MyCompany

Security Assessment Finding Report

MAY XX,XXXX

Author: Daniel Hidalgo Pagés

TABLE OF CONTENTS

Confidentiality Statement

This document is the exclusive property of MyCompany and the author Daniel Hidalgo Pagés. This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires the consent of MyCompany. MyCompany may share this document with auditors under non-disclosure agreements to demonstrate penetration test requirement compliance.

Disclaimer

A penetration test is considered a snapshot in time. The findings and recommendations reflect the information gathered during the assessment and not any changes or modifications made outside of that period. Time-limited engagements do not allow for a full evaluation of all security controls. Daniel Hidalgo Pagés prioritized the assessment to identify the weakest security controls an attacker would exploit. Daniel Hidalgo Pagés recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

Executive Summary

The penetration test conducted for MyCompany aimed to evaluate the security posture of the organization's network and web assets. This assessment was performed as a black box test, meaning no prior knowledge of the internal workings of the target systems was provided to the testers. The primary goals were to identify vulnerabilities, assess potential risks, and recommend remediation actions to enhance overall security.

Objectives and Scope

Test Objective: To evaluate the security of MyCompany infrastructure by identifying vulnerabilities and assessing their potential impact.

Summary of findings

The penetration test uncovered multiple critical vulnerabilities, including buffer overflow, local file inclusion (LFI), and an outdated Python version 2.7.9 vulnerability (Python Exploit for CGI). These vulnerabilities exposed the web server (10.20.30.40) to severe risks, including remote code execution and unauthorized access.

Furthermore, the test revealed an escalation of privileges to root on the web server via a misconfigured cron job running with root privileges. This allowed for complete control over the system, posing a significant threat to the organization's data integrity and confidentiality.

Additionally, the test identified a critical directory traversal vulnerability in the Nginx web server (version 1.14.0) running on the web server. Exploiting this flaw could allow an attacker to access and modify sensitive files outside the web root directory, further compromising the server's security.

Overall Conclusions

The findings underscore the urgent need for MyCompany to address the identified vulnerabilities to mitigate potential security breaches and data compromises. Failure to address these issues promptly could result in severe repercussions, including financial losses and damage to the organization's reputation.

Vulnerability 1: Local File Inclusion (LFI) on Web Server (10.20.30.40)

CVSS Score: 7.5 (High)

Description: The web server at IP address 10.20.30.40 is vulnerable to local file inclusion attacks.

Impact/Risk: High

Proof of Concept (PoC):

Note: add screenshots at each step

1. Initial discovery:

Conducted a web application scan using Burp Suite to identify potential vulnerabilities.

Discovered the presence of file inclusion parameters in the URL.

2. Identification of LFI Vulnerability:

Detected potential LFI vulnerability by manipulating the file inclusion parameter to access sensitive files such as /etc/passwd.

3. Exploitation with Burp Suite:

Utilized Burp Suite to further assess and exploit the LFI vulnerability.

Burp Suite confirmed the vulnerability and allowed access to sensitive system files.

Tools Used: Nmap, Burp Suite

Recommendations:

- Implement input validation and sanitization to prevent LFI vulnerabilities.

- Disable unnecessary file inclusion functionalities and restrict file access permissions.

Vulnerability 3: Outdated Python Version 2.7.9 Vulnerability (Python Exploit for CGI) on Web Server (10.20.30.40)

CVSS Score: 9.0 (Critical)

Description: The web server at IP address 10.20.30.40 is running an outdated Python version 2.7.9, which is vulnerable to exploits through CGI.

Impact/Risk: High

Proof of Concept (PoC):

Note: add screenshots at each step

1. Initial discovery:

Conducted a vulnerability scan using Nessus to identify outdated software and potential vulnerabilities.

Discovered the web server is running Python version 2.7.9 with CGI enabled.

2. Identification of Python CGI Exploit:

Detected potential Python CGI exploit by sending crafted input to the CGI script, resulting in remote code execution.

3. Exploitation with Custom Python Script:

Developed a custom Python script to exploit the CGI vulnerability.

The script confirmed the vulnerability and provided a reverse shell with web server privileges.

Tools Used: Nessus, Custom Python Script

Recommendations:

- Upgrade Python to the latest version to mitigate known vulnerabilities.
- Disable or restrict CGI scripts if not necessary for web application functionality.

Remediation Report

In this section, we will provide detailed recommendations to remediate the identified vulnerabilities and enhance the security posture of MyCompany. Each remediation step is tailored to address the specific vulnerabilities found during the penetration test.

Local File Inclusion (LFI) on Web Server (10.20.30.40)

Recommendation:

- Implement strict input validation and sanitization to ensure that file paths are not manipulated by user input.
- Disable unnecessary file inclusion functionalities within the web application.
- Restrict file access permissions to limit the impact of LFI vulnerabilities.
- Conduct regular security audits and penetration testing to identify and mitigate LFI vulnerabilities.

Outdated Python Version 2.7.9 Vulnerability (Python Exploit for CGI) on Web Server (10.20.30.40)

Recommendation:

- Upgrade Python to the latest version to mitigate known vulnerabilities and security issues.
- Disable or restrict the use of CGI scripts if they are not necessary for the web application's functionality.
- Implement application whitelisting to control which scripts can be executed by the web server.
- Conduct regular vulnerability scans to identify and address outdated software components.

Host summary table

Host (IP)	Open Ports	Services	Obtained Access?	Vulnerabilities Exploited
10.90.60.80	22, 80	SSH, HTTP	YES	Local File Inclusion (LFI), Outdated Python Version 2.7.9 Vulnerability