Windows Fundamentals1

Introduction to Windows

As a penetration tester, it is important to have knowledge of a wide variety of technologies. A thorough understanding of Windows and Linux operating systems is beneficial in a wide range of assessment types. The majority of systems that we encounter during assessments, whether on-premise or in the cloud, will be based on these two operating systems. It is important to understand how to attack and defend these operating systems and how they can each be used as a platform to perform further penetration testing activities.

The Windows Operating System

Microsoft first introduced the Windows operating system on November 20, 1985. The first version of Windows was a graphical operating system shell for MS-DOS. Later versions of Windows Desktop introduced the Windows File Manager, Program Manager, and Print Manager programs.

Windows 95 was the first full integration of Windows and DOS and offered built-in Internet support for the first time. This version also debuted the Internet Explorer web browser. Since the initial version, there have been over a dozen versions of Windows released, such as Windows XP, Vista, and 8, up to the current version: Windows 10. Over time, Microsoft has offered various editions of each Windows Desktop release catering to everyone from casual consumers to enterprise customers.

Windows Server was first released in 1993 with the release of Windows NT 3.1 Advanced Server. Windows NT saw several updates over the years, adding in technologies such as Internet Information Services (IIS), various networking protocols, Administrative Wizards to facilitate admin tasks, and more. With the release of Windows 2000, Microsoft debuted Active Directory, originally intended to help sysadmins set up file sharing, data encryption, VPNs, etc. Windows Server 2000 also included the Microsoft Management Console (MMC) and supported dynamic disk volumes.

Windows Server 2003 came next with server roles, a built-in firewall, the Volume Shadow Copy Service, and more. Windows Server 2008 included failover clustering, Hyper-V virtualization software, Server Core, Event Viewer, and major enhancements to Active Directory. Over the years, Microsoft released further Server versions, including Server 2012, Server 2016, and most recently, Server 2019. This latest version added support for Kubernetes, Linux containers, and more advanced security features.

As new versions of Windows are introduced, older versions are deprecated and no longer receive Microsoft updates (unless a long-term support contract is purchased in some cases). Windows Server 2008 and 2012 reached end of life for security updates on January 14, 2020. Currently, only Server 2012 R2 and later are in support. However, Microsoft has released out-of-band patches for earlier versions of Windows in the past few years due to the discovery of the critical SMBv1 vulnerability (EternalBlue).

Many versions of Windows are now deemed "legacy" and are no longer supported. Organizations often find themselves running various older operating systems to support critical applications or due to operational or budgetary concerns. An assessor needs to understand the differences between versions and the various misconfigurations and vulnerabilities inherent to each.

Windows Versions

The following is a list of the major Windows operating systems and associated version numbers:

Operating System Names	Version Number
Windows NT 4	4.0
Windows 2000	5.0
Windows XP	5.1
Windows Server 2003, 2003 R2	5.2
Windows Vista, Server 2008	6.0
Windows 7, Server 2008 R2	6.1
Windows 8, Server 2012	6.2
Windows 8.1, Server 2012 R2	6.3
Windows 10, Server 2016, Server 2019	10.0

We can use the Get-WmiObject cmdlet to find information about the operating system. This cmdlet can be used to get instances of WMI classes or information about available WMI classes. There are a variety of ways to find the version and build number of our system. We can easily obtain this information using the `win32_OperatingSystem` class, which shows that we are on a Windows 10 host, build number 19041.

```
PS C:\htb> Get-WmiObject -Class win32_OperatingSystem | select Version,BuildNumber

Version    BuildNumber
-------    -----------
10.0.19041 19041
```

Some other useful classes that can be used with `Get-WmiObject` are `Win32_Process` to get a process listing, `Win32_Service` to get a listing of services, and `Win32_Bios` to get Basic Input/Output System (`BIOS`) information. The BIOS is firmware installed on a computer's motherboard that controls the computer's essential functions, such as power management, input/output interfaces, and system configuration. We can use the `ComputerName` parameter to get information about remote computers. `Get-WmiObject` can be used to start and stop services on local and remote computers, and more. Further information about the cmdlet can be found here and here.

Accessing Windows

Local Access Concepts

If you are reading these words right now, you have local access to a computer of some kind. Be it a smartphone, tablet, laptop, Raspberry Pi, or Desktop. Local access is the most common way to access any computer, including computers running Windows. `Input` is likely happening through a keyboard, trackpad &/or mouse. `Output` is coming from the display screen(s). Organizations with office space where employees work on a day-to-day basis build security policies and security controls around the idea that their employees are working in dedicated workspaces on computers owned by the organization. It is becoming more common to see organizations increasing their support for remote work with their non-technical and technical workforces. This is not a new reality for technical professionals working in IT, Software Development & Infosec. On any given day, a technical professional could be accessing multiple machines locally and remotely. With that, let's discuss the concept of remote access.

Remote Access Concepts

Remote Access is accessing a computer over a network. Local access to a computer is needed before one can access another computer remotely. There are countless methods for remote access. In this module, we will mainly use remote access methods to connect to and interact with Windows operating systems. Advances in Networking & Internet technologies have given birth to entire industries that rely completely on remote access & remote administration of computer systems.

Consider MSPs & MSSPs, both industries are primarily dependent on managing their client's computer systems remotely. This functionality allows them to centralize management, standardize what technologies are used, automate numerous tasks, enable remote work arrangements and allow for quick response time when issues surface, or potential security threats emerge. Remote access is not just limited to MSPs & MSSPs. Organizations with IT, Software Development &/or Security teams use remote access methods daily to build applications, manage servers and administer employee workstations. Some of the most common remote access technologies include but aren't limited to:

- Virtual Private Networks (VPN)
- Secure Shell (SSH)
- File Transfer Protocol (FTP)
- Virtual Network Computing (VNC)
- Windows Remote Management (or PowerShell Remoting) (WinRM)
- Remote Desktop Protocol (RDP)

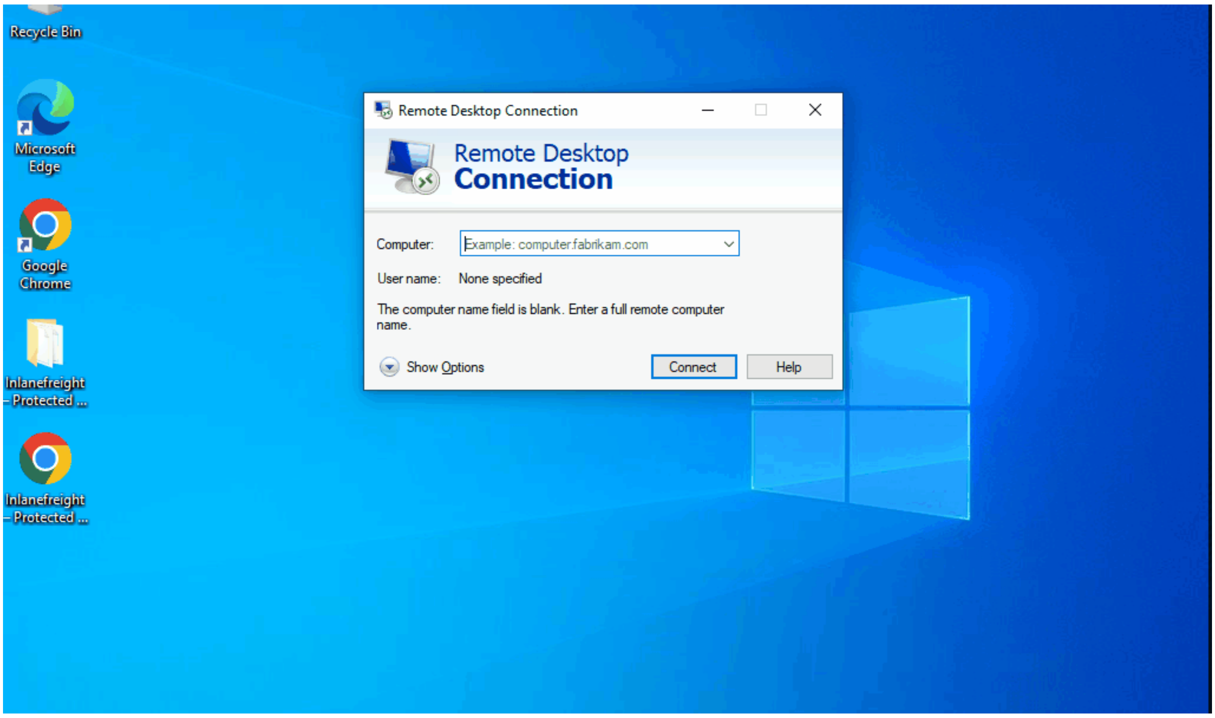
We will focus primarily on using RDP in this module.

Remote Desktop Protocol (RDP)

RDP uses a client/server architecture where a client-side application is used to specify a computer's target IP address or hostname over a network where RDP access is enabled. The target computer where RDP remote access is enabled is considered the server. It is important to note that RDP listens by default on logical port `3389`. Keep in mind that an IP address is used as a logical identifier for a computer on a network, and a logical port is an identifier assigned to an application. In simpler terms, we could consider a network subnet a street in a town (the corporate network), an IP address in that subnet assigned to a host as a house on that street, and logical ports as windows/doors that can be used to access the house.

Once a request (encapsulated inside a packet) has reached a destination computer via its IP address, the request will be directed to an application hosted on the computer based on the port specified in that request (included as a header inside a packet). IP addressing and protocol encapsulation are covered in greater detail in the module Introduction to Networking. From a networking perspective, in this module, we only need to understand that every computer has an IP address assigned to communicate over a network, and applications hosted on target computers listen on specific logical ports.

We can use RDP to connect to a Windows target from an attack host running Linux or Windows. If we are connecting to a Windows target from a Windows host, we can use the built-in RDP client application called `Remote Desktop Connection` (mstsc.exe). Check out the clip below to see basic usage:


Recycle Bin
Microsoft Edge
Google Chrome
Inlanefreight - Protected ...
Inlanefreight - Protected ...
Remote Desktop Connection
Show Options
Connect
Help

Remote Desktop Connection

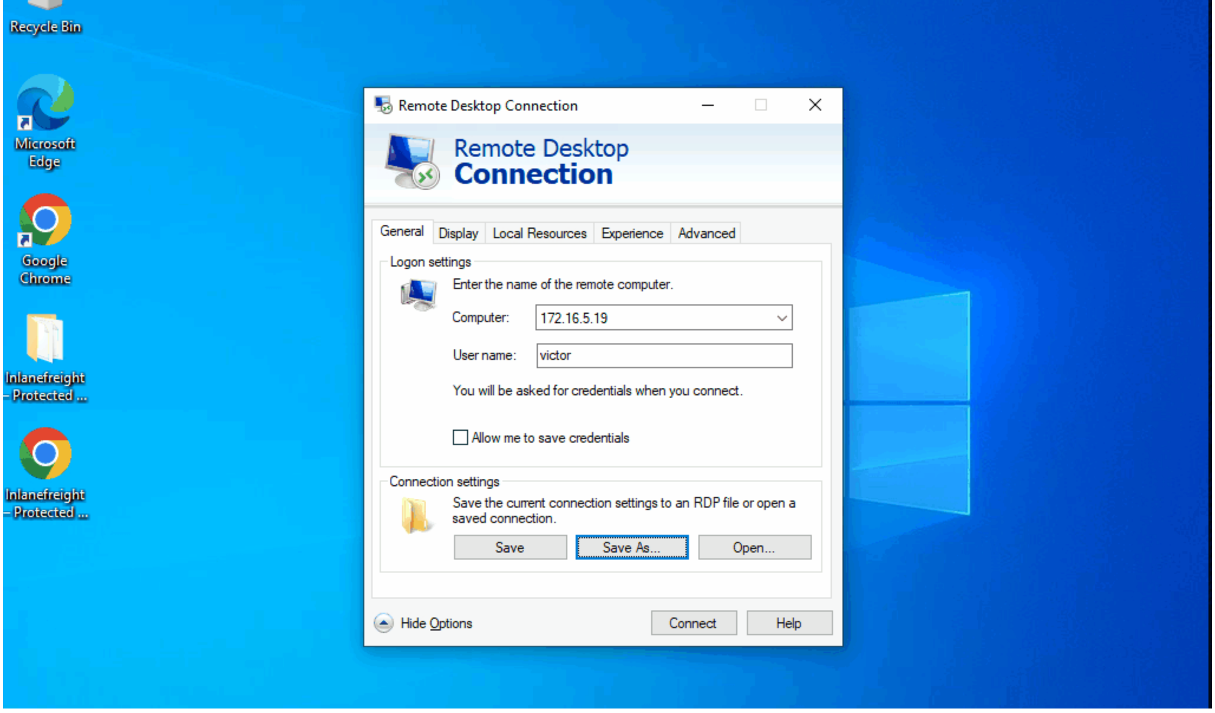
Computer: Example: computer.fabrikam.com

User name: None specified

The computer name field is blank. Enter a full remote computer name.

For this to work, remote access must already be allowed on the target Windows system. By default, remote access is not allowed on Windows operating systems. The HTB Academy team has configured many of our Windows targets to permit RDP access once connected to the Academy labs via VPN.

Remote Desktop Connection also allows us to save connection profiles. This is a common habit among IT admins because it makes connecting to remote systems more convenient.



Remote Desktop Connection

Logon settings

Enter the name of the remote computer.

Computer: 172.16.5.19

User name: victor

You will be asked for credentials when you connect.

□ Allow me to save credentials

Connection settings

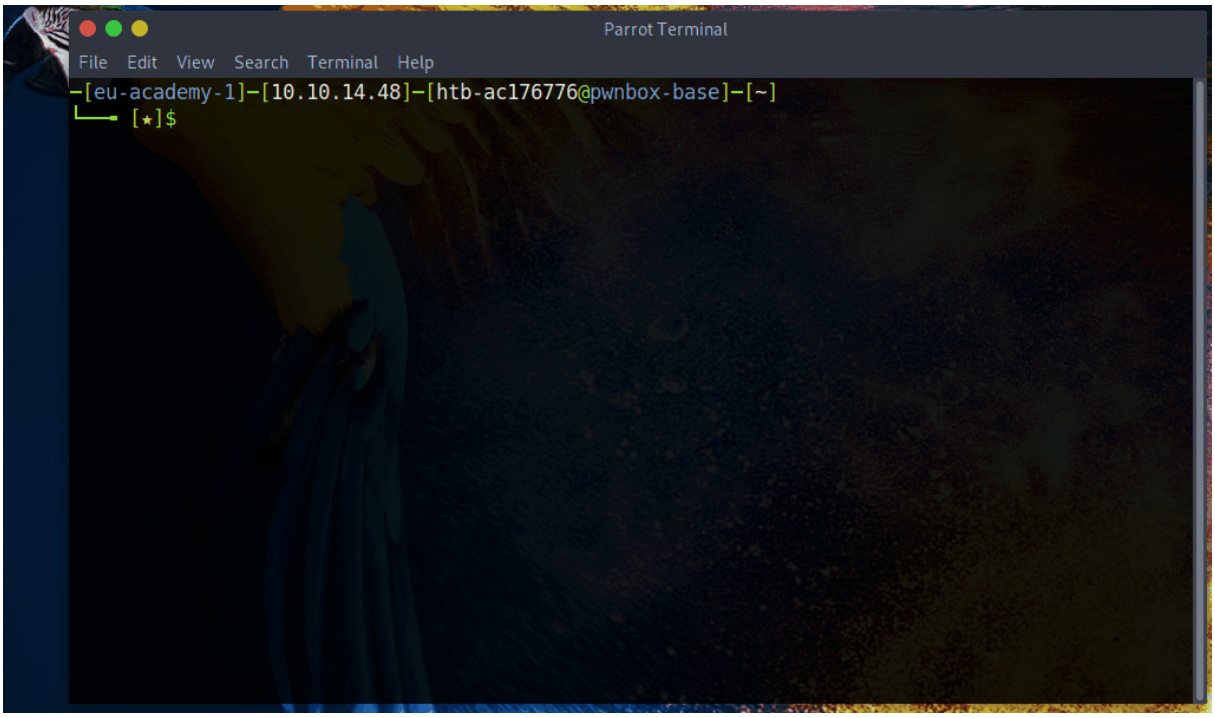
Save the current connection settings to an RDP file or open a saved connection.

As pentesters, we can benefit from looking for these saved Remote Desktop Files (`.rdp`) while on an engagement.

Many other Remote Desktop client applications exist, some of which are listed in this Microsoft article called Remote Desktop clients. We will not cover every Remote Desktop client application in this module.

Using xfreerdp

From a Linux-based attack host we can use a tool called xfreerdp to remotely access Windows targets. You will notice that we use xfreerdp across multiple modules because of its ease of use, feature set, command line utility, and efficiency. Check out the clip below to see basic usage from Pwnbox:



```
─[eu-academy-1]─[10.10.14.48]─[htb-ac176776@pwnbox-base]─[~]
└──╼ [★]$
```

Remember that we can also copy and paste in xfreerdp commands in the command line, so we do not need to enter options manually. There are several options available to us with xfreerdp, such as drive redirection to be able to tranfer files to/from the target host, which are worth practicing and we will cover in other modules within HTB Academy.

Other RDP clients exist, such as Remmina and rdesktop, and we encourage you to experiment with others and see what works best for you. Now that we have covered these concepts let's apply them by spawning the target below and connecting to it using RDP with the credentials provided.

Connecting to the Windows Target

Connect via Remote Desktop (RDP) using the following command:

```
xfreerdp /v:<targetIp> /u:htb-student /p:Password
```

Note: It may take 1-2 minutes for your target instance to spawn.

Operating System Structure

In Windows operating systems, the root directory is <drive_letter>:\ (commonly C drive). The root directory (also known as the boot partition) is where the operating system is installed. Other physical and virtual drives are assigned other letters, for example, Data (E:). The directory structure of the boot partition is as follows:

Directory	Function
Perflogs	Can hold Windows performance logs but is empty by default.
Program Files	On 32-bit systems, all 16-bit and 32-bit programs are installed here. On 64-bit systems, only 64-bit programs are installed here.
Program Files (x86)	32-bit and 16-bit programs are installed here on 64-bit editions of Windows.
ProgramData	This is a hidden folder that contains data that is essential for certain installed programs to run. This data is accessible by the program no matter what user is running it.
Users	This folder contains user profiles for each user that logs onto the system and contains the two folders Public and Default.
Default	This is the default user profile template for all created users. Whenever a new user is added to the system, their profile is based on the Default profile.
Public	This folder is intended for computer users to share files and is accessible to all users by default. This folder is shared over the network by default but requires a valid network account to access.
AppData	Per user application data and settings are stored in a hidden user subfolder (i.e., cliff.moore\AppData). Each of these folders contains three subfolders. The Roaming folder contains machine-independent data that should follow the user's profile, such as custom dictionaries. The Local folder is specific to the computer itself and is never synchronized across the network. LocalLow is similar to the Local folder, but it has a lower data integrity level. Therefore it can be used, for example, by a web browser set to protected or safe mode.
Windows	The majority of the files required for the Windows operating system are contained here.
System, System32, SysWOW64	Contains all DLLs required for the core features of Windows and the Windows API. The operating system searches these folders any time a program asks to load a DLL without specifying an absolute path.
WinSxS	The Windows Component Store contains a copy of all Windows components, updates, and service packs.

Exploring Directories Using Command Line

We can explore the file system using the dir command.

```
C:\htb> dir c:\ /a
 Volume in drive C has no label.
 Volume Serial Number is F416-77BE

 Directory of c:\

08/16/2020  10:33 AM    <DIR>          $Recycle.Bin
06/25/2020  06:25 PM    <DIR>          $WinREAgent
07/02/2020  12:55 PM             1,024 AMTAG.BIN
06/25/2020  03:38 PM    <JUNCTION>     Documents and Settings [C:\Users]
08/13/2020  06:03 PM             8,192 DumpStack.log
08/17/2020  12:11 PM             8,192 DumpStack.log.tmp
08/27/2020  10:42 AM    37,752,373,248 hiberfil.sys
08/17/2020  12:11 PM    13,421,772,800 pagefile.sys
12/07/2019  05:14 AM    <DIR>          PerfLogs
08/24/2020  10:38 AM    <DIR>          Program Files
07/09/2020  06:08 PM    <DIR>          Program Files (x86)
08/24/2020  10:41 AM    <DIR>          ProgramData
06/25/2020  03:38 PM    <DIR>          Recovery
06/25/2020  03:57 PM             2,918 RHDSetup.log
08/17/2020  12:11 PM        16,777,216 swapfile.sys
08/26/2020  02:51 PM    <DIR>          System Volume Information
08/16/2020  10:33 AM    <DIR>          Users
08/17/2020  11:38 PM    <DIR>          Windows
               7 File(s) 51,190,943,590 bytes
              13 Dir(s)  261,310,697,472 bytes free
```

The tree utility is useful for graphically displaying the directory structure of a path or disk.

```
C:\htb> tree "c:\Program Files (x86)\VMware"
Folder PATH listing
Volume serial number is F416-77BE
C:\PROGRAM FILES (X86)\VMWARE
├───VMware VIX
│   ├───doc
│   │   ├───errors
│   │   ├───features
│   │   ├───lang
│   │   │   └───c
│   │   │       └───functions
│   │   └───types
│   ├───samples
│   └───Workstation-15.0.0
│       ├───32bit
│       └───64bit
└───VMware Workstation
    ├───env
    ├───hostd
    │   ├───coreLocale
    │   │   └───en
    │   ├───docroot
    │   │   ├───client
    │   │   └───sdk
    │   ├───extensions
    │   │   └───hostdiag
    │   │       └───locale
    │   │           └───en
    │   └───vimLocale
    │
    │       └───en
    ├───ico
    ├───messages
    │   ├───ja
    │   └───zh_CN
    ├───OVFTool
    │   ├───env
    │   │   └───en
    │   └───schemas
    │       ├───DMTF
    │       └───vmware
    ├───Resources
    ├───tools-upgraders
    └───x64
```

The `tree` command can provide us with a large amount of information. The following command can be used to walk through all the files in the C drive, one screen at a time. This command can be modified to be run against any directory.

```
tree c:\ /f | more
```

File System

There are 5 types of Windows file systems: FAT12, FAT16, FAT32, NTFS, and exFAT. FAT12 and FAT16 are no longer used on modern Windows operating systems. We will touch upon the FAT32 and exFAT file systems for this training, but our main focus will be the NTFS file system.

FAT32 (File Allocation Table) is widely used across many types of storage devices such as USB memory sticks and SD cards but can also be used to format hard drives. The "32" in the name refers to the fact that FAT32 uses 32 bits of data for identifying data clusters on a storage device.

`Pros of FAT32:`

- Device compatibility - it can be used on computers, digital cameras, gaming consoles, smartphones, tablets, and more.
- Operating system cross-compatibility - It works on all Windows operating systems starting from Windows 95 and is also supported by MacOS and Linux.

`Cons of FAT32:`

- Can only be used with files that are less than 4GB.
- No built-in data protection or file compression features.
- Must use third-party tools for file encryption.

NTFS (New Technology File System) is the default Windows file system since Windows NT 3.1. In addition to making up for the shortcomings of FAT32, NTFS also has better support for metadata and better performance due to improved data structuring.

`Pros of NTFS:`

- NTFS is reliable and can restore the consistency of the file system in the event of a system failure or power loss.
- Provides security by allowing us to set granular permissions on both files and folders.
- Supports very large-sized partitions.
- Has journaling built-in, meaning that file modifications (addition, modification, deletion) are logged.

`Cons of NTFS:`

- Most mobile devices do not support NTFS natively.
- Older media devices such as TVs and digital cameras do not offer support for NTFS storage devices.

Permissions

The NTFS file system has many basic and advanced permissions. Some of the key permission types are:

Permission Type	Description
Full Control	Allows reading, writing, changing, deleting of files/folders.
Modify	Allows reading, writing, and deleting of files/folders.
List Folder Contents	Allows for viewing and listing folders and subfolders as well as executing files. Folders only inherit this permission.
Read and Execute	Allows for viewing and listing files and subfolders as well as executing files. Files and folders inherit this permission.
Write	Allows for adding files to folders and subfolders and writing to a file.
Read	Allows for viewing and listing of folders and subfolders and viewing a file's contents.
Traverse Folder	This allows or denies the ability to move through folders to reach other files or folders. For example, a user may not have permission to list the directory contents or view files in the documents or web apps directory in this example c:\users\bsmith\documents\webapps\backups\backup_02042020.zip but with Traverse Folder permissions applied, they can access the backup archive.

Files and folders inherit the NTFS permissions of their parent folder for ease of administration, so administrators do not need to explicitly set permissions for each file and folder, as this would be extremely time-consuming. If permissions do need to be set explicitly, an administrator can disable permissions inheritance for the necessary files and folders and then set the permissions directly on each.

Integrity Control Access Control List (icacls)

NTFS permissions on files and folders in Windows can be managed using the File Explorer GUI under the security tab. Apart from the GUI, we can also achieve a fine level of granularity over NTFS file permissions in Windows from the command line using the icacls utility.

We can list out the NTFS permissions on a specific directory by running either `icacls` from within the working directory or `icacls C:\Windows` against a directory not currently in.

```
C:\htb> icacls c:\windows
c:\windows NT SERVICE\TrustedInstaller:(F)
           NT SERVICE\TrustedInstaller:(CI)(IO)(F)
           NT AUTHORITY\SYSTEM:(M)
           NT AUTHORITY\SYSTEM:(OI)(CI)(IO)(F)
           BUILTIN\Administrators:(M)
           BUILTIN\Administrators:(OI)(CI)(IO)(F)
           BUILTIN\Users:(RX)
           BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
           CREATOR OWNER:(OI)(CI)(IO)(F)
           APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(RX)
           APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)
           APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(RX)
           APPLICATION PACKAGE AUTHORITY\ALL RESTRICTED APPLICATION PACKAGES:(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files
```

The resource access level is listed after each user in the output. The possible inheritance settings are:

- `(CI)`: container inherit
- `(OI)`: object inherit
- `(IO)`: inherit only
- `(NP)`: do not propagate inherit
- `(I)`: permission inherited from parent container

In the above example, the `NT AUTHORITY\SYSTEM` account has object inherit, container inherit, inherit only, and full access permissions. This means that this account has full control over all file system objects in this directory and subdirectories.

Basic access permissions are as follows:

- `F` : full access
- `D` : delete access
- `N` : no access
- `M` : modify access
- `RX` : read and execute access
- `R` : read-only access
- `W` : write-only access

We can add and remove permissions via the command line using `icacls`. Here we are executing `icacls` in the context of a local administrator account showing the `C:\users` directory where the `joe` user does not have any write permissions.

```
C:\htb> icacls c:\Users
c:\Users NT AUTHORITY\SYSTEM:(OI)(CI)(F)
         BUILTIN\Administrators:(OI)(CI)(F)
         BUILTIN\Users:(RX)
         BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
         Everyone:(RX)
         Everyone:(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files
```

Using the command `icacls c:\users /grant joe:f` we can grant the joe user full control over the directory, but given that `(oi)` and `(ci)` were not included in the command, the joe user will only have rights over the `c:\users` folder but not over the user subdirectories and files contained within them.

```
C:\htb> icacls c:\users /grant joe:f
processed file: c:\users
Successfully processed 1 files; Failed processing 0 files
```

```
C:\htb> >icacls c:\users
c:\users WS01\joe:(F)
         NT AUTHORITY\SYSTEM:(OI)(CI)(F)
         BUILTIN\Administrators:(OI)(CI)(F)
         BUILTIN\Users:(RX)
         BUILTIN\Users:(OI)(CI)(IO)(GR,GE)
         Everyone:(RX)
         Everyone:(OI)(CI)(IO)(GR,GE)

Successfully processed 1 files; Failed processing 0 files
```

These permissions can be revoked using the command `icacls c:\users /remove joe`.

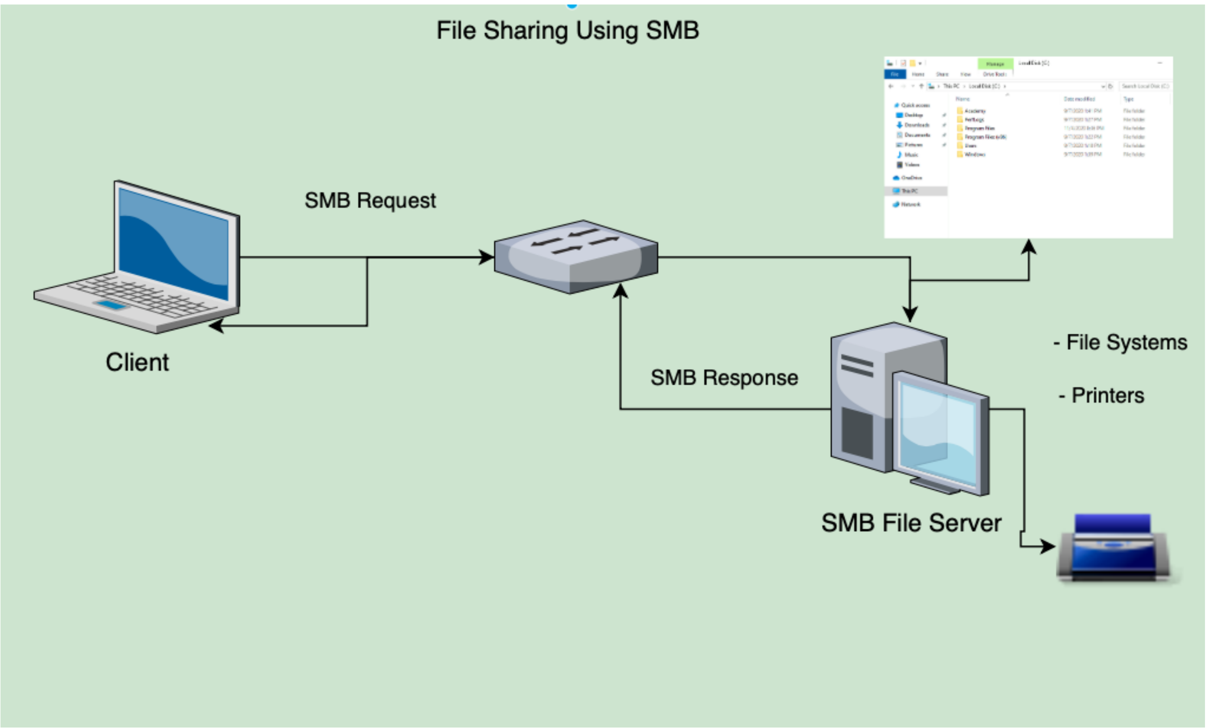
`icacls` is very powerful and can be used in a domain setting to give certain users or groups specific permissions over a file or folder, explicitly deny access, enable or disable inheritance permissions, and change directory/file ownership.

A full listing of `icacls` command-line arguments and detailed permission settings can be found here.

NTFS vs. Share Permissions

Microsoft owns over 70% of the global market share on desktop operating systems with Windows. This explains why most malware authors choose to write malware for Windows and why many perceive Windows as less secure than other operating systems. From a business perspective it just makes sense for malware authors to expend resources on writing malware for Windows. It is a high-value target. The idea that any OS is immune to malware is a technical fallacy. If software can be written for an operating system then a virus can be written for an operating system. Keep in mind that a virus, by definition, is software written with malicious intent and can be written for any OS. Many variants of malware written for Windows can spread over the network via network shares with lenient permissions applied. It is also worth noting that to this day, the infamous `EternalBlue` vulnerability still haunts unpatched Windows systems running `SMBv1` and often paves the way for ransomware to shut down organizations.

The `Server Message Block protocol` (`SMB`) is used in Windows to connect shared resources like files and printers. It is used in large, medium, and small enterprise environments. See the image below to visualize this concept:



Note: Any time you see a visualization/diagram of a concept, take your time to understand it thoroughly. A picture can be worth a thousand words but very tempting to skip over when reading.

NTFS permissions and share permissions are often understood to be the same. Please know that they are not the same but often apply to the same shared resource. Let's take a look at the individual permissions that can be set to secure/grant objects access to a network share hosted on a Windows OS running the NTFS file system.

Share permissions

Permission	Description
`Full Control`	Users are permitted to perform all actions given by Change and Read permissions as well as change permissions for NTFS files and subfolders
`Change`	Users are permitted to read, edit, delete and add files and subfolders
`Read`	Users are allowed to view file & subfolder contents

NTFS Basic permissions

Permission	Description
`Full Control`	Users are permitted to add, edit, move, delete files & folders as well as change NTFS permissions that apply to all allowed folders
`Modify`	Users are permitted or denied permissions to view and modify files and folders. This includes adding or deleting files
`Read & Execute`	Users are permitted or denied permissions to read the contents of files and execute programs
`List folder contents`	Users are permitted or denied permissions to view a listing of files and subfolders
`Read`	Users are permitted or denied permissions to read the contents of files
`Write`	Users are permitted or denied permissions to write changes to a file and add new files to a folder
`Special Permissions`	A variety of advanced permissions options

NTFS special permissions

Permission	Description
`Full control`	Users are permitted or denied permissions to add, edit, move, delete files & folders as well as change NTFS permissions that apply to all permitted folders
`Traverse folder / execute file`	Users are permitted or denied permissions to access a subfolder within a directory structure even if the user is denied access to contents at the parent folder level. Users may also be permitted or denied permissions to execute programs
`List folder/read data`	Users are permitted or denied permissions to view files and folders contained in the parent folder. Users can also be permitted to open and view files
`Read attributes`	Users are permitted or denied permissions to view basic attributes of a file or folder. Examples of basic attributes: system, archive, read-only, and hidden
`Read extended attributes`	Users are permitted or denied permissions to view extended attributes of a file or folder. Attributes differ depending on the program
`Create files/write data`	Users are permitted or denied permissions to create files within a folder and make changes to a file
`Create folders/append data`	Users are permitted or denied permissions to create subfolders within a folder. Data can be added to files but pre-existing content cannot be overwritten
`Write attributes`	Users are permitted or denied to change file attributes. This permission does not grant access to creating files or folders
`Write extended attributes`	Users are permitted or denied permissions to change extended attributes on a file or folder. Attributes differ depending on the program
`Delete subfolders and files`	Users are permitted or denied permissions to delete subfolders and files. Parent folders will not be deleted
`Delete`	Users are permitted or denied permissions to delete parent folders, subfolders and files.
`Read permissions`	Users are permitted or denied permissions to read permissions of a folder

Permission	Description
`Change permissions`	Users are permitted or denied permissions to change permissions of a file or folder
`Take ownership`	Users are permitted or denied permission to take ownership of a file or folder. The owner of a file has full permissions to change any permissions

Keep in mind that NTFS permissions apply to the system where the folder and files are hosted. Folders created in NTFS inherit permissions from parent folders by default. It is possible to disable inheritance to set custom permissions on parent and subfolders, as we will do later in this module. The share permissions apply when the folder is being accessed through SMB, typically from a different system over the network. This means someone logged in locally to the machine or via RDP can access the shared folder and files by simply navigating to the location on the file system and only need to consider NTFS permissions. The permissions at the NTFS level provide administrators much more granular control over what users can do within a folder or file.

Creating a Network Share

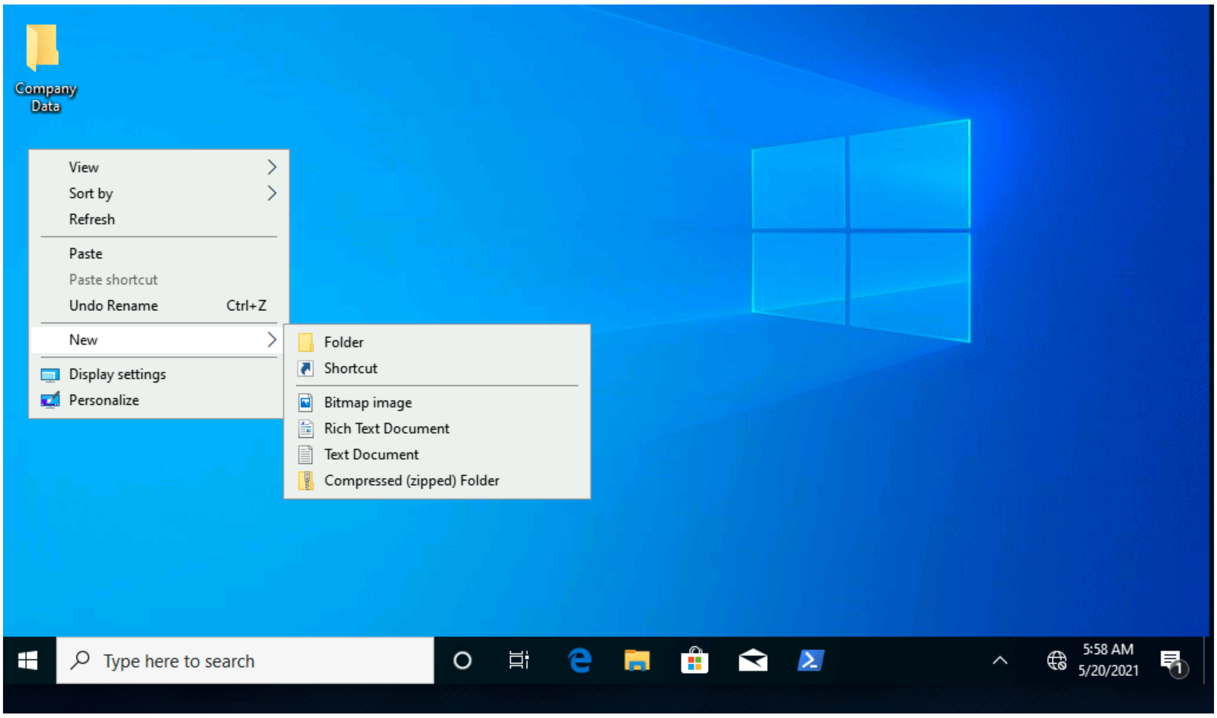
To get a solid fundamental understanding of SMB and it's relationship to NTFS, we will create a network share on the `Windows 10 target box`.

Note: It is an ideal learning experience to have the Pwnbox open full screen on a separate monitor so we may have at least one display dedicated to displaying the written content and one display for the boxes we are interacting with. Alternatively, if we only have access to one display, we can use that one for interactions with boxes and a smartphone or tablet to reference the written content.

In this case, we will create a shared folder by first creating a new folder on the Windows 10 desktop. Keep in mind that in most large enterprise environments, shares are created on a Storage Area Network (SAN), Network Attached Storage device (NAS), or a separate partition on drives accessed via a server operating system like Windows Server. If we ever come across shares on a desktop operating system, it will either be a small business or it could be a beachhead system used by a penetration tester or malicious attacker to gather and exfiltrate data.

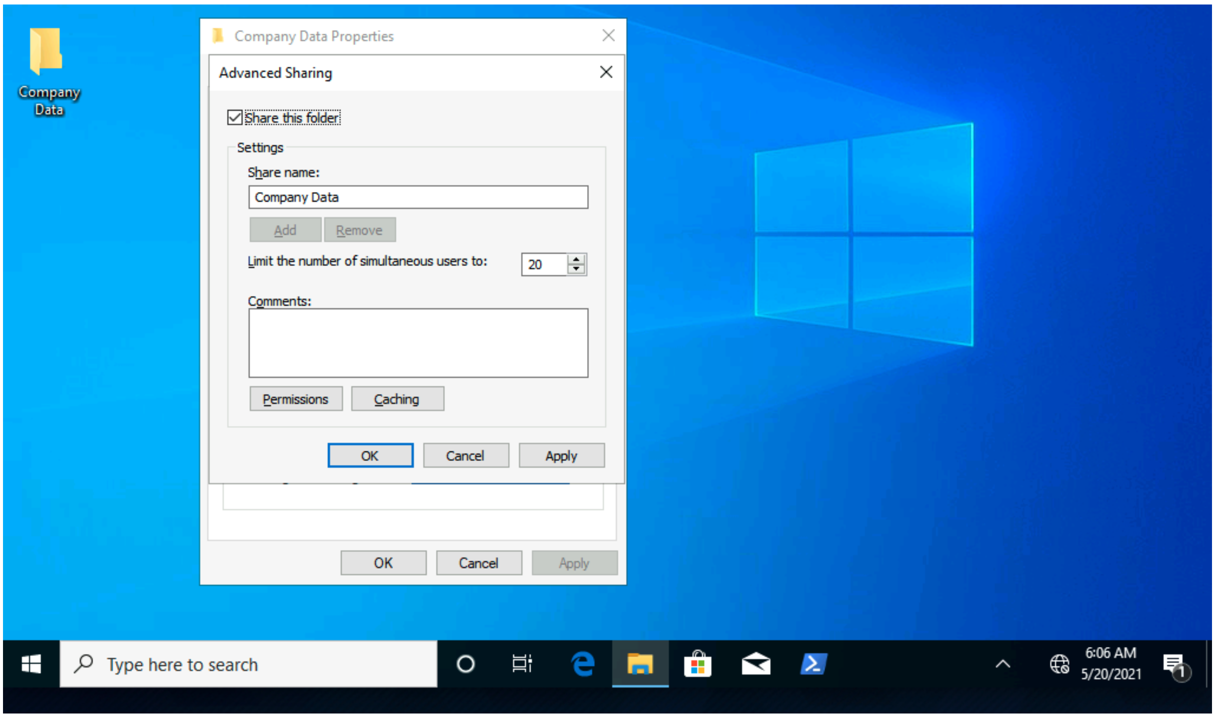
We will go through this process using the GUI in Windows.

Creating the Folder



We are going to use the `Advanced Sharing` option to configure our share.

Making the Folder a Share


Company Data Properties

Advanced Sharing

☑ Share this folder

Settings

Share name:

Company Data

Add Remove

Limit the number of simultaneous users to: 20

Comments:

Permissions Caching

OK Cancel Apply

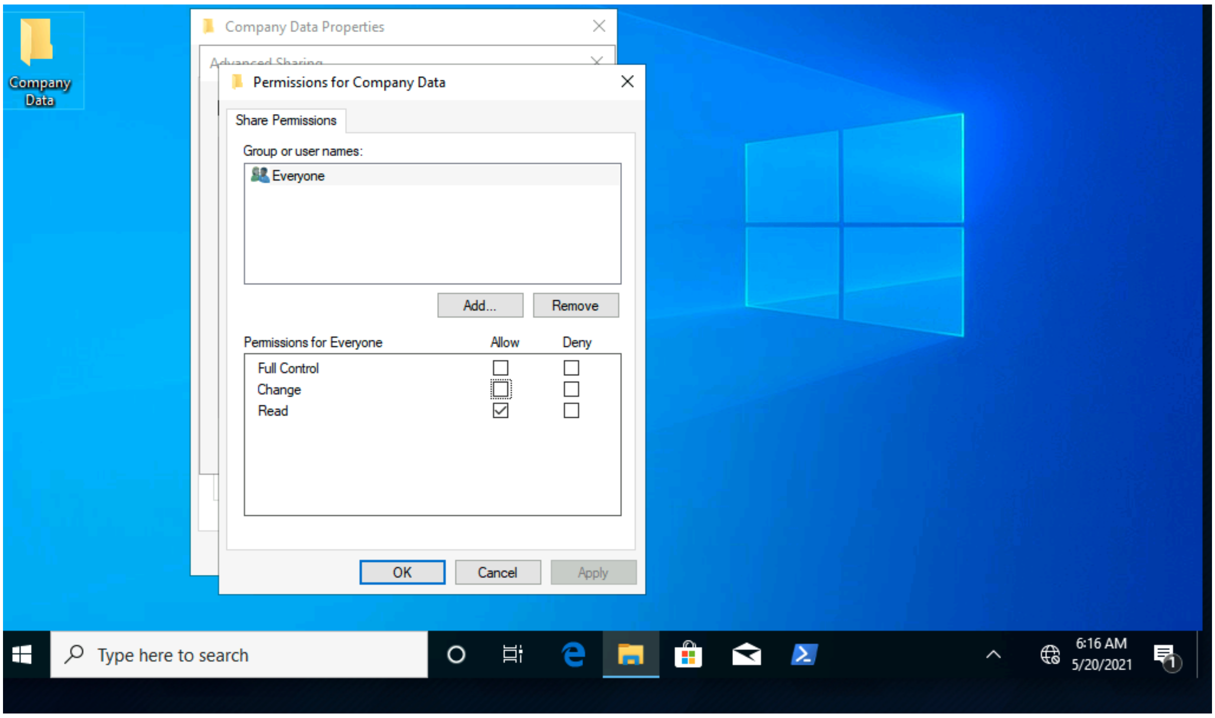
OK Cancel Apply

Notice how the share name automatically defaults to the name of the folder. Also, we can see that it is possible to limit the number of users that can be connected to this share simultaneously. In a real-world environment it is a good practice for administrators to set this number according to the number of users that regularly need access to the resource being shared.

Similar to NTFS permissions, there is an `access control list` (`ACL`) for shared resources. We can consider this the SMB permissions list. Keep in mind that with shared resources, both the SMB and NTFS permissions lists apply to every resource that gets shared in Windows. The ACL contains `access control entries` (`ACEs`). Typically these ACEs are made up of `users` & `groups` (also called security principals) as they are a suitable mechanism for managing and tracking access to shared resources.

Notice the default access control entry and permissions settings.

Share Permissions ACL (Sharing Tab)



Group or user names:

Everyone

Permissions for Everyone	Allow	Deny
Full Control	□	□
Change	□	□
Read	☑	□

For now, we are going to apply these settings to test the effect of this ACL and the permissions applied as-is. We will test connectivity from the Pwnbox by opening terminal and using `smbclient`.

Note: A server is technically a software function used to service the requests of a client. In this case, the Pwnbox is our client, and the Windows 10 target box is our server.

Using smbclient to Connect to the Share

```
smbclient -L IPaddressOfTarget -U htb-student
Enter WORKGROUP\htb-student's password:

        Sharename       Type      Comment
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        C$              Disk      Default share
        Company Data    Disk
        IPC$            IPC       Remote IPC
```

What could potentially block us from accessing this share if all our entries are correct and our permissions list has the Everyone group present with at least Read permissions?

Windows Defender Firewall Considerations

It is the Windows Defender Firewall that could potentially be blocking access to the SMB share. Since we are connecting from a Linux-based system the firewall has blocked access from any device that is not joined to the same `workgroup`. It is also important to note that when a Windows system is part of a workgroup, all `netlogon` requests are authenticated against that particular Windows system's `SAM` database. When a Windows system is joined to a Windows Domain environment, all netlogon requests are authenticated against `Active Directory`. The primary difference between a workgroup and a Windows Domain in terms of authentication, is with a workgroup the local SAM database is used and in a Windows Domain a centralized network-based database (Active Directory) is used. We must know this information when attempting to logon & authenticate with a Windows system. Consider where the htb-student account is hosted to properly connect to the target.

In terms of the firewall blocking connections, this can be tested by completely deactivating each firewall profile in Windows or by enabling specific predefined inbound firewall rules in the `Windows Defender Firewall advanced security settings`. Like most firewalls, Windows Defender Firewall permits or denies traffic (access & connection requests in this case) flowing `inbound` &/or `outbound`

The different inbound and outbound rules are associated with the different firewall profiles in defender.

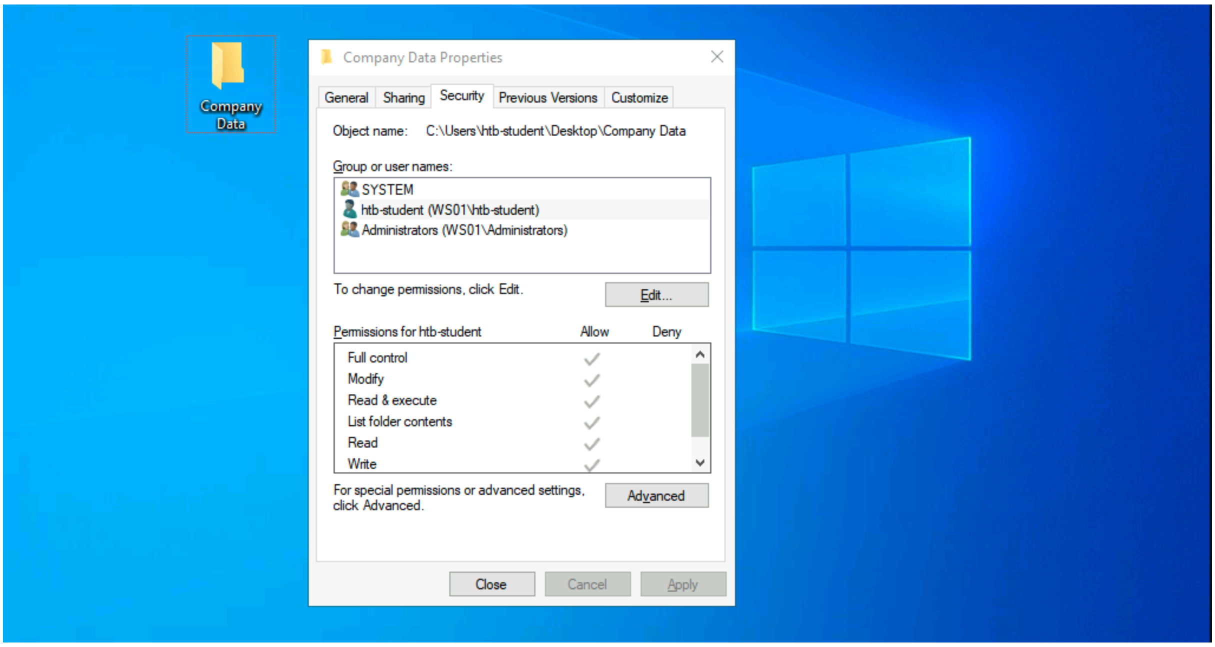
Windows Defender Firewall Profiles:

- `Public`
- `Private`
- `Domain`

It is a best practice to enable predefined rules or add custom exceptions rather than deactivating the firewall altogether. Unfortunately, it is very common for firewalls to be left completely deactivated for the sake of convenience or lack of understanding. Firewall rules on desktop systems can be centrally managed when joined to a Windows Domain environment through the use of Group Policy. Group Policy concepts and configurations are outside of the scope of this module.

Once the proper `inbound` firewall rules are enabled we will successfully connect to the share. Keep in mind that we can only connect to the share because the user account we are using (`htb-student`) is in the `Everyone group`. Recall that we left the specific share permissions for the Everyone group set to Read, which quite literally means we will only be able to Read files on this share. Once a connection is established with a share, we can create a `mount point` from our Pwnbox to the Windows 10 target box's file system. This is where we must also consider that NTFS permissions apply alongside share permissions. Recall that NTFS is the default file system in Windows. Lets jump back to our xfreerdp session with our Windows 10 target box and take a look at the NTFS permissions on the Company Data folder.

NTFS Permissions ACL (Security Tab)



Object name: C:\Users\htb-student\Desktop\Company Data

Group or user names:
SYSTEM
htb-student (WS01\htb-student)
Administrators (WS01\Administrators)

To change permissions, click Edit.

Permissions for htb-student	Allow	Deny
Full control	✓	
Modify	✓	
Read & execute	✓	
List folder contents	✓	
Read	✓	
Write	✓	

For special permissions or advanced settings, click Advanced.

There's more granular control with NTFS permissions that can be applied to users and groups. Anytime we see a gray checkmark next to a permission, it was inherited from a parent directory. By default, all NTFS permissions are inherited from the parent directory. In the Windows world, the `C:\ drive` is the parent directory to rule all directories unless a system administrator were to disable inheritance inside a newly created folder's advanced Security settings.

In many cases, the system administrator(s) of an organization would be responsible for deciding what permissions a user or group of users gets over network resources. This is why many spear-phishing attacks are directed at system administrators and other IT leaders. They have lots of influence over what is allowed in the environments they oversee, even more so than an organization's non-technical c-level leaders in many cases. For example, the doctors or executives working in a hospital will not have administrative rights over the network, but the system administrators will.

Now lets give the Everyone group `Full control` at the share level and test the impact of the change by trying to create a mount point to the share from the Desktop of our Pwnbox

Mounting to the Share

```
sudo mount -t cifs -o username=htb-student,password=Academy_WinFun! //ipaddoftarget/"Company Data" /home/user/Desktop/
```

If this command is not working check the syntax. If the syntax is correct yet the command is still not working, `cifs-utils` may need to be installed. This can be done with the following command:

Installing CIFS Utilities

```
sudo apt-get install cifs-utils
```

Once we have successfully created the mount point on the Desktop on our Pwnbox, we should look at a couple of tools built-in to Windows that will allow us to track and monitor what we have done.

The `net share` command allows us to view all the shared folders on the system. Notice the share we created and also the C:\ drive.

`Do you remember us sharing the C:\ drive?`

We didn't manually share C:. The most important drive with the most critical files on a Windows system is shared via SMB at install. This means anyone with the proper access could remotely access the entire C:\ of each Windows system on a network.

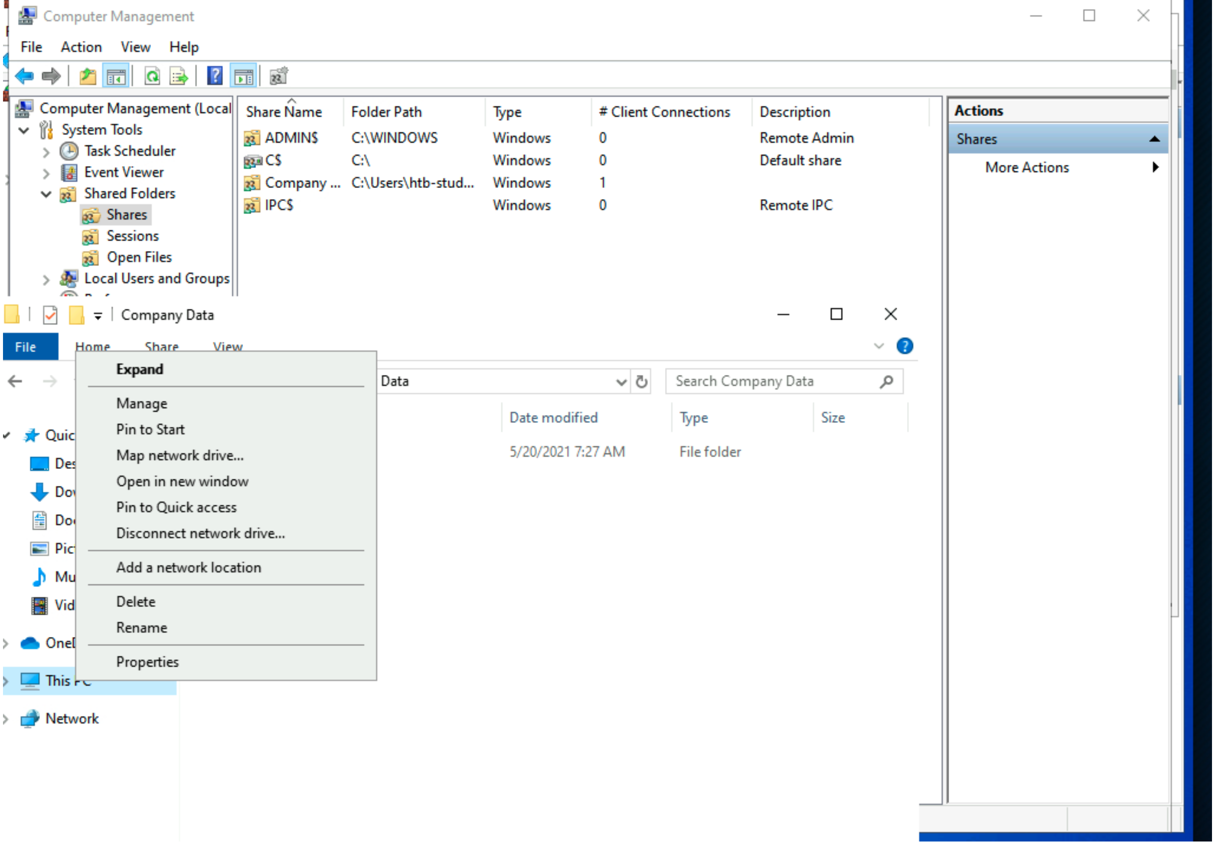
We can also see the share we created.

Displaying Shares using net share

```
C:\Users\htb-student> net share

Share name   Resource                        Remark

-------------------------------------------------------------------------------
C$           C:\                             Default share
IPC$                                         Remote IPC
ADMIN$       C:\WINDOWS                      Remote Admin
Company Data C:\Users\htb-student\Desktop\Company Data

The command completed successfully.
```

`Computer Management` is another tool we can use to identify and monitor shared resources on a Windows system.

Monitoring Shares from Computer Management



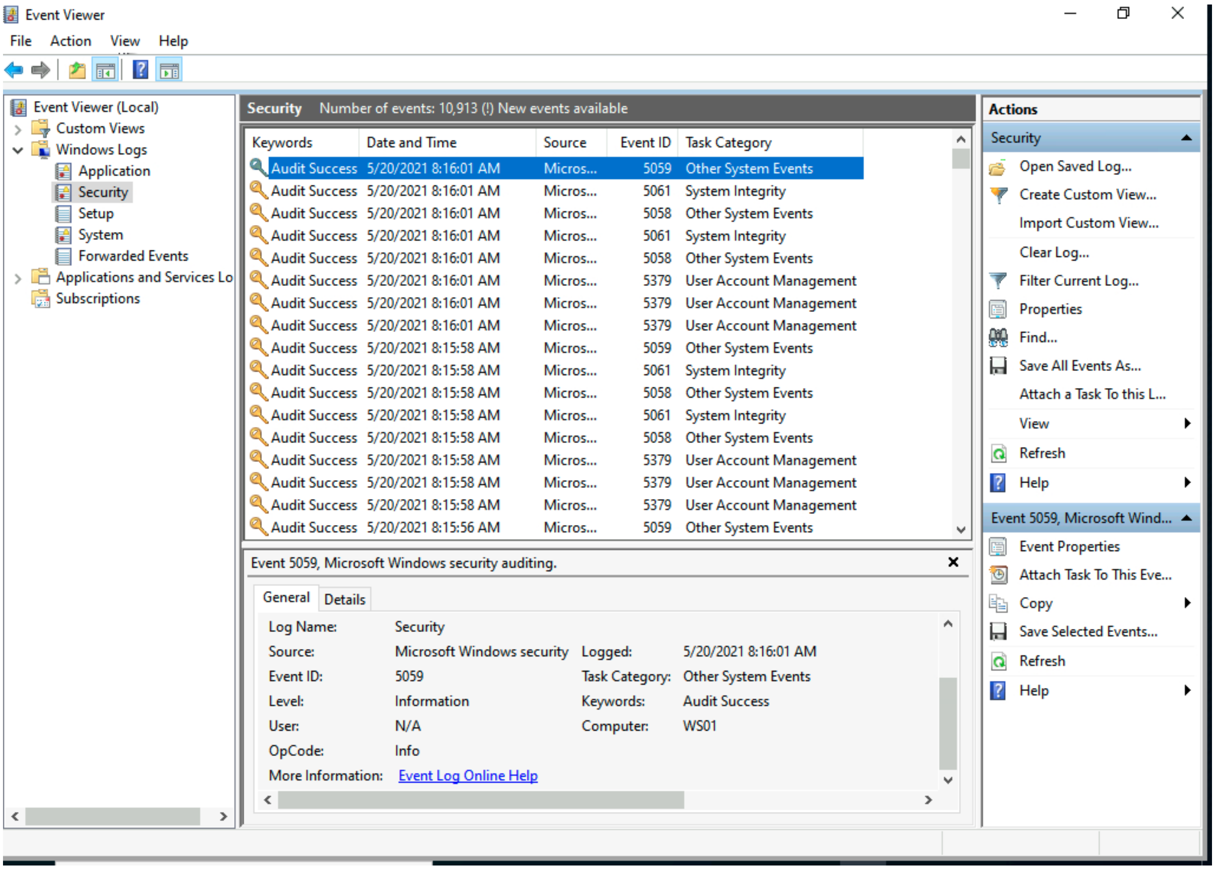
Share Name	Folder Path	Type	# Client Connections	Description
ADMIN$	C:\WINDOWS	Windows	0	Remote Admin
C$	C:\	Windows	0	Default share
Company ...	C:\Users\htb-stud...	Windows	1	
IPC$		Windows	0	Remote IPC

	Date modified	Type	Size
	5/20/2021 7:27 AM	File folder	

We can poke around in `Shares`, `Sessions`, and `Open Files` to get an idea of what information this provides us. Should there be a situation where we assist an individual or organization with responding to a breach related to SMB, these are some great places to check and start to understand how the breach may have happened and what may have been left behind.

Viewing Share access logs in Event Viewer

`Event Viewer` is another good place to investigate actions completed on Windows. Almost every operating system has a logging mechanism and a utility to view the logs that were captured. Know that a log is like a journal entry for a computer, where the computer writes down all the actions that were performed and numerous details associated with that action. We can view the logs created for every action we performed when accessing the Windows 10 target box, as well as when creating, editing and accessing the shared folder.



Security Number of events: 10,913 (!) New events available

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	5/20/2021 8:16:01 AM	Micros...	5059	Other System Events
Audit Success	5/20/2021 8:16:01 AM	Micros...	5061	System Integrity
Audit Success	5/20/2021 8:16:01 AM	Micros...	5058	Other System Events
Audit Success	5/20/2021 8:16:01 AM	Micros...	5061	System Integrity
Audit Success	5/20/2021 8:16:01 AM	Micros...	5058	Other System Events
Audit Success	5/20/2021 8:16:01 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:16:01 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:16:01 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:15:58 AM	Micros...	5059	Other System Events
Audit Success	5/20/2021 8:15:58 AM	Micros...	5061	System Integrity
Audit Success	5/20/2021 8:15:58 AM	Micros...	5058	Other System Events
Audit Success	5/20/2021 8:15:58 AM	Micros...	5061	System Integrity
Audit Success	5/20/2021 8:15:58 AM	Micros...	5058	Other System Events
Audit Success	5/20/2021 8:15:58 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:15:58 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:15:58 AM	Micros...	5379	User Account Management
Audit Success	5/20/2021 8:15:56 AM	Micros...	5059	Other System Events

Event 5059, Microsoft Windows security auditing.

General Details

Log Name:	Security		
Source:	Microsoft Windows security	Logged:	5/20/2021 8:16:01 AM
Event ID:	5059	Task Category:	Other System Events
Level:	Information	Keywords:	Audit Success
User:	N/A	Computer:	WS01
OpCode:	Info		
More Information:	Event Log Online Help		

Windows Services & Processes

Windows Services

Services are a major component of the Windows operating system. They allow for the creation and management of long-running processes. Windows services can be started automatically at system boot without user intervention. These services can continue to run in the background even after the user logs out of their account on the system.

Applications can also be created to install as a service, such as a network monitoring application installed on a server. Services on Windows are responsible for many functions within the Windows operating system, such as networking functions, performing system diagnostics, managing user credentials, controlling Windows updates, and more.

Windows services are managed via the Service Control Manager (SCM) system, accessible via the `services.msc` MMC add-in.

This add-in provides a GUI interface for interacting with and managing services and displays information about each installed service. This information includes the service Name, Description, Status, Startup Type, and the user that the service runs under.

It is also possible to query and manage services via the command line using `sc.exe` using PowerShell cmdlets such as `Get-Service`.

```
PS C:\htb> Get-Service | ? {$_.Status -eq "Running"} | select -First 2 |fl

Name                : AdobeARMservice
DisplayName         : Adobe Acrobat Update Service
Status              : Running
DependentServices   : {}
ServicesDependedOn  : {}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : True
ServiceType         : Win32OwnProcess

Name                : Appinfo
DisplayName         : Application Information
Status              : Running
DependentServices   : {}
ServicesDependedOn  : {RpcSs, ProfSvc}
CanPauseAndContinue : False
CanShutdown         : False
CanStop             : True
ServiceType         : Win32OwnProcess, Win32ShareProcess
```

Service statuses can appear as Running, Stopped, or Paused, and they can be set to start manually, automatically, or on a delay at system boot. Services can also be shown in the state of Starting or Stopping if some action has triggered them to either start or stop. Windows has three categories of services: Local Services, Network Services, and System Services. Services can usually only be created, modified, and deleted by users with administrative privileges. Misconfigurations around service permissions are a common privilege escalation vector on Windows systems.

In Windows, we have some critical system services that cannot be stopped and restarted without a system restart. If we update any file or resource in use by one of these services, we must restart the system.

Service	Description
smss.exe	Session Manager SubSystem. Responsible for handling sessions on the system.
csrss.exe	Client Server Runtime Process. The user-mode portion of the Windows subsystem.
wininit.exe	Starts the Wininit file .ini file that lists all of the changes to be made to Windows when the computer is restarted after installing a program.
logonui.exe	Used for facilitating user login into a PC
lsass.exe	The Local Security Authentication Server verifies the validity of user logons to a PC or server. It generates the process responsible for authenticating users for the Winlogon service.
services.exe	Manages the operation of starting and stopping services.
winlogon.exe	Responsible for handling the secure attention sequence, loading a user profile on logon, and locking the computer when a screensaver is running.
System	A background system process that runs the Windows kernel.
svchost.exe with RPCSS	Manages system services that run from dynamic-link libraries (files with the extension .dll) such as "Automatic Updates," "Windows Firewall," and "Plug and Play." Uses the Remote Procedure Call (RPC) Service (RPCSS).
svchost.exe with Dcom/PnP	Manages system services that run from dynamic-link libraries (files with the extension .dll) such as "Automatic Updates," "Windows Firewall," and "Plug and Play." Uses the Distributed Component Object Model (DCOM) and Plug and Play (PnP) services.

This link has a list of Windows components, including key services.

Processes

Processes run in the background on Windows systems. They either run automatically as part of the Windows operating system or are started by other installed applications.

Processes associated with installed applications can often be terminated without causing a severe impact on the operating system. Certain processes are critical and, if terminated, will stop certain components of the operating system from running properly. Some examples include the Windows Logon Application, System, System Idle Process, Windows Start-Up Application, Client Server Runtime, Windows Session Manager, Service Host, and Local Security Authority Subsystem Service (LSASS) process.

Local Security Authority Subsystem Service (LSASS)

`lsass.exe` is the process that is responsible for enforcing the security policy on Windows systems. When a user attempts to log on to the system, this process verifies their log on attempt and creates access tokens based on the user's permission levels. LSASS is also responsible for user account password changes. All events associated with this process (logon/logoff attempts, etc.) are logged within the Windows Security Log. LSASS is an extremely high-value target as several tools exist to extract both cleartext and hashed credentials stored in memory by this process.

Sysinternals Tools

The SysInternals Tools suite is a set of portable Windows applications that can be used to administer Windows systems (for the most part without requiring installation). The tools can be either downloaded from the Microsoft website or by loading them directly from an internet-accessible file share by typing `\\live.sysinternals.com\tools` into a Windows Explorer window.

For example, we can run procdump.exe directly from this share without downloading it directly to disk.

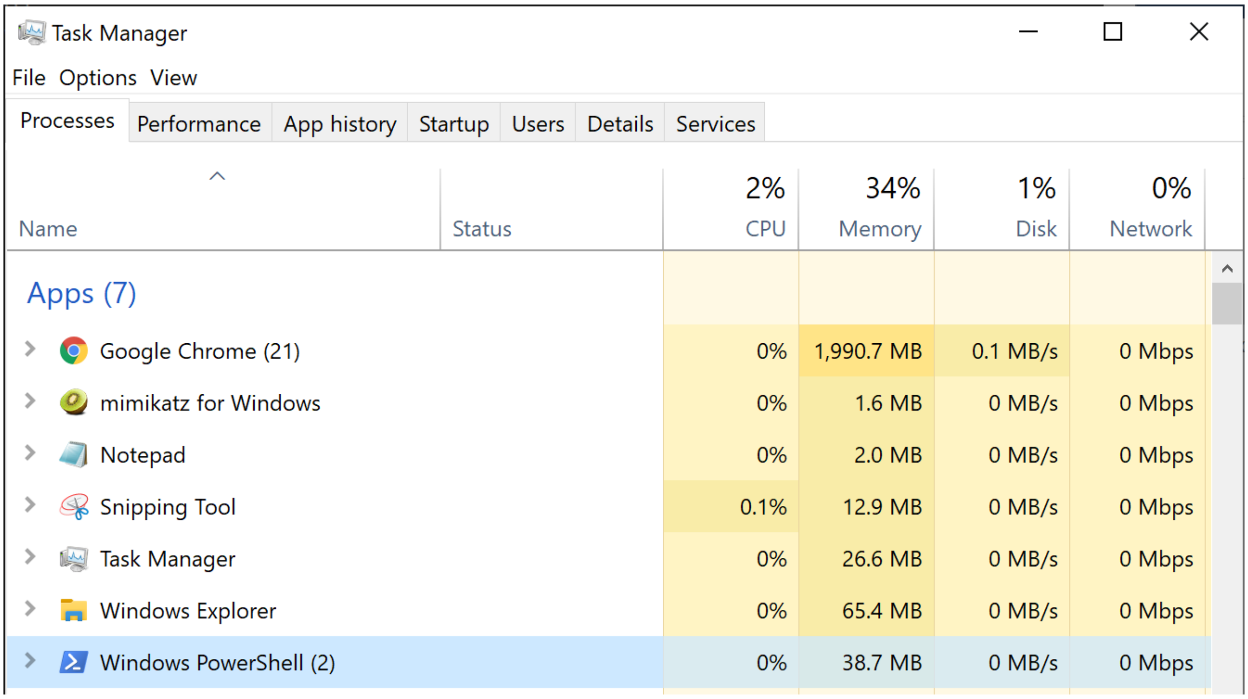
```
C:\htb> \\live.sysinternals.com\tools\procdump.exe -accepteula

ProcDump v9.0 - Sysinternals process dump utility
Copyright (C) 2009-2017 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

Monitors a process and writes a dump file when the process exceeds the
specified criteria or has an exception.

Capture Usage:
   procdump.exe [-mm] [-ma] [-mp] [-mc Mask] [-md Callback_DLL] [-mk]
                [-n Count]
                [-s Seconds]
                [-c|-cl CPU_Usage [-u]]
                [-m|-ml Commit_Usage]
                [-p|-pl Counter_Threshold]
                [-h]
                [-e [1 [-g] [-b]]]
                [-l]
                [-t]
                [-f  Include_Filter, ...]
                [-fx Exclude_Filter, ...]
                [-o]
                [-r [1..5] [-a]]
                [-wer]
                [-64]
                {
                 {{[-w] Process_Name | Service_Name | PID} [Dump_File | Dump_Folder]}
                |
                 {-x Dump_Folder Image_File [Argument, ...]}
                }

<SNIP>
```

The suite includes tools such as `Process Explorer`, an enhanced version of `Task Manager`, and `Process Monitor`, which can be used to monitor file system, registry, and network activity related to any process running on the system. Some additional tools are TCPView, which is used to monitor internet activity, and PSExec, which can be used to manage/connect to systems via the SMB protocol remotely.

These tools can be useful for penetration testers to, for example, discover interesting processes and possible privilege escalation paths as well as for lateral movement.

Task Manager

Windows Task Manager is a powerful tool for managing Windows systems. It provides information about running processes, system performance, running services, startup programs, logged-in users/logged in user processes, and services. Task Manager can be opened by right-clicking on the taskbar and selecting `Task Manager`, pressing ctrl + shift + Esc, pressing ctrl + alt + del and selecting `Task Manager`, opening the start menu and typing `Task Manager`, or typing `taskmgr` from a CMD or PowerShell console.



Name	Status	2% CPU	34% Memory	1% Disk	0% Network
Apps (7)					
Google Chrome (21)		0%	1,990.7 MB	0.1 MB/s	0 Mbps
mimikatz for Windows		0%	1.6 MB	0 MB/s	0 Mbps
Notepad		0%	2.0 MB	0 MB/s	0 Mbps
Snipping Tool		0.1%	12.9 MB	0 MB/s	0 Mbps
Task Manager		0%	26.6 MB	0 MB/s	0 Mbps
Windows Explorer		0%	65.4 MB	0 MB/s	0 Mbps
Windows PowerShell (2)		0%	38.7 MB	0 MB/s	0 Mbps

Tab	Description
Processes tab	Shows a list of running applications and background processes along with the CPU, memory, disk, network, and power usage for each.
Performance tab	Shows graphs and data such as CPU utilization, system uptime, memory usage, disk and, networking, and GPU usage. We can also open the `Resource Monitor`, which gives us a much more in-depth view of the current CPU, Memory, Disk, and Network resource usage.

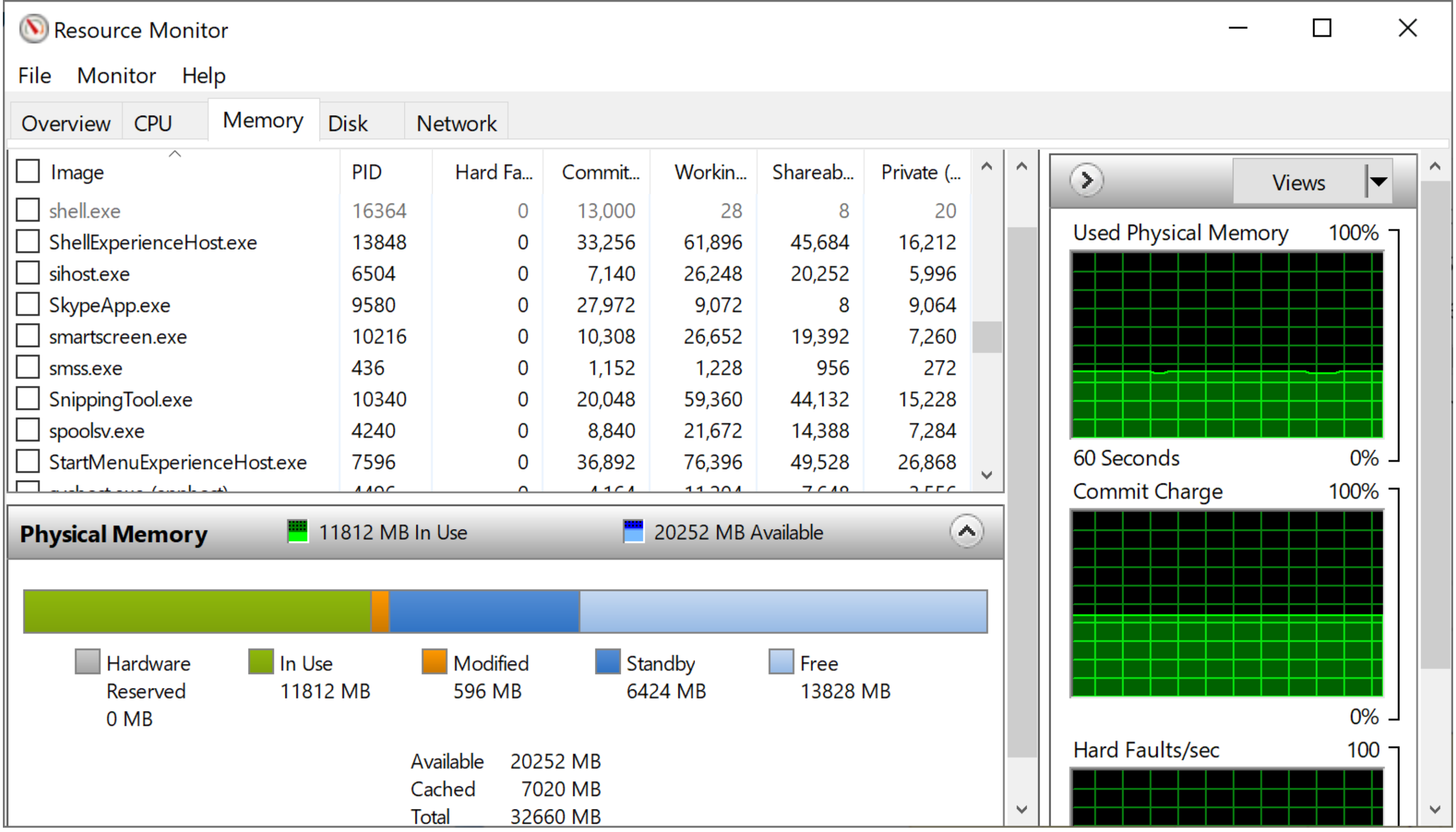


Image	PID	Hard Fa...	Commit...	Workin...	Shareab...	Private (...
□ shell.exe	16364	0	13,000	28	8	20
□ ShellExperienceHost.exe	13848	0	33,256	61,896	45,684	16,212
□ sihost.exe	6504	0	7,140	26,248	20,252	5,996
□ SkypeApp.exe	9580	0	27,972	9,072	8	9,064
□ smartscreen.exe	10216	0	10,308	26,652	19,392	7,260
□ smss.exe	436	0	1,152	1,228	956	272
□ SnippingTool.exe	10340	0	20,048	59,360	44,132	15,228
□ spoolsv.exe	4240	0	8,840	21,672	14,388	7,284
□ StartMenuExperienceHost.exe	7596	0	36,892	76,396	49,528	26,868

Physical Memory 🟩 11812 MB In Use 🟦 20252 MB Available

Hardware Reserved	In Use	Modified	Standby	Free
0 MB	11812 MB	596 MB	6424 MB	13828 MB

Available 20252 MB
Cached 7020 MB
Total 32660 MB

Tab	Description
App history tab	Shows resource usage for the current user account for each application for a period of time.
Startup tab	Shows which applications are configured to start at boot as well as the impact on the startup process.
Users tab	Shows logged in users and the processes/resource usage associated with their session.
Details tab	Shows the name, process ID (PID), status, associated username, CPU, and memory usage for each running application.
Services tab	Shows the name, PID, description, and status of each installed service. The Services add-in can be accessed from this tab as well.

Process Explorer

Process Explorer is a part of the Sysinternals tool suite. This tool can show which handles and DLL processes are loaded when a program runs. Process Explorer shows a list of currently running processes, and from there, we can see what handles the process has selected in one view or the DLLs and memory-swapped files that have been loaded in another view. We can also search within the tool to show which processes tie back to a specific handle or DLL. The tool can also be used to analyze parent-child process relationships to see what child processes are spawned by an application and help troubleshoot any issues such as orphaned processed that can be left behind when a process is terminated.

Service Permissions

Recall that services allow for the management of long-running processes and are a critical part of Windows operating systems. Sysadmins often overlook them as potential threat vectors that can be used to load malicious DLLs, execute applications without access to an admin account, escalate privileges and even maintain persistence. These threat vectors in Windows services often come into existence through service permissions misconfigurations put in place by 3rd party software and easy to make mistakes by admins during install processes.

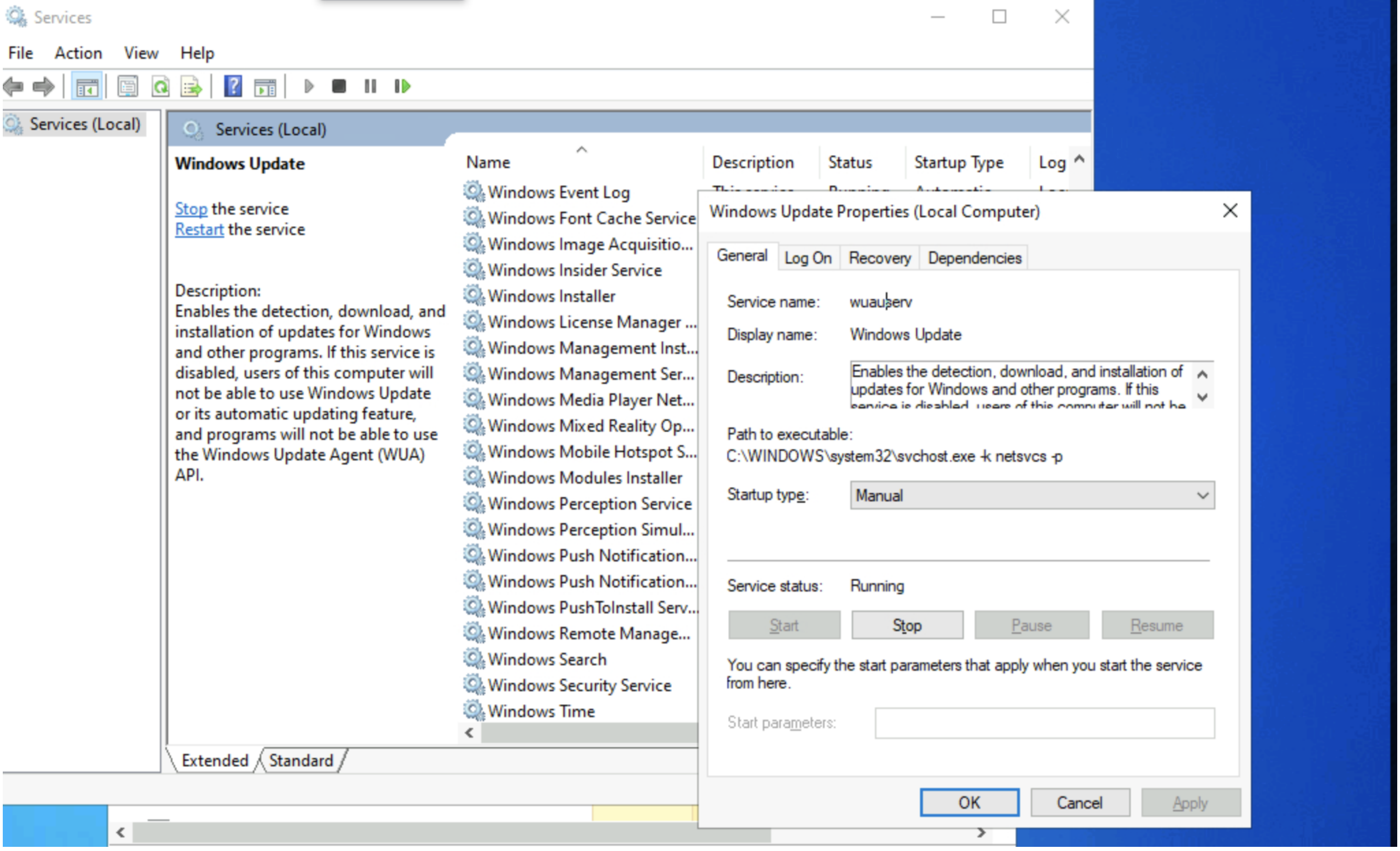
The first step in realizing the importance of service permissions is simply understanding that they exist and being mindful of them. On server operating systems, critical network services like DHCP and Active Directory Domain Services commonly get installed using the account assigned to the admin performing the install. Part of the install process includes assigning a specific service to run using the credentials and privileges of a designated user, which by default is set within the currently logged-on user context.

For example, if we are logged on as Bob on a server during DHCP install, then that service will be configured to run as Bob unless specified otherwise. What bad things could come of this? Well, what if Bob leaves the organization or gets fired? The typical business practice would be to disable Bob's account as part of his exit process. In this case, what would happen to DHCP and other services running using Bob's account? Those services would fail to start. DHCP or Dynamic Host Configuration Protocol is responsible for leasing IP addresses to computers on the network. If this service stops on a Windows DHCP server, clients requesting an IP address will not receive one. This means a service misconfiguration could lead to downtime and loss of productivity. It is highly recommended to create an individual user account to run critical network services. These are referred to as service accounts.

We should also be mindful of service permissions and the permissions of the directories they execute from because it is possible to replace the path to an executable with a malicious DLL or executable file. Let's examine the permissions of services running on Windows 10 to get an even better understanding of this.

Examining Services using services.msc

As discussed in the processes and services section, we can use `services.msc` to view and manage just about every detail regarding all services. Let's take a closer look at the service associated with `Windows Update` (`wuauserv`).



Services

Services (Local)

Windows Update

Stop the service
Restart the service

Description:
Enables the detection, download, and installation of updates for Windows and other programs. If this service is disabled, users of this computer will not be able to use Windows Update or its automatic updating feature, and programs will not be able to use the Windows Update Agent (WUA) API.

Name: Windows Event Log, Windows Font Cache Service, Windows Image Acquisitio..., Windows Insider Service, Windows Installer, Windows License Manager ..., Windows Management Inst..., Windows Management Ser..., Windows Media Player Net..., Windows Mixed Reality Op..., Windows Mobile Hotspot S..., Windows Modules Installer, Windows Perception Service, Windows Perception Simul..., Windows Push Notification..., Windows Push Notification..., Windows PushToInstall Serv..., Windows Remote Manage..., Windows Search, Windows Security Service, Windows Time

Windows Update Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: wuauserv
Display name: Windows Update
Description: Enables the detection, download, and installation of updates for Windows and other programs. If this service is disabled, users of this computer will not be

Path to executable:
C:\WINDOWS\system32\svchost.exe -k netsvcs -p

Startup type: Manual

Service status: Running

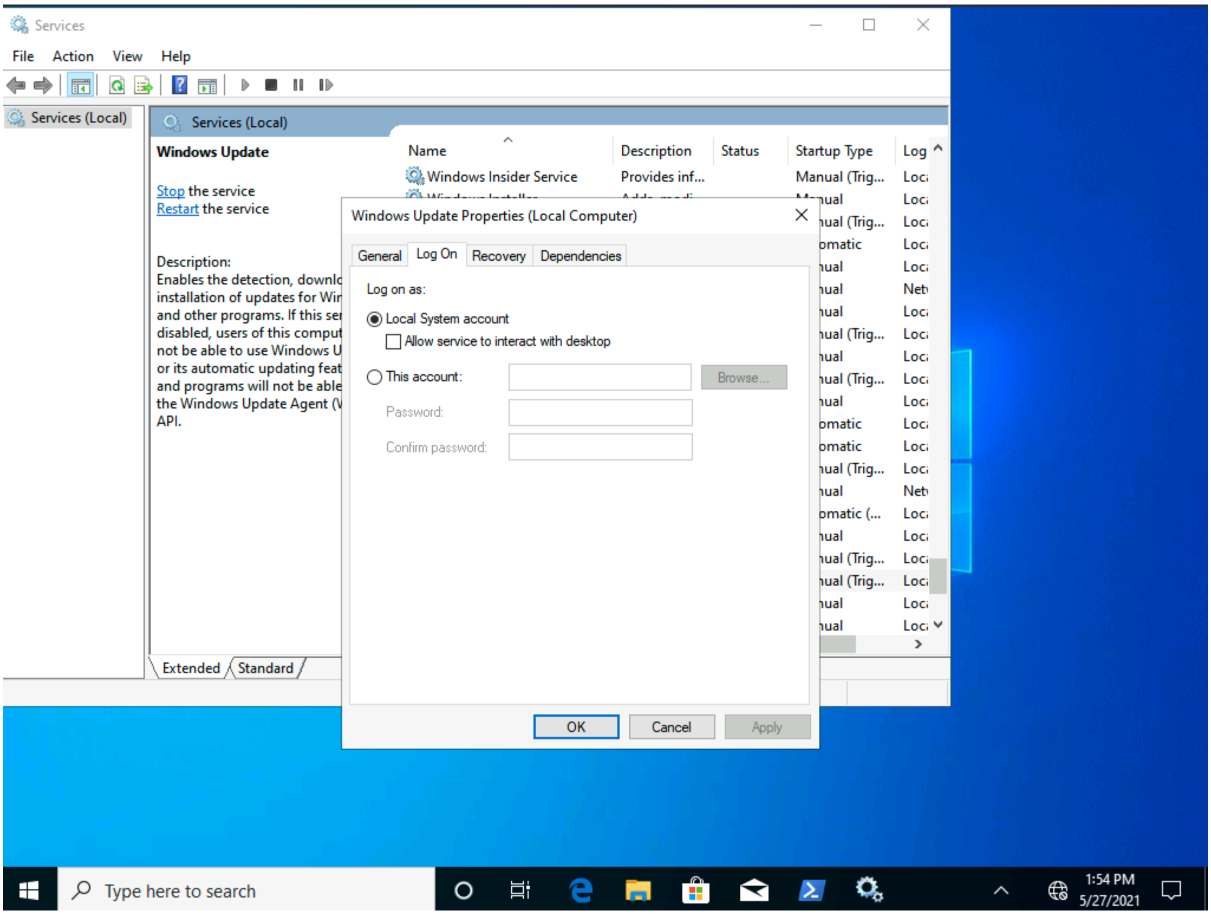
Start | Stop | Pause | Resume

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

Start parameters:

OK | Cancel | Apply

Make a note of the different properties available for viewing and configuration. Knowing the service name is especially useful when using command-line tools to examine and manage services. Path to the executable is the full path to the program and command to execute when the service starts. If the NTFS permissions of the destination directory are configured with weak permissions, an attacker could replace the original executable with one created for malicious purposes. We discuss NTFS permissions more in the NTFS vs. Share permissions section of this module.



Windows Update Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

● Local System account

□ Allow service to interact with desktop

○ This account: | Browse...

Password:

Confirm password:

OK | Cancel | Apply

Most services run with LocalSystem privileges by default which is the highest level of access allowed on an individual Windows OS. Not all applications need Local System account-level permissions, so it is beneficial to perform research on a case-by-case basis when considering installing new applications in a Windows environment. It is a good practice to identify applications that can run with the least privileges possible to align with the principle of least privilege.

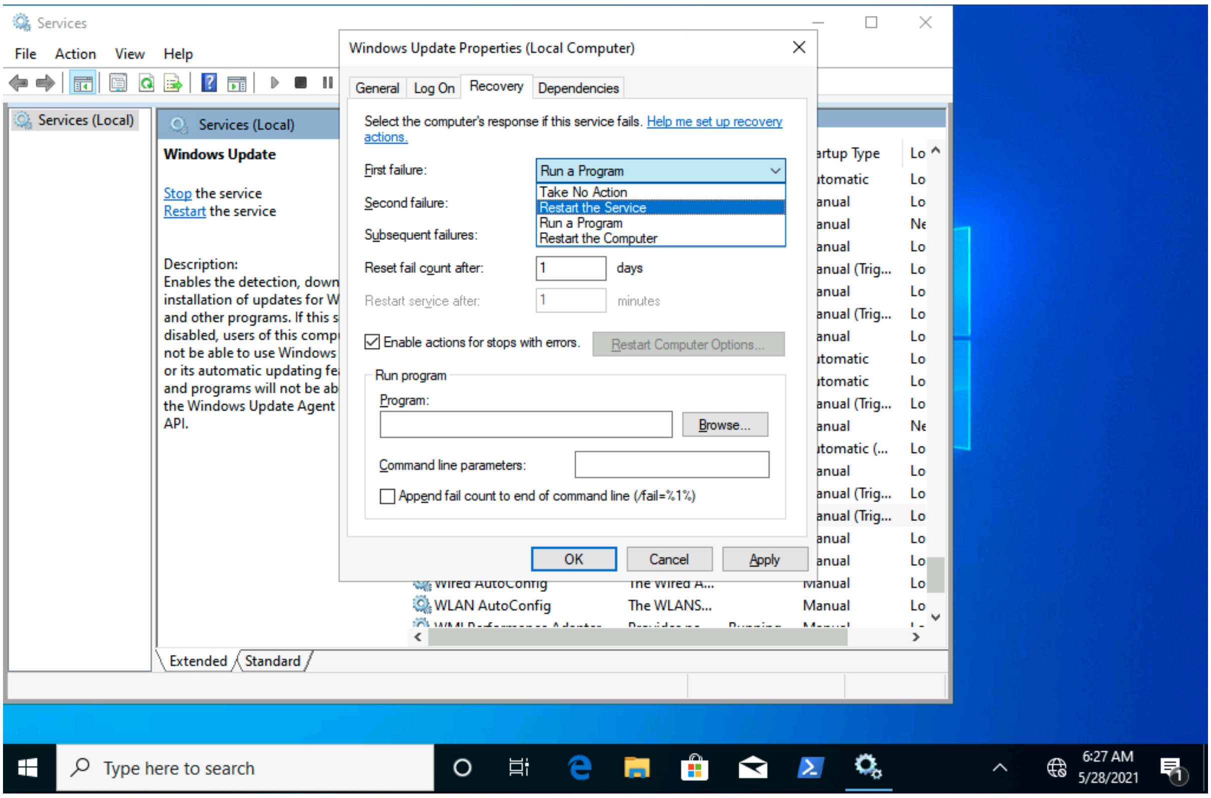
[Here is one breakdown of the principle of least privilege](#)

Notable built-in service accounts in Windows:

- LocalService

- NetworkService
- LocalSystem

Note: We can also create new accounts and use them for the sole purpose of running a service.



Windows Update Properties (Local Computer)

General | Log On | Recovery | Dependencies

Select the computer's response if this service fails. Help me set up recovery actions.

First failure: Run a Program

- Take No Action
- Restart the Service
- Run a Program
- Restart the Computer

Second failure:

Subsequent failures:

Reset fail count after: 1 days

Restart service after: 1 minutes

☑ Enable actions for stops with errors. Restart Computer Options...

Run program

Program:

Command line parameters:

☐ Append fail count to end of command line (/fail=%1%)

The recovery tab allows steps to be configured should a service fail. Notice how this service can be set to run a program after the first failure. This is yet another vector that an attacker could use to run malicious programs by utilizing a legitimate service.

Examining services using sc

Sc can also be used to configure and manage services. Let's experiment with a few commands.

```
C:\Users\htb-student>sc qc wuauserv
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: wuauserv
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\WINDOWS\system32\svchost.exe -k netsvcs -p
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Windows Update
        DEPENDENCIES       : rpcss
        SERVICE_START_NAME : LocalSystem
```

The `sc qc` command is used to query the service. This is where knowing the names of services can come in handy. If we wanted to query a service on a device over the network, we could specify the hostname or IP address immediately after `sc`.

```
C:\Users\htb-student>sc //hostname or ip of box query ServiceName
```

We can also use sc to start and stop services.

```
C:\Users\htb-student> sc stop wuauserv

[SC] OpenService FAILED 5:

Access is denied.
```

Notice how we are denied access from performing this action without running it within an administrative context. If we run a command prompt with `elevated privileges`, we will be permitted to complete this action.

```
C:\WINDOWS\system32> sc config wuauserv binPath=C:\Winbows\Perfectlylegitprogram.exe

[SC] ChangeServiceConfig SUCCESS

C:\WINDOWS\system32> sc qc wuauserv

[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: wuauserv
```

```
        TYPE               : 20  WIN32_SHARE_PROCESS
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Winbows\Perfectlylegitprogram.exe
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : Windows Update
        DEPENDENCIES       : rpcss
        SERVICE_START_NAME : LocalSystem
```

If we were investigating a situation where we suspected that the system had malware, sc would give us the ability to quickly search and analyze commonly targeted services and newly created services. It's also much more script-friendly than utilizing GUI tools like `services.msc`.

Another helpful way we can examine service permissions using `sc` is through the `sdshow` command.

```
C:\WINDOWS\system32> sc sdshow wuauserv

D:(A;;CCLCSWRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;SY)S:(AU;FA;CCDCLCSWRPWPDTLOSDRCWDWO;;;WD)
```

At an initial glance, the output looks crazy. It almost seems that we have done something wrong in our command, but there is a meaning to this madness. Every named object in Windows is a securable object, and even some unnamed objects are securable. If it's securable in a Windows OS, it will have a security descriptor. Security descriptors identify the object's owner and a primary group containing a `Discretionary Access Control List` (`DACL`) and a `System Access Control List` (`SACL`).

Generally, a DACL is used for controlling access to an object, and a SACL is used to account for and log access attempts. This section will examine the DACL, but the same concepts would apply to a SACL.

```
D:(A;;CCLCSWRPLORC;;;AU)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;SY)
```

This amalgamation of characters crunched together and delimited by opened and closed parentheses is in a format known as the `Security Descriptor Definition Language` (`SDDL`).

We may be tempted to read from left to right because that is how the English language is typically written, but it can be much different when interacting with computers. Read the entire security descriptor for the `Windows Update` (`wuauserv`) service in this order starting with the first letter and set of parentheses:

`D: (A;;CCLCSWRPLORC;;;AU)`

1. D: - the proceeding characters are DACL permissions
2. AU: - defines the security principal Authenticated Users
3. A;; - access is allowed
4. CC - SERVICE_QUERY_CONFIG is the full name, and it is a query to the service control manager (SCM) for the service configuration
5. LC - SERVICE_QUERY_STATUS is the full name, and it is a query to the service control manager (SCM) for the current status of the service
6. SW - SERVICE_ENUMERATE_DEPENDENTS is the full name, and it will enumerate a list of dependent services
7. RP - SERVICE_START is the full name, and it will start the service
8. LO - SERVICE_INTERROGATE is the full name, and it will query the service for its current status
9. RC - READ_CONTROL is the full name, and it will query the security descriptor of the service

As we read the security descriptor, it can be easy to get lost in the seemingly random order of characters, but recall that we are essentially viewing access control entries in an access control list. Each set of 2 characters in between the semi-colons represents actions allowed to be performed by a specific user or group.

`;;CCLCSWRPLORC;;;`

After the last set of semi-colons, the characters specify the security principal (User and/or Group) that is permitted to perform those actions.

`;;;AU`

The character immediately after the opening parentheses and before the first set of semi-colons defines whether the actions are Allowed or Denied.

`A;;`

This entire security descriptor associated with the `Windows Update` (`wuauserv`) service has three sets of access control entries because there are three different security principals. Each security principal has specific permissions applied.

Examine service permissions using PowerShell

Using the `Get-Acl` PowerShell cmdlet, we can examine service permissions by targeting the path of a specific service in the registry.

```
PS C:\Users\htb-student> Get-ACL -Path HKLM:\System\CurrentControlSet\Services\wuauserv | Format-List

Path   : Microsoft.PowerShell.Core\Registry::HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\wuauserv
Owner  : NT AUTHORITY\SYSTEM
Group  : NT AUTHORITY\SYSTEM
Access : BUILTIN\Users Allow  ReadKey
         BUILTIN\Users Allow  -2147483648
         BUILTIN\Administrators Allow  FullControl
         BUILTIN\Administrators Allow  268435456
         NT AUTHORITY\SYSTEM Allow  FullControl
         NT AUTHORITY\SYSTEM Allow  268435456
         CREATOR OWNER Allow  268435456
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  ReadKey
         APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES Allow  -2147483648
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         ReadKey
         S-1-15-3-1024-1065365936-1281604716-3511738428-1654721687-432734479-3232135806-4053264122-3456934681 Allow
         -2147483648
Audit  :
```

```
Sddl   : O:SYG:SYD:AI(A;ID;KR;;;BU)(A;CIIOID;GR;;;BU)(A;ID;KA;;;BA)(A;CIIOID;GA;;;BA)(A;ID;KA;;;SY)(A;CIIOID;GA;;;SY)(A
           ;CIIOID;GA;;;CO)(A;ID;KR;;;AC)(A;CIIOID;GR;;;AC)(A;ID;KR;;;S-1-15-3-1024-1065365936-1281604716-3511738428-1654
           721687-432734479-3232135806-4053264122-3456934681)(A;CIIOID;GR;;;S-1-15-3-1024-1065365936-1281604716-351173842
           8-1654721687-432734479-3232135806-4053264122-3456934681)
```

Notice how this command returns specific account permissions in an easy-to-read format and in SDDL. Also, the SID that represents each security principal (User and/or Group) is present in the SDDL. This is something we do not get when running `sc` from the command prompt.

Knowing how to interact with services and their associated permissions from the command line makes it easier to script out these tasks. While it is good to know how to perform these tasks from the GUI, it does not scale well as we start getting into larger network environments and Domains.

Windows Sessions

Interactive

An interactive, or local logon session, is initiated by a user authenticating to a local or domain system by entering their credentials. An interactive logon can be initiated by logging directly into the system, by requesting a secondary logon session using the `runas` command via the command line, or through a Remote Desktop connection.

Non-interactive

Non-interactive accounts in Windows differ from standard user accounts as they do not require login credentials. There are 3 types of non-interactive accounts: the Local System Account, Local Service Account, and the Network Service Account. Non-interactive accounts are generally used by the Windows operating system to automatically start services and applications without requiring user interaction. These accounts have no password associated with them and are usually used to start services when the system boots or to run scheduled tasks.

There are differences between the three types of accounts:

Account	Description
Local System Account	Also known as the `NT AUTHORITY\SYSTEM` account, this is the most powerful account in Windows systems. It is used for a variety of OS-related tasks, such as starting Windows services. This account is more powerful than accounts in the local administrators group.
Local Service Account	Known as the `NT AUTHORITY\LocalService` account, this is a less privileged version of the SYSTEM account and has similar privileges to a local user account. It is granted limited functionality and can start some services.
Network Service Account	This is known as the `NT AUTHORITY\NetworkService` account and is similar to a standard domain user account. It has similar privileges to the Local Service Account on the local machine. It can establish authenticated sessions for certain network services.

Interacting with the Windows Operating System

Graphical User Interface

The concept of a graphical user interface (GUI) was introduced in the late 1970s by the Xerox Palo Alto research laboratory. It was added to Apple and Microsoft operating systems to address usability concerns for everyday users that would likely have difficulty navigating the command line. Most casual Windows computer users do not ever need to interact with the operating system via the command line. As the name alludes to, a GUI provides users with an interactive point and click interface for interacting with the operating system and installed applications and services.

The introduction of the GUI opened up widespread appeal and access to computers across many demographics since users would be able to interact with their computer without having to memorize commands or know any programming language. Systems administrators commonly use GUI-based systems for administering Active Directory, configuring IIS, or interacting with databases.

Remote Desktop Protocol (RDP)

RDP is a proprietary Microsoft protocol which allows a user to connect to a remote system over a network connection and obtain a graphical user interface. The user connects using RDP client software to a target system running RDP server software. RDP uses port 3389 to open a dedicated network channel for sending data back and forth. When connecting via RDP, a user can access the GUI as if they were actually sitting at the computer and logging into it locally. RDP is often used by system administrators to administer remote systems quickly. It can also allow users to access their work computers when traveling or working from home after connecting to a Virtual Private Network (VPN).

Windows Command Line

Command-line interfaces give users greater control over their systems and can be used to perform a wide variety of day-to-day, administrative, and troubleshooting tasks. It can be leveraged to introduce automation to perform certain tasks quickly (such as adding many users to a domain at once). In Windows operating systems, the main two ways to interact with the system from the command line are via the Command Prompt (CMD) and PowerShell.

The Windows Command Reference from Microsoft is a comprehensive A-Z command reference which includes an overview, usage examples, and command syntax for most Windows commands, and familiarity with it is recommended.

CMD

The Command Prompt (cmd.exe) is used to enter and execute commands. A user can enter one-off commands such as `ipconfig` to view IP address information or perform more advanced tasks such as setting up scheduled tasks or creating scripts and batch files. The Command prompt can be opened from the Start Menu, by typing `cmd` in the run dialogue box, or by directly launching the binary from `C:\Windows\system32\cmd.exe`.

After launching `cmd.exe` we can type `help` to see a listing of available commands.

```
C:\htb> help
For more information on a specific command, type HELP command-name
ASSOC          Displays or modifies file extension associations.
```

```
ATTRIB         Displays or changes file attributes.
BREAK          Sets or clears extended CTRL+C checking.
BCDEDIT        Sets properties in boot database to control boot loading.
CACLS          Displays or modifies access control lists (ACLs) of files.
CALL           Calls one batch program from another.
CD             Displays the name of or changes the current directory.
CHCP           Displays or sets the active code page number.
CHDIR          Displays the name of or changes the current directory.
CHKDSK         Checks a disk and displays a status report.
CHKNTFS        Displays or modifies the checking of disk at boot time.
CLS            Clears the screen.
CMD            Starts a new instance of the Windows command interpreter.
COLOR          Sets the default console foreground and background colors.
COMP           Compares the contents of two files or sets of files.
COMPACT        Displays or alters the compression of files on NTFS partitions.
CONVERT        Converts FAT volumes to NTFS.  You cannot convert the
               current drive.
COPY           Copies one or more files to another location.

<SNIP>
```

For more information about a specific command, we can type `help <command name>`.

```
C:\htb> help schtasks

SCHTASKS /parameter [arguments]

Description:
    Enables an administrator to create, delete, query, change, run and
    end scheduled tasks on a local or remote system.

Parameter List:
    /Create         Creates a new scheduled task.

    /Delete         Deletes the scheduled task(s).

    /Query          Displays all scheduled tasks.

    /Change         Changes the properties of scheduled task.

    /Run            Runs the scheduled task on demand.

    /End            Stops the currently running scheduled task.

    /ShowSid        Shows the security identifier corresponding to a scheduled task name.

    /?              Displays this help message.

Examples:
    SCHTASKS
    SCHTASKS /?
    SCHTASKS /Run /?
    SCHTASKS /End /?
    SCHTASKS /Create /?
    SCHTASKS /Delete /?
    SCHTASKS /Query  /?
    SCHTASKS /Change /?
    SCHTASKS /ShowSid /?
```

Note that certain commands have their own help menus, which can be accessed by typing `<command> /?`. For example, information about the `ipconfig` command can be seen below.

```
C:\htb> ipconfig /?

USAGE:
    ipconfig [/allcompartments] [/? | /all |
                                 /renew [adapter] | /release [adapter] |
                                 /renew6 [adapter] | /release6 [adapter] |
                                 /flushdns | /displaydns | /registerdns |
                                 /showclassid adapter |
                                 /setclassid adapter [classid] |
                                 /showclassid6 adapter |
                                 /setclassid6 adapter [classid] ]

where
    adapter             Connection name
                       (wildcard characters * and ? allowed, see examples)

    Options:
       /?               Display this help message
       /all             Display full configuration information.
       /release         Release the IPv4 address for the specified adapter.
       /release6        Release the IPv6 address for the specified adapter.
       /renew           Renew the IPv4 address for the specified adapter.
       /renew6          Renew the IPv6 address for the specified adapter.
       /flushdns        Purges the DNS Resolver cache.
       /registerdns     Refreshes all DHCP leases and re-registers DNS names
       /displaydns      Display the contents of the DNS Resolver Cache.
       /showclassid     Displays all the dhcp class IDs allowed for adapter.
       /setclassid      Modifies the dhcp class id.
       /showclassid6    Displays all the IPv6 DHCP class IDs allowed for adapter.
       /setclassid6     Modifies the IPv6 DHCP class id.
```

```
<SNIP
```

PowerShell

Windows PowerShell is a command shell that was designed by Microsoft to be more geared towards system administrators. PowerShell, like the Windows command line, has an interactive command prompt as well as a powerful scripting environment. PowerShell is built on top of the .NET Framework, which is used for building and running applications on Windows. This makes it a very powerful tool for interfacing directly with the operating system.

Like the command prompt, PowerShell gives us direct access to the file system, and we run the majority of the same commands that we can within a cmd shell.

Cmdlets

PowerShell utilizes cmdlets, which are small single-function tools built into the shell. There are more than 100 core cmdlets, and many additional ones have been written, or we can author our own to perform more complex tasks. PowerShell also supports both simple and complex scripts used for system administration tasks, automation, and more.

Cmdlets are in the form of `Verb-Noun`. For example, the command `Get-ChildItem` can be used to list our current directory. Cmdlets also take arguments or flags. We can type `Get-ChildItem -` and hit the tab key to iterate through the arguments. A command such as `Get-ChildItem -Recurse` will show us the contents of our current working directory and all subdirectories. Another example would be `Get-ChildItem -Path C:\Users\Administrator\Documents` to get the contents of another directory. Finally, we can combine arguments such as this to list all subdirectories' contents within another directory recursively: `Get-ChildItem -Path C:\Users\Administrator\Downloads -Recurse`.

Aliases

Many cmdlets in PowerShell also have aliases. For example, the aliases for the cmdlet `Set-Location`, to change directories, is either `cd` or `sl`. Meanwhile, the aliases for `Get-ChildItem` are `ls` and `gci`. We can view all available aliases by typing `Get-Alias`.

```
PS C:\htb> get-alias

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           % -> ForEach-Object
Alias           ? -> Where-Object
Alias           ac -> Add-Content
Alias           asnp -> Add-PSSnapin
Alias           cat -> Get-Content
Alias           cd -> Set-Location
Alias           CFS -> ConvertFrom-String                          3.1.0.0    Microsoft.PowerShell.Utility
Alias           chdir -> Set-Location
Alias           clc -> Clear-Content
Alias           clear -> Clear-Host
Alias           clhy -> Clear-History
Alias           cli -> Clear-Item
Alias           clp -> Clear-ItemProperty
```

We can also set up our own aliases with `New-Alias` and get the alias for any cmdlet with `Get-Alias -Name`.

```
PS C:\htb> New-Alias -Name "Show-Files" Get-ChildItem
PS C:\> Get-Alias -Name "Show-Files"

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Alias           Show-Files
```

PowerShell has a help system for cmdlets, functions, scripts, and concepts. This is not installed by default, but we can either run the `Get-Help <cmdlet-name> -Online` command to open the online help for a cmdlet or function in our web browser. We can type `Update-Help` to download and install help files locally.

```
PS C:\htb> help

TOPIC
    Windows PowerShell Help System

SHORT DESCRIPTION
    Displays help about Windows PowerShell cmdlets and concepts.

LONG DESCRIPTION
    Windows PowerShell Help describes Windows PowerShell cmdlets,
    functions, scripts, and modules, and explains concepts, including
    the elements of the Windows PowerShell language.

    Windows PowerShell does not include help files, but you can read the
    help topics online, or use the Update-Help cmdlet to download help files
    to your computer and then use the Get-Help cmdlet to display the help
    topics at the command line.

    You can also use the Update-Help cmdlet to download updated help files
    as they are released so that your local help content is never obsolete.

    Without help files, Get-Help displays auto-generated help for cmdlets,
    functions, and scripts.

  ONLINE HELP
    You can find help for Windows PowerShell online in the TechNet Library
    beginning at http://go.microsoft.com/fwlink/?LinkID=108518.

    To open online help for any cmdlet or function, type:

        Get-Help <cmdlet-name> -Online

<SNIP>
```

Typing a command such as `Get-Help Get-AppPackage` will return just the partial help unless the Help files are installed.

```
PS C:\htb>  Get-Help Get-AppPackage

NAME
    Get-AppxPackage

SYNTAX
    Get-AppxPackage [[-Name] <string>] [[-Publisher] <string>] [-AllUsers] [-PackageTypeFilter {None | Main |
    Framework | Resource | Bundle | Xap | Optional | All}] [-User <string>] [-Volume <AppxVolume>]
    [<CommonParameters>]

ALIASES
    Get-AppPackage

REMARKS
    Get-Help cannot find the Help files for this cmdlet on this computer. It is displaying only partial help.
        -- To download and install Help files for the module that includes this cmdlet, use Update-Help.
```

Running Scripts

The PowerShell ISE (Integrated Scripting Environment) allows users to write PowerShell scripts on the fly. It also has an autocomplete/lookup function for PowerShell commands. The PowerShell ISE allows us to write and run scripts in the same console, which allows for quick debugging.

We can run PowerShell scripts in a variety of ways. If we know the functions, we can run the script either locally or after loading into memory with a download cradle like the below example.

```
PS C:\htb> .\PowerView.ps1;Get-LocalGroup |fl

Description     : Users of Docker Desktop
Name            : docker-users
SID             : S-1-5-21-674899381-4069889467-2080702030-1004
PrincipalSource : Local
ObjectClass     : Group

Description     : VMware User Group
Name            : __vmware__
SID             : S-1-5-21-674899381-4069889467-2080702030-1003
PrincipalSource : Local
ObjectClass     : Group

Description     : Members of this group can remotely query authorization attributes and permissions for resources on
                  this computer.
Name            : Access Control Assistance Operators
SID             : S-1-5-32-579
PrincipalSource : Local
ObjectClass     : Group

Description     : Administrators have complete and unrestricted access to the computer/domain
Name            : Administrators
SID             : S-1-5-32-544
PrincipalSource : Local

<SNIP>
```

One common way to work with a script in PowerShell is to import it so that all functions are then available within our current PowerShell console session: `Import-Module .\PowerView.ps1`. We can then either start a command and cycle through the options or type `Get-Module` to list all loaded modules and their associated commands.

```
PS C:\htb> Get-Module | select Name,ExportedCommands | fl

Name             : Appx
ExportedCommands : {[Add-AppxPackage, Add-AppxPackage], [Add-AppxVolume, Add-AppxVolume], [Dismount-AppxVolume,
                   Dismount-AppxVolume], [Get-AppxDefaultVolume, Get-AppxDefaultVolume]...}

Name             : Microsoft.PowerShell.LocalAccounts
ExportedCommands : {[Add-LocalGroupMember, Add-LocalGroupMember], [Disable-LocalUser, Disable-LocalUser],
                   [Enable-LocalUser, Enable-LocalUser], [Get-LocalGroup, Get-LocalGroup]...}

Name             : Microsoft.PowerShell.Management
ExportedCommands : {[Add-Computer, Add-Computer], [Add-Content, Add-Content], [Checkpoint-Computer,
                   Checkpoint-Computer], [Clear-Content, Clear-Content]...}

Name             : Microsoft.PowerShell.Utility
ExportedCommands : {[Add-Member, Add-Member], [Add-Type, Add-Type], [Clear-Variable, Clear-Variable], [Compare-Object,
                   Compare-Object]...}

Name             : PSReadline
ExportedCommands : {[Get-PSReadLineKeyHandler, Get-PSReadLineKeyHandler], [Get-PSReadLineOption,
                   Get-PSReadLineOption], [Remove-PSReadLineKeyHandler, Remove-PSReadLineKeyHandler],
                    [Set-PSReadLineKeyHandler, Set-PSReadLineKeyHandler]...}
```

Execution Policy

Sometimes we will find that we are unable to run scripts on a system. This is due to a security feature called the `execution policy`, which attempts to prevent the execution of malicious scripts. The possible policies are:

Policy	Description
`AllSigned`	All scripts can run, but a trusted publisher must sign scripts and configuration files. This includes both remote and local scripts. We receive a prompt before running scripts signed by publishers that we have not yet listed as either trusted or untrusted.
`Bypass`	No scripts or configuration files are blocked, and the user receives no warnings or prompts.
`Default`	This sets the default execution policy, `Restricted` for Windows desktop machines and `RemoteSigned` for Windows servers.
`RemoteSigned`	Scripts can run but requires a digital signature on scripts that are downloaded from the internet. Digital signatures are not required for scripts that are written locally.
`Restricted`	This allows individual commands but does not allow scripts to be run. All script file types, including configuration files (`.ps1xml`), module script files (`.psm1`), and PowerShell profiles (`.ps1`) are blocked.
`Undefined`	No execution policy is set for the current scope. If the execution policy for ALL scopes is set to undefined, then the default execution policy of `Restricted` will be used.
`Unrestricted`	This is the default execution policy for non-Windows computers, and it cannot be changed. This policy allows for unsigned scripts to be run but warns the user before running scripts that are not from the local intranet zone.

Below is an example of the current execution policy for all scopes.

```
PS C:\htb> Get-ExecutionPolicy -List

        Scope ExecutionPolicy
        ----- ---------------
MachinePolicy       Undefined
   UserPolicy       Undefined
      Process       Undefined
  CurrentUser       Undefined
 LocalMachine    RemoteSigned
```

The execution policy is not meant to be a security control that restricts user actions. A user can easily bypass the policy by either typing the script contents directly into the PowerShell window, downloading and invoking the script, or specifying the script as an encoded command. It can also be bypassed by adjusting the execution policy (if the user has the proper rights) or setting the execution policy for the current process scope (which can be done by almost any user as it does not require a configuration change and will only be set for the duration of the user's session).

Below is an example of changing the execution policy for the current process (session).

```
PS C:\htb> Set-ExecutionPolicy Bypass -Scope Process

Execution Policy Change
The execution policy helps protect you from scripts that you do not trust. Changing the execution policy might expose
you to the security risks described in the about_Execution_Policies help topic at
https:/go.microsoft.com/fwlink/?LinkID=135170. Do you want to change the execution policy?
[Y] Yes  [A] Yes to All  [N] No  [L] No to All  [S] Suspend  [?] Help (default is "N"): Y
```

We can now see that the execution policy has been changed.

```
PS C:\htb>  Get-ExecutionPolicy -List

        Scope ExecutionPolicy
        ----- ---------------
MachinePolicy       Undefined
   UserPolicy       Undefined
      Process          Bypass
  CurrentUser       Undefined
 LocalMachine    RemoteSigned
```

Windows Management Instrumentation (WMI)

WMI is a subsystem of PowerShell that provides system administrators with powerful tools for system monitoring. The goal of WMI is to consolidate device and application management across corporate networks. WMI is a core part of the Windows operating system and has come pre-installed since Windows 2000. It is made up of the following components:

Component Name	Description
WMI service	The Windows Management Instrumentation process, which runs automatically at boot and acts as an intermediary between WMI providers, the WMI repository, and managing applications.
Managed objects	Any logical or physical components that can be managed by WMI.
WMI providers	Objects that monitor events/data related to a specific object.
Classes	These are used by the WMI providers to pass data to the WMI service.
Methods	These are attached to classes and allow actions to be performed. For example, methods can be used to start/stop processes on remote machines.
WMI repository	A database that stores all static data related to WMI.
CIM Object Manager	The system that requests data from WMI providers and returns it to the application requesting it.
WMI API	Enables applications to access the WMI infrastructure.
WMI Consumer	Sends queries to objects via the CIM Object Manager.

Some of the uses for WMI are:

- Status information for local/remote systems
- Configuring security settings on remote machines/applications
- Setting and changing user and group permissions
- Setting/modifying system properties
- Code execution
- Scheduling processes
- Setting up logging

These tasks can all be performed using a combination of PowerShell and the WMI Command-Line Interface (WMIC). WMI can be run via the Windows command prompt by typing `WMIC` to open an interactive shell or by running a command directly such as `wmic computersystem get name` to get the hostname. We can view a listing of WMIC commands and aliases by typing `WMIC /?`.

```
C:\htb> wmic /?

WMIC is deprecated.

[global switches] <command>

The following global switches are available:
/NAMESPACE           Path for the namespace the alias operate against.
/ROLE                Path for the role containing the alias definitions.
/NODE                Servers the alias will operate against.
/IMPLEVEL            Client impersonation level.
/AUTHLEVEL           Client authentication level.
/LOCALE              Language id the client should use.
/PRIVILEGES          Enable or disable all privileges.
/TRACE               Outputs debugging information to stderr.
/RECORD              Logs all input commands and output.
/INTERACTIVE         Sets or resets the interactive mode.
/FAILFAST            Sets or resets the FailFast mode.
/USER                User to be used during the session.
/PASSWORD            Password to be used for session login.
/OUTPUT              Specifies the mode for output redirection.
/APPEND              Specifies the mode for output redirection.
/AGGREGATE           Sets or resets aggregate mode.
/AUTHORITY           Specifies the <authority type> for the connection.
/?[:<BRIEF|FULL>]    Usage information.

For more information on a specific global switch, type: switch-name /?

Press any key to continue, or press the ESCAPE key to stop
```

The following command example lists information about the operating system.

```
C:\htb> wmic os list brief

BuildNumber  Organization  RegisteredUser  SerialNumber             SystemDirectory      Version
19041                      Owner           00123-00123-00123-AAOEM  C:\Windows\system32  10.0.19041
```

WMIC uses aliases and associated verbs, adverbs, and switches. The above command example uses `LIST` to show data and the adverb `BRIEF` to provide just the core set of properties. An in-depth listing of verbs, switches, and adverbs is available here. WMI can be used with PowerShell by using the `Get-WmiObject` module. This module is used to get instances of WMI classes or information about available classes. This module can be used against local or remote machines.

Here we can get information about the operating system.

```
PS C:\htb> Get-WmiObject -Class Win32_OperatingSystem | select SystemDirectory,BuildNumber,SerialNumber,Version | ft

SystemDirectory     BuildNumber SerialNumber            Version
---------------     ----------- ------------            -------
C:\Windows\system32 19041       00123-00123-00123-AAOEM 10.0.19041
```

We can also use the `Invoke-WmiMethod` module, which is used to call the methods of WMI objects. A simple example is renaming a file. We can see that the command completed properly because the `ReturnValue` is set to 0.

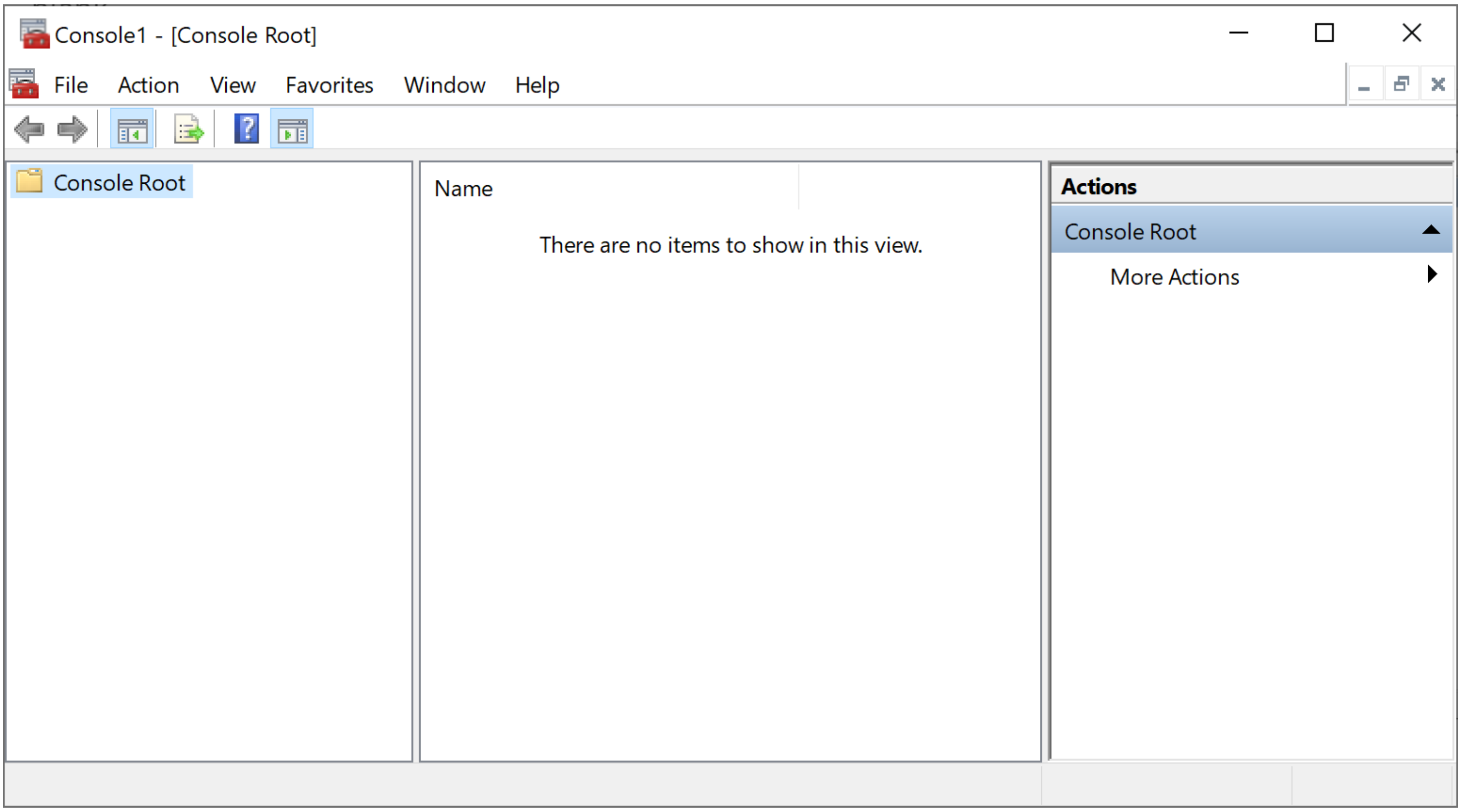
```
PS C:\htb> Invoke-WmiMethod -Path "CIM_DataFile.Name='C:\users\public\spns.csv'" -Name Rename -ArgumentList "C:\Users\Public\kerberoasted_users.csv"

__GENUS          : 2
__CLASS          : __PARAMETERS
__SUPERCLASS     :
__DYNASTY        : __PARAMETERS
__RELPATH        :
__PROPERTY_COUNT : 1
__DERIVATION     : {}
__SERVER         :
__NAMESPACE      :
__PATH           :
ReturnValue      : 0
PSComputerName   :
```

This section provides a brief overview of `WMI`, `WMIC`, and combining `WMIC` and `PowerShell`. `WMI` has a wide variety of uses for both blue team and red team operators. Later sections of this course will show some ways that `WMI` can be leveraged offensively for both enumeration and lateral movement.

Microsoft Management Console (MMC)

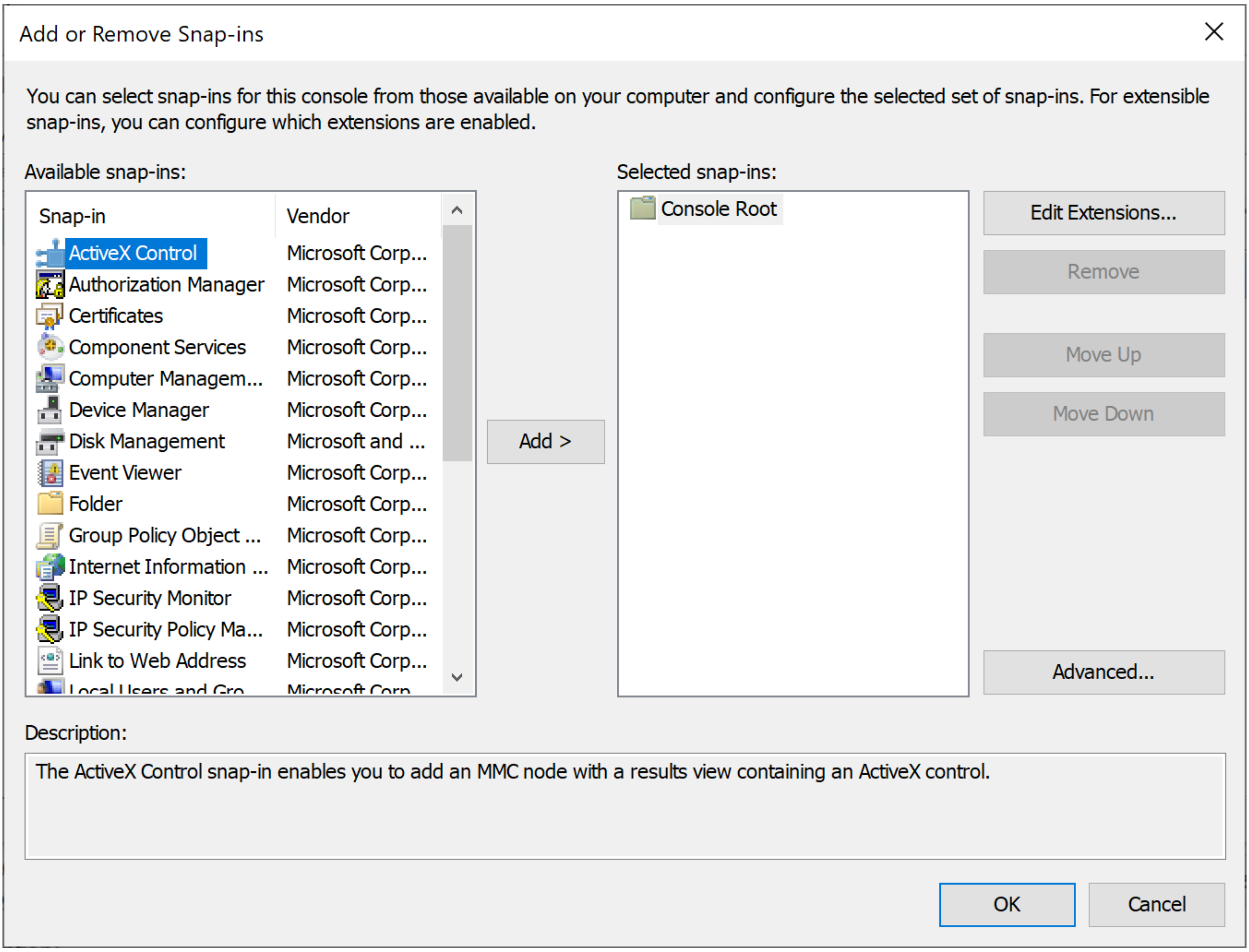
The MMC can be used to group snap-ins, or administrative tools, to manage hardware, software, and network components within a Windows host. It has been around since Windows Server 2000 and runs on all Windows versions. We can also use MMC to create custom tools and distribute them to users. MMC works with the concept of snap-ins, allowing administrators to create a customized console with only the administrative tools needed to manage several services. These snap-ins can be added to manage both local and remote systems.

We can open MMC by just typing `mmc` in the Start menu. When we open MMC for the first time, it will be blank.



There are no items to show in this view.

From here, we can browse to File --> `Add or Remove Snap-ins`, and begin customizing our administrative console.



You can select snap-ins for this console from those available on your computer and configure the selected set of snap-ins. For extensible snap-ins, you can configure which extensions are enabled.

Available snap-ins:

Snap-in	Vendor
ActiveX Control	Microsoft Corp...
Authorization Manager	Microsoft Corp...
Certificates	Microsoft Corp...
Component Services	Microsoft Corp...
Computer Managem...	Microsoft Corp...
Device Manager	Microsoft Corp...
Disk Management	Microsoft and ...
Event Viewer	Microsoft Corp...
Folder	Microsoft Corp...
Group Policy Object ...	Microsoft Corp...
Internet Information ...	Microsoft Corp...
IP Security Monitor	Microsoft Corp...
IP Security Policy Ma...	Microsoft Corp...
Link to Web Address	Microsoft Corp...
Local Users and Gro...	Microsoft Corp...

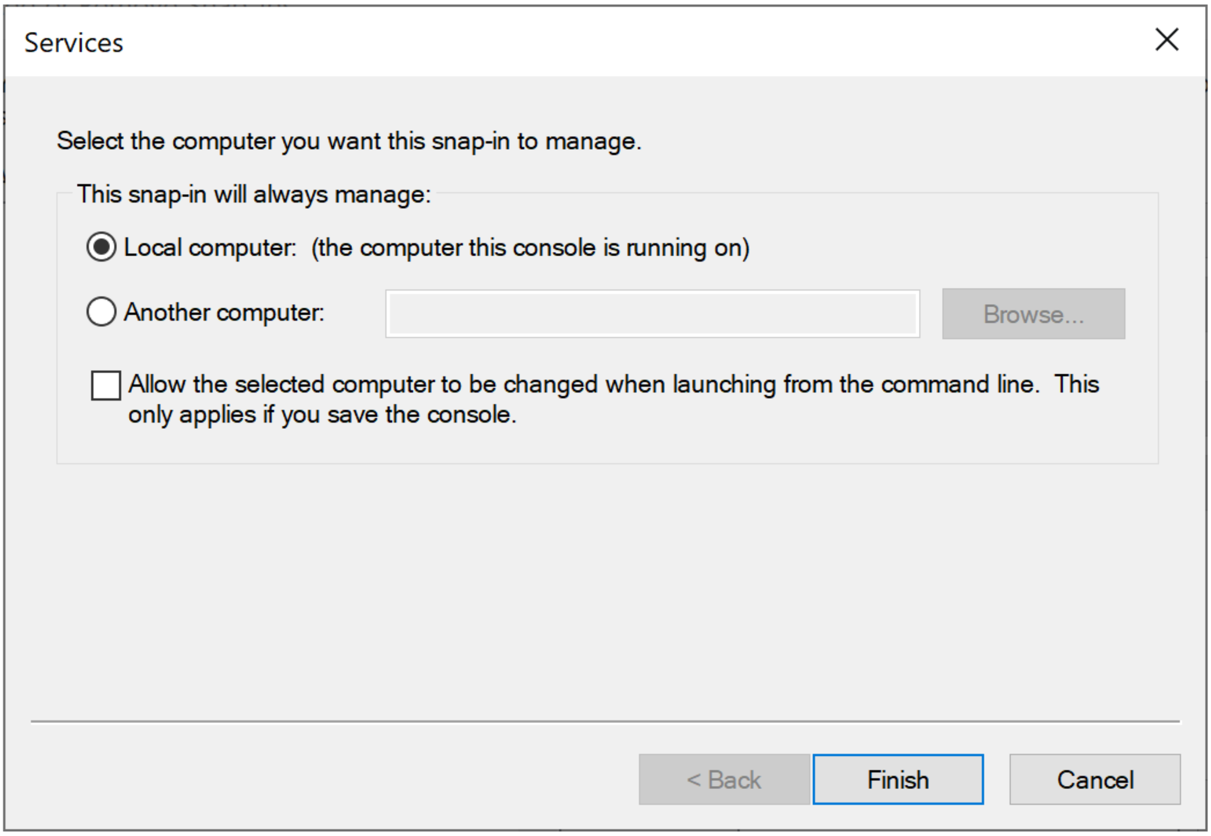
Selected snap-ins:

Console Root

Description:

The ActiveX Control snap-in enables you to add an MMC node with a results view containing an ActiveX control.

As we begin adding snap-ins, we will be asked if we want to add the snap-in to manage just the local computer or if it will be used to manage another computer on the network.



Services

Select the computer you want this snap-in to manage.

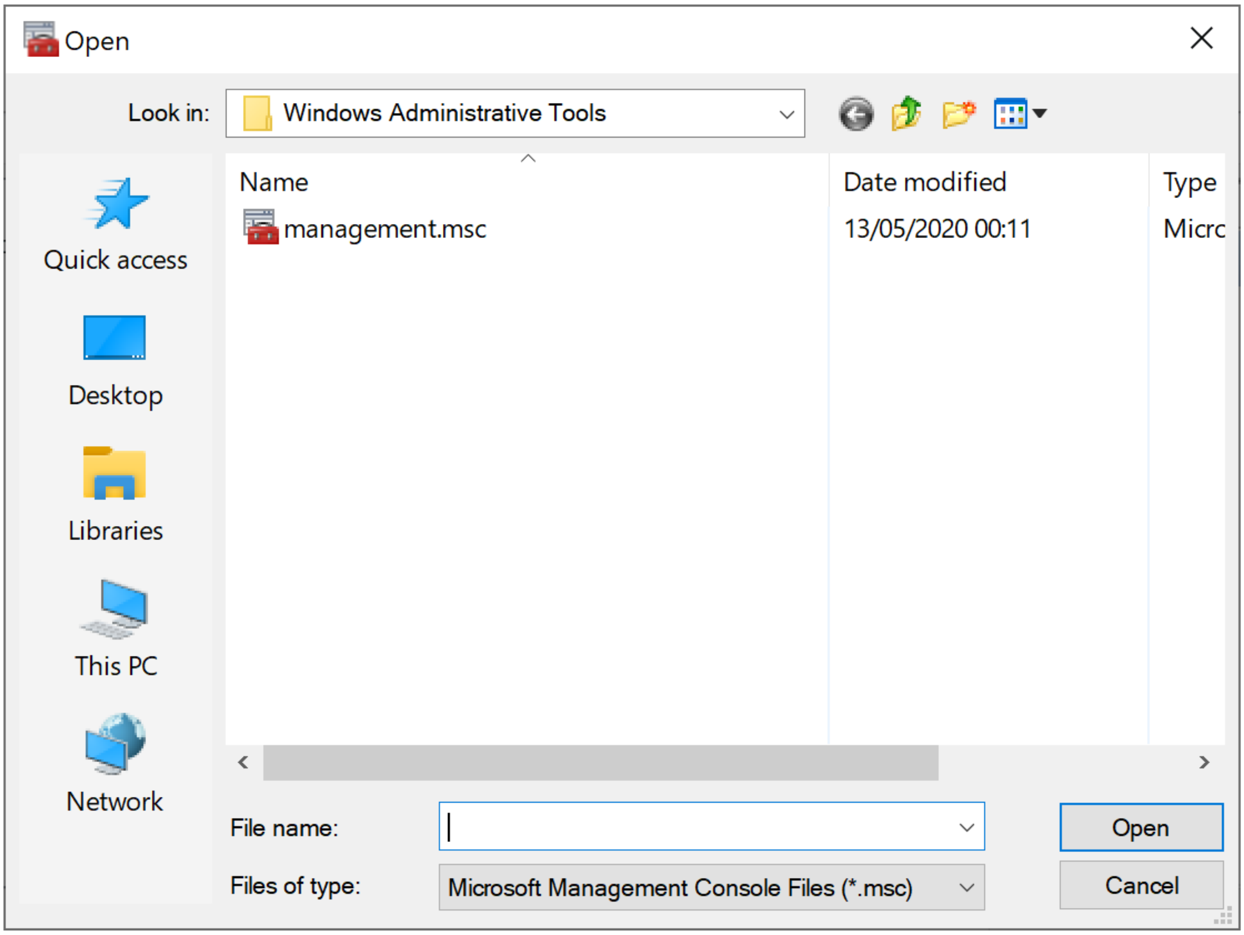
This snap-in will always manage:

● Local computer: (the computer this console is running on)

○ Another computer:

□ Allow the selected computer to be changed when launching from the command line. This only applies if you save the console.

Once we have finished adding snap-ins, they will appear on the left-hand side of MMC. From here, we can save the set of snap-ins as a .msc file, so they will all be loaded the next time we open MMC. By default, they are saved in the Windows Administrative Tools directory under the Start menu. Next time we open MMC, we can choose to load any of the views that we have created.



Open

Look in: Windows Administrative Tools

Name	Date modified	Type
management.msc	13/05/2020 00:11	Micrc

File name:

Files of type: Microsoft Management Console Files (*.msc)

Windows Subsystem for Linux (WSL)

WSL is a feature that allows Linux binaries to be run natively on Windows 10 and Windows Server 2019. It was originally intended for developers who needed to run Bash, Ruby, and native Linux command-line tools such as `sed`, `awk`, `grep`, etc., directly on their Windows workstation. The second version of WSL, released in May 2019, introduced a real Linux kernel utilizing a subset of Hyper-V features.

WSL can be installed by running the PowerShell command `Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux` as an Administrator. Once this feature is enabled, we can either download a Linux distro from the Microsoft Store and install it or manually download the Linux distro of our choice and unpack and install it from the command line.

WSL installs an application called `Bash.exe`, which can be run by merely typing `bash` into a Windows console to spawn a Bash shell. We have the full look and feel of a Linux host from this shell, including the standard Linux directory structure.

```
PS C:\htb> ls /

bin dev home lib lLib64 media opt root sbin srv tmp var
boot etc init 1lib32 Libx32 mnt proc run Snap sys usr
```

We can access the C$ volume and other volumes on the host operating system via the `mnt` directory, making the transition from the WSL host and the Windows host OS seamless. Once in this bash shell, we can interact with WSL as we would interact with any Linux-based operating system: running commands, installing updates/packages, etc.

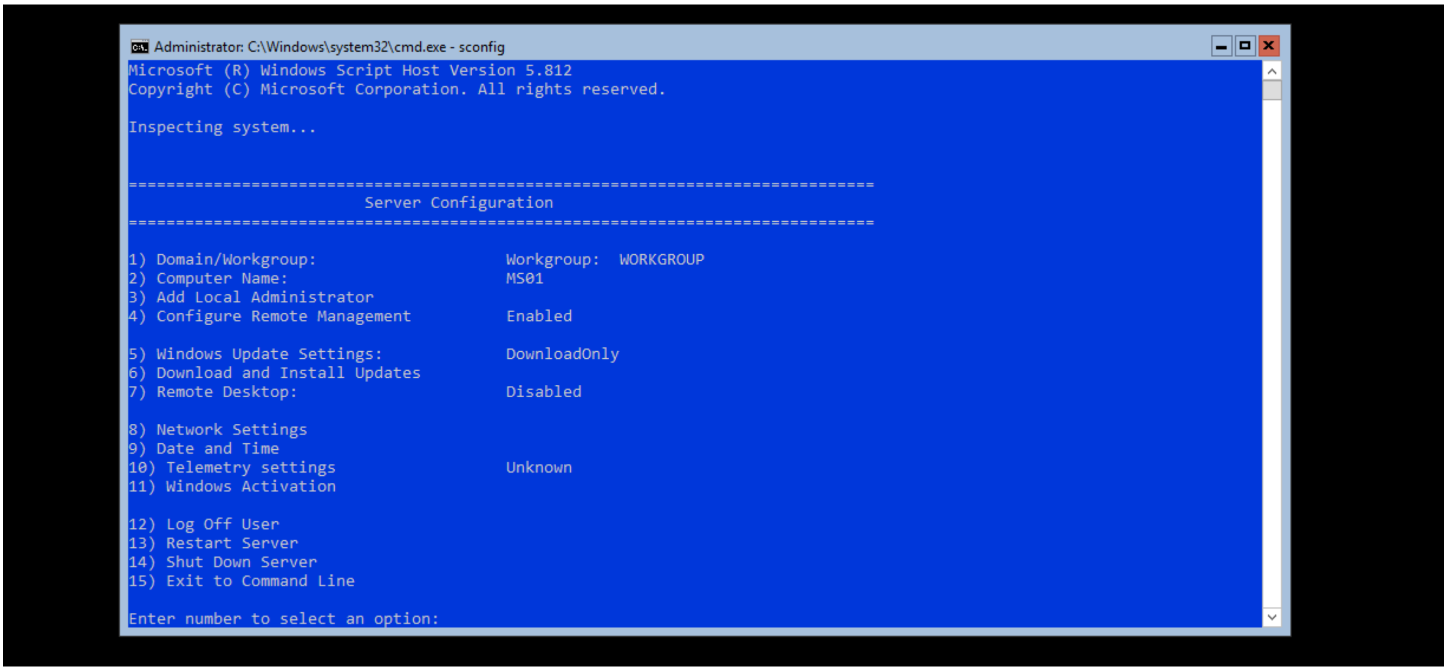
```
PS C:\htb> uname -a

Linux WS01 4.4.0-18362-Microsoft #476-Microsoft Frit Nov 01 16:53:00
PST 2019 x86_64 x86 _64 x86_64 GNU/Linux
```

Desktop Experience vs. Server Core

Windows Server Core was first released with Windows Server 2008 as a minimalistic Server environment only containing key Server functionality. As a result, Server Core has lower management requirements, a smaller attack surface, and uses less disk space and memory than its Desktop Experience (GUI) counterpart. In Server Core, all configuration and maintenance tasks are performed via the command-line, PowerShell, or remote management with MMC or Remote Server Administration Tools (RSAT).

While Server Core aims to have a smaller footprint by lacking a GUI, some graphical programs are still supported, such as Registry Editor, Notepad, System Information, Windows Installer, Task Manager, and PowerShell. It also supports some Sysinternals suite tools such as Active Directory Explorer, Process Explorer, Process Monitor, and TCPView.

As of Windows Server 2019, Server Core or Desktop Experience must be selected at installation, and neither can be rolled back (i.e., converting Server Core to Desktop Experience). Once installed, the initial setup for Server Core can be done via `Sconfig`, which is a text-based interface (actually a VBScript executed by WScript). `Sconfig` is used for performing a variety of common commands such as configuring networking, checking for/installing Windows updates, account management, configuring remote management, activating Windows, and more.



```
Microsoft (R) Windows Script Host Version 5.812
Copyright (C) Microsoft Corporation. All rights reserved.

Inspecting system...


===============================================================================
                         Server Configuration
===============================================================================

1) Domain/Workgroup:                    Workgroup:  WORKGROUP
2) Computer Name:                       MS01
3) Add Local Administrator
4) Configure Remote Management          Enabled

5) Windows Update Settings:             DownloadOnly
6) Download and Install Updates
7) Remote Desktop:                      Disabled

8) Network Settings
9) Date and Time
10) Telemetry settings                  Unknown
11) Windows Activation

12) Log Off User
13) Restart Server
14) Shut Down Server
15) Exit to Command Line

Enter number to select an option:
```

Certain server applications cannot run on Server Core, including Microsoft Server Virtual Machine Manager 2019 (SCVMM), System Center Data Protection Manager 2019, SharePoint Server 2019, Project Server 2019.

In summary, Server Core is lighter weight and less resource-intensive but has a steeper learning curve and can be more difficult to manage. It also has some limitations, such as performing management tasks using certain GUI programs.

In any environment, the determination between using Desktop Experience or Server Core for a server installation should be made by both the business need and intended use of the server and the skill level of the administrators responsible for maintaining it. The following table shows some of the applications available on Server Core vs. Desktop Experience. This is a list of common applications and not an exhaustive list.

Application	Server Core	Desktop Experience
Command prompt	Available	Available
Windows PowerShell/ Microsoft .NET	Available	Available
Regedit	Available	Available
Diskmgmt.msc	Not Available	Available
Server Manager	Not Available	Available
Mmc.exe	Not Available	Available
Eventvwr	Not Available	Available
Services.msc	Not Available	Available
Control Panel	Not Available	Available
Windows Explorer	Not Available	Available
Taskmgr	Available	Available
Internet Explorer or Edge	Not Available	Available

Application	Server Core	Desktop Experience
Remote Desktop Services	Available	Available

Windows Security

Security is a critical topic in Windows operating systems. Windows systems have many moving parts that present a vast attack surface. Due to the many built-in applications, features, and layers of settings, Windows systems can be easily misconfigured, thus opening them up to attack even if they are fully patched.

It has many built-in features that can be abused and has suffered from a wide variety of critical vulnerabilities, resulting in widely used and very effective remote and local exploits.

Microsoft has improved upon Windows security over the years. As our world's interconnectedness continues to expand and attackers become more sophisticated, Microsoft has continued to add new features that can be used by systems administrators to harden systems and actively block and detect attempts at intrusion and misuse.

Windows follows certain security principles. These are units in the system that can be authorized or authenticated for a particular action. These units include users, computers on the network, threads, or processes. The principles are designed to make it more difficult for attackers or malicious software to gain unauthorized access and exploit the system in an unintended way.

Security Identifier (SID)

Each of the security principals on the system has a unique security identifier (SID). The system automatically generates SIDs. This means that even if, for example, we have two identical users on the system, Windows can distinguish the two and their rights based on their SIDs. SIDs are string values with different lengths, which are stored in the security database. These SIDs are added to the user's access token to identify all actions that the user is authorized to take.

A SID consists of the Identifier Authority and the Relative ID (RID). In an Active Directory (AD) domain environment, the SID also includes the domain SID.

```
PS C:\htb> whoami /user

USER INFORMATION
----------------

User Name           SID
=================== =============================================
ws01\bob S-1-5-21-674899381-4069889467-2080702030-1002
```

The SID is broken down into this pattern.

```
(SID)-(revision level)-(identifier-authority)-(subauthority1)-(subauthority2)-(etc)
```

Let's break down the SID piece by piece.

Number	Meaning	Description
S	SID	Identifies the string as a SID.
1	Revision Level	To date, this has never changed and has always been `1`.
5	Identifier-authority	A 48-bit string that identifies the authority (the computer or network) that created the SID.
21	Subauthority1	This is a variable number that identifies the user's relation or group described by the SID to the authority that created it. It tells us in what order this authority created the user's account.
674899381-4069889467-2080702030	Subauthority2	Tells us which computer (or domain) created the number
1002	Subauthority3	The RID that distinguishes one account from another. Tells us whether this user is a normal user, a guest, an administrator, or part of some other group

Security Accounts Manager (SAM) and Access Control Entries (ACE)

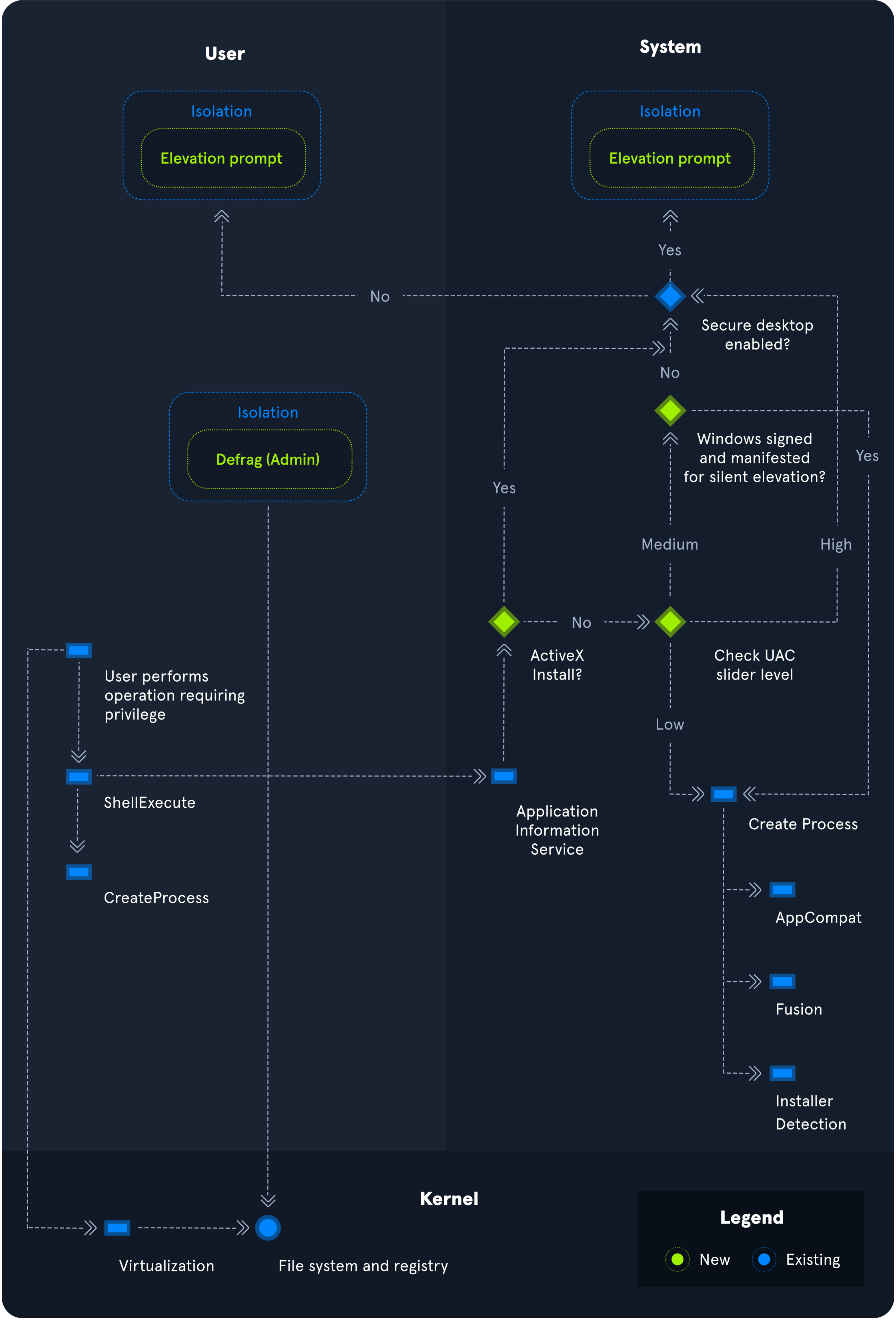
SAM grants rights to a network to execute specific processes.

The access rights themselves are managed by Access Control Entries (ACE) in Access Control Lists (ACL). The ACLs contain ACEs that define which users, groups, or processes have access to a file or to execute a process, for example.

The permissions to access a securable object are given by the security descriptor, classified into two types of ACLs: the `Discretionary Access Control List (DACL)` or `System Access Control List (SACL)`. Every thread and process started or initiated by a user goes through an authorization process. An integral part of this process is access tokens, validated by the Local Security Authority (LSA). In addition to the SID, these access tokens contain other security-relevant information. Understanding these functionalities is an essential part of learning how to use and work around these security mechanisms during the privilege escalation phase.

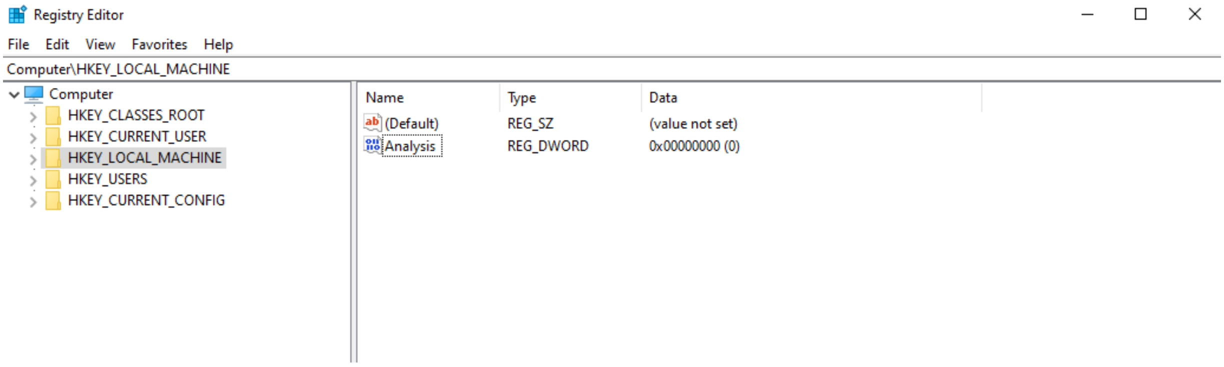
User Account Control (UAC)

User Account Control (UAC) is a security feature in Windows to prevent malware from running or manipulating processes that could damage the computer or its contents. There is the Admin Approval Mode in UAC, which is designed to prevent unwanted software from being installed without the administrator's knowledge or to prevent system-wide changes from being made. Surely you have already seen the consent prompt if you have installed a specific software, and your system has asked for confirmation if you want to have it installed. Since the installation requires administrator rights, a window pops up, asking you if you want to confirm the installation. With a standard user who has no rights for the installation, execution will be denied, or you will be asked for the administrator password. This consent prompt interrupts the execution of scripts or binaries that malware or attackers try to execute until the user enters the password or confirms execution. To understand how UAC works, we need to know how it is structured and how it works, and what triggers the consent prompt. The following diagram, adapted from the source here, illustrates how UAC works.


User
System
Isolation
Elevation prompt
Isolation
Elevation prompt
Yes
No
Secure desktop enabled?
No
Isolation
Defrag (Admin)
Windows signed and manifested for silent elevation?
Yes
Yes
Medium
High
No
ActiveX Install?
Check UAC slider level
Low
User performs operation requiring privilege
ShellExecute
Application Information Service
Create Process
CreateProcess
AppCompat
Fusion
Installer Detection
Kernel
Virtualization
File system and registry
Legend
New
Existing

Registry

The Registry is a hierarchical database in Windows critical for the operating system. It stores low-level settings for the Windows operating system and applications that choose to use it. It is divided into computer-specific and user-specific data. We can open the Registry Editor by typing `regedit` from the command line or Windows search bar.



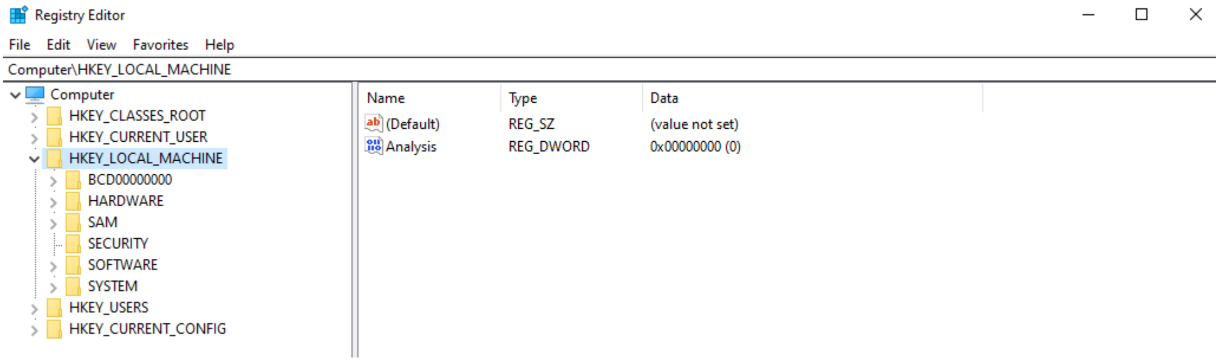
Name	Type	Data
(Default)	REG_SZ	(value not set)
Analysis	REG_DWORD	0x00000000 (0)

The tree-structure consists of main folders (root keys) in which subfolders (subkeys) with their entries/files (values) are located. There are 11 different types of values that can be entered in a subkey.

Value	Type
REG_BINARY	Binary data in any form.
REG_DWORD	A 32-bit number.
REG_DWORD_LITTLE_ENDIAN	A 32-bit number in little-endian format. Windows is designed to run on little-endian computer architectures. Therefore, this value is defined as REG_DWORD in the Windows header files.
REG_DWORD_BIG_ENDIAN	A 32-bit number in big-endian format. Some UNIX systems support big-endian architectures.
REG_EXPAND_SZ	A null-terminated string that contains unexpanded references to environment variables (for example, "%PATH%"). It will be a Unicode or ANSI string depending on whether you use the Unicode or ANSI functions. To expand the environment variable references, use the **ExpandEnvironmentStrings** function.
REG_LINK	A null-terminated Unicode string containing the target path of a symbolic link created by calling the **RegCreateKeyEx** function with REG_OPTION_CREATE_LINK.
REG_MULTI_SZ	A sequence of null-terminated strings, terminated by an empty string (\0). The following is an example: *String1*\0 *String2*\0 *String3*\0 *LastString*\0\0 The first \0 terminates the first string, the second to the last \0 terminates the last string, and the final \0 terminates the sequence. Note that the final terminator must be factored into the length of the string.
REG_NONE	No defined value type.
REG_QWORD	A 64-bit number.
REG_QWORD_LITTLE_ENDIAN	A 64-bit number in little-endian format. Windows is designed to run on little-endian computer architectures. Therefore, this value is defined as REG_QWORD in the Windows header files.
REG_SZ	A null-terminated string. This will be either a Unicode or an ANSI string, depending on whether you use the Unicode or ANSI functions.

Source: https://docs.microsoft.com/en-us/windows/win32/sysinfo/registry-value-types

Each folder under `Computer` is a key. The root keys all start with `HKEY`. A key such as `HKEY-LOCAL-MACHINE` is abbreviated to `HKLM`. HKLM contains all settings that are relevant to the local system. This root key contains six subkeys like `SAM`, `SECURITY`, `SYSTEM`, `SOFTWARE`, `HARDWARE`, and `BCD`, loaded into memory at boot time (except `HARDWARE` which is dynamically loaded).



Name	Type	Data
(Default)	REG_SZ	(value not set)
Analysis	REG_DWORD	0x00000000 (0)

The entire system registry is stored in several files on the operating system. You can find these under `C:\Windows\System32\Config\`.

```
PS C:\htb> ls

    Directory: C:\Windows\system32\config

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         12/7/2019   4:14 AM                Journal
d-----         12/7/2019   4:14 AM                RegBack
d-----         12/7/2019   4:14 AM                systemprofile
d-----         8/12/2020   1:43 AM                TxR
-a----         8/13/2020   6:02 PM        1048576 BBI
-a----         6/25/2020   4:36 PM          28672 BCD-Template
-a----         8/30/2020  12:17 PM       33816576 COMPONENTS
-a----         8/13/2020   6:02 PM         524288 DEFAULT
-a----         8/26/2020   7:51 PM        4603904 DRIVERS
-a----         6/25/2020   3:37 PM          32768 ELAM
-a----         8/13/2020   6:02 PM          65536 SAM
-a----         8/13/2020   6:02 PM          65536 SECURITY
-a----         8/13/2020   6:02 PM       87818240 SOFTWARE
```

```
-a----         8/13/2020   6:02 PM       17039360 SYSTEM
```

The user-specific registry hive (HKCU) is stored in the user folder (i.e., `C:\Users\<USERNAME>\Ntuser.dat`).

```
PS C:\htb> gci -Hidden

    Directory: C:\Users\bob

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d--h--         6/25/2020   5:12 PM                AppData
d--hsl         6/25/2020   5:12 PM                Application Data
d--hsl         6/25/2020   5:12 PM                Cookies
d--hsl         6/25/2020   5:12 PM                Local Settings
d--h--         6/25/2020   5:12 PM                MicrosoftEdgeBackups
d--hsl         6/25/2020   5:12 PM                My Documents
d--hsl         6/25/2020   5:12 PM                NetHood
d--hsl         6/25/2020   5:12 PM                PrintHood
d--hsl         6/25/2020   5:12 PM                Recent
d--hsl         6/25/2020   5:12 PM                SendTo
d--hsl         6/25/2020   5:12 PM                Start Menu
d--hsl         6/25/2020   5:12 PM                Templates
-a-h--         8/13/2020   6:01 PM        2883584 NTUSER.DAT
-a-hs-         6/25/2020   5:12 PM         524288 ntuser.dat.LOG1
-a-hs-         6/25/2020   5:12 PM        1011712 ntuser.dat.LOG2
-a-hs-         8/17/2020   5:46 PM        1048576 NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.0.regtrans-ms
-a-hs-         8/17/2020  12:13 PM        1048576 NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.1.regtrans-ms
-a-hs-         8/17/2020  12:13 PM        1048576 NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.2.regtrans-ms
-a-hs-         8/17/2020   5:46 PM          65536 NTUSER.DAT{53b39e87-18c4-11ea-a811-000d3aa4692b}.TxR.blf
-a-hs-         6/25/2020   5:15 PM          65536 NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TM.blf
-a-hs-         6/25/2020   5:12 PM         524288 NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000
                                                  00000000001.regtrans-ms
-a-hs-         6/25/2020   5:12 PM         524288 NTUSER.DAT{53b39e88-18c4-11ea-a811-000d3aa4692b}.TMContainer000000000
                                                  00000000002.regtrans-ms
---hs-         6/25/2020   5:12 PM             20 ntuser.ini
```

Run and RunOnce Registry Keys

There are also so-called registry hives, which contain a logical group of keys, subkeys, and values to support software and files loaded into memory when the operating system is started or a user logs in. These hives are useful for maintaining access to the system. These are called Run and RunOnce registry keys.

The Windows registry includes the following four keys:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce
```

Here is an example of the `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` key while logged in to a system.

```
PS C:\htb> reg query HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run
    SecurityHealth    REG_EXPAND_SZ    %windir%\system32\SecurityHealthSystray.exe
    RTHDVCPL    REG_SZ    "C:\Program Files\Realtek\Audio\HDA\RtkNGUI64.exe" -s
    Greenshot    REG_SZ    C:\Program Files\Greenshot\Greenshot.exe
```

Here is an example of the `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run` showing applications running under the current user while logged in to a system.

```
PS C:\htb> reg query HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
    OneDrive    REG_SZ    "C:\Users\bob\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
    OPENVPN-GUI    REG_SZ    C:\Program Files\OpenVPN\bin\openvpn-gui.exe
    Docker Desktop    REG_SZ    C:\Program Files\Docker\Docker\Docker Desktop.exe
```

Application Whitelisting

An application whitelist is a list of approved software applications or executables allowed to be present and run on a system. The goal is to protect the environment from harmful malware and unapproved software that does not align with the specific business needs of an organization. Implementing an enforced whitelist can be a challenge, especially in a large network. An organization should implement a whitelist in audit mode initially to make sure that all necessary applications are whitelisted and not blocked by an error of omission, which can cause more problems than it fixes.

Blacklisting, in contrast, specifies a list of harmful or disallowed software/applications to block, and all others are allowed to run/be installed. Whitelisting is based on a "zero trust" principle in which all software/applications are deemed "bad" except for those specifically allowed. Maintaining a whitelist generally has less overhead as a system administrator will only need to specify what is allowed and not constantly update a "blacklist" with new malicious applications.

Whitelisting is recommended by organizations such as NIST, especially in high-security environments.

AppLocker

AppLocker is Microsoft's application whitelisting solution and was first introduced in Windows 7. AppLocker gives system administrators control over which applications and files users can run. It gives granular control over executables, scripts, Windows installer files, DLLs, packaged apps, and packed app installers.

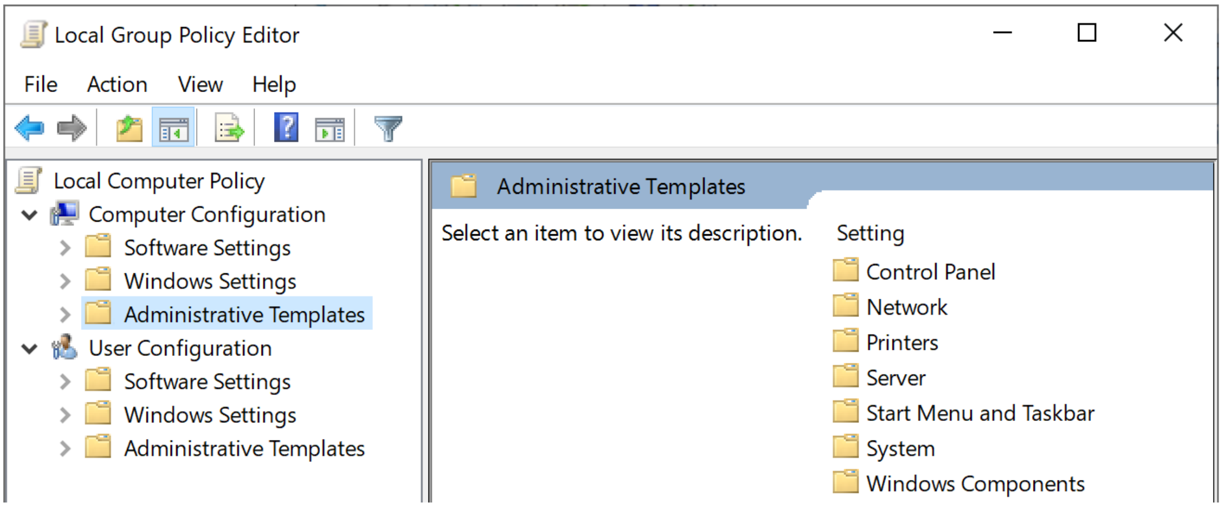
It allows for creating rules based on file attributes such as the publisher's name (which can be derived from the digital signature), product name, file name, and version. Rules can also be set up based on file paths and hashes. Rules can be applied to either security groups or individual users, based on the business need. AppLocker can be deployed in audit mode first to test the impact before enforcing all of the rules.

Local Group Policy

Group Policy allows administrators to set, configure, and adjust a variety of settings. In a domain environment, group policies are pushed down from a Domain Controller onto all domain-joined machines that Group Policy objects (GPOs) are linked to. These settings can also be defined on individual machines using Local Group Policy.

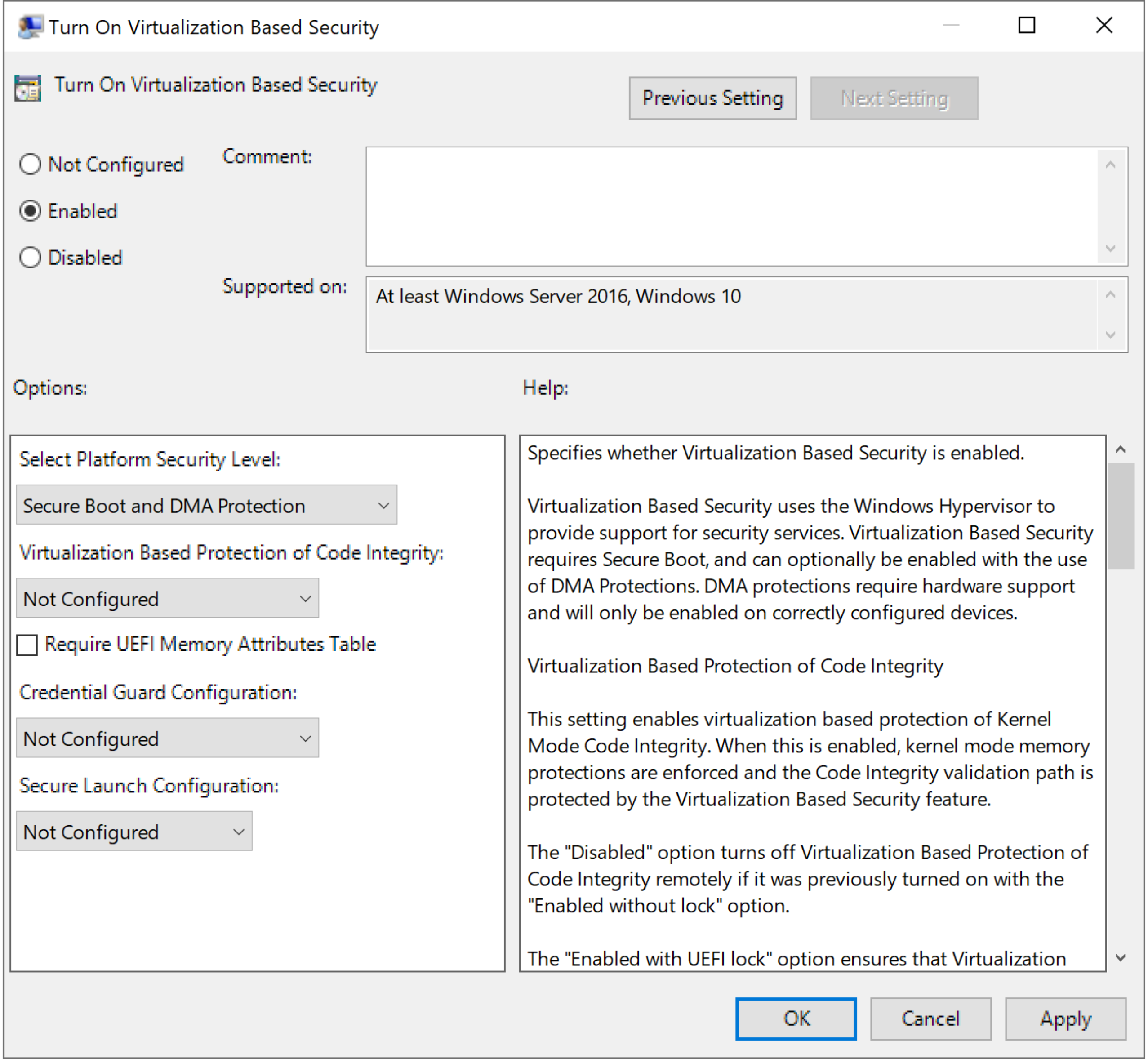
Group Policy can be configured locally, in both domain environments and non-domain environments. Local Group Policy can be used to tweak certain graphical and network settings that are otherwise not accessible via the Control Panel. It can also be used to lock down an individual computer policy with stringent security settings, such as only allowing certain programs to be installed/run or enforcing strict user account password requirements.

We can open the Local Group Policy Editor by opening the Start menu and typing `gpedit.msc`. The editor is split into two categories under Local Computer Policy - `Computer Configuration` and `User Configuration`.



Select an item to view its description.

For example, we can open the Local Computer Policy to enable Credential Guard by enabling the setting `Turn On Virtualization Based Security`. Credential Guard is a feature in Windows 10 that protects against credential theft attacks by isolating the operating system's LSA process.



○ Not Configured
● Enabled
○ Disabled

Supported on: At least Windows Server 2016, Windows 10

Select Platform Security Level: Secure Boot and DMA Protection

Virtualization Based Protection of Code Integrity: Not Configured

□ Require UEFI Memory Attributes Table

Credential Guard Configuration: Not Configured

Secure Launch Configuration: Not Configured

Specifies whether Virtualization Based Security is enabled.

Virtualization Based Security uses the Windows Hypervisor to provide support for security services. Virtualization Based Security requires Secure Boot, and can optionally be enabled with the use of DMA Protections. DMA protections require hardware support and will only be enabled on correctly configured devices.

Virtualization Based Protection of Code Integrity

This setting enables virtualization based protection of Kernel Mode Code Integrity. When this is enabled, kernel mode memory protections are enforced and the Code Integrity validation path is protected by the Virtualization Based Security feature.

The "Disabled" option turns off Virtualization Based Protection of Code Integrity remotely if it was previously turned on with the "Enabled without lock" option.

The "Enabled with UEFI lock" option ensures that Virtualization

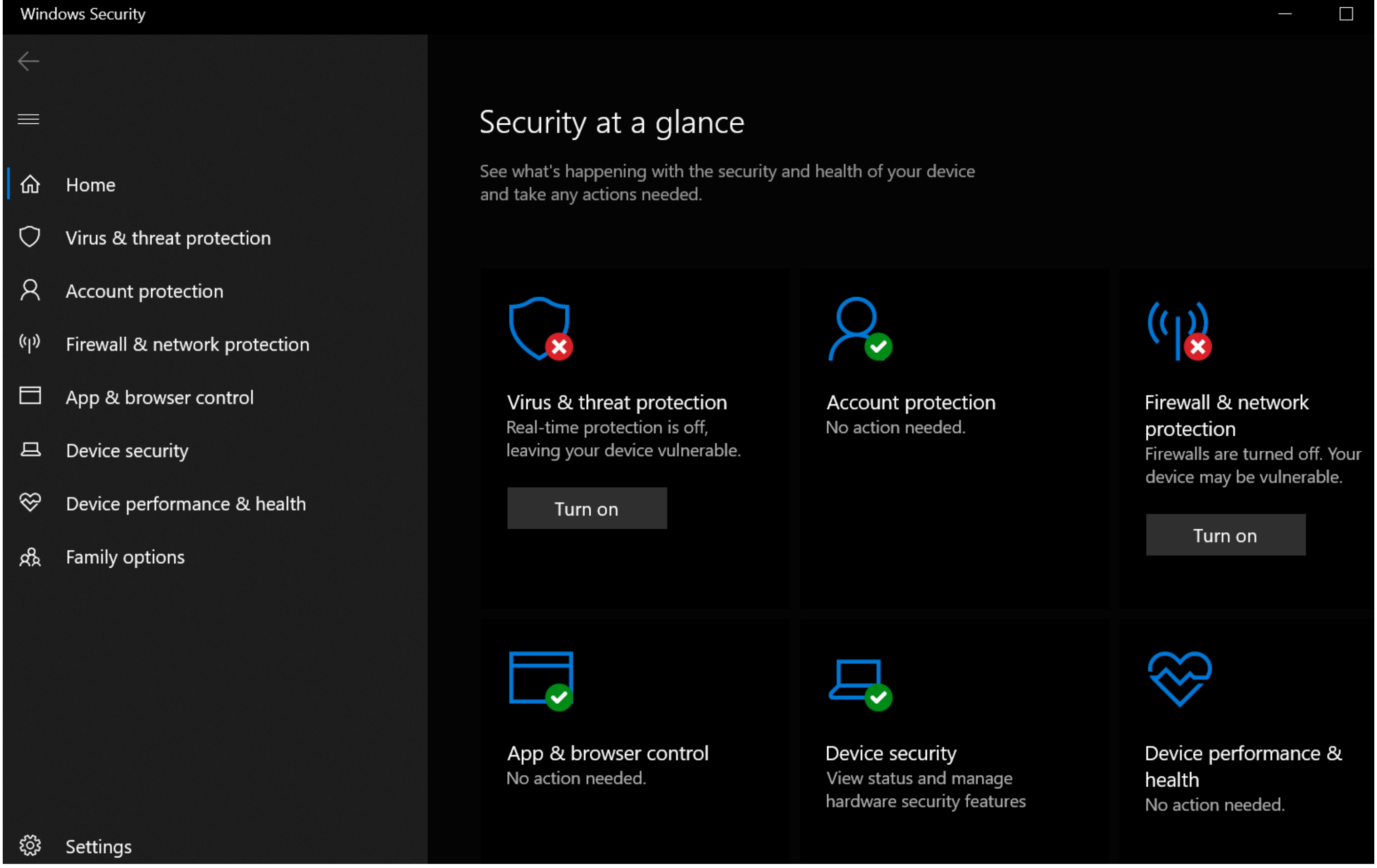
We can also enable fine-tuned account auditing and configure AppLocker from the Local Group Policy Editor. It is worth exploring Local Group Policy and learning about the wide variety of ways it can be used to lock down a Windows system.

Windows Defender Antivirus

Windows Defender Antivirus (Defender), formerly known as Windows Defender, is built-in antivirus that ships for free with Windows operating systems. It was first released as a downloadable anti-spyware tool for Windows XP and Server 2003. Defender started coming prepackaged as part of the operating system with Windows Vista/Server 2008. The program was renamed to Windows Defender Antivirus with the Windows 10 Creators Update.

Defender comes with several features such as real-time protection, which protects the device from known threats in real-time and cloud-delivered protection, which works in conjunction with automatic sample submission to upload suspicious files for analysis. When files are submitted to the cloud protection service, they are "locked" to prevent any potentially malicious behavior until the analysis is complete. Another feature is Tamper Protection, which prevents security settings from being changed through the Registry, PowerShell cmdlets, or group policy.

Windows Defender is managed from the Security Center, from which a variety of additional security features and settings can be enabled and managed.



Security at a glance

See what's happening with the security and health of your device and take any actions needed.

Virus & threat protection
Real-time protection is off, leaving your device vulnerable.

Account protection
No action needed.

Firewall & network protection
Firewalls are turned off. Your device may be vulnerable.

App & browser control
No action needed.

Device security
View status and manage hardware security features

Device performance & health
No action needed.

Real-time protection settings can be tweaked to add files, folders, and memory areas to controlled folder access to prevent unauthorized changes. We can also add files or folders to an exclusion list, so they are not scanned. An example would be excluding a folder of tools used for penetration testing from scanning as they will be flagged malicious and quarantined or removed from the system. Controlled folder access is Defender's built-in Ransomware protection.

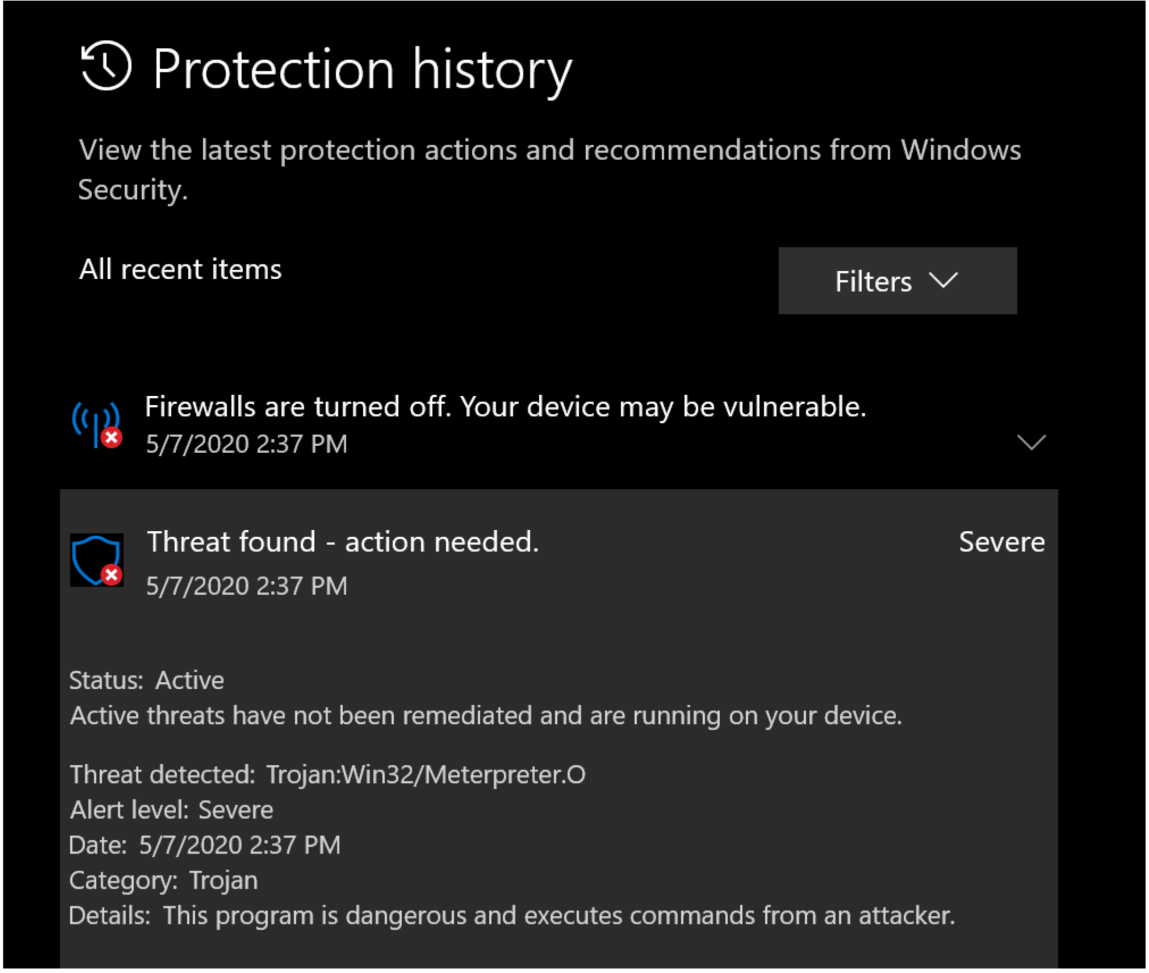
We can use the PowerShell cmdlet `Get-MpComputerStatus` to check which protection settings are enabled.

```
PS C:\htb> Get-MpComputerStatus | findstr "True"
AMServiceEnabled                 : True
AntispywareEnabled               : True
AntivirusEnabled                 : True
BehaviorMonitorEnabled           : True
IoavProtectionEnabled            : True
IsTamperProtected                : True
NISEnabled                       : True
OnAccessProtectionEnabled        : True
RealTimeProtectionEnabled        : True
```

While no antivirus solution is perfect, Windows Defender does very well in monthly detection rate tests compared to other solutions, even paid ones. Also, since it comes preinstalled as part of the operating system, it does not introduce "bloat" to the system, such as other programs that add browser extensions and trackers. Other products are known to slow down the system due to the way they hook into the operating system.

Windows Defender is not without its flaws and should be part of a defense-in-depth strategy built around core principles of configuration and patch management, not treated as a silver bullet for protecting our systems. Definitions are updated constantly, and new versions of Windows Defender are built-in to major operating releases such as Windows 10, version 1909, which is the most recent version at the time of writing.

Windows Defender will pick up payloads from common open-source frameworks such as Metasploit or unaltered versions of tools such as Mimikatz.



Protection history

View the latest protection actions and recommendations from Windows Security.

All recent items

Firewalls are turned off. Your device may be vulnerable.
5/7/2020 2:37 PM

Threat found - action needed. Severe
5/7/2020 2:37 PM

Status: Active
Active threats have not been remediated and are running on your device.

Threat detected: Trojan:Win32/Meterpreter.O
Alert level: Severe
Date: 5/7/2020 2:37 PM
Category: Trojan
Details: This program is dangerous and executes commands from an attacker.

Though it is becoming increasingly difficult, it is still possible to fully bypass Windows Defender protections enforced by the latest version with the most up-to-date definitions installed.

Skills Assessment - Windows Fundamentals

Inlanefreight recently had an incident where a disgruntled employee in marketing accessed an internally hosted HR share and deleted several confidential files & folders. Thankfully, the IT team had good backups and restored all of the deleted data. There are now concerns that this disgruntled employee was able to access the HR share in the first place. After performing a security assessment, you have found that the IT team may not fully understand how permissions work in Windows. You are conducting training and a demonstration to show the department good security practices with file sharing in a Windows environment as well as viewing services from the command line to check for any potential tampering.

Note: It is important that each step is completed in the order they are presented. Starting with step 1 and working your way through step 8, including all associated specifications with each step. Please know that each step is designed to give you the opportunity to apply the skills & concepts taught throughout this module. Take your time, have fun and feel free to reach out if you get stuck.

In this demonstration, you are:

1. Creating a shared folder called Company Data

2. Creating a subfolder called HR inside of the Company Data folder

3. Creating a user called Jim

- `Uncheck: User must change password at logon`

4. Creating a security group called HR

5. Adding Jim to the HR security group

6. Adding the HR security group to the shared Company Data folder and NTFS permissions list

- `Remove the default group that is present`
- `Share Permissions: Allow Change & Read`
- `Disable Inheritance before issuing specific NTFS permissions`
- `NTFS permissions: Modify, Read & Execute, List folder contents, Read, Write`

7. Adding the HR security group to the NTFS permissions list of the HR subfolder

- `Remove the default group that is present`
- `Disable Inheritance before issuing specific NTFS permissions`
- `NTFS permissions: Modify, Read & Execute, List folder contents, Read, and Write`

8. Using PowerShell to list details about a service