HARDEN AD

Secure your domain in minutes

User Manual

Community Edition - Version 2.9.4

About this document

Welcome to **Harden**[AD]! The whole community members hope you will find our tools useful and helps you in securing more efficiently your Active Directory by leveraging your security posture through a whole bunch of known good practices recommended by Security Expert from all around the world.

This document is solely intended to assist you in setting up and deploy the solution. Even if the script is self-sufficient and can be run “as-is” in a test environment, we do not recommend proceeding that way for your production environment.

Begin your journey with us: read through the manual, follow our instructions, then move to a better secure AD!

Getting trouble or having a question?

The community is here to help. Please, feel free to join us through our mail **contact@hardenad.net** or join our team members on LinkedIn or discord (CADIM).

SUMMARY

1 Introduction to Harden AD

1.1 Files and folders

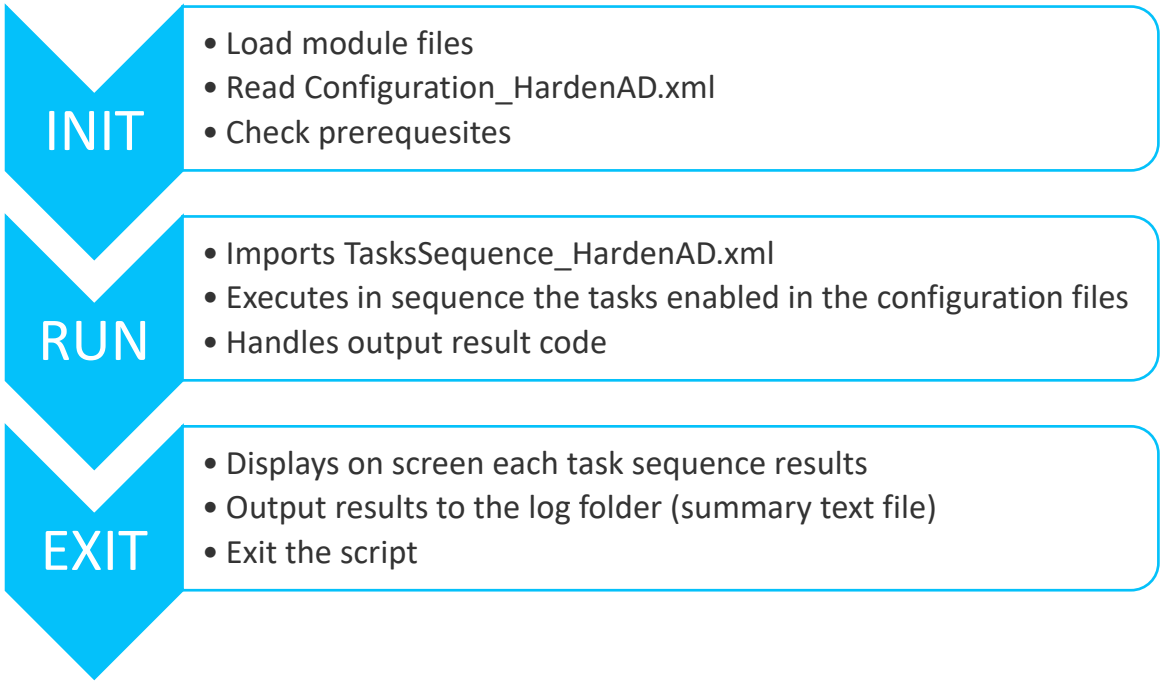
Here is the folders hierarchy you always should maintain:

TREE	DESCRIPTION
HardenAD.ps1	Main script.
Configs	Folder that contains configuration files for the script.
Configuration_HardenAD.xml	Configuration file for the script only. Do not modify.
Logo_harden.txt	Text file with the Harden AD ASCII logo file.
TasksSequence_HardenAD.xml	Configuration file that defines how the script should implement the Harden AD model in your environment.
Inputs	Folder that contains data files to be used during the script execution.
GroupPolicies	Folder that contains all Group Policies related data
{…}	Folder that contains a backup of a GPO to be imported.
Hardenad.migtable	A translation file to dynamically replace data within a backup GPO.
WmiFilters	Filter containing Group Policy filters
[…].mof	A backup file of a wmiFilter file.
LocalAdminPwdSolution	Folder that contains the binaries for LAPS
Binaries	Folder that contains the MSI files for deployment
LAPS.x64.msi	MSI file for 64 bits operating systems
LAPS.x86.msi	MSI file for 32 bits operating systems
LogonScripts	Folder that contains the logon script for the deployment of LAPS
Deploy-laps_x64.bat	Batch file for 64 bits operating systems
Deploy-laps_x86.bat	Batch file for 32 bits operating systems
PolicyDefinitions	Folder that contains files used by the Group Policy Management Console
AdmPwd.admx	ADMX file for GPMC
En-US	
AdmPwd.adml	ADLM file for GPMC (english)
ScheduleTasks	Folder that contains data to create scheduled tasks to be run
TaskSchedulesScripts	Folder that contains script files and its subsidiaries
MCS-GroupsFlushing	Folder to be copied on the system
MCS-GroupsFlushing.csv	CSV file that contains ScTsk script data
MCS-GroupsFlushing.ps1	Script to be run
TaskSchedulesXml	Folder that contains XML file for schedule tasks creation
MCS-GroupsFlushing.xml	XML file used to create the ScTsk MCS-GroupFlushing
Logs	Folder that contains logs (deprecated)
Debug	Folder that contains functions debug log (deprecated)
Modules	Folder that contains modules file loaded by the script
AccessControlList.psm1	Module to handle ACL
Domain.psm1	Module to handle domain level targeted scripts
Engine.psm1	Module dedicated to the script engine
File.psm1	Module to handle file manipulation
GroupPolicy.psm1	Module to handle Group Policy scripts
Object.psm1	Module to handle AD objects
OrganizationalUnit.psm1	Module to handle Organizational Units

TREE	DESCRIPTION
Outputs	Folder that contains files generated by the script
HardenAD.kdbx	File that contains admin users name and password
Tools	Folder that contains side-tools
GpoManagers	Tool to manage GroupPolicy data
Translate-BackupID_to_GpoName.ps1	Script generating a translation file to link backupID to GPO name.
KeePass-2.48.1	Portable KeePass edition (used by the script)
[…]	Binaries
XmlBuilder	(Not yet implemented)
XmlBuilder.ps1	(Not yet implemented)

1.2 Global sequence overview

This section provides a global overview of the script execution.



2 Security Model

2.1 Principle

Harden[AD] has developed a security model based on common recommendation from security experts, including Microsoft company, and our own knowledge of security design acceptance by administrators. The below model is a proposition that could easily be adapted to your needs or security requirement. The model is divided in different strategies:

1. **Default AD Behavior:** out of the box an Active Directory includes a lot of default settings that can be harmful for your environment – this tool will ensure that everything is scaled properly.
2. **Organizational Unit design:** its purpose is to ensure that objects can be managed by the right delegated accounts, without compromising any of the m by upscaling the required privileges or giving access to a most privileged object – this is also known as the Identity Tiering Model.
3. **Identity Tiering Model:** too often administrators use the same account to perform administration tasks on servers and workstations, and sometime this account is also their daily production identity. The security model split roles in tiers, each one of them defining a security realm that could not be mixed with other.
4. **Group Policies:** not only should the objects be protected, but the end systems need also to be tuned to reflect your security rules – such settings are applied through group policies. This is also the dynamic parts that you could always adapt or complete to maintain your security posture at the highest level.

2.2 Default AD behavior

2.2.1 Domain joining

2.2.1.1 Default behavior

By design, Active Directory will allow any authenticated user to perform a domain joining operation up to 10 computer objects. This is a security risks as it could also allow somehow to integrate your IT environment with a compromised machine or add one which can be easily used as an entry-point to hack your systems (think of the ability to gain local administrator privilege or to preinstall hacking tools before joining your domain…).

2.2.1.2 Recommended configuration

The proper configuration is to deny to everyone the ability to add a new computer object to the domain.

2.2.1.3 Impact

Once correctly set, only an account with the ability to create computer object and reset their password would be able to add a new computer to the domain. By default, some groups are granted accordingly:

- Domain Admins
- Enterprise Admins

HardenAD will delegate this permission to some groups (see later in this documentation).

2.2.1.4 More about it?

Sneaky Active Directory Persistence #16: Computer Accounts & Domain Controller Silver Tickets – Active Directory Security (adsecurity.org)

MachineAccountQuota - The Hacker Recipes

2.2.2 Active Directory Recycle Bin

2.2.2.1 Default behavior

Starting with release 2008 R2, Active Directory offers the ability to enable a recycle bin: this is a major improvement in a security design as it offers more flexibility and traceability upon objects deletion – a standard approach used by attackers is to grant themselves rights and maintains privileges out of tracks by deleting an account immediately (through scripts, it will take less than 30 seconds).

Nore this will let us tracks illegal activity, but also this will ease your daily activity when an object needs to be restored: no longer need to reboot a DC into a non-authoritative mode and begin a complex restoration process, the recycle bin is accessible through a GUI and not-only restore the object itself but also its group memberships!

This option is disabled by default.

2.2.2.2 Recommended configuration

The option should be enabled. Even more if you never heard about *repadmin /showobjmeta*.

2.2.2.3 Impact

None.

2.2.2.4 More about it?

Introduction to Active Directory Administrative Center Enhancements (Level 100) | Microsoft Docs

How Active Directory Recycle Bin works: Enable AD Recycle Bin - TechNet Articles - United States (English) - TechNet Wiki (microsoft.com)

The AD Recycle Bin: Understanding, Implementing, Best Practices, and Troubleshooting - Microsoft Tech Community

Azure AD Connect sync: Enable AD recycle bin - Microsoft Entra | Microsoft Docs

2.2.3 Site Links

2.2.3.1 Default behavior

When Active Directory is installed, at least one site link is created and used to automatically generate the replication topology. No option is set upon it, which leave the replication between domain controller not belonging to the same AD Site to its factory default: grouping the whole changes in a stack and inform about changes other DC once a time every 180 minutes (3 hours).

The AD Site Link object can be setup in an asynchronous mode where the DC will inform all its partners outside its site that new metadata are available to them – then external DCs will begin a replication request, the same way they do for emergency replication (password reset, account disabled). With a such setting enabled, you can modify group membership on any DC and having the object updated to any other DC in less than 5 minutes!

From a security point-of-view, this is also a good practice as you can react to an on-going attack very quickly, without seeking first the best DC to use to block your malicious insider actions.

2.2.3.2 Recommended configuration

Setup the *notify* option on every site link.

2.2.3.3 Impact

This would increase network traffic between DC across your WAN links, however if you made few modifications upon objects, this would remain quite low.

2.2.3.4 More about this?

Setting Site Link Properties | Microsoft Docs

Set Active Directory To "Use notify" Replication - TechNet Articles - United States (English) - TechNet Wiki (microsoft.com)

2.2.4 Group Policy Central Store

2.2.4.1 Default behavior

By design, each DC keep a local repository for the Group Policy ADM, ADMX and ADML files: those files are stored in c:\windows\policyDefinitions and are not replicated between DC. This behavior leads to a risk of misconfiguration between DC (file not updated or missing) and could let an attacker exploiting these differences. The ADM files are also heavily used and could generate a SYSVOL bloat.

To remediate to the sysvol bloat risks and ease in maintaining files across DC, the Central Store has been added to Active Directory by moving the policyDefinitions folder to the SYSVOL directory.

2.2.4.2 Recommended configuration

The ADM files should no more be used, and the GPO Central Store should be activated.

2.2.4.3 Impact

None.

2.2.4.4 More about it?

How to Implement the Central Store for Group Policy Admin Templates, Completely (Hint: Remove Those .ADM files!) - Microsoft Tech Community

Create and manage Central Store - Windows Client | Microsoft Docs

Avoiding Group Policy Bloat (techgenix.com)

2.2.5 Default object creation path

2.2.5.1 Default behavior

By design, a new computer object is created in a container named *computers* located at the root of the domain; same is for a new user object, but this time in a container named *users,* again at the root of the domain. Problem is that a GPO could not be linked to a container - however it inherits those from its hierarchy units. Indeed, any GPO linked at the root of your domain will be applied by any new object located in its the default location.

This configuration is not perfect from a production perspective, nor is it from a security one. Firstly, you could not enforce a strategy for newer objects without impacting the whole domain – and blocking inheritance is really something to avoid. Secondly, it is a common scenario to not apply some of the most restrictive strategies to a brand-new system under configuration (integration process), raising the needs to customized GPO specifically designated for mass deployment.

Active Directory let you the ability to modify the default location for computer and user objects.

2.2.5.2 Recommended configuration

Default location should be changed to point to an organizational unit instead of a container.

2.2.5.3 Impact

None for users. IT team needs to be aware of the new location.

2.2.5.4 More about this?

Redirect users and computers containers - Windows Server | Microsoft Docs

2.3 Organizational Units design

2.3.1 Tiering model

Before digging deeper into the OU strategy, let's talk a bit about the administration tiering model. You may have heard the word "tiering" coupled to "network" which is related to an isolation of a network segment protected by a firewall. We could apply the same logical protection to our AD objects by dedicating administration accounts to a "tier".

2.3.1.1 Logical Tiers

A logical tier regroups systems within a common perimeter. A perimeter designs a risk boundary where an attacker can evolve without gaining more privilege. As of, a tier should not be administered by an account which is granted in another tier:

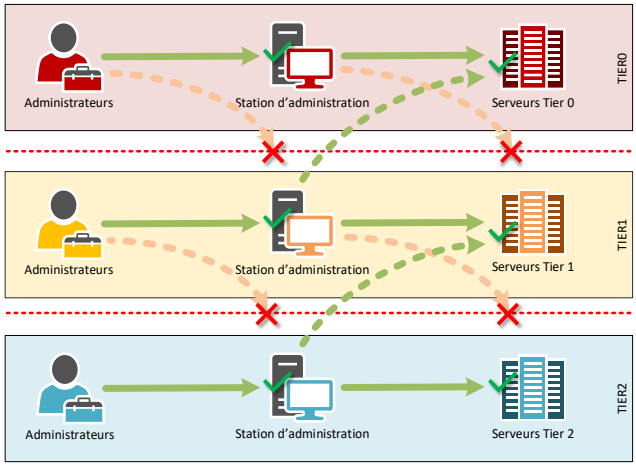


Figure 1 administration tier concept

Nowadays, it is conventional to use a five tiers model:

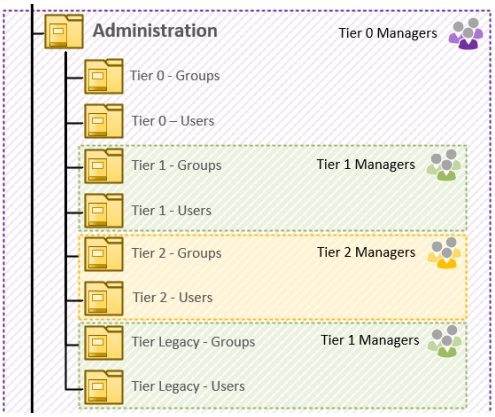
- **Tier Cloud:** cloud identity providers, such as Azure AD.
- **Tier 0:** on-premises identity providers such as Active Directory.
- **Tier 1:** back-end systems, typically servers.
- **Tier 2:** front-end systems, typically workstation, printers, ….
- **Tier Legacy:** any systems that is no more updated by the editor.

Harden[AD] will not manage the Tier Cloud (another product called Harden[365] is focus on this specific aspect) and focus only on the on-premises Active Directory. To achieve this, the model uses group memberships for delegation permissions and Group Policy Object to enforce the strategy when needed.

2.3.2 Organizational Unit design

Harden[AD] allow you to setup your own design but also integrated a default model that could fetch for many companies.

2.3.2.1 OU: Administration



This organizational unit host administrative groups and accounts used by the delegation model.

Each tier contains 2 OU: one for accounts and one for groups. Both are set to allow the corresponding tier manager to deal with objects (see schema). The Tier 1 managers manage the Tier Legacy.

Those organizational units should only contain operators, administrators and managers accounts and delegation groups – including group policies group for applying a denying application.

The organizational unit “groups” contain three sub-organizational units specifically crafted for the HardenAD delegation model.

2.3.2.1.1 Dedicated OU: Groups | Deleg

This organizational unit contains groups used to delegate permission on Active Directory objects within the Tier.

2.3.2.1.2 Dedicated OU: Groups | GPO

This organizational unit contain groups used to allow or deny a group policy to apply on a system.

2.3.2.1.3 Dedicated OU: Groups | Local Admins

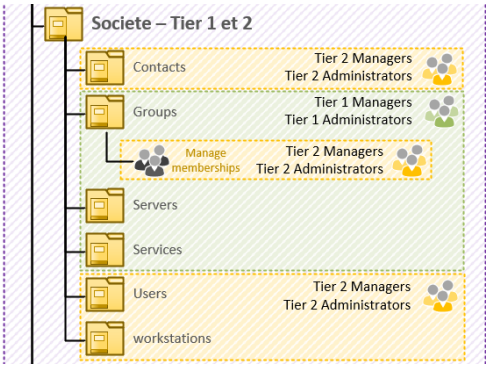
This organizational unit contain groups used to grant local administrator privilege to a system.

2.3.2.2 OU: Company Tier 0

The tier 0 OU contains objects relevant for this tier (and only this tier). Objects are handled by domain admins or equivalent only.

Within this Tier, you could handle: Domain Controllers, Public Keys, infrastructure servers, Azure AD Connect servers, Domain Naming servers, Vault servers.

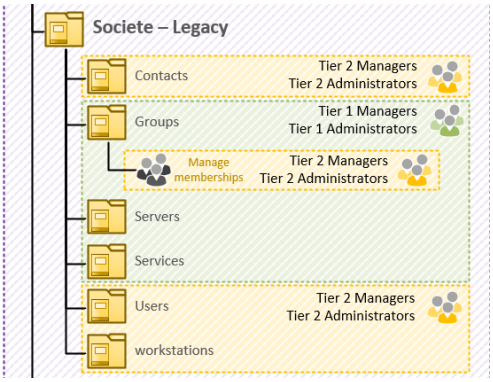
2.3.2.3 OU: Company Tier 1 and 2



This is your production organizational unit, and, in our design, a unique team based in a unique location handle everything.

The delegations are set to let Tier 1 managers and administrators manage groups, service accounts and server system objects, when Tier 2 will manage groups memberships, contacts, users, and workstation system objects.

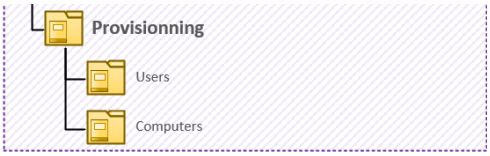
2.3.2.4 OU: Company Tier Legacy



This OU hosts objects dedicated to legacy systems.

The delegations are set to let Tier 1 managers and administrators manage groups, service accounts and server system objects, when Tier 2 will handle groups memberships, contacts, users and workstation system objects.

2.3.2.5 OU: Provisioning



This organizational unit will manage newly created objects in your domain when there is no target path defined. Domain admins members oversee this section.

2.4 Identity tiering model

2.4.1 Account types

2.4.1.1 Administrative account types

Each tier contains three kinds of administrative accounts:

Type	Description
Manager	Fully empowered to deal with objects under their responsibility: they could create, delete and modify them. Such account could perform any tasks within the administrator realm.
Administrator	Allowed to modify existing objects by relocating them in a different OU, changing group membership, …
Operator	Dedicated to performing administration tasks on a system with more privileges than a standard user account (local administrator, managing application, …).

Important notice

Manager and **Administrator** accounts should not login to any system other than its own tier admin stations.

2.4.2 Identity Matrix

The below delegation matrix could help you in identifying your account needs:

The account should…	I am a Simple User	I am an Operator	I am an Administrator	I am a Manager
Login to a system (standard user)	Yes	No	No	No
Login to a system (local administrator)	No	Yes	No	No
Administer an application	No	Yes	No	No
Read a password in a vault	No	Yes	No	No
Add a computer to the domain	No	Yes	No	No
Remove a computer from the domain	No	Yes	No	No
Change group memberships	No	No	Yes	No
Reset a password	No	No	Yes	No
Disable an object	No	No	Yes	No
Modify an object property	No	No	Yes	No
Create an object	No	No	No	Yes
Delete an object	No	No	No	Yes
Relocate an object	No	No	No	Yes

Important notice

If a user from your team qualify to both administrator and manager type, then only create a manager account.

2.4.3 Naming convention

A naming convention ease to identify object either for automation or human reading. Harden[AD] use a naming convention for group and user. You can use the default one or choose to use another that best fits your organization.

2.4.3.1 Groups

Convention is:

- [Group Scope]-[Group Type]-[Tier]_[Name]

Where:

- [Group Scope] equal **G** (global), **L** (domain local) or **U** (universal)
- [Group Type] equal **S** (security) or **D** (distribution)
- [Tier] equal **T0** (tier 0), **T1** (tier 1), **T2** (tier 2) or **TL** (tier legacy)
- [Name] equal whatever you want

Example:

- For a global security group that will contain all Tier 0 manager: **G-S-T0_Managers**.

Please be advised that some groups are automatically created by the script while processing tasks and will offers you the ability to set a naming convention (or reuse the one provided).

2.4.3.2 User

A user admin account should be named in a way that one person could use multiple accounts and easily remembers its login name.

Convention is:

- [UserID]-[Tier][Role]

Where:

- [UserID] equal to the user unique ID (adminXX, AXXX, firstname.lastname, …)
- [Tier] equal T0 (tier 0), T1 (tier 1), T2 (tier 2) or TL (tier legacy)
- [Role] equal M (manager), A (administrator) or O (operator)

Example:

- For WEARIS Mickey, a tier 2 operator which unique id is Admin01: ADMIN01-T2O
- For Justin SECOND, a tier 1 administrator which unique ID is JSECOND: JSECOND-T1A

2.4.4 Local Admin Password Solution

The Local Admin Password Solution will be implemented by this script and will delegate permission to read and reset password to operators, administrators, and managers.

The default design will teach the script to:

- Assign self-permission on all servers and workstations organizational units
- Assign password reading right for server to a dedicated group (L-S-T1-DELEG_LAPS_PwdRead)
- Assign password reading right for station to a dedicated group (L-S-T2-DELEG_LAPS_PwdRead)
- Assign password reset right for server to a dedicated group (L-S-T1-DELEG_LAPS_PwdReset)
- Assign password reset right for station to a dedicated group (L-S-T2-DELEG_LAPS_PwdReset)

The preferred method to administer station and server, when local administrator privilege is required, is to use the system built-in administrator account – in case of credential thief, the account will not be elevated on other system. The account password will be stored within Active Directory.

You can have more information about LAPS here: Download Local Administrator Password Solution (LAPS) from Official Microsoft Download Center (documentation is included)

2.5 Group policy objects

The script comes with a full set of security features integrated through GPOs. This section gives you inputs about them.

2.5.1 WMI Filters

The following WMI filters are added to the domain:

Name	Description
Windows_8.1_And_Older_Only	Only client OS older or equal to Windows 8.1
Windows_10_And_Newer_Only	Only client OS newer or equal to Windows 10
Windows_Client_Only	Only client OS
Windows_Server_2012_R2_And_Older_Only	Only server OS older or equal to Windows Server 2012 R2
Windows_Server_2016_And_Newer_Only	Only server OS newer or equal to Winows Server 2016
Windows_Server_Only	Only server OS
Windows_X64_Only	Only 64 bits systems
Windows_X86_Only	Only 32 bits systems

You can choose to modify the WMI filter name through the configuration file to reflect your company strategy.

2.5.2 Allow or deny a GPO to a system

Because exception always arise, you may need to manage which object should apply a GPO or should exceptionally not. As a result, Harden[AD] now create two groups to ease you in managing exception or application of each GPO: one for applying and another one for denying execution.

By design, those groups are stored in the administration OU, within its corresponding tier, in GROUPS | GPO. A naming convention is set by default to help administrators in identifying the group role:

- G-S-**[tier]**-GPO_**[GpoName]**_**[mode]**

Where:

- **[Tier]** is replaced by T0 (tier 0), T1 (tier 1) or T2 (tier 2)
- **[GpoName]** is replaced by the GPO Name, shortened (see here under)
- **[Mode]** is replaced by apply (to apply the gpo) or deny (to deny execution)

To avoid long and complex GPO name, those one is shortened when the group is created:

- By removing blank space between,
- By using a dictionary of keywords to replace known words.

2.5.3 GPO list

By design, the following GPO are included in the community edition release.

GPO Name	Description
Updates	
HAD-Auto-Update_S1_Thu_0h-Srv	GPO to update servers
HAD-Auto-Update_S1_Thu_1h-Srv	GPO to update servers
HAD-Disable_Win11_Update-Wks	GPO to disable Windows automatic update
HAD-Win10_Auto_Update-Wks	GPO to update Windows 10 computers with Windows Update for Business
HAD-Win8_Auto_Update-Wks	GPO to update Windows 7 and 8 computers
Legacy Protocol	
HAD-Disable_NTLM1_LM-All	GPO to disable LM and NTLMV1 protocol

HAD-Configure_NTLMv2_128bits-All	GPO to configure NTLM V2 128 bits encryption
HAD-Disable_LDAP_Bind_Simple-All	GPO to disable LDAP Bind Simple
HAD-Disable_LLMNR-All	GPO to disable LLMNR
HAD-Disable_LMHASH-All	GPO to disable LMHASH
HAD-Disable_SMBv1-All	GPO to disable SMB V1 protocol
HAD-Disable_SSL2_SSL3-All	GPO to disable SSL2 and SSL 3.0 protocol
HAD-Disable_Wdigest-All	GPO to disable WDigest protocol
Tiering	
HAD-Login_Restrictions-Paw	GPO to manage who can open a session on a Paw
HAD-Login_Restrictions-All-T0	GPO to manage who can open a session on a Tier 0 computers
HAD-Login_Restrictions-Paw-T0	GPO to manage who can open a session on a Tier 0 Paw
HAD-Login_Restrictions-All-T1	GPO to manage who can open a session on T1 servers
HAD-Login_Restrictions-Paw-T1	GPO to manage who can open a session on Tier 1 Paw
HAD-Login_Restrictions-All-T12	GPO to manage who can open a session on workstations (Tier 2) and Tier 1 servers
HAD-Login_Restrictions-All-T2	GPO to manage who can open a session on workstations (Tier 2)
HAD-Login_Restrictions-Paw-T2	GPO to manage who can open a session on Tier 2 Paw
HAD-Login_Restrictions-All-TLeg	GPO to manage who can open a session on Tier Legacy computers
HAD-Login_Restrictions-Paw-TLeg	GPO to manage who can open a session on Tier Legacy Paw

Logs	
HAD-ANSSI_Logs-All	GPO to configure Windows audit log
HAD-PowerShell_Logs-All	GPO to enable PowerShell V5 logs
Local Accounts	
HAD-Local_Admins-Srv-T0	GPO to manage who are members of local administrators' group on Tier 0 servers
HAD-Local_Admins-Wks-T0	GPO to manage who are members of local administrators' group on Tier 0 workstations
HAD-Local_Admins-Srv-T1	GPO to manage who are members of local administrators' group on Tier 1 servers
HAD-Local_Admins-Wks-T2	GPO to manage who are local administrators on a Tier 2 workstations
HAD-Local_Admins-Wks-Tleg	GPO to manage who are local administrators on Tier Legacy workstations
HAD-Local_Admins-Srv-Tleg	GPO to manage who are local administrators on Tier Legacy servers
HAD-Local_Accounts-All	GPO to enable and rename administrator local account and to disable and rename guest local account
Bitlocker	
HAD-BitLocker_TPM_Only-Wks	GPO to activate bitlocker TPM only
HAD-BitLocker_TPM_and_PIN-Wks	GPO to activate bitlocker TPM and PIN
LAPS	
HAD-Deploy_LAPS_x64-Wks	GPO to activate LAPS x64
HAD-Deploy_LAPS_x86-Wks	GPO to activate LAPS x86

Protocol	
HAD-Configure_SMB-Signing-All	GPO to sign SMB communication
HAD-Block_WinRM-All	GPO to block WINRM
HAD-Activate_NLA_for_RDP-All	GPO to enable NLA for RDP access
HAD-Activate_RDP-All	GPO to enable RDP
HardenAD Enging	
HAD-Scheduled_Tasks_Motor-DC	GPO to add scheduled task of DC: add local admin groups automatically
Firewall	
HAD-Configure_Windows_Defender-All	GPO to enable and configure Microsoft Defender
HAD-Firewall_Audit_Only-All	GPO to configure Microsoft Firewall on audit mode
HAD-Firewall_Block_Inbound-All	GPO to enable Microsoft Firewall (inboud trafic)
Printers	
HAD-Disable_Print-Spooler_Service_Srv	GPO to disable Print spooler
Session	
HAD-Block_Ms_3rd_Party-All	Gpo to block third party Microsoft Windows components
HAD-Block_MSLive_Accounts-All	GPO to block authentication with Microsoft Live accounts
HAD-Block_Anonymous_SAM_Enum-All	GPO to block anonymous access and SAM enumeration
HAD-Configure_Screenlock-All	GPO to lock inactive session automatically
HAD-Logon_No_Cache-Srv	GPO to disable logon cache for server

HAD-Logon_Cache-Wks	GPO to configure session cache for workstations
Exploits	
HAD-Block_BloodHound-All	GPO to block BloodHound tool (Netcease)
HAD-Secure_NetLogon-All	GPO to secure Netlogon (encryption and password change of computer accounts)
Misc	
HAD-Network_Access_Settings-All	Configure Network Access Settings on All Members
HAD-Network_Access_Settings-DC	Configure Network Access Settings on DC
HAD-Network_Access_Settings-Wks	Configure Network Access Settings on Workstations
HAD-IPv4_Priority-All	GPO to configure IPV4 as default IP protocol instead of IPv6
HAD-Disable_Posix_Subsystem-All	GPO do disable Posix subsystem
HAD-Enforce_Symbolic_Links-All	GPO to enforce symbolic links
HAD-FIPS_Cryptography-All	GPO to enable FIPS cryptography
HAD-GPO_Refresh_Cycle_20min-All	GPO to force group policy refresh interval to 20 minutes
HAD-Clear_PageFile_Shutdown-All	GPO to clear Pagefile at shutdown
HAD-Block_Camera_On_Lockon-All	Gpo to prevent enabling lock screen camera
UAC	
HAD-Enforce_UAC-All	GPO to enable UAC
HAD-UAC_Enforced-All	GPO to apply UAC to Builtin administrator account
Services	

HAD-Disable_Allow_Game_DVR_Service-Wks	GPO to disable AllowGameDVR service

3 Using the file TasksSequences_hardenAD.xml

3.1 Presentation

The **TasksSequences_HardenAD.xml** file contains everything needed to customize and apply the security model following your need. The present chapter will explain each item and to use them.

As every XML file, the data are stored through attributes: we will name the first level of attributes as **section**, as they will contains a set of attributes scoped in thematic. The below sub-chapter will present its section in a logical order.

3.2 Section: Organizational Unit

3.2.1 How it works

This section is dedicated to the Organization Unit tree models. You can add as many as you want to call them later through the **sequence** section. Each model is delimited through the attribute **OU** – this attribute contains a sub-attribute named **ChildOU** which represents an organizational unit. All models are regrouped through the parent attribute **ouTree**.

When you create your model, you can integrate sub-organizational units anywhere you need (there is no deep limit).

Here is a structural example:

```
<ouTree>
    <OU Class=”example” Name=”Demo OU” Description=”Demo desc.”>
            <ChildOU Name=”level 1 with child” Description=”This one with child OU”>
                    <ChildOU Name=”level 2” Description=”Child of level 1 with child”/>
            </ChildOU>
    </OU>
</ouTree>
```

3.2.2 ouTree

This attribute needs to be the parent of all of yours **OU** attributes.

3.2.2.1 OU

This attribute should always be a child of the attribute **ouTree**. It could only contain the attribute **ChildOU** as child.

Construction:

```
<OU Class="" Name="" Description="">
```

Parameters:

Class	the name used as reference to call through the sequence section
Name	the name of the organizational unit.
Description	the description field you will see in the console.

3.2.2.2 ChildOU

This attribute could be either a child of the attribute **OU** or itself.

Construction:

```
<ChildOU Name="" Description="">
```

Parameters:

Name	the name of the organizational unit.
Description	the description field you will see in the console.

3.3 Section: DelegationACEs

3.3.1 How it works

This section is intended to delegate permission through Active Directory. When a delegation is set, the script will add security access rule (ACL) to the appropriate organizational unit. Each attribute **ACL** is relevant to one permission.

Here is a structural example:

```
<DelegationACEs>
    <ACL Trustee=”My User”>
        <Right=”CreateChild”/>
        <RightType=”All”/>
        <Inheritance=”All”/>
        <InheritedObjects=””/>
        <ObjectType=”User”/>
        <TargetDN=”OU=example,RootDN”/>
    </ACL>
</DelegationACEs>
```

3.3.2 ACL

This attribute is a child of the attribute **DelegationACEs.**

Construction:

```
<ACL Trustee="" Right="" RightType="" Inheritance="" InheritedObjects="" ObjectType="" TargetDN="">
```

Parameters:

Trustee	Specifies the identity of the object that's getting permissions added. You can specify either the distinguished name (DN) of the object or the object's name if it's unique.
Right	Specifies the rights that you want to add on the Active Directory object. Valid values include: AccessSystemSecurity, CreateChild, DeleteChild, ListChildren, Self, ReadProperty, WriteProperty, DeleteTree, ListObject, ExtendedRight, Delete, ReadControl, GenericExecute, GenericWrite, GenericRead, WriteDacl, WriteOwner, GenericAll and Synchronize. You can specify multiple values separated by commas.
RightType	Allow or deny
Inheritance	Specify how permissions are inherited: • None • All • Descendents: all descendent object as defined in InheritedObjects • Children • SelfAndChildren
InheritedObjects	what kind of object inherits this access control entry (User, Group, Computer, Contact, …)
ObjectType	what type of object the permission should be applied to. The ObjectType parameter can only be used if the Right parameter is set to CreateChild or DeleteChild.
TargetDN	Distinguished Name of the target Organization Unit to modify. You can use the keyword **RootDN** to refer to the domain distinguished name.

3.4 Section: Translation

3.4.1 How it works

The attribute **Translation** is used as a dictionary by the script to dynamically call a value – this is easier to refer to %Admin-T0% instead of “G-S-T0-Admins” and this will also simplify the script execution as you will guarantee yourself to not have a mistyping input somewhere.

Here is a structural example:

```
<Translation>
    <WellKnownID objectClass=”text” translateFrom=”%rootDN%” TranslateTo=”DC=My,DC=Domain”/>
    <Keyword LongName=”Servers” ShortenName=”Srv”/>
</Translation>
```

3.4.2 WellKnownID

The attribute is a child of **Translation**. Each attribute is a unique input. You can add as many as you want.

Construction:

```
<WellKnownID ObjectClass="" TranslateFrom="" TranslateTo="">
```

Parameters:

ObjectClass	Type of the object. Needed as it will be used to rewrite the GPO migration table. Use text, group, user or SID.
TranslateFrom	Your referral to be translated. It is mandatory to use "%" to delimit the string to replace.
TranslateTo	The new value to use.

3.4.3 Keyword

The attribute is a child of **Translation**. Each attribute is a unique input. You can add as many as you want. This attribute is used to shorten GPO group name.

Construction:

```
<Keyword LongName="" ShortenName="">
```

Parameters:

LongName	Word to identify
ShortenName	Word shortened

3.5 Section: GroupPolicies

3.5.1 How it works

This attribute includes all attributes used to manage Group Policies. There are three attributes used in here:

- WMIFilter: to manager WMI Filter
- GlobalGpoSettings: to define global parameters to all GPO
- GPO: to manage each GPO individually

Here is a Structural example:

```
<GroupPolicies>
    <WmiFilters>
            <Filter Name=”example” source=”example.mof”/>
    </WmiFilters>
    <GlobalGpoSettings GroupName=G-S-%tier%_%GpoName%” Tier0=”T0” Tier1=”T1” Tier2=”T2”>
            <GpoTier0 OU=”OU=GPO,OU=Groups T0,OU=Admin,%RootDN%”/>
            <GpoTier1 OU=”OU=GPO,OU=Groups T1,OU=Admin,%RootDN%”/>
            <GpoTier2 OU=”OU=GPO,OU=Groups T2,OU=Admin,%RootDN%”/>
    </GlobalGpoSettings>
    <GPO BackupID=”{xxx-xxx}” Validation=”Yes” Name=”Demo GPO” Description=”Desc.”>
            <GpoMode Mode=”Both” Tier=Tier0”/>
            <GpoFilter WMI=”example”/>
            <GpoLink Path=”OU=admin,%RootDN% Enabled=”Yes” Enforced=”No”/>
    </GPO>
</GroupPolicies>
```

3.5.2 WMIFilter

The attribute **WMIFilter** could only contain the child attributes **Filter**.

3.5.2.1 Filter

The attribute **Filter** is used to import a new WMIFilter to Active Directory.

Construction:

```
<Filter Name="" Source="">
```

Parameters:

Name	The name you will give to the WMI Filter.
Source	The MOF file used to import the WMI Filter settings. The MOF file should be present in the folder Inputs | GroupPolicies | WmiFilters.

3.5.3 GlobalGpoSettings

The attribute **GlobalGpoSettings** allow you to define the baseline for the group to be created for each GPO.

Construction:

```
<GlobalGpoSettings GroupName="" Tier0="" Tier1="" Tier2="">
```

Parameters:

GroupName	Base name to create a GPO's groups. Use Translation table.
Tier0	Short name to use for GPO linked at Tier 0 level.
Tier1	Short name to use for GPO linked at Tier 1 level.
Tier2	Short name to use for GPO linked at Tier 2 level.

3.5.3.1 GpoTier0

This attribute **GpoTier0** is used to define the target organizational unit storing the associated groups (Tier 0).

Construction:

```
<GpoTier0 OU="">
```

Parameters:

OU	Distinguished name of the target organizational unit.

3.5.3.2 GpoTier1

This attribute **GpoTier1** is used to define the target organizational unit storing the associated groups (Tier 1).

Construction:

```
<GpoTier1 OU="">
```

Parameters:

OU	Distinguished name of the target organizational unit.

3.5.3.3 GpoTier2

This attribute **GpoTier2** is used to define the target organizational unit storing the associated groups (Tier 2).

Construction:

```
<GpoTier0 OU="">
```

Parameters:

OU	Distinguished name of the target organizational unit.

3.5.4 GPO

The attribute <GPO> is used to define the Group Policy Objects to add to your domain. You can add as many as needed to the <GroupPolicies> attribute. This attribute requires the sub-attributes <GpoMode>, <GpoFilter> and <GpoLink> in its definition.

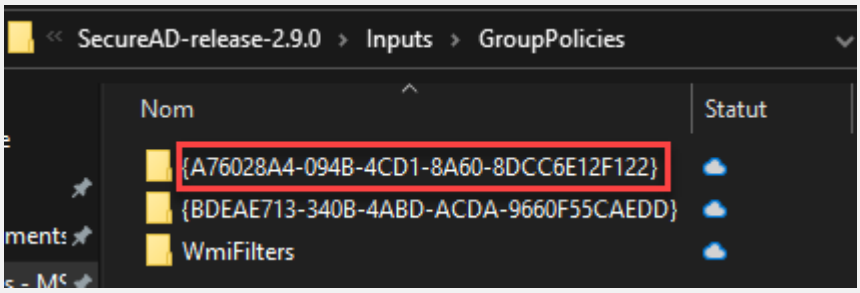
Here is an example:

```
<GPO BackupID="{xxx-xxx}" Validation="Yes" Name="Demo GPO" Description="Desc.">
    <GpoMode Mode="Both" Tier=Tier0"/>
    <GpoFilter WMI="example"/>
    <GpoLink Path="OU=admin,%RootDN% Enabled="Yes" Enforced="No"/>
</GPO>
```

Construction:

```
<GPO BackupID="" Validation="" Name="" Description="">
```

Parameters:

BackupID	Refer to the folder name containing the GPO backup, as shown in the below capture: 
Validation	Accepted values are **Yes** or **No**. When the value is set to Yes, the GPO will be imported either by adding the new GPO or by overwriting the existing one.
Name	Name of the GPO, as shown in the Group Policy Management Console – also used to generate the group names to manage allow or deny delegation
Description	A description to add to the GPO object, detailing the purpose of the object.

3.5.4.1 GpoMode

The attribute **GpoMode** is used to define how the GPO will works and the belonging Tier for configuration.

Construction:

```
<GpoMode Mode="" Tier="">
```

Parameters:

Mode	BOTH	to create both group filter to Allow and Deny GPO application (default mode)
	ALLOW	to create only the Allow group – let you manage whom get the GPO applied.
	DENY	to create on the Deny group – let you manage whom will be filtered out.
Tier	Tier0	to create management group (allow, deny) to tier 0 Group OU.
	Tier1	to create management group (allow, deny) to tier 1 Group OU (not recommended)
	Tier2	to create management group (allow, deny) to tier 2 Group OU (not recommended)

3.5.4.2 GpoFilter

The **GpoFilter** attribute let you assign a WMI filter to the corresponding GPO.

Construction:

```
<GpoFilter WMI="">
```

Parameters:

WMI	WMI Filter name. You can refer to a WMI filter declared within the section <WmiFilter>.

3.5.4.3 GpoLink

The **GpoLink** attribute let you set the linage position within your organizational units.

Construction:

```
<GpoLink Path="" Enabled="" Enforced="">
```

Parameters:

Path	DistinguishedName of the target OU where the GPO must be linked. Use "RootDN" to refer to the DN of the domain.
Enabled	**Yes** or **No**. Teach the script to enable the link or not.
Enforced	**Yes** or **No**. Teach the script to enforce this stratagy to any descending GPO that normally overcome its parameters.

3.6 Section: Accounts

3.6.1 How it works

The **Account** section list all user identities to be generated by the script.

The script first check if the account is already present within the domain and if not, generate it. The newly created user is then stored with its random password in a keepass file.

Here is an example:

```
<Accounts>
<User DisplayName=”Herby N’bee” Surname=”N’Bee” GivenName=”Herby” SamAccountName=”hnbee.t0m”
Descripton=”T0 M example account”/>
</Accounts>
```

Keepass location file: Outputs | HardenAD.kbdx

The keepass password is defined through the calling section (default value: **H4rd3n@D!!**).

3.6.2 User

The **User** attribute let you create a new identity in your domain.

Construction:

```
<User DisplayName="" Surname="" GivenName="" samAccountName="" Description="" Path="">
```

Parameters:

DisplayName	Display name of the account (LDAP attribute DisplayName)
Surname	Surname of the account
GivenName	First name of the account
SamAccountName	Login name. Also used to set the UserPrincipalName attribute.
Description	Description of the user.
Path	DistinguishedName of the OU containing the account. you can use ROOTDN to refer to the domain DN.

3.7 Section: Groups

3.7.1 How it works

The **Groups** Section list all group identities to be generated by the script.

The script first check if the group is already present in the domain (if so, nothing is done), and if not, it generates it. The section contains two attributes: <group> and its child <member>.

Here is an example:

```
<Groups>
  <Group name=”MyGroup” Category="Security" Scope="Global" Description="Example”
Path="OU=Groupes,OU=Tier 0,OU=00 - Administration,ROOTDN">
    <Member SamAccountName=”hnbee.t0m”/>
  </Group>
</Groups>
```

3.7.2 Group

The attribute **Group** define the new group to be created.

Construction:

```
<Group Name="" Category="" Scope="" Description="" Path="">
```

Parameters:

Name	Name of the group
Category	**Security** or **Distribution**
Scope	**DomainLocal**, **Global** or **Univeral**.
SamAccountName	Login name. Also used to set the UserPrincipalName attribute.
Description	Description of the user.
Path	DistinguishedName of the OU containing the account. you can use ROOTDN to refer to the domain DN.

3.7.2.1 Member

The attribute **Member** define the member to be populated in the group.

Construction:

```
<Member samAccountName="">
```

Parameters:

samAccountName	samAccountName to be added as member.

3.8 Section: DefaultMembers

3.8.1 How it works

The **DefaultMembers** Section list strictly allowed members of a specified group.

When the script runs, all groups present in this section will be flushed – once done, the group membership is filled-up with the member listed in this section.

Here is an example:

```
<DefaultMembers>
  <Group Target=”MyGroup”>
    <Member>hnbee.t0m</member>
  </Group>
</DefaultMembers>
```

3.8.2 Group

The **Group** attribute set the target to be enforced. You can use a SID, a SamAccountName or use a keyword.

Construction:

```
<Group Target="">
```

Parameters:

Target	Use a Group SID or the SamAccountName.

Keyword(s):

%DomainSID%	When used, the script will retrieve the domain SID.

You can also refer to keywords from the translation section.

3.8.2.1 Member

Use this attribute to set the group membership. You can use the samAccount, the SID or a keyord.

Construction:

```
<member>yourValue</member>
```

Parameters:

yourValue	Replace by a SID or a SamAccountName. You can use keywords too.

Keyword(s):

%DomainSID%	When used, the script will retrieve the domain SID.

You can also refer to keywords from the translation section.

3.9 Section: TaskSchedules

3.9.1 How it works

This section defines the schedule tasks to be added on current system. Schedule tasks are expected to be run on the Primary Domain Controller to avoid using a service account granted with too powerful privileges.

Please be advised that this solution is an alternative to a real scheduling solution (soon to come).

Here is an example:

```
<TaskSchedules>
  <SchedTask Name="Sensitive Groups Flushing" Xml="GroupsFlushing.xml">
     <SchedCmd>powershell.exe</SchedCmd>
     <SchedArg>-windowstyle hidden -file .\MCS-GroupsFlushing.ps1</SchedArg>
     <SchedDir>%BaseDir%\MCS-GroupsFlushing</SchedDir>
     <SchedDsc>Flush highly sensitive groups</SchedDsc>
     <SchedPth>HardenAD</SchedPth>
   </SchedTask>
</TaskSchedules>
```

To add a schedule task, you will need to create it first on a system and export it as an XML backup file. The backup and any associated files to be used by the schedule must be in place in **Inputs | ScheduleTasks**:

- **TasksSchedulesScripts**: the script and all required files to be executed/used. The files must be placed in a folder named as the schedule task will.
- **TasksSechedulesXml**: this is where your backup files to import are located.

When the script deploys the schedule tasks, the content of **TaskSchedulesScripts** will be copied to the scheduling repository on the system.

3.9.2 SchedTask

This sub-section defines a schedule to add to the system. You will need an XML backup file to create it. The sub attributes **SchedCmd**, **SchedArg**, **SchedDir**, **SchedDsc** and **SchedPath** are mandatory.

Construction:

```
<SchedTask Name="" Xml="">
```

Parameters:

Name	The Task Schedule name.
Xml	The backup Xml file.

3.9.2.1 SchedCmd

The executable to be used. It will replace the %command% in the xml file.

Construction:

```
<SchedCmd>YourCommand</SchedCmd>
```

Parameters:

YourCommand	The command to run, without parameters.

3.9.2.2 SchedArg

The parameters to be appended to the script command. If a script is specified, it should present in **Inputs | ScheduleTasks | TasksSchedulesScripts.**

Construction:

```
<SchedArg>YourArgs</SchedArg>
```

Parameters:

YourArgs	Your arguments.

3.9.2.3 SchedDir

The effective path context when the schedule is run - use **%BaseDir%** to refer to the directory value speciefied in the "BaseDir" parameters of the section <TaskSchedules>

Construction:

```
<SchedDir>YourLocalDirectory</SchedDir>
```

Parameters:

YourDir	Your local directory to store scheduling files.

3.9.2.4 SchedDsc

The description field for this task.

Construction:

```
<SchedDsc>YourDesc</SchedDsc>
```

Parameters:

YourDesc	Your task description.

3.9.2.5 SchedPth

The folder name where the task will be stored in the Tasks Scheduler console.

Construction:

```
<SchedPth>YourFolder</SchedPth>
```

Parameters:

YourFolder	Your folder in the task scheduler console.

3.10 Section: LocalAdminPasswordSolution

3.10.1 How it works

This section defines the settings to deploy Local Admin Password Solution (LAPS) within the domain. The script will extend your schema if needed and proceed with delegation on computer objects to let your teams being able to read the password of a computer administrator account.

The configuration relies on 3 permission levels:

1. **Self-Permission**. This will allow a computer system to store the account password in a dedicated attribute of its own object in Active Directory.
2. **Password reading**. This allows a user to read a password by deciphering the attribute on the computer object in Active Directory.
3. **Password Reset**. This allows a user to read and reset the password stored in Active Directory. This is useful when you need to forcefully change the local password on a system.

An add-on is also required on the target systems and is deployed through a group policy objects.

This section will use the attributes <AdmPwdSelfPermission>, <AdmPwdPasswordReader> and <AdmPwdPasswordReset>.

Here is an example:

```
<LocalAdminPasswordSolution>
  <AdmPwdSelfPermission Target="OU=Servers,OU=Harden - Tier 1 and 2,%RootDN%"/>
  <AdmPwdSelfPermission Target="OU=Workstations,OU=Harden - Tier 1 and 2,%RootDN%"/>

  <AdmPwdPasswordReader Target="OU=Servers,OU=Harden - Tier 1 and 2,%RootDN%"
   Id="%domain%\%T1-LAPS-PasswordReader%"/>
  <AdmPwdPasswordReader Target="OU=Workstations,OU=Harden - Tier 1 and 2,%RootDN%"
   Id="%domain%\%T2-LAPS-PasswordReader%"/>

  <AdmPwdPasswordReset  Target="OU=Servers,OU=Harden - Tier 1 and 2,%RootDN%"
   Id="%domain%\%T1-LAPS-PasswordReset%"/>
  <AdmPwdPasswordReset  Target="OU=Workstations,OU=Harden - Tier 1 and 2,%RootDN%"
   Id="%domain%\%T2-LAPS-PasswordReset%"/>
</LocalAdminPasswordSolution>
```

You can read more aout LAPS here: https://docs.microsoft.com/en-us/defender-for-identity/security-assessment-laps

3.10.2 AdmPWdSelfPermission

This attribute will let you define which computer objects will be allowed to store the local password. You can either choose to delegate this permission from the root of your domain or target any organizational unit that will contains computer objects (if so, add as many times as needed, one per OU path).

Construction:

```
<AdmPwdSelfPermission Target=””/>
```

Parameters:

Target	The distinguished name of your target OU. You can use %RootDN% to refer to the distinguished name of the domain.

Keyword(s):

%RootDN%	When used, the script will retrieve the domain distinguished name.

3.10.3 AdmPWdPasswordReader

This attribute will allow you to assign the “read-only” permission to a target organizational unit. You can use as many as needed.

Construction:

```
<AdmPwdPasswordReader Target=”” Id=””/>
```

Parameters:

Target	The distinguished name of your target OU. You can use %RootDN% to refer to the distinguished name of the domain.
Id	The netbios name of the group (or user) which will be delegated on. You can use the translation table to ease in filling-up the script.

Keyword(s):

%RootDN%	When used, the script will retrieve the domain distinguished name.
%Domain%	When used, the script will translate this with the domain netBios name.

You can also refer to keywords from the translation section.

3.10.4 AdmPwdPasswordReset

This attribute will allow you to assign the “read and reset” permission to a target organizational unit. You can use as many as needed.

Construction:

```
<AdmPwdPasswordReset Target="" Id=""/>
```

Parameters:

Target	The distinguished name of your target OU. You can use %RootDN% to refer to the distinguished name of the domain.
Id	The netbios name of the group (or user) which will be delegated on. You can use the translation table to ease in filling-up the script.

Keyword(s):

%RootDN%	When used, the script will retrieve the domain distinguished name.
%Domain%	When used, the script will translate this with the domain netBios name.

You can also refer to keywords from the translation section.

3.11 Section: Sequence

3.11.1 How it works

the <Sequence> section define the different tasks to iterate in "sequence" (can't resist to this one). Each item will drive one hardening item. The script will go through each <Id> attribute in sequence, ordering ids from the lowest to the highest.

Here is an example:

```
<Id Number="010" Name="Restrict computer junction to the domain">
  <CallingFunction>Set-msDSMachineAccountQuota</CallingFunction>
  <UseParameters>0</UseParameters>
  <TaskEnabled>Yes</TaskEnabled>
  <TaskDescription>set `(msDSMachineAccountQuota` to `(0`</TaskDescription>
</id>
```

3.11.2 ID

Each task ID is executed in sequence, based on its **Number** value (ascending).

Each <Id> requires the sub-attributes **CallingFunction**, **UseParameters**, **TaskEnabled** and **TaskDescription**.

Construction:

```
<Id Number="" Name="">
```

Parameters:

Number	Define the sequence ordering, the lowest will be run first.
Name	Screen output display name. Also referred throughout log files.

3.11.2.1 CallingFunction

Name of the function to be called. This should be one from any of the .psm1 files present in the modules directory.

Construction:

```
<CallingFunction>My-Function</CallingFunction>
```

3.11.2.2 UseParameters

Mandatory, even if empty. Specify parameter(s) to pass to **CallingFunction**. Use one per parameter and sort them in sequence (the script will use input ordering).

Construction:

```
<UseParameters>MyParameter</UseParameter>
```

3.11.2.3 TaskEnabled

Accept value YES or NO. When set to NO, the task is disabled and will not be applied.

Construction:

```
<TaskEnabled>YES or NO</TaskEnabled>
```

3.11.2.4 TaskDescription

Text displayed on the screen to let the user knows what is going on while the script is running.

Construction:

```
<TaskDescription>This is your text</TaskDescription>
```

Text output customization

To deal with highlight color in display, use the ` (tild) to initiate and end a color change in your string, then use one of the three characters specified in value the **AltBaseHTxt** (A, B, or C) to select your color - the color will switch back to normal when the sequence ends.

The default highlight values are:

- **`[my text`**: magenta
- **`(my text`**: yellow
- **`{my text`**: gray

The values **AltBaseHTxt** is set within the script hardenad.ps1

4 Prepare your deployment

Once you get familiar with the xml configuration file, you can start to customize the script to fetch your needs. The below steps will help you in your journey to Active Directory hardening.

4.1 Prepare your OU design

The base design will most likely be the one you need – however, you can choose to modify it. As an example, a company who has recently installed a brand-new active directory domain may not have any legacy objects but need to host in its domain some of its client's computer; the administrator then chooses to rename the Tier Legacy to Tier 3 and apply the same strategy as the legacy model.

You can also extend the model by splitting some of the organizational units into sub-OU to delegate administrative tasks to the local IT.

4.2 Prepare your translation Table

First, take time to think about your naming convention and object or name you will frequently re-use through the configuration file. If you intend to use long and complex name, it may be a good point to use a translation ID that will refer to it.

First, update the existing <wellKnownID> reference (mandatory):

- **%RootDN%** should match to your domain distinguished name.
- **%Domain%** should match to your domaine netbios name.
- **%domainDNS%** should match to your fully qualified domaine name.

Second, update the group name reference following your needs. By design, the configuration uses a naming convention to reflect the group scope, the group type, the belonging tier and a short description (in this order). If you plan to use additional translation id, add them now. Keep track of your new ids to use them later.

Finally, you can fulfill the Keywords with your own shorten word – this will be helpful to reduce the group name associated with the GPOs.

4.3 Prepare your accounts

The configuration file will need to be filled-up with your team members. Before adding them, you will need to identify which privilege will be granted to each of your teammates (refers to the section 2.4).

Once done, add your them to the <Accounts> section.

4.4 Prepare your groups

The script come with a bunch of required groups to apply its security design. However, you can choose to adapt them by changing the group name or its description (or anything else). As an example, as the configuration file is intended to be use internationally, the descriptions and names are written in English: you can translate them to your local language.

If you need custom group, you can add them to the existing list – there is no obligation to only create security

groups. If you create a new group, remember to place it to the proper organizational unit:

- In the organizational unit named “groups” within the production organizational unit, which is supposed to host the day-to-day groups (such as permission for an application or a team group) – remember that those groups will be managed by the Tier 1 administrators and that membership could be managed by the Tier 2 administrators.
- In the organizational unit named “groups” within a tier administration organizational unit, when your group will either be used only by the administration accounts, to perform delegation or grant administrative privileges.

Do not hesitate to contact a member of the Harden[AD] community for any help.

4.5 Review the delegation matrix

If you have made any change to the file, take time to review the <delegationACEs> section and look forward to any name you may need to adapt.

If needed, you can add additional delegation – refers to the Microsoft documentation about ACLs: https://docs.microsoft.com/fr-fr/windows-server/identity/ad-ds/plan/delegating-administration-by-using-ou-objects

4.6 Review the default group membership

The <DefaultMembers> section assign the expected member to some default groups and remove all other. Before running the script, take time review your existing groups and ensure that those changes will not create trouble. If needed, add your exception to the list, however, please ensure with a security specialist that your custom configuration will not raise a new risk to your security.

You can also add any custom group from your AD that needs to be verified and purged if needed.

4.7 Setup your schedule tasks base directory

By design, the script will store all the schedule tasks to c:_ADM\TasksSechedulesScripts – you should review if this path needs to be changed (the script will take care of creating the folders if needed).

4.8 Setup Local Admin Password Solution (LAPS)

The <LocalAdminPasswordSolution> section is set to match the default organizational unit tree present in the configuration file. If you have made some change to the organizational design, or create your own, then you need to adapt this section to your design:

- **AdmSelfPwdPermission** should target every OU containing a computer objects, excepted the Domain Controllers’ one.
- **AdmPwdPasswordReader** should assign to a group the permission to read the password to a target OU. Create as many groups as needed to split permission, if needed.
- **AdmPwdPasswordReset** should assign to a group the permission to read and force a reset of the password to a target OU. Create as many groups as needed to split permission, if needed.

4.9 Prepare the Group Policies

The script includes a set of group policies to add to your domain. Before executing the script, you should review

each of them and decide if you can add them without any risks (refers to section 2.5 for details). While reviewing them, you can choose to:

- Deny the overwriting of an existing GPO: set the **validation** attribute to YES.
- Not linking a GPO to an OU: remove the **GpoLink** attribute (or comment it for later use).
- Disable a GPO link: set the **Enabled** attribute to NO.
- Enforce a GPO: set the **Enforced** attribute to YES.
- Change or add a gpo's link to an OU: modify or add the **GpoLink** attribute to reflect your need.

Regarding the WMI filter, you can choose to modify their name – do not forget to adapt the GPO reference to avoid script issue later.

4.10 Review the tasks sequences

Once everything is adapted to your domain, you can finally review the tasks sequence. Each task could be disable by settings the **TaskEnabled** to NO.

The configuration file is self-explanatory so keep time to read it – the Harden**AD** community could help you if any questions arise, through the forum (https://hardenad.net/forums/).

5 Execute the script

The time has come to run the script to enforce your security! Before running it, you should execute an audit with PingCastle – this will give you an idea of the maturity level at the whole beginning.

5.1 Running pingCastle

PingCastle is available at https://www.pingCastle.com – download it, then run:

1. Right-click and select “run as administrator”
2. Select the option **6-advanced**
3. Select the option **4-noenumlimit**
4. Select the menu entry **1-healthcheck-Score** to start the analysis
5. Press “enter” to use you’re your current domain context, or type-in your domain target

Once done, the script will have generated an HTML report right next to the executable.

5.2 Running the script

Connect to one of your domain controllers (we recommend using the one holding the PDC Emulator role) and run PowerShell in an elevated mode (run as administrator). Wait until the scripts then review the execution output.

If the script reports for warnings or errors, you can check in the logs folder for details.

5.3 Running pingCastle. Again.

You could run again the PingCastle tool and compare the output. Then plot out a list of actions to perform and pursue your journey to a hardened directory!

6 Harden Community Edition Licencing

6.1 License description

The *Community Edition license* allow:

- From the association website (https://hardenad.net), accessing all the documentations
- Downloading the Harden Community Edition source code from the community repository (github)
- Deploying the Community Edition to your company – you are limited to 500 users for this edition if you are a private brand; public services company are not limited
- Modifying the source code, following this rules:
  - You have to maintain your code under the *Community Edition License*
  - You have to publish your code to the community repository (github) thourgh a dedicated branch
  - You have to maintain in any documentations and applications the graphical charter and all logos.

6.2 Beneficiaries

The *Harden Community Edition* license is free.

Deploying the *Harden Community Edition* in a private company with more than 500 employees is unallowed.