LONDON METROPOLITAN UNIVERSITY

Islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title
CC6051NI Ethical Hacking

Assessment Type
50% Individual Coursework

Semester
2022-2023 Spring

Practical Hacking Methods and Techniques

Student Name: Aadarsha Muni Shakya
London Met ID: 20049438
College ID: NP01NT4S210023
Assignment Due Date: 3rd May 2023
Assignment Submission Date: 2nd May 2023
Submitted To: Aditya Sharma
Word Count (Where Required): 2086

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Acknowledgment

I would like to express my sincere gratitude to all the individuals who have supported and contributed to this project. Their invaluable guidance, expertise, and encouragement have been instrumental in making this endeavor a success.

Abstract

In this report web application vulnerability related to SQL injection is covered. Moreover, contents related to what SQL injection is, how it is done and recommendation to mitigate SQL injections are provided.

Table of Contents

List of Figures

1. Introduction

SQL injection is a sort of cyber-attack that focuses on exploiting flaws in online applications that are linked to databases. Hackers install malicious code into an application's database using SQL, a database management language. This allows them to access and modify sensitive information such as user login passwords, financial information, and other secret data. (PortSwigger, 2023) Worryingly, this attack may be carried out remotely, without the attacker requiring physical access to the targeted machine. SQL injection attacks may cause considerable impact to enterprises, including financial losses, data theft, and reputational damage. As a result, it's critical for developers and organizations to understand SQL injection vulnerabilities and take the appropriate precautions to avoid them.

1.1. Current Scenario

SQL injection is still a serious worry in the cybersecurity sector, with hackers continuing to exploit this vulnerability. Despite frequent warnings, many businesses are still not doing enough to safeguard their databases. SQL injection attacks are still frequent, according to recent research, and fraudsters are discovering new techniques to circumvent existing security measures.

The growing usage of online apps and APIs, which enable new access points for hackers, exacerbates the problem. These apps are frequently created without enough security testing, making them vulnerable to SQL injection attacks. Furthermore, with the increased prevalence of IoT devices, attackers now have even more options to target web interfaces.

To overcome this challenge, firms must have a comprehensive security strategy. This involves doing frequent vulnerability assessments and penetration testing, as well as training developers on safe coding standards. Organizations may limit the risk of SQL injection attacks and secure sensitive data from unauthorized access by implementing these procedures.

1.2. Aim and Objectives

1.2.1. Aim

The aim of this report is to learn about vulnerabilities related to SQL injection, how it is done, and recommendations related to SQL injections.

1.2.2. Objectives

- Checking if websites are vulnerable to SQL Injection
- Learn to inject SQL in vulnerable web application
- Learn to bypass client-side filtering
- Learn about Burp Suit
- Learn about Server-side filtering

2. Background and Literature Review

2.1. Background

The study discusses a SQL injection vulnerability discovered on Mutillidae's website login page. SQL injection is a sort of attack that occurs when a website fails to properly validate the information that visitors enter the site. It gives attackers access to critical information and potentially gives them control of the site.

Mutillidae is a site that is intentionally created to be susceptible for individuals to practice detecting and addressing security flaws in a safe setting. Furthermore, the Burp suite is utilized to circumvent client-side filtering, allowing this assault on Mutillidae's login page to succeed.

In summary, the study goes into detail on the vulnerability discovered on Mutillidae's login page, as well as what individuals can do to avoid similar problems from occurring on other websites. It also demonstrates how attackers may exploit the vulnerability and how Burp Suite can assist make the assault more successful.

2.2. Literature Review

2.2.1. Case Study

Consider the following scenario: a hacker tries to gain unauthorized access to a web application by exploiting a flaw in the SQL injection of the password field on the login page. The attacker can obtain sensitive information from the application's database via a Boolean-based SQL injection approach. The attacker can enter the following information into the login page's password field: OR EXISTS(SELECT * FROM users WHERE name='sonakshi' AND password='%a%') AND ''='

This data is a SQL injection attack payload with a Boolean statement. The payload is meant to take advantage of a flaw in the system by adding a SELECT statement into the SQL query used to authenticate users. The attacker intends to gain access to information about the user "sonakshi" if her password begins with the letter "a." In the SQL query, the LIKE operator looks for any instances of the letter "a" in the password field.

If the attacker is successful, the application's database will display all user records with the name "sonakshi" and a password that contains the letter "a." The attacker can exploit this information to obtain unauthorized access to the application or other private data in the database. (Gopal, 2016)

2.3. Tools and Technologies

To carry out this SQL injection, Tools like Burp Suit, Mutillidae are used. OWASP Mutillidae is an excellent online application that has been purposefully designed to be vulnerable, making it an excellent training tool for web security specialists. Over 40 challenges and vulnerabilities are provided by the program, allowing users to improve their skills in a secure and regulated environment. Mutillidae II differs from previous comparable programs in that it does not require users to provide a "magic" statement to access vulnerabilities, making it really susceptible. The program has built-in suggestions, tutorials, and video lessons to assist users in navigating the system and completing tasks, making it simple to learn and grasp the ideas. Mutillidae II is a wonderful tool for anybody interested in learning more about online security, thanks to its user-friendly design and comprehensive capabilities. (OWASP, 2022)

Similarly, Burp Suite is a powerful platform and graphical tool for web application security testing. It is a comprehensive solution that covers every stage of the testing process, from assessing the application's attack surface to discovering and exploiting vulnerabilities. This program is intended to assist security experts in successfully assessing the security of online applications. Burp Suite's extensive feature set simplifies testing and makes it easy to use for both expert and inexperienced users. Using Burp Suite, security specialists may swiftly detect any security flaws or vulnerabilities in online applications and devise effective solutions to fix them. Overall, Burp Suite is a must-have tool for everyone working in online application security testing. (PortSwigger, 2023)

3. Attack Demonstration

3.1. Demonstration

As described earlier, the attack is carried out in in Mutillidae which is hosted in the metasploitable 2 web server. Down below the webserver is hosted in 192.168.164.138.

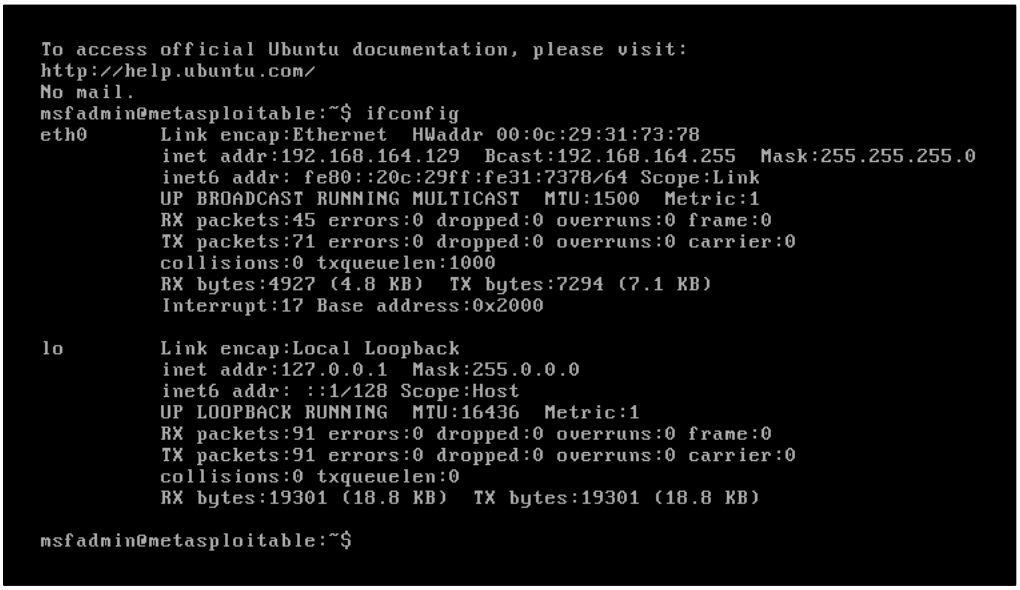

Figure 1: Metasploitable 2

Figure 2: Accessing the websites hosted on Metasploitable 2

Figure 3: Accessing Mutillidae

Figure 4: Navigating to Login/Register page

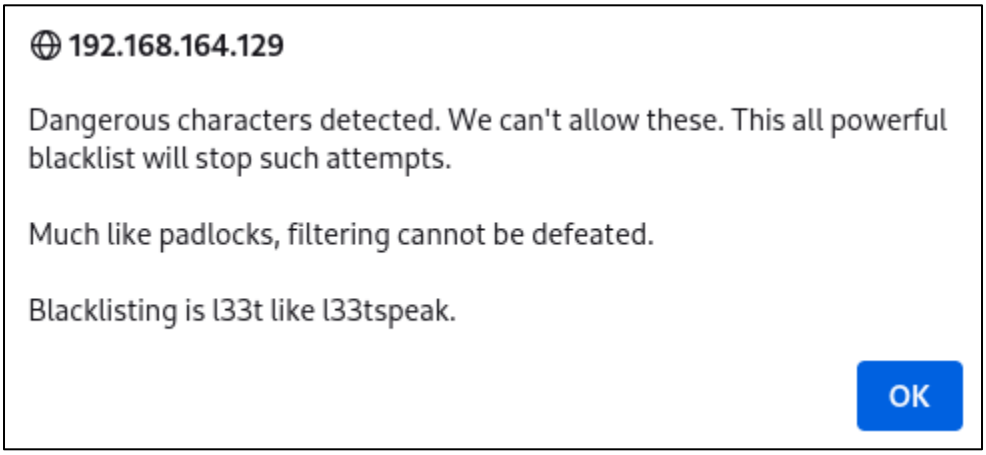

Figure 5: Error after entering special characters in text fields

Science Special characters cannot be used, Burp suit is used to bypass the client-side filtering. Down below an example of how burp suit works is shown.

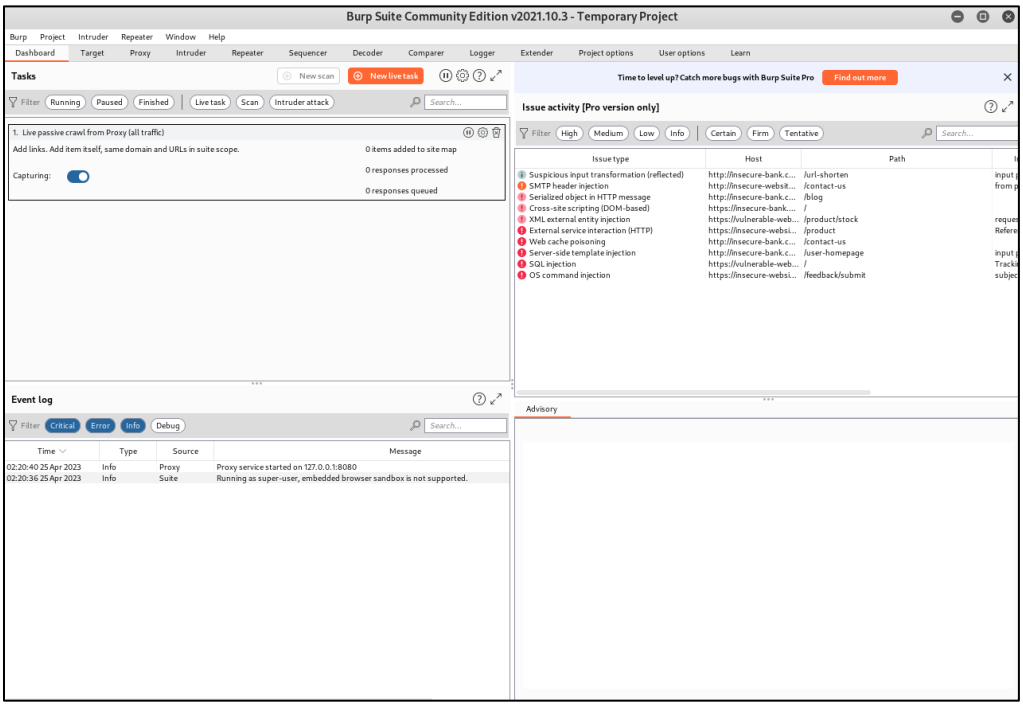

Figure 6: Opening Burp Suit

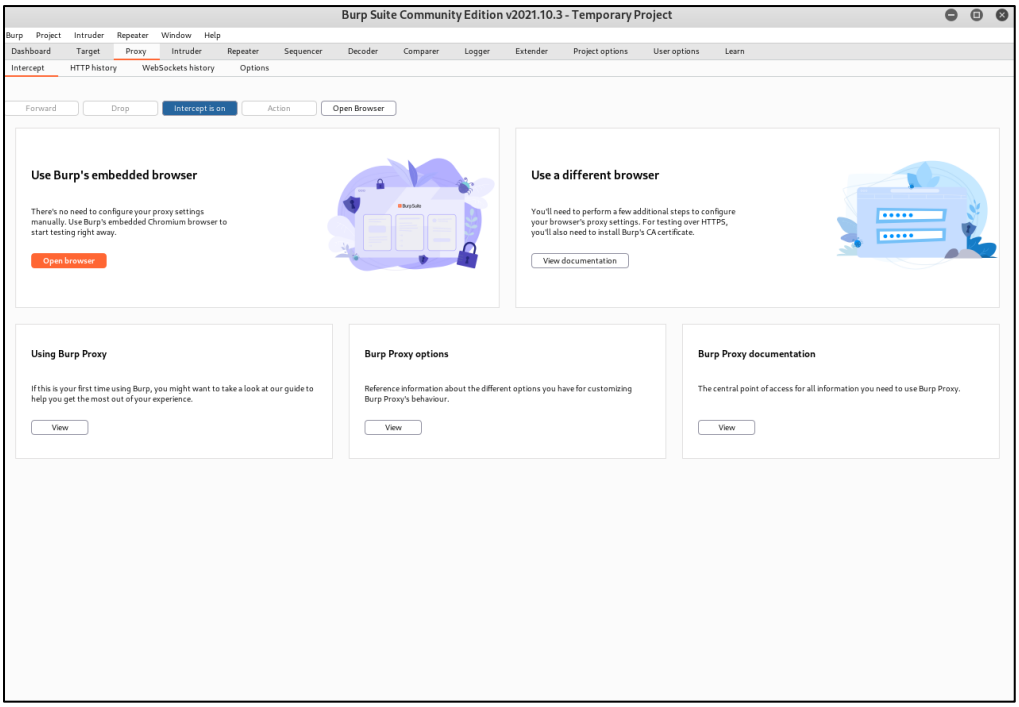

Figure 7: Navigating to Proxy tab and turning Intercept on

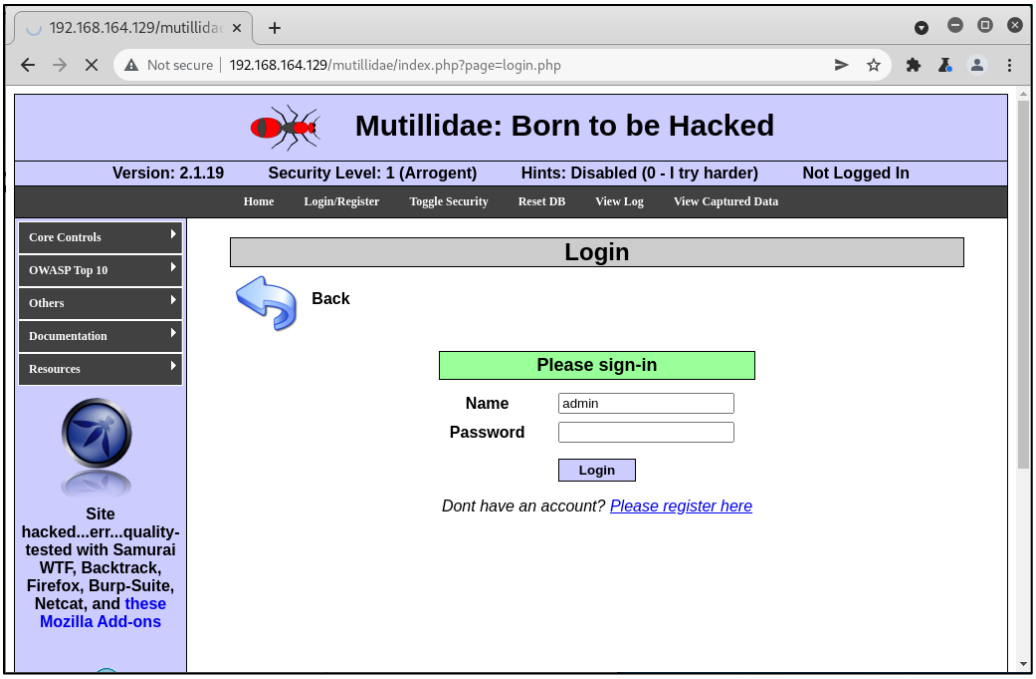

Figure 8: Navigating to Home

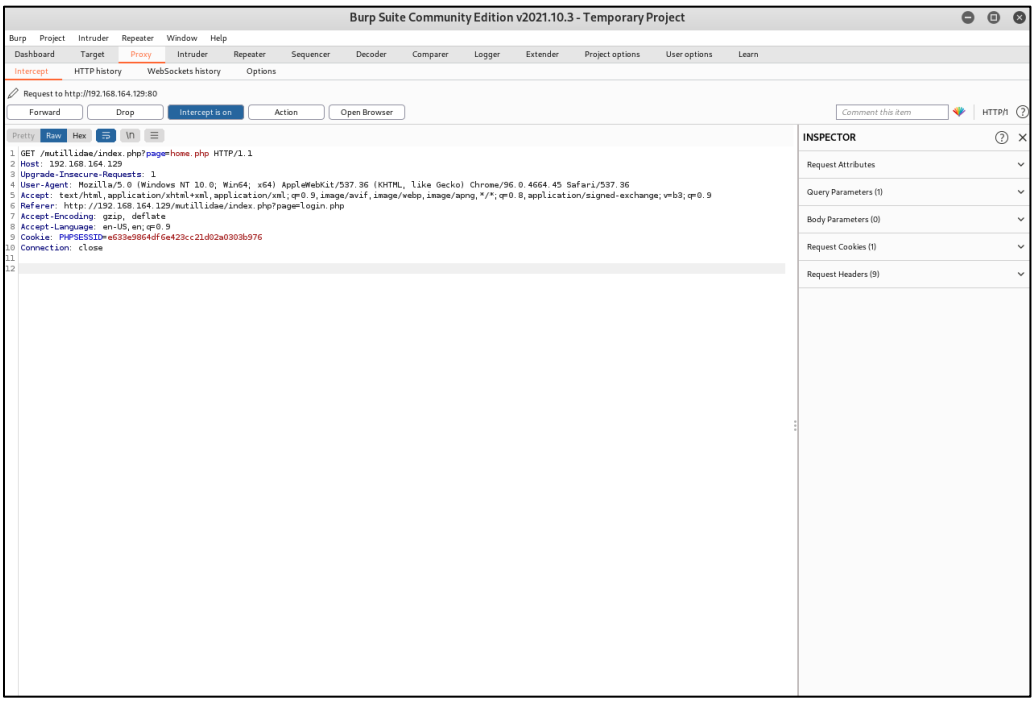

Figure 9: Burp Intercepted the request

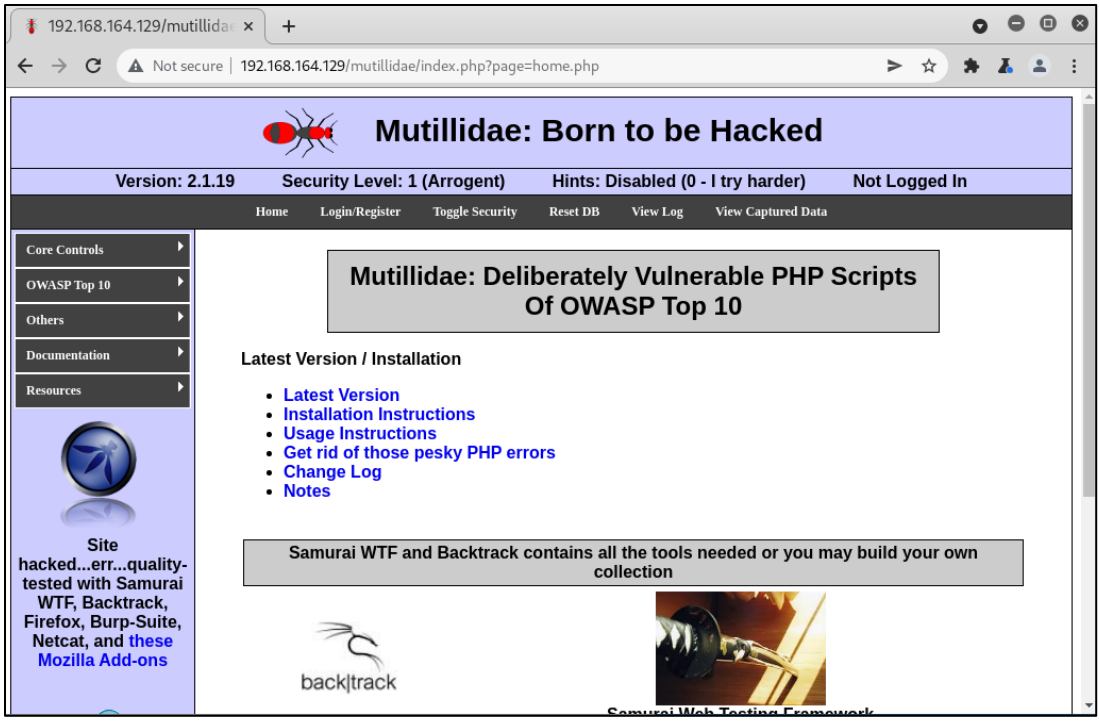

Figure 10: Home page is displayed after Forward is clicked on Burp Suit

Finally, the SQL injection can be executed. At first, admin and random password with no special characters are entered and login in pressed. As a result, Authentication error message is displayed.

Figure 11: Logging in with username admin and a random password

Next, the intercept is turned on to capture the request after client-side filtering.

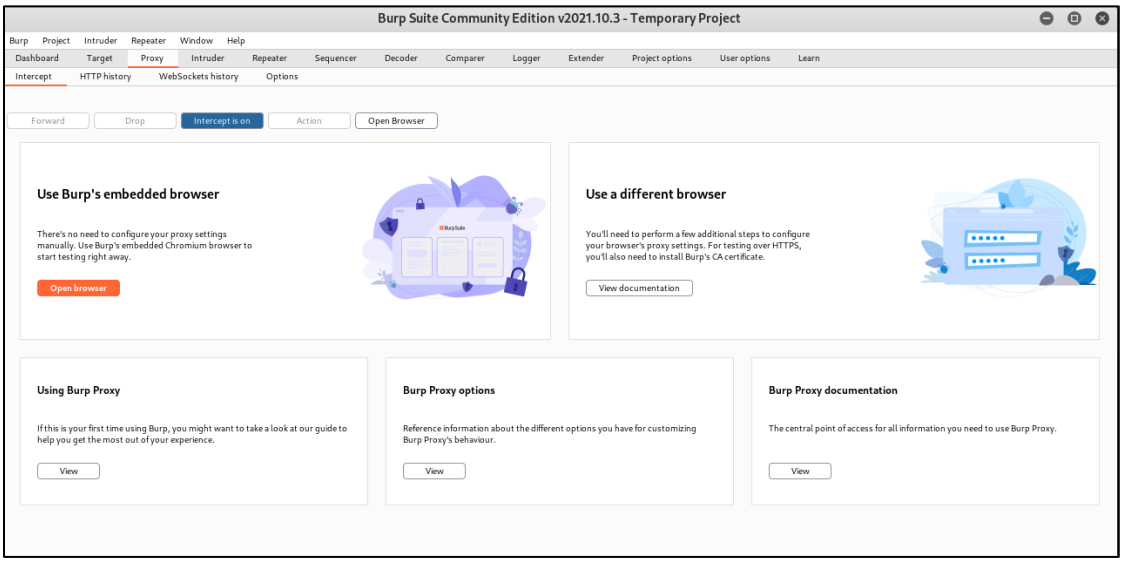

Figure 12: Intercept On in Burp suit

Now the username is passed as admin and password is passes as password. And login button is clicked

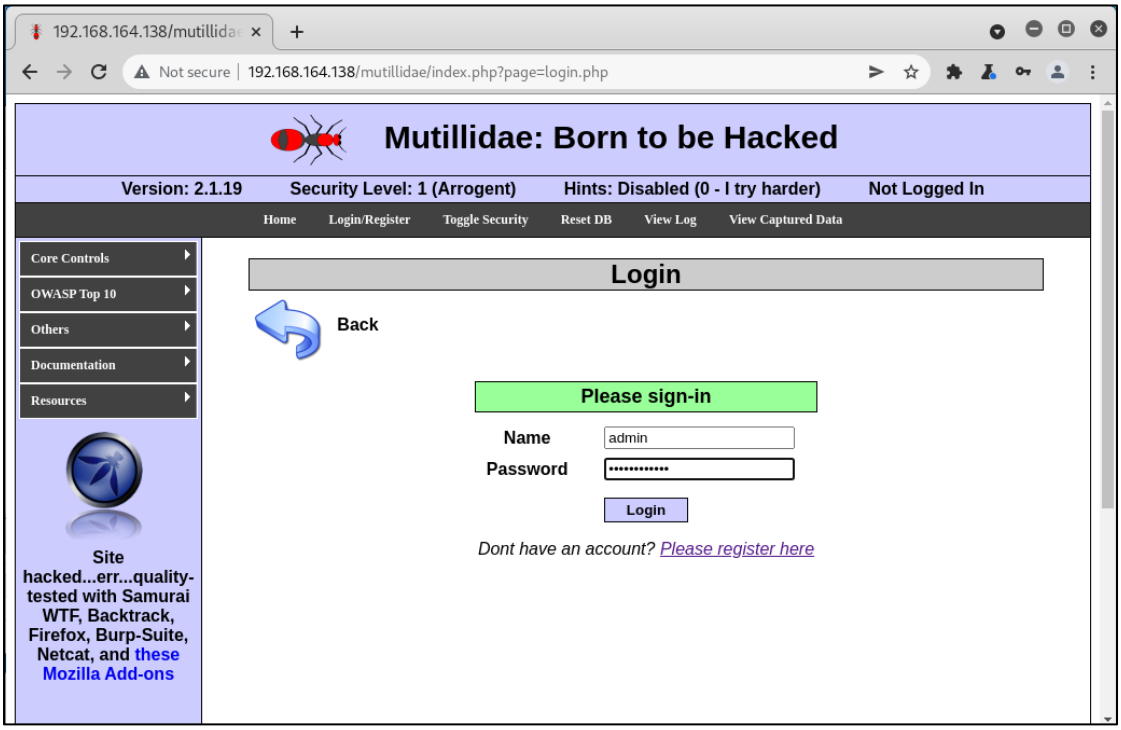

Figure 13: Assigning username and password

After login is clicked, the request is intercepted in burp suit.

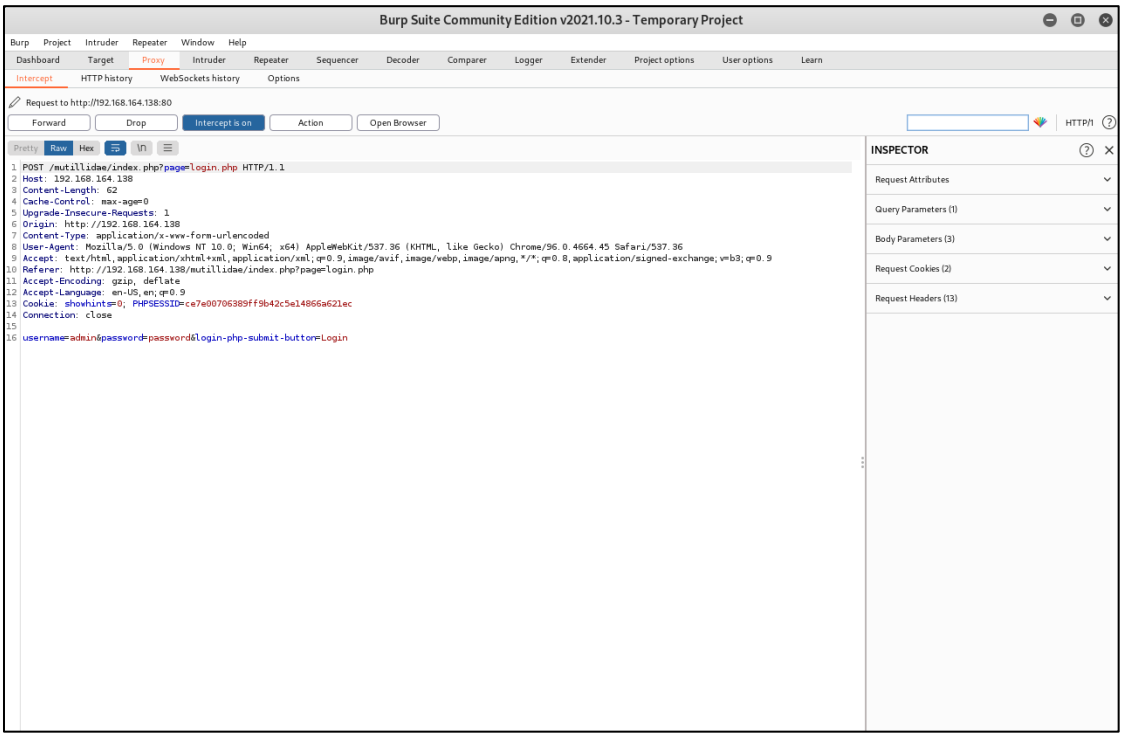

Figure 14: Burp Suit intercepting the request

SQL script is created in a tricky way using ‘or’ statement ad passing a true statement. According to the SQL script the code to inject is copied.

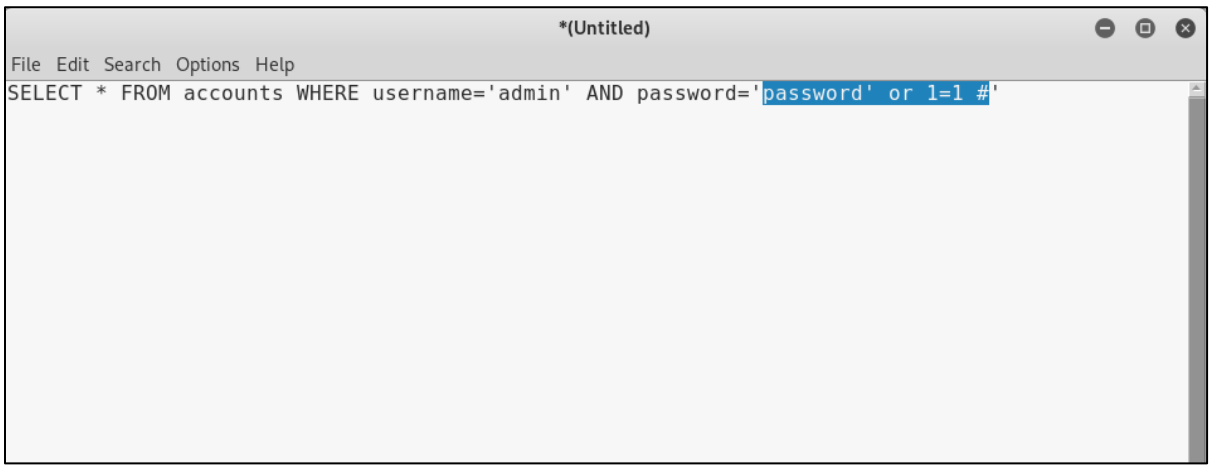

Figure 15: Creating Injection for SQL

The copied script is now pasted after password variable and forward in clicked.

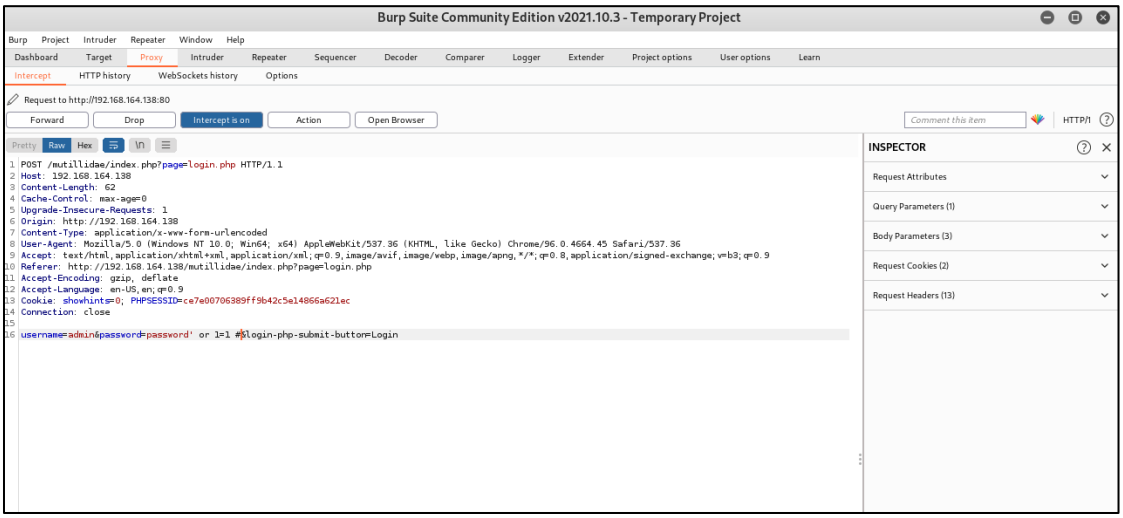

Figure 16: SQL injection

Finally, the injection is successful as the user is logged in as Admin.

Figure 17: Logged in as Admin

The SQL OR statement is used in SQL injection to send an extra value by placing a single quotation after the password, for example password'. The OR statement is then used, and a true statement is passed, making the entire sentence legitimate. As an example, the assertion 1=1 is utilized. Finally, a hash is used to terminate the sentence, commenting out the last single quotation. In total, "password' or 1=1 #" is injected into the burp suit password parameter. Finally, the forward button is pushed, and the user is signed into the online application as Admin.

3.2. Recommendation

There are several strategies available in server-side filtering to avoid SQL injections vulnerabilities. Input validation is a well-known approach. It validates the user input format, length, and type. For example, if a website requests a phone number, the input can be compared to a regular expression that allows only valid phone numbers. (dbForge Fusion for SQL Server, 2023)

Another method is to utilize parameterized queries, which take user input out of SQL code and use it as a literal value. Because the input is not recognized as executable code, attackers find it difficult to inject malicious code. Similarly, developers can employ escaping characters to prevent special characters like as quotations from being viewed as code by preceding them with a backslash.

Whitelisting is another method for limiting input to predetermined valid characters or values. However, server-side screening is insufficient to prevent SQL injection attacks. Other security procedures, such as input validation and safe coding methods, should be used by developers to protect the security of their apps. Developers may limit the risk of SQL injection attacks and keep their apps safe by using these strategies.

For illustration, passing the input through “mysql_real_escape_string()” function will go through every input passes and look for special characters and remove them. Since the special characters are removed in the server-side the vulnerabilities related to SQL injection will be prevented. (PHP, 2023)

4. Conclusion

For conclusion, SQL injection is a sort of cyber assault that targets databases and internet applications. The purpose of this assault is to exploit any vulnerabilities in the application's code and inject malicious code into the database using SQL, a database management language. Once the attacker has gained access, he or she can steal sensitive information such as login credentials, financial information, and other confidential information. The most terrifying aspect of the assault is that it may be carried out remotely without physical contact to the targeted system. Companies may suffer financial losses, data breaches, and reputational harm as a result of this sort of assault. As a result, developers and organizations must be aware of SQL injection vulnerabilities and take the appropriate precautions to avoid them.

The Electronic Transaction Act 2063 (ETA 2063) has made a good contribution to reducing SQL injection attacks in Nepal by making such attacks illegal and encouraging improved security measures for online services. Furthermore, it provides legal protection for ethical hackers who conduct security testing using SQL injection techniques, provided they have been authorized by the web application's owner. Overall, the ETA 2063 has played an essential role in increasing awareness about cybercrime and fostering a more secure online environment in Nepal. (Department of Information Technology, 2006)

4.1. Legal, Social, and Ethical Issues

4.1.1. Legal Issues

Companies may face substantial legal implications if they do not take appropriate precautions to secure their systems from SQL injection attacks. consumers may sue them, they may face fines and penalties, and they may be held accountable for any harm caused to consumers as a result of a data breach. Customers have the right to sue organizations that fail to appropriately protect their personal data, which can result in a ruined reputation for the company. Noncompliance with data privacy and protection regulations, as well as other regulatory obligations, may potentially result in legal action and further penalties. (imperva, 2022)

4.1.2. Social Issues

SQL injection assaults, in addition to the legal ramifications, can have a substantial societal impact. When personal information is exposed, it can diminish customer faith in a firm, leading to bad press and reputational harm. Customers that lose faith may opt to take their business elsewhere, which can have financial ramifications for the organization. Individuals and corporations may potentially incur financial losses as a result of data breaches caused by SQL injection attacks, which can have a cascading impact and cause other issues. Companies must take these societal ramifications seriously and emphasize the security of their consumers' data. (Dizdar, 2022)

4.1.3. Ethical Issues

From an ethical standpoint, using SQL injection vulnerabilities to obtain access to and steal personal data is a violation of a person's right to privacy. Companies that acquire client data have an ethical obligation to protect that data from unwanted access, which SQL injection attacks take advantage of. As a result, businesses must accept responsibility for safeguarding their customers' personal information and preventing such assaults from occurring. It is not just a legal and regulatory requirement, but also an ethical commitment to customers. (EC-Council, 2023)

5. References

dbForge Fusion for SQL Server, 2023. *Server-Side and Client-Side Sorting and Filtering.* [Online]
Available at: https://docs.devart.com/fusion-for-sql-server/building-queries-with-query-builder/query-builder-overview.html
[Accessed 20 April 2023].

Department of Information Technology, 2006. *Acts and Laws.* [Online]
Available at: https://doit.gov.np/en/list/resource/2?parent=12
[Accessed 28 April 2023].

Dizdar, A., 2022. *SQL Injection Attack: Real Life Attacks and Code Examples.* [Online]
Available at: https://brightsec.com/blog/sql-injection-attack/
[Accessed 28 April 2023].

EC-Council, 2023. *The Ultimate Guide to SQL Injection | Certified Ethical Hacker | EC-Council.* [Online]
Available at: https://www.eccouncil.org/cybersecurity/what-is-sql-injection-attack/
[Accessed 28 April 2023].

Gopal, G., 2016. *(PDF) CASE STUDY OF SQL INJECTION ATTACKS.* [Online]
Available at: https://www.researchgate.net/publication/309404360_CASE_STUDY_OF_SQL_INJECTION_ATTACKS
[Accessed 25 April 2023].

imperva, 2022. *What is SQL Injection | SQLI Attack Example & Prevention Methods | Imperva.* [Online]
Available at: https://www.imperva.com/learn/application-security/sql-injection-sqli/#:~:text=The%20impact%20SQL%20injection%20can,highly%20detrimental%20to%20a%20business.
[Accessed 28 April 2023].

OWASP, 2022. *OWASP Mutillidae II | OWASP Foundation.* [Online]
Available at: https://owasp.org/www-project-mutillidae-ii/
[Accessed 25 April 2023].

PHP, 2023. *PHP: mysql_real_escape_string - Manual.* [Online] Available at: https://www.php.net/manual/en/function.mysql-real-escape-string.php [Accessed 20 April 2023].

PortSwigger, 2023. *Burp Suite documentation - PortSwigger.* [Online] Available at: https://portswigger.net/burp/documentation [Accessed 25 April 2023].

PortSwigger, 2023. *https://www.php.net/manual/en/function.mysql-real-escape-string.php.* [Online] Available at: https://portswigger.net/web-security/sql-injection [Accessed 19 April 2023].