



**POLYTECHNIQUE
MONTRÉAL**

UNIVERSITÉ
D'INGÉNIERIE

Département d'informatique

INF3405

Réseaux Informatique

TP3

Section de laboratoire < **05** >

Soumis à Liliane-Caroline Demers

<Maximiliano Falicoff, Corentin Glaus, Anthony Marcelo Guzman Soto>

<2013658, 2021624, 2019666>

Remis le 21 Avril 2021 (Session H21)

8. A

1. tcp.port==5000 && ip.src== 192.168.11.137 && ip.dst ==192.168.11.138
2. Lors du click droit pour faire follow packet, on a l'option Follow TCP Stream, on utilise ici le protocole TCP pour la couche 4.
3. Client vers serveur:

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	96	19	19,792%	0	0,000%
Between first and last packet	83,231 sec	40,059 sec			
Avg. packets/sec	1,153	0,474			
Avg. packet size	273 bytes	367 bytes			
Bytes	26234	6967	26,557%	0	0.000%
Avg. bytes/sec	315,195	173,916			
Avg. MBit/sec	0,003	0,001			

Client vers serveur:

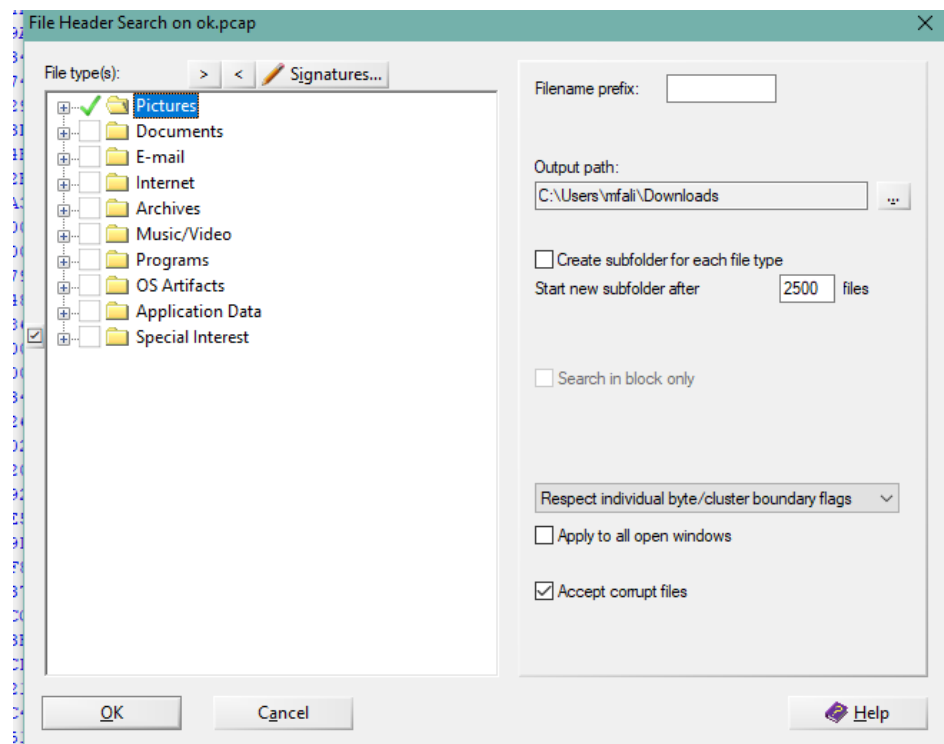
Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	96	16	16,667%	0	0,000%
Between first and last packet	83,231 sec	66,317 sec			
Avg. packets/sec	1,153	0,241			
Avg. packet size	273 bytes	801 bytes			
Bytes	26234	12810	48,830%	0	0.000%
Avg. bytes/sec	315,195	193,164			
Avg. MBit/sec	0,003	0,002			

4. On observe que dans transmission serveur -> client on a un paquet ayant un length de 5880. Dû au fait que le fichier est de taille assez large, la machine serveur va segmenter le fichier et l'envoyer en plus petits morceaux.

```
. V. BZG.  
.!...:.....ls  
2  
getFilesList  
[FILE] polyImage.jpg-  
download polyImage.jpg  
1  
downladingFile  
polyImage.jpg  
rdy  
11604
```

- 5.
6. Nous avons obtenu les données des paquets entre les communications Client-Serveur sous le format RAW. On l'a sauvegardé sous la forme pcap. Cela nous permet de le rentrer dans Winhex. On sait que l'entête d'un fichier jpeg est FF D8 FF E0

et se termine par FF D9. On enlève les segments avant et après.
On peut maintenant l'extraire avec l'outil recover by File Type
(picture) de Winhex



7. Nous pouvons conclure que la sécurité n'est pas adéquate puisque n'importe qui est connecté sur le serveur pour avoir accès à nos images privées.

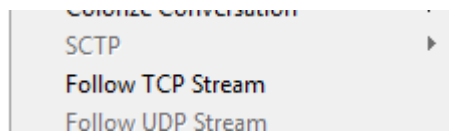
9.

A.

B.

Mode 1

1. Le premier échange de TCP est le SYN qui sert à synchroniser le SN (Sequence Number) et demande l'établissement d'une connexion.



2. Port source: 49842, port destination: 50000

Transmission Control Protocol, Src Port: 49842 (49842), Dst Port: 5000 (5000), Seq: 0, Len: 0

3. client -> serveur

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	37	7	18,919%	0	0,000%
Between first and last packet	31,032 sec	0,008 sec			
Avg. packets/sec	1,192	892,731			
Avg. packet size	324 bytes	630 bytes			
Bytes	11990	4408	36,764%	0	0.000%
Avg. bytes/sec	386,374	562165,289			
Avg. MBit/sec	0,003	4,497			

serveur -> client

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	37	4	10,811%	0	0,000%
Between first and last packet	31,032 sec	0,007 sec			
Avg. packets/sec	1,192	533,694			
Avg. packet size	324 bytes	57 bytes			
Bytes	11990	228	1,902%	0	0.000%
Avg. bytes/sec	386,374	30420,579			
Avg. MBit/sec	0,003	0,243			

Il y a 7 paquets envoyés du client vers le serveur avec 4408 octets envoyés et 4 paquets envoyés du serveur vers le client avec 228 octets envoyés.

4. Le Client a envoyé 3 paquets pour transmettre les données.

[illegible]

Mode 2

Protocol
TCP
TCP
TCP

1. Le premier échange de TCP est le SYN TCP
2. Port source: 49842, port destination: 50000

Transmission Control Protocol, Src Port: 49842 (49842), Dst Port: 5000 (5000), Seq: 0, Len: 0

3. client -> serveur

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	243	104	42,798%	0	0,000%
Between first and last packet	28,551 sec	0,007 sec			
Avg. packets/sec	8,511	14268,207			
Avg. packet size	164 bytes	93 bytes			
Bytes	39891	9646	24,181%	0	0.000%
Avg. bytes/sec	1397,175	1323376,174			
Avg. MBit/sec	0,011	10,587			

serveur -> client

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	243	47	19,342%	0	0,000%
Between first and last packet	28,551 sec	0,006 sec			
Avg. packets/sec	8,511	7263,265			
Avg. packet size	164 bytes	54 bytes			
Bytes	39891	2550	6,392%	0	0.000%
Avg. bytes/sec	1397,175	394070,786			
Avg. MBit/sec	0,011	3,153			

Il y a 104 paquets envoyés du client vers le serveur avec 9646 octets envoyés et 47 paquets envoyés du serveur vers le client avec 2550 octets envoyés.

[illegible]

Mode 3

[illegible]

A diagram of a packet header. It consists of two stacked rectangular boxes. The top box is light blue and contains the text 'UDP' in white. The bottom box is dark blue and contains the text 'ICMP' in light blue.

1. **ICMP** On utilise UDP pour ce mode. On a pas de SYN mais on observe l'apparence d'un paquet ICMP, cela est dû au fait qu'il fonctionne comme un PING
2. Port source: 50251, port destination: 5010

Source port: 50251 Destination port: 5010

3. client -> serveur 443,180 | serveur -> client

4. Le Client a envoyé 1 paquet pour transmettre les données.

Mode 4

Protocol
UDP
ICMP
UDP
UDP
UDP
UDP
UDP
UDP
UDP
UDP
UDP
UDP
UDP

1. On utilise UDP pour ce mode. On a pas de SYN mais on observe l'apparence d'un paquet ICMP, cela est dû au fait qu'il fonctionne comme un PING.
2. Port source: 56233, port destination: 5010

source port: 56233 Destination port: 5010

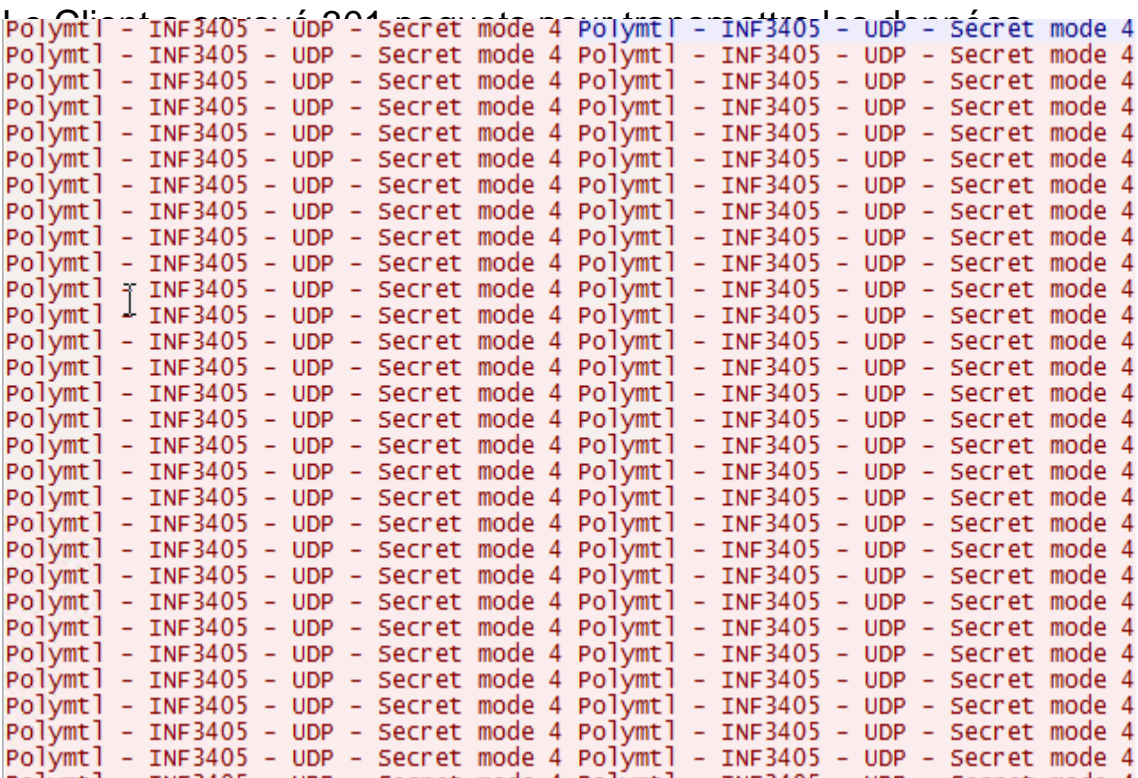
3. client -> serveur

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	335	301	89,851%	0	0,000%
Between first and last packet	19,297 sec	0,015 sec			
Avg. packets/sec	17,360	20054,732			
Avg. packet size	80 bytes	82 bytes			
Bytes	26838	24710	92,071%	0	0,000%
Avg. bytes/sec	1390,793	1646353,600			
Avg. MBit/sec	0,011	13,171			

serveur -> client

Traffic	Captured	Displayed	Displayed %	Marked	Marked %
Packets	335	1	0,299%	0	0,000%
Between first and last packet	19,297 sec				
Avg. packets/sec	17,360				
Avg. packet size	80 bytes				
Bytes	26838	110	0,410%	0	0,000%
Avg. bytes/sec	1390,793				
Avg. MBit/sec	0,011				

Il y a 301 paquets envoyés du client vers le serveur avec 24710 octets envoyés et 1 paquet envoyé du serveur vers le client avec 110 octets envoyés.

4. 

C.

1. Le mode 2 est plus rapide en moyenne que le mode 1. Cela peut être dû au Tcp slow start. En effet, le TCP augmente graduellement sa vitesse. Cela explique pourquoi le mode 1 qui a 7 paquets à une vitesse moyenne plus lente que le mode 2 qui a 104 paquets.
2. Le mode 3 utilise le protocole IPv4 alors que le mode 4 utilise UDP. Le mode 3 (443,180 Mbits) est plus rapide que le mode 4 (13,171 Mbits). De plus, le mode 3 est fragmenté ce qui fait en sorte qu'il aille plus vite que l'autre puisqu'on peut envoyer plus d'information en même temps. Aussi, le modèle 3 va utiliser moins de data pour la somme des headers que celui du modèle 4. Cela fait que nous allons utiliser moins de temps pour lire du data qui n'est pas utile.
3. Dû au fait que UDP n'offre pas d'accusé de réception, TCP est plus fiable au niveau
4. Le dernier message a [FIN, ACK], car le client est notifié de la réception de son dernier paquet et envoie fin, car le client veut cesser ses communications avec le serveur.