A picture containing text, sign

Description automatically generated

Fake Company

Web Application Security Assessment

## Business Confidential

*Date: March 6th, 2025 Project: ABC123 Version 1.0*

# Confidentiality Statement

This document is the exclusive property of Fake Company and TCM Security (TCMS). This document contains proprietary and confidential information. Duplication, redistribution, or use, in whole or in part, in any form, requires consent of both Fake Company and TCMS.

Fake Company 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. TCMS prioritized the assessment to identify the weakest security controls an attacker would exploit. TCMS recommends conducting similar assessments on an annual basis by internal or third-party assessors to ensure the continued success of the controls.

# Contact Information

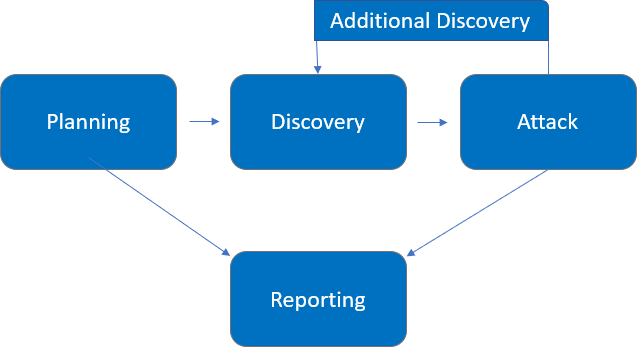
|  |  |  |
| --- | --- | --- |
| **Name** | **Title** | **Contact Information** |
| Fake Company | | |
| John Smith | Information Security Manager | Email: <jsmith@fakecompany.com> |
| TCM Security | | |
| Shawn | Penetration Tester | Email: <tester@tcm-sec.com> |

# Assessment Overview

From February 4th, 2024 to February 7th, 2024, Fake Company engaged TCMS to evaluate the security posture of its web application compared to current industry best practices that included an web application penetration test. All testing performed is based on the NIST *SP 800-115 Technical Guide to Information Security Testing and Assessment, OWASP Testing Guide (v4), and customized testing frameworks*.

Phases of penetration testing activities include the following:

* Planning – Customer goals are gathered and rules of engagement obtained.
* Discovery – Perform scanning and enumeration to identify potential vulnerabilities, weak areas, and exploits.
* Attack – Confirm potential vulnerabilities through exploitation and perform additional discovery upon new access.
* Reporting – Document all found vulnerabilities and exploits, failed attempts, and company strengths and weaknesses.



# 

# Assessment Components

### Web Application Penetration Test

A web application penetration test emulates the role of an attacker targeting the application's functionality, logic, and security controls. The engineer will analyze the application to identify potential vulnerabilities, including common and advanced attacks such as injection flaws (SQLi, XSS), authentication and session management issues, access control weaknesses (e.g., IDOR), and business logic vulnerabilities. The goal is to exploit these vulnerabilities to access unauthorized functionality or data, escalate privileges, and potentially compromise user accounts or sensitive information. The engineer will also assess the application's defenses against techniques such as brute force, token impersonation, API abuse, and data exfiltration.

# Finding Severity Ratings

The following table defines levels of severity and corresponding CVSS score range that are used throughout the document to assess vulnerability and risk impact.

|  |  |  |
| --- | --- | --- |
| **Severity** | **CVSS V4**  **Score Range** | **Definition** |
| Critical | 9.0-10.0 | The vulnerability has a severe, widespread impact and is trivially exploitable. Attackers can fully compromise systems, execute arbitrary code remotely, or cause catastrophic failures with little effort. Immediate remediation is highly recommended. |
| High | 7.0-8.9 | The vulnerability has a significant impact and is relatively easy to exploit. It could lead to unauthorized access, data theft, system compromise, or service disruption with substantial consequences. |
| Moderate | 4.0-6.9 | The vulnerability has a moderate impact and may be somewhat easy to exploit. Attackers can gain partial access, disrupt functionality, or cause moderate damage, but not in a widespread or severe manner. |
| Low | 0.1-3.9 | The vulnerability has limited impact and is difficult to exploit. Exploiting it may require significant effort, special conditions, or does not result in serious consequences. |
| Informational | N/A | No vulnerability exists. Additional information is provided regarding items noticed during testing, strong controls, and additional documentation. |

# Risk Factors

Risk is measured by two factors: Likelihood and Impact:

### Likelihood

Likelihood measures the potential of a vulnerability being exploited. Ratings are given based on the difficulty of the attack, the available tools, attacker skill level, and client environment.

### Impact

Impact measures the potential vulnerability’s effect on operations, including confidentiality, integrity, and availability of client systems and/or data, reputational harm, and financial loss.

# Scope

|  |  |
| --- | --- |
| **Assessment** | **Details** |
| Web Application Penetration Test | http://fakecompany.com |

### Scope Exclusions

Per client request, TCMS did not perform any of the following attacks during testing:

* Denial of Service (DoS)
* Attacks against underlying infrastructure and hardware running the application (e.g. the operating system or container hosting the web application)

All other attacks not specified above were permitted by Fake Company.

### Client Allowances

Fake Company provided TCMS the following allowances:

* Access to network via vpn and port allowances

# Testing Summary

TCMS evaluated Fake Company’s web application security posture through penetration testing from February 4th, 2025, to February 7th, 2025. The following sections provide a high-level overview of vulnerabilities discovered, successful and unsuccessful attempts, and strengths and weaknesses.

### Scoping and Time Limitations

Scoping during the engagement did not permit denial of service or attacks on the underlying infrastructure.

Time limitations were in place for testing. Web application penetration testing was permitted for two

1. business days.

### Executive Summary

The web application assessment evaluated Fake Company’s web application's security posture from an external perspective. The TCMS team used automated vulnerability scanning tools such as ZAP and Nikto, alongside manual techniques, to evaluate common and advanced web application security vulnerabilities. The team focused on key areas such as authentication, authorization, session management, input validation, and various security headers.

The TCMS team found a critical vulnerability within the web application that could potentially be exploited by an attacker. The following findings are categorized according to the OWASP top 10.

**Broken Access Control:**

OWASP Summary: Broken Access Control occurs when an application does not properly restrict users from performing actions outside of their allowed privileges.

Finding: The web application allows users to access and administrative panel and delete other users accounts. This indicates inadequate access control mechanisms in place to prevent unauthorized actions.

**Additional Findings:**

**Lack of Content Security Policy (CSP)**:

The application does not implement a Content Security Policy (CSP), increasing the risk of attacks such as Cross-Site Scripting (XSS) and data injection. Without CSP, malicious scripts could be executed in users' browsers, potentially compromising sensitive data and overall security. Implementing a well-defined CSP helps mitigate these risks by restricting the sources from which scripts and other resources can be loaded.

### 

### Tester Notes and Recommendations

On a positive note, our testing team triggered lockout mechanisms during the initial login page assessment and vulnerability scanning.

Overall, the Fake Company application performed as expected for a first-time penetration test. We recommend that the Fake Company team thoroughly review the recommendations made in this report, patch the findings, and re-test annually to improve their overall internal security posture.

# Technical Findings and Remediation Recommendations

### Broken Access Control

### Severity: Critical

### Impact:

### Broken Access Control occurs when an application does not properly enforce restrictions on what authenticated or unauthenticated users can do. Specifically:

### Accessing Other Users' Records: Unauthorized access to an administrative panel without authentication indicates a lack of proper access controls, allowing attackers to view sensitive data and perform administrative actions.

### Deleting Other Users' Records: The ability to delete records without verification of ownership or authorization can result in data loss and operational disruption.

### Both issues stem from insufficient access controls, posing a significant risk to data security and application integrity.

### Reproduction steps:

### During unauthenticated testing, the tester noticed through ZAP spider that a request was made to <https://0a92003503d1c1f88610f73500d0008a.web-security-academy.net/robots.txt> with a 200 response.

### A computer screen shot of a computer code

### While inspecting the response the tester noticed a previously undiscovered directory of /administrator-panel.

### A white screen with black text AI-generated content may be incorrect.

### Back in the browser we confirm that we do have unauthenticated access to this administrative panel.

### A screenshot of a computer AI-generated content may be incorrect.

### With unauthenticated access to the administrative panel, the tester was able to delete all users from the application.

### A screenshot of a computer AI-generated content may be incorrect.

### 

### Remediation Recommendations:

### To remediate broken access control vulnerabilities like unauthorized editing or deletion of user records:

### Implement Object-Level Access Controls: Validate ownership of resources on every request by verifying that the user has the correct permissions to access or modify the target resource.

### Use Secure Authorization Checks: Enforce role- or permission-based access control on server-side operations. Avoid relying on client-side mechanisms for critical authorization decisions.

### Audit and Test Access Controls: Regularly test access controls through automated tools and manual testing to ensure they cover all endpoints.

### Log and Monitor Access Attempts: Implement logging for access requests to detect and respond to unauthorized attempts effectively.

### For more information: <https://owasp.org/Top10/A01_2021-Broken_Access_Control/>

### Lack of Content Security Policy

### Severity: Low

### Impact:

### The absence or misconfiguration of a Content Security Policy (CSP) increases the risk of attacks such as Cross-Site Scripting (XSS) and data injection. Without a properly defined CSP, attackers can exploit security weaknesses to execute malicious scripts, compromise user data, and manipulate website content, potentially leading to data breaches and unauthorized access.

### Reproduction steps:

### Capture application responses in a proxy like Burp Suite or observe the headers in the developer tools. Note the lack of Content Security Policy (CSP).

### A screenshot of a computer AI-generated content may be incorrect.

### Remediation Recommendations:

### To mitigate risks associated with the lack of a Content Security Policy (CSP), implement a restrictive CSP header to control the sources from which content can be loaded. A well-defined CSP helps prevent Cross-Site Scripting (XSS) and other injection-based attacks by limiting the execution of unauthorized scripts.

### Recommended CSP Implementation:

### Configure the web server or application to include a CSP header with a restrictive policy. Below is an example of a strong CSP header:

### Content-Security-Policy: default-src 'self'; script-src 'self' https://trusted-scripts.example.com; style-src 'self' 'unsafe-inline'; img-src 'self' data:; object-src 'none'; frame-ancestors 'none'; upgrade-insecure-requests; block-all-mixed-content

### Additional Scans and Reports

TCMS provides all clients with all report information gathered during testing. This includes Nikto files and full vulnerability scans in detailed formats. These reports contain raw vulnerability scans and additional vulnerabilities not exploited by TCM Security.

The reports identify hygiene issues needing attention but are less likely to lead to a breach, i.e. defense-in-depth opportunities. For more information, please see the documents in your shared drive folder labeled “Additional Scans and Reports”.



Last Page