

HW防守 | Linux应急响应基础

原创 璠淳 Timeline Sec 昨天

0x00 引言

简单说一下，我们为什么会推出关于HW防守的文章，目前关于该行动，会发现越来越多的厂商需求该行动的人员具备分析溯源的能力了。

其中原因一是由于该行动规则的需求，溯源能力可以很好的进行攻击队画像描述，追踪到更多的信息线索，从而给防守方增加更多的该行动的分。

二是任何产品目前都是需要人来驱动的，都有其局限性，比如日志采集的灵活度及完整性等，我们主观的收集一些信息，可以更好的辅助产品，也可以兼顾不同种类安全产品的一些边界问题。

目前已经推出windows基础篇及此篇linux基础篇试试水，方便大家进行该行动的时候查阅知识点进行基础溯源，同时也欢迎大家反馈想法与意见，如果后续效果可以的话，我们会推出一些真实脱敏的溯源加分案例进行交流。

0x01 技能树

- Linux常用命令
- 常见日志的位置以及分析方法
- 熟悉常规黑客的攻击手法
- 常规安全事件的处置思路

0x02 linux 常用命令

查找与文本操作

1、find

根目录下所有.jsp后缀文件

```
1 find / -name *.jsp
```

最近3天修改过的文件

```
1 find -type f -mtime -3
```

最近3天创建的文件

```
1 find -type f -ctime -3
```

2、grep、strings、more、head、tail

过滤出不带有某个关键词的行并输出行号

```
1 grep -nv 'root' /etc/passwd
```

查看根目录下 含有root信息的文件，并标注行号

```
1 grep -nr root /
```

查看根目录下后缀为.jsp .jspx文件，并从大到小排列

```
1 grep -nr -v "404" ./ | grep -E "\.jsp | \.jspx" | more
```

显示文件前十行

```
1 head /etc/passwd
```

实时展示文件内容

```
1 tail -f 文件名
```

3、awk、sort、uniq

awk的F参数是指定分隔符，print \$1意思是打印第一列，sort命令是用来排序的，uniq命令是用来把相邻的重复数据聚合到一起，加个c参数意思就是把重复次数统计出来，为什么先要用sort聚合一次呢，就是因为uniq命令只会聚合相邻的重复数据，最后那个sort命令刚才说了是用于排序的，他的n参数是以数字排序，r参数是倒叙排序

```
1 awk -F " " '{print $1}' access.log | sort | uniq -c | sort -nr
```

案例：

我们以空格为分界线（\$1为第一行）对access.log日志进行分析，筛查提取访问IP 从大到小排序，并提示访问次数。

```
access_log error_log
fanchun@192 ~$ cat access_log | head
127.0.0.1 - - [12/May/2020:10:37:11 +0800] "GET / HTTP/1.1" 200 45
127.0.0.1 - - [12/May/2020:10:37:11 +0800] "GET /favicon.ico HTTP/1.1" 404 196
25.25.25.236 - - [12/May/2020:17:48:05 +0800] "GET / HTTP/1.1" 200 45
25.25.25.236 - - [12/May/2020:17:48:05 +0800] "GET /WPAPI/ HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "POST /index.php?s=/home/user/checkcode/ HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "POST /index.php?s=captcha HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "POST /index.php?s=index/index/index HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "GET /index.php?ids%5B0,UpdAtexml(0,Concat(0xa,Md5(2333)),0)%5D=1 HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "GET /index.php?s=index/%5Cthink%5Cview%5Cdriver%5CPhp/display&content=%3C?php%20var_dump(md5(2333));?%3E HTTP/1.1" 404 196
127.0.0.1 - - [13/May/2020:20:26:48 +0800] "GET /index.php?s=my-show-id-%5Cx5C.%5Cx5CTp!%5Cx5CBedy%5Cx5CHome%5Cx5Cmy_1%7B~print_r(md5(2333))%7D%5D HTTP/1.1" 404 196
fanchun@192 ~$ awk -F " " '{print $1}' access_log | sort | uniq -c | sort -nr
893 10.100.110.7
57 10.10.2.66
29 127.0.0.1
3 10.111.65.99
2 25.25.25.236
2 25.25.24.133
1 ::1
```

系统状态命令

1、lsof

查看某个用户启动了什么进程

```
1 lsof -u root
```

某个端口是哪个进程打开的

```
1 lsof -i:8080
```

2、last、lastb、lastlog

登录失败记录：/var/log/btmp

```
1 lastb
```

最后一次登录: /var/log/lastlog

```
1 lastlog
```

登录成功记录: /var/log/wtmp

```
1 last
```

3、crontab

查看计划任务是否有恶意脚本或者恶意命令

```
1 crontab -l
```

4、netstat

a参数是列出所有连接, n是不要解析机器名, p列出进程名

```
1 netstat -anp
```

5、ps

查看进程信息

```
1 ps -ef
2 ps -aux
```

6、top

查看进程cpu占比 (动态任务, 可实时查看最高cpu占有率)

```
1 top
```

7、stat

查看某个文件是否被修改过

```
1 stat
```

8、last和lastb (对应日志wtmp/btmp)

last查看成功登陆的IP (用于查看登陆成功信息)

登陆用户---连接方式---时间

```
root@kali:~# last
root      :1              :1              Sat Apr 25 01:42      gone - no logout
reboot    system boot    4.19.0-kali3-amd Sat Apr 25 01:42      still running
root      :1              :1              Sat Apr 25 01:40 - 01:42 (00:01)
reboot    system boot    4.19.0-kali3-amd Sat Apr 25 01:40 - 01:42 (00:02)
root      :1              :1              Sat Apr 25 01:26 - 01:40 (00:13)
reboot    system boot    4.19.0-kali3-amd Sat Apr 25 01:26 - 01:40 (00:14)
root      :1              :1              Sat Apr 25 01:22 - 01:25 (00:03)
reboot    system boot    4.19.0-kali3-amd Sat Apr 25 01:22 - 01:25 (00:03)
root      :1              :1              Sat Apr 25 01:00 - down (00:14)
```

lastb查看连接失败的IP (可用于查看爆破信息)

登陆用户---登陆方式---登陆IP---时间

```
root@kali:~# lastb
root      ssh:notty      192.168.2.1      Sat May 2 12:03 - 12:03 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:03 - 12:03 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:03 - 12:03 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:02 - 12:02 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:02 - 12:02 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:02 - 12:02 (00:00)
root      ssh:notty      192.168.2.1      Sat May 2 12:02 - 12:02 (00:00)
```

0x03 日志分析

1、安全日志 /var/log/secure

作用：安全日志secure包含验证和授权方面信息

分析：是否有IP爆破成功

2、用户信息 /etc/passwd

内容含义：注册名、口令、用户标识号、组标识号、用户名、用户主目录、命令解释程序

分析：是否存在攻击者创建的恶意用户

3、命令执行记录 ~/.bash_history

作用：命令执行记录 ~/.bash_history

分析：是否有账户执行过恶意操作系统命令

4、root邮箱 /var/spool/mail/root

作用：root邮箱 /var/spool/mail/root

分析：root邮箱的一个文件，在该文件中包含大量信息，当日志被删除可查询本文件

5、中间件日志(Web日志access_log)

1 nginx、apache、tomcat、jboss、weblogic、websphere

作用：记录访问信息

分析：请求次数过大，访问敏感路径的IP

位置：/var/log下 access.log文件（apache默认位置）

位置：/var/log/nginx下 access名称日志（nginx日志位置）

位置：tomcat、weblogic等日志均存放在安装路径下logs文件下

访问日志结构：访问IP---时间---请求方式---请求路径---请求协议----请求状态---字节数

```
10.100.110.7 - - [01/Jun/2020:18:03:42 +0800] "GET /scripts/remothtmlview.php?phpAds_path=http://xxxxxxx HTTP/1.1" 404 196 2020年4月18日 下午11:29 24.9 MB 应用程
10.100.110.7 - - [01/Jun/2020:18:03:42 +0800] "GET /SiteScope/cgi/go.exe/SiteScope?page=eventLog&machine=6logName=System&account=administrator HTTP/1.1" 404 196 程
10.100.110.7 - - [01/Jun/2020:18:03:42 +0800] "GET /cgi-bin/search/search.cgi?keys=*&pre-any&category=../../../../../../../../../../../../etc HTTP/1.1" 404 196 用程
10.100.110.7 - - [01/Jun/2020:18:03:43 +0800] "GET /DB4Web/www.microsoft.com:23/foo HTTP/1.1" 404 196 2020年4月9日 上午8:37 333 MB 应用程
10.100.110.7 - - [01/Jun/2020:18:03:43 +0800] "GET /scripts/search/search.cgi?keys=*&pre-any&category=../../../../../../../../../../../../etc HTTP/1.1" 404 196 2020年4月9日 下午6:42 737.6 MB 应用程
10.100.110.7 - - [01/Jun/2020:18:03:43 +0800] "GET /nx/common/cds/menu.inc.php?c_path=https://xxxxxxx/ HTTP/1.1" 404 196 2020年4月19日 下午10:05 356.4 MB 应用程
10.100.110.7 - - [01/Jun/2020:18:03:43 +0800] "GET /ifx/7LO../../../../../../../../etc/passwd HTTP/1.1" 404 196 2020年4月17日 上午10:43 469 KB 应用程
10.100.110.7 - - [01/Jun/2020:18:03:43 +0800] "PROPFIND / HTTP/1.1" 405 225 ZeroTier One
```

6.登陆日志（可直接使用命令调取该信息，对应命令last/lastb）

位置：/var/log/wtmp #成功连接的IP信息

位置：/var/log/btmp #连接失败的IP信息

7.cron(定制任务日志)日志

位置：/var/log/cron

作用：查看历史计划任务（对该文件进行分析调取恶意病毒执行的计划任务，获取准确时间）


```

May 31 03:20:01 nfs CROND[10544]: (root) CMD (/usr/lib64/sa/sa1 1 1)

[root@nfs ~]# cat /var/log/cron
May 31 03:40:01 nfs CROND[26634]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 03:40:01 nfs run-parts(/etc/cron.daily)[26640]: finished man-db.cron
May 31 03:40:01 nfs anacron[24716]: Job `cron.daily' terminated
May 31 03:40:01 nfs anacron[24716]: Normal exit (1 job run)
May 31 03:50:01 nfs CROND[27129]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:00:01 nfs CROND[27617]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:01:01 nfs CROND[27673]: (root) CMD (run-parts /etc/cron.hourly)
May 31 04:01:01 nfs run-parts(/etc/cron.hourly)[27673]: starting 0anacron
May 31 04:01:01 nfs run-parts(/etc/cron.hourly)[27682]: finished 0anacron
May 31 04:10:01 nfs CROND[28135]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:20:01 nfs CROND[28625]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:30:01 nfs CROND[29112]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:40:01 nfs CROND[29600]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 04:50:01 nfs CROND[30090]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 05:00:01 nfs CROND[30578]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 05:01:01 nfs CROND[30634]: (root) CMD (run-parts /etc/cron.hourly)
May 31 05:01:01 nfs run-parts(/etc/cron.hourly)[30634]: starting 0anacron
May 31 05:01:01 nfs run-parts(/etc/cron.hourly)[30643]: finished 0anacron
May 31 05:10:01 nfs CROND[31083]: (root) CMD (/usr/lib64/sa/sa1 1 1)
May 31 05:20:01 nfs CROND[31574]: (root) CMD (/usr/lib64/sa/sa1 1 1)

```

8、history日志

位置： ~/.bash_history

作用：操作命令记录，可筛查攻击者执行命令信息

```

[root@nfs ~]# cat ~/.bash_history
yum -y install nfs-util
yum -y install nfs-utils
more /etc/redhat-release
ss -tnulp | grep 111
systemctl start rpcbind
rpm -qa nfs-utils
vim /etc/exports
exports -v
exportfs
exportfs -v
systemctl start nfs
vim /etc/exports
exports -r
exportfs -r
vim /etc/exports
exportfs -r
vim /etc/exports
exportfs -r
systemctl start nfs
rpcinfo -p 39.106.102.123
showmount -e localhost
touch /home/nfs/1.txt
echo "hello nfs" >>/data/1.txt
echo "hello nfs" >>/home/nfs/1.txt
vim /etc/exports
exportfs -v
service nfs-server start
vim /etc/exports
iptables -I INPUT -p udp -m multiport --dports 875,2049,111,37747,3116
iptables -I INPUT -p tcp -m multiport --dports 875,2049,111,28300,5076

```

9、其他日志

redis、sql server、mysql、oracle等

作用：记录访问信息

分析：敏感操作

web日志分析思路：



寻找Webshell的方法：

1、文件内容中的恶意函数

PHP: eval(、 system(、 assert(
JSP: getRunTime(、 FileOutputStream(
ASP: eval(、 execute(、 ExecuteGlobal (

2、Web日志中的webshell特征

Darkblade: goaction=login
JspSpy: o=login
PhpSpy: action=phpinfo
Regeorg: cmd=connect
Other: cmd=

3、贴合Web业务中的url来分析Web日志

4、每天新增的动态脚本文件

5、低频访问的脚本文件

本篇完

欢迎投稿HW防守相关文章！



关注我们看更多HW相关文章

Timeline Sec 团队

安全路上，与你并肩前行