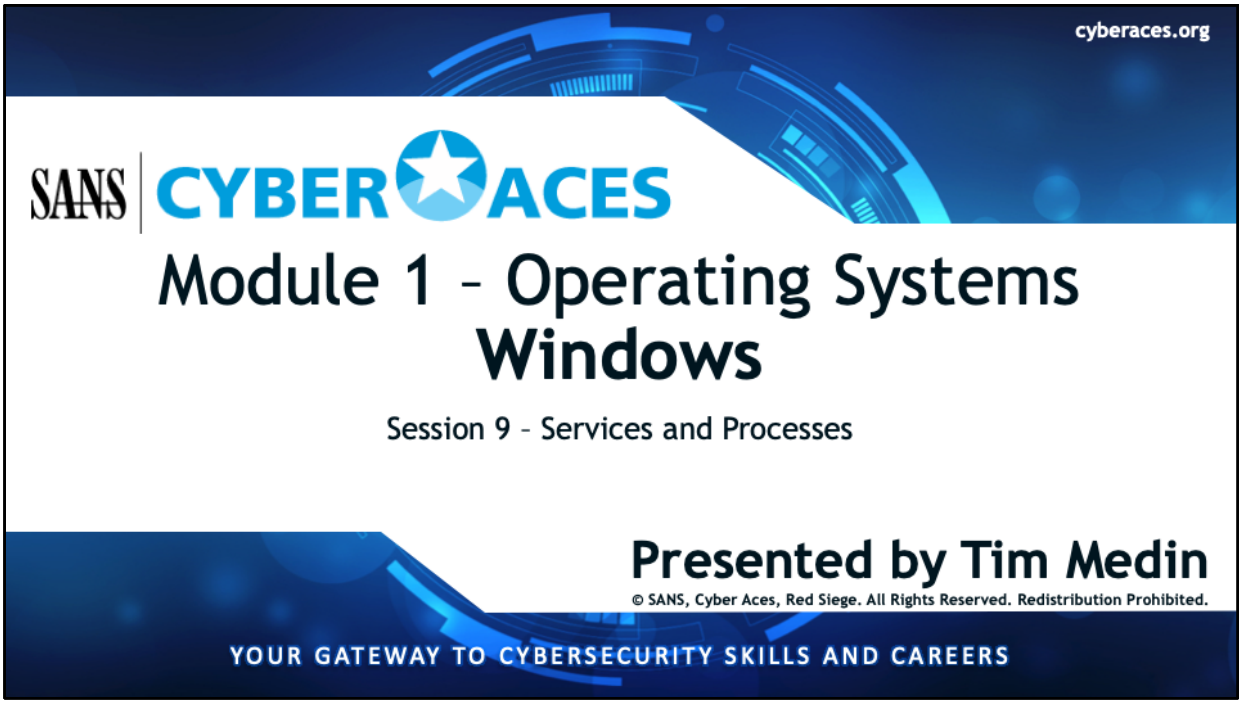


cyberaces.org

SANS | CYBER ACES

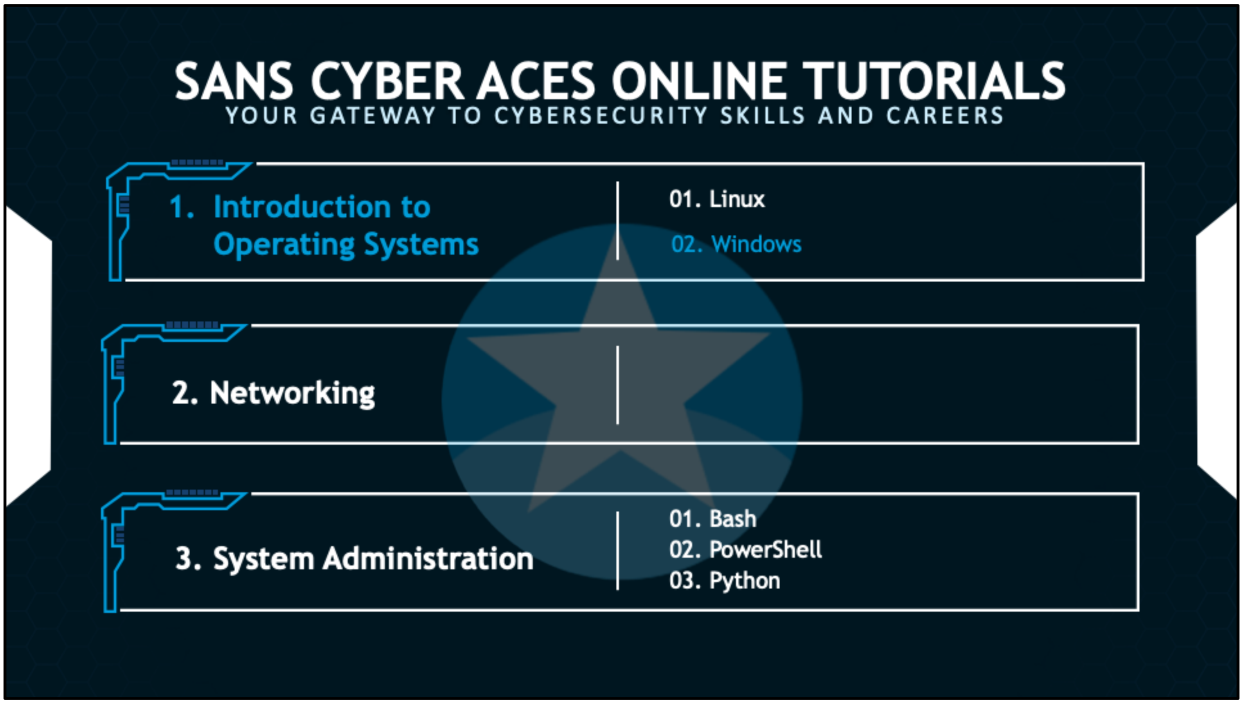
Module 1 – Operating Systems Windows

Session 9 – Services and Processes

Presented by Tim Medin

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

Welcome to Cyber Aces Online, Module 1! A firm understanding of operating systems is essential to being able to secure or attack one. This module dives in to Microsoft Windows Operating System.



SANS CYBER ACES ONLINE TUTORIALS

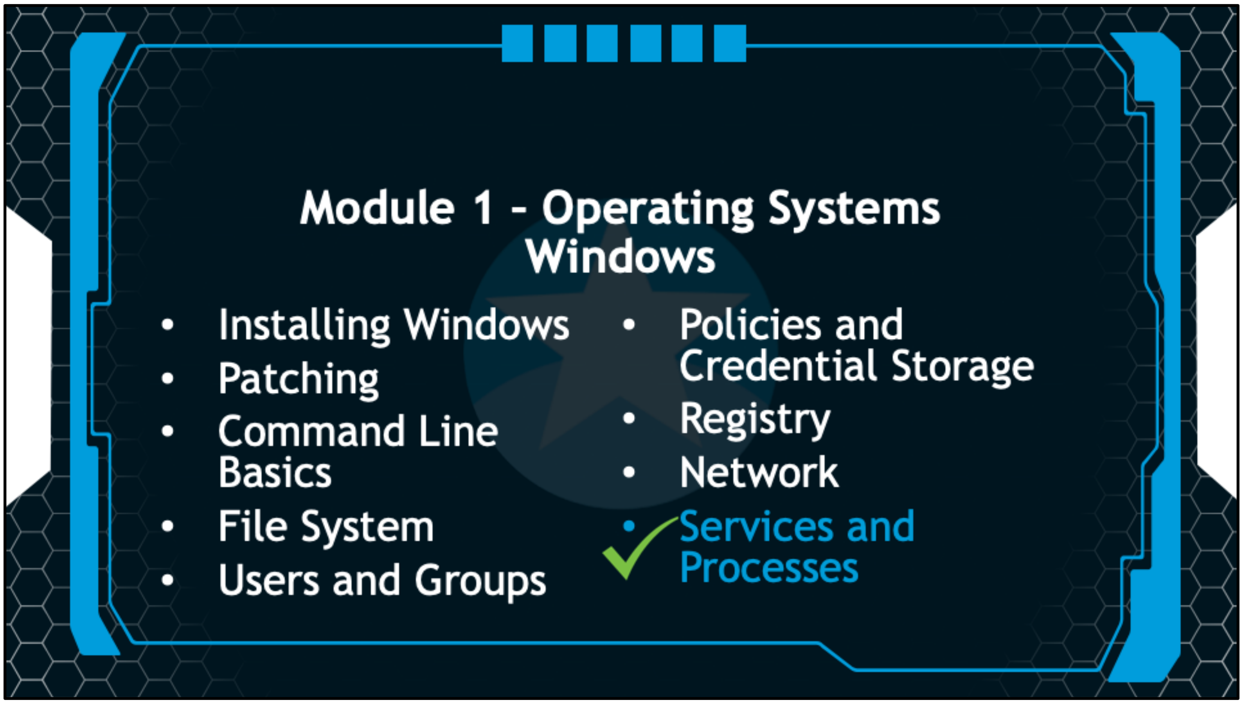
YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

1. Introduction to Operating Systems
   - 01. Linux
   - 02. Windows
2. Networking
3. System Administration
   - 01. Bash
   - 02. PowerShell
   - 03. Python

This training material was originally developed to help students, teachers, and mentors prepare for the Cyber Aces Online Competition. This module focuses on the basics of what an operating systems is as well as the two predominant OS's, Windows and Linux. This session is part of Module 1, Introduction to Operating Systems. This module is split into two sections, Linux and Windows. In this session, we will continue our examination of Windows.

The three modules of Cyber Aces Online are Operating Systems, Networking, and System Administration.

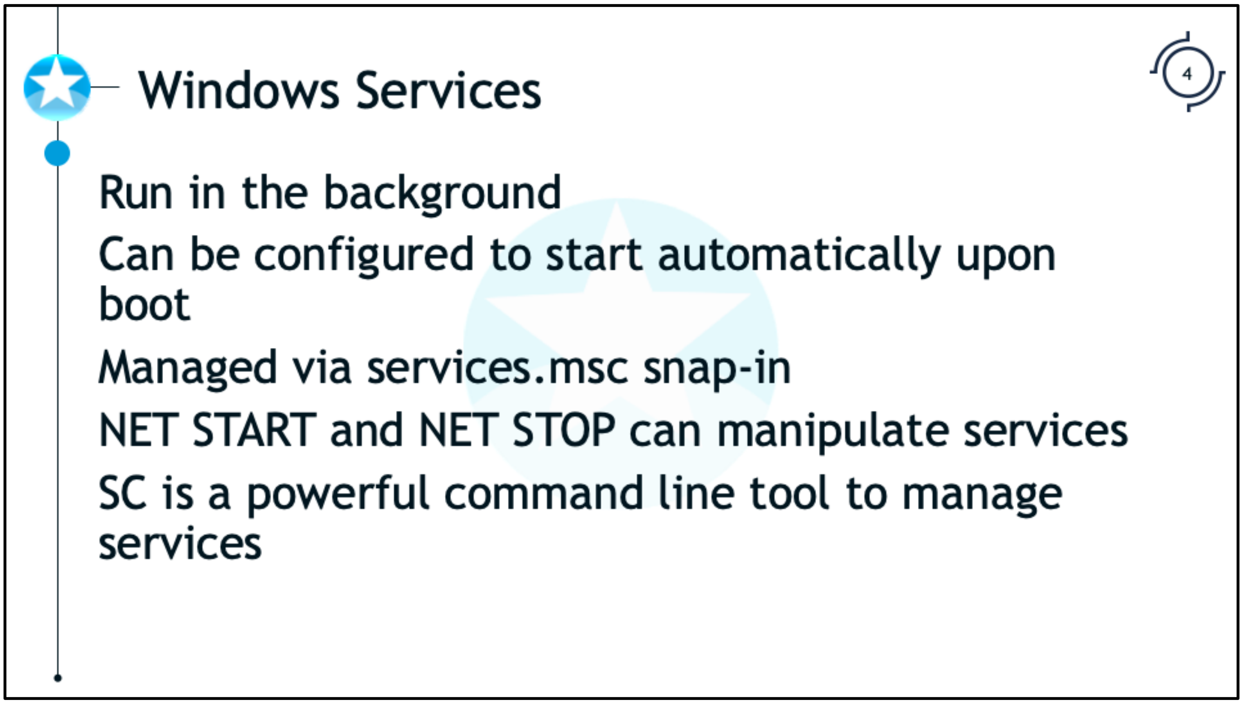
For more information about the Cyber Aces program, please visit the Cyber Aces website at https://CyberAces.org/.



Module 1 - Operating Systems Windows

- Installing Windows
- Patching
- Command Line Basics
- File System
- Users and Groups
- Policies and Credential Storage
- Registry
- Network
- ✓ Services and Processes

In this session we will discuss Windows services and processes.



Windows Services

- Run in the background
- Can be configured to start automatically upon boot
- Managed via services.msc snap-in
- NET START and NET STOP can manipulate services
- SC is a powerful command line tool to manage services

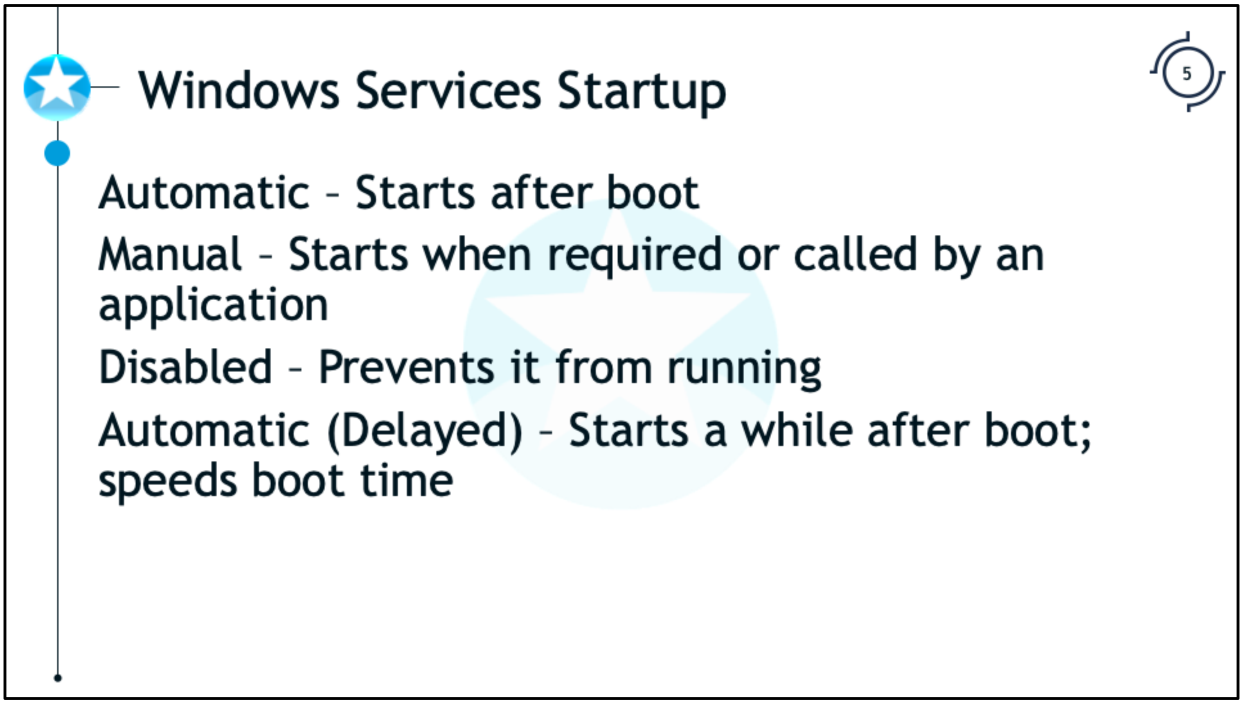
Software installed on the system is typically configured to run one of two ways: it can be executed as an interactive user process, or it can run in the background as a service. Services are run in the background and can be configured to start automatically after the system has booted. Services can be managed several ways. Most users manage services using the "SERVICES.MSC" MMC snap-in (Start -> Run, type "services.msc", hit enter). Services can also be started, stopped or queried using the NET command from the command line.

View all services: **`net start`**

Start the Print Spooler service: **`net start "print spooler"`**

Stop the Print Spooler service: **`net stop "print spooler"`**

But the most powerful interface to manage Windows services is the command line based Services Controller utility "SC.EXE"



Windows Services Startup

Automatic – Starts after boot

Manual – Starts when required or called by an application

Disabled – Prevents it from running

Automatic (Delayed) – Starts a while after boot; speeds boot time

Windows Services Startup

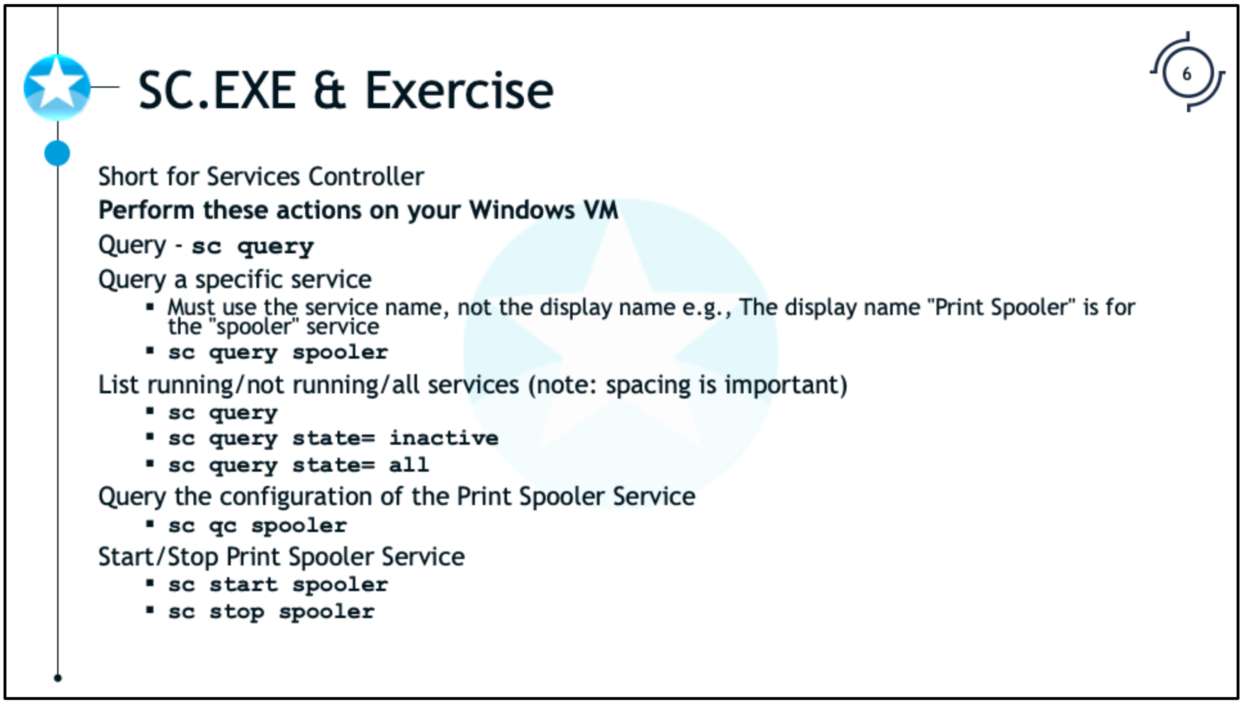
Windows Services can be set to various start-up modes, including preventing the service from starting at all. The startup options are:

Automatic – Starts after boot

Manual – Starts only when required or called by another service or application

Disabled – Will not run, even if another service attempts to start it

Automatic (Delayed) – Starts after boot is completed in order to prevent high load during boot. This option was added with Windows Vista.



SC.EXE & Exercise

Short for Services Controller

Perform these actions on your Windows VM

Query - **`sc query`**

Query a specific service

- Must use the service name, not the display name e.g., The display name "Print Spooler" is for the "spooler" service
- **`sc query spooler`**

List running/not running/all services (note: spacing is important)

- **`sc query`**
- **`sc query state= inactive`**
- **`sc query state= all`**

Query the configuration of the Print Spooler Service

- **`sc qc spooler`**

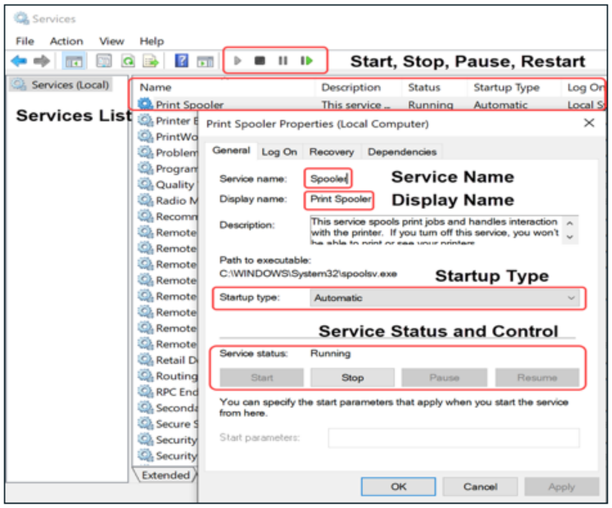
Start/Stop Print Spooler Service

- **`sc start spooler`**
- **`sc stop spooler`**

The SC command can be used to create, stop, start, query, modify or delete Windows services. For help on the command run: **`sc /?`**

When using the SC command, you must use the service name, which is different from the display name. The service name is typically shorter, all lower case, and contains no spaces. The service name can be found via "sc query" or the Services snap-in.

Services Snap-in



Name	Description	Status	Startup Type	Log On
Print Spooler	This service ...	Running	Automatic	Local S

Print Spooler Properties (Local Computer)

Service name: Spooler

Display name: Print Spooler

Description: This service spools print jobs and handles interaction with the printer. If you turn off this service, you won't be able to print or see your printers

Path to executable:
C:\WINDOWS\System32\spoolsv.exe

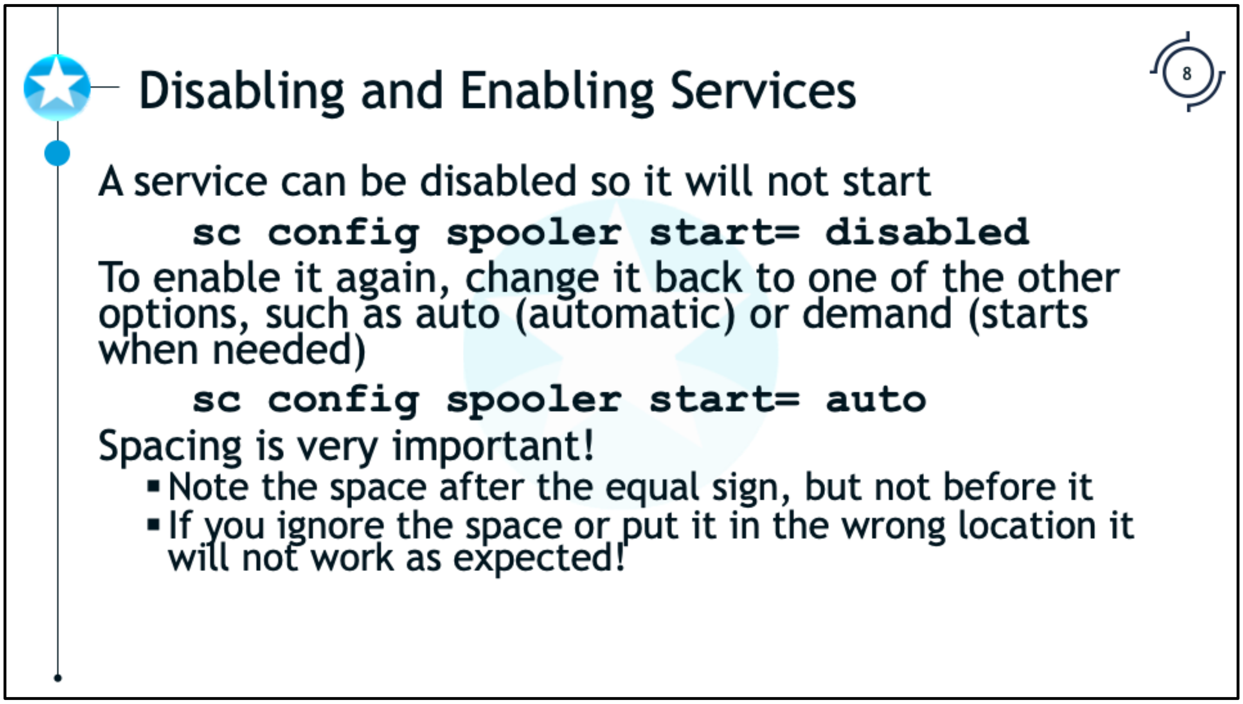
Startup type: Automatic

Service status: Running

You can specify the start parameters that apply when you start the service from here.

The Services Snap-in (Start -> Administrative Tools -> Services or Start -> Run, type "services.msc", hit enter) is the GUI front end for Windows services. The snap-in allows for services to be stopped, started, restarted (stop then start), or paused. This can be done by using the VCR controls shown in the menu, by right clicking on the service, or after double clicking on the service and using the control buttons.

When a service is opened (double click or enter key) the service name is visible. This name is important for command line interaction with the service.



Disabling and Enabling Services

A service can be disabled so it will not start

```
sc config spooler start= disabled
```

To enable it again, change it back to one of the other options, such as auto (automatic) or demand (starts when needed)

```
sc config spooler start= auto
```

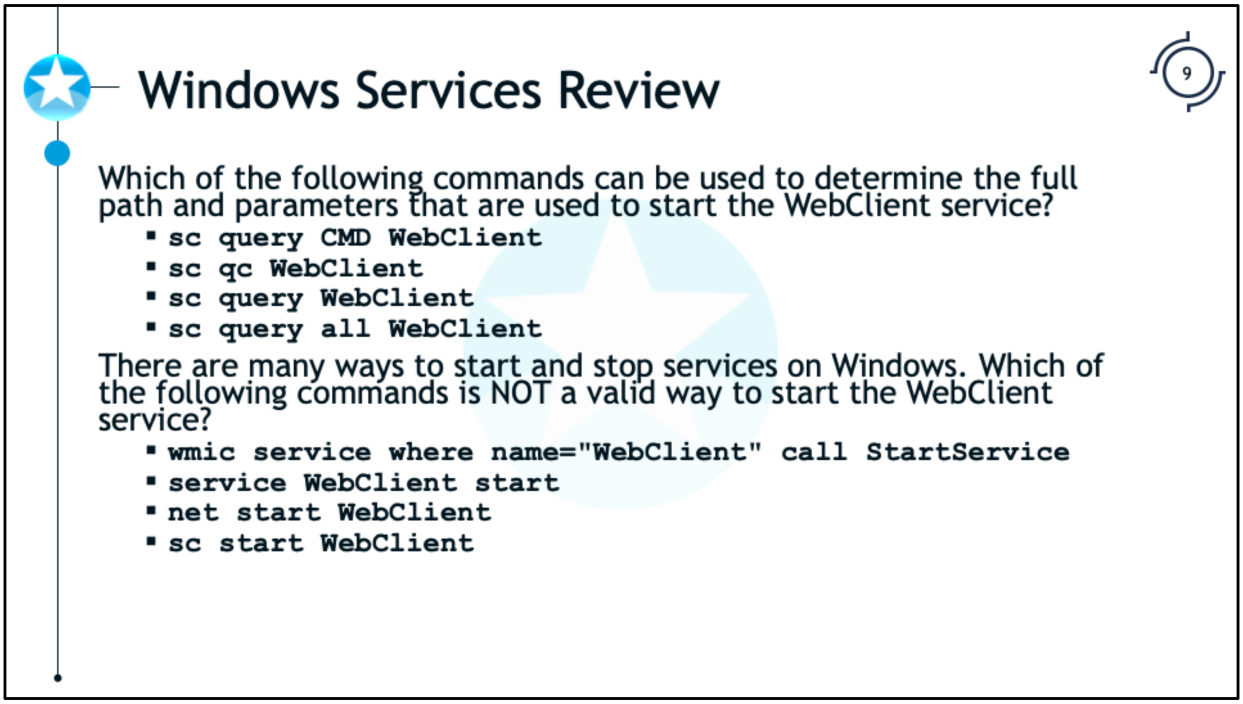
Spacing is very important!

- Note the space after the equal sign, but not before it
- If you ignore the space or put it in the wrong location it will not work as expected!

Microsoft offers full documentation on the command here:
https://www.redsiege.com/ca/sc

The “start” type for a service supports the following options:

- boot: A device driver that is loaded by the boot loader.
- system: A device driver that is started during kernel initialization.
- auto: A service that automatically starts each time the computer is restarted and runs even if no one logs on to the computer.
- demand: A service that must be manually started. This is the default value if start= is not specified.
- disabled: A service that cannot be started. To start a disabled service, change the start type to some other value.



Windows Services Review

Which of the following commands can be used to determine the full path and parameters that are used to start the WebClient service?

- `sc query CMD WebClient`
- `sc qc WebClient`
- `sc query WebClient`
- `sc query all WebClient`

There are many ways to start and stop services on Windows. Which of the following commands is NOT a valid way to start the WebClient service?

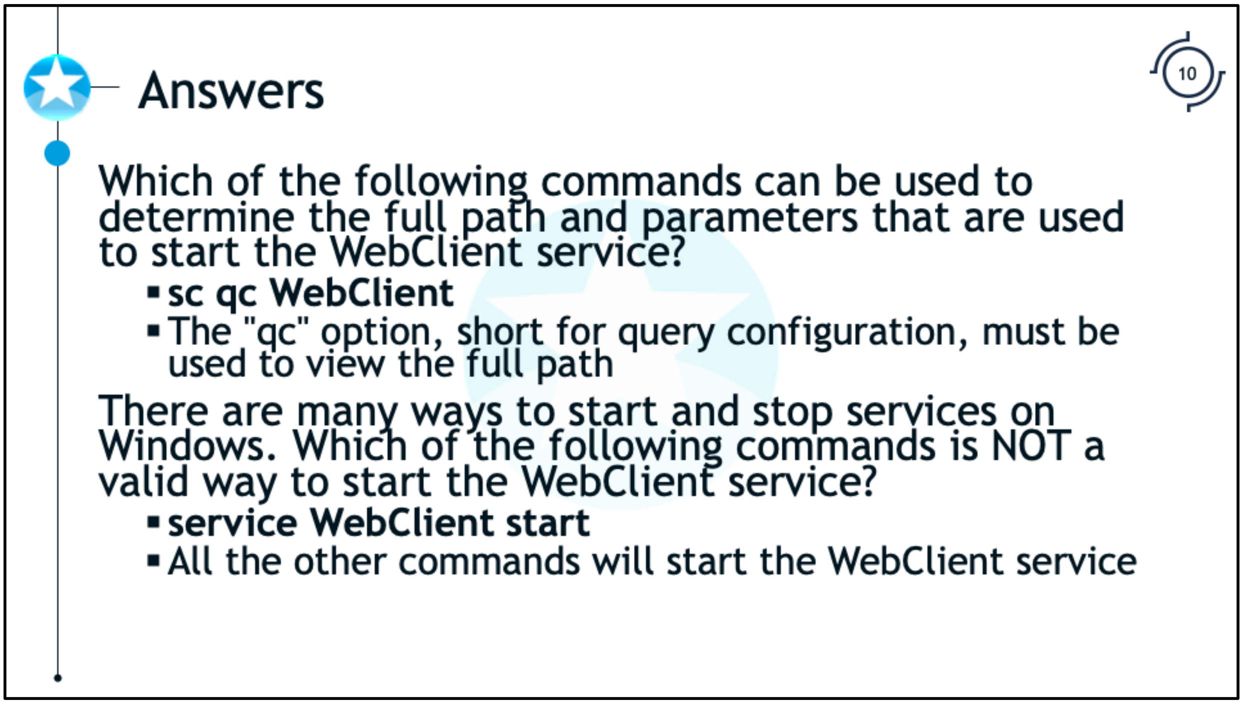
- `wmic service where name="WebClient" call StartService`
- `service WebClient start`
- `net start WebClient`
- `sc start WebClient`

Which of the following commands can be used to determine the full path and parameters that are used to start the WebClient service?

```
sc query CMD WebClient
sc qc WebClient
sc query WebClient
sc query all WebClient
```

There are many ways to start and stop services on Windows. Which of the following commands is NOT a valid way to start the WebClient service?

```
wmic service where name="WebClient" call
StartService
service WebClient start
net start WebClient
sc start WebClient
```



Answers

Which of the following commands can be used to determine the full path and parameters that are used to start the WebClient service?

- **sc qc WebClient**
- The "qc" option, short for query configuration, must be used to view the full path

There are many ways to start and stop services on Windows. Which of the following commands is NOT a valid way to start the WebClient service?

- **service WebClient start**
- All the other commands will start the WebClient service

Which of the following commands can be used to determine the full path and parameters that are used to start the WebClient service?

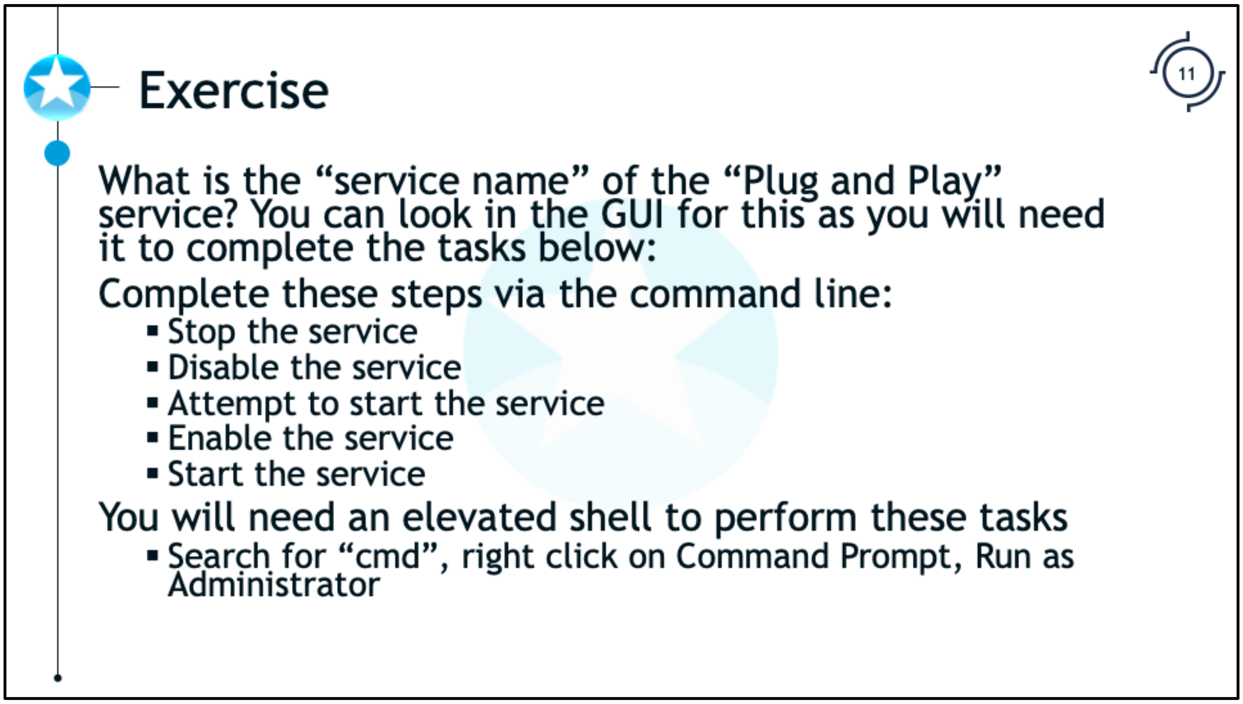
sc qc WebClient

The "qc" option, short for query configuration, must be used to view the full path

There are many ways to start and stop services on Windows. Which of the following commands is NOT a valid way to start the WebClient service?

service WebClient start

All the other commands will start the WebClient service



Exercise

What is the “service name” of the “Plug and Play” service? You can look in the GUI for this as you will need it to complete the tasks below:

Complete these steps via the command line:

- Stop the service
- Disable the service
- Attempt to start the service
- Enable the service
- Start the service

You will need an elevated shell to perform these tasks

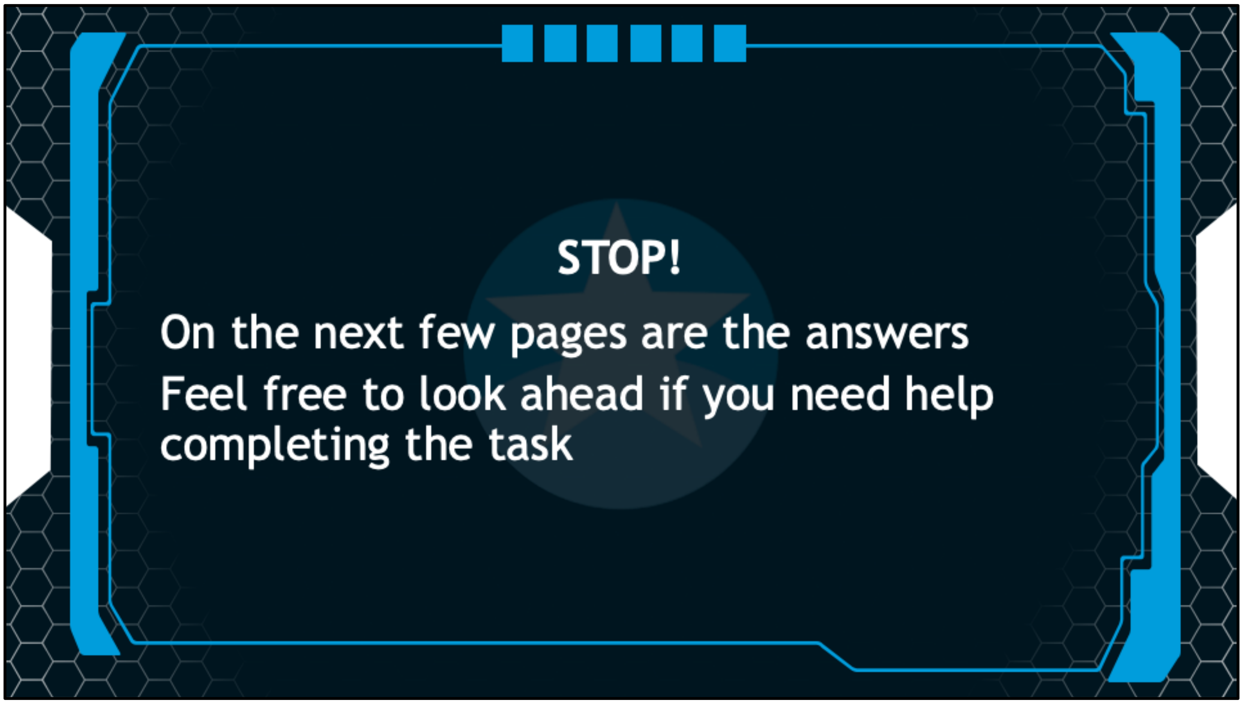
- Search for “cmd”, right click on Command Prompt, Run as Administrator

What is the “service name” of the “Plug and Play” service? You can look in the GUI for this as you will need it to complete the tasks below:

Complete these steps via the command line:

Stop the service
Disable the service
Attempt to start the service
Enable the service
Start the service

You will need an elevated shell to perform these tasks. Search for “cmd”, right click on Command Prompt, Run as Administrator.



STOP!

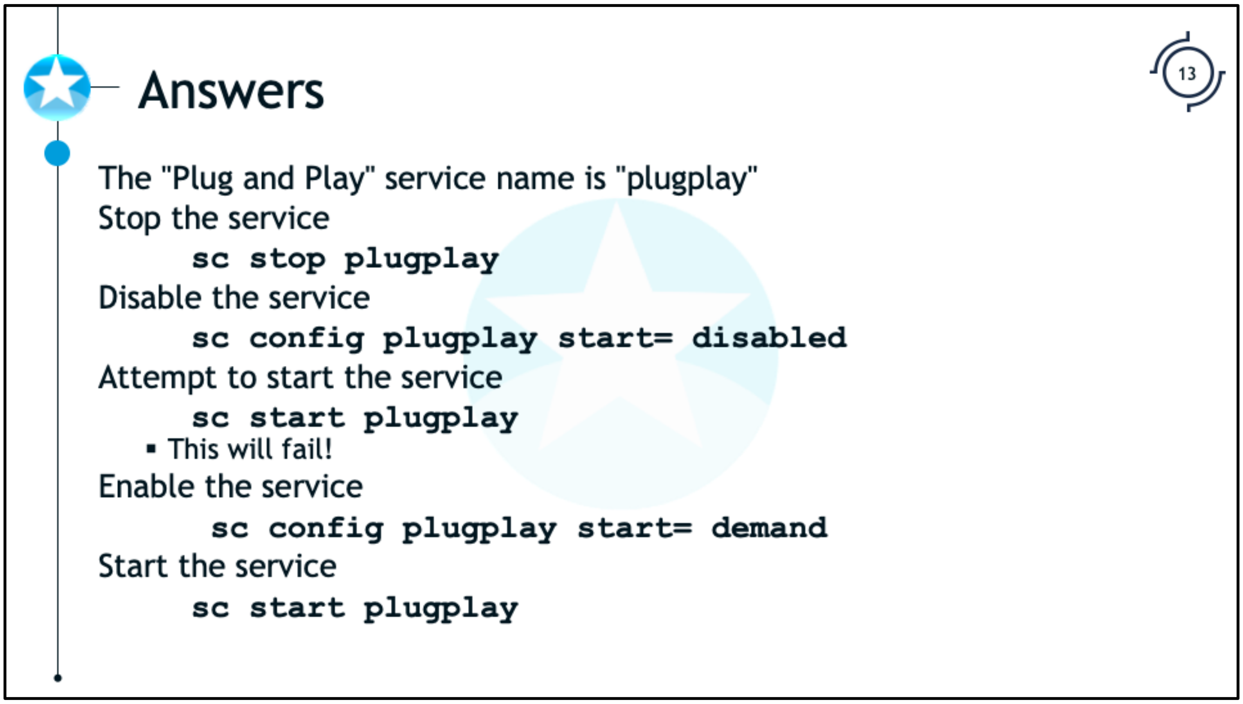
On the next few pages are the answers

Feel free to look ahead if you need help completing the task

STOP!

On the next few pages are the answers

Feel free to look ahead if you need help completing the task



Answers

The "Plug and Play" service name is "plugplay"

Stop the service

```
sc stop plugplay
```

Disable the service

```
sc config plugplay start= disabled
```

Attempt to start the service

```
sc start plugplay
```

- This will fail!

Enable the service

```
sc config plugplay start= demand
```

Start the service

```
sc start plugplay
```

The "Plug and Play" service name is "plugplay".

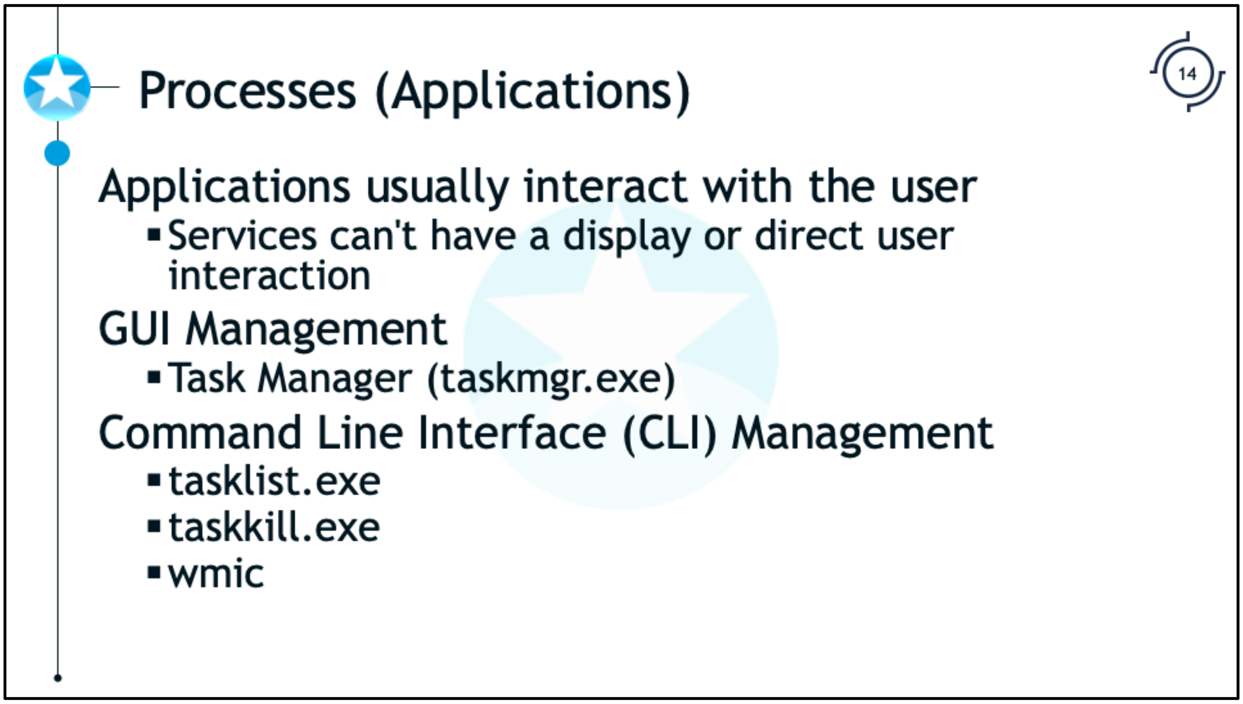
Stop the service: **`sc stop plugplay`**

Disable the service: **`sc config plugplay start= disabled`**

Attempt to start the service (will fail since it is disabled): **`sc start plugplay`**

Enable the service: **`sc config plugplay start= demand`**

Start the service: **`sc start plugplay`**

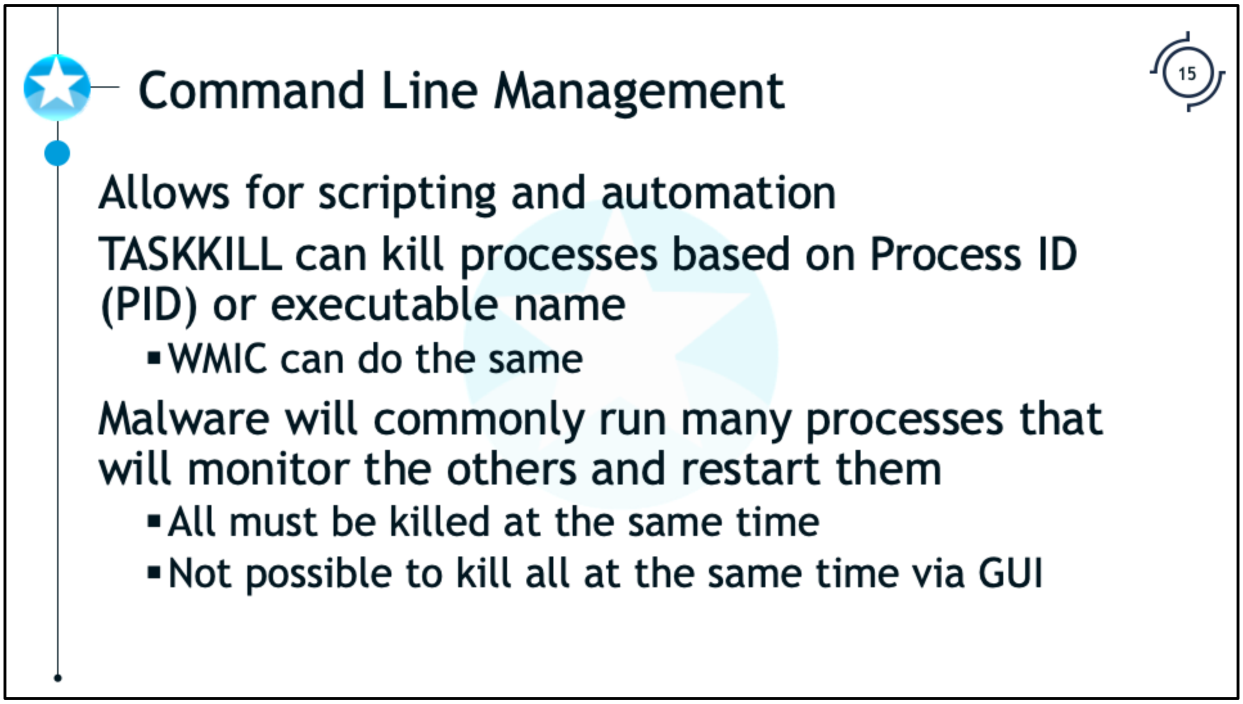


Processes (Applications)

- Applications usually interact with the user
  - Services can't have a display or direct user interaction
- GUI Management
  - Task Manager (taskmgr.exe)
- Command Line Interface (CLI) Management
  - tasklist.exe
  - taskkill.exe
  - wmic

Unlike services, Applications usually interact with the user, but they don't have to run visibly on the screen or show up on the start bar. Windows provides the user with several ways to manage running applications. From the GUI, users can use TASK MANAGER to monitor, start and kill applications. At the command line, TASKLIST.EXE and its partner TASKKILL.EXE can monitor or kill tasks. WMIC, a command line power-house, also boasts the ability to control processes among its many other uses.

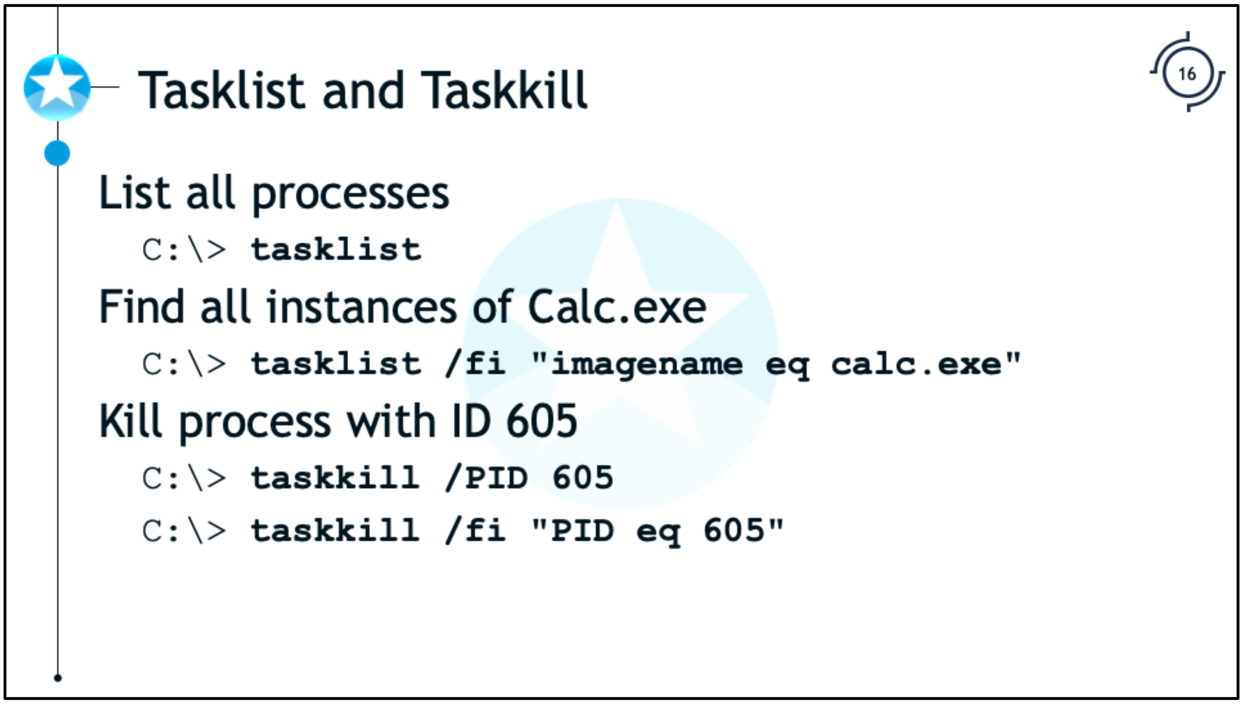
The tasklist and taskkill commands are mentioned earlier in the basic commands section.



Command Line Management

- Allows for scripting and automation
- TASKKILL can kill processes based on Process ID (PID) or executable name
  - WMIC can do the same
- Malware will commonly run many processes that will monitor the others and restart them
  - All must be killed at the same time
  - Not possible to kill all at the same time via GUI

From the command line, you have several options for controlling running process lists. TASKKILL.EXE can kill applications based upon their Process ID Number (PID) or the name of the executable. "WMIC" can also be used to manage tasks from the command line. One significant advantage to using the command line version of tools is that they can be scripted and run quickly. It is not uncommon for malicious code to launch several copies of malicious processes. Those processes monitor the other processes to make sure they are still running. If any of the malicious processes notices the other process has stopped, they re-launch the process. So to kill all the copies of the malicious code, you have to kill them all at the same time. This is impossible to do using the GUI based Task Manager.



Tasklist and Taskkill

List all processes

```
C:\> tasklist
```

Find all instances of Calc.exe

```
C:\> tasklist /fi "imagename eq calc.exe"
```

Kill process with ID 605

```
C:\> taskkill /PID 605
C:\> taskkill /fi "PID eq 605"
```

The Tasklist command (without any additional options) will list the processes that are running on the system. The tasklist command can be used to look for a specific proceses, by name

```
C:\> tasklist /fi "imagename eq calc.exe"
```

Or by process ID

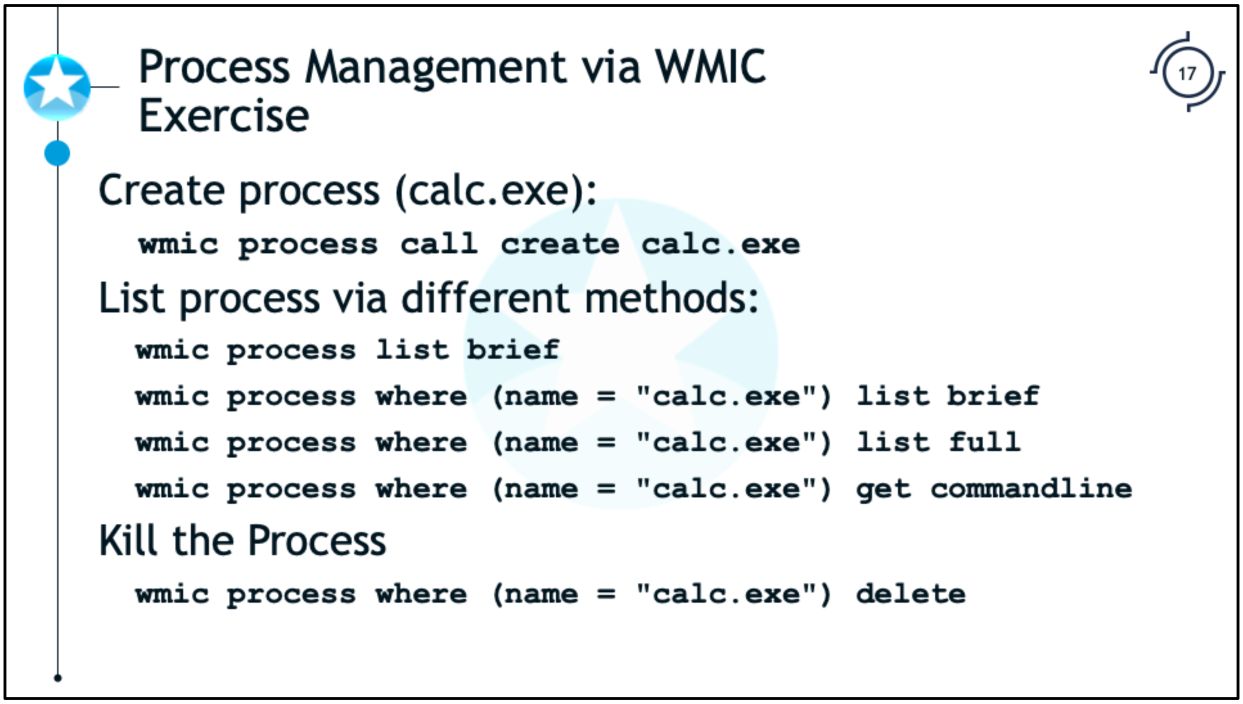
```
C:\> tasklist /fi "pid eq 3088"
```

The taskkill command can kill processes based on the process ID (PID), name, and other criteria (see the help page for more details).

```
C:\> taskkill /PID 605
C:\> tasklist /fi "pid eq 3088"
```

To loop and kill processes based on user, name, executable path and other methods check out:

https://redsiege.com/ca/tasks



Process Management via WMIC Exercise

Create process (calc.exe):

```
wmic process call create calc.exe
```

List process via different methods:

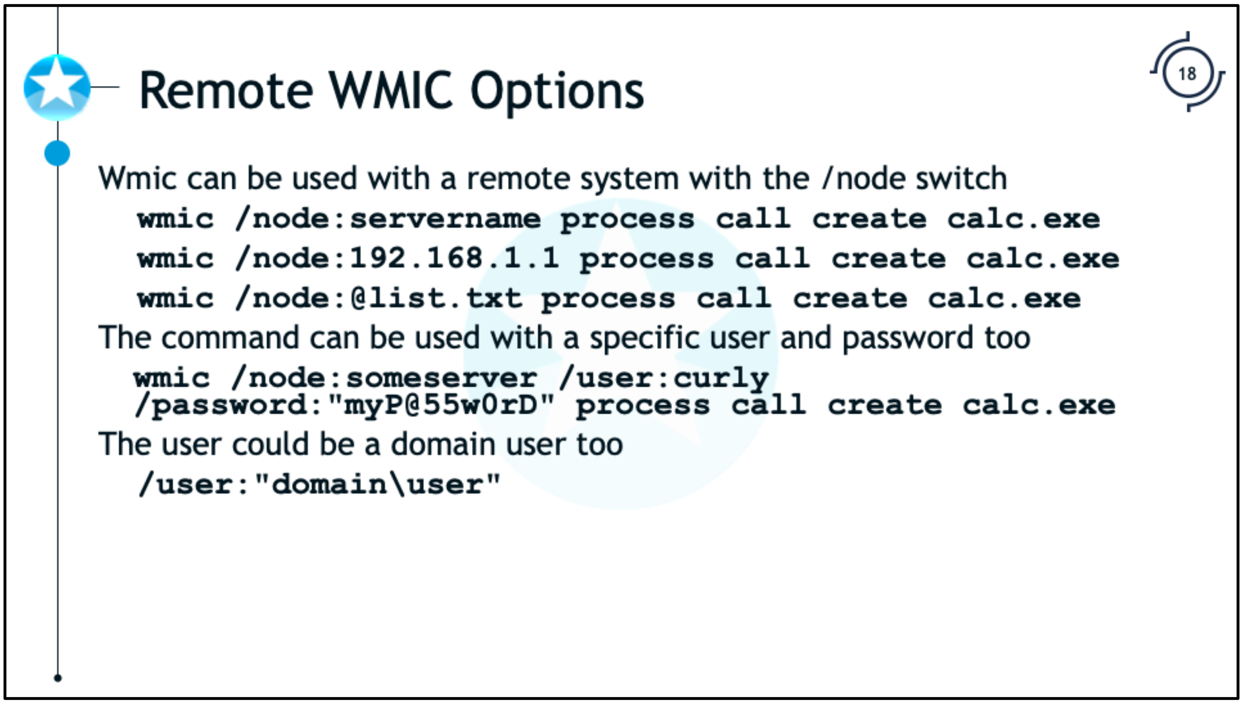
```
wmic process list brief
wmic process where (name = "calc.exe") list brief
wmic process where (name = "calc.exe") list full
wmic process where (name = "calc.exe") get commandline
```

Kill the Process

```
wmic process where (name = "calc.exe") delete
```

When it comes to managing processes from the command line, TASKLIST is not the only sheriff in town. Processes, like most aspects of the Windows Operating System, can also be controlled at the command line with the WMIC command. In this section, we will focus on using WMIC to manage processes, but WMIC is a very powerful tool and can do much much more.

WMIC Intro Guide: https://redsiege.com/ca/wmic



Remote WMIC Options

Wmic can be used with a remote system with the /node switch

```
wmic /node:servername process call create calc.exe
wmic /node:192.168.1.1 process call create calc.exe
wmic /node:@list.txt process call create calc.exe
```

The command can be used with a specific user and password too

```
wmic /node:someserver /user:curly
/password:"myP@55w0rD" process call create calc.exe
```

The user could be a domain user too

```
/user:"domain\user"
```

The WMIC command can be used to run commands on remote systems. It requires the correct permissions and the appropriate credentials. The /node switch can be used to connect to a remote system. You can specify the remote host by name:

```
C:\> wmic /node:servername process call create calc.exe
```

By IP address:

```
C:\> wmic /node:4.5.6.7 process call create calc.exe
```

Or a list of IP addresses and/or names in a text file:

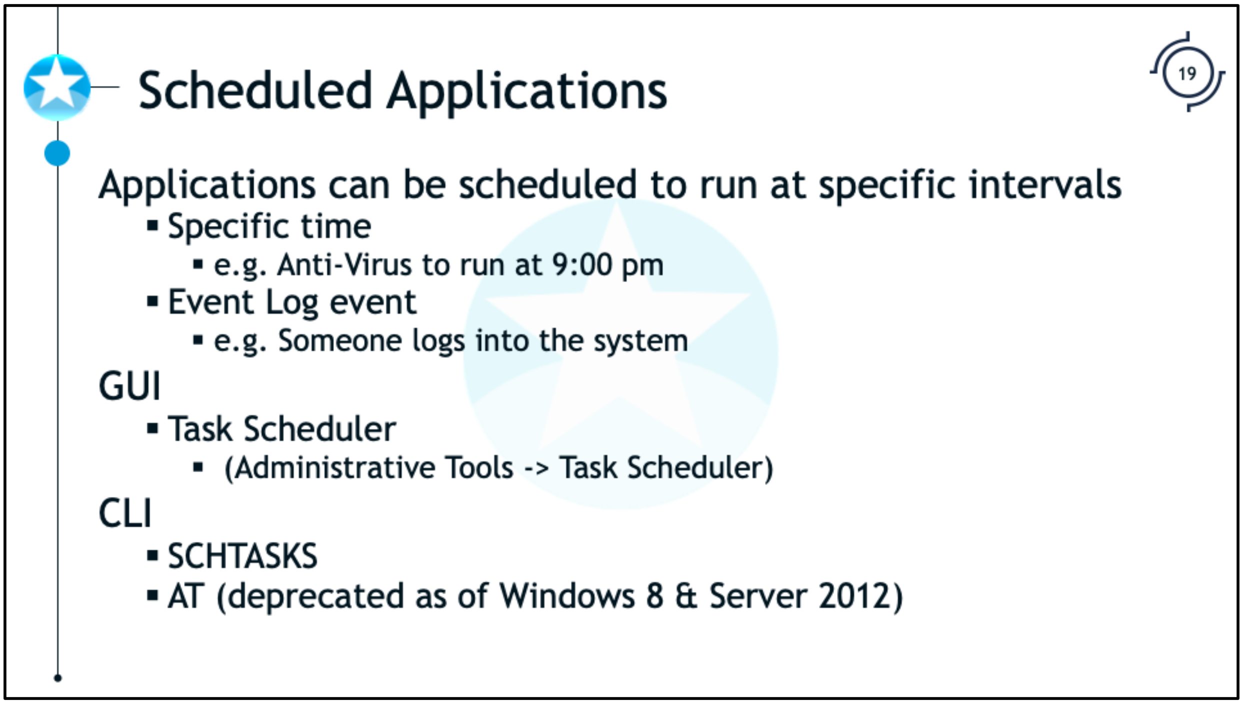
```
C:\> wmic /node:@list.txt process call create calc.exe
```

These commands will authenticate to the remote system as the currently logged in user. You can specify a specific user and password with the /user and /password switches respectively:

```
C:\> wmic /node:someserver /user:curly
/password:"myP@55w0rD" process call create calc.exe
```

The user can be a domain user by specifying the domain as part of the username:

```
C:\> wmic /node:someserver /user:mydomain\curly
/password:"myP@55w0rD" process call create calc.exe
```



Scheduled Applications

Applications can be scheduled to run at specific intervals

- Specific time
  - e.g. Anti-Virus to run at 9:00 pm
- Event Log event
  - e.g. Someone logs into the system

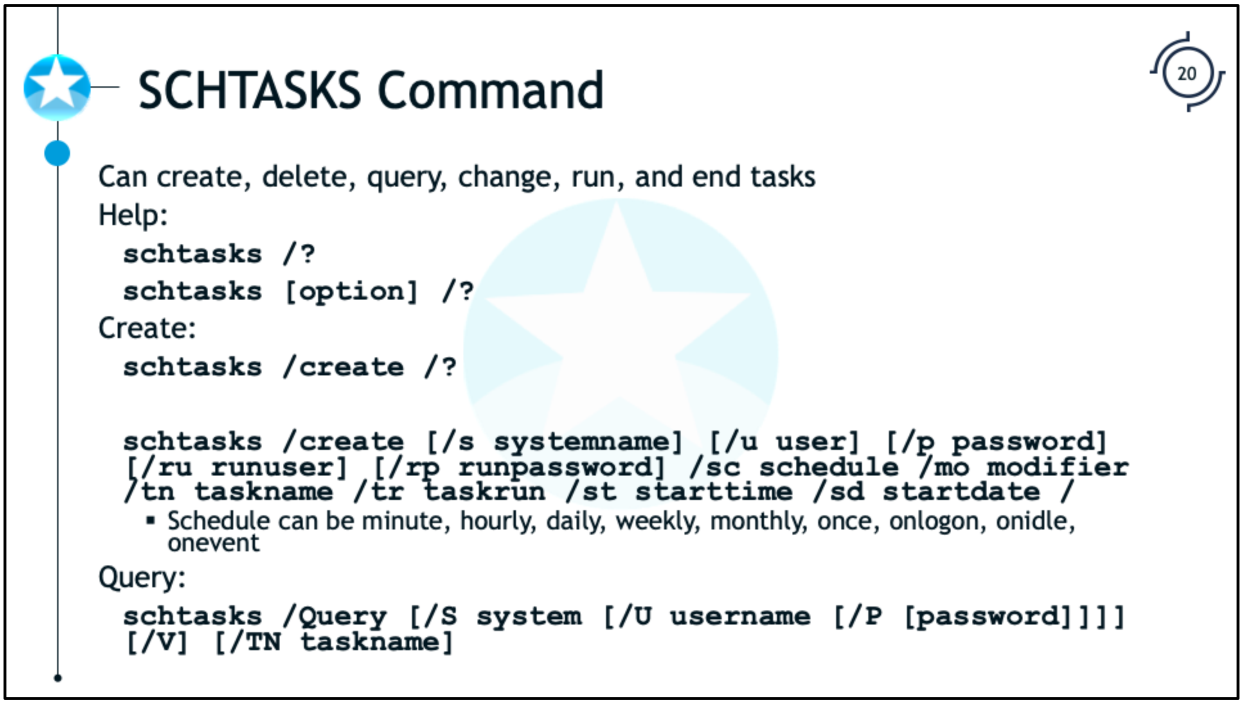
GUI

- Task Scheduler
  - (Administrative Tools -> Task Scheduler)

CLI

- SCHTASKS
- AT (deprecated as of Windows 8 & Server 2012)

Applications can be run interactively by the user, but they can also be scheduled to run at specific intervals. Scheduled tasks can be triggered by the date and time or by events that occur in the operating system. For example, you can schedule your antivirus software to scan your computer every night at 9:00 pm. Or, you could setup a scheduled task to send you an email every time the EventLog records someone has logged into the system. Tasks can be scheduled through the GUI using the "Task Scheduler" which is located in "Control Panel" → "Administrative Tools" → "Task Scheduler". You can also manage tasks through "SCHTASKS" and, before Windows 8, the "AT" command.



SCHTASKS Command

Can create, delete, query, change, run, and end tasks

Help:

```
schtasks /?
schtasks [option] /?
```

Create:

```
schtasks /create /?


schtasks /create [/s systemname] [/u user] [/p password]
[/ru runuser] [/rp runpassword] /sc schedule /mo modifier
/tn taskname /tr taskrun /st starttime /sd startdate /
```

- Schedule can be minute, hourly, daily, weekly, monthly, once, onlogon, onidle, onevent

Query:

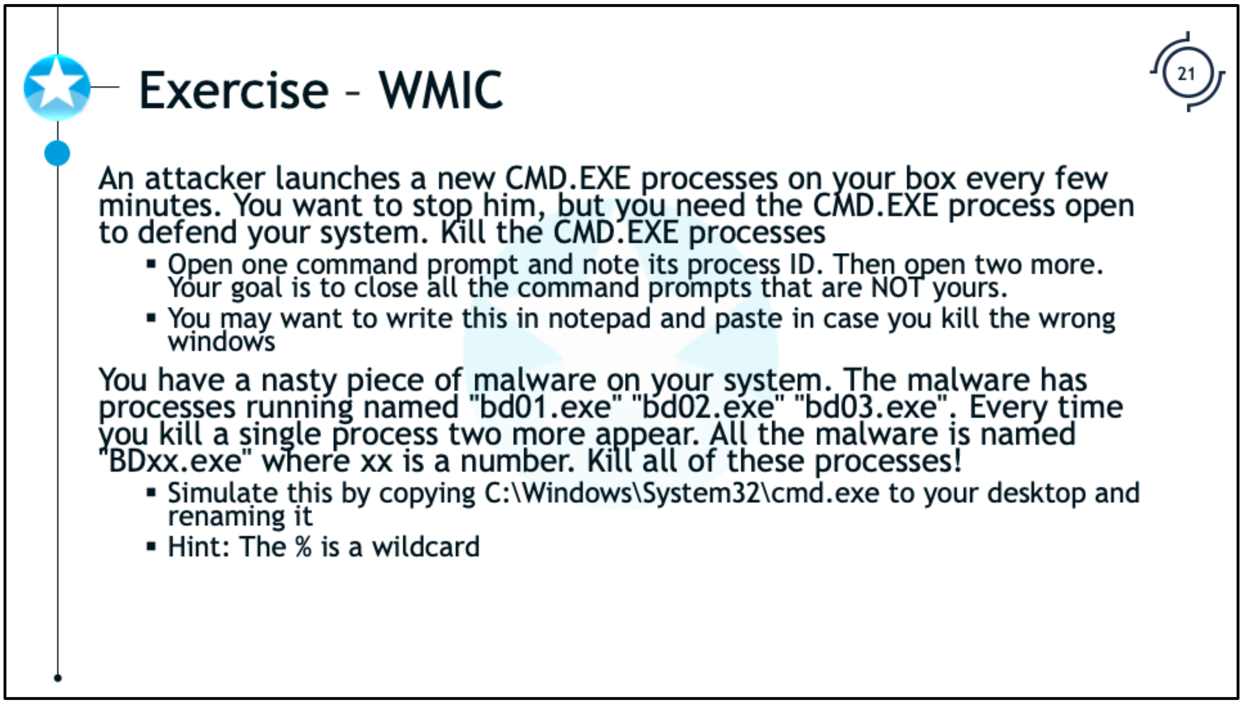
```
schtasks /Query [/S system [/U username [/P [password]]]]
[/V] [/TN taskname]
```

Enables an administrator to create, delete, query, change, run, and end scheduled tasks on a local or remote computer. Running Schtasks.exe without arguments displays the status and next run time for each registered task.

The most common available options for schtasks are:

- /Run – run the specified task
- /End – stops a currently running task
- /Create – create a new scheduled task
- /Delete – delete an existing task
- /Query – display all scheduled tasks
- /Change – modify and existing scheduled task

Creating a task allows us to specify the user/password to create the task as, as well as credentials the task should run under. The options also allow us to schedule more granularly than the AT command. See the help page for additional details on scheduling.



Exercise – WMIC

An attacker launches a new CMD.EXE processes on your box every few minutes. You want to stop him, but you need the CMD.EXE process open to defend your system. Kill the CMD.EXE processes

- Open one command prompt and note its process ID. Then open two more. Your goal is to close all the command prompts that are NOT yours.
- You may want to write this in notepad and paste in case you kill the wrong windows

You have a nasty piece of malware on your system. The malware has processes running named "bd01.exe" "bd02.exe" "bd03.exe". Every time you kill a single process two more appear. All the malware is named "BDxx.exe" where xx is a number. Kill all of these processes!

- Simulate this by copying C:\Windows\System32\cmd.exe to your desktop and renaming it
- Hint: The % is a wildcard

An attacker launches a new CMD.EXE processes on your box every few minutes. You want to stop him, but you need the CMD.EXE process open to defend your system. Kill the CMD.EXE processes

Open one command prompt and note its process ID. Then open two more.

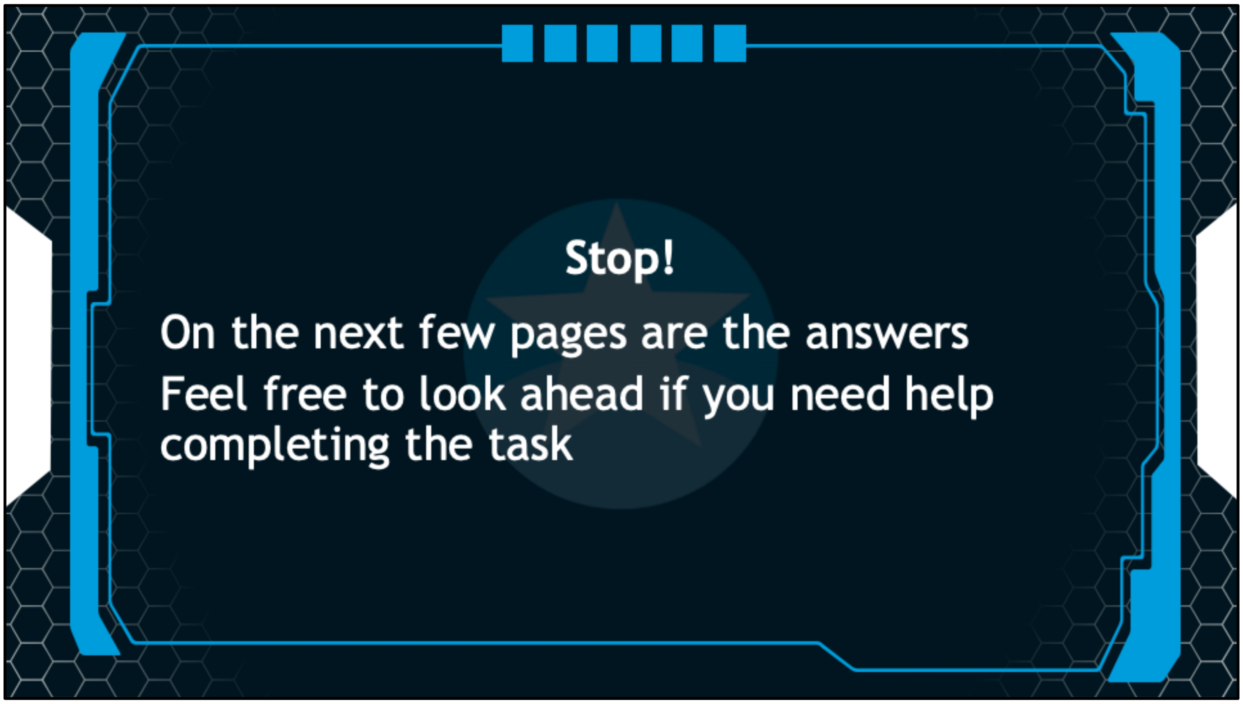
Your goal is to close all the command prompts that are NOT yours.

You may want to write this in notepad and paste in case you kill the wrong windows

You have a nasty piece of malware on your system. The malware has processes running named "bd01.exe" "bd02.exe" "bd03.exe". Every time you kill a single process two more appear. All the malware is named "BDxx.exe" where xx is a number. Kill all of these processes!

Simulate this by copying C:\Windows\System32\cmd.exe to your desktop and renaming it

Hint: The % is a wildcard



Stop!

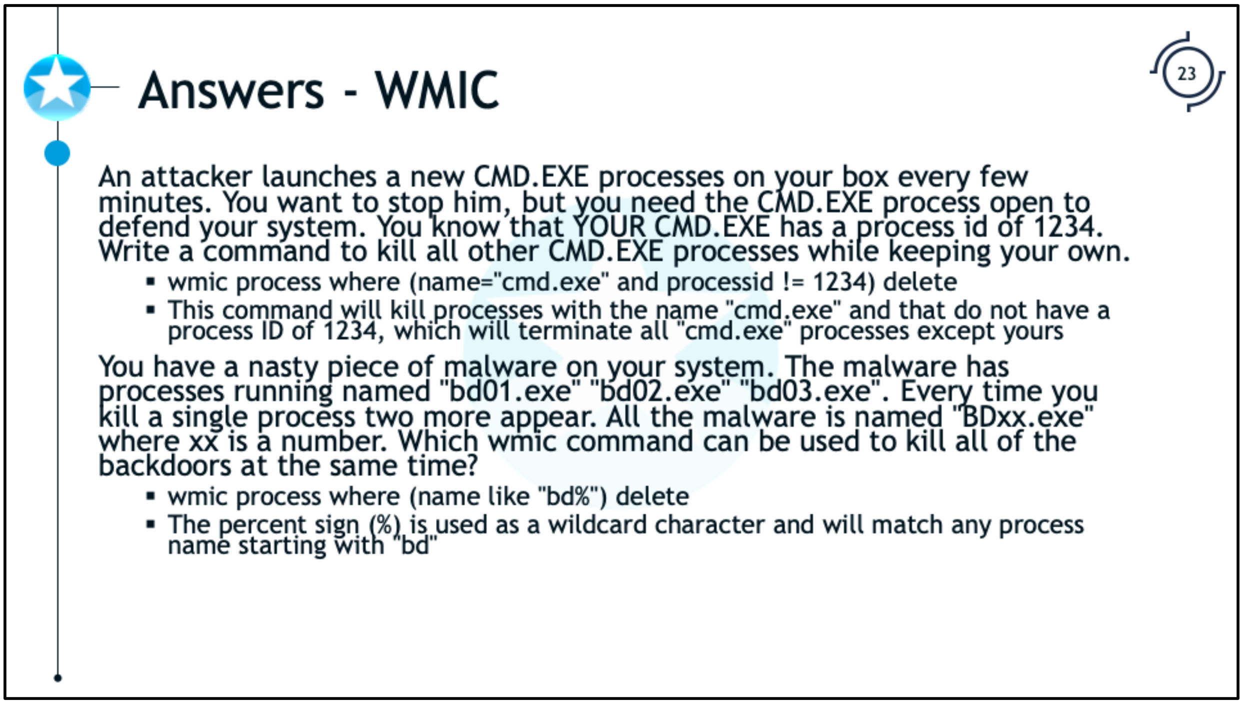
On the next few pages are the answers

Feel free to look ahead if you need help completing the task

STOP!

On the next few pages are the answers

Feel free to look ahead if you need help completing the task



Answers - WMIC

An attacker launches a new CMD.EXE processes on your box every few minutes. You want to stop him, but you need the CMD.EXE process open to defend your system. You know that YOUR CMD.EXE has a process id of 1234. Write a command to kill all other CMD.EXE processes while keeping your own.

- wmic process where (name="cmd.exe" and processid != 1234) delete
- This command will kill processes with the name "cmd.exe" and that do not have a process ID of 1234, which will terminate all "cmd.exe" processes except yours

You have a nasty piece of malware on your system. The malware has processes running named "bd01.exe" "bd02.exe" "bd03.exe". Every time you kill a single process two more appear. All the malware is named "BDxx.exe" where xx is a number. Which wmic command can be used to kill all of the backdoors at the same time?

- wmic process where (name like "bd%") delete
- The percent sign (%) is used as a wildcard character and will match any process name starting with "bd"

An attacker launches a new CMD.EXE processes on your box every few minutes. You want to stop him, but you need the CMD.EXE process open to defend your system. You know that YOUR CMD.EXE has a process id of 1234. Write a command to kill all other CMD.EXE processes while keeping your own.

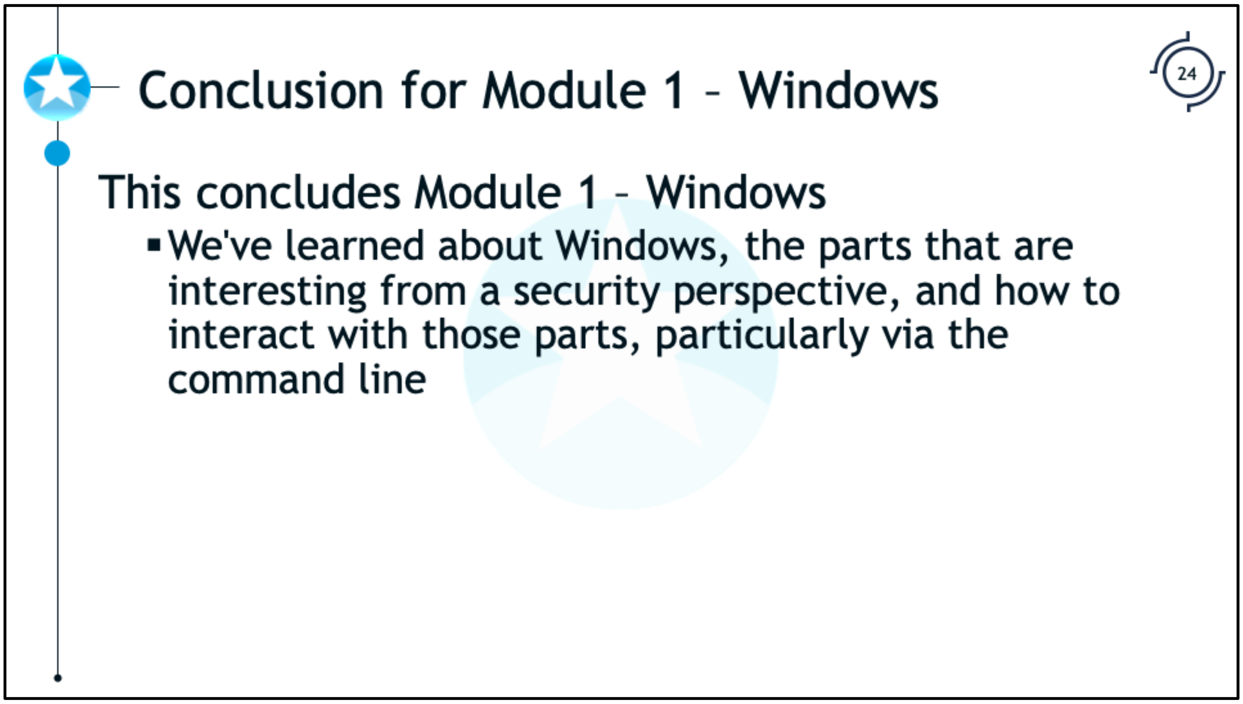
```
wmic process where (name="cmd.exe" and processid !=
1234) delete
```

This command will kill processes with the name "cmd.exe" and that do not have a process ID of 1234, which will terminate all "cmd.exe" processes except yours

You have a nasty piece of malware on your system. The malware has processes running named "bd01.exe" "bd02.exe" "bd03.exe". Every time you kill a single process two more appear. All the malware is named "BDxx.exe" where xx is a number. Which wmic command can be used to kill all of the backdoors at the same time?

```
wmic process where (name like "bd%") delete
```

The percent sign (%) is used as a wildcard character and will match any process name starting with "bd"



Conclusion for Module 1 – Windows

This concludes Module 1 – Windows

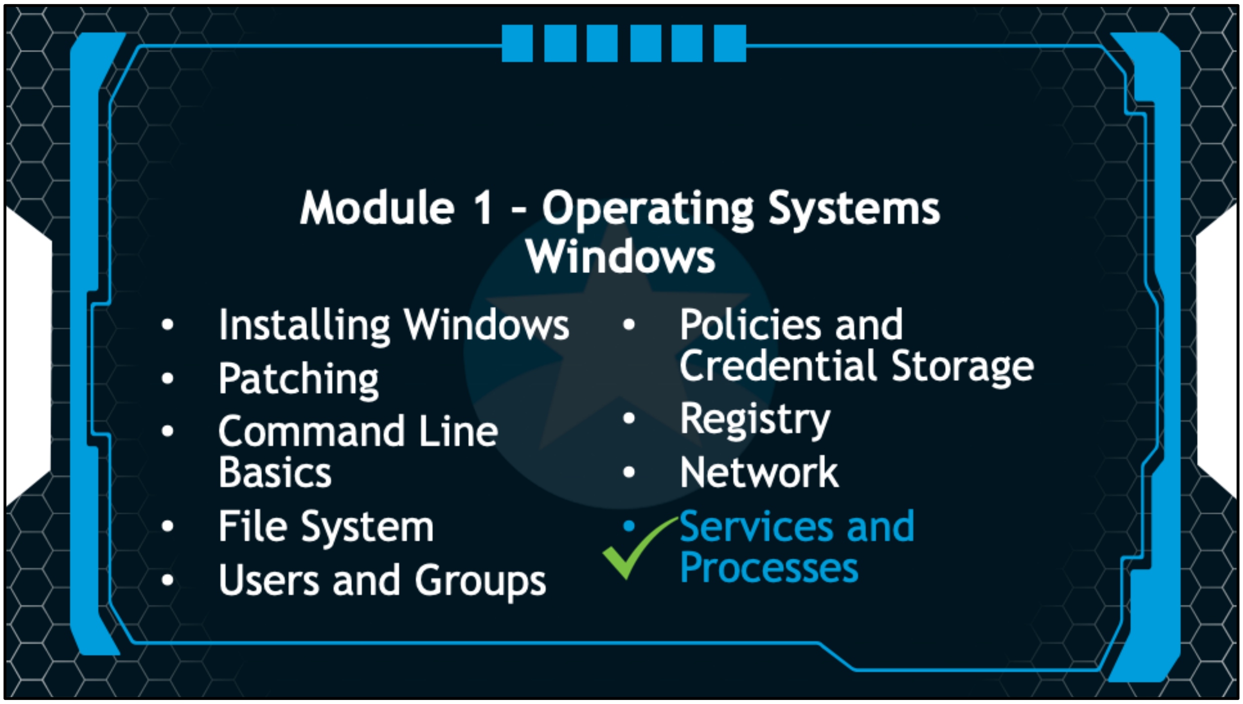
- We've learned about Windows, the parts that are interesting from a security perspective, and how to interact with those parts, particularly via the command line

The Windows Operating system is often in the cross hairs of today's attackers. Microsoft has responded by adding a large number of security features to the operating system and we have only briefly touched on a few of them here. Here are just a few additional references for your exploration:

https://www.redsiege.com/ca/mssecurity

https://www.redsiege.com/ca/sanssecurity

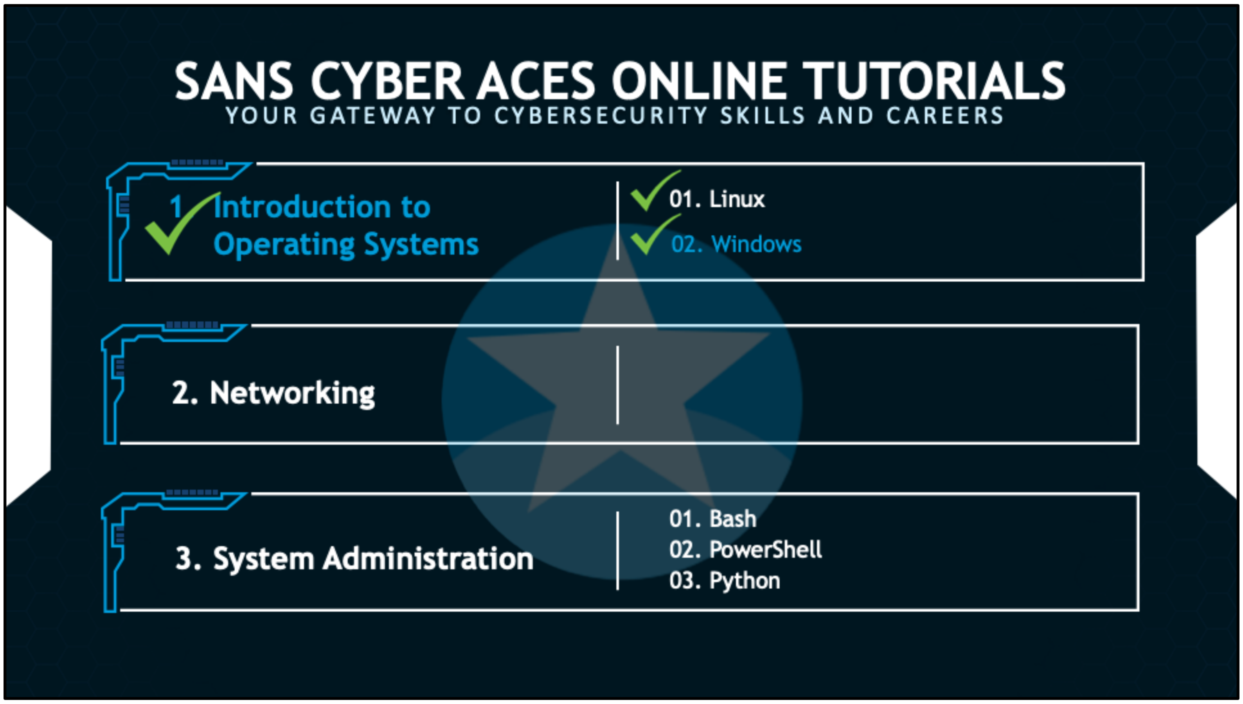
https://www.redsiege.com/ca/darknetsecurity



Module 1 - Operating Systems Windows

- Installing Windows
- Patching
- Command Line Basics
- File System
- Users and Groups
- Policies and Credential Storage
- Registry
- Network
- ✓ Services and Processes

You have successfully completed the session on Windows services and processes.



SANS CYBER ACES ONLINE TUTORIALS

YOUR GATEWAY TO CYBERSECURITY SKILLS AND CAREERS

✓ 1. Introduction to Operating Systems

- ✓ 01. Linux
- ✓ 02. Windows

2. Networking

3. System Administration

- 01. Bash
- 02. PowerShell
- 03. Python

This concludes module one, Introduction to Operating Systems. You have just completed the portion on Windows. In the next module, we will discuss networking.