Architectures Réseaux Télécoms	Manuel : Analyseur réseau Wireshark	Création : 28/12/2006 15:20:00 Mise à jour : 04/05/200707 09:52:00 AM
Auteur : David ROUMANET		

1.SOMMAIRE

1. SOMMAIRE	1
2. OBJET	2
3. CAPTURE	3
 3.1 Edition des préférences	3
 4.1 LECTURE DES TRAMES. 4.1.1 Fenêtre de résumé. 4.1.2 Fenêtre d'arborescence de protocole. 4.1.3 Fenêtre de vue des données. 4.2 ANALYSE RAPIDE. 4.2.1 Expert Info Composite	8 8 8 9 9 9 9 11 11 11 11 11 12 13

2.OBJET

Le logiciel Wireshark (anciennement Ethereal) permet la capture et l'analyse de trames sur Ethernet.

Son utilité est indéniable pour contrôler le bon fonctionnement de réseau ou vérifier les trames transitant sur une interface d'un commutateur ou analyser les trafics inutiles ou ceux impactant les performances du réseau.

Voici un petit récapitulatif des capacités de Wireshark :

- Décoder les trames (niveau 2 et 3)
- Calculer le débit moyen sur la durée de la capture (Mbps)
- Tracer un graphe du trafic pour tout ou partie des flux capturés
- Afficher les temps de réponses des trames TCP (basé sur les acquittements)
- Indiquer les erreurs ou les alertes détectées (paquets perdus, retransmis, dupliqués...)
- Suivre un dialogue TCP (notamment HTTP)
- Donner les statistiques sur les tailles des trames réseaux
- Etc.

Pour cela, il suffit de l'installer sur un PC munit d'une interface réseau 100 Mbps ou plus et fonctionnant sous Windows ou Linux. La version décrite ici fonctionne sous Windows XP. Il s'agit de la version 0.99.4.

Récupérer la dernière version du programme sur le site http://www.wireshark.org/

Suivre les instructions d'installation (en particulier l'installation de WinPCAP s'il n'est pas déjà installé sur le poste).

Une fois l'installation terminé, le **poste** devient une **sonde** réseau prête à fonctionner.

3.CAPTURE

La première opération est la capture de trames. Cependant il y a plusieurs cas possibles :

- Capture en temps réel de manière manuelle (l'utilisateur démarre et arrête la capture)
- Capture avec limitations automatiques (dans le temps ou sur la taille...)
- Capture sur une période donnée avec rotation (utilisation d'un « buffer » circulaire)

D'autre part, il peut-être utile de limiter les données capturées à celles qui sont en cause lors de l'analyse :

- Filtrage par protocoles
- Filtrage par adresses
- Limitation de la taille des paquets capturés

3.1Edition des préférences

L'édition des préférences permet de choisir l'apparence de Wireshark mais aussi de choisir l'interface de capture à utiliser :



🖪 Wireshark: Preferences		
🗉 User Interface	Capture	
Layout	Default interface:	\NPF_{E5E80F62-2D97-41F8-9382-F3D64A6B5AB9}
Columns	T-b-f	Generic dialup adapter: \Device\NPF_GenericDialupAdapt
Font	Interfaces:	Intel(R) PRO/100 VE Network Connection (Microsoft's Pa
Colors	Capture packets in promiscuous mode:	AT&T (Microsoft's Packet Scheduler) : \Device\NPF_{325,
Pripting		<
Name Resolution	Opdate list of packets in real time:	
Protocols	Automatic scrolling in live capture:	
	Hide capture info dialog:	
		QK Apply Cancel

Une fois les préférences modifiées, il est possible de procéder à notre première capture.

3.2Choisir les paramètres de capture

Pour faire une **capture manuelle**, il suffit de cliquer sur l'icône "Start a new live capture" ou dans le menu, choisir [Capture] [Start]... il suffira ensuite d'arrêter la capture en cliquant sur l'icône à sa droite.



Cette option est intéressante pour tester le trafic et déterminer la quantité d'informations passant sur l'interface, cependant il est préférable de faire une capture automatisé.



Pour faire une **capture automatisée**, il faut cliquer sur l'icône "Show capture options…" ou dans le menu, choisir [Capture] [Options…] ou encore utiliser le raccourci-clavier [CTRL+K]



Par défaut, l'interface présente la plupart des options en grisées car elles ne sont pas actives :

📶 Wireshark: Capture Options 📃 🗆 🔯										
Capture										
Interface: Intel(R) PRO/100 VE Network Connection (Microsoft's Packet Scheduler) : \Device\NP										
IP address: 139.160.140.83										
Link-layer header type: Ethernet 💉 Buffer size: 1 👘 megabyte(s) Wireless Settings										
🔽 Capture pack	Capture packets in promiscuous mode									
📃 Limit each pao	Limit each packet to 68 🗘 bytes									
Capture Filter:			•							
Capture File(s)			Display Options							
File:		Brows	se							
Use <u>m</u> ultiple f	iles									
Next file ever	у 1	megabyte(s)	Automatic scroling in live capture							
Next file ever	у 1	minute(s)	Hide capture info dialog							
💌 Ring buffer w	ith 2	🗘 files	-Name Resolution							
Stop capture	after 1	🔷 file(s)								
Stop Capture			Enable MAC name resolution							
🔲 after	1	packet(s)	Enable network name resolution							
🔲 after	1	megabyte(s)								
🔲 after	1	minute(s)	Enable transport name resolution							
Help			Start Cancel							

Il est primordial que la capture se fasse en mode « promiscuous ». D'autre part, si le poste n'est pas très puissant, il est préférable de désactiver la case « Update list of packets in real time ».

3.2.1 Limitation de la taille des paquets

L'analyse des trames se faisant généralement sur les premiers octets (les entêtes), il est utile de limiter la taille des paquets capturés à une taille maximum : pour cela, il suffit de cocher la case « Limit each packet to » et de choisir un nombre entre 68 octets (entête TCP) et 132 (informations complémentaires pour des flux HTTP ou TNS par exemple).

Cela n'a aucune influence sur les statistiques concernant les tailles de trames puisque cette information est inscrite dans l'entête des trames Ethernet.

3.2.2Arrêt automatique sur seuil

Il est possible de limiter la capture sur 3 critères : nombre de paquets, taille de la capture et délai dans le temps. Ces trois critères peuvent être combinés. Cet arrêt automatique permet de limiter le travail d'analyse plus tard et de ne pas écraser un événement important.

3.2.3Captures circulaires

C'est le mode le plus intéressant, surtout si la sonde dispose d'un espace disque suffisant. En effet, les problèmes réseaux sont souvents fugitifs et lorsque un incident survient, le temps d'activer une capture ne permet pas de trouver l'origine du problème. D'un autre coté, une capture

linéaire permet de remonter dans l'historique des trames capturées mais la manipulation d'un fichier unique et souvent de taille imposante et difficile. La capture circulaire résout ces problèmes :

Capture	File(s)									
File: D	D:\Audits\test Browse									
Use multiple files										
📃 Nex	t file every	1	\$	megabyte(s) 💉						
🗹 Nex	t file every	1	-	hour(s) 🗸 🗸						
🗹 Ring) buffer with	24	-	files						
📃 Stop	o capture after	1	\$	file(s)						

Dans l'exemple ci-dessus, Wireshark va créer 24 fichiers contenant chacun une heure de capture. Une fois la 24^{ème} heure écoulée, Wireshark va supprimer le premier fichier de la liste et va créer un nouveau fichier.



Avantages :

- Limiter le risque de dépassement de taille de disque,
- Conserver un historique sur 24 heures,
- Permettre la copie des fichiers intermédiaires (sauvegarde ou analyse sur autre poste),
- Localiser facilement un événement dans l'ensemble des fichiers.

Attention : la quantité de données capturées pouvant être très importante, il peut-être préférable de limiter chaque fichier à une taille comprise entre 2Mo et 10Mo afin de faciliter le travail d'analyse. En effet, l'utilisation des outils de Wireshark peut prendre beaucoup de temps sur un poste aux capacités limitées.

L'astuce pour déterminer le bon nombre de fichier pour effectuer la rotation est d'effectuer une première capture manuelle pour chronométrer combien de temps il faut pour remplir la taille choisie.

3.2.4Filtres de capture

Si le flux à surveiller est bien identifié (serveur, plage réseau, numéros de ports), il est possible de n'enregistrer que les trames qui lui correspondent. Pour cela, Wireshark permet d'appliquer un filtre sur les paquets à enregistrer.

📶 Wireshark: Capture Filter 📃 🗆 🔀										
Edit	Filter									
	Ethernet type 0x0806 (ARP)									
	No Broadcast and no Multicast									
New	No ARP									
<u> <u> </u></u>	IP only									
	IP address 192.168.0.1									
	IPX only									
	TCP only									
	UDP only									
	TCP or UDP port 80 (HTTP)									
Delete	HTTP TCP port (80)									
	No ARP and no DNS									
	Non-HTTP and non-SMTP to/from www.wireshark.org									
	Suppression OSPF/HSRP & STP	2								
Properties										
Filter name	: Suppression OSPF/HSRP & STP									
Filter string: not stp and not net 224.0.0.0 mask 255.255.255.240										
Help										

La syntaxe du filtre de capture est accessible en cliquant sur le bouton [Help]. Ce filtre permet de combiner plusieurs conditions (and et or) ainsi que d'inverser les filtres (not).

3.3Statistiques instantanées de capture

Une fois tous les paramètres de capture définis, la capture démarre. Wireshark affiche une boite de dialogue qui indique en temps réel la répartition des protocoles

🕜 Wireshark	: Capture from	m Intel(R) PRO/1.	– 🗆 🛛				
Captured Pack	kets						
Total	213	% of total					
SCTP	0		0,0%				
тср	47		22,1%				
UDP	61		28,6%				
ICMP	12		5,6%				
ARP	60		28,2%				
OSPF	6		2,8%				
GRE	0		0,0%				
NetBIOS	0		0,0%				
IPX	8		3,8%				
VINES	0		0,0%				
Other	19		8,9%				
Running	00:00:30						
Stop							

4.ANALYSES

Une fois les captures effectuées, il est possible de faire le travail d'analyse. C'est la partie la plus complexe mais si les options de captures ont été judicieusement utilisées, ce travail ne sera pas trop long.

4.1Lecture des trames

L'affichage de Wireshark se décompose en fenêtre qu'il est possible de redimensionner :

4.1.1Fenêtre de résumé

Dans cette fenêtre, Wireshark affiche un résumé des informations : adresses (niveau 3 ou par défaut niveau 2), estampillage horaire, protocole et description succinte. La coloration permet de retrouver rapidement certains protocoles (broadcast, requêtes ARP, etc.) et elle est personnalisable dans le menu [View] [Coloring Rules...].

No. +		Time				Source	Destination	Protocol	Info	^
19	557	2006-11-29	19:33	1:55	793332	00:09:6b:b0:da:be	ff:ff:ff:ff:ff	Intel A	Sequence: 3351399424, Sender ID 256, Tear	_
15	558	2006-11-29	19:33	1:55.	.835471	00:0b:db:8d:7d:11	ff:ff:ff:ff:ff:ff	ARP	Who has 10.196.22.67? Tell 10.196.23.23	=
15	559	2006-11-29	19:33	1:56	.196238	00:80:f4:00:64:07	00:80:f4:00:65:20	LLC	U, func=UI; DSAP 0x24 Individual, SSAP 0>	
15	5.60	2006-11-29	19:33	1:56	.851720	00000000.000400222ed9	00000000.ffffffffffffff	IPX SAP	General Response	
15	561	2006-11-29	19:33	1:56	.852212	00000000.000400222ed9	00000000.ffffffffffffff	IPX SAP	General Response	
15	562	2006-11-29	19:33	1:56	.852648	00000000.000400222ed9	00000000.ffffffffffffff	IPX SAP	General Response	
15	563	2006-11-29	19:33	1:56	.853059	00000000.000400222ed9	00000000.ffffffffffffff	IPX SAP	General Response	
15	564	2006-11-29	19:33	1:56	.862636	00:09:6b:b0:da:be	ff:ff:ff:ff:ff:ff	Intel A	Sequence: 3368176640, Sender ID 256, Tear	
15	565	2006-11-29	19:33	1:57	.317178	00:80:f4:00:64:05	00:80:f4:00:03:10	LLC	U, func=UI; DSAP 0x24 Individual, SSAP 0>	
15	566	2006-11-29	19:33	1:57	.925888	00:09:6b:b0:da:be	ff:ff:ff:ff:ff:ff	Intel A	Sequence: 3384953856, Sender ID 256, Tear	
15	567	2006-11-29	19:33	1:57	.943504	10.196.22.86	10.196.22.54	TCP	[TCP Retransmission] 2733 > 502 [PSH, ACH	
15	568	2006-11-29	19:33	1:57	.947756	00:80:f4:00:65:17	ff:ff:ff:ff:ff:ff	ARP	Who has 10.196.22.86? Tell 10.196.22.54	
15	569	2006-11-29	19:33	1:57	.947915	00:16:35:75:0e:4e	00:80:f4:00:65:17	ARP	10.196.22.86 is at 00:16:35:75:0e:4e	
15	570	2006-11-29	19:33	1:57	.951050	10.196.22.54	10.196.22.86	TCP	502 > 2733 [RST] Seq=74468 Len=0	
15	571	2006-11-29	19:33	1:58	.145842	10.196.22.86	10.196.22.54	TCP	2735 > 502 [SYN] Seq=0 Len=0 MSS=1460	
15	572	2006-11-29	19:33	1:58	.149970	10.196.22.54	10.196.22.86	TCP	502 > 2735 [SYN, ACK] Seq=0 Ack=1 Win=409	
15	573	2006-11-29	19:33	1:58	.150164	10.196.22.86	10.196.22.54	TCP	2735 > 502 [ACK] Seg=1 Ack=1 Win=17520 Lt	
15	574	2006-11-29	19:33	1:58	.150412	10.196.22.86	10.196.22.54	TCP	2735 > 502 [PSH, ACK] Seg=1 Ack=1 Win=17!	
15	575	2006-11-29	19:33	1:58	.180560	10.196.22.54	10.196.22.86	TCP	502 > 2735 [PSH, ACK] Seg=1 Ack=29 Win=4(
15	576	2006-11-29	19:33	1:58	.181673	10.196.22.86	10.196.22.54	TCP	2735 > 502 [PSH, ACK] Seg=29 Ack=20 Win=:	_
15	577	2006-11-29	19:33	1:58	231942	10.196.22.54	10.196.22.86	TCP	502 > 2735 [PSH, ACK] Seg=20 Ack=57 Win=4	Y
<							1111			

A partir de cette vue, il est possible de marquer des paquets : menu [Edit] [Mark packet (toggle)] ou séquence clavier [CTRL]+[M]. Cela permet lors d'une sauvegarde ou d'un export de limiter le nombre de trames sauvegardés.

4.1.2Fenêtre d'arborescence de protocole

Cette fenêtre détaille le paquet sélectionné dans la fenêtre de résumé : la trame est décomposée de manière hiérarchique, du plus bas niveau (frame) jusqu'au niveau du protocole le plus élevé connu par Wireshark.

```
    Frame 1 (80 bytes on wire, 80 bytes captured)
    Ethernet II, Src: 00:16:35:75:0e:4e (00:16:35:75:0e:4e), Dst: 00:80:f4:00:65:17 (00:80:f4:00:65:17)
    Destination: 00:80:f4:00:65:17 (00:80:f4:00:65:17)
    Bource: 00:16:35:75:0e:4e (00:16:35:75:0e:4e)
Type: IP (0x0800)
    Internet Protocol, Src: 10.196.22.86 (10.196.22.86), Dst: 10.196.22.54 (10.196.22.54)
    Transmission Control Protocol, Src Port: 2733 (2733), Dst Port: 502 (502), Seq: 0, Ack: 0, Len: 26
Data (26 bytes)
```

4.1.3Fenêtre de vue des données

Cette fenêtre affiche les données brutes : chaque champ sélectionné dans la fenêtre d'arborescence de protocole et indiqué en inverse vidéo dans cette fenêtre. L'inverse est possible aussi. De plus, la barre d'état affiche également le type de donnée sélectionnée.

D000 00 80 f4 00 65 17 00 16 35 75 0e 4e 03 00 45 00 0010 00 42 6a ae 40 00 80 64 f4 0a cf 16 56 0a c4 0020 16 36 0a ad 01 f6 a8 4d 6d fd 00 3e x6 c5 0 a 0020 16 36 0a ad 01 f6 a8 4d 6d fd 00 3e x6 c5 18 0030 42 42 48 57 00 00 00 00 14 00 f1 50 0040 17 00 21 65 39 65 f9 1b 36 07 68 07 90 01 64 00 </th <th>e Su.N<mark></mark>E. .Bj.@ MV. .GM m>.P. BBHWS. !e9e G.hd.</th>	e Su.N <mark></mark> E. .Bj.@ MV. .GM m>.P. BBHWS. !e9e G.hd.
Type (eth.type), 2 bytes	P: 18067 D: 18067 M: 0

4.2Analyse rapide

Pour obtenir rapidement des indications concernant les erreurs dans la capture, il faut utiliser le module expert.

4.2.1Expert Info Composite

Ce module est accessible via le menu [Analyze] [Expert Info Composite]. Il permet une analyse rapide (bien que ce soit l'analyse la plus complexe).

Merpins_panne_19-3159-13.pcap - Wireshark				- 2 🛛
Eile Edit View Go Capture Analyze Statistics	Help			
Display Filters	(e 🖉
Prepare a Filter	-			,
Eilter: Firewall ACL Rule	es 🖉 🖉 Expre	ession <u>⊂</u> lear <u>A</u> pply		
No Time Stabled Protocol:	s Shift+Ctrl+R estination	Protocol	Info	<u>^</u>
1556 2006-11-29 19:31: 🕏 Decode As	D:80:f4:0	0:17:00 LLC	U, func=UI; DSAP 0x24 I	Individual, SSAP 0
1557 2006-11-29 19:31: 28 User Specified De	ecodes F:ff:ff:1	ff:ff:ff Intel A:	Sequence: 3351399424, S Who has 10 196 22 672	Sender ID 256, Tear 🗐
1559 2006-11-29 19:31: Eollow TCP Strea	m D:80:f4:0	00:65:20 LLC	U, func=UI; DSAP 0x24 I	ndividual, SSAP Ox
1560 2006-11-29 19:31: 1561 2006-11-29 19:31: Eollow SSL Stream	m 0000000.1	FTTTTTTTTTTT IPX SAP (FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	General Kesponse General Response	
1562 2006-11-29 19:31: Expert Info	000000.1	FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF	General Response	
1563 2006-11-29 19:31: Expert Info Comp 1564 2006-11-29 19:31:50:002050 00:05	posite	ff:ff:ff Intel A:	General Kesponse Sequence: 3368176640, S	ender ID 256, Tear
1565 2006-11-29 19:31:57.317178 00:80	:f4:00:64:05 00:80:f4:0	0:03:10 LLC	U, func=UI; DSAP 0x24 I	ndividual, SSAP 0>
1567 2006-11-29 19:31:57.925888 00:09	6.22.86 10.196.22.	.54 TCP	[TCP Retransmission] 27	33 > 502 [PSH, ACH
1568 2006-11-29 19:31:57.947756 00:80	:f4:00:65:17 ff:ff:ff:f	ff:ff:ff ARP 1	Who has 10.196.22.86?	Tell 10.196.22.54
1570 2006-11-29 19:31:57.951050 10.19	6.22.54 10.196.22.	.86 TCP !	502 > 2733 [RST] Seq=74	468 Len=0
1571 2006-11-29 19:31:58.145842 10.19	6.22.86 10.196.22. 6.22.54 10.196.22	.54 TCP	2735 > 502 [SYN] Seq=0 502 > 2735 [SYN: ACK] S	Len=O MSS=1460 Seq=O Ack=1 Win=400
1573 2006-11-29 19:31:58.150164 10.19	6.22.86 10.196.22	.54 TCP	2735 > 502 [ACK] Seq=1	Ack=1 Win=17520 Le
1574 2006-11-29 19:31:58.150412 10.19	6.22.86 10.196.22. 6.22.54 10.196.22	.54 TCP 2	2735 > 502 [PSH, ACK] S 502 > 2735 [PSH_ ACK] S	Seq=1 Ack=1 Win=175
1576 2006-11-29 19:31:58 181673 10 19	6 77 86 10 196 77	54 TCP	2735 5 502 FPSH ACKI S	Sen=29 Ack=20 Win="
<				
표 Frame 1 (80 bytes on wire, 80 bytes ca	ptured)			
Ethernet II, Src: 00:16:35:75:0e:4e (0 Internet Brateral Series 10, 106, 22, 06 (0:16:35:75:0e:4e), Dst: 00	:80:f4:00:65:17 (00:	80:f4:00:65:17)	
Internet Protocol, Src: 10.196.22.86 (■ Transmission Control Protocol, Src Por	10.196.22.86), Dst: 10.196 t: 2733 (2733). Dst Port:	502 (502), Sen: 0, A	y Ack: 0. Len: 26	
Data (26 bytes)		, , , , , , , , , , , , , , , , , , ,		
0000 00 80 t4 00 65 17 00 16 35 75 0e 4	le 08 00 45 00e 5	u.NE.		
0010 00 42 8a ae 40 00 80 08 40 14 0a 0 0020 16 36 0a ad 01 f6 a8 4d 6d fd 00 3	e 8c dc 50 18 .6M n	n>P.		
0030 42 42 48 57 00 00 00 00 00 01 00 1 0040 17 00 21 65 39 65 f9 1b 36 07 68 0	L4 00 f1 35 00 BBHW)7 90 01 64 00 !e9e 6	5. 5.hd.		
File: "D:\Audits\Merpins_automates\Merpins_panne_19-31	59-13.pcap" 2333 KB 00:10:00	P: 18067 D: 18067 M: 0		.:

Chaque trame va être analysée et les drapeaux (flag) ainsi que les numéros de séquences seront suivis. Le résultat est trié en 5 catégories :

- ERRORS : les problèmes réels comme des pertes de données. L'impact est donc visible.
- WARNINGS : les problèmes potentiels mais pas forcément réels.
- NOTES : les problèmes légers comme les retransmissions suspectées
- CHAT : le suivi des sessions (SYNchronisation, ReSeT, etc.)
- Details : est une vue des 4 catégories précédentes permettant de trier les données par type.

<u>7</u> 1	0-19	4-41-28	3_000	13_200	7010311	3435	- Wire	shark														-6	
Eile	⊑dit	⊻iew	<u>G</u> ο	⊆apture	Analyz	e St	atistics	Help															
	ř.	0	@ (\$	D	8	×	¢,	8	٩	\$	⇔	¢	Ŧ	₽ [Q, Q,	0	**	2	¥	
Eilter:											•	Expr	ession	. <u>⊂</u> lear	Apply								
No		Time					Source				Destin	ation			Protoco	l Info							
	1	2007-0	01-03 01-03	11:34	:35.651	1848	10.19	3.3.9 4.41.	28		10.19	94.41 98.3.	.28 9		TCP TCP	1521	> 183	3 [PSH, 1 FACK	ACK]	Seq=() Ack:	0 Ack= =42949	0 Win=2 65836 W	2
	3	2007-	01-03	11:34	:35.652	2447	10.19	3.3.9			10.19	94.41	.28		TNS	Resp	onse,	Data (6), Da	ta[Ma	lforme	d Packe	đ
	1	2007-0	01-03 01-03		1 Wiresl	hark:	8603 E	xpert	Infos												-	553	1
	6	2007-	01-03	11:		_																65 5 cko	8
	8	2007-	01-03	11:	Errors: 3	1 Wa	rnings: :	Note	s: 5 🤇	Ihats: 5	52 Det	ails										Wi	1
	9 10	2007-	01-03 01-03	11:	Group		✓ Pro	tocol	Summa	ry				Count							1	cke Wi	et i
	11	2007-	01-03	11:	🖃 Malfo	ormed	TNS	5 1	Malforn	ned Pack	ket (Exce	eption o	occured)	2468							<u> </u>	655	
	12	2007	01-03 01-03	11:	P	acket:	1															655 cke	e d
	14	2007-	01-03		P	acket:	3															ned	1
	16	2007-	01-03	11:	P	acket:	7															D W	: I /-
	17	2007-0	01 - 03	11:	P	acket:	9															655 cko	8
	19	2007-	01-03	11:	P	acket:	13															D W	, - 1-
	20	2007-0	01 - 03 01 - 03	11:	P	acket:	14															=65	E
	21	2007-	01-01		P	acket:	15																
	_				P	acket:	18															_	
E Fr	ame heri	3 (15 net IT	14 b) Sri	/tes -• oc	P	acket:	24															 Image: A set of the set of the	
⊥ Ir	iteri	net Pr	otoc	51, S										lose									
± Tr	ansi	missio	n Coi	itrol																			
± Tr	ans	parent	Net	work su	ibscraci	e pro	,																
	lalfo	ormed	Pack	et: TNS																			
0.000		A				- 4 . 7		<u>a ac</u>	<u> </u>					-									
0010	00	dc 2a	9 9 a . 1 f 5 .	ae cs 0 40 00 3	10 US	e4 07 cd 77	a 58 0 a 0a c	6 03	00 45	a c2	. *.@	.,		E.									
0020	29 60	1c 05 f4 29	; f1 () hf (07 29 6 00 00 0	52 4f)7 e3 i	56 9) 00 00	: 833 0060	d 96 0 00	91 50 00 00) 10	?)60 I	V=	Ρ.									
0040	08	00 00	02	c1 08 0	0 00		22 1	foo	00 03	5 c0													
0050	00	00 03	c0 (03 CU C 03 O7 C	0 00 0	00 00 03 bt	F 5b 1	5 00	00 03	1 80 3 c0			<u>.</u>										
0070	19	5b 00		03 c0 0 06 61 0	3 21	00 00		0 00 f 00	00 01	L 80	• [• • •	! a		••									
0090	05	59 00	ŏŏ	03 be 3	8 29	00 00	0 0 e 0	2 10	ca 01	ĹŎŠ	.Y	.8)											
00a0 00h0	00	02 63	8C (01 20 0 f3 ef f	0 00 i	06 0) 01 0(2 01 1	7 01 4 c3	1a 01 0h 0k	L 64 1 4 m	c	••		.d									~
File: "I	D:\Au	idits\PLM	1\10-19	94-41-28_	00013_20	00701	0311343	5" 512) KB 00	:02:13		[i	P: 15170) D: 151	70 M: 0								

En cliquant sur une erreur, le module affiche la trame dans le programme principal.

Attention : les onglets du module d'analyse expert indiquent le nombre de types d'erreurs reconnus. En cliquant sur l'onglet, chaque type d'erreur est affiché de manière condensé : il suffit d'explorer l'arborescence pour pouvoir afficher les trames.

Attention : le module d'analyse est une aide précieuse mais il ne permet pas un diagnostic à 100%. J'ai eu dans certains cas (protocole TNS d'Oracle) des messages « TNS unreassembled packets » qui étaient finalement dus à la multiplicité de requêtes simultanées : Wireshark n'est pas capable de différencier les différentes requêtes...

D'autres outils permettent l'analyse des protocoles utilisés et les temps de réponses ou bande passante.

4.3Analyse normale

La qualification d'un réseau nécessite de pouvoir déterminer l'utilisation de celui-ci. Cela inclut l'utilisation de la bande passante, les protocoles présents ainsi que leur proportion, les temps de latence, la répartition des tailles de paquets, etc.

4.3.1Informations sur la capture

Wireshark affiche les informations sur le fichier de capture avec notamment le débit moyen lors de la capture. Pour cela, il faut aller dans le menu [Statistics] [Summary].

📶 Wireshark: Sun	nmary							
File Name: Length: Format: Packet size limit:	D:\Audits\Merpins_automates\Merpins_panne_19-3159-13.pcap 2389415 bytes Wireshark/tcpdump/ libpcap 65535 bytes							
Time First packet: Last packet: Elapsed:	2006-11-2 2006-11-2 00:10:00	29 19:30:50 29 19:40:51						
Capture Interface: Dropped packets: Capture filter: Display	unknown unknown unknown							
Display filter: Marked packets:	none O							
Traffic		Captured	Displayed					
Between first and I Packets Avg. packets/sec Avg. packet size Bytes Avg. bytes/sec Avg. MBit/sec	ast packet	600,908 sec 18067 30,066 116,000 bytes 2100319 3495,242 0,028						
		⊆lose						

La durée de capture, ainsi que les dates de début et de fin sont indiquées de manière claire.

4.3.2Répartition des protocoles

Wireshark est capable de donner la répartition des protocoles sur une capture. Dans ce cas, plus la capture est grande, plus elle sera significative. Dans le menu [Statistics], sélectionner [Protocol Hierarchy] :



Wireshark analyse alors l'ensemble des trames et fournit une table donnant le pourcentage d'utilisation sur le nombre totale de trame : ainsi le pourcentage de la sous-catégorie « Malformed Packet » sous « Transparent Network Substrate Protocol » se rapporte bien à la totalité des trames de la capture.

Wireshark: Protocol Hierarchy Statistics							- 🗆 🛛
Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
🖃 Frame	100,00%	15170	5000882	0,299	0	0	0,000
Ethernet	100,00%	15170	5000882	0,299	0	0	0,000
Internet Protocol	100,00%	15170	5000882	0,299	0	0	0,000
Transmission Control Protocol	100,00%	15170	5000882	0,299	3893	275594	0,016
Malformed Packet	0,77%	117	68598	0,004	117	68598	0,004
Transparent Network Substrate Protocol	22,28%	3380	3324402	0,199	1109	392346	0,023
Malformed Packet	14,29%	2168	2869228	0,171	2099	2827790	0,169
Transparent Network Substrate Protocol	0,33%	50	30004	0,002	0	0	0,000
Malformed Packet	0,30%	46	27552	0,002	44	26326	0,002
\pm Transparent Network Substrate Protocol	0,01%	2	1226	0,000	0	0	0,000
Transparent Network Substrate Protocol	0,03%	4	2452	0,000	0	0	0,000
 Malformed Packet 	0,03%	4	2452	0,000	2	1226	0,000
Malformed Packet	0,13%	19	11434	0,001	19	11434	0,001
Transparent Network Substrate Protocol	0,68%	103	62828	0,004	0	0	0,000
Malformed Packet	0,59%	90	54859	0,003	86	52407	0,003
Malformed Packet	0,01%	2	1226	0,000	2	1226	0,000
Transparent Network Substrate Protocol	0,01%	2	1226	0,000	0	0	0,000
Malformed Packet	0,01%	2	1226	0,000	2	1226	0,000
Transparent Network Substrate Protocol	0,09%	13	7969	0,000	0	0	0,000
Malformed Packet	0,07%	11	6743	0,000	6	3678	0,000
Transparent Network Substrate Protocol	0,03%	5	3065	0,000	0	0	0,000
Malformed Packet	0,01%	2	1226	0,000	2	1226	0,000
Short Frame	0,02%	3	1839	0,000	3	1839	0,000
Short Frame	0,01%	2	1226	0,000	2	1226	0,000
Data	7,33%	1112	665286	0,040	1112	665286	0,040
DCE RPC	0,05%	7	6994	0,000	0	0	0,000
Malformed Packet	0,03%	4	2452	0,000	0	0	0,000
NetWare Core Protocol	43,88%	6656	657412	0,039	6656	657412	0,039
	<u>ο</u> κ						

Il n'est – hélas – pas possible de copier les informations contenues dans cette fenêtre, ni même, les trier par colonnes.

4.3.3Répartition des tailles de paquets

Wireshark est capable d'afficher la répartition des paquets par taille. Dans le menu [Statistics], choisir [Packet Length...]

Une fenêtre s'affiche permettant de filtrer sur quels éléments la répartition doit être calculée : il n'est pas nécessaire de remplir le champ...

📶 Wireshark: Packet L	ength Stats	Tree 🗕 🗆 🔯
Eilter:		
	reate Stat	

En cliquant sur le bouton [Create Stat], Wireshark ouvre une fenêtre contenant la répartition demandée par tranche.

Comme pour la répartition hiérarchique de protocoles, il n'est – hélas – pas possible de copier les informations contenues dans cette fenêtre, ni même, les trier par colonnes.

📶 Packet Length				
Topic / Item	Count	Rate	Percent	
😑 Packet Length	18067	0,030066		
0-19	0	0,000000	0,00%	
20-39	0	0,000000	0,00%	
40-79	6413	0,010672	35,50%	
80-159	8173	0,013601	45,24%	
160-319	3457	0,005753	19,13%	
320-639	20	0,000033	0,11%	
640-1279	4	0,000007	0,02%	
1280-2559	0	0,000000	0,00%	
2560-5119	0	0,000000	0,00%	
5120-	0	0,000000	0,00%	
		⊆lose		

4.3.4Conversations

Wireshark est capable de montrer les conversations durant la capture, menu [Statistics] [Conversations].

				IPv4 Convers	ations			
Address A	Address B	Packets	* Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B	<u>^</u>
10.196.22.54	10.196.22.86	16127	1946278	8183	1327740	7944	618538	
10.196.22.31	10.196.22.86	272	34162	135	15130	137	19032	
10.196.22.31	10.196.23.255	24	2760	24	2760	0	0	
10.196.22.133	10.196.23.255	19	2407	19	2407	0	0	
10.196.6.33	10.196.22.86	6	1722	3	210	3	1512	_
10.196.22.87	10.196.23.255	5	611	5	611	0	0	
10.196.22.86	10.196.23.255	4	519	4	519	0	0	
10.196.22.85	10.196.23.255	4	519	4	519	0	0	
10.196.22.83	10.196.23.255	3	276	3	276	0	0	
10.196.23.27	10.196.23.255	2	486	2	486	0	0	
10.196.22.88	10.196.23.255	2	335	2	335	0	0	
10.196.22.134	10.196.23.255	2	335	2	335	0	0	
10.196.23.15	10.196.23.255	2	496	2	496	0	0	
10.196.22.117	139.160.126.198	1	90	0	0	1	90	~
				⊆ору				

Les onglets permettent de choisir le type d'adressage (Ethernet, IPX, Ipv4) et même par protocoles (TCP ou UDP).

Il est possible de trier les données par colonnes (en cliquant sur le titre de la colonne une fois ou deux fois pour changer l'ordre) et de copier le résultat dans le presse-papier (bouton [Copy].

La troisième et la quatrième colonne (Packets et Bytes) sont respectivement la somme des colonnes 'Packets A->B + Packet B->A' et 'Bytes A->B + Bytes B-> A'.

4.4Analyse graphique "Time-Sequence" (tcptrace)

Cet outil graphique permet de voir rapidement la forme des échanges pour un flux sélectionné :



Les petits traits verticaux noirs (en forme de 'l') sont des trames envoyés (du premier au dernier octet, donc la hauteur représente la longueur de la trame).

La courbe bleue qui semble suit les traits verticaux noirs correspond à l'accusé de réception des trames : séquence ACK. Elle indique le délai d'acquittement de chaque trame et lorsqu'il y a une retransmission, un petit trait vers le bas est ajouté (Duplicate ACK).