Etudiants:

HERMANN BANKOLE

EDWINE FAYE CACOU

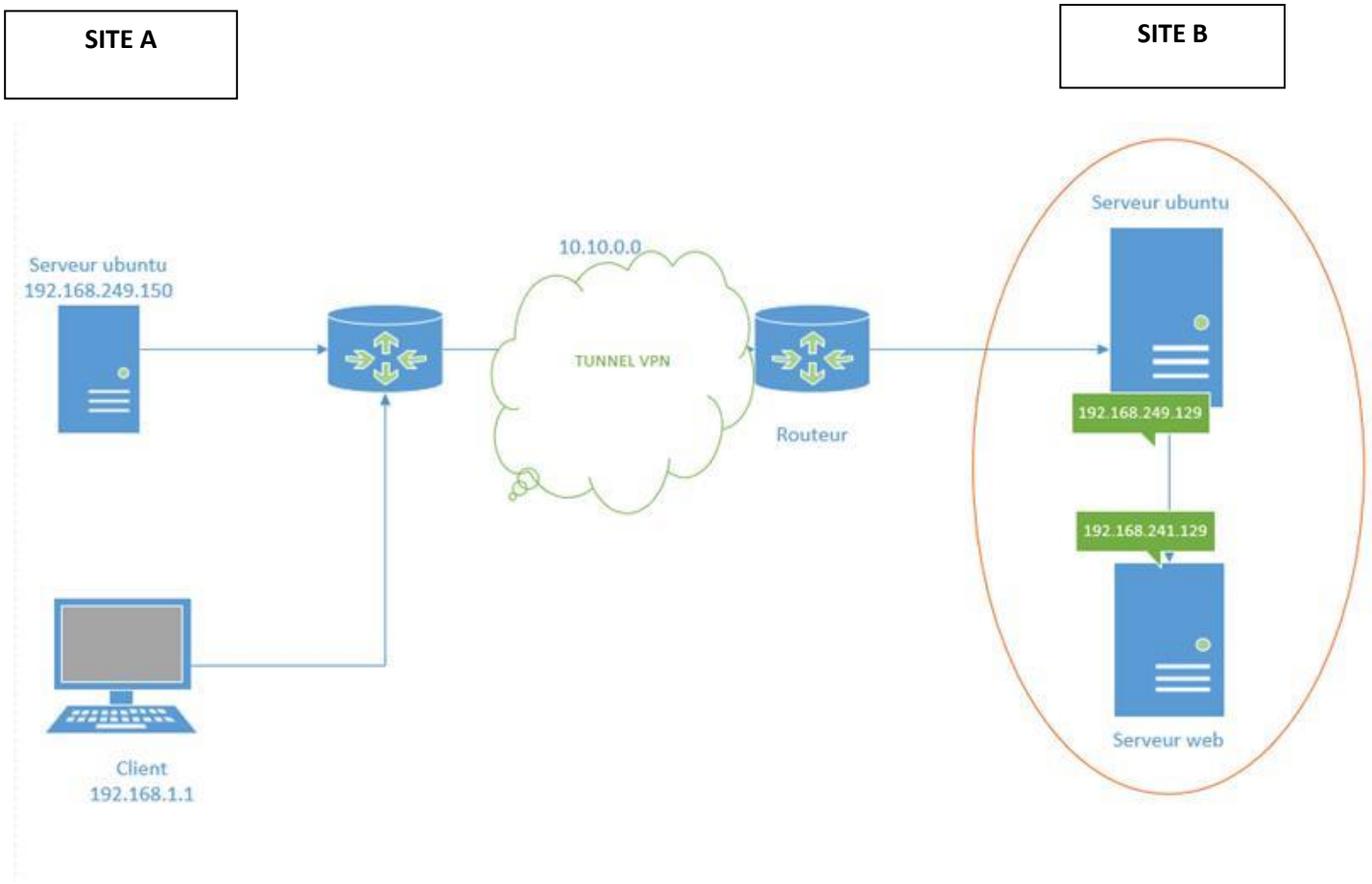
Animateur:

ALEX FALZON

SECURITE INFORMATIQUE

MISE EN PLACE D'UN VPN AVEC LES SOLUTIONS OPEN VPN ET WIREGUARD ET DETECTION DE FAILLES SUR LES SITES WEB AVEC JOOMSCAN

MISE EN PLACE D'UN VPN LAN TO LAN VIA OPENVPN



SCRIPT MISE EN PLACE DU SERVEUR OPENVPN

```
#!/bin/bash
#
#
#
# Creation serveur VPN
#


echo -e "---------------------------------"
echo -e "---------------------------------"


# Vérifie si le dossier easy-rsa existe
if [ -d "/etc/openvpn/easy-rsa»]; then
        echo "Le dossier1 existe !";
else
        echo "Le dossier n'existe pas, copie de /usr/share/easy-rsa/"
        cp -r /usr/share/easy-rsa/ /etc/openvpn/
fi


chown -R root:root  /etc/openvpn/easy-rsa/
cd /etc/openvpn/easy-rsa
cp openssl-1.0.0.cnf openssl.cnf


echo ""
echo "modif vars"
echo ""
```

 # Modification du fichiers VAR

sed -i 's/export KEY_COUNTRY=\"US\"/export KEY_COUNTRY=\"FR\"/' vars

sed -i 's/export KEY_PROVINCE=\"CA\"/export KEY_PROVINCE=\"IDF\"/' vars

sed -i 's/export KEY_CITY=\"SanFrancisco\"/export KEY_CITY=\"Paris\"/' vars

sed -i 's/export KEY_ORG=\"Fort-Funston\"/export KEY_ORG=\"Insta\"/' vars

sed -i 's/export KEY_EMAIL=\"me@myhost.mydomain\"/export KEY_EMAIL=\"Ihbankole@inta.fr\"/' vars

sed -i 's/export KEY_OU=\"MyOrganization\"/export KEY_OU=\"IT\"/' vars

echo "----"

echo ""

echo "Cette opération peut prendre un certain temps."

echo ""

echo "----"

 # Génération des clés serveurs

source vars

./clean-all

./build-dh

./pkitool --initca

./pkitool --server server

openvpn --genkey --secret keys/ta.key

cp keys/ta.key /etc/openvpn/

 # Copie des cles et certificats dans /etc/openvpn/

echo ""

echo "Copie des cles serveurs"

echo ""

cp keys/ca.crt keys/ta.key keys/server.crt keys/server.key keys/dh2048.pem /etc/openvpn/

mkdir -p /etc/openvpn/jail/tmp

mkdir /etc/openvpn/clientconf

 # Création du fichier de configuration du client

cat >> /etc/openvpn/server.conf << EOF

Voir Fichier de configuration

EOF

 # règles obligatoires pour ouvrir déverrouiller l'accès :

echo "Autorisation de l'interface TUN0"

iptables -I FORWARD -i tun0 -j ACCEPT

iptables -I FORWARD -o tun0 -j ACCEPT

iptables -I OUTPUT -o tun0 -j ACCEPT

sed -i 's/#net.ipv4.ip_forward=1/net.ipv4.ip_forward=1/' /etc/sysctl.conf

```
echo ""

echo ""

echo "----"

echo "Installation terminée."

echo "----"

echo ""

echo ""



echo ""

echo ""

echo "Le serveur va redémarrer dans 5....."

echo "Le serveur va redémarrer dans 4...."

echo "Le serveur va redémarrer dans 3..."

echo "Le serveur va redémarrer dans 2.."

echo "Le serveur va redémarrer dans 1."

init 6
```

apt install openvpn

sudo apt-get install easy-rsa

cp openssl-easyrsa.cnf openssl.cnf

./easyrsa init-pk

./easyrsa build-ca

./easyrsa gen-req serverbanks nopass

./easyrsa sign-req server serverbanks

openssl dhparam -out dh2048.pem 2048

./easyrsa gen-req openvpn nopass

./easyrsa sign-req client openvpn

Fichier de Configuration (Serveur)

```
#################################################
# Sample OpenVPN 2.0 config file for            #
# multi-client server.                          #
#                                               #
# This file is for the server side              #
# of a many-clients <-> one-server              #
# OpenVPN configuration.                        #
#                                               #
# OpenVPN also supports                         #
# single-machine <-> single-machine             #
# configurations (See the Examples page         #
# on the web site for more info).               #
```

```
#                                     #
# This config should work on Windows          #
# or Linux/BSD systems.  Remember on          #
# Windows to quote pathnames and use          #
# double backslashes, e.g.:                   #
# "C:\\Program Files\\OpenVPN\\config\\foo.key" #
#                                     #
# Comments are preceded with '#' or ';'       #
#################################################

# Which local IP address should OpenVPN
# listen on? (optional)
;local a.b.c.d

# Which TCP/UDP port should OpenVPN listen on?
# If you want to run multiple OpenVPN instances
# on the same machine, use a different port
# number for each one.  You will need to
# open up this port on your firewall.
port 1194

# TCP or UDP server?
proto tcp
;proto udp

# "dev tun" will create a routed IP tunnel,
# "dev tap" will create an ethernet tunnel.
# Use "dev tap0" if you are ethernet bridging
```

```
# and have precreated a tap0 virtual interface
# and bridged it with your ethernet interface.
# If you want to control access policies
# over the VPN, you must create firewall
# rules for the the TUN/TAP interface.
# On non-Windows systems, you can give
# an explicit unit number, such as tun0.
# On Windows, use "dev-node" for this.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
# from the Network Connections panel if you
# have more than one.  On XP SP2 or higher,
# you may need to selectively disable the
# Windows firewall for the TAP adapter.
# Non-Windows systems usually don't need this.
;dev-node MyTap

# SSL/TLS root certificate (ca), certificate
# (cert), and private key (key).  Each client
# and the server must have their own cert and
# key file.  The server and all clients will
# use the same ca file.
#
```

```
# See the "easy-rsa" directory for a series
# of scripts for generating RSA certificates
# and private keys.  Remember to use
# a unique Common Name for the server
# and each of the client certificates.
#
# Any X509 key management system can be used.
# OpenVPN can also use a PKCS #12 formatted key file
# (see "pkcs12" directive in man page).
ca ./server/ca.crt
cert ./server/serverbanks.crt
key ./server/serverbanks.key  # This file should be kept secret

# Diffie hellman parameters.
# Generate your own with:
#   openssl dhparam -out dh2048.pem 2048
dh ./server/dh.pem

# Network topology
# Should be subnet (addressing via IP)
# unless Windows clients v2.0.9 and lower have to
# be supported (then net30, i.e. a /30 per client)
# Defaults to net30 (not recommended)
;topology subnet

# Configure server mode and supply a VPN subnet
# for OpenVPN to draw client addresses from.
# The server will take 10.8.0.1 for itself,
```

the rest will be made available to clients.

Each client will be able to reach the server

on 10.8.0.1. Comment this line out if you are

ethernet bridging. See the man page for more info.

server 10.10.0.0 255.255.255.0

Maintain a record of client <-> virtual IP address

associations in this file. If OpenVPN goes down or

is restarted, reconnecting clients can be assigned

the same virtual IP address from the pool that was

previously assigned.

ifconfig-pool-persist /var/log/openvpn/ipp.txt

Configure server mode for ethernet bridging.

You must first use your OS's bridging capability

to bridge the TAP interface with the ethernet

NIC interface. Then you must manually set the

IP/netmask on the bridge interface, here we

assume 10.8.0.4/255.255.255.0. Finally we

must set aside an IP range in this subnet

(start=10.8.0.50 end=10.8.0.100) to allocate

to connecting clients. Leave this line commented

out unless you are ethernet bridging.

;server-bridge 10.8.0.4 255.255.255.0 10.8.0.50 10.8.0.100

Configure server mode for ethernet bridging

using a DHCP-proxy, where clients talk

to the OpenVPN server-side DHCP server

```
# to receive their IP address allocation
# and DNS server addresses.  You must first use
# your OS's bridging capability to bridge the TAP
# interface with the ethernet NIC interface.
# Note: this mode only works on clients (such as
# Windows), where the client-side TAP adapter is
# bound to a DHCP client.
;server-bridge

# Push routes to the client to allow it
# to reach other private subnets behind
# the server.  Remember that these
# private subnets will also need
# to know to route the OpenVPN client
# address pool (10.8.0.0/255.255.255.0)
# back to the OpenVPN server.
push "route 192.168.241.0 255.255.255.0"
;push "route 192.168.20.0 255.255.255.0"

# To assign specific IP addresses to specific
# clients or if a connecting client has a private
# subnet behind it that should also have VPN access,
# use the subdirectory "ccd" for client-specific
# configuration files (see man page for more info).

# EXAMPLE: Suppose the client
# having the certificate common name "Thelonious"
# also has a small subnet behind his connecting
```

machine, such as 192.168.40.128/255.255.255.248.

First, uncomment out these lines:

client-config-dir ccd

route 192.168.1.0 255.255.255.0

Then create a file ccd/Thelonious with this line:

iroute 192.168.40.128 255.255.255.248

This will allow Thelonious' private subnet to

access the VPN. This example will only work

if you are routing, not bridging, i.e. you are

using "dev tun" and "server" directives.

EXAMPLE: Suppose you want to give

Thelonious a fixed VPN IP address of 10.9.0.1.

First uncomment out these lines:

;client-config-dir ccd

;route 10.9.0.0 255.255.255.252

Then add this line to ccd/Thelonious:

ifconfig-push 10.9.0.1 10.9.0.2

Suppose that you want to enable different

firewall access policies for different groups

of clients. There are two methods:

(1) Run multiple OpenVPN daemons, one for each

group, and firewall the TUN/TAP interface

for each group/daemon appropriately.

(2) (Advanced) Create a script to dynamically

modify the firewall in response to access

from different clients. See man

```
#     page for more info on learn-address script.
;learn-address ./script

# If enabled, this directive will configure
# all clients to redirect their default
# network gateway through the VPN, causing
# all IP traffic such as web browsing and
# and DNS lookups to go through the VPN
# (The OpenVPN server machine may need to NAT
# or bridge the TUN/TAP interface to the internet
# in order for this to work properly).
;push "redirect-gateway def1 bypass-dhcp"

# Certain Windows-specific network settings
# can be pushed to clients, such as DNS
# or WINS server addresses.  CAVEAT:
# http://openvpn.net/faq.html#dhcpcaveats
# The addresses below refer to the public
# DNS servers provided by opendns.com.
;push "dhcp-option DNS 8.8.8.8"
;push "dhcp-option DNS 1.1.1.1"

# Uncomment this directive to allow different
# clients to be able to "see" each other.
# By default, clients will only see the server.
# To force clients to only see the server, you
# will also need to appropriately firewall the
# server's TUN/TAP interface.
```

client-to-client

push "route 192.168.1.0 255.255.255.0"

Uncomment this directive if multiple clients

might connect with the same certificate/key

files or common names. This is recommended

only for testing purposes. For production use,

each client should have its own certificate/key

pair.

#

IF YOU HAVE NOT GENERATED INDIVIDUAL

CERTIFICATE/KEY PAIRS FOR EACH CLIENT,

EACH HAVING ITS OWN UNIQUE "COMMON NAME",

UNCOMMENT THIS LINE OUT.

;duplicate-cn

The keepalive directive causes ping-like

messages to be sent back and forth over

the link so that each side knows when

the other side has gone down.

Ping every 10 seconds, assume that remote

peer is down if no ping received during

a 120 second time period.

keepalive 10 120

For extra security beyond that provided

by SSL/TLS, create an "HMAC firewall"

to help block DoS attacks and UDP port flooding.

#

Generate with:

openvpn --genkey --secret ta.key

#

The server and each client must have

a copy of this key.

The second parameter should be '0'

on the server and '1' on the clients.

;tls-auth ta.key 0 # This file is secret

Select a cryptographic cipher.

This config item must be copied to

the client config file as well.

Note that v2.4 client/server will automatically

negotiate AES-256-GCM in TLS mode.

See also the ncp-cipher option in the manpage

;cipher AES-256-CBC

Enable compression on the VPN link and push the

option to the client (v2.4+ only, for earlier

versions see below)

;compress lz4-v2

;push "compress lz4-v2"

For compression compatible with older clients use comp-lzo

If you enable it here, you must also

enable it in the client config file.

comp-lzo

```
# The maximum number of concurrently connected
# clients we want to allow.
;max-clients 100

# It's a good idea to reduce the OpenVPN
# daemon's privileges after initialization.
#
# You can uncomment this out on
# non-Windows systems.
;user nobody
;group nogroup

# The persist options will try to avoid
# accessing certain resources on restart
# that may no longer be accessible because
# of the privilege downgrade.
persist-key
persist-tun

# Output a short status file showing
# current connections, truncated
# and rewritten every minute.
status /var/log/openvpn/openvpn-status.log

# By default, log messages will go to the syslog (or
# on Windows, if running as a service, they will go to
# the "\Program Files\OpenVPN\log" directory).
# Use log or log-append to override this default.
```

```
# "log" will truncate the log file on OpenVPN startup,
# while "log-append" will append to it.  Use one
# or the other (but not both).
;log         /var/log/openvpn/openvpn.log
;log-append  /var/log/openvpn/openvpn.log

# Set the appropriate level of log
# file verbosity.
#
# 0 is silent, except for fatal errors
# 4 is reasonable for general usage
# 5 and 6 can help to debug connection problems
# 9 is extremely verbose
verb 3

# Silence repeating messages.  At most 20
# sequential messages of the same message
# category will be output to the log.
;mute 20

# Notify the client that when the server restarts so it
# can automatically reconnect.
;explicit-exit-notify 1
management localhost 5500
```

Fichier de Configuration (Client)

```
##############################################
# Sample client-side OpenVPN 2.0 config file #
# for connecting to multi-client server.     #
#                                            #
# This configuration can be used by multiple #
# clients, however each client should have   #
# its own cert and key files.                #
#                                            #
# On Windows, you might want to rename this  #
# file so it has a .ovpn extension           #
##############################################

# Specify that we are a client and that we
# will be pulling certain config file directives
# from the server.
client

# Use the same setting as you are using on
# the server.
# On most systems, the VPN will not function
# unless you partially or fully disable
# the firewall for the TUN/TAP interface.
;dev tap
dev tun

# Windows needs the TAP-Win32 adapter name
```

from the Network Connections panel

if you have more than one. On XP SP2,

you may need to disable the firewall

for the TAP adapter.

;dev-node MyTap

Are we connecting to a TCP or

UDP server? Use the same setting as

on the server.

proto tcp

;proto udp

The hostname/IP and port of the server.

You can have multiple remote entries

to load balance between the servers.

remote 192.168.249.129 1194

;remote my-server-2 1194

Choose a random host from the remote

list for load-balancing. Otherwise

try hosts in the order specified.

;remote-random

Keep trying indefinitely to resolve the

host name of the OpenVPN server. Very useful

on machines which are not permanently connected

to the internet such as laptops.

resolv-retry infinite

Most clients don't need to bind to

a specific local port number.

nobind

Downgrade privileges after initialization (non-Windows only)

;user nobody

;group nogroup

Try to preserve some state across restarts.

persist-key

persist-tun

If you are connecting through an

HTTP proxy to reach the actual OpenVPN

server, put the proxy server/IP and

port number here. See the man page

if your proxy server requires

authentication.

;http-proxy-retry # retry on connection failures

;http-proxy [proxy server] [proxy port #]

Wireless networks often produce a lot

of duplicate packets. Set this flag

to silence duplicate packet warnings.

;mute-replay-warnings

SSL/TLS parms.

See the server config file for more

description. It's best to use

a separate .crt/.key file pair

for each client. A single ca

file can be used for all clients.

ca ./client/ca.crt

cert ./client/openvpn.crt

key ./client/openvpn.key

Verify server certificate by checking that the

certicate has the correct key usage set.

This is an important precaution to protect against

a potential attack discussed here:

http://openvpn.net/howto.html#mitm

#

To use this feature, you will need to generate

your server certificates with the keyUsage set to

digitalSignature, keyEncipherment

and the extendedKeyUsage to

serverAuth

EasyRSA can do this for you.

;remote-cert-tls server

If a tls-auth key is used on the server

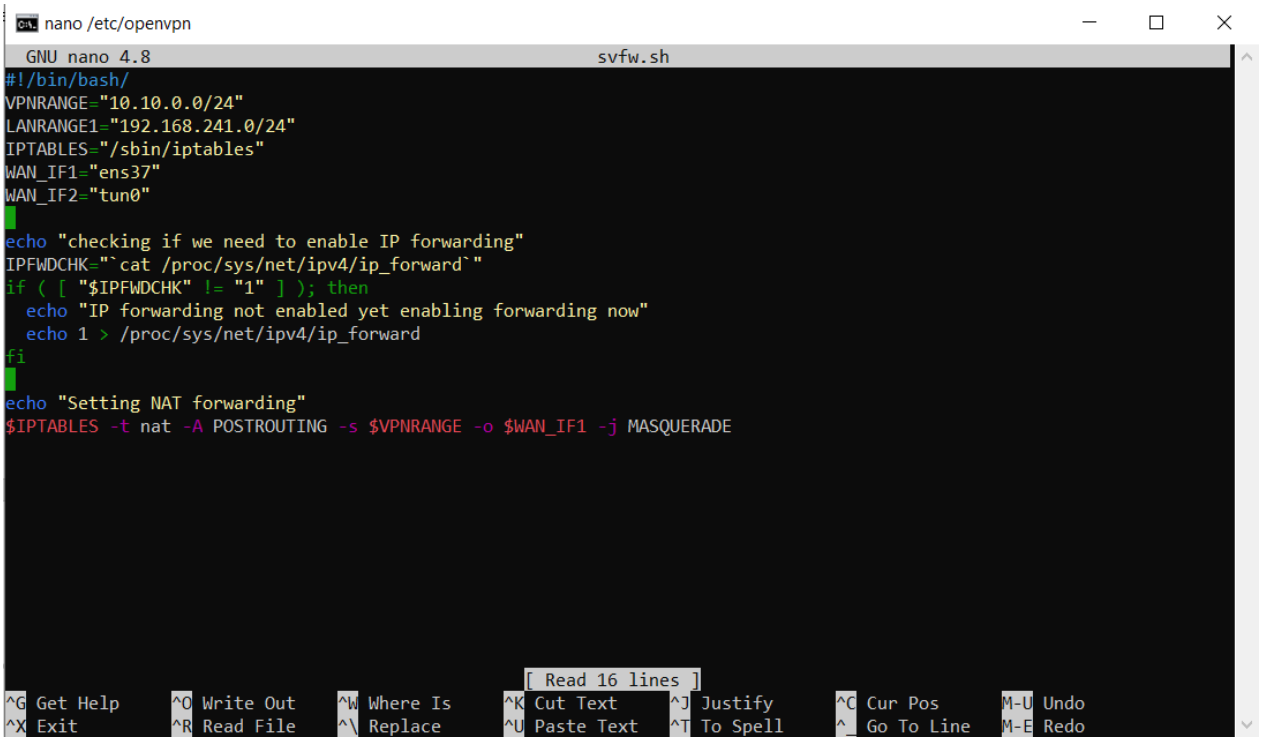
then every client must also have the key.

;tls-auth ta.key 1

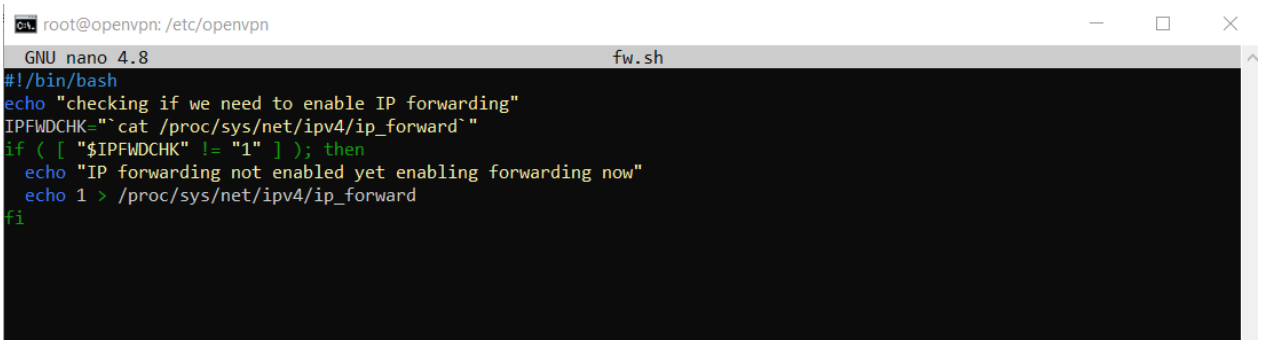
Select a cryptographic cipher.

```
# If the cipher option is used on the server
# then you must also specify it here.
# Note that v2.4 client/server will automatically
# negotiate AES-256-GCM in TLS mode.
# See also the ncp-cipher option in the manpage
;cipher AES-256-CBC

# Enable compression on the VPN link.
# Don't enable this unless it is also
# enabled in the server config file.
comp-lzo

# Set log file verbosity.
verb 3

# Silence repeating messages
;mute 20
```

SCRIPT POUR ROUTAGE (SERVER)



SCRIPT POUR ROUTAGE (CLIENT)



Routage sur serveur site B

route add -net 10.10.0.0 netmask 255.255.255.0 gw 192.168.249.150 ens37

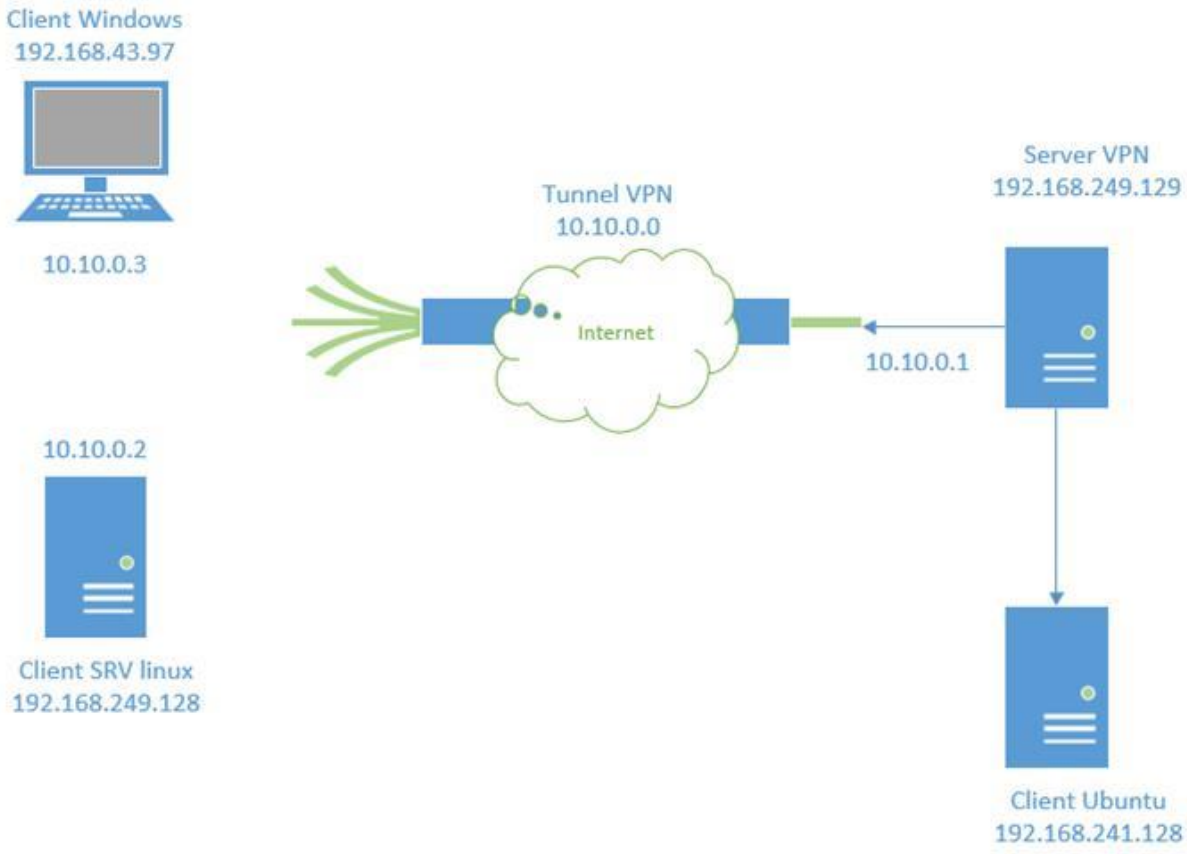
route add -net 192.168.241.0 netmask 255.255.255.0 gw 192.168.249.150 ens37

Routage sur serveur site A

route add -net 10.10.0.0 netmask 255.255.255.0 gw 192.168.249.129 ens37

route add -net 192.168.241.0 netmask 255.255.255.0 gw 192.168.249.129 ens37

MISE EN PLACE D'UN VPN Client TO LAN VIA WIREGUARD



Lors de l'installation d'un système de gestion de contenu pour votre site Web, il est facile d'être paresseux et de supposer qu'il fera tout le travail à votre place. Un CMS comme Joomla rend certainement les choses plus pratiques et vous permet de publier un site Web soigné très rapidement, mais cela ne signifie pas que vous ne devriez pas prendre plus de temps pour le sécuriser.

Si votre site Web exécute Joomla, vous pouvez utiliser l'utilitaire JoomScan contre votre site pour découvrir des vulnérabilités ou simplement des informations générales qui peuvent aider à une attaque contre votre site. Une fois que vous êtes conscient des points faibles du site, vous pouvez prendre les mesures appropriées pour le sécuriser. JoomScan fonctionne de manière similaire à WPScan, qui est utilisé pour analyser les sites WordPress à la recherche de vulnérabilités.

Dans ce guide, nous verrons comment utiliser JoomScan sur Kali Linux. JoomScan en lui-même n'est pas un outil qui peut être utilisé à des fins malveillantes lors de simples analyses sur un site, à moins que vous ne considériez le trafic supplémentaire lui-même comme malveillant. Mais les informations qu'il révèle sur un site peuvent être exploitées par des attaquants pour lancer une attaque. Par conséquent, assurez-vous que vous êtes autorisé à analyser un site Web lorsque vous utilisez cet outil.

COMMENT UTILISER JOOMSCAN

Vous pouvez installer JoomScan sur votre système (ou le mettre à jour, s'il est déjà installé) avec le gestionnaire de paquets apt en utilisant les commandes suivantes dans le terminal.

$ sudo apt update

$ sudo apt install joomscan

Il faut configuré un serveur de test avec Apache et Joomla installés.

Suivez nos exemples de commandes ci-dessous pendant que nous vérifions la sécurité de notre site Web de test.

Utilisez l'option --url et spécifiez l'URL du site Joomla afin de le scanner avec JoomScan.

$ joomscan --url http://example.com

JoomScan effectuera ensuite une analyse du site Web, qui se termine généralement en quelques secondes.

Certaines choses révélées par l'analyse sont les suivantes :

- Type de pare-feu utilisé pour protéger le site
- Quelle version de Joomla est en cours d'exécution
- Si cette version a des vulnérabilités principales
- Répertoires avec listes disponibles
- URL de connexion administrateur
- URL trouvées dans le fichier robots.txt
- Sauvegarde et fichiers journaux
- Page d'enregistrement de l'utilisateur

Certaines de ces informations sont utiles aux attaquants. L'analyse montre que les listes de répertoires sont activées, ce qui permet potentiellement aux attaquants de trouver des fichiers que le propriétaire pensait être cachés. Connaître l'URL d'administration signifie qu'un attaquant peut utiliser Hydra ou un autre outil similaire pour lancer une attaque par dictionnaire contre les informations de connexion.

JoomScan peut également énumérer les composants, ce qui révélera quel logiciel Joomla supplémentaire le propriétaire du site a installé. Si l'un d'entre eux a des failles de sécurité connues, il agira comme un autre vecteur d'attaque.

$ joomscan --url http://example.com --enumerate-components

Non seulement JoomScan listera les composants qu'un site utilise, mais s'ils contiennent des vulnérabilités connues, JoomScan vous en avertira et vous fournira un lien pour que vous puissiez en savoir plus.

D'autres options pour JoomScan incluent la possibilité de définir un agent utilisateur ou un agent aléatoire.

$ joomscan --url http://example.com --user-agent "Googlebot / 2.1 (+ http: //www.googlebot.com/bot.html)"

OU ALORS

$ joomscan --url http://example.com --random-agent

Utilisez un proxy pour analyser le site Joomla avec l'option --proxy.

$ joomscan --url www.example.com --proxy http://127.0.0.1:8080

Pour voir toutes ces options à tout moment, consultez le menu d'aide de JoomScan.

$ joomscan - Aide

WEBOGRAPHIE

openvpn [Wiki ubuntu-fr] (ubuntu-fr.org)

TUTOS.EU : Mettre en place un serveur openvpn

LAN to LAN OpenVPN - TUSER.NLTUSER.NL