Windows Server 2016 and Windows 10 Security

Christian Toinard
Full Professor in Computer Sciences
INSA-CVL Bourges – LIFO Laboratory
http://www.toinard.net

[Largely inspired for details from:
"Inside Out" Windows Server 2016, Orin Thomas, Microsoft Press]

''I hereby certify that I have no conflict of interest of any kind. Regarding, Microsoft, I never worked for them either directly or indirectly. I have no specific advantage or financial benefit associated with this course. The purpose only is to improve the knowledge, avoid obscurantism and try to make the Operating System community less guided by ideology, ignorance or scientific pretention.''

The good news

- Microsoft abandons the useless ''SElinux'' approach associated with .Net3.5 since 1) no administrator can compute such **a complex fine-grain policy** and 2) **the computed policy is obviously too permissive** since it is a **static approach** that does not take into account the context of a process (e.g. during the lifecycle of a page, a Firefox process needs different privileges that cannot be dynamically enforced i.e. neither with ''SELinux'' or .Net3.5)

- ***Yes we can:*** Microsoft introduces **advanced security features that make mandatory protection more easy and efficient** (thus Microsoft prevents its customers from developing an inefficient .Net4.0 sandbox for each process!!)**:**
    - **A microkernel approach that enables to share a host between different isolated OSs** (***No we can't***: Linux does not provide such a method!!).
        - As defunct Hubert Zimmermann said ''OS security and safety is a matter of architecture'' (not a question of hacking the kernel!! See the different memory inspections available for attackers with a Linux kernel).
    - An **integrated microkernel protection** such as Credential Guard or Device Guard that prevents memory inspection from the ''guest'' OS and provides a safe ''Trusted Computation Based'' that cannot be compromised even if the attacker as a complete control of the "guest" OS.
    - **Just Enough Administration** that is a simpler but still a time consuming task to enforce separation of privileges and least privileges.

- Microsoft improves all the existing **security mechanisms regarding dynamic and easier management:**
    - **AppLocker controls which applications a user can run**. It helps to automatically compute the required rules. Associated with a GPO, it enables to enforce a policy with the required scale.
    - **BitLocker encrypts the resource while guarantying the availability through Active Directory**. Thus, the recovery of the different hosts in a domain is enforced through the Active Directory service.
    - **Dynamic Access Control (DAC)** provides a way to dynamically assign access permissions to content based on the content's

properties and information about the user and device that are attempting to access the content. For example, you can configure DAC so that only people who have Don Funk as a manager are able to open files that contain the word Cake.
    - **Active Directory Rights Management Services (AD RMS)** uses encryption and limits access to documents such as corporate emails, documents, and web pages. Companies can encrypt information and the policies embedded in the documents prevent the protected content from being decrypted except by specified people or groups, in certain environments, under certain conditions, and for certain periods of time.

- Microsoft improves all the services **regarding security:**
    - **Certificate Services** is available on both Server Core and Server with a GUI system, with Server Core being the more secure option because of its reduced attack surface. Because Certificate Services is an important security role, you should consider deploying it on a computer separate from other roles. You should also limit access to the server hosting Certificate Services so that only those directly responsible for performing administration tasks have access.
    - **File Server Resource Manager** and **SMB3.1.1** provide different security functionalities such as file classification/expiration/encryption and AES encryption. Associated with DAC and AD RMS, advanced access control and encryption can be enforced.

- Microsoft goes further in**:**
    - The **automation of the administration** since all the services currently are supporting PowerShell scripting and cmdlets. Apart Microsoft, PowerShell Gallery extends the functionality. Secure remote access is available for a wide range of server, host including nano server and server core.
    - **All the functionalities** (networks, hypervisors, containers, IoT, Cloud, …) **propose advanced security mechanisms**.

The bad(?) news

- Microsoft has a “poor” **communication that may be not sufficient to act against some ideological positions and obscurantism**.
- **Microsoft has no competitor**, making his position fragile since technical advances need competition.
- **Microsoft is still adopting available/research advances**. It continues to provide an industrial response that reuses the best practices and research results.
- **Microsoft does not address a “pure” microkernel approach**. May be Windows Server 2020 will provide a Chorus like approach in order to protect the different kernel functionalities.
  © Christian Toinard (an old and stupid computer scientist that worked and taught with the Chorus microkernel in the 2000’s)

Role Based Access Control

Role Based Access Control (RBAC) operationalizes the principle of **least privilege**.

The idea is instead of having all-powerful administrator accounts that can perform any task on any system, you instead parcel out administrative privileges limiting what an account can do, and where an account can do it.

For example, rather than giving help desk support staff the ability to reset the passwords of any user in the domain, you instead delegate them the right to only be able to reset the passwords of user accounts in a specific organizational unit. This way you allow the support staff to reset the passwords of a specific class of user account, for example those that are used by people in the Philosophy department, without giving support staff the ability to reset the passwords of domain administrator accounts.

You delegate privileges in Active Directory over **organizational units** using the **Delegation Of Control Wizard**, as shown in the figure below.

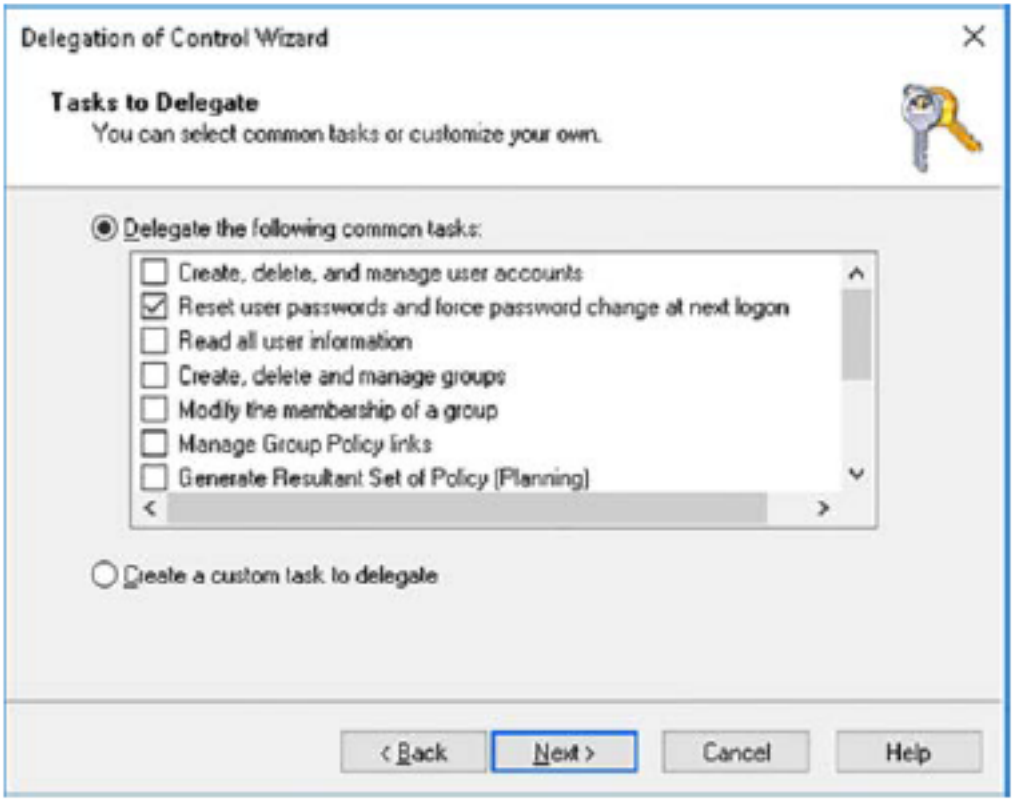

FIGURE Delegating the reset password right

Password policies

Account policies include both password policies and account lockout policies. You configure the **domain password and lockout policies** by configuring **the default domain GPO**. You can **override the domain policies for specific security groups or user accounts using fine-grained password policies**. You configure fine-grained password policies using **the Active Directory Administrative Center console**.

User rights

Rather than adding a user or service account to the local Administrators group, you should use **Group Policy to assign the specific rights** that a user account needs to perform the tasks it needs to perform.

Assign rights as close to the object as possible. For example, don't give a user account the ability to log on through Remote Desktop Services to all computers in an OU if the user account only needs remote desktop access to one or two computers.

Account security options

There are additional security options that you should consider configuring for highly privileged accounts. These include configuring the following settings:

- **Logon Hours**. Use this setting to configure **when users can use an account**. AD DS does not authenticate someone attempting to sign in outside of these hours. By default, accounts are configured so that users can sign in at all times. Consider configuring privileged accounts so that users can use them only at specific times.

- **Logon Workstations**. Use this setting to **limit the computers that a particular account can sign in to**. By default, users can use an account to sign in to any computer in the domain. Use this setting to ensure that users can use privileged accounts only on specific, specially configured administrative workstations, or certain servers.

- **Password Never Expires**. **Use this setting reluctantly** to configure this option for privileged accounts because it absolves the account from the domain password policy. You can use products such as Microsoft Identity Manager to assist with password management for privileged accounts.

- **Smart Card Is Required For Interactive Logon**. Use this setting to ensure that a smart card must be present for the account sign-in to occur. In high security environments, you should deploy smart cards and enable this option to **ensure that only an authorized person, who has both the smart card and the account credentials, can use the privileged** account.

- **Account Is Sensitive And Cannot Be Delegated**. Use this setting to **ensure that trusted applications cannot forward the account credentials** to other services or computers on the network. Enable this setting for highly privileged accounts.

- **This Account Supports Kerberos AES 256 Bit Encryption**. Use this setting to allow Kerberos AES 256-bit encryption. Where possible, you would **configure this option for privileged accounts** and have them use this form of Kerberos encryption over the AES 128-bit encryption option.

- **Account Expires**. Use this setting to configure an **expiration date for an account**. Configuring privileged accounts with expiration dates ensures that privileged accounts do not remain in AD DS after they are no longer in use.

When you join a computer to a domain for the first time, it creates a computer account in the Computers container. If you want to create an account in a specific container, you'll need to pre-stage the computer account and place it in the appropriate OU.

Computer accounts can be made members of the domain security group. This is because, by default, these services use the computer's credentials when interacting with resources on the network. This gives the computer account the rights and privileges assigned toany security group that the computer is a member of.

Computer accounts have automatically assigned passwords that are updated every 30 days. If a computer doesn't connect to a domain within 30

days, a new password is assigned the next time a connection is established. If you disable a computer account, the computer cannot connect to the domain and domain users are unable to sign onto the computer until the account is re-enabled.

Resetting a computer account removes the relationship between the computer and the domain. To fix this, you'll need to either join the computer back to the domain, or **reestablish the broken trust relationship using the following PowerShell command**, specifying the credentials of a member of the Domain Administrator group:

Test-ComputerSecureChannel -credential <domain>\<admin> -Repair

If you delete a computer account, it's necessary to rejoin the computer to the domain manually. When you delete a computer account, **all information associated with the account is removed from Active Directory**. You should only ever delete computer accounts once a computer has been decommissioned.

Service accounts

Service accounts allow services running on a computer to interact with the operating system as well as resources on the network. Windows Server 2016 uses **three types of built-in service accounts**, each of which is suitable for a specific set of circumstances. These accounts are as follows:

- **The Local System (NT AUTHORITY\SYSTEM)** account has privileges that are **equivalent to** those assigned to a user account that **is a member of the local Administrators group on the computer**. A service that uses this account can act by using the computer account's credentials when interacting with other resources on the network.

- **The Local Service (NT AUTHORITY\LocalService)** account has privileges that are **equivalent to** those assigned to **a user account that is a member of the local Users group** on the computer. A service that uses this account can access resources on the network without credentials. You use this account when a **service does not need to interact with resources on the network and does not need local Administrator privileges** on the computer on which it is running.

- **The Network Service (NT AUTHORITY\NetworkService)** account has privileges that are **equivalent to** those assigned to **a user account that is a member of the local Users group on the computer**. A service that uses **this account accesses resources on the network by using the computer account's credentials**.

In the past, when you've needed to create **a custom service account, you've probably created a user account** and then assigned it an appropriate set of permissions. The challenge with this type of account is password management. Many organizations configuring custom service accounts with passwords that never expire.

Now, a Group Managed Service Account (gMSA) is a special type of account that has AD DS manage its password. The password of a gMSA is updated every 30 days. You **don't need to know the password of a gMSA**, even when configuring a service to use that password because gMSA accounts provide you with a domain based service account identity without all the hassle of service account password management.

The requirements are 1) Client computers must run at least Windows 8, 2) You must create a key distribution services (KDS) root key for your domain and 3) At least one domain controller must be running Windows Server 2012 or later.

You **create a gMSA using the New-ADServiceAccount cmdlet**. When creating the account you specify a hostname as well as service principle names. For example, to create a new gMSA with the name called SYD-SVC1 that is associated with the hostname SYD-SVC1.adatum.com, run the command:

New-ADServiceAccount SYD-SVC1 -DNSHOSTNAME SYD-SVC1.adatum.com

gMSAs are stored in the Managed Service Accounts container, which you can view using Active Directory Administrative Center, as shown below :

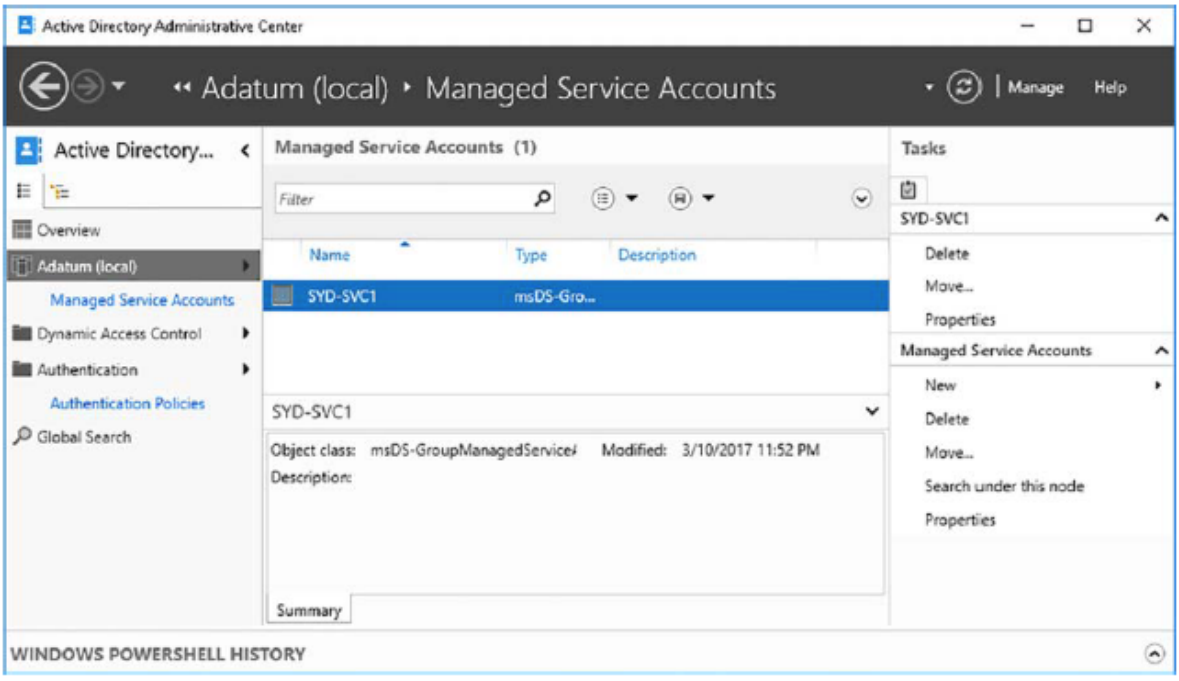

FIGURE Managed Service Accounts container

Once the gMSA is created, you need to **configure permissions for specific computers to be able to install and use the account**. The simplest way to do this is to create a security group, and then use the Set-ADServiceAccount cmdlet to assign the group permissions to the account.

Protected accounts

The Protected Users group, shown in the figure below, provides you with a method of **protecting highly privileged user accounts from being compromised**. It does this by blocking the use of less secure security and authentication options.

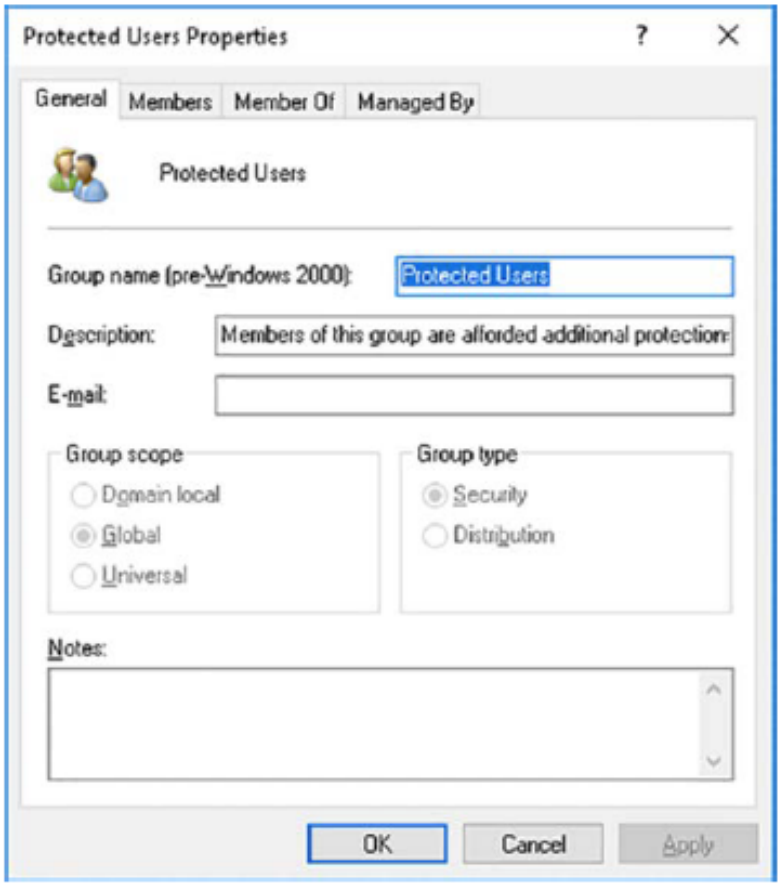

FIGURE Protected Users group

Accounts that are members of this group cannot use the following security options:

- Default credential delegation (CredSSP)
- Windows Digest
- NTLM – NTOWF
- Kerberos long term keys
- Sign-on offline

User accounts that are members of the Protected Users group cannot:

- Use NT LAN Manager (NTLM) for authentication.
- Use DES for Kerberos pre-authentication.
- Use RC4 cipher suites for Kerberos pre-authentication.
- Be delegated using constrained delegation.
- Be delegated using unconstrained delegation.
- Renew user tickets (TGTs) past the initial 240-minute lifetime.

Only user accounts should be added to the Protected Users group. **You should not add computer accounts or service accounts to this group. In secure environments, all privileged accounts should be members of this group**.

Authentication policies and silos

Authentication policy silos define relationships between the user, computer, and managed service accounts. **A user, computer, and managed service account can only belong to a single authentication policy silo**. This provides a more robust method, beyond configuring logon restrictions, and restricting which accounts can access specific servers in your environment. Accounts in an authentication policy silo are associated with a silo claim. For example, you can configure accounts so that only accounts that are associated with a specific silo claim are able to access particularly sensitive servers. With this, only accounts that are associated with a Certificate Services silo claim are able to sign on to a computer that has the Active Directory Certificate Services role installed.
Authentication policies allow you to configure settings, such as TGT lifetime and access control conditions, which specify conditions that must be met before a user can sign in to a computer.
For example, you might configure an authentication policy that specifies a TGT lifetime of 120 minutes and limit a user account so that users can only use it with specific devices, such as privileged access workstations.
You configure Authentication Policies and Authentication Policy Silos using Active Directory Administrative Center, as shown below.

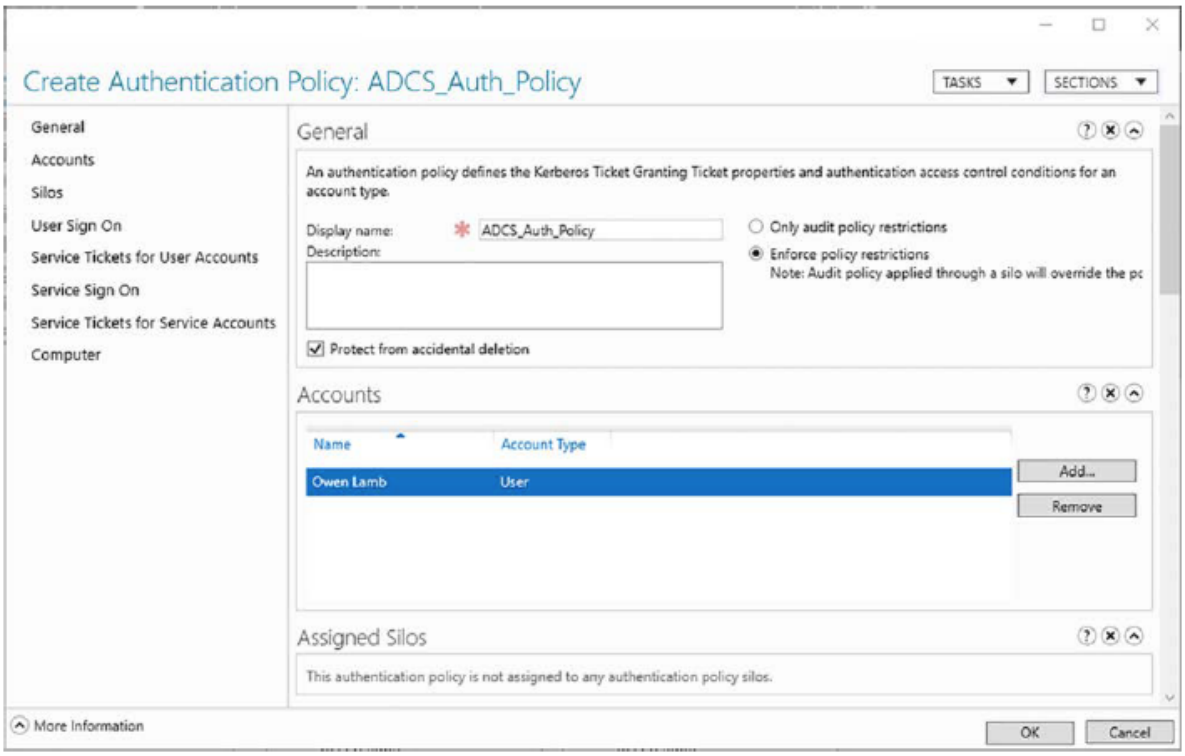

FIGURE Authentication policies

Credential guard

Credential Guard is a feature new to Windows Server 2016 and Windows 10 that allows you to leverage virtualization-based security to isolate secrets, such as cached user credentials, **in a special separate virtualized operating system**. The special separate virtualized operating system is configured so that only specific processes in the host operating system can access this secret data. The processes running in the separate virtualized operating system are termed trustlets.

Credential Guard is primarily a response to pass-the-hash or pass-the-ticket attacks. Should a host that has credential guard be compromised by an attacker, that attacker won't be able to successfully run a pass-the-hash attack tool to extract cached credentials and then use them to access other computers on the network.

Credential guard includes the following features and solutions:

- Stores derived domain credentials in a virtualized environment that is protected from the running operating system.
- You can manage Credential Guard by using Group Policy, Windows Management Instrumentation (WMI), or Windows PowerShell.

Credential Guard does not allow:

- Unconstrained Kerberos delegation
- NT LAN Manager version 1 (NTLMv1)
- Microsoft Challenge Handshake Authentication Protocol (MS-CHAPv2)
- Digest Authentication
- Credential Security Support Provider (CredSSP)
- Kerberos DES encryption

Credential Guard can be used in conjunction with the Protected Users group in a layered approach to the protection of highly privileged accounts. The Protected Users group remains useful because your organization may not have computers that support Credential Guard. **You can deploy Credential Guard only on computers that meet certain hardware requirements**.

Credential Guard has the following requirements:

- Windows Server 2016 or Windows 10 Enterprise
- UEFI firmware version 2.3.1 or higher
- Secure Boot
- Intel VT-x or AMD-V virtualization extensions
- Second Level Address Translation
- x64 processor architecture
- A VT-d or AMD-Vi IOMMU input/output memory management unit
- TPM 1.2 or 2.0
- Secure firmware update process
- Firmware updated to support Secure MOR implementation

To enable Credential Guard on an appropriately configured computer, you need to configure the Turn On Virtualization Based Security policy, which is located in the Computer Configuration\Administrative Templates\System\Device Guard node of a GPO. This is the same policy that you also use to configure Device Guard.

While configuring this policy, you must first **set the policy to Enabled**, and then you must set the **platform security level to either Secure Boot or to Secure**

Boot and DMA Protection. Secure Boot with DMA Protection ensures that Credential Guard is used with Direct Memory Access protection.
Once this is done, you need to then **set the Credential Guard Configuration option to Enabled with UEFI lock, or Enabled without lock**. If you set the Enabled with UEFI lock, credential guard cannot be remotely disabled and can only be disabled by having someone with local Administrator privileges sign on and disable credential guard configuration locally. The Enabled Without Lock option allows Credential Guard to be remotely disabled.

Virtual Secure Mode

It includes Credential Guard and Device Guard. VSM is a secure execution environment in which **secrets and keys are maintained** and critical security processes run as Trustlets (small trusted processes) **in a secure virtualized partition**. Moreover, trusted binaries and files form **a "Trusted Computation Based" preventing malicious binaries and files from being executed**.

Just Enough Administration

Just Enough Administration (JEA) allows you to implement Separation of Privileges and Least Privileges through Windows PowerShell remoting. JEA allows you to specify **which Power-Shell cmdlets and functions can be used when connected to a specific endpoint**. You can go further and specify which parameters within those cmdlets and functions are authorized and even specify which values can be used with those parameters.
For example, you could create a JEA endpoint where a user is able to run the Restart-Service command, but only where the Name parameter is set to DHCPServer. This would allow the user to restart the DHCPServer on the computer they connected to, but not restart any other service on the computer.

JEA endpoints can leverage virtual accounts. This means that activities performed on the computer through the endpoint use a special temporary virtual account rather than the user's account. This temporary virtual account has local Administrator privileges, but is constrained to only using the cmdlets, functions, parameters, and values defined by JEA. The benefits of this include:

- **The user's credentials are not stored on the remote system**. If the remote system is compromised, the user's credentials are not subject to credential theft and cannot be used to traverse the network to gain access to other hosts.
- **The user account used to connect to the endpoint does not need to be privileged**. The endpoint simply needs to be configured to allow connections from specified user accounts.
- **The virtual account is limited to the system on which it is hosted**. The virtual account cannot be used to connect to remote systems. Attackers cannot use a compromised virtual account to access other protected servers.
- **The virtual account** has local administrator privileges, but **is limited to performing only the activities defined by JEA**. **You have the option** of configuring the virtual account with the membership of a group other than the local administrators group, **to further reduce privileges**.

JEA works on Windows Server 2016 and Windows 10, version 1511 or later directly. It also functions on previous versions of Windows client and server as long as Windows Management 5.0 is installed. Virtual accounts are not available when using JEA on Windows 7 or Windows 2008 R2 and all activities are performed using the privileges assigned to the account connecting to the JEA endpoint.

JEA drawbacks
The biggest drawback to JEA is the **amount of time that it takes to configure**. You'll need to customize each JEA endpoint, which requires that you understand exactly which cmdlets, functions, parameters, and values are required to perform specific tasks. The other drawback of JEA is that it relies upon administrative tasks being performed using the command line. While most Windows Server administrators are comfortable performing tasks using PowerShell, some may be reluctant to be placed in a position where they can only use PowerShell for specific tasks rather than falling back on the RSAT consoles.

Role-capability files

A role-capability file is **a special file that allows you to specify what tasks can be performed when connected to a JEA endpoint**. Only tasks that are explicitly allowed in the role-capability file can be performed. You can create a new blank role-capability file by using the *New-PSRoleCapabilityFile* cmdlet. Role-capability files use the *.psrc* extension. For example, to create a new role capability file that allows someone to manage a DNS server, run the command:

New-PSRoleCapabilityFile -Path .\DNSOps.psrc

Once the .psrc file is created, you edit the role capability file and add the cmdlets, functions, and external commands that are available when a user is connected to the endpoint.

You can edit a role capability file in PowerShell ISE as shown below.

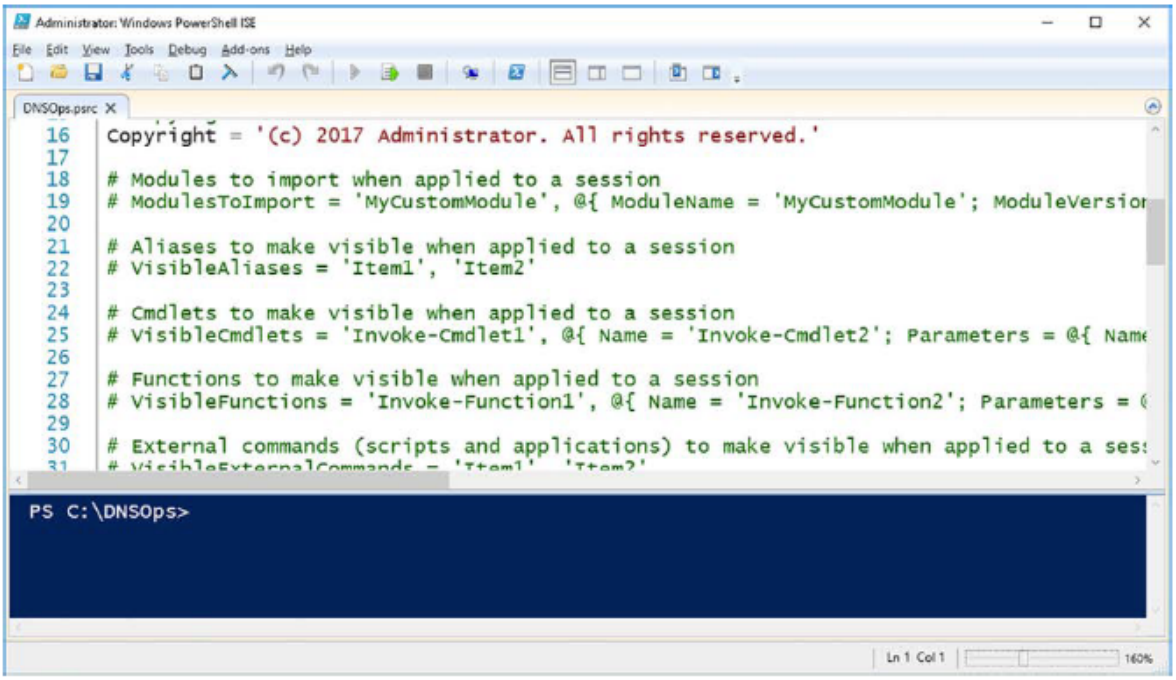

FIGURE JEA role capability file

The following table describes the options available in a role capability file.

Capability	Description
ModulesToImport	JEA auto-loads standard modules, so you probably don't need to use this unless you need to import custom modules
VisibleAliases	Specify which aliases to make available in the JEA session. Even if an aliased cmdlet is available, the alias won't be unless it's here
VisibleCmdlets	Lists which Windows PowerShell cmdlets are available in the session. You can extend this by allowing all parameters and parameter values to be used or you can limit cmdlets to particular parameters and parameter values. For example, if you wanted to allow the Restart-Service cmdlet to only be used to restart the DNS service, use the following syntax: `VisibleCmdlets = @{ Name = 'Restart-Service'; Parameters = @{ Name='Name'; ValidateSet = 'DNS'}}`
VisibleFunctions	This field lists which Windows PowerShell functions are available in the session. You can choose to list functions, allowing all parameters and parameter values to be used, or you can limit functions to particular parameters and parameter values. For example, if you wanted to allow the Add-DNSServerResourceRecord, Get-DNSServerResourceRecord, and Remove-DNSServerResource functions to be used, you would use the following syntax: `VisibleFunctions = 'Add-DNSServerResourceRecord', 'Get-DNSServerResourceRecord','Remove-DNSServerResourceRecord'`
VisibleExternalCommands	This field allows users who are connected to the session to run external commands. For example, you can use this field to allow access to c:\windows\system32\whoami.exe so that users connected to the JEA session can identify their security context by using the following syntax: `VisibleExternalCommands = 'C:\Windows\System32\whoami.exe'`
VisibleProviders	This field lists Windows PowerShell providers that are visible to the session
ScriptsToProcess	This field allows you to configure Windows PowerShell scripts to run automatically when the session is started
AliasDefinitions	This field allows you to define Windows PowerShell aliases for the JEA session
FunctionDefinitions	This field allows you to define Windows PowerShell functions for the JEA session
VariableDefinitions	This field allows you to define Windows PowerShell variables for the JEA session
EnvironmentVariables	This field allows you to specify environment variables for the JEA session
TypesToProcess	This field allows you to configure Windows PowerShell type files to load for the JEA session
FormatsToProcess	This field allows you to configure Windows PowerShell formats to load for the JEA session
AssembliesToLoad	This field allows you to specify which assemblies to load for the JEA session

Session-configuration files

Session-configuration files **determine which role capabilities are mapped to specific security groups**. For example, if you wanted to allow only members of the CONTOS\DNSOps security group to connect to the JEA endpoint that is defined by the DNSOps role capability file, you would configure this in the session configuration file.

For example, to create a new session configuration file for the DNSOps role, run the following command:

New-PSSessionConfigurationFile -Path .\DNSOps.pssc -Full

Session-configuration files have elements described in the following table:

Field	Explanation
SessionType	This field allows you to configure the session's default settings. If you set this to RestrictedRemoteServer, you can use the Get-Command, Get-FormatData, Select-Object, Get-Help, Measure-Object, Exit-PSSession, Clear-Host, and Out-Default cmdlets. The session execution policy is set to RemoteSigned. Example: `SessionType = 'RestrictedRemoteServer'`
RoleDefinitions	You use the RoleDefinitions entry to assign role capabilities to specific security groups. These groups do not need to have any privileges and can be standard security groups. Example: `RoleDefinitions =@{'CONTOSO\DNSOps' = @ {RoleCapabilities='DNSOps'}}`
RunAsVirtualAccount	When enabled, this field allows JEA to use a privileged virtual account created just for the JEA session. This virtual account has local Administrators privileges on member servers, and is member of the Domain Admins group on a Domain Controller. Use this option to ensure that credentials are not cached on the server that hosts the endpoint. Remember that you can configure the virtual account to be a member of groups other than the local Administrators group
TranscriptDirectory	This field allows you to specify the location where JEA activity transcripts are stored
RunAsVirtualAccountGroups	If you do not want the virtual account to be a member of the local Administrators group (or Domain Admins on a domain controller) you can instead use this field to specify the groups in which the virtual account is a member

JEA endpoints

A JEA endpoint is a Windows PowerShell endpoint that you configure so that only specific authenticated users can connect to it. When those users do connect, they only have access to the Windows PowerShell cmdlets, parameters, and values defined by the appropriate session-configuration file that links security groups and role capabilities. When you use endpoints with virtual accounts, the actual activity uses the virtual account. **This means that no domain-based administrative credentials are stored on the server that hosts the endpoint**.

A server can have multiple JEA endpoints, and **each JEA endpoint can be used for a different administrative task**. For example, a DNSOps endpoint to

perform DNS administrative tasks and an IISOps endpoint to perform Internet Information Server related administrative tasks. Users do not have to have privileged accounts. **Once connected, users have the privileges assigned to the virtual account configured in the session-configuration file**.
For example, to create the endpoint DNSOps using the *DNSOps.pssc* session-configuration file, use the following command and then restart the WinRM service:
Register-PSSessionConfiguration -Name DNSOps -Path .\DNSOps.pssc
You can use the *Get-PSSessionConfigurationFile* to determine which endpoints are present on a computer.
A user wanting to connect to a JEA session endpoint uses the Enter-PSSession cmdlet with the ConfigurationName parameter. For example, to connect to the DNSOps JEA endpoint on server MEL-DNS1, you would use the command:
Enter-PSSession -ComputerName MEL-DNS1 -ConfigurationName DNSOps

Privileged Access Management

Privileged Access Management (PAM), also known as Just In Time (JIT) Administration, works on the concept that users can be granted permission for a finite amount of time rather than permanently. It uses temporary membership of a security group that has been delegated privileges, rather than permanent membership of a security group that has been delegated privileges, to accomplish this goal.

For example, a user named Rooslan requires that his password be reset and submits a ticket to Oksana, who works at the service desk, to have Oksana perform this task. In a traditional environment, Oksana would use an account that is a member of a security group that has been delegated the reset-password privilege. In a PAM environment, Oksana requests the privilege to change Rooslan's user account password. The PAM process adds Oksana's user account temporarily to a group that has been delegated the reset password privilege on Rooslan's user account. Oksana then reset's Rooslan's user account password. After a certain period of time has elapsed, the PAM process removes Oksana's account from this group, removing the reset-password privilege for Rooslan's user account from Oksana's user account.

Containers

Windows Server supports two container types: the Windows Server container and Hyper-V containers.
Windows 10 supports only Hyper-V containers. Windows Server 2016 supports Windows Server containers and Hyper-V containers.

Windows Server Containers
Windows Server Containers provide an isolated application execution environment. This is accomplished through process and namespace isolation. Windows Server containers share a kernel with all other containers running on the container host. Windows Server containers also share a kernel with the container host. This is why you can't run Windows Server containers directly on Windows 10.

Hyper-V Containers
A Hyper-V container is a highly optimized virtual machine that also provides an isolated application execution environment. **Hyper-V containers don't share the kernel with the container host, nor other containers on the same container host**. Hyper-V containers require that the Hyper-V role be installed on the container host. If the container host is a Hyper-V virtual machine, you will need to enable nested virtualization. By default, a container will start as a Windows Server container. You can start a container as a Hyper-V container by using the *--isolation=hyperv* option.
For example, the following commande create a Hyper-V container from the microsoft/windowsservercore image:
Docker run -it --isolation=hyperv microsoft/windowsservercore cmd

Shielded VMs

It is **a way to secure VM even from administrators**.
Shielded VMs provide protection for the data and state of the VM against inspection, theft, and tampering from administrator privileges.
Shielded VMs work for Generation 2 VMs that provide the necessary secure startup, UEFI firmware, and virtual Trusted Platform Module (vTPM) 2.0 support required.

A new **Host Guardian Service** instance is deployed in the environment, which stores the keys required for an approved Hyper-V host that can prove its health to run shielded VMs.
A shielded VM provides the following benefits:

- BitLocker encrypted drives (utilizing its vTPM)
- A hardened VM worker process (VMWP) that encrypts live migration traffic in addition to its runtime state file, saved state, checkpoints, and even Hyper-V Replica files
- No console access in addition to blocking Windows PowerShell Direct, Guest File Copy Integration Components, and other services that provide possible paths from a user or process with administrative privileges to the VM

How is this security possible? First, it requires **an attestation that the Hyper-V host has not been compromised** before shielded VM is enforced.
This attestation can happen in one of two ways:

1) The preferred way is by using the TPM 2.0 that is present in the Hyper-V host. Using the TPM, the boot path of the server is assured, which guarantees no malware or root kits are on the server that could compromise the security. **The TPM secures communication to and from the HGS attestation service**.
2) For hosts that do not have a TPM 2.0, an **alternate Active Directory–based attestation** is possible; It does not provide the same levels of assurance and protection.

After the attestation, the **host receives a health certificate** from the attestation service on the HGS. The keys are encrypted during transmission and can be decrypted only within a protected enclave that is the Virtual Secure Mode (VSM) including Credential Guard. VSM is a secure execution environment in which secrets and keys are maintained and critical security processes run as Trustlets (small trusted processes) in a secure virtualized partition.

The Windows operating system, even the kernel, has no access to VSM. Only safe processes (Trustlets) that are Microsoft signed are allowed to cross the "bridge" to access VSM. A vTPM Trustlet is used for each VM, separate from the rest of the VM process, which runs in a new type of protected VM worker process. This means that there is **no way to access the memory used to store these keys, even with complete kernel access**.