

黑客入侵应急分析手工排查

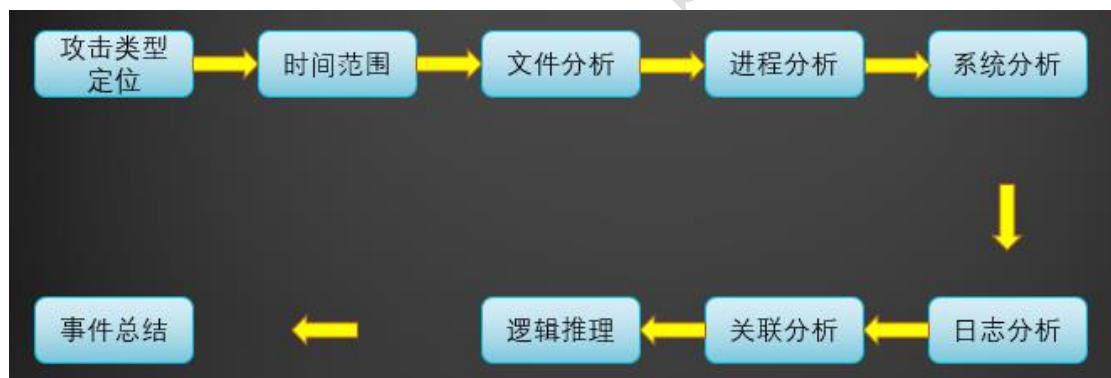
1 事件分类

常见的安全事件：

1. Web 入侵：挂马、篡改、Webshell
2. 系统入侵：系统异常、RDP 爆破、SSH 爆破、主机漏洞
3. 病毒木马：远控、后门、勒索软件
4. 信息泄漏：拖裤、数据库登录（弱口令）
5. 网络流量：频繁发包、批量请求、DDOS 攻击

2 排查思路

一个常规的入侵事件后的系统排查思路：



1. 文件分析
 - a) 文件日期、新增文件、可疑/异常文件、最近使用文件、浏览器下载文件
 - b) Webshell 排查与分析
 - c) 核心应用关联目录文件分析
2. 进程分析
 - a) 当前活动进程 & 远程连接
 - b) 启动进程&计划任务
 - c) 进程工具分析
 - i. Windows:Pchunter
 - ii. Linux: Chkrootkit&Rkhunter

3. 系统信息

- a) 环境变量
- b) 帐号信息
- c) History
- d) 系统配置文件

4. 日志分析

- a) 操作系统日志
 - i. Windows: 事件查看器 (eventvwr)
 - ii. Linux: /var/log/
- b) 应用日志分析
 - i. Access.log
 - ii. Error.log

3 分析排查

3.1 Linux 系列分析排查

3.1.1 文件分析

1. 敏感目录的文件分析 (类/tmp 目录, 命令目录/usr/bin /usr/sbin)

例如:

查看 tmp 目录下的文件: `ls -alt /tmp/`

查看开机启动项内容: `ls -alt /etc/init.d/`

查看指定目录下文件时间的排序: `ls -alt | head -n 10`

针对可疑文件可以使用 `stat` 进行创建修改时间、访问时间的详细查看, 若修改时间距离事件日期接近, 有线性关联, 说明可能被篡改或者其他。

PS: 若黑客通过 `touch -r` 修改了文件的日期, 会增加时间界定难度。

```

root@sm0nk:~# stat /usr/bin/lsof
  File: '/usr/bin/lsof'
  Size: 163136      Blocks: 320        IO Block: 4096   regular file
Device: 801h/2049d Inode: 132360     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/         root)   Gid: (    0/         root)
Access: 2017-04-06 11:30:30.744100682 +0800
Modify: 2015-09-14 05:46:13.000000000 +0800
Change: 2016-09-07 20:28:18.073346621 +0800
 Birth: -

```

2. 新增文件分析

例如要查找 24 小时内被修改的 JSP 文件: `find ./ -mtime 0 -name "*.jsp"`

(最后一次修改发生在距离当前时间 $n \times 24$ 小时至 $(n+1) \times 24$ 小时)

查找 72 小时内新增的文件 `find / -ctime -2`

PS: `-ctime` 内容未改变权限改变时候也可以查出

根据确定时间去反推变更的文件

`ls -al /tmp | grep "Feb 27"`

3. 特殊权限的文件

查找 777 的权限的文件 `find / *.*.jsp -perm 4777`

4. 隐藏的文件 (以 "." 开头的具有隐藏属性的文件)

5. 在文件分析过程中, 手工排查频率较高的命令是 `find grep ls` 核心目的是为了关联推理出可疑文件。

3.1.2 进程命令

1. 使用 netstat 网络连接命令, 分析可疑端口、可疑 IP、可疑 PID 及程序进程

`netstat -antlp | more`

```

root@sm0nk:~# netstat -antlp | more
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      2177/gnome-session
tcp6       0      0 :::22                  :::*                   LISTEN      2177/gnome-session
udp        0      0 0.0.0.0:68              0.0.0.0:*               LISTEN      11295/NetworkManager
raw6       0      0 0.0.0.0:58              0.0.0.0:*               LISTEN      11295/NetworkManager
Active UNIX domain sockets (servers and established)
Proto RefCnt Flags       Type       State       I-Node  PID/Program name      Path
unix    2      [ ACC ]     STREAM    LISTENING   20477   11295/NetworkManager  /tmp/.ICE-unix/2177
unix    2      [ ACC ]     STREAM    LISTENING   15622   11295/NetworkManager  /tmp/.ICE-unix/2177
unix    3      [ ]       DGRAM          11272   1/init                 /run/systemd/notify
unix    2      [ ]       DGRAM          11274   1/init                 /run/systemd/cgroups-agent
unix    2      [ ACC ]     Pictu STREAM    LISTENING   11276   1/init                 /run/systemd/private
unix    2      [ ACC ]     SEQPACKET LISTENING   11281   1/init                 /run/udev/control
unix    2      [ ACC ]     Video  STREAM    LISTENING   22323   631/gdm3               @/tmp/dbus-7NyFWPXE
unix    2      [ ACC ]     STREAM    LISTENING   11295   1/init                 /run/systemd/journal/stdout
unix    2      [ ACC ]     STREAM    LISTENING   20412   2171/Xorg               @/tmp/.X11-unix/X0
unix    7      [ ]       DGRAM          11298   1/init                 /run/systemd/journal/socket
unix   15      [ ]       DGRAM          11303   1/init                 /run/systemd/journal/dev-log
unix    2      [ ACC ]     Floppy STREAM    LISTENING   11307   1/init                 /run/lvm/lvmetad.socket
unix    2      [ ACC ]     STREAM    LISTENING   11313   1/init                 /run/lvm/lvmpolld.socket
unix    2      [ ACC ]     STREAM    LISTENING   14625   662/gnome-session-b    /tmp/.ICE-unix/662

```

2. 使用 ps 命令，分析进程

ps aux | grep pid | grep -v grep

```
root@sm0nk:~# ps aux | grep 2624
root 2624 0.0 0.2 67812 5568 ? Ss 10:44 0:00 /usr/sbin/sshd -D
root 5364 0.0 0.0 12728 904 pts/1 S+ 19:10 0:00 grep 2624
root@sm0nk:~# ps aux | grep 2624 | grep -v grep
root 2624 0.0 0.2 67812 5568 ? Ss 10:44 0:00 /usr/sbin/sshd -D
root@sm0nk:~#
```

将 netstat 与 ps 结合，可参考 vinc 牛的案例：

```
[root@i-9kp9tipm tmp]# netstat -antlp
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	1273/sshd
tcp	0	0	127.0.0.1:25	0.0.0.0:*	LISTEN	1655/master
tcp	0	64	192.168.100.3:22	218.247.17.100:29737	ESTABLISHED	2007/sshd
tcp	0	0	192.168.100.3:22	124.207.112.10:54772	ESTABLISHED	1919/sshd
tcp	0	1	192.168.100.3:35806	43.241.157.58:6001	SYN_SENT	1742/getty
tcp	0	1	192.168.100.3:48358	211.149.149.191:45693	SYN_SENT	1677/abcfg
tcp	0	1	192.168.100.3:47268	61.147.73.76:1233	SYN_SENT	1683/VI
tcp	0	0	:::3306	:::*	LISTEN	1508/mysqld
tcp	0	0	:::22	:::*	LISTEN	1273/sshd
tcp	0	0	:::1:25	:::*	LISTEN	1655/master

发现了3个可以进程1742、1677、1683

看一下这些可执行程序在什么地方

```
[root@i-9kp9tipm tmp]# ps aux | grep 1677
root 1677 0.0 0.0 93636 892 ? Ssl 10:05 0:01 /usr/local/tomcat/abcfg
[root@i-9kp9tipm tmp]# ps aux | grep 1683
root 1683 0.0 0.0 73088 824 ? Ssl 10:05 0:01 /usr/local/tomcat/VI
[root@i-9kp9tipm init.d]# ps aux | grep 1742 | grep -v grep
```

（可以使用 lsof -i:1677 查看指定端口对应的程序）

3. 使用 ls 以及 stat 查看系统命令是否被替换。

两种思路：第一种查看命令目录最近的时间排序，第二种根据确定时间去匹配。

ls -alt /usr/bin | head -10

ls -al /bin /usr/bin /usr/sbin/ /sbin/ | grep "Jan 15"

```
root@sm0nk:~# ls -alt /usr/bin | head -10
total 436584
drwxr-xr-x 2 root root          90112 Jan 15 19:05 .
lrwxrwxrwx 1 root root           37 Jan 15 19:05 vmware-hgfsclient -> /usr/lib/vmware-tools/bin64/apploader
-rwxr-xr-x 1 root root       410536 Jan 15 19:04 vmware-config-tools.pl
-rwxr-xr-x 1 root root         9201 Jan 15 19:04 vm-support
-rwxr-xr-x 1 root root       196237 Jan 15 19:04 vmware-uninstall-tools.pl
-rwxr-xr-x 1 root root        22856 Dec 12 04:03 funzip
-rwxr-xr-x 2 root root       170920 Dec 12 04:03 unzip
-rwxr-xr-x 1 root root       76536 Dec 12 04:03 unzipsfx
-rwxr-xr-x 1 root root         2953 Dec 12 04:03 zipgrep
root@sm0nk:~# ls -al /bin /usr/bin /usr/sbin /sbin | grep "Jan 15"
drwxr-xr-x 2 root root         4096 Jan 15 19:04 .
-r-xr-xr-x 1 root root       63440 Jan 15 19:04 mount.vmhgfs
drwxr-xr-x 2 root root          90112 Jan 15 19:05 .
-rwxr-xr-x 1 root root         9201 Jan 15 19:04 vm-support
-rwxr-xr-x 1 root root       410536 Jan 15 19:04 vmware-config-tools.pl
lrwxrwxrwx 1 root root           37 Jan 15 19:05 vmware-hgfsclient -> /usr/lib/vmware-tools/bin64/apploader
-rwxr-xr-x 1 root root       196237 Jan 15 19:04 vmware-uninstall-tools.pl
```

PS: 如果日期数字<10, 中间需要两个空格。比如1月1日, grep "Jan 1"

4. 隐藏进程查看

```
ps -ef | awk '{print}' | sort -n | uniq >1
ls /proc | sort -n | uniq >2
diff 1 2
```

3.1.3 系统信息

```
history (cat /root/.bash_history)

/etc/passwd

crontab /etc/cron*

rc.local /etc/init.d chkconfig

last

$PATH

strings
```

1. 查看分析 history (cat /root/.bash_history), 曾经的命令操作痕迹, 以便进一步排查溯源。

运气好有可能通过记录关联到如下信息:

- a) wget 远程某主机 (域名&IP) 的远控文件;
- b) 尝试连接内网某主机 (ssh scp), 便于分析攻击者意图;
- c) 打包某敏感数据或代码, tar zip 类命令
- d) 对系统进行配置, 包括命令修改、远控木马类, 可找到攻击者关联信息...

2. 查看分析用户相关分析

- a) useradd userdel 的命令时间变化 (stat), 以及是否包含可疑信息

b) `cat /etc/passwd` 分析可疑帐号，可登录帐号

查看 UID 为 0 的帐号：`awk -F: '{if($3==0)print $1}' /etc/passwd`

查看能够登录的帐号：`cat /etc/passwd | grep -E "/bin/bash$"`

PS: UID 为 0 的帐号也不一定是可疑帐号，Freebsd 默认存在 `toor` 帐号，且 uid 为 0。

(`toor` 在 BSD 官网解释为 `root` 替代帐号，属于可信帐号)

```
root@sm0nk:~# awk -F: '{if($3==0)print $1}' /etc/passwd
root
sm0nk
root@sm0nk:~# cat /etc/passwd | grep -E "/bin/bash$"
root:x:0:0:root:/root:/bin/bash
postgres:x:116:119:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
couchdb:x:124:129:CouchDB Administrator,,,:/var/lib/couchdb:/bin/bash
sm0nk:x:0:0::/home/sm0nk:/bin/bash
root@sm0nk:~# 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
```

3. 查看分析任务计划

a) 通过 `crontab -l` 查看当前的任务计划有哪些，是否有后门木马程序启动相关信息；

b) 查看 `etc` 目录任务计划相关文件，`ls /etc/cron*`

4. 查看 linux 开机启动程序

a) 查看 `rc.local` 文件 (`/etc/init.d/rc.local` `/etc/rc.local`)

b) `ls -alt /etc/init.d/`

c) `chkconfig`

5. 查看系统用户登录信息

a) 使用 `lastlog` 命令，系统中所有用户最近一次登录信息。

b) 使用 `lastb` 命令，用于显示用户错误的登录列表

c) 使用 `last` 命令，用于显示用户最近登录信息(数据源为 `/var/log/wtmp`, `/var/log/btmp`)

`utmp` 文件中保存的是当前正在本系统中的用户的信息。

wtmp 文件中保存的是登录过本系统的用户的信息。

/var/log/wtmp 文件结构和 /var/run/utmp 文件结构一样，都是引用 /usr/include/bits/utmp.h 中的 struct utmp

```
root@sm0nkali:~# last -f /var/run/utmp
root    pts/0      192.168.27.3    Tue Apr 25 02:21    still logged in
reboot  system boot  4.3.0-kali1-amd64 Thu Feb 16 01:19    still running

utmp begins Thu Feb 16 01:19:49 2017
root@sm0nkali:~# last -f /var/log/btmp
root    ssh:notty   192.168.27.3    Fri Apr 14 02:33    gone - no logout
root    ssh:notty   192.168.27.3    Fri Apr 14 02:33    - 02:33 (00:00)

btmp begins Fri Apr 14 02:33:13 2017
root@sm0nkali:~# last -a
root    pts/0      Tue Apr 25 02:21    still logged in      192.168.27.3
root    pts/0      Mon Apr 24 21:20 - 01:20 (03:59)      192.168.27.3
root    pts/0      Thu Apr 20 01:43 - 04:11 (02:27)      192.168.27.3
root    pts/0      Wed Apr 19 21:35 - 00:51 (03:15)      192.168.27.3
root    pts/0      Wed Apr 19 02:10 - 04:36 (02:26)      192.168.27.3
root    pts/2      Fri Apr 14 02:42 - 03:06 (00:24)      192.168.27.3
root    pts/1      Fri Apr 14 02:33 - 03:06 (00:33)      192.168.27.3
root    pts/0      Fri Apr 14 01:25 - 03:34 (02:09)      192.168.27.3
root    pts/1      Tue Apr 11 05:17 - 08:45 (03:27)      192.168.27.3
root    pts/0      Mon Apr 10 23:02 - 08:45 (09:42)      192.168.27.3
sm0nk  pts/1      Wed Apr 5 22:45 - 22:51 (00:05)      192.168.27.3
sm0nk  pts/1      Wed Apr 5 22:44 - 22:45 (00:00)      192.168.27.3
root    pts/0      Wed Apr 5 22:44 - 23:21 (00:37)      192.168.27.3

wtmp begins Wed Apr 5 22:44:09 2017
root@sm0nkali:~# lastb
```

6. 系统路径分析

- a) echo \$PATH 分析有无敏感可疑信息

7. 指定信息检索

- a) strings 命令在对象文件或二进制文件中查找可打印的字符串
- b) 分析 sshd 文件, 是否包括 IP 信息 strings /usr/bin/.sshd | egrep '[1-9]{1,3}\.[1-9]{1,3}\.'

PS: 此正则不严谨, 但匹配 IP 已够用

- c) 根据关键字匹配命令内是否包含信息 (如 IP 地址、时间信息、远控信息、木马特征、代号名称)

8. 查看 ssh 相关目录有无可疑的公钥存在。

- a) Redis (6379) 未授权恶意入侵, 即可直接通过 redis 到目标主机导入公钥。
- b) 目录: /etc/ssh ./.ssh/

3.1.4 后门排查

除以上文件、进程、系统 分析外, 推荐工具 chkrootkit rkhunter

www.chkrootkit.org rkhunter.sourceforge.net

➤ chkrootkit

(迭代更新了 20 年)主要功能:

1. 检测是否被植入后门、木马、rootkit
2. 检测系统命令是否正常
3. 检测登录日志
4. 详细参考 README

```
root@sm0nk:~/Desktop/PenTest/chkrootkit-0.52# ./chkrootkit -h
Usage: ./chkrootkit [options] [test ...]
Options:
  -h          show this help and exit
  -V          show version information and exit
  -l          show available tests and exit
  -d          debug
  -q          quiet mode
  -x          expert mode
  -r dir      use dir as the root directory
  -p dir1:dir2:dirN path for the external commands used by chkrootkit
  -n          skip NFS mounted dirs

root@sm0nk:~/Desktop/PenTest/chkrootkit-0.52# ./chkrootkit -l
./chkrootkit: tests: aliens asp bindshell lkm rexedcs sniffer w55808 wted scalper slapper z2 chkutmp OSX RSPLUG amd basenane biff chfn chsh cron cron
tab date du dirname echo egrep env find fingerd gpm grep hdparm su ifconfig inetd inetdconf identd init killall ldsopreload login ls lsof mail minge
tty netstat named passwd pidof pop2 pop3 ps pstree rpcinfo rlogind rshd slogin sendmail sshd syslogd tar tcpd tcpdump top telnetd timed traceroute vd
ir w write
```

➤ rkhunter 主要功能:

1. 系统命令 (Binary) 检测, 包括 Md5 校验
2. Rootkit 检测
3. 本机敏感目录、系统配置、服务及套间异常检测
4. 三方应用版本检测

```
root@sm0nk:~# rkhunter --checkall --sk
[ Rootkit Hunter version 1.4.2 ]

Checking system commands...

Performing 'strings' command checks
Checking 'strings' command [ OK ]

Performing 'shared libraries' checks
Checking for preloading variables [ None found ]
Checking for preloaded libraries [ None found ]
Checking LD_LIBRARY_PATH variable [ Not found ]

Performing file properties checks
Checking for prerequisites [ Warning ]
/usr/local/bin/rkhunter [ OK ]
/usr/sbin/adduser [ Warning ]
/usr/sbin/chroot [ OK ]
/usr/sbin/cron [ OK ]
/usr/sbin/groupadd [ OK ]
/usr/sbin/groupdel [ OK ]
/usr/sbin/groupmod [ OK ]
/usr/sbin/grpck [ OK ]
```

➤ 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:

- S MD5 校验码
- S 文件尺寸
- L 符号连接
- T 文件修改日期
- D 设备
- U 用户
- G 用户组
- M 模式 e (包括权限和文件类型)

借用 sobug 文章案例: 如下图可知 ps, pstree, netstat, sshd 等等系统关键进程被篡改了

The screenshot shows the output of the command `rpm -Va` in a terminal. The output lists various system files and their checksums. Several files are highlighted with red boxes and red arrows, indicating they have been modified. The highlighted files are:

- `/bin/netstat`
- `/usr/bin/ssh-keygen`
- `/usr/bin/scp`
- `/usr/bin/sftp`
- `/usr/bin/ssh`
- `/usr/bin/ssh-add`
- `/usr/bin/ssh-agent`
- `/usr/bin/ssh-keyscan`
- `/usr/bin/pidstat`
- `/usr/sbin/sshd`
- `/usr/bin/pstree`

➤ Webshell 查找

Webshell 的排查可以通过文件、流量、日志三种方式进行分析,基于文件的命名特征和内容特征,相对操作性较高,在入侵后应急过程中频率也比较高。

可根据 **webshell** 特征进行命令查找，简单的可使用(当然会存在漏报和误报)

```
find /var/www/ -name "*.php" |xargs egrep
'assert|phpspy|c99sh|milw0rm|eval|\\(gunerpress|\\(base64_decoolcode|spider_bc
|shell_exec|passthru|\\(\\$\\_POST\\[|eval
\\(str_rot13|\\.chr\\(|\\$\\{\\\"\\_P|eval\\(\\$\\_R|file_put_contents\\(\\.\\*\\$\\_|base64
_decode'
```

1. **Webshell** 的排查可以通过具备 **shell** 特征的关键函数进行过滤匹配；
2. **Github** 上存在各种版本的 **webshell** 查杀脚本，当然都有自己的特点，可使用河马 **shell** 查杀 (shellpub.com)

综上所述，通过 **chkrootkit**、**rkhunter**、**RPM check**、**Webshell Check** 等手段得出以下应对措施：

1. 根据进程、连接等信息关联的程序，查看木马活动信息。
2. 假如系统的命令（例如 **netstat ls** 等）被替换，为了进一步排查，需要下载一新的或者从其他未感染的主机拷贝新的命令。
3. 发现可疑可执行的木马文件，不要急于删除，先打包备份一份。
4. 发现可疑的文本木马文件，使用文本工具对其内容进行分析，包括回连 **IP** 地址、加密方式、关键字（以便扩大整个目录的文件特征提取）等。

3.1.5 日志分析

日志文件

/var/log/message	包括整体系统信息
/var/log/auth.log	包含系统授权信息，包括用户登录和使用的权限机制等
/var/log/userlog	记录所有等级用户信息的日志。
/var/log/cron	记录 crontab 命令是否被正确的执行
/var/log/xferlog(vsftpd.log)	记录 Linux FTP 日志
/var/log/lastlog	记录登录的用户，可以使用命令 lastlog 查看
/var/log/secure	记录大多数应用输入的账号与密码，登录成功与否

<code>var/log/wtmp</code>	记录登录系统成功的账户信息，等同于命令 <code>last</code>
<code>var/log/faillog</code>	记录登录系统不成功的账号信息，一般会被黑客删除

1. 日志查看分析，`grep,sed,sort,awk` 综合运用

2. 基于时间的日志管理：

`/var/log/wtmp`

`/var/run/utmp`

`/var/log/lastlog(lastlog)`

`/var/log/btmp(lastb)`

3. 登录日志可以关注 `Accepted`、`Failed password`、`invalid` 特殊关键字

4. 登录相关命令

lastlog 记录最近几次成功登录的事件和最后一次不成功的登录

who 命令查询 `utmp` 文件并报告当前登录的每个用户。**Who** 的缺省输出包括用户名、终端类型、登录日期及远程主机

w 命令查询 `utmp` 文件并显示当前系统中每个用户和它所运行的进程信息

users 用单独的一行打印出当前登录的用户，每个显示的用户名对应一个登录会话。如果一个用户有不止一个登录会话，那他的用户名把显示相同的次数

last 命令往回搜索 `wtmp` 来显示自从文件第一次创建以来登录过的用户

finger 命令用来查找并显示用户信息，系统管理员通过使用该命令可以知道某个时候到底有多少用户在使用这台 Linux 主机。

5. 几个语句

定位有多少 IP 在爆破主机的 root 帐号

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

登录成功的 IP 有哪些

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

`tail -400f demo.log` #监控最后 400 行日志文件的变化 等价与 `tail -n 400 -f`（-f 参数是实时）

`less demo.log` #查看日志文件，支持上下滚屏，查找功能

`uniq -c demo.log` #标记该行重复的数量，不重复值为 1

```
grep -c 'ERROR' demo.log    #输出文件 demo.log 中查找所有包行 ERROR 的行的数量
```

3.1.6 相关处置

kill -9

chattr -i

rm

setfacl

ssh

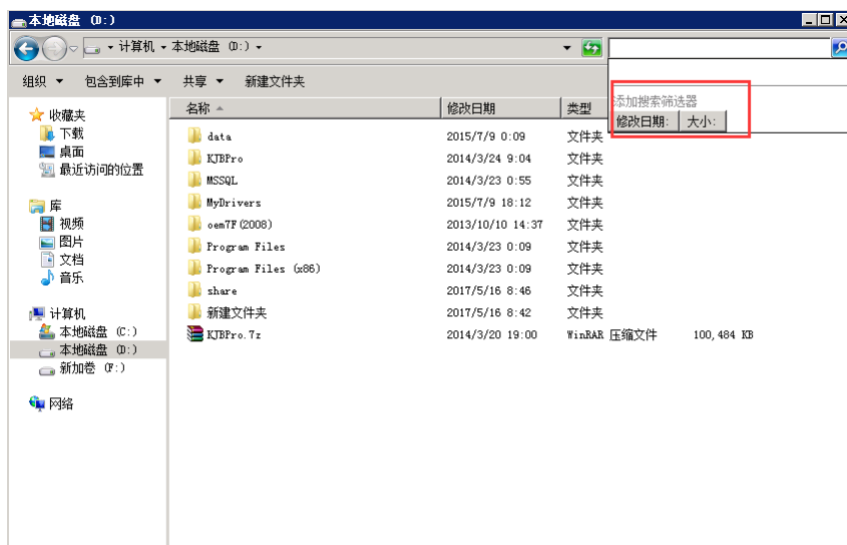
chmod

3.2 Windows 系列分析排查

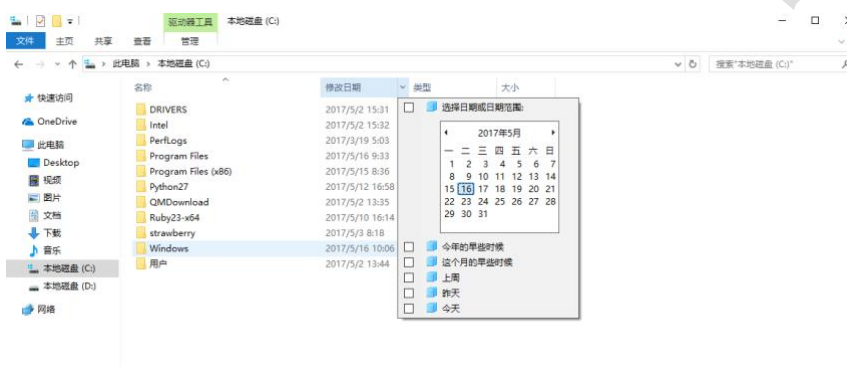
3.2.1 文件分析

1. 开机启动有无异常文件
2. 各个盘下的 temp(tmp)相关目录下查看有无异常文件
3. 浏览器浏览痕迹、浏览器下载文件、浏览器 cookie 信息
4. 查看文件时间，创建时间、修改时间、访问时间。对应 linux 的 ctime mtime atime，通过对文件右键属性即可看到详细的时间（也可以通过 `dir /tc 1.aspx` 来查看创建时间），黑客通过菜刀类工具改变的是修改时间。所以如果修改时间在创建时间之前明显是可疑文件。
5. 查看用户 recent 相关文件，通过分析最近打开分析可疑文件
 - a) C:\Documents and Settings\Administrator\Recent
 - b) C:\Documents and Settings\Default User\Recent
 - c) 开始,运行 %UserProfile%\Recent
6. 根据文件夹内文件列表时间进行排序，查找可疑文件。当然也可以搜索指定日期范围的文件及文件件

Server 2008 R2 系列



Win10 系列



7. 关键字匹配, 通过确定后的入侵时间, 以及 webshell 或 js 文件的关键字 (比如博彩类), 可以在 IIS 日志中进行过滤匹配, 比如经常使用:

知道是上传目录, 在 web log 中查看指定时间范围包括上传文件夹的访问请求

```
findstr /s /m /l "UploadFiles" *.log
```

某次博彩事件中的六合彩信息是 six.js

```
findstr /s /m /l "six.js" *.aspx
```

根据 shell 名关键字去搜索 D 盘 spy 相关的文件有哪些

```
for /r d:\ %i in (*spy*.aspx) do @echo %i
```

3.2.2 进程命令

1. netstat -ano 查看目前的网络连接, 定位可疑的 ESTABLISHED

2. 根据 netstat 定位出的 pid, 再通过 tasklist 命令进行进程定位

```
C:\Users\sm0nk>netstat -ano | findstr ESTABLISHED
TCP    127.0.0.1:443          127.0.0.1:12844      ESTABLISHED    5316
TCP    127.0.0.1:443          127.0.0.1:12868      ESTABLISHED    5316
TCP    127.0.0.1:443          127.0.0.1:12869      ESTABLISHED    5316
TCP    127.0.0.1:443          127.0.0.1:12870      ESTABLISHED    5316
TCP    127.0.0.1:1975        127.0.0.1:1976      ESTABLISHED     8
TCP    127.0.0.1:1976        127.0.0.1:1975      ESTABLISHED     8
TCP    127.0.0.1:2271        127.0.0.1:2272      ESTABLISHED    5316
TCP    127.0.0.1:2272        127.0.0.1:2271      ESTABLISHED    5316
TCP    127.0.0.1:12844        127.0.0.1:443        ESTABLISHED   12992
TCP    127.0.0.1:12845        127.0.0.1:12846      ESTABLISHED   12992
TCP    127.0.0.1:12846        127.0.0.1:12845      ESTABLISHED   12992
TCP    127.0.0.1:12868        127.0.0.1:443        ESTABLISHED   12992
TCP    127.0.0.1:12869        127.0.0.1:443        ESTABLISHED   12992
TCP    127.0.0.1:12870        127.0.0.1:443        ESTABLISHED   12992
TCP    192.168.1.102:2089     180.163.21.35:80      ESTABLISHED   1444
TCP    192.168.1.102:2465     192.168.3.141:445     ESTABLISHED     4
TCP    192.168.1.102:2492     192.168.3.143:22      ESTABLISHED   8548
TCP    192.168.1.102:6427     23.79.16.113:443      ESTABLISHED  10404
TCP    192.168.1.102:6614     111.221.29.75:443     ESTABLISHED   4052
TCP    192.168.1.102:7259     101.227.162.139:80    ESTABLISHED   6696
TCP    192.168.1.102:12410    52.41.66.130:443      ESTABLISHED     8
TCP    192.168.1.102:12877    23.33.164.43:443      ESTABLISHED  12992
TCP    192.168.1.102:13211    14.17.42.118:80       ESTABLISHED     8
TCP    192.168.1.102:13214    101.226.99.117:80     ESTABLISHED   6696
TCP    [::1]:8307             [::1]:12849           ESTABLISHED   5316
TCP    [::1]:8307             [::1]:12871           ESTABLISHED   5316
TCP    [::1]:8307             [::1]:12872           ESTABLISHED   5316
TCP    [::1]:8307             [::1]:12873           ESTABLISHED   5316
TCP    [::1]:12849            [::1]:8307            ESTABLISHED   5316
TCP    [::1]:12871            [::1]:8307            ESTABLISHED   5316
TCP    [::1]:12872            [::1]:8307            ESTABLISHED   5316
TCP    [::1]:12873            [::1]:8307            ESTABLISHED   5316

C:\Users\sm0nk>tasklist /svc | findstr 10404
WinStore.App.exe      10404 暂缺
```

3. 通过 tasklist 命令查看可疑程序

3.2.3 系统信息

1. 使用 set 命令查看变量的设置
2. Windows 的计划任务;
3. Windows 的帐号信息, 如隐藏帐号等
4. 配套的注册表信息检索查看, SAM 文件以及远控软件类
5. 查看 systeminfo 信息, 系统版本以及补丁信息

例如系统的远程命令执行漏洞 MS08-067、MS09-001、MS17-010 (永恒之蓝) ...

若进行漏洞比对, 建议使用 Windows-Exploit-Suggester

<https://github.com/GDSSecurity/Windows-Exploit-Suggester/>

3.2.4 后门排查

PC Hunter 是一个 Windows 系统信息查看软件

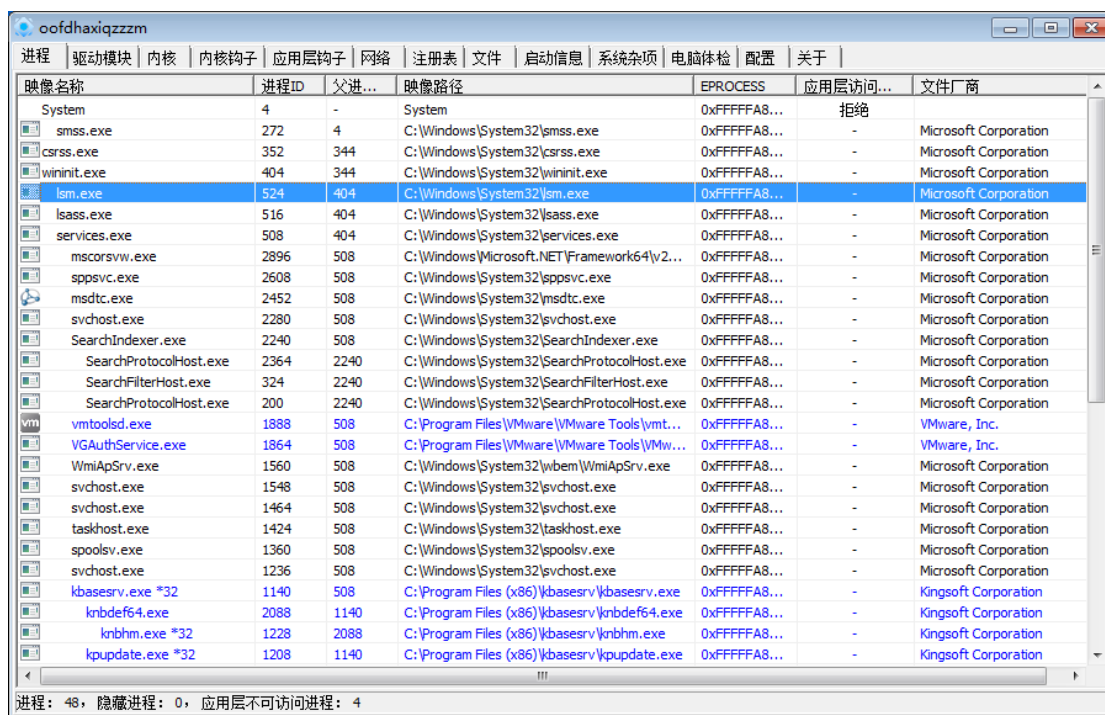
<http://www.xuetr.com/>

功能列表如下：

- 1.进程、线程、进程模块、进程窗口、进程内存信息查看，杀进程、杀线程、卸载模块等功能
- 2.内核驱动模块查看，支持内核驱动模块的内存拷贝
- 3.SSDT、Shadow SSDT、FSD、KBD、TCPIP、Classnpnp、Atapi、Acpi、SCSI、IDT、GDT 信息查看，并能检测和恢复 ssdt hook 和 inline hook
- 4.CreateProcess、CreateThread、LoadImage、CmpCallback、BugCheckCallback、Shutdown、Lego 等 Notify Routine 信息查看，并支持对这些 Notify Routine 的删除
- 5.端口信息查看，目前不支持 2000 系统
- 6.查看消息钩子
- 7.内核模块的 iat、eat、inline hook、patches 检测和恢复
- 8.磁盘、卷、键盘、网络层等过滤驱动检测，并支持删除
- 9.注册表编辑
- 10.进程 iat、eat、inline hook、patches 检测和恢复
- 11.文件系统查看，支持基本的文件操作
- 12.查看（编辑）IE 插件、SPI、启动项、服务、Host 文件、映像劫持、文件关联、系统防火墙规则、IME
- 13.ObjectType Hook 检测和恢复
- 14.DPC 定时器检测和删除
- 15.MBR Rootkit 检测和修复
- 16.内核对象劫持检测
- 17.WorkerThread 枚举
- 18.Ndis 中一些回调信息枚举
- 19.硬件调试寄存器、调试相关 API 检测
- 20.枚举 SFilter/Fltmgr 的回调

21. 系统用户名检测

PS: 最简单的使用方法, 根据颜色去辨识——可疑进程, 隐藏服务、被挂钩函数: 红色, 然后根据程序右键功能去定位具体的程序和移除功能。根据可疑的进程名等进行互联网信息检索然后统一清除并关联注册表。



映像名称	进程ID	父进...	映像路径	EPROCESS	应用层访问...	文件厂商
System	4	-	System	0xFFFFFA8...	拒绝	
smss.exe	272	4	C:\Windows\System32\smss.exe	0xFFFFFA8...	-	Microsoft Corporation
csrss.exe	352	344	C:\Windows\System32\csrss.exe	0xFFFFFA8...	-	Microsoft Corporation
wininit.exe	404	344	C:\Windows\System32\wininit.exe	0xFFFFFA8...	-	Microsoft Corporation
lsass.exe	524	404	C:\Windows\System32\lsass.exe	0xFFFFFA8...	-	Microsoft Corporation
lsass.exe	516	404	C:\Windows\System32\lsass.exe	0xFFFFFA8...	-	Microsoft Corporation
services.exe	508	404	C:\Windows\System32\services.exe	0xFFFFFA8...	-	Microsoft Corporation
mscorsvw.exe	2896	508	C:\Windows\Microsoft.NET\Framework64\v2...	0xFFFFFA8...	-	Microsoft Corporation
sppsvc.exe	2608	508	C:\Windows\System32\sppsvc.exe	0xFFFFFA8...	-	Microsoft Corporation
msdtc.exe	2452	508	C:\Windows\System32\msdtc.exe	0xFFFFFA8...	-	Microsoft Corporation
svchost.exe	2280	508	C:\Windows\System32\svchost.exe	0xFFFFFA8...	-	Microsoft Corporation
SearchIndexer.exe	2240	508	C:\Windows\System32\SearchIndexer.exe	0xFFFFFA8...	-	Microsoft Corporation
SearchProtocolHost.exe	2364	2240	C:\Windows\System32\SearchProtocolHost.exe	0xFFFFFA8...	-	Microsoft Corporation
SearchFilterHost.exe	324	2240	C:\Windows\System32\SearchFilterHost.exe	0xFFFFFA8...	-	Microsoft Corporation
SearchProtocolHost.exe	200	2240	C:\Windows\System32\SearchProtocolHost.exe	0xFFFFFA8...	-	Microsoft Corporation
vmtoolsd.exe	1888	508	C:\Program Files\VMware\VMware Tools\vmtoolsd.exe	0xFFFFFA8...	-	VMware, Inc.
VGAuthService.exe	1864	508	C:\Program Files\VMware\VMware Tools\VGAuthService.exe	0xFFFFFA8...	-	VMware, Inc.
WmiApSrv.exe	1560	508	C:\Windows\System32\wbem\WmiApSrv.exe	0xFFFFFA8...	-	Microsoft Corporation
svchost.exe	1548	508	C:\Windows\System32\svchost.exe	0xFFFFFA8...	-	Microsoft Corporation
svchost.exe	1464	508	C:\Windows\System32\svchost.exe	0xFFFFFA8...	-	Microsoft Corporation
taskhost.exe	1424	508	C:\Windows\System32\taskhost.exe	0xFFFFFA8...	-	Microsoft Corporation
spoolsv.exe	1360	508	C:\Windows\System32\spoolsv.exe	0xFFFFFA8...	-	Microsoft Corporation
svchost.exe	1236	508	C:\Windows\System32\svchost.exe	0xFFFFFA8...	-	Microsoft Corporation
kbasesrv.exe *32	1140	508	C:\Program Files (x86)\kbasesrv\kbasesrv.exe	0xFFFFFA8...	-	Kingsoft Corporation
knbdef64.exe	2088	1140	C:\Program Files (x86)\kbasesrv\knbdef64.exe	0xFFFFFA8...	-	Kingsoft Corporation
knbhm.exe *32	1228	2088	C:\Program Files (x86)\kbasesrv\knbhm.exe	0xFFFFFA8...	-	Kingsoft Corporation
kpupdate.exe *32	1208	1140	C:\Program Files (x86)\kbasesrv\kpupdate.exe	0xFFFFFA8...	-	Kingsoft Corporation

进程: 48, 隐藏进程: 0, 应用层不可访问进程: 4

Webshell 排查

1. 可以使用 hm

序号	类型	路径
1	一句话后门-建议清理	UploadFiles\Temp\201704\201704270600022520.jpg

```

C:\Users\sm0nk\Desktop\Center\nmhms0510>hm.exe scan .

[1.0.0]

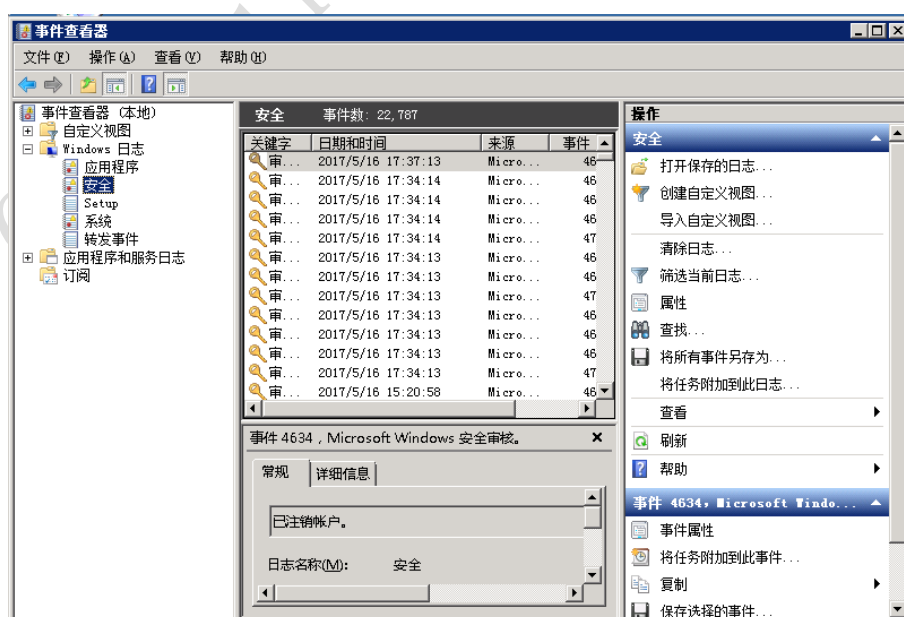
[+] 准备扫描环境 2017-05-10 22:06:45
[*] 开始扫描 2017-05-10 22:06:45
|###-----| 1598/3744 42% [elapsed: 2.0312944s left: 2s, 786.69 iters/sec]
2017/05/10 22:06:48 hm.exe.profile could not open file
2017/05/10 22:06:48 hm.log could not open file
|#####| 3745/3745 100% [elapsed: 4.0313848s left: 0, 928.96 iters/sec]
+-----+
| 类型 | 云查杀 | 数量 |
+-----+
| 后门 |      0 |      1 |
| 疑似 |      0 |     615 |
+-----+
|      | 总计 |     616 |
+-----+
[+] 详细结果已经保存到当前目录下的result.csv文件

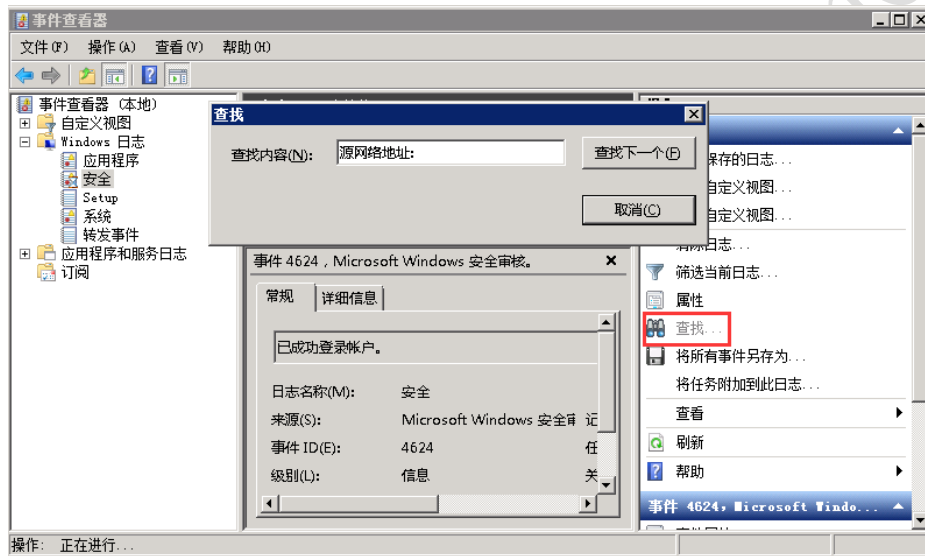
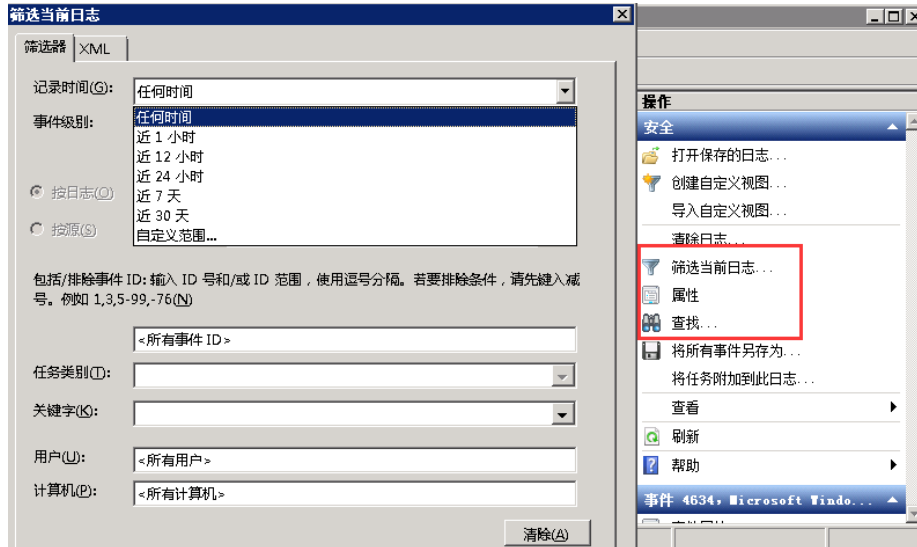
```

2. 也可以使用盾类（D 盾、暗组盾），如果可以把 web 目录导出，可以在自己虚拟机进行分析

3.2.5 日志分析

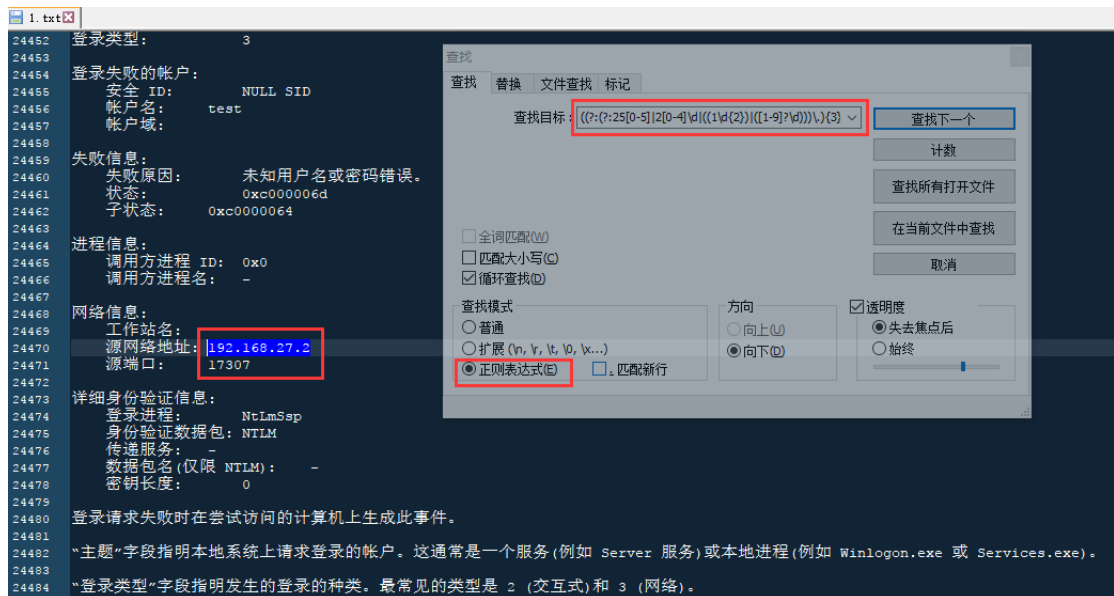
1. 打开事件管理器（开始—管理工具—事件查看/开始运行 eventvwr）
2. 主要分析安全日志，可以借助自带的筛选功能





3. 可以把日志导出为文本格式，然后使用 notepad++ 打开，使用正则模式去匹配远程登录过的 IP 地址，在界定事件日期范围的基础，可以提高效率

正则式: `((?:{?:25[0-5]|2[0-4]\d|((1\d{2})|((1-9)?\d)))\.){3}{?:25[0-5]|2[0-4]\d|((1\d{2})|((1-9)?\d)))`



4. 强大的日志分析工具 [Log Parser](#)

管理员: Log Parser 2.2

```
LogParser -c -i:<input_format> -o:<output_format> <from_entity>
<into_entity> [<where_clause>] [<input_format_options>]
[<output_format_options>] [-multiSite[:ON|OFF]]
[-q[:ON|OFF]] [-e:<max_errors>] [-iw[:ON|OFF]]
[-stats[:ON|OFF]] [-queryInfo]

-i:<input_format> : one of IISW3C, NCSA, IIS, IISODBC, BIN, IISMSID,
HTTPERR, URLSCAN, CSV, TSV, W3C, XML, EVT, ETW,
NETMON, REG, ADS, TEXTLINE, TEXTWORD, FS, COM (if
omitted, will guess from the FROM clause)
-o:<output_format> : one of CSV, TSV, XML, DATAGRID, CHART, SYSLOG,
NEUROVIEW, NAT, W3C, IIS, SQL, TPL, NULL (if omitted,
will guess from the INTO clause)
-q[:ON|OFF] : quiet mode; default is OFF
-e:<max_errors> : max # of parse errors before aborting; default is -1
(ignore all)
-iw[:ON|OFF] : ignore warnings; default is OFF
-stats[:ON|OFF] : display statistics after executing query; default is
ON
-c : use built-in conversion query
-multiSite[:ON|OFF] : send BIN conversion output to multiple files
depending on the SiteID value; default is OFF
-saveDefaults : save specified options as default values
-restoreDefaults : restore factory defaults
-queryInfo : display query processing information (does not
execute the query)
```

Examples:

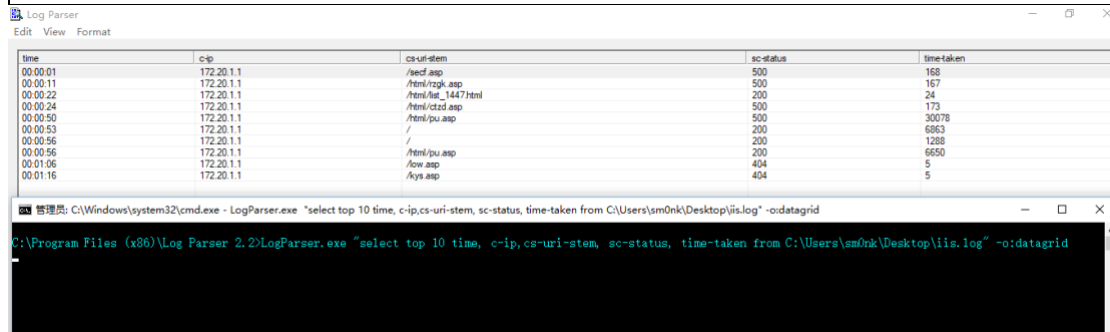
```
LogParser "SELECT date, REVERSEDNS(c-ip) AS Client, COUNT(*) FROM file.log
WHERE sc-status<>200 GROUP BY date, Client" -e:10
LogParser file:myQuery.sql?myInput=C:\temp\ex*.log+myOutput=results.csv
LogParser -c -i:BIN -o:W3C file1.log file2.log "ComputerName IS NOT NULL"
```

Help:

```
-h GRAMMAR : SQL Language Grammar
-h FUNCTIONS [ <function> ] : Functions Syntax
-h EXAMPLES : Example queries and commands
-h -i:<input_format> : Help on <input_format>
-h -o:<output_format> : Help on <output_format>
-h -c : Conversion help
```

#分析 IIS 日志

LogParser.exe "select top 10 time, c-ip,cs-uri-stem, sc-status, time-taken from
C:\Users\sm0nk\Desktop\iis.log" -o:datagrid



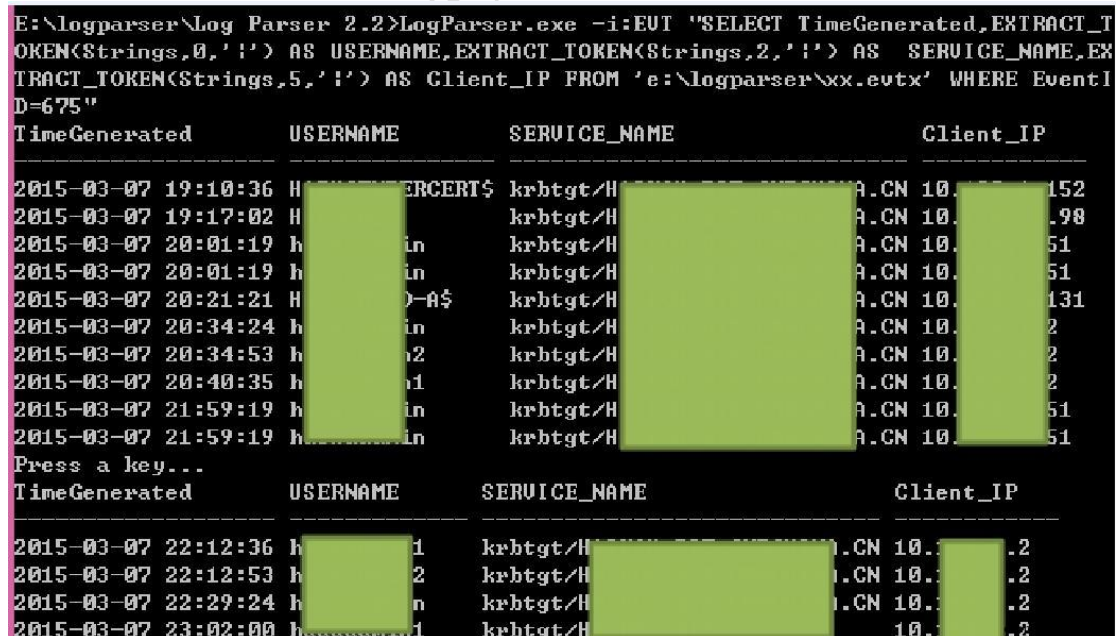
The screenshot shows the LogParser application window with a table of data. Below it, a command prompt window shows the execution of the LogParser command.

time	c-ip	cs-uri-stem	sc-status	time-taken
00:00:01	172.20.1.1	/feed.asp	500	160
00:00:11	172.20.1.1	/html/rzqk.asp	500	167
00:00:22	172.20.1.1	/html/ist_1447.html	200	24
00:00:24	172.20.1.1	/html/ctzd.asp	500	173
00:00:50	172.20.1.1	/html/spu.asp	500	30078
00:00:53	172.20.1.1	/	200	6863
00:00:56	172.20.1.1	/	200	1288
00:00:56	172.20.1.1	/html/spu.asp	200	6650
00:01:06	172.20.1.1	/low.asp	404	5
00:01:16	172.20.1.1	/kys.asp	404	5

```
C:\Windows\system32\cmd.exe - LogParser.exe "select top 10 time, c-ip,cs-uri-stem, sc-status, time-taken from C:\Users\sm0nk\Desktop\iis.log" -o:datagrid  
C:\Program Files (x86)\Log Parser 2.2\LogParser.exe "select top 10 time, c-ip,cs-uri-stem, sc-status, time-taken from C:\Users\sm0nk\Desktop\iis.log" -o:datagrid
```

有了这些我们就可以对 windows 日志进行了分析了 比如我们分析域控日志的时候, 想要查询账户登陆过程中, 用户正确, 密码错误的情况, 我们需要统计出源 IP, 时间, 用户名时, 我们可以这么写 (当然也可以结合一些统计函数, 分组统计等等):

```
LogParser.exe -i:EVT "SELECT TimeGenerated,EXTRACT\_TOKEN(Strings,0,'|') AS  
USERNAME,EXTRACT\_TOKEN(Strings,2,'|') AS  
SERVICE\_NAME,EXTRACT\_TOKEN(Strings,5,'|') AS Client_IP FROM 'e:\logparser\xx.evtx'  
WHERE EventID=675"
```



The screenshot shows a command prompt window with the output of a LogParser query. The output is a table with columns: TimeGenerated, USERNAME, SERVICE_NAME, and Client_IP. The data shows several login attempts with varying success and client IP addresses.

TimeGenerated	USERNAME	SERVICE_NAME	Client_IP
2015-03-07 19:10:36	h...	kerbtgt/H...	10.1.1.152
2015-03-07 19:17:02	h...	kerbtgt/H...	10.1.1.198
2015-03-07 20:01:19	h...	kerbtgt/H...	10.1.1.51
2015-03-07 20:01:19	h...	kerbtgt/H...	10.1.1.51
2015-03-07 20:21:21	h...	kerbtgt/H...	10.1.1.131
2015-03-07 20:34:24	h...	kerbtgt/H...	10.1.1.2
2015-03-07 20:34:53	h...	kerbtgt/H...	10.1.1.2
2015-03-07 20:40:35	h...	kerbtgt/H...	10.1.1.2
2015-03-07 21:59:19	h...	kerbtgt/H...	10.1.1.51
2015-03-07 21:59:19	h...	kerbtgt/H...	10.1.1.51
Press a key...			
TimeGenerated	USERNAME	SERVICE_NAME	Client_IP
2015-03-07 22:12:36	h...	kerbtgt/H...	10.1.1.2
2015-03-07 22:12:53	h...	kerbtgt/H...	10.1.1.2
2015-03-07 22:29:24	h...	kerbtgt/H...	10.1.1.2
2015-03-07 23:02:00	h...	kerbtgt/H...	10.1.1.2

事件 ID 是很好的索引

Windows server 2008 系列参考 event ID:

4624 - 帐户已成功登录

4625 - 帐户登录失败

4648 - 试图使用明确的凭证登录（例如远程桌面）

3.2.6 相关处置

1. 通过网络连接锁定的可疑进程，进行定位恶意程序后删除(taskkill)
2. 木马查杀，可配合 pchunter 进行进一步专业分析，使用工具功能进行强制停止以及删除
3. 最后清理后，统一查看网络连接、进程、内核钩子等是否正常。

3.3 应用类

➤ Mysql MSSQL 数据库类

1. 检查 mysql\lib\plugin 目录没有发现异常文件（参考 UDF 提权）
2. Mysql : select * from mysql.func
3. MSSQL，检查 xp_cmdshell 等存储过程正常与否

➤ Apache、tomcat、Nginx、IIS 的 Web 日志类

无论任何 web 服务器其实日志需要关注的东西是一致的，即 access_log 和 error_log。

一般在确定 ip 地址后，通过：

find . access_log |grep xargs ip 攻击地址

find . access_log| grep xargs 木马文件名

页面访问排名前十的 IP

```
cat access.log | cut -f1 -d " " | sort | uniq -c | sort -k 1 -r | head -10
```

页面访问排名前十的 URL

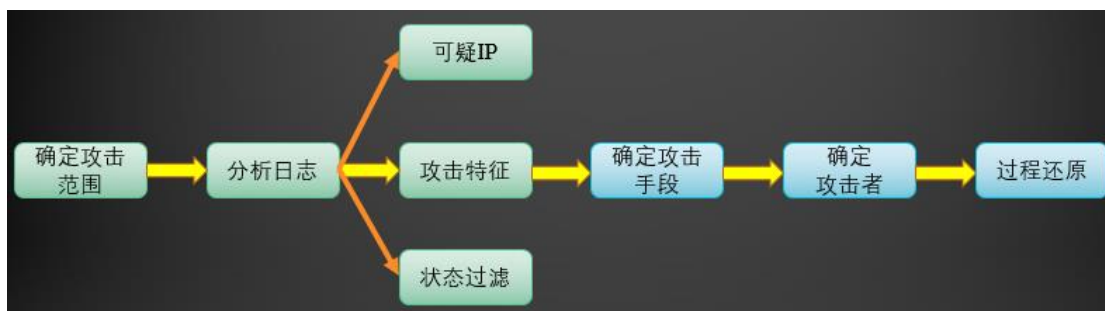
```
cat access.log | cut -f4 -d " " | sort | uniq -c | sort -k 1 -r | head -10
```

查看最耗时的页面

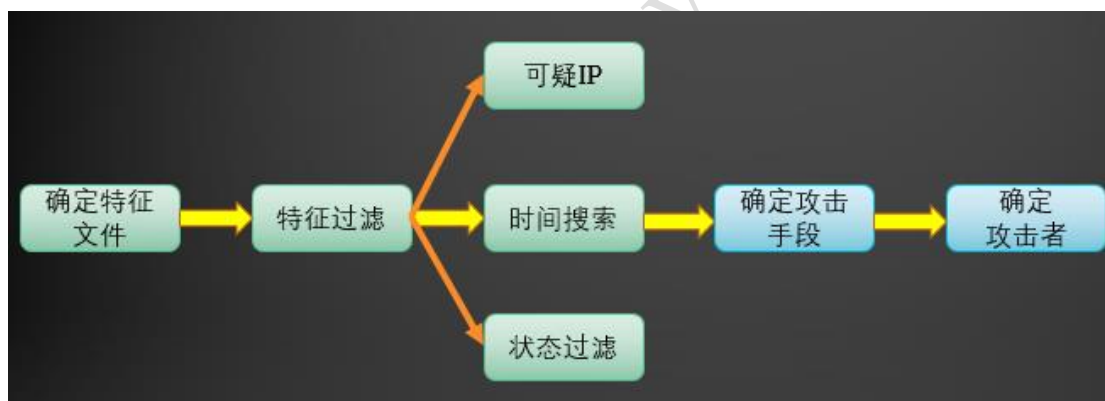
```
cat access.log | sort -k 2 -n -r | head 10
```

在对 WEB 日志进行安全分析时，可以按照下面两种思路展开，逐步深入，还原整个攻击过程。

1. 首先确定受到攻击、入侵的时间范围，以此为线索，查找这个时间范围内可疑的日志，进一步排查，最终确定攻击者，还原攻击过程。



2. 一般攻击者在入侵网站后，通常会上传一个后门文件，以方便自己以后访问。我们也可以以该文件为线索来展开分析。



Web 日志安全分析，完全可以平台化，涉及的知识面也比较丰富，大多数是特征匹配的方式，但基本也结合了模型和学习的新技能。有兴趣的同学可移步 [《Web 日志安全分析浅谈》](#)

4 应急总结

1. 核心思路是“顺藤摸瓜”
2. 碎片信息的关联分析
3. 时间范围的界定以及关键操作时间点串联

4. Web 入侵类, shell 定位很重要
5. 假设与求证
6. 攻击画像与路线确认

5 渗透反辅

1. 密码读取
 - a) Windows: Mimikatz
 - b) Linux: mimipenguin
2. 帐号信息
 - a) 操作系统帐号
 - b) 数据库帐号
 - c) 应用帐号信息
3. 敏感信息
 - a) 配置信息
 - b) 数据库信息
 - c) 服务端口信息
 - d) 指纹信息
4. 滚雪球式线性拓展
 - a) 密码口令类拓展 (远控)
 - b) 典型漏洞批量利用
5. 操作系统攻防 TIPS
 - a) Ubuntu 系统, 在命令输入前加个空格, 命令操作不会被记录到 history
 - b) 针对 linux 的文件时间, 可以采用 touch -r 进行迷惑排查者
6. 常见的入侵方式 Getshell 方法
 - a) WEB 入侵
 - i. 典型漏洞: 注入 Getshell, 上传 Getshell, 命令执行 Getshell, 文件包含 Getshell, 代码执行 Getshell, 编辑器 getshell, 后台管理 Getshell, 数据库操作 Getshell

- ii. 容器相关: Tomcat、Axis2、WebLogic 等中间件弱口令上传 war 包等, Websphere、weblogic、jboss 反序列化, Struts2 代码执行漏洞, Spring 命令执行漏洞
- b) 系统入侵
 - i. SSH 破解后登录操作
 - ii. RDP 破解后登录操作
 - iii. MSSQL 破解后远控操作
 - iv. SMB 远程命令执行 (MS08-067、MS17-010、CVE-2017-7494)
- c) 典型应用
 - i. Mail 暴力破解后信息挖掘及漏洞利用
 - ii. VPN 暴力破解后绕过边界
 - iii. Redis 未授权访问或弱口令可导 ssh 公钥或命令执行
 - iv. Rsync 未授权访问类
 - v. MongoDB 未授权访问类
 - vi. Elasticsearch 命令执行漏洞
 - vii. Memcache 未授权访问漏洞
 - viii. 服务相关口令 (mysql ldap zebra squid vnc smb)

6 资源参考

<https://www.waitalone.cn/linux-find-webshell.html>

<http://vinc.top/category/yjxy/>

<http://www.shellpub.com/>

http://linux.vbird.org/linux_security/0420rkhunter.php

<https://cisofy.com/download/lynis/>

<https://sobug.com/article/detail/27?from=message&isappinstalled=1>

<http://www.freebuf.com/articles/web/23358.html>

<https://www.microsoft.com/en-us/download/details.aspx?id=24659>

<http://www.cnblogs.com/downmoon/archive/2009/09/02/1558409.html>

<http://wooyun.jozxing.cc/static/drops/tips-7462.html>

<http://bobao.360.cn/learning/detail/3830.html>

<https://yq.aliyun.com/ziliao/65679>

<http://secsky.sinaapp.com/188.html>

http://blog.sina.com.cn/s/blog_d7058b150102wu07.html

<http://www.sleuthkit.org/autopsy/>

<https://xianzhi.aliyun.com/forum/read/1723.html>

7 FAQ

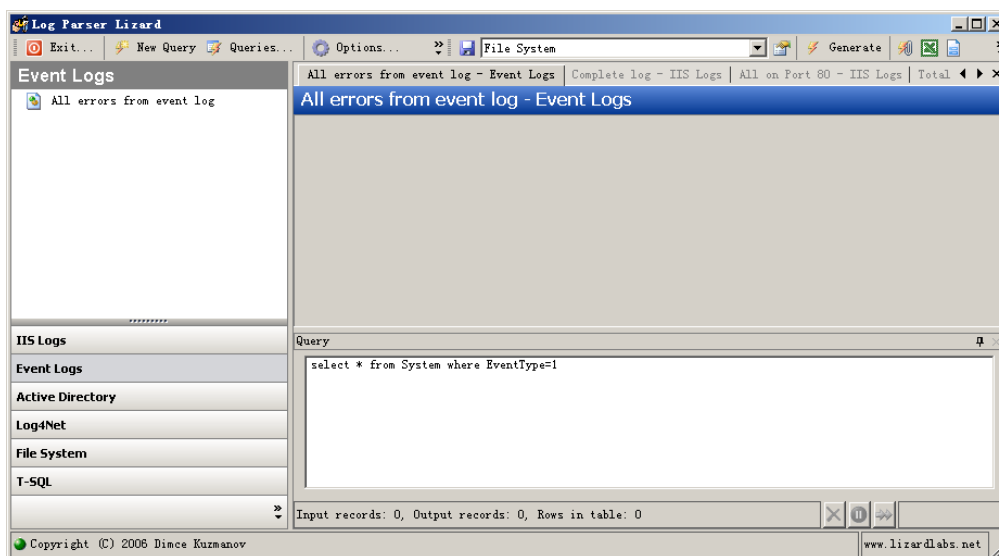
1. 应急需求有哪些分类：

- a) 被谁入侵了？ 关联 攻击 IP 攻击者信息
- b) 怎么入侵的？ 关联 入侵时间轴、漏洞信息
- c) 为什么被入侵？ 关联 行业特性、数据信息、漏洞信息
- d) 数据是否被窃取？ 关联 日志审计
- e) 怎么办？ 关联 隔离、排查分析、删马（解密）、加固、新运营

2. 关于 windows 的日志工具（log parser）有无图形界面版？

Log Parser Lizard 是一款用 Vc++.net 写的 logParser 增强工具。主要有以下特点：

- a) 封装了 logParser 命令，带图形界面，大大降低了 LogParser 的使用难度。
- b) 集成了几个开源工具，如 log4net 等。可以对 IIS logs\EventLogs\active directory\log4net\File Systems\T-SQL 进行方便的查询。
- c) 集成了 Infragistics.UltraChart.Core.v4.3、Infragistics.Excel.v4.3.dll 等，使查询结果可以方便的以图表或 EXCEL 格式展示。
- d) 集成了常见的查询命令，范围包含六大模块:IIS
- e) 可以将查询过的命令保存下来，方便再次使用。



PS:软件是比较老的, 对新系统兼容性不好, 还是建议微软原生态 log parser

3. 在 linux 日志中, 有无黑客入侵后的操作命令的统计

- 可以根据 history 信息进行溯源分析, 但一般可能会被清除
- 还有方法是需要结合 accton 和 lastcomm

```
[root@localhost sm0nk]# accton off
Turning off process accounting.
[root@localhost sm0nk]# accton /home/sm0nk/pacct.log
Turning on process accounting, file set to '/home/sm0nk/pacct.log'.
[root@localhost sm0nk]# lastcomm -f /home/sm0nk/pacct.log
bash          SF      root    pts/0      0.00 secs Thu Jun 1 15:28
bash          SF      root    pts/0      0.00 secs Thu Jun 1 15:28
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:28
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:28
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:28
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:28
cat           sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
ksmtuned      F       root    —          0.00 secs Thu Jun 1 15:27
awk           root    —          0.00 secs Thu Jun 1 15:27
ksmtuned      F       root    —          0.00 secs Thu Jun 1 15:27
ksmtuned      F       root    —          0.00 secs Thu Jun 1 15:27
pgrep         root    —          0.02 secs Thu Jun 1 15:27
ksmtuned      F       root    —          0.00 secs Thu Jun 1 15:27
awk           root    —          0.00 secs Thu Jun 1 15:27
sleep         root    —          0.00 secs Thu Jun 1 15:26
bash          F       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
gdbus         X       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
bash          F       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
pool         X       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
uname         sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
id            sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:27
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:27
bash          F       root    pts/0      0.00 secs Thu Jun 1 15:27
accton        S       root    pts/0      0.00 secs Thu Jun 1 15:27
[root@localhost sm0nk]# lastcomm -f /home/sm0nk/pacct.log --user sm0nk
cat           sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
bash          F       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
gdbus         X       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
bash          F       sm0nk   pts/1      0.00 secs Thu Jun 1 15:27
```

4. 3.2.3 提到了 Windows-Exploit-Suggester, 有无 linux 版?

Linux_Exploit_Suggester https://github.com/PenturaLabs/Linux_Exploit_Suggester


```
$ perl ./Linux_Exploit_Suggester.pl -k 2.6.28

Kernel local: 2.6.28

Possible Exploits:
[+] sock_sendpage2
    Alt: proto_ops    CVE-2009-2692
    Source: http://www.exploit-db.com/exploits/9436
[+] half_nelson3
    Alt: econet    CVE-2010-4073
    Source: http://www.exploit-db.com/exploits/17787/
[+] reiserfs
    CVE-2010-1146
    Source: http://www.exploit-db.com/exploits/12130/
[+] pktdcdrv
    CVE-2010-3437
    Source: http://www.exploit-db.com/exploits/15150/
[+] american-sign-language
    CVE-2010-4347
    Source: http://www.securityfocus.com/bid/45408/
[+] half_nelson
    Alt: econet    CVE-2010-3848
```

5. 有无 linux 自动化信息收集的脚本工具?

LinEnum <https://github.com/rebootuser/LinEnum>

```
root@kali:~/Desktop/PenTest/LinEnum# ./LinEnum.sh
#####
# Local Linux Enumeration & Privilege Escalation Script #
# www.rebootuser.com                                     #
# PenTest                                                #
#####
Debug Info
thorough tests = disabled

Scan started at:
Thu Jun  1 21:14:25 EDT 2017

### SYSTEM #####
Kernel information:
Linux kali 4.9.0-kali3-amd64 #1 SMP Debian 4.9.18-1kali1 (2017-04-04) x86_64 GNU/Linux

Kernel information (continued):
Linux version 4.9.0-kali3-amd64 (devel@kali.org) (gcc version 6.3.0 20170321 (Debian 6.3.0-11) ) #1 SMP Debian 4.9.18-1kali1 (2017-04-04)
```

6. 检测病毒文件的几个网站

<https://x.threatbook.cn/>

<http://www.virscan.org>

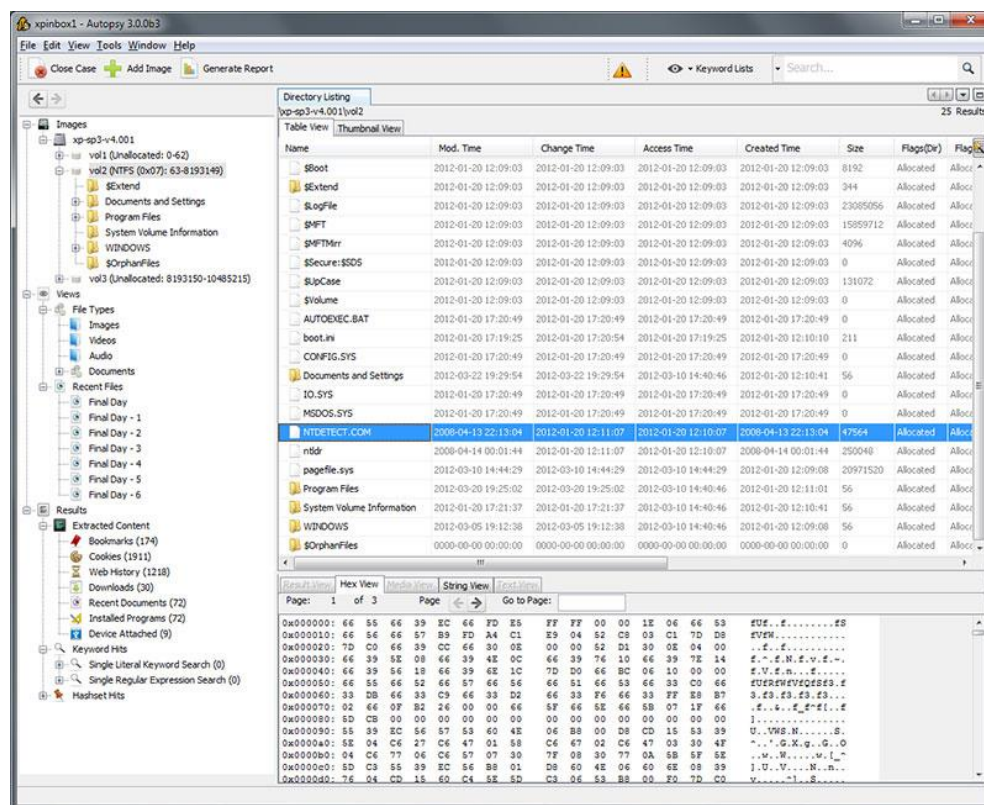
<https://www.virustotal.com/>

<https://fireeye.ijinshan.com/>

7. 有无综合的取证分析工具

Autopsy 是 sleuthkit 提供的平台工具, Windows 和 Linux 磁盘映像静态分析、恢复被删文件、时间线分析, 网络浏览历史, 关键字搜索和邮件分析等功能

<http://www.sleuthkit.org/autopsy/>



8. 关于业务逻辑的排查方法说明

新型业务安全中安全事件, 例如撞库、薅羊毛、支付、逻辑校验等敏感环节, 未在本文体现, 所以后续有必要针对业务侧的应急排查方法归纳。