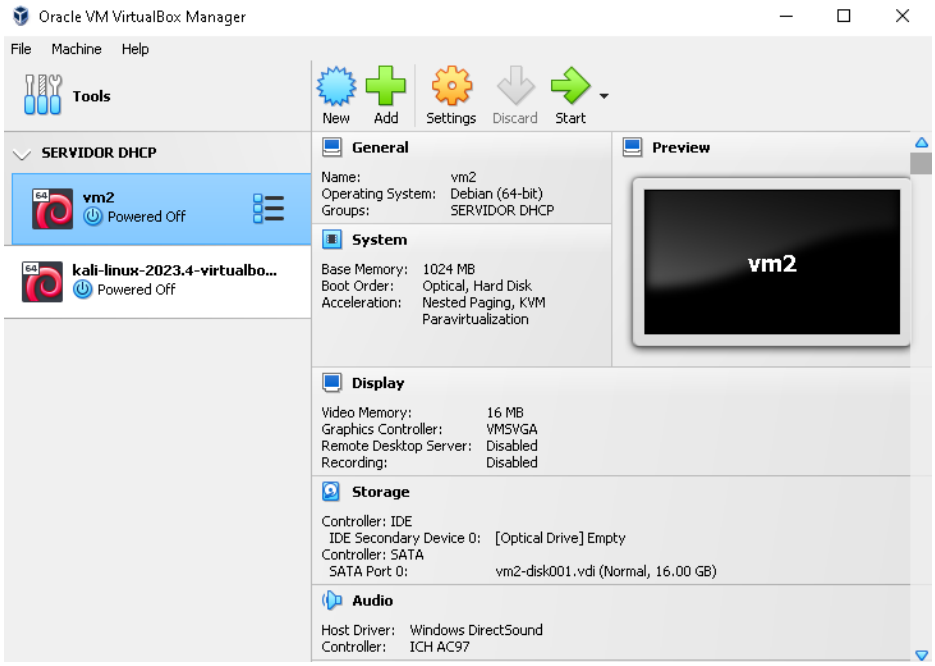
How to

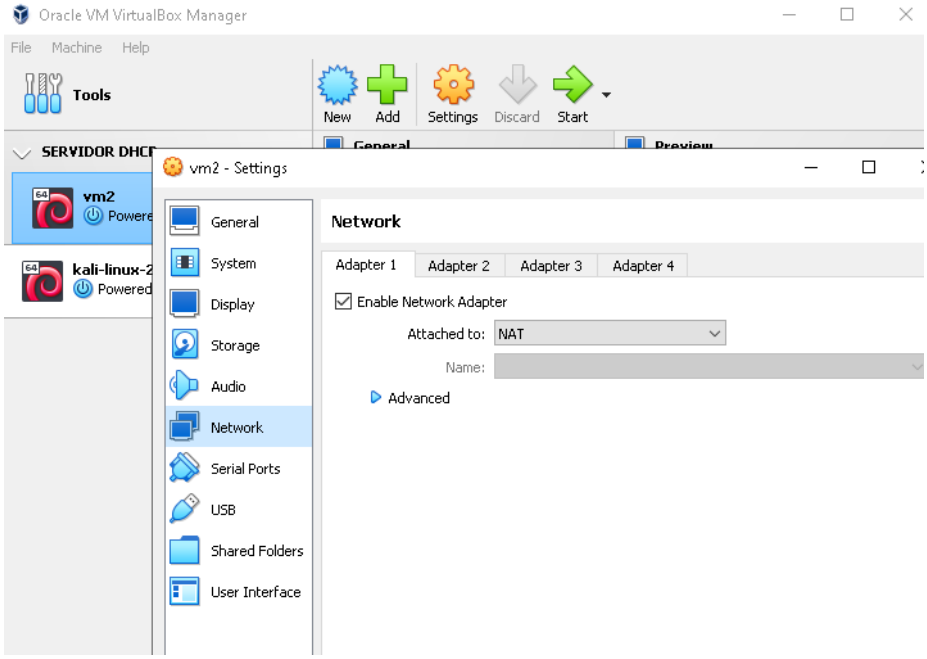
- **VM Debian – Server – IP 172.16.80.10**
- **VM Kali – Client – 172.16.80.20**

1. Configurar uma comunicação entre Server x Cliente

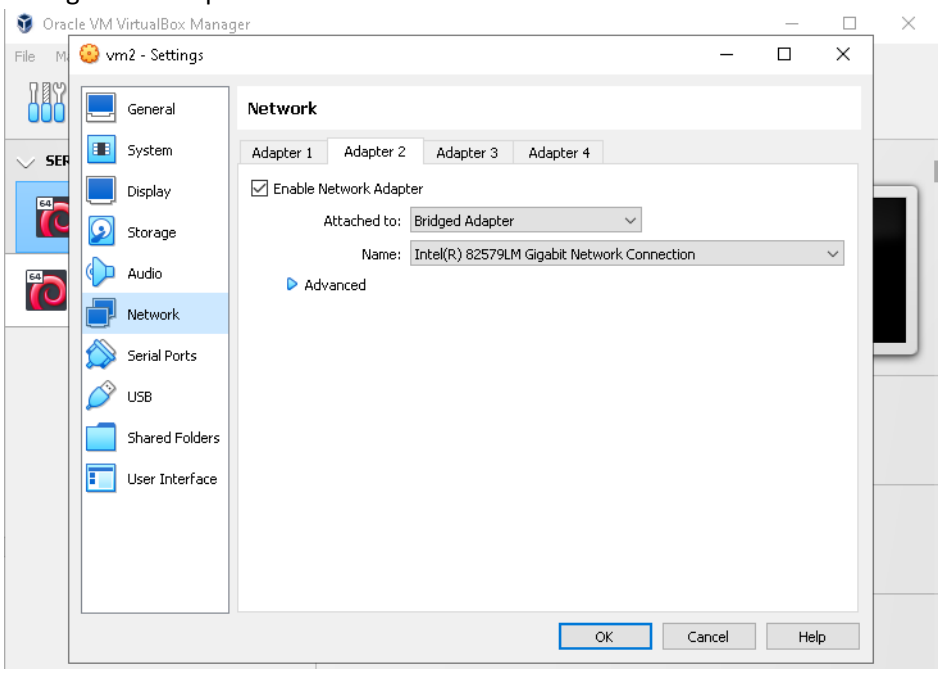
a. Configure as duas VMs para funcionar



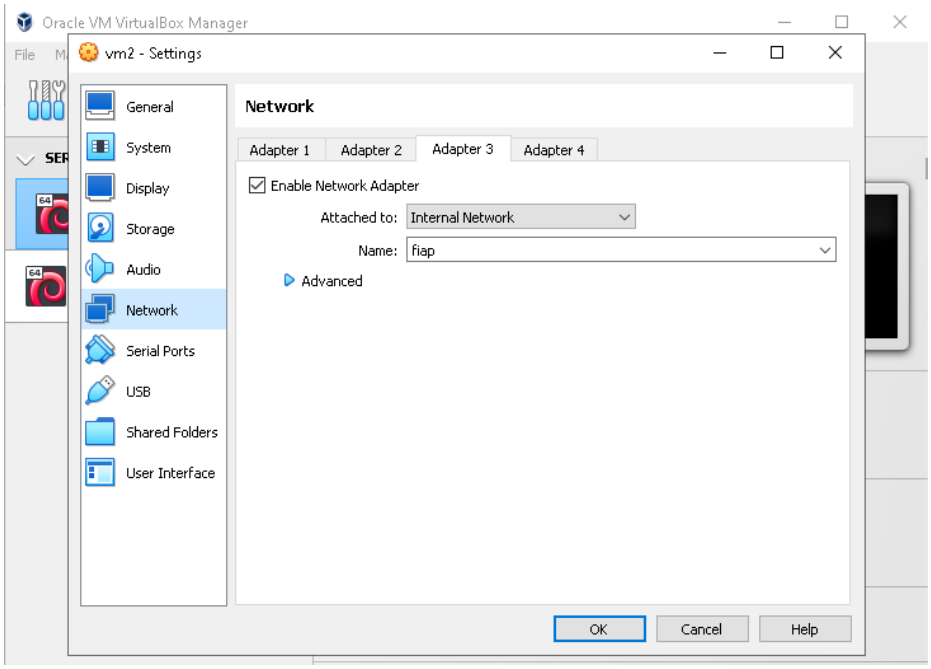
b. Clique em Settings e vá para a aba de Network



c. Configurar o Adapter 2



d. Configurar o Adapter 3



e. Repetir a configuração para a outra máquina virtual
f. Iniciar as duas máquinas, logando com o usuário root
    i. VM Debian - User root – Senha fiap
    ii. VM Kali – User kali – Senha kali
g. Na máquina Kali, rode o comando ‘sudo su’, digitando a senha ‘kali’ quando necessário.
h. Rodar o seguinte comando em cada uma das máquinas: ‘nano /etc/network/interfaces’

i. Na máquina Kali, vá até o final do arquivo e digite o seguinte:

```
# Placa de rede interna – fiap

allow-hotplug eth2

iface eth2 inet static

        address 172.16.80.20
```

j. Ainda no Kali, aperte as teclas ‘Ctrl O’ -> ‘Enter’ -> ‘Ctrl X’

k. Na máquina Debian, vá até o final do arquivo e digite o seguinte:

```
# Placa de rede interna – fiap

allow-hotplug enp0s9

iface enp0s9 inet static

        address 172.16.80.10
```

l. Digite o comando ‘init 6’ em ambas máquinas para reiniciar

m. Digite os comandos ‘ip -br -c a’ e verifique se as interfaces eth2 e enp0s9 estão com os IPs 172.16.80.20 e 172.16.80.10 respectivamente.

n. Na máquina Kali rode o comando ‘ping 172.16.80.10’ e verifique se há resposta do servidor.

```
┌──(root㉿kali)-[/home/kali]
└─# ping 172.16.80.10
PING 172.16.80.10 (172.16.80.10) 56(84) bytes of data.
64 bytes from 172.16.80.10: icmp_seq=1 ttl=64 time=1.50 ms
64 bytes from 172.16.80.10: icmp_seq=2 ttl=64 time=1.62 ms
64 bytes from 172.16.80.10: icmp_seq=3 ttl=64 time=1.41 ms
64 bytes from 172.16.80.10: icmp_seq=4 ttl=64 time=1.67 ms
64 bytes from 172.16.80.10: icmp_seq=5 ttl=64 time=1.54 ms
64 bytes from 172.16.80.10: icmp_seq=6 ttl=64 time=1.47 ms
^C
--- 172.16.80.10 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5010ms
rtt min/avg/max/mdev = 1.410/1.533/1.669/0.087 ms

┌──(root㉿kali)-[/home/kali]
└─#
```

(Aperte ‘Ctrl c’ para sair do comando)

o. Na máquina Debian rode o comando ‘ping 172.16.80.20’ e verifique se há resposta do cliente.

```
root@debian:~# ping 172.16.80.20
PING 172.16.80.20 (172.16.80.20) 56(84) bytes of data.
64 bytes from 172.16.80.20: icmp_seq=1 ttl=64 time=1.09 ms
64 bytes from 172.16.80.20: icmp_seq=2 ttl=64 time=0.906 ms
64 bytes from 172.16.80.20: icmp_seq=3 ttl=64 time=0.947 ms
64 bytes from 172.16.80.20: icmp_seq=4 ttl=64 time=0.918 ms
^[S64 bytes from 172.16.80.20: icmp_seq=5 ttl=64 time=1.04 ms
64 bytes from 172.16.80.20: icmp_seq=6 ttl=64 time=0.889 ms
^[S64 bytes from 172.16.80.20: icmp_seq=7 ttl=64 time=0.887 ms
64 bytes from 172.16.80.20: icmp_seq=8 ttl=64 time=0.913 ms
64 bytes from 172.16.80.20: icmp_seq=9 ttl=64 time=0.946 ms
^[S64 bytes from 172.16.80.20: icmp_seq=10 ttl=64 time=1.06 ms
^C
--- 172.16.80.20 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9008ms
rtt min/avg/max/mdev = 0.887/0.959/1.092/0.071 ms
root@debian:~#
```

(Aperte ‘Ctrl c’ para sair do comando)

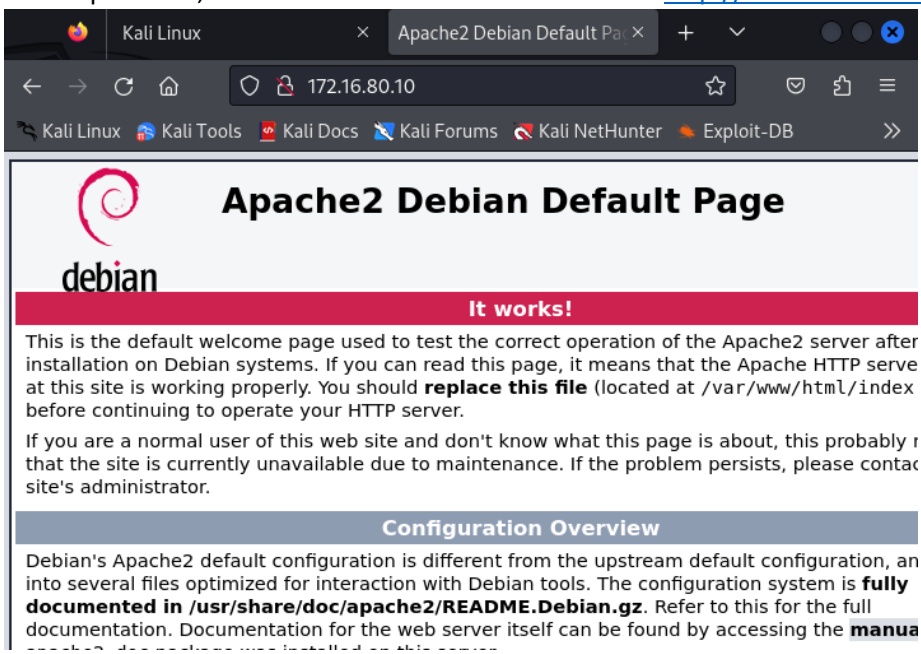
p. Isso nos mostra que as duas máquinas conseguem se comunicar

2. Simular vulnerabilidade de acesso ao Apache

a. Primeiro precisamos instalar o serviço apache2 no nosso servidor.
b. Na máquina Debian rode o comando ‘apt update’ e em seguida ‘apt install apache2’ (digite ‘S’ se solicitado)
c. Instale o pacote ‘net-tools’ para facilitar o uso do serviço apache2. Para isso rode o comando ‘apt install net-tools’
d. Para iniciar o serviço apache2 rode o comando ‘service apache2 start’
e. Para verificar se o serviço está funcionando, rode o comando ‘netstat -nltp’, a fim de verificar se a porta 80 está sendo usada pelo apache.

```
root@debian:~# netstat -nltp
Conexões Internet Ativas (sem os servidores)
Proto Recv-Q Send-Q Endereço Local          Endereço Remoto         Estado      PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               OUÇA       466/sshd: /usr/sbin
tcp6       0      0 :::80                   :::*                    OUÇA       467/apache2
tcp6       0      0 :::22                   :::*                    OUÇA       466/sshd: /usr/sbin
root@debian:~#
```

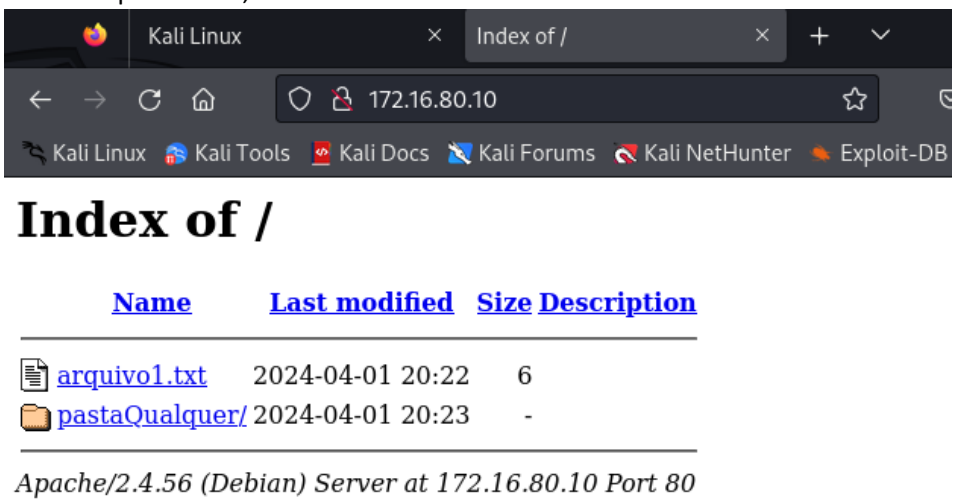
f. Na máquina Kali, abra o Mozilla Firefox e acesse a URL: http://172.16.80.10:80



g. Agora para visualizar a vulnerabilidade, volte na máquina Debian e rode os seguintes comandos:

```
service apache2 stop

mv /var/www/html/index.html /var/www/

touch /var/www/html/arquivo1.txt

echo “Teste” > /var/www/html/arquivo1.txt

mkdir /var/www/html/pastaQualquer

service apache2 start
```

h. Perceba que a tela exibe informações sensíveis, como os arquivos dentro do servidor, o sistema operacional, o servidor e sua versão etc.



i. Isso se trata da vulnerabilidade em questão, pois abre portas para pessoas maliciosas explorarem falhas na versão do Servidor, no SO e entre outras falhas de segurança.

3. Corrigir a vulnerabilidade

a. Para corrigir a vulnerabilidade, precisamos alterar algumas configurações do apache2.
b. Primeiro vamos configurar o servidor para o modo produção, de forma que esconda o SO e a versão do servidor.
c. Na máquina Debian rode o comando ‘nano /etc/apache2/conf-enabled/security.conf’

d. Onde estiver escrito ‘ServerTokens OS’, troque o ‘OS’ para ‘Prod’

```
  GNU nano 5.4                                security.conf
#
# Disable access to the entire file system except for the directories that
# are explicitly allowed later.
#
# This currently breaks the configurations that come with some web application
# Debian packages.
#
#<Directory />
#   AllowOverride None
#   Require all denied
#</Directory>


# Changing the following options will not really affect the security of the
# server, but might make attacks slightly more difficult in some cases.

#
# ServerTokens
# This directive configures what you return as the Server HTTP response
# Header. The default is 'Full' which sends information about the OS-Type
# and compiled in modules.
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
ServerTokens Prod
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
                                    [ Escritas 73 linhas ]
^G Ajuda      ^O Gravar     ^W Onde está? ^K Recortar   ^T Executar   ^C Local      M-U Desfazer
^X Sair       ^R Ler o arq  ^\ Substituir ^U Colar      ^J Justificar ^_ Ir p/ linha M-E Refazer
```

e. Aperte ‘Ctrl O’ -> Enter -> ‘Ctrl X’

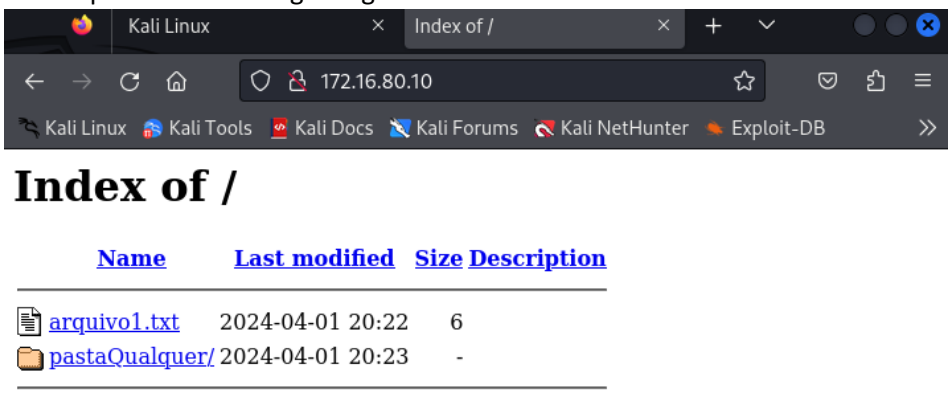
f. Digite o comando novamente (nano /etc/apache2/conf-enabled/security.conf)

g. Altere onde estiver escrito ‘ServerSignature On’ para ‘ServerSignature Off’

```
  GNU nano 5.4                                security.conf
# and compiled in modules.
# Set to one of:  Full | OS | Minimal | Minor | Major | Prod
# where Full conveys the most information, and Prod the least.
#ServerTokens Minimal
ServerTokens Prod
#ServerTokens Full

#
# Optionally add a line containing the server version and virtual host
# name to server-generated pages (internal error documents, FTP directory
# listings, mod_status and mod_info output etc., but not CGI generated
# documents or custom error documents).
# Set to "EMail" to also include a mailto: link to the ServerAdmin.
# Set to one of:  On | Off | EMail
#ServerSignature Off
ServerSignature Off

#
# Allow TRACE method
#
# Set to "extended" to also reflect the request body (only for testing and
# diagnostic purposes).
#
# Set to one of:  On | Off | extended
TraceEnable Off
#TraceEnable On

#
# Forbid access to version control directories
#
# If you use version control systems in your document root, you should
# probably deny access to their directories. For example, for subversion:
#
                                    [ Escritas 73 linhas ]
^G Ajuda      ^O Gravar     ^W Onde está? ^K Recortar   ^T Executar   ^C Local      M-U Desfazer
^X Sair       ^R Ler o arq  ^\ Substituir ^U Colar      ^J Justificar ^_ Ir p/ linha M-E Refazer
```

h. Reinicie o serviço com o comando ‘service apache2 stop’ -> ‘service apache2 start’

i. Na máquina Kali recarregue a guia do Mozilla

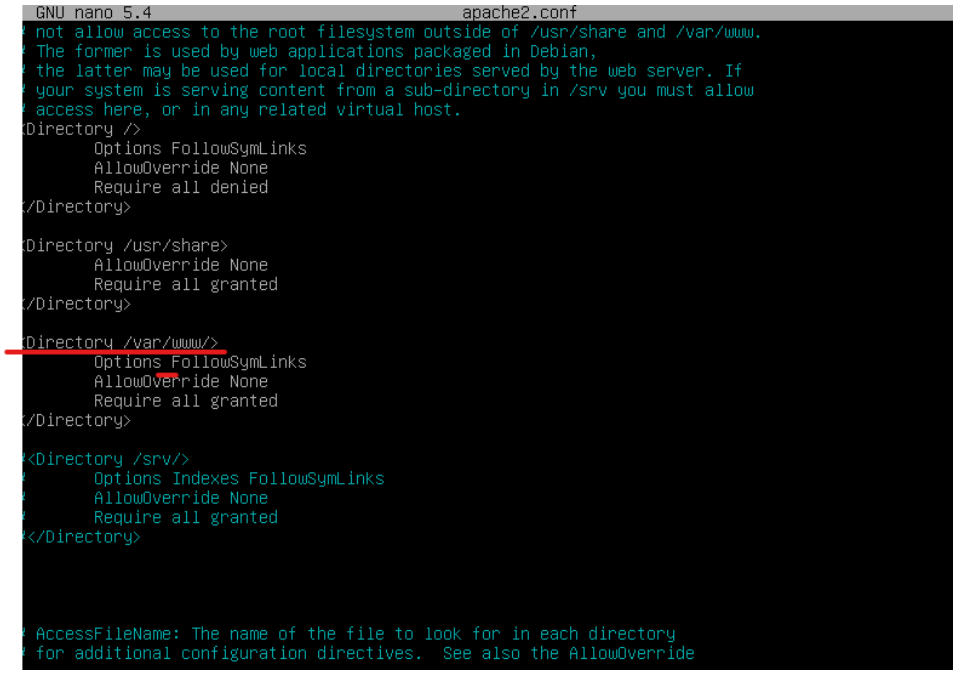


j. Perceba que o banner exibindo informações do servidor e do sistema operacional sumiram.

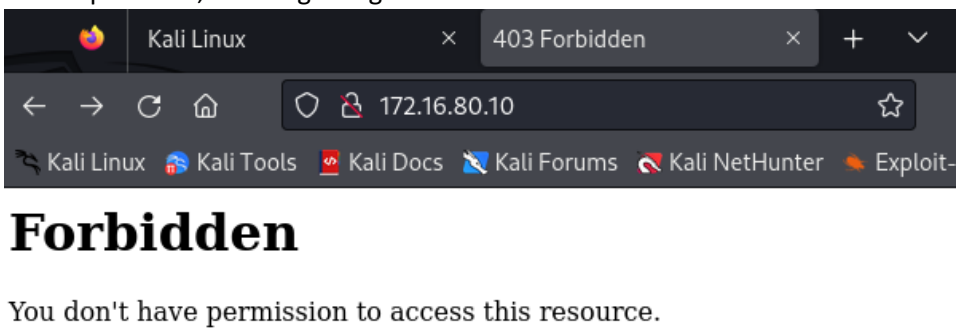
k. Agora para impedir o acesso à tela com as pastas e arquivos do servidor, precisamos realizar uma outra configuração.

l. Rode o comando ‘nano /etc/apache2/apache2.conf’

m. Procure pelo texto ‘<Directory /var/www/>’, e na linha abaixo, remova a palavra ‘Indexes’



n. Na máquina Kali, recarregue a guia



o. Perceba que agora já não conseguimos mais ver os arquivos do servidor
p. Agora podemos adicionar um index.html novamente para visualizar a tela inicial do servidor.
q. Para isso rode o comando ‘mv /var/www/index.html /var/www/html/’
r. Reinicie o servidor com os comandos ‘service apache2 stop’ e ‘service apache2 start’
s. Na máquina Kali recarregue a guia e perceba que a tela exibe o que aparecia antes: