Adresse ip:

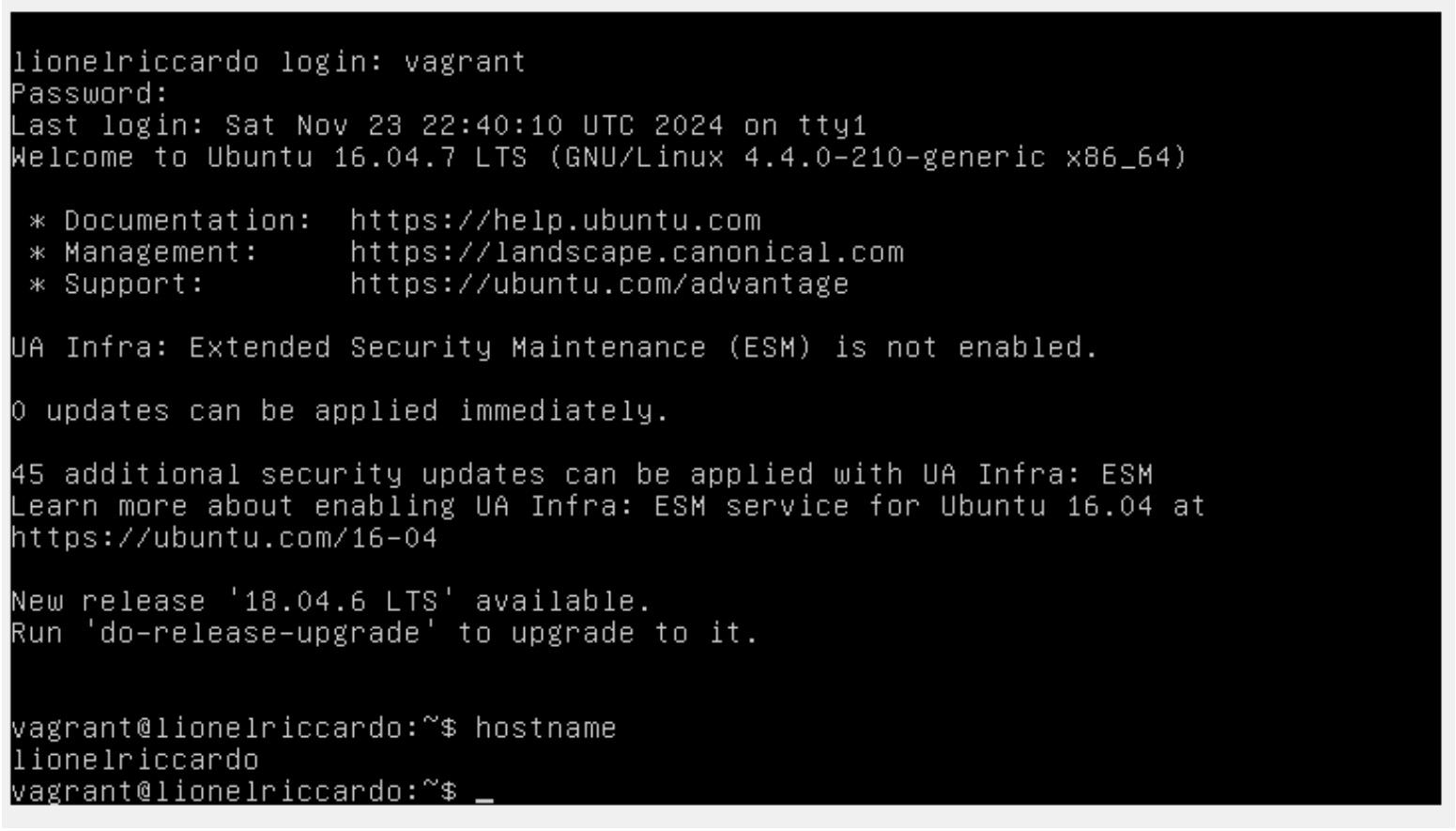
Premièrement, nous nous sommes assurés que l'adresse IP correspondait bien à celle demandée dans le projet et qu'elle était bien assignée à l'interface eth1. Lors de l'ouverture de la machine, nous avons utilisé la commande ip a pour vérifier la configuration actuelle de l'adresse IP. Nous avons constaté que l'adresse IP ne correspondait pas à celle demandée et qu'elle était assignée à l'interface enp0s8 au lieu de eth1.

Pour corriger cela, nous avons modifié le fichier de configuration réseau avec la commande sudo nano /etc/netplan/01-netcfg.yaml, où nous avons ajusté les paramètres pour qu'ils correspondent à ceux demandés dans le projet. Après avoir sauvegardé les modifications, nous avons appliqué les nouveaux paramètres avec la commande sudo netplan apply.

```
vagrant@ubuntu:~$ sudo ip addr add 192.168.56.10/24 dev eth1
sudo: unable to resolve host ubuntu: Connection refused
vagrant@ubuntu:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: rename2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default q
len 1000
    link/ether 02:be:82:6b:cc:1d brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP gr
oup default qlen 1000
    link/ether 08:00:27:71:b6:ed brd ff:ff:ff:ff:ff:ff
    inet 192.168.56.10/24 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe71:b6ed/64 scope link
       valid_lft forever preferred_lft forever
vagrant@ubuntu:~$
```

Changement de nom VM:

Nous avons ensuite changé le nom de la VM, comme demandé, en modifiant le fichier /etc/hostname.



SECTION 1:

Pour la première question, nous devions vérifier que SELinux était en mode enforcing. Nous avons vérifié cela avec la commande sestatus. Lorsque nous avons exécuté cette commande, nous nous sommes aperçus que SELinux n'était pas installé. Nous l'avons donc installé avec la commande sudo apt install selinux selinux-utils selinux-basics -y.

Après l'installation, nous l'avons activé avec sudo selinux-activate.

Ensuite, nous avons modifié le fichier de configuration avec sudo nano /etc/selinux/config, où nous avons défini le mode enforcing.

Enfin, nous avons redémarré la machine pour appliquer les modifications.

Dans la deuxième capture d'écran, on voit des messages générés par SELinux indiquant que certaines actions ont été bloquées. Ces messages commencent par avc: denied et montrent, par exemple, que la commande dpkg a essayé d'accéder à des fichiers temporaires mais a été bloquée. Chaque message donne des détails sur ce qui a été bloqué et pourquoi, grâce aux contextes de sécurité utilisés par SELinux (scontext et tcontext). Cela prouve que SELinux est bien activé en mode enforcing et qu'il applique les règles de sécurité en bloquant les actions non autorisées. Ces messages montrent donc que le système est sécurisé et que les politiques fonctionnent correctement.

```
Press Enter for maintenance
(or press Control-D to continue):
root@lionelriccardo:~# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      30
root@lionelriccardo:~# sudo cat /var/log/audit/audit.log | grep AVC
cat: /var/log/audit/audit.log: No such file or directory
root@lionelriccardo:~# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:             default
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Max kernel policy version:      30
```

```
script_t:s0 tclass=process permissive=0
[  166.790314] audit: type=1400 audit(1732483066.680:7): avc:  denied  { transit
ion } for  pid=1329 comm="dpkg" path="/var/lib/dpkg/tmp.ci/preinst" dev="sda1" i
no=258474 scontext=system_u:system_r:kernel_t:s0 tcontext=system_u:system_r:dpkg
_script_t:s0 tclass=process permissive=0
[  166.796590] audit: type=1400 audit(1732483066.688:8): avc:  denied  { transit
ion } for  pid=1330 comm="dpkg" path="/var/lib/dpkg/tmp.ci/postrm" dev="sda1" in
o=258479 scontext=system_u:system_r:kernel_t:s0 tcontext=system_u:system_r:dpkg_
script_t:s0 tclass=process permissive=0
[  180.236020] audit: type=1400 audit(1732483080.124:9): avc:  denied  { transit
ion } for  pid=1408 comm="dpkg" path="/var/lib/dpkg/tmp.ci/preinst" dev="sda1" i
no=258474 scontext=system_u:system_r:kernel_t:s0 tcontext=system_u:system_r:dpkg
_script_t:s0 tclass=process permissive=0
[  180.238520] audit: type=1400 audit(1732483080.124:10): avc:  denied  { transi
tion } for  pid=1409 comm="dpkg" path="/var/lib/dpkg/tmp.ci/postrm" dev="sda1" i
no=258479 scontext=system_u:system_r:kernel_t:s0 tcontext=system_u:system_r:dpkg
_script_t:s0 tclass=process permissive=0
[  210.782512] audit: type=1400 audit(1732483110.672:12): avc:  denied  { transi
tion } for  pid=1489 comm="dpkg" path="/var/lib/dpkg/tmp.ci/preinst" dev="sda1"
ino=258474 scontext=system_u:system_r:kernel_t:s0 tcontext=system_u:system_r:dpk
g_script_t:s0 tclass=process permissive=1
[  211.407072] audit: type=1107 audit(1732483111.292:17): pid=1 uid=0 auid=42949
67295 ses=4294967295 subj=system_u:system_r:kernel_t:s0 msg='avc:  received sete
nforce notice (enforcing=0)
root@lionelriccardo:~# _
```

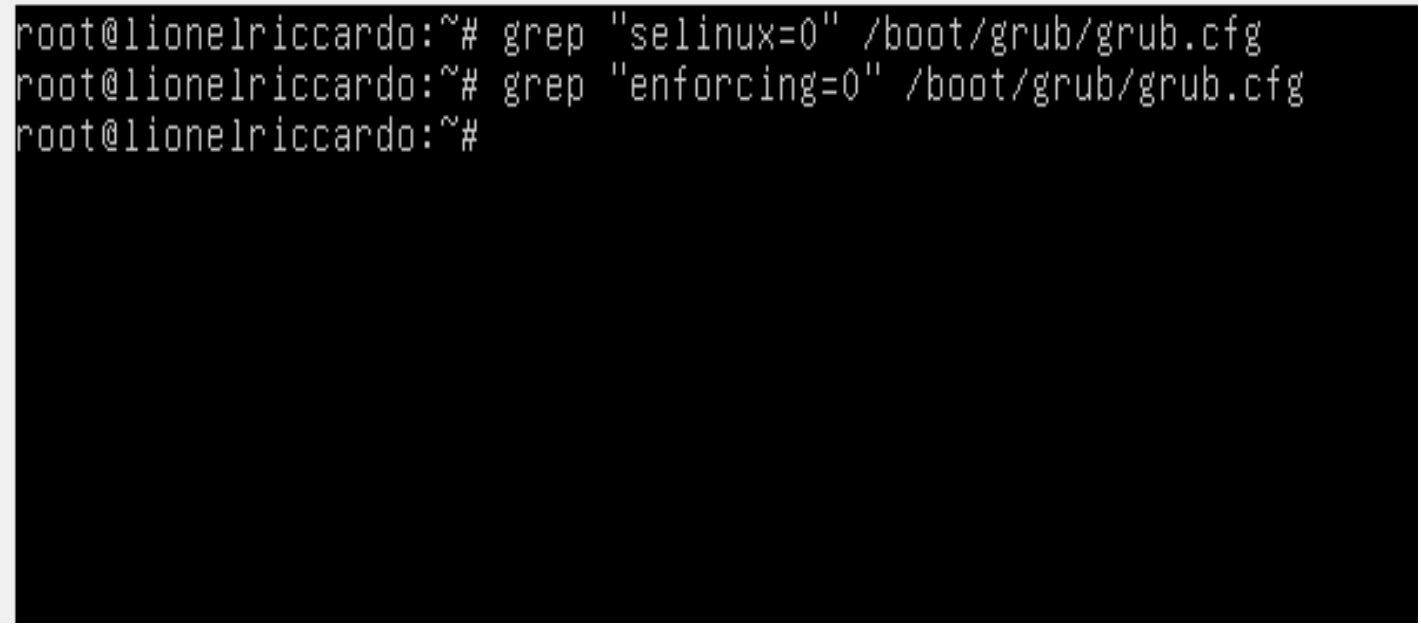
SECTION 2:

Dans cette capture d’écran, nous avons vérifié que SELinux n’était pas désactivé dans le fichier de configuration du chargeur de démarrage GRUB.

Nous avons utilisé deux commandes pour effectuer cette vérification : la commande grep "selinux=0" /boot/grub/grub.cfg pour rechercher si l’option selinux=0 (qui désactive SELinux) est présente dans le fichier de configuration de GRUB, et la commande grep "enforcing=0" /boot/grub/grub.cfg pour vérifier si SELinux est configuré en mode permissif via l’option enforcing=0.

Les deux commandes n’ont retourné aucun résultat, ce qui confirme que SELinux n’est pas désactivé ou forcé en mode permissif dans la configuration GRUB. Cette vérification garantit que SELinux peut fonctionner correctement en mode enforcing, comme demandé dans le projet.

Enfin, cette étape est essentielle pour s'assurer que SELinux est bien appliqué au niveau du démarrage du système.



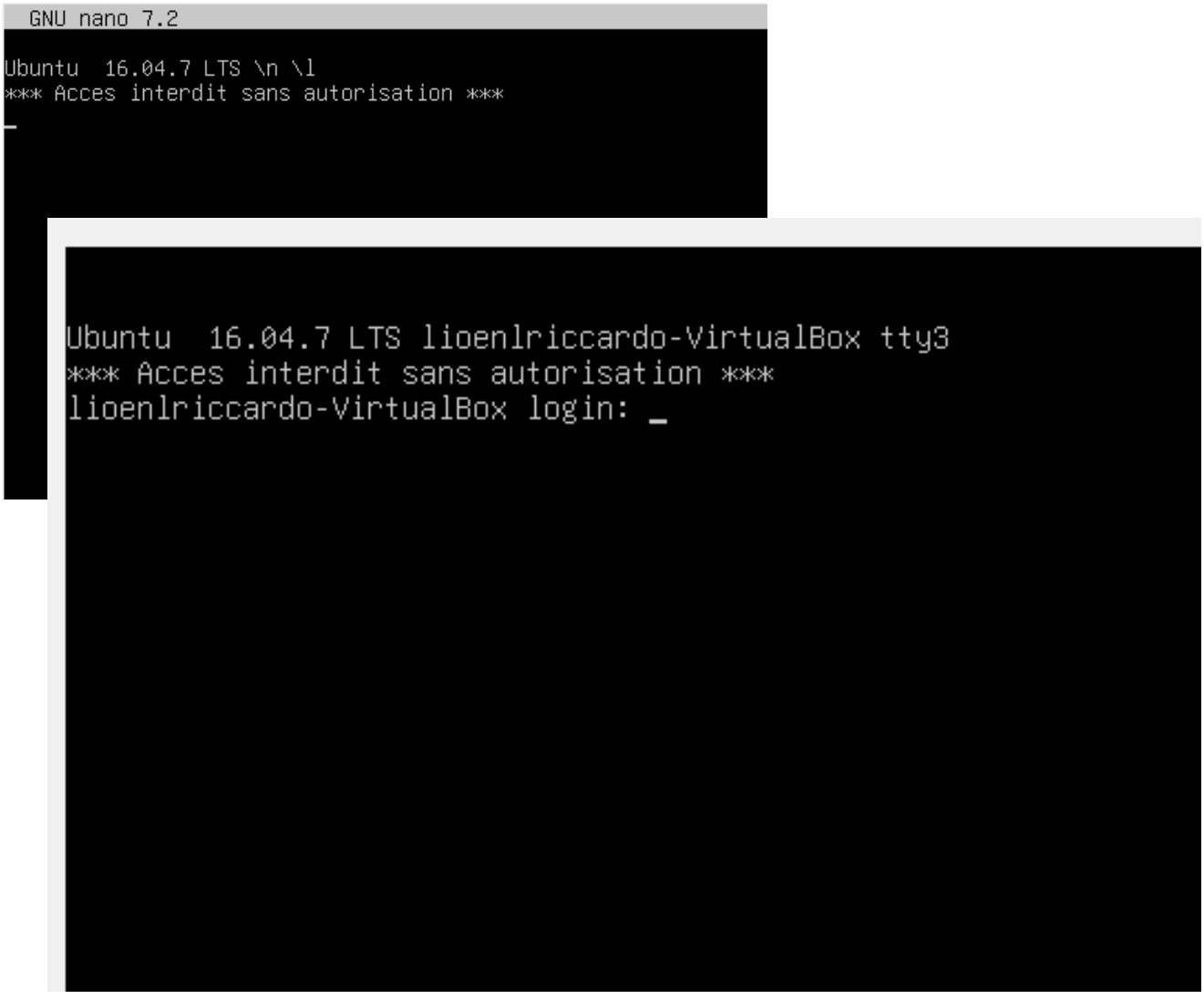
SECTION 3:

Dans cette étape, nous avons commencé par créer un nouvel utilisateur nommé victor, comme demandé dans le projet. Pour cela, nous avons utilisé la commande sudo useradd victor afin d'ajouter l'utilisateur au système. Une fois cette étape réalisée, nous avons associé cet utilisateur au rôle SELinux staff_u. Cette association est importante car elle définit les permissions de sécurité de l'utilisateur en fonction des politiques SELinux. Pour effectuer cette configuration, nous avons utilisé la commande sudo semanage login -a -s staff_u victor.

Enfin, pour nous assurer que tout était correctement configuré, nous avons vérifié l'association en exécutant la commande sudo semanage login -l | grep victor. Le résultat a confirmé que l'utilisateur victor était bien lié au rôle staff_u, comme demandé. Cette étape garantit que l'utilisateur est correctement intégré dans le cadre des règles de sécurité définies par SELinux.

SECTION 4:

Dans cette étape, nous avons configuré un message de bannière qui s'affiche avant la connexion, comme demandé dans le projet. Ce message a pour but de prévenir que l'accès au système est interdit sans autorisation. Pour cela, nous avons modifié le fichier /etc/issue avec la commande sudo nano /etc/issue et y avons ajouté le texte * Accès interdit sans autorisation ***. Une fois les modifications enregistrées, nous avons redémarré la machine pour vérifier si le message s'affichait correctement. Comme on peut le voir sur la capture d'écran, le message apparaît bien avant l'écran de connexion, ce qui prouve que la configuration a été réalisée avec succès.**

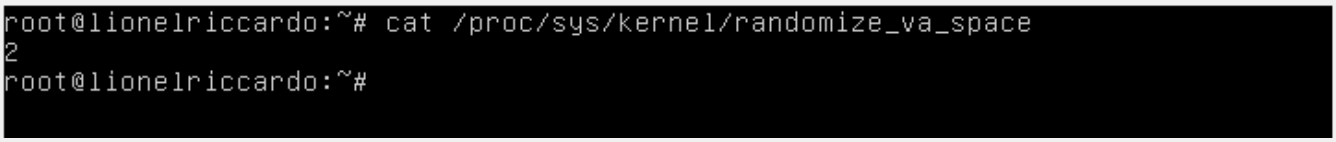


SECTION 5:

Dans cette étape, nous avons vérifié que l'ASLR était activé pour sécuriser le système. Avec la commande sudo sysctl kernel.randomize_va_space, nous avons confirmé que la valeur était 2, ce qui signifie que l'ASLR est activé.

Nous avons aussi utilisé sysctl -a | grep randomize_va_space et cat /proc/sys/kernel/randomize_va_space pour confirmer cette configuration.

Enfin, la commande dmesg | grep randomize_va_space n'a signalé aucun problème. La capture d'écran montre que tout est correctement configuré.



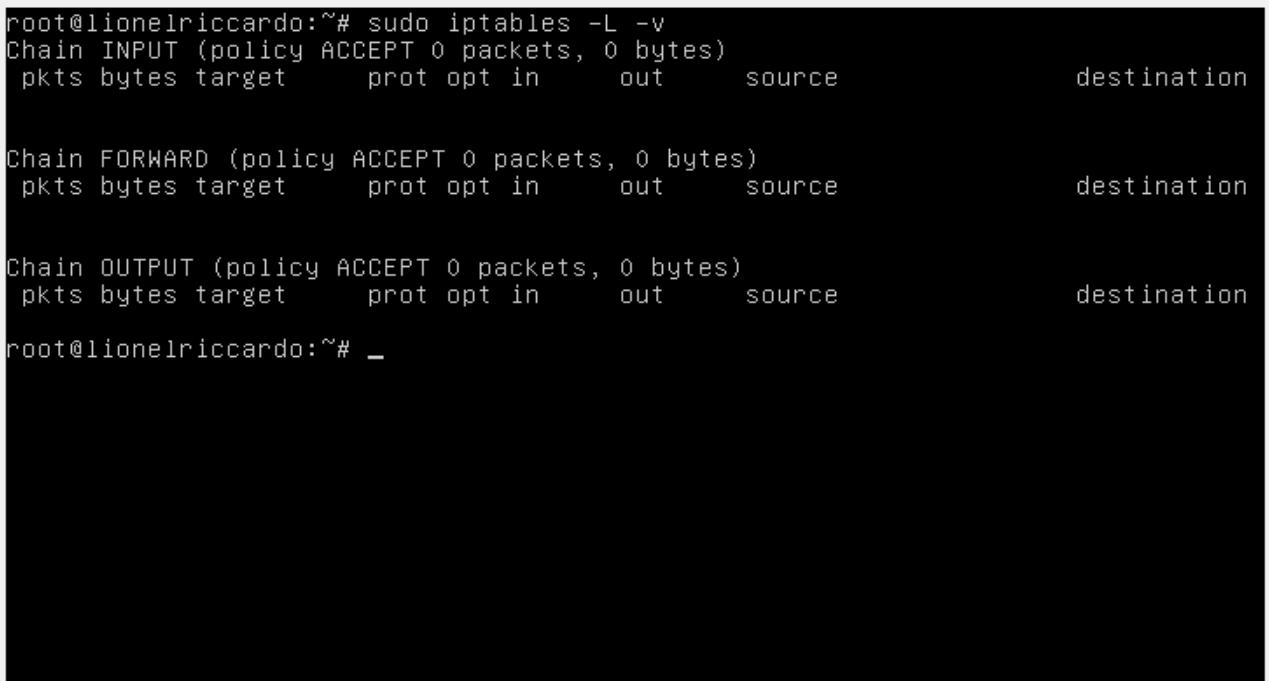
SECTION 6:

Dans cette section, nous avons configuré le pare-feu iptables pour bloquer les connexions entrantes par défaut, conformément aux exigences du projet.

Initialement, nous avons vérifié les règles en place avec la commande sudo iptables -L -v, qui montrait que la politique par défaut pour la chaîne INPUT était réglée sur ACCEPT, autorisant toutes les connexions entrantes.

Nous avons ensuite modifié cette configuration en réglant la politique de la chaîne INPUT sur DROP à l'aide de la commande sudo iptables -P INPUT DROP. Pour confirmer que le changement avait bien été pris en compte, nous avons réexécuté la commande sudo iptables -L -v, et la sortie montrait clairement que la politique par défaut pour INPUT était désormais réglée sur DROP.

Cette configuration garantit que toutes les connexions entrantes non autorisées sont bloquées, renforçant ainsi la sécurité du système.



```
root@lionelriccardo:~# sudo iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@lionelriccardo:~# sudo iptables -P INPUT DROP
root@lionelriccardo:~#
```

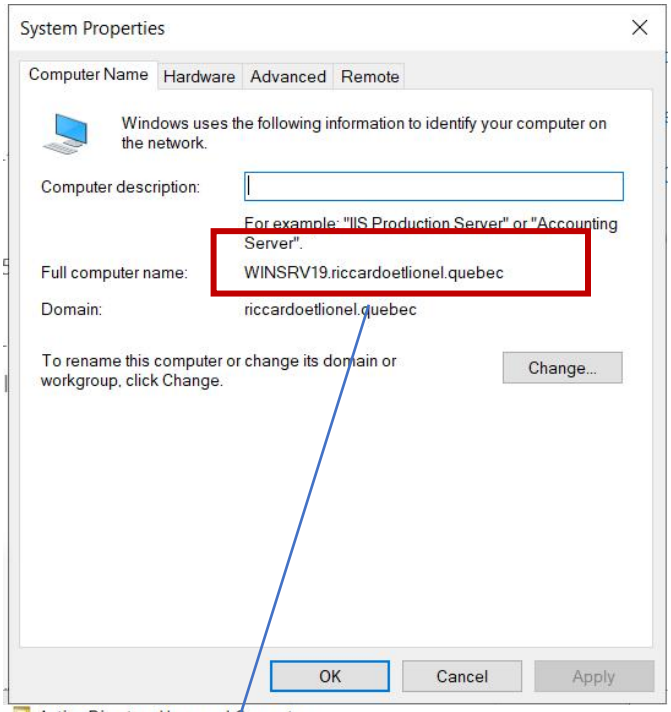
```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@lionelriccardo:~# sudo iptables -P INPUT DROP
root@lionelriccardo:~# sudo iptables -L -v
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

root@lionelriccardo:~#
```

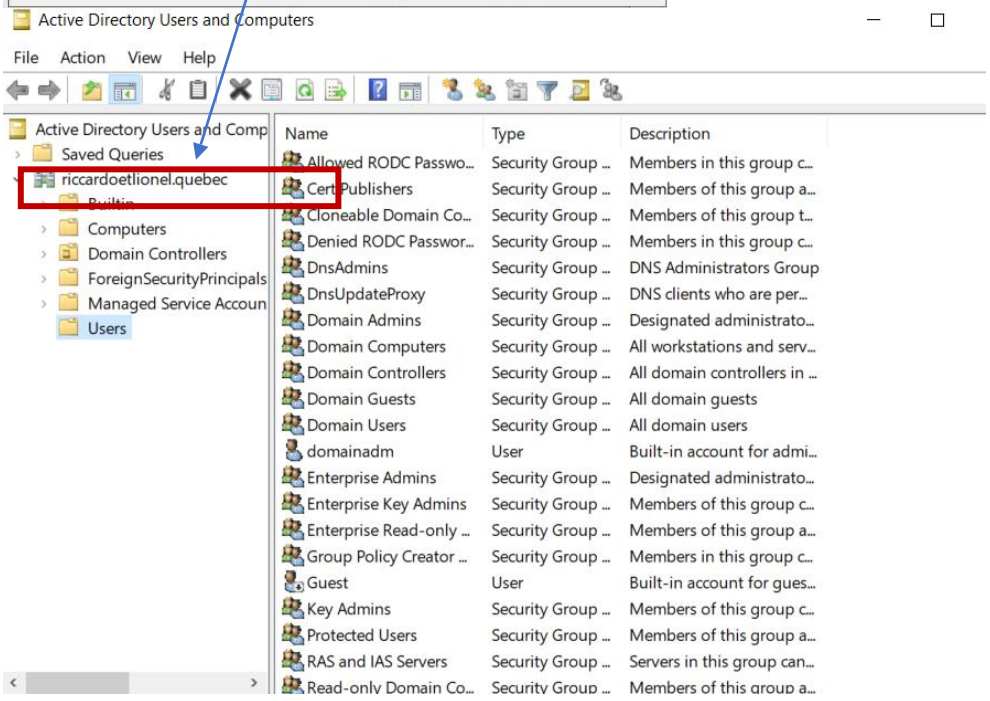
CONCLUSION:

Pour conclure, ce projet nous a permis de sécuriser un système Linux en suivant plusieurs étapes importantes. Nous avons activé SELinux, vérifié l'ASLR, configuré un message d'avertissement à l'écran de connexion et mis en place un pare-feu pour bloquer les connexions entrantes. Chaque étape a été vérifiée pour s'assurer qu'elle était bien appliquée, ce qui garantit que le système est maintenant mieux protégé.

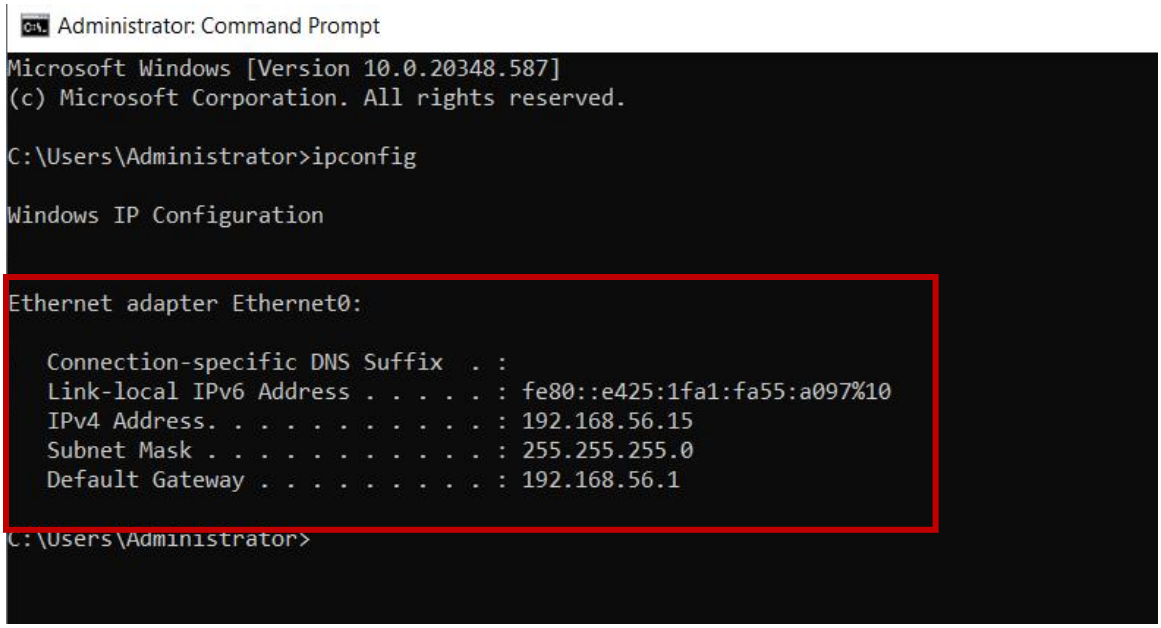
Partie 2 - Windows

Voici mon Windows Seveur nom:**WINSRV2019** nom de domaine : **riccardoetlionel.quebec**

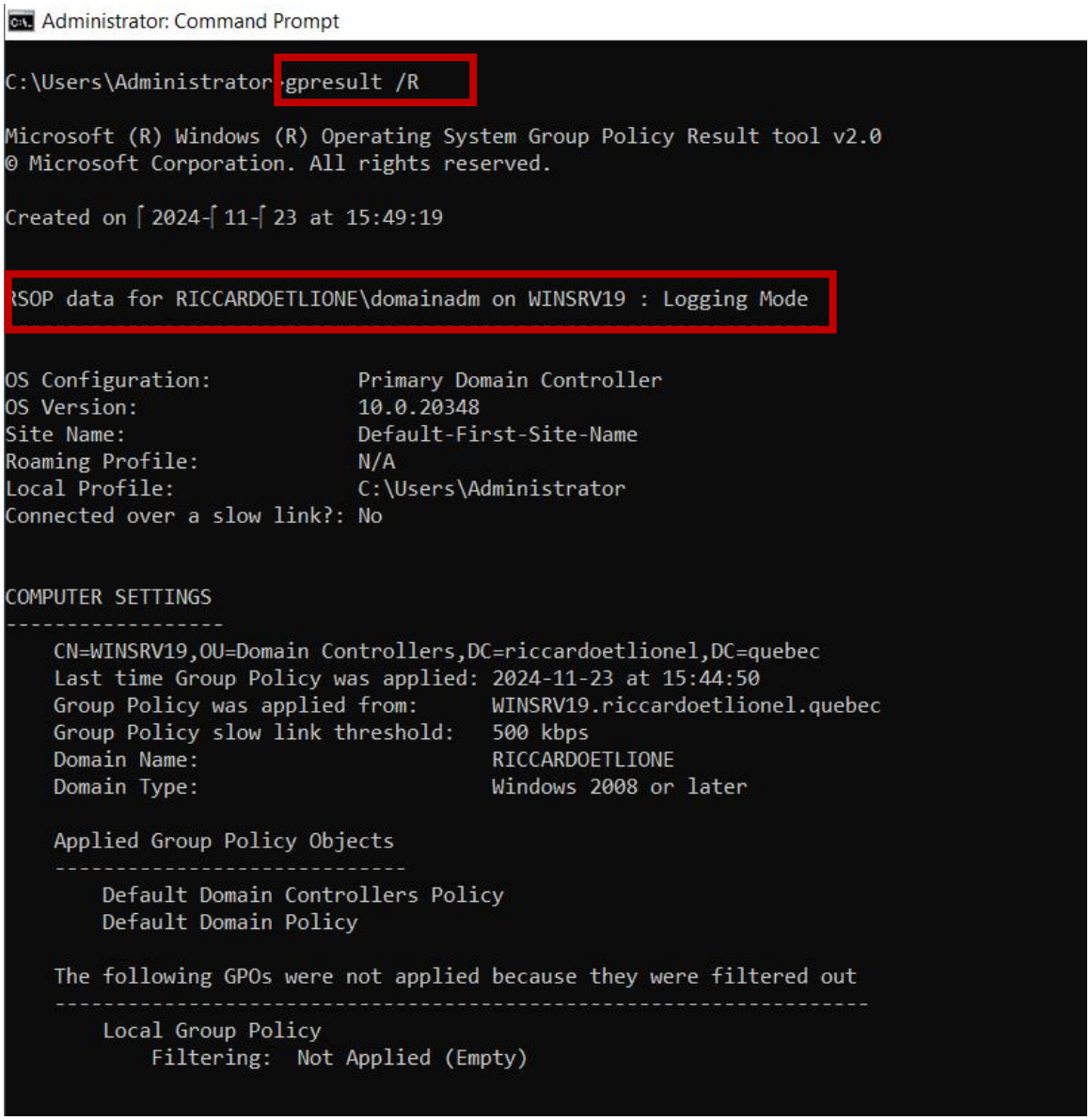


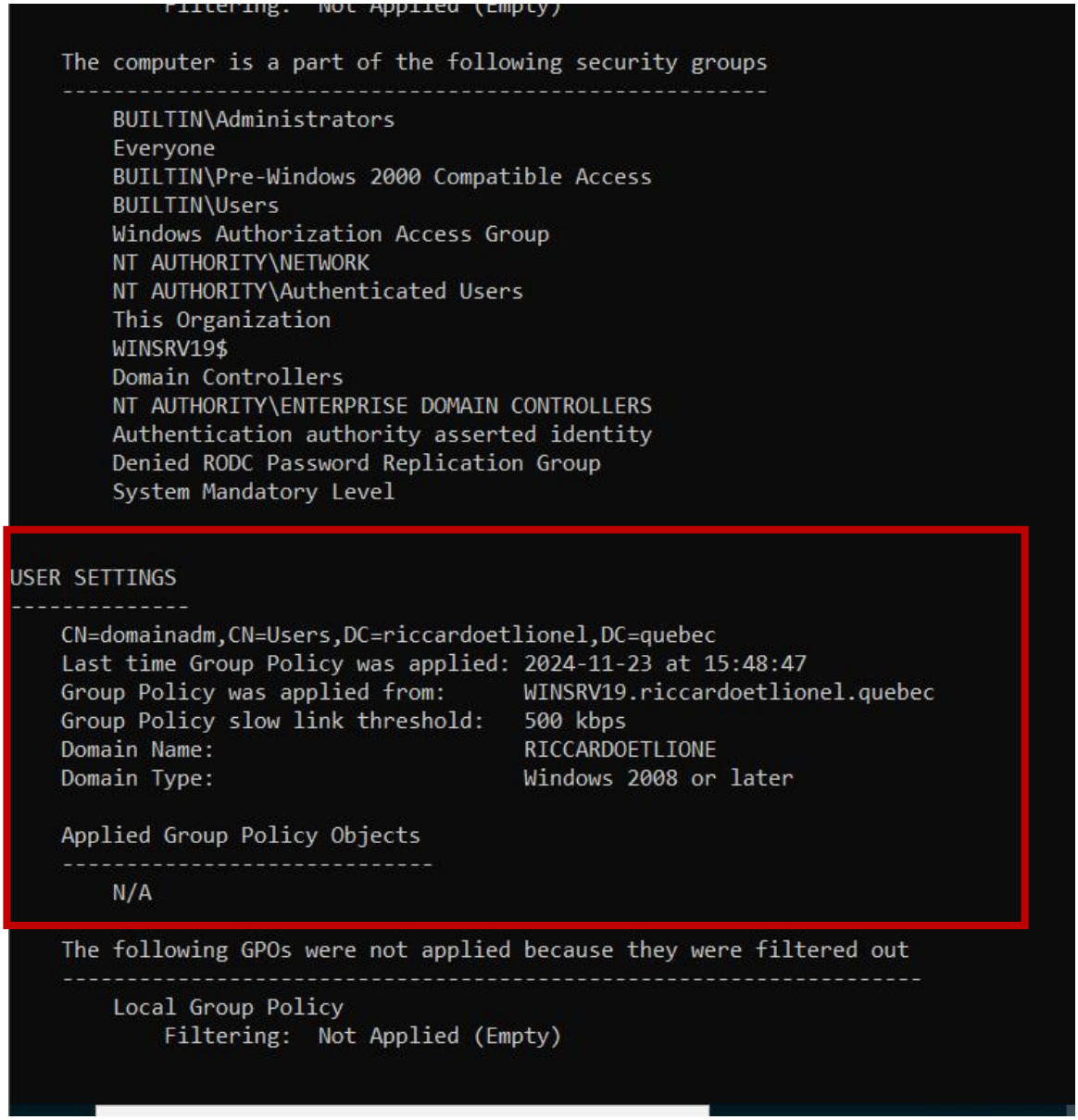


Voici L'information de ma carte réseau :



1- Pour renomme le compte "Administrator" en "domainadm" :
   - Ouvrez une session avec un compte ayant des privilèges administratifs.
   - Ouvrez l'outil **Utilisateurs et ordinateurs Active Directory**
   - Dans l'arborescence de la console, développez le domaine dans**Users**, localisez le compte "Administrator'
   - Faites un clic droit sur le compte "Administrator" et sélectionner**Renommer**.
   - Entrez "domainadm" comme nouveau nom et appuyez**Entrée**.
   - Ouvrez l'invite de commande en tant qu'administrateur.
   - Tapez **gpresult /R** et appuyez sur Entrée pour obtenir un résumé des paramètres de stratégie de groupe appliqués.

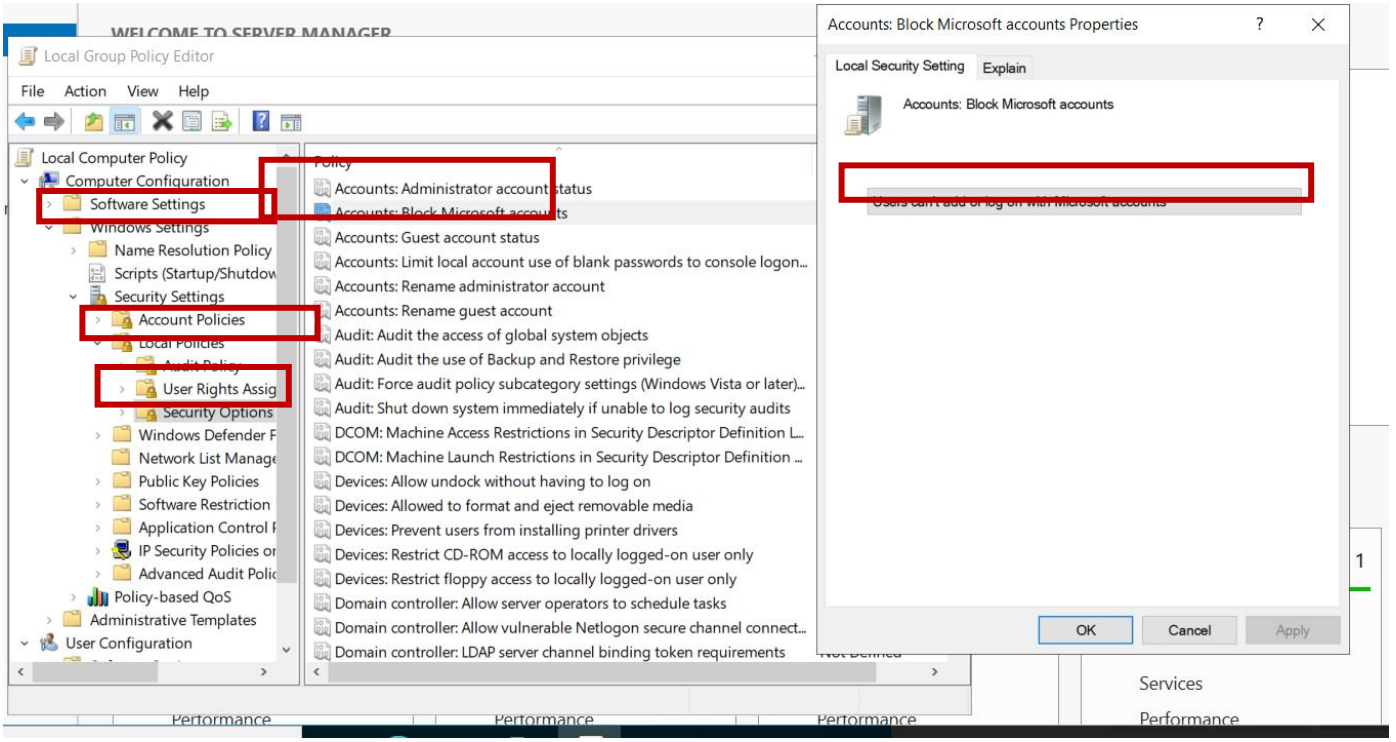

Administrator: Command Prompt
C:\Users\Administrator>gpresult /R
Microsoft (R) Windows (R) Operating System Group Policy Result tool v2.0
© Microsoft Corporation. All rights reserved.
Created on ‎2024-‎11-‎23 at 15:49:19
\SOP data for RICCARDOETLIONE\domainadm on WINSRV19 : Logging Mode
OS Configuration: Primary Domain Controller
OS Version: 10.0.20348
Site Name: Default-First-Site-Name
Roaming Profile: N/A
Local Profile: C:\Users\Administrator
Connected over a slow link?: No
COMPUTER SETTINGS
CN=WINSRV19,OU=Domain Controllers,DC=riccardoetlionel,DC=quebec
Last time Group Policy was applied: 2024-11-23 at 15:44:50
Group Policy was applied from: WINSRV19.riccardoetlionel.quebec
Group Policy slow link threshold: 500 kbps
Domain Name: RICCARDOETLIONE
Domain Type: Windows 2008 or later
Applied Group Policy Objects
Default Domain Controllers Policy
Default Domain Policy
The following GPOs were not applied because they were filtered out
Local Group Policy
Filtering: Not Applied (Empty)



2- Pour empêcher l'arrêt d'un serveur sans se connecter Voici comment procéder :

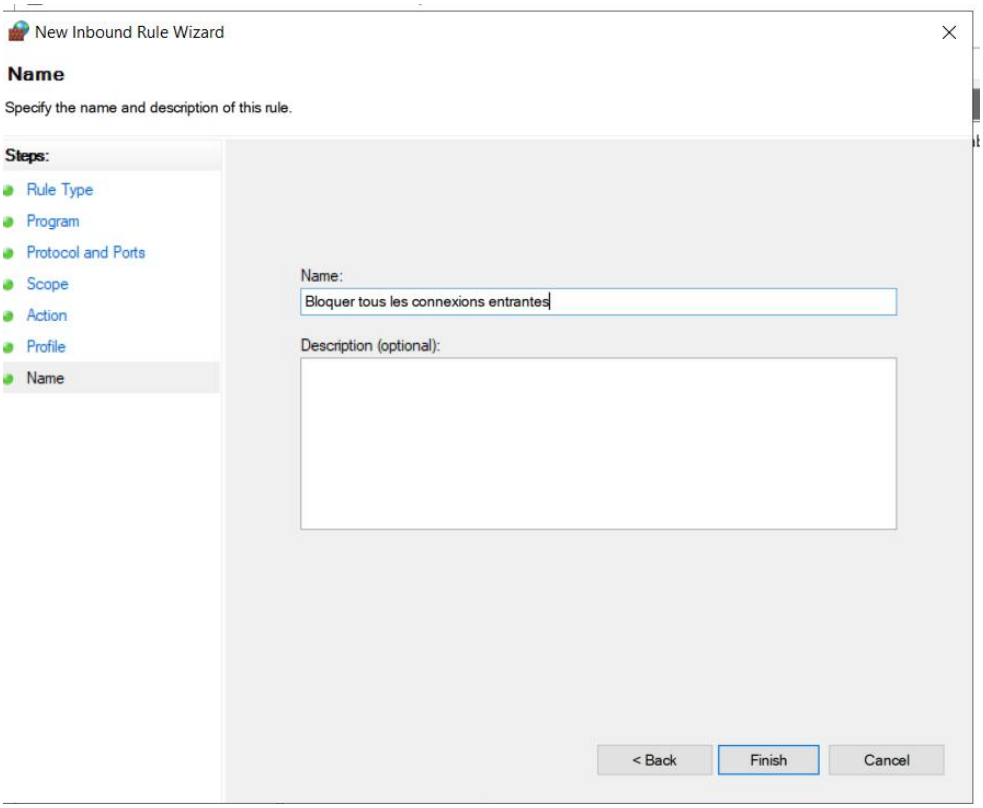
- **Ouvrir l'Éditeur de stratégie de groupe locale**
- Appuyez sur win + R, tapez gpedit.msc et appuyée sur**Entrée.**
- Allez dans **Configuration de l'ordinateur > Paramètres Windows > Paramètres de sécurité > Stratégies locales > Options de sécurité**
- Recherche l'option **Arrêt : autoriser le système à être arrêté sans avoir à se connecter.**
- Double-Cliquez dessus et sélectionnez **Désactivé**
- Cliquez sur Appliquer puis sur **OK**
- **Redémarrer le Serveur pour que les modifications prennent effet.**

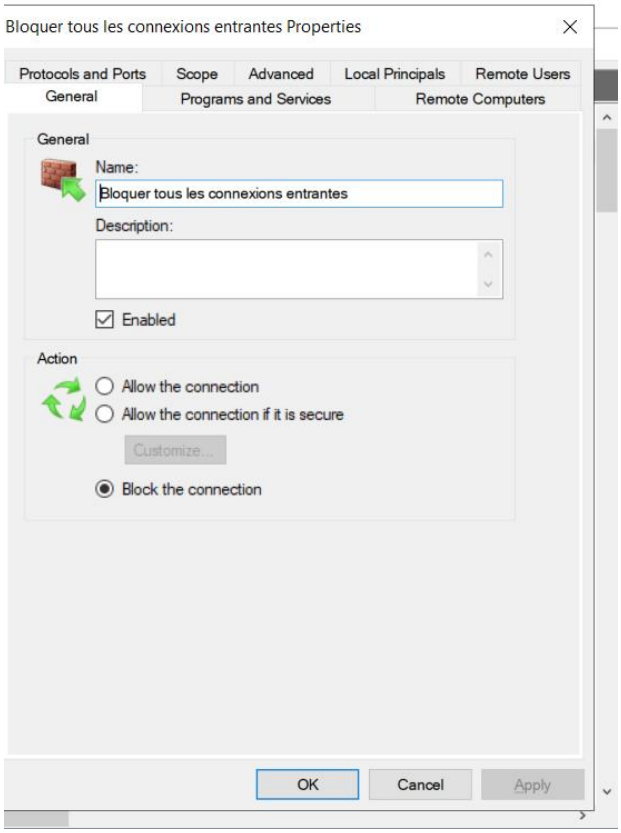
- Cliquez sur **Appliquer** puis sur **OK**

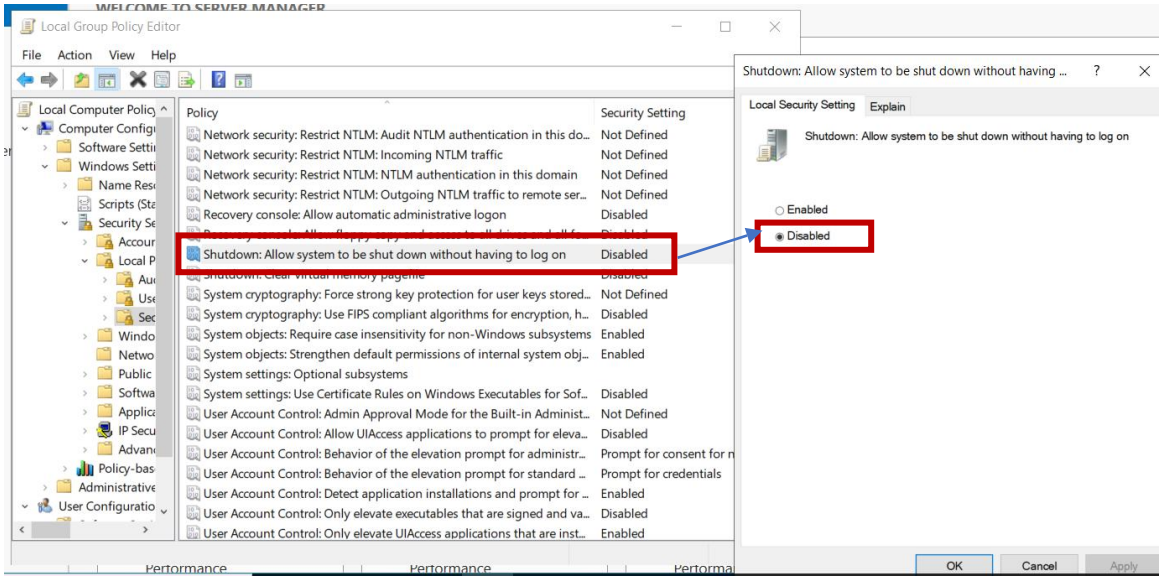


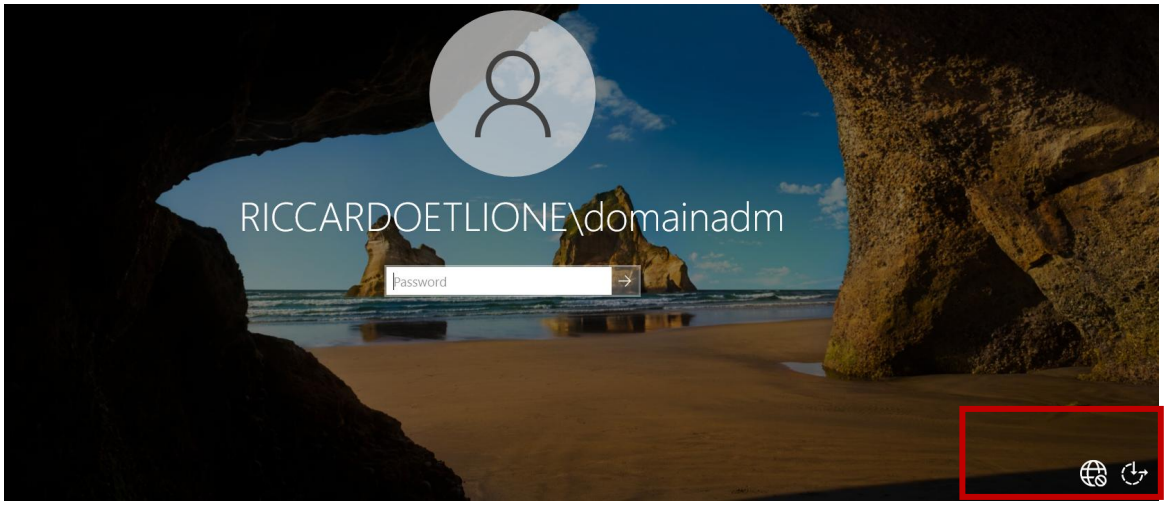
4- Pour vous assurer que le pare-feu bloque par défaut les connexions entrantes voici comment procéder :

- **Ouvrir le Pare-feu Windows avec sécurité avancée**:
- Appuyez sur win + R, tapez wf.msc et appuyez sur**Entrée**
- Dans le volet de gauche, sélectionnez**Règle de trafic entrant**
- Dans le volet de droite, cliquez sur**Nouvelle règle**.
- Sélectionnez **Personnalisé** et cliquez sur **suivant**
- Choisissez **Tous les programmes** et cliquez sur **Suivant**
- Dans la page **Protocole et ports**, laissez les paramètres par défaut et cliquez sur **Suivant**
- Dans la Page **Étendue**, Laissez les paramètres et cliquez sur**Suivant**
- Dans la page **Action**, sélectionnez **Bloquer la connexion** et cliquer sur **Suivant**
- Dans la page **Profil**, cochez tous les cases (Domaine,Privé,Public) et cliquez sur **Suivant**
- Donnez un nom à la règle et cliquez sur**Terminer**


New Inbound Rule Wizard
Name
Specify the name and description of this rule.
Steps:
Rule Type
Program
Protocol and Ports
Scope
Action
Profile
Name
Name:
Bloquer tous les connexions entrantes
Description (optional):
< Back
Finish
Cancel


Bloquer tous les connexions entrantes Properties
Protocols and Ports
Scope
Advanced
Local Principals
Remote Users
General
Programs and Services
Remote Computers
General
Name:
Bloquer tous les connexions entrantes
Description:
Enabled
Action
Allow the connection
Allow the connection if it is secure
Customize...
Block the connection
OK
Cancel
Apply





3- Pour empêcher les utilisateurs de se connecter avec un compte Microsoft sur un ordinateur Windows :

- **Ouvrir l'Éditeur de stratégie de groupe locale**
- Appuyez sur win + R, tapez gpedit.msc et appuyée sur**Entrée.**
- Allez dans **Configuration de l'ordinateur > Paramètres Windows > Paramètres de sécurité > Stratégies locales > Options de sécurité**
- Recherchez **Comptes : bloquez les comptes Microsoft**
- Double-cliquez dessus et sélectionnez**Les utilisateurs ne peuvent pas ajouter ou se connecter avec des comptes Microsoft**