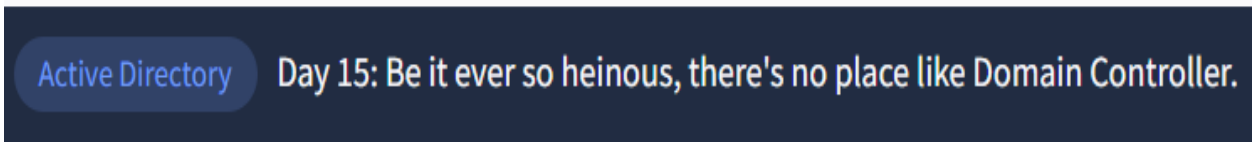


25.12.2024

Nikhil Kumar

The Story

Ahead of SOC-mas, the team decided to do a routine security check of one of their Active Directory domain controllers. Upon some quick auditing, the team noticed something was off. Could it be? The domain controller has been breached? With sweat on their brows, the SOC team smashed the glass and hit the panic alarm. There's only one person who can save us...

Learning Objectives

- Learn about the structures of Active Directory.
- Learn about common Active Directory attacks.
- Investigate a breach against an Active Directory.

Introducing Active Directory

Before diving into Active Directory, let us understand how network infrastructures can be mapped out and ensure that access to resources is well managed. This is typically done through **Directory Services,** which map and provide access to network resources within an organisation. The **Lightweight Directory Access Protocol (LDAP)** forms the core of Directory Services. It provides a mechanism for accessing and managing directory data to ensure that searching for and retrieving information about subjects and objects such as users, computers, and groups is quick.

Active Directory (AD) is, therefore, a Directory Service at the heart of most enterprise networks that stores information about objects in a network. The associated objects can include:

- **Users**: Individual accounts representing people or services
- **Groups**: Collections of users or other objects, often with specific permissions
- **Computers**: Machines that belong to the domain governed by AD policies
- **Printers** and other **resources**: Network-accessible devices or services

The building blocks of an AD architecture include:

- **Domains**: Logical groupings of network resources such as users, computers, and services. They serve as the main boundary for AD administration and can be identified by their **Domain Component and Domain Controller** name. Everything inside a domain is subject to the same security policies and permissions.
- **Organisational Units (OUs)**: OUs are containers within a domain that help group objects based on departments, locations or functions for easier management. Administrators can apply Group Policy settings to specific OUs, allowing more granular control of security settings or access permissions.
- **Forest**: A collection of one or more domains that share a standard schema, configuration, and global catalogue. The forest is the top-level container in AD.
- **Trust Relationships**: Domains within a forest (and across forests) can establish trust relationships that allow users in one domain to access resources in another, subject to permission.

Combining all these components allows us to establish the **Distinguished Name (DN)** that an object belongs to within the AD. The structure of the name would be as follows:

```
DN=CN=Mayor Malware, OU=Management, DC=wareville, DC=thm
```

Core Active Directory Components

Active Directory contains several key components that allow it to provide a wide range of services. Understanding these components will give one a clear picture of how AD supports administrative and security operations.

- **Domain Controllers (DCs):** Domain Controllers are the servers that host Active Directory services. They store the AD database and handle authentication and authorisation requests, such as logging in users or verifying access to resources. Multiple DCs can exist within a domain for redundancy. When changes are made to AD (such as adding users or updating passwords), these changes are replicated across all DCs, ensuring that the directory remains consistent.
- **Global Catalog:** The Global Catalog (GC) is a searchable database within AD that contains a subset of information from all objects in the directory. This allows users and services to locate objects in any domain in the forest, even if those objects reside in different domains.
- **LDAP (Lightweight Directory Access Protocol):** AD uses this protocol to query and modify the directory. The protocol allows for fast searching and retrieving of information about objects such as users, computers, and groups.
- **Kerberos Authentication:** The default authentication protocol used by AD provides secure authentication by using tickets rather than passwords.

Group Policy

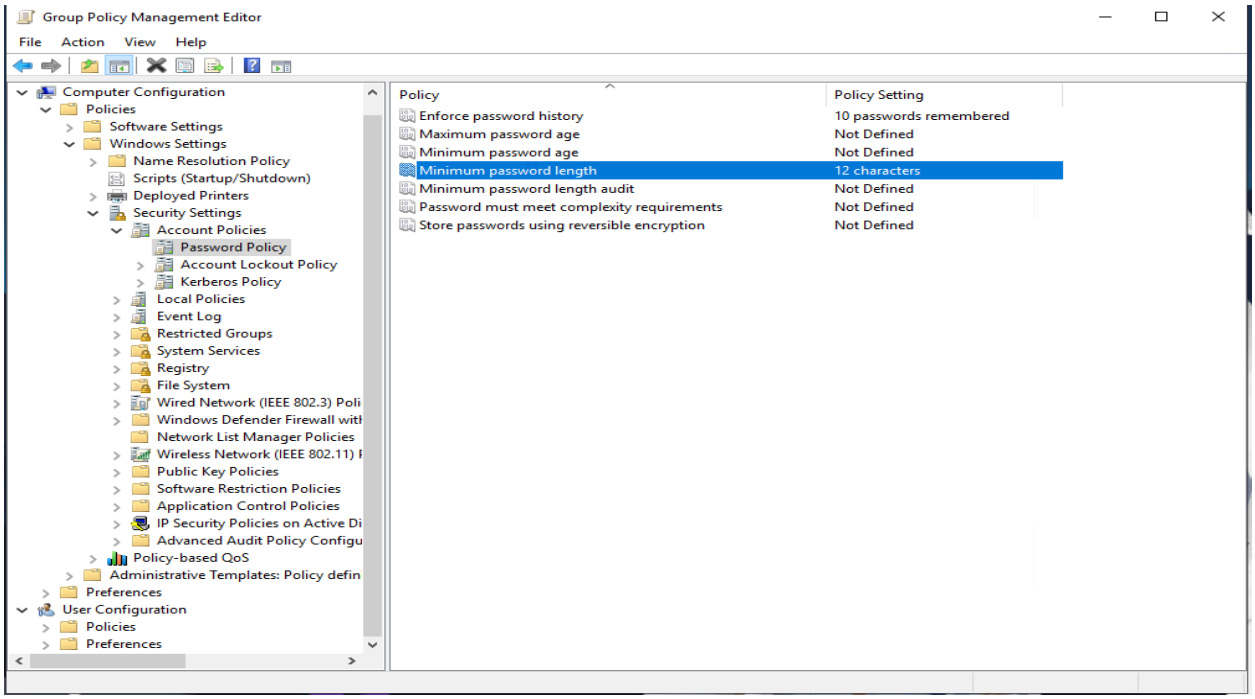
One of Active Directory's most powerful features is **Group Policy**, which allows administrators to enforce policies across the domain. Group Policies can be applied to users and computers to enforce password policies, software deployment, firewall settings, and more.

Group Policy Objects (GPOs) are the containers that hold these policies. A GPO can be linked to the entire domain, an OU, or a site, giving the flexibility in applying policies.

Let us say that McSkidy wants to ensure that all users within Wareville's SOC follow a strict password policy, enforcing minimum password lengths and complexity rules. Here is how it would be done:

1. Using the Run window, open **Group Policy Management** from your server by typing `gpmc.msc`.
2. Right-click your domain and select **"Create a GPO in this domain, and Link it here"**. Name the new GPO **"Password Policy"**.
3. Edit the GPO by navigating to **Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Account Policies -> Password Policy**.
4. Configure the following settings:
    - Minimum password length: 12 characters
    - Enforce password history: 10 passwords
    - Maximum password age: 90 days
    - Password must meet complexity requirements: Enabled
5. Click **OK**, then link this GPO to the domain or specific OUs you want to target.

This policy will now be applied across the domain, ensuring all users meet these password requirements.



Common Active Directory Attacks

Adversaries are always looking for ways to breach and exploit Active Directory environments to destabilise and cause havoc to organisations. Working with Glitch to secure SOC-mas requires us to know common attacks and their mitigation measures.

Golden Ticket Attack

A **Golden Ticket** attack allows attackers to exploit the Kerberos protocol and impersonate any account on the AD by forging a Ticket Granting Ticket (TGT). By compromising the **krbtgt** account and using its password hash, the attackers gain complete control over the domain for as long as the forged ticket remains valid. The attack requires four critical pieces of information to be successful:

- Fully Qualified Domain Name (FQDN) of the domain
- SID of the domain
- Username of an account to impersonate
- KRBTGT account password hash

Detection for this type of attack involves monitoring for unusual activity involving the **krbtgt**

- **Event ID 4768**: Look for TGT requests for high-privilege accounts.
- **Event ID 4672**: This logs when special privileges (such as SeTcbPrivilege) are assigned to a user.

Pass-the-Hash

This type of attack steals the hash of a password and can be used to authenticate to services without needing the actual password. This is possible because the NTLM protocol allows authentication based on password hashes.

Key ways to mitigate this attack are enforcing strong password policies, conducting regular audits on account privileges, and implementing multi-factor authentication across the domain.

Kerberoasting

Kerberoasting is an attack targeting Kerberos in which the attacker requests service tickets for accounts with Service Principal Names (SPNs), extracts the tickets and password hashes, and then attempts to crack them offline to retrieve the plaintext password.

Mitigation for this type of attack involves ensuring that service accounts are secured with strong passwords, and therefore, implementing secure policies across the AD would be the defence.

Pass-the-Ticket

In a **Pass-the-Ticket** attack, attackers steal Kerberos tickets from a compromised machine and use them to authenticate as the user or service whose ticket was stolen.

This attack can be detected through monitoring for suspicious logins using **Event ID 4768** (TGT request), especially if a user is logging in from unusual locations or devices. Additionally, **Event ID 4624** (successful login) will reveal tickets being used for authentication.

Malicious GPOs

Adversaries are known to abuse Group Policy to create persistent, privileged access accounts and distribute and execute malware by setting up policies that mimic software deployment across entire domains. With escalated privileges across the domain, attackers can create GPOs to accomplish goals at scale, including disabling core security software and features such as firewalls, antivirus, security updates, and logging. Additionally, scheduled tasks can be created to execute malicious scripts or exfiltration data from affected devices across the domain.

To mitigate against the exploitation of Group Policy, GPOs need to be regularly audited for unauthorised changes. Strict permissions and procedures for GPO modifications should also be enforced.

Skeleton Key Attack

In a **Skeleton Key** attack, attackers install a malware backdoor to log into any account using a master password. The legitimate password for each account would remain unchanged, but attackers can bypass it using the skeleton key password.

Investigating an Active Directory Breach

Group Policy

As previously discussed in this task, Group Policy is a means to distribute configurations and policies to enrolled devices in the domain. For attackers, Group Policy is a lucrative means of spreading malicious scripts to multiple devices.

Reviewing Group Policy Objects (GPOs) is a great investigation step. In this section, we will use PowerShell to audit our GPOs. First, we can use the `Get-GPO` cmdlet to list all GPOs installed on the domain controller.

Listing all GPOs viaPowerShell

```
PS C:\Users\Administrator> Get-GPO -All

DisplayName      : Default Domain Policy
DomainName       : wareville.thm
Owner            : WAREVILLE\Domain Admins
Id               : 31b2f340-016d-11d2-945f-00c04fb984f9
GpoStatus        : AllSettingsEnabled
Description      :
CreationTime     : 10/14/2024 12:17:31 PM
ModificationTime : 10/14/2024 12:19:28 PM
UserVersion      : AD Version: 0, SysVol Version: 0
ComputerVersion  : AD Version: 3, SysVol Version: 3
WmiFilter        :
```

```
DisplayName       : Default Domain Controllers Policy
DomainName        : wareville.thm
Owner             : WAREVILLE\Domain Admins
Id                : 6ac1786c-016f-11d2-945f-00c04fb984f9
GpoStatus         : AllSettingsEnabled
Description       :
CreationTime      : 10/14/2024 12:17:31 PM
ModificationTime  : 10/14/2024 12:17:30 PM
UserVersion       : AD Version: 0, SysVol Version: 0
ComputerVersion   : AD Version: 1, SysVol Version: 1
WmiFilter         :

DisplayName       : SetWallpaper GPO
DomainName        : wareville.thm
Owner             : WAREVILLE\Domain Admins
Id                : d634d7c1-db7a-4c7a-bf32-efca23d93a56
GpoStatus         : AllSettingsEnabled
Description       : Set the wallpaper of every domain joined
machine
CreationTime      : 10/30/2024 9:01:36 AM
ModificationTime  : 10/30/2024 9:01:36 AM
UserVersion       : AD Version: 0, SysVol Version: 0
ComputerVersion   : AD Version: 0, SysVol Version: 0
WmiFilter         :
```

This would allow us to look for out-of-place GPOs. We can export a GPO to an HTML file for further investigation to make it easier to see what configurations the policy enforces. For this example, we will export the "SetWallpaper" GPO.

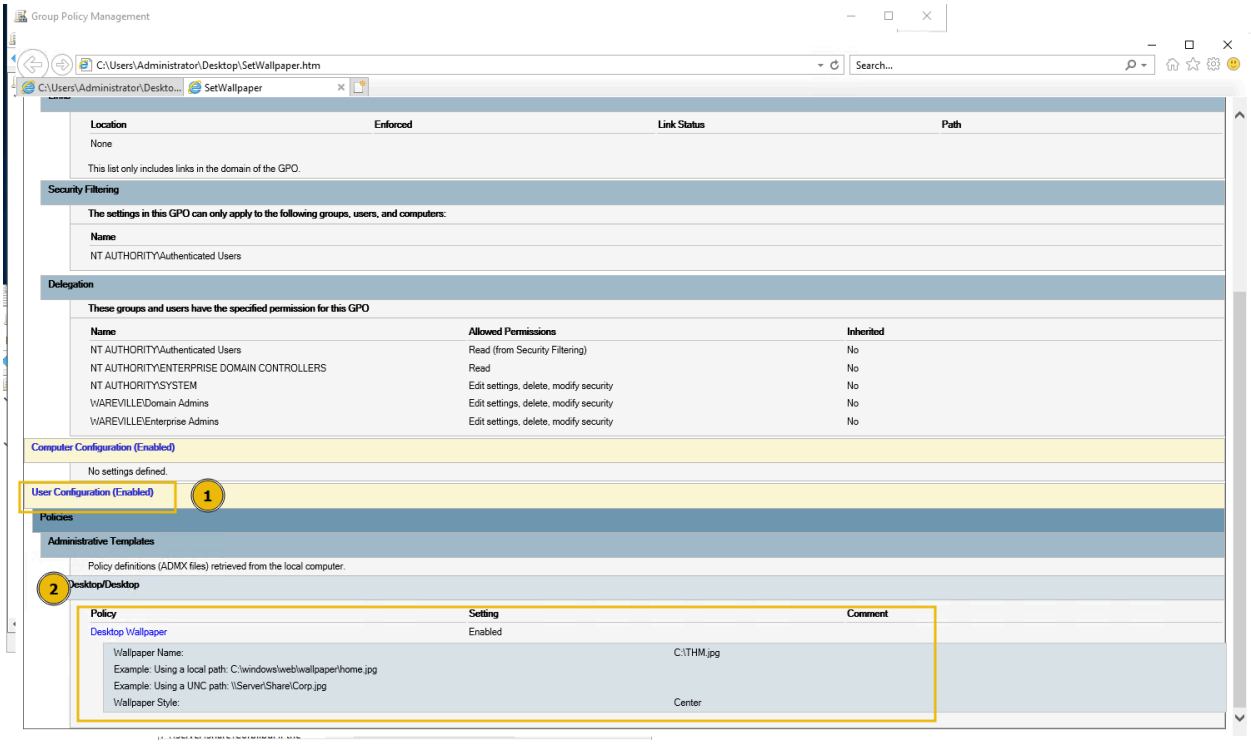
Please note that this is a demonstration GPO, and isn't present on the practical machine for today's task.

Exporting SetWallpaperGPO

```
PS C:\Users\Administrator> Get-GPOReport -Name "SetWallpaper"
-ReportType HTML -Path ".\SetWallpaper.html"
```

Then, when opening the HTML file in the browser, we are presented with an overview of things such as:

- When the policy was created and modified.
- What devices or users the GPO applies to.
- The permissions over the GPO.
- The user or computer configurations that it enforces.



From the screenshot above, we can see that the policy sets the Desktop Wallpaper of devices using the image located in C:\THM.jpg on the domain controller.

Domains are naturally likely to have many GPOs. We can use the same Get-GPO cmdlet, with a bit of *PowerShell-fu* to list only those GPOs that were recently modified. This is a handy snippet because it highlights policies that were recently modified - perhaps by an attacker.

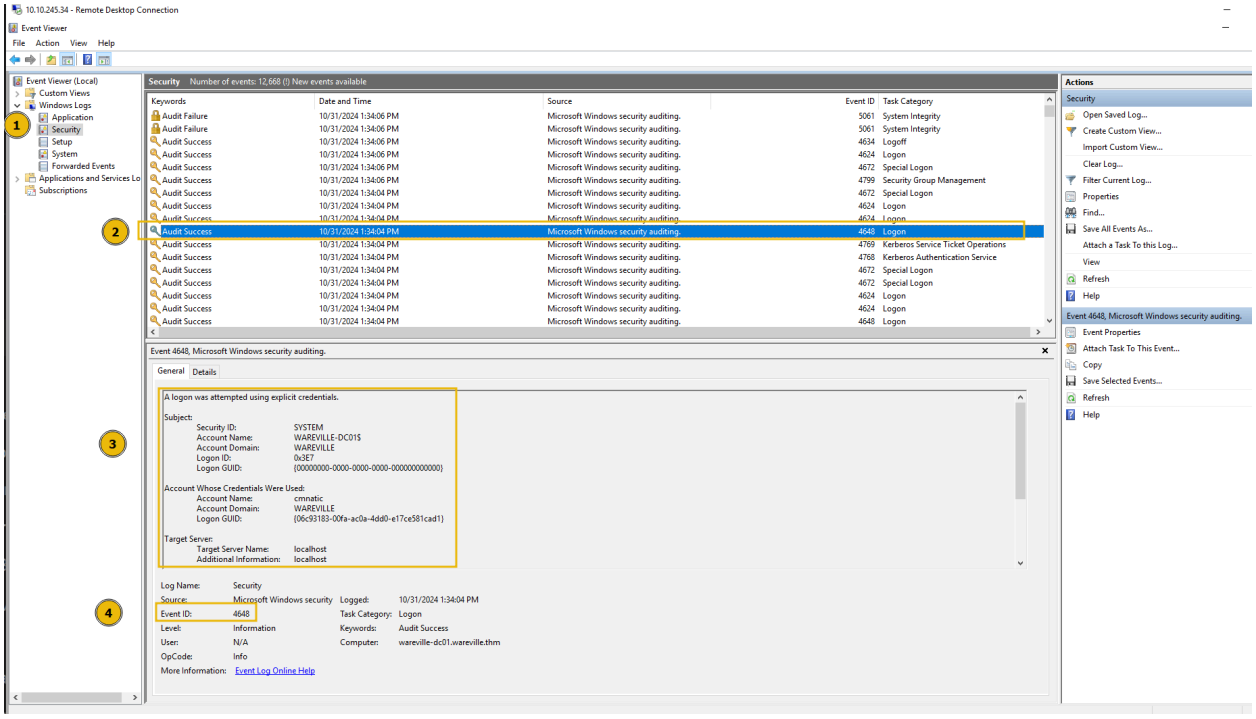
Listing recently modified GPOs

```
PS C:\Users\Administrator\Desktop> Get-GPO -All |
Where-Object { $_.ModificationTime } | Select-Object
DisplayName, ModificationTime


DisplayName                                   ModificationTime
-----------                                   ----------------
Default Domain Policy                         10/14/2024
12:19:28 PM
Default Domain Controllers Policy             10/14/2024
12:17:30 PM
SetWallpaper                                10/31/2024 1:01:04 PM
```

Event Viewer

Windows comes packaged with the Event Viewer. This invaluable repository stores a record of system activity, including security events, service behaviours, and so forth.

For example, within the "Security" tab of Event Viewer, we can see the history of user logins, attempts and logoffs. The screenshot below shows a record of the user "cmnatic" attempting to log into the device.



All categories of events are given an event ID. The table below provides notable event IDs for today's task.

Event ID	Description
4624	A user account has logged on
4625	A user account failed to log on
4672	Special privileges (i.e. SeTcbPrivilege) have been assigned to a user
4768	A TGT (Kerberos) ticket was requested for a high-privileged account

User Auditing

User accounts are a valuable and often successful method of attack. You can use Event Viewer IDs to review user events and PowerShell to audit their status. Attack methods such as password spraying will eventually result in user accounts being locked out, depending on the domain controller's lockout policy.

To view all locked accounts, you can use the Search-ADAccount cmdlet, applying some filters to show information such as the la00000000st time the user had successfully logged in.

```
Search-ADAccount -LockedOut | Select-Object Name,
SamAccountName, LockedOut, LastLogonDate, DistinguishedName
```

Additionally, a great way to quickly review the user accounts present on a domain, as well as their group membership, is by using the `Get-ADUser` cmdlet, demonstrated below:

Listing all users and their groups usingPowerShell

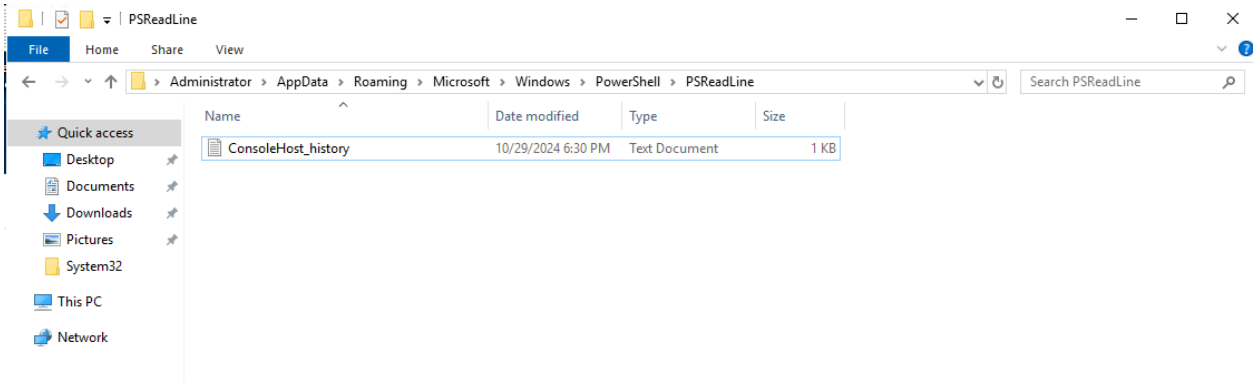
```
PS C:\Users\Administrator\Desktop> Get-ADUser -Filter *
-Properties MemberOf | Select-Object Name, SamAccountName,
@{Name="Groups";Expression={$_.MemberOf}}


Name           SamAccountName Groups
----           -------------- ------
Administrator  Administrator  {CN=Group Policy Creator
Owners,CN=Users,DC=wareville,DC=thm, CN=Domain
Admins,CN=Users,DC=wareville,DC=thm, CN=Enterprise
Admins,CN=Users,DC=wareville,DC=thm, CN=Schema ...
Guest          Guest
CN=Guests,CN=Builtin,DC=wareville,DC=thm
```

```
krbtgt          krbtgt          CN=Denied RODC Password
Replication Group,CN=Users,DC=wareville,DC=thm

tryhackme       tryhackme       CN=Domain
Admins,CN=Users,DC=wareville,DC=thm

DAVID           DAVID

James           James

NewAccount      NewAccount
cmnatic         cmnatic         {CN=Domain
Admins,CN=Users,DC=wareville,DC=thm, CN=Remote Desktop
Users,CN=Builtin,DC=wareville,DC=thm}
```

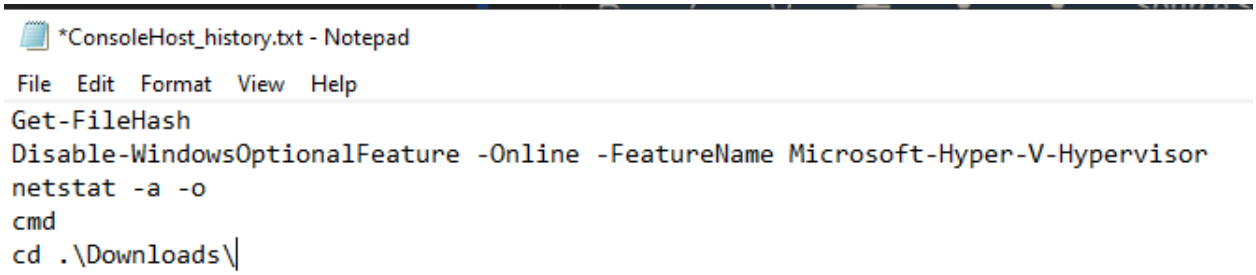
Reviewing PowerShell History and Logs

PowerShell, like Bash on Linux, keeps a history of the commands inputted into the session. Reviewing these can be a fantastic way to see recent actions taken by the user account on the machine.

On a Windows Server, this history file is located at `%APPDATA%\Microsoft\Windows\PowerShell\PSReadLine\ConsoleHost_history.txt`.



You can use the in-built Notepad on Windows or your favourite text editor to review the PowerShell command history.



Additionally, logs are recorded for every PowerShell process executed on a system. These logs are located within the Event Viewer under `Application and Services Logs -> Microsoft -> Windows -> PowerShell -> Operational` or also under `Application and Service Logs -> Windows PowerShell`. The logs have a wealth of information useful for incident response.