溯源的思路

看对方的目的是什么，就是最终目标是做什么。然后根据自己经验 看看达到这个目标 需要进行什么操作逆推回去。看看这些过程都会留下什么日志。

下手的几个点

> 网站源码分析
>
> 代码修改，可疑文件，webshell
>
> 日志分析
>
> 根据时间， 文件，ip等，可以的登录，读取存储，定时任务，错误日志
>
> 系统的信息分析
>
> 执行的代码，可疑的用户（passwd），ssh可疑的公钥
>
> 分析进程端口
>
> 端口占用，可疑进程，杀死进程

网站源码文件分析

分析网站源码可以帮助我们获取网站被入侵时间, 黑客如何的 IP, 等信息, 对于接下来的日志分析有很大帮助。

1. 查杀后门

可以使用 D 盾查杀是否存在网站后门，如果存在 webshell，记录下该 webshell 的信息。

找到 webshell 后，就可以根据该文件的路径，在日志里查找有关信息，例如访问该文件的 IP、时间等。可以根据这些信息确定网站别入侵的时间，从而缩小搜索范围，运气好了可以直接根据 IP 找到黑客。

2. diff 源码，查找被修改的地方，记录被修改代码的信息。

diff 工具推荐-diffmerge

可以根据被修改的文件的修改时间，缩小搜索范围。

3. 查看指定目录下文件时间的排序

可以根据文件的排序迅速找到被黑客修改的文件，从而找到入侵时间。

```
➜  ~ ls -alt | head -n 10

总用量 2432

drwxr-xr-x 35 yang yang   4096 6 月  28 21:43 .

-rw-------  1 yang yang  41214 6 月  28 21:43 .zsh_history

-rw-------  1 yang yang 413115 6 月  28 21:42 .xsession-errors
```

```
drwxr-xr-x  2 yang yang   4096 6 月  28 21:41 .sogouinput

drwxr-xr-x  6 yang yang   4096 6 月  28 20:40 Desktop

drwxr-xr-x 16 yang yang   4096 6 月  28 18:30 .cache

drwxr-xr-x 27 yang yang   4096 6 月  28 09:53 .config

drwx------  2 yang yang   4096 6 月  28 07:54 .gconf

-rw-------  1 yang yang     49 6 月  28 07:54 .Xauthority
```

4. 使用 find 指令查找限定时间范围的文件

例：查看 10 分钟内修改过的文件

```
➜  html sudo find ./ -cmin -10 -name "*.php"

./1.php
```

5. 查看文件详细信息

```
➜  html stat waf.php

  文件：waf.php

  大小：0               块：0          IO 块：4096   普通空文件

设备：802h/2050d    Inode：837154      硬链接：1

权限：(0644/-rw-r--r--)  Uid：(    0/    root)   Gid：(    0/    root)

最近访问：2018-06-21 18:51:19.492195229 +0800

最近更改：2018-06-20 21:28:56.907316842 +0800

最近改动：2018-06-20 21:28:56.907316842 +0800

创建时间：-
```

日志分析

网站日志分析

网站日志一般为

```
- access.log

- error.log
```

根据上一步分析网站源码得到的信息在对日志文件进行筛选分析，因为日志文件会记录很多信息，如果一条一条分析，不是很现实。

1. 根据时间筛选

```
sudo cat access.log| grep '27/Jun/2018'
```

2. 根据特殊文件名筛选

```
sudo cat access.log| grep '文件名'
```

3. 根据 ip 筛选

```
sudo cat access.log| grep 'ip'
```

4. 对访问服务器的 IP 进行统计排序

```
sudo cat /var/log/apache2/access.log | cut -f1 -d ' '| sort | uniq -c
```

[web-log 分析工具]

系统日志分析

> /var/log/wtmp 登录进入，退出，数据交换、关机和重启纪录
>
> /var/run/utmp 有关当前登录用户的信息记录
>
> /var/log/lastlog 文件记录用户最后登录的信息，可用 lastlog 命令来查看。
>
> /var/log/secure 记录登入系统存取数据的文件，例如 pop3/ssh/telnet/ftp 等都会被记录。
>
> /var/log/cron 与定时任务相关的日志信息
>
> /var/log/message 系统启动后的信息和错误日志

/var/log/wtmp 和/var/run/utmp 两个文件无法直接使用 cat 命令输出，但是可以使用一些命令来查看，比如 w/who/finger/id/last/ac/uptime

1. w 命令

该命令查询 /var/log/wtmp 文件并显示 当前 系统中每个用户和它所运行的进程信息：

```
➜  ~ w

 17:47:16 up  9:53,  1 user,  load average: 2.45, 1.81, 1.62

USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT

yang     tty1     :0               07:54    9:53m  1:15m  1:51
/usr/bin/startdde
```

2. last

该命令往回搜索 /var/log/wtmp 文件来显示自从该文件第一次创建以来所有登录过的用户：

如果指明了用户，则该命令只显示该用户的近期活动：

```
➜  ~ last

yang     tty1         :0               Thu Jun 28 07:54   still logged in

reboot   system boot  4.15.0-21deepin- Thu Jun 28 07:53   still running

yang     tty1         :0               Wed Jun 27 08:52 - 22:02  (13:10)

reboot   system boot  4.15.0-21deepin- Wed Jun 27 08:51 - 22:03  (13:11)

yang     tty1         :0               Tue Jun 26 10:01 - down   (12:39)

reboot   system boot  4.15.0-21deepin- Tue Jun 26 10:00 - 22:41  (12:40)

reboot   system boot  4.15.0-21deepin- Tue Jun 26 09:54 - 22:41  (12:46)

//指定用户

➜  ~ last reboot

reboot   system boot  4.15.0-21deepin- Thu Jun 28 07:53   still running

reboot   system boot  4.15.0-21deepin- Wed Jun 27 08:51 - 22:03  (13:11)

reboot   system boot  4.15.0-21deepin- Tue Jun 26 10:00 - 22:41  (12:40)

reboot   system boot  4.15.0-21deepin- Tue Jun 26 09:54 - 22:41  (12:46)

reboot   system boot  4.15.0-21deepin- Mon Jun 25 08:14 - 20:49  (12:34)

reboot   system boot  4.15.0-21deepin- Sun Jun 24 21:46 - 22:54  (01:07)
```

3. lastlog 命令

/var/log/lastlog 文件在每次有用户登录时被查询。可以使用 lastlog 命令来检查某特定用户上次登录的时间，并格式化输出上次登录日志 /var/log/lastlog 的内容。它根据 UID 排序显示登录名、端口号（tty）和上次登录时间。如果一个用户从未登录过，lastlog 显示 Never logged(从未登录过)。注意需要以 root 运行该命令：

```
➜  ~ lastlog

用户名            端口     来自             最后登陆时间

root                                        **从未登录过**

daemon                                      **从未登录过**

bin                                         **从未登录过**

sys                                         **从未登录过**
```

```
sync                                            **从未登录过**

games                                           **从未登录过**

man                                             **从未登录过**

lp                                              **从未登录过**

mail                                            **从未登录过**

news                                            **从未登录过**

uucp                                            **从未登录过**

//lastlog -u 'uid' 该指令仅输出 uid 为 0 的用户。

➜  ~ lastlog -u 0

用户名              端口       来自                最后登陆时间

root                                            **从未登录过**
```

\4. id 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，那他的用户名将显示相同的次数：

```
➜  ~ id

uid=1000(yang) gid=1000(yang) 组
=1000(yang),7(lp),27(sudo),100(users),109(netdev),113(lpadmin),117(scanner),123(
sambashare)

➜  ~ id yang

uid=1000(yang) gid=1000(yang) 组
=1000(yang),7(lp),27(sudo),100(users),109(netdev),113(lpadmin),117(scanner),123(
sambashare)
```

系统信息分析

> history
>
> /etc/passwd
>
> ls -alt /etc/init.d
>
> 查看用户登录信息 (lastlog,lastb,last)
>
> 查看是否有 ssh 可疑公钥

1. history

可使用该指令查看服务器上使用过的历史指令。通过 history 信息可能获得以下敏感信息

- wget (远程某主机的远控文件)

- ssh 尝试连接内网的某些机器

- tar zip 可以知道攻击者打包了哪些敏感数据

- 可知道攻击者对服务器做了哪些配置上的修改 (添加用户，留后门等)

2. /etc/passwd

可通过该文件分析可疑账号

3. 分析服务器的开机自启程序，分析是否存在后门木马程序。

1\. ls -alt /etc/init.d

2\. /etc/init.d/rc.local /etc/rc.local

3\. chkconfig

4. 查看登录信息

1\. lastlog(查看系统中所有用户最近一次的登录信息)

2\. lasstb (查看用户的错误登录信息)

3\. last(显示用户最近登录信息)

5. 查看 ssh 相关目录

redis 未授权访问漏洞可直接向服务器写入公钥，从而实现无密码登录服务器。

所以要查看/etc/.ssh ~/.ssh 目录下有无可疑公钥

分析进程 (端口)

检查服务器是否有黑客留下的木马程序。

1. 查看端口占用情况

```
➜  ~ netstat -apn|more

(Not all processes could be identified, non-owned process info

 will not be shown, you would have to be root to see it all.)

Active Internet connections (servers and established)

Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name

tcp        0      0 0.0.0.0:902             0.0.0.0:*               LISTEN      -

tcp        0      0 0.0.0.0:139             0.0.0.0:*               LISTEN      -

tcp        0      0 127.0.0.1:4300          0.0.0.0:*               LISTEN
16378/wineserver.re

tcp        0      0 127.0.0.1:4301          0.0.0.0:*               LISTEN
16378/wineserver.re

tcp        0      0 127.0.0.1:8307          0.0.0.0:*               LISTEN      -

tcp        0      0 127.0.0.1:5939          0.0.0.0:*               LISTEN      -


tcp        0      0 127.0.0.1:631           0.0.0.0:*               LISTEN      -

tcp        0      0 127.0.0.1:1080          0.0.0.0:*               LISTEN      -

tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN      -

tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN      -

tcp        0      0 192.168.10.119:33930    112.34.111.124:443      ESTABLISHED
2798/chrome
```

2. 根据上一步得出的可疑端口的 pid 分析进程

指令：ps aux|grep ‘pid’

```
➜  ~ ps aux | grep '2798'

yang      2798  2.6 10.8 1864144 767000 ?      SLl  08:41  20:45
/opt/google/chrome/chrome

yang     21564  0.0  0.0  14536   948 pts/0    S+   21:52   0:00 grep --
color=auto --exclude-dir=.bzr --exclude-dir=CVS --exclude-dir=.git --exclude-
dir=.hg --exclude-dir=.svn 2798
```

3. 结束进程

```
kill PID

killall <进程名>

kill - <PID>
```

总结

整理完这篇总结，感觉溯源是一个很细节的事情，需要注意每一个细节，这篇总结也可以是一个备忘，以后在遇到溯源的活，做的时候就可以更系统一些。