

Mérési utasítás

WireShark használata, TCP kapcsolatok analizálása

A Wireshark (korábbi nevén Ethereal) a legfejlettebb hálózati sniffer és analizátor program. 1998-óta fejlesztik, jelenleg a GPL 2 licensz alatt. Nem igen találni ilyen széleskörű szolgáltatásokkal és ismeretekkel rendelkező hálózati analizátor programot. Támogatott operációs rendszerek: Windows, Linux, OS X, Solaris, FreeBSD, NetBSD és még sok egyéb. Grafikus interaktív interfésszel rendelkezik. Az OSI ISO modell 2-7 rétegének minden implementációját tudja analizálni. A program által jelenleg ismert protokollok száma jelenleg több mint 81000!

A Wireshark analizátor funkcióit több könyv, illetve elektronikus irodalom írja le több száz oldal terjedelemben, így gyakorlaton csak az alap funkciókkal ismerkedünk meg.



1. Feladat.

Amennyiben nincs telepítve a számítógépre, telepítse a wireshark-ot.

apt-get install wireshark



Nézzük át a wireshark kezelőfelületét.

<u>F</u> ile <u>E</u>	dit ⊻	ew <u>(</u>	<u>Go</u> ca	pture	<u>A</u> na	alyze	e <u>S</u> ta	atistics	<u>H</u> el	р												
e: C	1 01					×	¢		Ì			Ŵ	T	2	[Ð	Q	Q	+ +	M	V
🗹 <u>F</u> ilte	r:										•		<u>x</u> pre	ssior	יייי אין איי	<u>T</u> örlé	s 🖌	<u>A</u> lkalr	naz			

Az első gombbal hívhatjuk elő a wireshark által elérhető és használható hálózati interfészeket.

🖸 Wirest	ark: Capture Interfaces					_ - ×
Device	Description	IP	Packets	Packets/s	۲	Stop
🛒 wifio					≧ (<u>S</u> tart	<u>مار O</u> ptions
🛒 ath0		192.168.1.112			<mark>≧(</mark> <u>S</u> tart	<u>مار O</u> ptions
🛒 eth4	I	92.168.100.215	1223153	1192	<mark>≧(</mark> <u>S</u> tart	<u>مار O</u> ptions
🛒 any	Pseudo-device that captures on all interfaces		1223153	1192	≧ (<u>S</u> tart	<u>مار O</u> ptions
🛒 lo		127.0.0.1			≧ (<u>S</u> tart	<u>مار O</u> ptions
	×	<u>B</u> ezárás				

Ezen az ábrán láthatóak a "sniffelhető" interfészek, IP címekkel, és az áthaladt csomagok számával. A második gombbal állíthatjuk be az analizálás tulajdonságait.

<u>File Edit View Go Capture Analyze Statistics H</u>elp

	🗁 🎇 × 🕸 📇 🗟 💠 🗢	<u>⊼ ⊻ [</u>] ; Q Q @ [] ; X								
Filter:	▼ ♣	Expression 🦕 Törlés 🖌 Alkalmaz								
	Wireshark: Capture Options Capture									
	Interface: eth4 IP address: 192.168.100.215, fe80::211:11ff:febb:d801 Link-layer header type: Ethernet ↓ ☑ Capture packets in promiscuous mode									
	Limit each packet to	Display Options								
	File: Browse	<u>Update list of packets in real time</u>								
	□ Next file every 1 ⁺ megabyte(s)	☐ <u>H</u> ide capture info dialog								
	Stop Capture after 1 + file(s)	Name Resolution								
	□ after 1	Enable <u>n</u> etwork name resolution								
	after	Mégsem								



Széchenyi István Egyetem Győr Távközlési Tanszék

Legfelül látható, hogy jelen esetben az eth4-es interfészt használjuk. A "Capture packet in promiscuous mode" kapcsolót mindig hagyjuk bekapcsolva, így ún. monitor módba állítjuk a hálókártyát. Be lehet itt állítani, hogy a Wireshark fájlba mentse el az elkapott csomagokat. Megadhatjuk az analizálás leállásának feltételeit is, csomagszám, elkapott csomagok mérete és időkorlát alapján.

A Display options menüben lehet a csomagelkapás közbeni információkat beállítani. Automatikus "real-time" kijelzés, valamint ennek függvényében a képernyő görgetése, és az elkapott csomagok számának kijelzése.

Az utolsó részben lehet a névfeloldás lehetőségeinek beállítása, vagyis nem IP címeket kell ez esetben keresnünk, hanem az ezekhez hozzárendelt szimbolikus neveket, valamint a MAC-ben az első 3 byte helyett a gyártó neve.

A következő két gomb a csomag elkapás indítása, illetve leállítása.

2. feladat.

Indítsunk egy csomagelkapást az eth4-en, úgy hogy a leállítás feltétele legyen 1 perc, valamint a képernyő automatikusan gördüljön a csomagokkal. (amennyiben a wireshark megkérdezi, nem kell menteni az előző listát.) Majd a böngészőt elindítva kérje le az *index.hu* honlapot.

🔼 (Unti	Untitled) - Wireshark							
<u>F</u> ile <u>E</u> c	dit ⊻iew <u>G</u> o g	Capture Analyze Statistics	5 <u>H</u> elp					
ð ë	i 🖻 🜒 📦	🕒 🖬 🗙 🏟 📇	💽 🗢 🗢 🛧 🛃 🔳	₃ (Q, Q, @, M ¥ K K Ø				
Filter			💌 💠 Expression 🏷 I	irlés 🖋 <u>A</u> lkalmaz				
No	Time	Source	Destination Protocol I	nfo 🔨				
	2 7.574377	192.168.100.215	192.168.100.1 DNS S	tandard query AAAA index.hu				
	3 7.574612	192.168.100.1	192.168.100.215 DNS 9	tandard query response				
	4 7.574692	192.168.100.215	192.168.100.1 DNS S	tandard query AAAA index.hu.tilb.sze.hu				
	5 7.574902	192.168.100.1	192.168.100.215 DNS S	Tandard query response, No such name				
	7 7 580520	217 20 130 97	192 168 100 215 TCP	www.s.57711 [SYN_ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=57597556 TSER=8				
	8 7.580549	192.168.100.215	217.20.130.97 TCP 5	7711 > www [ACK] Seg=1 Ack=1 Win=5840 Len=0 TSV=87097506 TSER=57597556				
	9 7.580825	192.168.100.215	217.20.130.97 HTTP 0	ET / HTTP/1.1				
1	0 7.586409	217.20.130.97	192.168.100.215 TCP w	ww > 57711 [ACK] Seq=1 Ack=398 Win=6912 Len=0 TSV=57597557 TSER=87097506				
1	1 7.604638	217.20.130.97	192.168.100.215 TCP [TCP segment of a reassembled PDU]				
1	2 7.604651	192.168.100.215	217.20.130.97 TCP 5	7711 > www [ACK] Seq=398 Ack=1449 Win=8736 Len=0 TSV=87097512 TSER=57597562				
1	3 7.604785	217.20.130.97	192.168.100.215 TCP [TCP segment of a reassembled PDU				
	4 7.604796	192.168.100.215	217.20.130.97 TCP 5	//11 > WWW [ACK] Seq=398 ACK=289/ W1n=11632 Len=0 ISV=8/09/512 ISER=5/59/562				
	57.004505	217.20.130.37	152.100.100.215					
P Frame	543 (66 Dyles	s on wire, 66 bytes capt	ured)					
P Ether	net II, src: :	SUNMICTO_20:49:55 (00:14	:41:20:49:55), Dst: Intet_bb:d8:01					
Inter	net Protocol,	Src: 217.20.130.97 (217	.20.130.97), Dst: 192.168.100.215	(192.168.100.215)				
P Irans	mission contro	DI Protocol, Src Port: W	WW (80), DST Port: 5//16 (5//16),	Seq: 395, ACK: 4/9, Len: 0				
0000 0	0 11 11 bb d8	01 00 14 4f 20 49 55 08						
0010 0	0 34 95 23 40	00 3a 06 2a ab d9 14 82	2 61 c0 a8 .4.#0.:. *a					
0020 6	4 d7 00 50 e1	74 73 81 dd 74 b8 2e 0a	c7 80 11 dP.tst					
0030 0	0 36 al 66 00	00 01 01 08 0a 03 6a 53	ae 05 31 .6.fjS1					
File: "/tm	np/etherXXXXG8	BA9U" 152 KB 00:00:25		P: 548 D: 548 M: 0 Drops: 0				
T		😰 2 🖷 mc - feke	e5:~ - Parancs 🔋 Index - Konqueror	e =				
	1 22	3 4 📶 (Untitled) - Wireshark					



Széchenyi István Egyetem Győr Távközlési Tanszék

A Wireshark az elkapott csomagok sorszámát, a forrás és cél IP-t, a protokoll nevét valamint a csomag részletét jeleníti meg első látásra. Alul látható, hogy a Wireshark a különböző protokollokat sorrendbe helyezi. Először a csomag méretét adja meg, majd az Ethernet opciókat. Itt található a forrás és cél MAC cím. Alant az IP protokoll adatai láthatóak mint a forrás és cél IP. Majd végezetül a TCP tulajdonságokat nézhetjük meg. Mint például a forrás és cél port, valamint a különböző TCP bitek értékét (SYN, ACK, FIN stb.).

Jól megfigyelhető a képen, hogy először a mi gépünk lekéri a DNS bejegyzést a névkiszolgálótól, majd megkezdi IP cím alapján az index.hu kezdőlapját letölteni.

A hálózatokon sokszor rengeteg "szemét" csomag kering, mint például feszítőfa, illetve más egyéb routing protokoll. Ha ezeket figyelmen kívül szeretnénk hagyni, a csomagszűrőkhöz kell nyúlnunk.

Csomagszűrők két helyen alkalmazhatók:

- 1. csomagelkapásnál
- 2. megjelenítésnél

Ha csomagok elkapásánál használunk szűrőt, akkor csak a szűrési feltételeknek megfelelő csomagokat fogja a Wireshark eltárolni. Az eltárolt csomagok közül pedig megjelenítési szűrővel választhatjuk ki, hogy melyek jelenjenek meg a képernyőn. A két fajta szűrő szintaxisa sajnos különböző!

A csomagelkapási beállításokon (2. gomb) belül lehet csomagszűrőket alkalmazni.

A csomagszűrési beállításokon belül több előre definiált szűrő áll rendelkezésünkre.

😰 (Untitled) - Wireshark	_ # X
Elle Edit View Go Capture Analyze Statistics Help	
No. Time Source Destination Protocol 2 7.574377 192.166.100.1 192.166.100.1 DVS 3 7.574522 192.166.100.1 DVS DVS 4 7.574622 192.166.100.1 DVS DVS 5 7.574572 192.166.100.1 DVS DVS 6 7.576522 192.166.100.1 DVS DVS 7 7.60502 217. Capture Ethernet 1, Social and no Multicast 9 7.805761 192.166.100.215, fe80:211:11ffebb:080 Image: Social and no Multicast 9 7.805761 192.166.100.215, fe80:211:11ffebb:080 Image: Social and no Multicast 11 7.605761 192.100.1 DVs Image: Social and no Multicast 13 7.604780 227.100.100.215, fe80:211:11ffebb:080 Image: Social and no Multicast 13 7.604780 227.100.100.215 Image: Social and no Multicast 14 7.604780 227.100.100.100.100.100.100.100.100.100.10	597556 TSER-6 597556 67097506 ER-6-5797562 SER-57597562
000 00 14 4f 20 49 55 00 11 1b bd 01 00 00 14 4f 20 49 55 00 11 1b bd 01 00 10 10	
🔽 🔥 🗽 🚺 2 - fekete5:~ - Parancs 🔋 Index - Konqueror	0
🔛 🏠 🧏 3 4 🗧 (Untitled) - Wireshark	<u> </u>



Meg lehet adni protokollszűrést, IP cím szűrést, forrás és célport szűrést.

3. feladat

Hajtsuk végre az előző feladatot, úgy hogy most filterként beállítjuk, hogy csak a 80-as portot érintő kommunikációt vizsgáljuk. (*Capture Filter port 80*).

🔼 (Unti	tled) - Wiresha	ark							_ 6 X
Eile Eo	dit ⊻iew <u>G</u> o	<u>C</u> apture <u>A</u> nalyze <u>S</u> tat	stics <u>H</u> elp						
8	i 🗟 🚳 👜	(🖻 🖬 × 🐵	l 🔹 🗢 🖘 🐴	2	, O, O, O	. 🖭 🌌 🔛 💥	0		
Filter			▼ 💠 Expre	ssion 🗞 I	örlés 🖌 <u>A</u> lkalmaz				
No	Time	Source	Destination	Protocol Ir	nfo				
	1 0.000000	192.168.100.215	217.20.130.97	TCP 3	5436 > www [SYN]	Seq=0 Len=0 MSS=1460 TSV=	87348385 TSER=0 WS=4		
	2 0.005135	217.20.130.97	192.168.100.215	TCP w	ww > 35436 [SYN,	ACK] Seq=0 Ack=1 Win=5792	Len=0 MSS=1460 TSV=	57608712 TS	ER=87
	3 0.005190	192.168.100.215	217.20.130.97	TCP 3	5436 > www [ACK]	Seq=1 Ack=1 Win=5840 Len=	0 TSV=87348386 TSER=	57608712	
	4 0.005067	192.168.100.215	217.20.130.97	HTTP G	ET / HTTP/1.1	C		0-07240200	
	5 0.009864	217.20.130.97	192.168.100.215	TCP W	WW > 35436 [ACK]	Seq=1 ACK=441 W1n=6912 Le	n=0 ISV=5/608/13 ISE	8=87348386	
	7 0 018243	192 168 100 215	217 20 130 97		5436 S May [ACK]	Seg=441 Ack=1449 Win=8736	Len-0 TSV-87348389	TCED-576087	15
	8 0 018289	217 20 130 97	192 168 100 215	1 90T	TCP segment of a	reassembled PDU1	- Ecil-0 134-0/340303	1521-570007	
	9 0.018307	192,168,100,215	217,20,130,97	TCP 3	5436 > www [ACK]	Seg=441 Ack=2897 Win=1163	2 Len=0 TSV=87348389	TSER=57608	715
1	0 0.018417	217.20.130.97	192.168.100.215	TCP [TCP segment of a	reassembled PDU]			
1	1 0.018436	192.168.100.215	217.20.130.97	TCP 3	5436 > www [ACK]	Seq=441 Ack=4345 Win=1452	8 Len=0 TSV=87348389	TSER=57608	715
1	2 0.026410	217.20.130.97	192.168.100.215	TCP [TCP segment of a	reassembled PDU]			
1	.3 0.026443	192.168.100.215	217.20.130.97	TCP 3	5436 > www [ACK]	Seq=441 Ack=5793 Win=1742	4 Len=0 TSV=87348391	TSER=57608	716
1	4 0.026566	217.20.130.97	192.168.100.215	TCP [TCP segment of a	reassembled PDU]			-
▶ Frame	e 2332 (66 by	tes on wire, 66 bytes	captured)						
Ether	net II, Src:	Intel_bb:d8:01 (00:11	:11:bb:d8:01), Dst: SunMi	cro_20:49:5	5 (00:14:4f:20:49	9:55)			
▶ Inter	net Protocol	Src: 192.168.100.215	(192.168.100.215), Dst:	80.48.15.22	(80.48.15.22)				
▶ Trans	mission Cont	rol Protocol. Src Port	: 35956 (35956). Dst Port	: www (80).	Seg: 381. Ack: 2	26498. Len: 0			
0000 -	0 14 46 05 15	55 00 11 11 kk /5 5	00.00.45.00.0.5.						
0000 0	0 14 4T 20 49 0 34 15 30 40	0 00 11 11 00 d8 0	UBUU 45 000 IU	E.					-
0010 0	0 34 13 30 40 f 16 8c 74 00	50 f6 f3 cf 4b 27 0	74 e8 80 14 + P	K' +					
0030 0	e dc 44 c9 00	00 01 01 08 0a 05 3	d9 21 59 57	4.!YW					
File: "/tn	np/etherXXXXIV	ICAA9U" 1865 KB 00:00:	94		P: 2332 D: 2332	M: 0 Drops: 0			•
	· · · ·	12 mm - f	ekete5	Kopqueror	a				
<u>K</u> 4	A A	3 4 🖉 (Unti	led) - Wireshark	- Konqueror				🖹 📑	12:34

Most csak a 80-as portot érintő kommunikációt jelenítjük meg.

4. feladat

Hajtsuk végre az előző feladatot úgy, hogy a csomagelkapás leállításának feltétele 3 csomag elkapása legyen. Ezzel az előző feladatból csak a "three way handshake" vagyis a 3 utas kézfogást kaptuk meg.

Ez a TCP protokoll kapcsolat felépítési fázisa.





(Untitled) - Wireshark Elle Edit View Go Capture Analyze Statistics Help ▼ 🕂 Expression... 🧞 Törlés 🖌 Alkalmaz Filter: Destination Protocol Info Source Time 0 Ack=1 Wir 14 TSE Frame 1 (74 bytes on wire, 74 bytes captured) Ethernet II, Src: Intel_bbid8:01 (00:11:11:bbid8:01), Dst: SumMicro_02:49:55 (00:14:4f:20:49:55) Internet Protocol, Src: 192.168.100.215 (192.168.100.215), Dst: 217.20.130.97 (217.20.130.97) Transmission Control Protocol, Src Port: 39371 (39371), Dst Port: www (80), Seq: 0, Len: 0 Source port: 39371 (39371) Destination port: www (80) Sequence number: 0 (relative sequence number) Header length: 40 bytes Flags: 0x02 (SYN) Window size: 5840 Checksum: 0xed9b [correct] ▶ Options: (20 bytes)
 00
 14
 4f
 20
 49
 55
 00
 11
 11
 bb
 d8
 01
 08
 00
 45
 00
 ..0
 IU..
 ...
 E.

 00
 3c
 17
 ea
 40
 06
 al
 dc
 0c
 a8
 4d
 dd
 14
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 ...
 010 0020 0030 File: "/tmp/etherXXX11L38U" 286 Bytes 00:00:00 P: 3 D: 3 M: 0 Drops: 0 🛤 mc - fekete5:~ - Parancs 📳 Index - Kongueror 2 🔣 🏫 😣 R 🖪 82:41 3 4 🛛 (Untitled) - Wireshark

A csomagokat "kibontva" látható, hogy a 3 utas kézfogás egy TCP SYN bittel kezdődik egy sequence number=0-val, majd a szerver visszaküldi a TCP SYN,ACK bitekkel egy sequence number=0 és Acknowledge number=1-el, majd ismét válaszolunk egy TCP ACK bittel, ahol mind a sequence number mind az acknowledge number 1-re van állítva.

Természetesen ezek csak jelen helyzetben ilyen értékűek a könnyebb megértés érdekében.

Ezzel létrejött a TCP kapcsolat.



5. feladat

Hajtsuk végre az előző feladatot úgy, hogy vegyük ki a csomagelkapás leállítási feltételt, és most a http://poisson.tilb.sze.hu lapot kérjük le. (Az egyszerűség kedvéért.)

🔼 (Untit	🖉 (Untitled) - Wireshark								
<u>F</u> ile <u>E</u> di	Elle Edit View Go Capture Analyze Statistics Help								
	or or or	🗁 🗔 × 🟟 📇	🗟 🗢 🗢 🖗	5 ± 🗐 🗟 (Q, Q, 🗹 📓 🖄 📓 🛠 🕲					
Filter:			💌 🕂 Expre	pression 🦕 Törlés 🖌 Alkalmaz					
No	Time	Source	Destination	Protocol Info					
1	1 0.000000	192.168.100.215	193.224.129.164	TCP 34106 > www [SYN] Seq=0 Len=0 MSS=1460 TSV=88802857 TSER=0 WS=4					
2	2 0.000212	193.224.129.164	192.168.100.215	TCP www > 34106 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 TSV=154180776 TSER=€					
3	3 0.000232	192.168.100.215	193.224.129.164	TCP 34106 > www [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=88802857 TSER=154180776					
4	4 0.000614	192.168.100.215	193.224.129.164	HIP GET / HIP/1.1 TO MARK 20105 [ACK] Coop1 Ack-207 Min-5012 Long TO/-164190776 TCED-09902967					
6	5 0 001392	193.224.129.164	192.168.100.215	HTTP HTTP/1 1 200 0K (Hetr/bhal)					
7	7 0.001414	192,168,100,215	193.224.129.164	TCP 34106 > www [ACK] Seg=397 Ack=394 Win=6912 Len=0 TSV=88802857 TSER=154180776					
6	8 15.002883	193.224.129.164	192.168.100.215	TCP www > 34106 [FIN, ACK] Seq=394 Ack=397 Win=6912 Len=0 TSV=154184527 TSER=888026					
9	9 15.045051	192.168.100.215	193.224.129.164	TCP 34106 > www [ACK] Seq=397 Ack=395 Win=6912 Len=0 TSV=88806618 TSER=154184527					
10	0 16.046991	192.168.100.215	193.224.129.164	TCP 34106 > www [FIN, ACK] Seq=397 Ack=395 Win=6912 Len=0 TSV=88806869 TSER=1541845					
11	1 16.047275	193.224.129.164	192.168.100.215	TCP www > 34106 [ACK] Seq=395 Ack=398 Win=6912 Len=0 TSV=154184788 TSER=88806869					
				-					
b. Ensure	O (CC huter								
P Frame	8 (66 Dyles	on wire, 66 bytes capt	(red)						
V Ethern	net II, SIC:	Summicro_20:49:55 (00:	100 004 100 104) Det: In						
Interr	net Protocol,	SFC: 193.224.129.164	193.224.129.164), Dst:	: 192.168.100.215 (192.168.100.215)					
Transn Transn	mission contr	rol Protocol, Src Port:	WWW (80), DST Port: 341	(34106), Seq: 394, ACK: 397, Len: 0					
Sour	rce port: www	v (80)							
Dest	tination port	t: 34106 (34106)							
Sequ	uence number:	: 394 (relative sequ	nce number)						
Ackr	nowledgement	number: 397 (relati	re ack number)						
Head	der length: 3	32 bytes							
▶ Flag	gs: Ox11 (FIM	N, ACK)							
Wind	dow size: 691	l2 (scaled)							
Chec	cksum: Oxf8d9	5 [correct]							
▶ Opti	ions: (12 byt	tes)							
▶ [SEC	Q/ACK analysi	is]							
0000 00	11 11 bb d8	01 00 14 4f 20 49 55	08 00 45 00 0	0 TIL E					
0010 00	34 bd 48 40	00 3f 06 15 77 c1 e0	81 a4 c0 a8 .4.H@.?	·					
0020 64	d7 00 50 85	3a a6 94 23 77 65 67	9f ba 8011 dP.: #	. #weg					
0030 00) 36 f8 d5 00	00 01 01 08 0a 09 30	ab 4f 05 4b .6	0.0.K					
File: "/tm	p/etherXXXXY	9918U" 1731 Bytes 00:00:	16	P: 11 D: 11 M: 0 Drops: 0					
		🖪 2 🖷 jampy@	dev: ~ - Parancsé 👿 http://	o://poisson.tilb.sze.hu					
🔛 😭	1 253	3 4 📶 (Untitle	d) - Wireshark	🗵 🖂 (4. j					

Itt az utolsó négy csomagban megfigyelhető a 4 utas kézfogás, mely a TCP kapcsolat lebontását jelenti. Először a szerver küld egy FIN bitet amelyre mi ACK bittel válaszolunk. Majd mi is küldünk egy FIN bitet, amelyre a szerver válaszol ACK-al.

