Servidores Web de Altas Prestaciones (2015-2016)
Grado en Ingeniería Informática
Universidad de Granada

Wireshark

Jiménez Bernal, Manuel
manuasir@correo.ugr.es

Castro Guerrero, Juan Miguel
jcastroguerrero@correo.ugr.es

18 de septiembre de 2016

Índice

Índice de figuras

1. Introducción

En el presente documento se pretenden explicar los diferentes motivos por los cuales la herramienta Wireshark debería ser una herramienta indispensable en el mantenimiento de un entorno de red con servidores y del cual se pretenda que se mantenga seguro.
La herramienta Wireshark es un analizador de protocolos open-source cuyo principal objetivo es el análisis de tráfico para solucionar problemas en redes de comunicaciones aunque también es utilizado en el desarrollo de software y protocolos, como herramienta didáctica en entornos de pruebas y laboratorios, y para hallar fallos de seguridad en redes.
En un entorno de granja de servidores, la herramienta tiene un uso muy interesante en la monitorización del tráfico que se maneja para determinar el correcto funcionamiento de los protocolos que en él actúan, mantener la carga bajo parámetros establecidos para determinar sobrecargas y en definitiva asegurar que las conexiones que entran y salen son las esperadas en todos los casos.

2. Funcionalidades de Wireshark

- Captura de paquetes de datos en vivo a partir de una interfaz de red.
- Abrir archivos que contienen datos de paquetes capturados con tcpdump / WinDump, Wireshark, y una serie de otros programas de captura de paquetes.
- Importar paquetes de archivos de texto que contienen los códigos hexadecimales de paquetes de datos.
- Visualización de paquetes con información muy detallada de protocolo.
- Guardar los datos de los paquetes capturados.
- Exportar algunos o todos los paquetes en una serie de formatos de archivo de captura.
- Búsqueda de paquetes mediante una gran variedad de filtros.
- Crear varias estadísticas.

3. Guía de instalación en sistemas Windows y Linux

3.1. Windows

Desde el sitio web oficial de Wireshark se descargan los instaladores tanto para arquitecturas x86 como AMD64, habiendo que escoger el que se adecue a cada sistema. De la misma manera también es posible descargar los códigos fuente del programa, recompilarlos e incluso modificarlos en caso de que fuese necesario. El código oficial también se encuentra en la comunidad de repositorios GitHub para todo aquel desarrollador que

quisiera realizar un fork del mismo o realizar sus propuestas de mejora mediante pull request [2]. El software está desarrollando en C El proceso de instalacción consiste en seguir las intrucciones que presenta el asistente.

Figura 3.1: Paquetes disponibles para sistemas Windows

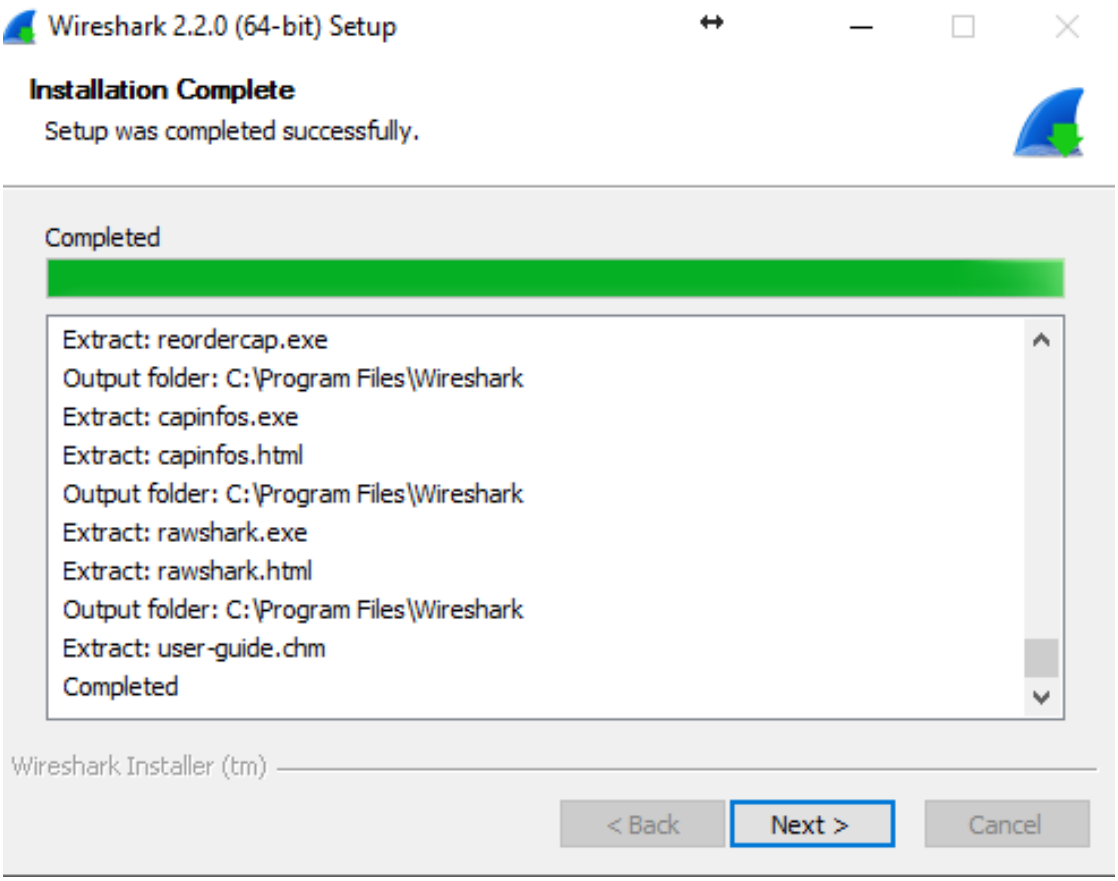


Figura 3.2: Pantalla del asistente de instalación de Wireshark

3.2. Linux

Para instalar Wireshark en los sistemas Linux basados en Debian y que usen el sistema de paquetes dpkg desde el shell basta con ejecutar las siguientes secuencias de comandos:

```
$ sudo apt-get update

$ sudo apt-get install wireshark
```

Para sistemas basados en Red Hat con el gestor de paquetes yum:

```
# yum check-update

# yum instal wireshark
```

Una vez descargado se abrirá una ventana para configurar el archivo wireshark-common en la que debemos permitir a los no superusuarios capturar paquetes. En caso de que no apareciera esta ventana debemos ejecutar el comando:
Como alternativa, una vez instalado podemos ejecutar wireshark con el comando

```
$ sudo wireshark
```

y no sería necesario el paso anterior, ya que lo estaríamos ejecutando como superusuario. Una vez instalado se procede a realizar una primera escucha de pruebas, tras ejecutar el programa y configurar la tarjeta de red (NIC) por la que se va a analizar el contenido de los paquetes, se procede a ejecutar el Start y el programa comienza a capturar paquetes de cualquier protocolo, al no habérsele establecido un filtro. Un ejemplo de filtro podría ser el siguiente

```
"ud.port == 53 || tcp.port == 80"
```

Para generar tráfico accedemos a una terminal desde la que se realiza algún comando que implique tráfico TCP y resolución de nombres. En el ejemplo se usó el siguiente comando:

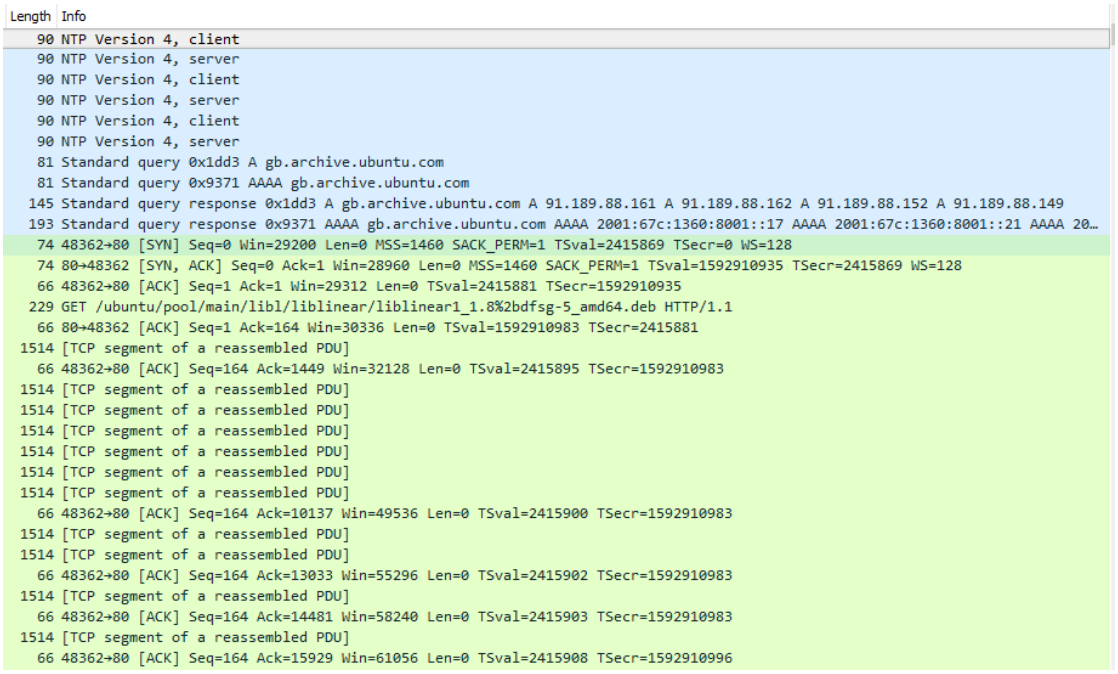


Figura 3.3: Ejemplo de captura tras realizar un host de la red el comando 'apt-get update'

```
$ sudo apt-get update
```

Tras lo cual se comprueba que en color azul se aprecia con facilidad las conexiones DNS para resolver la Ip que direcciona hacie el repositorio oficial de Ubuntu. Esto puede resultar de utilidad en caso de querer comprobar que los servidores a los cuales se accede son legítimos y no han sido modificados maliciosamente para que se apunte a otros no lícitos.

4. Casos de uso

A continuación se presentan distintos escenarios de relevancia en el ámbito de los servidores web en los cuales se va a analizar tráfico en distintos entornos habituales para un administrador de redes

4.1. Análisis de tráfico en un escaneado de puertos

Para este escenario se ha instalado el paquete nmap en una de las máquinas de nuestra red. Esta herramienta realiza un escaneo de puertos en los hosts que se le indiquen por parámetro y devuelve un feedback con aquellos que se encuentren en modo 'listening'

,con el fin de monitorizar actividad que pueda llegar a comprometer vulnerabilidades en esos puertos. Por otra parte este tipo de monitorización puede ser útil al administrador del sistema para comprobar si algún host remoto está intentando escanear el estado de los puertos sin permiso. Para la simulación se atacó la IP de un servidor con los siguientes resultados:

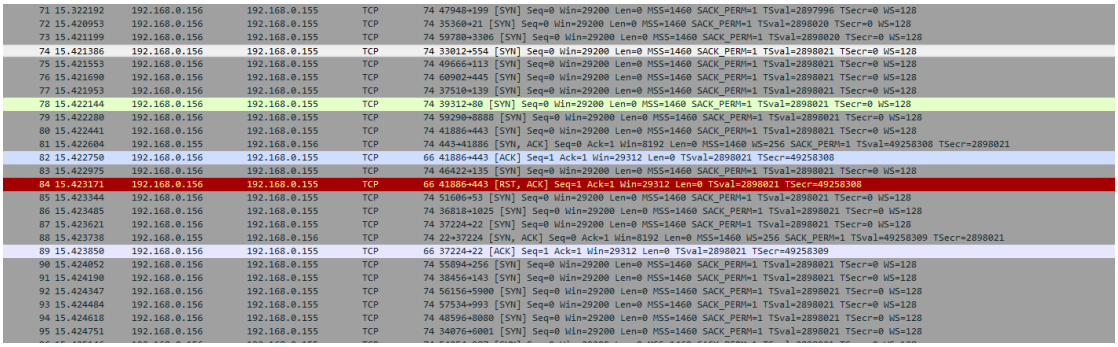


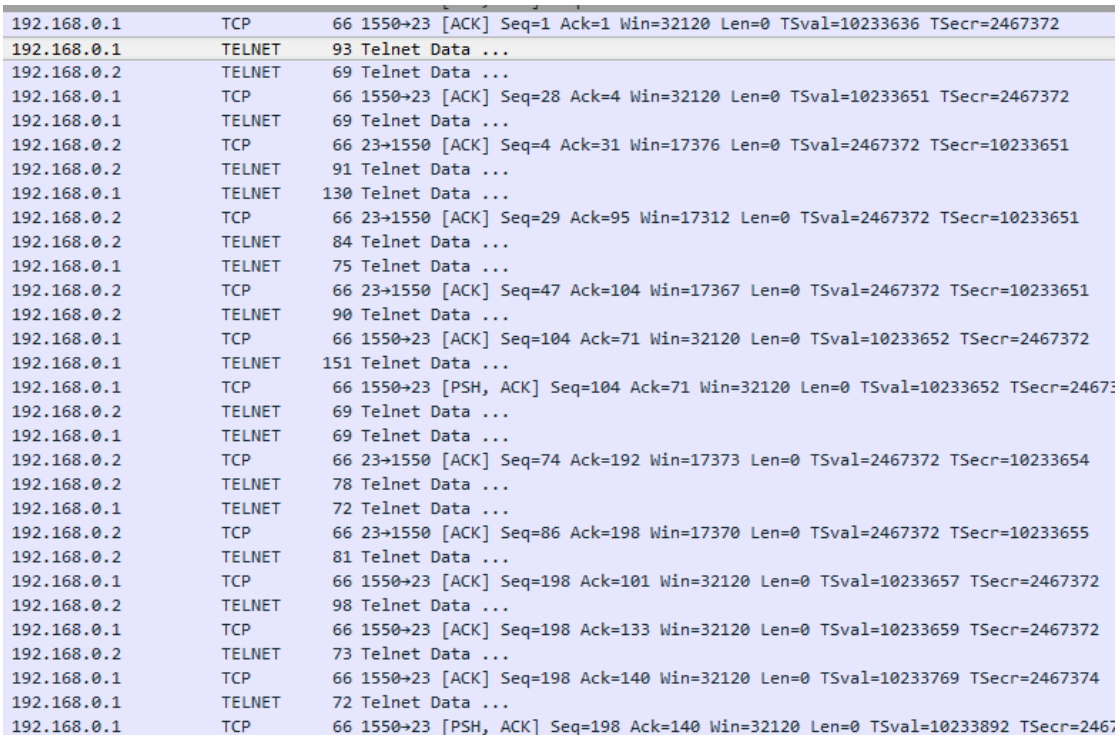
Figura 4.1: Captura de escaneado de puertos

Como se comprueba en la figura, se reafirma que Wireshark es una herramienta excepcional en cuanto a las facilidades que proporciona al usuario con su interfaz gráfica resaltando con colores los diferentes tipos de mensajes que se intercambian en los proocolos. De esta forma se comprueba cómo una máquina con ip 192.168.0.156 está realizando un escaneo de puertos contra la máquina de IP 192.168.0.155. En éste intercambio de mensajes se aprecian cómo los paquetes que son respondidos con un ACK (acuse de recibo) son aquellos que se envían a los puertos que efectivamente tiene escuchando la máquina objetivo, en la figura el puerto 443 y 80, puertos bien conocidos por ser dedicados a usos de servidores Web y el puerto 22 dedicado por defecto a las conexiones del protocolo SSH. De esta forma descubrimos cómo opera la herramienta NMAP por debajo, realizando un barrido de peticiones a todos los puertos conocidos y señalando como abiertos los cuales envíen un ACK.

4.2. Análisis de tráfico en una sesión telnet

Telnet es un protocolo de red cuyo propósito es proveer de una comunicación bidireccional, principalmente proporcionando conexión remota con las terminales de hosts remotos. La comunicación entre estos terminales se realiza por la red sin usar encriptación, por lo que es un protocolo que se considera no seguro [3]. Pese a que hace años que el uso de este se vio reducido frente al protocolo SSH, que proporciona comunicación cifrada, telnet sigue usándose en dispositivos que se usan hoy en día como dispositivos de red del tipo router o switches. El fabricante Cisco es uno de los distribuidores que opera globalmente,y en sus manuales de configuración de los dispositivos se indica que la comunicación con los dispositivos se realiza mediante telnet, enviándose tanto los comandos como las credenciales en texto plano [1]. Un atacante puede hacer un uso malintencionado de Wireshark para interceptar estas comunicaciones y conseguir las credenciales de acceso a

los router. En la siguiente simulación se pretende mostrar la captura como resultado de una escucha al protocolo telnet:



192.168.0.1	TCP	66	1550→23 [ACK] Seq=1 Ack=1 Win=32120 Len=0 TSval=10233636 TSecr=2467372
192.168.0.1	TELNET	93	Telnet Data ...
192.168.0.2	TELNET	69	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [ACK] Seq=28 Ack=4 Win=32120 Len=0 TSval=10233651 TSecr=2467372
192.168.0.1	TELNET	69	Telnet Data ...
192.168.0.2	TCP	66	23→1550 [ACK] Seq=4 Ack=31 Win=17376 Len=0 TSval=2467372 TSecr=10233651
192.168.0.2	TELNET	91	Telnet Data ...
192.168.0.1	TELNET	130	Telnet Data ...
192.168.0.2	TCP	66	23→1550 [ACK] Seq=29 Ack=95 Win=17312 Len=0 TSval=2467372 TSecr=10233651
192.168.0.2	TELNET	84	Telnet Data ...
192.168.0.1	TELNET	75	Telnet Data ...
192.168.0.2	TCP	66	23→1550 [ACK] Seq=47 Ack=104 Win=17367 Len=0 TSval=2467372 TSecr=10233651
192.168.0.2	TELNET	90	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [ACK] Seq=104 Ack=71 Win=32120 Len=0 TSval=10233652 TSecr=2467372
192.168.0.1	TELNET	151	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [PSH, ACK] Seq=104 Ack=71 Win=32120 Len=0 TSval=10233652 TSecr=24673
192.168.0.2	TELNET	69	Telnet Data ...
192.168.0.1	TELNET	69	Telnet Data ...
192.168.0.2	TCP	66	23→1550 [ACK] Seq=74 Ack=192 Win=17373 Len=0 TSval=2467372 TSecr=10233654
192.168.0.2	TELNET	78	Telnet Data ...
192.168.0.1	TELNET	72	Telnet Data ...
192.168.0.2	TCP	66	23→1550 [ACK] Seq=86 Ack=198 Win=17370 Len=0 TSval=2467372 TSecr=10233655
192.168.0.2	TELNET	81	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [ACK] Seq=198 Ack=101 Win=32120 Len=0 TSval=10233657 TSecr=2467372
192.168.0.2	TELNET	98	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [ACK] Seq=198 Ack=133 Win=32120 Len=0 TSval=10233659 TSecr=2467372
192.168.0.2	TELNET	73	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [ACK] Seq=198 Ack=140 Win=32120 Len=0 TSval=10233769 TSecr=2467374
192.168.0.1	TELNET	72	Telnet Data ...
192.168.0.1	TCP	66	1550→23 [PSH, ACK] Seq=198 Ack=140 Win=32120 Len=0 TSval=10233892 TSecr=2467

Figura 4.2: Escucha de rotocolo telnet

Una vez se encuentre Wireshark capturando los paquetes, ofrece la funcionalidad de concatenar y ordenar los paquetes para reconstruir el flujo de comunicación con la herramienta Follow TCP stream,de esta manera se puede apreciar con detalle el contenido o payload de las tramas, al que podemos acceder y el cual es completamente legible al no estar cifrada la comunicación:

```
.......... ..!.."..'.....#..%..%.......... ..!.."..".........P. ....
.......................".....'.....#..&..&..$..&..&..$.. .....#....
0.0....'..DISPLAY.bam.zing.org:0.0......xterm-color.............!....
OpenBSD/i386 (oof) (ttyp2)

login: fake
......Password:user

......Last login: Sat Nov 27 20:11:43 on ttyp2 from bam.zing.org
Warning: no Kerberos tickets issued.
OpenBSD 2.6-beta (OOF) #4: Tue Oct 12 20:42:32 CDT 1999

Welcome to OpenBSD: The proactively secure Unix-like operating system

Please use the sendbug(1) utility to report bugs in the system.
Before reporting a bug, please try to reproduce it with the latest
version of the code.  With bug reports, please try to ensure that
enough information to reproduce the problem is enclosed, and if a
known fix for it exists, include that as well.

$ /sbin/ping www.yahoo.com
PING www.yahoo.com (204.71.200.67): 56 data bytes
64 bytes from 204.71.200.67: icmp_seq=0 ttl=241 time=69.885 ms
64 bytes from 204.71.200.67: icmp_seq=1 ttl=241 time=73.591 ms
64 bytes from 204.71.200.67: icmp_seq=2 ttl=241 time=72.302 ms
64 bytes from 204.71.200.67: icmp_seq=3 ttl=241 time=73.493 ms
64 bytes from 204.71.200.67: icmp_seq=4 ttl=241 time=75.068 ms
64 bytes from 204.71.200.67: icmp_seq=5 ttl=241 time=70.239 ms
..........
.--- www.yahoo.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 69.885/72.429/75.068 ms
$ ls
$ ls -a
.         ..        .cshrc    .login    .mailrc   .profile  .rhosts
$ exit
```

Figura 4.3: Contenido de la comunicación con telnet

Referencias

[1] Set up your cisco router. `http://www.cisco.com/public/technotes/smbsa/en/us/internet/proc_setup_router.pdf`, 2006.

[2] Github wireshark. `https://github.com/wireshark/wireshark`, 2016.

[3] J. Reynolds J. Poster. Rfc 854. https://tools.ietf.org/html/rfc854, 1983.