

# Web Vulnerability Scanner - Project Report

## -By Prakanth V

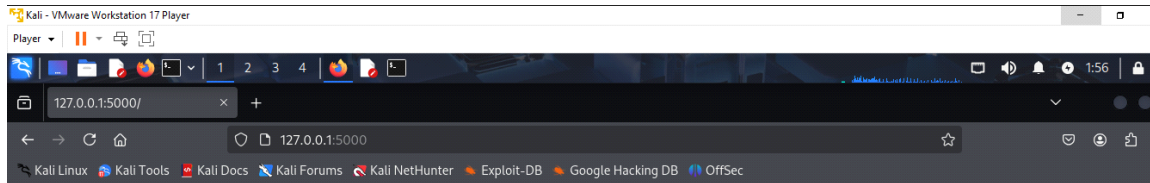
### Objective

Build a Python-based web vulnerability scanner that identifies common web application vulnerabilities such as Cross-Site Scripting (XSS) and SQL Injection (SQLi). The scanner uses a web interface for scanning URLs and displaying results in real-time.

### Tools & Technologies

Tool	Purpose
Python 3.8+	Core scripting language
Flask	Web framework
Requests	HTTP request handling
BeautifulSoup	HTML parsing and crawling
Regex	Pattern matching for error detection

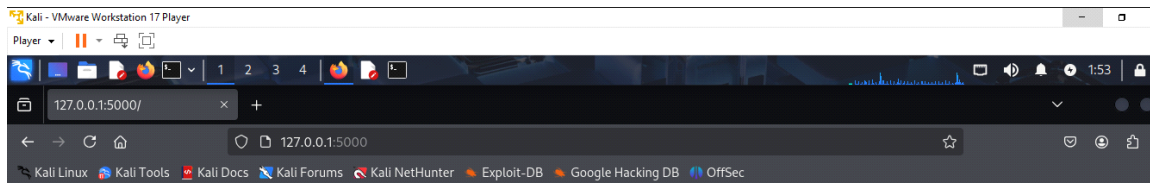
## Screenshots



### Web Vulnerability Scanner

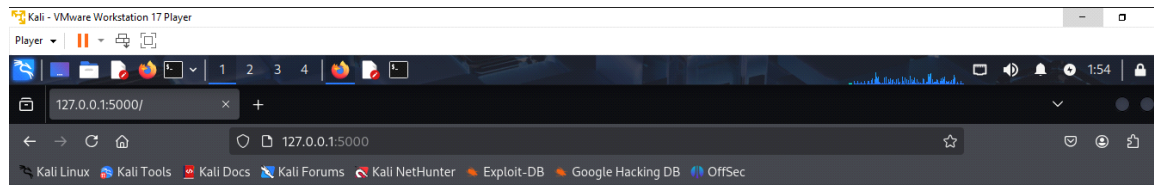
Target URL:

Scan Now

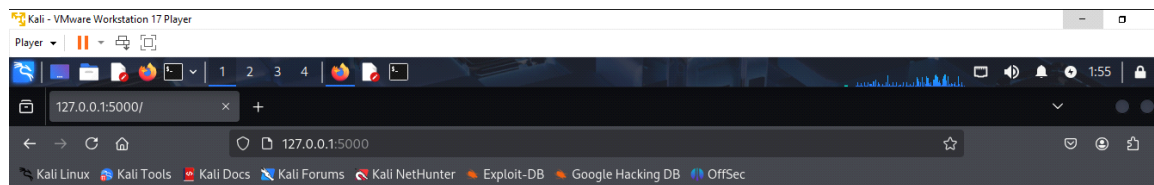


### Scan Results for http://testphp.vulnweb.com/

- **XSS** at http://testphp.vulnweb.com/index.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/index.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/index.php  
Payload: '; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/categories.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/categories.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/categories.php  
Payload: '; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/artists.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/artists.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**



- **SQL Injection** at http://testphp.vulnweb.com/disclaimer.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/cart.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/cart.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/cart.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/guestbook.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/guestbook.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/guestbook.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/categories.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/categories.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**



- **XSS** at http://testphp.vulnweb.com/categories.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/categories.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/artists.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/artists.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/artists.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**
- **XSS** at http://testphp.vulnweb.com/cart.php  
Payload: <script>alert(1)</script>  
Severity: **High**
- **XSS** at http://testphp.vulnweb.com/cart.php  
Payload: " onmouseover="alert(1)"  
Severity: **High**
- **SQL Injection** at http://testphp.vulnweb.com/cart.php  
Payload: ' ; DROP TABLE users; --  
Severity: **Critical**

[← Run another scan](#)

## How It Works

1. User inputs a URL.
2. The crawler finds all internal links and form elements.
3. Payload injection occurs in form inputs.
4. Responses are analyzed for reflection or error signatures.

5. Detected vulnerabilities are displayed in a table.

## Vulnerability Testing

### XSS (Cross-Site Scripting)

Payload: `<script>alert(1)</script>`

Detection: If payload is reflected in HTML response.

### SQL Injection

Payload: `' OR 1=1 --`

Detection: If response contains SQL errors (e.g., SQL syntax error).

### Example Report Entry

Type	URL	Payload	Severity
SQL Injection	<a href="http://test.com/search">http://test.com/search</a>	<code>' OR 1=1 --</code>	Critical
XSS	<a href="http://test.com/comment">http://test.com/comment</a>	<code>&lt;script&gt;alert(1)&lt;/script&gt;</code>	High

### Ethical Notice

⚠ This tool is for educational purposes only. Only scan domains you own or are authorized to test.