**TP1 - Installation d’un serveur Windows**

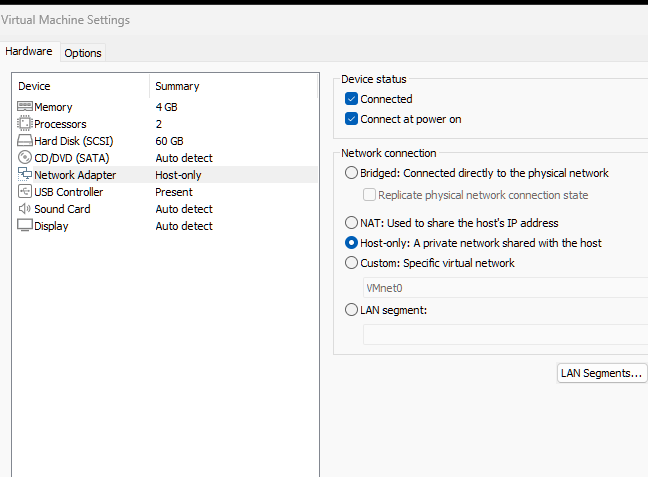
Ifrim Vasile-Alexandru M2SSI

1. Installez Virtualbox et ses extensions. Téléchargez Windows Server 2012 R2 (version d’essai - langue anglaise).

For the following TP I chose to use VMWare. I downloaded a Windows Server 2012 R2 from <https://www.microsoft.com/en-us/evalcenter/download-windows-server-2012-r2>.

1. Créez un environnement virtuel pour Windows sur Virtualbox. Documentez vos choix concernant les ressources attribuées à la machine virtuelle en vous appuyant sur des ressources officielles. Expliquez votre choix de configuration Réseau.

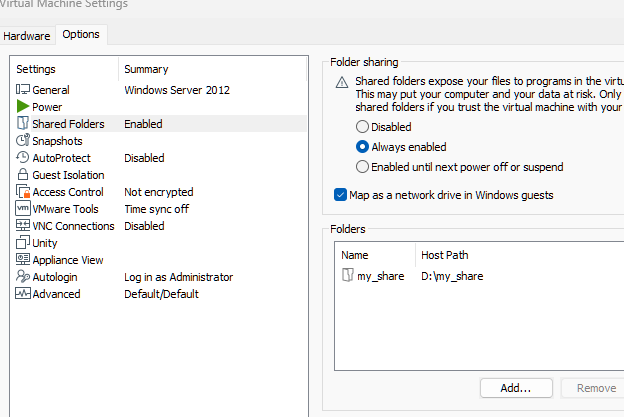
According to <https://www.techtarget.com/searchwindowsserver/tip/Be-aware-of-essential-Windows-Server-2012-hardware-requirements>, Windows Server 2012 R2 has the following minimum requirements: single 1.4GHz 64-bit processor, 512 MB RAM, 32 GB storage and 10/100 Mbps Ethernet network connection. For more performance, I can spare more than this even if a bit redundant. More importantly is the network configuration: I had issues with the adapter in bridged mode on the eduroam network – no internet access, default Windows automatic ip address. So, whenever internet access was needed by the guest machine, i switched to NAT. For anything else, a Host-only network was used for the ability to access from host to guest ( and visibility between guests).



1. Montez le disque téléchargé sur la VM et lancez la machine virtuelle. Installez le serveur Windows Server 2012 R2 avec Interface Graphique
2. Créez un compte administrateur. Justifiez votre choix de nommage. A la fin de l’installation, effectuez une snapshot de la machine.

I chose to just name it Administrator, but for a secure name common and descriptive names should be avoided ( Admin, Administrator, root, srvAdmin, srv\_mng) to obscure the fact that that account has any privileges more than your next user, making it hard for attackers to target it. Also, for any server management consistent naming conventions should be used.

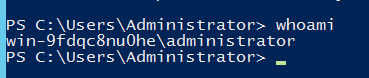
1. Créez un répertoire d’échange (shared folder ) entre votre machine hôte et votre machine virtuelle. Assurez-vous que vous êtes bien en mesure d’échanger des données entre votre hôte et la VM.



1. Ouvrez l’intepréteur de commande par défaut de Windows. Identifiez les commandes adaptées qui permettent d’afficher les informations suivantes :

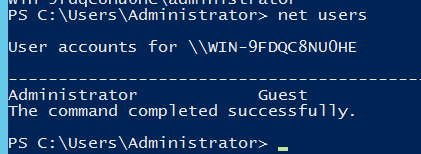
* l’utilisateur courant

>whoami



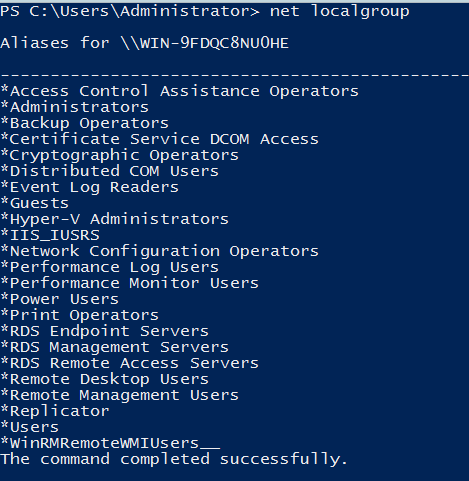
* la liste des utilisateurs locaux

>net users



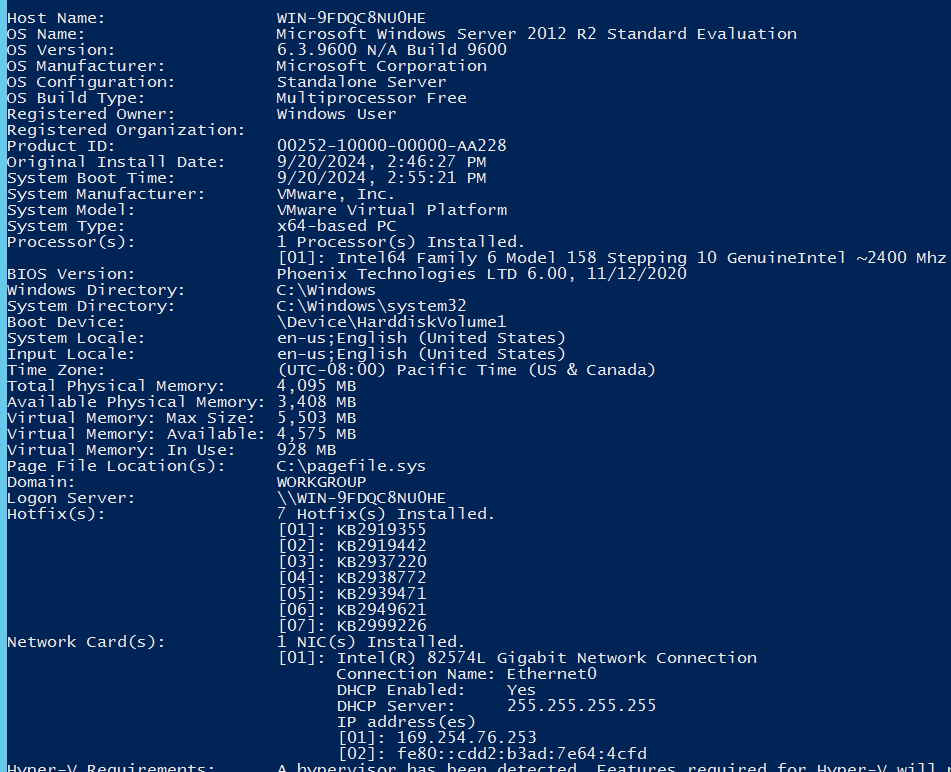
* la liste des groupes locaux

>net localgroup



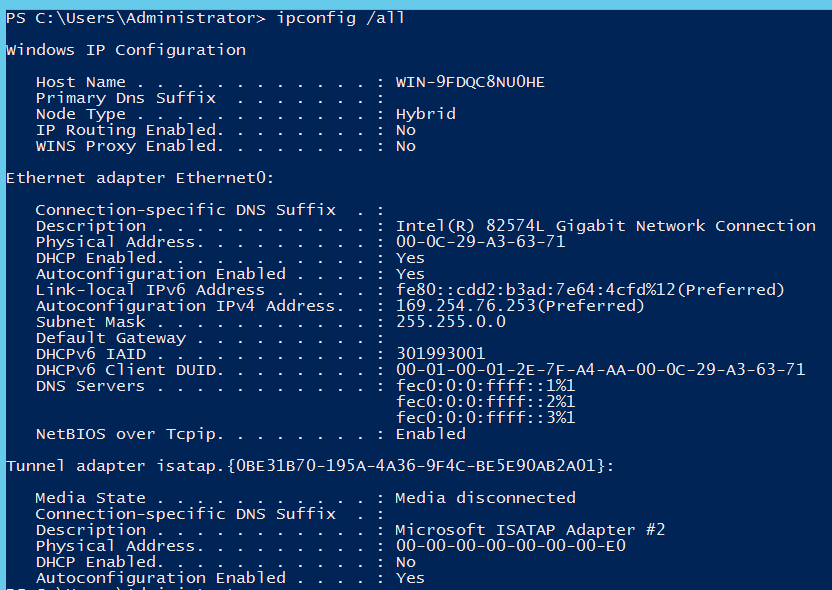
* la version du système

>systeminfo



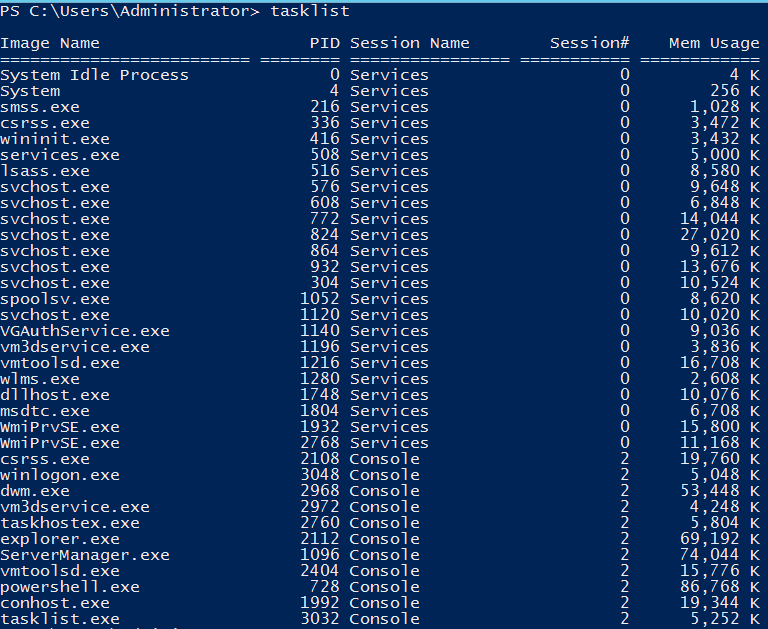
* informations détaillées sur les interfaces réseau configurées

>ipconfig /all



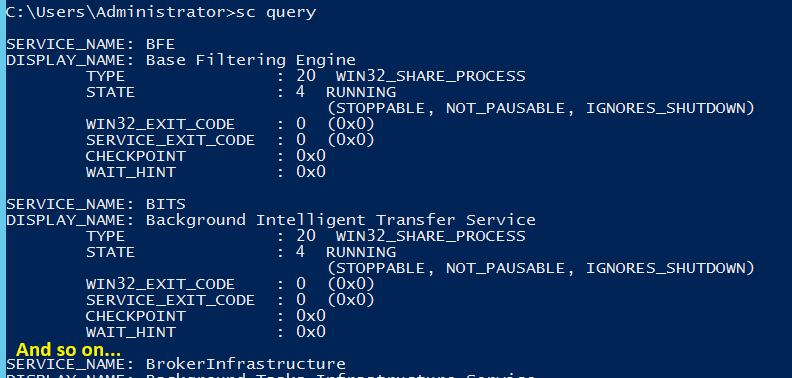
* la liste des processus en cours d’exécution

>tasklist



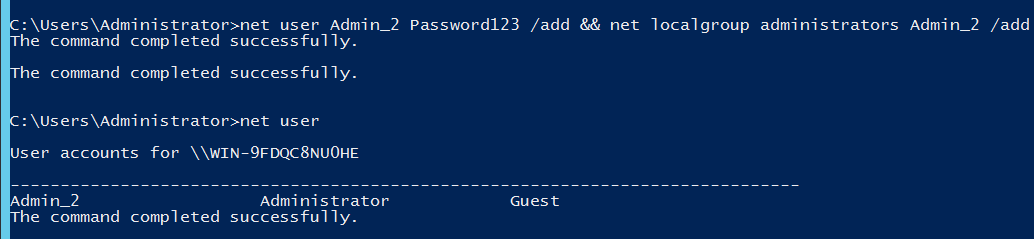
* la liste des services en cours d’exécution

>sc query



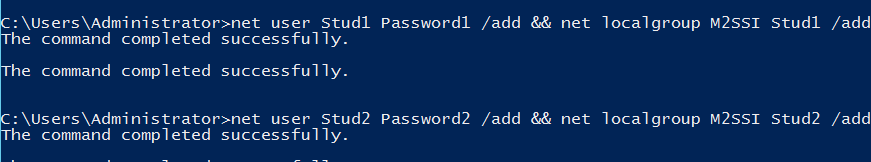
1. Identifiez 3 services en cours d’exécution sur le serveur (au choix, services vous semblant intéressants). Décrire ces services, leurs usages. Identifiez les documentations officielles respectives, utiles pour un durcissement de ces services.
2. Créez via l’instance 4 comptes supplémentaires sur le système:

* ◦ 1 compte administrateur supplémentaire

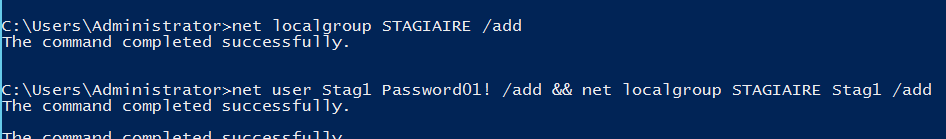


* 2 comptes standards au sein d’un groupe M2SSI





* un compte standard dans le groupe STAGIAIRE au mot de passe Password01!

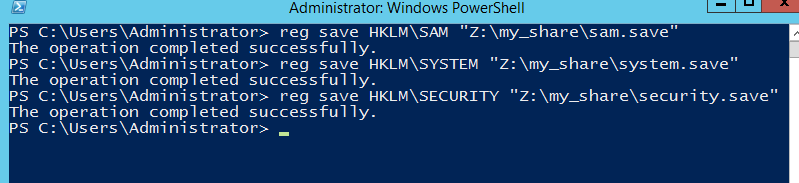


1. Où sont stockés les comptes et leurs secrets configurés sur le système de fichier? Proposez 2 méthodes différentes pour récupérer ces fichiers pour une analyse à froid. Décrivez la démarche de récupération. Identifiez les noms d’utilisateurs et leur mot de passe stocké au sein de ces données.

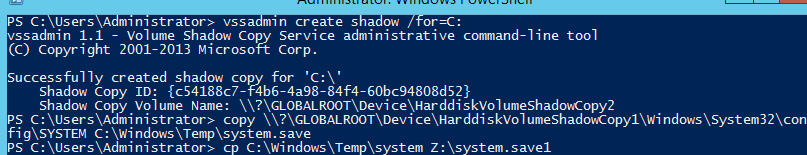
<https://www.thehacker.recipes/ad/movement/credentials/dumping/sam-and-lsa-secrets>

* Methode 1 – Live Windows

Copy from reg:



Or do a shadow copy of C:\Windows\System32\config:

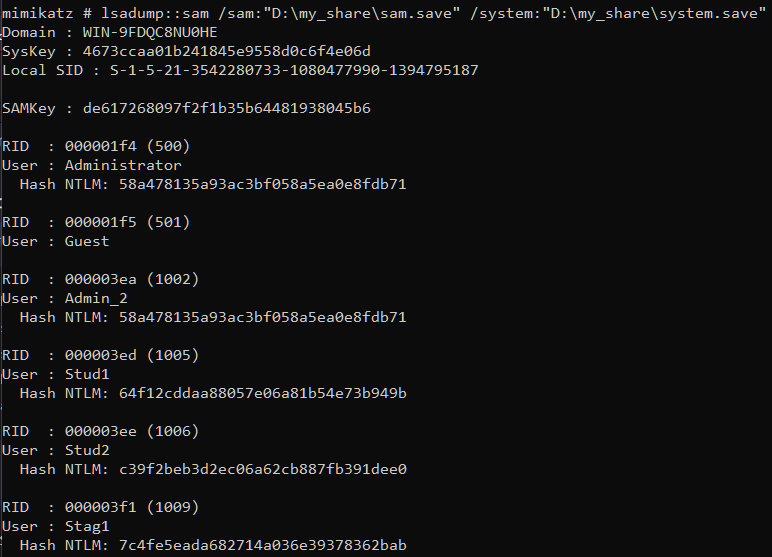


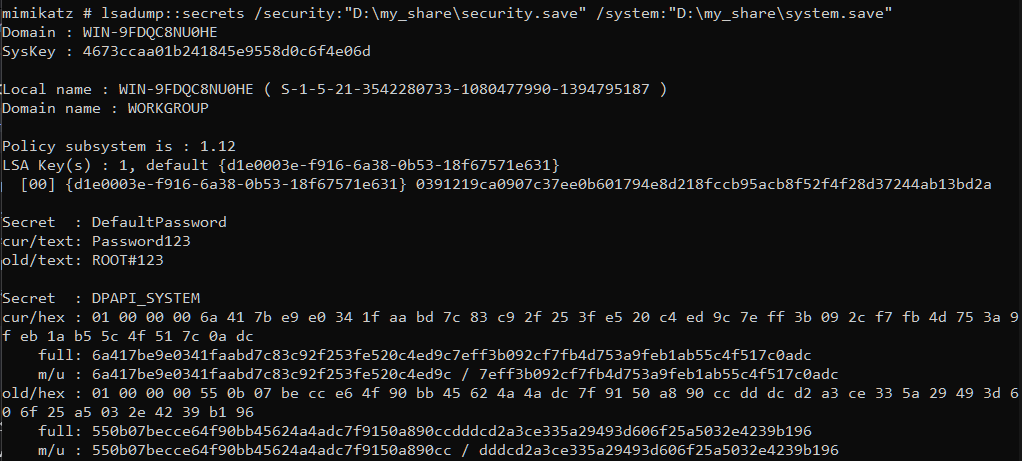
* Methode 2 - Offline System Access

Bootable Live CD/USB with a Linux -> Navigate to C:\Windows\System32\config -> Copy the SAM, SYSTEM and SECURITY files.

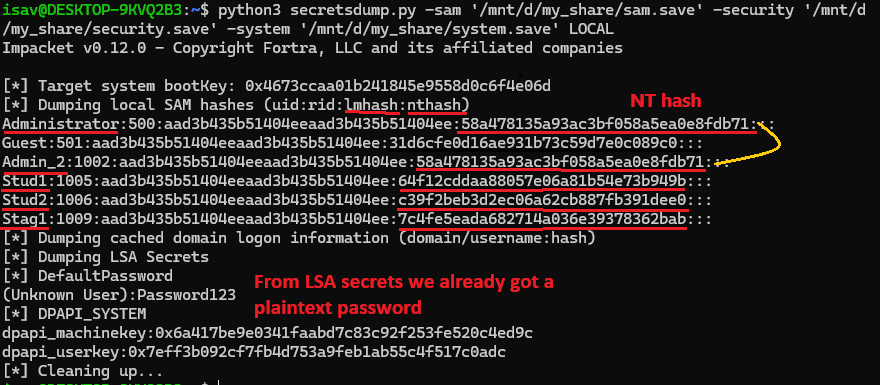
In Windows environments, passwords are stored in a hashed format in registry hives like SAM (Security Account Manager, stores locally cached credentials – SAM secrets: LM or NT hashes) and SECURITY (stores domain cached credentials – LSA secrets: plaintext, LM or NT hashes, etc). SYSTEM contains enough info to decrypt SAM secrets and LSA secrets. For now, we are only interested in locally cached credentials.

Mimicatz dump of secrets [<https://github.com/gentilkiwi/mimikatz/releases>]:

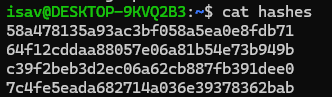




Impacket’s secretsdump dump:

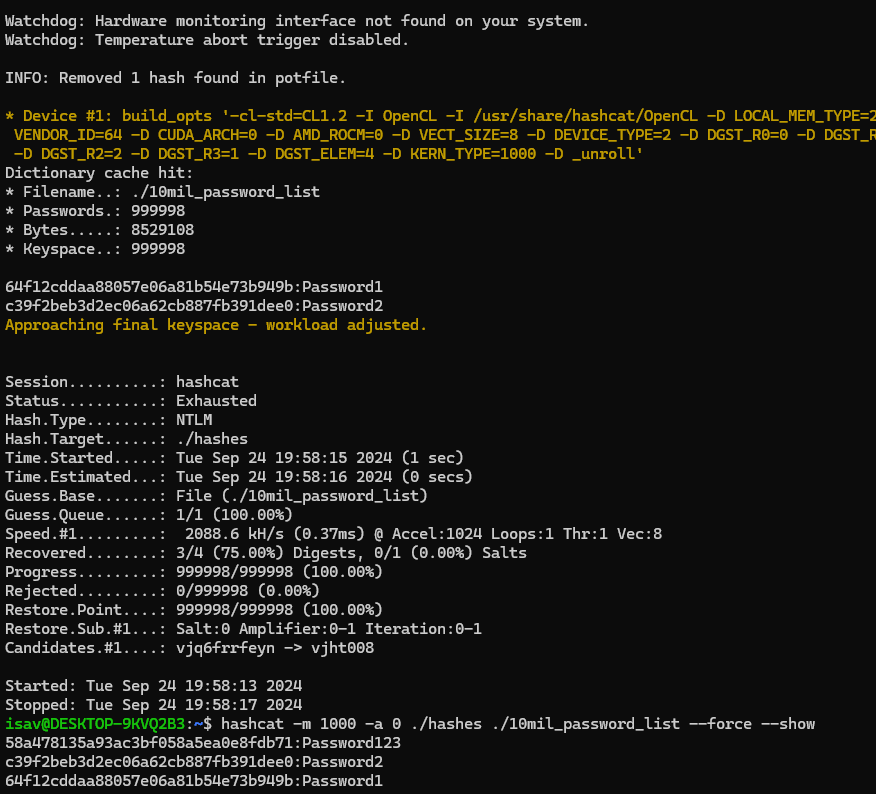


We paste these NTLM hashes in a ‚hashes’ input file.



Time to crack the hashes with hashcat [<https://tools.kali.org/password-attacks/hashcat>].  
>hashcat -m 1000 -a 0 ./hashes ./10mil\_password\_list –force

In this command, -m stands for hash mode(e.g, 1000 stands for NTLM hash), -a stands for attack mode (e.g., 0 stands for straight attack); ‚10mil\_password\_list’ is a dictionary of weak passwords found on the Internet.



Dictionary cracking of weak passwords through Hashcat

In summary, we have exfiltrated the registry hives that contain credentials and important information for decryption so we can do a cold analysis on a separate machine. With the help of tools for dumping, we got a table of users <-> hashes, and with hashcat we managed to crack the hashes, resulting in plaintext passwords.

1. Passwords are stored in a cryptographic format - NTLM hashes. The NTLM hash is encoded by taking the user’s password and converting it into a 16-byte key using an MD4 hash function. This key is divided into two halves of 8 bytes each, which are used as input to three rounds of DES encryption to generate a 16-byte output that represents the NTLM hash. More on this, <https://www.vaadata.com/blog/understanding-ntlm-authentication-and-ntlm-relay-attacks/#what-is-ntlm>.

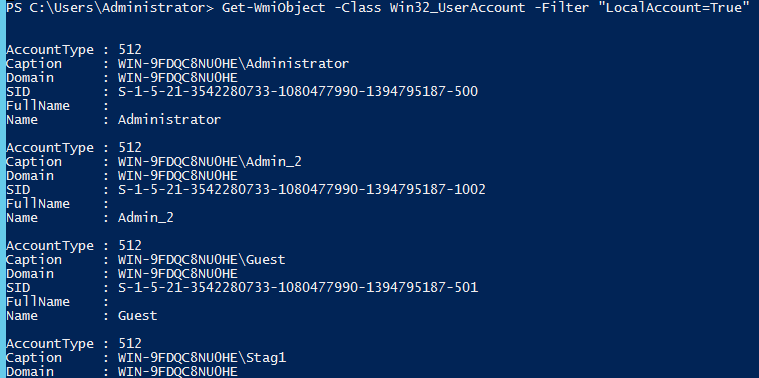
NTLM lack **salting** and are vulnerable to **brute-force attacks** and **rainbow tables**. With modern hardware, NTLM hashes can be cracked relatively quickly, especially if the passwords are weak or common.

1. PowerShell is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework. PowerShell runs on Windows, Linux, and macOS [[learn.microsoft.com/en-us/powershell/scripting](https://learn.microsoft.com/en-us/powershell/scripting/overview?view=powershell-7.4)].

* Lister les comptes sur le systeme

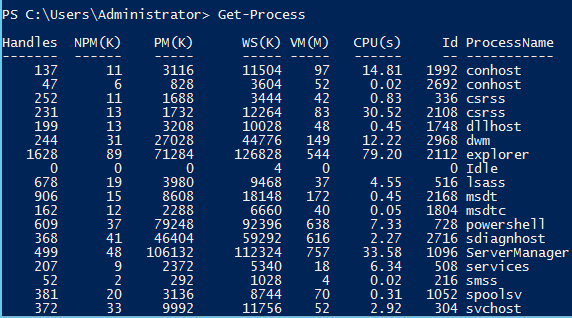
>Get-WmiObject -Class Win32\_UserAccount -Filter "LocalAccount=True"

[[superuser.com/questions/1718616/list-all-user-accounts-with-an-active-presence-on-the-local-device](https://superuser.com/questions/1718616/list-all-user-accounts-with-an-active-presence-on-the-local-device)]



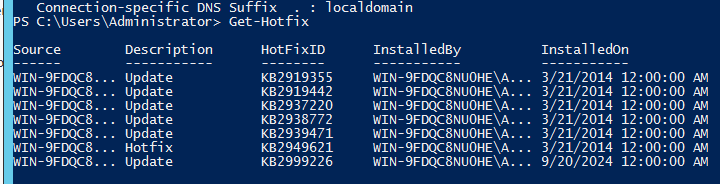
* Lister les processus en cours d’execution

>Get-Process



* Lister les mises a jour installees sur votre serveur

>Get-Hotfix

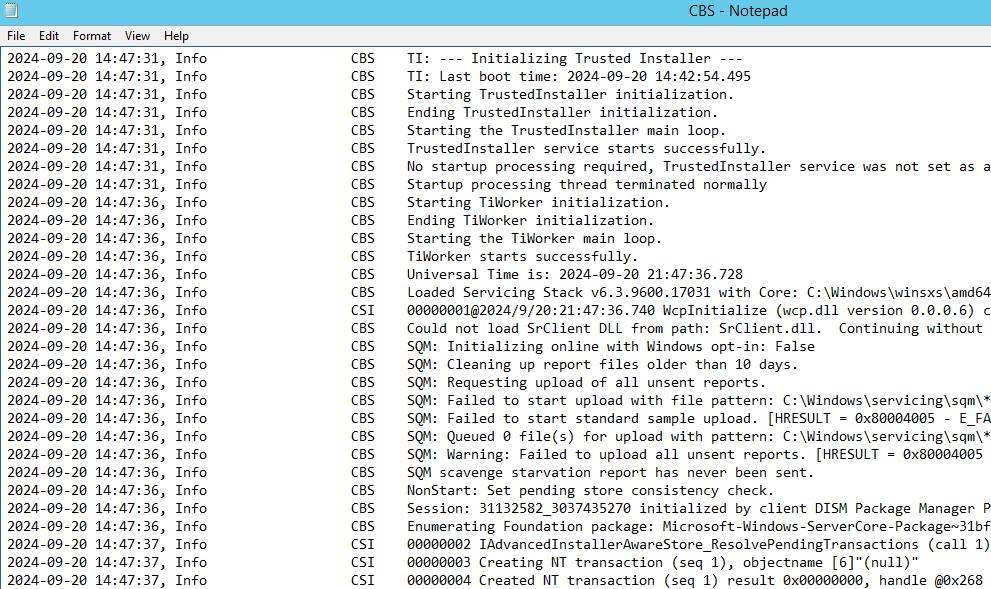


1. Proposez une stratégie technique pour vérifier par analyse à froid que le système est à jour dans ses correctifs de sécurité.

The documentation found at [https://learn.microsoft.com/en-us/windows/deployment/ update/windows-update-logs](https://learn.microsoft.com/en-us/windows/deployment/%20update/windows-update-logs) suggests to us that for our Windows Server 2012 R2 machine, we are interested is just the following file:

| **Log file** | **Location** | **Description** | **When to use** |
| --- | --- | --- | --- |
| CBS.log | %systemroot%\Logs\CBS | This log provides insight on the update installation part in the servicing stack. | To troubleshoot the issues related to Windows Update installation. |

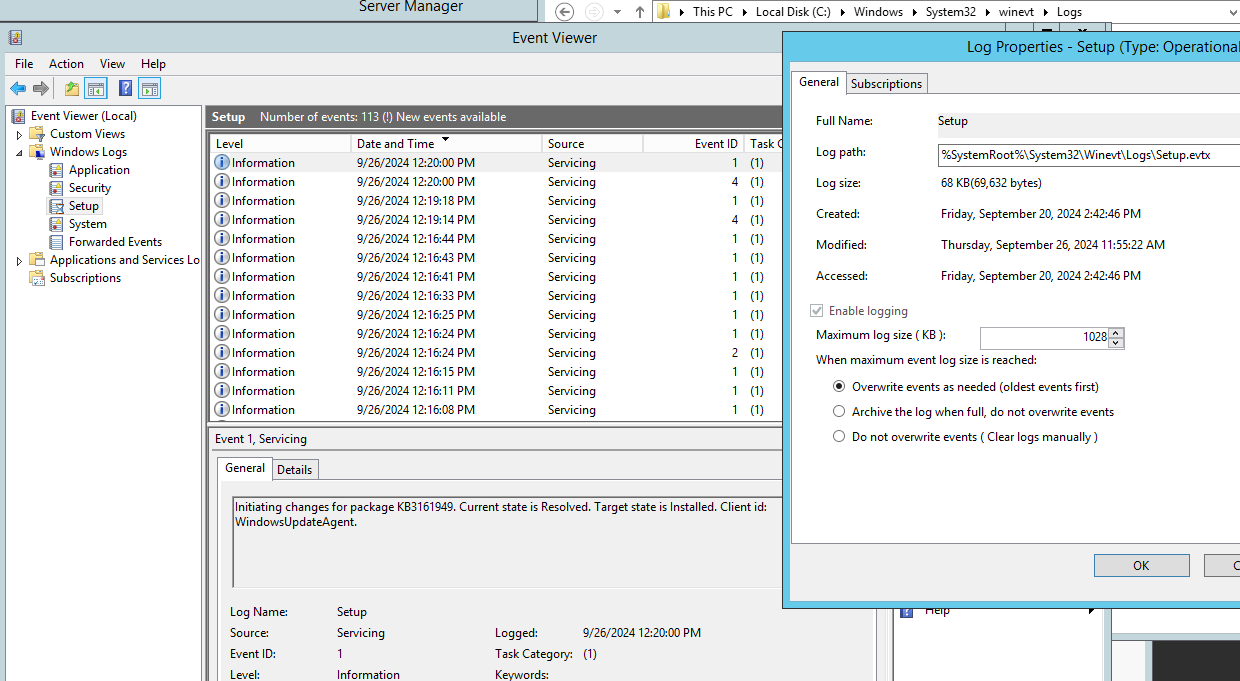
To obtain it, we can either boot from a live USB and copy our file or use shadow copy (a short summary/example of the steps is found in exercise 9).



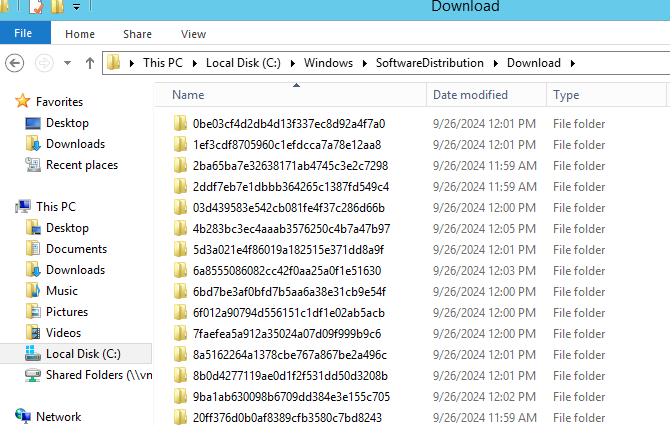
Example of CBS.log

Theoretically, we would cross-referrence the information gathered with [Microsoft Security Update Guide](https://msrc.microsoft.com/update-guide/en-US). For newer versions of Windows, we would also check the C:\Windows\Logs\ WindowsUpdate\windowsupdate.log.

In my exercise, after starting an update and looking through EventViewer -> Windows Logs, I could find that update events are logged in the Setup category. So another file of interest is located at C:\System32\Winevt\Logs\Setup.evtx. Also, temporary files for updates are stored at C:\Windows\SoftwareDistribution\Download.



EventViewer -> Windows Logs -> Setup : events for any setup/update operation



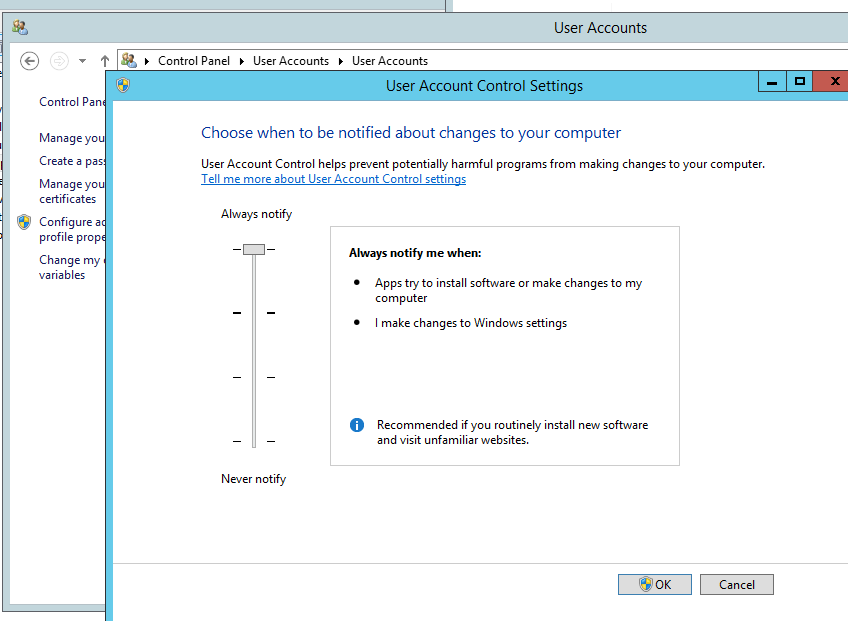
Temporary files for updates are stored in %WINDIR%\SoftwareDistribution\Download

1. Qu’est ce que l’UAC? Comment est-il configuré par défaut ? Quelles sont vos recommandations ? Appliquez-les.

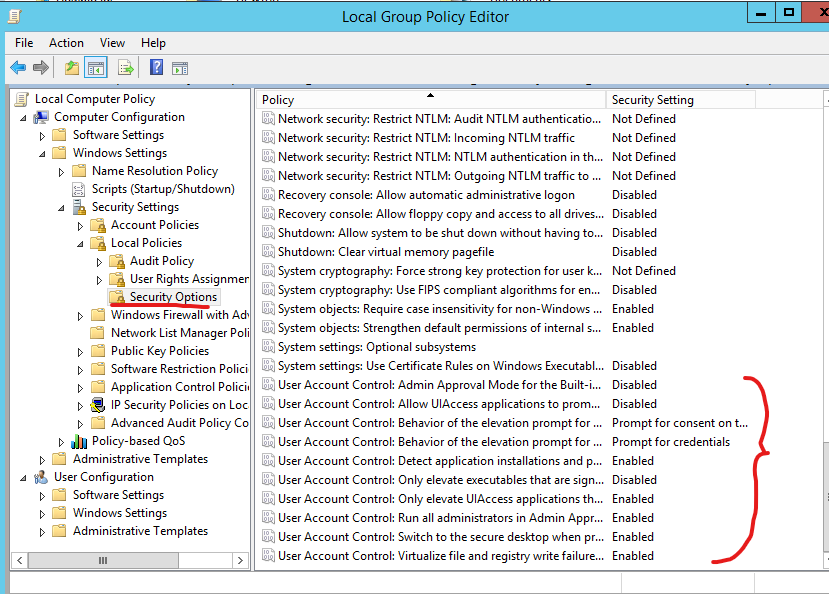
<https://learn.microsoft.com/en-us/windows/security/application-security/application-control/user-account-control/>

User Account Control (UAC) is a Windows security feature designed to protect the operating system from unauthorized changes. When changes to the system require administrator-level permission, UAC notifies the user, giving the opportunity to approve or deny the change. UAC improves the security of Windows devices by limiting the access that malicious code has to execute with administrator privileges.

UAC is enabled by default, and you can configure it if you have administrative privileges. Going to ‚Control Panel’ -> ‚User Accounts’ -> ‚Change User Account Control setting’ we can check and modify the level of alerts. In a professional environment, it’s best to keep it at the highest level.

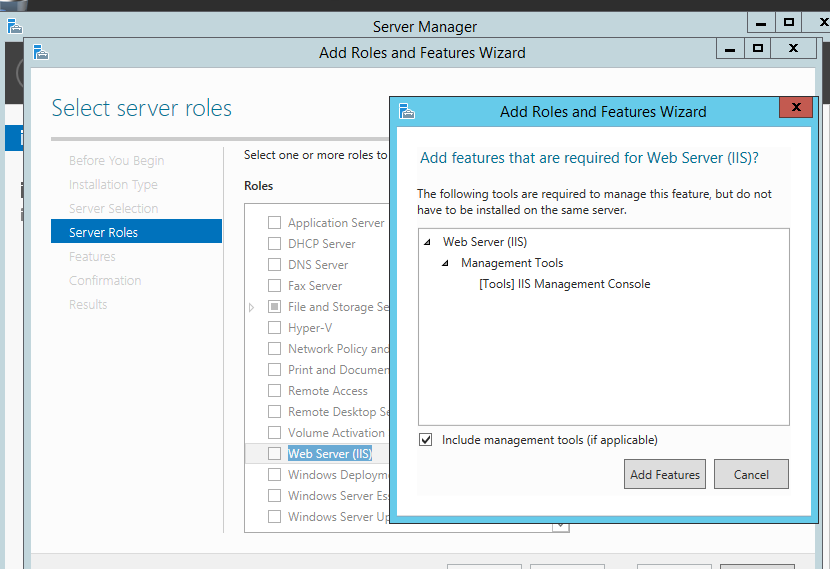


For more security options related to UAC, we have to go Group Policies Manager, by pressing Win+R and executing „gpedit.msc”. In the following image i have highlighted the section we are interested in. An administrator should check all of these options so he can enable everything deemed necessary to properly secure the system.

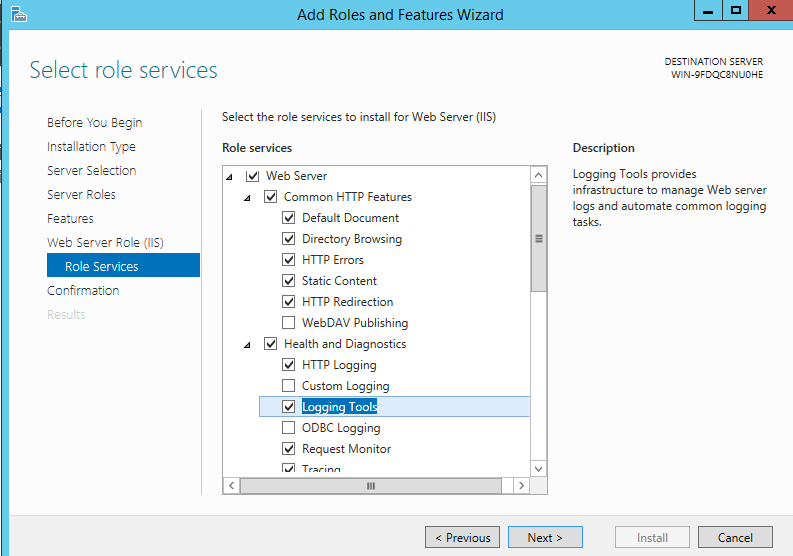


1. Installez un serveur IIS suivant votre serveur Windows. Configurez le en suivant les meilleures pratiques de sécurité. Expliquez votre démarche et votre mise en oeuvre.

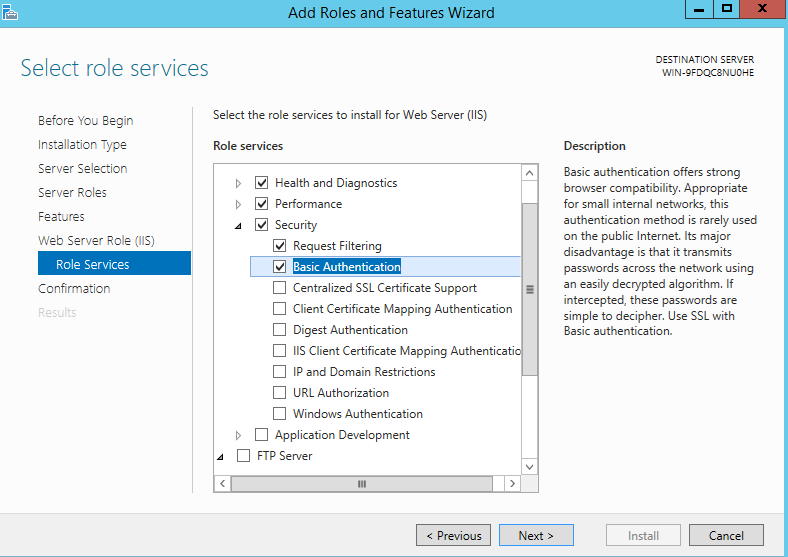
Following the documentation found at [learn.microsoft.com/en-us/iis/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012](https://learn.microsoft.com/en-us/iis/get-started/whats-new-in-iis-8/installing-iis-8-on-windows-server-2012), to install an IIS web server we need to go to Server Manager -> Add roles and features and select the option for it.

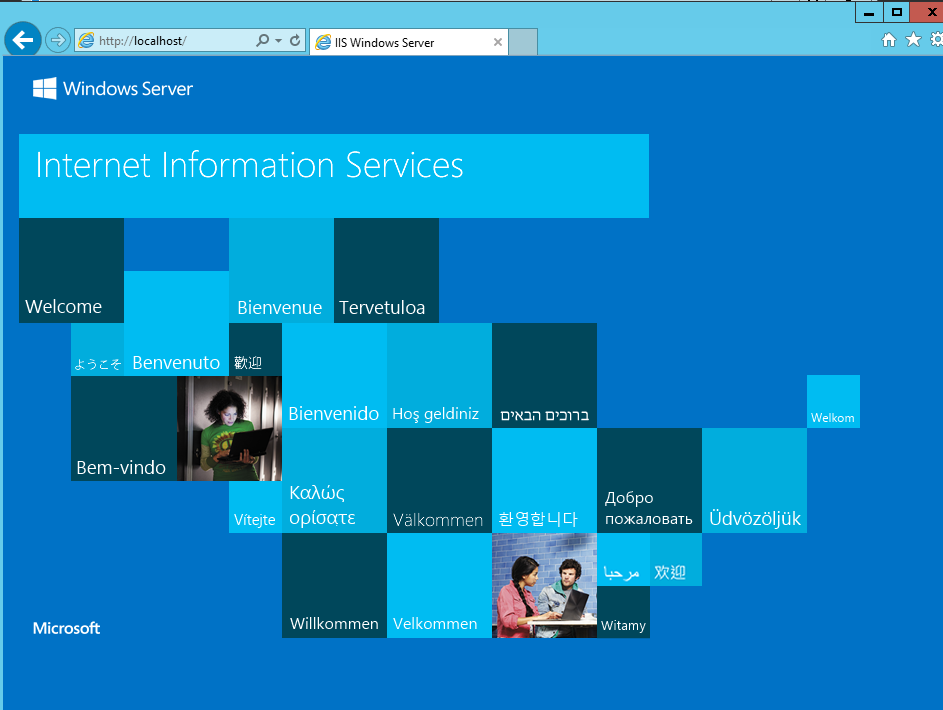


The next step is to check what role services should our web server have. We take into consideration Logging, Requests monitoring, authentification and encryption tools, etc., but limit IIS features to essentials only.



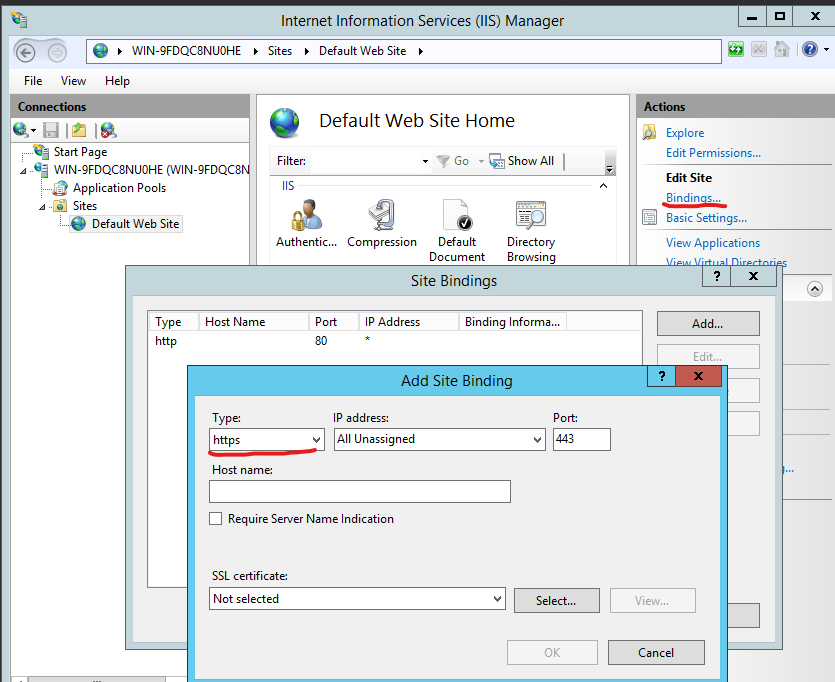
Install Wizard – role services for IIS web server





Our web server is up!

IIS manager is the management application. From here, we can disable any unused modules ( this reduces the attack surface), configure access rights to the web directories as to ensure the least privileges, enable logging and monitoring, or configure HTTPS and disable HTTP, setting proper authentication and authorization and many more. As a bonus step, the web server could be tested for vulnerabilities to identify potential security issues.



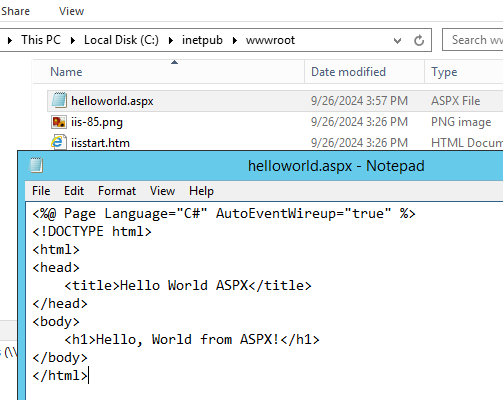
Configuring HTTPS

1. Déposez une page Helloworld ASPX à la racine de votre serveur. Exposez cette ressource sur une interface réseau. Montrez que vous pouvez accéder à la page depuis votre système Hôte.

Sources:  
<https://www.youtube.com/watch?v=VMyMag9_vmc>

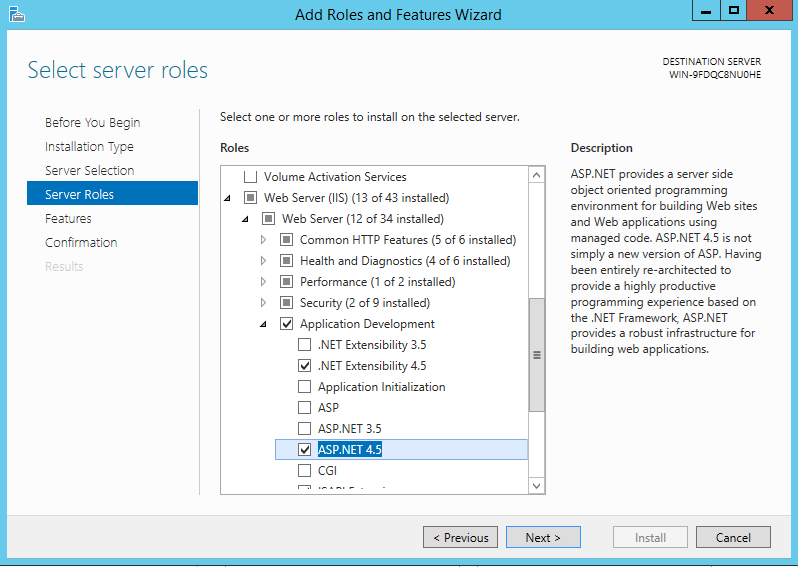
<https://learn.microsoft.com/en-us/aspnet/core/tutorials/publish-to-iis?view=aspnetcore-8.0&tabs=visual-studio>

We will start by creating a file named „helloworld.aspx” inside or web server root ( C:\inetpub\wwwroot).

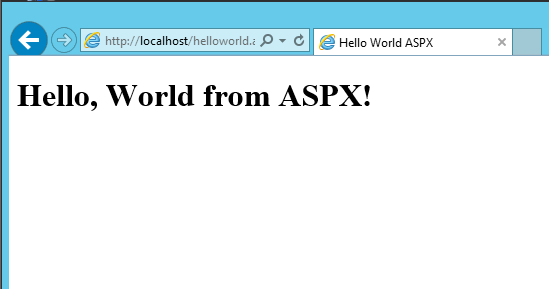


helloworld.aspx

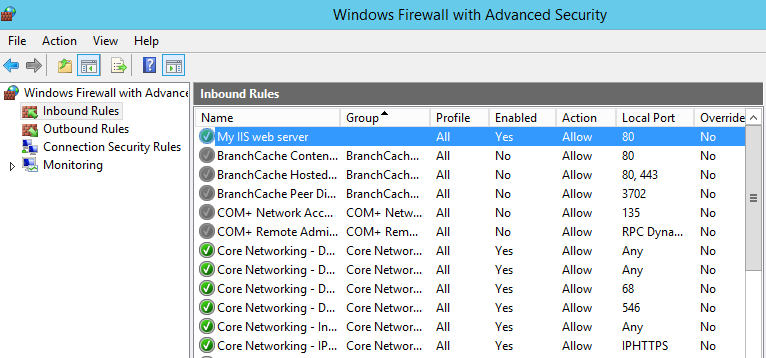
Next step is to enable the ASP.NET feature. Server manager -> Add feature and roles -> Install wizard -> Roles, check under IIS - Application development -> ASP.NET.



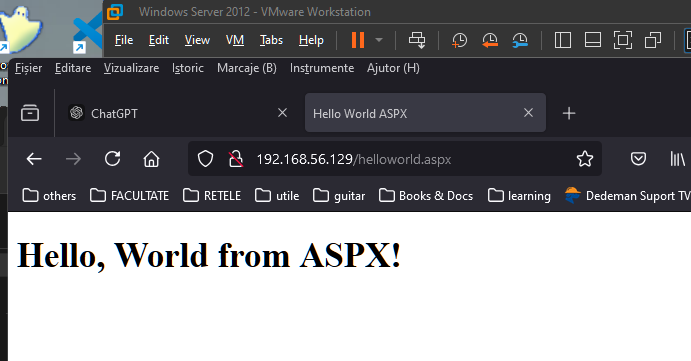
After installation is done, we can check if helloworld.aspx is accesible by navigating to <http://localhost/helloworld.aspx>.



To bind it to the network interface, we check in IIS manager -> Sites -> ‚Default Web Site’ -> bindings. We ensure that there is a HTTP binding with an appropiate IP address ( or unassigned, for our case) and port 80. Then, in Windows Firewall we create a new inbound rule for port 80 to allow traffic.



Finally, we test access from the host machine.



Accesible from the host machine

1. Citez deux méthodes pour se connecter à distance sur votre serveur Windows (méthodes "nativement" supportées). Décrivez rapidement les 2 services / protocoles et les ports par défaut sur lesquels ils sont exposés. Depuis un hôte Linux, connectez-vous à distance à votre serveur en utilisant des clients initiés en ligne de commande.

There are two primary methods natively supported:

* Remote Desktop Protocol (RDP)

Remote Desktop Services (RDS) use RDP (TCP, port 3389) to allow graphical remote access [[https://learn.microsoft.com/en-us/windows-server/remote/remote-desktop-services/ remote-desktop-services-overview](https://learn.microsoft.com/en-us/windows-server/remote/remote-desktop-services/%20remote-desktop-services-overview)].win

* Windows Management Instrumentation (WMI) over Windows Remote Management (WinRM)

Windows Remote Management (WinRM) is the Microsoft implementation of the [WS-Management protocol](https://learn.microsoft.com/en-us/windows/win32/winrm/ws-management-protocol), which is a standard Simple Object Access Protocol (SOAP)-based, firewall-friendly protocol that allows interoperation between hardware and operating systems from different vendors [[https://learn.microsoft.com/en-us/windows/win32/ winrm/portal](https://learn.microsoft.com/en-us/windows/win32/%20winrm/portal)]. It uses ports 5985 for HTTP (unsecured, rare) and 5986 for HTTPS (secured with SSL/TLS).

For remote access to our Windows Server we need to check:

* RDP is enabled ( System -> Remote -> Allow remote connections
* Firewall inbound rules permit traffic
* Firewall profile is set to Private / Domain. In my case, this was blocking

In Powershell:

>Get-NetConnectionProfile # check profile and get exact network name

>Set-NetConnectionProfile -Name " Network-Name" -NetworkCategory Private

To connect from a Linux host, we can use ‚rdesktop’ [<https://blog.helpwire.app/rdesktop-linux/>].

>rdesktop $ipaddress

