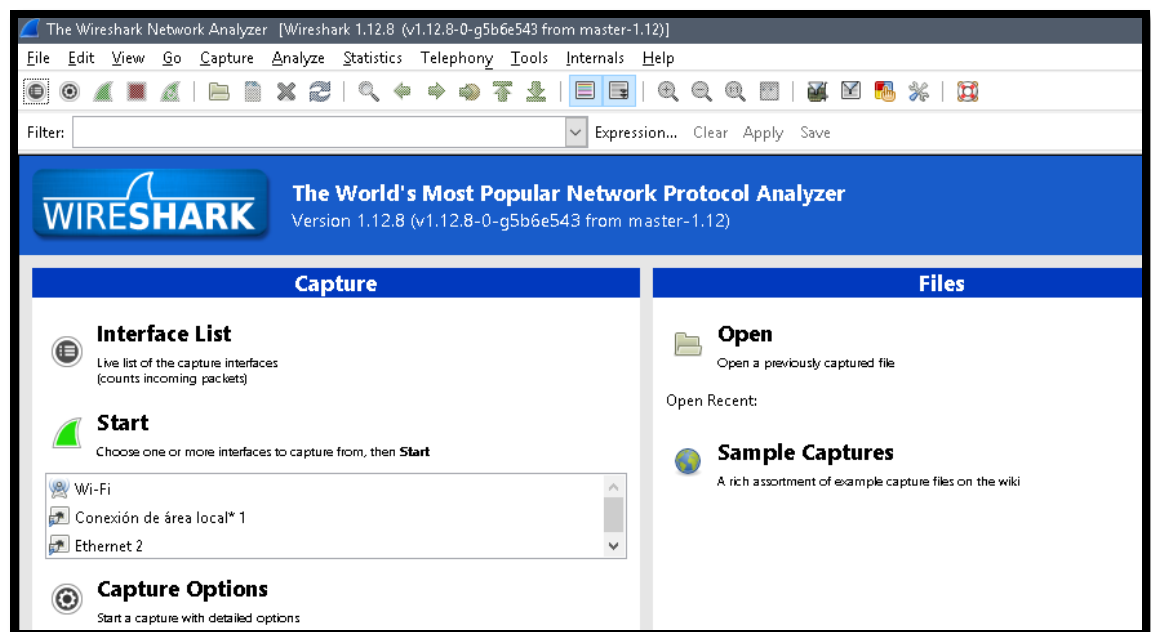


MONITOREO DE PAQUETES CON WIRESHARK

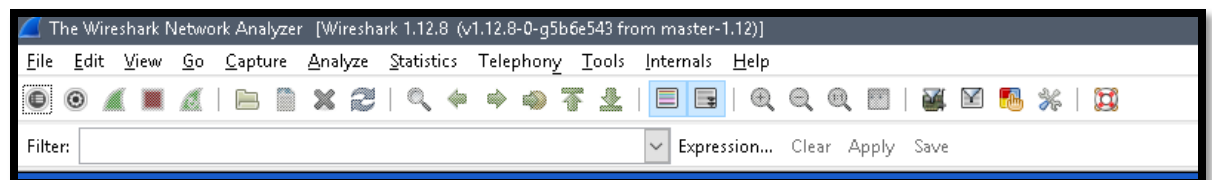
Elaborado por: Yersson Stiven Malaver Barreto

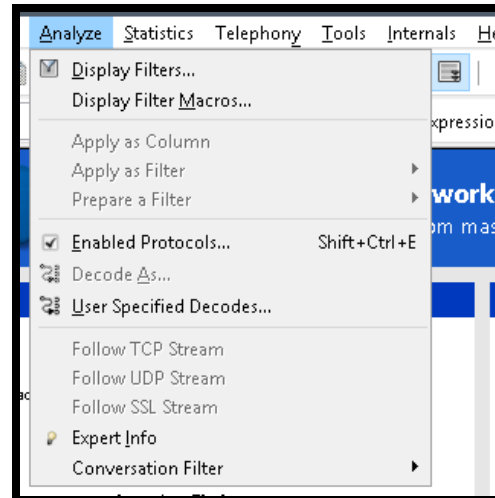
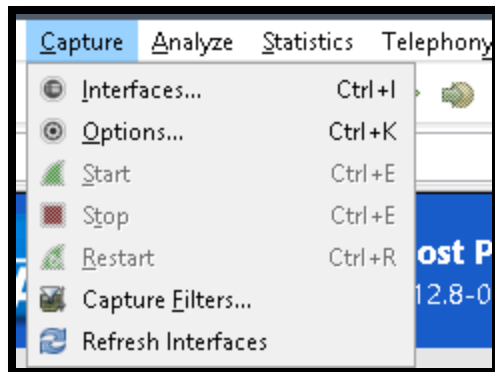
Elementos a tener en cuenta de Wireshark:

- 1) Desde un usuario privilegiado (Administrador) abrimos nuestra herramienta de monitoreo Wireshark, esto es debido a que para poder trabajar e inicializar de manera efectiva las interfaces de red se deben poseer ciertos privilegios.

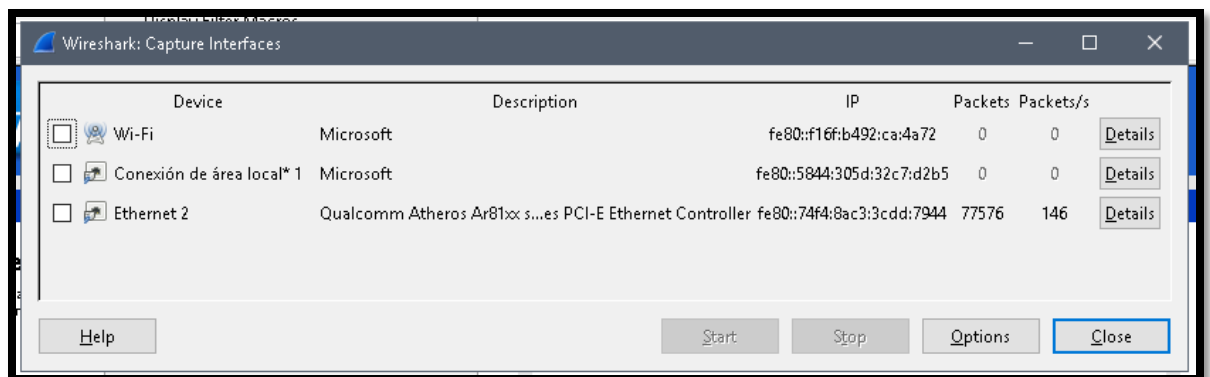


- 2) Una vez tenemos la aplicación abierta, pasamos a conocer las funcionalidades de los botones de la interfaz principal y en seguida pasamos a los sub-menús que nos serán útiles para el manejo del software, es importante que este paso lo realice con su tutor, previa consulta por parte suya de los componentes del programa.

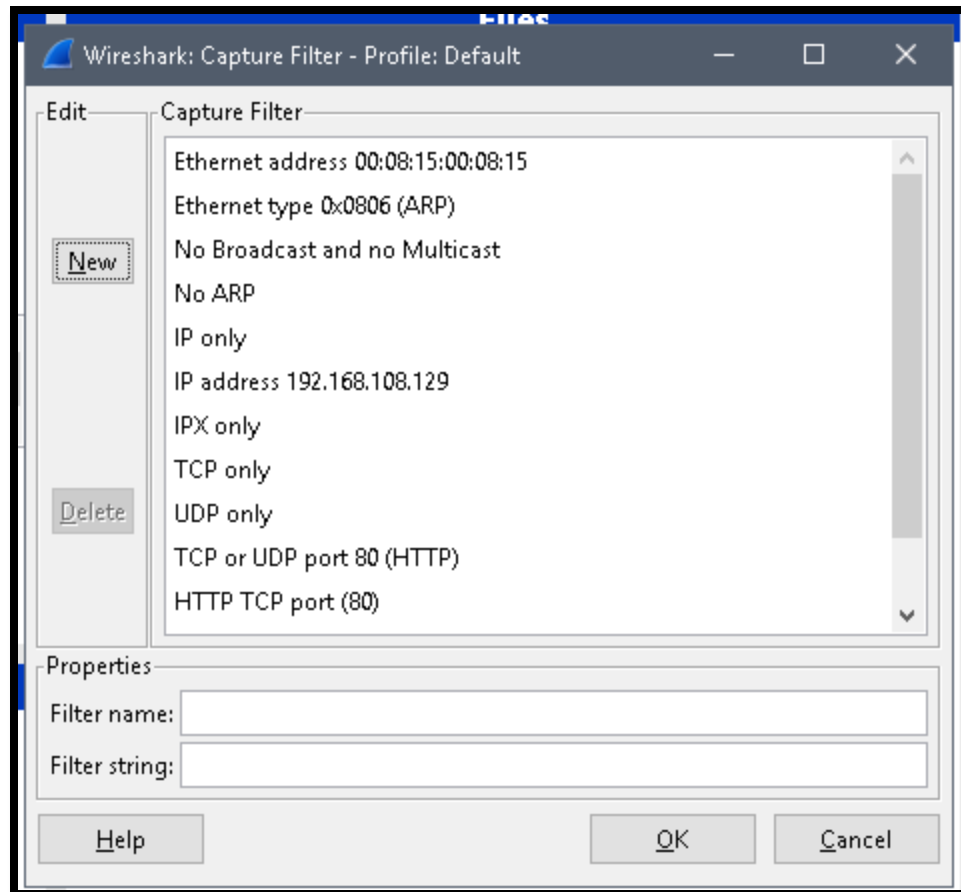




- 3) Luego del reconocimiento de los botones, menús y sub-menús del programa es importante reconocer por donde haremos la captura, para ello nos dirigimos al menú “capture” y seleccionamos la opción “interfaces”, allí podremos ver por cuál de las interfaces capturaremos tráfico.



- 4) Mediante el botón “Capture filter”, podemos seleccionar el filtro a utilizar para la captura de los datos, aquí se puede elegir o crear uno personalizado.



PARTE II - La práctica:

- 1) Desde la maquina atacante capturemos tráfico de red con Wireshark, es importante establecer comunicación con la maquina víctima, para ello podríamos hacer ping, o utilizar la maquina servidor que configuramos en la práctica 1.

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.10586]
(c) 2015 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Arsen>ping 192.168.0.112

Haciendo ping a 192.168.0.112 con 32 bytes de datos:
Respuesta desde 192.168.0.112: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.112: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.112: bytes=32 tiempo<1m TTL=128
Respuesta desde 192.168.0.112: bytes=32 tiempo<1m TTL=128

Estadísticas de ping para 192.168.0.112:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
              (0% perdidos),
    Tiempos aproximados de ida y vuelta en milisegundos:
        Mínimo = 0ms, Máximo = 0ms, Media = 0ms

C:\Users\Arsen>

```

2) Luego de capturado el tráfico de red, paramos la captura y guardamos el archivo generado. Respondemos las siguientes preguntas:

¿Qué información nos da la herramienta del tráfico capturado?

¿Cómo podemos interpretar lo capturado?

No.	Time	Source	Destination	Protocol	Length	Info
689	10.4748730	192.168.0.110	192.168.0.112	TCP	54	5938->15230 [ACK] Seq=92505 Ack=6329 win=256 Len=0
690	10.5027640	192.168.0.111	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x2e403751
691	10.5167440	HewlettP_b2:f5:75	Broadcast	ARP	60	who has 192.168.0.35? Tell 192.168.0.111
692	10.6710400	HewlettP_f5:72:4b	Broadcast	ARP	60	who has 192.168.0.207? Tell 192.168.0.5
693	10.8263210	192.168.0.111	239.255.255.250	SSDP	175	M-SEARCH * HTTP/1.1
694	11.0150670	192.168.0.254	192.168.0.255	NBNS	92	Name query NB NPI888429<00>
695	11.0531860	3comEuro_52:bd:81	Broadcast	ARP	60	who has 192.168.0.49? Tell 192.168.0.35
696	11.0919860	G-ProCom_ca:8a:6c	Broadcast	ARP	60	who has 192.168.0.26? Tell 192.168.0.214
697	11.1671260	192.168.0.110	192.168.0.112	TCP	345	5938->15230 [PSH, ACK] Seq=92505 Ack=6329 win=256 Len=291
698	11.1677110	192.168.0.110	192.168.0.112	TCP	254	5938->15230 [PSH, ACK] Seq=92796 Ack=6329 win=256 Len=200
699	11.1680340	192.168.0.112	192.168.0.110	TCP	60	15230->5938 [ACK] Seq=6329 Ack=92996 win=16149 Len=0
700	11.1714780	192.168.0.110	192.168.0.112	TCP	1466	5938->15230 [PSH, ACK] Seq=92996 Ack=6329 win=256 Len=1412
... 11.1717500 192.168.0.110 192.168.0.112 TCP 401 5938->15230 [PSH, ACK] Seq=94408 Ack=6329 win=256 Len=427						
<div> <div> Frame 1: 105 bytes on wire (840 bits), 105 bytes captured (840 bits) on interface 0 </div> <div> Ethernet II, Src: HewlettP_4b:64:fa (8c:dc:d4:4b:64:fa), Dst: dell_5d:2a:7a (f0:4d:a2:5d:2a:7a) </div> <div> Internet Protocol Version 4, Src: 192.168.0.112 (192.168.0.112), Dst: 192.168.0.110 (192.168.0.110) </div> <div> Transmission Control Protocol, Src Port: 15230 (15230), Dst Port: 5938 (5938), Seq: 1, Ack: 1, Len: 51 </div> <div> Data (51 bytes) </div> </div>						
0000	f0 4d a2 5d 2a 7a 8c dc	d4 4b 64 fa 08 00 45 00	.M.]*z.. .Kd...E.			
0010	00 5b 7d 01 40 00 80 06	fb 6c c0 a8 00 70 c0 a8	.[].@... .[]...p..			
0020	00 6e 3b 7e 17 32 90 fb	04 d3 b5 22 44 02 50 18	.n;~.2... .."d.P.			
0030	3f 7a e5 30 00 00 11 30	6b 00 1b 00 00 00 fa 17	?Z.0...0 k.....			
0040	01 00 f8 6a 01 00 1b 00	00 00 18 00 00 00 0c 00	...j.... ..k....			
0050	00 00 d9 0e 01 00 07 00	00 00 50 e1 6b a2 9a 22	...j.... ..P.k....			
0060	a6 04 c7 22 08 d2 77 b9	37	...n..w. 7			

Tráfico capturado.

Respuesta: La herramienta nos muestra el tráfico capturado a manera de registros. Con datos como Host fuente, el host de destino, protocolo utilizado, longitud del paquete.

La herramienta permite mostrar el contenido de cada registro en la parte inferior de la ventana. Allí se listan los componentes del paquete. Si seleccionamos un componente, Wireshark no muestra el contenido de ese elemento.

- 3) Ahora iniciamos una nueva captura y con un navegador web desde nuestra víctima (máquina virtual que se sugirió) abrimos una página cualquiera de internet.

Con lo anterior respondemos:

¿Qué paquetes distintos a la anterior captura se pueden identificar?

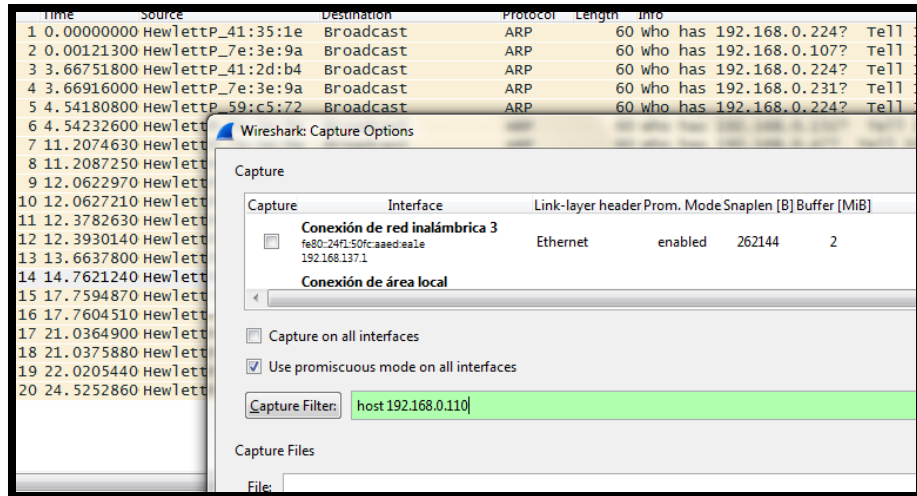
348	3.55703200	fe80::8a51:fbff:feff02::1:2		DHCPv6	127	Solicit	XID: 0x3d44 CID: 00020000000b8851fbeaa757
349	3.57806200	HewlettP_b0:f4:fd	Broadcast	ARP	60	who has	192.168.0.170? Tell 192.168.0.137
350	3.59508500	192.168.0.112	192.168.0.110	TCP	103	15230->5938	[PSH, ACK] Seq=2782 Ack=28486 win=16178 Len=49
351	3.59517500	192.168.0.110	192.168.0.112	TCP	54	5938->15230	[ACK] Seq=28486 Ack=2831 win=253 Len=0
352	3.61249400	192.168.0.112	192.168.0.110	TCP	104	15230->5938	[PSH, ACK] Seq=2831 Ack=28486 win=16178 Len=50
353	3.63944700	192.168.0.110	192.168.0.112	TCP	187	5938->15230	[PSH, ACK] Seq=28486 Ack=2881 win=253 Len=133
354	3.63977300	192.168.0.110	192.168.0.112	TCP	107	5938->15230	[PSH, ACK] Seq=28619 Ack=2881 win=253 Len=53
355	3.64022300	192.168.0.112	192.168.0.110	TCP	60	15230->5938	[ACK] Seq=2881 Ack=28672 win=16132 Len=0
356	3.64156900	192.168.0.112	192.168.0.110	TCP	104	15230->5938	[PSH, ACK] Seq=2881 Ack=28672 win=16132 Len=50
357	3.69184800	192.168.0.110	192.168.0.112	TCP	54	5938->15230	[ACK] Seq=28672 Ack=2931 win=253 Len=0
358	3.71389500	3comEuro_52:bd:ce	Spanning-tree-(for-STP	120	MST. Root = 32768/0/00:uf:cb:da:5c:00	Cost = 400000	Port = 0x800e
359	3.74310700	HewlettP_53:bb:22	Broadcast	ARP	60	who has	192.168.0.174? Tell 192.168.0.104

¿Cómo se puede interpretar la captura en términos de TCP/IP?

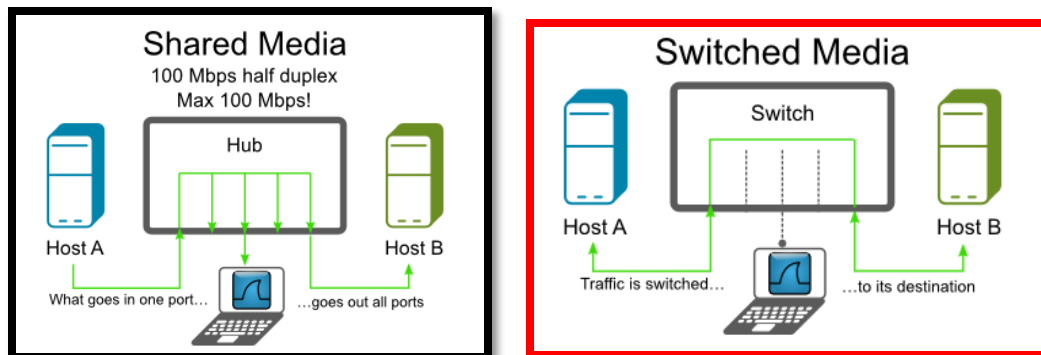
Este protocolo utiliza varias propiedades para operar, Wireshark nos muestra los detalles como por ejemplo la longitud del paquete, los datos de las cabeceras. Además nos muestra los datos del paquete en caracteres legibles al usuario.

Ahora vamos a usar un filtro predeterminado en las capturas, esto con el fin de disminuir capturas que en el momento no nos interesan. Teniendo clara la IP del equipo que deseamos monitorear vamos a usar el filtro de direcciones IP, en opciones de captura en el campo de filtro de captura escribimos lo siguiente:

host
ip_target



En las redes modernas ya no están fácil capturar los paquetes de un Ip distinta la nuestra. Los Switches tienen un sistema que impide la captura de los paquetes del objetivo. Como se menciona en esta página:
<https://wiki.wireshark.org/CaptureSetup/Ethernet>

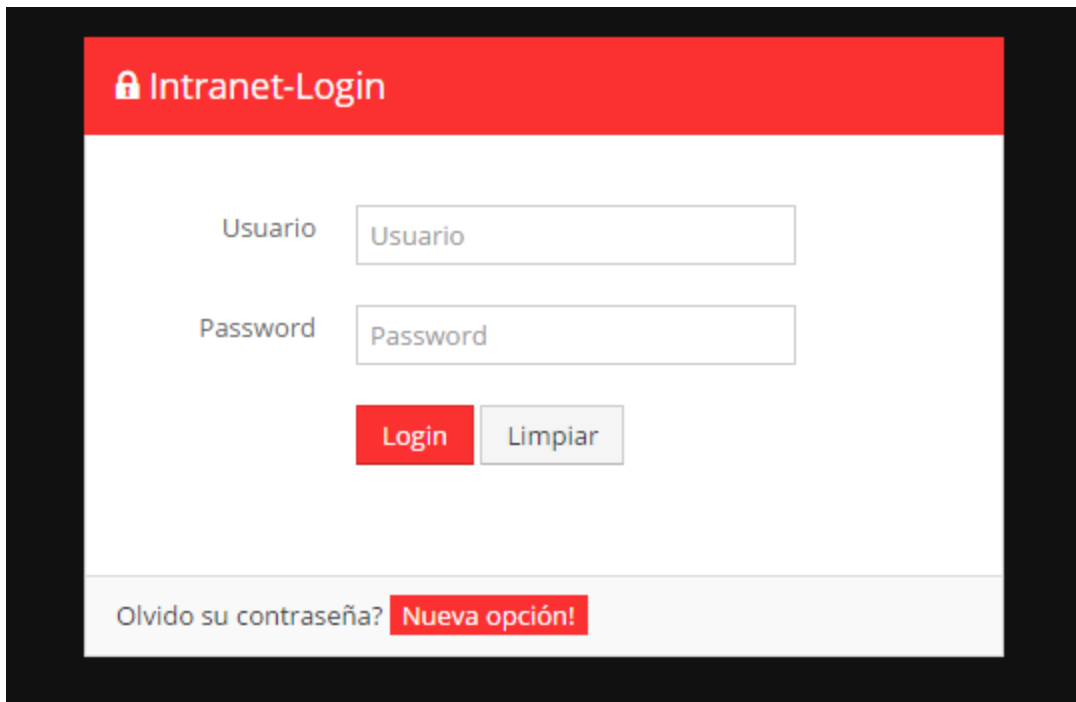


La solución que proponen es utilizar un Ethernet Hub o situarse directamente en la Ip que deseamos escanear.

No.	Time	Source	Destination	Protocol	Length	Info
98	9.36554300	192.168.0.110	192.168.0.112	TCP	103	5938→26277 [PSH, ACK] Seq=442 Ack=442 win=254
99	9.36628200	192.168.0.112	192.168.0.110	TCP	103	26277→5938 [PSH, ACK] Seq=442 Ack=491 win=160
100	9.38152300	192.168.0.112	64.4.46.81	TLSv1.2	1051	Application Data
101	9.41725200	192.168.0.110	192.168.0.112	TCP	60	5938→26277 [ACK] Seq=491 Ack=491 win=254 Len=
102	9.54648100	64.4.46.81	192.168.0.112	TLSv1.2	363	Application Data
103	9.54656400	192.168.0.112	64.4.46.81	TLSv1.2	779	Application Data
104	9.78027000	64.4.46.81	192.168.0.112	TLSv1.2	955	Application Data
105	9.93341000	64.4.46.81	192.168.0.112	TLSv1.2	203	Application Data
106	9.93348300	192.168.0.112	64.4.46.81	TCP	54	40773→443 [ACK] Seq=12695 Ack=10202 win=63190
107	10.3028950	74.125.196.189	192.168.0.112	QUIC	84	CID: 0, Seq: 140
108	10.3028960	74.125.196.189	192.168.0.112	QUIC	62	CID: 0, Seq: 141
109	10.3030820	192.168.0.112	74.125.196.189	QUIC	86	CID: 10322257214438298934, Seq: 12
110	10.3754260	192.168.0.110	192.168.0.112	TCP	103	5938→26277 [PSH, ACK] Seq=491 Ack=491 win=254
111	10.3763820	192.168.0.112	192.168.0.110	TCP	103	26277→5938 [PSH, ACK] Seq=491 Ack=540 win=160
112	10.4266960	192.168.0.110	192.168.0.112	TCP	60	5938→26277 [ACK] Seq=540 Ack=540 win=254 Len=
113	10.6358690	192.168.0.112	74.125.196.189	QUIC	276	CID: 10322257214438298934, Seq: 13
114	10.7145070	74.125.196.189	192.168.0.112	QUIC	122	CID: 0, Seq: 142
115	10.7145090	74.125.196.189	192.168.0.112	QUIC	82	CID: 0, Seq: 143
116	10.7148070	192.168.0.112	74.125.196.189	QUIC	83	CID: 10322257214438298934, Seq: 14
117	10.9790330	74.125.196.189	192.168.0.112	QUIC	85	CID: 0, Seq: 144
118	11.0045240	192.168.0.112	74.125.196.189	QUIC	80	CID: 10322257214438298934, Seq: 15
119	11.4004410	192.168.0.110	192.168.0.112	TCP	103	5938→26277 [PSH, ACK] Seq=540 Ack=540 win=254
120	11.4009670	192.168.0.112	192.168.0.110	TCP	103	26277→5938 [PSH, ACK] Seq=540 Ack=589 win=160
121	11.4516550	192.168.0.110	192.168.0.112	TCP	60	5938→26277 [ACK] Seq=589 Ack=589 win=254 Len=
122	11.8357190	192.168.0.112	40.117.100.83	TLSv1.2	587	Application Data
Frame 122: 587 bytes on wire (4696 bits), 587 bytes captured (4696 bits) on interface 0						
Ethernet II, Src: HewlettP_4b:64:fa (8c:dc:d4:4b:64:fa), Dst: 3comEuro_52:bd:81 (40:01:c6:52:bd:81)						
Internet Protocol Version 4, Src: 192.168.0.112 (192.168.0.112), Dst: 40.117.100.83 (40.117.100.83)						
Transmission Control Protocol, Src Port: 58851 (58851), Dst Port: 443 (443), Seq: 1, Ack: 1, Len: 533						
Application Data (533 bytes)						
0000	40 01 c6 52 bd 81 8c dc	d4 4b 64 fa 08 00 45 00	@..R....Kd...E.			
0010	02 3d 71 7a 40 00 80 06	00 00 c0 a8 00 70 28 75	..qz@....p(u			
0020	64 53 e5 e3 01 bb 02 b9	2e 8c 6b 51 8f 5f 50 18	dS.....kq..P.			
0030	40 34 50 10 00 00 17 03	03 02 10 7d 19 68 29 12	@4P.....}.h).			
0040	bb 6f 47 d0 96 9b ad fc	c5 fa 13 7c 84 cd 5a cf	.OG..... .Z.			
0050	ba ef 04 03 9c c7 db f1	f4 0a 42 3a 17 2a 9c 80B:.*..			
0060	c2 40 1d 76 38 f3 ed 87	6a 61 37 60 31 cd e3 91	..v8...ja7'l...			
0070	b3 32 62 d1 f9 f3 82 47	d5 53 69 cc b2 2d 75 ee	.2b...G.Si...u.			
0080	48 36 4d 4e 10 22 a7 5a	4d 94 71 44 87 f8 90 7b	H6MN...Z M.qD...{			
0090	04 9f 39 76 7c 7c 28 b7	51 ce 76 fc 5e 40 37 76	..9v (. Q.v.^@7v			
00a0	3d 09 d6 99 47 31 4b a0	ea ab 2a 47 9c 90 41 e7	...G1K...*G..A.			
00b0	e3 73 e0 b5 5e 58 ea 44	4f 2b 90 12 49 05 ef 01	.S..^X.D 0+..I...			
00c0	7e 47 cb 6e 73 a0 30 3d	76 33 87 70 f6 03 db c7	~G.ns.0= v3.p...			
00d0	00 db b7 8c b8 36 0b 6b	dd 41 0b 92 3c 95 f8 eb6.k.A..<...			
00e0	bc 0e fe b0 8a 50 95 34	54 7d 4d 31 07 3e 37 79P.4 T}M1.>7y			
00f0	aa a9 17 99 7c 5a e9 e4	72 7d dc d0 53 19 ce dd	...[Z..r}..S...			
0100	7a 7d 35 44 c3 c2 b8 c2	65 35 72 fb c7 c7 b5	..%o...f...			

En vista del problema de captura mencionado anteriormente, se hará una prueba con la maquina local, y ver si podemos capturar los datos de navegación:

Sito web objetivo: <http://190.60.95.8:8080/intranet/>



Capturamos los paquetes y obtenemos los datos de acceso al sitio:

Filter: tcp.stream eq 2

Time	Source	Destination	Protocol	Length	Info
14	1.89784100	192.168.0.112	TCP	66	20640-8080 [SYN] Seq=0 win=8
15	1.91139000	190.60.95.8	TCP	66	8080-20640 [SYN, ACK] Seq=0
16	1.91147600	192.168.0.112	TCP	54	20640-8080 [ACK] Seq=1 Ack=1
17	1.91193200	192.168.0.112	HTTP	720	POST /intranet/Loginuser HTTP/1.1
18	1.92601700	190.60.95.8	TCP	60	8080-20640 [ACK] Seq=1 Ack=0
19	1.92990000	190.60.95.8	TCP	1514	[TCP segment of a reassembled
20	1.92990200	190.60.95.8	TCP	735	[TCP previous segment not c
21	1.92990200	190.60.95.8	HTTP	1514	[TCP out-of-order] HTTP/1.1
22	1.92999900	192.168.0.112	TCP	66	20640-8080 [ACK] Seq=667 Ack
23	1.93004200	192.168.0.112	TCP	54	20640-8080 [ACK] Seq=667 Ack
44	4.17112800	192.168.0.112	HTTP	693	POST /intranet/Loginuser HTTP/1.1
46	4.18658500	190.60.95.8	TCP	1514	[TCP segment of a reassembled
47	4.18658500	190.60.95.8	TCP	470	[TCP previous segment not c
48	4.18670000	192.168.0.112	TCP	66	20640-8080 [ACK] Seq=1306 Ack
49	4.18683200	190.60.95.8	HTTP	1514	[TCP out-of-order] HTTP/1.1
50	4.18689000	192.168.0.112	TCP	54	20640-8080 [ACK] Seq=1306 Ack

Stream Content

```

Content-Length: 45
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://190.60.95.8:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.94 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://190.60.95.8:8080/intranet/Loginuser
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.8
Cookie: JSESSIONID=BC0D25D0F87A5DAAA9F2C91B062C4BD

usuario=pruebalogeo&password=micontrasea45678HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 3452
Date: Thu, 19 May 2016 13:17:33 GMT
  
```


Time	Source	Destination	Protocol	Length	Info
14	1.89784100	192.168.0.112	190.60.95.8	TCP	66 20640→8080 [SYN] Seq=0 win=8192 Len=0 M
15	1.91139000	190.60.95.8	192.168.0.112	TCP	66 8080→20640 [SYN, ACK] Seq=0 Ack=1 win=1
16	1.91147600	192.168.0.112	190.60.95.8	TCP	54 20640→8080 [ACK] Seq=1 Ack=1 win=65700
17	1.91193200	192.168.0.112	190.60.95.8	HTTP	720 POST /intranet/LoginUser HTTP/1.1 (app
18	1.92601700	190.60.95.8	192.168.0.112	TCP	60 8080→20640 [ACK] Seq=1 Ack=667 win=1600
19	1.92990000	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]
20	1.92990200	190.60.95.8	192.168.0.112	TCP	735 [TCP Previous segment not captured] [TC
21	1.92990200	190.60.95.8	192.168.0.112	HTTP	1514 [TCP out-of-order] HTTP/1.1 200 OK (te
22	1.92999900	192.168.0.112	190.60.95.8	TCP	66 20640→8080 [ACK] Seq=667 Ack=1461 win=6
23	1.93004200	192.168.0.112	190.60.95.8	TCP	54 20640→8080 [ACK] Seq=667 Ack=3602 win=6
44	4.17112800	192.168.0.112	190.60.95.8	HTTP	693 POST /intranet/LoginUser HTTP/1.1 (app
46	4.18658500	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]
47	4.18658500	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]
48	4.18658500	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]
49	4.18658500	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]
50	4.18658500	190.60.95.8	192.168.0.112	TCP	1514 [TCP segment of a reassembled PDU]

Follow TCP Stream (tcp.stream eq 2)

Stream Content

```

Content-Length: 45
Cache-Control: max-age=0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Origin: http://190.60.95.8:8080
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/50.0.2661.94 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Referer: http://190.60.95.8:8080/intranet/LoginUser
Accept-Encoding: gzip, deflate
Accept-Language: es-419,es;q=0.8
Cookie: JSESSIONID=BC0D225DDF87A5DAAA9F2C91B062C4BD

usuario=pruebaalgeo&password=micontrase456788HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 3452
Date: Thu, 19 May 2016 13:17:33 GMT

```

Entire conversation (8242 bytes)

Find Save As Print ASCII EBCDIC Hex Dump C Arrays Raw

Help Filter Out This Stream Close

Con esta herramienta se ha comprendido un poco mejor como viaja la información en las redes informáticas. Y también como hay herramientas que permiten hacer seguimiento a los paquetes de nuestros usuarios. Muestran la importancia de implementar buena seguridad en nuestros sistemas de información.

REFERENCIAS BIBLIOGRÁFICAS

- ✓ Cambero, L. (2015). Análisis de Tráfico con Wireshark (Parte1) [Video]. Disponible en: <https://youtu.be/lxCENEGhztU>
- ✓ Gómez, R. (2014). Tutorial como Instalar y Configurar Wireshark [Video]. Disponible en: <https://youtu.be/10AMih59955U>
- ✓ iTutosPc (2013). Como Descargar e Instalar VirtualBox 4.3.2 en Windows 8 32 y 64 BITS 2016 [Video]. Disponible en: https://youtu.be/H_GgZmT1u7s
- ✓ Samboni Núñez, D. M. (2012). Diapositivas online tituladas "Manual Básico de WireShark". Publicadas en el sitio web 2.0 de alojamiento de diapositivas denominado "Slideshare", el 26 de Febrero de 2012. Extraídas el 18 de Mayo de 2016, <http://es.slideshare.net/DIANYSS2012/manual-bsico-de-wireshark>
- ✓ windowscracker1 (2015). Descargar e Instalar VMWare Workstation 11 Full - Virtualiza varios sistemas operativos [2016] [Video]. Disponible en: https://youtu.be/3ajD6m_gyHE