U.S. DEPARTMENT OF ENERGY'S
CYBERFORCE® PROGRAM

CyberForce® 101
Windows Hardening

October 2023

cyberforcecompetition@anl.gov

Windows Hardening 101

Pre-Requisites

- **Intro to PowerShell**
- **Intro to Windows**
- **Useful Protocols**
- **Typical Services**

Managing Users and Groups

Getting the list of users should always be one of the first steps so that you can verify if there are any unknown accounts that need to be removed from the local system. You can do that in PowerShell with the cmdlet `Get-LocalUser`.

```
It's also important to be able to disable and remove users.

```powershell
# disable a user
Disable-LocalUser -Name "Guest"
# remove a user
Remove-LocalUser -Name "Test"
```

Groups can be handled very similarly to users. You want to start by verifying what groups exist with the cmdlet `Get-LocalGroup`. You also want to see what users are linked to that group, which can be found with the cmdlet `Get-LocalGroupMember` with the group name.

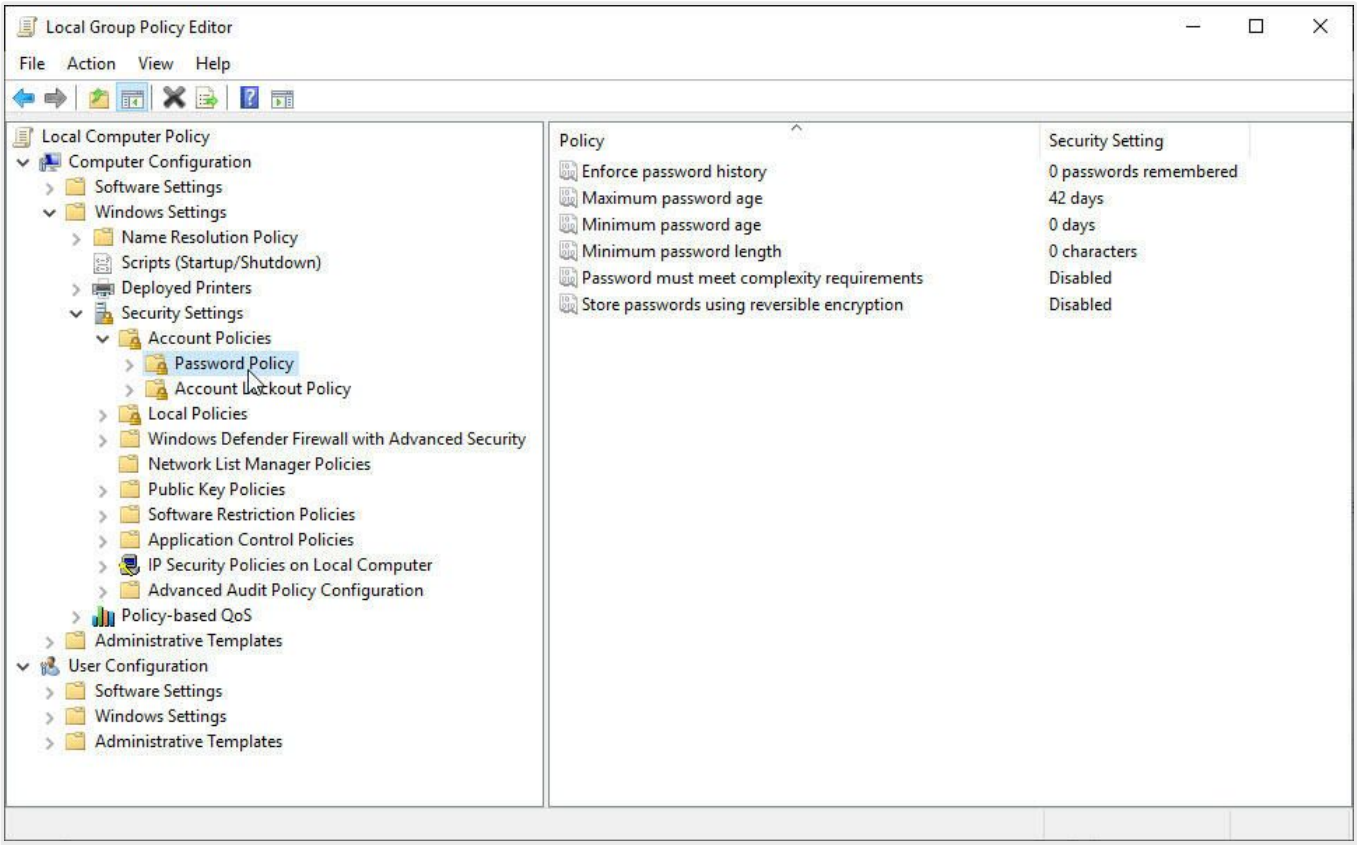
```
# add user to group
Add-LocalGroupMember -Group "Administrators" -Member Test
# remove user from group
Remove-LocalGroupMember -Group "Administrators" -Member Test
# remove group
Remove-LocalGroup -Name "Bad"
```

Passwords
```

Resetting a user's password is another important command. A lot of times users can have weak passwords if the Local Security Policy settings are not configured properly.

```
# creates password variable
$Password = Read-Host -AsSecureString
-> [ENTER PASSWORD HERE]
# create a user variable
$UserAccount = Get-LocalUser -Name Test
# pipe the user variable in to a password change
$UserAccount | Set-LocalUser -Password $Password
```

To modify password policies, open Run and type `secpol.msc` and hit enter. In the left pane, click on `Account Policies > Password Policy`. In the right pane, you will see settings for configuring the Password Policy. Here you can set policies for password history, maximum password age, minimum password age, minimum password length, minimum password length audits, and complexity requirements.



From TechRepublic

Enabling Firewalls and Antivirus

You can get the information on the current rule set for the Domain, Public, and Private firewall profiles by using the cmdlet `Get-NetFirewallProfile`. You can also enable all these in PowerShell.

```
# enable all profiles
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
# enable logging on all profiles
Set-NetFirewallProfile -All -LogAllowed 1
```

You also should check the status of Windows Defender to see if it's been disabled or removed from the system. If there are features like AntivirusEnabled that are set to False, these need to be changed to True.

```
# show status of defender
Get-MpComputerStatus
```

We also need to update the Anti-Malware signatures to allow for better detection and protection of the system. We can do this by using the cmdlet `Update-MpSignature`.

To start a scan and see if there is any malware that can be found through the Defender's signature database, we can use the cmdlet `Start-MpScan` with the Parameter `-ScanType FullScan`. The cmdlet `Get-MpThreat` lists previous threats, and the cmdlet `Get-MpThreatDetection` lists active and past threats. If Windows Defender has been removed or isn't working, reinstall it with the command below.

```
Get-AppxPackage Microsoft.SecHealthUI -AllUsers | Reset-AppxPackage
```

If you find a threat, you can use the cmdlet `Remove-MpThreat` to get rid of it.

Running Malware Scans

Running malware scans are important to ensure there isn't any malware that has been downloaded onto the systems. Tools like Malwarebytes are useful for this.

You should first run the scans and then if anything is found, make sure to quarantine them and remove them completely.

Remove Malicious Scheduled Tasks

Within Administrator PowerShell, you should list suspicious scheduled tasks and remove them accordingly. Malicious tasks can be used to hide malware and should be checked to ensure your system doesn't have any.

```
# Set User Env Variable
$User = "DOE"

# List Scheduled Tasks in Ready/Running State and Output to File
```

```
Get-ScheduledTask | Where State -in "Ready","Running" | Set-Content -Path
C:\Users\$User\Desktop\Connections.txt

# Enable a Task (Good for Update or Windows Defender)
Enable-ScheduledTask -TaskName "<NAME>"

# Enable ALl Tasks in a Folder
Get-ScheduledTask -TaskPath "\WindowsDefender\" | Enable-ScheduledTask

# Disable a Task (Disable Any Malicious Task)
Disable-ScheduledTask -TaskName

# Disable ALl Tasks in a Folder
Get-ScheduledTask -TaskPath "\WindowsDefender\" | Disable-ScheduledTask

# Unregister or Remove a Scheduled Task
Unregister-ScheduledTask -TaskName "<NAME>"

# Stop a Scheduled Task
Stop-ScheduledTask -TaskName "<NAME>"

# Stop All Tasks in a Folder
Get-ScheduledTask -TaskPath "<PATH>" | Stop-ScheduledTask
```

Check for Malicious Shares

Valid Accounts can be used to interact with a remote network share using Server Message Block (SMB). Then, an attacker can perform actions as the logged-on user. Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. This can be used with Valid Accounts to remotely access a networked system over SMB.

Therefore, you should check for any malicious shares on your system.

```
# List SMB Shares
Get-SmbShare

# Remove a SMB Share
Remove-SmbShare -Name <NAME>

# Check if SMBv1 is Enabled (BAD IF ENABLED)
Get-SmbServerConfiguration | Select EnableSMB1Protocol

# Disable SMBv1 if Enabled
Set-SmbServerConfiguration -EnableSMB1Protocol 0
```

```
# Check if SMBv2 is Enabled (replacement for SMBv1)
Get-SmbServerConfiguration | Select EnableSMB2Protocol

# Enable SMBv2 if Disabled
Set-SmbServerConfiguration -EnableSMB2Protocol 1
```

Remove Malicious Connections

In PowerShell, we want to check established/listening ports on the system and remove any suspicious outliers we find.

```
# Base Command
Get-NetTcpConnection

# List Listening and Established
Get-NetTcpConnection -State Listen,Established

# List Listening and Established and Sort Remote Port Least to Greatest
Get-NetTcpConnection -State Listen,Established | Sort-Object RemotePort

# List All Property Details of Connections
Get-NetTcpConnection -State Listen,Established | Select-Object -Property *

# List Specific Property Details of Connections
Get-NetTcpConnection -State Listen,Established | Select-Object -Property
State,CreationTime,OwningProcess,Local*,Remote*
```

Checking Services

If we find any malicious services, we want to stop them from running on the system.

```
# List Services
Get-Service
# Find Service Information
Get-Service -Name <NAME> | Select-Object -Property *
# Stop Service (e.g., get rid of print "Spooler")
Stop-Service -Name <NAME>
# Change Service Startup Type to Disabled
Set-Service <NAME> -StartupType Disabled
```

Check Processes

We want to check for any malicious processes and kill them if found.

```
# List Processes
Get-Process
# Find Process Information
Get-Service -Name <NAME> | Select-Object -Property *
# Stop the Process
Stop-Process -Name <NAME>
# OR
Stop-Process -Id <INTEGER_ID>
```

Check for Malicious Files

As you navigate through the system, ensure that there are no suspicious files. In particular, you want to check the Temp folder for any. File names containing hashes and random things will be there because that is where temporary files are stored.

If you find anything suspicious, you want to make sure you remove it.

```
Delete a File
Remove-Item -Path <PATH> -Force
# Recursively Remove Files
Remove-Item -Path <PATH> -Recurse
# Remove Files with Special Characters (Get-ChildItem = ls) (` = ESC Char)
Get-ChildItem | Where-Object Name -Like '*`[*' | ForEach-Object {Remove-Item -
LiteralPath $_.Name}
```

Uninstall Unnecessary Software

Sometimes you will find unnecessary software that shouldn't be on the system. This can be malicious or just not needed but can lead to further security issues.

First, you should get a list of the apps installed.

```
Get-AppxPackage | select Name,PackageFullName | Format-List
```

To remove the app, use the command below.

```
# for current account
Remove-AppxPackage [AppName]
# for all users
Remove-AppxPackage -allusers [AppName]
```

Update the System

Security patches need to be applied to ensure that your system isn't vulnerable. You can update the system by checking in the settings.

Sources

1. How to manage Microsoft Defender Antivirus with PowerShell
2. How to manage your users' Windows passwords with Group Policy
3. Remote Services: SMB/Windows Admin Shares
4. How to uninstall Windows apps with PowerShell