Abdi Osman

Information Management & Technology student at Syracuse University

Case Study 1: Vulnerability Management Lab on Kali Linux

Overview

This project demonstrates how to set up a complete vulnerability management lab using Kali Linux. I deployed OWASP Juice Shop (an intentionally vulnerable web app), scanned it using common security tools, identified weaknesses, and documented mitigation strategies.

Lab Environment

Component | Details

Host OS | Windows 1

Virtualization | VirtualBox running Kali Linux

Web App | OWASP Juice Shop (Node.js)

Tools Used

Tool | Purpose

Nmap | Port scanning, service detection

Nikto | Web server vulnerability scanning

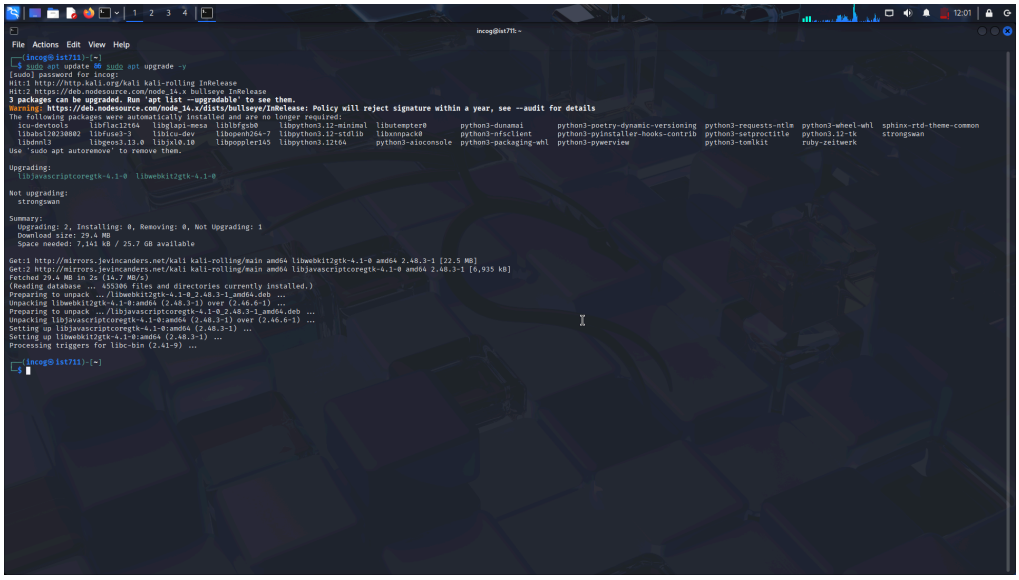
Sqlmap | SQL injection testing

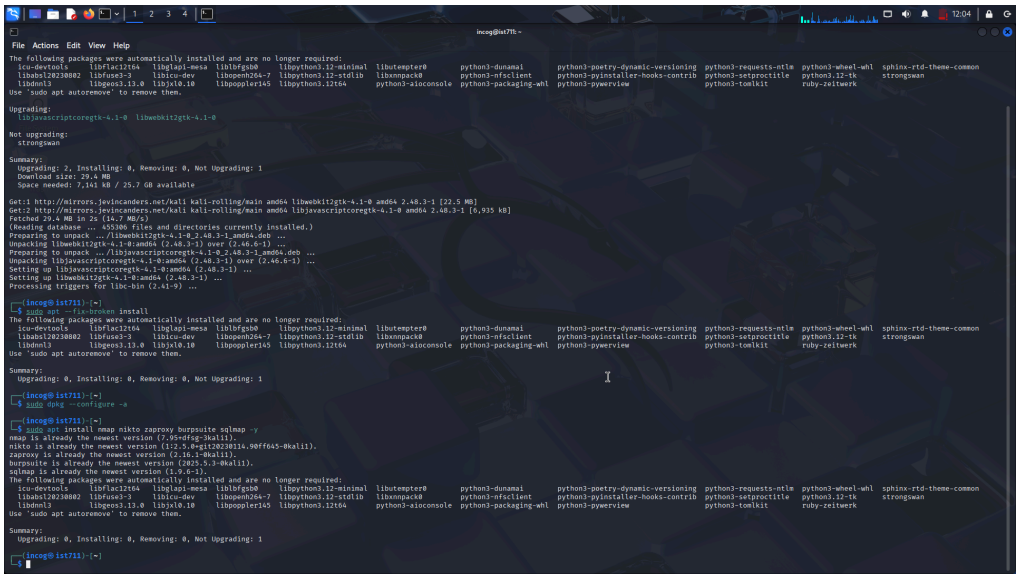
Burp Suite | Manual web vulnerability testing

OWASP ZAP | Automated web security scanner

1. **Environment Setup**

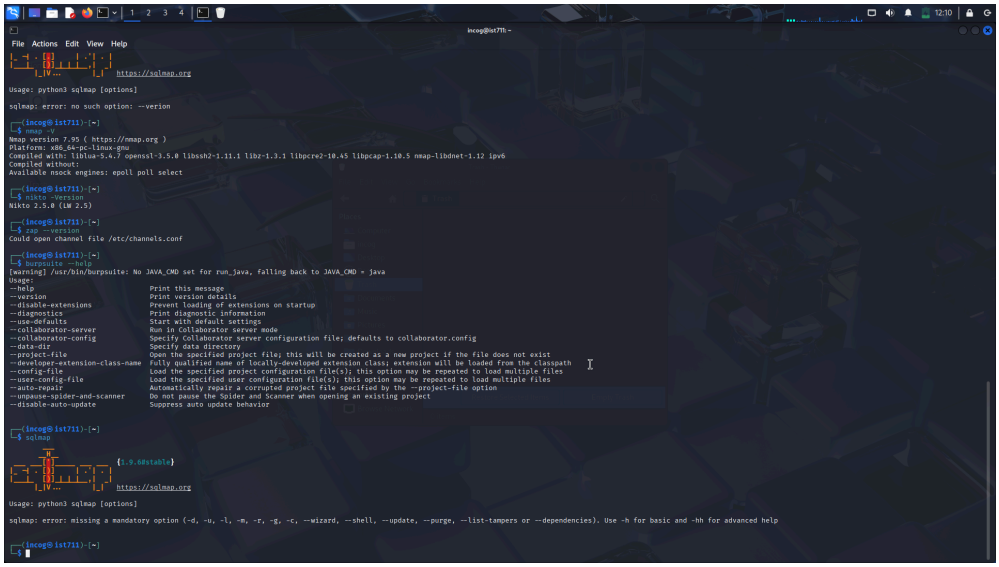
- Installed Kali Linux on VirtualBox (Windows 11 host).
- Updated the system packages:

```bash

- sudo apt update && sudo apt upgrade -y




2. **Installed Security Tools**

- sudo apt install nmap nikto zaproxy burpsuite sqlmap git nodejs npm -y
```



Followed by:

- nmap -V
- nikto -Version
- zap --version
- burpsuite --help
- sqlmap --version

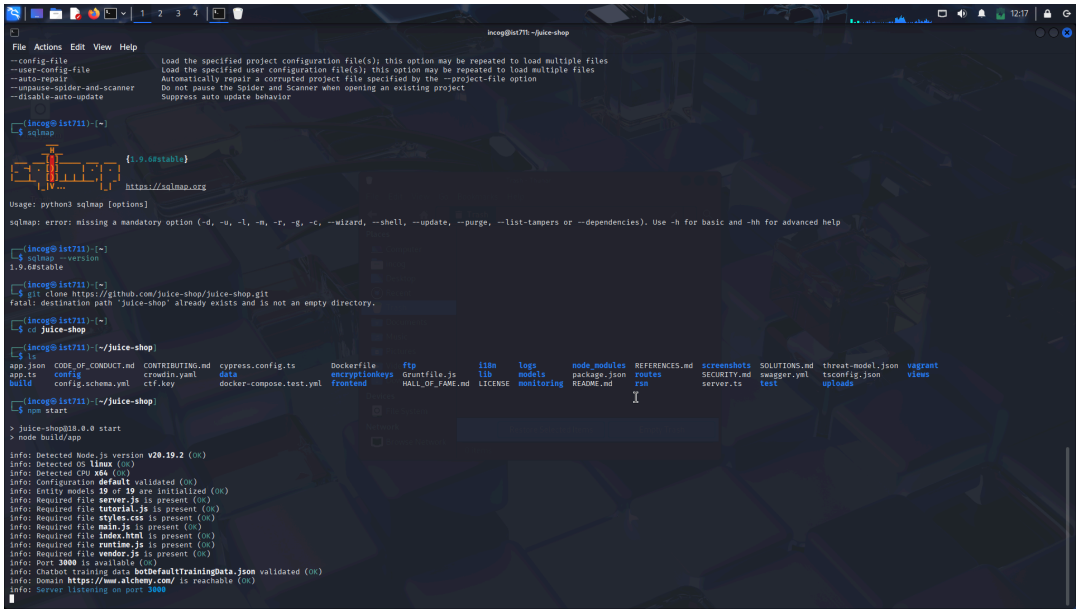


3. Deployed OWASP Juice Shop

- git clone https://github.com/juice-shop/juice-shop.git
- cd juice-shop
- npm install
- npm start

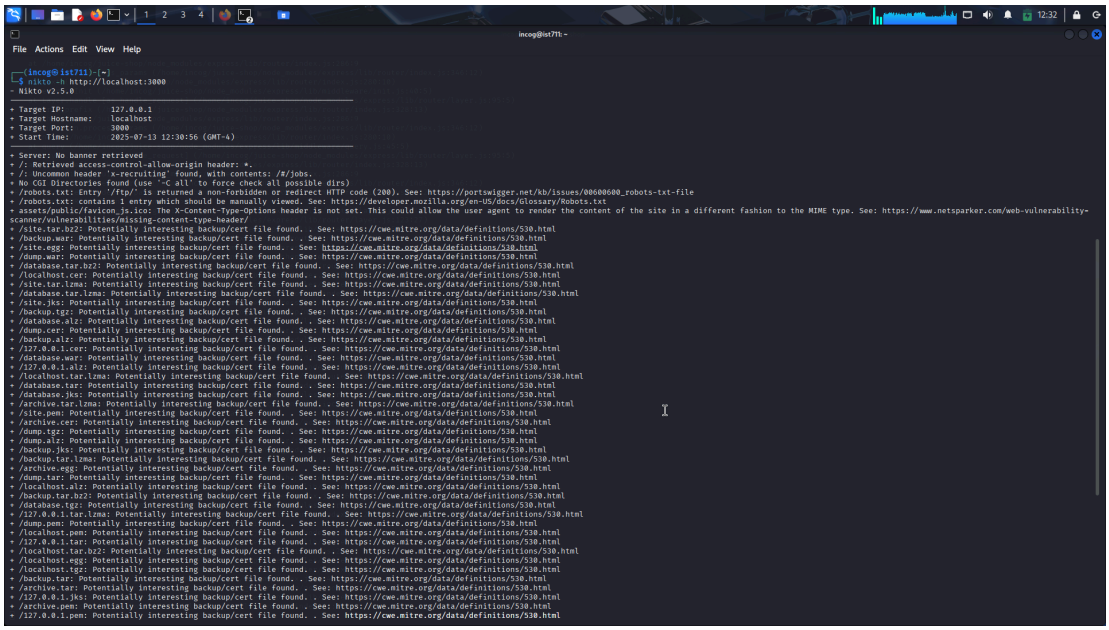
4. Vulnerability Scanning & Results

- Nmap Scan
- nmap -sV -p- localhost
    - Found open port 3000 running HTTP service.



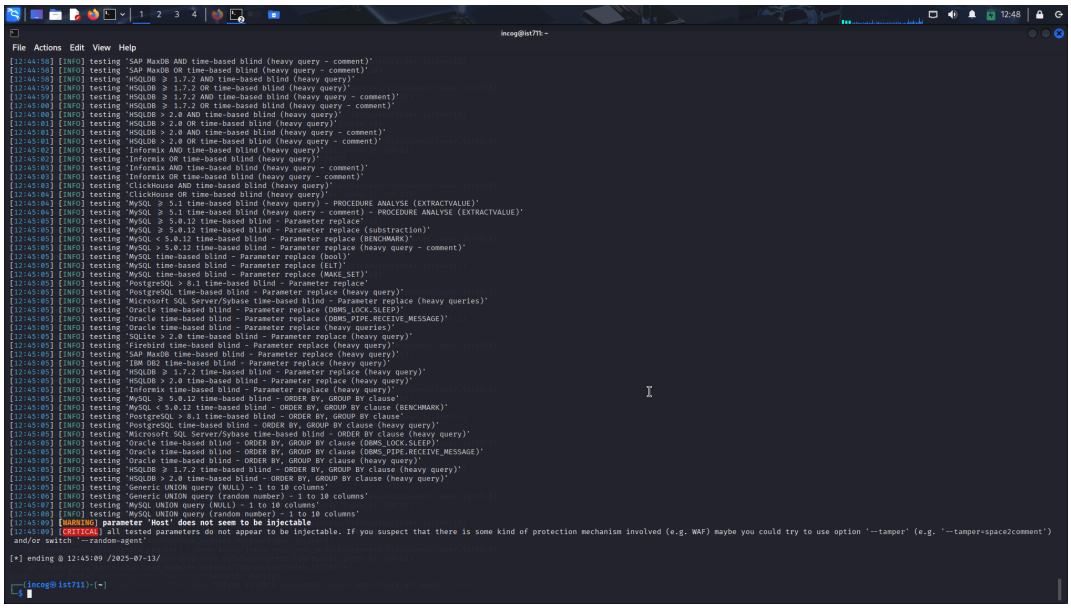
5. Nikto Web Vulnerability Scan

- nikto -h http://localhost:3000
    - Identified missing security headers.
    - Found potentially sensitive .tar, .zip, and .cert backup files.



6. SQLMap Injection Testing

- sqlmap -u "http://localhost:3000/?id=1" --batch --risk=3 --level=5
  - Extensively tested multiple injection techniques across:
    - MySQL, PostgreSQL, Oracle, MSSQL, SQLite, Firebird, ClickHouse, SAP MaxDB and more.
  - Ultimately determined:
    - [CRITICAL] all tested parameters do not appear to be injectable



Vulnerability Summary Table

Vulnerability	Tool	Description	Risk	Recommendation
Open port 3000 externally exposed	Nmap	Juice Shop HTTP server running on 3000	Medium	Use firewall to restrict access
Missing X-Content-Type-Options	Nikto	HTTP header not set	Low	Add nosniff header in server config
Potential sensitive backups found	Nikto	.tar, .zip, .cert files present	High	Remove or secure backups
Outdated JavaScript dependencies	npm	Multiple high/critical CVEs on install	High	Patch or containerize app
SQL injection attempts safe	sqlmap	No injection vulnerabilities found	N/A	Maintain monitoring

Recommendations & Next Steps

1. Restrict Juice Shop port 3000 to only trusted IPs.
2. Add secure HTTP headers (X-Content-Type-Options, Strict-Transport-Security, etc.).
3. Remove or properly secure backup/archive files on the web server.

4. Keep Node dependencies up to date to reduce known CVEs.
5. Run regular automated scans (Nikto, sqlmap) and periodic manual reviews (Burp Suite, ZAP).

Conclusion

Through this cybersecurity project, I successfully demonstrated the process of identifying, analyzing, and mitigating vulnerabilities within a deliberately insecure web application. Beginning with setting up a vulnerable environment using tools like OWASP Juice Shop on a Kali Linux virtual machine, I employed industry standard scanning and testing utilities such as OWASP ZAP and Burp Suite to uncover multiple security flaws. These included issues like broken authentication, sensitive data exposure, and improper input validation.

By systematically documenting each vulnerability and replicating exploits, I not only highlighted the risks they posed but also practiced responsible remediation techniques. I implemented secure coding practices and configuration adjustments to close these security gaps, then re-tested the application to verify the effectiveness of the fixes.

This project underscored the critical importance of proactive security testing and continuous vulnerability management in protecting web applications from real world cyber threats. It also helped solidify my hands-on experience with common attack vectors and defensive measures, reinforcing concepts that are vital in the cybersecurity field.

Ultimately, completing this project enhanced both my technical skill set and my understanding of secure software development lifecycles. It also produced a tangible portfolio piece that illustrates my ability to conduct security assessments, document findings, and implement meaningful improvements to all essential capabilities for roles in cybersecurity analysis and consulting.

Credits

- OWASP Juice Shop
- Tools: Nmap, Nikto, sqlmap, OWASP ZAP, Burp Suite
- *Chatgpt*, chatgpt.com/. Accessed 13 July 2025.