Introduction to Windows and CMD

Cybersecurity

Windows Administration and Hardening Day 1

Class Objectives

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

- Leverage the Windows Command Prompt (CMD) to navigate and manage directories and files.
- Use wmic and Task Manager to manage processes and retrieve system info.
- Create, manage, and display user information using the command-line tool net.
- Manage password policies using gpedit.
- Schedule tasks using Task Scheduler.

Welcome to Windows

Welcome to Windows

While many IT professionals prefer Mac OS and Linux, Windows is still the leader for desktop operating systems.

The popularity of Windows machines makes them the most common target for today's attackers.

Malware can specifically target vulnerabilities in unpatched and unsecure Windows machines and servers.

Windows in a Professional Context

Windows knowledge is essential for the following roles, among many others:

SOC Analyst

SOC analysts must monitor and detect suspicious activity on Windows machines.

System Administrator

The large majority of system administrators work with one or many Microsoft products and services: Windows PCs, Windows Servers, Office 365, and Exchange, etc.

Penetration Testing

Due to Windows' wide usage by businesses, penetration testers must exploit Windows and Microsoft-related platforms.

Endpoint Forensics

Being the most commonly supported endpoint device for businesses, forensics investigators must understand how Windows works.

Windows System Administrator

Today we will complete common system administration tasks using command-line and GUI tools to troubleshoot a problematic Windows PC.

01 Audit processes with Task Manager.

02 Use the command line to gather info and create files.

03 Enforce password policies.

04 Manage users.

05 Automate tasks.

Learning Windows

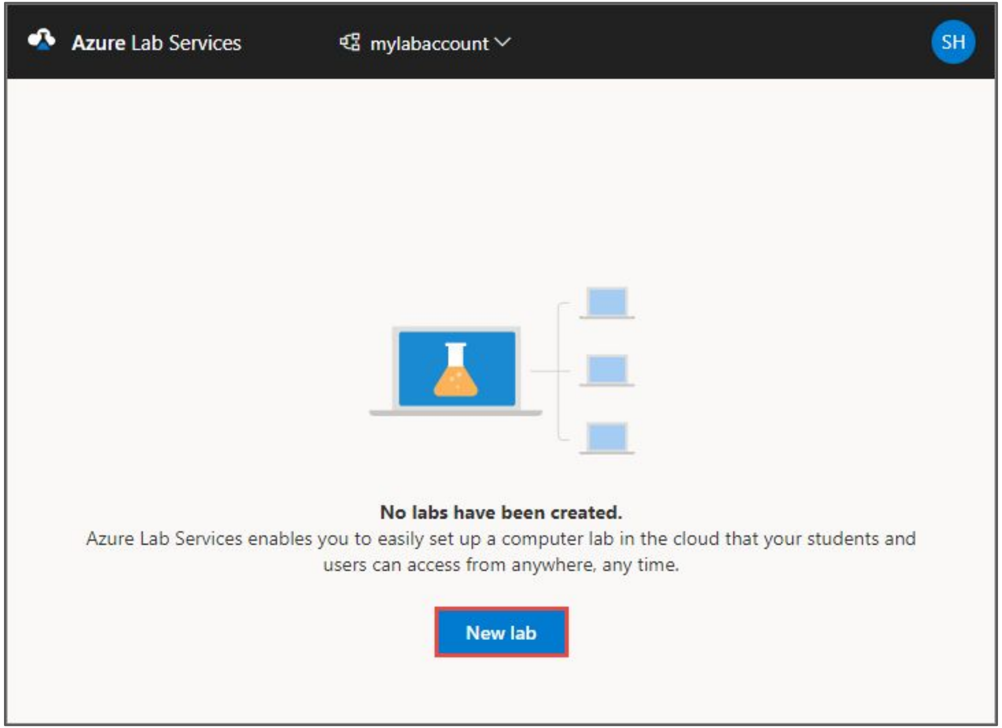
Today, we'll learn the "Windows way" of performing basic sysadmin tasks.

We've already learned how to do many of these tasks on Linux.

Since the topics covered today are similar to Linux, we will move quickly and emphasize the syntax and OS differences for completing tasks in Windows.

Launching Your Windows Lab

Before we get started, let's take a moment to log into and launch our **Azure Lab environment**.



No labs have been created.

Azure Lab Services enables you to easily set up a computer lab in the cloud that your students and users can access from anywhere, any time.

https://docs.microsoft.com/

Introduction to Task Manager

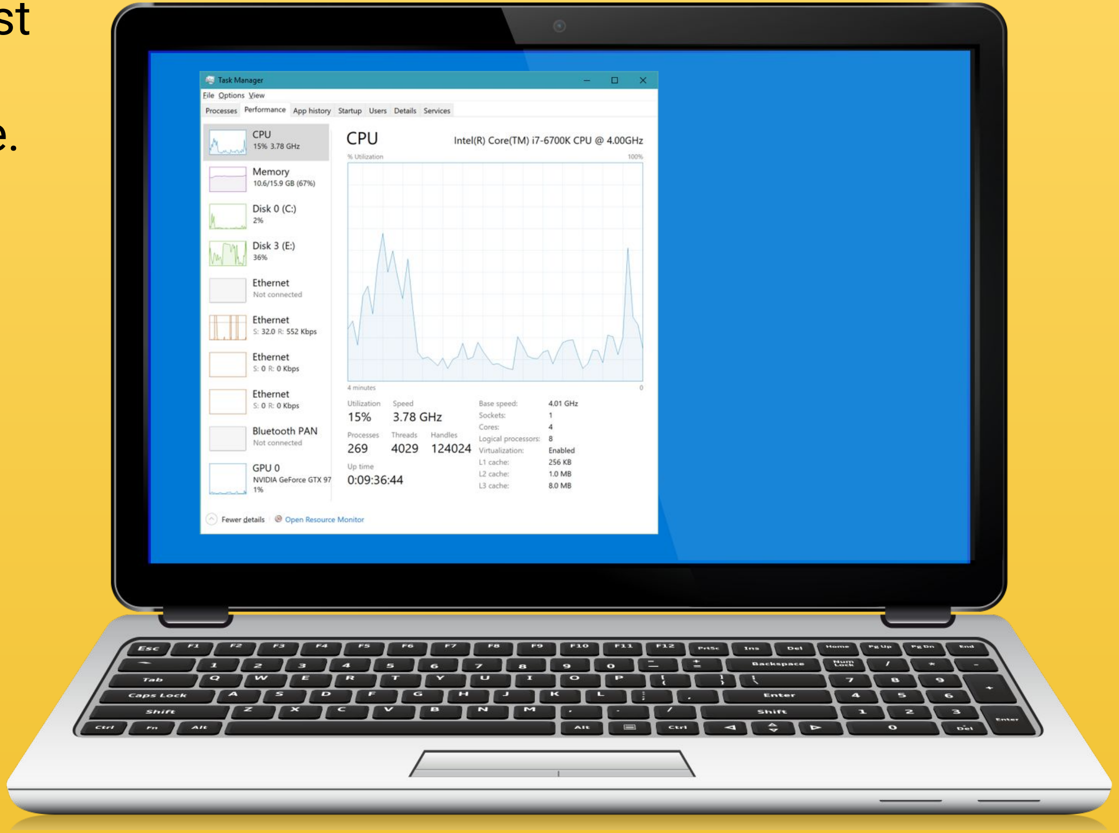
Did you notice the **excessive number of processes** that started up when you logged into the Windows 10 VM?

This is what a Windows workstation can be like if not maintained by an organization’s system administrator.

Task Manager

Task Manager is one of the most important Windows tools for troubleshooting resource usage.

- We'll audit and manage tasks and processes to identify errant or malicious actions taking place without users' or administrators' knowledge.
- Processes in Windows are much like the processes and PIDs you saw in the Linux units.



CPU
15% 3.78 GHz

Memory
10.6/15.9 GB (67%)

Disk 0 (C:)
2%

Disk 3 (E:)
36%

Ethernet
Not connected

Ethernet
S: 32.0 R: 552 Kbps

Ethernet
S: 0 R: 0 Kbps

Ethernet
S: 0 R: 0 Kbps

Bluetooth PAN
Not connected

GPU 0
NVIDIA GeForce GTX 97
1%

CPU Intel(R) Core(TM) i7-6700K CPU @ 4.00GHz

% Utilization 100%

4 minutes 0

Utilization 15%

Speed 3.78 GHz

Processes 269

Threads 4029

Handles 124024

Up time 0:09:36:44

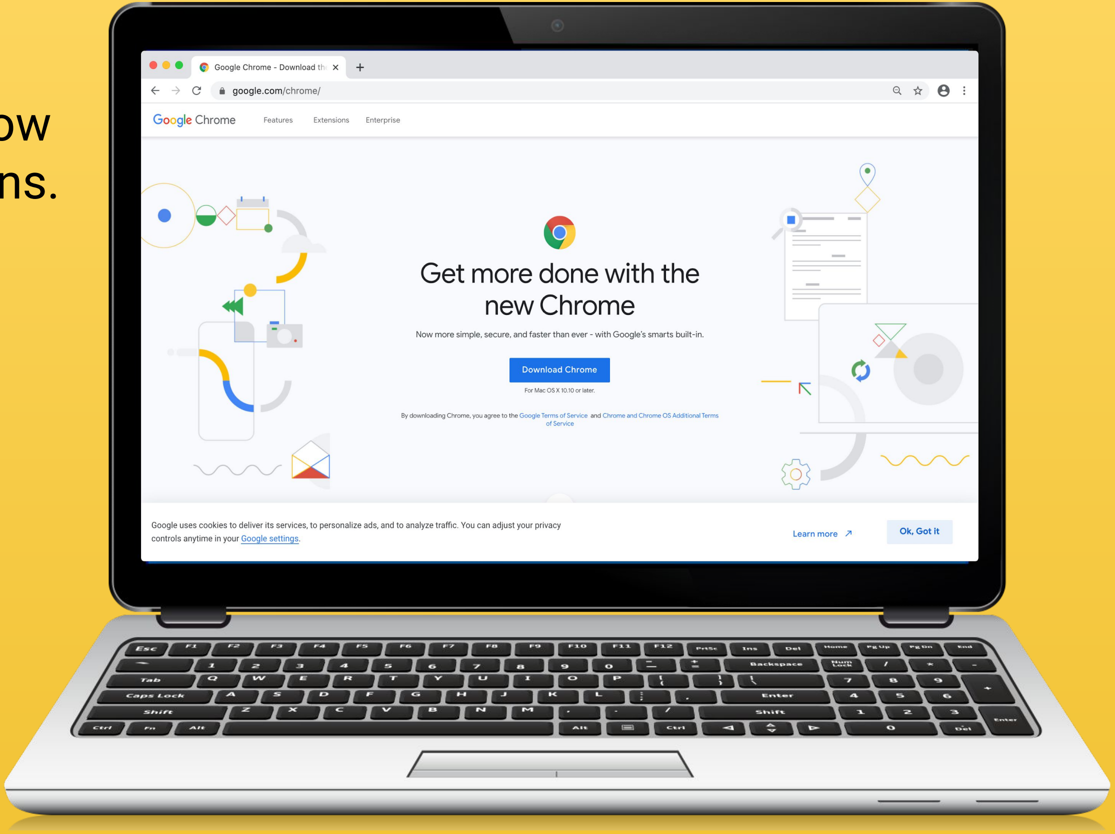
Base speed: 4.01 GHz
Sockets: 1
Cores: 4
Logical processors: 8
Virtualization: Enabled
L1 cache: 256 KB
L2 cache: 1.0 MB
L3 cache: 8.0 MB

Task Manager

Some programs, if left running while not in use, can take up excessive resources or even allow for unwanted remote connections.

Some examples are:

- Google Chrome, which is well-known for its high memory usage.
- Teamviewer, the remote desktop application, has had critical issues that have left systems extremely vulnerable and accessible from public connections.



Get more done with the new Chrome

Now more simple, secure, and faster than ever - with Google's smarts built-in.

For Mac OS X 10.10 or later.

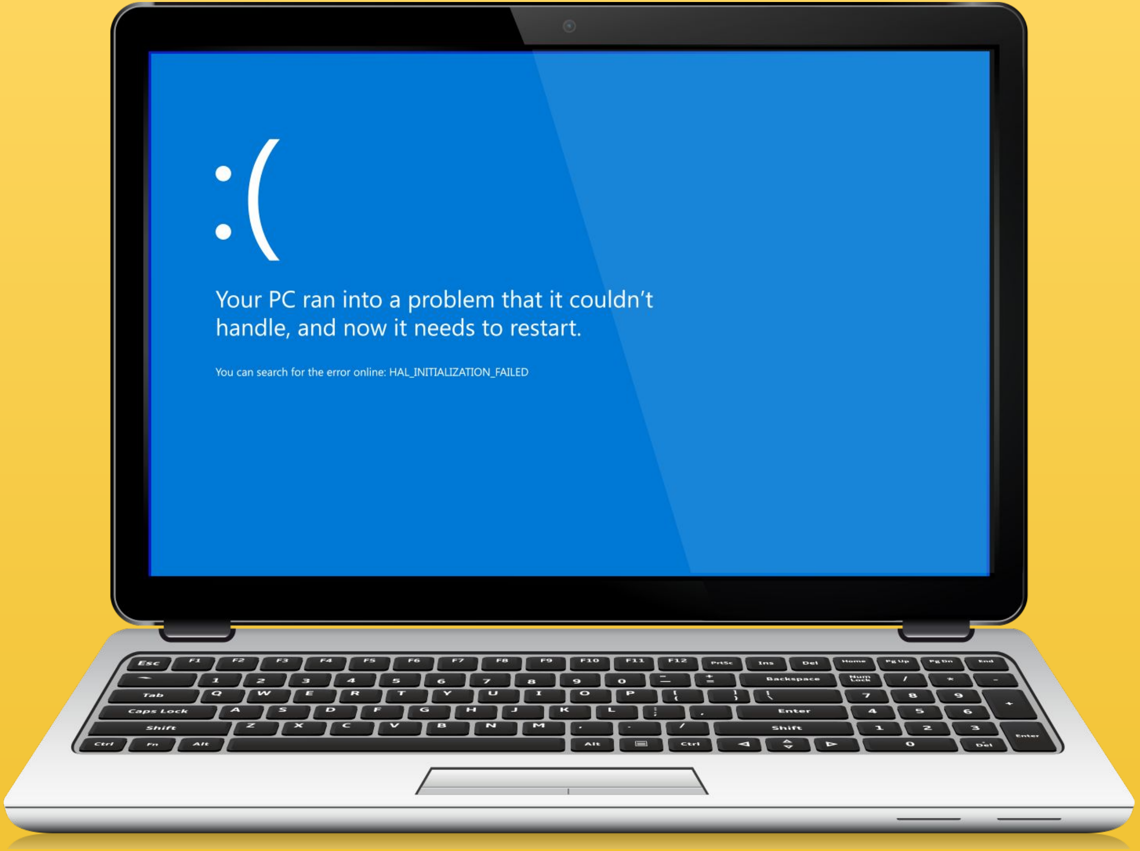
By downloading Chrome, you agree to the Google Terms of Service and Chrome and Chrome OS Additional Terms of Service

Google uses cookies to deliver its services, to personalize ads, and to analyze traffic. You can adjust your privacy controls anytime in your Google settings.

Task Manager

Some processes can even cause memory leaks that can result in system instability and abrupt system crashes.

When a Windows system crashes, you are often stuck with what what is known as the **blue screen of death**.

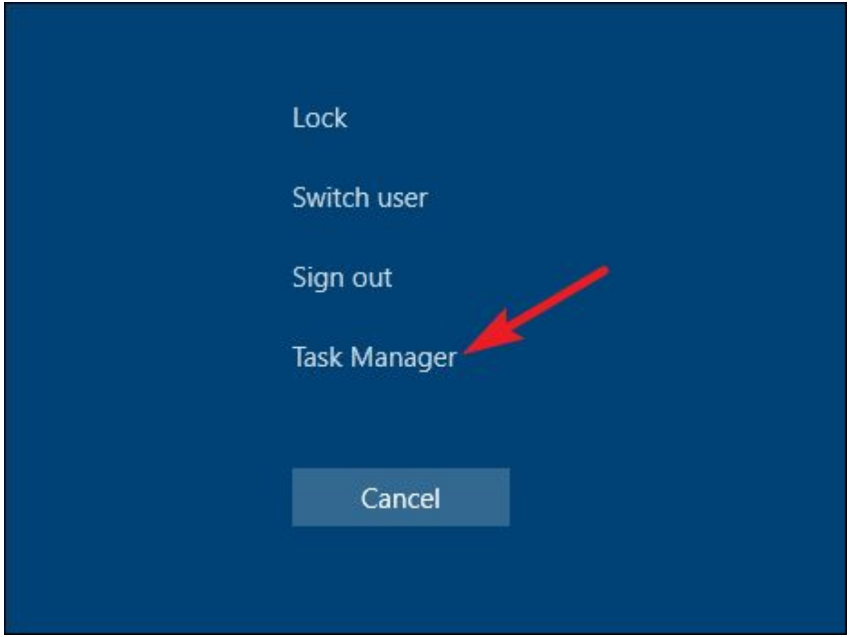


:(

Your PC ran into a problem that it couldn't handle, and now it needs to restart.

You can search for the error online: HAL_INITIALIZATION_FAILED

Let’s open up Task Manager, check out the processes, and **end a process**.



Instructor Demonstration

Task Manager and Ending Processes

Disabling Startup Applications (Task Manager)

Managing startup applications is important for system and security administration.

Startup applications can slow boot time due to their execution priority.

They may use excessive resources in the background, causing random system slowdowns.

They may use the network in the background. For example, they can initiate their own automatic updates, taking up network bandwidth but also become a security risk by opening ports to listen to.

They may require special permissions for their functionality. These can pose security risks if they are compromised through malware, which can then potentially run these rogue processes as administrators.

Introduction to Command Prompt (CMD)

Windows Directory and File Structure

The default Windows directory structure:

```
C:\
├── PerfLogs\
├── Program Files\
├── Program Files (x86)\
├── ProgramData\ [hidden folder]
├── Users\
│     └── [username]\
│           └── Desktop\
│           └── Documents\
└── Windows\
      └── System32\
            └── Config\
            └── Drivers\
                    └── etc\
                          └──hosts
                          └──networks
      └── Spool
    └── Temp
```

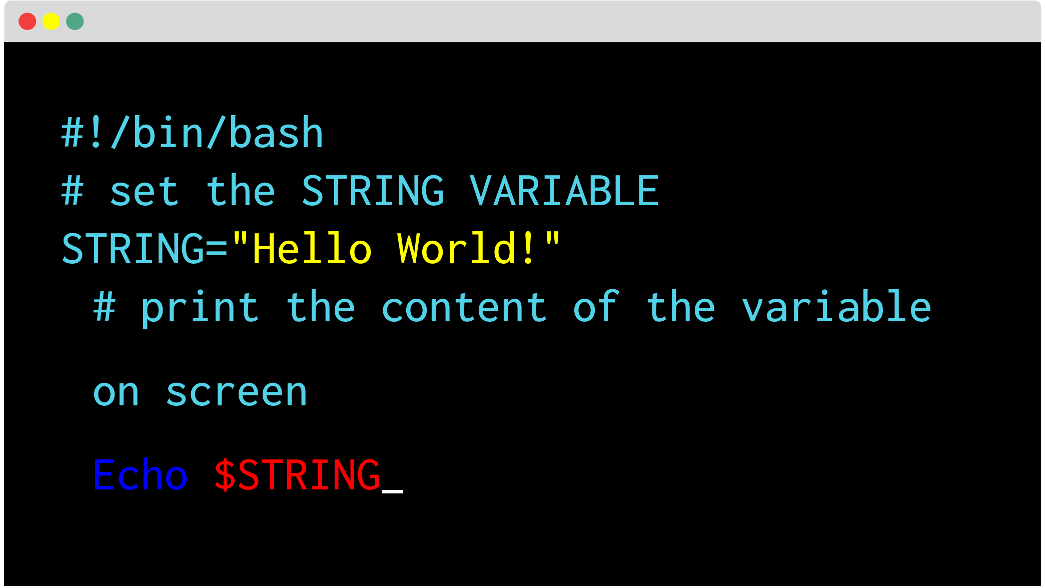
Windows Directory and File Structure

The default Windows directory structure:

```
C:\ (or whichever drive Windows is installed on) is the root drive.
 ├── PerfLogs\
 ├── Program Files\ Where 64-bit applications are installed.
 ├── Program Files (x86)\ Where 32-bit applications are installed.
 ├── ProgramData\ [hidden folder]  Where applications-specific settings reside.
 ├── Users\ Directory of all users (similar to the Linux /home directory).
 │     └── [username]\ Each specific user’s home folder.
 │           └── Desktop\ The desktop and document folders for the current user.
 │           └── Documents\
 └── Windows\ Holds Windows-specific programs and libraries.
       └── System32\
             └── Config\
             └── Drivers\
                    └── etc\
                          └──hosts
                          └──networks
       └── Spool
     └── Temp
```

Remember environment variables from the bash programming unit?

In Windows, they work the same way—they're preset by the system and usable in the command line and scripts.



```
#!/bin/bash
# set the STRING VARIABLE
STRING="Hello World!"
 # print the content of the variable

 on screen

 Echo $STRING_
```

Environment variables (envvars)
are special values that contain
information about the system,
such as the user’s home directory or
the system's program files directory.

Common ENV Variables

Envvars can be used for the following:

- Shortening long directory paths.
- Grabbing the current time.
- Finding the location of your system files.

Environment Variables	Default Value
%CD%	Current directory
%DATE%	Current date
%OS%	Windows
%ProgramFiles%	C:\Program Files
%ProgramFiles(x86)%	C:\Program Files (x86)
%TIME%	Current time
%USERPROFILE%	C:\Users\{username}
%SYSTEMDRIVE%	C:\
%SYSTEMROOT%	C:\Windows

Common ENV Variables

Linux variables are designated with a **$**, while Windows ENV variables are enclosed with % signs.

For example, to navigate to the 64-bit **Program Files** folder, we run:

```
cd %ProgramFiles%
```

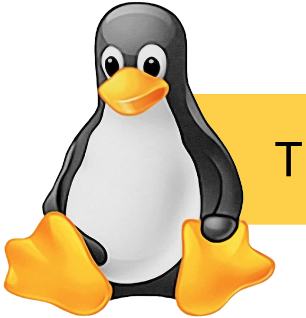
```
cd %USERPROFILE%\Desktop
```

Common ENV Variables

We can combine environment variables with regular directory names:

```
cd %USERPROFILE%\Desktop
```

`%USERPROFILE%` is a variable assigned to the value of the current user’s home directory.



This is the same as **`$HOME`** in Linux.

Command Prompt (CMD)

Windows Command Prompt (CMD or **`cmd.exe`**), is the command-line interface for Windows, comparable to a Unix shell, such as Bash for Linux.

`CMD` Command	Action	Linux Counterpart
`cd` or **`chdir`**	Change directory	**`cd`**
`dir`	List contents of directory	**`ls`**
`md` or **`mkdir`**	Create directory	
`copy`	Copy file	**`cp`**
`move`	Move (cut and paste) files	**`mv`**
`del` or **`erase`**	Delete files and directories	
`rd` or **`rmdir`**	Remove a directory if empty	
`find`	Search a file for specified string	
`exit`	Close CMD	
`type`	Show contents of specified file	**`cat`**

Command Prompt (CMD)

Windows Command Prompt (CMD or **`cmd.exe`**), is the command-line interface for Windows, comparable to a Unix shell, such as Bash for Linux.

CMD Command	Action	Linux Counterpart
`cd` or **`chdir`**	Change directory	**`cd`**
`dir`	List contents of directory	**`ls`**
`md` or **`mkdir`**	Create directory	
`copy`	Copy file	**`cp`**
`move`	Move (cut and paste) files	**`mv`**
`del` or **`erase`**	Delete files and directories	
`rd` or **`rmdir`**	Remove a directory if empty	
`find`	Search a file for specified string	
`exit`	Close CMD	
`type`	Show contents of specified file	**`cat`**

NOTE

Command prompts are not case sensitive with files and directories.

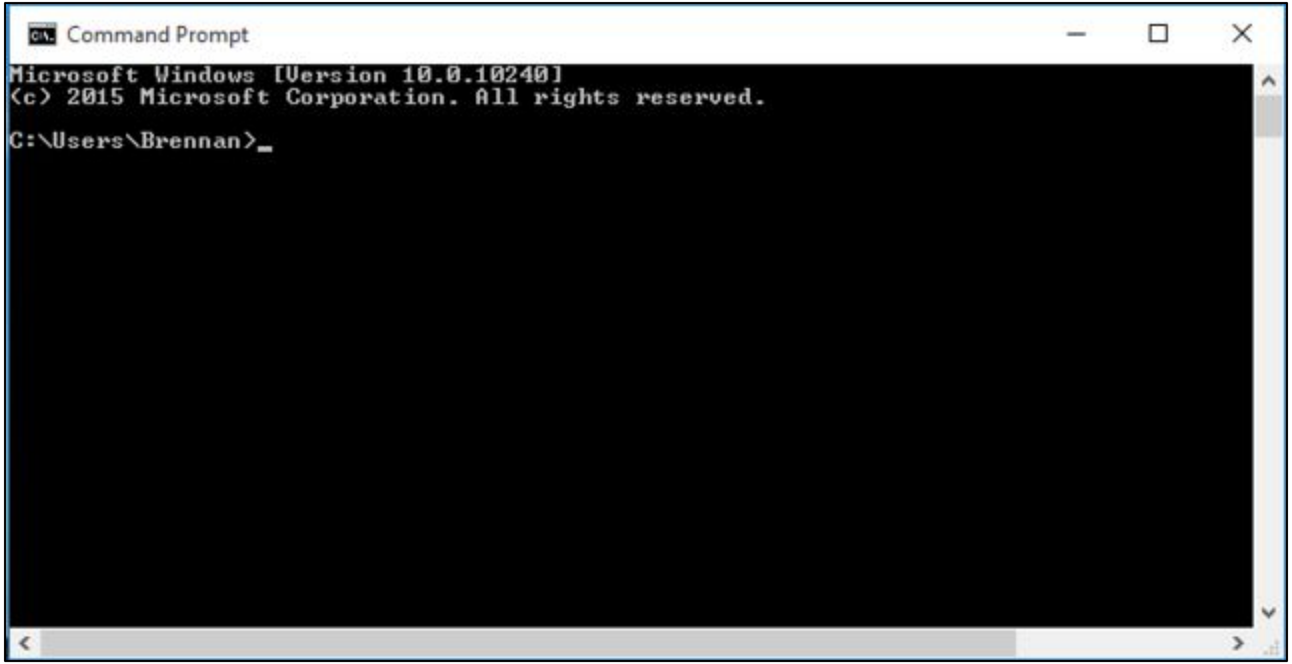
```
cd “Program files”
```

is the same as

```
cd “PROGRAM FILES”
```

Use quotes around the name of a file or directory that contains spaces.

In the next guided tour, we will create and manage files within **Windows CMD**.



```
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Brennan>_
```

https://en.wikipedia.org

Instructor Demonstration

CMD: Navigation and Output

Activity: Task Manager and CMD

In this activity, you will use CMD and Task Manager to output various details of a Windows workstation into a report file.

Suggested Time:

15 Minutes

Time’s Up! Let’s Review.

Questions?

Windows Management Instrumentation Command (`wmic`)

Windows Management Instrumentation Command (wmic) is a tool used to query system information and diagnostics, such as OS and hard disk info.

Sysadmins can also use `wmic` to launch, terminate, install, and uninstall processes.

`wmic` Structure and Conventions

```
wmic [global switches] [alias] [verbs] [properties]
```

```
[global switches]
```

Not to be confused with normal switches, these are `wmic`-specific global commands that alter behavior. They can do things like specify a file to append output to. Today, we will use the command `/APPEND`.

For example:

`wmic /APPEND:report.txt os get buildnumber`	Will append the Windows build number to the report.txt file.

`wmic` Structure and Conventions

```
wmic [global switches] [alias] [verbs] [properties]
```

`[alias]` is the Windows component that wmic queries.

Common aliases include:

`os (operating system)`	Contains properties specific to the operating system, such as the Windows edition name and build number.
`Logicaldisk`	Contains properties specific to the disk drives, such as drive name, file system, free space, size, and volume serial number.

wmic Structure and Conventions

```
wmic [global switches] [alias] [verbs] [properties]
```

`[verbs]` are actions we want to complete with the `wmic` command.

For example:

If we are using `wmic os` to find operating system information, we can then use the `get` verb, followed by the various `[properties]` we want to find.

`wmic` Structure and Conventions

```
wmic [global switches] [alias] [verbs] [properties]
```

```
[properties]
```

Common properties to retrieve using get:

`get caption`	Returns a one-line description of the given alias.
`get /value`	Gets all of the properties and values of an alias and lists each on a separate line.

Applying `wmic`

Let’s go through a few examples:

```
wmic os get /value
```

```
wmic os get caption, buildnumber
```

```
wmic /APPEND:report.txt os get caption
```

```
wmic logicaldisk get caption, filesystem, freespace, size, volumeserialnumber
```

```
wmic /APPEND:report.txt logicaldisk get caption, filesystem, freespace
```

`wmic` Demo

In the next demo, we will move through different programs, understand their importance in a sysadmin context, and get and append them to our report.

We'll retrieve the following properties:

Build Details

The Build Number and Caption

Logical Disk Info

The caption, filesystem, freespace, size, and serial number

Instructor Demonstration

`wmic` Demo

Activity: Creating a Report with `wmic` Output

In this activity, you will continue baselining the Windows system using `wmic` queries.

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Questions?

Countdown timer

15:00

(with alarm)

Users and Password Policies

Next, we’ll use the command-line tool net to manage user accounts, groups, and password policies.

Using `net`

We'll be using the following **`net`** utilities:

`net user`

For adding, removing and managing users.

`net localgroup`

For adding, removing, and managing local groups.

`net accounts`

For viewing password and logon requirements for users to enforce password security policies.

Using `net`

`net` lets us set the following password policies:

- Time before a password expires.
- Minimum number of characters required for a password.
- Minimum number of days before a password can be changed.
- Number of times a password must be unique before it can be reused again.

If using the password **`apples2apples`**, you'll have to change it to two new passwords before you can use **`apples2apples`** again.

net Demo Scenario

Your CIO is curious about the groups and password policies on the Windows workstation.

We need to retrieve more information from this workstation using the net command-line utility.

We'll use the **`net`** tool to do the following:

- Enumerate users to see **`net`** output.
- Enumerate azadmin's groups and password policies.
- Enumerate local groups with **`net localgroup`**.
- Enumerate the Windows workstation's current password policies with **`net accounts`**.

Instructor Demonstration

`net`

Activity: Users, Groups and Password Policies

In this activity, you will use the `net` utility to retrieve more information about the Windows workstation.

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Questions?

Creating Users and Setting Password Policy

Password Policies

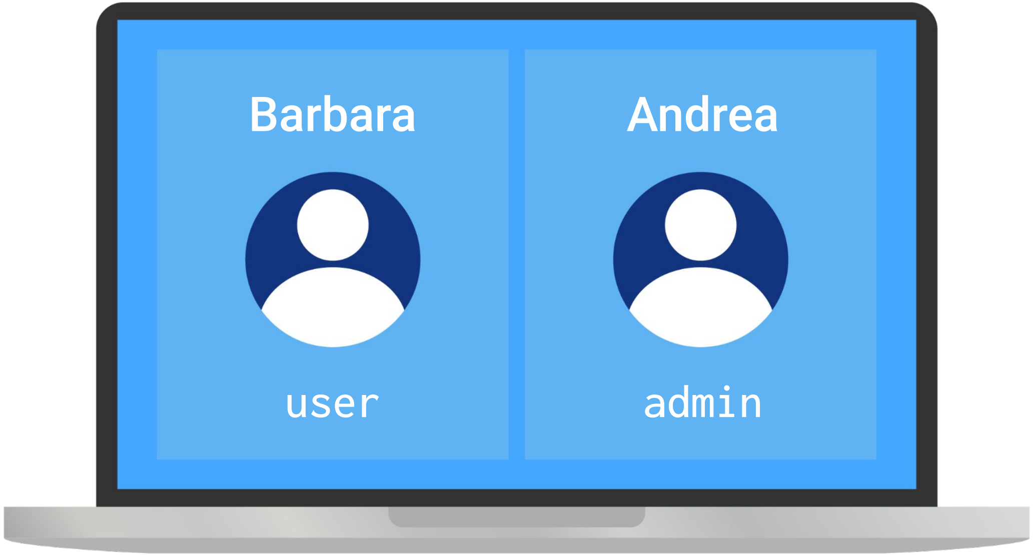
We’ve discussed the importance of password policies in earlier Linux units.

Now we’ll establish password policies for new users in Windows.

Password Policies

In the next demonstration, we'll use the following scenario:

A new regular user (Barbara) and new administrator (Andrea) need to be added to the workstation.



- We'll use **`net user`** to create user accounts for Andrea, the new senior developer, and Barbara, the new sales representative.
- We will create these users and set their password policies to make sure they follow company-wide policies.

Instructor Demonstration

Adding Users and Setting Password Policies

Activity: Create Users and Set Passwords

In this activity, you will create users and set password policies for two new users.

Suggested Time:

10 Minutes

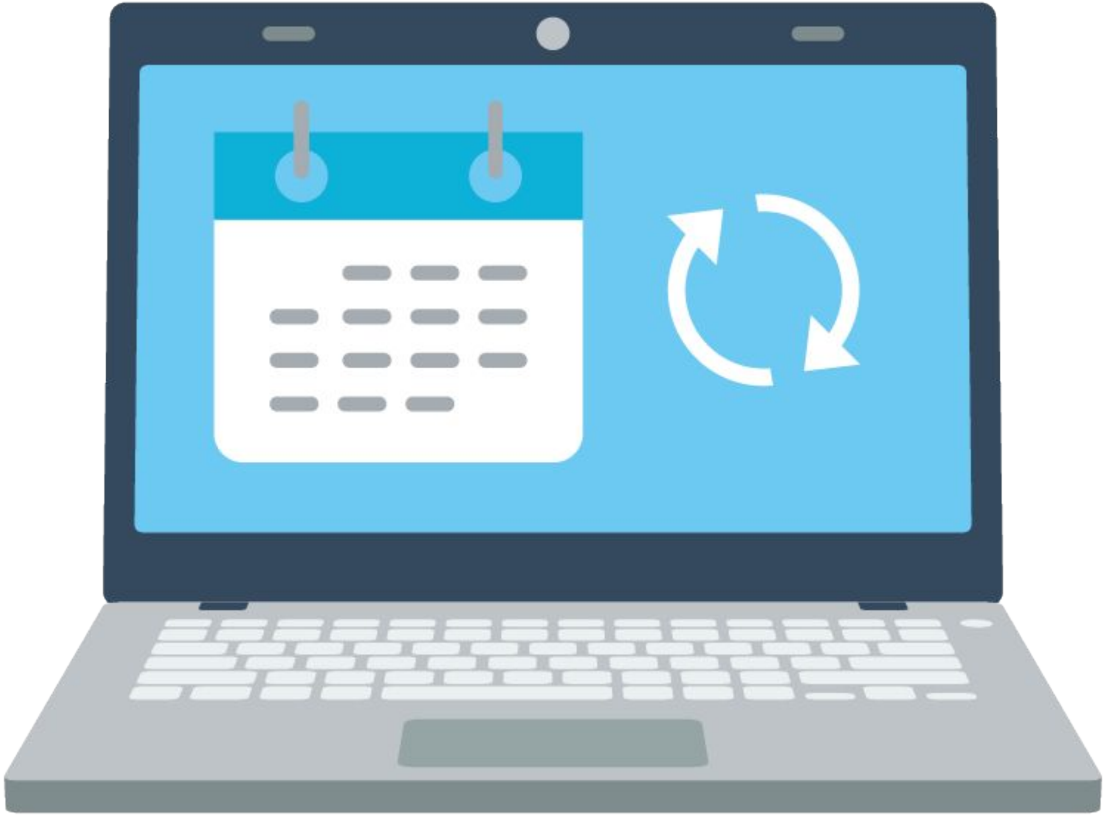
Time’s Up! Let’s Review.

Questions?

Task Scheduling
(Optional / Time Permitting)

Task Scheduler is a GUI tool that allows system administrators to automate the execution of scripts and applications on a Windows system.

Similar to cron jobs, tasks can be set to execute at specific times, or a certain amount of time after a user logs in.



Task Scheduling

Properly managing systems with scheduled tasks allows us to automate security and system administration actions such as:

Checking for updates for endpoint security software

Sending important logs to systems such as SIEM

Scheduling system maintenance scripts

Task Scheduling Demo Setup

In this demo, we will use the administrative user, Andrew, to create scheduled tasks that will automate the reports we’ve been working on.

- The CIO wants us to schedule reports to be created on a daily basis.
- We will use **Task Scheduler** to create a task that runs each day.

Activity: Task Scheduling

In this activity, we will use Task Scheduler to schedule reports to be created every day.

Suggested Time:

10 Minutes

Time’s Up! Let’s Review.

Questions?

⚠️ Important ⚠️

Make sure to shut down your Windows RDP Host Machine.

You are provided **30 hours** of Azure lab access.

- If you exceed that quota, you will be provided an additional **10 hours**.
- If you exceed those additional hours, you will be provided an additional **5 hours**.

Once you exceed that final quota, you will not be provided any additional hours.

It is extremely important that you preserve your allotted hours by **shutting off your machines** at the end of each class.

Questions?

The End