TP3 - Wireshark

1 Analyse de Trames avec Wireshark

1.1 Préambule

université

BORDEAUX

Voici les informations utiles sur la machine qui a servi à capturer ces traces réseaux. Ces informations sont obtenues avec les commandes Linux suivantes :

root@mydebian:~\$ /sbin/ifconfig

eth0: fla	gs=4163 net 10.	S <up,broal< th=""><th>DCAST,RU netmask</th><th>NNING, 255.255.</th><th>MULTIC</th><th>CAST> t</th><th>mtu proadca</th><th>1500 ast 10.0</th><th>0.2.2</th><th>55</th><th></th></up,broal<>	DCAST,RU netmask	NNING, 255.255.	MULTIC	CAST> t	mtu proadca	1500 ast 10.0	0.2.2	55	
B	X nacke	1.04.00.12	2.34.30 vtes 804	.3 (7 8 k	(iR)	1000		lernet)			
R. T.	X erron X packe	rs 0 drop ets 63 by	pped 0 ytes 102	overruns 204 (9.9	S O KiB)	fram	le O				
T	X error	rs 0 drop	pped 0 c	verruns	0 0	carri	er O	collis	ions	0	
lo: flags	=73 <up,< td=""><td>LOOPBACK</td><td>,RUNNING</td><td>> mtu 6</td><td>65536</td><td></td><td></td><td></td><td></td><td></td><td></td></up,<>	LOOPBACK	,RUNNING	> mtu 6	65536						
i	net 127	.0.0.1 r	netmask	255.0.0.	0						
1	oop ty	queuelen	1 (Loc	al Loopt	back)						
R	X packe	ets 0 byt	tes 0 (C).O B)							
R	X error	rs 0 drop	oped 0	overruns	s 0	fram	le 0				
T	X packe	ets 0 byt	tes 0 (C).O B)							
T	X erron	rs 0 drop	pped 0 c	verruns	0 0	carri	er O	collis	ions	0	
root@myde	bian:~\$	/sbin/ro	oute -n								
Kernel IP	routir	lg table									
Destinati	on	Gateway		Genmask			Flags	Metric	Ref	Use) Iface
0.0.0.0		10.0.2.2		0.0.0.0			UG	0	0	C) eth0
10.0.2.0		0.0.0.0		255.255.	255.0		U	0	0	C) eth0
root@myde	bian:~\$	cat /eto	c/resolv	.conf							

nameserver 10.0.2.3

Vous devez reconnaître l'adresse IP de la machine utilisée (et son masque de réseau), l'adresse IP de la passerelle vers Internet (gateway), ainsi que l'adresse du serveur DNS local...

université BORDEAUX

					traffic	pcap _ 🗆 ×
<u>File</u> <u>E</u> dit	<u>V</u> iew <u>G</u> o	<u>Capture</u> <u>A</u> nalyze <u>S</u> tatist	ics Telephon <u>y W</u> ireless	<u>T</u> ools <u>H</u> elp	D	
	101	📮 🚺 🗶 🚺	« » °\$ IS »I			<u>₽</u>
Apply a	display filter .	<ctrl-></ctrl->				Expression +
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	08:00:27:0b:03:37 52:54:00:12:35:02	ff:ff:ff:ff:ff:ff:ff	ARP	42 60	Who has 10.0.2.2? Tell 10.0.2.15 10.0.2.2 is at 52:54:00:12:35:02
3	0.000374	10.0.2.15	193.50.111.150	DNS	73	Standard query 0xa7eb A www.perdu.com
4	0.000583	10.0.2.15 193.50.111.150	193.50.111.150	DNS	73 201	Standard query 0x8103 AAAA www.perdu.com A 208.97.177.124 NS ns3.dreambost.com NS
6	0.156191	193.50.111.150	10.0.2.15	DNS	134	Standard query response 0x8f03 AAAA www.perdu.com SOA ns1.dreamhost.com
Γ ⁷	0.159073	10.0.2.15 208 97 177 124	208.97.177.124	TCP	74	38196 → 80 [SYN] Seq=0 W1n=29200 Len=0 MSS=1460 SACK_PERM=1 TSva1=4090081 TSecr=0 WS=128 80 → 38196 [SYN_ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460
9	0.261621	10.0.2.15	208.97.177.124	TCP	54	38196 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
+ 10	0.262996	10.0.2.15	208.97.177.124	HTTP TCP	194	GET / HTTP/1.1 80 - 38196 [ACK] Sen=1 Ack=141 Win=65535 Len=0
÷ 12	0.366463	208.97.177.124	10.0.2.15	HTTP	549	HTTP/1.1 200 OK (text/html)
13	0.366517	10.0.2.15	208.97.177.124	TCP	54	38196 → 80 [ACK] Seq=141 Ack=496 Win=30016 Len=0 20106 → 80 [ETN ACK] Seq=141 Ack=406 Win=20016 Len=0
15	0.383833	208.97.177.124	10.0.2.15	TCP	60	80 → 38196 [ACK] Seq=496 Ack=142 Win=65535 Len=0
16	0.485155	208.97.177.124	10.0.2.15	TCP	60 54	80 → 38196 [FIN, ACK] Seq=496 Ack=142 Win=65535 Len=0
- 1/	0.403213	10.0.2.15	200.01.111.124	TOP	54	30130 - 00 [AOK] 364-142 ACK-437 WIN-50010 E81-0
4						
Frame :	10: 194 bytes	s on wire (1552 bits).	194 bytes captured (15	52 bits)		
Ethern	et II, Src: 0	08:00:27:0b:03:37, Dst:	52:54:00:12:35:02			
 Intern Transm. 	et Protocol v ission Contro	/ersion 4, Src: 10.0.2. Dl Protocol, Src Port:	15, Dst: 208.97.177.12 38196, Dst Port: 80, S	4 eq: 1, Ack:	1, Len: 140	
 Hypert 	ext Transfer	Protocol				
0000 52	54 00 12 35	02 08 00 27 0b 03 37 0	38 00 45 00 RT5	'7E.		
0010 b1	7c 95 34 00	50 78 5c d2 80 00 21 3	34 02 50 18 . .4.Px\	!4.P.		
0030 72	10 8e 93 00	00 47 45 54 20 2f 20 4	48 54 54 50 rGE	T / HTTP		
0040 21 0050 3a	20 57 67 65	74 2f 31 2e 31 39 2e 3	32 20 28 6c : Wget/1	.19.2 (1		
0060 69	6e 75 78 2d	67 6e 75 29 0d 0a 41	63 63 65 70 inux-gnu)Accep		
0080 6e	3a 20 2a 21 63 6f 64 69	2a 00 0a 01 03 03 05 6e 67 3a 20 69 64 65 0	70 74 20 65 t: "/" Se 74 69 74 ncoding:	identit		
0090 79	0d 0a 48 6f	73 74 3a 20 77 77 77 7	2e 70 65 72 yHost:	www.per		
00a0 64	75 2e 63 6f 6e 3a 20 4b	65 65 70 2d 41 6c 69	55 63 74 69 du.com 76 65 0d 0a on: Keen	-Alive		
00c0 0d	0a					
🔵 🍸 t	raffic					Packets: 17 · Displayed: 17 (100.0%) · Load time: 0:0.0 Profile: Default

1.2 Prise en main de Wireshark

A l'aide de l'outil Wireshark, ouvrez le fichier ping.pcap.

On voit apparaître dans la partie du haut une ligne pour chaque trame Ethernet que la carte réseau de la machine a traitée (en émission comme en réception). Ici on voit donc une séquence de trames avec un petit résumé qui indique la source, la destination, le protocole et quelques infos supplémentaires.

Lorsqu'une trame est sélectionnée, le contenu brut (i.e. octet par octet) apparaît dans la partie du bas, et une version décodée apparaît dans la partie du milieu. Il est alors possible d'inspecter cette trame en profondeur, couche par couche, en affichant à chaque niveau tous les détails sur le protocole utilisé. Cliquez sur \triangleright pour explorer un niveau.

1.3 Ping

La commande *ping* disponible sur toutes les plateformes permet de tester rapidement si une machine est joignable sur le réseau Internet à partir de son nom ou de son IP... Par exemple :

```
$ ping -4 -n www.google.com
```

```
PING www.google.com (172.217.19.132) 56(84) bytes of data.
64 bytes from 172.217.19.132: icmp_seq=1 ttl=63 time=15.4 ms
64 bytes from 172.217.19.132: icmp_seq=2 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=3 ttl=63 time=15.7 ms
64 bytes from 172.217.19.132: icmp_seq=4 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=5 ttl=63 time=15.8 ms
64 bytes from 172.217.19.132: icmp_seq=6 ttl=63 time=15.7 ms
64 bytes from 172.217.19.132: icmp_seq=6 ttl=63 time=15.7 ms
64 bytes from 172.217.19.132: icmp_seq=6 ttl=63 time=15.7 ms
```

```
--- www.google.com ping statistics ---
```

université [®]BORDEAUX

7 packets transmitted, 7 received, 0% packet loss, time 26ms rtt min/avg/max/mdev = $15.442/15.722/15.847/0.207~{\rm ms}$

Par défaut, la commande *ping* envoie une requête echo-request (protocole ICMP) chaque seconde et affiche en retour la réponse echo-reply avec quelques statistiques... Notamment, le temps affiché correspond au temps d'aller-retour du message (RTT = Round-Time Trip). On considère souvent que la Latence représente la moitié du RTT.

La trace enregistrée se divise en trois étapes principales identifiées avec des couleurs différentes dans Wireshark :

- 1. trames 1-2 (ARP) : on cherche à découvrir l'adresse Ethernet du serveur DNS (situé dans notre réseau local);
- 2. trames 3-4 (DNS) : on cherche à trouver l'adresse IP de la machine cible (www.google.com);
- 3. trames 5-6 (ARP) : on cherche à découvrir l'adresse Ethernet de la passerelle, afin de connaître la porte de sortie du réseau local vers Internet;
- 4. trames 7-16 (ICMP) : 5 pings et leurs réponses...

Répondez aux questions suivantes concernant la trace ping.pcap en vous aidant de Wireshark.

- A quelle adresse Ethernet est destinée la requête ARP (trame 1) émise par la machine cliente? Il s'agit en fait de l'adresse de diffusion (broadcast). Elle ne correspond à aucune machine particulière! A votre avis pourquoi doit on procéder ainsi?
- Quel est le protocole de transport utilisé pour les échanges DNS (trames 3-4)?
 Observez en détail la réponse DNS (section Answers) et découvrez ainsi l'adresse
 IP de la machine www.google.com retourné par le serveur DNS.
- La requête ARP WHO HAS (trame 5) cherche à trouver l'adresse Ethernet de la machine 10.0.2.2. Pourquoi cette machine et non pas la machine cible www.google.com ? Vérifiez l'adresse Ethernet destination utilisée pour envoyer la trame 7.
- Observez la première requête / réponse ICMP (trames 7-8) et observez la valeur du champs type dans l'en-tête ICMP...

1.4 Une page Web : je suis perdu !

Que se passe-t-il quand je consulte une page web (par exemple, http://www.perdu. com) sur Internet avec mon navigateur préféré?

La trace enregistrée http.pcap effectue les mêmes étapes préliminaires que dans la trace précédente : requêtes ARP et DNS... Ouvrez-la dans Wireshark, et concentrons nous ici sur la conversation TCP/IP (trames 7-16). On demande la page d'accueil "/" ou "/index.html" du serveur web (www.perdu.com) en utilisant le protocole HTTP au dessus de TCP, ce qui implique plusieurs étapes intermédiaires :

- 1. la connexion TCP en trois temps (trames 7-9);
- 2. la requête HTTP ainsi que la réponse incluant le code HTML (trames 10-13);
- 3. la phase de déconnexion (trames 14-16).

Répondez aux questions suivantes concernant la trace en vous aidant de Wireshark.

université BORDEAUX

- Considérons la première trame TCP qui ouvre la connexion (trame 7). Trouvez dans l'en-tête TCP le port source et le port de destination. Ce dernier est standard pour tous les serveur web (80). A quoi correspond le flag SYN dans cette en-tête?
- Identifiez dans la conversation TCP les trames correspondant à la requête HTTP et à la réponse HTTP...
- Dans l'en-tête de la requête HTTP, on observe sur la première ligne qu'il s'agit de la requête GET / HTTP/1.1. Identifiez le rôle des champs suivants : User-Agent, Host, Connection.
- La réponse HTTP commence par la ligne suivante "HTTP/1.1 200 OK" qui indique que tout s'est bien passé (code 200). Vous devez déjà connaître le fameux code d'erreur 404 (cf. liste des codes HTTP).
- Observez maintenant les différents champs dans la réponse HTTP et en déduire le logiciel serveur, la longueur et le type de contenu dans cette réponse.
- Immédiatement après l'en-tête HTTP, vous pouvez identifier le code HTML de la page web : <html>...</html>.
- Trames 7-16 : Pour lire plus facilement la conversation TCP, vous pouvez faire un "clic droit" sur un des paquets TCP et sélectionner *Suivre (Follow)* \rightarrow *flux TCP (TCP Stream)* dans le menu déroulant. Notez qu'il est possible de reconstruire précisément le fil de la conversation grâce aux numéros de séquence (en octets) qui se trouve dans l'en-tête TCP.