T.Y. Instrumentation and Control Engineering Vishwakarma Institute of Technology Pune, India

Design & Development of an intentionally vulnerable web application for enactment of SQL injection & Cross Site Scripting vulnerabilities: HackerLearn

By

Sakshi Patil (11910968), Shaunak Deshpande(11911180), Shrawani Shinde (11911031), Shruti Yadnik(11911408).

***Abstract — In today's world, every human being relies on the internet for their basic to complex needs. Because the internet provides users with a great amount of information, its accessibility is essential. Major objectives of security are availability, integrity and confidentiality. Cross-site scripting (XSS) and SQL Injection Attack (SQLIA) are two common and serious security flaws in web applications and databases.According to our research, there is not a specific platform for cybersecurity students to practise these critical attacks. The proposed technology allows students to rehearse these attacks and gain a better understanding of the situation.***

***Keywords*** *— Security, Cyber Attacks, SQL Injection, XSS Attack*

1. INTRODUCTION

Cybersecurity is important because it protects all categories of data from theft and damage. This includes sensitive data, personally identifiable information (PII), protected health information (PHI), personal information, intellectual property, data, and governmental and industry information systems. Without a cybersecurity program, your organization cannot defend itself against data breach campaigns, which makes it an irresistible target for cybercriminals.

Both inherent risk and residual risk are increasing, driven by global connectivity and usage of cloud services, like Amazon Web Services, to store sensitive data and personal information. Widespread poor configuration of cloud services paired with increasingly sophisticated cyber criminals means the risk that your organization suffers from a successful cyber attack or data breach is on the rise.Business leaders can no longer solely rely on out-of-the-box cybersecurity solutions like antivirus software and

firewalls, cybercriminals are getting smarter and their tactics are becoming more resilient to conventional cyber defenses. It's important to cover all the fields of cybersecurity to stay well-protected.

Many cyber attacks are opportunistic, with hackers spotting vulnerabilities in a computer system’s defenses and exploiting them. This may involve finding flaws in the code of a website that allows them to insert their own code and then bypass security or authentication processes. It could also mean they install ‘malware’ – software which is specifically designed to damage a system – via a vulnerable third party site.Although terminology such as ‘cyber attacks’ and ‘hackers’ may conjure up images of sophisticated teams of computer experts with high-tech equipment, poring over lines of code, the reality is often quite different. Cyber attacks are much more likely to occur through mundane errors like a user choosing an easy-to-guess password or not changing the default password on something like a router.Cyber attacks are usually either criminally or politically motivated, although some hackers enjoy bringing down computer systems a thrill or sense of achievement.

Politically motivated cyber attacks may occur for propaganda reasons, to harm the image of a particular state or government in the minds of the public. It might also have more pernicious intent, such as to leak sensitive intelligence, private communications or embarrassing data.

Cyber attacks could potentially go even further, for example, government-backed hackers could theoretically create software to corrupt and destroy a weapons program, or other crucial infrastructure.

In the present world of technology and innovation, the best way to build cyber security experience is to practice. The main concern of our project is to develop a website where users can practice different cyber attacks like SQL Injection, XSS.

1. **LITERATURE REVIEW**

SQL injection is a technique that exploits a security vulnerability occurring in the database layer of an application. The attack takes advantage of poor input validation in code and website administration. It allows attackers to obtain unauthorized access to the back-and database to change the intended application generated SQL queries. Researchers have proposed various solutions to address SQL injection problems. However, many of them have limitations and often cannot address all kind of injection problems. What’s more, new types of SQL injection attacks have arisen over the years. To better counter these attacks, identifying and understanding existing techniques are very important. In this research we present all SQL injection attack types and also different techniques and tools which can detect or prevent these attacks.

In [1] The proposed technique was implemented using PHP Scripting Language and Apache XAMPP Server. The PHP was selected as a scripting Language, because it is the most widely used server-side scripting language in building database-driven web-based application while the Apache XAMPP Server was chosen due to its cross-platform compatibility, it supports any operating system, and it also supports both PHP scripting language and SQL.

Cross Site Scripting is one of the first vulnerabilities to be identified. Malicious code is injected from the source to the user browser. This can result in stealing of cookies with Personal Information such as Credit Card details, login credentials etc [2].

SQL injection is one of the most lethal attacks that can be launched against any web application involving databases. 64% web applications worldwide are vulnerable to SQL injection attack improper input mechanisms.[3]

SQL injection and XSS can be prevented using input

sanitization. Two sanitization methods include blacklisting

and whitelisting. Blacklisting, as name suggests, listing out

the inputs which are flagged as malicious in nature. Usage

of ‘<script>’ tags or use of operators such as “=” etc is not

allowed as they could play with the logic of codes and

queries running in the backend. Whitelisting is the polar

opposite of Blacklisting. It refers to the list of

symbols/characters that can be allowed.[5]

**CYBER ATTACKS :**

Two attacks will be discussed in this paper .

## XSS(Cross Site Scripting)

XSS, an acronym for Cross Site Scripting, is an injection attack. The injected code is usually written in JavaScript which is the browser side script. This attack occurs when input is not properly validated or encoded. It allows the hackers to force through the SOP (Same Origin Policy).

In many websites, cookies are used to store your login credentials temporarily so that you may not have to log in again and again. But this is highly risky. It would be advised to keep a user logged for a short period of time when inactive rather than remembering login credentials in the form of cookies as these cookies are easily accessible from the user side with the browser using Javascript or any other browser in-build tools, utilities etc[2]. Thus your Sensitive Data confidentiality might be compromised. It can be used to capture login credentials, get access to unauthorized data, masquerading, performing unauthorized tasks etc.

## SQL injection

SQL injection is also an injection attack. As the name suggests, SQL code is injected into websites . Database vulnerabilities are exploited in this attack.

For most websites, data displayed on the web page arrives from SQL database servers. Login/ Registration forms, comments etc, everything is loaded into the backend from the databases. When we try to login to a website, an SQL query is run in the backend to verify your credentials. In case of registering an account, additional rows are added into the database. If sanitization is not practiced, input fields of the page could be filled with SQL code which could logically end up commenting out a certain section of the SQL query that is run in the backend which could result in granting login permission without knowing the actual credentials. This one of several examples of how SQL injection is implemented. It can be used to retrieve hidden data, subverting application logic, examining databases, UNION attacks(retrieve data from various databases) etc

1. **METHODOLOGY/EXPERIMENTAL**

The project starts with a landing page which gives the user an option to register onto the website (for new users) or login to their account (for repeat users). The authentication system has been implemented using flask, flask-login and bcrypt. The hash of the password entered by the user gets stored during registration, and during login, the hash of the entered password is compared with the stored password. After login, the user is greeted by an ecommerce page, where the user can search for products. Along with that, there is also an option to open up the blog page of the ecommerce website, and leave a review for the products bought from the website. All this is setup so that this looks like an actual ecommerce business to the user. But the primary purpose of this web app is for practicing cyber security attacks in a secured environment. In the product search page, the user sees an input box, which can be used for practicing SQL injection attacks. The user can give malicious SQL queries into the search engine and try to steal user data. The user can also practice blind SQLI attacks here. Along with this, the blog and the product review section can be hit by Cross Site Scripting (XSS) attacks. The backend is written in Flask, a Python based microframework, which makes it lightweight, but also more susceptible to cyber threats, making it the perfect choice for our backend. The user can also perform Cookie stealing using the Cross site scripting vulnerability in the webapp. The SQL injection attack allows the user to get the passwords of the users, making it a good exercise in cracking Bcrypt hashes. Lastly there is a logout button that the user can click on to log out of the session. The session management is done by the flask\_login module in python.

1. **RESULT AND DISCUSSION**

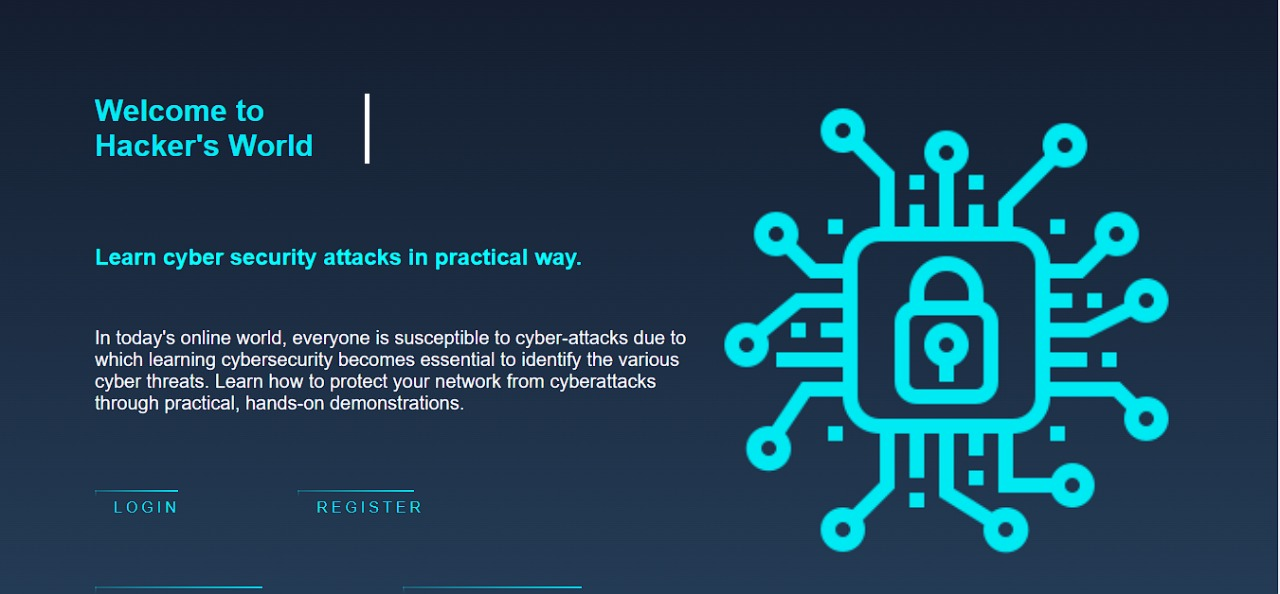


Fig.3. Home page

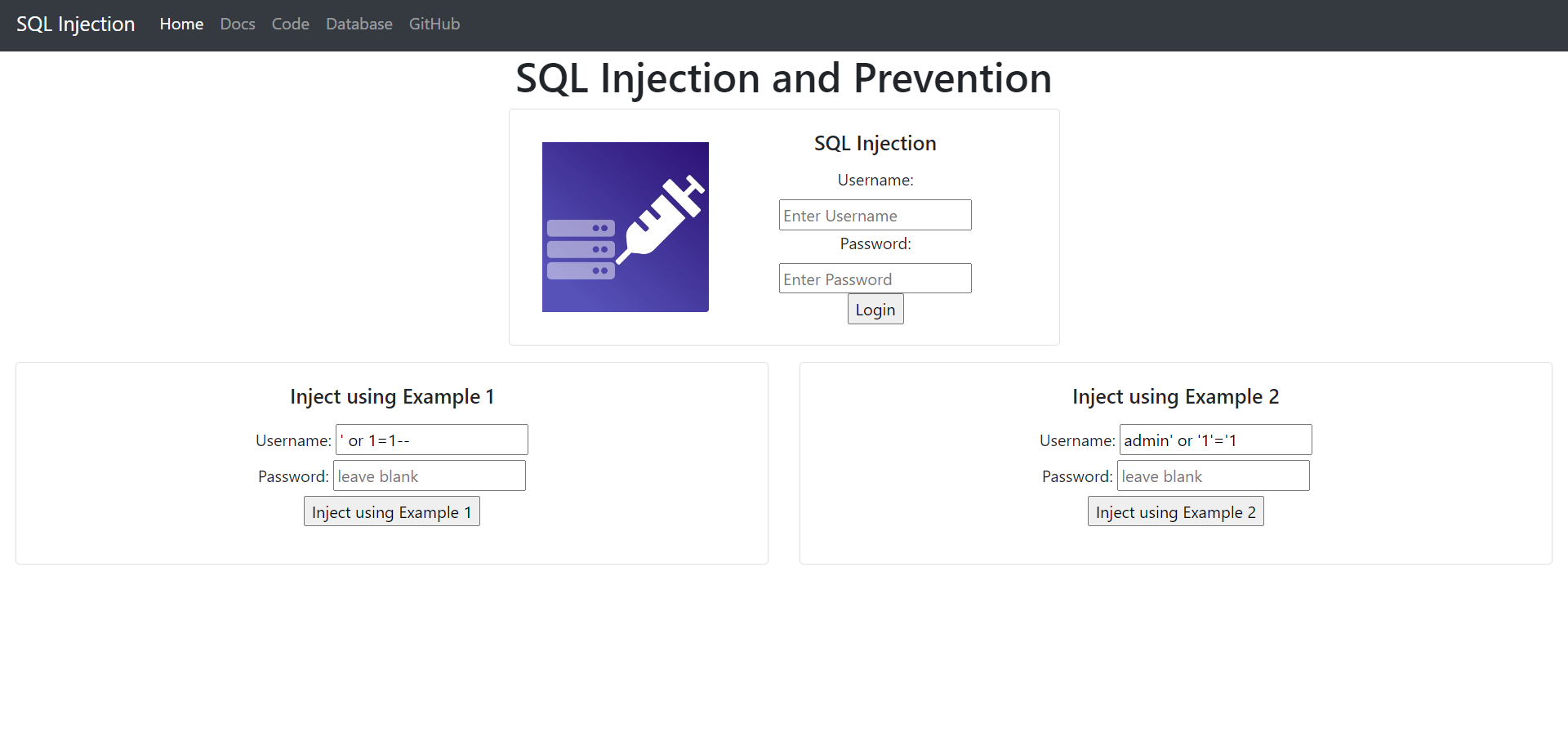
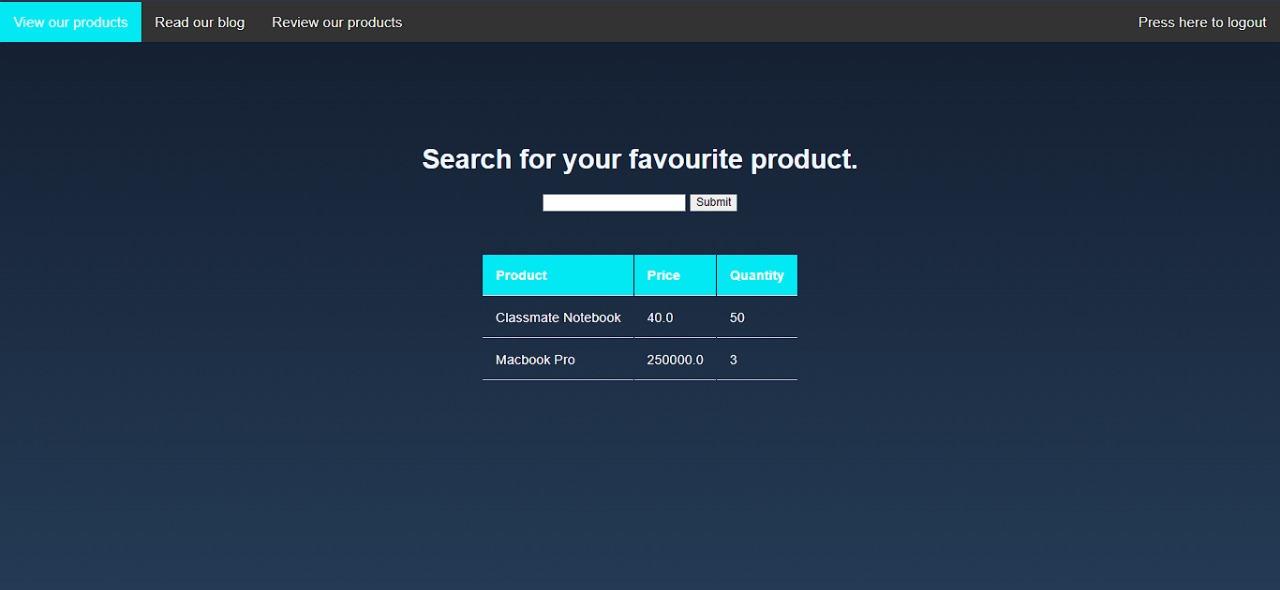


Fig4. Webpage for simulating SQL injection attack on database



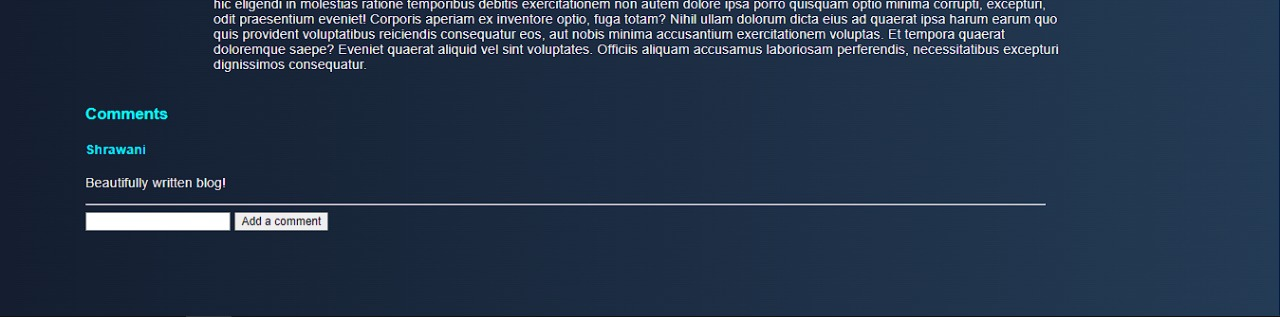


Fig.6.Webpage for simulating stored XSS attack on Blog comments

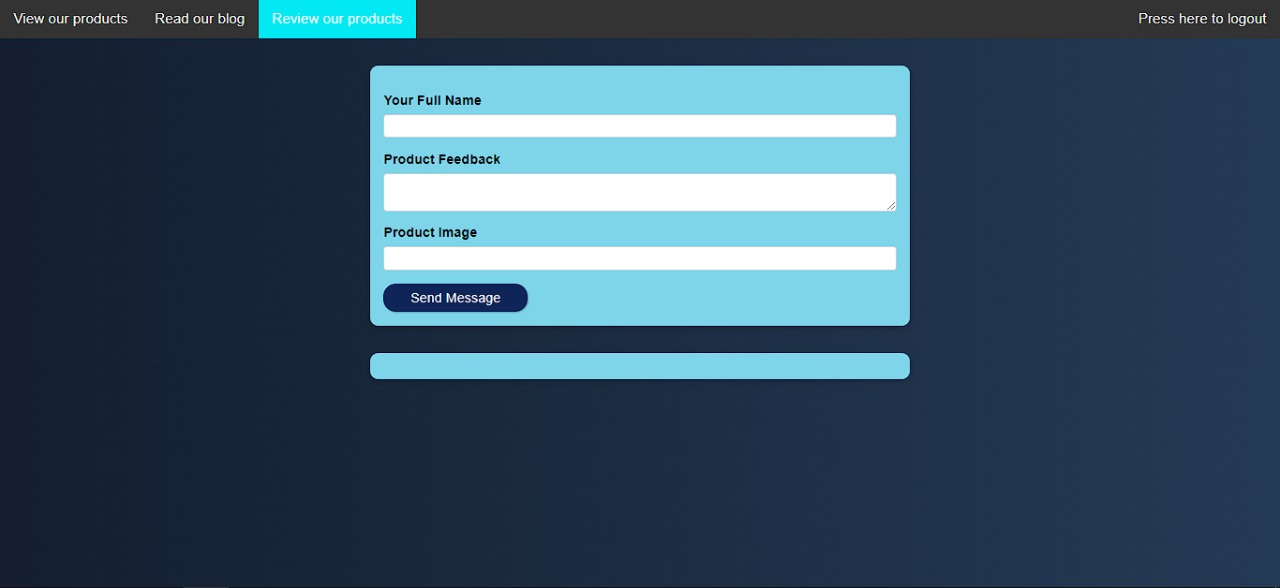


Fig.7.Webpage for simulating stored XSS attack on Product Reviews

Fig.5. Webpage for simulating SQL injection attack on search for a product

1. **FUTURE SCOPE**

The proposed application is capable of simulating SQL injection and XSS attack successfully. In future, we intend to include simulation functionalities of more cyber attacks such as CSRF, Man-in-the-middle, Zero-day attack, and so on, so that the web application can serve as a full-fledged cyber attack simulation platform.

1. **CONCLUSION**

SQL injection and cross-site scripting attacks are two of the most dangerous threats to web applications. The proposed web application serves as a simulation platform to practise attacks in a secured environment. It will assist students and enthusiasts interested in cyber security to obtain a better knowledge of how such cyber attacks occur.

1. **ACKNOWLEDGMENT**

We would like to express our gratitude to Prof. Anil Kadu, our project guide, for thoroughly educating us on all aspects of the project. We would also like to express our gratitude to our college, Vishwakarma Institute of Technology, for giving us the opportunity to work on and present a project at this level.

1. **REFERENCES**
2. Prakhar Tripathi , Rahul Thingla; ‘Cross site scripting (XSS) and SQL-injection attack detection in
3. web application’, International Conference on Sustainable Computing in Science, Technology & Management (SUSCOM-2019).
4. ‘XSS- What are Cross Site Scripting Attacks’- An article by Maximilian Schwarzmüller.

[3] Adit Bhosle, “Combination Attack: XSS+SQL Injection Attack Demonstration”, International Research Journal of Engineering and Technology (IRJET).

[4] Robinson, Memen Akbar, SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application Firewall, International Journal of Informatics Visualization. VOL 2 (2018) NO 4.

[5] Muhammad Saidu Aliero, Imran GhaniAn algorithm for detecting SQL injection vulnerability using black-box testing,, Journal of Ambient Intelligence and Humanized Computing , 7th Febuary 2019.

[6] Robinson, Memen Akbar, SQL Injection and Cross Site Scripting Prevention Using OWASP Web Application

Firewall, International Journal of Informatics

Visualization. VOL 2 (2018) NO 4.

[7] Ouissem Ben Fredj, Omar Cheikhrouhou, An OWASP Top Ten Driven Survey on Web Application Protection Methods, Risks and Security of Internet and Systems, 15th international Conference, CRiSIS 2020, Paris, France, November 4-6, 202