



**ACADÉMIE  
DE GRENOBLE**

*Liberté  
Égalité  
Fraternité*



Réalisation en cours de formation

## **Administration & sécurisation réseau des équipements**



0 - Sommaire :

## Sommaire

- 1 – Mise en place d'un environnement de travail fonctionnel *(Page 3)*
  - 1.1 Installation des postes de travail (sertir & tirer câbles RJ45...)
  - 1.2 Installation des principaux équipements réseaux nécessaires
  - 1.3 Configuration du serveur Proxmox
- 2 – Administration & sécurisation réseau *(Page 3 → Page 5)*
  - 2.1 Notion de VLANs & LACP sur switch Cisco
  - 2.2 Configuration pare-feu (VPN IPsec, HA, NAT/PAT...) sur Stormshield
  - 2.3 Configuration Access-List & HSRP sur routeur Cisco
- 3 – TP réalisés *(Page 5 → Page 14)*
  - 3.1 Analyse de trames avec Wireshark
  - 3.2 Routage statique sur Cisco Packet Tracer
  - 3.3 Routage dynamique (OSPF / RIP v2) sur GNS3
  - 3.4 PENTEST avec Kali Linux (Brut force, Exploitation de failles avec Metasploit)
  - 3.5 Exercice d'injection SQL sur un site web local en PHP (phpMyAdmin)

## 1 – Mise en place d'un environnement de travail fonctionnel :

### - 1.1 Installation des postes de travail (sertir & tirer câbles RJ45...)

Au début de l'année 1, étant donné que nous étions la première promotion, on a eu la chance de pouvoir installer nos postes de travail. On a donc dû tous tirer 4 câbles de la baie vers notre station d'accueil puis tester leurs bons fonctionnements. Enfin on a installé nos équipements et périphériques c'est-à-dire 1 Dock & Station Lenovo all-in-one, 2 écrans Acer 24 pouces, clavier et souris de la marque Logitech.

### - 1.2 Installation des principaux équipements réseaux nécessaires

Une fois notre lieu de travail individuel installé, on a eu la chance de mettre en place nos premiers équipements réseaux, notamment le switch Cisco et les routeurs. On a pu faire nos premières configurations et se connecter avec notre ordinateur. Plus tard dans l'année nous sommes passé d'une petite baie de brassage à une plus grande d'au moins 40U pour donner 5U à chacun.

### - 1.3 Configuration du serveur Proxmox

On a mis en place des VM Ubuntu et Windows serveur toute au long de l'année pour différents projets. Cet hyperviseur a notamment été plus qu'utile pour le projet Noscea (entreprise fictive) de 1<sup>ère</sup> année et pour mes projets de fin d'année sur l'épreuve E6.

## 2 – Administration & sécurisation réseau :

### - 2.1 Notion de VLANs & LACP sur switch Cisco

L'utilisation des équipements de la marque Cisco a été une évidence pour apprendre à utiliser des VLANs, nommés « mode access » ou « mode trunk » correspondant respectivement à « untagged » et « tagged ». Cela correspondant à la protection d'un réseau en cloisonnant en différentes parties, différents réseaux, accessible ou non selon la configuration souhaitée. J'ai eu l'occasion de les utiliser plus particulièrement lors de la réalisation d'un de mes projets pour l'épreuve E6. J'ai également utilisé le LACP de Cisco dans mon projet, appelé EtherChannel, qui permet de créer une redondance de switch. C'est-à-dire, lorsqu'il y a 2 switches avec la même configuration au lieu d'un en cas de pannes.

## - 2.2 Configuration pare-feu (VPN IPsec, HA, NAT/PAT...) sur Stormshield

Les pare-feux de la marque Stormshield nous ont servis à faire pas mal de chose, je pense notamment au VPN IPsec que j'ai eu l'occasion d'établir avec un camarade. La HA -High-Availability) ou Haute-disponibilité en français qui permet d'avoir une redondance de pare-feu très recommandé par l'ANSSI dans le cas où l'un des 2 pare-feux tombe ou même pour mettre à jour le firmware sans coupure. J'ai pu monter une HA dans mon projet de l'épreuve E6. Ensuite, forcément qui dit pare-feu, dit règle de filtrage et NAT, j'ai également pu mettre en place des règles de filtrage lors de mon projet comme l'ANSSI le recommande :

SECURITY POLICY / FILTER - NAT

(5) Accès internet

Edit

Export

FILTERING

NAT

Searching...

+ New rule

Delete

Cut

Copy

Paste

Search in logs

Search in monitoring

	Status	Action	Source	Destination	Dest. port	Protocol	Security inspection	Comments
Règles d'autorisation des flux à destination du pare-feu (contains 1 rules, from 1 to 1)								
1		pass	Network_internals	firewall_all	firewall_srv https		IPS	Admin from Internal Network- Updated on 2024-11-...
Règles d'autorisation des flux émis par le pare-feu (contains 1 rules, from 2 to 2)								
2		pass	Network_internals	Firewall_all	Any	icmp	IDS	Allow Ping from Internal Network- Updated on 2024-...
Règle de protection du pare-feu (contains 1 rules, from 3 to 3)								
3		block	Any	Firewall_all	Any		IPS	Résolution DNS pour Users
Règles d'autorisation des flux métiers - Internet (contains 1 rules, from 4 to 4)								
4		pass	Network_internals	Internet	http https dns_udp		IPS	Accès internet
Règles "antiparasites" (contains 2 rules, from 5 to 6)								
5		block	Network_internals	Any	netbios-ns netbios-dgm		IPS	Block Netbios to extern
6		block	Network_internals	Any	microsoft-ds_tcp		IPS	Block SMB to extern
Block All (contains 1 rules, from 7 to 7)								
7		block	Any	Any	Any		IPS	Block all - Updated on 2024-11-05 15:18:50 by admin...

SECURITY POLICY / FILTER - NAT

(5) Accès internet

Edit

Export

FILTERING

NAT

Searching...

+ New rule

Delete

Cut

Copy

Paste

Search in logs

Search in monitoring

	Status	Original traffic (before translation)			Traffic after translation			Protocol	Options	Comments
		Source	Destination	Dest. port	Source	Src. port	Destination			
PAT - Internet (contains 1 rules, from 1 to 1)										
1		Any	Internet	Any	→	Firewall_out	↔	ephemera		Created on 2024-11-05 15:20:46 by admin (192.168.210.3)

## - 2.3 Configuration d'Access-List & HSRP sur routeur Cisco

On a aussi vu qu'un réseau pouvait utiliser des routeurs avec des Access-List pour autoriser un flux (White List IP) avec deny all ou alors pour bloquer seulement une IP en particulier (Black List), il existe 2 types d'Access List : ACL -standard & -étendue

Standard:

N°1 à 99 | nom (permit | deny) (host | Network + wildcard | any)

Exemple: Access-list 1 permit any (= Autorise tout)



Etendue:

N°100 -|nom (permit | deny) protocol source+wildcard [Operateur Port]  
destination+wildcard [Operateur Port] [established(sit Tcp)]

Exemple: Access-list 100 deny TCP 192.168.210.0 0.0.0.255 192.168.208.0 0.0.0.255

Résultat sur le routeur :

```
Router#sh access-lists
Standard IP access list 1
  10 permit 192.168.210.0, wildcard bits 0.0.0.255 (5071 matches)
Extended IP access list 100
  10 permit tcp 192.168.208.0 0.0.0.255 192.168.210.0 0.0.0.255 established
  20 permit tcp 192.168.208.0 0.0.0.255 192.168.210.0 0.0.0.255 eq www
Router#
```

Comme l’EtherChannel des switch Cisco ou encore la HA de Stormshield, les routeurs Cisco ont une redondance possible avec la configuration du HSRP (Hot Standby Router Protocol). Lors de la mise en place du HSRP, les deux routeurs ne feront qu’un, avec une seule adresse IP. Le principe est simple, comme chaque redondance, le but est d’assurer une continuité des services en cas de panne d’un des équipements.

### 3 – TP réalisés :

#### - 3.1 Analyse de trames avec Wireshark

Wireshark est un outil très utile pour analyser les trames réseau. J’ai eu l’occasion de l’utiliser pour différentes tâches comme lors d’un exercice de recherche d’un mot de passe chiffré dans une trame ou encore dans le TP téléphonie avec les trames VOIP.

TP mot de passe, extraire mot de passe d’une trame FTP :

tcp.port==21						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.20.144.150	10.20.144.151	TCP	74	35974 → 21 [SYN] Seq=0 Win=32648 Len=0 MSS=1380 WS=1 TSval=1657560000 TSecr=0
2	0.000320	10.20.144.151	10.20.144.150	TCP	78	21 → 35974 [SYN, ACK] Seq=0 Ack=1 Win=16384 Len=0 MSS=1356 WS=1 TSval=1657390000 TSecr=165
3	0.000570	10.20.144.150	10.20.144.151	TCP	66	35974 → 21 [ACK] Seq=1 Ack=1 Win=32648 Len=0 TSval=1657560000 TSecr=1657390000
4	0.060630	10.20.144.151	10.20.144.150	FTP	106	Response: 220-OTCP at fran.csg.stercomm.com.
5	0.275440	10.20.144.150	10.20.144.151	TCP	66	35974 → 21 [ACK] Seq=1 Ack=37 Win=32648 Len=0 TSval=1657560500 TSecr=1657390000
6	0.275760	10.20.144.151	10.20.144.150	FTP	126	Response: 220 Connection will close if idle more than 5 minutes.
7	0.276140	10.20.144.150	10.20.144.151	TCP	66	35974 → 21 [ACK] Seq=1 Ack=93 Win=32648 Len=0 TSval=1657560500 TSecr=1657390000
8	4.216600	10.20.144.150	10.20.144.151	FTP	81	Request: USER cdt3500
9	4.217350	10.20.144.151	10.20.144.150	FTP	91	Response: 331 Enter password.
10	4.217630	10.20.144.150	10.20.144.151	TCP	66	35974 → 21 [PSH, ACK] Seq=16 Ack=114 Win=32648 Len=0 TSval=1657564500 TSecr=1657394000
11	7.639420	10.20.144.150	10.20.144.151	FTP	81	Request: PASS cdt3500

Extraire un mdp chiffré d’un protocole POP :

POP utilise le chiffrement MD5 et la clé est :4ddd4137b84ff2db7291b568289717f0

Le mot de passe chiffré est : [1755.1.5f403625.BcWGgpKzUPRC8vscWn0wuA==@vps-7e2f5a72](#)

On peut ensuite utiliser Hashcat pour cracker le mot de passe chiffré :

Etat AUTORISATION.

Notez qu'à mesure que la longueur du secret partagé augmente, fait la difficulté de le dériver. A ce titre, partagé les secrets doivent être de longues chaînes (considérablement plus longues que l'exemple à 8 caractères présenté ci-dessous).

Réponses possibles :

+OK maildrop verrouillé et prêt  
-Autorisation ERR refusée

Exemples:

S : +OK Serveur POP3 prêt <1896.697170952@dbc.mtview.ca.us>  
C : APOP mrose c4c9334bac560ecc979e58001b3e22fb  
S: +OK maildrop contient 1 message (369 octets)

Dans cet exemple, le secret partagé est la chaîne `tanstaaf`. Par conséquent, l'algorithme MD5 est appliqué à la chaîne

<1896.697170952@dbc.mtview.ca.us>tanstaaf

qui produit une valeur de résumé de

c4c9334bac560ecc979e58001b3e22fb

En utilisant hashcat, la commande est :

.\hashcat.exe hashcat -m 0 -a 7 "4ddd4137b84ff2db7291b568289717f0"

"<1755.1.5f403625.BcWGgpKzUPRC8vscWn0wuA==@vps-7e2f5a72>" "C:\Users\Utilisateur\Desktop\BTS-Cours\Cyber\TP Analyse de trame\rockyou.txt"

```
PS C:\Users\Utilisateur\Desktop\BTS-Cours\Cyber\TP Analyse de trame\hashcat-6.2.6> .\hashcat.exe -m 0 -a 7 "4ddd4137b84ff2db7291b568289717f0" "<1755.1.5f403625.BcWGgpKzUPRC8vscWn0wuA==@vps-7e2f5a72>" "C:\Users\Utilisateur\Desktop\BTS-Cours\Cyber\TP Analyse de trame\rockyou.txt"
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]
* Device #1: Intel(R) Iris(R) Xe Graphics, 3168/6443 MB (1610 MB allocatable), 80MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Dictionary cache hit:
* Filename..: C:\Users\Utilisateur\Desktop\BTS-Cours\Cyber\TP Analyse de trame\rockyou.txt
* Passwords.: 14344384
* Bytes.....: 139921497
* Keyspace...: 14344384

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates

Optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Hash
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Pure kernels can crack longer passwords, but drastically reduce performance.
If you want to switch to optimized kernels, append -O to your commandline.
See the above message to find out about the exact limits.

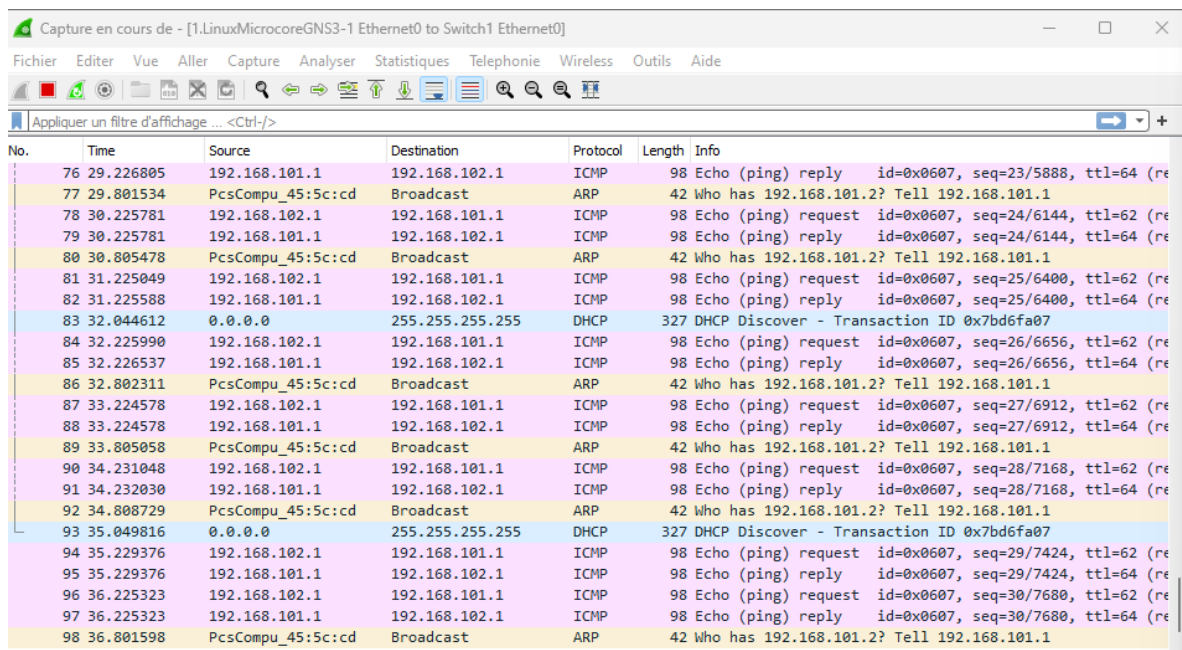
Watchdog: Hardware monitoring interface not found on your system.
```

4ddd4137b84ff2db7291b568289717f0:<1755.1.5f403625.BcWGgpKzUPRC8vscWn0wuA==@vps-7e2f5a72>100%popprincess

Session.....: hashcat  
Status.....: Cracked

J'ai par exemple, également pu analyser le protocole ICMP qui a traversé l'une des liaisons entre deux machines. ICMP qui signifie (Internet Control Message Protocol) est un protocole de la couche réseau utilisé par les appareils du réseau pour diagnostiquer les problèmes de communication du réseau. ICMP est principalement utilisé pour déterminer si les données atteignent ou non leur destination en temps voulu grâce à un « ping ». Voici ce qu'on a obtenu respectivement dans l'ordre d'affichage des pings de la machine 1 vers la 2 et de la machine 2 vers la 1 :

- Machine 1 : « ping 192.168.102.1 »



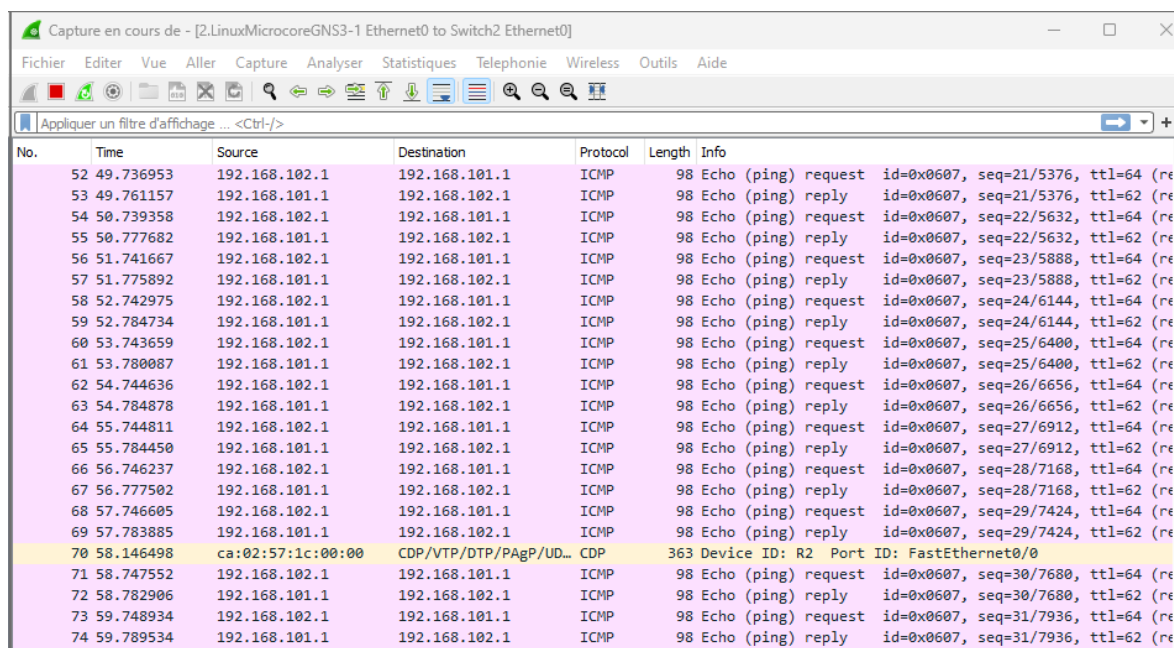
Capture en cours de - [1.LinuxMicrocoreGNS3-1 Ethernet0 to Switch1 Ethernet0]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
76	29.226805	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=23/5888, ttl=64 (re
77	29.801534	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1
78	30.225781	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=24/6144, ttl=62 (re
79	30.225781	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=24/6144, ttl=64 (re
80	30.805478	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1
81	31.225049	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=25/6400, ttl=62 (re
82	31.225588	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=25/6400, ttl=64 (re
83	32.044612	0.0.0.0	255.255.255.255	DHCP	327	DHCP Discover - Transaction ID 0x7bd6fa07
84	32.225990	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=26/6656, ttl=62 (re
85	32.226537	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=26/6656, ttl=64 (re
86	32.802311	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1
87	33.224578	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=27/6912, ttl=62 (re
88	33.224578	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=27/6912, ttl=64 (re
89	33.805058	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1
90	34.231048	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=28/7168, ttl=62 (re
91	34.232030	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=28/7168, ttl=64 (re
92	34.808729	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1
93	35.049816	0.0.0.0	255.255.255.255	DHCP	327	DHCP Discover - Transaction ID 0x7bd6fa07
94	35.229376	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=29/7424, ttl=62 (re
95	35.229376	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=29/7424, ttl=64 (re
96	36.225323	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=30/7680, ttl=62 (re
97	36.225323	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=30/7680, ttl=64 (re
98	36.801598	PcsCompu_45:5c:cd	Broadcast	ARP	42	Who has 192.168.101.2? Tell 192.168.101.1

- Machine 2 : « ping 192.168.101.1 »



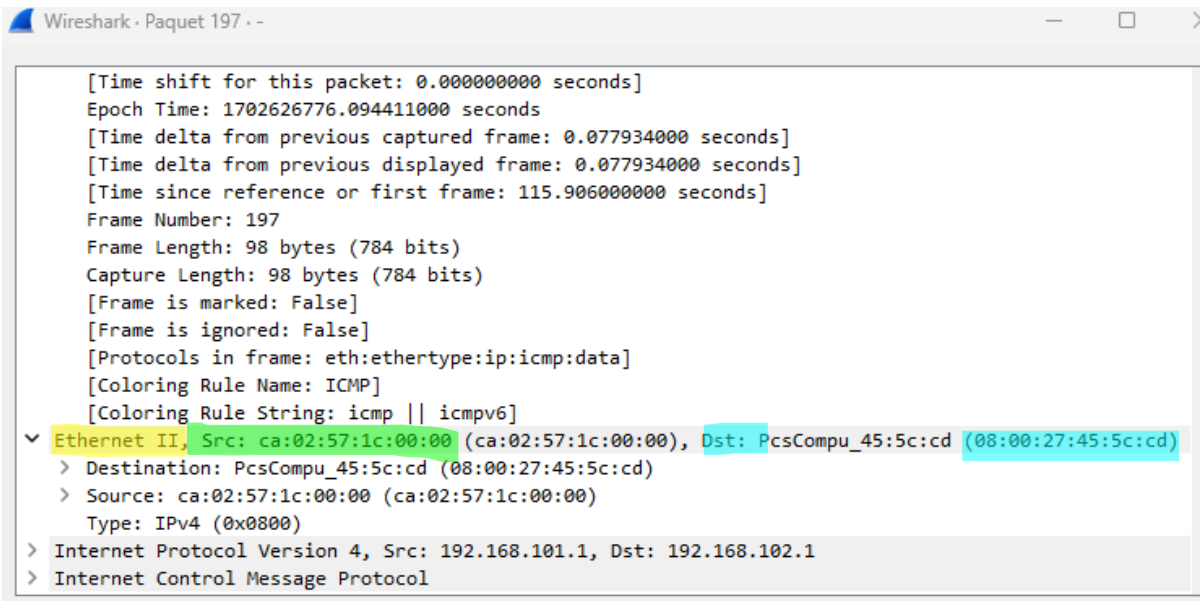
Capture en cours de - [2.LinuxMicrocoreGNS3-1 Ethernet0 to Switch2 Ethernet0]

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

Appliquer un filtre d'affichage ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
52	49.736953	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=21/5376, ttl=64 (re
53	49.761157	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=21/5376, ttl=62 (re
54	50.739358	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=22/5632, ttl=64 (re
55	50.777682	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=22/5632, ttl=62 (re
56	51.741667	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=23/5888, ttl=64 (re
57	51.775892	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=23/5888, ttl=62 (re
58	52.742975	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=24/6144, ttl=64 (re
59	52.784734	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=24/6144, ttl=62 (re
60	53.743659	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=25/6400, ttl=64 (re
61	53.780087	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=25/6400, ttl=62 (re
62	54.744636	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=26/6656, ttl=64 (re
63	54.784878	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=26/6656, ttl=62 (re
64	55.744811	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=27/6912, ttl=64 (re
65	55.784450	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=27/6912, ttl=62 (re
66	56.746237	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=28/7168, ttl=64 (re
67	56.777502	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=28/7168, ttl=62 (re
68	57.746605	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=29/7424, ttl=64 (re
69	57.783885	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=29/7424, ttl=62 (re
70	58.146498	ca:92:57:1c:00:00	CDP/VTP/DTP/PagP/UD...	CDP	363	Device ID: R2 Port ID: FastEthernet0/0
71	58.747552	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=30/7680, ttl=64 (re
72	58.782906	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=30/7680, ttl=62 (re
73	59.748934	192.168.102.1	192.168.101.1	ICMP	98	Echo (ping) request id=0x0607, seq=31/7936, ttl=64 (re
74	59.789534	192.168.101.1	192.168.102.1	ICMP	98	Echo (ping) reply id=0x0607, seq=31/7936, ttl=62 (re

Prenons au hasard une des trames ci-dessus de la machine 2 et on obtient :



```
Wireshark - Paquet 197 -  
[Time shift for this packet: 0.000000000 seconds]  
Epoch Time: 1702626776.094411000 seconds  
[Time delta from previous captured frame: 0.077934000 seconds]  
[Time delta from previous displayed frame: 0.077934000 seconds]  
[Time since reference or first frame: 115.906000000 seconds]  
Frame Number: 197  
Frame Length: 98 bytes (784 bits)  
Capture Length: 98 bytes (784 bits)  
[Frame is marked: False]  
[Frame is ignored: False]  
[Protocols in frame: eth:ethertype:ip:icmp:data]  
[Coloring Rule Name: ICMP]  
[Coloring Rule String: icmp || icmpv6]  
▼ Ethernet II, Src: ca:02:57:1c:00:00 (ca:02:57:1c:00:00), Dst: PcsCompu_45:5c:cd (08:00:27:45:5c:cd)  
  > Destination: PcsCompu_45:5c:cd (08:00:27:45:5c:cd)  
  > Source: ca:02:57:1c:00:00 (ca:02:57:1c:00:00)  
    Type: IPv4 (0x0800)  
  > Internet Protocol Version 4, Src: 192.168.101.1, Dst: 192.168.102.1  
  > Internet Control Message Protocol
```

Cette analyse de trame est très utile pour comprendre ce qu'il se passe dans ce protocole, intéressons-nous au déroulé « Ethernet II » surligné en jaune. Sur cette ligne, 2 adresses uniques MAC d'équipements y sont notifiées, celle de l'équipement source en vert  
Source: ca:02:57:1c:00:00 (ca:02:57:1c:00:00), celui qui envoie, en l'occurrence la machine 2, et celle de la machine 1 Destination: PcsCompu\_45:5c:cd (08:00:27:45:5c:cd), qui est celle de destination en bleu.

On en conclut alors que lorsque le paquet ICMP du ping dépasse le routeur, il y a un changement dans la trame au niveau des adresses MAC qui permettent d'identifier la provenance du paquet et sa destination, elle sera alors inversée si l'on analyse la trame du ping de la machine 1 vers la machine 2.

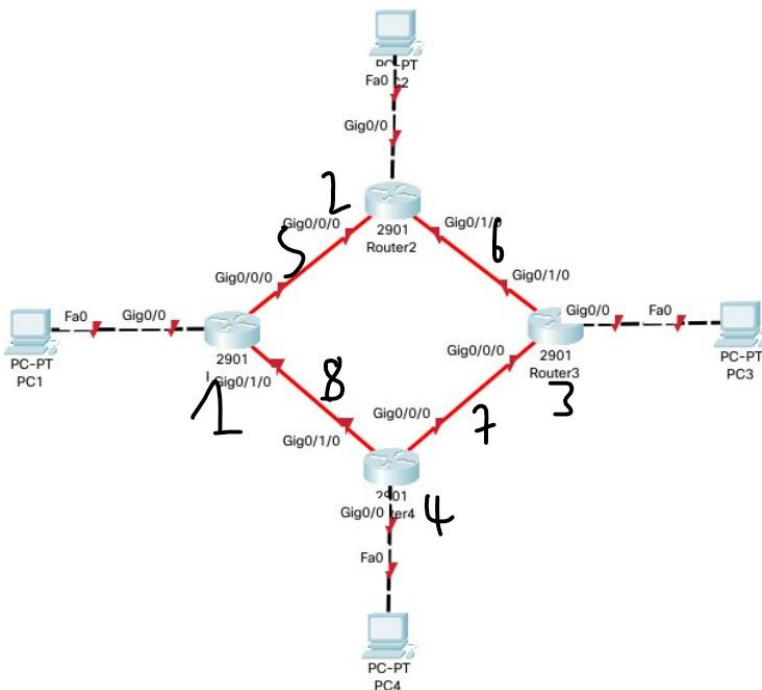
### - 3.2 Routage statique sur Cisco Packet Tracer

Le routage statique est complexe car plus il y a de routeurs, plus les routes à prévoir sont longues et on risque d'en oublier certaines. C'est pour cette raison qu'il faut d'abord créer les tables de routages avant de configurer chaque équipement :



Table de Routage du routeur 1		
Réseau à atteindre	Prochaine étape	Adresse IP de la prochaine étape
2	Routeur 2	10.12.2.1 / 16
3	Routeur 2	10.12.2.1 / 16
4	Routeur 4	10.14.4.1 / 16
6	Routeur 2	10.12.2.1 / 16
7	Routeur 4	10.14.4.1 / 16
Table de Routage du routeur 2		
Réseau à atteindre	Prochaine étape	Adresse IP de la prochaine étape
1	Routeur 1	10.12.1.1 / 16
3	Routeur 3	10.23.3.1 / 16
4	Routeur 3	10.23.3.1 / 16
7	Routeur 1	10.12.1.1 / 16
8	Routeur 1	10.12.1.1 / 16
Table de Routage du routeur 3		
Réseau à atteindre	Prochaine étape	Adresse IP de la prochaine étape
1	Routeur 2	10.23.2.1 / 16
2	Routeur 2	10.23.2.1 / 16
4	Routeur 4	10.34.4.1 / 16
5	Routeur 2	10.23.2.1 / 16
8	Routeur 4	10.34.4.1 / 16

Tables de routage associé à ce réseau :



Le but d'une route est de pouvoir accéder d'un appareil A vers un appareil B en ayant toute une infrastructure entre, des routeurs par exemple. Si les tables de routage ne sont pas justes, aucune connexion sera établie entre l'appareil A et l'appareil B. Une simple requête ICMP (ping) suffit à tester cette connexion filaire.

### - 3.3 Routage dynamique (OSPF / RIP v2) sur GNS3

OSPF : Open shortest Path First, c'est un protocole de routage interne IP permettant le routage d'adresse IP en dynamique tout comme le protocole RIP.

RIP V2: Routing Information Protocol version 2, c'est également un protocole de routage interne IP destiné plus particulièrement aux petites infrastructures.

Leur but est de permettre que les routeurs échangent activement des informations sur l'état du réseau. Lorsqu'un changement survient, comme l'ajout d'un nouveau réseau ou la défaillance d'un lien, ces protocoles de routage dynamique mettent à jour leurs tables de routage et propagent ces informations à travers le réseau. Voici quelques points permettant de les comparer :

#### Complexité :

RIPV2 est plus simple à configurer car il utilise une métrique simple basée sur le nombre de sauts.

OSPF est plus complexe à configurer car il prend en compte des facteurs tels que la bande passante, la charge de trafic et la fiabilité des liaisons.

#### Taille du réseau :

RIPV2 convient mieux aux petits réseaux en raison de sa simplicité.

OSPF est mieux adapté aux réseaux de taille moyenne à grande car il offre une meilleure évolutivité et une gestion plus efficace des grandes topologies.

#### Convergence :

RIPV2 peut prendre plus de temps pour converger dans de grandes topologies en raison de son délai de convergence lent.

OSPF converge généralement plus rapidement grâce à son protocole de mise à jour d'état de liaison.

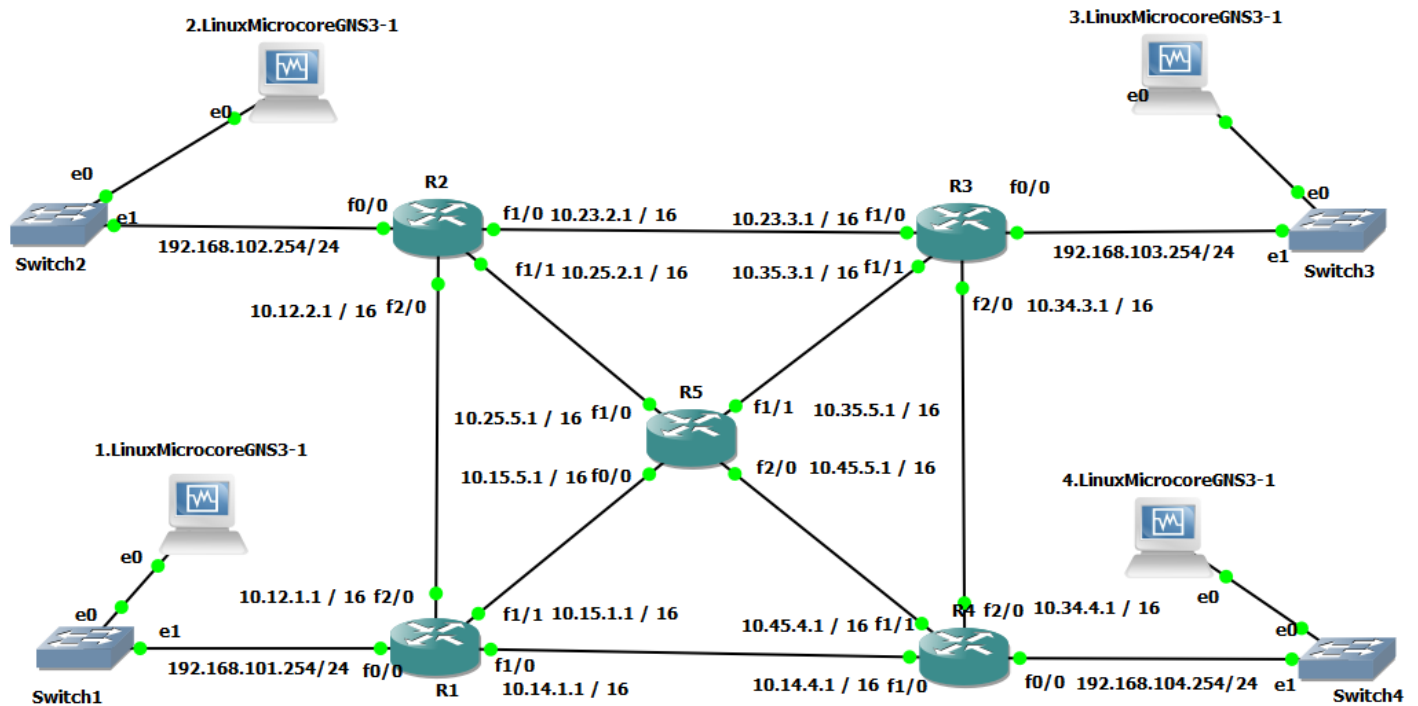
#### Sécurité :

RIPV2 ne prend pas en charge nativement l'authentification des mises à jour de routage, ce qui le rend plus vulnérable aux attaques de type "man-in-the-middle".

OSPF prend en charge l'authentification des messages OSPF, ce qui le rend plus sécurisé contre les attaques de routage malveillantes.

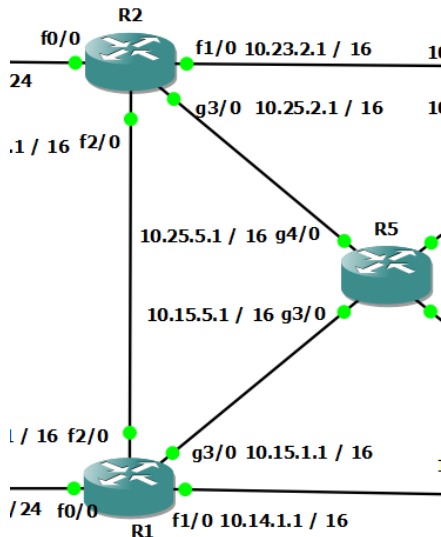
En résumé, OSPF est généralement considéré comme meilleur en termes de sécurité en raison de son support intégré pour l'authentification des messages OSPF et de sa capacité à

gérer efficacement de grandes topologies réseau. Cependant, pour les réseaux plus petits et moins complexes, RIPV2 peut être plus facile à configurer et suffisamment efficace. De plus, le protocole OSPF est plus récent et s'adapte normes actuelles. D'ailleurs lors de la création du protocole OSPF, le but était de remplacer le protocole RIP. C'est pour cela que j'ai choisis le protocole OSPF pour le TP.



Ci-dessus, cette infrastructure est composée de 4 postes (pc/laptop) déployé en machines virtuelles sous linux TinyCore sur VirtualBox nommé de 1 à 4. Elles sont liées chacune à leur switch respectif et ces derniers sont reliés à un routeur également nommé respectivement aux machines. Ces routeurs sont reliés entre eux et un cinquième routeur a été rajouté au milieu au cas où un équipement venait à tomber en panne ou une liaison céderait.

Il y avait également une partie bonus du TP qui consistait à monter différentes technologies sur les routeurs (FastEthernet, GigaBit...) pour ensuite faire un « traceroute » et voir si le chemin le plus court est le plus rapide. J'ai donc remplacé les liaisons FastEthernet entre R1 <> R5 et R2 <> R5 par des liaisons GigaBit. Ce qui donne ceci :



Le ping passait d'une machine à l'autre donc le routage dynamique à bien fonctionné.

### - 3.4 PENTEST avec Kali Linux (Brut force, Exploitation de failles avec Metasploit)

On a eu l'occasion d'utiliser le potentiel de l'OS Kali sous linux spécialisé dans la sécurité informatique. On a tout d'abord revu les commandes linux de bases et utilisé Matasploit pour des recherches de vulnérabilité existante sur différents paramètres comme sur Tomcat, Windows 10, 11...

Ensuite on s'est intéressé au brute force en utilisant l'outil puissant Hydra qui permet de cracker des mots de passe. Pour cela, on a récupéré la liste de mots de passes en lignes Rockyou.txt (un dictionnaire contenant pratiquement tous les mots de passe les plus courants et connus du + simple ou + complexe). Une fois fait, on a tenté de cracker le mot de passe en connaissant le login d'un serveur WEB non sécurisé que l'un d'entre nous avait déployé.

Enfin, chacun a utilisé NMAP pour découvrir le serveur Web, puis Metasploit pour savoir comment tenter d'attaquer le serveur puis on a attaqué par force brute mais le résultat n'était pas celui attendu car vu que tout le monde essayait de cracker le mot de passe, le serveur s'est plutôt pris une attaque de type DoS (déni de service) pas forcément souhaité au début...

### - 3.5 Exercice d'injection SQL sur un site web local en PHP (phpMyAdmin)

On s'est intéressé aux injections SQL qui consiste à injecter du code SQL dans des champs input, des commandes du type : ' admin OR 1 = 1 - - . Pour cela, chacun à hébergé une page php avec phpMyAdmin via l'intermédiaire de XAMMP pour héberger localement notre page



web. On a ensuite pu injecter notre code malveillant dans la page d'authentification avec « admin » comme utilisateur et on a accédé à la page sans connaître le mot de passe. J'ai ensuite demandé à ChatGPT, à la demande du professeur, de m'expliquer la différence entre un code php sensible aux injections SQL et insensibles et de me modifier mon code existant pour qu'il devienne insensible aux attaques de ce type tout en m'expliquant la différence.

Le code non sécurisé :

```
<?php
// Connexion à la base de données (sans protection contre l'injection SQL)
$conn = new mysqli("localhost", "root", "", "test_db");

if ($conn->connect_error) {
    die("Échec de la connexion : " . $conn->connect_error);
}

// Vérification des identifiants (vulnérable à l'injection SQL)
if (isset($_POST['username']) && isset($_POST['password'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Requête non sécurisée (vulnérable aux injections SQL)
    $sql = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        echo "<h2>Connexion réussie ! Bienvenue, " . htmlspecialchars($username) . "</h2>";
    } else {
        echo "<h2>Identifiants incorrects.</h2>";
    }
}
?>

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Vulnérable</title>
</head>
<body>
    <h2>Page de connexion (Vulnérable à l'injection SQL)</h2>
    <form method="post">
        <label>Nom d'utilisateur :</label>
        <input type="text" name="username" required><br>
        <label>Mot de passe :</label>
        <input type="password" name="password" required><br>
        <input type="submit" value="Se connecter">
    </form>
</body>
</html>
```

Le code sécurisé (requête préparé) :

```
<?php
// Connexion sécurisée à la base de données avec mysqli
$conn = new mysqli("localhost", "root", "", "test_db");

// Vérification de la connexion
if ($conn->connect_error) {
    die("Échec de la connexion : " . $conn->connect_error);
}

// Vérification des identifiants (sécurisé avec requêtes préparées)
if (isset($_POST['username']) && isset($_POST['password'])) {
    $username = $_POST['username'];
    $password = $_POST['password'];

    // Requête préparée pour éviter l'injection SQL
    $stmt = $conn->prepare("SELECT * FROM users WHERE username = ? AND password = ?");
    $stmt->bind_param("ss", $username, $password); // "ss" signifie deux chaînes (string)

    $stmt->execute();
    $result = $stmt->get_result();

    if ($result->num_rows > 0) {
        echo "<h2>Connexion réussie ! Bienvenue, " . htmlspecialchars($username) . "</h2>";
    } else {
        echo "<h2>Identifiants incorrects.</h2>";
    }

    $stmt->close();
}
$conn->close();
?>

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Login - Sécurisé</title>
</head>
<body>
    <h2>Page de connexion (Sécurisée contre l'injection SQL)</h2>
    <form method="post">
        <label>Nom d'utilisateur :</label>
        <input type="text" name="username" required><br>
        <label>Mot de passe :</label>
        <input type="password" name="password" required><br>
        <input type="submit" value="Se connecter">
    </form>
</body>
</html>
```