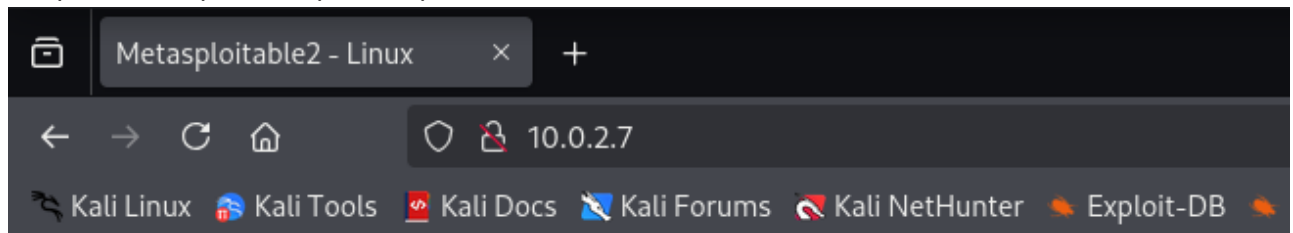


Wireshark análisis de tráfico

Tools

- Máquina Kali Linux (atacante)
- Debia (victima)
- Máquina Metasploitable (servidor)



metasploitable2

Warning: Never expose this VM to an untrusted network!

Contact: msfdev[at]metasploit.com

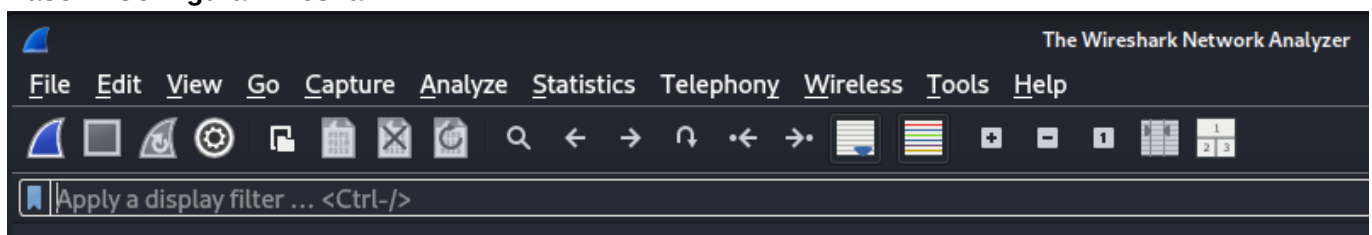
Login with msfadmin/msfadmin to get started

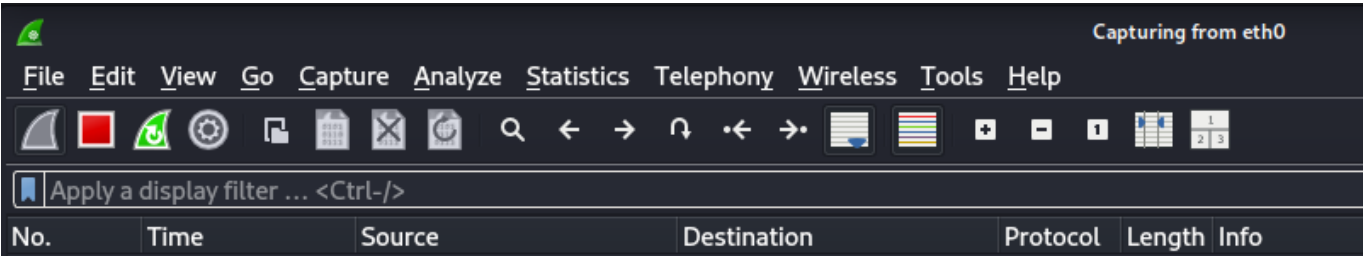
- [TWiki](#)
- [phpMyAdmin](#)
- [Mutillidae](#)
- [DVWA](#)
- [WebDAV](#)

Análisis de tráfico en Wireshark

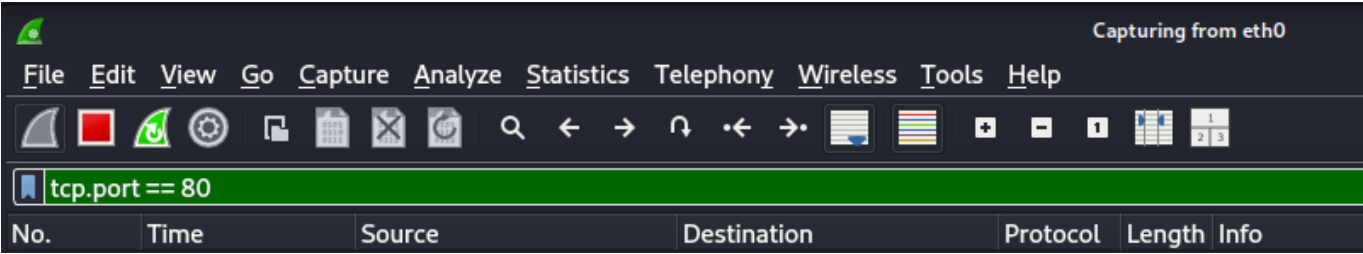
1 - Capturar y analizar el trafico HTTP

Paso 1: Configura Wireshark

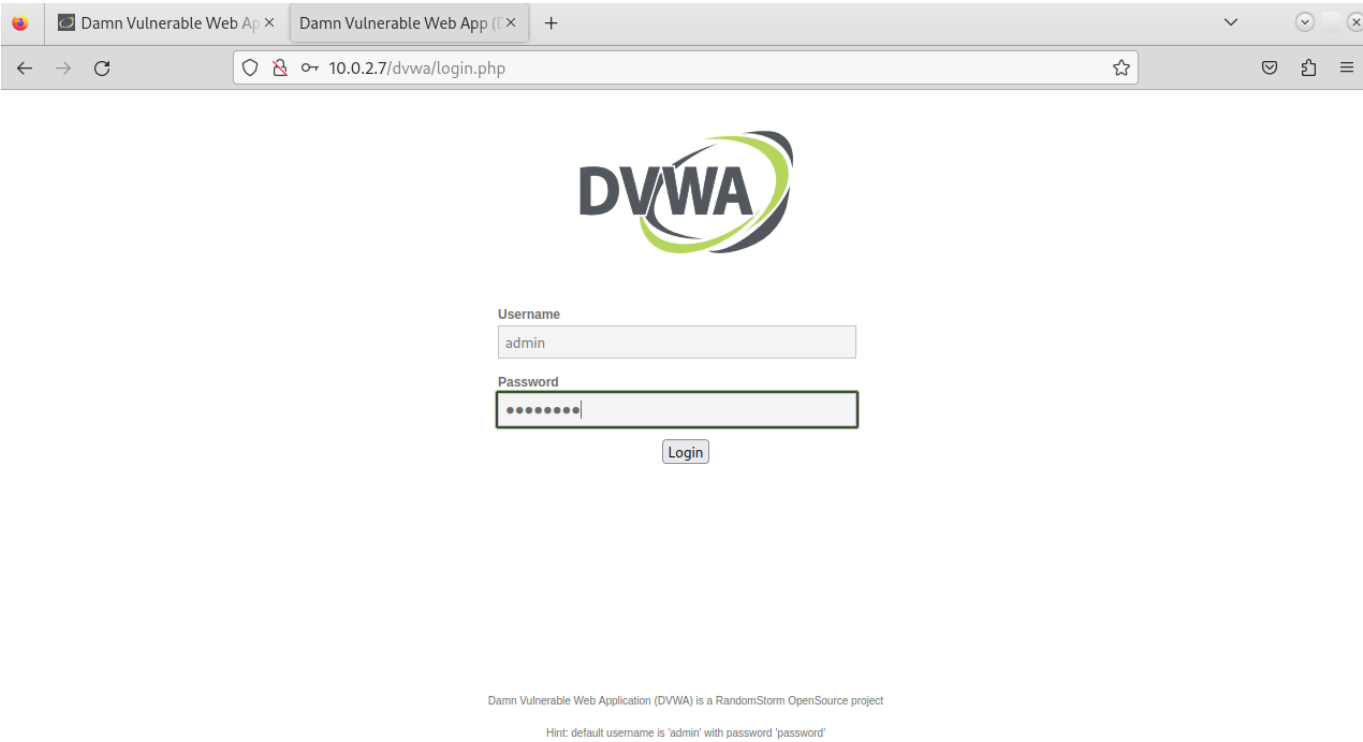




Paso 2: Aplicar filtro TCP para tráfico HTTP



Paso 3: Realizar una solicitud HTTP en el navegador



Paso 4: Analizar los paquetes capturados

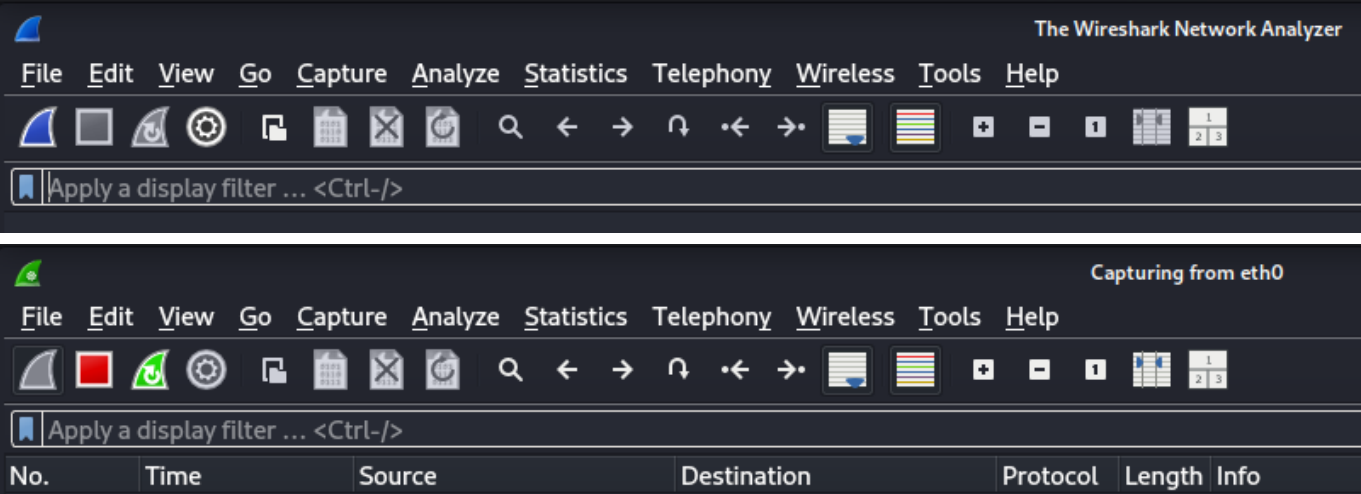
[illegible]

Explicación del resultado esperado:

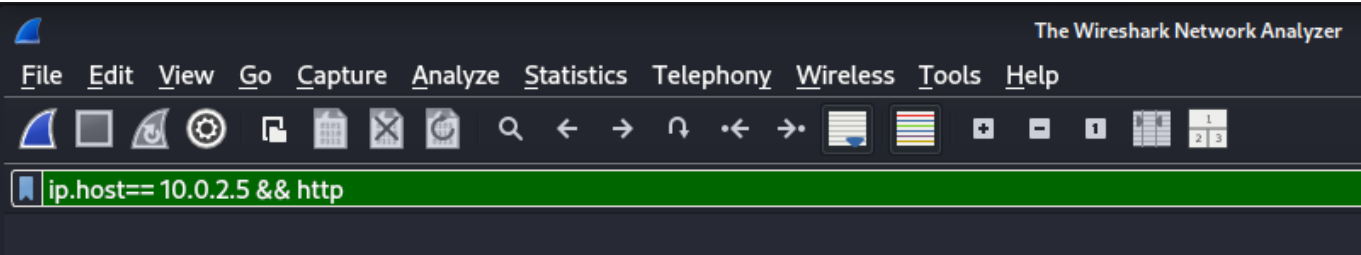
- La negociación TCP se ve como tres paquetes consecutivos con banderas y números de secuencia específicos. En la sección de solicitud HTTP, hay el nombre de usuario utilizado para autenticación "admin" junto con su contraseña correspondiente "password".

2 - Capturar y analizar ataque XSS

Paso 1: Configura Wireshark



Paso 2: Aplicar filtro HTTP desde Kali Linux



Paso 3: Realizar un ataque XSS en DVWA

Home

Instructions

Setup

Brute Force

Command Execution

CSRF

File Inclusion

SQL Injection

SQL Injection (Blind)

Upload

XSS reflected

XSS stored

DVWA Security

PHP Info

About

Logout

Username: admin

Security Level: low

PHPIDS: disabled

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

More info

<http://hackers.org/xss.html>

http://en.wikipedia.org/wiki/Cross-site_scripting

<http://www.cgisecurity.com/xss-faq.html>

View Source

View Help

DVWA

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

Submit

More info

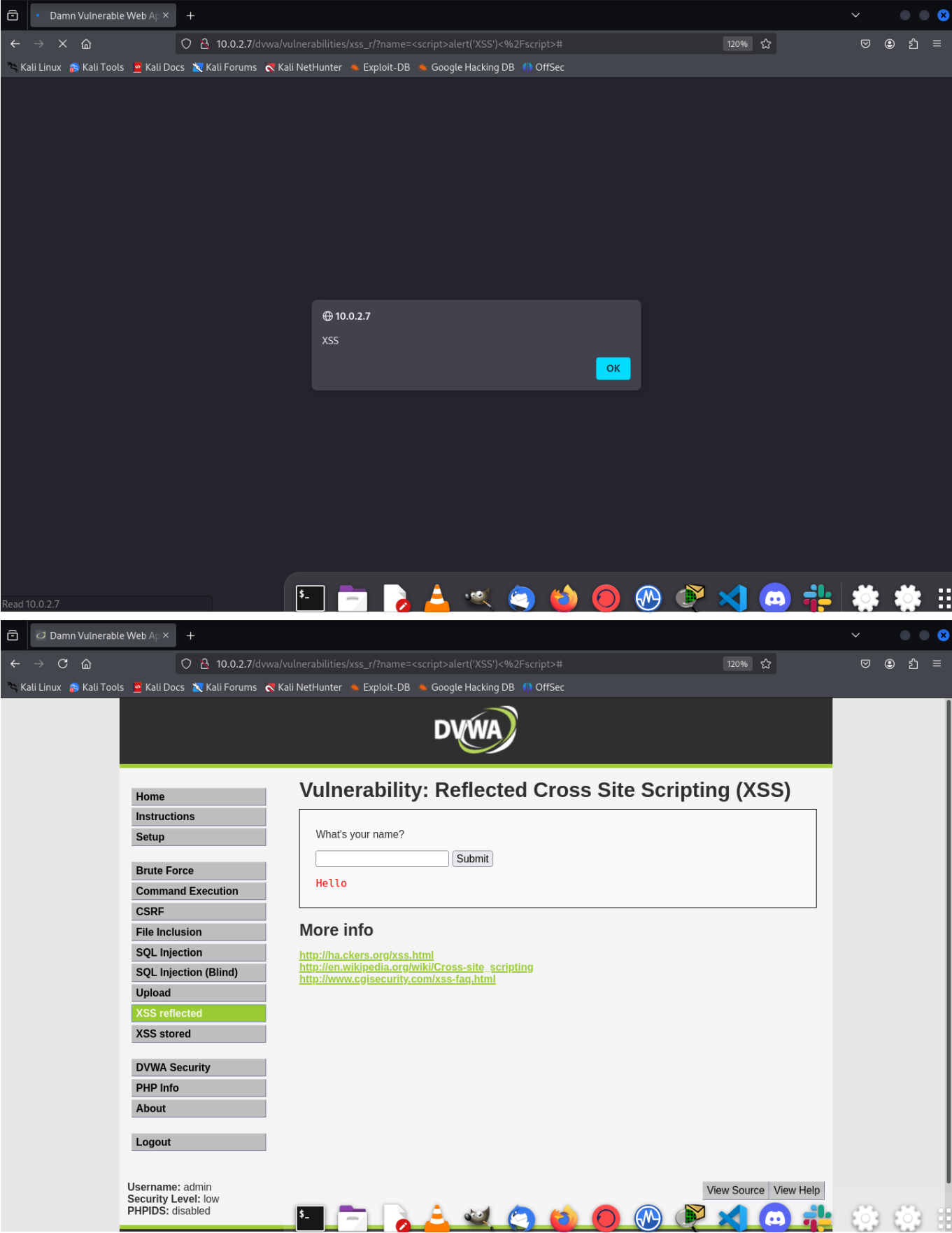
<http://hackers.org/xss.html>

http://en.wikipedia.org/wiki/Cross-site_scripting

<http://www.cgisecurity.com/xss-faq.html>

View Source

View Help



Paso 4: Analizar los paquetes capturados

The top screenshot shows the Wireshark interface with the packet list and packet details. The packet list shows a GET request to `/dwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS%27%29%3C%2Fscript%3E`. The packet details show the request URI and query parameters.

The bottom screenshot shows the Wireshark interface with the packet list and packet details. The packet list shows a GET request to `/dwa/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%27XSS%27%29%3C%2Fscript%3E`. The packet details show the request URI and query parameters.

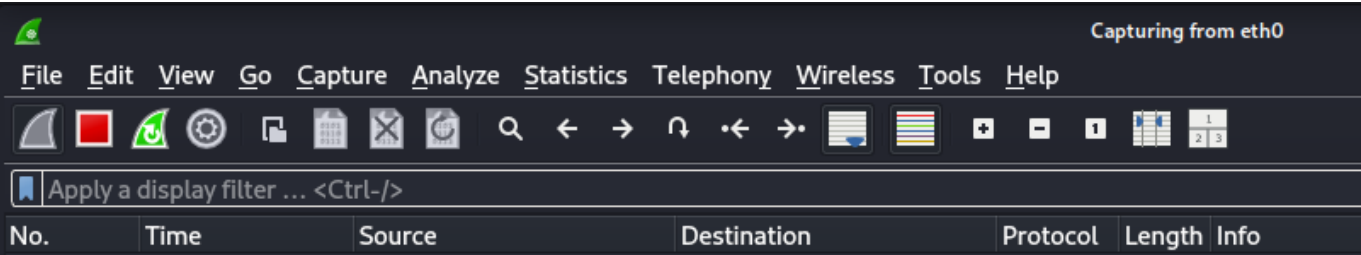
Explicación del resultado esperado:

- La solicitud HTTP contiene la carga útil de XSS en su sección de datos. Esta carga puede ser utilizada para ejecutar código JavaScript.

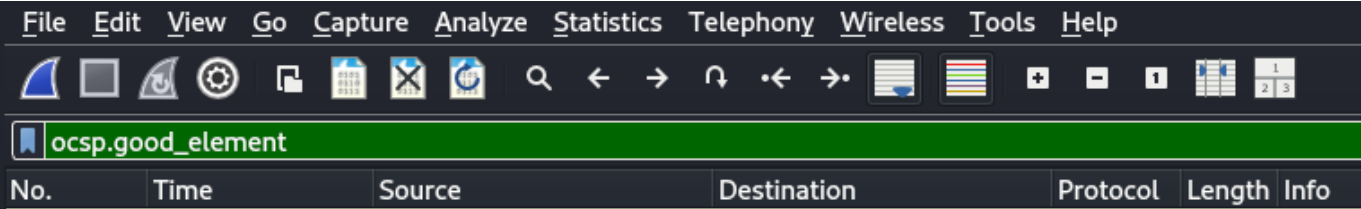
3 - Capturar y analizar la validación del certificado de Google

Instrucciones paso a paso:

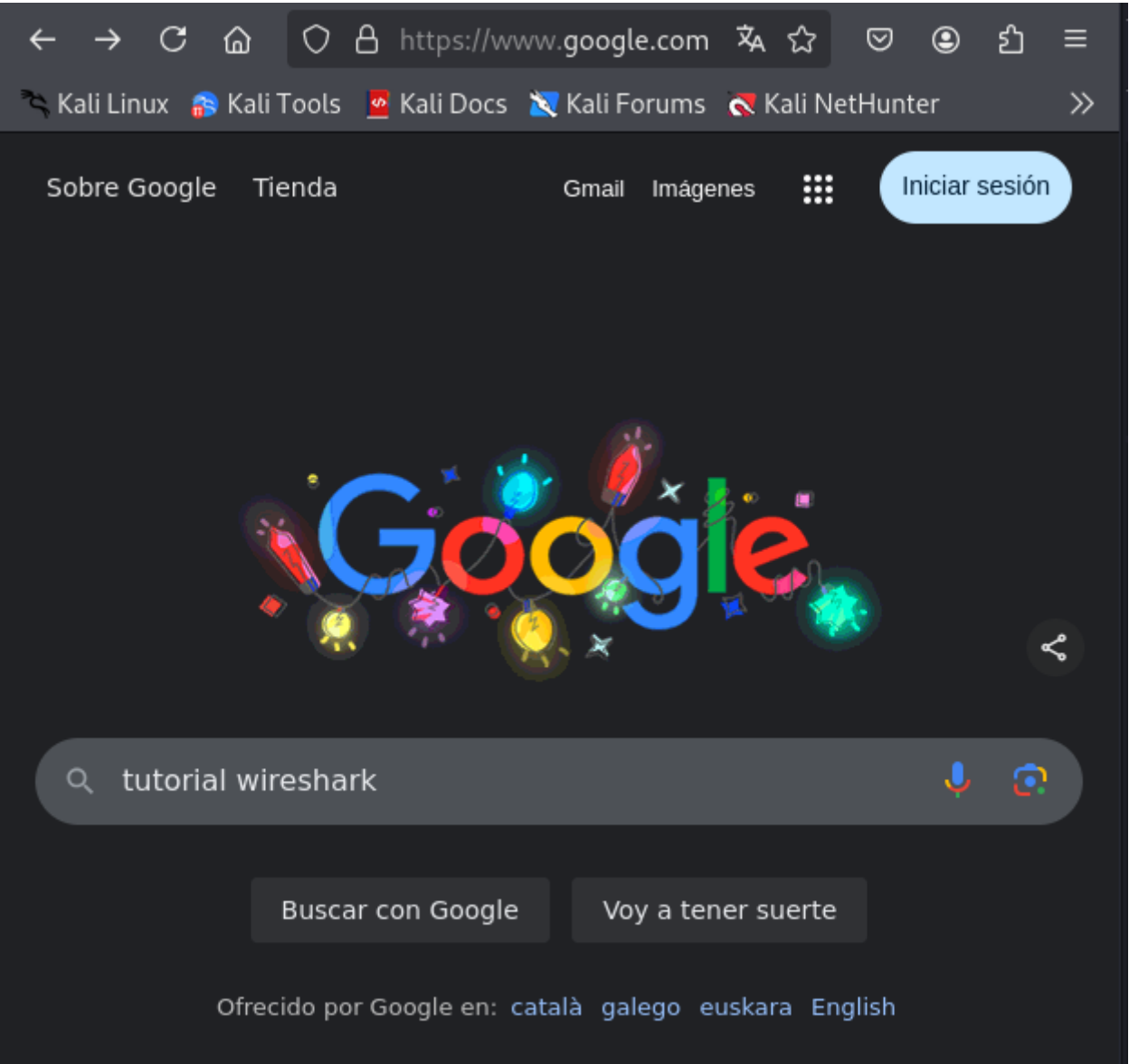
Paso 1: Configura Wireshark



Paso 2: Aplicar filtro TLS desde Kali Linux



Paso 3: Realizar una búsqueda en Google



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

oosp.good_element

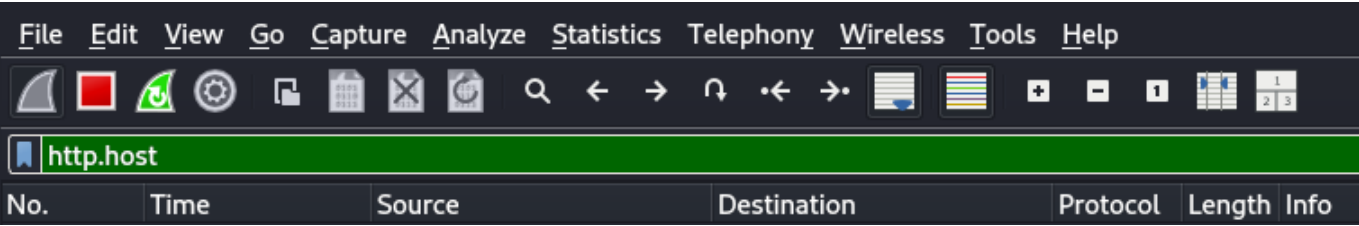
No.	Time	Source	Destination	Protocol	Length	Info
146	0.853317612	142.250.201.67	10.0.2.5	OOSP	755	Response
147	0.853317822	142.250.201.67	10.0.2.5	OOSP	755	Response
190	0.877200192	142.250.201.67	10.0.2.5	OOSP	755	Response
532	1.164746213	142.250.201.67	10.0.2.5	OOSP	756	Response
880	1.372197417	142.250.201.67	10.0.2.5	OOSP	755	Response

- ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 443
- ▶ Hypertext Transfer Protocol
- ▼ Online Certificate Status Protocol
 - responseStatus: successful (0)
 - ▼ responseBytes
 - ResponseType Id: 1.3.6.1.5.5.7.48.1.1 (id-pkcs-ocsp-response)
 - ▼ BasicOCSPResponse
 - ▼ tbsResponseData
 - ▼ responderID: byKey (2)
 - byKey: de1b1eed7915d43e3724c321bbec34f0a2f041e7b2
 - producedAt: Dec 12, 2024 14:14:35.000000Z
 - responses: 1 item
 - ▼ SingleResponse
 - certID: 1
 - certStatus: good (0)
 - good
 - thisUpdate: Dec 12, 2024 14:14:35.000000Z
 - nextUpdate: Dec 19, 2024 13:14:34.000000Z
 - signatureAlgorithm (sha256WithRSAEncryption)
 - padding: 0
 - signature [1]: 3b84c3886d348010cd1e9af0156a02f041e7b2

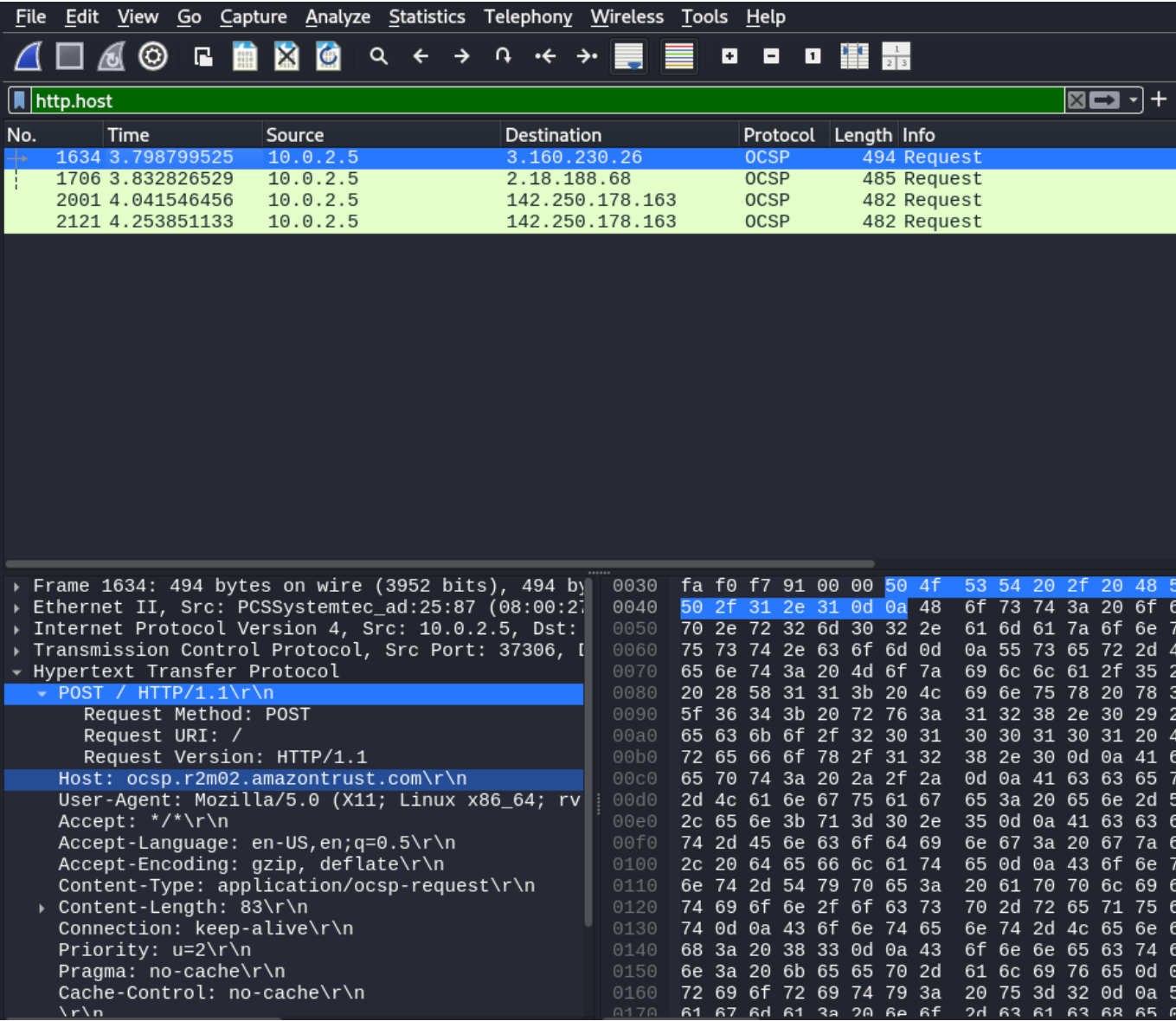
- El resultado muestra en la sección de negociación SSL/TLS, el certificado google válido con la confirmacion good, indincado a la vez informacion sobre las versiones y sus posibles vulnerabilidades.

The image shows the top portion of the Wireshark application window. At the top right, it says "Capturing from eth0". Below this is a menu bar with the following items: File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Underneath the menu bar is a toolbar containing various icons for file operations (like Open, Save, Print), navigation (Back, Forward, Home), and analysis (Packet List, Packet Details, Packet Bytes). Below the toolbar is a text input field with the placeholder text "Apply a display filter ... <Ctrl-/>". At the very bottom, the top of a table is visible, with headers: No., Time, Source, Destination, Protocol, Length, and Info.

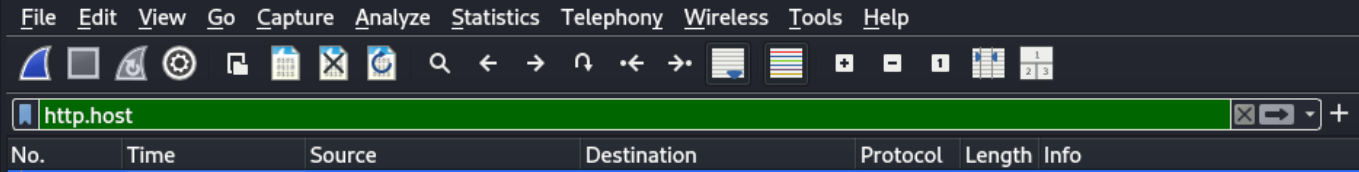
Paso 2: Aplicar filtro HTTP desde Kali Linux



Paso 3: Realizar una solicitud al sitio CocaCola



Paso 4: Analizar los paquetes capturados

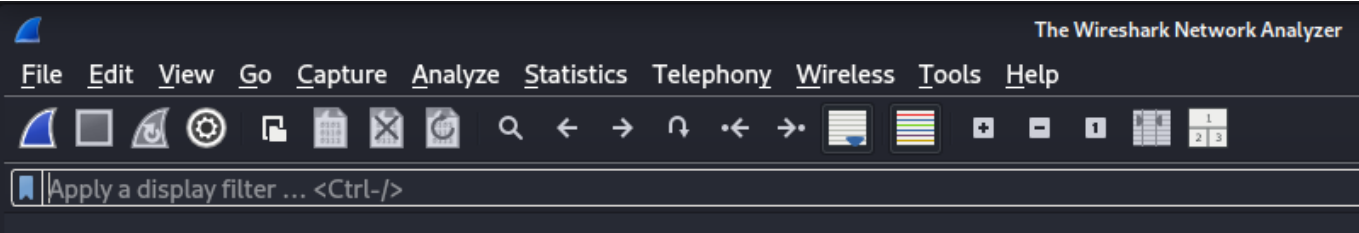


Explicación del resultado esperado:

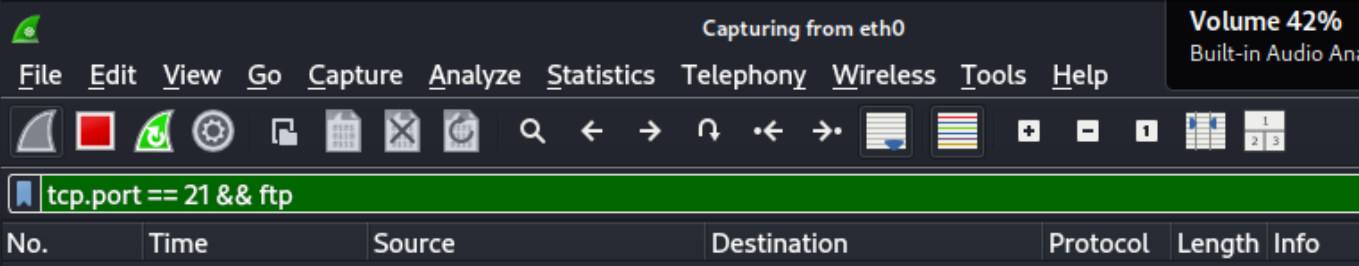
- La solicitud HTTP contiene la URL completa de la página, el metodo usado y la version de HTTP, en este caso es "POST / HTTP/1.1" y tambien se aprecia informacion sobre el Host usado en este caso es "amazontrust.com"

5 - Capturar y analizar el archivo my.cnf a través de FTP al servidor Metasploitable desde la máquina Kali Linux

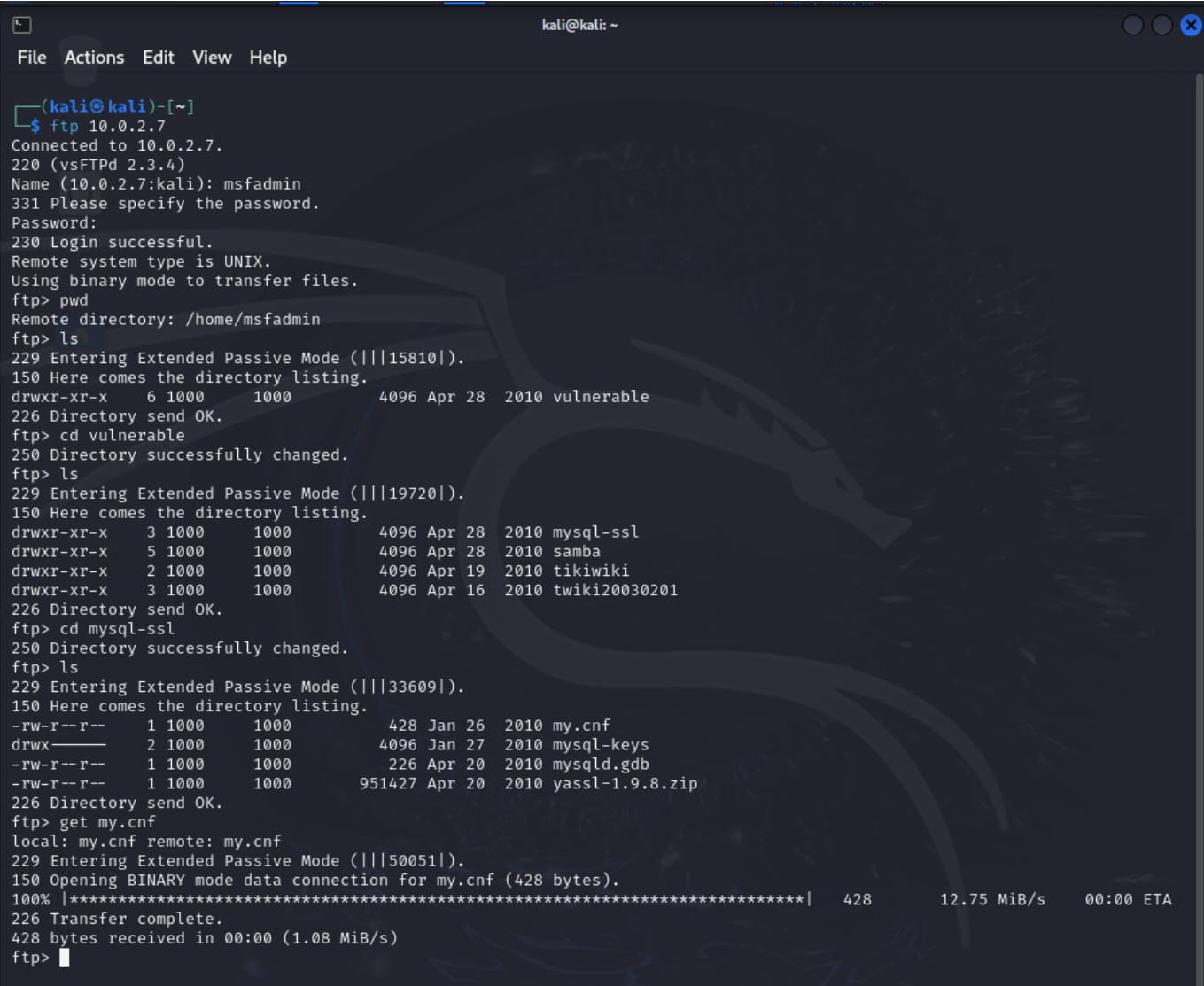
Paso 1: Configura Wireshark



Paso 2: Aplicar filtro FTP



Paso 3: Conectar via FTP al servidor Metasploitable desde Kali Linux & Paso 4: Descargar el archivo my.cnf



Paso 5: Analizar los paquetes capturados

FileEditViewGoCaptureAnalyzeStatisticsTelephonyWirelessToolsHelp

tcp.port==21&&ftp

No.	Time	Source	Destination	Protocol	Length	Info
23	7.719184953	10.0.2.7	10.0.2.5	FTP	88	Response: 257 "/home/msfadmin"
25	8.927921919	10.0.2.5	10.0.2.7	FTP	72	Request: EPSV
26	8.927554812	10.0.2.7	10.0.2.5	FTP	115	Response: 229 Entering Extended Passive Mode (23536).
31	8.927939798	10.0.2.5	10.0.2.7	FTP	72	Request: LIST
32	8.928178425	10.0.2.7	10.0.2.5	FTP	105	Response: 150 Here comes the directory listing.
34	8.928333202	10.0.2.7	10.0.2.5	FTP	90	Response: 226 Directory send OK.
40	22.226816046	10.0.2.5	10.0.2.7	FTP	92	Request: CWD vulnerable/mysql-ssl
41	22.227173354	10.0.2.7	10.0.2.5	FTP	103	Response: 250 Directory successfully changed.
43	23.546667295	10.0.2.5	10.0.2.7	FTP	72	Request: EPSV
44	23.547060589	10.0.2.7	10.0.2.5	FTP	115	Response: 229 Entering Extended Passive Mode (11640).
49	23.547399544	10.0.2.5	10.0.2.7	FTP	72	Request: LIST
50	23.547547726	10.0.2.7	10.0.2.5	FTP	105	Response: 150 Here comes the directory listing.
56	23.547831610	10.0.2.7	10.0.2.5	FTP	90	Response: 226 Directory send OK.
58	39.659112282	10.0.2.5	10.0.2.7	FTP	74	Request: TYPE I
59	39.659456328	10.0.2.7	10.0.2.5	FTP	97	Response: 200 Switching to Binary mode.
60	39.659516261	10.0.2.5	10.0.2.7	FTP	79	Request: SIZE my.cnf
61	39.659679197	10.0.2.7	10.0.2.5	FTP	75	Response: 213 428
62	39.659737696	10.0.2.5	10.0.2.7	FTP	72	Request: EPSV
63	39.659938703	10.0.2.7	10.0.2.5	FTP	115	Response: 229 Entering Extended Passive Mode (50721).
67	39.660235268	10.0.2.5	10.0.2.7	FTP	79	Request: RETR my.cnf
68	39.660544582	10.0.2.7	10.0.2.5	FTP	131	Response: 150 Opening BINARY mode data connection for my.cnf (428 bytes).
74	39.661081120	10.0.2.7	10.0.2.5	FTP	90	Response: 226 Transfer complete.
76	39.661142551	10.0.2.5	10.0.2.7	FTP	79	Request: MDTM my.cnf
77	39.661273121	10.0.2.5	10.0.2.5	FTP	86	Response: 213 20100126185452

Frame 74: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface eth0, ...

Ethernet II, Src: PCSSystemtec_9c:f0:63 (08:00:27:9c:f0:63), Dst: PCSSystemtec_ad:25:87 ...

Internet Protocol Version 4, Src: 10.0.2.7, Dst: 10.0.2.5 ...

Transmission Control Protocol, Src Port: 21, Dst Port: 34870, Seq: 621, Ack: 138, Len: ...

File Transfer Protocol (FTP) ...

226 Transfer complete.\r\n

Response code: Closing data connection (226)

Response arg: Transfer complete.

[Current working directory: /home/msfadmin/vulnerable/mysql-ssl]

0000 08 00 27 ad 25 87 08 00 27 9c f0 63 08 00 45 00 ... '%...'.c.E

0010 00 4c 52 50 40 00 40 06 d0 50 0a 00 02 07 0a 00 ... LRP@...P...

0020 02 05 00 15 88 36 b8 20 f9 07 eb 78 88 fb 80 18 ... 6...x...

0030 00 2e 38 b7 00 00 01 01 08 0a 00 00 b4 2c 57 e48...W...

0040 9c bd 32 32 36 20 54 72 61 6e 73 66 65 72 20 63 ... 226 Transfer c

0050 6f 6d 70 6c 65 74 65 2e 0d 0a ... mplete..%

Explicación del resultado esperado:

- Los paquetes de datos contienen detalles sobre el archivo **my.cnf** descargado y mas contenido relacionado con las comunicaciones FTP realizadas.

/