Final Report
Web Application Penetration Test

For: JProg

Submitted by:

MainNerve, LLC.
201 E Pikes Peak Ave Suite 2025
Colorado Springs, CO 80903
www.mainnerve.com

April 6, 2026

Table of Contents

Summary of Changes

Change Information	Reason for Change	Date
Version 1.0	Initial Release	4/6/2026

1. How to Use This Report

This is the final report for the web application penetration test performed by MainNerve. This document was prepared in accordance with security best practices recommended by such organizations as NIST and WSTG. Remediation recommendations, where applicable, are found in line within each section. These recommendations specifically address the security concerns discussed in the respective sections. Wherever possible, screenshots are included to demonstrate the methods and findings encountered during the period of this test.

MainNerve has also provided a risk rating that describes the impact of each vulnerability or misconfiguration discovered during this test. This rating does not cover vulnerability scans or other assessments (if applicable) not directly part of this penetration test. The rating is based on the DREAD Threat Risk Modeling algorithm and is intended to assist with determining a proper course of action when performing remediation or analysis of the findings.

This report describes penetration testing that represents a point-in-time snapshot of the web application security posture. In accordance with security best practices, regular security assessments should be commissioned, especially after major changes to systems and/or networks.

2. Executive Summary

MainNerve conducted credentialed web application penetration testing for JProg ("Client"). The scope of the penetration test included the web application hosted at *https://adr2025build571.jprog.net/careware/rs/index.htm*.

Testing was performed using a combination of publicly available tools, commercial software, and proprietary utilities. Both manual and automated testing techniques were employed to identify and validate exploitable vulnerabilities.

The primary objective of this engagement was to compromise or gain unauthorized access to resources within the targeted web application. MainNerve followed the Web Security Testing Guide (WSTG) framework as the foundation for its web application penetration testing methodology.

The overall risk posed to JProg is **Medium**. The findings in this report should be remediated to minimize or significantly reduce the impact or risk of future malicious attacks.

2.1 Summary of Findings

Severity	#	Finding	Description
MEDIUM	1	**User Account Enumeration via Account Lockout Message**	The application provides differing responses during authentication, enabling attackers to identify valid user accounts.
MEDIUM	2	**Use of Outdated Software Components**	The application relies on outdated software versions that may contain known vulnerabilities exploitable by attackers.
LOW	3	**Insecure Content Security Policy**	The application implements a misconfigured Content Security Policy that fails to adequately prevent injection attacks such as XSS.

3. Project Scope

MainNerve completed a gray box or "partial knowledge" web application penetration test. The Client identified an approved target from which MainNerve did an application layer penetration test.

The following system was included in the entirety of this web application penetration test. No other systems were included nor detected during any phase of the testing.

Hostname	IP Address
https://adr2025build571.jprog.net/careware/rs/index.htm	34.208.56.24

3.1 Statement of Methodology

MainNerve employs a testing approach based on the Web Security Testing Guide framework that is acceptable for all compliance and regulatory standards. The methodology includes specific phases, such as, Reconnaissance, Mapping, Discovery, and Exploitation, with continual Reporting throughout the test.

The scope of the test included:

- Gray box web application penetration test of the following categories:
  - Deployment Management
  - Identity Management
  - Authentication
  - Authorization
  - Session Management
  - Input Validation / Injection
  - Error Handling
  - Logic / Flow
  - Cryptography
  - Client Side
- Attempts to penetrate the application by compromising or otherwise gaining unauthorized read-only access to the URL identified in the Rules of Engagement
- Penetration testing of the most critical vulnerabilities and/or misconfigurations found, if any
- Provide remediation recommendations as appropriate

It is also important to note that our assessment represents a point in time. Hence, this penetration test does not reflect zero-day exploits or other exploits not previously known. The Client should continue to regularly assess and/or audit its security posture.

4. Testing Checklist

The following are tests from the OWASP Web Security Testing Guide that were performed during this penetration test:

Status	Description
PASS	Test resulted in a favorable result.
LOW **MEDIUM** **HIGH**	Test resulted in one or more findings with a risk rating of LOW, MEDIUM, or HIGH.
DONE	A process used to aid in the performance of a security assessment, preparatory functions.
N/A	A test that is not applicable to the application being tested. This will usually have a note regarding the reason the test is not applicable.

Test ID	Test Name	Status	Notes
WSTG-INFO	**Information Gathering**		
WSTG-INFO-01	Conduct Search Engine Discovery and Reconnaissance for Information Leakage	DONE	
WSTG-INFO-02	Fingerprint Web Server	DONE	
WSTG-INFO-03	Review Webserver Metafiles for Information Leakage	DONE	
WSTG-INFO-04	Enumerate Applications on Webserver	N/A	Not in scope
WSTG-INFO-05	Review Webpage Content for Information Leakage	DONE	
WSTG-INFO-06	Identify Application Entry Points	DONE	
WSTG-INFO-07	Map Execution Paths Through Application	DONE	
WSTG-INFO-08	Fingerprint Web Application Framework	DONE	
WSTG-INFO-09	Fingerprint Web Application	INFO-08	Test merged with INFO-08
WSTG-INFO-10	Map Application Architecture	N/A	Not in scope

Test ID	Test Name	Status	Notes
WSTG-CONF	**Configuration and Deploy Management Testing**		
WSTG-CONF-01	Test Network Infrastructure Configuration	PASS	No issues found
WSTG-CONF-02	Test Application Platform Configuration	**MEDIUM**	See finding 5.2
WSTG-CONF-03	Test File Extensions Handling for Sensitive Information	PASS	No issues found
WSTG-CONF-04	Review Old Backup and Unreferenced Files for Sensitive Information	PASS	No issues found
WSTG-CONF-05	Enumerate Infrastructure and Application Admin Interfaces	PASS	No issues found
WSTG-CONF-06	Test HTTP Methods	PASS	No issues found
WSTG-CONF-07	Test HTTP Strict Transport Security	PASS	No issues found
WSTG-CONF-08	Test RIA Cross Domain Policy	N/A	Not in scope
WSTG-CONF-09	Test File Permission	PASS	No issues found
WSTG-CONF-10	Test for Subdomain Takeover	N/A	Not in scope
WSTG-CONF-11	Test Cloud Storage	N/A	Not found in application
WSTG-CONF-12	Testing for Content Security Policy	**LOW**	See finding 5.3

Test ID	Test Name	Status	Notes
WSTG-IDNT	**Identity Management Testing**		
WSTG-IDNT-01	Test Role Definitions	PASS	No issues found
WSTG-IDNT-02	Test User Registration Process	N/A	Not found in application
WSTG-IDNT-03	Test Account Provisioning Process	PASS	No issues found
WSTG-IDNT-04	Testing for Account Enumeration and Guessable User Account	**MEDIUM**	See finding 5.1
WSTG-IDNT-05	Testing for Weak or Unenforced Username Policy	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-ATHN	**Authentication Testing**		
WSTG-ATHN-01	Testing for Credentials Transported Over an Encrypted Channel	CRYP-03	Test merged with CRYPT-03
WSTG-ATHN-02	Testing for Default Credentials	PASS	No issues found
WSTG-ATHN-03	Testing for Weak Lock Out Mechanism	PASS	No issues found
WSTG-ATHN-04	Testing for Bypassing Authentication Schema	PASS	No issues found
WSTG-ATHN-05	Testing for Vulnerable Remember Password	PASS	No issues found
WSTG-ATHN-06	Testing for Browser Cache Weakness	PASS	No issues found
WSTG-ATHN-07	Testing for Weak Password Policy	PASS	No issues found
WSTG-ATHN-08	Testing for Weak Security Question Answer	N/A	Disabled in application
WSTG-ATHN-09	Testing for Weak Password Change or Reset Functionalities	PASS	No issues found
WSTG-ATHN-10	Testing for Weaker Authentication in Alternative Channel	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-ATHZ	**Authorization Testing**		
WSTG-ATHZ-01	Testing Directory Traversal File Include	PASS	No issues found
WSTG-ATHZ-02	Testing for Bypassing Authorization Schema	PASS	No issues found
WSTG-ATHZ-03	Testing for Privilege Escalation	PASS	No issues found
WSTG-ATHZ-04	Testing for Insecure Direct Object References	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-SESS	**Session Management Testing**		
WSTG-SESS-01	Testing for Session Management Schema	PASS	No issues found
WSTG-SESS-02	Testing for Cookies Attributes	PASS	No issues found
WSTG-SESS-03	Testing for Session Fixation	PASS	No issues found
WSTG-SESS-04	Testing for Exposed Session Variables	PASS	No issues found
WSTG-SESS-05	Testing for Cross Site Request Forgery	PASS	No issues found
WSTG-SESS-06	Testing for Logout Functionality	PASS	No issues found
WSTG-SESS-07	Testing Session Timeout	PASS	No issues found
WSTG-SESS-08	Testing for Session Puzzling	PASS	No issues found
WSTG-SESS-09	Testing for Session Hijacking	PASS	No issues found
WSTG-SESS-10	Testing JSON Web Tokens	N/A	Not found in application

Test ID	Test Name	Status	Notes
WSTG-INPV	**Input Validation Testing**		
WSTG-INPV-01	Testing for Reflected Cross Site Scripting	PASS	No issues found
WSTG-INPV-02	Testing for Stored Cross Site Scripting	PASS	No issues found
WSTG-INPV-03	Testing for HTTP Verb Tampering	PASS	No issues found
WSTG-INPV-04	Testing for HTTP Parameter pollution	PASS	No issues found
WSTG-INPV-05	Testing for SQL Injection		
WSTG-INPV-06	Testing for LDAP Injection	N/A	Not found in application
WSTG-INPV-07	Testing for XML Injection	N/A	Not found in application
WSTG-INPV-08	Testing for SSI Injection	N/A	Not found in application
WSTG-INPV-09	Testing for XPath Injection	N/A	Not found in application
WSTG-INPV-10	Testing for IMAP SMTP Injection	PASS	No issues found
WSTG-INPV-11	Testing for Code Injection	PASS	No issues found
WSTG-INPV-12	Testing for Command Injection	PASS	No issues found
WSTG-INPV-13	Testing for Format String Injection	PASS	No issues found

WSTG-INPV-14	Testing for Incubated Vulnerabilities	PASS	No issues found
WSTG-INPV-15	Testing for HTTP Splitting Smuggling	PASS	No issues found
WSTG-INPV-16	Testing for HTTP Incoming Requests	PASS	No issues found
WSTG-INPV-17	Testing for Host Header Injection	PASS	No issues found
WSTG-INPV-18	Testing for Server-Side Template Injection	PASS	No issues found
WSTG-INPV-19	Testing for Server-Side Request Forgery	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-ERRH	**Error Handling**		
WSTG-ERRH-01	Testing for Improper Error Handling	PASS	No issues found
WSTG-ERRH-02	Testing for Stack Traces	ERRH-01	Test merged with ERRH-01

Test ID	Test Name	Status	Notes
WSTG-CRYP	**Cryptography**		
WSTG-CRYP-01	Testing for Weak Transport Layer Security	PASS	No issues found
WSTG-CRYP-02	Testing for Padding Oracle	PASS	No issues found
WSTG-CRYP-03	Testing for Sensitive Information Sent Via Unencrypted Channels	PASS	No issues found
WSTG-CRYP-04	Testing for Weak Encryption	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-BUSLOGIC	**Business Logic Testing**		
WSTG-BUSL-01	Test Business Logic Data Validation	PASS	No issues found
WSTG-BUSL-02	Test Ability to Forge Requests	PASS	No issues found
WSTG-BUSL-03	Test Integrity Checks	PASS	No issues found
WSTG-BUSL-04	Test for Process Timing	PASS	No issues found

WSTG-BUSL-05	Test Number of Times a Function Can Be Used Limits	PASS	No issues found
WSTG-BUSL-06	Testing for the Circumvention of Workflows	PASS	No issues found
WSTG-BUSL-07	Test Defenses Against Application Misuse	PASS	No issues found
WSTG-BUSL-08	Test Upload of Unexpected File Types	PASS	No issues found
WSTG-BUSL-09	Test Upload of Malicious Files	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-CLIENT	**Client-side Testing**		
WSTG-CLNT-01	Testing for DOM Based Cross Site Scripting	PASS	No issues found
WSTG-CLNT-02	Testing for JavaScript Execution	PASS	No issues found
WSTG-CLNT-03	Testing for HTML Injection	PASS	No issues found
WSTG-CLNT-04	Testing for Client-Side URL Redirect	PASS	No issues found
WSTG-CLNT-05	Testing for CSS Injection	PASS	No issues found
WSTG-CLNT-06	Testing for Client-Side Resource Manipulation	PASS	No issues found
WSTG-CLNT-07	Test Cross Origin Resource Sharing	PASS	No issues found
WSTG-CLNT-08	Testing for Cross Site Flashing	PASS	No issues found
WSTG-CLNT-09	Testing for Clickjacking	PASS	No issues found
WSTG-CLNT-10	Testing WebSockets	N/A	Not found in application
WSTG-CLNT-11	Test Web Messaging	N/A	Not found in application
WSTG-CLNT-12	Test Browser Storage	PASS	No issues found
WSTG-CLNT-13	Testing for Cross Site Script Inclusion	PASS	No issues found

Test ID	Test Name	Status	Notes
WSTG-APIT	**API Testing**		
WSTG-APIT-01	Testing GraphQL	PASS	No issues found

5. Web Application Testing Results

MainNerve began testing of the web application by spidering/crawling the application in order to determine potential injection points and logic flaws. This provided a foundational understanding of the application's structure and allowed the team to identify areas for deeper analysis, such as form submissions, API endpoints, and authentication mechanisms. Following the initial mapping, the team conducted manual and automated testing to identify common web application vulnerabilities. The test team analyzed request/response behavior, session management, and access control mechanisms. Particular attention was paid to potential injection points, including URL parameters, form fields, headers, and cookies. As the testing progressed, the team performed targeted tests based on findings from the enumeration phase.

5.1 User Account Enumeration via Account Lockout Message

Current Rating
MEDIUM

MainNerve identified a username enumeration vulnerability within the application's login mechanism. This issue arises when the application discloses whether a submitted username exists based on differing authentication responses. Specifically, when an invalid username is provided, no account lockout occurs (Figure 1). In contrast, when a valid username is entered incorrect passwords are attempted repeatedly, the account becomes locked after a threshold of failed login attempts (Figure 2). This discrepancy in behavior allows an attacker to reliably distinguish between valid and invalid usernames without requiring knowledge of account credentials. As a result, this vulnerability increases the likelihood of successful password-based attacks, such as brute-force or credential stuffing, by enabling attackers to compile a list of valid accounts for targeted exploitation.

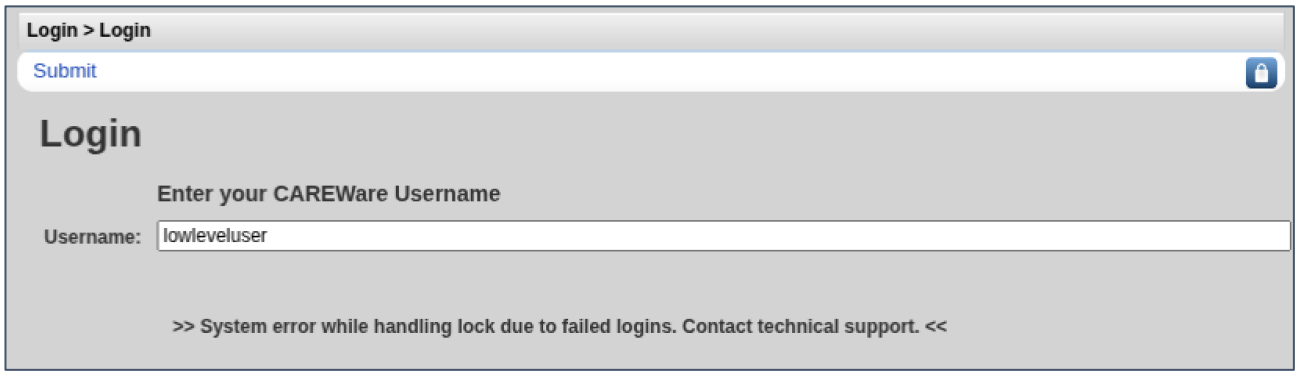

Figure 1 - Application response for a valid account lockout

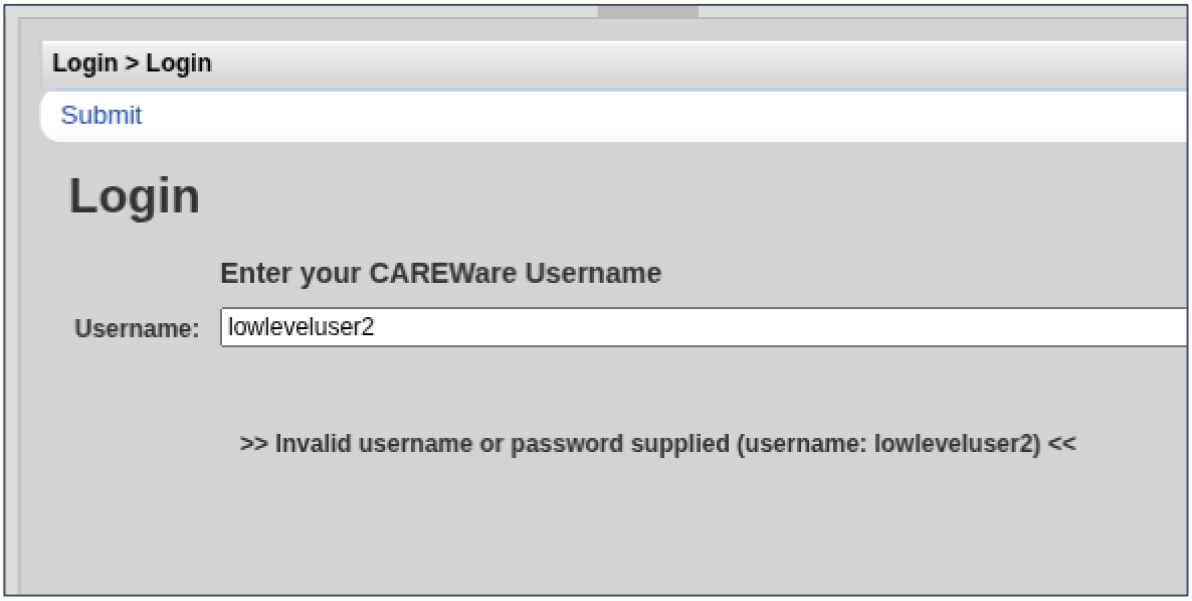

Figure 2 – Application response to a non-existent username with no lockout action

Recommendation:

Implement identical account lockout policies for both valid and invalid usernames or disable account lockout entirely in favor of rate limiting and CAPTCHA challenges. Additionally, account lockout notifications should not differentiate between valid and invalid usernames.

Reference:

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/03-Identity_Management_Testing/04-Testing_for_Account_Enumeration_and_Guessable_User_Account

5.2 Use of Outdated Software Components

Current Rating
MEDIUM

The application infrastructure was operating on Microsoft SQL Server 2022 RTM (build 16.0.1000.6) and Windows Server 2022 Datacenter (build 20348.10) (Figure 3). Both systems are running baseline RTM versions and have not been updated with numerous security patches released over the past two years. As a result, they remain susceptible to publicly known vulnerabilities and exploits affecting critical services, including database authentication mechanisms. These RTM versions, originally released in November 2022, do not incorporate important security fixes delivered through subsequent cumulative updates. The absence of these updates significantly increases the attack surface and exposes the infrastructure to known exploitation techniques that have since been mitigated in newer builds.

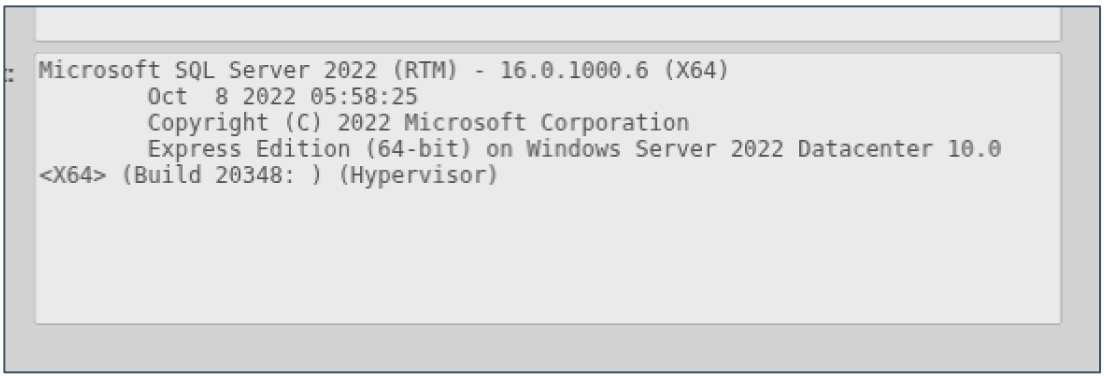

Figure 3 - Shows the outdated version of Microsoft SQL Server

Recommendation:

Apply the latest cumulative updates to both systems: SQL Server 2022 RTM CU24 and Windows Server 2022 KB5068787 (OS Build 20348.4405). Establish a regular patching schedule to apply security updates within 30 days of release and configure automatic update policies where possible.

Reference:

https://my.f5.com/manage/s/article/K17045144

5.3 Insecure Content Security Policy

Current Rating
LOW

The application implements a Content Security Policy (CSP); however, it does not define a default-src directive. The default-src directive serves as a fallback for all resource types that do not have explicitly defined source directives (e.g., script-src, img-src, style-src). In the absence of a default-src directive, any resource type without a specific directive is not restricted by the policy, allowing the browser to load resources from arbitrary origins. This misconfiguration weakens the intended protections of CSP and reduces its effectiveness as a defense-in-depth control against client-side attacks. Additionally, the CSP does not include the upgrade-insecure-requests directive. This directive instructs browsers to automatically upgrade HTTP requests to HTTPS. Without it, users may be exposed to mixed content issues, where insecure (HTTP) resources are loaded within secure (HTTPS) pages, increasing the risk of man-in-the-middle (MITM) attacks.

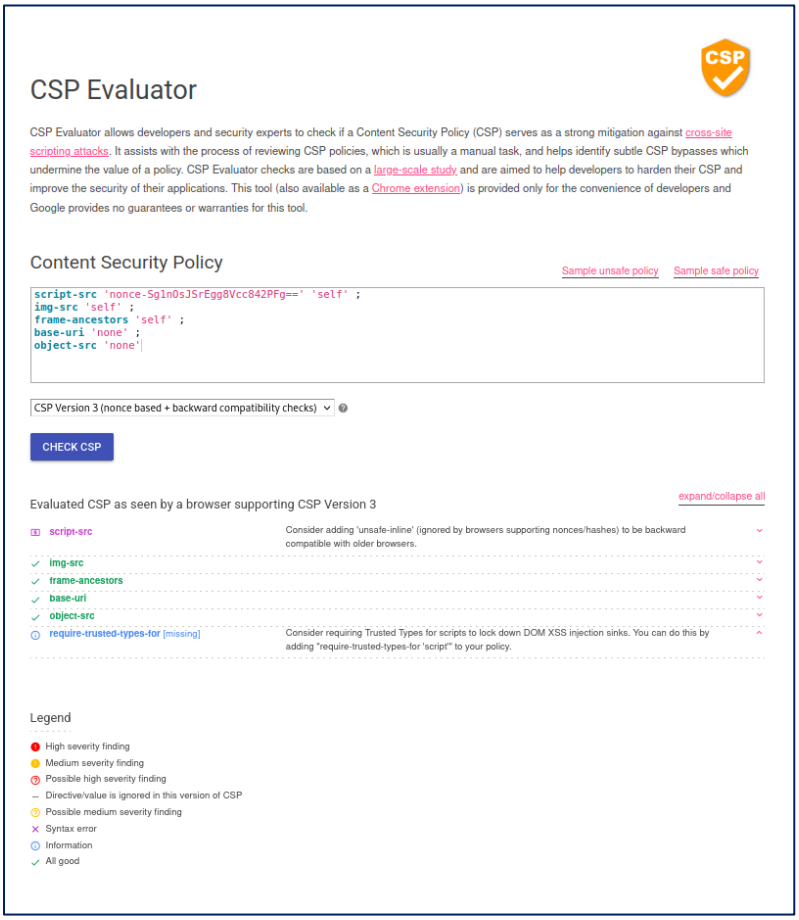

Figure 4 - Shows the application's CSP missing the default-src and upgrade-insecure-requests directives

Recommendation:

Implement a restrictive default-src directive to enforce a secure baseline for all resource types and add the upgrade-insecure-requests directive to ensure all resources are loaded over HTTPS.

Reference:

https://owasp.org/www-project-web-security-testing-guide/latest/4-Web_Application_Security_Testing/02-Configuration_and_Deployment_Management_Testing/12-Test_for_Content_Security_Policy

6. Testing Tools

For this penetration test, MainNerve may use any of the following tools:

Tool Name	Type
Burp Suite Professional	Web application testing proxy (Automated/Manual)
Nmap	Host/Service discovery and enumeration
SSL Scan	SSL/TLS configuration analysis
Gobuster	Directory and file enumeration tool
NSLookup	Domain name information query tool

7. Risk Rating Overview

MainNerve uses the DREAD threat modeling algorithm to determine the amount of risk posed to a system or infrastructure by vulnerabilities discovered during an assessment.

Damage Potential	**R**eproducibility	**E**xploitability	**A**ffected Users	**D**iscoverability
•If a threat exploit occurs, how much damage will be caused?	•How difficult is it to reproduce the attack?	•What is needed to exploit this threat?	•How many users will be affected?	•How easy is it to discover this threat?

Rating (Value)	High (10)	Medium (5)	Low (0)
Damage Potential	An attacker can gain privileged access; full system/network breach is possible	Some sensitive data may be accessed by an attacker, but full system breach is improbable	No sensitive data or information leakage
Reproducibility	An attacker can easily circumvent security controls	An attacker can bypass security controls under certain conditions	Very hard or impossible, even for administrators of the system
Exploitability	Requires little and/or novice-level skills	Requires moderate skills and malware is publicly available	Requires very high skill level with custom or advanced attack tools
Affected Users	All users. Critical processes significantly affected	Some users are affected. Critical processes operational	Very little to no impact on users or critical processes
Discoverability	No detection controls or measures in place	Partial security measures are implemented, providing limited detection capabilities without comprehensive coverage or blocking of attacks	Security controls in place that detect and block attacks; vulnerability not overt

7.1 Risk Value Calculation

MainNerve assigns a Risk Value to each relevant finding discovered during this assessment using the following calculation.

To determine the priority level, MainNerve uses the following table of values. The *overall* Risk Rating is calculated by the highest priority rating. That is, if an organization has four (4) Medium priorities and one (1) High, the overall Threat Risk Impact Rating would be High.

Risk Value	Priority
0 - 4	Low
5 – 7	Medium
8 and greater	High

7.2 Risk Impact Rating

Based on the findings and analysis by MainNerve, the risks are calculated in the following table. It should be noted that the DREAD algorithm, as calculated below, is used to compute a risk value, which is an average of all five categories.

Findings	D	R	E	A	D	Value	Priority
User Account Enumeration via Account Lockout Messages	7	5	5	5	5	27/5=5.4	**Medium**
Use of Outdated Software Components	5	5	5	5	5	25/5=5.0	**Medium**
Insecure Content Security Policy	3	3	2	2	7	17/5=3.4	**Low**

Based on these findings, MainNerve has determined that the overall risk impact is **Medium**. As a matter of due diligence, the Client should assume that the findings of this report are a point-in-time assessment and review the findings in this report with the recommendations to reduce its risk exposure through effective remediation. Periodic assessments should also be completed as a matter of best practice while ensuring a higher level of security.