Penetration Testing Report
El Banco Bank

Introduction

This penetration test was conducted as part of a gray-box security assessment for El Banco Bank. The objective of this engagement was to identify security vulnerabilities that malicious attackers could exploit to gain unauthorized access to sensitive data and assets.

The scope of the test included reconnaissance, scanning, and log analysis to detect potential attack vectors and recommend appropriate mitigations.

Tasks and Findings

Task 1: Half-Open Scan

A half-open SYN scan was performed using Nmap to quickly identify all open ports in the network without establishing full TCP connections.

Command Used:

nmap -sS -p- -T4 192.80.0.16

Findings:

- The following open ports were identified:
    - Port 22 (SSH)
    - Port 80 (HTTP)
    - Port 443 (HTTPS)
    - Port 3306 (MySQL)
- Potential risks:
    - Open SSH port may be vulnerable to brute force attacks.
    - Open MySQL port may be exposed to SQL injection risks.
    - Web servers (ports 80 and 443) may be susceptible to application-layer attacks.

Mitigation:

- Implement firewall rules to restrict access to SSH and MySQL.
- Enforce strong authentication mechanisms.
- Use intrusion detection systems (IDS) to monitor access attempts.

Task 2: Aggressive Scan with OS & Version Detection

An aggressive scan was performed to gather details about the target system, including OS detection, version detection, script scanning, and traceroute.

Command Used:

nmap -A -T4 192.80.0.16

Findings:

- The target system is running **Linux Ubuntu 20.04 LTS**.
- The web server identified is **Apache 2.4.41**.
- The database server is **MySQL 5.7.34**.
- The system allows directory traversal via file inclusion vulnerabilities.

Mitigation:

- Keep all software updated with the latest security patches.
- Disable unnecessary services and ports.
- Implement proper access control mechanisms.
- Configure web application firewalls (WAF) to detect and block malicious requests.

Task 3: Log Analysis and Attack Identification

Suspicious HTTP GET requests were found in the server logs, indicating potential security threats.

Log 1: XSS Attack
18.66.78.71 - - [22/Dec/2021:16:18:20 +0300] "GET /media/system/js/caption.js HTTP/1.1" 200 751
"http://18.66.78.71/?wvstest=javascript:domxssExecutionSink(1,%22'%5C%22%3E%3Cxsstag%3E()locxss%22)"

Attack Type: Cross-Site Scripting (XSS)

Impact: Allows attackers to execute arbitrary JavaScript in a user's browser, leading to session hijacking and phishing attacks.

Mitigation:

- Implement input validation and output encoding.
- Use Content Security Policy (CSP) headers.
- Sanitize user inputs to prevent malicious script execution.

Log 2: Directory Traversal / Local File Inclusion (LFI) Attack
18.66.78.71 - - [22/Dec/2021:15:20:03 +0300] "GET /DVWA/vulnerabilities/fi/?page=%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2f..%2fetc%2fpasswd HTTP/1.1"

Attack Type: File Inclusion / Directory Traversal

Impact: Allows attackers to access sensitive system files, such as `/etc/passwd`, potentially leading to privilege escalation.

Mitigation:

- Validate and sanitize user inputs to prevent directory traversal.
- Disable directory listing in the web server configuration.
- Restrict file access permissions.

Log 3: SQL Injection Attack
18.66.78.71 - - [22/Dec/2021:15:19:59 +0300] "GET /DVWA/vulnerabilities/fi/?page=%27AND%201%3dcast(0x5f21403264696c656d6d61%20as%20varchar(8000))%20or%20%271%27%3d%27 HTTP/1.1"

Attack Type: SQL Injection (SQLi)

Impact: Enables attackers to manipulate database queries, extract sensitive information, and potentially take over the database.

Mitigation:

- Use parameterized queries and prepared statements.
- Implement Web Application Firewalls (WAF) to detect and block SQL injection attempts.
- Restrict database permissions to minimize the impact of a successful attack.

Conclusion

This penetration test revealed several critical vulnerabilities that could be exploited by malicious actors. Immediate steps should be taken to mitigate these risks to ensure the security of El Banco Bank's network and customer data.

Key Recommendations:

1. **Implement Strong Access Controls:**
    - Restrict access to sensitive services (SSH, MySQL) using firewall rules.
    - Enforce strong authentication and MFA.
2. **Patch and Update Systems:**
    - Regularly update OS, web servers, and databases.
    - Monitor security advisories for software vulnerabilities.
3. **Harden Web Applications:**
    - Implement WAF to detect and prevent XSS, SQLi, and LFI attacks.
    - Perform security audits and code reviews.
4. **Continuous Monitoring:**
    - Deploy IDS/IPS for real-time threat detection.
    - Regularly review logs for suspicious activities.