# 【Linux、SSH】记录一次Linux服务器被人使用SSH字典爆破

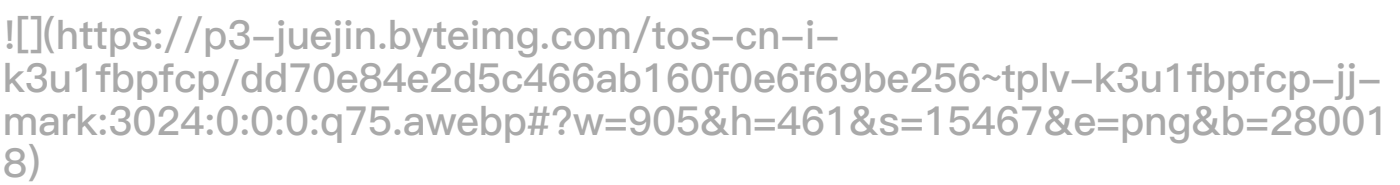
> 半年前的写的文章了，今天有了兴致发一下，记录一个运维小白服务器被攻击的经历。

2024年1.20凌晨睡了一觉，早上起来用termux远程ssh登录小主机发现密码没法登录了，一直显示密码错误，到了晚上用电脑ssh连接小主机，发现小主机真的没法登录了，直接把小主机接上屏幕查看，发现密码被人修改了，系统都进不去，而且主机风扇一直呼呼转，估计是cpu使用率上来了，

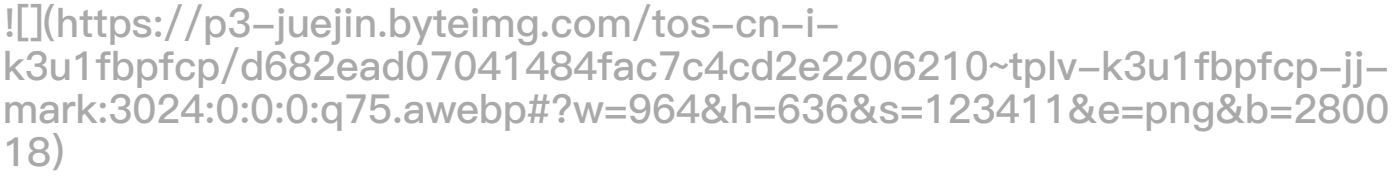
这下只能进到linux的安全模式进行密码修改了

重启ubuntu，开机的时候同时按住**shift+esc**键进入安全模式

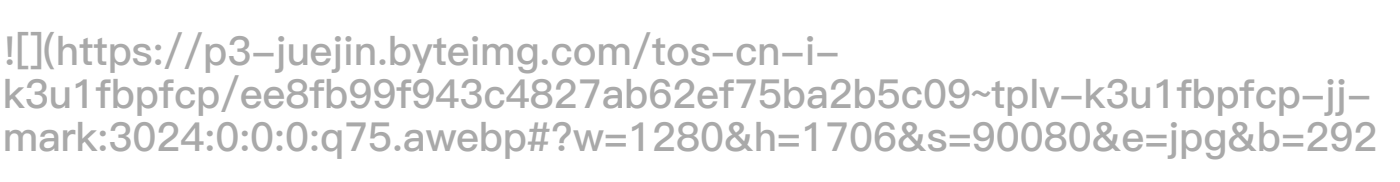
选择ubuntu高级选项然后enter进入



向下选择括号内有recovery mode的选项，然后按e进入编辑模式

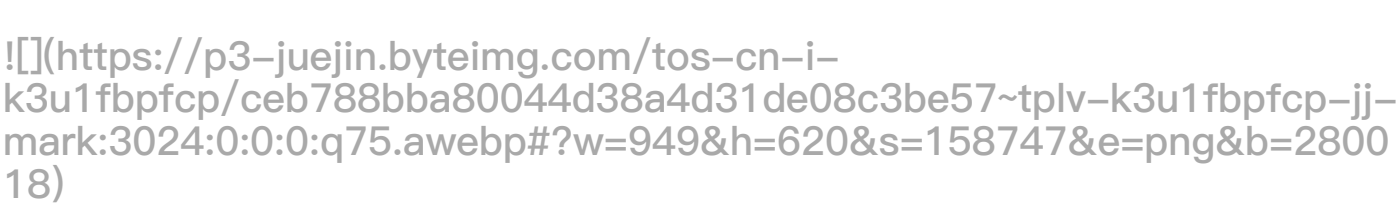


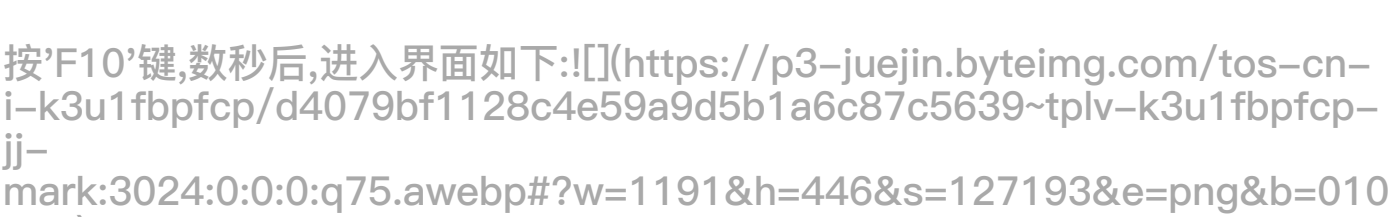
光标移动到前面单词为linux的那一行， 将那一行的ro recovery nomodeset 以及本行他后面的都删掉（比如图中的dis_ucode_ldr）



52b)

改为quiet splash rw init=/bin/bash,改完如下




按'F10'键,数秒后,进入界面如下: 


输入命令passwd 用户名 ,如下

```
这样的
：passwd 用户名
：密码
：确认密码
这样就好了，不加的话你的执行步骤是对的但用户密码有可能还是原来的
---
比如我用户名是 root，要修改的目标密码是123456。
输入passwd root
输入12345
确认输入12345
```

然后输入命令保存重启

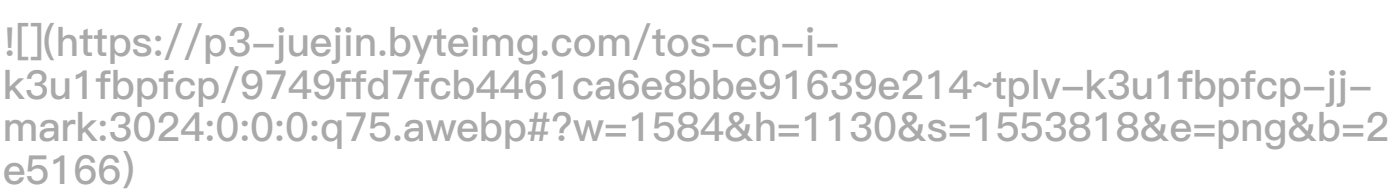
```
reboot -f #此处只输入reboot是没用的，必须加-f
```

重启之后就可以登录小主机了

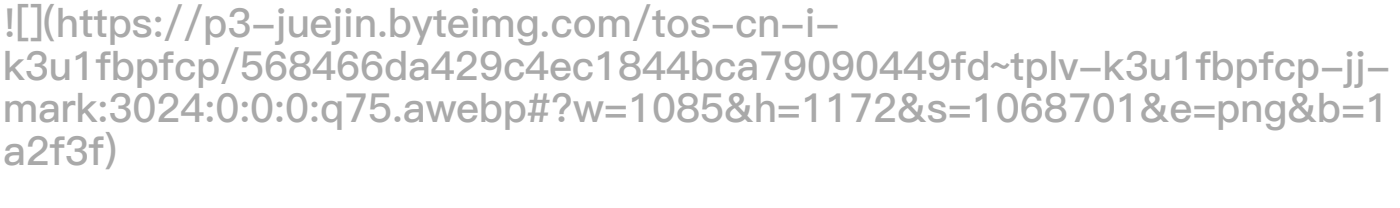
---

开机之后， 发现cpu一直居高不下， 开机几分钟cpu直顶99%

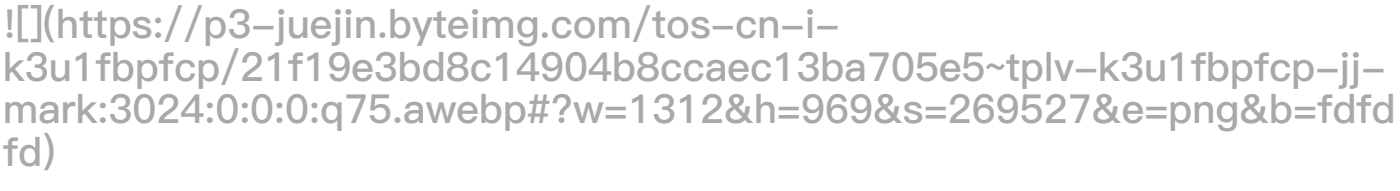
但是top命令查看了一下进程，发现没有进程占用过高，表现十分正常

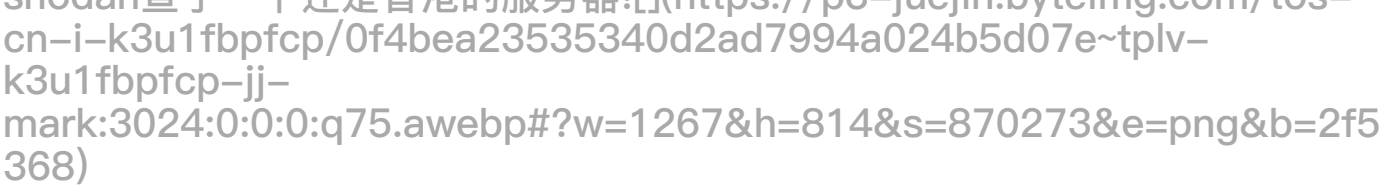


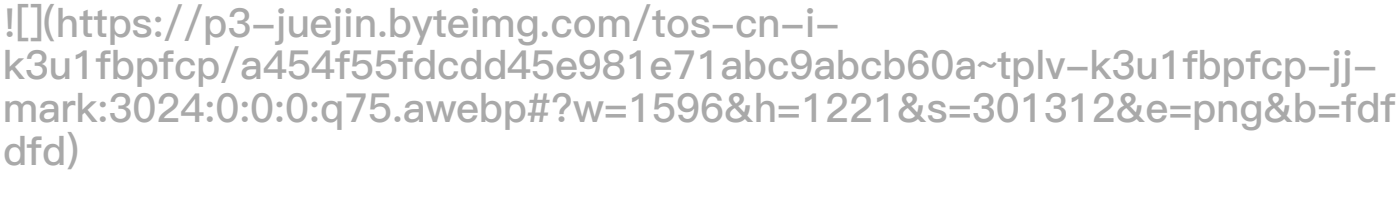
使用netstat -napt命令查看tcp进程服务，发现一个可疑的IP地址



查一下ip来源

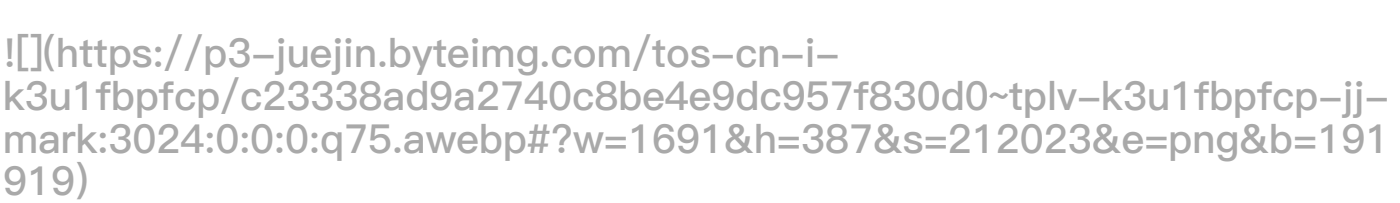


kill掉ntools进程之后第二次查看， 发现原来的可疑IP地址变了，但是用shodan查了一下还是香港的服务器 



使用shodan查询ip地址的方法：

[www.shodan.io/](http://cxyroad.com/ "https://www.shodan.io/")



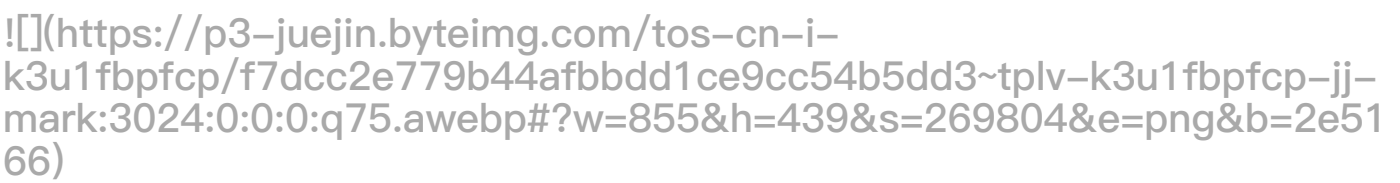
使用sysdig查看隐藏进程

```
sysdig -c topprocs_cpu # 该命令可以输出cpu占用的排行，经测试可以显示出被隐藏的进程
```

sysdig安装教程
[www.yundongfang.com/Yun6123.htm...](http://cxyroad.com/ "https://www.yundongfang.com/Yun6123.html")

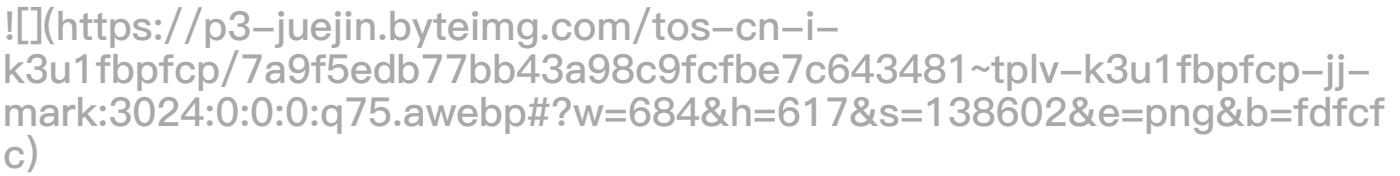
安装sysdig排错教程：[zhuanlan.zhihu.com/p/112788242](http://cxyroad.com/ "https://zhuanlan.zhihu.com/p/112788242")

使用sysdig查看到的隐藏进程：



有一个隐藏的进程占用率看起来十分离谱

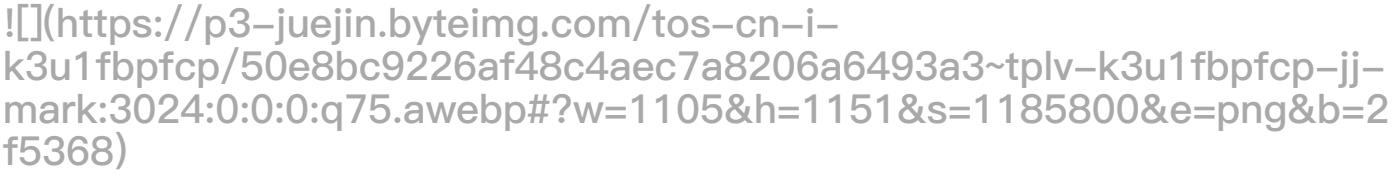
百度了一下这个工具，果然是挖矿软件



使用命令搜索隐藏进程的更多信息

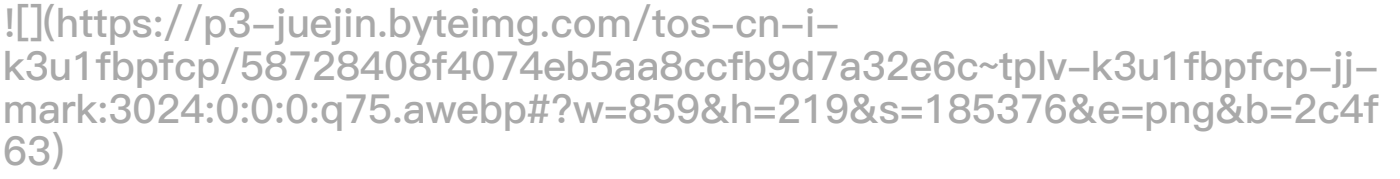
```
unhide proc # 自助搜索隐藏进程，linux系统中一切皆文件，proc目录下保存的就是所有正在运行程序的进程ID，即PID
```

发现以下结果， 全都是这个tmp/ntools， 跟百度上的石锤了



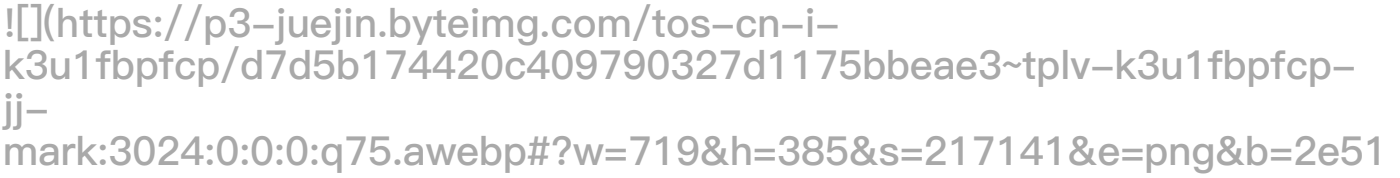
先把他kill掉看看，不过感觉是有定时任务的

直接kill ！ ！ ！ ！



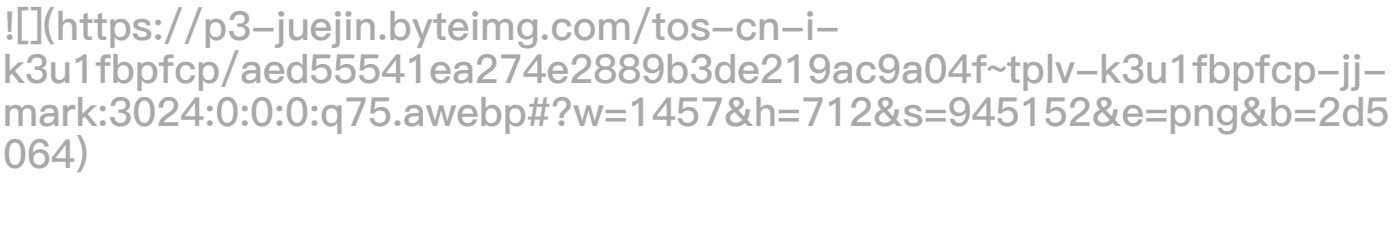
kill掉了之后发现机子风扇变慢了， 但是过了一会又转了

没错他又启动了！那么可以判断这一定是定时任务了!



66)

挖矿程序一般都设置了定时任务启动脚本程序，查看定时任务，crontab -l查看是找不到的。得看/etc/crontab文件，执行 cat /etc/crontab 。果然有任务在启动程序脚本 /var/log/oneav/cron.lock




不过后来查了一下这个是宝塔面板的定时任务, 初步判断应该不是这个原因引起的

---

查看ssh是否被暴力破解

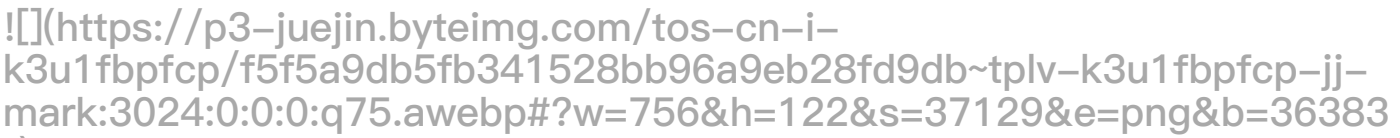
```
find /var/log -name 'secure*' -type f | while read line;do awk
'/Failed/{print $(NF-3)}' $line;done | awk '{a[$0]++}END{for (j in a) if(a[j]
> 20) print j"="a[j]}' | sort -n -t'=' -k 2
```

---

查看防火墙状态

```
sudo ufw status
```

发现防火墙是不活动的



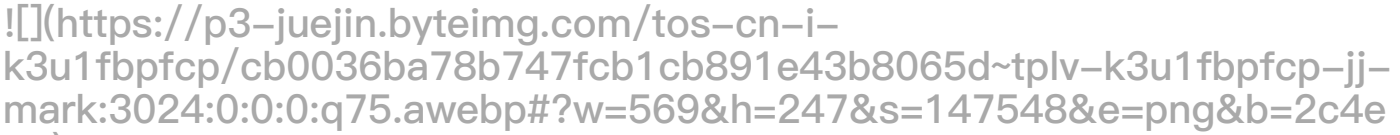

将防火墙开启

```
sudo ufw enable
sudo systemctl start ufw # 并且设置开机自启动 或者用sudo service ufw
start
```

同时修改防火墙配置，确保万无一失

```
sudo vim /etc/ufw/ufw.conf
```

将防火墙的enabled改为yes

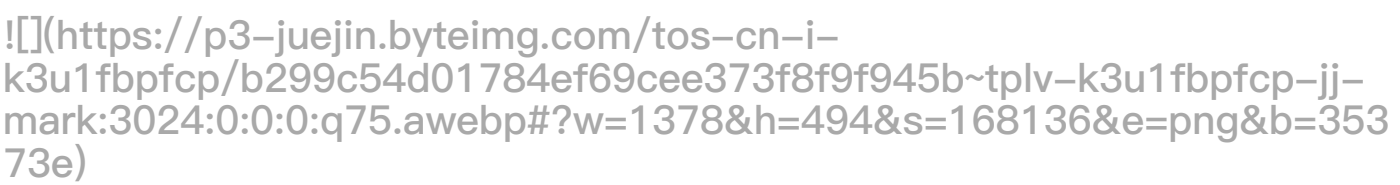



但是过了十秒钟左右配置文件中的ENABLED又被改成no了

并且sudo ufw status查看防火墙状态也是未激活状态

如下

k3u1fbpfcp/f525dd95a76742c5962ccc28895638d0~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=756&h=122&s=37129&e=png&b=36383f)



不管怎么改他都能将防火墙关闭

于是我想到了一个方法：

在修改防火墙之后直接将防火墙的配置文件/etc/ufw/ufw.conf修改权限为“只读”

在修改防火墙的配置文件后随即修改权限为“只读”

```
chmod 400 /etc/ufw/ufw.conf
```

* **4（读权限）：** 表示对文件或目录有读权限。
* **2（写权限）：** 表示对文件或目录有写权限。
* **1（执行权限）：** 对于文件，表示可以执行；对于目录，表示可以进入。

**数字1（第一个数字）：所有者权限。**

**数字2（第二个数字）：组权限。**

**数字3（第三个数字）：其他用户权限。**

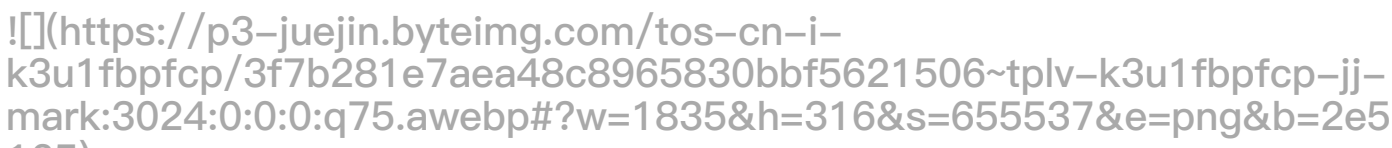
---

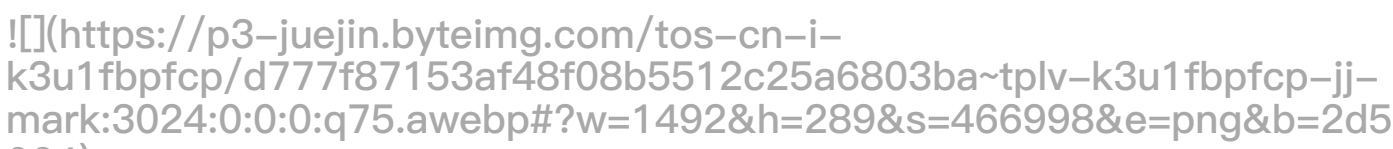
这样就可以有效阻止攻击者的程序修改防火墙的配置文件

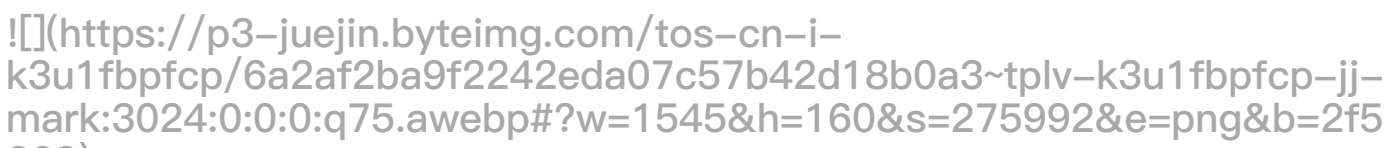
接着我们查看系统的日志

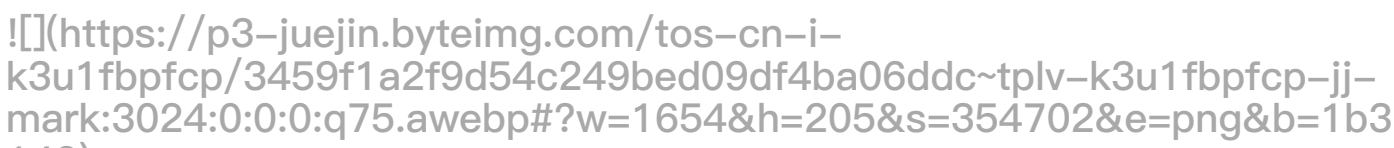
```
journalctl -e
```

一条一条日志进行检查









通过查看日志发现有一个unix系统（也是linux）似乎在与我们的主机建立会话

蓝色框内似乎是在执行一个命令行

绿色框可以知道这个pam_unix尝试使用root用户与我们的主机建立会话但是被关闭了

**连续对比了几次日志,发现这个cmd命令输入很稳定的一分钟执行一次, 而且有一次是用名称为smmsp的user发送邮件测试, 邮件程序被放在了以下三个文件夹里面, 猜测攻击者做了备用的邮件发送系统, 并且做了软链接**

**/etc/init.d/sendmail**

**/usr/share/sendmail/sendmail**

**/usr/libexec/sendmail/sendmail**

**经百度发现这个邮件系统其实是linux自带的**

---

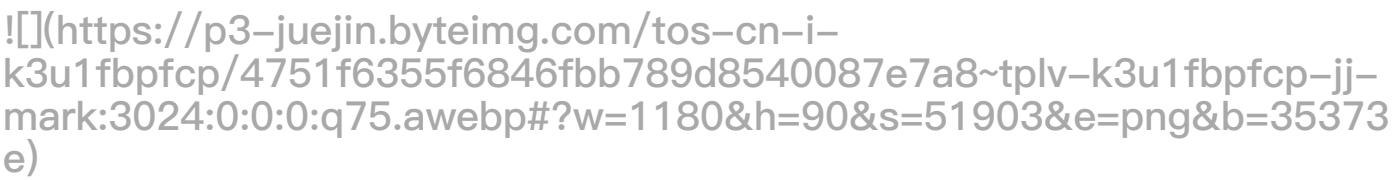
那么我们接着往下探

想到攻击者想要访问我们的服务器进行命令行输入，那应该是用到了ssh进行远程登录的

我们看一下ssh的日志， 查询ssh的日志

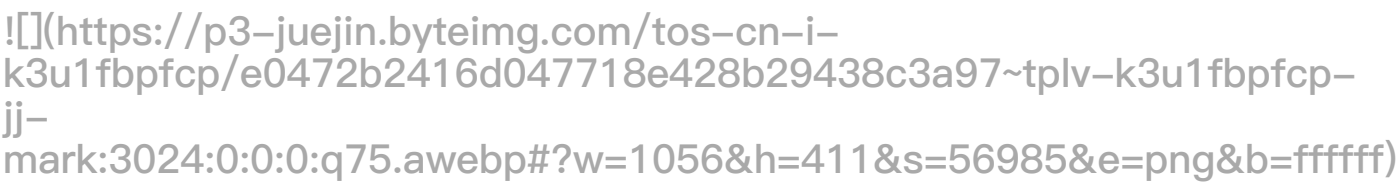
```
journalctl -u ssh -e #参数-e是直接将日志输出全部到底
```

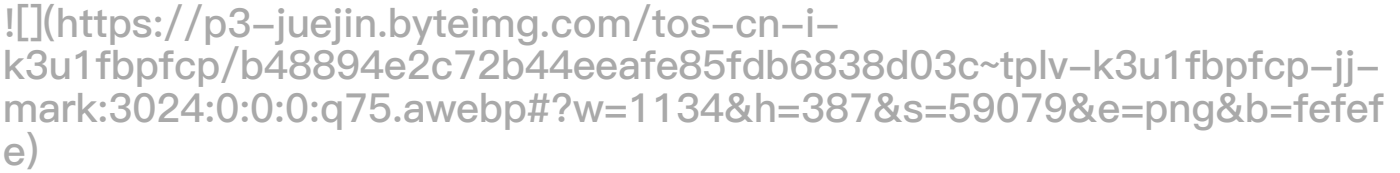
mark:3024:0:0:0:q75.awebp#?w=1885&h=1090&s=2156806&e=png&b=1a2e3e)




好家伙，我们发现上面一台unix机器一直在尝试用不同的user登录我们的机子，而且每次使用的IP地址都是不一样的，猜测这位攻击者在使用字典进行ssh爆破攻击

查了一下这些IP地址，都是国外的，联系到上面的几次可疑IP地址的归属地都不是大陆的，可以确定这就是攻击者使用梯子匿名攻击我们的服务器



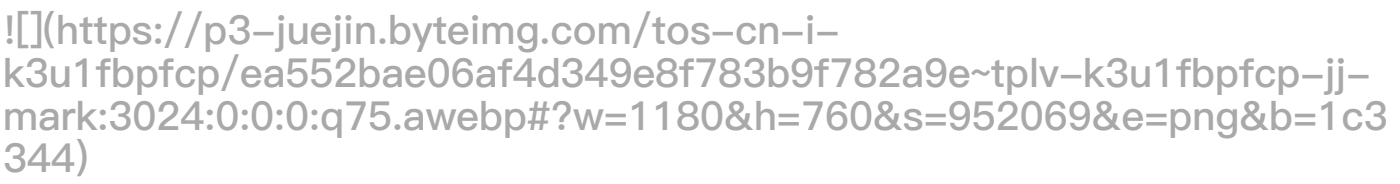

还好刚刚把防火墙立起来了而且加了只读权限,才重新把防火墙立起来，现在攻击者被拦截在外面无法通过ssh进行登录我们的机子

---

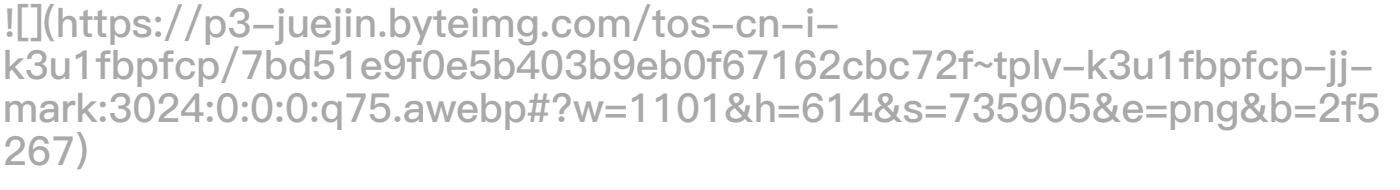
查看系统邮件是否有对外订阅

```
cat var/mail/root
```

发现有几条可疑的邮件



发现邮件通知了防火墙自动禁用的通知而且还有一个可疑的/bin/dtjbdjkw 1 1



前面有个单词cron(定时任务的全称是crontab)那么初步猜测这东西可能跟定时任务有关，查看定时任务的配置文件，看到都是很常规的定时任务，那我们直接进入到系统的这四个定时任务的文件中去查看是否有可疑文件:

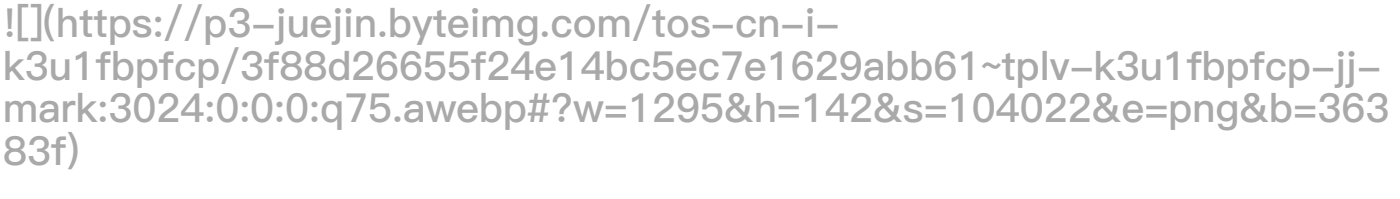
/etc/cron.hourly

/etc/cron.daily

/etc/cron.weekly

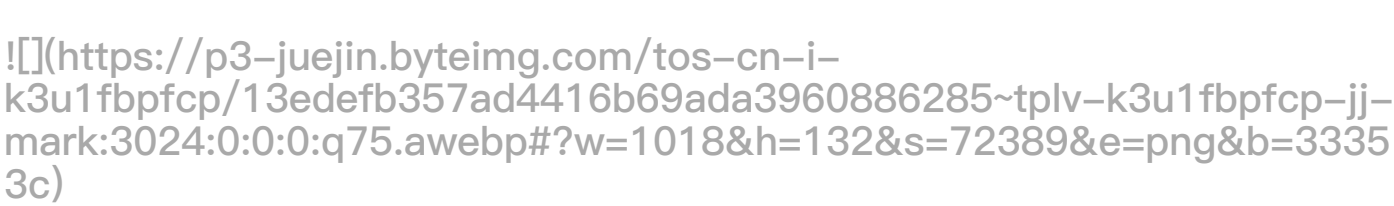
/etc/cron.monthly

这四个文件夹就是系统专门存放定时任务的地方, 分别查看这四个文件夹内部的异样



观察这几个定时任务的执行时间和 **journalctl -e 日志中异常脚本的启动时间间隔是否相似可以进行判断 相关程度**

接着查看后发现，有一个定时任务的文件名字很可疑，使用ls -al可以看到创建时间就是1.20号凌晨的时候

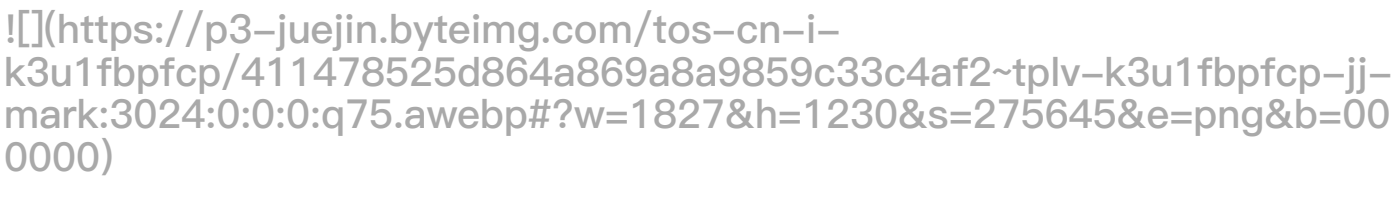



观察了三个定时任务目录后可以确认这个**Wsqsg3Wo**就是一个可疑文件, 我们直接**rm -rf Wsqsg3Wo** 将其删掉, 同时使用 **kill -9 pid** 结束掉ntools进程, 然后开启新的终端观察系统日志

```
journalctl -e
```

最终cpu不再处于高占用状态, 并且攻击者包ntools不再再生 但是还是能看到攻击者在不断地进行ssh字典爆破, 真是穷追不舍啊,

又过了几天之后使用**journalctl -e**查看日志




发现清除了定时任务后还有个自启动任务在以间隔一分钟的频率执行,但是因为我把它的脚本(定时任务)给移除了, 所以他每次执行的时候都找不到脚本, 就以执行失败告终

那么我们现在cd 到自启动目录下查看这个日志中出现的 **DJI2YwVz.service**

```
cd /lib/systemd/system/
```

cat DJI2YwVz.service

```

**DJI2YwVz.service** 的内容如下

这些配置的含义就是(由gpt进行解释)

* **[Service]** ： 这个部分定义了服务的启动行为。

* + **Type=simple**：表示这是一个简单的服务类型，它意味着主进程由 **ExecStart** \*\*\*\*指定的命令启动，并且当该命令退出时，整个服务被视为失败。但由于 \*\*\*\*\*RemainAfterExit=yes**，服务的状态将保持活动，即使 **ExecStart** \*\*\*\*的进程已经退出。
+ **ExecStart=/bin/dtjbdJkw**：这是启动服务时要运行的命令。这里指定了一个路径 **/bin/dtjbdJkw**，它应该是一个可执行文件。不过，这个文件名看起来有些不常见，可能是某个特定应用或脚本。
+ **RemainAfterExit=yes**：即使 \*\*\*\*\*ExecStart** \*\*\*\*指定的进程退出，服务也将保持活动状态。这通常与 **Type=oneshot** 一起使用，但在这里与 Type=simple 一起使用有些不寻常。
+ **Restart=always**：如果服务退出（无论是由于失败还是正常退出），它总是会被重新启动。
+ **RestartSec=60s**：在尝试重新启动服务之前，系统会等待60秒。

```
[Install]
WantedBy=multi-user.target

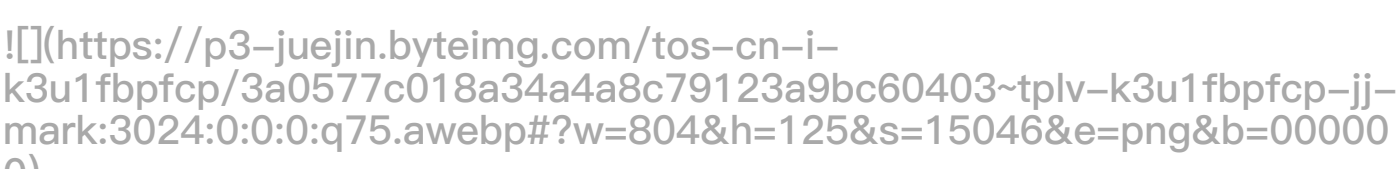
```

* **[Install]** ： 这个部分提供了服务的安装信息。

* + **WantedBy=multi-user.target**：表示当系统达到 **multi-user.target** 运行级别时，这个服务应该被启动。**multi-user.target** \*\*\*\*是一个标准的运行级别，表示系统已经准备好供多个用户登录和使用。
```

总的来说，这个服务单元文件定义了一个简单的服务，该服务依赖于网络，并在网络启动后运行 **/bin/dtjbdJkw** ****命令。如果命令退出，服务会尝试每隔60秒重新启动它。这个服务被配置为在系统达到多用户模式 **(表示系统已经完成基本初始化，并且允许多个用户同时登录和访问系统资源)** 时自动启动。

现在我们使用locate定位这个服务的位置

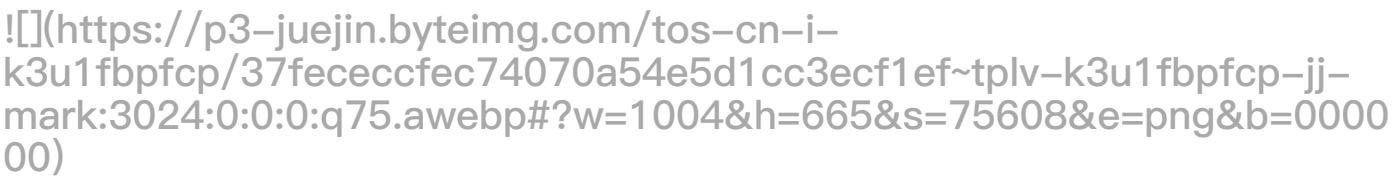



分别cd到这两个目录下去将这两个服务删除

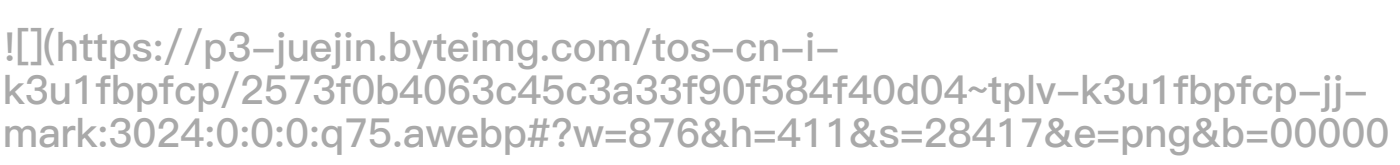
```
sudo rm DJI2YwVz.service
```

**查看系统自启动中的所有开启的服务**

```
systemctl list-unit-files --type=service | grep enabled
```




我们发现了一个可疑的包, cat命令查看一下



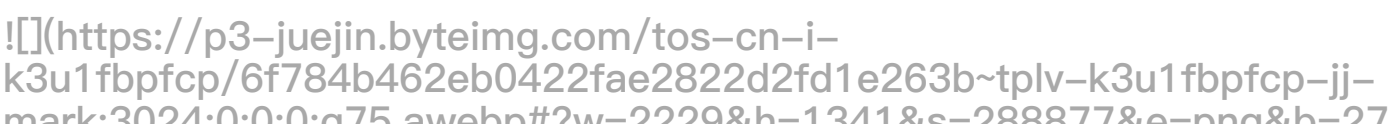
0)

果然还是这个病毒的启动文件, 使用rm删除它

```
cd /etc/systemd/system/
sudo rm DJI2YwVz.service
systemctl status YaljzjZB.service #再次查看是否删除成功
```

成功删除, 至此, 机子已经处于安全稳定的状态了, 日志中也只有攻击者ssh登录失败的信息

写一个python脚本来查看一下这七天内被可疑ip登录的ip信息和登陆次数(登录失败的),统计总共的攻击次数



可以看到, 这是过滤掉了本机的IP地址以及内网 192.168 开头的IP地址后的统计结果,这七天时间,总共有**100个**左右的IP地址尝试通过ssh登录主机, 总共登录次数为**15584次,** 这些ip归属地分布世界各地, 猜测使用了ip池子进行伪装.

**曾经我以为互联网到至今应该是很和平的状态, 但是经历了这次ssh字典爆破攻击后我才意识到网络攻击无处不在, 建议系统密码使用比较复杂的随机字符组合,七八十位都没问题, 数据可贵, 电脑该装杀毒软件的就装上, 别因为那占用那点内存而舍弃杀毒软件, 防网络攻击于未然 !**

原文链接: https://juejin.cn/post/7385410225829740544