Práctica de Laboratorio - Atacar una base de datos MySQL

NOTA: RESPONDER CADA PREGUNTA CON CAPTURAS DE PANTALLA COMPLETAS DONDE SE VEAN LAS SIGLAS DE SU NOMBRE AL LADO DEL RELOJ. EN LUGAR DE COLOCAR UN TEXTO DEBE DE SELECCIONAR EL TEXTO INVESTIGADO EN UNA PÁGINA WEB Y HACERLE CAPTURA DE PANTALLA COMPLETA, PARA LUEGO COLOCARLA AQUÍ. SI SON VARIAS PREGUNTAS, DEBE COLOCAR VARIAS CAPTURAS DE PANTALLAS COMPLETAS. NO COLOCAR INFORMACIÓN ENCONTRADA EN NETACAD, COLOCAR INFORMACIÓN INVESTIGADA EN OTRAS PÁGINAS WEB O LO REALIZADO POR USTED.

Objetivos

En esta práctica de laboratorio verán un archivo PCAP de un ataque anterior a una base de datos SQL.

Parte 1: Abrir Wireshark y cargar el archivo PCAP.

Parte 2: Visualizar el ataque de inyección SQL.

Parte 3: El ataque de inyección SQL continúa...

Parte 4: El ataque de inyección SQL proporciona información del sistema.

Parte 5: El ataque de inyección SQL e información de tablas

Parte 6: El ataque de inyección SQL concluye.

Antecedentes / Escenario

Los ataques de inyección SQL permiten que los hackers maliciosos escriban sentencias SQL en un sitio web y reciban una respuesta de la base de datos. Esto permite que los atacantes modifiquen los datos actuales de la base de datos, suplanten identidades y ejecuten malware variado.

Hemos creado un archivo PCAP para que vean un ataque anterior a una base de datos SQL. En esta práctica de laboratorio verán los ataques a la base de datos SQL y responderán las preguntas.

Recursos necesarios

- Máquina virtual Security Workstation

Instrucciones

Utilizarán Wireshark, un analizador de paquetes de red común, para analizar el tráfico de red. Después de iniciar Wireshark, abrirán una captura de red ya guardada y verán un ataque de inyección SQL paso a paso contra una base de datos SQL.

Parte 1: Abrir Wireshark y cargar el archivo PCAP.

La aplicación Wireshark se puede abrir por medio de diversos métodos en una estación de trabajo de Linux.

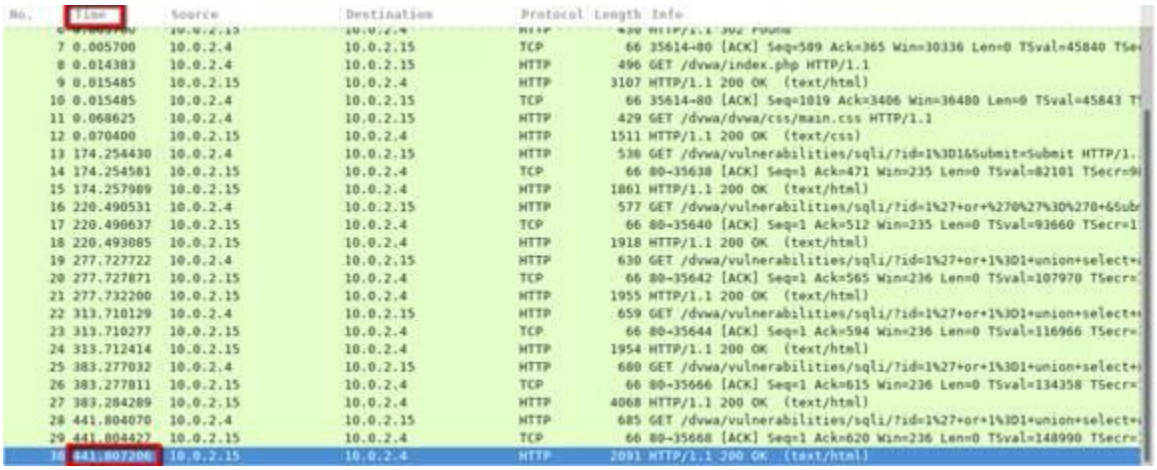
a. Inicie la máquina virtual Security Workstation.

b. En el escritorio haga clic en **Applications > CyberOPS > Wireshark** y luego busque la aplicación Wireshark.

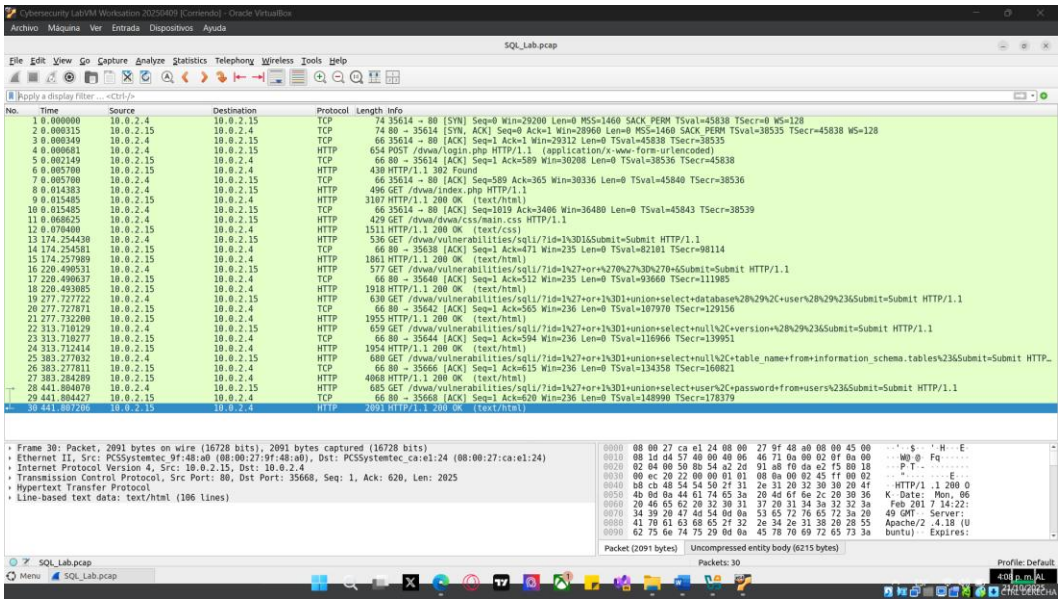
c. En la aplicación Wireshark, hagan clic en **Open** (Abrir) en el medio de la aplicación, en la sección Files (Archivos).

d. Diríjanse al directorio **/home/analyst/** y busquen **lab.support.files**. En el directorio **lab.support.files** abran el archivo **SQL_Lab.pcap**.

e. El archivo PCAP se abre dentro de Wireshark para mostrar el tráfico de red capturado. Este archivo de captura se extiende por un período de 8 minutos (441 segundos), la duración de este ataque de inyección SQL.



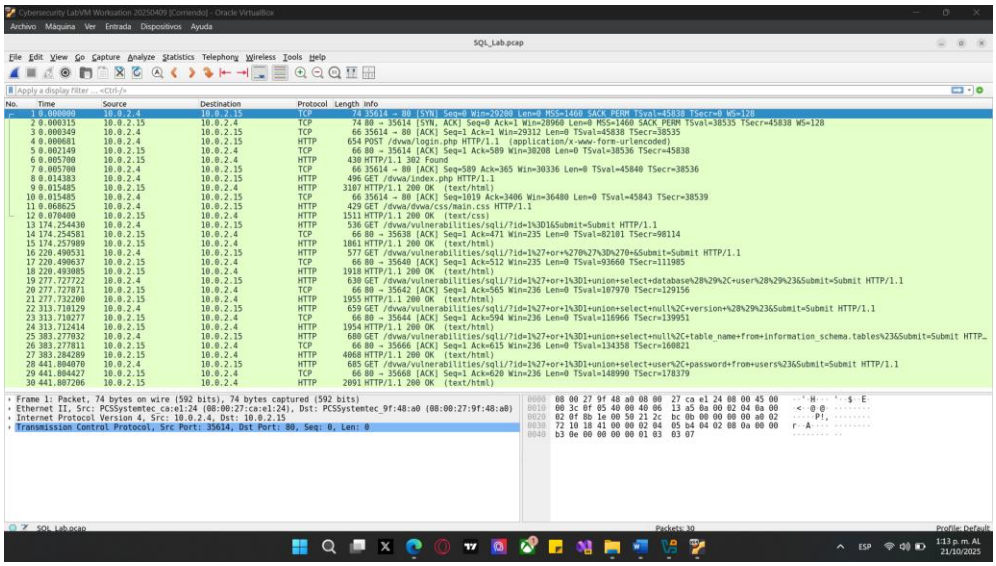
No.	Time	Source	Destination	Protocol	Length	Info
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614→80 [ACK] Seq=509 Ack=365 Win=30336 Len=0 TSval=45840 TSe
8	0.014383	10.0.2.4	10.0.2.15	HTTP	496	GET /dvwa/index.php HTTP/1.1
9	0.015485	10.0.2.15	10.0.2.4	HTTP	3107	HTTP/1.1 200 OK (text/html)
10	0.015485	10.0.2.4	10.0.2.15	TCP	66	35614→80 [ACK] Seq=1019 Ack=3406 Win=36480 Len=0 TSval=45843 T
11	0.068625	10.0.2.4	10.0.2.15	HTTP	429	GET /dvwa/dvwa/css/main.css HTTP/1.1
12	0.070400	10.0.2.15	10.0.2.4	HTTP	1511	HTTP/1.1 200 OK (text/css)
13	174.254430	10.0.2.4	10.0.2.15	HTTP	530	GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit HTTP/1.
14	174.254581	10.0.2.15	10.0.2.4	TCP	66	80→35638 [ACK] Seq=1 Ack=471 Win=235 Len=0 TSval=82101 TSecr=9
15	174.257989	10.0.2.15	10.0.2.4	HTTP	1861	HTTP/1.1 200 OK (text/html)
16	220.490531	10.0.2.4	10.0.2.15	HTTP	577	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+%270%27%3D%270+&Subr
17	220.490637	10.0.2.15	10.0.2.4	TCP	66	80→35640 [ACK] Seq=1 Ack=512 Win=235 Len=0 TSval=93660 TSecr=1
18	220.493085	10.0.2.15	10.0.2.4	HTTP	1918	HTTP/1.1 200 OK (text/html)
19	277.727722	10.0.2.4	10.0.2.15	HTTP	630	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+
20	277.727871	10.0.2.15	10.0.2.4	TCP	66	80→35642 [ACK] Seq=1 Ack=565 Win=236 Len=0 TSval=107970 TSecr=
21	277.732200	10.0.2.15	10.0.2.4	HTTP	1955	HTTP/1.1 200 OK (text/html)
22	313.710129	10.0.2.4	10.0.2.15	HTTP	659	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+
23	313.710277	10.0.2.15	10.0.2.4	TCP	66	80→35644 [ACK] Seq=1 Ack=594 Win=236 Len=0 TSval=116966 TSecr=
24	313.712414	10.0.2.15	10.0.2.4	HTTP	1954	HTTP/1.1 200 OK (text/html)
25	383.277032	10.0.2.4	10.0.2.15	HTTP	680	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+
26	383.277811	10.0.2.15	10.0.2.4	TCP	66	80→35666 [ACK] Seq=1 Ack=615 Win=236 Len=0 TSval=134358 TSecr=
27	383.284289	10.0.2.15	10.0.2.4	HTTP	4068	HTTP/1.1 200 OK (text/html)
28	441.804070	10.0.2.4	10.0.2.15	HTTP	685	GET /dvwa/vulnerabilities/sqli/?id=1%27+or+1%3D1+union+select+
29	441.804422	10.0.2.15	10.0.2.4	TCP	66	80→35668 [ACK] Seq=1 Ack=620 Win=236 Len=0 TSval=148990 TSecr=
30	441.807206	10.0.2.15	10.0.2.4	HTTP	2091	HTTP/1.1 200 OK (text/html)



En función de la información que aparece en pantalla, ¿cuáles son las dos direcciones IP involucradas en este ataque de inyección SQL?

Sale el tiempo, fuente, el destino, protocolo, la longitud e información

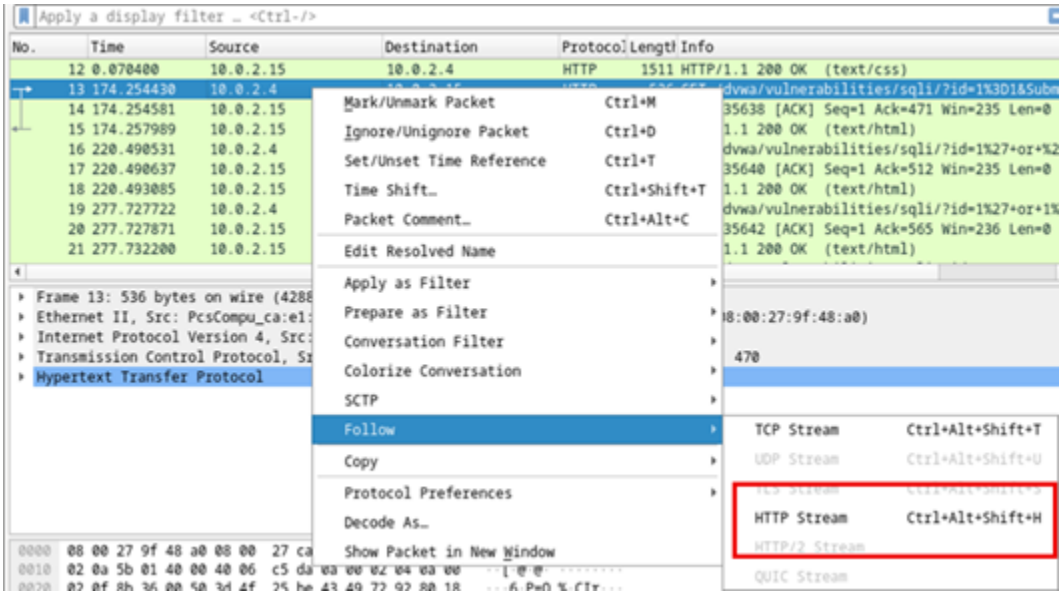
Las direcciones ip involucradas 10.0.2.4 10.0.2.15



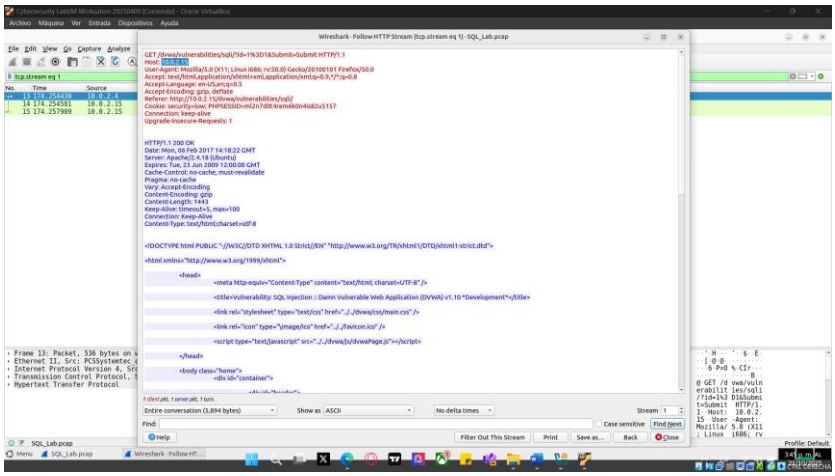
Parte 2: Ver el ataque de inyección SQL

En este paso visualizarán el comienzo de un ataque.

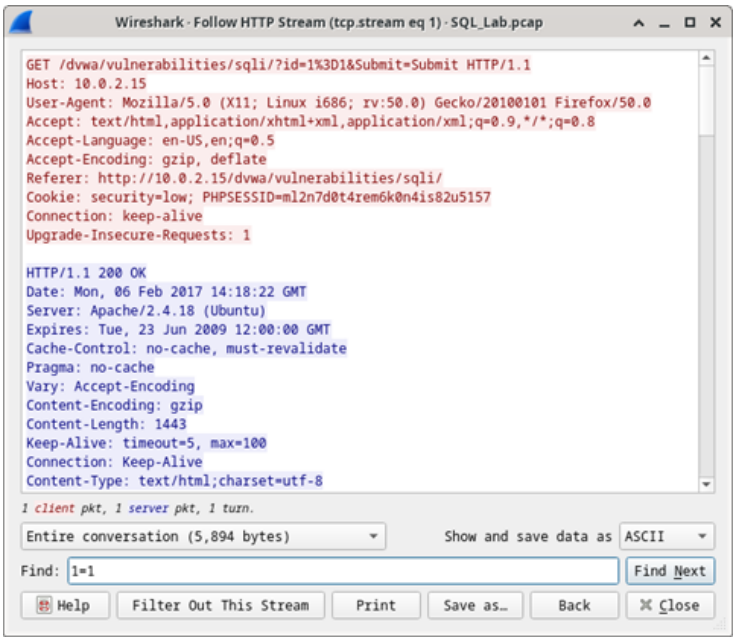
a. Dentro de la captura de Wireshark, haga clic derecho en la línea 13 y seleccione **Follow** > **HTTP Stream**. Se eligió la línea 13 porque es una solicitud GET HTTP. Esto será muy útil para seguir el flujo de datos a medida que lo ven las capas de aplicación y se genera una prueba de consulta para la inyección SQL.



El tráfico de origen se muestra en rojo. El origen ha enviado una solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.

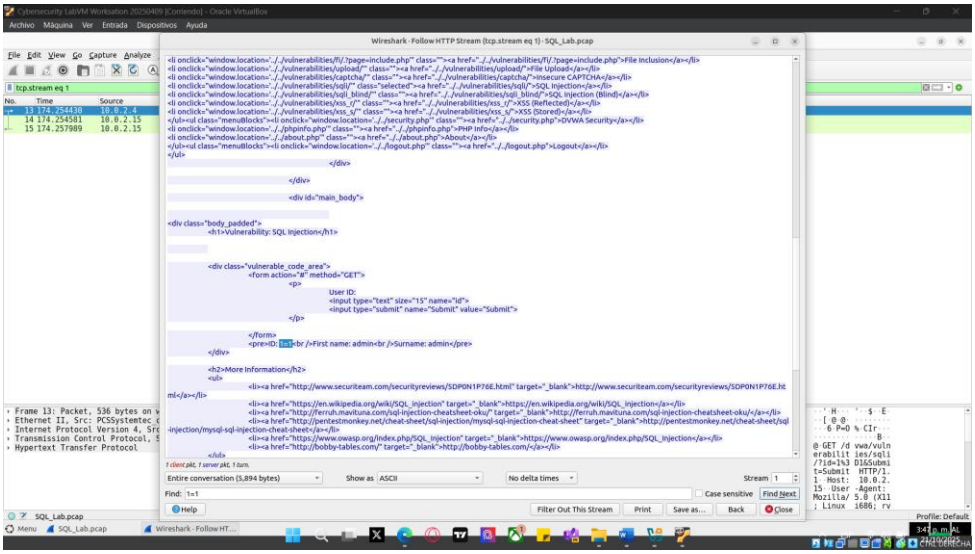


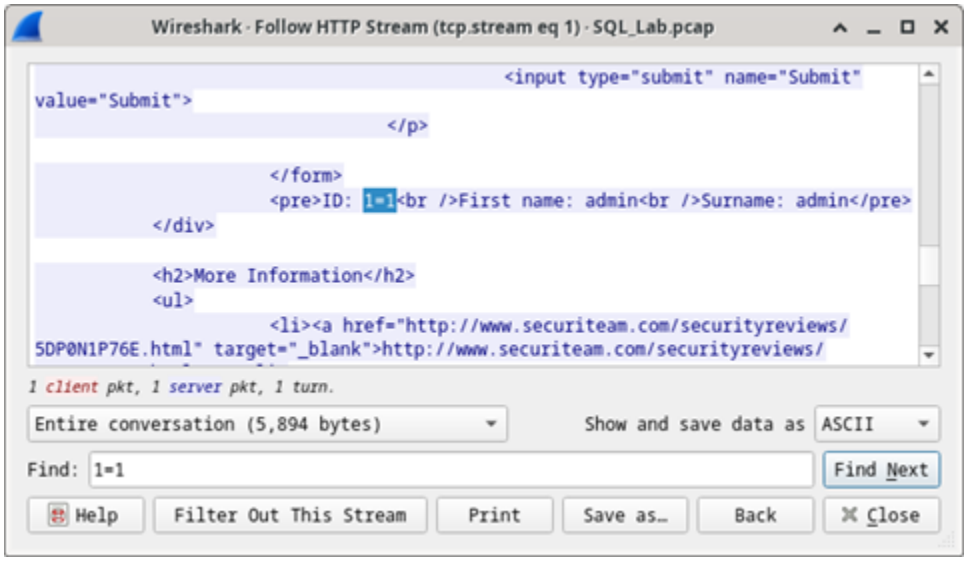
b. En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.



c. El atacante ha ingresado una consulta (1=1) en un cuadro de búsqueda de UserID en el destino 10.0.2.15 para ver si la aplicación es vulnerable a la inyección SQL. En lugar de responder con un mensaje de falla en el inicio de sesión, la aplicación respondió con un registro de la base de datos. El atacante ha verificado que puede ingresar un comando SQL y que la base de datos le responderá. La

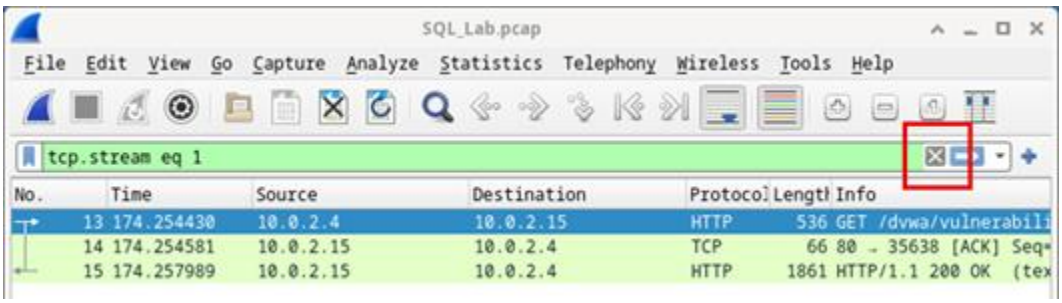
cadena de búsqueda 1=1 crea una sentencia SQL que siempre será verdadera. En el ejemplo no importa lo que se haya ingresado en el campo, siempre será verdadera.





d. Cierren la ventana Follow HTTP Stream.

e. Haga clic en **Clear display filter** para mostrar toda la conversación de Wireshark.



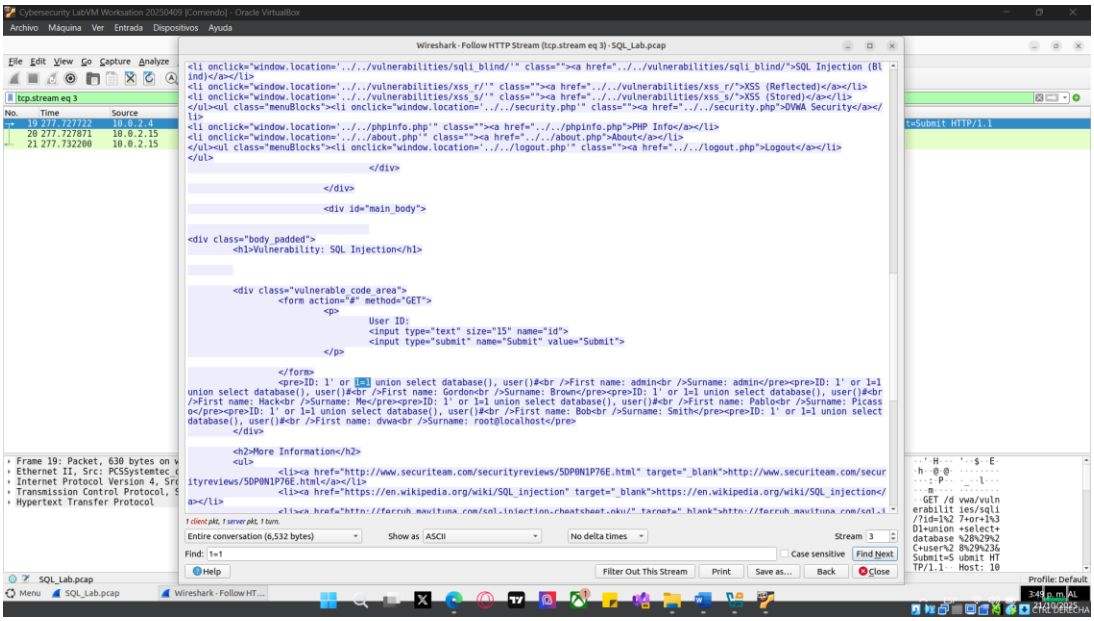
Parte 3: El ataque de inyección SQL continúa...

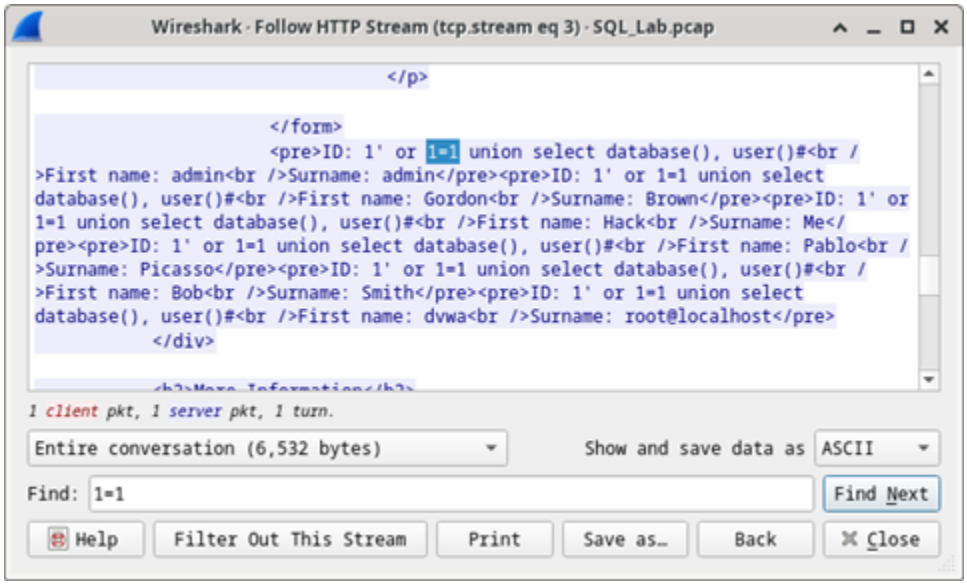
En este paso visualizarán cómo prosigue un ataque.

a. Dentro de la captura de Wireshark, haga clic derecho en la línea 19, y luego haga clic en **Follow** > **HTTP Stream**.

b. En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.

c. El atacante ha ingresado una consulta (1' or 1=1 union select database(), user()#) en un cuadro de búsqueda de ID de usuario en el destino 10.0.2.15. En lugar de responder con un mensaje de falla en el inicio de sesión, la aplicación respondió con la siguiente información:





El nombre de la base de datos es **dvwa** y su respectivo usuario es **root@localhost**. También se muestran varias cuentas de usuario.

d. Cierren la ventana Follow HTTP Stream.

e. Haga clic en **Clear display filter** para mostrar toda la conversación de Wireshark.

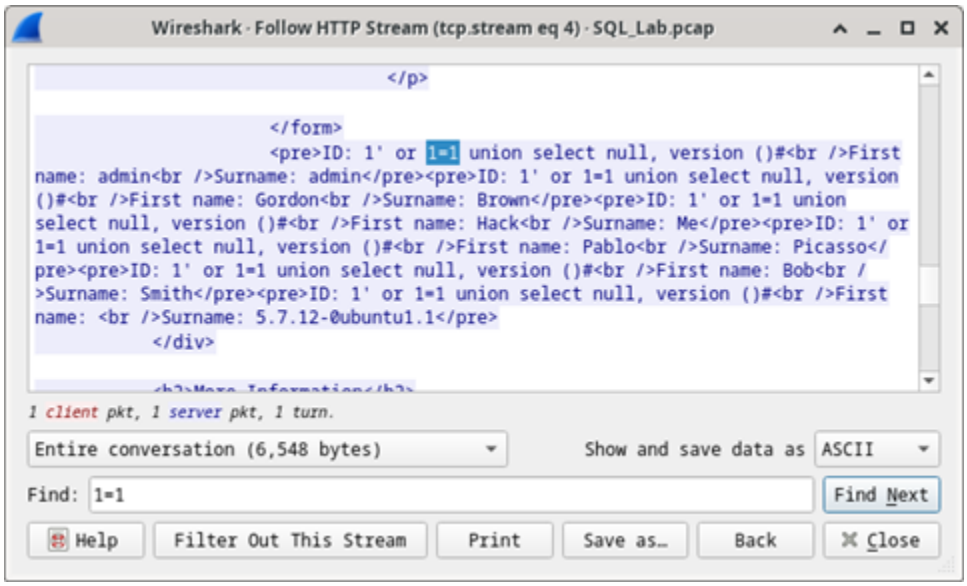
Parte 4: El ataque de inyección SQL proporciona información del sistema.

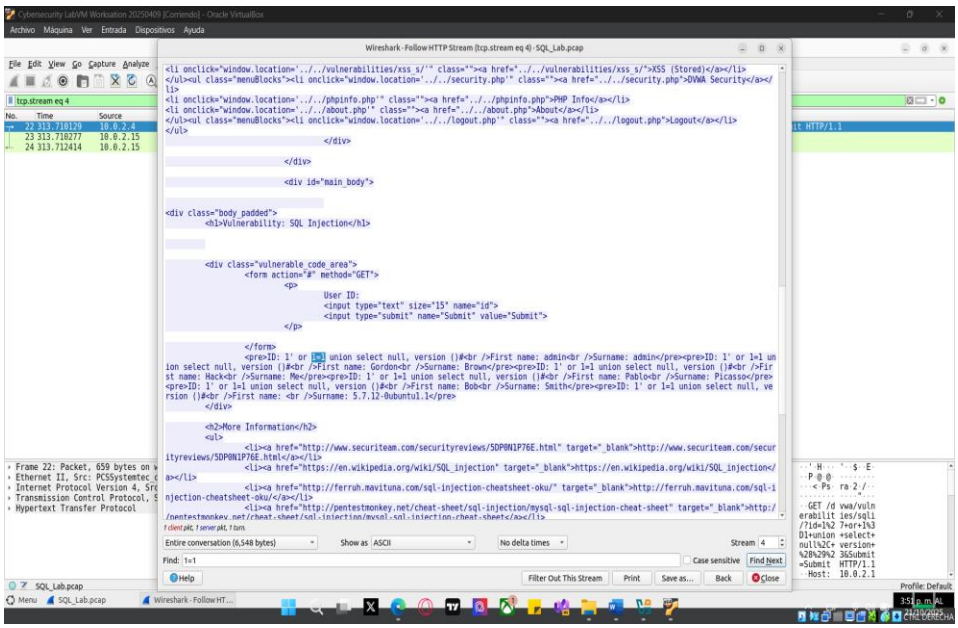
El atacante prosigue y comienza a buscar información más específica.

a. Dentro de la captura de Wireshark, haga clic derecho en la línea 22 y seleccione **Follow** > **HTTP Stream**. El tráfico de origen se muestra en rojo, y está enviando la solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.

b. En el apartado **Find**, escriba **1=1**. Haga clic en **Find Next**.

c. El atacante ha ingresado una consulta (1’ or 1=1 union select null, version()#) en un cuadro de búsqueda de ID de usuario en el destino 10.0.2.15 para localizar el identificador de la versión. Observe que es el identificador de versión se encuentra al final del resultado justo antes de</pre>. cierre del código HTML.





¿Cuál es la versión?

0ubuntu1.1

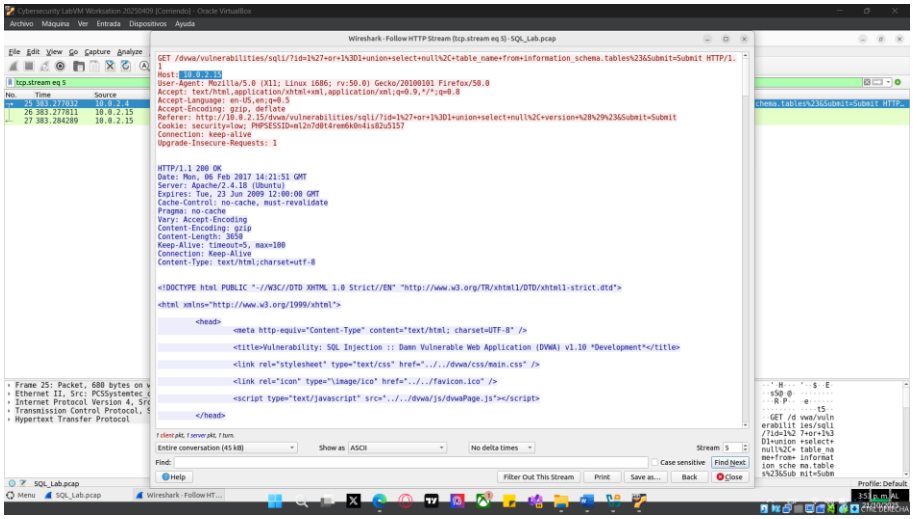
d. Cierren la ventana Follow HTTP Stream.

e. Haga clic en **Clear display filter** para mostrar toda la conversación de Wireshark.

Parte 5: El ataque de inyección SQL e información de tablas

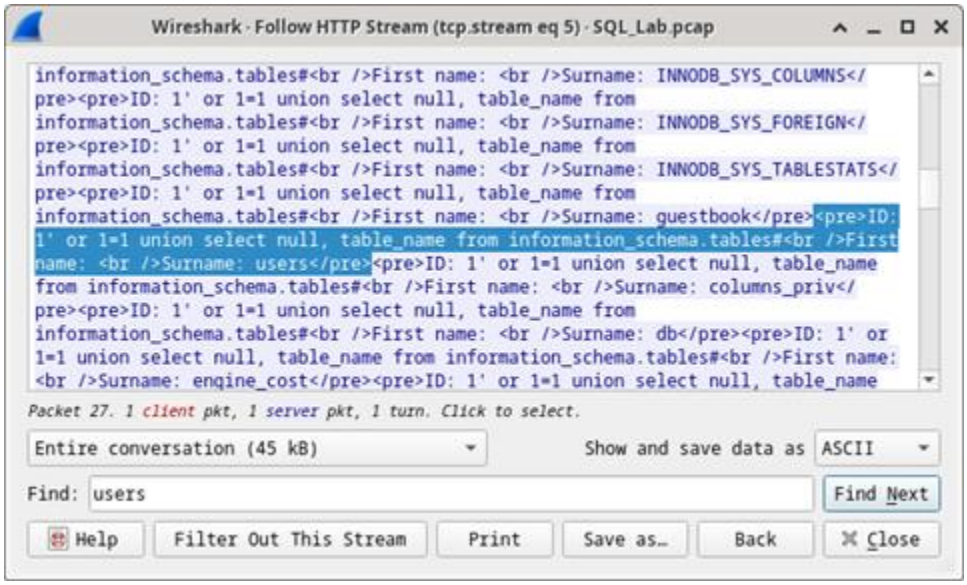
El atacante sabe que hay gran cantidad de tablas SQL repletas de información. Trata de encontrarlas.

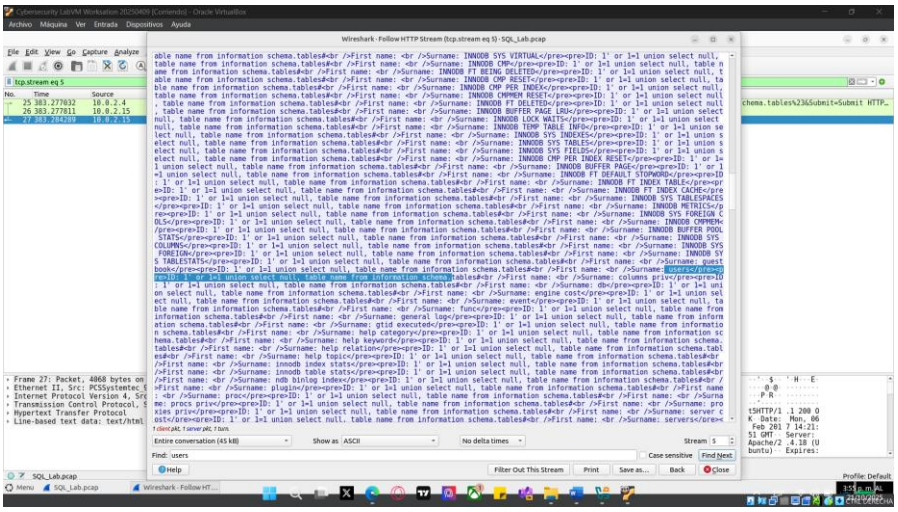
a. Dentro de la captura de Wireshark, haga clic derecho en la línea 25, y luego seleccione **Follow** > **HTTP Stream**. El origen se muestra en rojo. Ha enviado una solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.



b. En el apartado **Find**, escriba **users**. Haga clic en **Find Next**.

c. El atacante ha ingresado una consulta (1' or 1=1 union select null, table_name from information schema.tables#) en un cuadro de búsqueda de ID de usuario en el destino 10.0.2.15 para ver todas las tablas de la base de datos. Esto proporciona una enorme salida de muchas tablas, ya que el atacante especificó "null" sin más especificaciones.





¿Qué haría el comando modificado por el atacante: (**1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users'**)?

ver todas las tablas de la base de datos. Esto proporciona una enorme salida de muchas tablas, ya que el atacante especificó “null” sin más especificaciones.

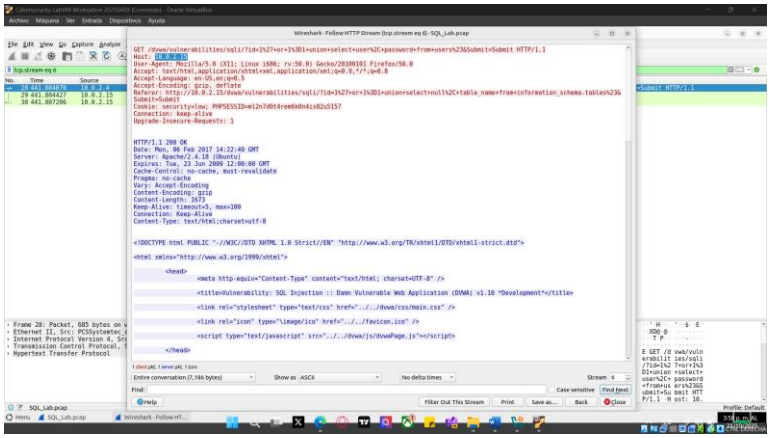
d. Cierren la ventana Follow HTTP Stream.

e. Haga clic en **Clear display filter** para mostrar toda la conversación de Wireshark.

Parte 6: El ataque de inyección SQL concluye

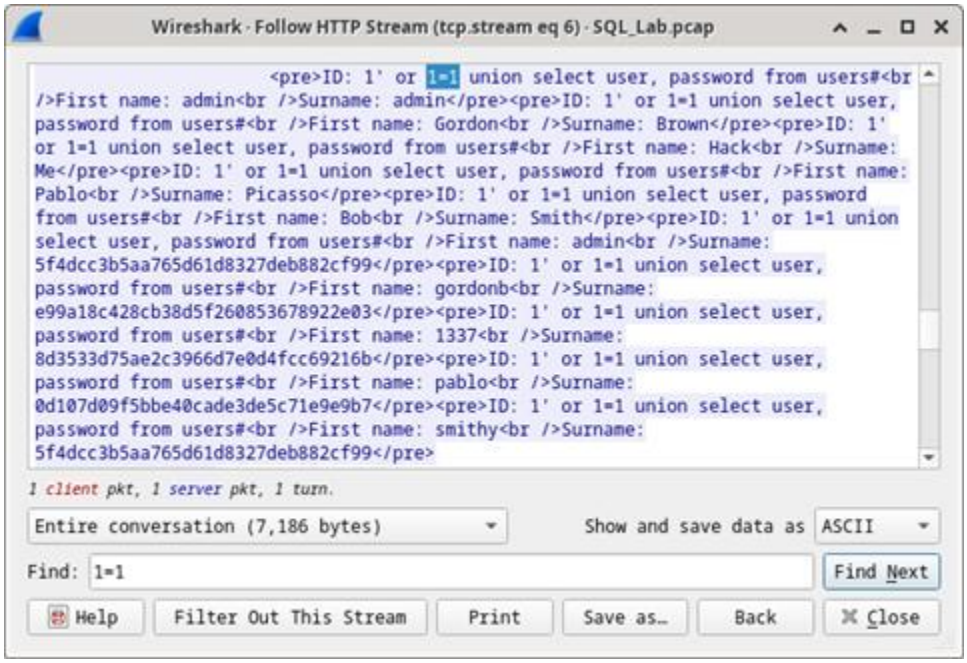
El ataque finaliza con el mejor premio posible: hashes de contraseñas.

a. Dentro de la captura de Wireshark, haga clic derecho en la línea 28 y luego seleccione **Follow** > **HTTP Stream**. El origen se muestra en rojo. Ha enviado una solicitud GET al host 10.0.2.15. En color azul, el dispositivo de destino le está respondiendo al origen.

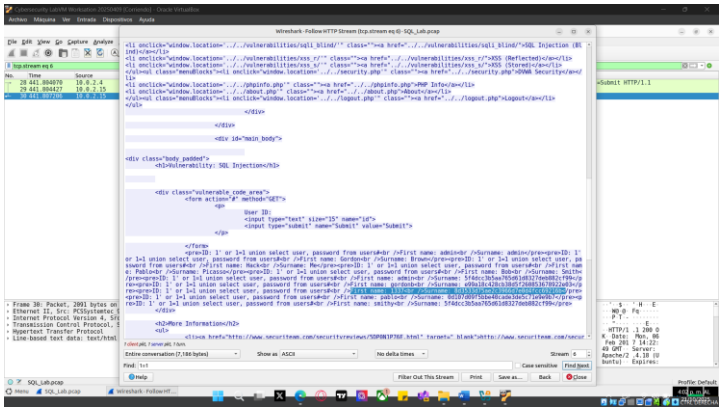


b. Hagan clic en **Find** (Buscar) e introduzcan **1=1**. Busquen esta entrada. Cuando se encuentre el texto, hagan clic en **Cancel** (Cancelar) en el cuadro de búsqueda de texto Find.

¡El atacante ha ingresado una consulta (1' or 1=1 union select user, password from users#) en un cuadro de búsqueda de ID de usuario en el destino 10.0.2.15 para obtener nombres de usuario y hashes de contraseñas!



¿Qué usuario tiene "8d3533d75ae2c3966d7e0d4fcc69216b" como hash de su contraseña?



c. Utilicen un sitio web como https://crackstation.net/ para copiar el hash de la contraseña en el decodificador de hashes de contraseñas y comiencen a decodificarlo.

¿Cuál es la contraseña en texto plano (plain-text)?

Charley

d. Cierren la ventana Follow HTTP Stream. Cierren todas las ventanas abiertas.

Preguntas de reflexión

1. ¿Cuál es el riesgo de hacer que las plataformas utilicen el lenguaje SQL?

Si la aplicación mezcla datos de usuarios con código SQL, un atacante puede "inyectar" comandos y robar, modificar o borrar información, incluso acceder a cuentas administrativas.

2. Realice una búsqueda en internet sobre "Evitar ataques de inyección SQL". ¿Cuáles son 2 métodos o pasos que se pueden utilizar para evitar ataques de inyección SQL?

Consultas parametrizadas: separar los datos del código SQL para que lo que escriba el usuario no se ejecute como comando.

Validar entradas y limitar permisos: permitir solo datos correctos (tipo, longitud, formato) y que la base de datos tenga solo los permisos necesarios.