Providence Memorial Park Antipolo

Marcos Highway 1870 Antipolo, Philippines
Email Address: Interment.pmpa@gmail.com
Contact Number: 0919-0638-018

TRANSFER OF RIGHT REQUEST

Details:

Transferor Name:	Zenith E Oliveros
Transferee Name:	Zenith E Oliveros
Location:	63
Type of Lot:	Court (8 Lots)
Date of Transfer:	2024-12-21
Day of Transfer:	Saturday
Time of Transfer:	10:00 AM
Payment Option:	Gcash

Payment Details:

Transfer Fee:	₱ 3,100.99
Notarial Fee:	₱ 250.99
Total Price:	₱ 3,351.98

REQUIREMENTS TO BRING AT THE DATE OF TRANSFER: 2024-12-21

TRANSFEROR/LOT OWNER
1. VALID ID WITH CLEAR SIGNATURE
- 3 COPIES WITH 3 SPECIMEN SIGNATURE

- IF MARRIED NEED VALID ID OF SPOUSE
- 3 COPIES WITH 3 SPECIMEN SIGNATURE
- MARRIAGE CONTRACT (PHOTO COPY)

IF SINGLE NEED BIRTH CERTIFICATE (PHOTO COPY)
- IF WIDOW NEED (CERTIFIED TRUE COPY OF DEATH CERTIFICATE)
- IF LOT OWNER DECEASED NEED CERTIFIED TRUE COPY OF DEATH CERTIFICATE

3. NOTARIZED DEED OF DEED OF RIGHTS
4. NOTARIZED JOINT AFFFIDAVIT OF CONFORMITY
5. SURRENDER ORIGINAL CERTIFICATE OF OWNERSHIP OR TITLE

TRANSFEREE
1. VALID ID WITH CLEAR SIGNATURE
- 3 COPIES WITH 3 SPECIMEN SIGNATURE

- IF MARRIED NEED VALID ID OF SPOUSE
- 3 COPIES WITH 3 SPECIMEN SIGNATURE
- MARRIAGE CONTRACT (PHOTO COPY)

- IF SINGLE NEED BIRTH CERTIFICATE (PHOTO COPY)
- IF WIDOW NEED (CERTIFIED TRUE COPY OF DEATH CERTIFICATE)
- IF LOT OWNER DECEASED NEED CERTIFIED TRUE COPY OF DEATH CERTIFICATE

SR. STO. NIÑO DE CEBU
RESOURCES & DEVELOPMENT CORPORATION

LEGAL DOCUMENTATION DIVISION

ANTIPOLO BRANCH

OFFICIAL REQUEST FORM

Date:	**December, 17, 2024**	Reference no:	**335**
Name:	**Zenith E Oliveros**	Civil Status:	**Married**
Address:	**San Jose**		
Contact No.:	**09284948360**	Email:	**nickoleibautista@gmail.com**
Project:	**VCE - PROVIDENCE MEMORIAL PARK ANTIPOLO**		
Block:	**Court of Serenity - Section - 4 - H - 111 to COS-Section4-G109**	Lot ID:	**63**

A Request for

TRANSFER OF RIGHTS

from: ***Zenith E Oliveros***

to: ***Zenith E Oliveros***

Requested By:

Zenith E Oliveros

(Buyer/Authorized Representative's Signature Over Printed Name)

This request Shall be subject for approval. Request Shall not be processed unless requirements are complete.

Verified by:	Endorsed by:	Recommended by:	Approved by:
LDA/LDS	BSM	LDO	LDM
Date:	Date:	Date:	Date:

NOTES: (to be filled up by **LDD** only)

AFFIDAVIT OF UNDERTAKING

I, Zenith E Oliveros, of legal age,Filipino citizen, with residential and postal address at San Jose and Zenith E Oliverosof legal age, Filipino citizen, with residential and postal address at San Jose,under oath, deposes and state, that:

1. That I purchased from Sr. Sto. Nino de Cebu Resources and Development Corporation (the 'Company') a parcel of land with house thereon at PROVIDENCE MEMORIAL PARK ANTIPOLO particularly Court of Serenity - Section - 4 - H - 111 to COS-Section4-G109 with OTP# 335 (the Subject 'Property');

2. As part of this transaction, I have provided contact information, including but not lomited to email addresses and phone numbers to wit:

Email Address: nickoleibautista@gmail.com
Contact Number: 09284948360

3. That I acknowledge and agree that the contact information provided to the Company will be used solely for the purpose of sending notices and any other correspondence related to the Property. This includesm but is not limited to, updates, maintenance notifications, payment reminders, legal notices, and
any other communication/request deemed necessary but the Company in connection with the Property.

4. That I acknowledge that all notices/reminders sent by the contact information provided are deemed received.

5. That Furthermore, I acknowledge and agree that the contact informaiton provided to the Company may be used to send request related to the Property. This inclides, but is not limited to, Move-in Request,
Construction Request, Refund Request, Transfer of Rights, Change of Name, Transfer of Lot, and any other communication deemed necessary to the Company in connection with the Property.

6. That I acknowledge that all request sent by the contact information provided are binding.

7. That I confirm that the contact information provided is accurate and up-to-date. I agree to notify the Company promptly in writing of any changes to the contact information to ensure continuous and accurate communication.

9. That I consent to receiving communications form the Company through carious methids, including but
not limited to email, phone calls, and text messages. That I acknowledge that electronic communications may be subkect to risks associated wih electronic transmission, including but not limited to unauthorized access, system failures, and transmission errors.

10. This Undertaking shall remain in effect for the duration of the ownership of the Property or until such
time as the I provide written notice to the Company requesting the cessation of such communications.

11. Finally, I have read and fully understood the contents of this Undertaking and that I have voluntarily affixed my signature above my printed name to confirm all matters stated herein.

In WITNESS HEREOF, I/We have to hereunto set our hande at _____,Philippines, on this _____

Zenith E Oliveros
Affiant

SIGNED IN THE PRESENCE OF:

_____ _____

REPUBLIC OF THE PHILIPPINES)_____)SS

BEFORE ME, a Notary Public for and in _____this day of _____.

Personally Appeared:	ID.No/CTC No.:	Date & Place Issued
Zenith E Oliveros	**LICENSE ID: D99-999-99-99**	**PHILIPPINES**

Known to me and to known to be the same persons who executed the foregoing instrument and they acknowleged to me that the same area their own free voluntary act and deed.

WITNESS HAND AND SEAL

Doc. No._____;
Page No._____;
Book. No._____;
Series of._____;
Republic of the Philipppines
City of _____)S.S.

Notarial Seal

DEED OF TRANSFER OF RIGHTS

KNOW ALL MEN BY THESE PRESENTS:

That I, Zenith E Oliveros, of legal age, Filipino citizen, married to Zenith E Oliveros with residential and postal address at San Jose, herein after referred to as the TRANSFEROR

-and-

Zenith E Oliveros of legal age, Filipino citizen, married to Zenith E Oliveros with residential and postal address atSan Jose herein after referred to as the TRANSFEREE.

For and in consideration of Ph 150,000 (ONE HUNDRED FIFTY THOUSAND PESOS ONLY) Total Contract Price and Memorial Maintenance Fund to me in hand paid in fully by TRANSFEREE, do hereby SELL, TRANSFER, AND CONVEY all my rights and interest in the purchaser of Memorial Lot particularly Court of Serenity - Section - 4 - H - 111 to COS-Section4-G109 at Providence Memorial Park , Brgy. Inarawan, Antipolo City, to the said TRANSFEREE, specified in Contract No.63, entered into by me and the Memorial Park owner.

That upon signing of this instrument TRANSFEREE shall be directly responsible for all instrument due payable to the memorial park owner and shall comply with all obligations pertaining to me and as stipulated in said Contract No. 63 and the stipulation of the Reservation Application when not contrary.

IN WITNESS WHEREOF, we have hereunto sign this _____ day of _____at _____ City

Zenith E Oliveros	Zenith E Oliveros
(Transferor)	(Transferee)

Zenith E Oliveros

Transferor-Spouse

ACKNOWLEDGMENT (REPUBLIC OF THE PHILIPPINES)SS
SIGNED IN PRESENCE OF (REPUBLIC OF THE PHILIPPINES)SS

BEFORE ME, a Notary Public for and in _____this _____day of _____ personally appeared:

Personally Appeared:	ID.No/CTC No.:	Date & Place Issued
Zenith E Oliveros	**LICENSE ID: D99-999-99-99**	**PHILIPPINES**
Zenith E Oliveros	**LICENSE ID: R99-55-666**	**PHILIPPINES**

All known to me and to known to be the same persons who executed the foregoing instrument and they acknowledged to me that the same are their own free voluntary act and deed.

WITNESS HAND AND SEAL

Doc. No._____;
Page No._____;
Book. No._____;
Series of._____;
Republic of the Philipppines
City of _____)S.S.

Notarial Seal

JOINT AFFIDAVIT OF CONFORMITY

We, Zenith E Oliveros, of legal age,Filipino citizen, married to Zenith E Oliveros with residential and postal address at San JoseandZenith E Oliverosof legal age, Filipino citizen,married to Zenith E Oliveros with residential and postal address atSan Jose,under oath, deposes and state, that:

That this Joint Affidavit refer to a Court (8 Lots) designated as Court of Serenity - Section - 4 - H - 111 to COS-Section4-G109 located at Brgy. Inarawan, Antipolo City consisting of 1 X 2.5 square meters (the
Property) known as PROVIDENCE MEMORIAL PARK - ANTIPOLO developed by Sr. Sto. Nino De Cebu Resources and Development Corporation (SNRDC)(the Developer);

That we jointly and severally undertake to pay the Capital Gains Tax and other taxes that the Government may require due to the transfer of any rights and obligations arising from this transaction;

That we will hold the Sr. Sto. Nino De Cebu Resources and Development Corporation (SNRDC) free and clear of any harm, liability, damage, or cost arising from any action, whether directly or indirectly, taken upon or as a consequence of my execution of this Affidavit;

That we shall be held personally liable to any person, natural or juridical, that may be prejudiced by my representation, in addition to other liabilities, civil or criminal, that may arise therefrom; hereby releasing and discharging the Sr. Sto. Nino De Cebu Resources and Development Corporation (SNRDC) from any and all further obligations in connection with the above.

That we execute this Affidavit freely and voluntarily to attest to the truth of all the foregoing for whatever legal purpose this may serve.

IN WITNESS WHEREOF, I have hereunto set my hand this ____ day of _____, _____ at_____, Philippines.

Zenith E Oliveros
(Affiant)

Zenith E Oliveros
(Affiant)

Zenith E Oliveros
Affiant-Spouse

ACKNOWLEDGMENT (REPUBLIC OF THE PHILIPPINES)SS

BEFORE ME, a Notary Public for and in _____this _____day of _____ personally appeared:

Personally Appeared:	ID.No/CTC No.:	Date & Place Issued
Zenith E Oliveros	**LICENSE ID: D99-999-99-99**	**PHILIPPINES**
Zenith E Oliveros	**LICENSE ID: R99-55-666**	**PHILIPPINES**

All known to me and to known to be the same persons who executed the foregoing instrument and they acknowledged to me that the same are their own free voluntary act and deed.

WITNESS HAND AND SEAL

Doc. No._____;
Page No._____;
Book. No._____;
Series of._____;
Republic of the Philipppines
City of _____)S.S.

Notarial Seal

Lab - Attacking a mySQL Database

Objectives

In this lab, you will view a PCAP file from a previous attack against a SQL database.

Part 1: Open Wireshark and load the PCAP file.

Part 2: View the SQL Injection Attack.

Part 3: The SQL Injection Attack continues…

Part 4: The SQL Injection Attack provides system information.

Part 5: The SQL Injection Attack and Table Information

Part 6: The SQL Injection Attack Concludes.

Background / Scenario

SQL injection attacks allow malicious hackers to type SQL statements in a web site and receive a response from the database. This allows attackers to tamper with current data in the database, spoof identities, and miscellaneous mischief.

A PCAP file has been created for you to view a previous attack against a SQL database. In this lab, you will view the SQL database attacks and answer the questions.

Required Resources

- CyberOps Workstation virtual machine

Instructions

You will use Wireshark, a common network packet analyzer, to analyze network traffic. After starting Wireshark, you will open a previously saved network capture and view a step by step SQL injection attack against a SQL database.

Part 1: Open Wireshark and load the PCAP file.

The Wireshark application can be opened using a variety of methods on a Linux workstation.

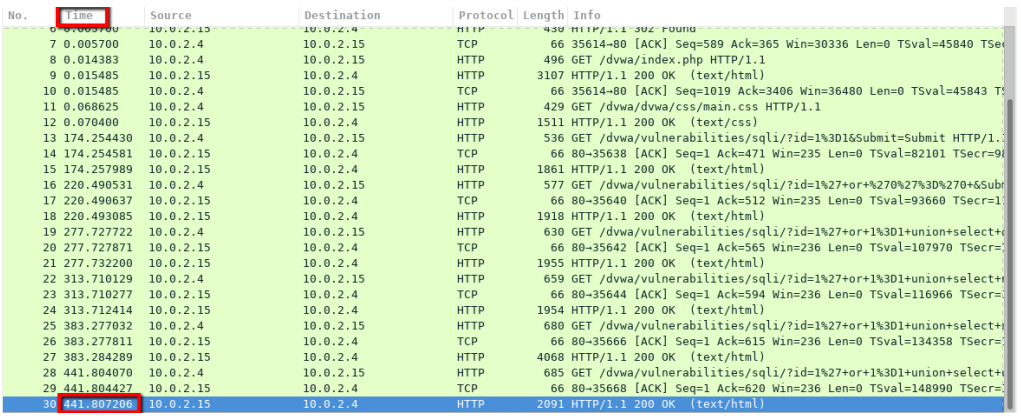
a. Start the CyberOps Workstation VM.

b. Click **Applications > CyberOPS > Wireshark** on the desktop and browse to the Wireshark application.

c. In the Wireshark application, click **Open** in the middle of the application under Files.

d. Browse through the **/home/analyst/** directory and search for **lab.support.files**. In the **lab.support.files** directory and open the **SQL_Lab.pcap** file.

e. The PCAP file opens within Wireshark and displays the captured network traffic. This capture file extends over an 8-minute (441 second) period, the duration of this SQL injection attack.



No.	Time	Source	Destination	Protocol	Length	Info
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 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	536	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+&Sub
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	441.804427	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)

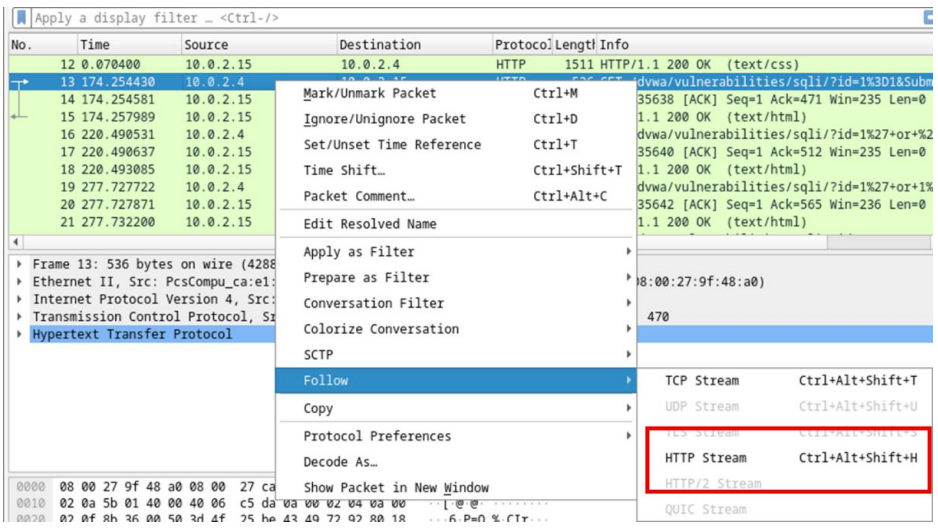
What are the two IP addresses involved in this SQL injection attack based on the information displayed?

Answer: 10.0.2.4 and 10.0.2.15

Part 2: View the SQL Injection Attack.

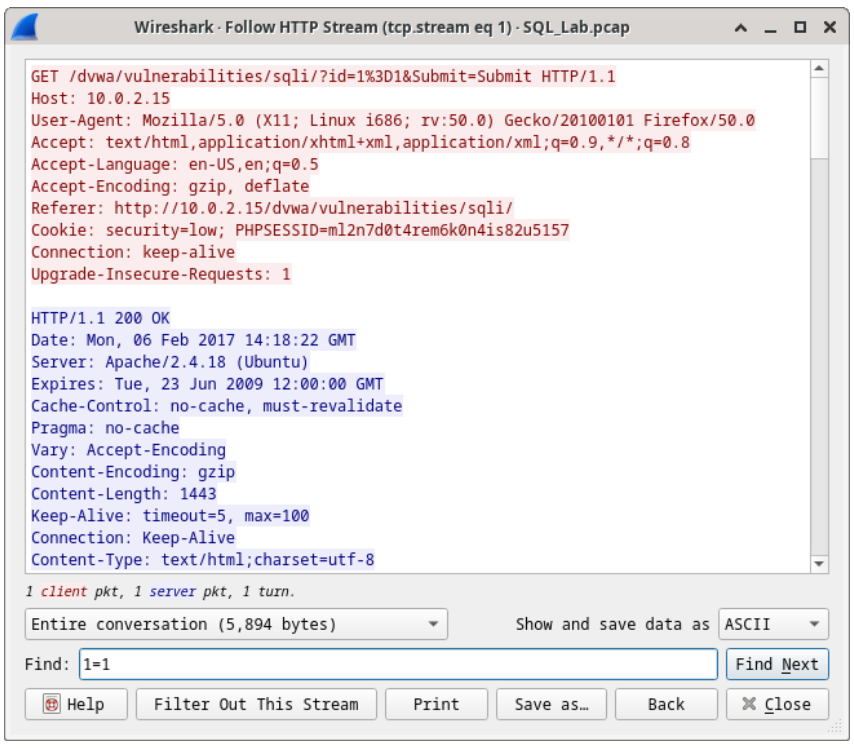
In this step, you will be viewing the beginning of an attack.

a. Within the Wireshark capture, right-click line 13 and select **Follow** > **HTTP Stream**. Line 13 was chosen because it is a GET HTTP request. This will be very helpful in following the data stream as the application layers sees it and leads up to the query testing for the SQL injection.

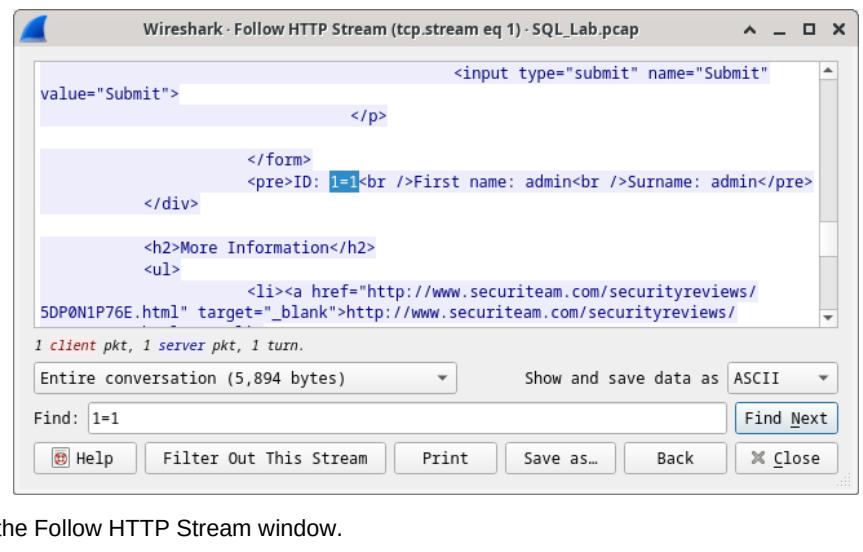


The source traffic is shown in red. The source has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

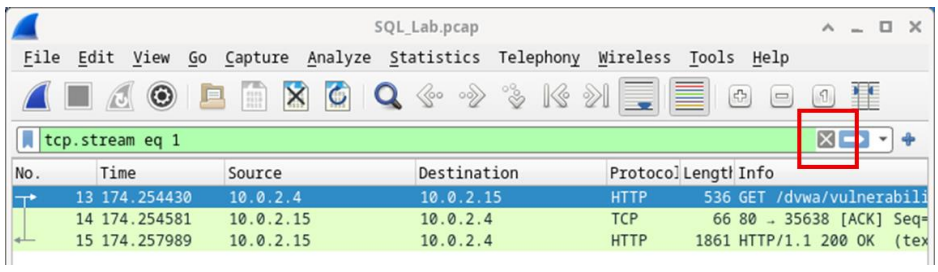


c. The attacker has entered a query (1=1) into a UserID search box on the target 10.0.2.15 to see if the application is vulnerable to SQL injection. Instead of the application responding with a login failure message, it responded with a record from a database. The attacker has verified they can input an SQL command and the database will respond. The search string 1=1 creates an SQL statement that will be always true. In the example, it does not matter what is entered into the field, it will always be true.



d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.



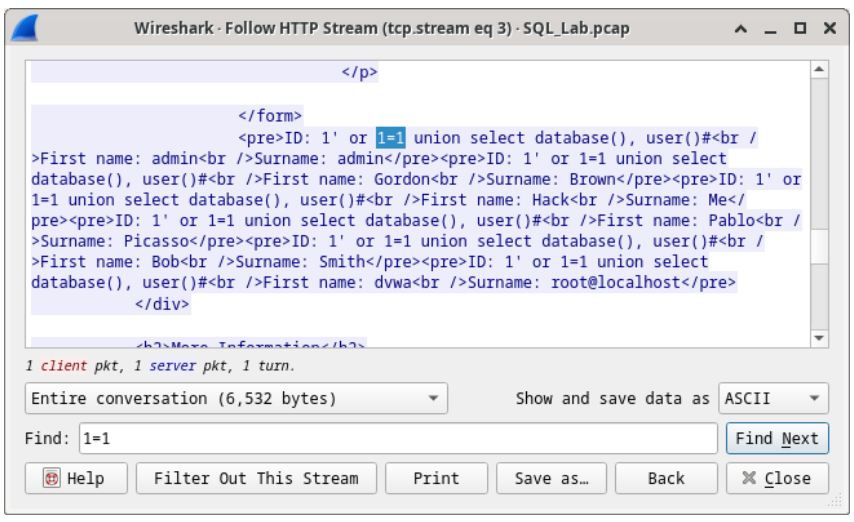
Part 3: The SQL Injection Attack continues...

In this step, you will be viewing the continuation of an attack.

a. Within the Wireshark capture, right-click line 19, and click **Follow** > **HTTP Stream**.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

c. The attacker has entered a query (1' or 1=1 union select database(), user()#) into a UserID search box on the target 10.0.2.15. Instead of the application responding with a login failure message, it responded with the following information:



The database name is **dvwa** and the database user is **root@localhost**. There are also multiple user accounts being displayed.

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

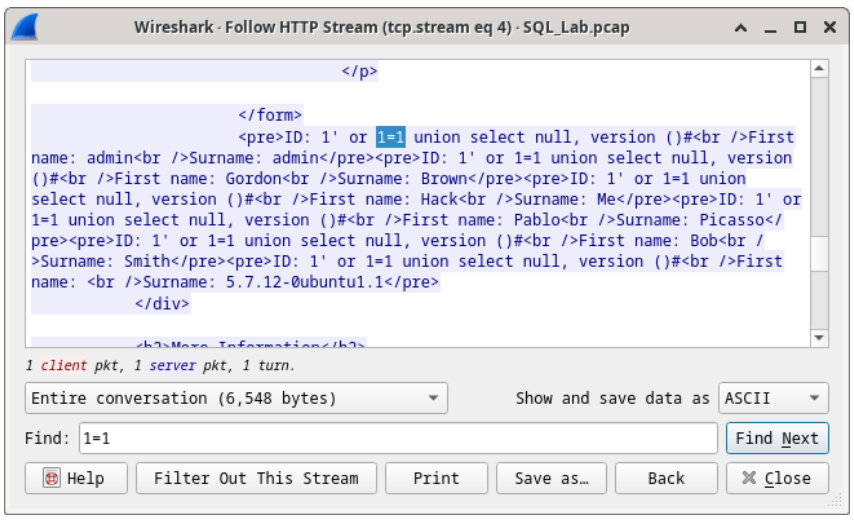
Part 4: The SQL Injection Attack provides system information.

The attacker continues and starts targeting more specific information.

a. Within the Wireshark capture, right-click line 22 and select **Follow** > **HTTP Stream**. In red, the source traffic is shown and is sending the GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

c. The attacker has entered a query (1' or 1=1 union select null, version ()#) into a UserID search box on the target 10.0.2.15 to locate the version identifier. Notice how the version identifier is at the end of the output right before the </pre>. closing HTML code.



What is the version?

Answer: MySQL 5.7.12-0

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

Part 5: The SQL Injection Attack and Table Information.

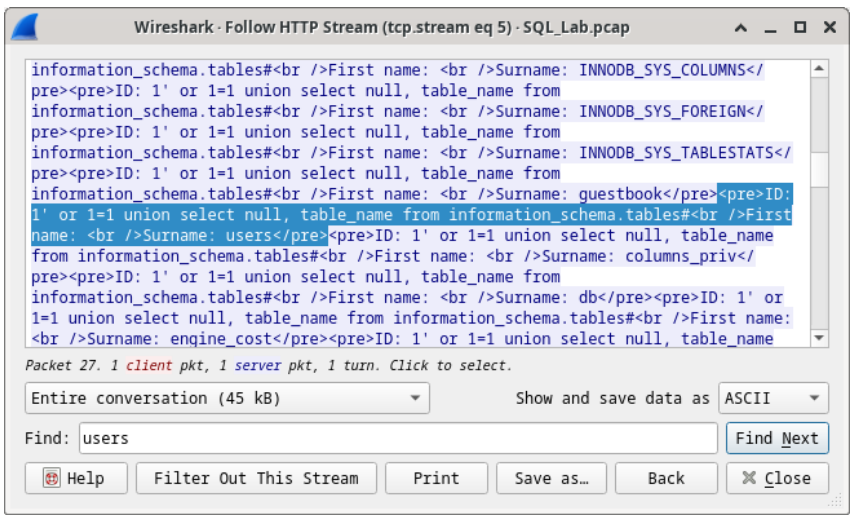
The attacker knows that there is a large number of SQL tables that are full of information. The attacker attempts to find them.

a. Within the Wireshark capture, right-click on line 25 and select **Follow** > **HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **users**. Click **Find Next**.

c. The attacker has entered a query (1'or 1=1 union select null, table_name from information_schema.tables#) into a UserID search box on the target 10.0.2.15 to view all the tables in the

database. This provides a huge output of many tables, as the attacker specified “null” without any further specifications.



What would the modified command of (**1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users'**) do for the attacker?

Answer: The database will respond with much shorter output filtered by occurrences of the word "users".

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

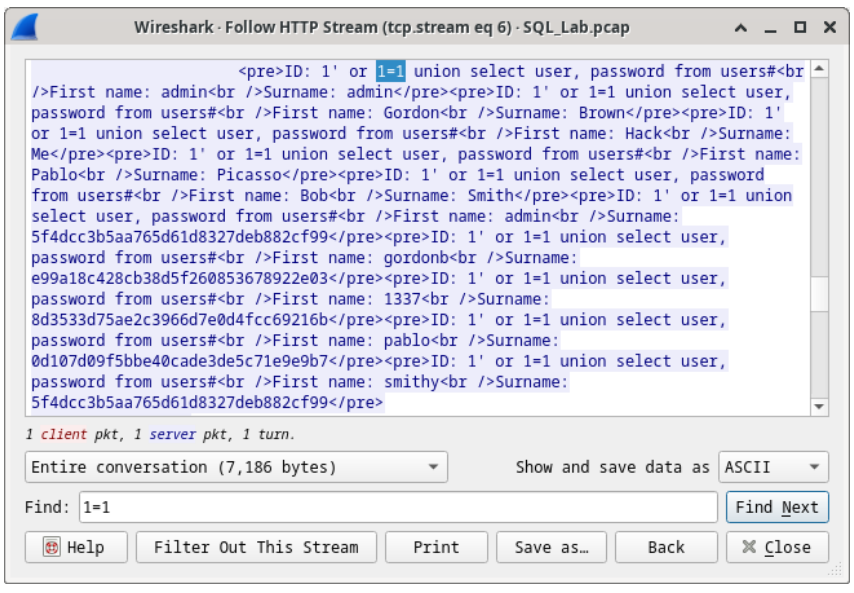
Part 6: The SQL Injection Attack Concludes.

The attack ends with the best prize of all; password hashes.

a. Within the Wireshark capture, right-click line 28 and select **Follow** > **HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. Click **Find** and type in **1=1**. Search for this entry. When the text is located, click **Cancel** in the Find text search box.

The attacker has entered a query (1'or 1=1 union select user, password from users#) into a UserID search box on the target 10.0.2.15 to pull usernames and password hashes!



Which user has the password hash of 8d3533d75ae2c3966d7e0d4fcc69216b?

Answer: Pablo

c. Using a website such as https://crackstation.net/, copy the password hash into the password hash cracker and get cracking.

What is the plain-text password?

Answer: Charley

d. Close the Follow HTTP Stream window. Close any open windows.

Reflection Questions

1. What is the risk of having platforms use the SQL langauge?

   Answer: Websites are generally database driven and use the SQL language. The severity of a SQL injection attack is up to the attacker.

2. Browse the internet and perform a search on "prevent SQL injection attacks". What are 2 methods or steps that can be taken to prevent SQL injection attacks?

   Answer: Filtering user input, implementing web application firewalls, disabling unnecessary database features/capabilities, monitoring SQL statements, using parameters with stored procedures, and using parameters with dynamic SQL

```
newhost:~ # sudo zypper install mariadb mariadb-tools
Loading repository data...
Reading installed packages...
'mariadb' is already installed.
No update candidate for 'mariadb-10.11.9-150600.4.6.1.x86_64'. The highest available version is already installed.
Resolving package dependencies...

The following 3 NEW packages are going to be installed:
  mariadb-tools perl-DBD-mysql perl-DBI

3 new packages to install.
Overall download size: 7.1 MiB. Already cached: 0 B. After the operation, additional 48.9 MiB will be used.

Backend:  classic_rpmtrans
Continue? [y/n/v/...? shows all options] (y): y
Retrieving: perl-DBI-1.642-3.9.1.x86_64 (Main Repository)                                                    (1/3), 740.5 KiB
Retrieving: perl-DBI-1.642-3.9.1.x86_64.rpm ..................................................................[done (769.9 KiB/s)]
Retrieving: perl-DBD-mysql-4.046-3.3.1.x86_64 (Main Repository)                                              (2/3), 154.0 KiB
Retrieving: perl-DBD-mysql-4.046-3.3.1.x86_64.rpm ............................................................[done (1002.8 KiB/s)]
Retrieving: mariadb-tools-10.11.9-150600.4.6.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)  (3/3),   6.2 MiB
Retrieving: mariadb-tools-10.11.9-150600.4.6.1.x86_64.rpm ....................................................[done (16.8 MiB/s)]

Checking for file conflicts: ................................................................................[done]
(1/3) Installing: perl-DBI-1.642-3.9.1.x86_64 ...............................................................[done]
(2/3) Installing: perl-DBD-mysql-4.046-3.3.1.x86_64 .........................................................[done]
(3/3) Installing: mariadb-tools-10.11.9-150600.4.6.1.x86_64 .................................................[done]
newhost:~ # _
```

```
Cleaning up...

All done!  If you've completed all of the above steps, your MariaDB
installation should now be secure.

Thanks for using MariaDB!
newhost:~ # mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 14
Server version: 10.11.9-MariaDB MariaDB package

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database my_database
    -> ;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> CREATE DATABASE my_database;
ERROR 1007 (HY000): Can't create database 'my_database'; database exists
MariaDB [(none)]> CREATE DATABASE test_database;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE DATABASE test_database
    ->
    -> ;
ERROR 1007 (HY000): Can't create database 'test_database'; database exists
MariaDB [(none)]> use test_database
Database changed
MariaDB [test_database]> create table my_table
    -> (
    -> exit
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '(
exit' at line 2
MariaDB [test_database]> create table my_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL, email VARCHAR(100));
Query OK, 0 rows affected (0.035 sec)

MariaDB [test_database]>
```

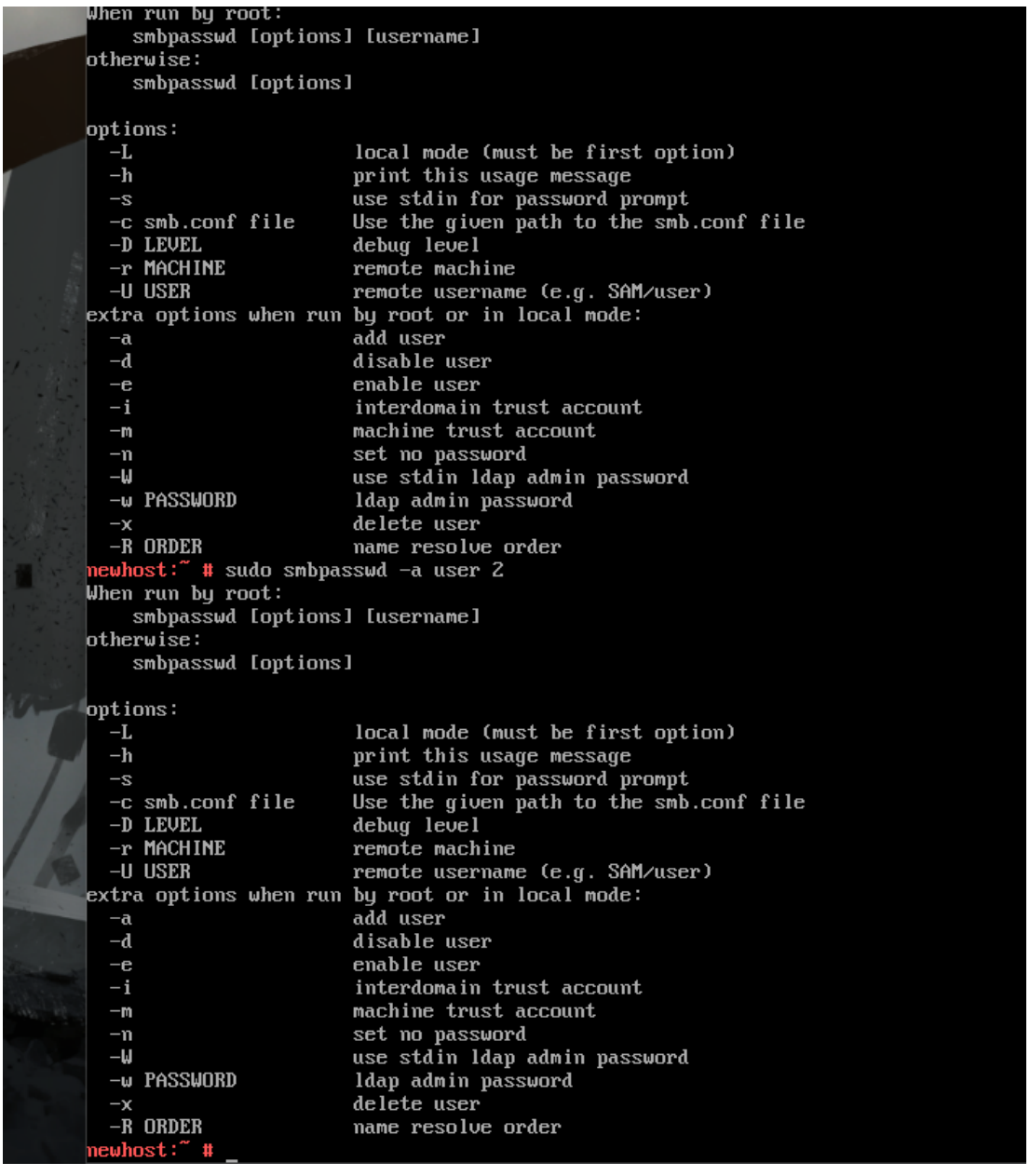
```
exit' at line 2
MariaDB [test_database]> create table my_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL, email VARCHAR(100));
Query OK, 0 rows affected (0.035 sec)

MariaDB [test_database]> INSERT INTO my_table (name,age,email) VALUES ('Alice', 30, 'alice@gmail.com'),('Railley', 21, 'nickoleibautista@gmail.com'), ('Vince',
21, 'railleynickoleivince@gmail.com');
Query OK, 3 rows affected (0.007 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [test_database]> SELECT * FROM my_table
    ->
    ->
    ->
    ->
```

```
    -> ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near
exit' at line 2
MariaDB [test_database]> create table my_table (id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, age INT NOT NULL, email VARCHAR(100));
Query OK, 0 rows affected (0.035 sec)

MariaDB [test_database]> INSERT INTO my_table (name,age,email) VALUES ('Alice', 30, 'alice@gmail.com'),('Railley', 21, 'nickoleibautista@gmail.com'), ('Vinc
21, 'railleynickoleivince@gmail.com');
Query OK, 3 rows affected (0.007 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [test_database]> SELECT * FROM my_table
    ->
    ->
    ->
    ->
    -> ;
+----+---------+-----+--------------------------------+
| id | name    | age | email                          |
+----+---------+-----+--------------------------------+
|  1 | Alice   |  30 | alice@gmail.com                |
|  2 | Railley |  21 | nickoleibautista@gmail.com     |
|  3 | Vince   |  21 | railleynickoleivince@gmail.com |
+----+---------+-----+--------------------------------+
3 rows in set (0.001 sec)
```

```
Have a lot of fun...
newhost:~ # su
newhost:~ # zypper install php php-mysql apache2 mariadb
Retrieving repository 'Update repository of openSUSE Backports' metadata .....[done]
Building repository 'Update repository of openSUSE Backports' cache .....[done]
Retrieving repository 'Update repository with updates from SUSE Linux Enterprise 15' metadata .....[done]
Building repository 'Update repository with updates from SUSE Linux Enterprise 15' cache .....[done]
Retrieving repository 'Main Update Repository' metadata .....[done]
Building repository 'Main Update Repository' cache .....[done]
Retrieving repository 'Update Repository (Non-Oss)' metadata .....[done]
Building repository 'Update Repository (Non-Oss)' cache .....[done]
Loading repository data...
Reading installed packages...
'mariadb' is already installed.
No update candidate for 'mariadb-10.11.9-150600.4.6.1.x86_64'. The highest available version is already installed.
'php-mysql' not found in package names. Trying capabilities.
'php7-mysql' providing 'php-mysql' is already installed.
'php' not found in package names. Trying capabilities.
'php7' providing 'php' is already installed.
Resolving package dependencies...

The following package is going to be upgraded:
  apache2

1 package to upgrade.
Overall download size: 550.5 KiB. Already cached: 0 B. No additional space will be used or freed after the operation.

Backend:  classic_rpmtrans
Continue? [y/n/v/...? shows all options] (y): y
Retrieving: apache2-2.4.58-150600.5.29.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)                    (1/1), 550.5 KiB
Retrieving: apache2-2.4.58-150600.5.29.1.x86_64.rpm .....[done (318.6 KiB/s)]

Checking for file conflicts: .....[done]
Please check /etc/permissions.local for settings of /usr/sbin/suexec .
Updating /etc/sysconfig/apache2 ...
Requesting apache restart (all instances)
(1/1) Installing: apache2-2.4.58-150600.5.29.1.x86_64 .....[done]
Running post-transaction scripts ----------------------------------------------------------------------------------------------------[-]
```

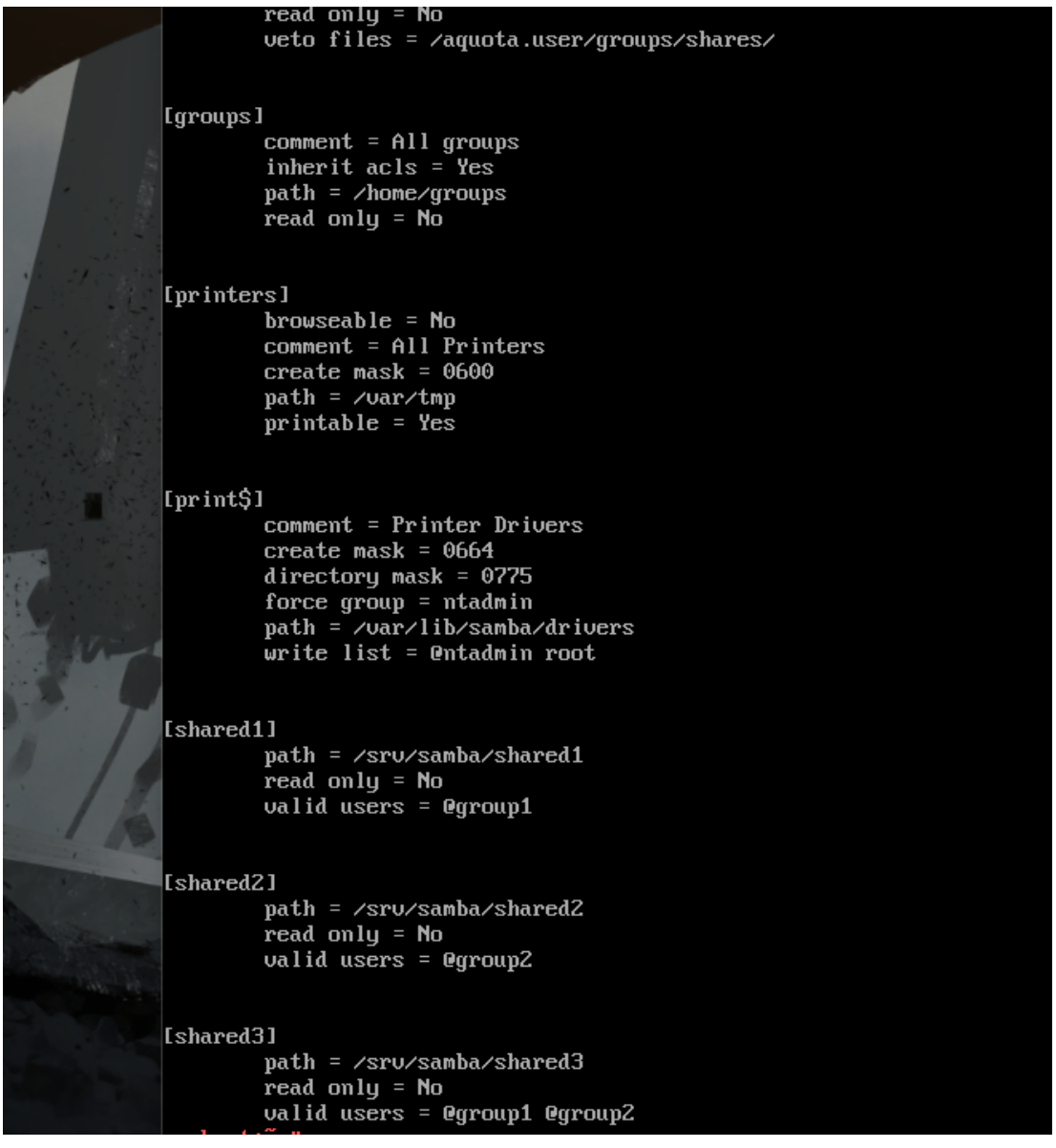
```
File  Machine  View  Input  Devices  Help
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'.

Switch to unix_socket authentication [Y/n] y
Enabled successfully!
Reloading privilege tables..
 ... Success!


You already have your root account protected, so you can safely answer 'n'.

Change the root password? [Y/n] n
 ... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them.  This is intended only for testing, and to make the installation
go a bit smoother.  You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] n
 ... skipping.

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] n
 ... skipping.

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.

Reload privilege tables now? [Y/n] y_
```

```
newhost:~ # sudo zypper install samba
Loading repository data...
Reading installed packages...
Resolving package dependencies...

The following package is going to be upgraded:
  samba-client-libs

The following recommended package was automatically selected:
  samba-libs-python3

The following package is suggested, but will not be installed:
  systemd-sysvcompat

The following 21 NEW packages are going to be installed:
  libsamba-policy0-python3 liburing1 perl-Crypt-SmbHash perl-Digest-MD4 perl-XML-LibXML perl-XML-NamespaceSupport perl-XML-SAX perl-XML-SAX-Base python3-ldb
  python3-talloc python3-tdb python3-tevent samba samba-client samba-dcerpc samba-libs samba-libs-python3 samba-python3 yast2-python3-bindings
  yast2-samba-client yast2-samba-server

1 package to upgrade, 21 new.
Overall download size: 12.4 MiB. Already cached: 0 B. After the operation, additional 35.4 MiB will be used.

Backend:  classic_rpmtrans
Continue? [y/n/v/...? shows all options] (y): y
Retrieving: liburing1-0.6-2.1.x86_64 (Main Repository)                                                                                    (1/22),  24.3 KiB
Retrieving: liburing1-0.6-2.1.x86_64.rpm ..........................................................................................[done (14.8 KiB/s)]
Retrieving: perl-Digest-MD4-1.9-1.28.x86_64 (Main Repository)                                                                             (2/22),  32.9 KiB
Retrieving: perl-Digest-MD4-1.9-1.28.x86_64.rpm ...................................................................................[done (23.6 KiB/s)]
Retrieving: perl-XML-NamespaceSupport-1.12-1.24.noarch (Main Repository)                                                                  (3/22),  26.2 KiB
Retrieving: perl-XML-NamespaceSupport-1.12-1.24.noarch.rpm ........................................................................[done (20.3 KiB/s)]
Retrieving: perl-XML-SAX-Base-1.09-1.25.noarch (Main Repository)                                                                          (4/22),  33.2 KiB
Retrieving: perl-XML-SAX-Base-1.09-1.25.noarch.rpm ................................................................................[done (16.1 KiB/s)]
Retrieving: python3-ldb-2.8.0-150600.1.4.x86_64 (Main Repository)                                                                         (5/22),  60.6 KiB
Retrieving: python3-ldb-2.8.0-150600.1.4.x86_64.rpm ...............................................................................[done (14.8 KiB/s)]
Retrieving: python3-talloc-2.4.1-150600.1.3.x86_64 (Main Repository)                                                                      (6/22),  23.7 KiB
Retrieving: python3-talloc-2.4.1-150600.1.3.x86_64.rpm ------------------------------------------------------------------------------------[-]
```


When run by root:
smbpasswd [options] [username]
otherwise:
smbpasswd [options]
options:
-L local mode (must be first option)
-h print this usage message
-s use stdin for password prompt
-c smb.conf file Use the given path to the smb.conf file
-D LEVEL debug level
-r MACHINE remote machine
-U USER remote username (e.g. SAM/user)
extra options when run by root or in local mode:
-a add user
-d disable user
-e enable user
-i interdomain trust account
-m machine trust account
-n set no password
-W use stdin ldap admin password
-w PASSWORD ldap admin password
-x delete user
-R ORDER name resolve order
newhost:~ # sudo smbpasswd -a user 2
When run by root:
smbpasswd [options] [username]
otherwise:
smbpasswd [options]
options:
-L local mode (must be first option)
-h print this usage message
-s use stdin for password prompt
-c smb.conf file Use the given path to the smb.conf file
-D LEVEL debug level
-r MACHINE remote machine
-U USER remote username (e.g. SAM/user)
extra options when run by root or in local mode:
-a add user
-d disable user
-e enable user
-i interdomain trust account
-m machine trust account
-n set no password
-W use stdin ldap admin password
-w PASSWORD ldap admin password
-x delete user
-R ORDER name resolve order
newhost:~ #

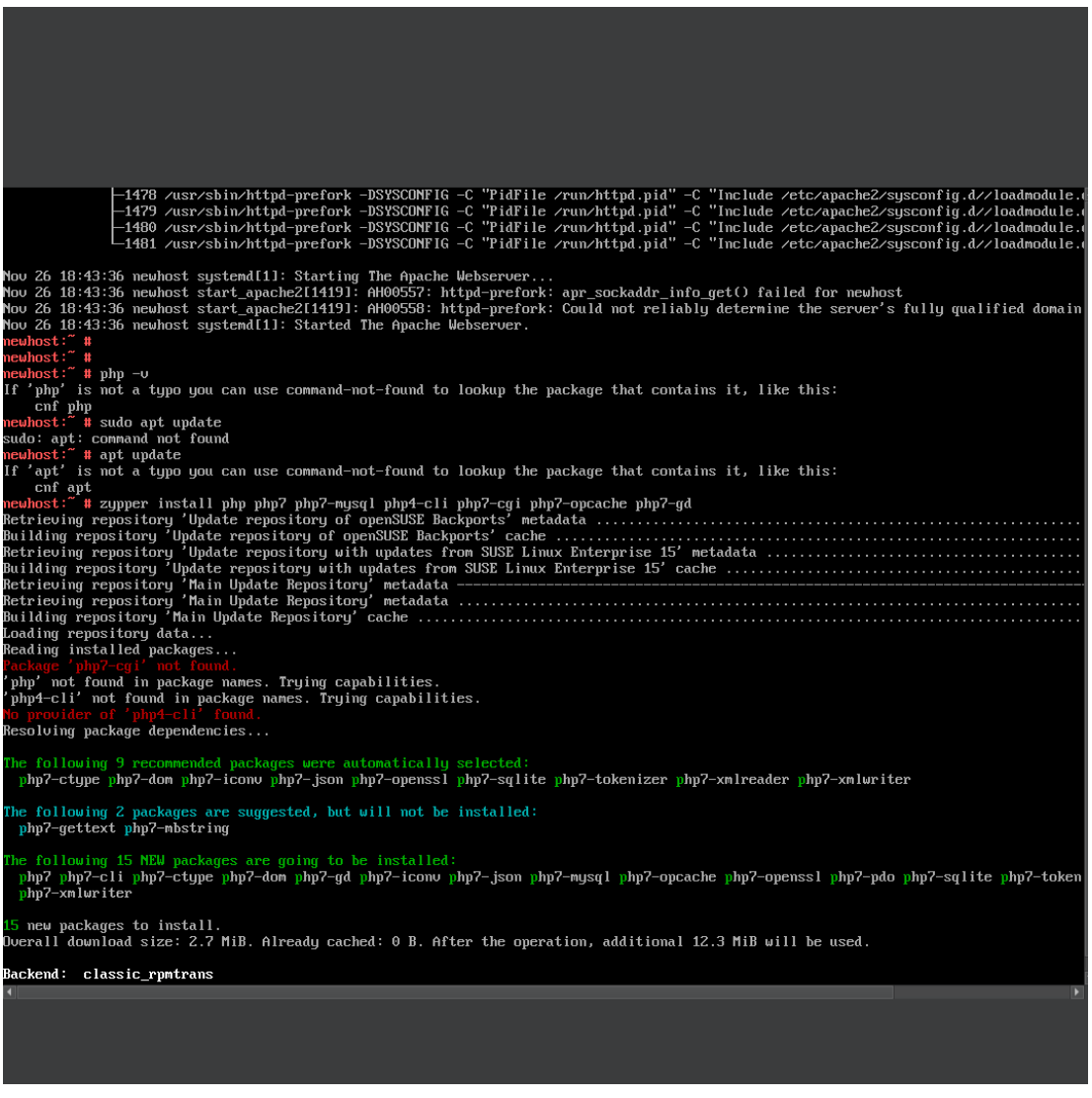
```
  -R ORDER              name resolve order
newhost:~ # sudo mkdir -p /srv/samba/shared1
newhost:~ # sudo mkdir -p /srv/samba/shared2
newhost:~ # sudo mkdir -p /srv/samba/shared3
newhost:~ # sudo chowd -R :group1 /srv/samba/shared1
sudo: chowd: command not found
newhost:~ # sudo chown -R :group1 /srv/samba/shared1
newhost:~ # sudo chmod 770 /srv/samba/shared1
newhost:~ # sudo chown -R :group2 /srv/samba/shared2
newhost:~ # sudo chmod 770 /srv/samba/shared2
newhost:~ # sudo chown -R :group1 /srv/samba/shared3
newhost:~ # sudo chown -R :group2 /srv/samba/shared3
newhost:~ # sudo chmod 770 /srv/samba/shared3
newhost:~ # _
```

```
        comment = Network Profiles Service
        path = %H
        read only = No
        store dos attributes = Yes
        create mask = 0600
        directory mask = 0700
[users]
        comment = All users
        path = /home
        read only = No
        inherit acls = Yes
        veto files = /aquota.user/groups/shares/
[groups]
        comment = All groups
        path = /home/groups
        read only = No
        inherit acls = Yes
[printers]
        comment = All Printers
        path = /var/tmp
        printable = Yes
        create mask = 0600
        browseable = No
[print$]
        comment = Printer Drivers
        path = /var/lib/samba/drivers
        write list = @ntadmin root
        force group = ntadmin
        create mask = 0664
        directory mask = 0775
[shared1]
        path = /srv/samba/shared1
        valid users = @group1
        read only = no
[shared2]
        path = /srv/samba/shared2
        valid users = @group2
        read only = no
[shared3]
        path = /srv/samba/shared3
        valid users = @group1, @group2
        read only = no
```



```
        read only = No
        veto files = /aquota.user/groups/shares/


[groups]
        comment = All groups
        inherit acls = Yes
        path = /home/groups
        read only = No


[printers]
        browseable = No
        comment = All Printers
        create mask = 0600
        path = /var/tmp
        printable = Yes


[print$]
        comment = Printer Drivers
        create mask = 0664
        directory mask = 0775
        force group = ntadmin
        path = /var/lib/samba/drivers
        write list = @ntadmin root


[shared1]
        path = /srv/samba/shared1
        read only = No
        valid users = @group1


[shared2]
        path = /srv/samba/shared2
        read only = No
        valid users = @group2


[shared3]
        path = /srv/samba/shared3
        read only = No
        valid users = @group1 @group2
```

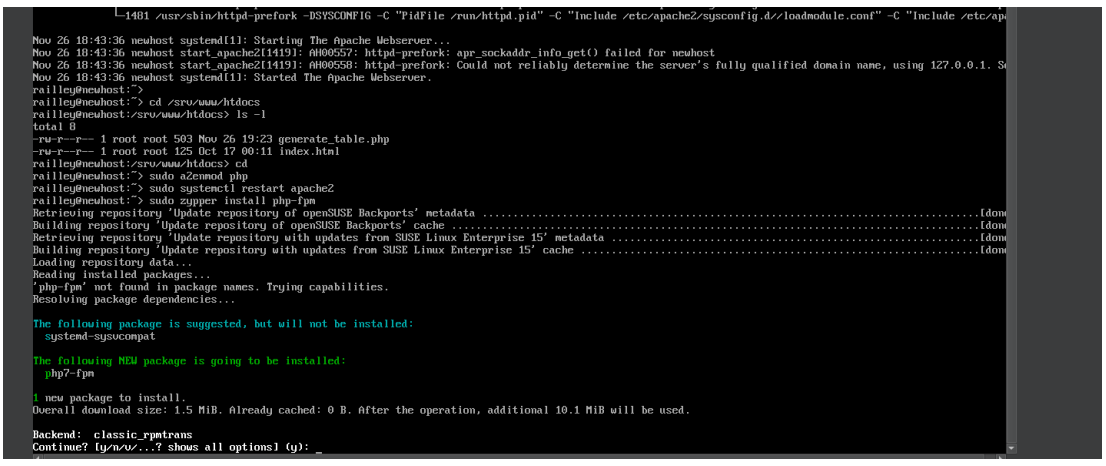
Installing php in opensuse



```
  php7 php7-cli php7-ctype php7-dom php7-gd php7-iconv php7-json php7-mysql php7-opcache php7-openssl php7-pdo php7-sqlite php7-tokenizer php7-xmlreader
  php7-xmlwriter

15 new packages to install.
Overall download size: 2.7 MiB. Already cached: 0 B. After the operation, additional 12.3 MiB will be used.

Backend:  classic_rpmtrans
Continue? [y/n/v/...? shows all options] (y): y
[ Invalid answer: 'y'
[y/n/v/...? shows all options] (y): y
Retrieving: php7-cli-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)                (1/15),   1.5 MiB
Retrieving: php7-cli-7.4.33-150400.4.40.1.x86_64.rpm ..........................................................................[done (770.1 KiB/s
Retrieving: php7-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)                   (2/15), 114.3 KiB
Retrieving: php7-7.4.33-150400.4.40.1.x86_64.rpm ..............................................................................[done
Retrieving: php7-dom-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)               (3/15),  95.0 KiB
Retrieving: php7-dom-7.4.33-150400.4.40.1.x86_64.rpm ..........................................................................[done (25.0 KiB/s
Retrieving: php7-openssl-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)           (4/15), 102.9 KiB
Retrieving: php7-openssl-7.4.33-150400.4.40.1.x86_64.rpm ......................................................................[done (350.3 KiB/s
Retrieving: php7-ctype-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)            (5/15),  49.7 KiB
Retrieving: php7-ctype-7.4.33-150400.4.40.1.x86_64.rpm ........................................................................[done (23.6 KiB/s
Retrieving: php7-iconv-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)            (6/15),  63.4 KiB
Retrieving: php7-iconv-7.4.33-150400.4.40.1.x86_64.rpm ........................................................................[done (33.7 KiB/s
Retrieving: php7-json-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)             (7/15),  66.1 KiB
Retrieving: php7-json-7.4.33-150400.4.40.1.x86_64.rpm .........................................................................[done (24.2 KiB/s
Retrieving: php7-pdo-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)              (8/15),  88.9 KiB
Retrieving: php7-pdo-7.4.33-150400.4.40.1.x86_64.rpm ..........................................................................[done (20.3 KiB/s
Retrieving: php7-sqlite-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)           (9/15),  72.5 KiB
Retrieving: php7-sqlite-7.4.33-150400.4.40.1.x86_64.rpm .......................................................................[done (20.3 KiB/s
Retrieving: php7-tokenizer-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)       (10/15),  53.0 KiB
Retrieving: php7-tokenizer-7.4.33-150400.4.40.1.x86_64.rpm ....................................................................[done (20.3 KiB/s
Retrieving: php7-xmlreader-7.4.33-150400.4.40.1.x86_64 (Update repository with updates from SUSE Linux Enterprise 15)       (11/15),  56.9 KiB
Retrieving: php7-xmlreader-7.4.33-150400.4.40.1.x86_64.rpm ----------------------------------------------------------------------[
```

```
newhost:~ # php -v
PHP 7.4.33 (cli) (built: Oct 11 2024 12:00:00) ( NTS )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
    with Zend OPcache v7.4.33, Copyright (c), by Zend Technologies
newhost:~ #
```

```php
<?php

$servername = "localhost";
$username = "railley@newhost";
$password = "railley";
$dbname = "sampledb";


$conn = new mysqli($servername, $username, $password, $dbname);

if ($conn -> connect_error)
{
        echo("Connection Failed:".$conn->connect_error);
}

echo "Table Details";

$sql ="select * from Users";

$result = $conn -> query($sql);

if ($result->num_rows>0)
{
        while($row = result->fetch_assoc())
        {
        echo "UserID: ".$row['UserID'].
        }
}else
{
 echo "no results";
}


$conn->close();

?>
```


└─1401 /usr/sbin/httpd-prefork -DSYSCONFIG -C "PidFile /run/httpd.pid" -C "Include /etc/apache2/sysconfig.d//loadmodule.conf" -C "Include /etc/ap
Nov 26 18:43:36 newhost systemd[1]: Starting The Apache Webserver...
Nov 26 18:43:36 newhost start_apache2[14191]: AH00557: httpd-prefork: apr_sockaddr_info_get() failed for newhost
Nov 26 18:43:36 newhost start_apache2[14191]: AH00558: httpd-prefork: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1. S
Nov 26 18:43:36 newhost systemd[1]: Started The Apache Webserver.
railley@newhost:~>
railley@newhost:~> cd /srv/www/htdocs
railley@newhost:/srv/www/htdocs> ls -l
total 8
-rw-r--r-- 1 root root 503 Nov 26 19:23 generate_table.php
-rw-r--r-- 1 root root 125 Oct 17 00:11 index.html
railley@newhost:/srv/www/htdocs> cd
railley@newhost:~> sudo a2enmod php
railley@newhost:~> sudo systemctl restart apache2
railley@newhost:~> sudo zypper install php-fpm
Retrieving repository 'Update repository of openSUSE Backports' metadata ..... [dow
Building repository 'Update repository of openSUSE Backports' cache ..... [dow
Retrieving repository 'Update repository with updates from SUSE Linux Enterprise 15' metadata ..... [dow
Building repository 'Update repository with updates from SUSE Linux Enterprise 15' cache ..... [dow
Loading repository data...
Reading installed packages...
'php-fpm' not found in package names. Trying capabilities.
Resolving package dependencies...
The following package is suggested, but will not be installed:
systemd-sysvcompat
The following NEW package is going to be installed:
php7-fpm
1 new package to install.
Overall download size: 1.5 MiB. Already cached: 0 B. After the operation, additional 10.1 MiB will be used.
Backend: classic_rpmtrans
Continue? [y/n/v/...? shows all options] (y): _

Lab - Attacking a mySQL Database

Objectives

In this lab, you will view a PCAP file from a previous attack against a SQL database.

Part 1: Open Wireshark and load the PCAP file.

Part 2: View the SQL Injection Attack.

Part 3: The SQL Injection Attack continues…

Part 4: The SQL Injection Attack provides system information.

Part 5: The SQL Injection Attack and Table Information

Part 6: The SQL Injection Attack Concludes.

Background / Scenario

SQL injection attacks allow malicious hackers to type SQL statements in a web site and receive a response from the database. This allows attackers to tamper with current data in the database, spoof identities, and miscellaneous mischief.

A PCAP file has been created for you to view a previous attack against a SQL database. In this lab, you will view the SQL database attacks and answer the questions.

Required Resources

- CyberOps Workstation virtual machine

Instructions

You will use Wireshark, a common network packet analyzer, to analyze network traffic. After starting Wireshark, you will open a previously saved network capture and view a step by step SQL injection attack against a SQL database.

Part 1: Open Wireshark and load the PCAP file.

The Wireshark application can be opened using a variety of methods on a Linux workstation.

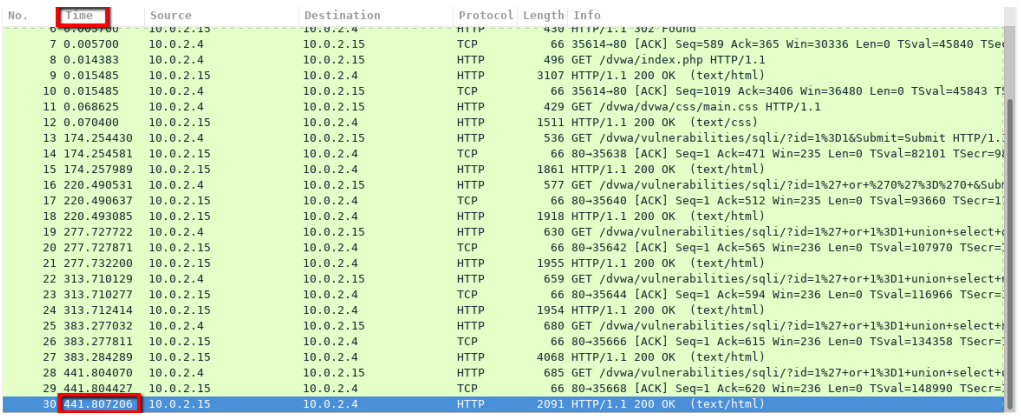
a. Start the CyberOps Workstation VM.

b. Click **Applications > CyberOPS > Wireshark** on the desktop and browse to the Wireshark application.

c. In the Wireshark application, click **Open** in the middle of the application under Files.

d. Browse through the **/home/analyst/** directory and search for **lab.support.files**. In the **lab.support.files** directory and open the **SQL_Lab.pcap** file.

e. The PCAP file opens within Wireshark and displays the captured network traffic. This capture file extends over an 8-minute (441 second) period, the duration of this SQL injection attack.



No.	Time	Source	Destination	Protocol	Length	Info
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 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	536	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+&Sub
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	441.804427	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)

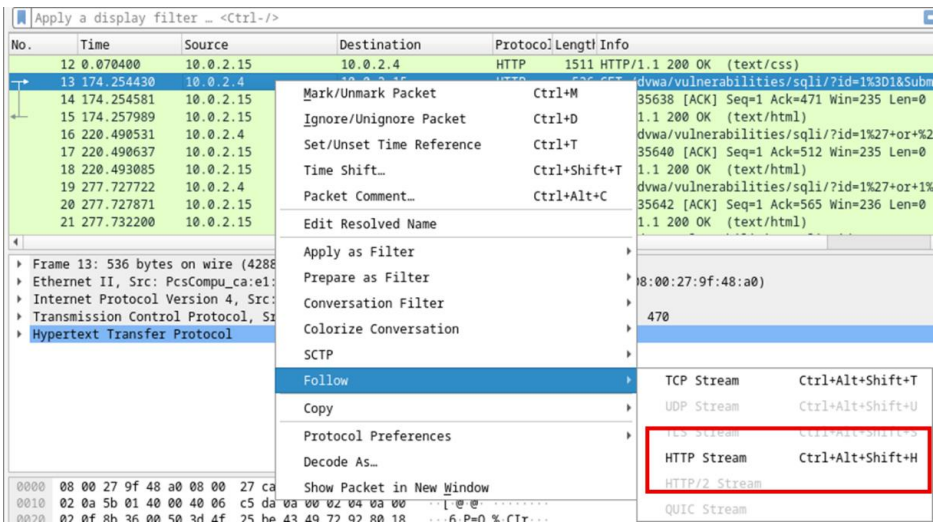
What are the two IP addresses involved in this SQL injection attack based on the information displayed?

Answer: 10.0.2.4 and 10.0.2.15

Part 2: View the SQL Injection Attack.

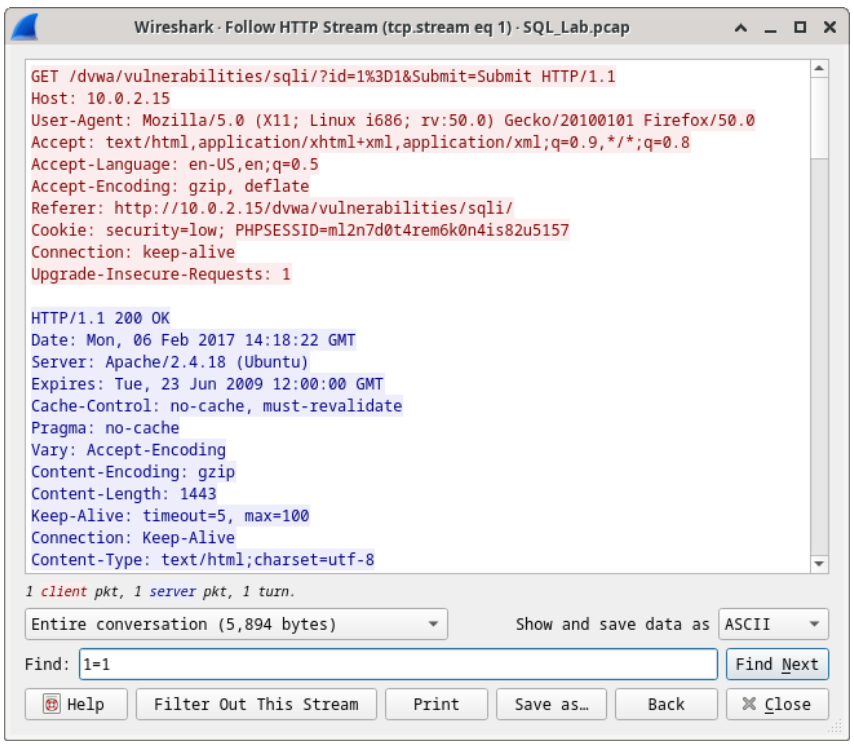
In this step, you will be viewing the beginning of an attack.

a. Within the Wireshark capture, right-click line 13 and select **Follow** > **HTTP Stream**. Line 13 was chosen because it is a GET HTTP request. This will be very helpful in following the data stream as the application layers sees it and leads up to the query testing for the SQL injection.

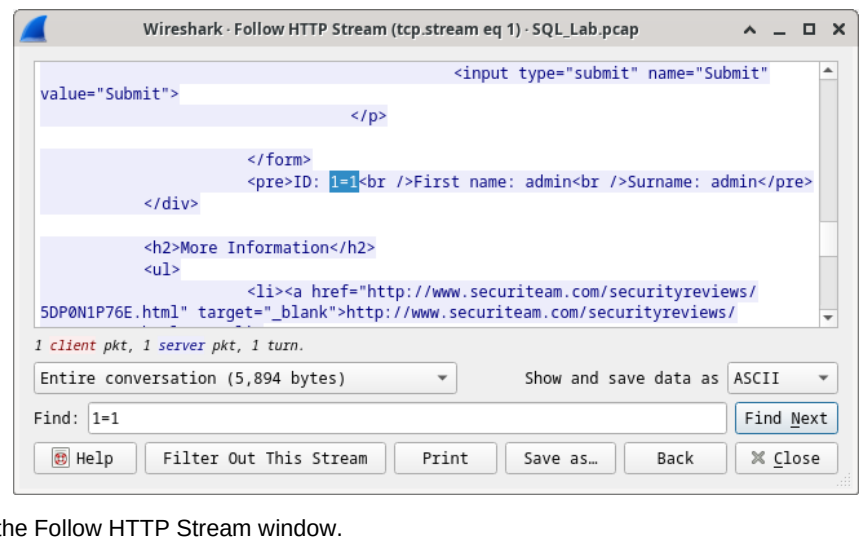


The source traffic is shown in red. The source has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

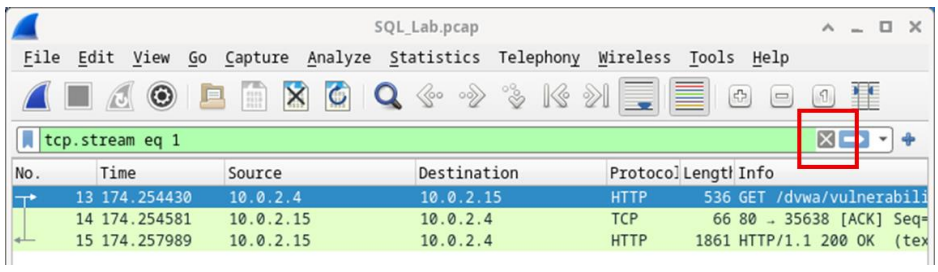


c. The attacker has entered a query (1=1) into a UserID search box on the target 10.0.2.15 to see if the application is vulnerable to SQL injection. Instead of the application responding with a login failure message, it responded with a record from a database. The attacker has verified they can input an SQL command and the database will respond. The search string 1=1 creates an SQL statement that will be always true. In the example, it does not matter what is entered into the field, it will always be true.



d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.



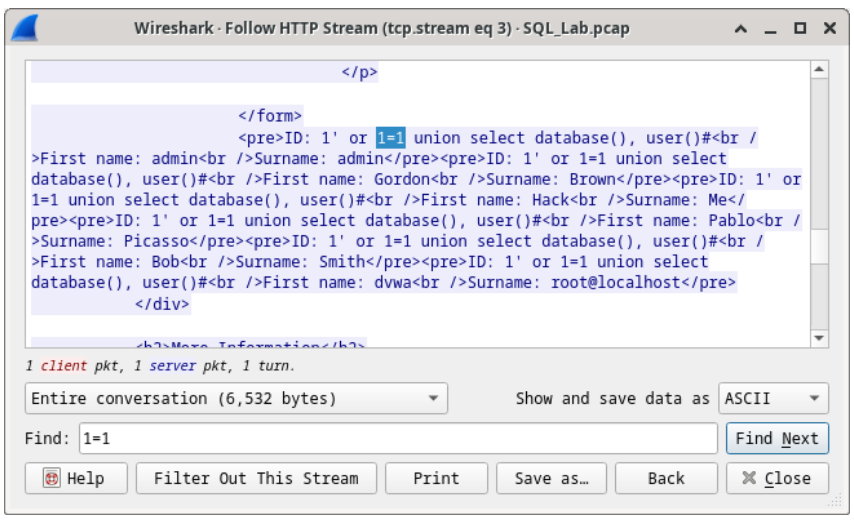
Part 3: The SQL Injection Attack continues...

In this step, you will be viewing the continuation of an attack.

a. Within the Wireshark capture, right-click line 19, and click **Follow** > **HTTP Stream**.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

c. The attacker has entered a query (1' or 1=1 union select database(), user()#) into a UserID search box on the target 10.0.2.15. Instead of the application responding with a login failure message, it responded with the following information:



The database name is **dvwa** and the database user is **root@localhost**. There are also multiple user accounts being displayed.

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

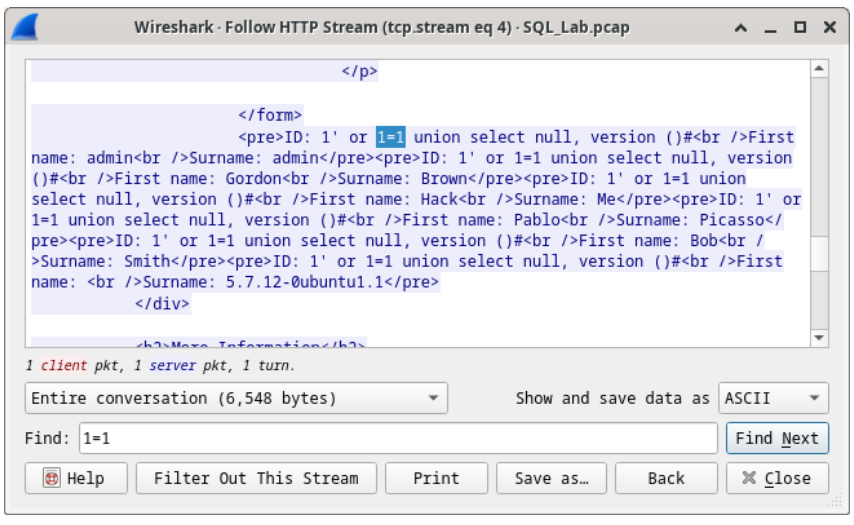
Part 4: The SQL Injection Attack provides system information.

The attacker continues and starts targeting more specific information.

a. Within the Wireshark capture, right-click line 22 and select **Follow** > **HTTP Stream**. In red, the source traffic is shown and is sending the GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **1=1**. Click **Find Next**.

c. The attacker has entered a query (1' or 1=1 union select null, version ()#) into a UserID search box on the target 10.0.2.15 to locate the version identifier. Notice how the version identifier is at the end of the output right before the </pre>. closing HTML code.



What is the version?

Answer: MySQL 5.7.12-0

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

Part 5: The SQL Injection Attack and Table Information.

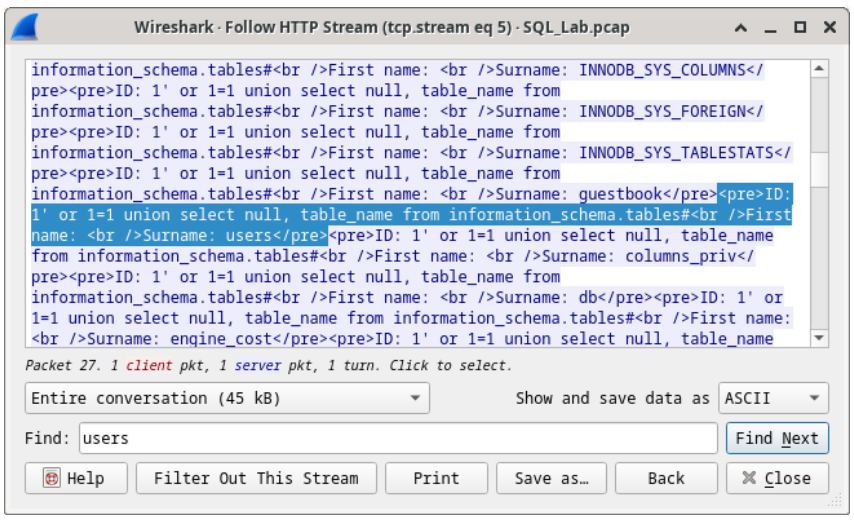
The attacker knows that there is a large number of SQL tables that are full of information. The attacker attempts to find them.

a. Within the Wireshark capture, right-click on line 25 and select **Follow** > **HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. In the **Find** field, enter **users**. Click **Find Next**.

c. The attacker has entered a query (1'or 1=1 union select null, table_name from information_schema.tables#) into a UserID search box on the target 10.0.2.15 to view all the tables in the

database. This provides a huge output of many tables, as the attacker specified “null” without any further specifications.



What would the modified command of (**1' OR 1=1 UNION SELECT null, column_name FROM INFORMATION_SCHEMA.columns WHERE table_name='users'**) do for the attacker?

Answer: The database will respond with much shorter output filtered by occurrences of the word "users".

d. Close the Follow HTTP Stream window.

e. Click **Clear display filter** to display the entire Wireshark conversation.

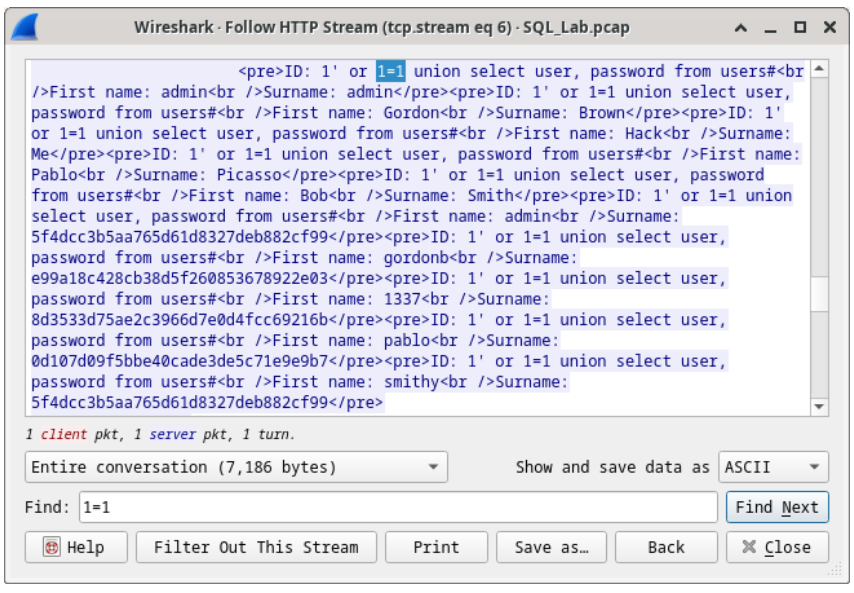
Part 6: The SQL Injection Attack Concludes.

The attack ends with the best prize of all; password hashes.

a. Within the Wireshark capture, right-click line 28 and select **Follow** > **HTTP Stream**. The source is shown in red. It has sent a GET request to host 10.0.2.15. In blue, the destination device is responding back to the source.

b. Click **Find** and type in **1=1**. Search for this entry. When the text is located, click **Cancel** in the Find text search box.

The attacker has entered a query (1'or 1=1 union select user, password from users#) into a UserID search box on the target 10.0.2.15 to pull usernames and password hashes!



Which user has the password hash of 8d3533d75ae2c3966d7e0d4fcc69216b?

Answer: Pablo

c. Using a website such as https://crackstation.net/, copy the password hash into the password hash cracker and get cracking.

What is the plain-text password?

Answer: Charley

d. Close the Follow HTTP Stream window. Close any open windows.

Reflection Questions

1. What is the risk of having platforms use the SQL langauge?

   Answer: Websites are generally database driven and use the SQL language. The severity of a SQL injection attack is up to the attacker.

2. Browse the internet and perform a search on "prevent SQL injection attacks". What are 2 methods or steps that can be taken to prevent SQL injection attacks?

   Answer: Filtering user input, implementing web application firewalls, disabling unnecessary database features/capabilities, monitoring SQL statements, using parameters with stored procedures, and using parameters with dynamic SQL