UV F2B505
SECURITE INFORMATIQUE ET RESEAUX
TP VPN

VO LE MINH QUAN
KOUNGA FREDERIC FRANCK

1. LE RESEAU EMULE

1.2 TOPOLOGIE DU RESEAU DE TEST

Bien vérifié => les machines sont bien configurées comme la topologie

2. UN VPN SSL : OPENVPN

2.1 CREATION D'UN VPN RESEAU A RESEAU

- *ipY* ?

- L'adresse IP du machine gateway à distance

- Quel est le rôle de *tun1* ?

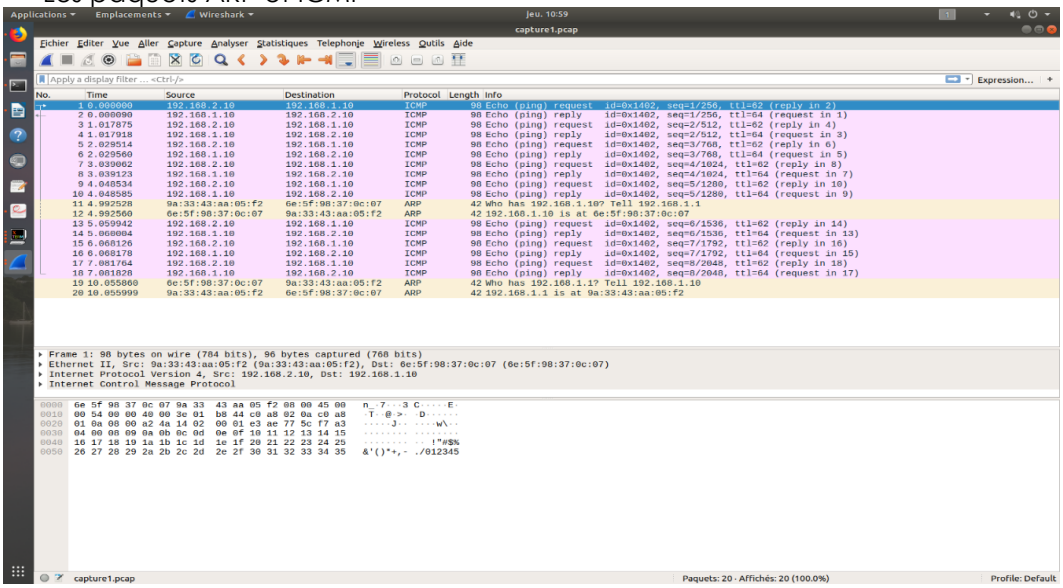
- Définir une nouvelle interface virtuelle

- Expliquez le sens de l'opération ?

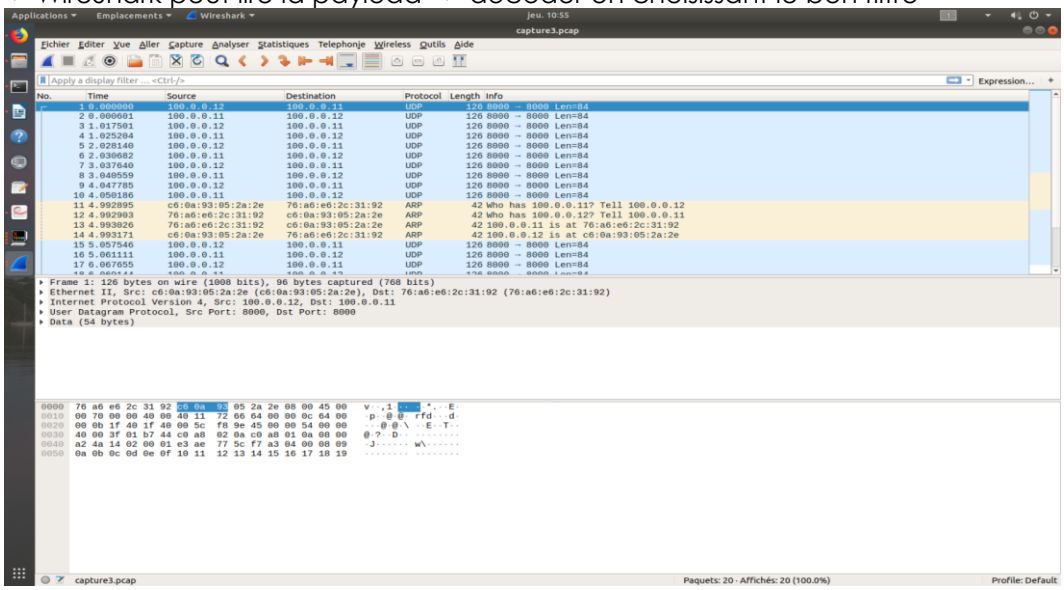
- Configurer la passerelle gw1 : *route add -net 192.168.2.0/24 gw 100.0.0.12*
  => faire le routage pour transmettre des paquets

- Quel type de trafic est échangé ? *ping* du pc2 au pc1

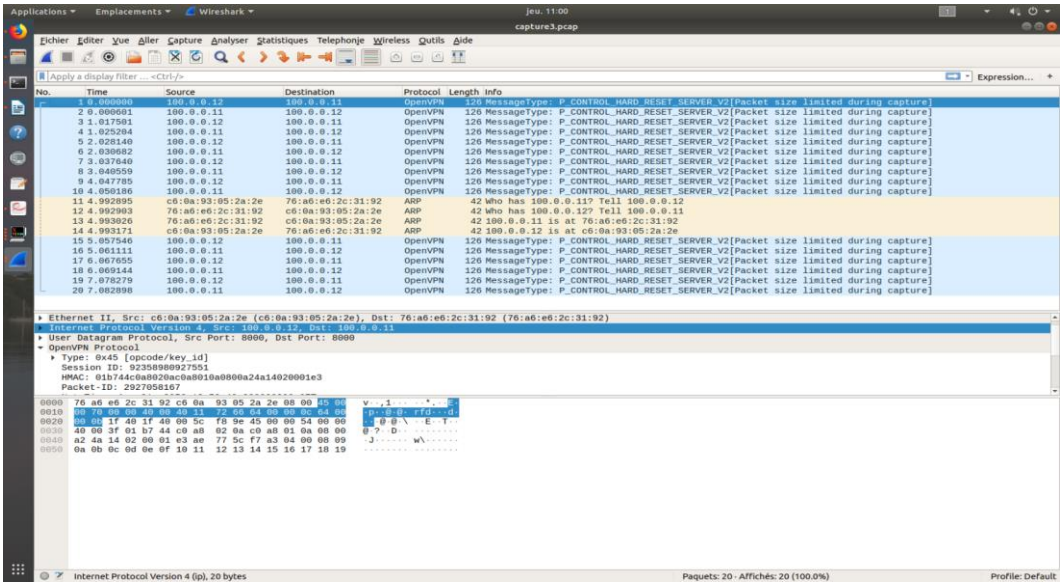
- Ecoute sur pc1
  => Les paquets ARP et ICMP



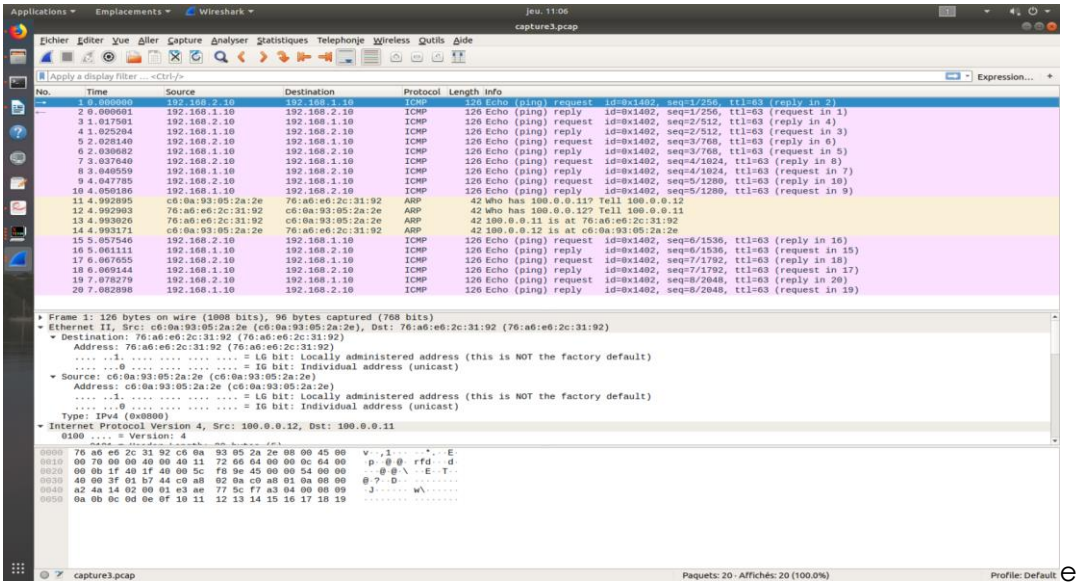
- Ecoute sur pc3 : 2 gateways communiquent entre eux par des paquets UDP, non sécurisé
  => WireShark peut lire la payload => décoder en choisissant le bon filtre



- Filtre OpenVPN => pas bon



- Décoder avec le filtre IPv4 => bon, on peut voir les paquets en clair



e

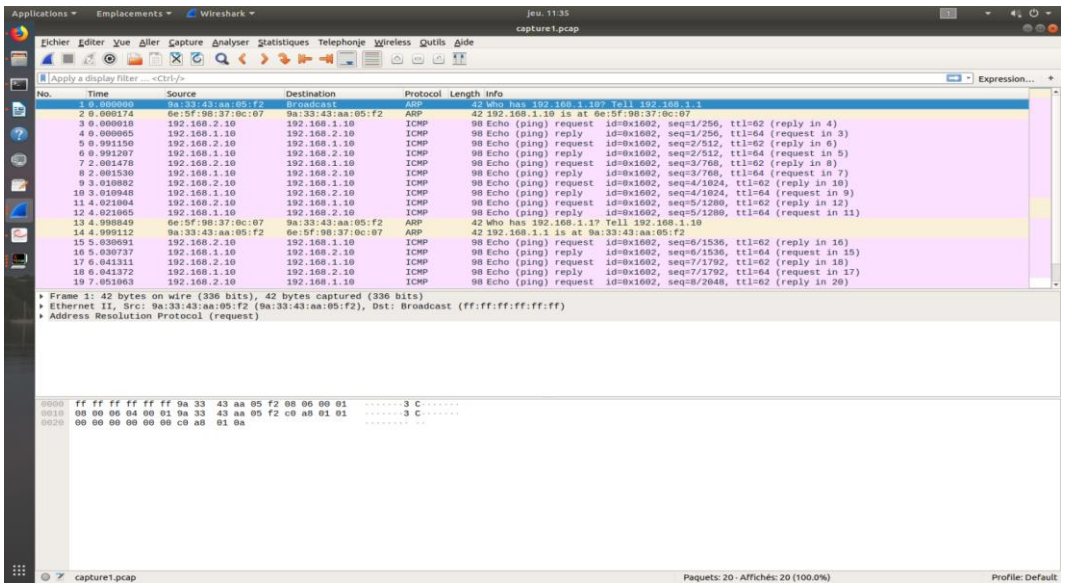
- Encapsulation

- Mettre des paquets IPv4 et ICMP dans UDP
- En général, c'est le fait d'injecter des paquets dans la charge utile d'autres paquets.

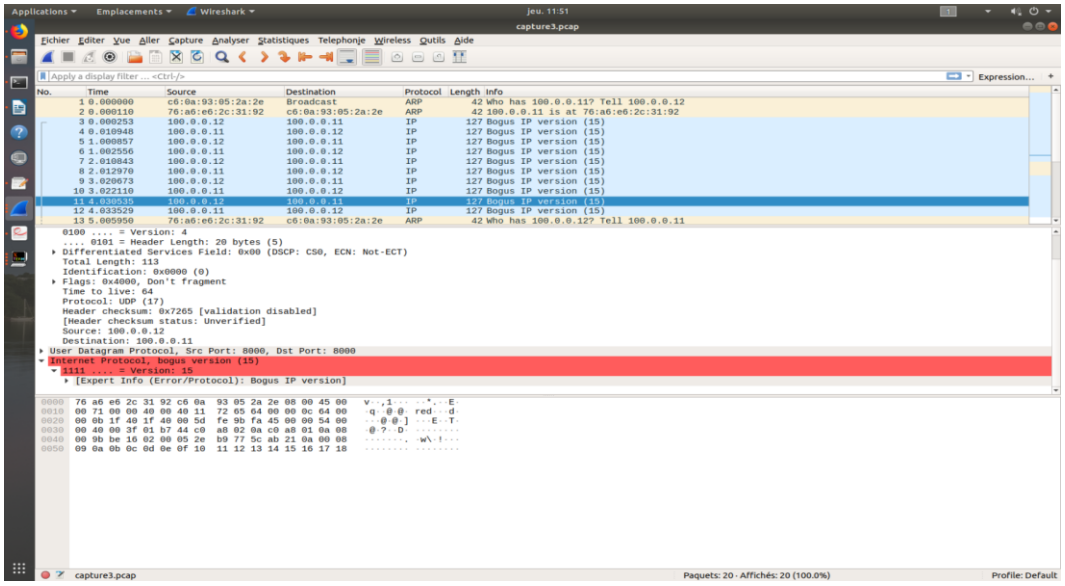
L'OPTION COMPRESSION

On émet un ping du pc1 au pc2

ECOUTE SUR LE PC1



ECOUTE SUR LE PC3



Remarque

=> On ne peut plus lire le contenu des paquets (Wireshark n'est pas très intelligent)

=> Ils sont compressés non sécurisés

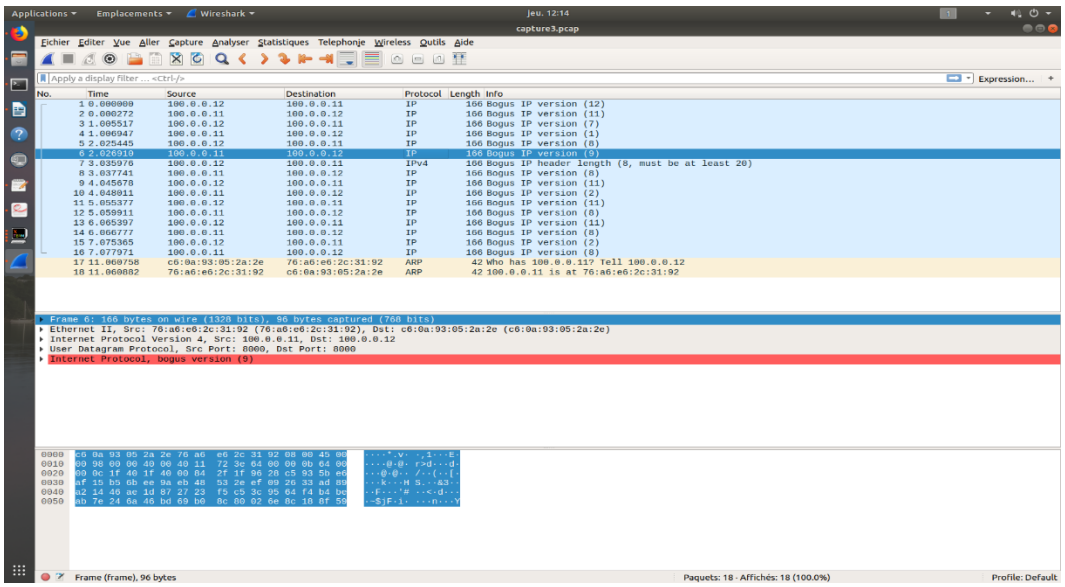
=> Peut-être que quelqu'un pourrait décompresser en utilisant l'algorithme exact

2.2 SECURISATION DU VPN

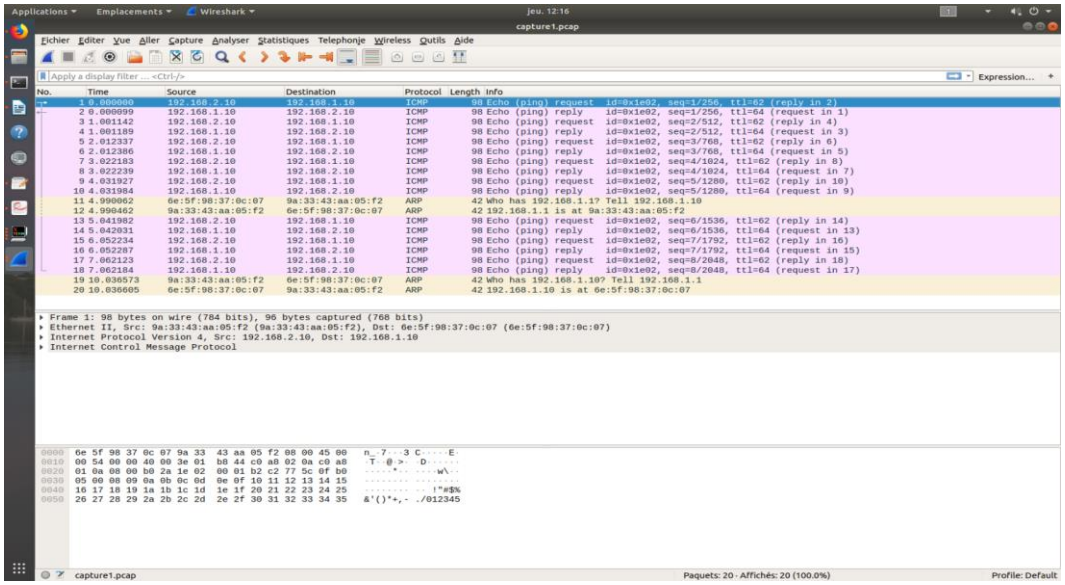
2.2.1 CHIFFREMENT

Les données sont bien chiffrées de manière symétrique

ECOUTE SUR PC3



ECOUTE SUR LE PC1



=> Bien chiffrée au pc3 et bien déchiffrée au pc1
=> Si on lance un gateway avec la clé et un autre sans clé =>Erreur d'établir la connexion

2.2.2 GENERATION DE CERTIFICATS

- Que génère-t-on à chaque étape

- 1e : la clé privée et le certificat de l'autorité de certification (CA)
- 2e : la clé privée et la demande de certificat pour gw1
- 3e : générer le certificat => signer la demande

- Quel est le rôle de ***-x509*** dans la première opération ?

- La commande x509 est un utilitaire de certificat polyvalent. Il peut être utilisé pour afficher des informations sur les certificats, convertir des certificats en divers formulaires, signer des

demandes de certificat comme une "mini-autorité de certification" ou modifier les paramètres de confiance du certificat.

- Quel est le rôle de ***-nodes*** ?

Cela signifie que OpenSSL ne chiffrera pas la clé privée

- Que décrit l'argument de ***-subj*** ? Que signifient les différents champs déclarés ?

- Il définit le nom de sujet pour la nouvelle demande ou remplace le nom de sujet lors du traitement d'une demande.
- Country Name, State or Province Name, Locality Name, Organization Name, Organizational Unit Name, Common Name, Email Address, Password, Company Name (optional)

2.2.3 GENERATION DU FICHIER DE PARAMETRES POUR L'ECHANGE DIFFIE – HELLMAN

- Que venez-vous de générer ?

- Un fichier de paramètres avec la longueur de 1024 bits pour l'échange Diffie-Hellman.

2.2.4 DEMARRER OPENVPN

On considère que gw1 : serveur et gw2 : client

CONFIGURATION DE PW1

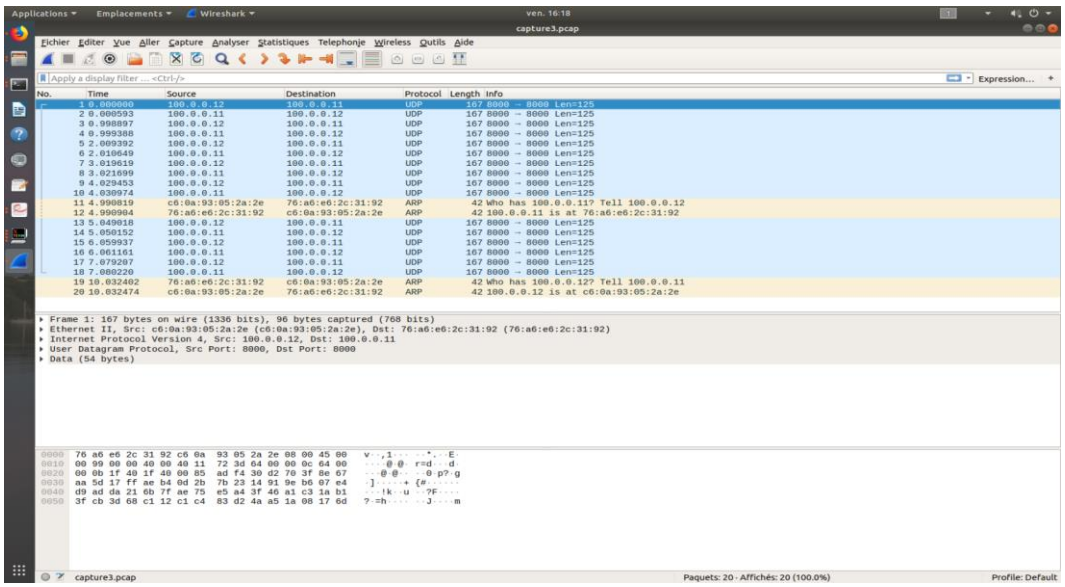
remote 100.0.0.12
port 8000
dev tun1
ifconfig 192.168.0.1 192.168.0.2
verb 5
tls-server
key /root/priv-gw1.pem
cert /root/crt-gw1.pem
ca /root/crt-ca.pem
dh /root/dh1024.pem

CONFIGURATION DE GW2

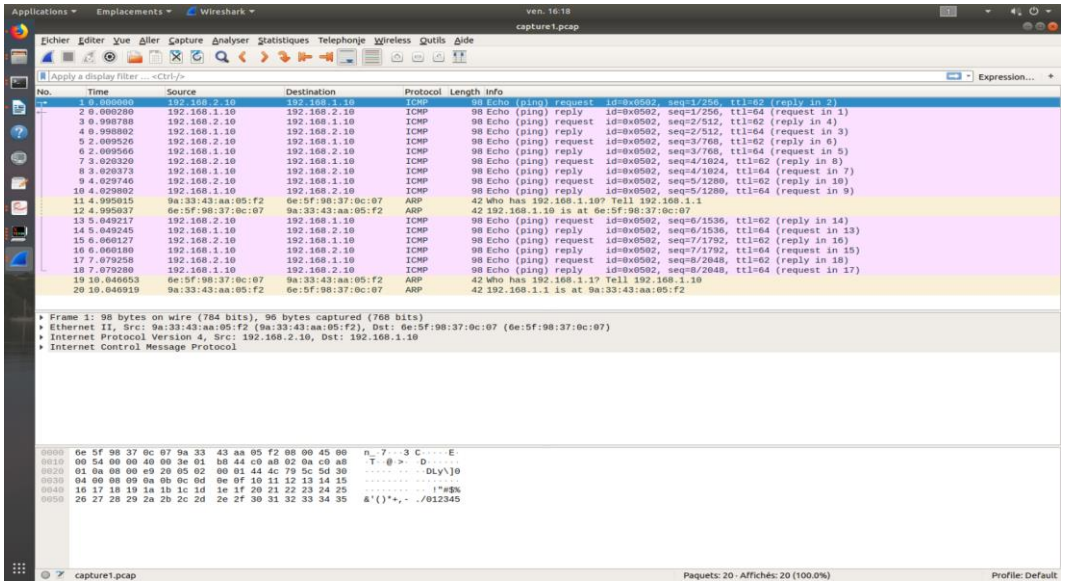
remote 100.0.0.11
port 8000
dev tun1
ifconfig 192.168.0.2 192.168.0.1
verb 5
tls-client
key /root/priv-gw2.pem
cert /root/crt-gw2.pem
ca /root/crt-ca.pem

RESULTAT

SUR LE PC3


Applications Emplacements Wireshark ven. 16:18
capture3.pcap
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
2 0.000593 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
3 0.998897 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
4 0.999388 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
5 2.009392 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
6 2.010649 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
7 3.019619 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
8 3.021099 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
9 4.029453 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
10 4.030974 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
11 4.990819 c6:0a:93:05:2a:2e 76:a6:e6:2c:31:92 ARP 42 Who has 100.0.0.11? Tell 100.0.0.12
12 4.990904 76:a6:e6:2c:31:92 c6:0a:93:05:2a:2e ARP 42 100.0.0.11 is at 76:a6:e6:2c:31:92
13 5.049018 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
14 5.050152 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
15 6.059937 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
16 6.061161 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
17 7.079207 100.0.0.12 100.0.0.11 UDP 167 8000 → 8000 Len=125
18 7.080220 100.0.0.11 100.0.0.12 UDP 167 8000 → 8000 Len=125
19 10.032402 76:a6:e6:2c:31:92 c6:0a:93:05:2a:2e ARP 42 Who has 100.0.0.12? Tell 100.0.0.11
20 10.032474 c6:0a:93:05:2a:2e 76:a6:e6:2c:31:92 ARP 42 100.0.0.12 is at c6:0a:93:05:2a:2e
Frame 1: 167 bytes on wire (1336 bits), 96 bytes captured (768 bits)
Ethernet II, Src: c6:0a:93:05:2a:2e (c6:0a:93:05:2a:2e), Dst: 76:a6:e6:2c:31:92 (76:a6:e6:2c:31:92)
Internet Protocol Version 4, Src: 100.0.0.12, Dst: 100.0.0.11
User Datagram Protocol, Src Port: 8000, Dst Port: 8000
Data (54 bytes)
0000 76 a6 e6 2c 31 92 c6 0a 93 05 2a 2e 08 00 45 00
0010 00 99 00 00 40 00 40 11 72 3d 64 00 00 0c 64 00
0020 00 0b 1f 40 1f 40 00 85 ad f4 30 d2 70 3f 8e 67
0030 aa 5d 17 ff ae b4 0d 2b 7b 23 14 91 9e b6 07 e4
0040 d9 ad da 21 6b 7f ae 75 e5 a4 3f 46 a1 c3 1a b1
0050 3f cb 3d 68 c1 12 c1 c4 83 d2 4a a5 1a 08 17 6d
capture3.pcap
Paquets: 20 · Affichés: 20 (100.0%)
Profile: Default

SUR LE PC1


Applications Emplacements Wireshark ven. 16:18
capture1.pcap
Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
1 0.000000 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=1/256, ttl=62 (reply in 2)
2 0.000280 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=1/256, ttl=64 (request in 1)
3 0.998788 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=2/512, ttl=62 (reply in 4)
4 0.998802 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=2/512, ttl=64 (request in 3)
5 2.009526 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=3/768, ttl=62 (reply in 6)
6 2.009566 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=3/768, ttl=64 (request in 5)
7 3.020320 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=4/1024, ttl=62 (reply in 8)
8 3.020373 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=4/1024, ttl=64 (request in 7)
9 4.029746 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=5/1280, ttl=62 (reply in 10)
10 4.029802 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=5/1280, ttl=64 (request in 9)
11 4.995015 9a:33:43:aa:05:f2 6e:5f:98:37:0c:07 ARP 42 Who has 192.168.1.10? Tell 192.168.1.1
12 4.995037 6e:5f:98:37:0c:07 9a:33:43:aa:05:f2 ARP 42 192.168.1.10 is at 6e:5f:98:37:0c:07
13 5.049217 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=6/1536, ttl=62 (reply in 14)
14 5.049245 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=6/1536, ttl=64 (request in 13)
15 6.060127 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=7/1792, ttl=62 (reply in 16)
16 6.060180 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=7/1792, ttl=64 (request in 15)
17 7.079258 192.168.2.10 192.168.1.10 ICMP 98 Echo (ping) request id=0x0502, seq=8/2048, ttl=62 (reply in 18)
18 7.079280 192.168.1.10 192.168.2.10 ICMP 98 Echo (ping) reply id=0x0502, seq=8/2048, ttl=64 (request in 17)
19 10.046053 6e:5f:98:37:0c:07 9a:33:43:aa:05:f2 ARP 42 Who has 192.168.1.1? Tell 192.168.1.10
20 10.046919 9a:33:43:aa:05:f2 6e:5f:98:37:0c:07 ARP 42 192.168.1.1 is at 9a:33:43:aa:05:f2
Frame 1: 98 bytes on wire (784 bits), 96 bytes captured (768 bits)
Ethernet II, Src: 9a:33:43:aa:05:f2 (9a:33:43:aa:05:f2), Dst: 6e:5f:98:37:0c:07 (6e:5f:98:37:0c:07)
Internet Protocol Version 4, Src: 192.168.2.10, Dst: 192.168.1.10
Internet Control Message Protocol
0000 6e 5f 98 37 0c 07 9a 33 43 aa 05 f2 08 00 45 00
0010 00 54 00 00 40 00 3e 01 b8 44 c0 a8 02 0a c0 a8
0020 01 0a 08 00 e9 20 05 02 00 01 44 4c 79 5c 5d 30
0030 04 00 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35
capture1.pcap
Paquets: 20 · Affichés: 20 (100.0%)
Profile: Default