**PRACTICAL 9**

**Aim:** Prepare document on SQL Injection and XSS with example. Attach necessary screenshots.

**Hardware Requirement:**

Computer/Laptop.

**Software Requirement:**

Ubuntu Linux, Burp Suite, MS Word.

1. **Cross-Site Scripting (XSS):**

Cross-Site Scripting (XSS) assaults are injection attacks in which malicious scripts are inserted into otherwise trustworthy and innocent websites. XSS attacks occur when an attacker utilizes a web application to transmit malicious code to a separate end user, usually in the form of a browser side script. The flaws that allow these attacks to succeed are common and can be found whenever a web application accepts user input in its output without verifying or encoding it.

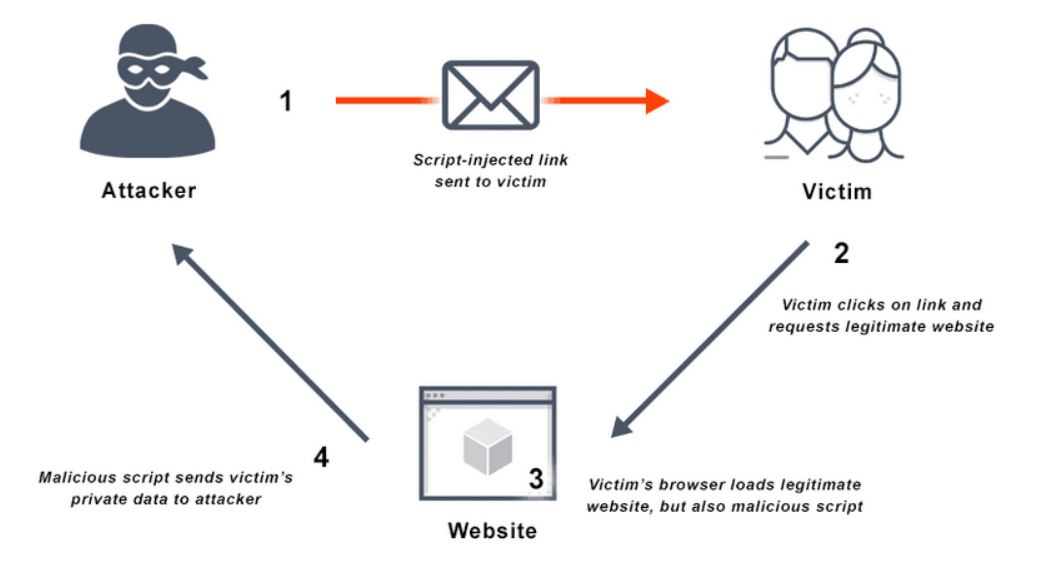
XSS can be used by an attacker to send a malicious script to an unwitting user. The browser of the end user has no means of knowing that the script should not be trusted and will run it nonetheless. The malicious script can access any cookies, session tokens, or other sensitive information stored by the browser and used with that site since it believes the script came from a trusted source. These programs can even rewrite the HTML page's content. For additional information on the many types of XSS bugs.

Cross-Site Scripting (XSS) attacks occur when:

1.Data is entered into a Web application from an untrusted source, most commonly a web request.

2.The information is contained in dynamic material that is provided to a web user without being checked for maliciousness.

Malicious content is frequently transmitted to the web browser in the form of a JavaScript section, but it can also be HTML, Flash, or any other type of code that the browser can run. The number of XSS attacks is nearly endless, but they typically involve sending private data to the attacker, such as cookies or other session information, redirecting the victim to attacker-controlled web content, or performing other malicious operations on the user's machine while posing as the vulnerable site.



**TYPE OF XSS ATTACKS:**

Stored XSS Attacks

Stored attacks are ones in which the injected script is kept on the target servers indefinitely, such as in a database, a message board, a visitor log, a comment box, and so on. When the victim requests information from the server, the malicious script is downloaded. Persistent or Type-I XSS are terms used to describe stored XSS.

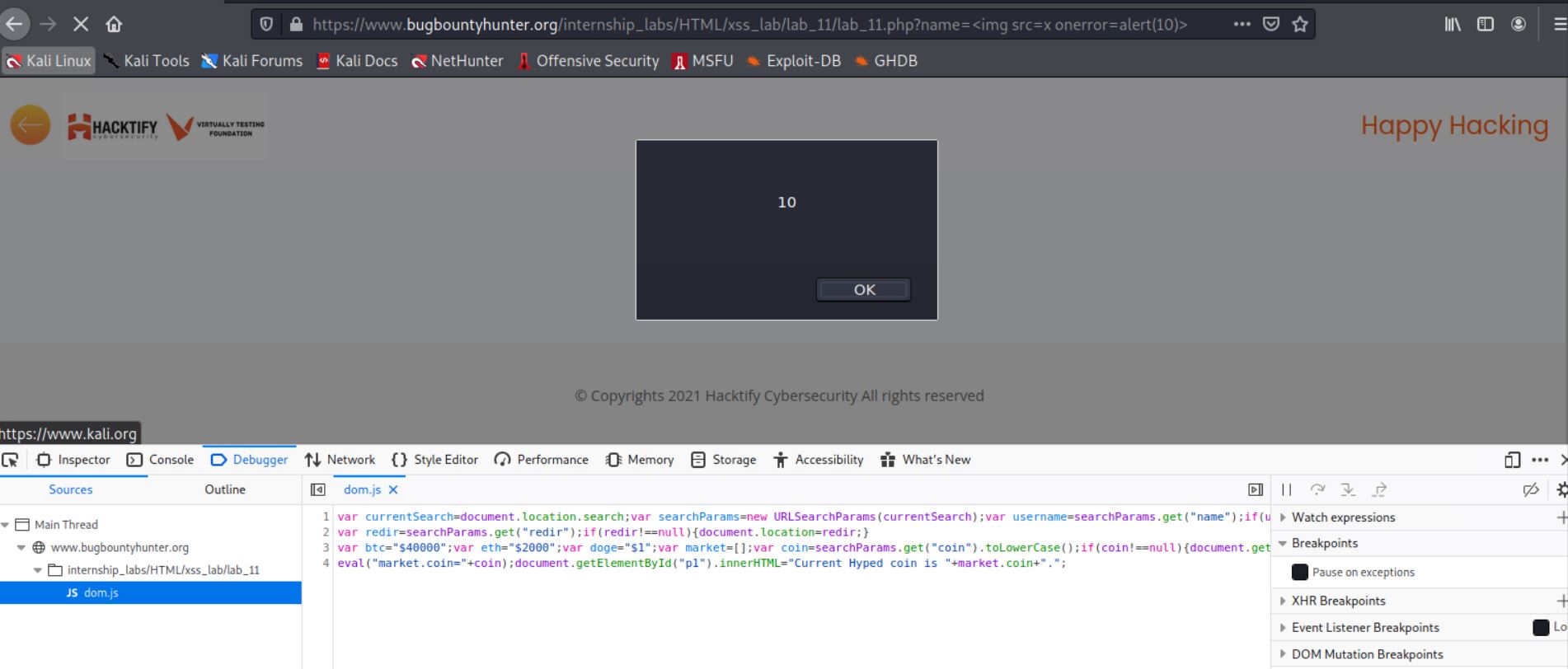
Blind Cross-site Scripting

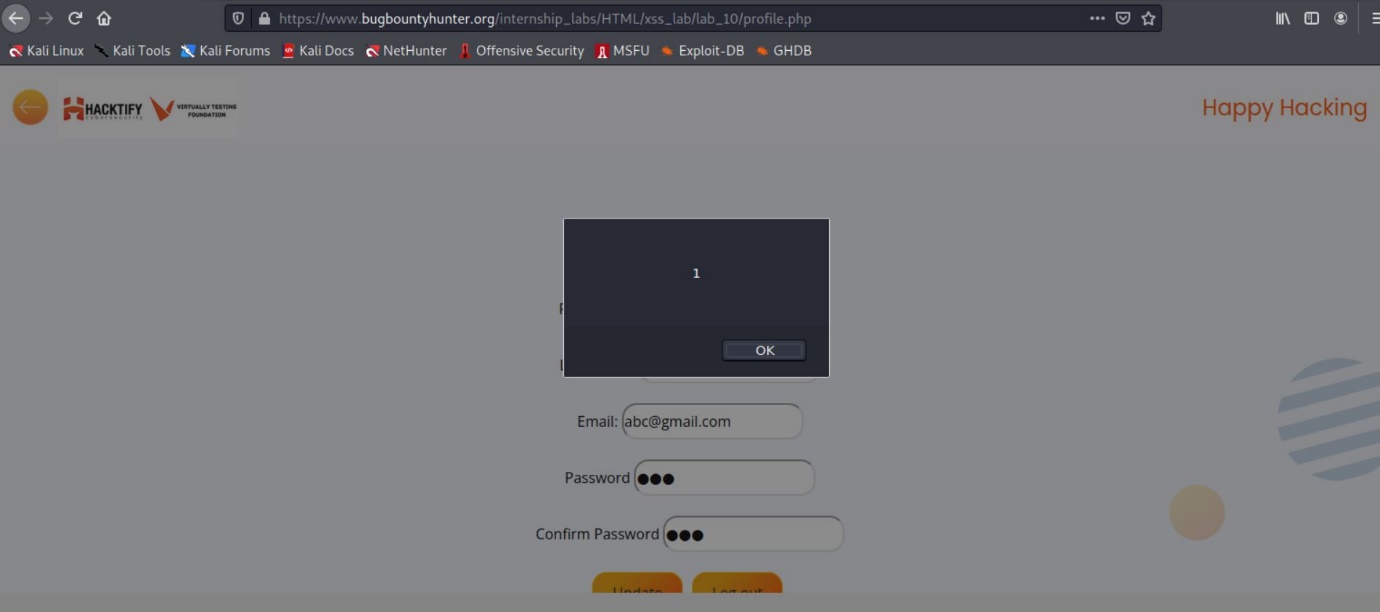
Persistent XSS is a type of blind cross-site scripting. When the attacker's payload is saved on the server and reflected back to the victim via the backend application, this is what happens. For example, with feedback forms, an attacker can submit a malicious payload using the form, and the attacker's payload will be executed whenever the backend user/admin of the application opens the attacker's submitted form via the backend application. Blind Cross-site Scripting is difficult to detect in a real-world setting, but XSS Hunter is one of the greatest tools for the job.

Reflected XSS Attacks

The injected script is mirrored off the web server, such as in an error message, a search result, or any other response that includes some or all of the input supplied to the server as part of the request. Reflected attacks are provided to victims through a different channel, such as an e-mail message or a different website. When a user is duped into clicking on a malicious link, filling out a specially constructed form, or simply navigating to a rogue site, the injected code goes to the susceptible web site, which replicates the user's actions.

**Practical Implementation:**





1. **SQL INJECTION(SQLi):**

SQL injection, often known as SQLI, is a typical attack vector in which malicious SQL code is used to manipulate backend databases and get access to data that was not intended to be displayed. This data could encompass everything from sensitive company data to user lists to private consumer information.

SQL injection can have a huge impact on a company's bottom line. A successful attack could result in the attacker reading illegal user lists, deleting entire tables, and, in some situations, gaining administrative rights to a database, all of which are extremely damaging to a corporation.

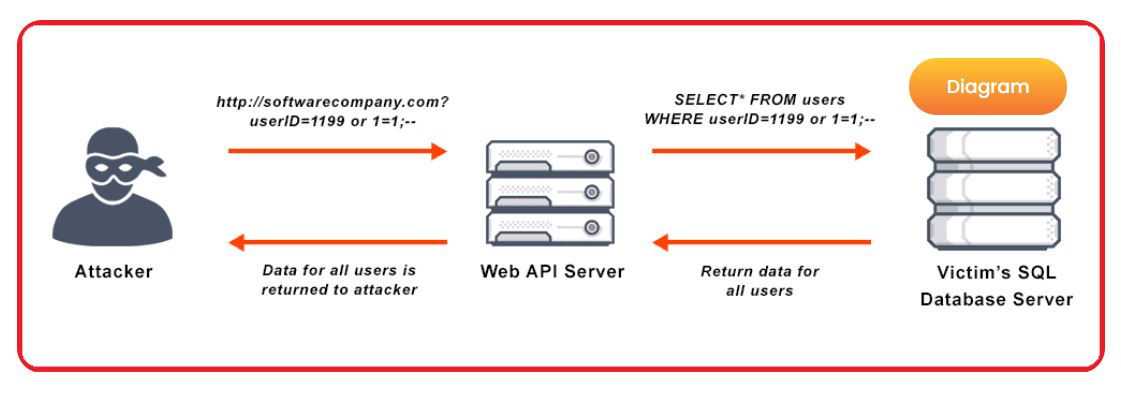
When estimating the cost of a SQLi, keep in mind the loss of customer trust if personal information such as phone numbers, addresses, and credit card data is stolen.

**What causes an SQL Injection?**

An attacker can use a SQL injection vulnerability to inject malicious data into a SQL statement. To completely comprehend the problem, we must first comprehend how SQL queries are handled by server-side scripting languages. For instance, a web application functionality generates a string using the SQL statement:

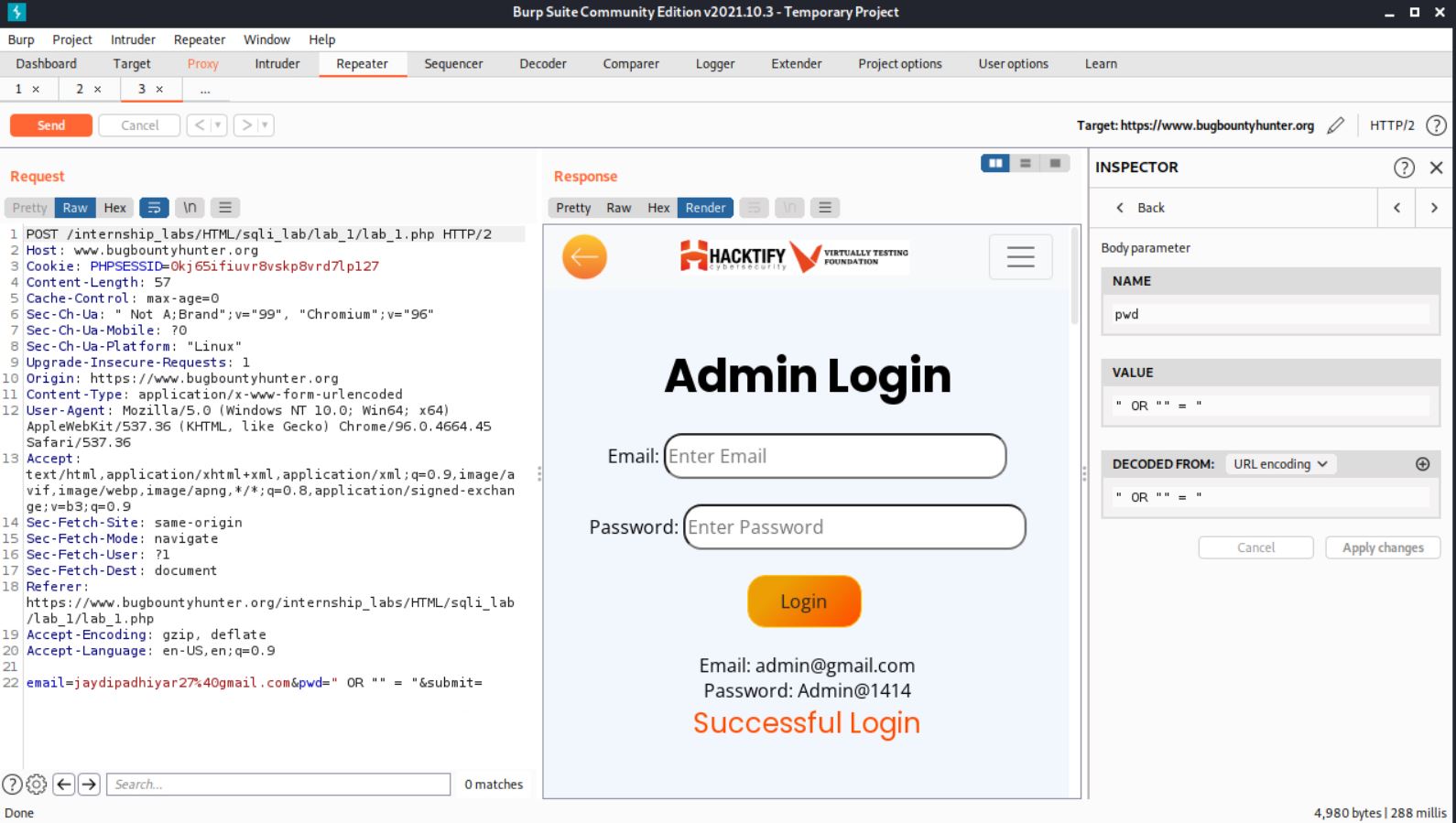
"SELECT \* FROM users WHERE username = 'Charithra' AND password = '1234' "; $query = " SELECT \* FROM users WHERE username = 'Charithra' AND password = '1234' "; $query = " SELECT \* FROM users W

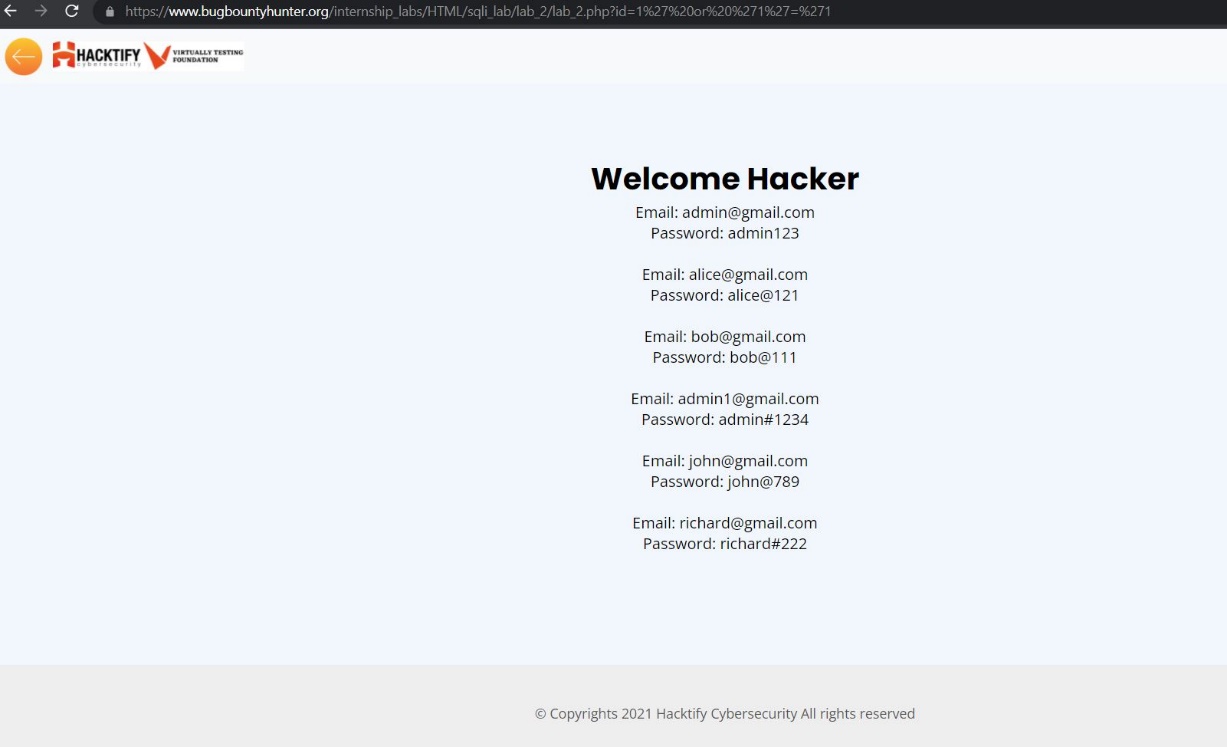
The above database query is a straightforward one that returns all fields in the users table that satisfy the supplied "WHERE" condition. In this case, the query must only return all of the table's data where the user has the name Charithra and the password '1234.'



**Practical Implementation:**

Payload: - “ OR “” = “





**CONCLUSION:**

Using these examples, we learned about SQL Injection and XSS in this practical.