Securing Active Directory with PowerShell & Other Tools

James Tyler

Overview

- Introduction
- Resources
- Security Philosophy
- Assessment Tools
- Configuration Recommendations
- Testing Strategies
- Question and Answer Session

POWERSHELL
FOR SYSTEMS ENGINEERS
LEVERAGING AUTOMATION AND CHATGPT TO SOLVE BUSINESS PROBLEMS
JAMES TYLER
NEW AT 5
JIM TYLER
NILES COMMUNITY SCHOOLS DIRECTOR OF TECHNOLOGY

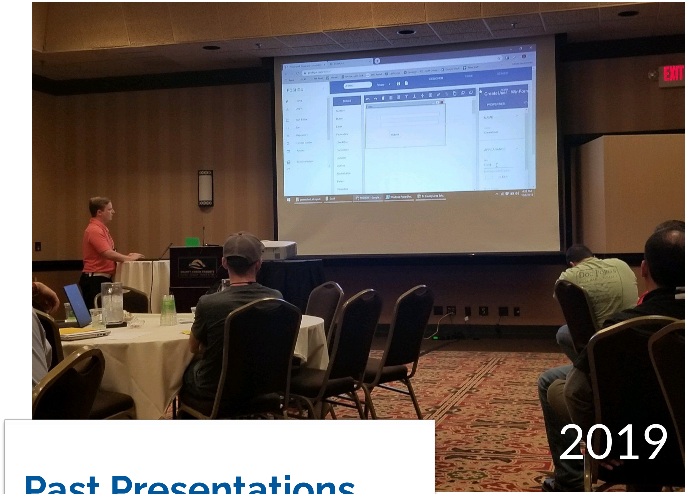
About Jim

- Director of Technology, Niles Community Schools
- Board Member, Watervliet Public Schools
- President, Watervliet Baseball/Softball Rec Council
- Board President, Children's Music Workshop
- Planning Commission Member, City of Watervliet
- Owner/Coach, Eagles Travel Ball, LLC
- Former Amazon Engineer
- Attending MAEDS since 2017
- Author of PowerShell for Systems Engineers
- Free PowerShell training on my YouTube Channel - 350K+ Views

-MS '17 Capella University, BA '13 WMU

Soap Box Time…

- Don’t fall victim to imposter syndrome!
- I’m going to put on my Chris Thomas hat…
- Everyone has something valuable to share
- **Do not be afraid to ask questions.**
- **Present next year!**



Past Presentations

2023 - PowerShell AI

2023 - PowerShell GUIs 2.0

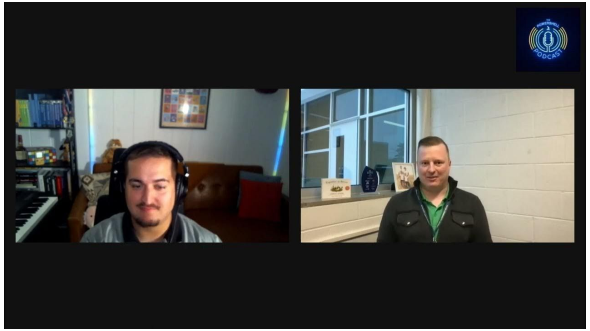
2022 - PowerShell GUIs

2022 - PowerShell Cloud Storage Methods (AWS, Azure, GCP)

2019 - *Brief* PowerShell

GUI demo

PowerShell Resources



- PowerShell Podcast -
  I appeared as a guest earlier this year
  https://www.youtube.com/watch?v=eWaCFWCId0w

- How to Harden Active Directory to Prevent Cyber Attacks - Spencer Alessi

- Cyber Threat Perspective Podcast - Spencer Alessi

- Steve Lee, Principal Software Engineering Manager of PowerShell
  https://x.com/steve_msft

PowerShell Videos

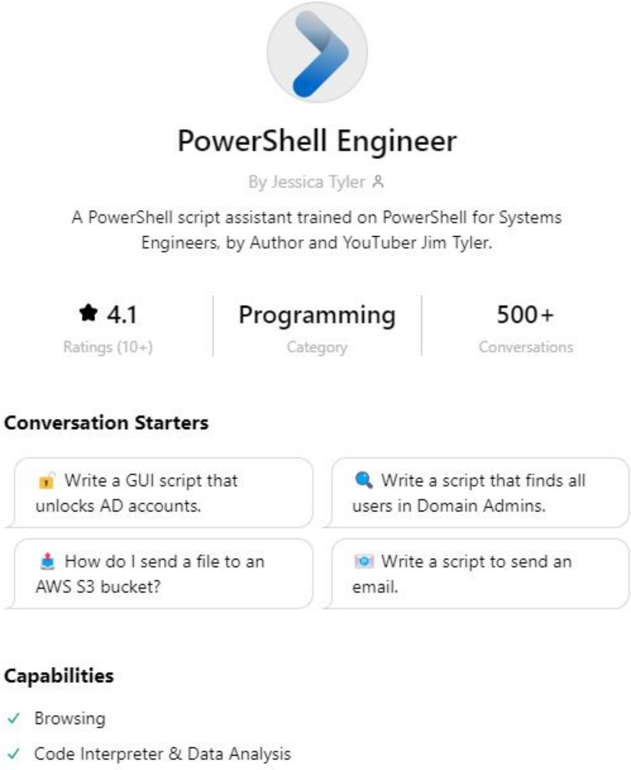
- Learn PowerShell in Less Than 2 Hours
  https://www.youtube.com/watch?v=ZOoCaWyifmI&t=4200s
- Learn PowerShell Automation in Less Than 1 Hour
  https://www.youtube.com/watch?v=ssS3dd6oQTU
- Learn PowerShell with Active Directory in Less Than 2 Hours
  https://www.youtube.com/watch?v=9oiEOYFe6PI

Learn PowerShell in Less ...

youtube.com

PowerShell Engineer GPT

- PowerShell Engineer GPT trained on my book, *PowerShell for Systems Engineers*: https://chatgpt.com/g/g-QvqZeqUjK-powershell-engineer



PowerShell Engineer

By Jessica Tyler

A PowerShell script assistant trained on PowerShell for Systems Engineers, by Author and YouTuber Jim Tyler.

★ 4.1
Ratings (10+)

Programming
Category

500+
Conversations

Conversation Starters

🔓 Write a GUI script that unlocks AD accounts.

🔍 Write a script that finds all users in Domain Admins.

📤 How do I send a file to an AWS S3 bucket?

📧 Write a script to send an email.

Capabilities

✓ Browsing

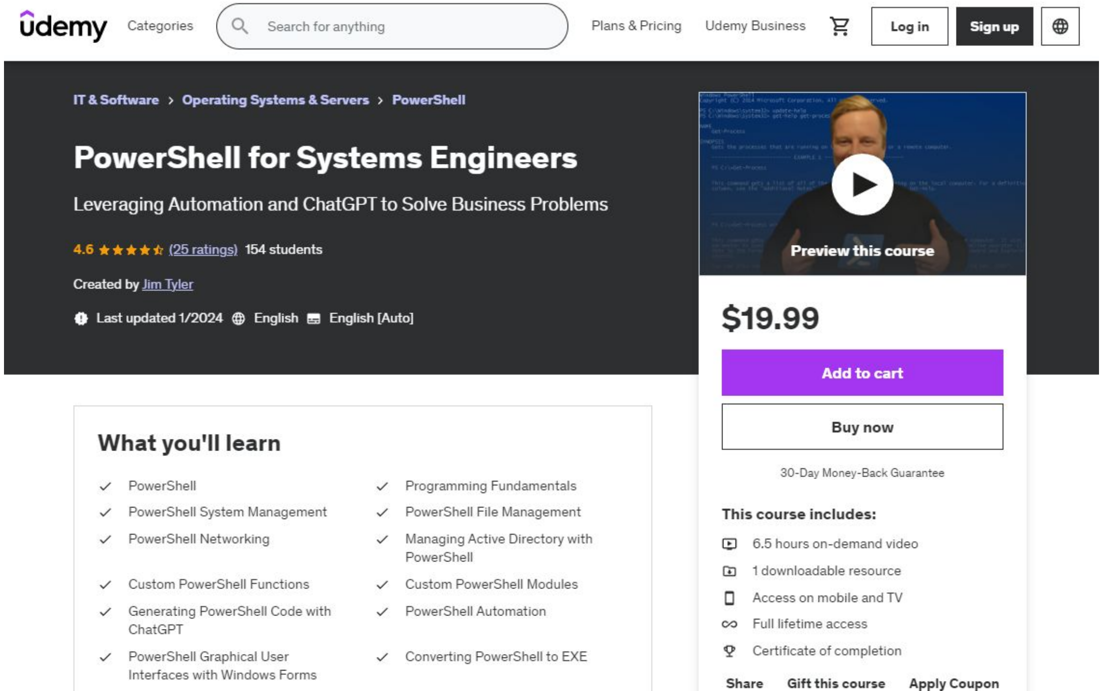
✓ Code Interpreter & Data Analysis

PowerShell Course on Udemy

- PowerShell Course based on my book, *PowerShell for Systems Engineers*: https://www.udemy.com/course/powershell-for-systems-engineers/?couponCode=MAEDS2024

Free to all MAEDS attendees.

MAEDS2024



IT & Software > Operating Systems & Servers > PowerShell

PowerShell for Systems Engineers

Leveraging Automation and ChatGPT to Solve Business Problems

4.6 ★★★★☆ (25 ratings) 154 students

Created by Jim Tyler

Last updated 1/2024 English English [Auto]

$19.99

30-Day Money-Back Guarantee

What you'll learn

- ✓ PowerShell
- ✓ PowerShell System Management
- ✓ PowerShell Networking
- ✓ Custom PowerShell Functions
- ✓ Generating PowerShell Code with ChatGPT
- ✓ PowerShell Graphical User Interfaces with Windows Forms
- ✓ Programming Fundamentals
- ✓ PowerShell File Management
- ✓ Managing Active Directory with PowerShell
- ✓ Custom PowerShell Modules
- ✓ PowerShell Automation
- ✓ Converting PowerShell to EXE

This course includes:

- 6.5 hours on-demand video
- 1 downloadable resource
- Access on mobile and TV
- Full lifetime access
- Certificate of completion

Code Examples & Slides

https://github.com/jimrtyler/maeds/

GitHub

Bias for Action

Speed matters in business. **Many decisions and actions are reversible** and do not need extensive study. We value calculated risk taking.

Bias for Action (Continued)

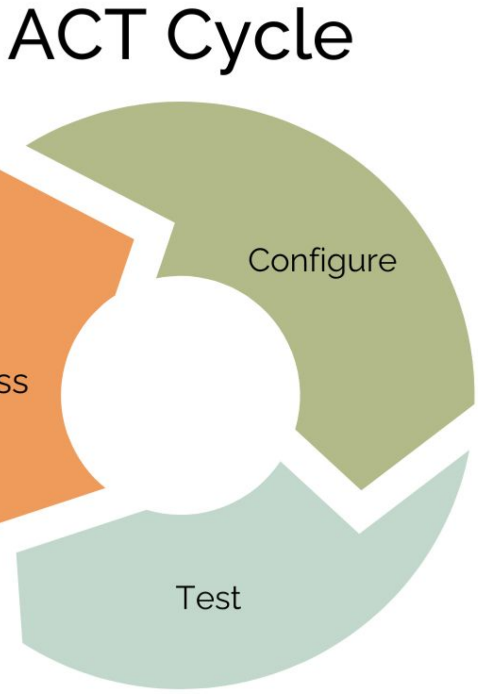
Many actions do not need extensive study. We are going to cover a lot of topics that I do not completely understand, but I know what remediation needs to be undergone.

Assumed Breach - A Better Model

Security teams should not operate under the assumption that a breach may happen, **but that it will happen**.

Security is Never a Destination

We need to follow a cyclical approach to information security.



Guiding Principle

Least Privilege / Need-to-Know

Limiting user access rights to only what is strictly necessary for their role.

Benefits:

- Reduces risk of unauthorized access.
- Mitigates damage from compromised accounts.

Guiding Principle

Minimize Privileged Accounts

Reducing the number of privileged accounts to a minimum.

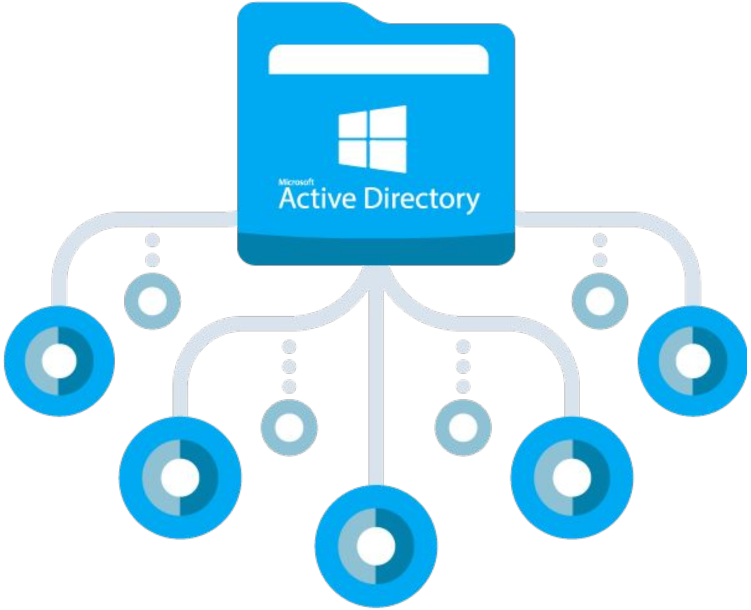
Why It Matters:

- Privileged accounts are prime targets for attackers.
- Fewer privileged accounts lead to reduced attack surface.

Best Practices:

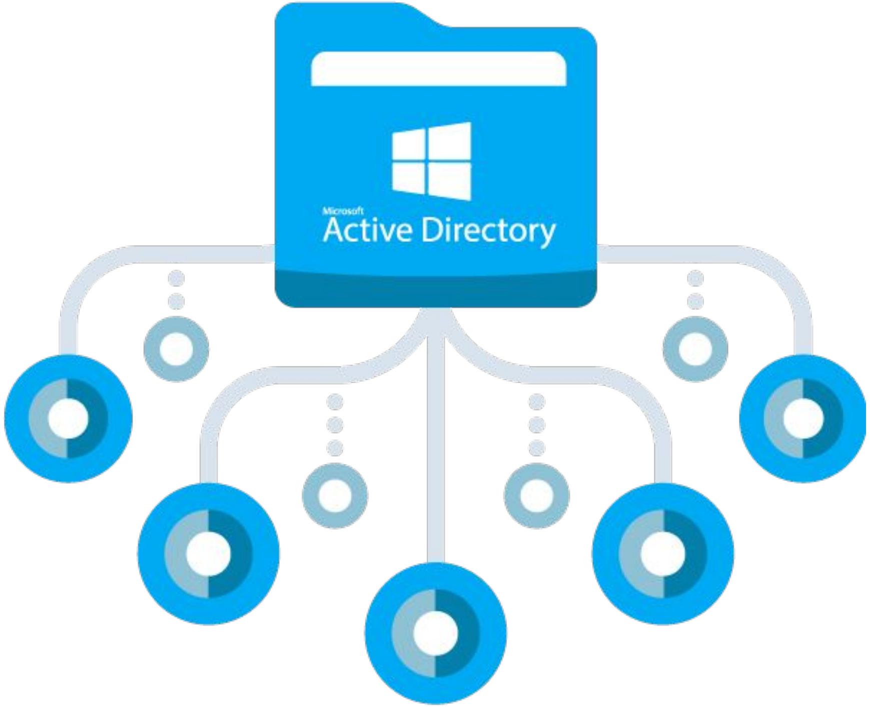
- Use Role-Based Access Control (RBAC).
- Assign temporary elevated privileges only when necessary.

Active Directory is Not Security Software



Why is Active Directory such a risk?

- Active Directory is a centralized, network control structure.
- Active Directory is very complex and a lot of things are hidden.
- Active Directory does not warn you about bad configurations.
- Active Directory users have a lot of permissions to a lot of resources.



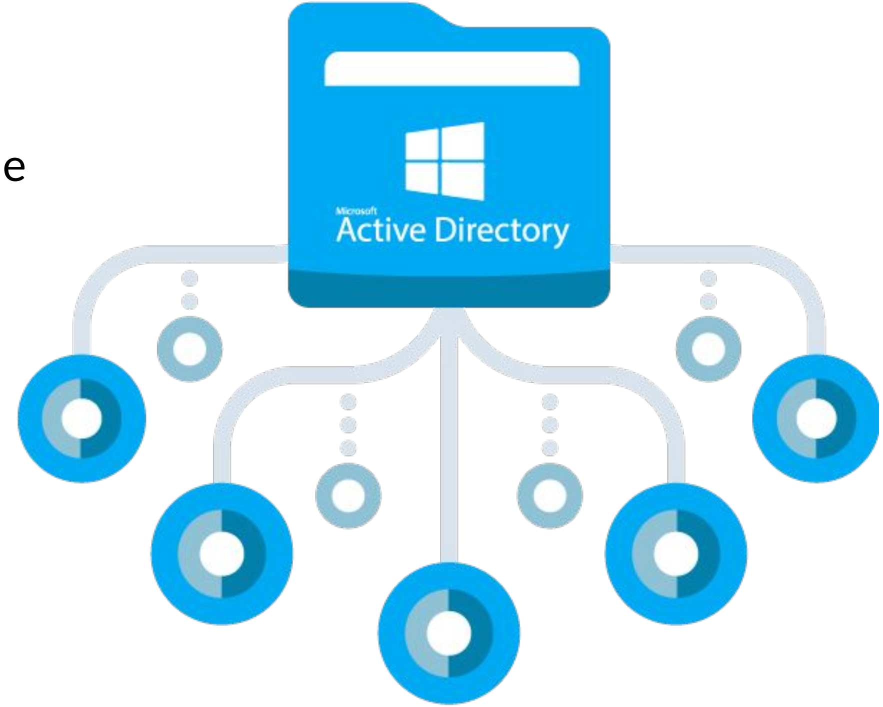
Why is Active Directory such a risk?

Active Directory is the starting point for

for many threat actors following this simple

pattern of compromise:

1. Credentials
2. Access
3. Control

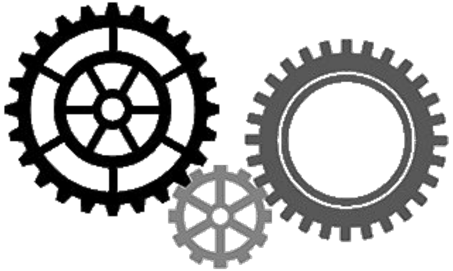


Known Active Directory Related Breaches

- Colonial Pipeline Attack (2021) - Utilized disabled account password hashes to laterally compromise other accounts, infiltrating the network via VPN that had no MFA.

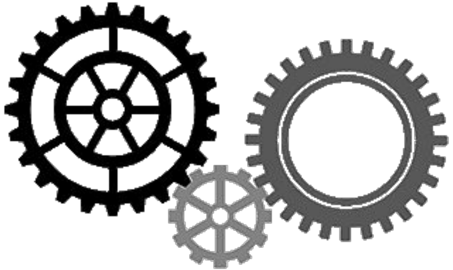
- Waikato District Health Board (New Zealand) 2021 - Redacted, but attackers used Windows 7 PCs as initial catalyst for account, presumably laterally moving around the network with AD permissions.

Step 1 - Assess



Objective: Evaluate and identify vulnerabilities and misconfigurations in Active Directory, including those related to ransomware attacks.

Step 1 - Assess



Ping Castle: Generate health reports and identify privilege escalation paths, outdated policies, and risks related to ransomware attacks.

Adeleg: Audit delegation rights to find over-privileged accounts and unnecessary permissions.

Script Sentry: Review PowerShell scripts running across the network for malicious or unauthorized executions.

Locksmith: Review privileged accounts and group memberships to ensure least privilege principles are enforced.

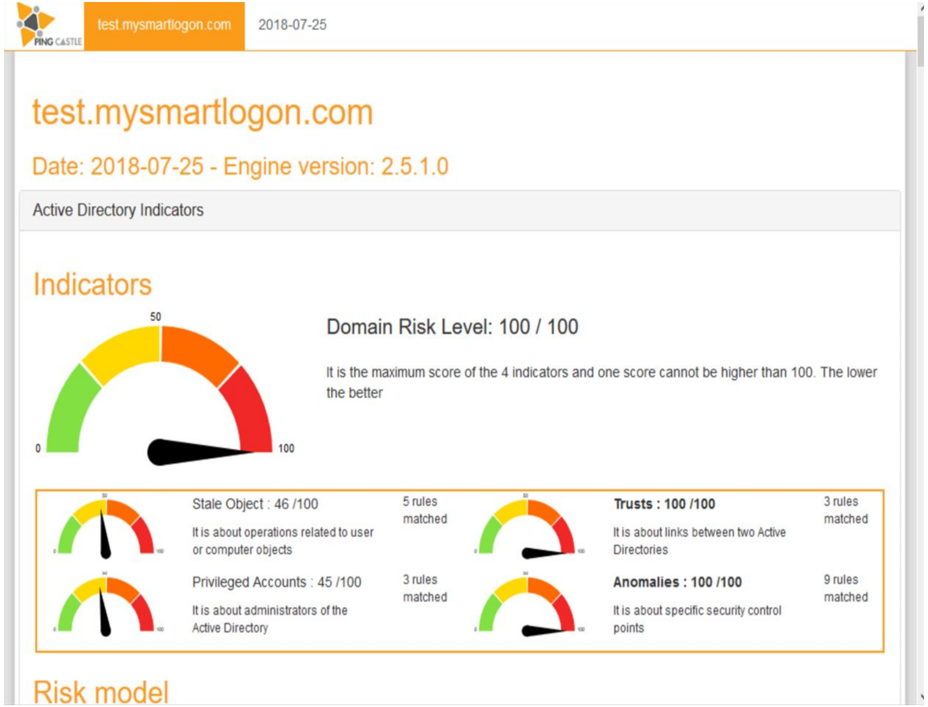
Critical Tools to Identify Misconfigurations

- PingCastle - https://www.pingcastle.com/
- Script Sentry - https://github.com/techspence/ScriptSentry
- ADeleg - https://github.com/mtth-bfft/adeleg
- Locksmith - https://github.com/TrimarcJake/Locksmith?tab=readme-ov-file

Ping Castle

- https://www.pingcastle.com/
- Provides health check analysis of Active Directory environment
- Provides actionable remediation steps for misconfigurations
- Reports should be run quarterly or annually

Ping Castle



test.mysmartlogon.com

Date: 2018-07-25 - Engine version: 2.5.1.0

Active Directory Indicators

Indicators

Domain Risk Level: 100 / 100

It is the maximum score of the 4 indicators and one score cannot be higher than 100. The lower the better

Stale Object : 46 /100 — 5 rules matched

It is about operations related to user or computer objects

Trusts : 100 /100 — 3 rules matched

It is about links between two Active Directories

Privileged Accounts : 45 /100 — 3 rules matched

It is about administrators of the Active Directory

Anomalies : 100 /100 — 9 rules matched

It is about specific security control points

Risk model

https://www.pingcastle.com/PingCastleFiles/ad_hc_test.mysmartlogon.com.html

Script Sentry

- https://github.com/techspence/ScriptSentry

- ScriptSentry finds misconfigured and dangerous logon scripts.

- Not in PSGallery, but easy to invoke:

```
# Run ScriptSentry and save output to a text file

IEX(Invoke-WebRequest
'https://raw.githubusercontent.com/techspence/ScriptSentry/main/I
nvoke-ScriptSentry.ps1')

Invoke-ScriptSentry | Out-File c:\temp\ScriptSentry.txt
```

Script Sentry - Output

```
########## Unsafe UNC folder permissions ##########


Type                       File                                 User     Rights

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

UnsafeUNCFolderPermission  \\eureka-dc01\fileshare1             Everyone FullControl

UnsafeUNCFolderPermission  \\eureka-dc01\fileshare1\accounting  Everyone FullControl

UnsafeUNCFolderPermission  \\eureka-dc01\fileshare1\IT          Everyone FullControl
```

Script Sentry - Output

```
########## Unsafe logon script permissions ##########

Type                        File                                                   User
Rights
----                        ----                                                   ----
------
UnsafeLogonScriptPermission \\eureka.local\sysvol\eureka.local\scripts\elevate.vbs NT AUTHORITY\Authenticated Users
ReadAndExecute, Synchronize
UnsafeLogonScriptPermission \\eureka.local\sysvol\eureka.local\scripts\run.vbs     NT AUTHORITY\Authenticated Users
ReadAndExecute, Synchronize
UnsafeLogonScriptPermission \\eureka.local\sysvol\eureka.local\scripts\test.cmd    EUREKA\Domain Users
```

Script Sentry - Output

```
########## Unsafe GPO logon script permissions ##########

Type                           File                             User                                             Rights
----                           ----                             ----                                             ------
UnsafeGPOLogonScriptPermission \\eureka-dc01\fileshare1\run.bat EUREKA\testuser Write, ReadAndExecute, Synchronize
UnsafeGPOLogonScriptPermission \\eureka-dc01\fileshare1\run.bat Everyone
```

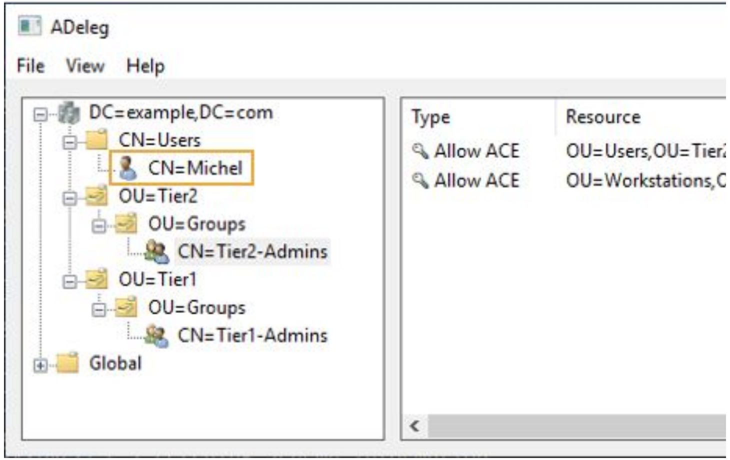
Script Sentry - Output

```
########## Plaintext credentials ##########



Type        File                                                     Credential

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

Credentials \\eureka.local\sysvol\eureka.local\scripts\ADCheck.ps1 $password = ConvertTo-SecureString -String
"Password2468!" -AsPlainText -Force

Credentials \\eureka.local\sysvol\eureka.local\scripts\shares.cmd  net use f: \\eureka-dc01\fileshare1\it
/user:itadmin Password2468!

Credentials \\eureka.local\sysvol\eureka.local\scripts\test.cmd    net use g: \\eureka-dc01\fileshare1
/user:user1 Password3355!
```

ADeleg

- https://github.com/mtth-bfft/adeleg

- An Active Directory delegation management tool. It allows you to make a detailed inventory of delegations set up so far in a forest, along with their potential issues:
  - Objects owned by users
  - Objects with ACEs (access control entries) for users
  - Non-canonical ACL
  - Disabled ACL inheritance
  - Default ACL modified in schema
  - Deleted delegation trustees



Type	Resource
Allow ACE	OU=Users,OU=Tier2
Allow ACE	OU=Workstations,C

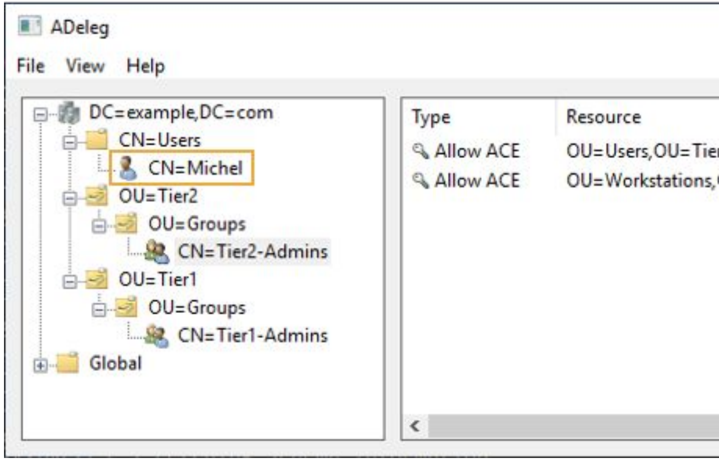
ADeleg - How it Works

This tool enumerates security descriptors of all objects, then filters out "expected" ACEs (access control entries):

- Inherited ACEs, since we are only interested in the original ACE upper in the tree;
- ACEs in the defaultSecurityDescriptor of the object class in the schema;
- Some special cases which need to be handled manually.

Special cases currently include:

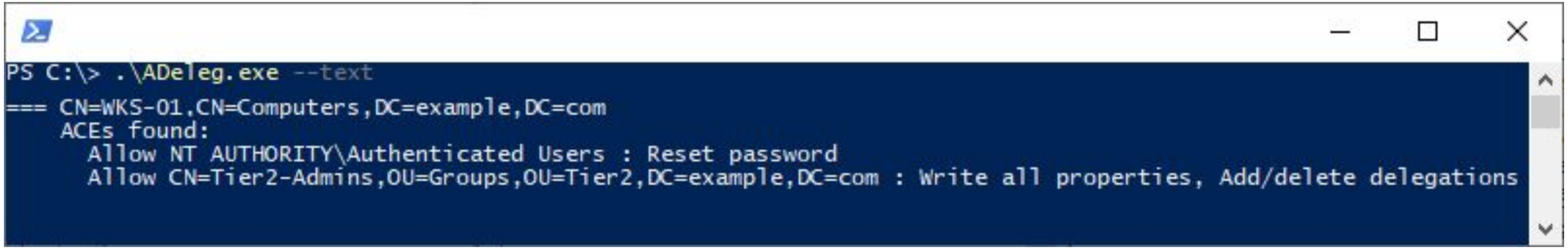
- object owners under a container with a CREATE_CHILD delegation
- ACEs for CREATOR_OWNER which are replaced and split in two in some cases during inheritance
- AdminSDHolder ACEs, for principals with adminCount set to 1
- KDS Root Keys, RODCs, ADCS, ADFS, Exchange, etc. are work in progress



Type	Resource
Allow ACE	OU=Users,OU=Tie
Allow ACE	OU=Workstations,(

ADeleg

You can use the tool from a terminal by passing any option to it (if you don't want to pass any particular option, just use `--text`):

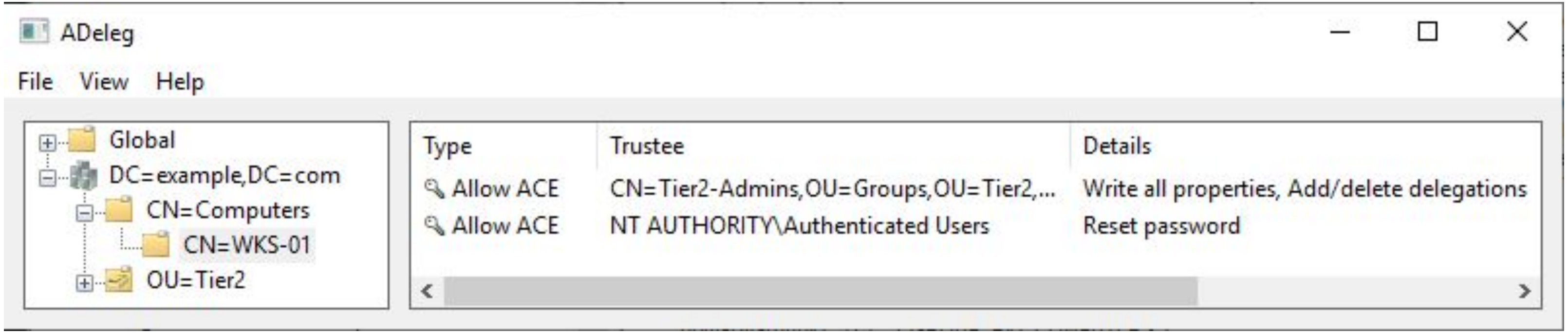


```
PS C:\> .\ADeleg.exe --text
=== CN=WKS-01,CN=Computers,DC=example,DC=com
    ACEs found:
      Allow NT AUTHORITY\Authenticated Users : Reset password
      Allow CN=Tier2-Admins,OU=Groups,OU=Tier2,DC=example,DC=com : Write all properties, Add/delete delegations
```

If you want to export results, you can choose a CSV output using --csv my.csv This is also suitable if you are interested in differences introduced since a previous dump (e.g. in PowerShell, diff (cat export_new.csv) (cat export_old.csv))

ADeleg

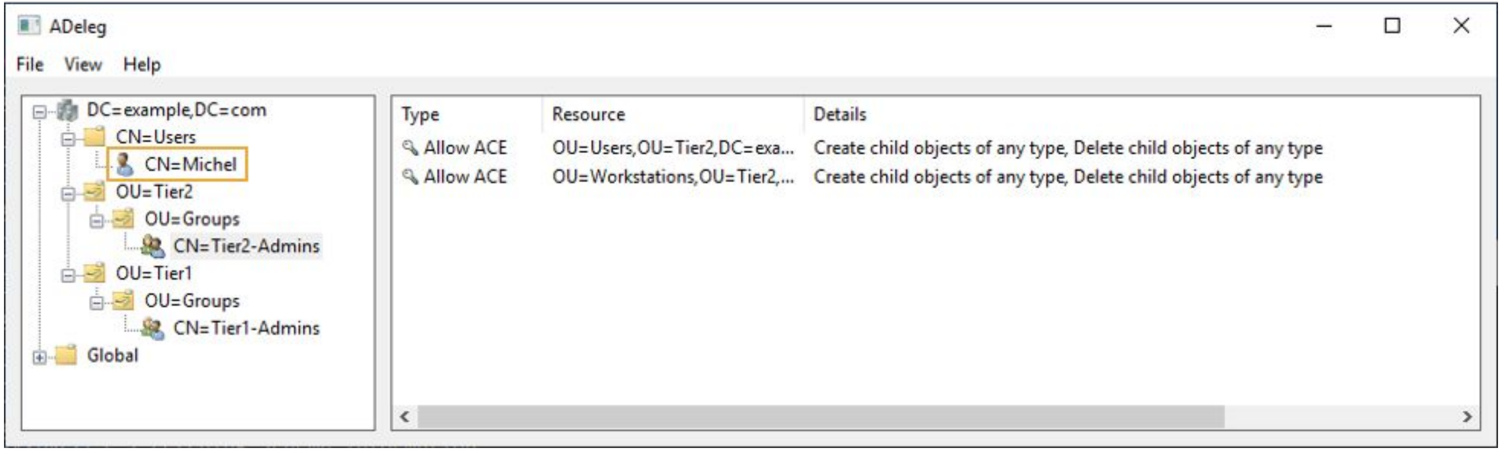
…but it does have a graphical user interface you can use.



Type	Trustee	Details
Allow ACE	CN=Tier2-Admins,OU=Groups,OU=Tier2,...	Write all properties, Add/delete delegations
Allow ACE	NT AUTHORITY\Authenticated Users	Reset password

ADeleg - How do I know if a result is important?

You should start reviewing delegations on your critical assets (domain controllers, domain admins, their admin workstations, servers with sensitive business data, etc.): **are these delegations needed for a user or service to do their work? Could they not work with fewer access rights, or on fewer objects?**



Type	Resource	Details
Allow ACE	OU=Users,OU=Tier2,DC=exa...	Create child objects of any type, Delete child objects of any type
Allow ACE	OU=Workstations,OU=Tier2,...	Create child objects of any type, Delete child objects of any type

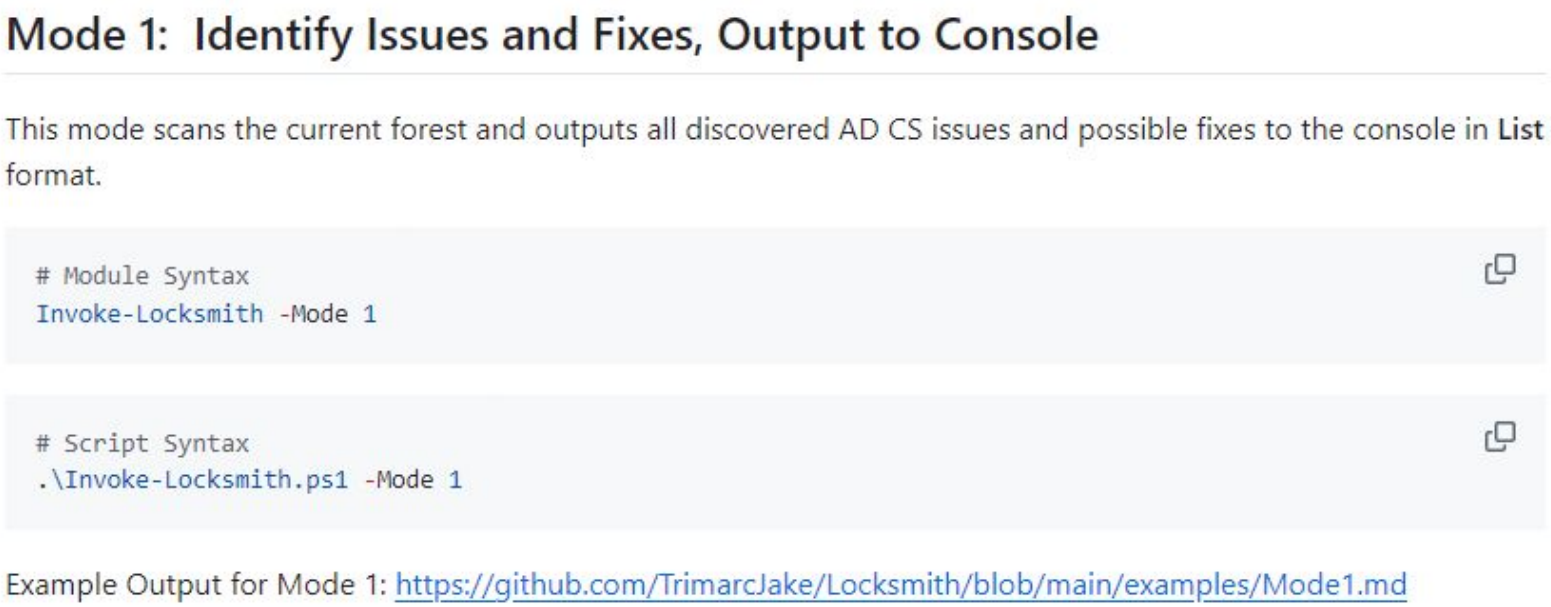
Locksmith

- Locksmith - https://github.com/TrimarcJake/Locksmith?tab=readme-ov-file
- A small tool built to find and fix common misconfigurations in Active Directory Certificate Services.

Locksmith

- Easy installation - Locksmith is in PSGallery
- `Install-Module -Name Locksmith -Scope CurrentUser`

Locksmith



Mode 1: Identify Issues and Fixes, Output to Console

This mode scans the current forest and outputs all discovered AD CS issues and possible fixes to the console in **List** format.

```
# Module Syntax
Invoke-Locksmith -Mode 1
```

```
# Script Syntax
.\Invoke-Locksmith.ps1 -Mode 1
```

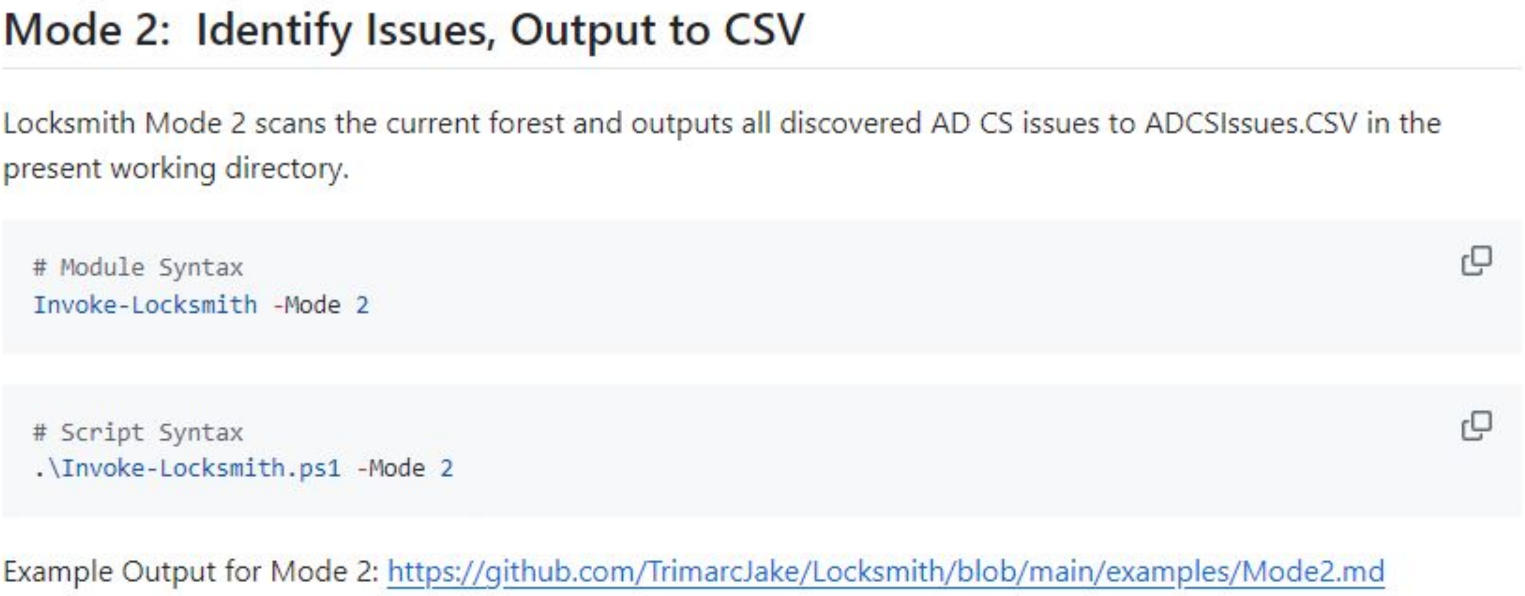
Example Output for Mode 1: https://github.com/TrimarcJake/Locksmith/blob/main/examples/Mode1.md

Locksmith

```
_       _____  _______ _     _ _______ _______ _____ _______ _     _
|      |     | |       |____/  |______ |  |  |   |      |    |_____|
|_____ |_____| |_____  |    \_ ______| |  |  | __|__    |    |     |
    .--.                  .--.                  .--.
   /.-. '----------.     /.-. '----------.     /.-. '----------.
   \'-' .---'-''-'-'     \'-' .--'--''-'-'     \'-' .--'--'-''-'
    '--'                  '--'                  '--'
```

```
########## ESC1 - Misconfigured Certificate Template ##########

Technique         : ESC1
Name              : ESC1-Vulnerable
DistinguishedName : CN=ESC1-Vulnerable,CN=Certificate Templates,CN=Public Key
                   Services,CN=Services,CN=Configuration,DC=horse,DC=local
Issue             : HORSE\kari can enroll in this Client Authentication template using a SAN without Manager
                   Approval
Fix               : Get-ADObject 'CN=ESC1-Vulnerable,CN=Certificate Templates,CN=Public Key
                   Services,CN=Services,CN=Configuration,DC=horse,DC=local' | Set-ADObject -Replace
                   @{'msPKI-Certificate-Name-Flag' = 0}
```

Locksmith



Mode 2: Identify Issues, Output to CSV

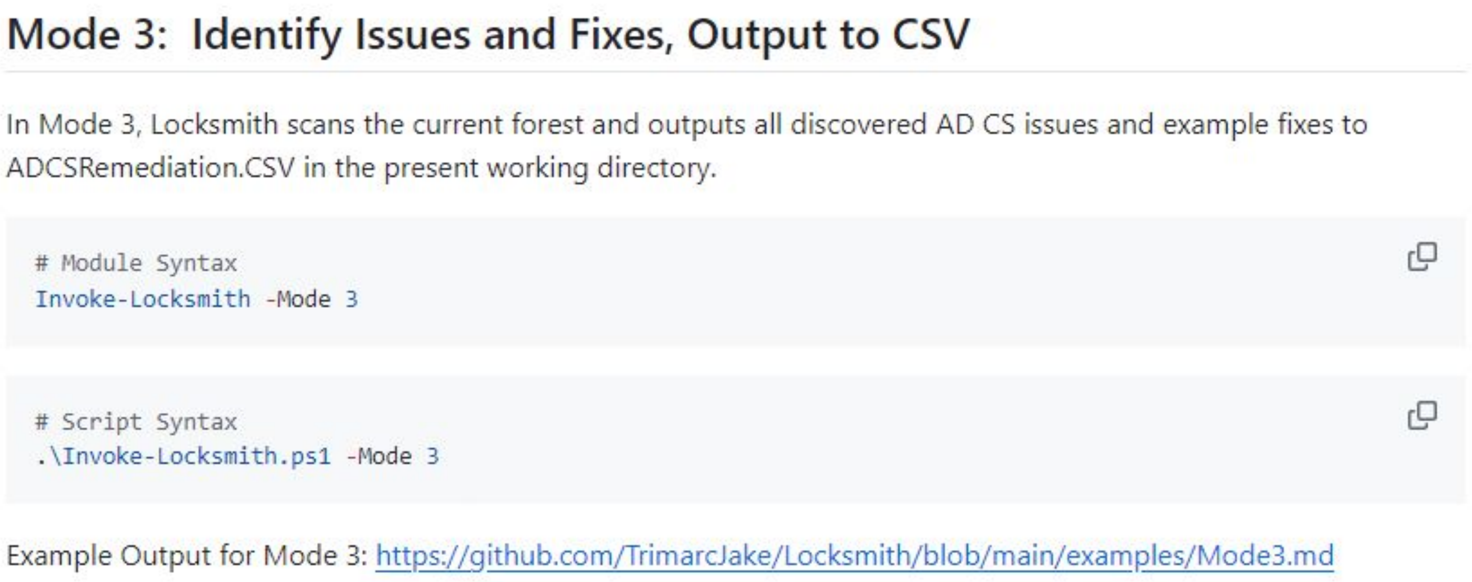
Locksmith Mode 2 scans the current forest and outputs all discovered AD CS issues to ADCSIssues.CSV in the present working directory.

```
# Module Syntax
Invoke-Locksmith -Mode 2
```

```
# Script Syntax
.\Invoke-Locksmith.ps1 -Mode 2
```

Example Output for Mode 2: https://github.com/TrimarcJake/Locksmith/blob/main/examples/Mode2.md

Locksmith



Mode 3: Identify Issues and Fixes, Output to CSV

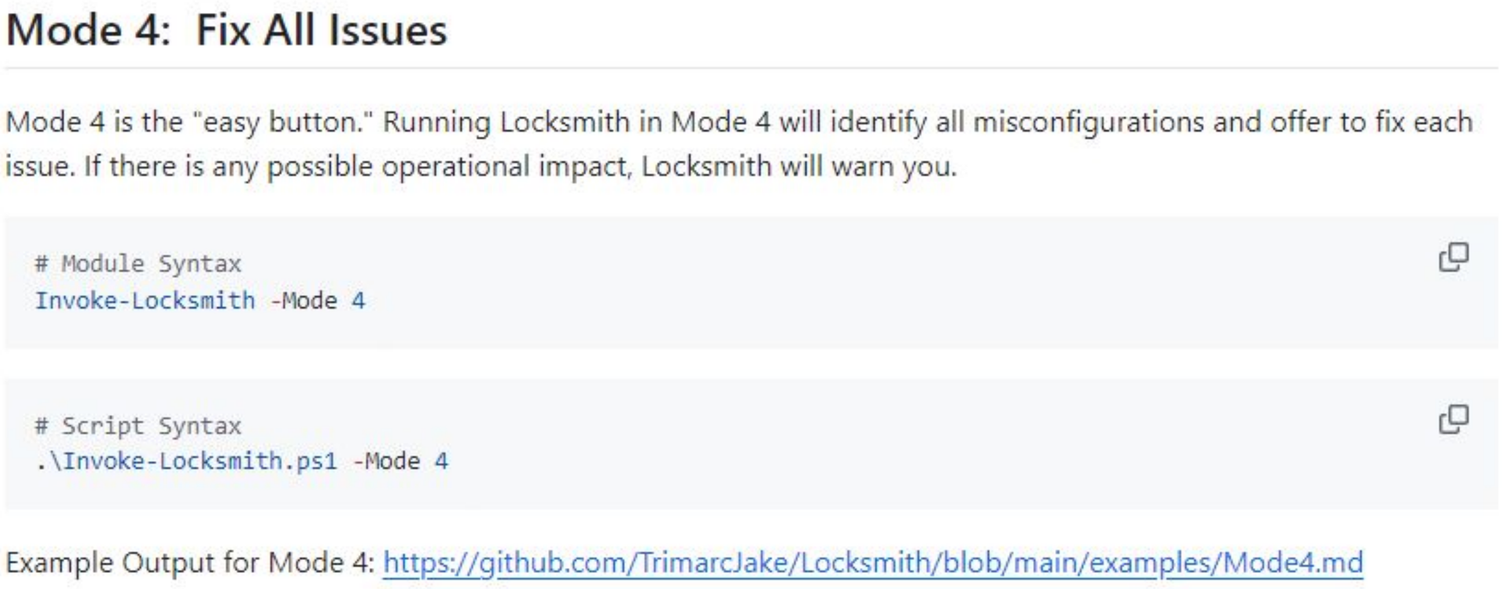
In Mode 3, Locksmith scans the current forest and outputs all discovered AD CS issues and example fixes to ADCSRemediation.CSV in the present working directory.

```
# Module Syntax
Invoke-Locksmith -Mode 3
```

```
# Script Syntax
.\Invoke-Locksmith.ps1 -Mode 3
```

Example Output for Mode 3: https://github.com/TrimarcJake/Locksmith/blob/main/examples/Mode3.md

Locksmith



Mode 4: Fix All Issues

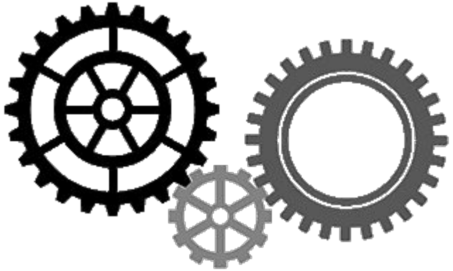
Mode 4 is the "easy button." Running Locksmith in Mode 4 will identify all misconfigurations and offer to fix each issue. If there is any possible operational impact, Locksmith will warn you.

```
# Module Syntax
Invoke-Locksmith -Mode 4
```

```
# Script Syntax
.\Invoke-Locksmith.ps1 -Mode 4
```

Example Output for Mode 4: https://github.com/TrimarcJake/Locksmith/blob/main/examples/Mode4.md

Step 2 - Configure



Storing Credentials and API Keys

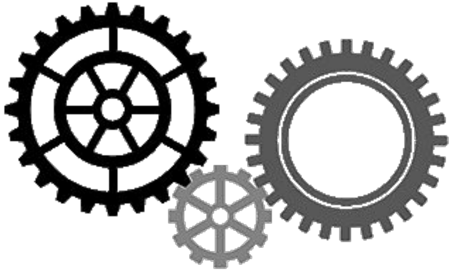
Domain Admins Audit: Generate health reports and identify privilege escalation paths, outdated policies, and risks related to ransomware attacks.

Disabled Users Audit: Audit delegation rights to find over-privileged accounts and unnecessary permissions.

Disabling SMB v1: Probably the biggest hole in ransomware.

KRBTGT Rotation: Rotating Kerberos Passwords

Step 2 - Configure



Restrict PowerShell Script Execution and Remoting

Disable Services and Protocols

Storing Credentials & Keys

- The PowerShell SecretManagement module provides a convenient way for a user to store and retrieve secrets.

- The SecretManagement module handles creating and configuring secrets, while the SecretStore module acts as the local secure vault for storing them.

Storing Credentials & Keys

```
Install-Module Microsoft.PowerShell.SecretManagement
```

```
Install-Module Microsoft.PowerShell.SecretStore
```

Storing Credentials & Keys

```
Install-Module Microsoft.PowerShell.SecretManagement
```

```
Install-Module Microsoft.PowerShell.SecretStore
```

Storing Credentials & Keys

To store credentials, you need to register a secret vault:

```
Register-SecretVault -Name PowerShellEngineer
-ModuleName Microsoft.PowerShell.SecretStore
-DefaultVault
```

Storing Credentials & Keys

To set a password on the Secret Store:

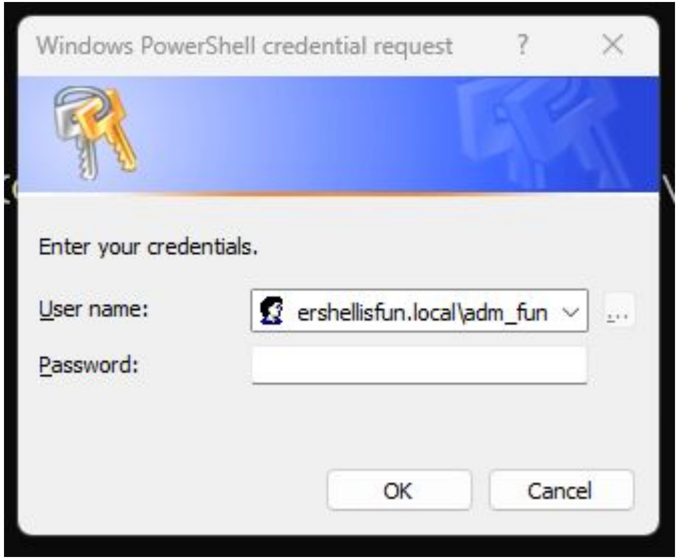
```
Get-SecretStoreConfiguration
```

Storing Credentials & Keys

To set a password on the Secret Store:

```
Set-Secret -Vault PowerShellEngineer -Name adm_ncs
-Secret (Get-Credential
PowerShellEngineer.local\adm_ncs) -Metadata
@{Description = "Admin account
PowerShellEngineer.local"}
```

Storing Credentials & Keys



Windows PowerShell credential request

Enter your credentials.

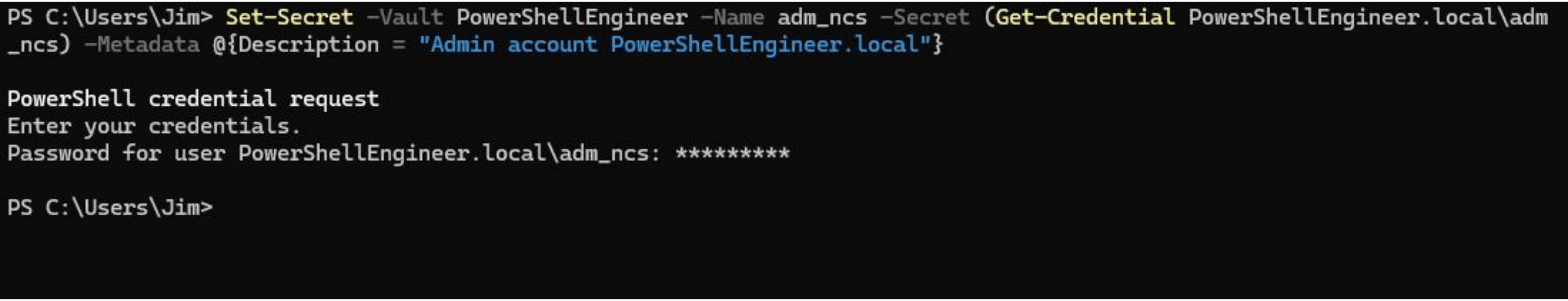
User name: ershellisfun.local\adm_fun

Password:

PowerShell v5

Storing Credentials & Keys

PowerShell v7



```
PS C:\Users\Jim> Set-Secret -Vault PowerShellEngineer -Name adm_ncs -Secret (Get-Credential PowerShellEngineer.local\adm
_ncs) -Metadata @{Description = "Admin account PowerShellEngineer.local"}

PowerShell credential request
Enter your credentials.
Password for user PowerShellEngineer.local\adm_ncs: *********

PS C:\Users\Jim>
```

Storing Credentials & Keys

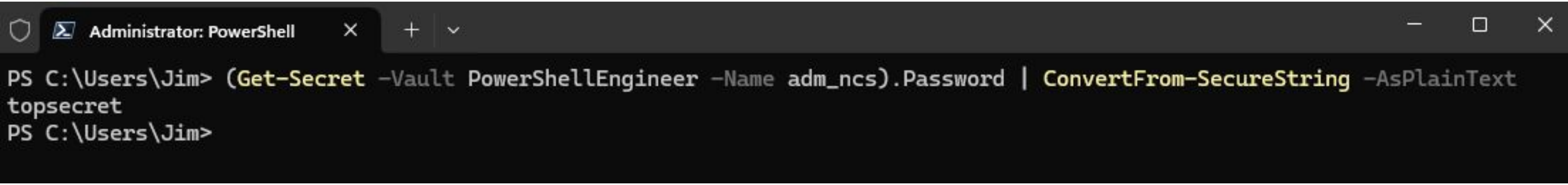
View a password:

```
(Get-Secret -Vault PowerShellEngineer -Name
adm_ncs).Password | ConvertFrom-SecureString
-AsPlainText
```

Storing Credentials & Keys

View a password:

```
(Get-Secret -Vault PowerShellEngineer -Name
adm_ncs).Password | ConvertFrom-SecureString
-AsPlainText
```



```
PS C:\Users\Jim> (Get-Secret -Vault PowerShellEngineer -Name adm_ncs).Password | ConvertFrom-SecureString -AsPlainText
topsecret
PS C:\Users\Jim>
```

Storing Credentials & Keys

Passing the credential safely to a command:

```
-credential (Get-Secret -Vault PowerShellEngineer
-Name adm_ncs)
```

Storing Credentials & Keys

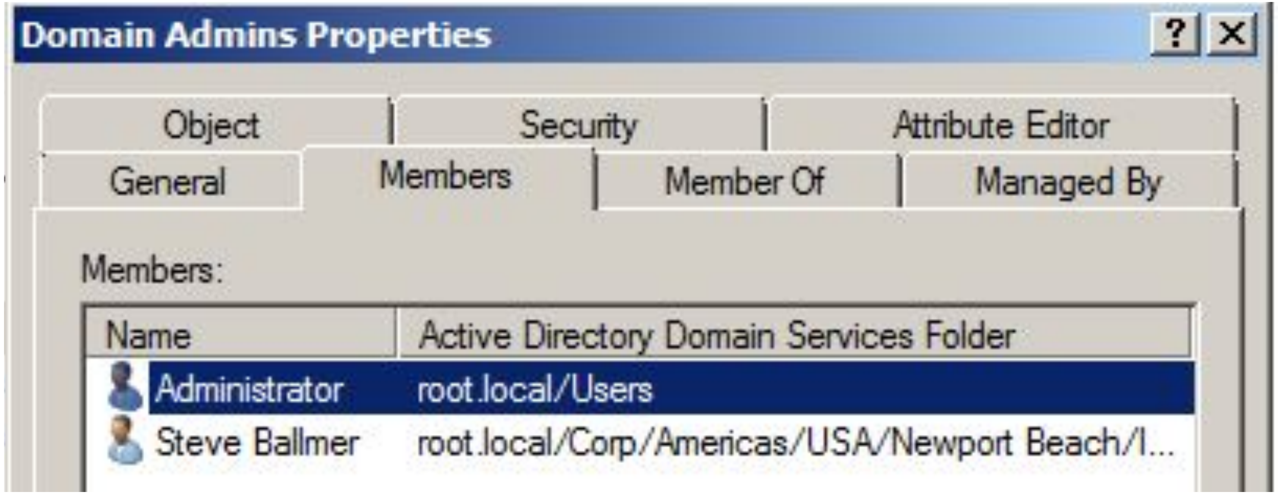
Passing the credential safely to a command:

```
Invoke-Command -ComputerName DC01 -ScriptBlock
{Restart-Service wuauserv -Force:$true
-Confirm:$false} -Credential (Get-Secret -Vault
PowerShellEngineer -Name adm_ncs)
```

Storing Credentials & Keys

```
Invoke-Command -ComputerName DC15 -ScriptBlock
{Restart-Service wuauserv -Force:$true
-Confirm:$false} -Credential (Get-Secret -Vault
PowerShellEngineer -Name adm_ncs)
```

Domain Admins Group



Domain Admins Properties

Members:

Name	Active Directory Domain Services Folder
Administrator	root.local/Users
Steve Ballmer	root.local/Corp/Americas/USA/Newport Beach/I...

Domain Admins Group

Domain Admins Group

The Domain Admins group is a built-in security group in Active Directory (AD) that has wide-ranging administrative privileges within a domain.

Domain Admins Group: Privileges

Members of the Domain Admins group have full control over all domain resources. They can manage user accounts, create and delete objects, modify group memberships, and perform other administrative tasks.

Domain Admins Group: Members

By default, the Administrator account is a member of the Domain Admins group. Additional users or groups can be added as needed, but this should be done sparingly due to the high level of access provided.

Domain Admins Group: Defaults

The Domain Admins group is part of the Administrators group on all domain-joined computers by default, giving its members local administrative rights on these machines.

Domain Admins Group: Defaults

Because of their extensive privileges, members of the Domain Admins group are high-value targets for attackers. Compromise of a Domain Admin account can lead to a full domain compromise, making it crucial to secure and monitor these accounts diligently.

LDAP Service Accounts Should NOT be Domain Admins

The typical use case for an LDAP account does not require write permissions. LDAP service accounts only need to be able to read users and group permissions in a directory.

KRBTGT Password Rotation

- The most important point of this process is that the Kerberos Ticket Granting Tickets (TGT) is encrypted and signed by the KRBTGT account. This means that anyone can create a valid Kerberos TGT if they have the KRBTGT password hash. Furthermore, despite the Active Directory domain policy for Kerberos ticket lifetime, the KDC trusts the TGT, so the custom ticket can include a custom ticket lifetime (even one that exceeds the domain kerberos policy).

- Prevents Golden Ticket attacks
- The password for the krbtgt account **should be rotated at least twice a year.**
- Script: https://github.com/microsoftarchive/New-KrbtgtKeys.ps1

Restrict PowerShell Script Execution and Remoting

- WinRM allows remote management using WS-Management. If not required, it should be disabled to reduce potential remote exploitation.

```
Disable-PSRemoting -Force
```

- PowerShell uses an execution policy to control how scripts can be executed. By default, the execution policy might be set to restricted, but if not, you can enforce this with this command:

```
Set-ExecutionPolicy Restricted -Force
```

Disable SMB v1

SMB v1 is an outdated protocol with several known vulnerabilities, including those exploited by ransomware like WannaCry. You can disable it using PowerShell with these commands:

```
Set-SmbServerConfiguration -EnableSMB1Protocol $false

Disable-WindowsOptionalFeature -Online -FeatureName
smb1protocol
```

Disable RDP (or Secure It)

RDP can be a major attack vector if not properly secured. If RDP is not needed, disable it. If needed, restrict access, use Network Level Authentication (NLA), and enable encryption.

```
Set-ItemProperty -Path
'HKLM:\System\CurrentControlSet\Control\Terminal Server\'
-Name "fDenyTSConnections" -Value 1
```

Disable RDP (or Secure It)

Secure RDP: If you must use RDP, ensure that: NLA is enabled, RDP is limited to necessary IP addresses, and that RDP sessions are monitored and logged.

To enable Network Level Authentication (NLA) for Remote Desktop Protocol (RDP) on Windows, you can do the following:

- Open the Control Panel
- Select System and Security
- Click Allow Remote Access
- In the Remote tab, check the box labeled Allow remote connections only from computers running Remote Desktop with Network Level Authentication

Disable Link-Local Multicast Name Resolution (LLMNR)

LLMNR is used for name resolution when DNS is unavailable, but it can be exploited in man-in-the-middle attacks.

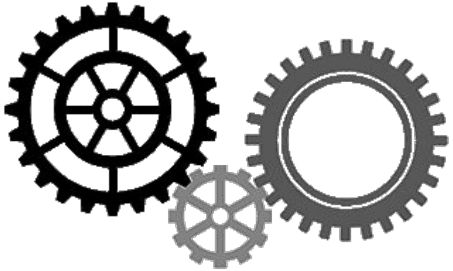
```
Set-ItemProperty -Path
"HKLM:\Software\Policies\Microsoft\Windows NT\DNSClient"
-Name "EnableMulticast" -Value 0
```

Disable ICMP

While disabling ICMP is not always recommended (since it helps in troubleshooting), blocking ICMP echo requests (pings) can reduce exposure to reconnaissance attacks.

```
New-NetFirewallRule -DisplayName "Block ICMPv4-In" -Protocol
ICMPv4 -IcmpType 8 -Action Block
```

Step 3 - Test



Using Kali Linux, there are three critical tools you can use to conduct penetration testing on your environment.

- Legion
- Mimikatz
- Hashcat / CrackMapExec

Step 3 - Test

Penetration testing, also known as ethical hacking, is a vital component of modern cybersecurity. It involves simulating real-world attacks on your network to identify vulnerabilities before malicious actors can exploit them. While this proactive approach is essential for safeguarding your infrastructure, it is crucial to obtain formal approval before conducting any penetration test on your network.

Step 3 - Before you test...

Obtain Formal Approval: Secure written authorization from senior management or legal teams before proceeding with any form of penetration testing.

Define the Scope and Objectives: Clearly outline what systems and areas will be tested, as well as the goals of the testing, to prevent unintended consequences.

Step 3 - Before you test...

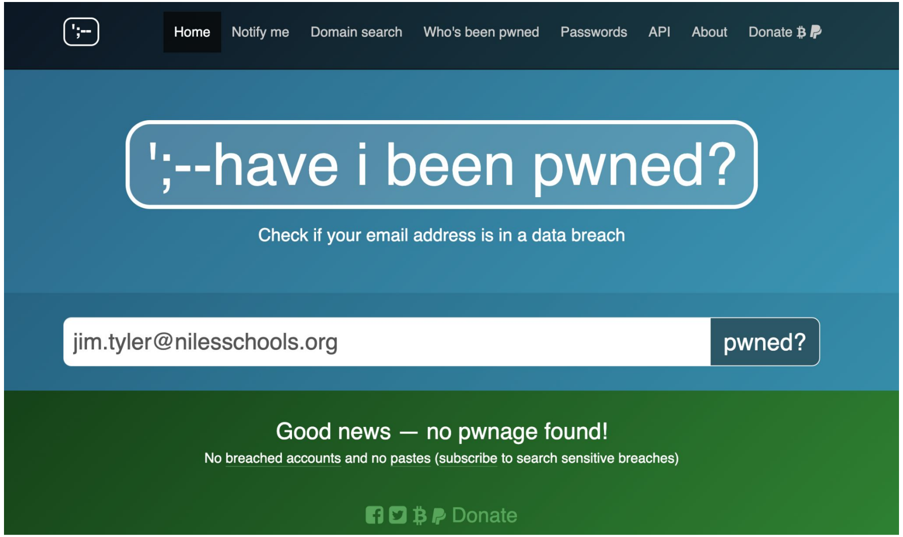
Coordinate with Security Teams: Ensure your security and IT teams are aware of the testing to avoid confusion and false alarms during the test window.

Use Qualified Professionals: Only allow certified and experienced professionals to conduct penetration tests to minimize the risk of mistakes or unintended disruptions.

Step 3 - Before you test...

Plan for Recovery: Prepare contingency plans for any service disruptions, and ensure that system settings are properly restored after testing is complete.

Have I been pwned?



Check if your email address is in a data breach

jim.tyler@nilesschools.org

Good news — no pwnage found!

No breached accounts and no pastes (subscribe to search sensitive breaches)

Checks to see if an email has been in a data breach. Useful for explaining why pentesting needs to be done, as these breaches are used for password spraying by threat actors.

https://www.haveibeenpwned.com

Kali Linux

Kali Linux is an open-source, Debian-based Linux distribution geared towards various information security tasks, such as Penetration Testing, Security Research, Computer Forensics and Reverse Engineering.

https://kali.org

Kali Linux

Virtual machine templates available:

https://www.kali.org/get-kali/#kali-virtual-machines

Including: VMWare, Hyper-V, Virtual Box

Legion

Description: Legion is a powerful GUI-based tool for network scanning and exploitation.

Features:

- Easy to set up and run
- Acts as a multitool for testing/scanning networks (similar to Nmap)
- Automates tasks like SMB enumeration and service discovery
- Can quickly identify vulnerable services in AD environments

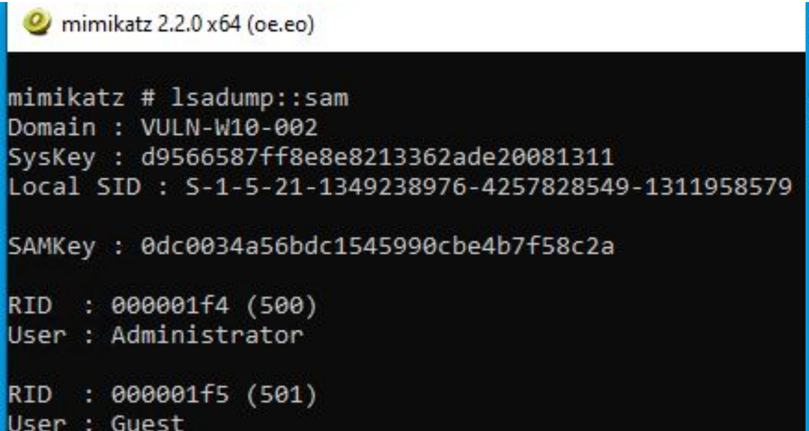
Legion

Steps:

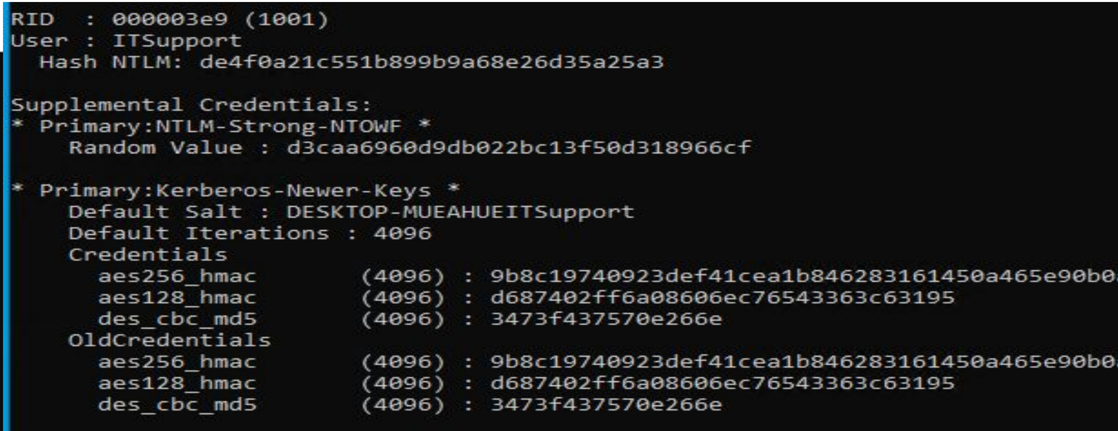
1. Launch Legion
2. Scan network ranges to identify AD services
3. Use integrated tools (SMB enumeration, DNS zone transfers, etc.)
4. Analyze output to identify potential AD entry points

Mimikatz

Mimikatz is used to extract credentials from Windows systems, including plaintext passwords and NTLM hashes.



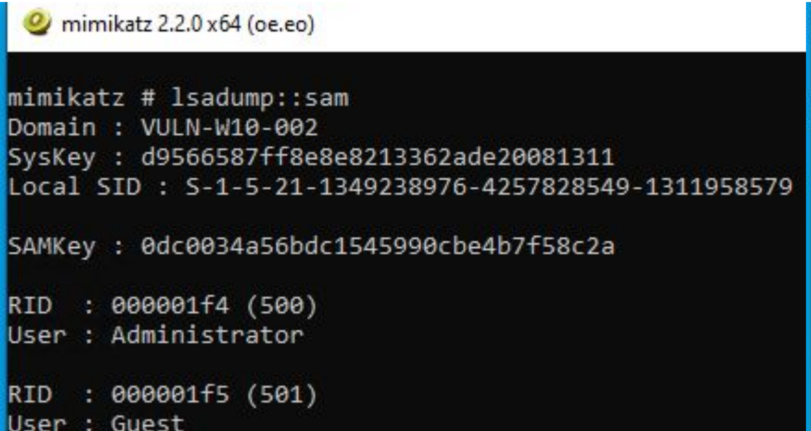
```
mimikatz # lsadump::sam
Domain : VULN-W10-002
SysKey : d9566587ff8e8e8213362ade20081311
Local SID : S-1-5-21-1349238976-4257828549-1311958579

SAMKey : 0dc0034a56bdc1545990cbe4b7f58c2a

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest
```



```
RID  : 000003e9 (1001)
User : ITSupport
  Hash NTLM: de4f0a21c551b899b9a68e26d35a25a3

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : d3caa6960d9db022bc13f50d318966cf

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-MUEAHUEITSupport
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 9b8c19740923def41cea1b846283161450a465e90b0
      aes128_hmac       (4096) : d687402ff6a08606ec76543363c63195
      des_cbc_md5       (4096) : 3473f437570e266e
    OldCredentials
      aes256_hmac       (4096) : 9b8c19740923def41cea1b846283161450a465e90b0
      aes128_hmac       (4096) : d687402ff6a08606ec76543363c63195
      des_cbc_md5       (4096) : 3473f437570e266e
```

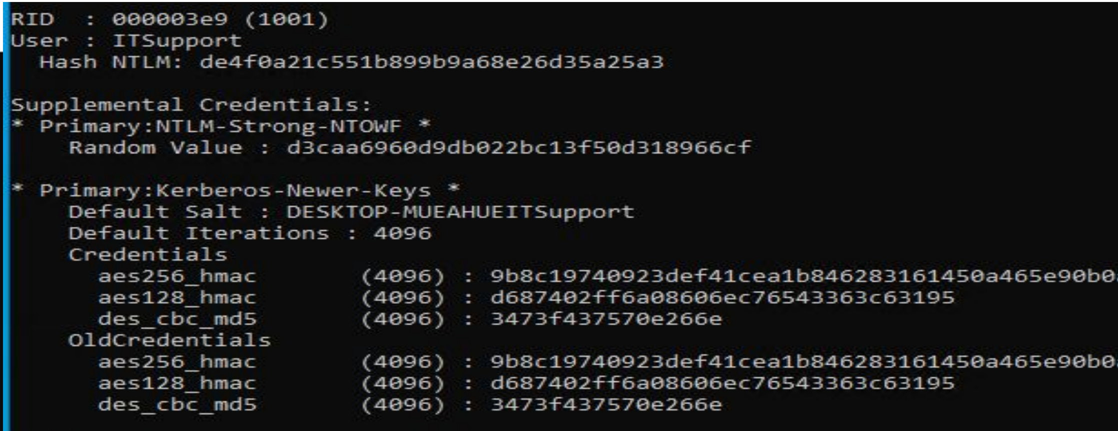
Mimikatz

Pass-the-Hash (PTH) Attack Leverages NTLM hashes to authenticate without cracking the password



mimikatz 2.2.0 x64 (oe.eo)

```
mimikatz # lsadump::sam
Domain : VULN-W10-002
SysKey : d9566587ff8e8e8213362ade20081311
Local SID : S-1-5-21-1349238976-4257828549-1311958579

SAMKey : 0dc0034a56bdc1545990cbe4b7f58c2a

RID  : 000001f4 (500)
User : Administrator

RID  : 000001f5 (501)
User : Guest
```



```
RID  : 000003e9 (1001)
User : ITSupport
  Hash NTLM: de4f0a21c551b899b9a68e26d35a25a3

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
    Random Value : d3caa6960d9db022bc13f50d318966cf

* Primary:Kerberos-Newer-Keys *
    Default Salt : DESKTOP-MUEAHUEITSupport
    Default Iterations : 4096
    Credentials
      aes256_hmac       (4096) : 9b8c19740923def41cea1b846283161450a465e90b0
      aes128_hmac       (4096) : d687402ff6a08606ec76543363c63195
      des_cbc_md5       (4096) : 3473f437570e266e
    OldCredentials
      aes256_hmac       (4096) : 9b8c19740923def41cea1b846283161450a465e90b0
      aes128_hmac       (4096) : d687402ff6a08606ec76543363c63195
      des_cbc_md5       (4096) : 3473f437570e266e
```

Hashcat

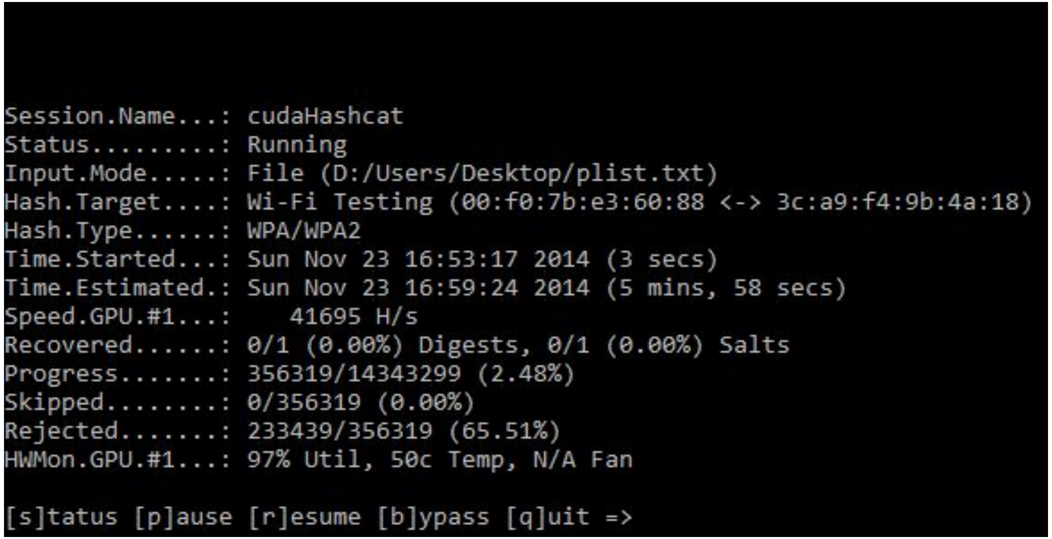
Hashcat is a powerful password-cracking tool that can crack NTLM hashes offline.

Works on Windows, Mac, and Linux.

https://hashcat.net/hashcat/

Hashcat

hashcat -m 1000 ntlm_hashes.txt wordlist.txt



```
Session.Name...: cudaHashcat
Status.........: Running
Input.Mode.....: File (D:/Users/Desktop/plist.txt)
Hash.Target....: Wi-Fi Testing (00:f0:7b:e3:60:88 <-> 3c:a9:f4:9b:4a:18)
Hash.Type......: WPA/WPA2
Time.Started...: Sun Nov 23 16:53:17 2014 (3 secs)
Time.Estimated.: Sun Nov 23 16:59:24 2014 (5 mins, 58 secs)
Speed.GPU.#1...:    41695 H/s
Recovered......: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.......: 356319/14343299 (2.48%)
Skipped........: 0/356319 (0.00%)
Rejected.......: 233439/356319 (65.51%)
HWMon.GPU.#1...: 97% Util, 50c Temp, N/A Fan

[s]tatus [p]ause [r]esume [b]ypass [q]uit =>
```

Source: JumpSec Labs

CrackMapExec

CrackMapExec (CME) is a post-exploitation tool targeting AD environments.

- Enumerate users, groups, and computers
- Test credentials across the network
- Perform SMB and LDAP attacks

https://github.com/byt3bl33d3r/CrackMapExec

CrackMapExec

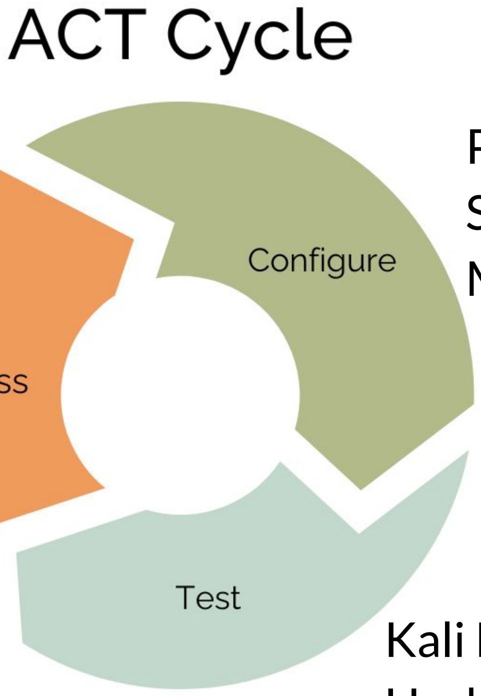
CrackMapExec (CME) is a post-exploitation tool targeting AD environments.

```
crackmapexec smb 10.0.0.1 -u admin -p
password123 --shares
```

https://github.com/byt3bl33d3r/CrackMapExec

Security is Never a Destination

We need to follow a cyclical approach to information security.



PowerShell, Remove/Disable Services & Protocols, Minimize Privileged Accounts

Ping Castle, Script Sentry, ADeleg, Locksmith

Kali Linux, Legion, Mimikatz, Hashcat, CrackMapExec

Auf Wiedersehen

Github:
https://github.com/jimrtyler/maeds

YouTube:
https://youtube.com/@PowerShellEngineer

LinkedIn:
https://linkedin.com/in/jamestyler

PowerShell Course based on my book, *PowerShell for Systems Engineers*:
https://www.udemy.com/course/powershell-for-systems-engineers/?couponCode=MAEDS2024

Free to all MAEDS
Attendees. Code: MAEDS2024