
ALMA MATER STUDIORUM UNIVERSITÀ DI BOLOGNA

Corso di Laurea in Tecnologie dei Sistemi Informatici - A.A. 2024/2025

Laboratorio di Sicurezza dei Sistemi e Privacy

- *Relazione Progetto Active Directory* -

Laboratorio Virtuale con Vagrant, Ansible e messa in sicurezza con GPO e ACL +
configurazione server RADIUS

Cacchi Alessandro

Indice

Introduzione.....	3
1. Configurazione dell'ambiente Active Directory	3
Preparazione dell'ambiente host	3
Struttura dell'ambiente	4
Provisioning con Vagrant	4
Risorse Hardware Assegnate	5
Configurazioni di rete	5
Configurazione delle interfacce di rete: public e private	5
Verifica manuale da VirtualBox	6
Configurazione automatizzata con Ansible	7
Avvio dell'ambiente e Risultato	8
2. Messa in sicurezza dell'ambiente Active Directory	9
Firewall – Domain Controller	9
Firewall – Client Windows 10	10
Creazione della struttura logica	11
Creazione dei gruppi AD	11
Creazione delle Organizational Unit (OU).....	12
Creazione di 25 utenti e assegnazione ai gruppi	12
Attivazione Utenti.....	13
3. GPO.....	14
Tutorial creazione GPO.....	14
GPO - Accesso remoto autorizzato (RDP)	14
GPO – Logging and Tracing GPO	15
GPO - Disable CMD	15
GPO - Block Software Download	16
GPO - Block Pannel Control	16
GPO - Disable Removable Storage	16
GPO - Block for Inactivity	17
GPO – Desktop Wallpaper.....	17
GPO - Block Windows Firewall	17
GPO - Block New Printers	17
Estensione GPO a più gruppi AD	18
Propagazione GPO create	19
4. Configurazione share di rete e ACL.....	19
5. Configurazione Server RADIUS.....	21
Installazione e Registrazione del Server	21

Configurazione di un RADIUS Client.....	22
Configurazione Network Policies	23
NP - Accesso Generale alla rete	23
NP - Accesso alla rete amministrativa (solo per il gruppo di sicurezza DL_IT).....	23
NP – Blocco accesso alla rete fuori orario	24
NP – Blocco dell’accesso alla rete guest per gruppi HR e Finance	25

Introduzione

Il presente progetto ha come obiettivo la creazione di un'infrastruttura Active Directory in ambiente simulato, con particolare attenzione alla configurazione iniziale e alla successiva messa in sicurezza del dominio. Per realizzare l'ambiente di test sono state utilizzate tecnologie di virtualizzazione (Vagrant) e automazione (Ansible), al fine di garantire ripetibilità e coerenza nella configurazione delle macchine.

Il laboratorio si compone principalmente di due macchine virtuali:

- un **Domain Controller** basato su Windows Server, responsabile della gestione centralizzata di utenti, gruppi, policy e autenticazioni;
- un **client Windows 10**, unito al dominio per testare le funzionalità e le policy applicate.

Nella fase iniziale è stata effettuata l'installazione e la configurazione del ruolo Active Directory Domain Services (AD DS), comprensiva della promozione a Domain Controller e della configurazione DNS. Successivamente, l'ambiente è stato popolato con utenti, gruppi e risorse tramite script automatici per simulare un'infrastruttura aziendale complessa.

Particolare attenzione è stata dedicata alla **messa in sicurezza** del dominio, attraverso l'uso di **Group Policy Objects (GPO)**, configurazioni di **Access Control List (ACL)**, controllo degli accessi RDP, e definizione di **regole firewall** mirate. Inoltre, è stata prevista la possibilità di estendere il progetto con la configurazione di un **server RADIUS**, con lo scopo di centralizzare la gestione dell'accesso alla rete.

Successivamente verranno illustrate passo dopo passo le attività svolte, le scelte progettuali adottate e le soluzioni implementate per garantire un ambiente Active Directory il più possibile coerente a uno scenario reale, con focus su sicurezza, organizzazione e automatizzazione.

1. Configurazione dell'ambiente Active Directory

Per la realizzazione dell'ambiente Active Directory è stato predisposto un laboratorio virtuale costituito da due macchine virtuali: un **Domain Controller** basato su Windows Server 2019 e un **client** con Windows 10. L'intero ambiente è stato automatizzato tramite **Vagrant** per la gestione delle VM e **Ansible** per la configurazione post-deployment.

Preparazione dell'ambiente host

Per automatizzare, facilitare e velocizzare il processo di preparazione dell'ambiente host e installare tutto quello che serve per lanciare e configurare l'ambiente virtuale, ho creato uno *script.ps1* che mi permettesse di:

- Installare Vagrant (se già non fosse presente)
- Installare WSL (se già non fosse presente)
- Installare Ansible (se già non fosse presente)

Script del file setup_host.ps1

```
# Verifica se Vagrant è installato
$vagrantInstalled = Get-Command vagrant -ErrorAction SilentlyContinue

if (-not $vagrantInstalled) {
    Write-Output "Vagrant non trovato. Scaricando e installando..."
    Start-Process "https://releases.hashicorp.com/vagrant/2.4.1/vagrant_2.4.1_x86_64.msi"
    exit
} else {
    Write-Output "[OK] Vagrant già installato"
}

# Verifica se WSL è installato
$wslInstalled = wsl --status 2>$null

if ($LASTEXITCODE -ne 0) {
    Write-Output "WSL non trovato. Installazione in corso..."
    wsl --install
    Write-Output "Riavvia il PC, poi rilancia lo script"
    exit
} else {
    Write-Output "[OK] WSL già installato"
}

# Lancia uno script in WSL per installare Ansible
wsl bash -c '/mnt/c/st-alessandro-cacchi/setup-scripts/setup_ansible.sh'
```

In alternativa, si può procedere manualmente alla preparazione dell'ambiente host. Per prima cosa, installare Vagrant → https://releases.hashicorp.com/vagrant/2.4.1/vagrant_2.4.1_x86_64.msi

Il resto del procedimento sarà spiegato [nelle sezioni successive](#).

Struttura dell'ambiente

L'ambiente è stato progettato con la seguente topologia:

Ruolo	Sistema Operativo	IP Privato	Hostname	RAM/CPU
Domain Controller	Windows Server 2019	10.10.10.100	winserver	6144MB – 2 vCPU
Client	Windows 10	10.10.10.101	win10	4096MB – 2 vCPU

Provisioning con Vagrant

L'ambiente virtuale è stato definito tramite il *Vagrantfile*, il quale descrive la configurazione delle macchine virtuali necessarie (Domain Controller e client Windows), includendo:

- le risorse assegnate (RAM, CPU)
- il networking (private_network e public_network)
- il port forwarding per WinRM, RDP e SSH
- la configurazione IP statica via script PowerShell integrato.

Grazie a Vagrant, è possibile ricreare l'intero ambiente con il comando - `vagrant up` - (se si vogliono avviare/creare entrambe le VM in contemporanea) oppure - `vagrant up nome della VM` - (se si vuole avviare/creare una sola VM specifica)

```
config.vm.define "winserver" do |server|
  server.vm.box = "StefanScherer/windows_2019"
  server.vm.box_version = "2021.05.15"

  server.vm.network "public_network", type: "dhcp"
  server.vm.network "private_network", ip: "10.10.10.100"

  server.vm.network "forwarded_port", guest: 3389, host: 3390, id: "rdp"
  server.vm.network "forwarded_port", guest: 22, host: 2222, id: "ssh"
  server.vm.network "forwarded_port", guest: 5985, host: 55985, id: "winrm"

  server.vm.provider "virtualbox" do |vb|
    vb.memory = "6144"
    vb.cpus = 2
  end
  server.vm.provision "shell", privileged: true, inline: <<-SHELL
    netsh interface ip set address "Ethernet 3" static 10.10.10.100 255.255.255.0 10.10.10.1
  SHELL
end
```

Risorse Hardware Assegnate

Ho deciso di assegnare risorse hardware differenti alle due macchine virtuali. In particolare, poiché più pesante, alla VM **Windows Server (Domain Controller)** ho assegnato **6 GB di RAM (6144 MB)** e **2 vCPU**, mentre alla VM **Windows 10 (Client)** ho destinato **4 GB di RAM (4096 MB)** e **2 vCPU**.

Configurazioni di rete

Per garantire la comunicazione tra le macchine virtuali, è stata curata in dettaglio la **configurazione di rete** dell'ambiente, sia in fase di test che nella definizione finale tramite Vagrant.

Configurazione delle interfacce di rete: public e private

Ogni macchina virtuale è stata dotata di **due interfacce di rete** distinte, configurate in modo complementare per gestire scenari sia di simulazione LAN interna che di accesso esterno.

1. Public Network (DHCP)

È stata configurata una **scheda di rete in modalità "public_network"**, che sfrutta il DHCP del sistema host:

```
win.vm.network "public_network", type: "dhcp"
```

2. Private Network (host-only)

La comunicazione tra il Domain Controller e il client Windows 10 avviene attraverso una **rete privata (host-only)**, creata tramite uno script python), con indirizzi IP statici assegnati manualmente:

```
server.vm.network "private_network", ip: "10.10.10.100"
```

```
win.vm.network "private_network", ip: "10.10.10.101"
```

Questa rete è essenziale per:

- La **comunicazione** tra il sistema **Host** e le **VM** senza richiedere accesso a una rete esterna
- La **risoluzione DNS** tra le macchine
- Il **join a dominio Active Directory**

Estratto dello script python per la creazione della rete host-only:

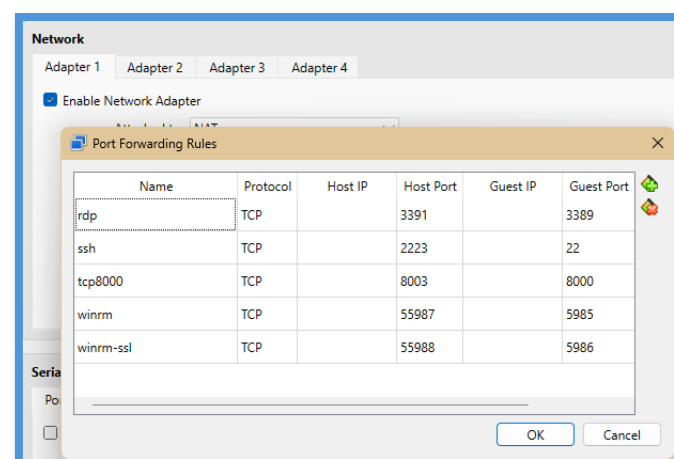
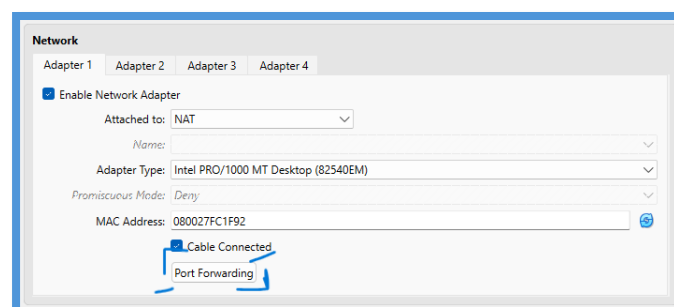
```
def get_existing_hostonly():  
    """Verifica se c'è una rete host-only con IP 10.10.10.1"""  
    output = run_command(f"{vboxmanage_cmd} list hostonlyifs")  
    blocks = output.split("\n\n")  
  
    for block in blocks:  
        name_match = re.search("Name:\s+(.*)", block)  
        ip_match = re.search("IPAddress:\s+([\d\.]+)", block)  
        if name_match and ip_match:  
            if ip_match.group(1) == "10.10.10.1":  
                return name_match.group(1)  
  
    return None
```

Verifica manuale da VirtualBox

In una prima fase, la configurazione delle schede di rete e il forwarding delle porte sono stati testati manualmente direttamente da **VirtualBox**, verificando:

- La raggiungibilità delle VM tramite **RDP (porta 3389)**
- Il funzionamento della connessione **WinRM (porta 5985)** per Ansible
- L'eventuale accesso tramite **SSH (porta 22)**

I test sono stati svolti sia per il **Domain Controller** che per il **client Windows 10**, e hanno permesso di individuare e stabilire le configurazioni di rete più adatte.



Automazione con Vagrant

Dopo aver validato la configurazione manuale, le stesse impostazioni sono state **codificate nel Vagrantfile**, per renderle **ripetibili e automatiche**:

Client Windows 10 (win10)

```
win.vm.network "forwarded_port", guest: 3389, host: 3391, id: "rdp"
win.vm.network "forwarded_port", guest: 22, host: 2223, id: "ssh"
win.vm.network "forwarded_port", guest: 5985, host: 55987, id: "winrm"
```

Domain Controller (winserver)

```
server.vm.network "forwarded_port", guest: 3389, host: 3390, id: "rdp"
server.vm.network "forwarded_port", guest: 22, host: 2222, id: "ssh"
server.vm.network "forwarded_port", guest: 5985, host: 55985, id: "winrm"
```

Configurazione automatizzata con Ansible

Dopo il provisioning, le VM vengono configurate tramite Ansible utilizzando la connessione **WinRM**. Il file `inventory.ini` definisce i due host con i parametri necessari alla comunicazione remota.

1. *configure_server.yml* – Configurazione del Domain Controller

- Sincronizzazione dell'orario (NTP) e fuso orario
- Installazione del ruolo **Active Directory Domain Services (AD DS)**
- Promozione a Domain Controller con dominio `server.local`
- Configurazione DNS interna
- Apertura delle porte firewall necessarie (WinRM, DNS, RDP, ICMP)
- Riavvio automatico al termine della promozione

2. *configure_win10.yml* – Configurazione del client

- Sincronizzazione NTP e fuso orario
- Abilitazione e configurazione dell'account Administrator
- Apertura delle porte firewall per RDP e WinRM
- Configurazione del DNS per puntare al Domain Controller
- Verifica della risoluzione del dominio `server.local`
- Join automatico al dominio e riavvio finale

Avvio dell'ambiente e Risultato

Ho cercato di rendere **quasi completamente automatizzato** l'avvio dell'ambiente. L'unico intervento manuale necessario è l'esecuzione di alcuni comandi iniziali da parte dell'utente.

Per prima cosa, è sufficiente lanciare il comando: **vagrant up**

Questo avvia entrambe le macchine virtuali (Domain Controller e client Windows 10), secondo quanto definito nel *Vagrantfile*.

Una volta che le VM risultano operative, si procede alla **configurazione automatizzata tramite Ansible**.

Se non si è lanciato il primo script per la preparazione dell'ambiente host ([*setup_host.ps1*](#)), si può procedere con l'installazione di WSL e Ansible in modo manuale

Per farlo, è necessario accedere al sottosistema Linux (WSL) da **PowerShell in modalità amministratore**.

Ho quindi per prima cosa installato WSL tramite riga di comando: **wsl --install** e inserito uno username e password

Entrato in WSL ho anche installato ansible con il comando: **sudo apt install ansible**

Una volta preparato l'ambiente, all'interno di WSL, è possibile lanciare i playbook con il comando: **ansible-playbook -i inventory.ini configure_win10.yml**

Una volta che le VM saranno attive, si potranno configurare tramite i playbook ansible. Per lanciare i playbook sarà necessario entrare in WSL da powershell (come amministratore). Per entrare in WSL basterà lanciare il comando **WSL** (da powershell in modalità amministratore).

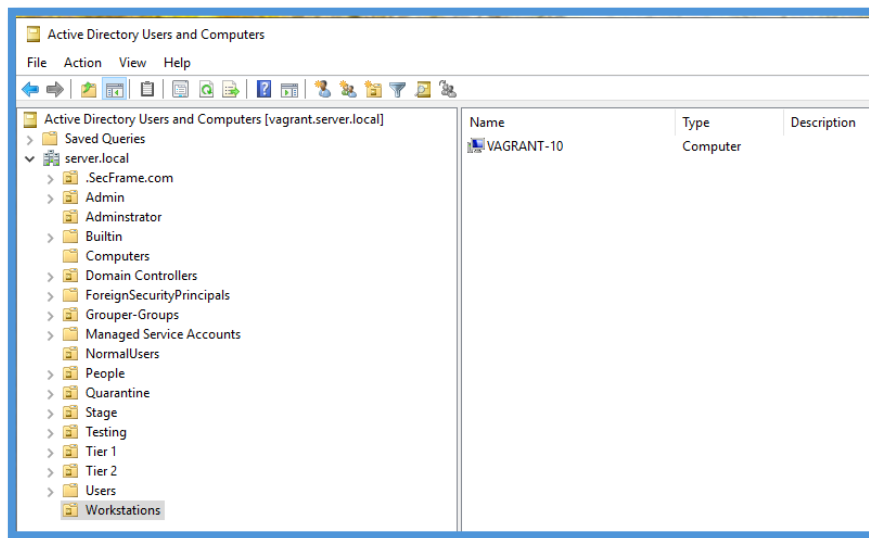
Se non si è ancora configurato un account, verranno richiesti un username e una password (che andranno memorizzate).

Da WSL poi potremmo lanciare il comando **ansible-playbook -i inventory.ini configure_server.yml** e attendere che venga eseguito.

Al termine dell'esecuzione dei playbook, l'ambiente risulterà completamente configurato:

- Il **Domain Controller** sarà operativo e fornirà correttamente i servizi DNS e Active Directory
- Il **client Windows 10** sarà correttamente unito al dominio `server.local`
- Entrambe le VM saranno accessibili tramite **RDP** e gestibili tramite **Ansible (WinRM)**

Per verificare il buon esito della configurazione, ho effettuato un'ulteriore verifica manuale tramite lo strumento **Active Directory Users and Computers** sul Domain Controller.



Dopo aver controllato che le VM fossero configurate in modo corretto (orario, promozione a DC, unione a DC, profili creati), sono passato alla fase successiva.

2. Messa in sicurezza dell'ambiente Active Directory

Per migliorare la sicurezza dell'infrastruttura Active Directory, ho configurato manualmente e via Ansible una serie di **regole del firewall** su entrambe le macchine (Domain Controller e client Windows 10).

Queste regole permettono di **consentire solo i servizi essenziali** e, all'occorrenza, **bloccare tutto il traffico indesiderato**, limitando la superficie di attacco della rete.

Firewall – Domain Controller

Sul Domain Controller (winserver) sono state attivate le seguenti regole:

Nome Regola	Porta	Protocollo	Funzione
Allow ICMPv4-In	N/A	ICMPv4	Permette il ping diagnostico
Allow RDP	3389	TCP	Abilita accesso remoto via RDP
Allow DNS TCP	53	TCP	Permette Richieste DNS TCP
Allow DNS UDP	53	UDP	Permette richieste DNS UDP
Allow WinRM HTTP	5985	TCP	Necessaria per comunicazione Ansible
Allow WinRM HTTPS	5986	TCP	Alternativa sicura a WinRM HTTP

Tutte le regole le ho configurate per i **tre profili di rete**: Domain, Private, Public, per garantire la raggiungibilità anche in caso di variazione del contesto di rete.

La configurazione è stata applicata automaticamente durante la prima esecuzione dello script Ansible:

Estratto di playbook relativo alla configurazione delle regole del Firewall

```
#----- Configura regole del Firewall -----
- name: Abilita ICMPv4 Echo Request (ping)
  community.windows.win_firewall_rule:
    name: "Allow ICMPv4-In"
    protocol: ICMPv4
    direction: in
    action: allow
    state: present
    enabled: yes
    profile:
      - Domain
      - Private
      - Public

- name: Abilita RDP (porta 3389)
  community.windows.win_firewall_rule:
    name: "Allow RDP"
    localport: 3389
    protocol: TCP
    direction: in
    action: allow
    state: present
    enabled: yes
    profile:
      - Domain
      - Private
      - Public

- name: Abilita DNS (porta 53 TCP)
  community.windows.win_firewall_rule:
    name: "Allow DNS TCP"
    localport: 53
    protocol: TCP
    direction: in
    action: allow
    state: present
    enabled: yes
    profile:
      - Domain
      - Private
      - Public
```

Firewall – Client Windows 10

Anche sul client (win10) ho configurate regole specifiche per garantire la compatibilità con Ansible e l'accesso remoto amministrativo, ma allo stesso tempo mantenere la superficie esposta ridotta.

Nome Regola	Porta	Protocollo	Funzione
WinRM Ansible HTTP	5985	TCP	Abilita gestione remota con Ansible
RDP Ansible TCP	3389	TCP	Abilita accesso remoto (RDP)

Queste regole consentono esclusivamente le porte necessarie per la **gestione remota e amministrativa**, lasciando tutte le altre porte bloccate di default, come da configurazione di Windows Defender Firewall.

Estratto di playbook relativo alla configurazione delle regole del Firewall

```
#----- Configurazione del Firewall -----  
- name: Abilita regola firewall per WinRM (porta 5985)  
  win_firewall_rule:  
    name: "WinRM Ansible HTTP"  
    localport: 5985  
    action: allow  
    direction: in  
    protocol: tcp  
    state: present  
    enable: yes  
  
- name: Abilita regola firewall per RDP (porta 3389)  
  win_firewall_rule:  
    name: "RDP Ansible TCP"  
    localport: 3389  
    action: allow  
    direction: in  
    protocol: tcp  
    state: present  
    enable: yes
```

Creazione della struttura logica

Per rappresentare una tipica struttura organizzativa aziendale, ho una gerarchia logica nel dominio Active Directory che include **gruppi di sicurezza, unità organizzative (OU) e utenti associati**.

Creazione dei gruppi AD

Ho creato **5 gruppi AD di tipo "Security" e ambito "Global"**, uno per ciascun dipartimento:

- DL_IT
- DL_Finance
- DL_HR
- DL_Marketing
- DL_Sales

Tutti i gruppi sono stati inseriti nella OU (Organizational Unit) "Gruppi", creata per migliorare l'organizzazione dell'AD.

Creazione OU "Gruppi" + Creazione appositi gruppi

```
PS C:\Users\Administrator> New-ADOrganizationalUnit -Name "Gruppi" -Path "DC=server,DC=local"  
PS C:\Users\Administrator> $dipartimenti = "IT", "Finance", "HR", "Marketing", "Sales"  
PS C:\Users\Administrator> foreach ($d in $dipartimenti) {  
>>   $groupName = "DL_$d"  
>>   New-ADGroup -Name $groupName -GroupScope Global -GroupCategory Security -Path "OU=Gruppi,DC=server,DC=local"  
>> }
```

Creazione delle Organizational Unit (OU)

Per separare logicamente gli utenti in base alla loro appartenenza dipartimentale, sono state create 5 OU:

- Utenti_IT
- Utenti_Finance
- Utenti_HR
- Utenti_Marketing
- Utenti_Sales

Questa divisione consente di applicare **GPO mirate**.

```
PS C:\Users\Administrator> foreach ($d in $dipartimenti) {  
>>     New-ADOrganizationalUnit -Name "Utenti_$d" -Path "DC=server,DC=local"  
>> }
```

Creazione di 25 utenti e assegnazione ai gruppi

Sono stati creati **25 utenti** in totale, 5 per ciascun dipartimento. Ogni utente è stato:

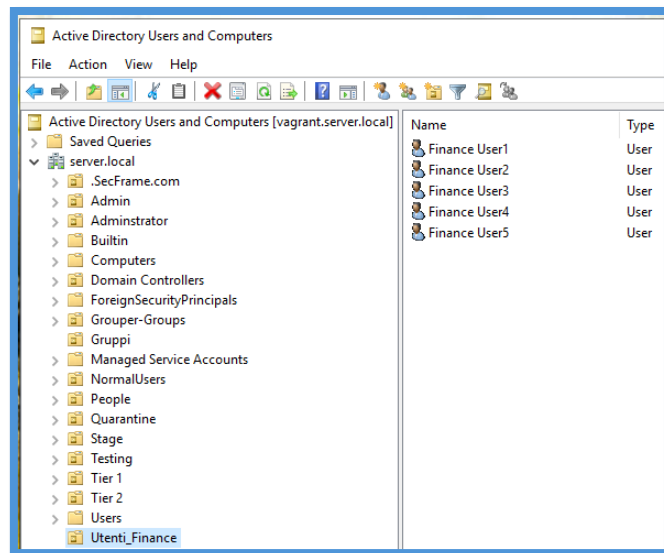
- Inserito nella rispettiva OU (es: OU=Utenti_IT)
- Aggiunto al gruppo di sicurezza corretto (es: DL_IT)
- Creato con una password iniziale sicura: Password123!
- Abilitato al momento della creazione.

Ho generato gli utenti con uno script PowerShell automatizzato, salvato localmente come `create_users.ps1`. Lo script ha iterato sulla lista dei dipartimenti, creando automaticamente utenti con nome coerente

Script create_users.ps1

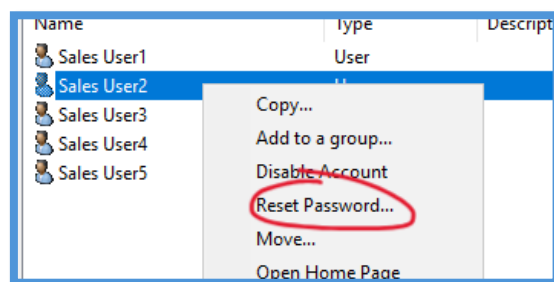
```
$departments = @(  
    @{ Name = "IT"; OU = "Utenti_IT"; Group = "DL_IT" },  
    @{ Name = "Finance"; OU = "Utenti_Finance"; Group = "DL_Finance" },  
    @{ Name = "HR"; OU = "Utenti_HR"; Group = "DL_HR" },  
    @{ Name = "Marketing"; OU = "Utenti_Marketing"; Group = "DL_Marketing" },  
    @{ Name = "Sales"; OU = "Utenti_Sales"; Group = "DL_Sales" }  
)  
  
foreach ($dept in $departments) {  
    for ($i = 1; $i -le 5; $i++) {  
        $username = ($dept.Name.Substring(0,1) + $dept.Name.ToLower() + $i)  
        $fullName = "$($dept.Name) User$i"  
        $password = ConvertTo-SecureString "Password123!" -AsPlainText -Force  
  
        Write-Output "Creazione utente: $username in OU=$($dept.OU) e gruppo=$($dept.Group)"  
  
        New-ADUser `   
            -Name $fullName `   
            -SamAccountName $username `   
            -UserPrincipalName "$username@server.local" `   
            -AccountPassword $password `   
            -Enabled $true `   
            -Path "OU=$($dept.OU),DC=server,DC=local"  
  
        Add-ADGroupMember -Identity $dept.Group -Members $username  
    }  
}
```

Poi dalla console Active Directory Users and Computers sono andato a verificare che fossero stati creati tutti gli utenti e fossero stati inseriti nel gruppo corretto

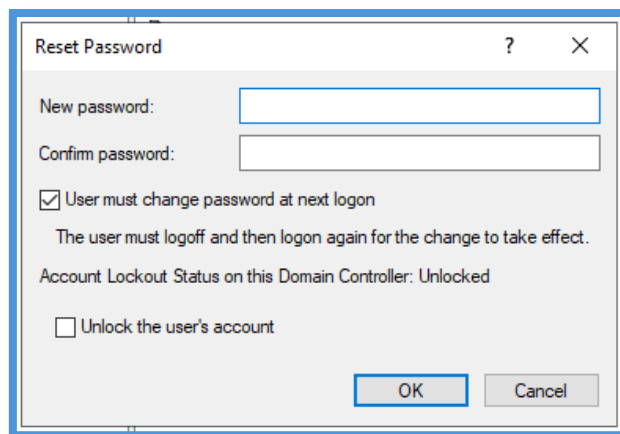


Attivazione Utenti - Manuale

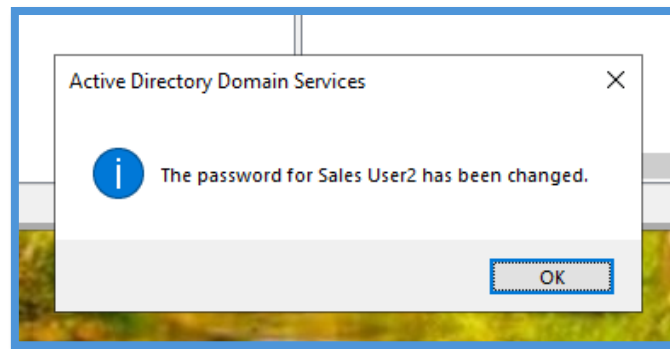
Nel caso in cui volessimo attivare un utente che non è attivato:



1. Come prima cosa a selezionare l'utente che si vuole attivare > tasto destro su di esso > "Reset Password..."



2. Inserito una nuova password → spuntato (se già non lo era) l'opzione "☑ User must change password at next logon" → confermato con "OK"



3. Dopo aver concluso la procedura sul Domain Controller → dirigersi sul Client per il login e cambiato la Password come richiesto.

3. GPO

Successivamente verranno le tecniche per creare svariate GPO, al fine di simulare una buona messa in sicurezza.

Creazione GPO

GPO - Accesso remoto autorizzato (RDP)

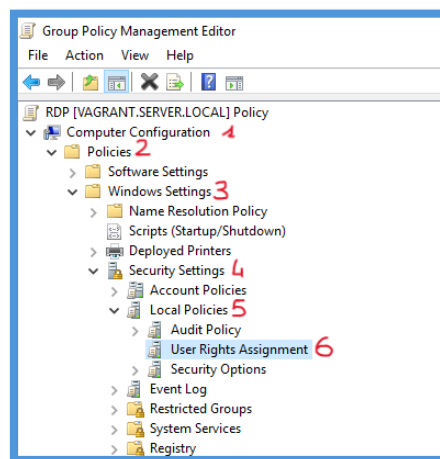
Questa Group Policy Object consente l'**accesso remoto tramite RDP (Remote Desktop Protocol)** esclusivamente agli utenti del **reparto IT**.

Attraverso la configurazione delle autorizzazioni per il servizio RDP la policy garantisce che **solo i membri del gruppo IT (es. DL_IT)** possano effettuare connessioni remote ai sistemi del dominio (es. al Domain Controller o a workstation IT).

La seguente parte iniziale sarà la medesima per tutte le altre GPO

1. Dirigersi su **Group Policy Management** (Start → Administrative Tools)
2. Espandere la struttura → **Forest: server.local → Domains → Server Local**
3. Clic destro sul nome del dominio → "**Create a GPO in this domain, and Link it here...**"
4. Inserire il nome della GPO (in questo caso Enable RDP)
5. Clic destro sulla GPO creata → **Edit**

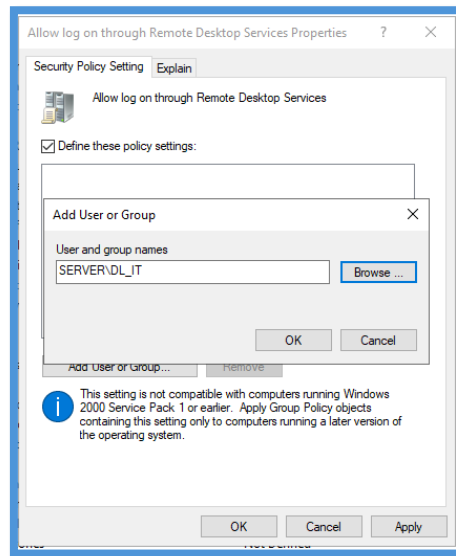
La parte successiva invece sarà simile per tutte le altre GPO:



1. **Percorso:** Computer configuration → Policies → Windows Settings → Security Settings → Local Policies → User Rights Assignment

Add workstations to domain	Not Defined
Adjust memory quotas for a process	Not Defined
Allow log on locally	Not Defined
Allow log on through Remote Desktop Services	Not Defined
Back up files and directories	Not Defined
Bypass traverse checking	Not Defined

2. Doppio clic su “**Allow log on through Remote Desktop Services**”



3. Aggiungere solo il gruppo **DL_IT**, cliccare su Apply e poi su OK

GPO – Logging and Tracing GPO

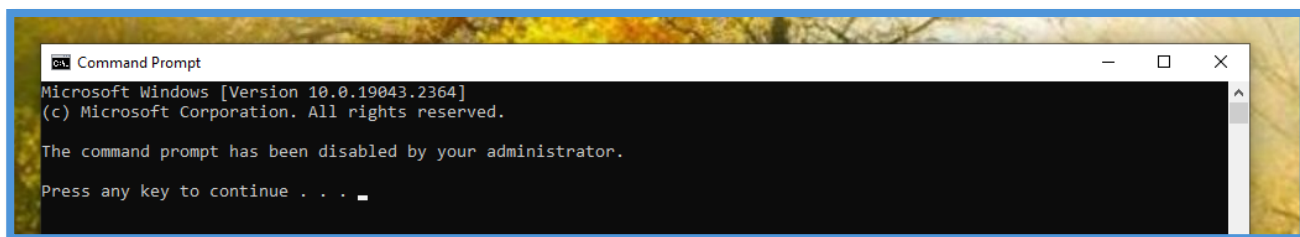
Questa GPO ci permette di tracciare ogni modifica di GPO o configurazione fatta dal gruppo IT per auditing e sicurezza

1. **Percorso:** Computer Configuration → Policies → Windows Settings → Advanced Audit Policy Configuration → Audit Policies → Policy Change
2. Doppio clic su → Policy Change
3. Abilitare le seguenti voci (una per volta)
 - a. Audit Authorization Policy Change → Success, Failure
 - b. Audit Policy Change → Success, Failure
 - c. Audit MPSSVC Rule-Level Policy Change → Success, Failure
4. Salvare e chiudere l'editor

GPO - Disable Command Prompt

Questa GPO ci permette di disabilitare il **Prompt dei Comandi** e l'esecuzione di script .bat e .cmd che potrebbero provocare un attacco

1. **Percorso:** User Configuration → Policies → Administrative Templates → System
2. Nel pannello di destra doppio clic su “Prevent Access to the command prompt”
3. Nella sezione Options, si può decidere se
 - a. Yes: Disabilita anche l'esecuzione di script .bat e .cmd
 - b. No: Permette l'esecuzione di script, ma blocca comunque l'accesso interattivo al CMD
4. Salvare e chiudere l'editor



Una volta propagate la GPO, si può testare il corretto funzionamento di essa provando ad aprire, su un utente a cui è stata propagata la regola, il command prompt. Automaticamente terminerà il processo.

GPO - Block Software Download

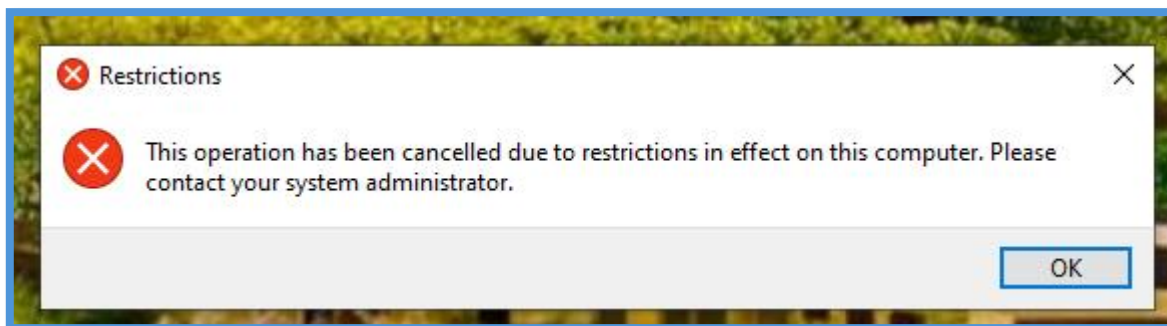
Questa Group Policy Object utilizza le **Software Restriction Policies** per impedire agli utenti di installare software non autorizzato all'interno dell'ambiente aziendale.

1. **Percorso:** Computer configuration → Policies → Windows Settings → Security Settings → Software Restriction Policies
2. Se ancora non esiste fare clic destro su "Software Restriction Policies" → New Software Restriction Policies
3. Selezionare Additional Rules → clic destro su New Path Rule
4. Aggiungere i seguenti path da bloccare:
 - a. C:\Windows\System32\msiexec.exe (Security Level: Disallowed)
 - b. *.exe (Security Level: Disallowed)
 - c. *.msi (Security Level: Disallowed)
5. Salvare e chiudere l'editor

GPO - Block Pannel Control

Questa GPO permette di bloccare il pannello di controllo agli utenti che non hanno l'autorizzazione a farlo

1. **Percorso:** User Configuration → Policies → Administrative Templates → Control Panel
2. Nel pannello di Destra doppio clic su → Prohibit access to Control Panel and PC settings
3. Impostare la Policy su **Enable**
4. Salvare e chiudere l'editor



Una volta propagate la GPO, si può testare il corretto funzionamento di essa provando ad aprire, su un utente a cui è stata propagata la regola, il pannello di controllo. Apparirà a schermo questo errore.

GPO - Disable Removable Storage

Questa GPO permette di disabilitare l'accesso agli Storage Rimovibili (es: Chiavette USB)

1. **Percorso:** Computer Configuration → Policies → Administrative Templates → System → Removable Storage Access

2. Nel pannello di Destra doppio clic su → All Removable Storage classes: Deny all access
3. Imposta la Policy su **Enable**
4. Salvare e chiudere l'editor

GPO - Block for Inactivity

Questa GPO blocca la sessione se l'utente è inattivo, utile per la sicurezza

1. **Percorso:** Computer Configuration → Administrative Tools → Control Panel → Personalization → Screen Saver Timeout
2. Impostare un tempo (es: 300 secondi = 5 minuti)
 - a. Force Specific Screen Saver → Desktop Wallpaper.jpg
 - b. Password protect the screen saver → Enabled
3. Salvare e chiudere l'editor

GPO – Desktop Wallpaper

Questa GPO imposta uno sfondo del desktop uguale per tutti gli utenti del dominio

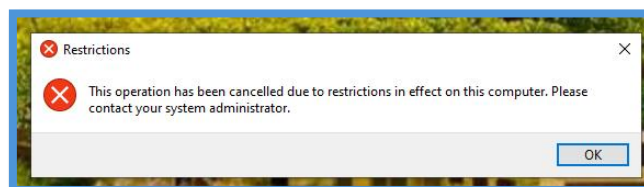
Prima di iniziare bisogna caricare un'immagine in .jpg (non altre estensioni) all'interno di una cartella condivisa.

1. **Percorso:** User Configuration → Administrative Templates → Desktop → Desktop
2. **Configurazione Policy:**
 - a. Imposta su **Enable**
 - b. Nella casella **Wallpaper Name** inserisci il percorso UNC completo dell'immagine
\\10.10.10.100\gpo_share\desktop_wallpaper.jpg
 - c. Inserisci **Fill** o **Stretch** a secondo della risoluzione dei client
3. Salvare e chiudere l'editor

GPO - Block Windows Firewall

Questa GPO blocca l'accesso a Windows Defender Firewall agli utenti che non hanno l'autorizzazione a farlo

1. **Percorso:** Computer Configuration → Administrative Templates → Network → Network Connections → Windows Defender Firewall → Standard Profile → Windows Defender Firewall: Protect all network connection
2. Imposta policy su **Disable**
3. Salvare e chiudere l'editor



Una volta propagate la GPO, si può testare il corretto funzionamento di essa provando ad aprire, su un utente a cui è stata propagata la regola, Windows Defender Firewall. Apparirà a schermo questo errore.

GPO - Block New Printers

Questa GPO Evita che gli utenti installino nuove stampanti senza il consenso dell'amministratore

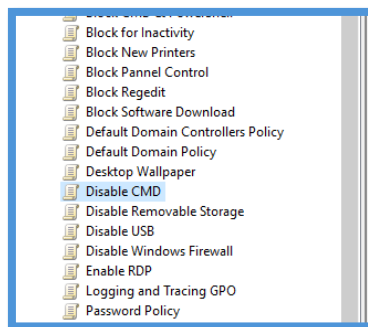
1. **Percorso:** Computer Configuration → Administrative Templates → Network → Network Connections → Windows Defender Firewall → Standard Profile → Windows Defender Firewall: Protect all network connection
2. Nel pannello di Destra doppio clic su → Printers
3. Imposta policy su **Enable**
4. Salvare e chiudere l'editor

Estensione GPO a più gruppi AD

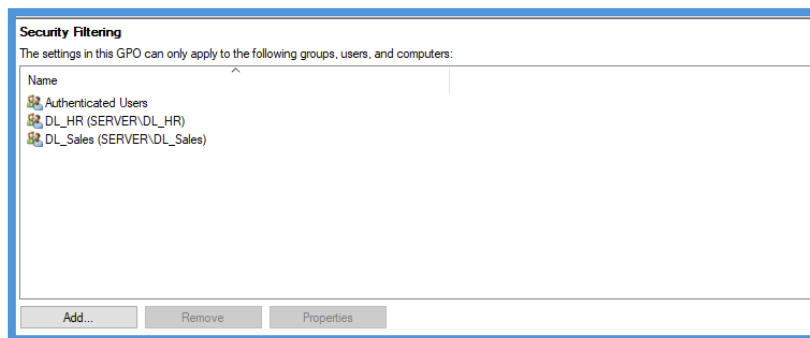
Alcune Group Policy Object sono state progettate per rispondere a esigenze comuni tra più dipartimenti, come ad esempio il blocco del command prompt, del pannello di controllo, del Windows Defender Firewall...

Per evitare duplicazioni e mantenere una gestione centralizzata ed efficiente, ho deciso di **linkare una singola GPO a più gruppi di sicurezza**, in un secondo momento, sfruttando la funzionalità di **Security Filtering**.

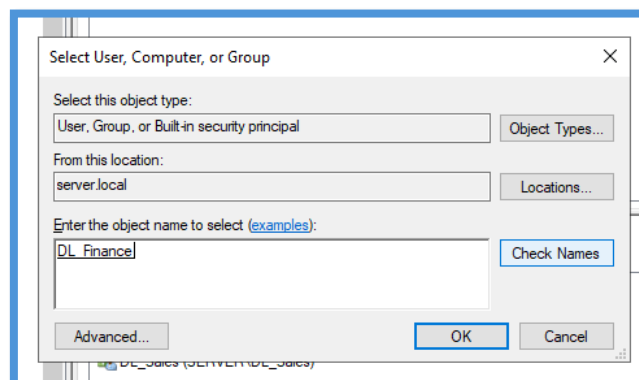
1. Per prima cosa ho espanso il gruppo **Group Policy**



2. Ho selezionato quindi la GPO



3. E ho aggiunto, tramite il tasto "Add...", il gruppo di sicurezza a cui volevo linkare la GPO



4. Fatto il check del nome e cliccato OK

Propagazione GPO create

Una volta create tutte le GPO, mi sono diretto sulla VM client (win10) e ho lanciato il seguente comando da PowerShell (modalità amministratore) → `gpupdate /force`

Ho poi fatto il log out e di nuovo il login all'account per riuscire a propagare correttamente tutte le regole.

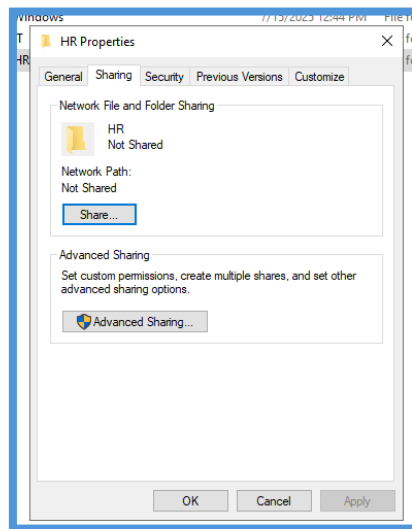
4. Configurazione share di rete e ACL

Sono andato a creare varie cartelle (una per ogni gruppo di sicurezza), manualmente e andando poi a condividerle e dare i permessi necessari:

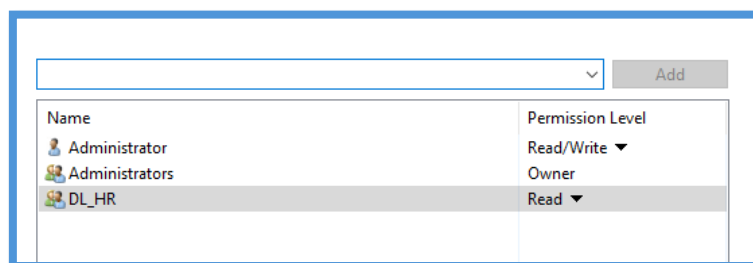
Mi sono diretto nel disco C:/ del domain controller (winserver) e sono andato a creare manualmente le cartelle.

Successivamente sono andato a condividerle e a dare i permessi:

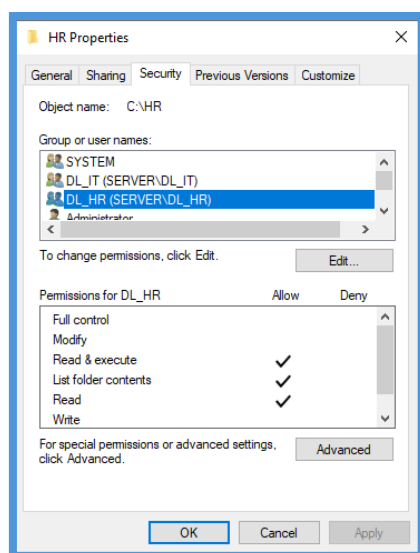
Tasto destro sulla cartella → Properties



Ho poi cliccato Shares e aggiunto il gruppo a cui volevo condividere la cartella. In questo caso DL_HR e poi anche DL_IT



Successivamente sono andato in security per dare i permessi necessari a ciascun gruppo:



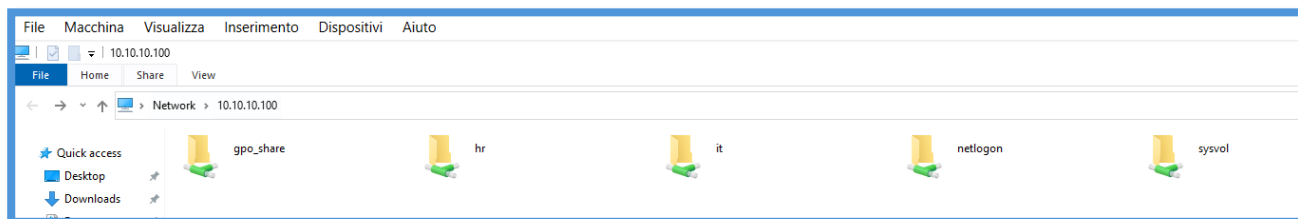
Ho fatto la stessa cosa per tutti quanti i gruppi. Quindi ogni gruppo avrà la cartella condivisa sia con loro stessi che con il gruppo DL_IT (es. tutti gli user di DL_Sales e DL_IT potranno visualizzare la cartella Sales).

Per accedere poi alla cartella da uno user di DL_HR, ad esempio, basterà fare:

1. Win + R
2. Cercare [\\10.10.10.100](http://10.10.10.100)

Si aprirà la schermata Network e vedremo la cartella condivisa

Es. da uno User IT



Quindi in conclusione:

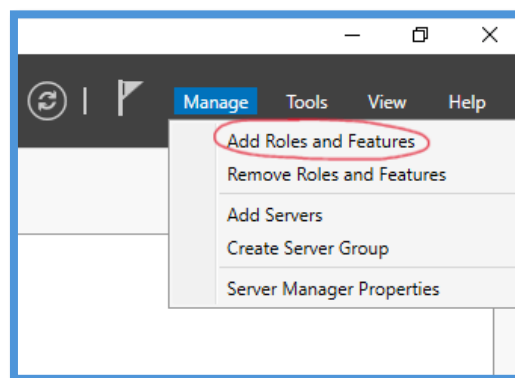
	IT	HR	Finance	Marketing	Sales
DL_IT	✓	✓	✓	✓	✓
DL_HR	⊘	✓	⊘	⊘	⊘
DL_Finance	⊘	⊘	✓	⊘	⊘
DL_Marketing	⊘	⊘	⊘	✓	✗
DL_Shares	⊘	⊘	⊘	⊘	✓

5. Configurazione Server RADIUS

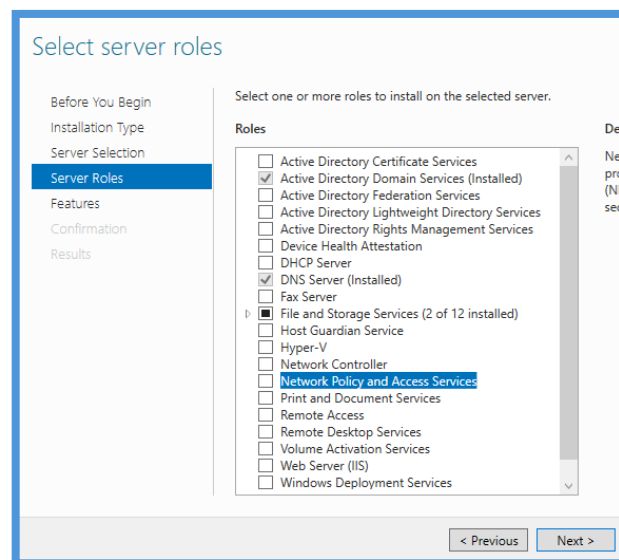
Per completare la simulazione di un'infrastruttura aziendale sicura e realistica, ho configurato un **server RADIUS (Remote Authentication Dial-In User Service)** all'interno del Domain Controller. Il servizio RADIUS è comunemente utilizzato per **centralizzare l'autenticazione degli accessi alla rete**, in particolare per dispositivi come switch, access point, VPN e firewall. Nel nostro ambiente virtuale, anche in assenza di dispositivi reali, è stato possibile simulare la presenza del servizio tramite il ruolo **Network Policy Server (NPS)**, valutato in base alle policy configurate.

Installazione e Registrazione del Server

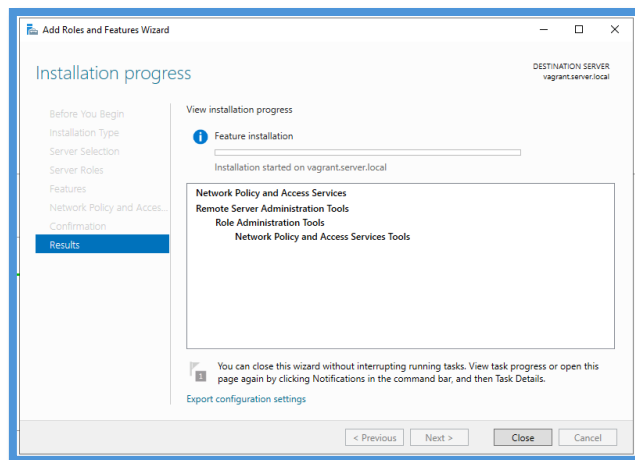
Tramite la **console GUI di Windows Server**, ho installato il ruolo **Network Policy and Access Services**, che include il componente NPS.



Ho premuto Next fino alla sezione Server Roles e ho quindi selezionato **Network Policy and Access Services**



Installato e al termine chiuso



Una volta completata l'installazione, il server è stato **registrato nel dominio Active Directory**, per permettere al servizio RADIUS di accedere alle informazioni sugli utenti e gruppi.

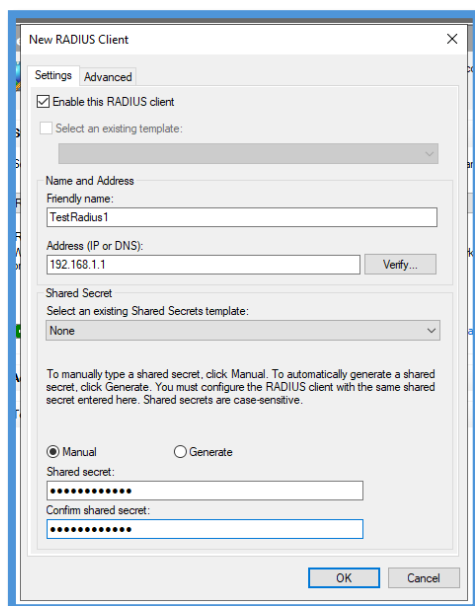
Configurazione di un RADIUS Client

Mi sono diretto nel Network Policy Server, clic destro su NPS (Local) e ho selezionato Register server in Active Directory, confermato 2 volte nelle 2 finestre che sono apparse e così ho permesso al server RADIUS di accedere ai dati utente di AD

Per completare la struttura, è stato aggiunto un **client RADIUS fittizio**, rappresentativo di un dispositivo di rete che invia richieste di autenticazione.

Sempre nel Network Policy Server, ho espanso il nodo: RADIUS Clients and Servers → RADIUS Clients poi tasto destro su esso → New

Ho compilato i campi, inserendo un segreto (*CyberLoop01!* in questo caso), e confermato:



Configurazione Network Policies

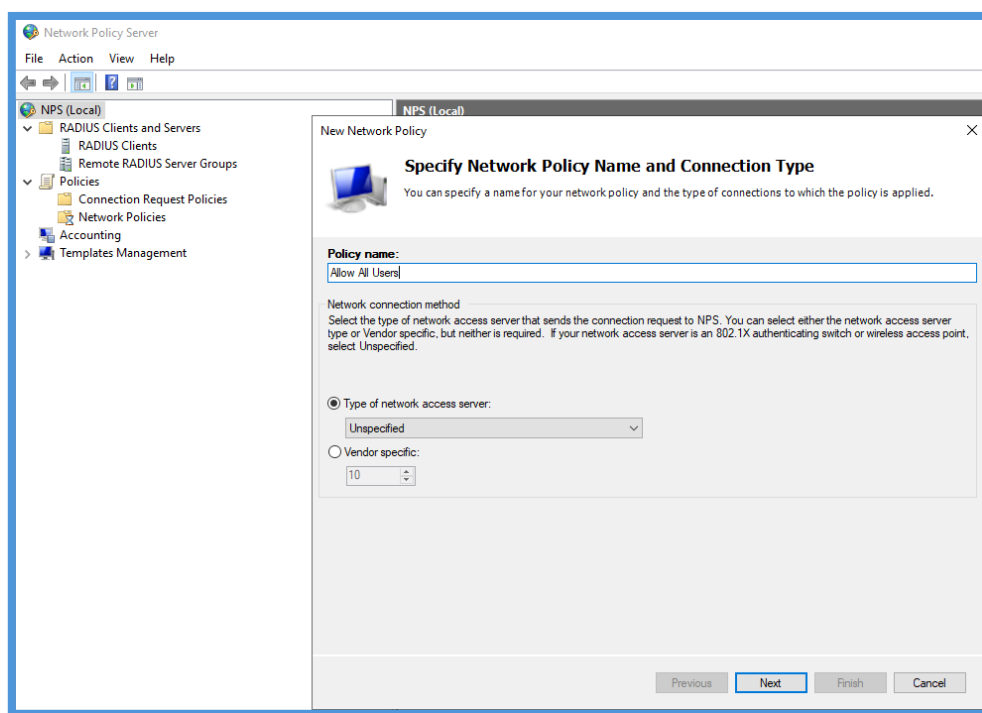
Ho configurato infine alcune Network Policies:

NP - Accesso Generale alla rete

Questa Network Policy chiamata **Allow All Users** che concede l'accesso alla rete a **tutti gli utenti del dominio** (Domain Users), utilizzando il protocollo sicuro **MS-CHAPv2**

Configurazione:

1. **Policy name:** *Allow All Users*
2. **Add → User Groups → Add Groups →** Inserire SERVER\Domain Users
3. **Access Granted**
4. **Cliccare Next fino alla fine e poi Finish**
5. Nella lista delle Network Policy, doppio click sulla policy appena creata
6. Constraints → Authenticated Methods → Add → **Microsoft: Secured password (EAP-MSCHAP v2)**
7. Ok e chiudere l'editor








NP - Accesso alla rete amministrativa (solo per il gruppo di sicurezza DL_IT)

Questa Network Policy chiamata **Allow IT Admin Network**, che consente l'accesso solo agli utenti appartenenti al gruppo **DL_IT**, rappresentanti il dipartimento IT

1. **Policy name:** *Allow IT Admin Network*
2. **Add → User Groups → Add Groups →** Inserire SERVER\DL_IT
3. Aggiungere una condizione su NAS Identifier = admin-network (si simula una rete amministrativa)
4. **Access Granted**

5. **Cliccare Next fino alla fine e poi Finish**
6. Nella lista delle Network Policy, doppio click sulla policy appena creata
7. Constraints → Authenticated Methods → Add → **Microsoft: Secured password (EAP-MSCHAP v2)**
8. Ok e chiudere l'editor

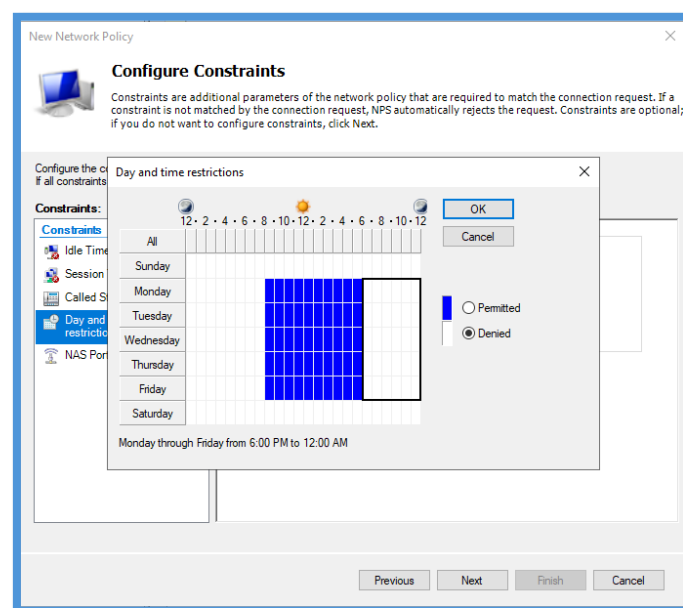
Network Policies				
 Network policies allow you to designate who is authorized to connect to the network and the circumstances under which they can or cannot connect.				
Policy Name	Status	Processing Order	Access Type	Source
 Allow IT Admin Network	Enabled	1	Grant Access	Unspecified
 Allow All Users	Enabled	2	Grant Access	Unspecified
 Connections to Microsoft Routing and Remote Access server	Enabled	999998	Deny Access	Unspecified
 Connections to other access servers	Enabled	999999	Deny Access	Unspecified

NP – Blocco accesso alla rete fuori orario

Questa Network Policy chiamata **Day and Time Restriction**, che impedisce agli utenti di connettersi fuori dall'orario lavorativo (impostato a 8.00 – 18.00)

Configurazione:

9. **Policy name: Day and Time Restriction → Next**
10. **Add → User Groups → Add Groups → Next → Next → Next**
11. **Day and time restrictions → Allow access only on these days and at these times**
12. **Edit:**



Ho poi creato una policy simile, ma togliendo le restrizioni, per il gruppo DL_IT. Così ho permesso l'accesso a qualunque ora e in qualunque giorno agli users del gruppo IT. La configurazione è esattamente la stessa, tranne per le restrizioni in cui sarà permesso l'accesso per tutta la settimana a tutte le ore

NP – Blocco dell'accesso alla rete guest per gruppi HR e Finance

Questa Network Policy chiamata **Deny HR Finance on Guest**, che impedisce agli utenti HR e Finance di connettersi alla rete guest aziendale. Simuliamo il fatto che questi due gruppi sono dei gruppi sensibili e devono esporsi il meno possibile.

Configurazione:

1. **Policy name:** *Deny HR Finance on Guest* → *Next*
2. **Add → User Groups → Add Groups:** aggiungere i gruppi **DL_HR** e **DL_Finance**
3. **Add... → NAS Identifier** → inserire la rete inventata (es: guest-wifi)
4. Confermare fino alla fine e cliccare **Finish**

Password			
User Name	Password	Nome AD	Password Precedente
Ffinance1	FinanceUser11!	Finance User1	FinanceUser1!
Hhr1	HRUser11!	HR User1	HRUser1!
lit1	ITUser11!	IT User1	ITUser1!
Mmarketing1	MarketingUser11!	Marketing User1	MarketingUser1!
Ssales1	SalesUser11!	Sales User1	SalesUser1!
Ssales2	SalesUser22!	Sales User2	SalesUser2!
TestRadius1	CyberLoop01!		