# SQL INJECTION VULNERABILITIES ANALYSIS USING SQLMAP: A CASE STUDY OF BANGLADESHI WEBSITES

**BY**

**SUMADHA CHAKMA**

**ID: 181-15-11169**

**AND**

**IRIN AKTAR PUSHPA**

**ID: 181-15-11153**

This Report Presented in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Computer Science and Engineering.

Supervised By

**Md. Sadekur Rahman**
Assistant professor
Department of CSE
Daffodil International University

Co-Supervised By

**Md. Zahid Hasan**
Assistant Professor
Department of CSE
Daffodil International University


**DAFFODIL INTERNATIONAL UNIVERSITY**
**DHAKA, BANGLADESH**
**JANUARY, 2022**

## APPROVAL

This Project/internship titled "SQL injection vulnerabilities analysis using SQLMAP: a case study of Bangladeshi websites" submitted by Irin Aktar Pushpa and Sumadha Chakma, ID No: 181-15-11153 and 181-15-11169 to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on 4th January, 2022.

## BOARD OF EXAMINERS

**Chairman**

**Dr. Sheak Rashed Haider Noori**
**Associate Professor and Associate Head**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Internal Examiner**

**Abdus Sattar**
**Assistant Professor**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

**Internal Examiner**

**Saiful Islam**
**Senior Lecturer**
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

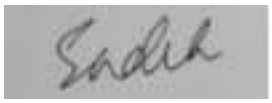
**External Examiner**

**Dr. Dewan Md. Farid**
**Professor**
Department of Computer Science and Engineering
United International University

# DECLARATION

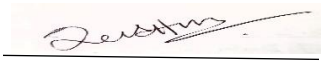
We hereby declare that this project has been done by us under the supervision of **Md. Sadekur Rahman, Assistant professor, Department of CSE** Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for award of any degree or diploma.

**Supervised by:**



**Md. Sadekur Rahman**
Assistant professor
Department of CSE
Daffodil International University

**Co-Supervised by:**

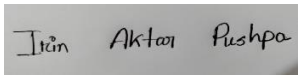


**Md. Zahid Hasan**
Assistant professor
Department of CSE
Daffodil International University

**Submitted by:**

**Sumadha Chakma**

ID: 181-15-11169
Department of CSE
Daffodil International University



**Irin Aktar Pushpa**

ID: 181-15-11153
Department of CSE
Daffodil International University

# ACKNOWLEDGEMENT

First, we express our heartiest thanks and gratefulness to Almighty God for His divine blessing makes it possible to complete the final year project/internship successfully. We are grateful and wish our profound indebtedness to our **supervisor, Md. Sadekur Rahman, Assistant professor**, **co-supervisor, Zahid Hasan, Associate Professor**, Department of CSE, Daffodil International University, Dhaka, Deep Knowledge & keen interest of our supervisor in the field of App development to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

We would like to express our heartiest gratitude to **Dr. Touhid Bhuiyan**, **Professor and Head**, Department of CSE, for his kind help to finish our project and also to other faculty members and the staff of the CSE department of Daffodil International University.

We would like to thank our entire course mate at Daffodil International University, who took part in this discussion while completing the course work.

Finally, we must acknowledge with due respect the constant support and patience of us parents.

# Abstract

To keep up with the global pace of digitalization, developing countries, like developed countries, are providing services to their citizens through various online portals, web applications, and web sites. When it comes to a web application, cybersecurity is one of the most discussed topics, and protecting the confidentiality and integrity of data has become critical. Unfortunately, many of those web-based services are vulnerable to serious security threats as a result of a lack of consideration for vulnerability issues during the development phase. Vulnerability statistics are required for these developing countries to gain insight into the current security status of the web services provided. SQLi is one of the most common techniques used by hackers to exploit a security flaw in a web application. In this paper, we used Sqlmap to detect SQLi vulnerabilities in Bangladeshi websites. We conducted the survey for 150 Bangladeshi websites because the country has been focusing on digitalization of government services for the last few years and already provides a variety of online services to its citizens. Among the 150 websites from various categories, the results show that the majority of them are vulnerable to error-based SQL Injection.

# TABLE OF CONTENTS


| CONTENTS | PAGE |
|---|---|
| Board of examiners | i |
| Declaration | ii |
| Acknowledgement | iii |
| Abstract | iv |

| CONTENTS | PAGE |
|---|---|
| **CHAPTER 1: INTRODUCTION** | **1-3** |
| 1.1 Introduction | 1 |
| 1.2 Motivation | 1 |
| 1.3 Rational of the Study | 2 |
| 1.4 Research Question | 2 |
| 1.5 Expected Outcome | 2 |
| 1.6 Project Management and Finance | 2 |
| 1.7 Report Layout | 3 |
| **CHAPTER 2: BACKGROUND** | **4-8** |
| 2.1 Preliminaries/Terminologies | 4 |
| 2.2 Related Works | 5 |
| 2.3 Comparative Analysis and Summary | 6 |
| 2.4 Scope of the Problem | 7 |
| 2.5 Challenges | 8 |
| **CHAPTER 3: RESEARCH METHODOLOGY** | **9-14** |
| 3.1 Research Subject and Instrumentation | 9 |
| 3.2 Data Collection Procedure/Dataset Utilized | 9 |
| 3.3 Statistical Analysis | 10 |

3.4 Proposed Methodology 11
3.5 Implementation Requirements 12

**CHAPTER 4: EXPERIMENTAL RESULTS AND DISCUSSION** **15-19**

4.1 Experimental Setup 15
4.2 Experimental Result and Analysis 16
4.3 Discussion 19

**CHAPTER 5: IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY** **19-20**

5.1 Impact on Society 19
5.2 Impact on Environment 20
5.3 Ethical Aspects 20
5.4 Sustainability Plan 20

**CHAPTER 6: SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH** **21-22**

6.1 Summary of the Study 21
6.2 Conclusions 21
6.3 Implication for Further Study 22

**References** **23-24**

# LIST OF FIGURES


| FIGURES NO | FIGURES TITLE | PAGE NO |
|---|---|---|
| Figure 3.1 | Visualization of Educational websites SQLi vulnerability of Bangladesh | 10 |
| Figure 3.2 | Visualization of BD Govt. websites SQLi vulnerability of Bangladesh | 10 |
| Figure 3.3 | Visualization of BD top NGOs websites SQLi vulnerability of Bangladesh | 11 |
| Figure 3.4 | Flow diagram of experimental process | 11 |
| Figure 3.5 | SQLMAP parameter | 12 |
| Figure 3.6 | command for fetch database (didn't detect attack) | 13 |
| Figure 3.7 | command for fetch database (detect attack) | 14 |
| Figure 4.1 | List information about Tables present in a specific database | 15 |
| Figure 4.2 | using command for dump the data | 16 |
| Figure 4.3 | show fetched databases | 17 |
| Figure 4.4 | fetched 19 tables from specific database | 17 |
| Figure 4.5 | fetched 19 tables from specific database | 18 |
| Figure 4.6 | Result of dump data from specific table | 18 |

# LIST OF TABLES


| TABLE NO | TABLE NAME | PAGE NO |
|---|---|---|
| 2.1 | Reference paper analyzed and summary | 6 |

# CHAPTER 1

# INTRODUCTION

**1.1 Introduction:** We live in such a century where ICT is a must thing we should know about. As the world is developing day by day with modern technologies, the work is getting easier. While technology gives us all the benefits, we shouldn't forget about its downsides that happen pretty much every moment. Bangladesh is trying to keep pace with the times as a developing country. Although Bangladesh is still lagging in many ways, almost everything in Bangladesh is web-based. It has become a trend to launch a web application for every service like-Financial transaction, various govt. and private organizations, Shopping purposes etc.

Cyber security issues such as cyber threats are rising due to a lack of public awareness. A cyber threat refers to a poisonous move that illegally accesses data, steals data, information and tries to damage our digital life [1]. In the pursuit of necessity, as web application requirements increase, so do the vulnerabilities of web applications. To make a secure web application, some criteria need to be maintained because security issues come up from the structure of the web application.

According to some study, it is showing that many web developers don't pay close attention to the high risk of SQLi while creating web pages/applications [2]. SQL injection vulnerability has been at the forefront of web vulnerabilities since 1998. But still, web developers don't take care of this vulnerability. That's why it is still going strong and becoming more popular among web vulnerabilities. SQLi is a code-based technique that allows an attacker to view, interfere and ruin backend databases.

As Bangladesh is moving towards a developed country, as citizens of Bangladesh, we should fully know any weaknesses in the country's website and make everyone aware of this. In this project, we selected some different category websites of Bangladesh to detect SQLi websites vulnerability.

**1.2 Motivation:** Over the past 20 years, many Govt. & non-Govt. Institutions made a massive loss due to SQL injection attacks. Let us look back at some attacks that led to severe data breaches. Here are some examples of unprecedented attacks where hackers used SQLi - Ghostshell attack (In 2011, APT team Ghostshell targeted 53 universities and stole 36,000 personal records of students, faculty, and staff), Turkish Govt. (In 2013, APT group, Redheck Collective targeted the

Turkish Govt. website and erased debt to Govt. agencies),7-Eleven breach (In 2007, an attackers team targeted several companies, primarily 7-Eleven retail chain, and stole 130 million credit card numbers), HBGary breach (In 2011, hackers targeted HBGary to take down the IT security company's website) [3]. Now let us take a look at some recent SQL injection attack- Fortnite Vulnerability (in 2019, a SQL vulnerability was discovered in an online game called "Fortnite," which could let attackers access user accounts) [3]. Freepik data breach (In 2020, the attackers stole about 8.3M records (emails and passwords) from Freepik and Flaticon users using SQLi) [4].

**1.3 Rationale of the Study:** Though a lot of SQLi attacks are happening here and there, still a lot of institutes don't take it seriously. They are not yet conscious of the losses that can be caused by their lack of awareness. Recently in Bangladesh, 147 public and private organizations including banks and NBFIs (Non-Bank Financial Institutions) came under an attack showing their utter vulnerabilities [5]. To avoid such attacks, we should be fully aware of our own flaws in our websites. If we find any vulnerabilities in the website, we should take necessary action to protect the website.

**1.4 Research Questions**

1. How does Sqlmap work to detect SQL injection vulnerability?
2. How many vulnerabilities can Sqlmap check?
3. How much security our Bangladeshi websites maintain in terms of SQL injection?

**1.5 Expected Output:** The expected outcome from this research paper is to identify vulnerable websites of Bangladesh and to raise awareness about this matter. This research will help determine whether the BD websites are vulnerable by SQLi or not. We took and detected different BD websites to check the website vulnerability.

**1.6 Project Management and Finance:** Since our project is network based, no finance or grant is there to support this research. We are on our own. Some essential objects that we use in this research are-

- Laptop/ Computer
- Internet

**1.7 Report Layout**: The structure of this report is as follows. Chapter 1 covered the importance of SQL injection and how it is still a vigorous cyber-attack to date. It is also covered with some recent and some remarkable SQL injection attacks. Chapter 2 presented the background. This chapter covered our challenges and some related works, which we used as references in our report. Our experiences are covered in chapter 3. We've discussed the results and discussion in chapter 4. Chapter 5 consisted of the impact of the environment and society. At the end of chapter 6, we described the conclusion and implication of the further study.

# CHAPTER 2

## BACKGROUND

**2.1 Preliminaries/Terminologies:** SQL injection attack is a type of cyber-attack in which an attacker utilizes vulnerabilities in web applications by applying SQL query with the intention of stealing, deleting, destructing data. Shortly, attacker try to gain illegal access over the system [6]. Jeff Forristal, a hacker and cyber security researcher, originally revealed SQL injection in 1998. Even though it has been documented for more than 20 years, SQLi is still the most common vulnerability, according to OWASP (Open Web Application Security Project) [7]. SQLIA can classify into three different categories: In-band, Inferential & Out of band SQLi [2]

**1. in-band SQLi:** It's a prevalent and standard technique. When the attacker can use the same communication channel to begin the attack and view the result, this attack occurs. Union-based SQLi and Error-based SQLi are the common In-band SQL injections.

**Error-based SQLi:** The database is forced to return error messages when the error-based injection is used. [8]

**Union-based SQLi:** This method employs the UNION SQL operator, which combines multiple select statements generated by the database to produce a single HTTP response. This response may contain information that the attacker can use.

**2. Inferential SQLi:** In this kind of attack, the attacker can't use the same communication channel to view the results and view the result because, in this type of attack, the attacker can't retrieve data from the database. The attacker can only reorganize the database structure. It is also known as Blind SQL injection. Boolean and time-based are the most common inferential SQL injections.

**Boolean-based blind:** It is also known as a content-based blind injection. SQLi attacks based on content/Boolean logic cause the web application to return different results depending on whether the malicious SQL query returns a TRUE or FALSE result.

**Time-based blind:** A query sent by the attacker forces the application to wait for a certain amount of time before returning a response. The attacker uses the response time to determine whether the result of a query is TRUE or FALSE.

**3. Out of band SQLi:** This category is opposite to In-band SQLi. When the attacker cannot use the same communication channel to start the attack and view the result, this type of attack occurs. Out of band SQL injection attacks are not widespread.

Sqlmap is one of the most popular tools to detect SQLi and exploitation. We usually use it to prevent overloading the web server or being blocked by IPS/WAF devices by favoring manual detection. It is an open-source penetration tool where a user/attacker can detect SQLi vulnerabilities in web applications. Once an exposure appears on that particular website, the user can dump the whole or any appointed tables/columns of the database, retrieve data, edit data from the database, check specific files and many more. Sqlmap can automate basic SQL injection techniques like error-based, Boolean-based, union-based, union-based, time delay, and stack queries.

### 2.2 Related Works

So far, a lot of work has been done regarding SQL injection attacks. In [9], the authors work with the security and privacy of online users related to the Turkish government sites, which positively affects the functionality and accessibility of the Turkish govt. Website.

In [2], the authors presented how the lack of input variable filtering can easily detect SQLIA on any website. They have emphasized the issue of input variable filtering to secure databases here and proposed a technique called CombinedDetect to improve filtering based on JavaScript.

In [10], the authors described the SQLi vulnerabilities web application in the BD domain, where User-based input checked the vulnerabilities of web applications.

In [8], the authors defined the mechanism of cyber resilience and the present condition of the BD government. Also, it showed a statistical analysis of various cyber-attacks on the website of Bangladesh.

In [11], the authors presented a survey; this survey sets up a strategy to protect against SQLi and XSS.

The authors [12] compared three SQLi prevention methods using product declaration, stored policy, and input confirmation.

In [13], the authors emphasized the security of the Government websites of Kosovo. They

presented three different scenarios to test the security system using various tools like acunetix and Sqlmap and found out which system scenario performs better.

In [14], the authors discussed how the machine learning approach is one of the best approaches to detect SQLi. They proposed a heuristic approach to detect SQL injection attacks.

In [15], the authors presented their experiments about five different security tools and discussed the limitations and extensions of each tool as well.

In [16], the authors proposed a scanner to improve the effectiveness of SQL and check the accuracy, and the authors took three more different tools to compare with the proposed scanner.

### 2.3 Comparative Analysis and Summary

Table-2.1: Reference paper analysis and summary

| **Author** | **Observed Attack type** | **Organization** | **Method** | **Tools** | **Comments** |
|---|---|---|---|---|---|
| Akgül Y | SQLi, Cross Site Scripting and more | Turkish Govt. Website | Survey based | Scanner Tools (Acunetix, Netsparker) | Ensures the security and privacy of online users related to government sites. |
| Thiyab et al | SQL queries, SQL Injection Attack | Dynamic Web Application | Interpretative base | Javascript & PHP | The Proposed technique to secure database was demonstrate but no comparison look was shown before and after applying the technique |
| Bhuiyan et al | Error based, Union Based, Double Query Injection | Govt, Private, Educational, Bank, Financial, HealthCare, E-commerce | Statistical analysis | Penetration Testing (Black Box) Method | All tests are implemented manually. No specific tools were used. |
| Johari and Sharma | SQLi, XSS | Security Engine | Survey based | Automatic test case generation for SQLi detection, Dynamic cookie re-writing for XSS detection | Presented a survey, which sets up a strategy to protect against SQLi and XSS. |
| Kumar and Anaswara | SQL injection attack, | Vulnerability detection and | Compare & Contrast base | Prepared statement(para | 3 SQLi prevention methods are compared. |

| | | | | | |
|---|---|---|---|---|---|
| | Vulnerability detection, Vulnerability prevention | prevention of any website | | meter Queries), Stored Procedure, White List input Validation | |
| Alam et al | SQL Injection Attack | .bd Domain | Analytical | Grouped the get and post based application | Describe the SQLi vulnerabilities existence web application in BD domain |
| Maraj et al | SQL injection, SQLi Attack | Govt. network of Kosovo | Compare & Contrast base | Acunetix,Sqlmap,Burp suite | Used 3 different Scenarios to test the security system and found the best system for protecting sensitive data |
| Hasan et al | SQL injection | Proposed a system for any web Application | Machine Learning | Heuristic approach | 23 classifiers were used to evaluate the proposed system and this system examines the flow of queries |
| Elia et al | SQL injection, Fault injection | comparison of different tools | Experimental | Scanner Tools (Acunetix, Apache, Mysql) | An exploratory test of 5 different security tools to detect SQL injection attacks |
| Aliero et al | SQL injection | Online farm application, Online news application and many more vulnerable application | Experimental | SQLIV, Nikito, Acunetix WVS, Vega, zap, Wapiti and many more. | Examine the accuracy of proposed scanner and compare the result with other different scanners |

**2.4 Scope of the Problem:** An attacker can use a vulnerability to obtain access to all of the databases on a website. The attacker utilizes application code to exploit the SQLi vulnerability, destroying the website's database.

After reading several research papers, we discovered that each author attempted to ensure the website's security in a unique way. Someone has explained how SQLi can be detected by filtering input variables. Many authors have mentioned various penetration methods to see vulnerability, such as the black box and white box methods. Some of them expressed concern about government websites and explained how security could be provided on these sites. A paper described how SQLi vulnerability could be checked using machine learning. In the vast majority of cases, we discovered that software used to scan for vulnerabilities was compromised.

We concluded after reading them all that there is no proper website security (SQLi) work for Bangladesh with Sqlmap. So, if we can use Sqlmap to separate the specific websites affected by the SQLi vulnerability from Bangladesh's websites, we can ensure the affected websites security. This is not software-based work because we used the Sqlmap tool in Kali Linux. During our research, we examined the SQLi vulnerability of BD websites to determine the type of vulnerability.

**2.5 Challenges:** Everywhere we face some challenges. Same here, we had to face some challenges which are below here-

When we want to scan many websites in a short time, it is seen that the whole work cannot be done correctly, which has to be scanned again later, which is challenging.

Secondly, when we try to classify the errors, which is a big challenge for us.

# CHAPTER 3

## RESEARCH METHODOLOGY

**3.1 Research Subject and Instrumentation:** The research subject of our study is 'SQL'. The Structured Query Language (SQL) queries, operating, and administration language for database applications. Frequently, user-supplied data is used to generate the SQL statement that accesses the database. The attackers can alter or manipulate the data provided by the user, gaining access to the database as a result. Attackers can change the SQL statement by substituting the user's supplied data with their data using SQL injection. As a result, attackers can gain direct access to a database server and obtain sensitive data [14].

This investigation discovered vulnerabilities using Sqlmap, a well-known tool for detecting SQL injection in any website. It's an open-source penetration tool for detecting SQLi vulnerabilities in online applications by a user/attacker.

**3.2 Data Collection Procedure/Dataset Utilized:** The processes for collecting data and checking for vulnerabilities are set up in order. We utilized the penetration testing tool "Sqlmap" to find the vulnerable web application. The most common URL format is "inurl:php?id= ". If the 'GET' parameter is highlighted, the website may be vulnerable to SQL injection in this mode, and an attacker may access data in the database. Furthermore, Sqlmap is usable when it is built on PHP. Here we have used this URL format to check the website's vulnerability.

After identifying the web application, we analyzed each website individually for vulnerabilities and grouped the injections based on the error messages. The data was then evaluated that used the GET and POST methods.[10]

### 3.3 Statistical Analysis

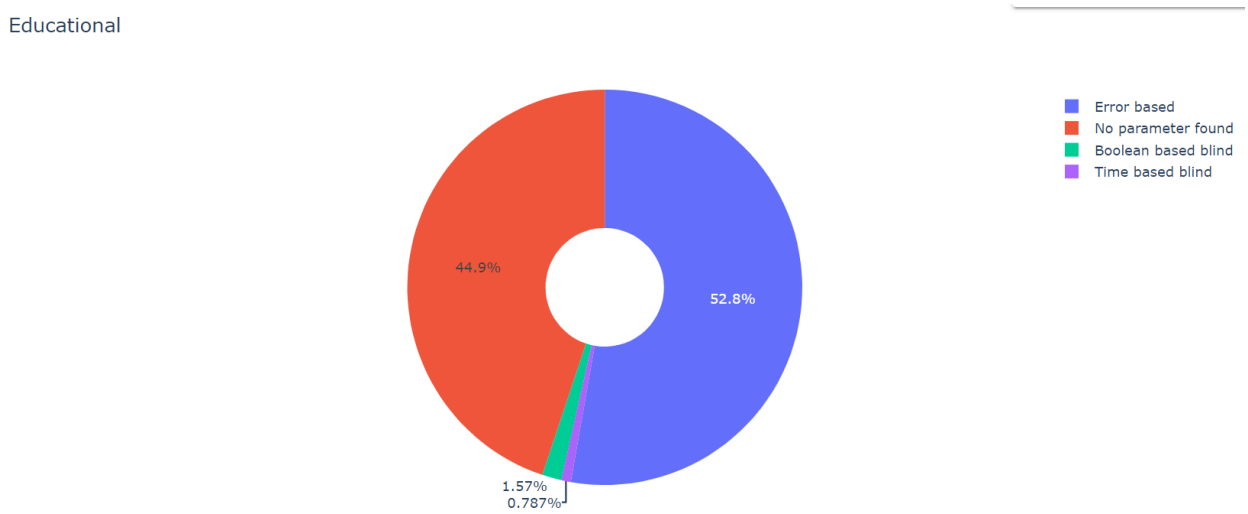



Figure-3.1: Visualization of Educational websites SQLi vulnerability of Bangladesh

We used Sqlmap to scan the websites of more than 125 educational institutions in Bangladesh, and the results are shown in Figure-3.1.

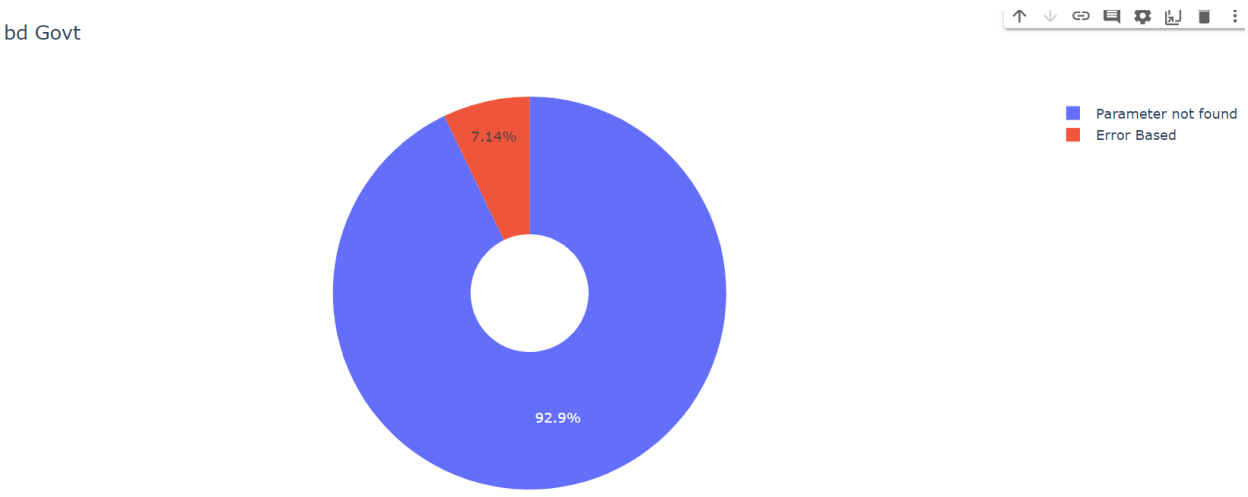



Figure-3.2: Visualization of BD Govt. websites SQLi vulnerability of Bangladesh

We used Sqlmap to scan the key websites of the Bangladesh government, and the results are given in Figure-3.2.

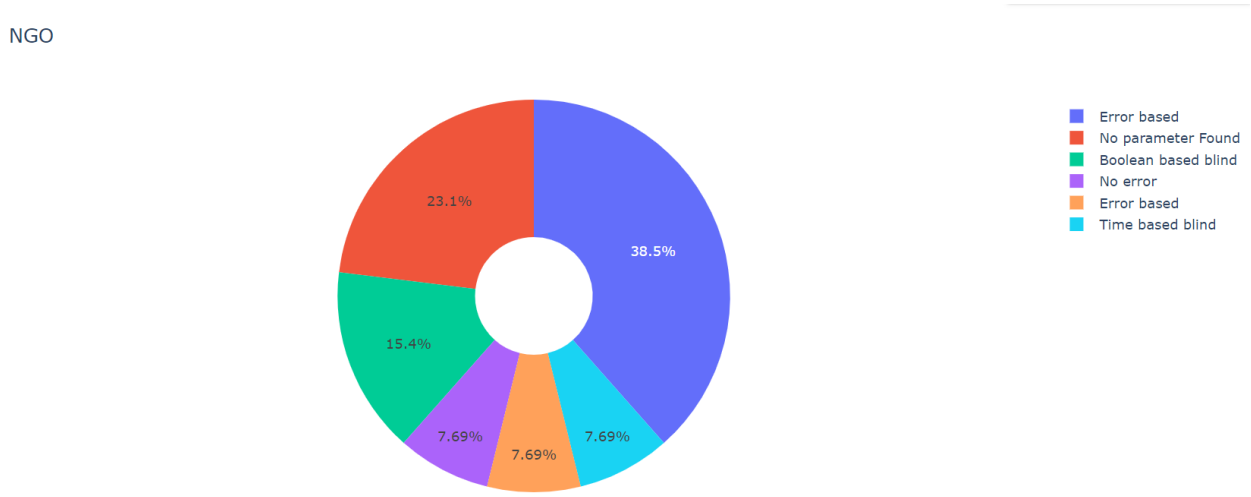



Figure-3.3: Visualization of BD top NGOs websites SQLi vulnerability of Bangladesh

We used Sqlmap to scan the top websites of Bangladeshi NGOs, and the results are presented in Figure-3.3.

## 3.4 Proposed Methodology

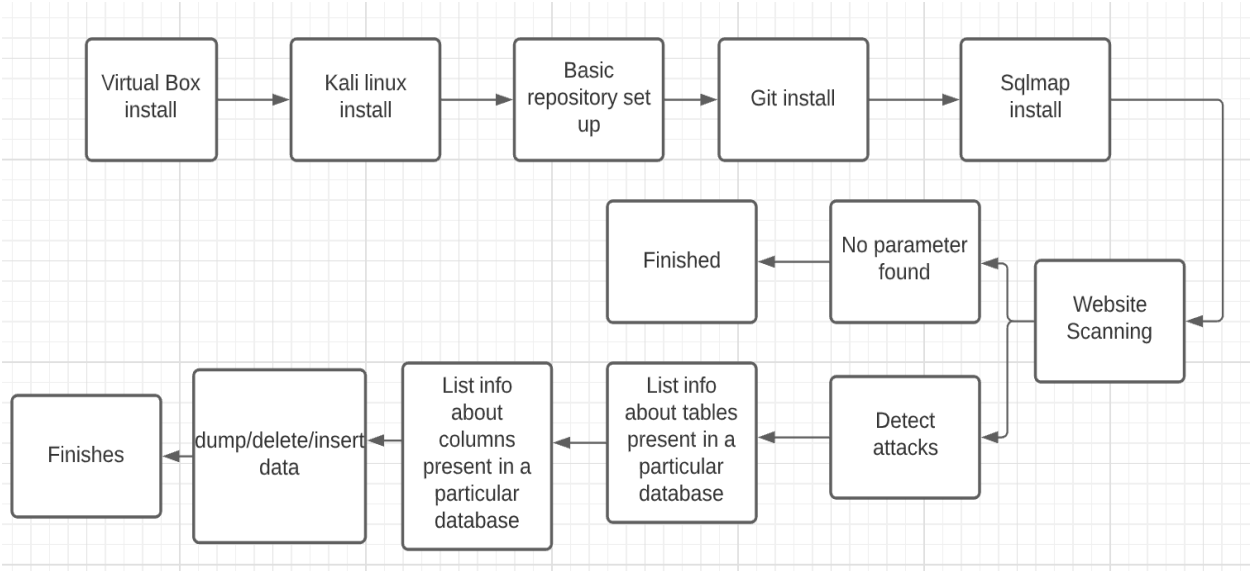



Figure-3.4: Flow diagram of experimental process

**3.5 Implementation Requirements:** After installing kali linux we installed Sqlmap to detect SQLi vulnerabilities of web applications. Sqlmap is one of the biggest platforms to detect SQLi vulnerabilities. To know if it's successfully installed or not we can use the command "sqlmap -h". If it's successfully installed then it will execute the command and show some basic details (figure-3.5). To scan any web applications with Sqlmap we have to maintain a format to detect SQL injection vulnerability. Here is the format- "inurl:php?id=".

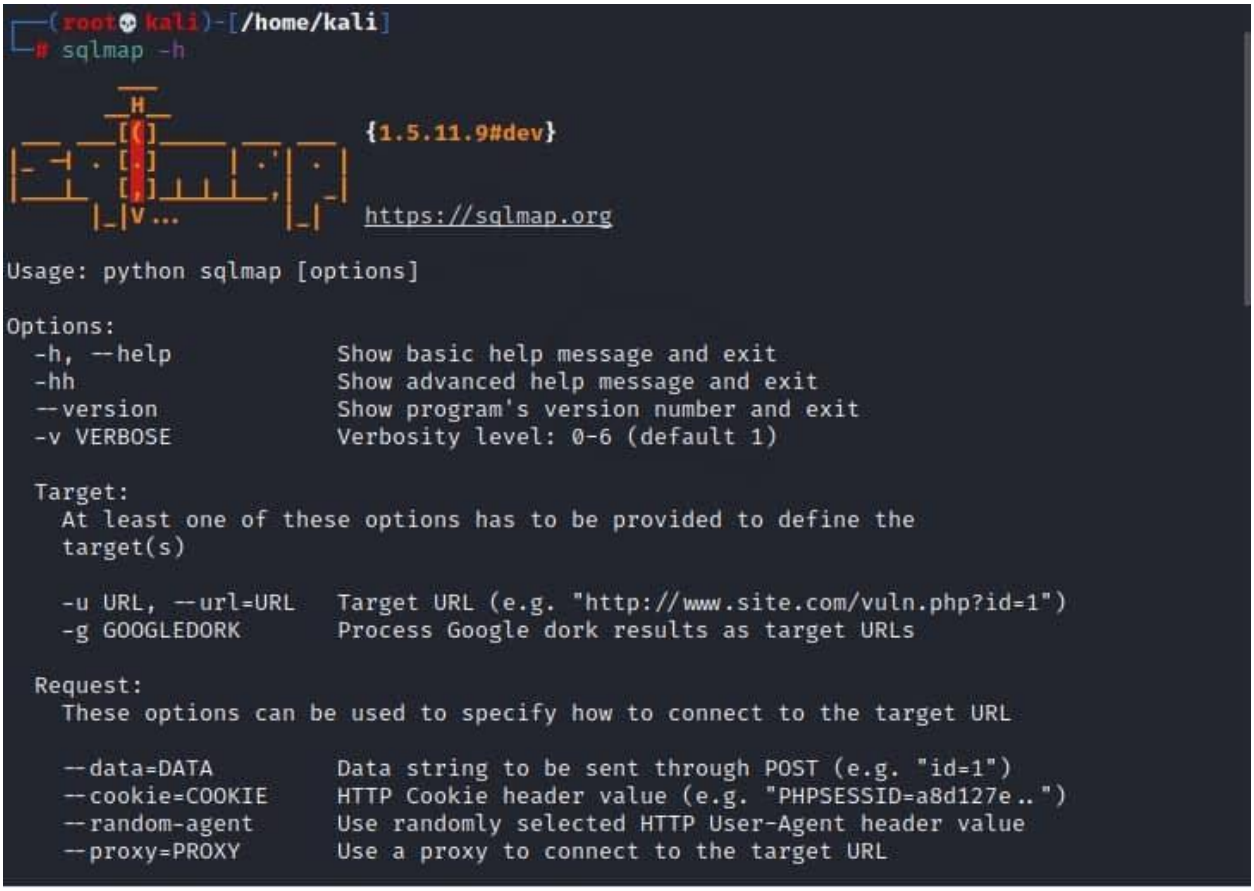



Figure-3.5: SQLMAP parameter

Figure 3.5 shows that it's successfully installed and shows some basic parameter details, commonly known as Sqlmap query.

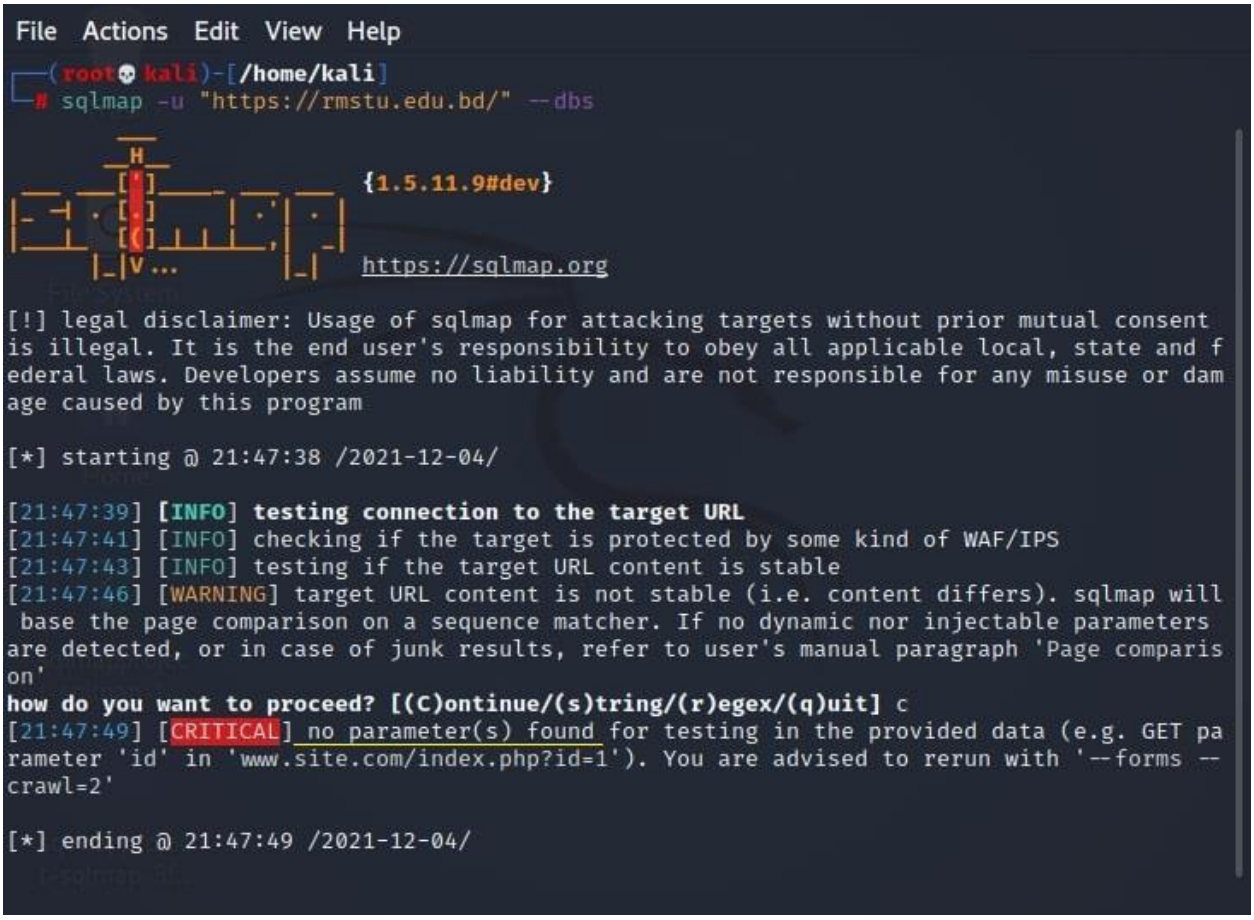



Figure-3.6: Command for fetch database (didn't detect attack)

In Figure-3.6, we can see the URL (https://rmstu.edu.bd/) did not satisfy Sqlmap URL format which led not to show any expected output. It also gave feedback about the targeted URL such as- in above figure shows that the problem with this executed URL was its ID. It also shows the connection stability of the targeted URL.

When Sqlmap is unable to locate difficult injection locations, several configuration changes are needed. But if it does satisfy the format then it will show an outcome.

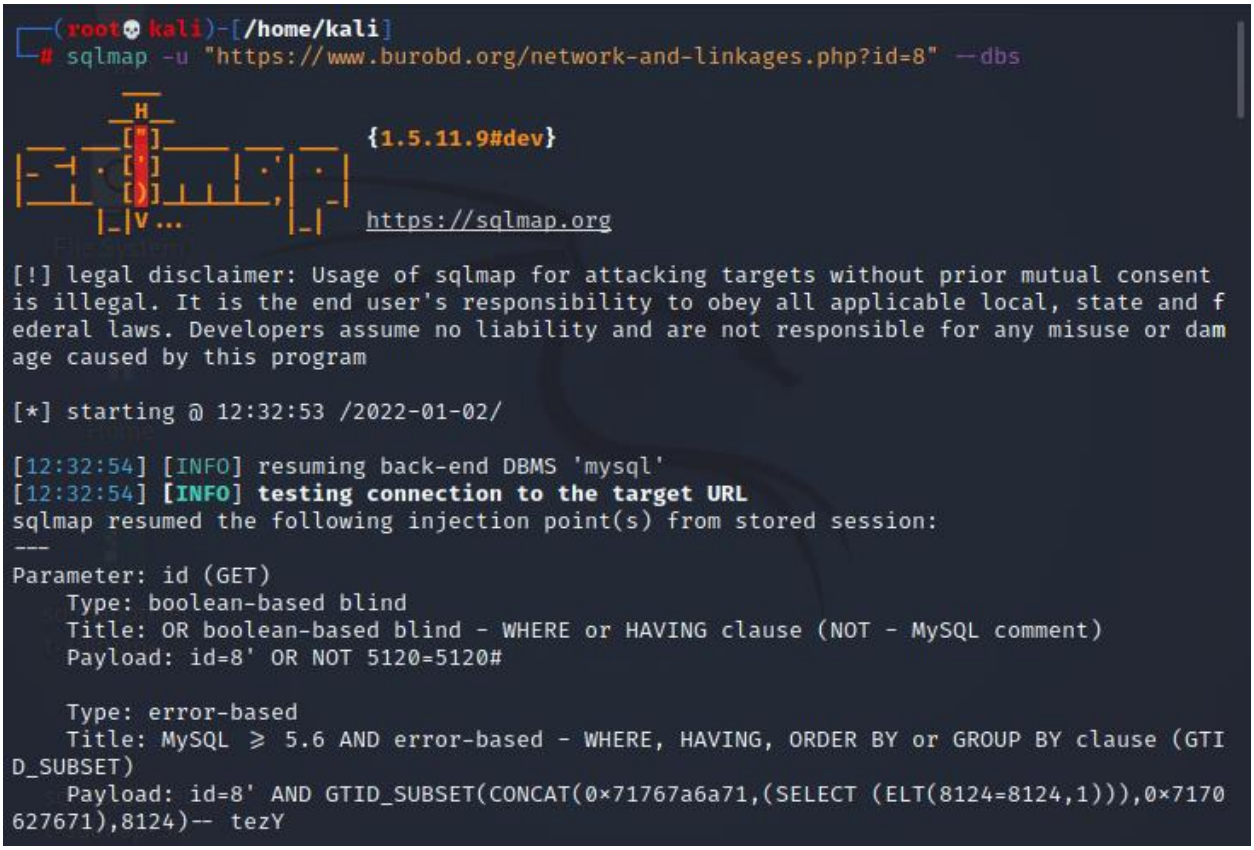



figure-3.7: Command for fetch database (detect attack)

In Figure-3.7, we can see the URL (https://www.burobd.org/network-and-linkages.php?id=8) satisfied Sqlmap URL format which led to the expected output. It shows the SQL injection types after getting executed successfully. In the above figure the lines marked below show the type of SQLi attack.

We can see in the above figure that we have entered the web URL to see if we have access to the databases of that URL. And it will display a list of all available databases.

# CHAPTER 4

## EXPERIMENTAL RESULTS AND DISCUSSION

### 4.1 Experimental Setup

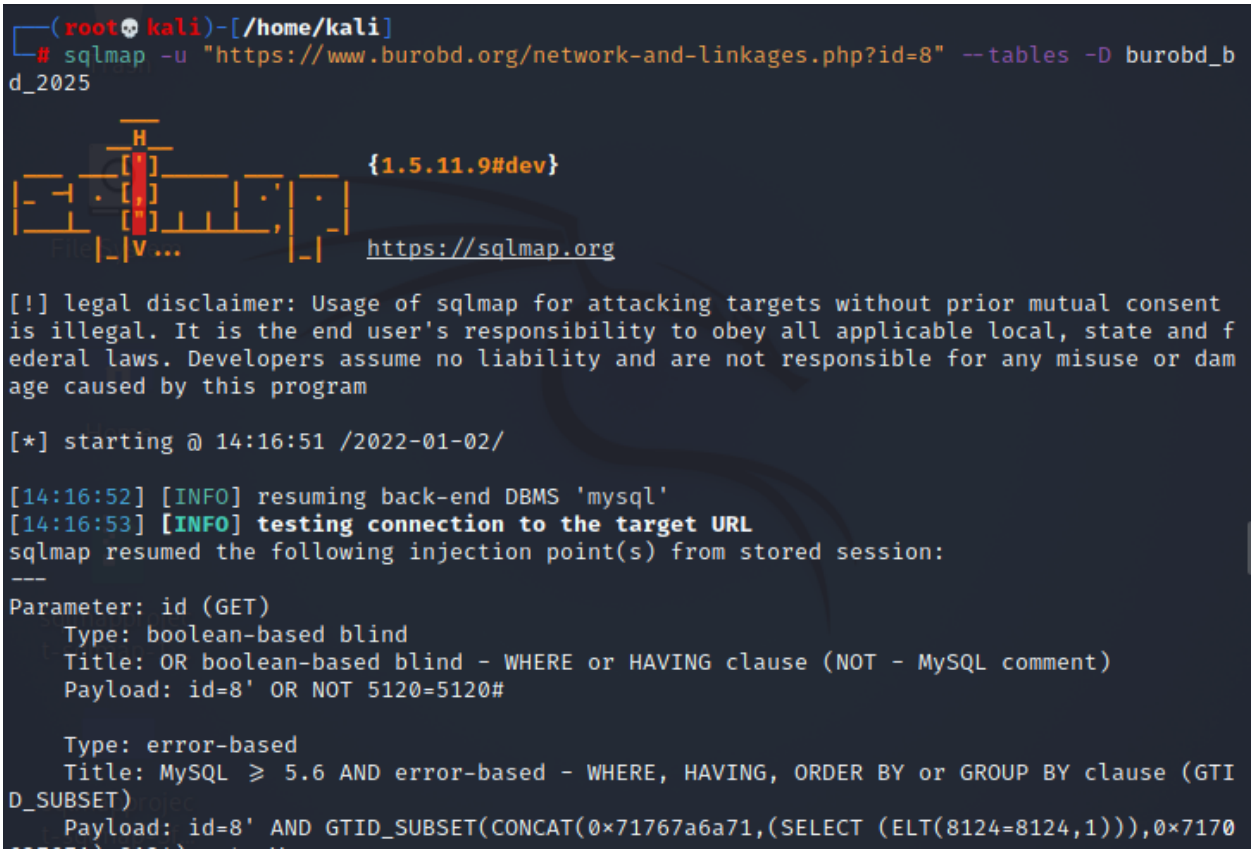



Figure-4.1: List information about Tables present in a specific database

In figure 4.1 we can see, for trying to access any of the databases, we ought to change our command moderately. We now enter the name of the database we want to access through “-D”, and even if we have access to the database, we want to see if we can access the tables. The “--tables” query is also used for this. We attempted to enter the database 'burobd_bd_2025' here.

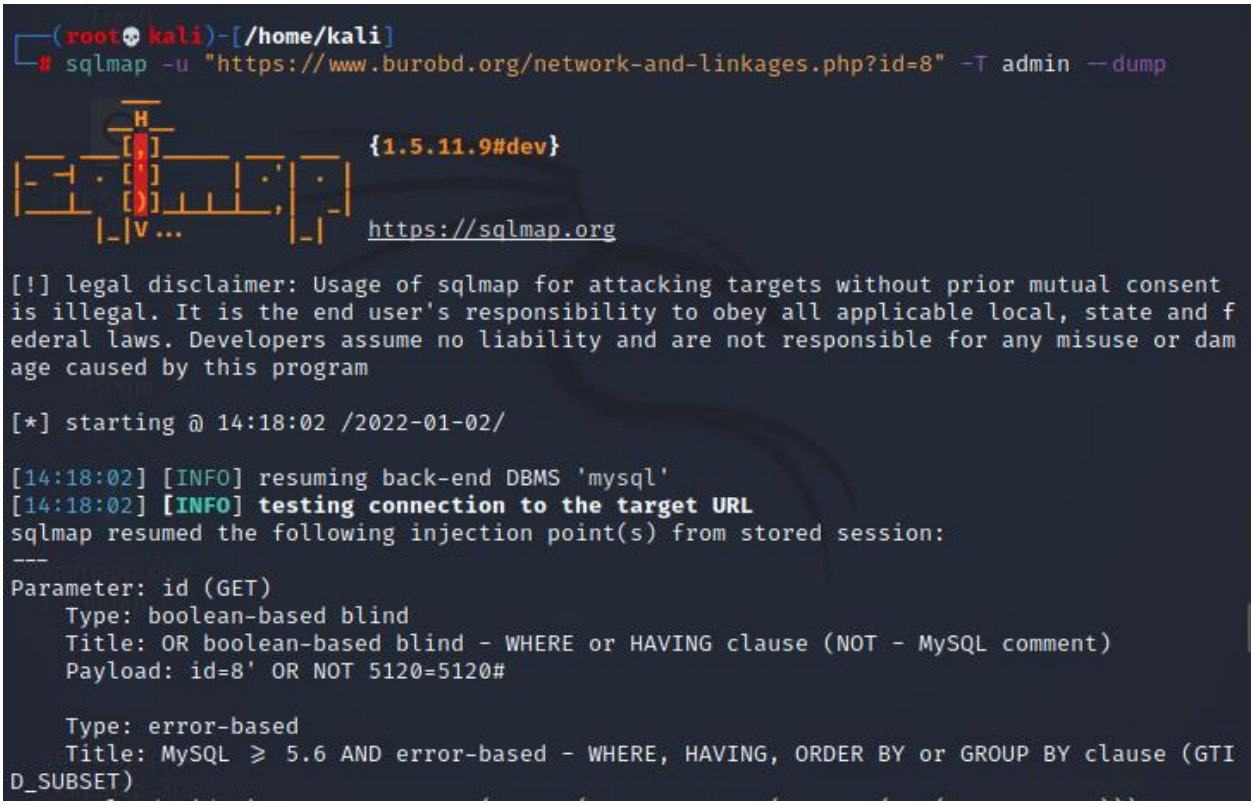



Figure-4.2: Using command for dump the data

Figure 4.2 shows a command which allows us to obtain data from a certain table while also collecting dump query data. We can also dump data from a particular column. We attempted to dump data from the "admin" database in this example.

### 4.2 Experimental Results & Analysis

The results of the experiments are presented and analyzed in this section. The setup was shown in the previous section. The results of the tool in web application testing are presented in this section.

This section also clarified how a hacker gains access to a website's databases after confirming a SQL injection vulnerability.

```
[12:33:00] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.6.40
back-end DBMS: MySQL ⩾ 5.6
[12:33:00] [INFO] fetching database names
[12:33:00] [INFO] resumed: 'information_schema'
[12:33:00] [INFO] resumed: 'burobd_bd_2025'
available databases [2]:
[*] burobd_bd_2025
[*] information_schema

[12:33:00] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/outp
ut/www.burobd.org'

[*] ending @ 12:33:00 /2022-01-02/
```

Figure-4.3: Show fetched databases

Figure 4.3 demonstrates that the URL used in figure 3.4 was able to access the databases, and there are two backend databases available. The databases are: information_schema, burobd_bd_2025. While scanning, the application may ask if you want to search other databases. If the 'Y' key is pressed here, the web application can be thoroughly examined.

```
[14:16:53] [INFO] the back-end DBMS is MySQL
web application technology: Apache, PHP 5.6.40
back-end DBMS: MySQL ⩾ 5.6
[14:16:53] [INFO] fetching tables for database: 'burobd_bd_2025'
Database: burobd_bd_2025
[31 tables]
+-------------------------+
| dynamic-page-code       |
| zone                    |
| admin                   |
| annualReports           |
| annualreports           |
| contactus               |
| gallery                 |
| galleryparallax         |
| header                  |
| homepage                |
| homesectionname         |
| homesections            |
| imagealbum              |
| isbkmap                 |
| job                     |
| linkheadings            |
| links                   |
| loginlinks              |
| map                     |
| news                    |
| noticeboard             |
```

Figure-4.4: Fetched 31 tables from specific database

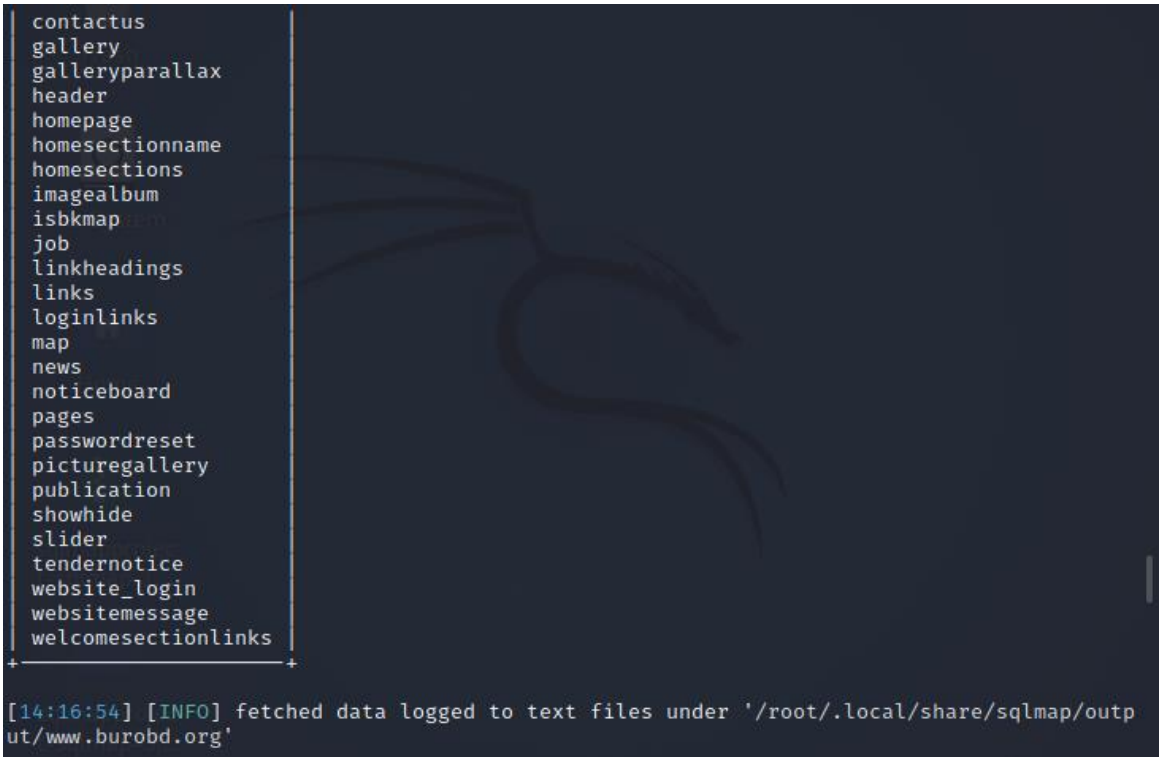



Figure-4.5: Fetched 31 tables from specific database

Figure 4.4 and 4.5 shows that 31 tables have been retrieved, indicating that this website is vulnerable.

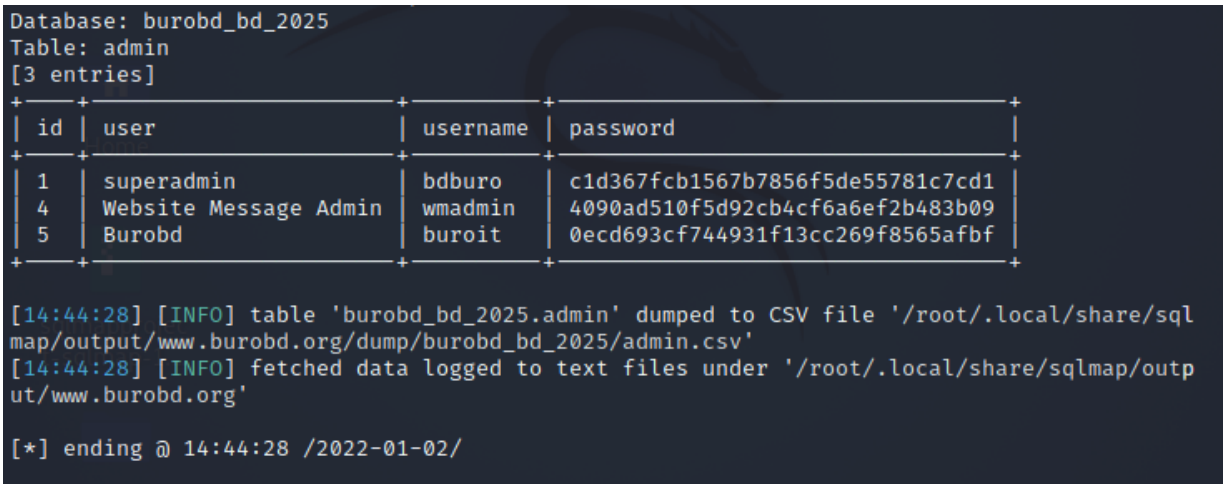



Figure-4.6: Result of dump data from specific table

In figure-4.6, we successfully retrieved the data despite the fact that the password is a bit jumbled and inaccurate, but we can check other vulnerable databases for data by using the format '#sqlmap -u"website url"-T table name --dump' that we executed earlier (figure-4.2).

**4.3 Discussion:** We looked at both public and private websites in Bangladesh for our study. In order to uncover SQLi vulnerabilities, we worked with a total of 150 websites. Not all websites are vulnerable to the same degree. Web applications from diverse genres, such as educational institutes, private NGOs, and government sites, are chosen for analysis. There are ten vulnerable online apps based on NGO, 14 vulnerable web applications based on government institutes, and 126 vulnerable web applications based on educational institutes, according to our research employing a data set for discovering insecure web applications.

There were no parameters detected on four of the ten NGO sites, while the remaining six were error-based, time-based, and Boolean-based attacks. No parameters were found on 13 of the 14 government sites, and only one error-based assault was detected. There were no parameters discovered on 57 of the 126 educational sites, 67 were error-based assaults, and two were Boolean-based attacks. There are two categories of data in our dataset: injected and non-injected. The findings reveal that an expert administrator does not properly maintain Bangladeshi websites [14].

# CHAPTER 5

## IMPACT ON SOCIETY, ENVIRONMENT AND SUSTAINABILITY

**5.1 Impact on Society**: The number of people who use social media apps is currently incalculable. The popularity of these apps is growing by the day. Social media apps have evolved into an excellent platform for entrepreneurs to promote their products and services. Web applications are now the backbone of today's businesses and are widely used. As web-based database applications, nearly all information systems and business applications (e-commerce, banking, transportation, webmail, blogs, and so on) are now available. The demand for web applications attracts adversaries looking to exploit their flaws [15].

Given the level of sensitive information handled by web applications, it is expected that all web applications be secure and resilient to various types of attacks [17]. This study will assist people in understanding website security. This research will increase public awareness of security and privacy issues as well. It will also increase web users' and web developers' awareness about the SQL injection vulnerability in web applications.

**5.2 Impact on Environment:** In terms of technology, we, the citizens of Bangladesh, are still far behind the citizens of other countries. We still have a long way to go before keeping up with the new world. However, Bangladesh is now much more prosperous than it was previously. The number of web pages in Bangladesh has skyrocketed in recent years. SQL Injection is extremely profitable for attackers, and there is a thriving black market for all types of digitally stolen goods, such as credit card numbers and bank accounts [15]. By deliberately injecting an SQLIA into the web application, this tool demonstrates potential vulnerabilities and assists developers in detecting design flaws [18]. This study will help make the environment better and make people more aware of the situation.

**5.3 Ethical Aspects:** A SQL injection attack is a legal offense that involves getting unauthorized access to a website's database. Attackers or cybercriminals use SQL injection to access any website to steal sensitive information [19]. Our primary purpose is to protect people from harmful behavior. Everyone should proceed with caution and knowledge when using websites.

**5.4 Sustainability Plan:** SQL injection attacks, and web-based attacks in general, are still a serious challenge in securing financial, health, and other sensitive data. The problem is only getting worse as more social activities rely on the internet. In the future, we can use a multi-source data analysis approach in this project to improve the accuracy of detecting SQL injection attacks [20].

Another thing we can do in the future is investigate more tools for detecting vulnerabilities in Bangladeshi websites and investigate more vulnerabilities in Bangladeshi websites [21].

# CHAPTER 6

## SUMMARY, CONCLUSION, RECOMMENDATION AND IMPLICATION FOR FUTURE RESEARCH

**6.1 Summary of the Study:** Humans have entered the information age in the twenty-first century, thanks to the continuous advancement of computer science and technology. The Internet is extremely important in this day and age. The Internet has become an important medium of communication between the two, whether it is for the exchange of information between people or the communication of business between enterprises. Simultaneously, network information security concerns are becoming more visible. More and more web applications based on database technology have become the target of attackers. No web application can claim to be 100% secure [22].

As a developing country, Bangladesh is also working to improve the quality of technology available to its citizens, yet concerns regarding web application security remain unmet. In this study, we look at a variety of web applications to see which ones are vulnerable to SQL injection. We used Kali Linux in this paper because it is a versatile and user-friendly operating system used to perform seamless and effective penetration testing and security auditing tasks. This fully open-source operating system allows users to carry out exploits in a variety of ways. As a tool, we used Sqlmap, a penetration testing tool. The primary reason for performing penetration testing on web pages is to identify vulnerabilities before an attacker does and fix them quickly [23].

**6.2 Conclusions:** In this paper, the detection of SQL injection attacks was tested. This test, which was performed on various types of Bangladesh websites, demonstrated that understanding a website's vulnerability is essential. The tool we used here has a variety of features, which is a plus for us. Another advantage of using the tool is that it employs the same technology as hackers but in a legal manner. Any website's SQL injection vulnerability is an excellent choice for attackers looking to acquire data from a website. That is why it is vital to assess website vulnerability so that we can take appropriate steps to protect websites from SQL injection attacks [24]. This research focused on a practical way to find SQLi vulnerabilities that are widely found in Bangladeshi websites, in addition to theoretical principles [25]. Our main goal was to find the SQL vulnerabilities in different web apps of Bangladesh while making it a simple and efficient practice for developers and software testers as well as the user [26]. Using this method will significantly

reduce the SQLi attack. As a result, we will merge the existing feature pattern matching method for SQL injection detection in the future study. We're attempting to improve our work using machine learning methods.

**6.3 Implication for Further Study:** All of the preliminary steps of the procedure that are required for future work have been fully utilized in our work. Since we have worked to detect web application vulnerabilities (SQLi attacks), we can work on how to prevent SQLi attacks in the future. Which method produces the best results?

We'll be concentrating our efforts in the future on detecting more serious vulnerabilities such as remote code execution, blind XSS, and others. We intend to investigate a variety of other web application vulnerabilities on each website in order to assess the potential cost of attacks and machine learning-based vulnerability detection approaches [26].

## References


[1] Upguard.com. 2021. *What is a Cyber Threat? | UpGuard*. [online] Available at: <https://www.upguard.com/blog/cyber-threat> [Accessed 7 August 2021].

[2] Rua Mohamed Thiyab, Musab A. M. Ali, Farooq Basil, & Abdulqader. (2017). The impact of SQL injection attacks on the security of databases in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 323-331). Sintok: School of Computing.

[3] NeuraLegion. 2021. *SQL Injection Attack: Real Life Attacks and Code Examples - NeuraLegion*. [online] Available at: <https://www.neuralegion.com/blog/sql-injection-attack/> [Accessed 4 August 2021].

[4] Gatlan, S., 2021. *Freepik data breach: Hackers stole 8.3M records via SQL injection*. [online] BleepingComputer. Available at: <https://www.bleepingcomputer.com/news/security/freepik-data-breach-hackers-stole-83m-records-via-sql-injection/> [Accessed 7 August 2021].

[5] Express, T., 2021. *Latest cyber attack hit at least 147 Bangladeshi entities*. [online] The Financial Express. Available at: <https://thefinancialexpress.com.bd/sci-tech/latest-cyber-attack-hit-at-least-147-bangladeshi-entities-1617416432> [Accessed 8 August 2021].

[6] 2021. [online] Available at: <https://www.acunetix.com/websitesecurity/sql-injection/> [Accessed 9 August 2021].

[7] Sucuri. 2021. *OWASP Top 10 Security Vulnerabilities 2021 | Sucuri*. [online] Available at: <https://sucuri.net/guides/owasp-top-10-security-vulnerabilities-2021/> [Accessed 11 September 2021].

[8] Bhuiyan, T., Alam, D. and Farah, T., 2015. Evaluating the Readiness of Cyber Resilient Bangladesh. *Journal of Internet Technology and Secured Transaction*, 4(3), pp.405-415.

[9] Y. Akgul, "Web Site Accessibility, Quality and Vulnerability Assessment: a Survey of Government Web Sites in the Turkish Republic", *Journal of Information Systems Engineering & Management*, vol. 1, no. 4, 2016. Available: https://www.researchgate.net/profile/Yakup-Akguel/publication/309735856_Web_Site_Accessibility_Quality_and_Vulnerability_Assessment_a_Survey_of_Government_Web_Sites_in_the_Turkish_Republic/links/5821f77208aed9ccec6382b5/Web-Site-Accessibility-Quality-and-Vulnerability-Assessment-a-Survey-of-Government-Web-Sites-in-the-Turkish-Republic.pdf. [Accessed 19 July 2021].

[10] D. Alam, M. Kabir, T. Bhuiyan and T. Farah, "A Case Study of SQL Injection Vulnerabilities Assessment of .bd Domain Web Applications", in *2015 Fourth International Conference on Cyber Security , Cyber Warfare, and Digital Forensic*, 2015.

[11] R. Johari and P. Sharma, "A Survey On Web Application Vulnerabilities(SQLIA,XSS)Exploitation and Security Engine for SQL Injection", in *International Conference on Communication Systems and Network Technologies*, 2012.

[12] B. Kumar and A. P.P., "Vulnerability detection and prevention of SQL injection", 2018.

[13] A. Maraj and E. Rogova, "Testing Techniques and Analysis of SQL Injection Attacks", in *2nd International Conference on Knowledge Engineering and Applications*, 2017.

[14] Hasan, M., Balbahaith, Z. and Tarique, M., 2019. Detection of SQL Injection Attacks: A Machine Learning Approach. In: *International Conference on Electrical and Computing Technologies and Applications*. [online] Available at: <https://ieeexplore.ieee.org/abstract/document/8959617/> [Accessed 11 September 2021].

[15] I. Alessandro Elia, J. Fonseca and M. Vieira, "Comparing SQL Injection Detection Tools Using Attack Injection: An Experimental Study", in *2010 IEEE 21st International Symposium on Software Reliability Engineering*, 2010.

[16] M. Aliero, I. Ghani, K. Qureshi and M. Rohani, "An algorithm for detecting SQL injection vulnerability using black-box testing", *Journal of Ambient Intelligence and Humanized Computing*, vol. 11, no. 1, pp. 249-266, 2019. Available: 10.1007/s12652-019-01235-z [Accessed 11 July 2021].

[17] O. Ojagbule, H. Wimmer and R. Haddad, "Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP", in SoutheastCon 2018, 2018.

[18] B. Nagpal, N. Singh, N. Chauhan and A. Panesar, "Tool based implementation of SQL injection for penetration testing," International Conference on Computing, Communication & Automation, 2015, pp. 746-749, doi: 10.1109/CCAA.2015.7148509.

[19] Abdus Satter, B M Mainul Hossain,"Vulnerabilities Assessment of Emerging Web-based
Services in Developing Countries", International Journal of Information Engineering and Electronic Business (IJIEEB), Vol.8, No.5, pp.1-8, 2016. DOI: 10.5815/ijieeb.2016.05.01 12:45 PM

[20] K. Ross, M. Moh, T. Moh and J. Yao, "Multi-Source Data Analysis and Evaluation of Machine Learning Techniques for SQL Injection Detection", in ACMSE '18: Proceedings of the ACMSE 2018 Conference, USA, 2018.

[21] P. Vamsi and A. Jain, "Practical Security Testing of Electronic Commerce Web Applications", in *Int. J. Advanced Networking and Applications*, 2021.

[22] Z. Xiao, Z. Zhou, W. Yang and C. Deng, "An approach for SQL injection detection based on behavior and response analysis," 2017 IEEE 9th International Conference on Communication Software and Networks (ICCSN), 2017, pp. 1437-1442, doi: 10.1109/ICCSN.2017.8230346.

[23] A. Ibrahim and S. Kant, "Penetration Testing Using SQL Injection to Recognize the Vulnerable Point on Web Pages", INdia, 2018.

[24] G. Su, F. Wang and Q. Li, "Research on SQL Injection Vulnerability Attack model," 2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS), 2018, pp. 217-221, doi: 10.1109/CCIS.2018.8691148. 12:08 PM

[25] A. Ciampa, C. Visaggio and M. Penta, "A heuristic-based approach for detecting SQL-injection vulnerabilities in Web applications", in ICSE, Cape Town South Africa, 2010.

[26] M. Moniruzzaman, F. Chowdhury and M. S. Ferdous, "Measuring Vulnerabilities of Bangladeshi Websites," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), 2019, pp. 1-7, doi: 10.1109/ECACE.2019.8679426.

SQL INJECTION VULNERABILITIES ANALYSIS USING SQLMAP A CASE STUDY OF BANGLADESHI WEBSITES

ORIGINALITY REPORT

| 9% | 7% | 3% | 6% |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Submitted to Daffodil International University<br>Student Paper | 2% |
| 2 | mecs-press.org<br>Internet Source | 2% |
| 3 | dspace.daffodilvarsity.edu.bd:8080<br>Internet Source | 1% |
| 4 | Olajide Ojagbule, Hayden Wimmer, Rami J. Haddad. "Vulnerability Analysis of Content Management Systems to SQL Injection Using SQLMAP", SoutheastCon 2018, 2018<br>Publication | 1% |
| 5 | Elia, Ivano Alessandro, Jose Fonseca, and Marco Vieira. "Comparing SQL Injection Detection Tools Using Attack Injection: An | 1% |