

MEZZOUR Omar

Rapport **Mini-projet**

Administration Linux
Avancé





Sujet Mini-projet

Intégration de **SELinux**
dans **Apache** via le Module
Dynamique SELinux pour la
Sécurisation et le
Déploiement

SOMMAIRE

I- Introduction

II- Objectif du projet

III- Recherche et documentation

1- Qu'est ce que SELinux ?

2- Qu'est ce que Apache et les modules ?

IV- Installation et configuration

1- Vérification de l'état de SELinux

2- Installation des outils SELinux

3- Installation d'Apache

4- Activation du support SELinux pour Apache

5- Démarrage et activation d'Apache

6- Configuration du pare-feu pour HTTP

V- Vérifications

1- Statut d'Apache

2- Logs d'Apache

3- Application SELinux

VI- Test

VII- Conclusion

INTRODUCTION

Le projet d'examination, de configuration et de déploiement du module SELinux pour Apache vise à renforcer la sécurité et la compatibilité de l'environnement Apache avec SELinux.

Ce module dynamique offre une solution pour intégrer les fonctionnalités de SELinux dans la configuration d'Apache, même en l'absence de spécificités SELinux natives dans Apache lui-même.

En sécurisant davantage le serveur web Apache grâce à SELinux, ce projet vise à offrir une couche supplémentaire de protection et à renforcer la robustesse du système contre les menaces potentielles.

Objectif du projet

L'objectif principal de ce projet est d'implémenter le module SELinux pour Apache, permettant ainsi une intégration transparente des politiques de sécurité de SELinux dans la configuration d'Apache.

Cela implique l'examen approfondi des paramètres de sécurité requis, la configuration adéquate du module pour s'aligner avec les politiques SELinux en place, et enfin, le déploiement réussi de cette solution pour renforcer la sécurité du serveur Apache.

En combinant les fonctionnalités de SELinux avec Apache, cet effort vise à limiter les risques potentiels d'exploitation tout en garantissant un fonctionnement optimal du serveur web.

RECHERCHE et DOCUMENTATION

1. Qu'est ce que SELinux ?

SELinux, acronyme de Security-Enhanced Linux, constitue une extension du noyau Linux qui implémente un contrôle d'accès obligatoire (MAC – Mandatory Access Control) plus rigoureux que le modèle DAC (Discretionary Access Control) classique.

Il fonctionne en définissant des politiques de sécurité détaillées pour restreindre spécifiquement les accès des utilisateurs, des processus et des fichiers à différentes ressources du système.

En attribuant des contextes de sécurité à chaque élément, SELinux permet un contrôle plus fin, réduisant ainsi les risques d'exploitation en limitant les actions possibles des utilisateurs et des programmes, ce qui renforce la sécurité du système dans son ensemble, bien que sa configuration requière une connaissance approfondie des politiques de sécurité et puisse initialement présenter une courbe d'apprentissage complexe pour les utilisateurs moins expérimentés.

RECHERCHE et DOCUMENTATION

2. Qu'est ce que Apache et les modules ?

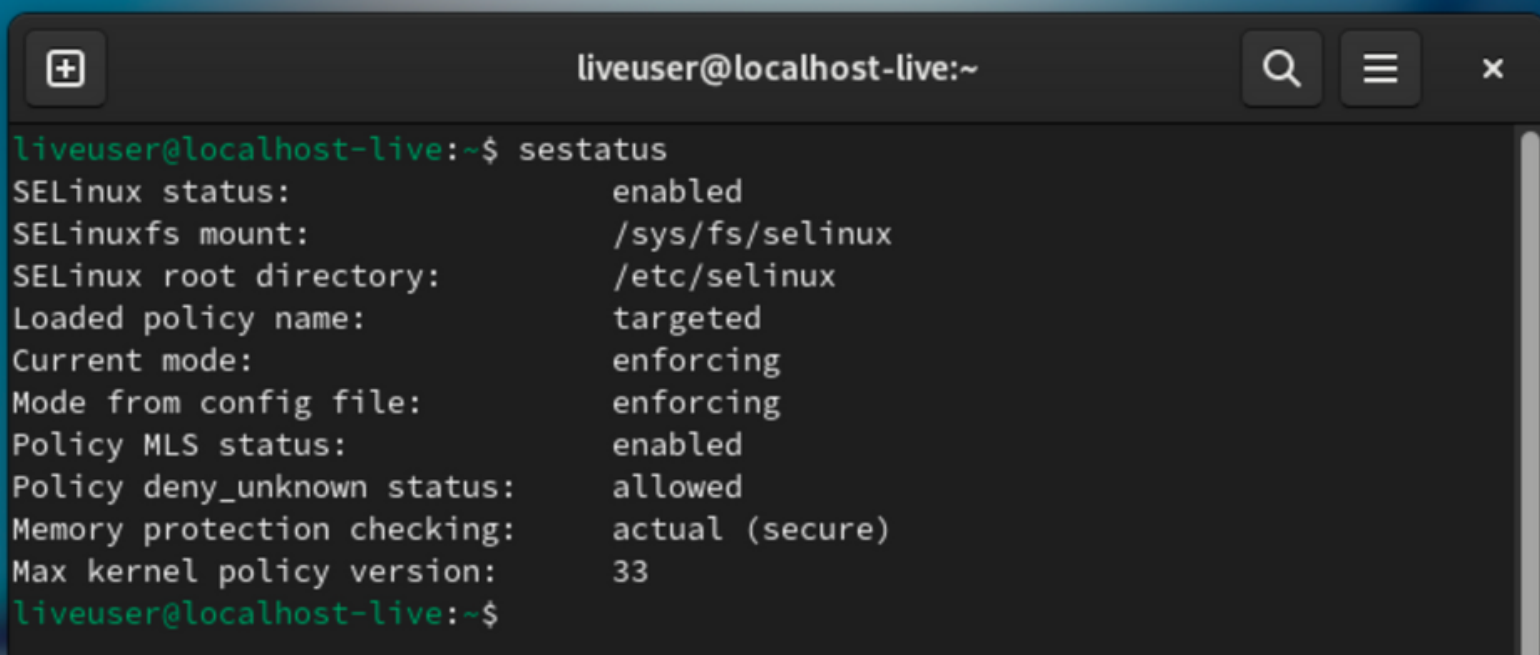
Apache est un serveur web open source populaire, largement utilisé pour héberger des sites web et fournir des contenus sur internet. Il constitue le fer de lance de nombreuses infrastructures web en offrant une plateforme stable et extensible pour répondre aux demandes HTTP.

Les modules Apache sont des extensions logicielles qui enrichissent ses fonctionnalités de base en ajoutant des fonctionnalités spécifiques, telles que la gestion de la sécurité, la compression, la prise en charge de langages de programmation, et bien plus encore.

Ces modules permettent aux administrateurs de personnaliser le comportement du serveur selon les besoins spécifiques de leurs applications web, offrant ainsi une souplesse et une adaptabilité accrues à l'ensemble du serveur.

INSTALLATION et CONFIGURATION

1. Vérification de l'état de SELinux

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows the command 'sestatus' and its output. The output indicates that SELinux is enabled, the mount point is /sys/fs/selinux, the root directory is /etc/selinux, the loaded policy is targeted, and the current mode is enforcing. Other details include the mode from the config file (enforcing), policy MLS status (enabled), policy deny_unknown status (allowed), memory protection checking (actual secure), and max kernel policy version (33).

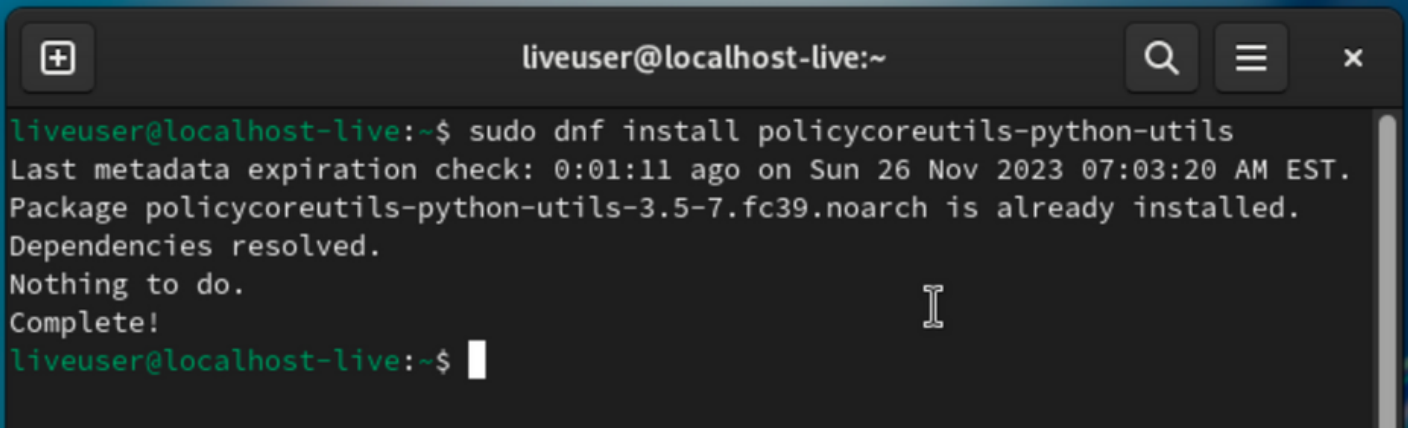
```
liveuser@localhost-live:~$ sestatus
SELinux status:                enabled
SELinuxfs mount:              /sys/fs/selinux
SELinux root directory:       /etc/selinux
Loaded policy name:            targeted
Current mode:                  enforcing
Mode from config file:         enforcing
Policy MLS status:             enabled
Policy deny_unknown status:    allowed
Memory protection checking:    actual (secure)
Max kernel policy version:     33
liveuser@localhost-live:~$
```

La commande "**sestatus**" sur Fedora affiche le statut actuel de SELinux sur le système. Lorsque le résultat indique "enabled" et que le mode est "**enforcing**", cela signifie que SELinux est activé et qu'il fonctionne en mode "**enforcing**".

En mode "**enforcing**", SELinux applique strictement les politiques de sécurité définies, ce qui signifie qu'il va fortement restreindre les actions des utilisateurs et des processus pour assurer une meilleure sécurité système.

INSTALLATION et CONFIGURATION

2. Installation des outils SELinux

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons. The terminal shows the command 'sudo dnf install policycoreutils-python-utils' being executed. The output indicates that the package 'policycoreutils-python-utils-3.5-7.fc39.noarch' is already installed, dependencies are resolved, and the installation is complete. The prompt returns to 'liveuser@localhost-live:~\$' with a cursor.

```
liveuser@localhost-live:~$ sudo dnf install policycoreutils-python-utils
Last metadata expiration check: 0:01:11 ago on Sun 26 Nov 2023 07:03:20 AM EST.
Package policycoreutils-python-utils-3.5-7.fc39.noarch is already installed.
Dependencies resolved.
Nothing to do.
Complete!
liveuser@localhost-live:~$
```

Ce package contient des utilitaires Python qui sont essentiels pour gérer les politiques de sécurité SELinux sur un système Fedora.

En exécutant cette commande avec les privilèges sudo, nous demandons au gestionnaire de paquets DNF d'installer ces outils, ce qui facilitera la manipulation et la configuration des règles de sécurité SELinux sur votre système Fedora.

INSTALLATION et CONFIGURATION

3. Installation d'Apache

```
liveuser@localhost-live:~$ sudo dnf install httpd
Fedora 39 - aarch64                               1.0 MB/s | 86 MB      01:26
Fedora 39 openh264 (From Cisco) - aarch64        149 B/s | 2.5 kB      00:17
Fedora 39 - aarch64 - Updates                     961 kB/s | 15 MB      00:16
Package httpd-2.4.57-3.fc39.aarch64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
liveuser@localhost-live:~$
```

Cette commande est utilisée pour installer le paquet HTTPD, qui correspond au serveur web Apache. En l'exécutant avec les privilèges sudo, on demande au gestionnaire de paquets DNF d'installer le logiciel Apache HTTP Server sur notre système Fedora.

Une fois installé, nous pourrions configurer et utiliser Apache pour héberger des sites web ou des applications web sur notre machine.

INSTALLATION et CONFIGURATION

4. Activation du support SELinux pour Apache

Vérifions d'abord si “httpd_use_nfs” est disponible :

```
liveuser@localhost-live:~$ sudo getsebool -a | grep httpd_use_nfs
httpd_use_nfs --> off
```

“httpd_use_nfs” est un paramètre de configuration pour SELinux spécifique à Apache.

Lorsqu'il est activé, il autorise le serveur web Apache à accéder aux partages NFS (Network File System) de manière sécurisée, conformément aux politiques de sécurité définies par SELinux.

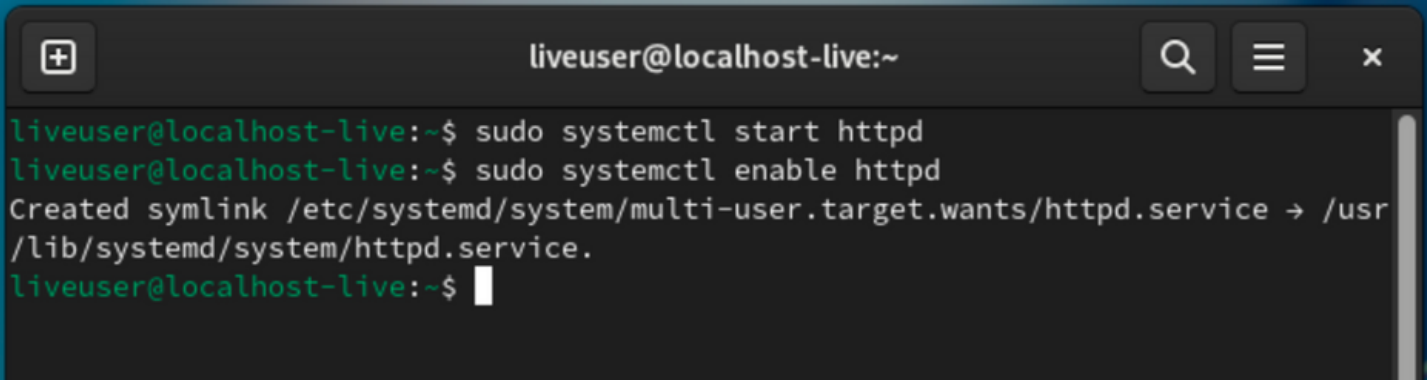
On remarque que le “httpd_use_nfs” est désactivé, donc il faut l'activer et revérifier s'il est activé :

```
liveuser@localhost-live:~$ sudo setsebool -P httpd_use_nfs 1
liveuser@localhost-live:~$ sudo getsebool -a | grep httpd_use_nfs
httpd_use_nfs --> on
liveuser@localhost-live:~$
```



INSTALLATION et CONFIGURATION

5. Démarrage et activation d'Apache

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal shows the following commands and output:

```
liveuser@localhost-live:~$ sudo systemctl start httpd
liveuser@localhost-live:~$ sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
liveuser@localhost-live:~$
```

La commande "**sudo systemctl start httpd**" est utilisée pour démarrer le service Apache. En exécutant cette commande avec les privilèges sudo, on active le serveur web Apache.

Ensuite, "**sudo systemctl enable httpd**" est utilisée pour activer le démarrage automatique du service Apache au démarrage du système. Cela garantit qu'Apache se lance automatiquement à chaque démarrage de Fedora, assurant ainsi la disponibilité du serveur web dès le démarrage du système.

INSTALLATION et CONFIGURATION

6. Configuration du pare-feu pour HTTP

Activer le pare-feu est crucial pour sécuriser un système informatique en contrôlant le trafic réseau entrant et sortant, limitant ainsi les accès non autorisés et réduisant la surface d'attaque potentielle.

Cela aide à prévenir les intrusions, à bloquer les logiciels malveillants et à protéger la confidentialité des données en filtrant le trafic suspect, assurant ainsi la conformité aux normes de sécurité et offrant une défense proactive contre les menaces en ligne.

INSTALLATION et CONFIGURATION

6. Configuration du pare-feu pour HTTP

```
liveuser@localhost-live:~$ sudo firewall-cmd --permanent --add-service=http
success
liveuser@localhost-live:~$ sudo firewall-cmd --reload
success
liveuser@localhost-live:~$
```

La première commande ajoute le service HTTP aux règles du pare-feu de façon permanente. Cela permet de permettre le trafic HTTP à travers le pare-feu, autorisant les requêtes entrantes sur le port 80, qui est le port par défaut pour le trafic HTTP.

La seconde commande recharge la configuration du pare-feu pour prendre en compte les modifications effectuées, permettant ainsi l'application des nouvelles règles de manière immédiate.

VÉRIFICATIONS

1. Statut d'Apache

```
liveuser@localhost-live:~ — sudo systemctl status httpd
liveuser@localhost-live:~$ sudo systemctl status httpd
• httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: di>
  Drop-In: /usr/lib/systemd/system/service.d
           └─10-timeout-abort.conf
  Active: active (running) since Sun 2023-11-26 07:08:53 EST; 1min 46s ago
  Docs: man:httpd.service(8)
  Main PID: 37692 (httpd)
  Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes>
  Tasks: 177 (limit: 2075)
  Memory: 34.8M
  CPU: 199ms
  CGroup: /system.slice/httpd.service
          └─37692 /usr/sbin/httpd -DFOREGROUND
             └─37693 /usr/sbin/httpd -DFOREGROUND
                └─37695 /usr/sbin/httpd -DFOREGROUND
                   └─37696 /usr/sbin/httpd -DFOREGROUND
                      └─37697 /usr/sbin/httpd -DFOREGROUND

Nov 26 07:08:53 localhost-live systemd[1]: Starting httpd.service - The Apache >
Nov 26 07:08:53 localhost-live (httpd)[37692]: httpd.service: Referenced but un>
Nov 26 07:08:53 localhost-live httpd[37692]: AH00558: httpd: Could not reliably>
Nov 26 07:08:53 localhost-live httpd[37692]: Server configured, listening on: p>
Nov 26 07:08:53 localhost-live systemd[1]: Started httpd.service - The Apache H>
```

VÉRIFICATIONS

1. Statut d'Apache

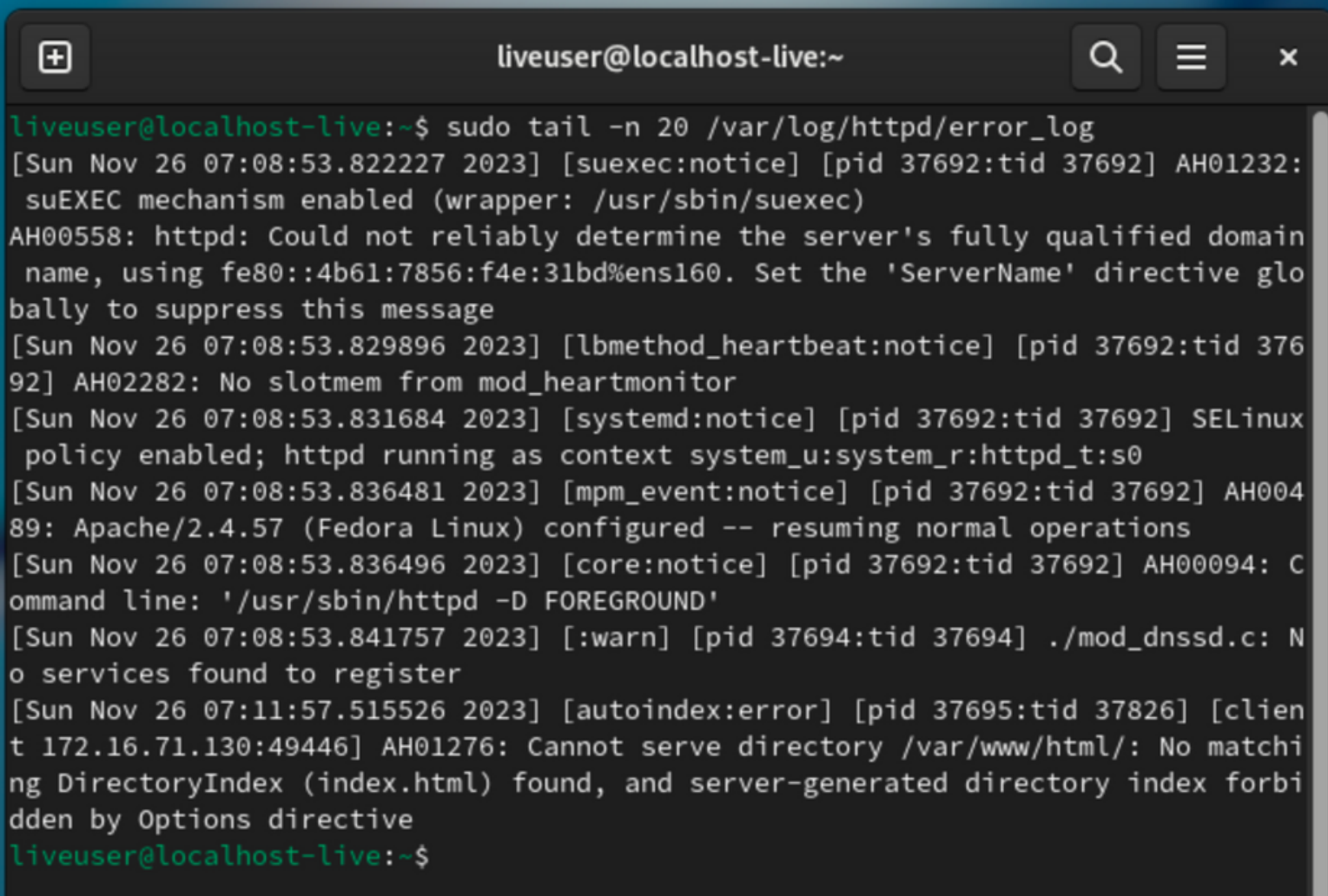
La commande "**sudo systemctl status httpd**" est utilisée pour vérifier le statut actuel du service Apache. En l'exécutant avec les privilèges sudo, elle affiche des informations détaillées sur l'état du service, y compris s'il est en cours d'exécution, les derniers logs d'activité et d'autres détails pertinents sur son fonctionnement.

Cela permet de vérifier si le service Apache fonctionne correctement et de diagnostiquer d'éventuels problèmes éventuels.

On remarque que le statut d'Apache est **activé**, donc il fonctionne correctement.

VÉRIFICATIONS

2. Logs d'Apache

A terminal window titled 'liveuser@localhost-live:~' with search, menu, and close icons in the title bar. The terminal displays the output of the command 'sudo tail -n 20 /var/log/httpd/error_log'. The logs show various system and Apache messages, including suEXEC mechanism enabled, SELinux policy enabled, and an error about a missing directory index.

```
liveuser@localhost-live:~$ sudo tail -n 20 /var/log/httpd/error_log
[Sun Nov 26 07:08:53.822227 2023] [suexec:notice] [pid 37692:tid 37692] AH01232:
suEXEC mechanism enabled (wrapper: /usr/sbin/suexec)
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fe80::4b61:7856:f4e:31bd%ens160. Set the 'ServerName' directive glo
bally to suppress this message
[Sun Nov 26 07:08:53.829896 2023] [lbmethod_heartbeat:notice] [pid 37692:tid 376
92] AH02282: No slotmem from mod_heartbeat
[Sun Nov 26 07:08:53.831684 2023] [systemd:notice] [pid 37692:tid 37692] SELinux
policy enabled; httpd running as context system_u:system_r:httpd_t:s0
[Sun Nov 26 07:08:53.836481 2023] [mpm_event:notice] [pid 37692:tid 37692] AH004
89: Apache/2.4.57 (Fedora Linux) configured -- resuming normal operations
[Sun Nov 26 07:08:53.836496 2023] [core:notice] [pid 37692:tid 37692] AH00094: C
ommand line: '/usr/sbin/httpd -D FOREGROUND'
[Sun Nov 26 07:08:53.841757 2023] [[:warn] [pid 37694:tid 37694] ./mod_dnssd.c: N
o services found to register
[Sun Nov 26 07:11:57.515526 2023] [autoindex:error] [pid 37695:tid 37826] [clien
t 172.16.71.130:49446] AH01276: Cannot serve directory /var/www/html/: No matchi
ng DirectoryIndex (index.html) found, and server-generated directory index forbi
dden by Options directive
liveuser@localhost-live:~$
```

Voici les logs, ils contiennent des informations importantes sur le fonctionnement d'Apache avec SELinux.

VÉRIFICATIONS

2. Logs d'Apache

1. Activation de suEXEC

Le message **[suexec:notice]** indique que le mécanisme suEXEC est activé. suEXEC est un mécanisme de sécurité pour exécuter des scripts CGI en tant qu'utilisateurs spécifiques plutôt que sous l'identité de l'utilisateur d'Apache.

2. Avertissement "ServerName"

Le message **AH00558** est un avertissement indiquant qu'Apache ne peut pas déterminer de manière fiable le nom de domaine complet du serveur. Pour résoudre cela, nous pouvons définir la directive **"ServerName"** globalement dans la configuration d'Apache.

3. Politique SELinux activée

Le message **[systemd:notice]** indique que la politique SELinux est activée, et Apache fonctionne sous le contexte **"system_u:system_r:httpd_t:s0"**. Cela montre qu'Apache est exécuté dans le contexte SELinux spécifié, ce qui est une bonne pratique pour la sécurité.

VÉRIFICATIONS

2. Logs d'Apache

4. Avertissement mod_dnssd

Le message **[`:warn`]** indique un avertissement concernant “`mod_dnssd.c`”, qui ne trouve pas de services à enregistrer. Cela peut ne pas être une préoccupation majeure, sauf si on utilise spécifiquement ce module pour des services liés à DNS-SD (Service Discovery).

5. Erreur autoindex

Le message **[`autoindex:error`]** indique qu'il y a une erreur avec la génération de l'index du répertoire. Cela se produit car il n'y a pas de fichier “`index.html`” dans le répertoire “`/var/www/html/`”. On peut soit créer un fichier “`index.html`”, soit configurer la directive “`DirectoryIndex`” pour spécifier les fichiers à utiliser comme index.

Pour résoudre l'erreur autoindex, nous pouvons ajouter une directive “`DirectoryIndex`” dans la configuration d'Apache pour spécifier les fichiers à utiliser comme index dans le répertoire “`/var/www/html/`”.

VÉRIFICATIONS

3. Application SELinux

```
liveuser@localhost-live:~  
liveuser@localhost-live:~$ sudo tail -n 20 /var/log/audit/audit.log  
type=BPF msg=audit(1701000712.726:364): prog-id=127 op=LOAD  
type=BPF msg=audit(1701000712.726:365): prog-id=128 op=LOAD  
type=SERVICE_START msg=audit(1701000712.820:366): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-timedated comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"  
type=SERVICE_START msg=audit(1701000741.737:367): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"  
type=SERVICE_STOP msg=audit(1701000742.898:368): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=systemd-timedated comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"  
type=BPF msg=audit(1701000742.905:369): prog-id=128 op=UNLOAD  
type=BPF msg=audit(1701000742.905:370): prog-id=127 op=UNLOAD  
type=BPF msg=audit(1701000742.905:371): prog-id=126 op=UNLOAD  
type=SERVICE_STOP msg=audit(1701000751.874:372): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=NetworkManager-dispatcher comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"  
type=USER_ACCT msg=audit(1701000813.576:373): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:accounting grantors=pam_unix,pam_localuser acct="liveuser" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=USER_CMD msg=audit(1701000813.576:374): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='cwd="/home/liveuser" cmd=7461696C202D6E203230202F7661722F6C6F672F6874740642F6572726F725F6C6F67 exe="/usr/bin/sudo" terminal=pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=CRED_REFR msg=audit(1701000813.578:375): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_localuser,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=USER_START msg=audit(1701000813.611:376): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:session_open grantors=pam_keyinit,pam_limits,pam_keyinit,pam_limits,pam_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=USER_END msg=audit(1701000813.618:377): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:session_close grantors=pam_keyinit,pam_limits,pam_keyinit,pam_limits,pam_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=CRED_DISP msg=audit(1701000813.619:378): pid=38901 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_localuser,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=SERVICE_STOP msg=audit(1701000832.715:379): pid=1 uid=0 auid=4294967295 ses=4294967295 subj=system_u:system_r:init_t:s0 msg='unit=pcscd comm="systemd" exe="/usr/lib/systemd/systemd" hostname=? addr=? terminal=? res=success'UID="root" AUID="unset"  
type=USER_ACCT msg=audit(1701000865.825:380): pid=38917 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:accounting grantors=pam_unix,pam_localuser acct="liveuser" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=USER_CMD msg=audit(1701000865.825:381): pid=38917 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='cwd="/home/liveuser" cmd=7461696C202D6E203230202F7661722F6C6F672F61756469742F61756469742E6C6F67 exe="/usr/bin/sudo" terminal=pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=CRED_REFR msg=audit(1701000865.827:382): pid=38917 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_env,pam_localuser,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
type=USER_START msg=audit(1701000865.844:383): pid=38917 uid=1000 auid=1000 ses=2 subj=unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:session_open grantors=pam_keyinit,pam_limits,pam_keyinit,pam_limits,pam_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=? addr=? terminal=/dev/pts/0 res=success'UID="liveuser" AUID="liveuser"  
liveuser@localhost-live:~$
```

Ces logs d'audit montrent les activités du système, enregistraient des informations sur les événements importants survenus sur le système.

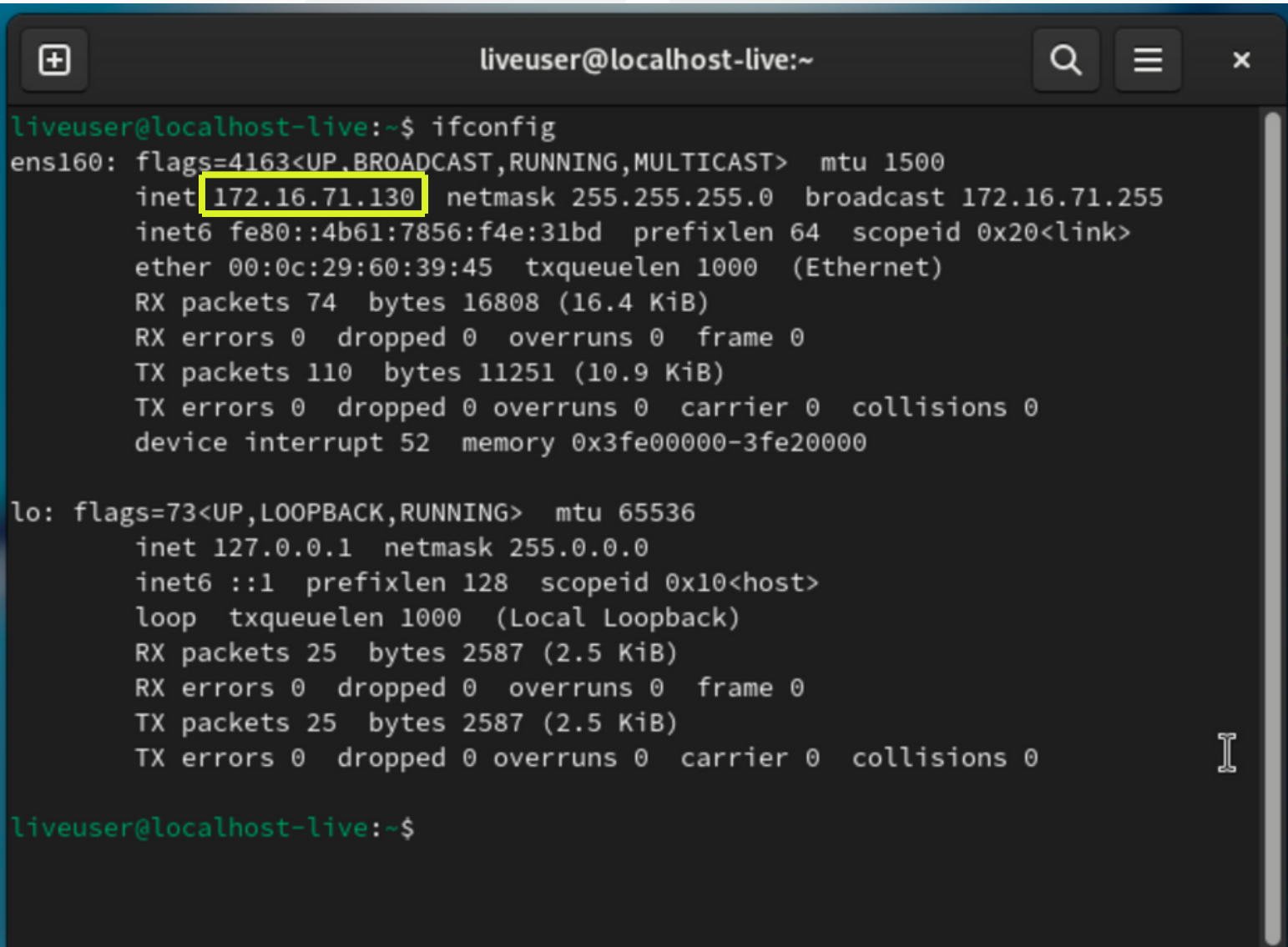
On remarque qu'il n'y a pas de messages indiquant une action bloquée par SELinux ou une politique spécifique de sécurité.

TEST

Accès via un navigateur web

Il faut ouvrir un navigateur et entrer l'adresse IP de notre serveur.

Trouvons d'abord notre adresse IP:



```
liveuser@localhost-live:~$ ifconfig
ens160: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.16.71.130 netmask 255.255.255.0 broadcast 172.16.71.255
    inet6 fe80::4b61:7856:f4e:31bd prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:60:39:45 txqueuelen 1000 (Ethernet)
    RX packets 74 bytes 16808 (16.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 110 bytes 11251 (10.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 52 memory 0x3fe00000-3fe20000

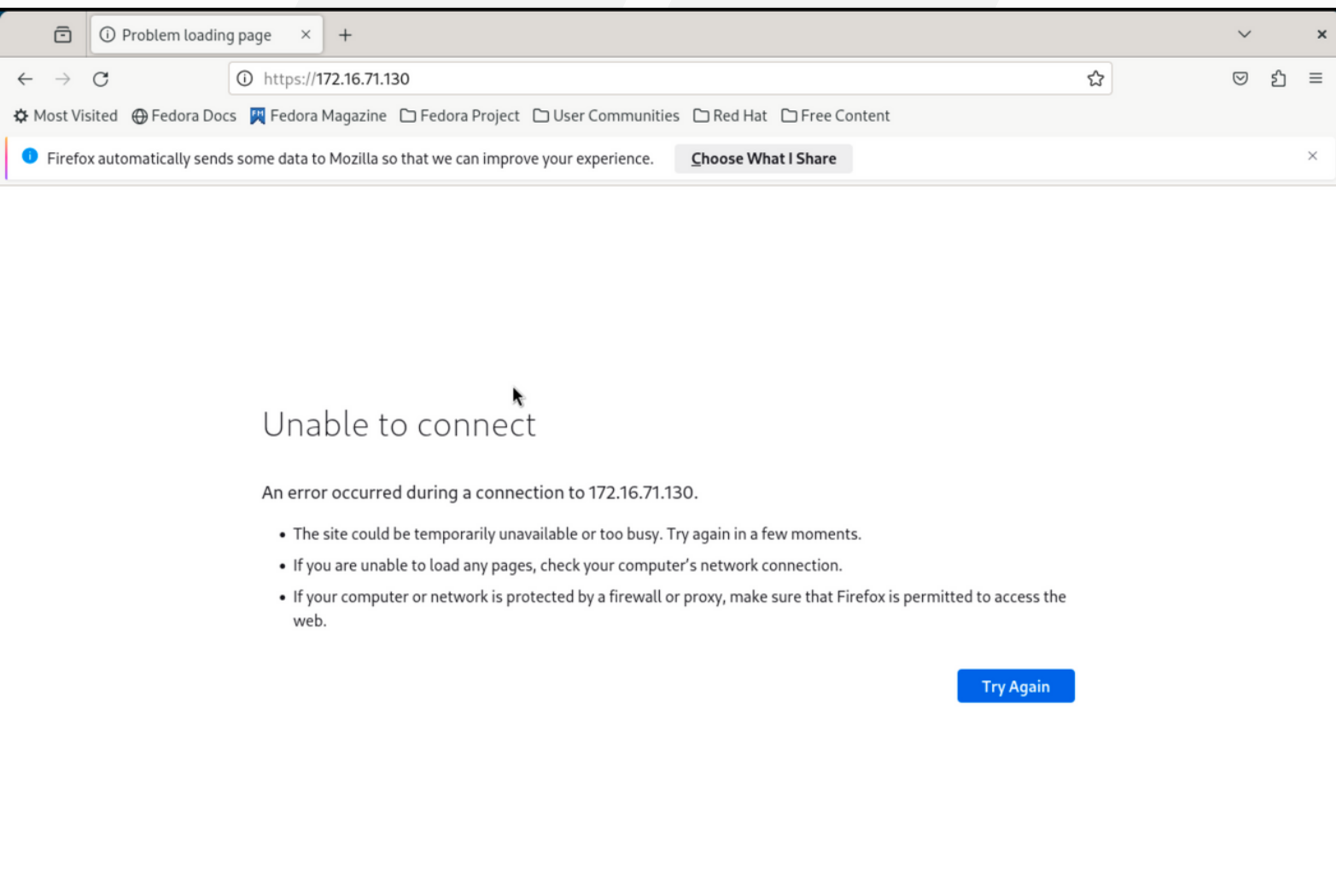
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 25 bytes 2587 (2.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 25 bytes 2587 (2.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

liveuser@localhost-live:~$
```

TEST

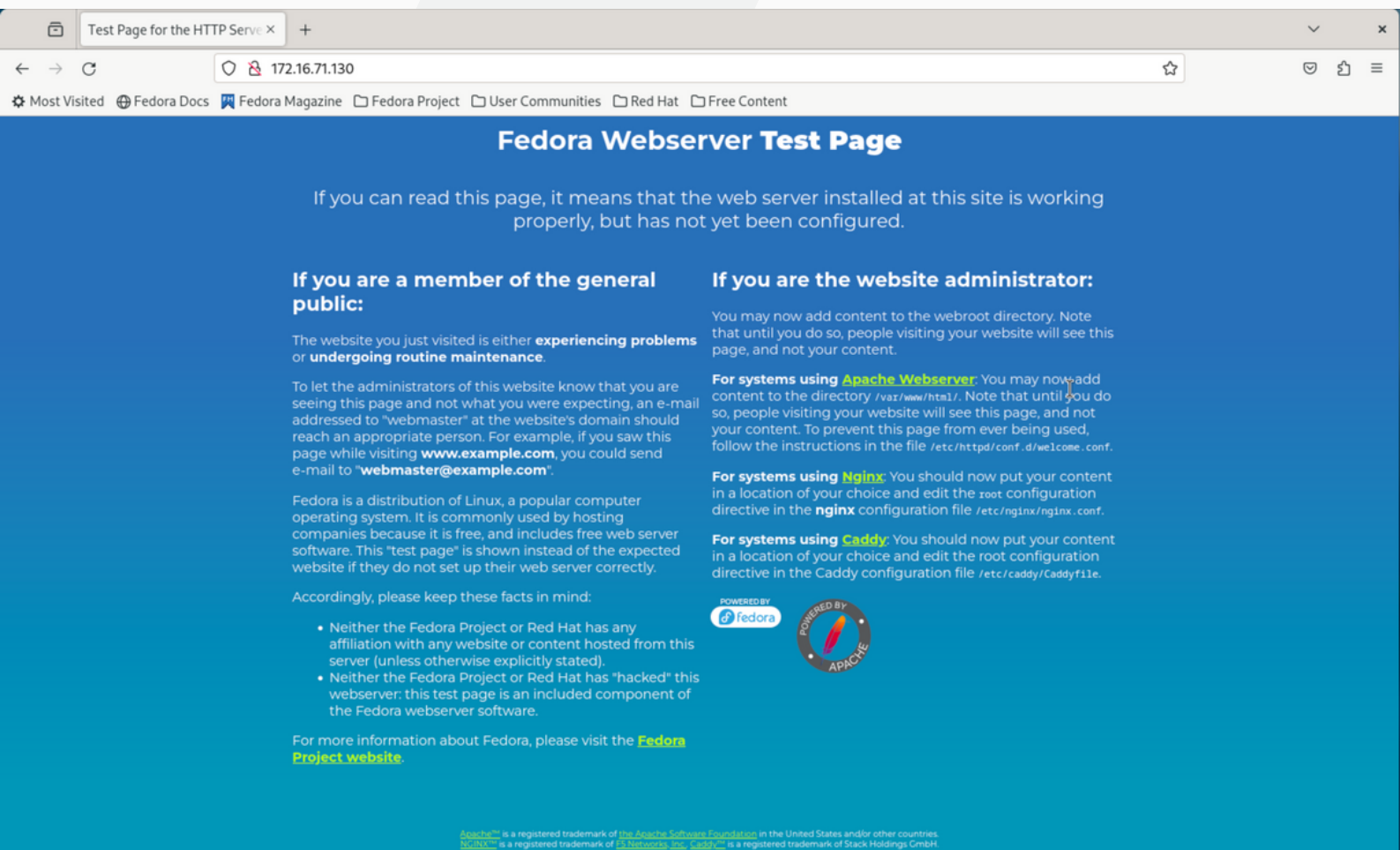
- Si Apache fonctionne correctement, nous devrions voir la page par défaut d'Apache.
- Si Apache ne fonctionne pas correctement, la page par défaut d'Apache ne s'affichera pas.

Avant l'intégration de SELinux dans Apache



TEST

Après l'intégration de SELinux dans Apache



On remarque que la page par défaut d'Apache s'est affichée, donc Apache fonctionne correctement.

CONCLUSION

En conclusion, l'intégration du module SELinux avec Apache représente une avancée significative en matière de sécurité pour les environnements web.

Ce projet offre une solution pratique pour étendre les capacités de sécurité de SELinux à Apache, renforçant ainsi la résilience du serveur face aux menaces en ligne.

En garantissant une compatibilité plus étroite entre ces deux composants, ce projet contribue à créer un environnement web plus sûr et plus robuste, offrant une tranquillité d'esprit aux administrateurs système et aux utilisateurs finaux.