Users and Groups Management

In this task, we will learn more about users and groups, especially within the Active Directory. Gathering information about the compromised machine is essential that could be used in the next stage. Account discovery is the first step once we have gained initial access to the compromised machine to understand what we have and what other accounts are in the system.



An Active Directory environment contains various accounts with the necessary permissions, access, and roles for different purposes. Common Active Directory service accounts include built-in local user accounts, domain user accounts, managed service accounts, and virtual accounts.

* The built-in local users' accounts are used to manage the system locally, which is not part of the AD environment.
* Domain user accounts with access to an active directory environment can use the AD services (managed by AD).
* AD managed service accounts are limited domain user account with higher privileges to manage AD services.
* Domain Administrators are user accounts that can manage information in an Active Directory environment, including AD configurations, users, groups, permissions, roles, services, etc. One of the red team goals in engagement is to hunt for information that leads to a domain administrator having complete control over the AD environment.

The following are Active Directory Administrators accounts:

|  |  |
| --- | --- |
| BUILTIN\Administrator | Local admin access on a domain controller |
| Domain Admins | Administrative access to all resources in the domain |
| Enterprise Admins | Available only in the forest root |
| Schema Admins | Capable of modifying domain/forest; useful for red teamers |
| Server Operators | Can manage domain servers |
| Account Operators | Can manage users that are not in privileged groups |

Now that we learn about various account types within the AD environment. Let's enumerate the Windows machine that we have access to during the initial access stage. As a current user, we have specific permissions to view or manage things within the machine and the AD environment.

**Active Directory (AD) Enum**

Now, enumerating in the AD environment requires different tools and techniques. Once we confirm that the machine is part of the AD environment, we can start hunting for any variable info that may be used later. In this stage, we are using PowerShell to enumerate for users and groups.

The following PowerShell command is to get all active directory user accounts. Note that we need to use  -Filter argument.

PowerShell

PS C:\Users\thm> Get-ADUser -Filter \*

DistinguishedName : CN=Administrator,CN=Users,DC=thmredteam,DC=com

Enabled : True

GivenName :

Name : Administrator

ObjectClass : user

ObjectGUID : 4094d220-fb71-4de1-b5b2-ba18f6583c65

SamAccountName : Administrator

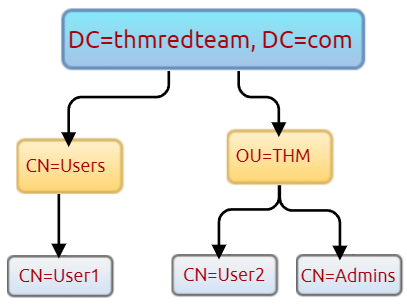
SID : S-1-5-21-1966530601-3185510712-10604624-500

Surname :

UserPrincipalName :

PS C:\Users\thm>

We can also use the [LDAP hierarchical tree structure](http://www.ietf.org/rfc/rfc2253.txt) to find a user within the AD environment. The Distinguished Name (DN) is a collection of comma-separated key and value pairs used to identify unique records within the directory. The DN consists of Domain Component (DC), OrganizationalUnitName (OU), Common Name (CN), and others. The following "CN=User1,CN=Users,DC=thmredteam,DC=com" is an example of DN, which can be visualized as follow:



Using the SearchBase option, we specify a specific Common-Name CN in the active directory. For example, we can specify to list any user(s) that part of Users.

PowerShell

PS C:\Users\thm> Get-ADUser -Filter \* -SearchBase "CN=Users,DC=THMREDTEAM,DC=COM"

DistinguishedName : CN=Administrator,CN=Users,DC=thmredteam,DC=com

Enabled : True

GivenName :

Name : Administrator

ObjectClass : user

ObjectGUID : 4094d220-fb71-4de1-b5b2-ba18f6583c65

SamAccountName : Administrator

SID : S-1-5-21-1966530601-3185510712-10604624-500

Surname :

UserPrincipalName :

Note that the result may contain more than one user depending on the configuration of the CN. Try the command to find all users within the THM OU and answer question 1 below.

***Answer the questions below***

Use the Get-ADUser -Filter \* -SearchBase command to list the available user accounts within THM OU in the thmredteam.com domain. How many users are available?



 Submit

 Hint

Once you run the previous command, what is the UserPrincipalName (email) of the admin account?



 Host Security Solution #1

Before performing further actions, we need to obtain general knowledge about the security solutions in place. Remember, it is important to enumerate antivirus and security detection methods on an endpoint in order to stay as undetected as possible and reduce the chance of getting caught.

This task will discuss the common security solution used in corporate networks, divided into Host and Network security solutions.

**Host Security Solutions**



It is a set of software applications used to monitor and detect abnormal and malicious activities within the host, including:

1. Antivirus software
2. Microsoft Windows Defender
3. Host-based Firewall
4. Security Event Logging and Monitoring
5. Host-based Intrusion Detection System (HIDS)/ Host-based Intrusion Prevention System (HIPS)
6. Endpoint Detection and Response (EDR)

Let's go more detail through the host-based security solutions that we may encounter during the red team engagement.

**Antivirus software (AV)**: Sometimes also known as anti-malware, it is mainly used to monitor, detect, and prevent, remove malicious software from being executed within the host. Most antivirus software applications use well-known features, including Background scanning, Full system scans, Virus definitions. In the background scanning, the antivirus software works in real-time and scans all open and used files in the background. The full system scan is essential when you first install the antivirus. The most interesting part is the virus definitions, where antivirus software replies to the pre-defined virus. That's why antivirus software needs to update from time to time.

There are various detection techniques that the antivirus uses, including

* Signature-based detection
* Heuristic-based detection
* Behavior-based detection

**Signature-based detection** is one of the common and traditional techniques used in antivirus software to identify malicious files. Often, researchers or users submit their infected files into an antivirus engine platform for further analysis by AV vendors, and if it confirms as malicious, then the signature gets registered in their database. The antivirus software compares the scanned file with a database of known signatures for possible attacks and malware on the client-side. If we have a match, then it considers a threat.

**Heuristic-based detection** uses machine learning to decide whether we have the malicious file or not. It scans and statically analyses in real-time in order to find suspicious properties in the application's code or check whether it uses uncommon Windows or system APIs. It does not rely on the signature-based attack in making the decisions, or sometimes it does. This depends on the implementation of the antivirus software.

Finally, **Behavior-based detection** relies on monitoring and examining the execution of applications to find abnormal behaviors and uncommon activities, such as creating/updating values in registry keys, killing/creating processes, etc.

As a red teamer, it is essential to be aware of whether antivirus exists or not. It prevents us from doing what we are attempting to do. We can enumerate AV software using Windows built-in tools, such as wmic.

PowerShell

PS C:\Users\thm> wmic /namespace:\\root\securitycenter2 path antivirusproduct

This also can be done using PowerShell, which gives the same result.

PowerShell

PS C:\Users\thm> Get-CimInstance -Namespace root/SecurityCenter2 -ClassName AntivirusProduct

displayName : Bitdefender Antivirus

instanceGuid : {BAF124F4-FA00-8560-3FDE-6C380446AEFB}

pathToSignedProductExe : C:\Program Files\Bitdefender\Bitdefender Security\wscfix.exe

pathToSignedReportingExe : C:\Program Files\Bitdefender\Bitdefender Security\bdservicehost.exe

productState : 266240

timestamp : Wed, 15 Dec 2021 12:40:10 GMT

PSComputerName :

displayName : Windows Defender

instanceGuid : {D58FFC3A-813B-4fae-9E44-DA132C9FAA36}

pathToSignedProductExe : windowsdefender://

pathToSignedReportingExe : %ProgramFiles%\Windows Defender\MsMpeng.exe

productState : 393472

timestamp : Fri, 15 Oct 2021 22:32:01 GMT

PSComputerName :

As a result, there is a third-party antivirus (Bitdefender Antivirus) and Windows Defender installed on the computer. **Note**that Windows servers may not have SecurityCenter2 namespace, which may not work on the attached VM. Instead, it works for Windows workstations!

2. **Microsoft Windows Defender**: It is a pre-installed antivirus security tool that runs on endpoints. It uses various algorithms in the detection, including machine learning, big-data analysis, in-depth threat resistance research, and Microsoft cloud infrastructure in protection against malware and viruses. MS Defender works in three protection modes: Active, Passive, Disable modes.

**Active** mode is used where the MS Defender runs as the primary antivirus software on the machine where provides protection and remediation. **Passive**mode is run when a 3rd party antivirus software is installed. Therefore, it works as secondary antivirus software where it scans files and detects threats but does not provide remediation. Finally, **Disable**mode is when the MS Defender is disabled or uninstalled from the system.

 We can use the following PowerShell command to check the service state of Windows Defender:

PowerShell

PS C:\Users\thm> Get-Service WinDefend

Status Name DisplayName

------ ---- -----------

Running WinDefend Windows Defender Antivirus Service

Next, we can start using the Get-MpComputerStatus cmdlet to get the current Windows Defender status. However, it provides the current status of security solution elements, including Anti-Spyware, Antivirus, LoavProtection, Real-time protection, etc. We can use select to specify what we need for as follows,

PowerShell

PS C:\Users\thm> Get-MpComputerStatus | select RealTimeProtectionEnabled

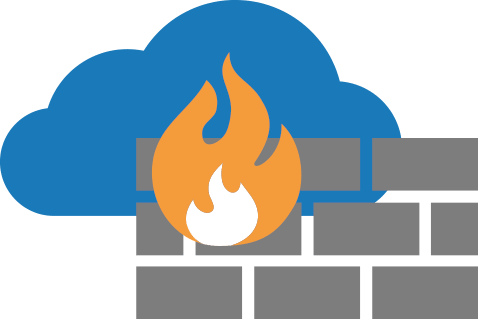
RealTimeProtectionEnabled

-------------------------

False

As a result, MpComputerStatus highlights whether Windows Defender is enabled or not.

3. **Host-based Firewall**: It is a security tool installed and run on a host machine that can prevent and block attacker or red teamers' attack attempts. Thus, it is essential to enumerate and gather details about the firewall and its rules within the machine we have initial access to.



The main purpose of the host-based firewall is to control the inbound and outbound traffic that goes through the device's interface. It protects the host from untrusted devices that are on the same network. A modern host-based firewall uses multiple levels of analyzing traffic, including packet analysis, while establishing the connection.

A firewall acts as control access at the network layer. It is capable of allowing and denying network packets. For example, a firewall can be configured to block ICMP packets sent through the ping command from other machines in the same network. Next-generation firewalls also can inspect other OSI layers, such as application layers. Therefore, it can detect and block SQL injection and other application-layer attacks.

PowerShell

PS C:\Users\thm> Get-NetFirewallProfile | Format-Table Name, Enabled

Name Enabled

---- -------

Domain True

Private True

Public True

If we have admin privileges on the current user we logged in with, then we try to disable one or more than one firewall profile using the Set-NetFirewallProfile cmdlet.

PowerShell

PS C:\Windows\system32> Set-NetFirewallProfile -Profile Domain, Public, Private -Enabled False

PS C:\Windows\system32> Get-NetFirewallProfile | Format-Table Name, Enabled

---- -------

Domain False

Private False

Public False

We can also learn and check the current Firewall rules, whether allowing or denying by the firewall.

PowerShell

PS C:\Users\thm> Get-NetFirewallRule | select DisplayName, Enabled, Description

DisplayName Enabled

----------- -------

Virtual Machine Monitoring (DCOM-In) False

Virtual Machine Monitoring (Echo Request - ICMPv4-In) False

Virtual Machine Monitoring (Echo Request - ICMPv6-In) False

Virtual Machine Monitoring (NB-Session-In) False

Virtual Machine Monitoring (RPC) False

SNMP Trap Service (UDP In) False

SNMP Trap Service (UDP In) False

Connected User Experiences and Telemetry True

Delivery Optimization (TCP-In) True

During the red team engagement, we have no clue what the firewall blocks. However, we can take advantage of some PowerShell cmdlets such as Test-NetConnection and TcpClient. Assume we know that a firewall is in place, and we need to test inbound connection without extra tools, then we can do the following:

PowerShell

PS C:\Users\thm> Test-NetConnection -ComputerName 127.0.0.1 -Port 80

ComputerName : 127.0.0.1

RemoteAddress : 127.0.0.1

RemotePort : 80

InterfaceAlias : Loopback Pseudo-Interface 1

SourceAddress : 127.0.0.1

TcpTestSucceeded : True

PS C:\Users\thm> (New-Object System.Net.Sockets.TcpClient("127.0.0.1", "80")).Connected

True

As a result, we can confirm the inbound connection on port 80 is open and allowed in the firewall. Note that we can also test for remote targets in the same network or domain names by specifying in the -ComputerName argument for the Test-NetConnection.

***Answer the questions below***

Enumerate the attached Windows machine and check whether the host-based firewall is enabled or not! (Y|N)

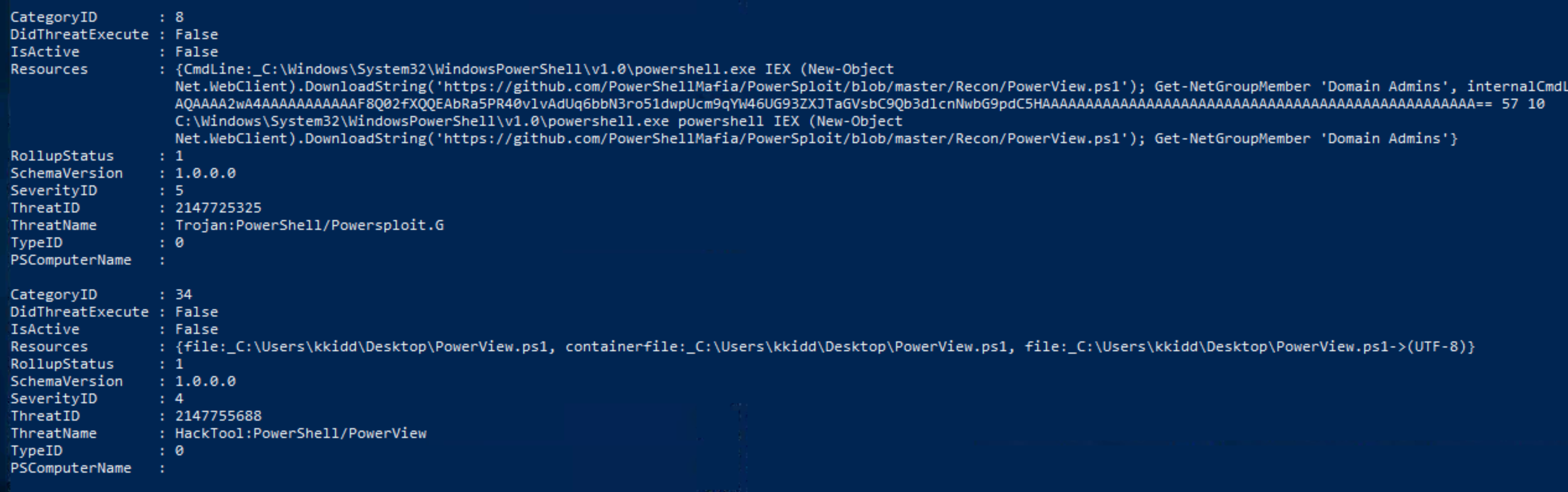


Correct Answer

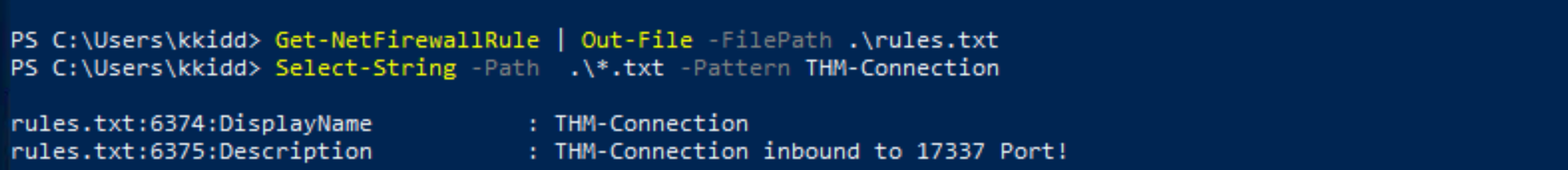
Using PowerShell cmdlets such Get-MpThreat can provide us with threats details that have been detected using MS Defender. Run it and answer the following: What is the file name that causes this alert to record?

Run Get-MpThreat and get the answer





Enumerate the firewall rules of the attached Windows machine. What is the port that is allowed under the **THM-Connection** rule?



In the next task, we will keep discussing the host security solution. I'm ready!

 Host Security Solution #2

In this task, we will keep discussing host security solutions.

4. **Security Event Logging and Monitoring**



By default, Operating systems log various activity events in the system using log files. The event logging feature is available to the IT system and network administrators to monitor and analyze important events, whether on the host or the network side. In cooperating networks, security teams utilize the logging event technique to track and investigate security incidents.

There are various categories where the Windows operating system logs event information, including the application, system, security, services, etc. In addition, security and network devices store event information into log files to allow the system administrators to get an insight into what is going on.

We can get a list of available event logs on the local machine using the Get-EventLog cmdlet.

PowerShell

PS C:\Users\thm> Get-EventLog -List

Max(K) Retain OverflowAction Entries Log

------ ------ -------------- ------- ---

512 7 OverwriteOlder 59 Active Directory Web Services

20,480 0 OverwriteAsNeeded 512 Application

512 0 OverwriteAsNeeded 170 Directory Service

102,400 0 OverwriteAsNeeded 67 DNS Server

20,480 0 OverwriteAsNeeded 4,345 System

15,360 0 OverwriteAsNeeded 1,692 Windows PowerShell

Sometimes, the list of available event logs gives you an insight into what applications and services are installed on the machine! For example, we can see that the local machine has Active Directory, DNS server, etc. For more information about the Get-EventLog cmdlet with examples, visit the [Microsoft documents website](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-eventlog?view=powershell-5.1).

In corporate networks, log agent software is installed on clients to collect and gather logs from different sensors to analyze and monitor activities within the network. We will discuss them more in the Network Security Solution task.

**System Monitor**sysmon



Windows System Monitor sysmon is a service and device driver. It is one of the Microsoft Sysinternals suites. The sysmon tool is not an essential tool (not installed by default), but it starts gathering and logging events once installed. These logs indicators can significantly help system administrators and blue teamers to track and investigate malicious activity and help with general troubleshooting.

One of the great features of the sysmon  tool is that it can log many important events, and you can also create your own rule(s) and configuration to monitor:

* Process creation and termination
* Network connections
* Modification on file
* Remote threats
* Process and memory access
* and many others

For learning more about sysmon, visit the Windows document page [here](https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon).

As a red teamer, one of the primary goals is to stay undetectable, so it is essential to be aware of these tools and avoid causing generating and alerting events. The following are some of the tricks that can be used to detect whether the sysmon is available in the victim machine or not.

We can look for a process or service that has been named "Sysmon" within the current process or services as follows,

PowerShell

PS C:\Users\thm> Get-Process | Where-Object { $\_.ProcessName -eq "Sysmon" }

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName

------- ------ ----- ----- ------ -- -- -----------

373 15 20212 31716 3316 0 Sysmon

or look for services as follows,

PowerShell

PS C:\Users\thm> Get-CimInstance win32\_service -Filter "Description = 'System Monitor service'"

**#** or

Get-Service | where-object {$\_.DisplayName -like "\*sysm\*"}

It also can be done by checking the Windows registry

PowerShell

PS C:\Users\thm> reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-Windows-Sysmon/Operational

All these commands confirm if the sysmon tool is installed. Once we detect it, we can try to find the sysmon configuration file if we have readable permission to understand what system administrators are monitoring.

PowerShell

PS C:\Users\thm> findstr /si '<ProcessCreate onmatch="exclude">' C:\tools\\*

C:\tools\Sysmon\sysmonconfig.xml:

C:\tools\Sysmon\sysmonconfig.xml:

For more detail about the Windows sysmon tool and how to utilize it within endpoints, we suggest trying the TryHackMe room: [Sysmon](https://tryhackme.com/room/sysmon" \t "_blank).

**Host-based Intrusion Detection/Prevention System (HIDS/HIPS)**



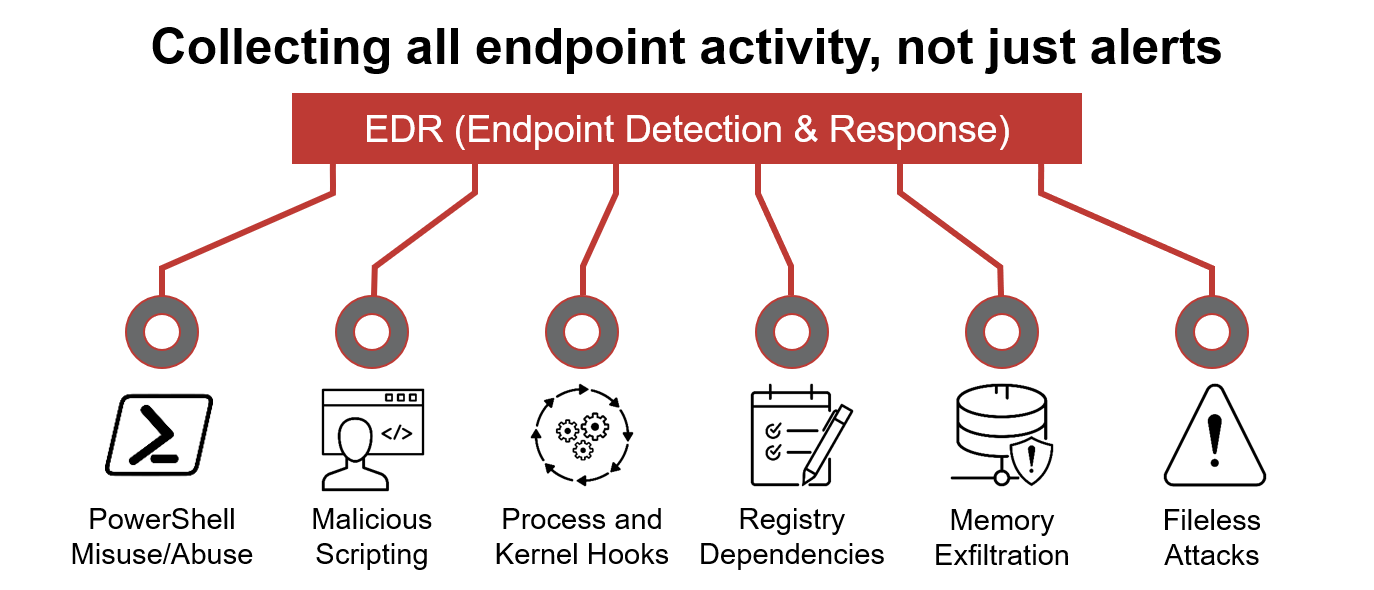
**HIDS**stands for Host-based Intrusion Detection System. It is software that has the ability to monitor and detect abnormal and malicious activities in a host. The primary purpose of HIDS is to detect suspicious activities and not to prevent them. There are two methods that the host-based or network intrusion detection system works, including:

* Signature-based IDS - it looks at checksums and message authentication.
* Anomaly-based IDS looks for unexpected activities, including abnormal bandwidth usage, protocols, and ports.

Host-based Intrusion Prevention Systems (**HIPS**) works by securing the operating system activities which where is installed. It is a detecting and prevention solution against well-known attacks and abnormal behaviors. HIPS is capable of auditing log files of the host, monitoring processes, and protecting system resources. HIPS is a mixture of best product features such as antivirus, behavior analysis, network, application firewall, etc.

There is also a network-based IDS/IPS, which we will be covering in the next task.

**Endpoint Detection and Response (EDR)**



It is also known as Endpoint Detection and Threat Response (EDTR). The EDR is a cybersecurity solution that defends against malware and other threats. EDRs can look for malicious files, monitor endpoint, system, and network events, and record them in a database for further analysis, detection, and investigation. EDRs are the next generation of antivirus and detect malicious activities on the host in real-time.

EDR analyze system data and behavior for making section threats, including

* Malware, including viruses, trojans, adware, keyloggers
* Exploit chains
* Ransomware

Below are some common EDR software for endpoints

* Cylance
* Crowdstrike
* Symantec
* SentinelOne
* Many others

Even though an attacker successfully delivered their payload and bypassed EDR in receiving reverse shell, EDR is still running and monitors the system. It may block us from doing something else if it flags an alert.

We can use scripts for enumerating security products within the machine, such as [Invoke-EDRChecker](https://github.com/PwnDexter/Invoke-EDRChecker) and [SharpEDRChecker](https://github.com/PwnDexter/SharpEDRChecker" \t "_blank). They check for commonly used Antivirus, EDR, logging monitor products by checking file metadata, processes, DLL loaded into current processes, Services, and drivers, directories.

***Answer the questions below***

We covered some of the common security endpoints we may encounter during the red team engagement. Let's discuss the network-based security solutions in the next task!

 Host Security Solution #2

In this task, we will keep discussing host security solutions.

4. **Security Event Logging and Monitoring**



By default, Operating systems log various activity events in the system using log files. The event logging feature is available to the IT system and network administrators to monitor and analyze important events, whether on the host or the network side. In cooperating networks, security teams utilize the logging event technique to track and investigate security incidents.

There are various categories where the Windows operating system logs event information, including the application, system, security, services, etc. In addition, security and network devices store event information into log files to allow the system administrators to get an insight into what is going on.

We can get a list of available event logs on the local machine using the Get-EventLog cmdlet.

PowerShell

PS C:\Users\thm> Get-EventLog -List

Max(K) Retain OverflowAction Entries Log

------ ------ -------------- ------- ---

512 7 OverwriteOlder 59 Active Directory Web Services

20,480 0 OverwriteAsNeeded 512 Application

512 0 OverwriteAsNeeded 170 Directory Service

102,400 0 OverwriteAsNeeded 67 DNS Server

20,480 0 OverwriteAsNeeded 4,345 System

15,360 0 OverwriteAsNeeded 1,692 Windows PowerShell

Sometimes, the list of available event logs gives you an insight into what applications and services are installed on the machine! For example, we can see that the local machine has Active Directory, DNS server, etc. For more information about the Get-EventLog cmdlet with examples, visit the [Microsoft documents website](https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.management/get-eventlog?view=powershell-5.1).

In corporate networks, log agent software is installed on clients to collect and gather logs from different sensors to analyze and monitor activities within the network. We will discuss them more in the Network Security Solution task.

**System Monitor**sysmon



Windows System Monitor sysmon is a service and device driver. It is one of the Microsoft Sysinternals suites. The sysmon tool is not an essential tool (not installed by default), but it starts gathering and logging events once installed. These logs indicators can significantly help system administrators and blue teamers to track and investigate malicious activity and help with general troubleshooting.

One of the great features of the sysmon  tool is that it can log many important events, and you can also create your own rule(s) and configuration to monitor:

* Process creation and termination
* Network connections
* Modification on file
* Remote threats
* Process and memory access
* and many others

For learning more about sysmon, visit the Windows document page [here](https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon).

As a red teamer, one of the primary goals is to stay undetectable, so it is essential to be aware of these tools and avoid causing generating and alerting events. The following are some of the tricks that can be used to detect whether the sysmon is available in the victim machine or not.

We can look for a process or service that has been named "Sysmon" within the current process or services as follows,

PowerShell

PS C:\Users\thm> Get-Process | Where-Object { $\_.ProcessName -eq "Sysmon" }

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName

------- ------ ----- ----- ------ -- -- -----------

373 15 20212 31716 3316 0 Sysmon

or look for services as follows,

PowerShell

PS C:\Users\thm> Get-CimInstance win32\_service -Filter "Description = 'System Monitor service'"

**#** or

Get-Service | where-object {$\_.DisplayName -like "\*sysm\*"}

It also can be done by checking the Windows registry

PowerShell

PS C:\Users\thm> reg query HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Channels\Microsoft-Windows-Sysmon/Operational

All these commands confirm if the sysmon tool is installed. Once we detect it, we can try to find the sysmon configuration file if we have readable permission to understand what system administrators are monitoring.

PowerShell

PS C:\Users\thm> findstr /si '<ProcessCreate onmatch="exclude">' C:\tools\\*

C:\tools\Sysmon\sysmonconfig.xml:

C:\tools\Sysmon\sysmonconfig.xml:

For more detail about the Windows sysmon tool and how to utilize it within endpoints, we suggest trying the TryHackMe room: [Sysmon](https://tryhackme.com/room/sysmon" \t "_blank).

**Host-based Intrusion Detection/Prevention System (HIDS/HIPS)**



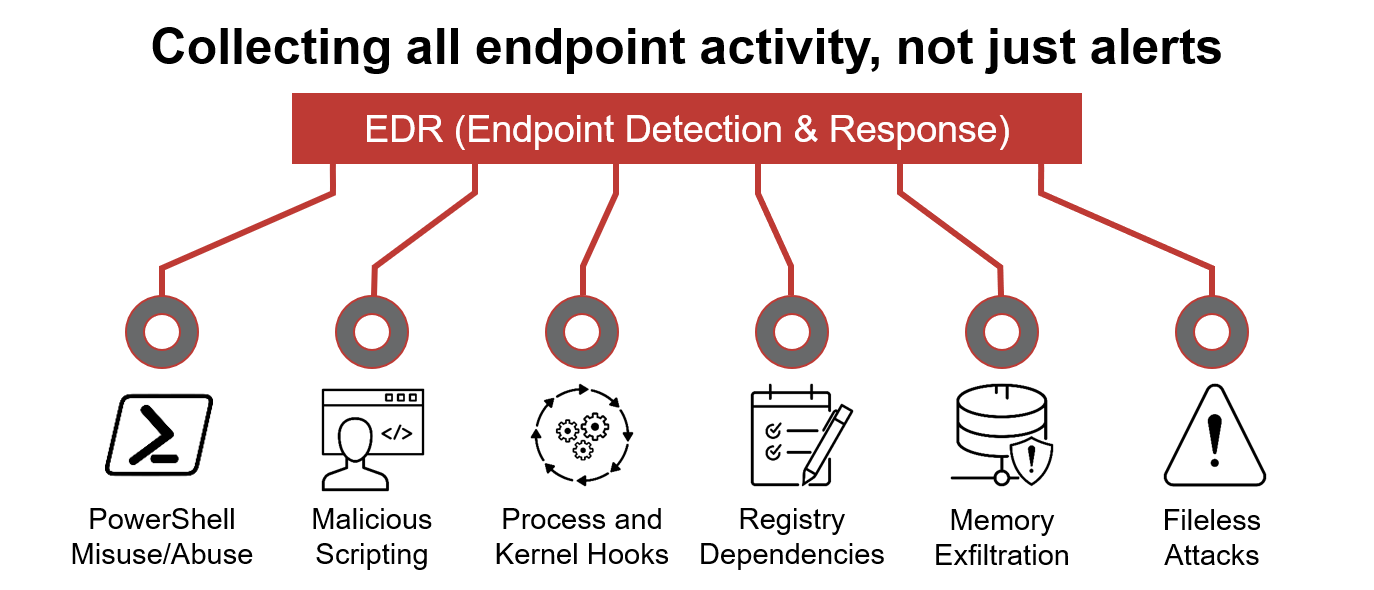
**HIDS**stands for Host-based Intrusion Detection System. It is software that has the ability to monitor and detect abnormal and malicious activities in a host. The primary purpose of HIDS is to detect suspicious activities and not to prevent them. There are two methods that the host-based or network intrusion detection system works, including:

* Signature-based IDS - it looks at checksums and message authentication.
* Anomaly-based IDS looks for unexpected activities, including abnormal bandwidth usage, protocols, and ports.

Host-based Intrusion Prevention Systems (**HIPS**) works by securing the operating system activities which where is installed. It is a detecting and prevention solution against well-known attacks and abnormal behaviors. HIPS is capable of auditing log files of the host, monitoring processes, and protecting system resources. HIPS is a mixture of best product features such as antivirus, behavior analysis, network, application firewall, etc.

There is also a network-based IDS/IPS, which we will be covering in the next task.

**Endpoint Detection and Response (EDR)**



It is also known as Endpoint Detection and Threat Response (EDTR). The EDR is a cybersecurity solution that defends against malware and other threats. EDRs can look for malicious files, monitor endpoint, system, and network events, and record them in a database for further analysis, detection, and investigation. EDRs are the next generation of antivirus and detect malicious activities on the host in real-time.

EDR analyze system data and behavior for making section threats, including

* Malware, including viruses, trojans, adware, keyloggers
* Exploit chains
* Ransomware

Below are some common EDR software for endpoints

* Cylance
* Crowdstrike
* Symantec
* SentinelOne
* Many others

Even though an attacker successfully delivered their payload and bypassed EDR in receiving reverse shell, EDR is still running and monitors the system. It may block us from doing something else if it flags an alert.

We can use scripts for enumerating security products within the machine, such as [Invoke-EDRChecker](https://github.com/PwnDexter/Invoke-EDRChecker) and [SharpEDRChecker](https://github.com/PwnDexter/SharpEDRChecker" \t "_blank). They check for commonly used Antivirus, EDR, logging monitor products by checking file metadata, processes, DLL loaded into current processes, Services, and drivers, directories.

***Answer the questions below***

We covered some of the common security endpoints we may encounter during the red team engagement. Let's discuss the network-based security solutions in the next task!

pplications and Services

This task will expand our knowledge needed to learn more about the system. We discussed account discovery and security products within the system in previous tasks. We will continue learning more about the system, including:

* Installed applications
* Services and processes
* Sharing files and printers
* Internal services: DNS and local web applications

It is necessary to understand what the system provides in order to get the benefit of the information.

**Installed Applications**

First, we start enumerating the system for installed applications by checking the application's name and version. As a red teamer, this information will benefit us. We may find vulnerable software installed to exploit and escalate our system privileges. Also, we may find some information, such as plain-text credentials, is left on the system that belongs to other systems or services.

 We will be using the wmic Windows command to list all installed applications and their version.

PowerShell

PS C:\Users\thm> wmic product get name,version

Name Version

Microsoft Visual C++ 2019 X64 Minimum Runtime - 14.28.29910 14.28.29910

AWS Tools for Windows 3.15.1248

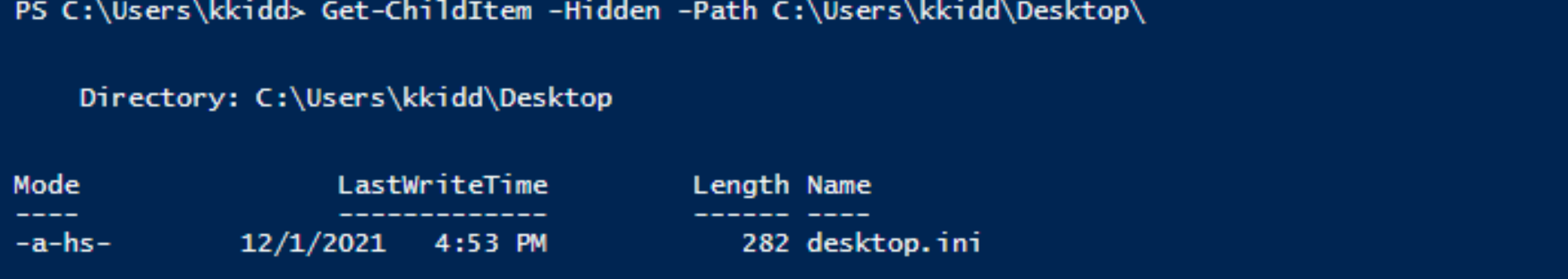
Amazon SSM Agent 3.0.529.0

aws-cfn-bootstrap 2.0.5

AWS PV Drivers 8.3.4

Microsoft Visual C++ 2019 X64 Additional Runtime - 14.28.29910 14.28.29910

Another interesting thing is to look for particular text strings, hidden directories, backup files. Then we can use the PowerShell cmdlets, Get-ChildItem, as follow:

**Services and Process**

Windows services enable the system administrator to create long-running executable applications in our own Windows sessions. Sometimes Windows services have misconfiguration permissions, which escalates the current user access level of permissions. Therefore, we must look at running services and perform services and processes reconnaissance.  For more details, you can read about process discovery on [Attack MITRE](https://attack.mitre.org/techniques/T1057/).

Process discovery is an enumeration step to understand what the system provides. The red team should get information and details about running services and processes on a system. We need to understand as much as possible about our targets. This information could help us understand common software running on other systems in the network. For example, the compromised system may have a custom client application used for internal purposes. Custom internally developed software is the most common root cause of escalation vectors. Thus, it is worth digging more to get details about the current process.

For more details about core Windows processes from the blue team perspective, check out the TryHackMe room: [Core Windows Process](https://tryhackme.com/room/btwindowsinternals).

**Sharing files and printers**



Sharing files and network resources is commonly used in personal and enterprise environments. System administrators misconfigure access permissions, and they may have useful information about other accounts and systems. For more information on printer hacking, we suggest trying out the following TryHackMe room: [Printer Hacking 101](https://tryhackme.com/room/printerhacking101).

**Internal services: DNS, local web applications, etc**

Internal network services are another source of information to expand our knowledge about other systems and the entire environment. To get more details about network services that are used for external and internal network services, we suggest trying out the following rooms: [Network Service](https://tryhackme.com/room/networkservices), [Network Service2](https://tryhackme.com/room/networkservices2).

The following are some of the internal services that are commonly used that we are interested in:

* DNS Services
* Email Services
* Network File Share
* Web application
* Database service

***Answer the questions below***

Let's try listing the running services using the Windows command prompt net start to check if there are any interesting running services.

PowerShell

PS C:\Users\thm> net start

These Windows services are started:

Active Directory Web Services

Amazon SSM Agent

Application Host Helper Service

Cryptographic Services

DCOM Server Process Launcher

DFS Namespace

DFS Replication

DHCP Client

Diagnostic Policy Service

THM Demo

DNS Client

We can see a service with the name THM Demo which we want to know more about.

Now let's look for the exact service name, which we need to find more information.

PowerShell

PS C:\Users\thm> wmic service where "name like 'THM Demo'" get Name,PathName

Name PathName

THM Service c:\Windows\thm-demo.exe

We find the file name and its path; now let's find more details using the Get-Process cmdlet.

PowerShell

PS C:\Users\thm> Get-Process -Name thm-demo

Handles NPM(K) PM(K) WS(K) CPU(s) Id SI ProcessName

------- ------ ----- ----- ------ -- -- -----------

82 9 13128 6200 3212 0 thm-service

Once we find its process ID, let's check if providing a network service by listing the listening ports within the system.

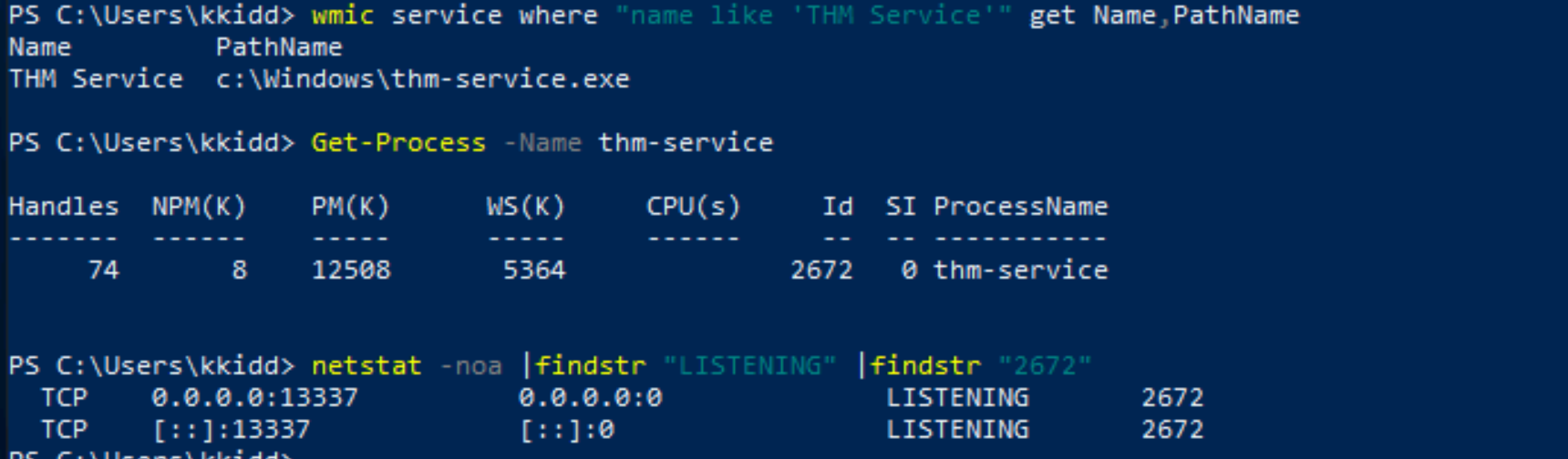
PowerShell

PS C:\Users\thm> netstat -noa |findstr "LISTENING" |findstr "3212"

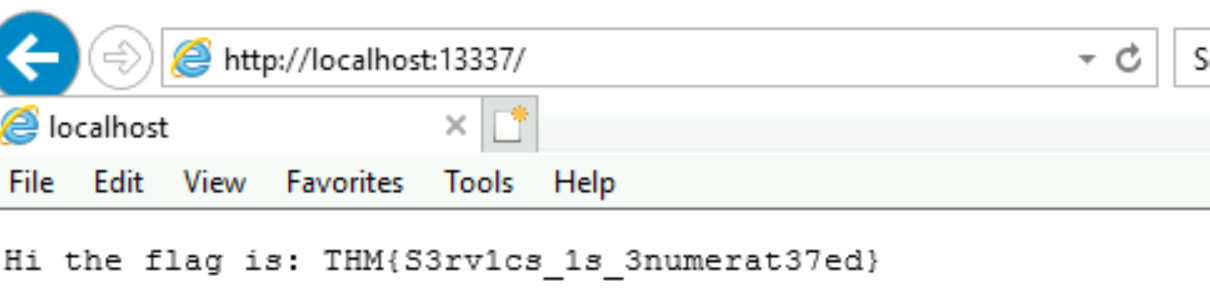
TCP 0.0.0.0:8080 0.0.0.0:0 LISTENING 3212

TCP [::]:8080 [::]:0 LISTENING 3212

Finally, we can see it is listening on port 8080. Now try to apply what we discussed and find the port number for THM Service. What is the port number?



Visit the localhost on the port you found in Question #1. What is the flag?

We mentioned that DNS service is a commonly used protocol in any active directory environment and network. The attached machine provides DNS services for AD. Let's enumerate the DNS by performing a zone transfer DNS and see if we can list all records.

We will perform DNS zone transfer using the Microsoft tool is nslookup.exe.

PowerShell

PS C:\Users\thm> nslookup.exe

Default Server: UnKnown

Address: ::1

Once we execute it, we provide the DNS server that we need to ask, which in this case is the target machine

NSlookup

> server 10.10.40.164

Default Server: [MACHINE\_IP]

Address: MACHINE\_IP

Now let's try the DNS zone transfer on the domain we find in the AD environment.

NSlookup

> ls -d thmredteam.com

[[10.10.40.164]]

thmredteam.com. SOA ad.thmredteam.com hostmaster.thmredteam.com. (732 900 600 86400 3600)

thmredteam.com. A MACHINE\_IP

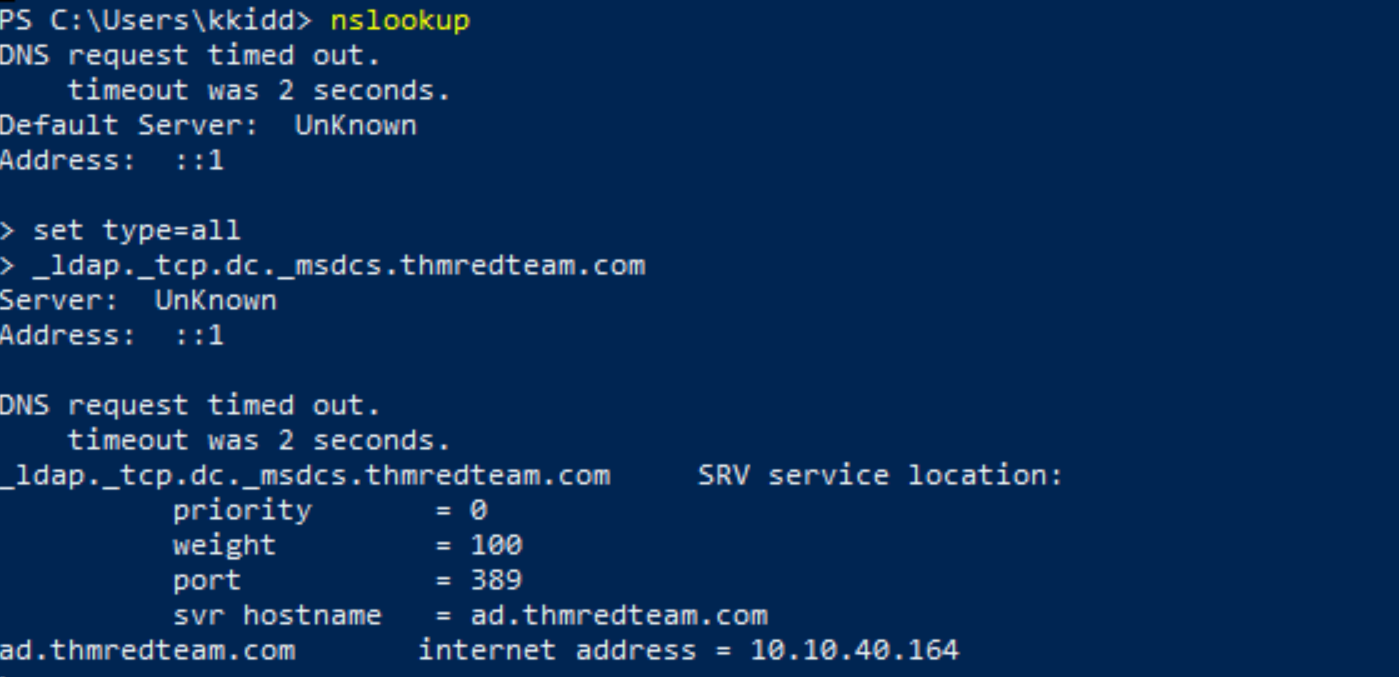
thmredteam.com. NS ad.thmredteam.com

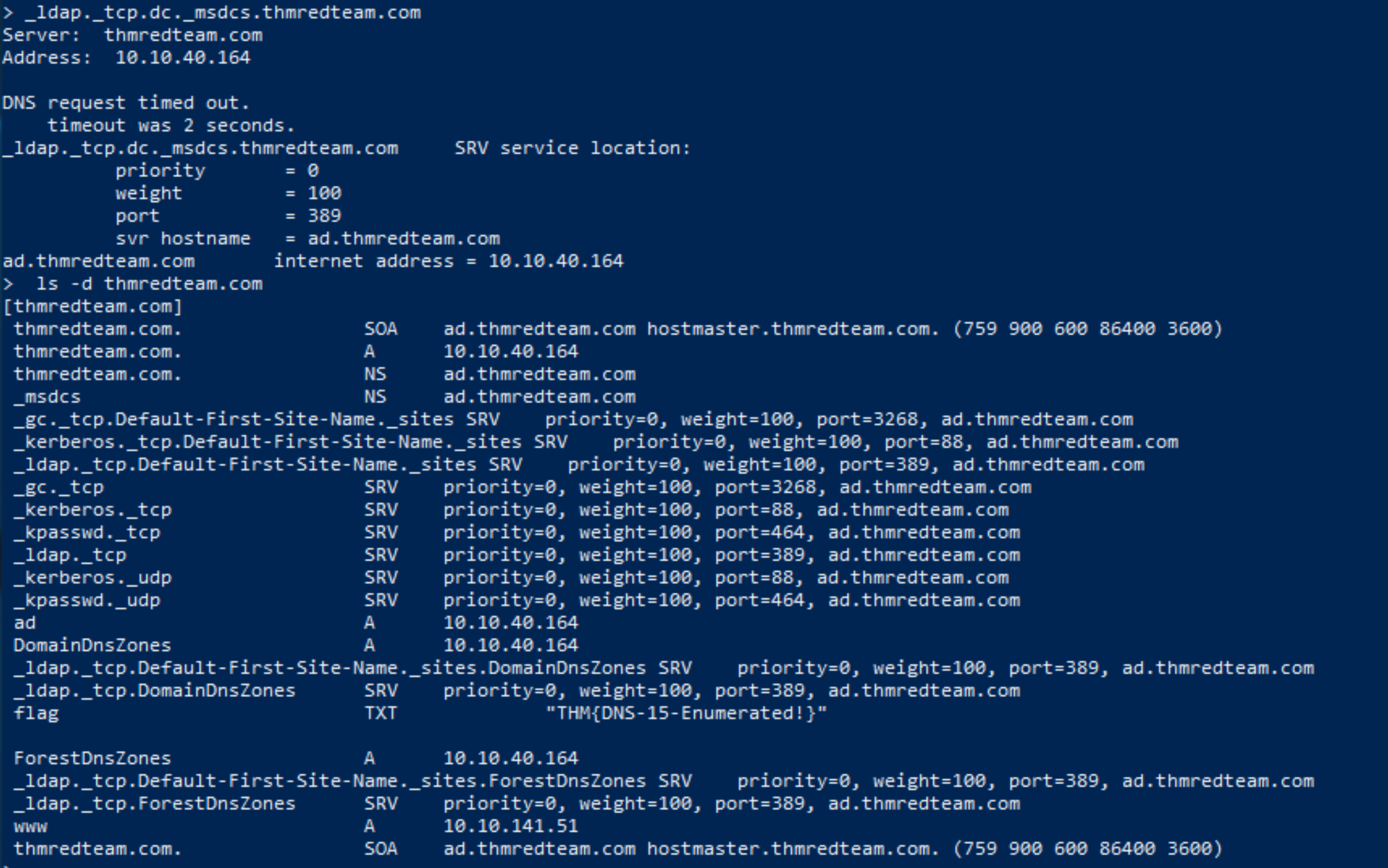
\*\*\*

ad A MACHINE\_IP

The previous output is an example of successfully performing the DNS zone transfer.

Now enumerate the domain name of the domain controller, thmredteam.com, using the nslookup.exe, and perform a DNS zone transfer. **What is the flag for one of the records?**





This room is an introduction to client systems in corporate environments. The student should have a better understanding of how clients are used in a corporate network including:

* Network Infrastructure
* AD environment
* security measures (HIPS, AV, etc.)
* Internal applications and services