

Week 4 – Windows Server Basics (Simulated using PowerShell on ARM Laptop)

Omprakash Nara

Objective:

The primary goal of Week 4 was to gain hands-on experience with Windows Server features such as Active Directory Domain Services (AD DS), user and group management, and security policies like password rules. However, due to hardware limitations on my ARM-based laptop (Snapdragon processor), it is not possible to run virtualization software like VirtualBox or VMware Workstation. Therefore, I simulated all key tasks using Windows PowerShell, which allowed me to complete the same user/group management functions and password policy configurations locally.

This simulation helped me understand how system administrators perform user account creation, group assignments, and enforce basic security policies — which are common tasks in managing enterprise environments.

Tasks Completed:

1. Creating Local Users:

Using PowerShell, I created two new local user accounts named “Alice” and “Bob.” These users were configured with passwords that meet typical enterprise standards (minimum 8 characters, using a mix of characters and numbers).

The exact command used was:

```
net user Alice "P@ssword123!" /add  
net user Bob "P@ssword123!" /add
```

This step simulates the kind of user provisioning an admin would do inside a domain environment or workgroup.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> whoami /groups | findstr /i "S-1-16-12288"
Mandatory Label\High Mandatory Level                                Label                                S-1-16-12288

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>>
The command completed successfully.

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>> net user Bob "P@ssword123!" /add
>>
The account already exists.

More help is available by typing NET HELPMSG 2224.

The command completed successfully.

PS C:\Windows\system32>
```

2. Creating Local Groups:

Next, I created two groups called “HR” and “IT” to represent organizational departments. These groups help control permissions and organize user access in real-world enterprise setups.

Groups created:

- HR
- IT

Command used:

```
net localgroup HR /add
```

```
net localgroup IT /add
```

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

C:\Windows\system32> whoami /groups | findstr /i "S-1-16-12288"
Mandatory Label\High Mandatory Level                                Label                                S-1-16-12288

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>>
The command completed successfully.

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>> net user Bob "P@ssword123!" /add
>>
The account already exists.

More help is available by typing NET HELPMSG 2224.

The command completed successfully.

PS C:\Windows\system32> net localgroup HR /add
>> net localgroup IT /add
>>
The command completed successfully.

The command completed successfully.

PS C:\Windows\system32>
```

3. Assigning Users to Groups:

After creating both the users and groups, I assigned each user to their appropriate department. This helps simulate real-world identity and access management:

- Alice was added to the HR group
- Bob was added to the IT group

Command used:

```
net localgroup HR Alice /add
```

```
net localgroup IT Bob /add
```

This reflects how system engineers group users for role-based access control in Active Directory or local group policy systems.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows
    whoami /groups | findstr /i "S-1-16-12288"
>> C:\Windows\system32>
Mandatory Label\High Mandatory Level                                Label                                S-1-16-12288

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>>
The command completed successfully.

PS C:\Windows\system32> net user Alice "P@ssword123!" /add
>> net user Bob "P@ssword123!" /add
>>
The account already exists.

More help is available by typing NET HELPMSG 2224.

The command completed successfully.

PS C:\Windows\system32> net localgroup HR /add
>> net localgroup IT /add
>>
The command completed successfully.

The command completed successfully.

PS C:\Windows\system32> net localgroup HR Alice /add
>> net localgroup IT Bob /add
>>
The command completed successfully.

The command completed successfully.

PS C:\Windows\system32>
```

4. Exporting and Reviewing Password Policy Settings:

Since I couldn't access Group Policy Management or Active Directory tools, I simulated password policy enforcement by exporting the local system's password and account settings using the secdit command.

I ran:

```
secdit /export /cfg "C:\Users\ompra\OneDrive\Desktop\policies.txt" /log
"C:\Users\ompra\OneDrive\Desktop\export_log.txt"
```

Then opened the exported file using Notepad to locate:

- MinimumPasswordLength
- MaximumPasswordAge

These settings help enforce good security practices across user accounts, ensuring users create strong passwords and change them regularly.

MinimumPasswordLength:

Electricity details.txt • CCloud soft.txt • References.md • clients details • public class omprakash • policies.txt

File Edit View H1 ≡ B I ↵

MinimumPasswordLength

^ × 🔍 ↓ ↑ ⚙ ×

```
[Unicode]
Unicode=yes
[System Access]
MinimumPasswordAge = 0
MaximumPasswordAge = 42
MinimumPasswordLength = 0
PasswordComplexity = 0
PasswordHistorySize = 0
LockoutBadCount = 10
ResetLockoutCount = 10
LockoutDuration = 10
AllowAdministratorLockout = 1
RequireLogonToChangePassword = 0
ForceLogoffWhenHourExpire = 0
NewAdministratorName = "Administrator"
NewGuestName = "Guest"
ClearTextPassword = 0
LSAAnonymousNameLookup = 0
EnableAdminAccount = 0
EnableGuestAccount = 0
[Event Audit]
AuditSystemEvents = 0
AuditLogonEvents = 0
AuditObjectAccess = 0
AuditPrivilegeUse = 0
AuditPolicyChange = 0
AuditAccountManage = 0
AuditProcessTracking = 0
AuditDSAccess = 0
AuditAccountLogon = 0
[Registry Values]
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Setup\RecoveryConsole\SecurityLevel=4,0
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Setup\RecoveryConsole\SetCommand=4,0
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\CachedLogonsCount=1,"10"
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ForceUnlockLogon=4,0
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon>PasswordExpiryWarning=4,5
MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ScRemoveOption=1,"0"
MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorAdmin=4,5
MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\ConsentPromptBehaviorUser=1,0
```

In 6 Col 22 21 of 8 379 characters Plain text 100% Windows (CRIF)

MaximumPasswordAge:

The screenshot shows a Windows File Explorer window displaying the contents of a file named "policies.txt". The file contains a list of system configuration settings, including various registry paths and values related to security, user interface, and network settings. The search bar at the top of the window shows the text "MaximumPasswordAge".

File Edit View H1 ≡ B I ↺ A

Electricity details.txt • Cloud soft.txt • References.md • clients details • public class omprakash • policies.txt +

MACHINE\System\CurrentControlSet\Control\Lsa\RestrictAnonymous=4,0
MACHINE\System\CurrentCont
MACHINE\System\CurrentCont
MACHINE\System\CurrentCont
MaximumPasswordAge x Q ↓ ↑ ⇅ X ductOptions,System
CurrentControlSet\Control\Server Applications\Software\Microsoft\Windows NT\CurrentVersion
MACHINE\System\CurrentControlSet\Control\SecurityProviders\Smb\SecurePipeServers\Winreg\AllowedPaths\Machine=7,System\CurrentControlSet\Control\Print\Printers,System
CurrentControlSet\Services\Eventlog\Software\Microsoft\OLAP Server\Software\Microsoft\Windows NT\CurrentVersion\Print\Software\Microsoft\Windows NT
CurrentVersion\Windows\System\CurrentControlSet\Control\ContentIndex\System\CurrentControlSet\Control\Terminal Server\System\CurrentControlSet\Control
Terminal Server\UserConfig\System\CurrentControlSet\Control\Terminal Server\DefaultUserConfiguration\Software\Microsoft\Windows NT\CurrentVersion
\Perflib\System\CurrentControlSet\Services\SysmonLog
MACHINE\System\CurrentControlSet\Control\Session Manager\Kernel\ObCaseInsensitive=4,1
MACHINE\System\CurrentControlSet\Control\Session Manager\Memory Management\CleanPageFileAtShutdown=4,0
MACHINE\System\CurrentControlSet\Control\Session Manager\ProtectionMode=4,1
MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems\optional=7,
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\AutoDisconnect=4,15
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\EnableForcedLogOff=4,1
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\EnableSecuritySignature=4,0
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\NullSessionPipes=7,
MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\RestrictNullSessAccess=4,1
MACHINE\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\EnablePlainTextPassword=4,0
MACHINE\System\CurrentControlSet\Services\LanmanWorkstation\Parameters\EnableSecuritySignature=4,1
MACHINE\System\CurrentControlSet\Services\LDAP\LDAPClientConfidentiality=4,1
MACHINE\System\CurrentControlSet\Services\LDAP\LDAPClientIntegrity=4,1
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\DisablePasswordChange=4,0
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\MaximumPasswordAge=4,30
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\RequireSignOrSeal=4,1
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\RequireStrongKey=4,1
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\SealSecureChannel=4,1
MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\SignSecureChannel=4,1
[Privilege Rights]
SeNetworkLogonRight = *S-1-1-0,*S-1-5-32-544,*S-1-5-32-545,*S-1-5-32-551
SeBackupPrivilege = *S-1-5-32-544,*S-1-5-32-551
SeChangeNotifyPrivilege = *S-1-1-0,*S-1-5-19,*S-1-5-20,*S-1-5-32-544,*S-1-5-32-545,*S-1-5-32-551,
*S-1-5-99-216390572-1995538116-3857911515-2404958512-2623887229
SeSystemtimePrivilege = *S-1-5-19,*S-1-5-32-544
SeCreatePagefilePrivilege = *S-1-5-32-544
SeDebugPrivilege = *S-1-5-32-544
SeRemoteShutdownPrivilege = *S-1-5-32-544
SeAuditPrivilege = *S-1-5-19,*S-1-5-20,*S-1-5-99-216390572-1995538116-3857911515-2404958512-2623887229
SeIncreaseQuotaPrivilege = *S-1-5-19,*S-1-5-20,*S-1-5-99-216390572-1995538116-3857911515-2404958512-2623887229

Outcome and Learnings:

Despite hardware limitations, I was able to successfully simulate the key concepts behind Windows Server administration using PowerShell. This included:

- Creating and managing user accounts
- Setting up and assigning users to local groups
- Reviewing system-wide password policies
- Understanding how user and group management is a critical part of identity and access control in enterprise networks

This exercise gave me confidence in using PowerShell for system administration and reinforced my understanding of how Windows Server environments are structured and maintained — even without a GUI or domain controller.