第1篇：SSH暴力破解

0x00 前言

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。

0x01 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 -anplt|grep 22
tcp        0      0 127.0.0.1:2208              0.0.0.0:*                   LISTEN      3215/hpiod
tcp        0      0 192.168.143.112:22          111.13.[redacted].208:80     SYN_RECV    -
tcp        0      0 192.168.143.112:22          123.59.[redacted].31:80      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
tcp        0      0 ::ffff:127.0.0.1:54063      ::ffff:127.0.0.1:3306       ESTABLISHED 22952/java
tcp        0      0 ::ffff:192.168.143.112:22   ::ffff:192.168.143.24:33474 ESTABLISHED 21307/sshd: root@no
tcp        0     52 ::ffff:192.168.143.112:22   ::ffff:192.168.143.22:48373 ESTABLISHED 21652/1
```

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。

0x02 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./qK12HN
8jFuXOfEYIKLID6hq0::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。

0x04 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。

后续持续更新内容，将发布在公众号Bypass--，同时公众号提供了该项目的PDF版本，关注后回复"应急响应" 即可下载。