Module 7: Windows Admin and Hardening

Part - 1. Introduction

Navigate to **Azure Labs**, login, and start the Windows machine

- Learning Objectives -

By the end of this study session you should be able to:

1. Provide some general examples of system hardening
2. Differentiate between a Domain Controller and Active Directory
3. Explain the LLMNR protocol vulnerability
4. Explain the concept of Access Control, and provide examples of it
5. Explain the "Living off the Land" attack method, and ways to mitigate it

- Key Vocab Review -

1. **Hardening -** Establishing baseline security on a computer system/network
2. **Domain Controller -** A server that acts as a network gatekeeper, handling authentication requests and enforcing security policies.
3. **Active Directory -** This program configures and stores the Domain Controller's settings.
4. **Organizational Unit *(OU)* -** The top-most organizational layer in an Active Directory used to organize groups, users, security policies and admin control.
5. **Group Policy Object *(GPO)* -** A technical access control, applicable to the computer itself as well as the users on it.

Part - 2. Module 7.3 in Class Activities *(Prerequisite for Module 7 HW)*

In this section we will:

- Initialize the lab
- Discover the Active Directory Users and Computers tool
- Create our core AD architecture

1. Download the RDP link

2. Connect, say yes to all prompts

3. Login to the RDP Host: azadmin : p4ssw0rd*

4. Open CMD as Admin and run this command: *slmgr.vbs /rearm*

5. Restart

Windows Lab Setup Guide

Windows Server: sysadmin : p4ssw0rd*

Windows 10: sysadmin : cybersecurity

Module 7: Windows Admin and Hardening

Let's suppose GoodCorp has multiple departments like most businesses do. We will use Sales, and Development as example departments. They have people working in each, and so we need to create user accounts for them, and we need to create a secure and manageable architecture/organizational structure. That's why we need a Domain Controller, to easily organize and manage the users and machines on our network.

Step 1 - Create OU's

* Our core architecture will consist of **4** Organizational Units, **2** Groups, and **2** Users.
See the **Solution Guide** for a more detailed breakdown.

1. ***Open Active Directory Users and Computers tool on the virtual Windows Server***
   `Windows Search -> Server Manager -> "Tools" -> ADUC`
2. ***Create top level/primary OU***
   `Right Click GOODCORP.NET -> NEW -> Organizational Unit`
   `NAME: GC Users`
3. ***Create OU for computers (same level as GC Users, not within it)***
   `Right Click GOODCORP.NET -> NEW -> Organizational Unit`
   `NAME: GC Computers`
   `Drag and drop the Windows 10 machine from "Computers" to "GC Computers"`
4. ***Create 2 sub layers of the OU tree (this goes INSIDE GC Users)***
   `Right Click GC Users -> NEW -> Organizational Unit`
   `NAME: Sales`
   `Right Click GC Users -> NEW -> Organizational Unit`
   `NAME: Development`

Step 2 - Create/Add Groups to OU Tree

* Now that we have our OU's, let's make the next layer, GROUPS.
* An *OU* is used to assign Group Policies (security controls), a *group is used to assign access to data.*
* ***You can only link a GPO to an OU, and you can only give access permissions for data to groups.***
* Learn more about group types and use cases here***: Complete Guide***

1. ***Create a GROUP for each department***

   ```
   Right Click Sales OU -> New -> Group
   NAME: Sales
   Options: Global, security
   ```

   ```
   Right Click Development OU -> New -> Group
   NAME: Development
   Options: Global, security
   ```

Step 3 - Create/Add Users to Groups

* Now that we have our GROUPS, let's make some USERS.

1. ***Create USERS (Bob and Andrew)***

   ```
   Right Click Sales OU -> NEW -> User
   NAME: Bob
   PW: Ilovesales!
   ```

   ```
   Right Click Development OU -> NEW -> User
   NAME: Andrew
   PW: Ilovedev!
   ```

2. ***Add USERS to their requisite department GROUPS***

   ```
   Right Click -> Username -> Add to Group
   Object Name: [name of the group]
   ```

3. ***Add the SALES GROUP to the Remote Desktop Users Group***

   ```
   Expand Sales OU -> Right Click Sales Group -> Add to group
   Object Name: Remote Desktop Users
   ```

** Before moving on it's a good idea to double-check your work. Take a moment to review.*

Part - 3. Making Group Policy Objects

Now that we have a structure for our Active Directory, let's add some security controls.

In this section we will:

- Discover the GPO Management Tool
- Create 4 GPOs (Disable Control Panel, Disable LLMNR, Account Lockout, Powershell Logging)
- Link the GPOs to their requisite Organizational Units

GPO 1 - Disable Control Panel

1. ***Open the Group Policy Management tool***

   ```
   Windows Search -> Server Manager -> "Tools" -> GPO Management
   ```

2. ***Create GPO for GOODCORP.net***

   ```
   Right Click Group Policy Objects under GOODCORP.net -> NEW
   NAME: Limit Settings
   ```

3. ***Configure the GPO***

   ```
   Right Click the GPO -> Edit -> Settings Page Visibility
   PATH: User Config/Policies/Admin Templates/Control Panel
   POLICY: Enabled
   VISIBILITY: showonly:about;themes
   APPLY
   ```

4. ***Link the GPO to the Sales OU***

   ```
   Right Click Sales OU -> "Link existing..." -> "Limit Settings" -> OK
   ```

GPO 2 - Disable LLMNR

LLMNR is an inherently insecure protocol, so we must disable it as part of hardening.

Learn more here: **LLMNR Vulnerability**

1. ***Create GPO for GOODCORP.net***

   ```
   Right Click Group Policy Objects under GOODCORP.net -> NEW
   NAME: NO LLMNR
   ```

2. ***Configure the GPO***

   ```
   Right Click the GPO -> Edit -> Turn Off Multicast Name Resolution
   PATH: Computer Config\Policies\Administrative Templates\Network\DNS Client
   POLICY: Enabled
   APPLY
   ```

3. ***Link the GPO to GC Computers OU***

   ```
   Right Click GC Computers OU -> "Link existing..." -> "NO LLMNR" -> OK
   ```

* Note that we linked our first GPO to a group, but this one is applied to the computers themselves.

* This is one example of how to use a GPO to make system-wide changes.

GPO 3 - Account Lockout

1. ***Create GPO for GOODCORP.net***

   ```
   Right Click Group Policy Objects under GOODCORP.net -> NEW
   Name: Account Lockout
   ```

2. ***Configure the GPO***

   ```
   Right Click the GPO -> Edit -> Account Lockout Policy
   PATH: Computer Config\Policies\Windows Settings\Security Settings\Account
         Policies\Account Lockout Policy
   DURATION: 15
   THRESHOLD: 10
   RESET: 10
   APPLY
   ```

3. ***Link the GPO to GC Computers OU***

   ```
   Right Click GC Computers OU -> "Link existing..." -> "Account Lockout" -> OK
   ```

GPO 4 - Powershell Logging

1. ***Create GPO for GOODCORP.net***

   ```
   Right Click Group Policy Objects under GOODCORP.net -> NEW
   NAME: Powershell Logging
   ```

2. ***Configure the GPO***

   ```
   Right Click the GPO -> Edit -> Turn Off Multicast Name Resolution
   PATH: Computer Config\Policies\Administrative Templates\Windows
         Components\Windows PowerShell
   MODULE LOGGING: Put a wildcard to log all (*)
   SCRIPT BLOCK LOGGING: Log script block invocation start/stop events (check the box)
   SCRIPT EXECUTION: Allow all scripts
   POWERSHELL TRANSCRIPTION: Send to default directory, and include invocation headers
   APPLY
   ```

3. ***Link the GPO to GC Computers OU***

   ```
   Right Click GC Computers OU -> "Link existing..." -> "Powershell Logging" -> OK
   ```

* Now we have 4 active GPO's and we are on our way to having a hardened system.

* If you can make 1 GPO, you can make any GPO, because the process doesn't change.

Part - 4. Powershell Script: Enumerate Access Control List

Now that we have an architecture, and GPO's, let's shift our focus over to Powershell.

This is not really a "hardening" task, but a useful tool for admins.

In this section we will:

- Discover PowerShell
- Create a short "for loop" type script
- Test the script

* It is recommended to use the Windows 10 lab machine, but the reality is you can do this on any machine with Powershell ISE, even your local host.

Students are given this template and asked to complete it:

```
foreach ($item in $directory) {
        <script block>
        }
```

*** "$item" is a default variable in Powershell, but "$directory" is not.**

1. ***Open Powershell ISE***

   ```
   Windows Search -> Powershell ISE -> Select
   ```

2. ***Enter the script:***

   ```
   $directory = Get-ChildItem .\
   foreach ($item in $directory) {
     Get-Acl $item
   }
   ```

3. Save

   * First, we define the "directory" variable as the output of Get-Childitem.
   That way we can use the item list as our input in the loop.
   In other words, $directory is not a path to a directory, but a list of items in the working directory.
   Then, for each item in $directory, the script will run "Get-Acl" on the item, and loop back through the list until complete.

4. ***Test the script***

   Navigate to any directory, and run the script using the full path, for example:

   ```
   C:\Users\sysadmin\Documents\enum_acls.ps1
   ```

BONUS

Navigate to `C:\Users\sysadmin\Documents` and since we ran a PowerShell script, and we have enabled Powershell logging via a GPO, we should have a log here if everything is working properly.

***To undo all changes and revert back to default, use this PowerShell command on each OU!**

```
Get-ADOrganizationalUnit -Identity 'OU=NAME,DC=GOODCORP,DC=NET' |
Set-ADOrganizationalUnit -ProtectedFromAccidentalDeletion:$false -PassThru |
Remove-ADObject -Recursive -Confirm:$false
```