**PROJECT REPORT**

**On**

**Vulnerability Scanner**

**Submitted by**

Harshpreet Singh Sandhu (Enroll No. R134218065)

Neelesh Kumar (Enroll. No. R134218102)

Naman Sharma(Enroll. No. R134218099)

Aarohi Mangal(Enroll. No. R134218001)

**Under the guidance of**

Mr. Keshav Kaushik

Department of Systemics



**SCHOOL OF COMPUTER SCIENCE**

UNIVERSITY OF PETROLEUM & ENERGY STUDIES

Bidholi Campus, Energy Acres, Dehradun – 248007.

**AUG-Dec-2021**

**CANDIDATE’S DECLARATION**

We hereby certify that the project work entitled **“Vulnerability Scanner”**in partial fulfillment of the requirements for the award ofthe Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING with specialization in **CYBER SECURITY AND FORENSICS** and submitted to the School of Computer Science, Department of Systemics, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August**-**2021** to **December-2021** under the supervision of **Mr. Keshav Kaushik , Dept. of Systemics.**

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

**Aarohi Mangal** R134218001

|  |  |
| --- | --- |
| This is to certify that the above statement made by the | candidate is correct to the best of |
| my knowledge. |  |
| Date: \_\_\_\_\_\_\_\_\_\_\_\_\_2020 | **Mr. Keshav Kaushik** |
|  | Project Guide |
| **Dr. Neelu Jyoti Ahuja** |  |
| HOD – Systemics |  |
| School of Computer Science |  |
| University of Petroleum & Energy Studies |  |
| Dehradun – 248 001 (Uttarakhand) |  |
|  |  |

**Harshpreet Singh Sandhu** R134218065

**Naman Sharma** R134218099

**Neelesh Kumar Gupta**

R134218102

**ACKNOWLEDGEMENT**

We wish to express our deep gratitude to our guide **Mr. Keshav Kaushik**, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, **Dr. Vidyanand Mishra**, for her great support in doing our project in **Dept. of Systemics** at **SoCS**.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally, we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us

**Name:** Aarohi Mangal Neelesh Kumar Harshpreet Sandhu Naman Sharma

**Roll No:** R134218001 R134218102 R134218065 R134218099

 **School of Computer Science**

**University of Petroleum & Energy Studies, Dehradun**

**PROJECT TITLE:**  **Vulnerability Scanner**

**ABSTRACT**

As of late, a ton of web applications have been delivered on the planet. Simultaneously, cyber-attacks against web application weaknesses have additionally expanded. In such a circumstance, it is important to make web applications safer. Anyway, really taking a look at all web weaknesses by hand is truly challenging and tedious. Therefore, a web vulnerability scanner is developed to detect some vulnerabilities like SQL Injection and Cross-site Scripting which are more frequently found in web applications.

**INTRODUCTION**

Complete web weaknesses scanner is utilized to observe the sites bug and after that shows the sorts of bug on that sites. This task is created in Python. As we probably are aware a consistently expanding number of high-profile information breaks have tormented associations over the previous decade. An extraordinary number of these come about through purported 'infusion assaults'; the main association in the field of web application security states; How information input is taken care of by Web applications is seemingly the main part of safety"

Two elements increment the stakes of the digital battle. Strategically and functionally, the expanding reliance of current innovatively progressed powers on organizations and data frameworks make new sorts of exploitable weaknesses. Second, as modem social orders including the militaries that reflect them have kept on advancing, they have become perpetually reliant upon a progression of interconnected, progressively powerless "basic foundations" for their viable working. These frameworks not just have altogether expanded the everyday proficiency of pretty much all aspects of our general public, however, they have likewise presented new sorts of weaknesses.

**PROBLEM STATEMENT**

Web security is an ever-changing field, and the threats never end. For developers of web applications and sites, the target is always keeping the data protected. Whether you’re creating your next big idea or are working to maintain your existing product or business, you’ll certainly want to keep the fruits of your labour protected.

**OBJECTIVES**

Building GUI based tool to find vulnerabilities

* Detect Cross-site scripting (XSS) vulnerability in web pages using requests and BeautifulSoup
* Display input field, payload, input type etc of XSS vulnerability
* Detect SQL injection vulnerability in web pages using requests and BeautifulSoup
* Display Table names, Column names etc

**METHODOLOGY**

In this scenario, we have to check for vulnerabilities like Cross site scripting and sql injection on a URL.

User has to enter the URL and then system will automatically detect whether the vulnerabilities lies or not.

**INPUT Type:** URL.

**SQL Injection**: Common attack vector that uses malicious SQL code for backend database manipulation to access information that was not intended to be displayed.

This information may include any number of items, including sensitive company data, user lists or private customer details.

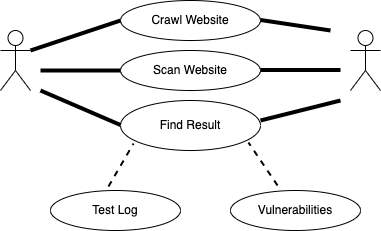
**Cross-site Scripting:** Cross-site scripting vulnerabilities normally allow an attacker to masquerade as a victim user, to carry out any actions that the user is able to perform, and to access any of the user's data.

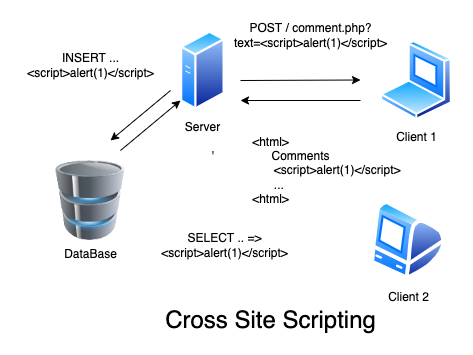
If the victim user has privileged access within the application, then the attacker might be able to gain full control over all of the application's functionality and data.

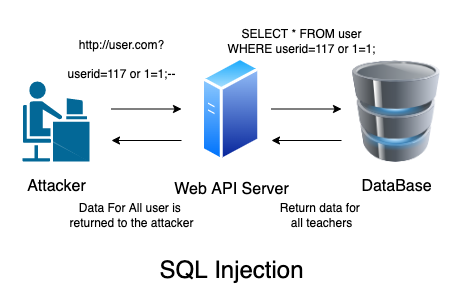
**Authentication:** Authentication is only providing security to the framework. Here each should go into the framework through login page. A user should provide his credentials like username and password for sign into the framework. For that the framework keeps up information for all users. At whatever point a user enters his username and password, it checks in the database for user presence. On the off chance that the user is exists he can be treated as a valid user. In any case the solicitation will toss back.

**Authorization:** Authorization is done when the user forgets the password and want to renew the password for that we are providing a service of OTP by which we send OTP on the registered no in a real time and take the OTP for authorization if OTP matches then he/she can renew its passwords.

**Data flow Diagram**







**Workflow**



**SYSTEM REQUIREMENTS SPECIFICATION**

Hardware requirement:

* RAM: 4GB
* Disk Space: 1GB

Software requirement:

* PyCharm (or any other Python IDE)
* VScode (installed with python 3)

Operating System:

* Windows
* linux

**REFERENCES**

1. https://docs.python.org/3/library/socket.html

# https://realpython.com/

# https://www.geeksforgeeks.org/

# https://www.tutorialspoint.com/python3/python\_networking.htm#:~:text=Sockets%20are%20the%20endpoints%20of,%2C%20UDP%2C%20and%20so%20on.

1. https://www.geeksforgeeks.org/python-program-to-generate-one-time-password-otp/
2. https://owasp.org/www-community/attacks/SQL\_Injection
3. https://owasp.org/www-community/attacks/xss/
4. https://www.balbix.com/insights/what-is-vulnerability-scanning/#:~:text=Vulnerability%20scanning%20is%20the%20process,the%20exposure%20of%20sensitive%20data.

**Synopsis Draft verified by**

**Project Guide HOD**

Mr. Keshav Kaushik

**(Name & Sign) (Dept. of Systemics)**

Keshav Kaushik

Keshav