|  |
| --- |
| April 30, 2025 |

|  |
| --- |
| Hackazon Vulnerability Assessment |

## Executive Summary

The vulnerability assessment was conducted between April 17, 2025, at 12:24 PM and April 17, 2025, at 12:29 PM to evaluate the vulnerabilities that are present in the web application Hackazon. The main purpose of this vulnerability assessment is to identify the vulnerabilities and weaknesses within the web application that could impact itself and its users. The scope of this assessment was directed specifically to the Hackazon web application to assess its security posture and recommend remedies to enhance it.

This vulnerability scan was performed using Tenable Nessus which is a widely recognized vulnerability scanning tool. The assessment identified 22 total vulnerabilities across the system which include 1 medium severity, 1 low severity, and 20 informational vulnerabilities. Based on these results, the overall inherent risk of the web application is low, but the medium and low-severity vulnerabilities should be addressed promptly to minimize risk.

Disclaimer: The results of this vulnerability assessment are based on what is found during the point in time. New vulnerabilities may emerge after the completion of this scan which may increase the risk profile. It is recommended to regularly perform security and vulnerability assessments to help maintain a strong security posture.

## Scan Results

The vulnerability scan was conducted using Tenable Nessus to assess the security posture of the web application Hackazon. The scan was done within the pre-defined scope of the publicly accessible web server. The scan was non-intrusive and meant to identify common web application vulnerabilities without disrupting services.

The vulnerabilities found in the scan results include medium, low, and informational severity findings which were categorized and ordered by their severity using Nessus’s rating system. Medium vulnerabilities represent security weaknesses that can be exploited with some effort and should be addressed as the priority in this case. Low vulnerabilities do not require as prompt a response as medium ones but should be addressed soon because they can still be exploited. Informational findings do not directly represent vulnerabilities but provide details and context of the system which could aid an attacker during exploitation.

The type of report provided includes a detailed vulnerability report with full technical details for each identified vulnerability which includes CVSS, VPR, and EPSS scores, plugin IDs, descriptions, and solutions.

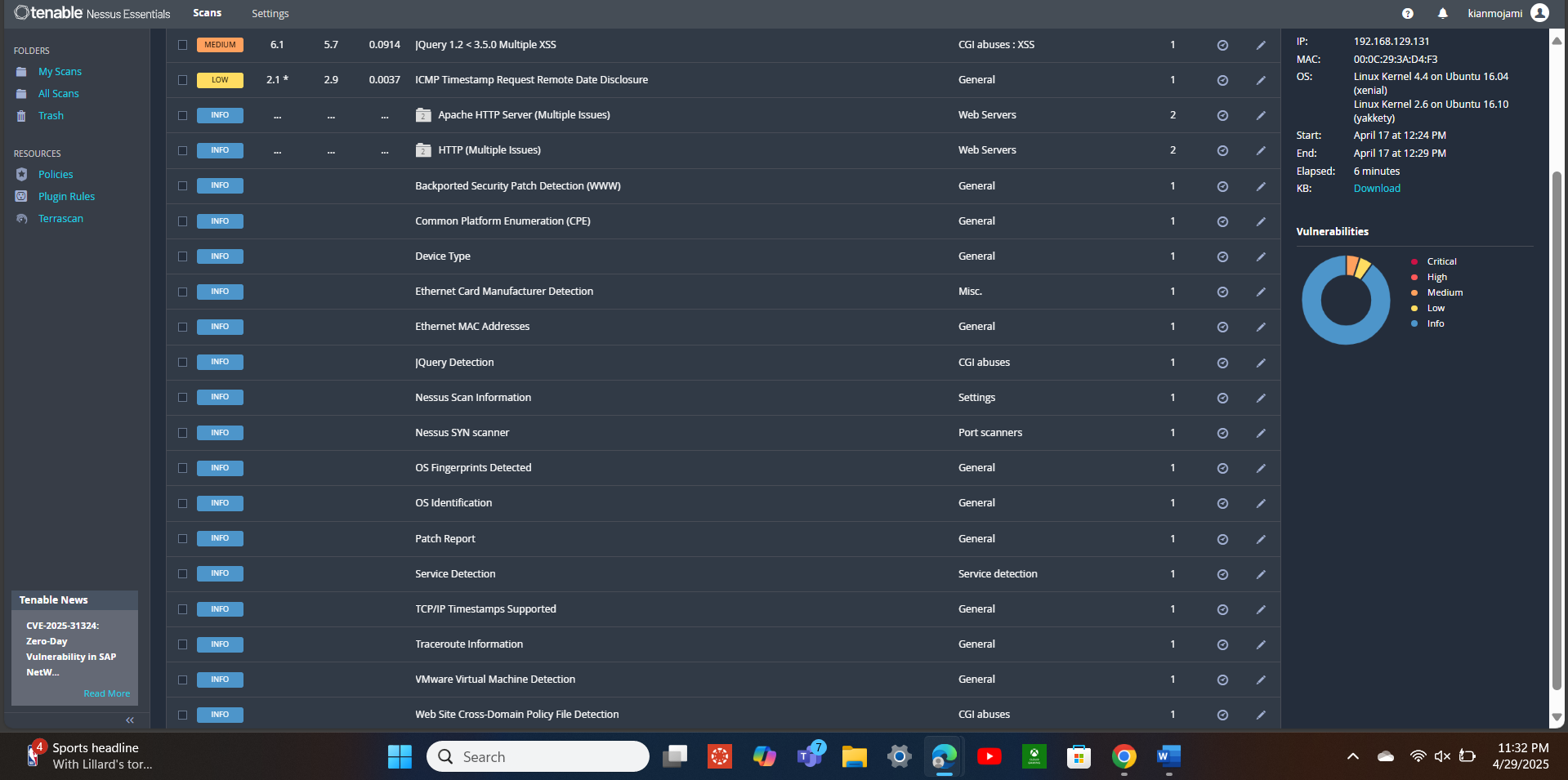


Figure 1: Nessus vulnerability results summary

## Methodology

This assessment was conducted using an automated vulnerability scanning method through Tenable Nessus, a widely known vulnerability scanner. The specific purpose of this scan is to find vulnerabilities, misconfigurations, and missing patches that could affect the security of the Hackazon web application. No penetration testing or exploitation attempts were conducted as the scope of this assessment was limited to non-intrusive identification of vulnerabilities on the web application.

The testing environment of the scan using the Tenable Nessus vulnerability scanner was conducted against Hackazon which is a publicly accessible web application. No systems that were not related to the web application were included in this scan. The scan was performed without authentication to simulate findings that are available to external users.

## Findings

The vulnerability assessment targets only the Hackazon web application as identified by the client. The application was fully accessible throughout the assessment process, which resulted in all its components being successfully scanned for vulnerabilities without issues.

No systems other than the Hackazon web application were included in the scan, and the system was fully accessible and online during the extraction of the scan which resulted in a fully complete assessment.

## Risk Assessment

The table below represents an index of all the vulnerabilities identified during the scan which are categorized by severity using Nessus’s risk levels: Critical, High, Medium, Low, and Informational.

Critical – Vulnerabilities that pose an immediate threat to the system and are highly exploitable.

High – Vulnerabilities that still pose a large threat and can be exploited to gain significant access.

Medium – Vulnerabilities that can have a moderate impact if exploited and should still be addressed.

Low – Vulnerabilities with a limited impact but can add to overall risk over time.

Informational – Not direct threats but can be used to provide insight into a system and should be made aware of.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Severity** | **Plugin Name** | **Description** | **Solution** | **Count** |
| Medium | jQuery 1.2 < 3.5.0 Multiple XSS | The web server is before 3.5 and after 1.2 and is affected by multiple cross-site scripting vulnerabilities. | Upgrade to jQuery version 3.5.0 or later. | 1 |
| Low | ICMP Timestamp Request Remote Date Disclosure | The remote host answers to an ICMP timestamp request. | Filter out the ICMP timestamp requests (13), and the outgoing ICMP timestamp replies (14). | 1 |
| Informational | Apache Banner Linux Distribution Disclosure | Nessus was able to extract the banner of the Apache web server and determine the Linux distribution. | Edit 'httpd.conf' and set the directive 'ServerTokens Prod' and restart Apache. | 1 |
| Informational | Apache HTTP Server Version | The remote host is running the Apache HTTP Server, and it is possible to see the version number in the banner. | Configure the web server to suppress version banners in HTTP responses. | 1 |
| Informational | HTTP Server Type and Version | Determines the type and the version of the remote web server. | Suppress or modify server response headers to minimize HTTP information disclosure. | 1 |
| Informational | Hypertext Transfer Protocol (HTTP) Information | This test gives some information about the remote HTTP protocol. | No action is required. This finding is informational and does not indicate a security issue. | 1 |
| Informational | Backported Security Patch Detection (WWW) | Security patches may have been 'backported' to the remote HTTP server without changing its version number. | No action is required. This finding is informational and does not indicate a security issue. | 1 |
| Informational | Device Type | Based on the remote operating system, it is possible to determine what the remote system type is. | Restrict unnecessary network access and limit OS fingerprinting by filtering inbound traffic and disabling unnecessary services. | 1 |
| Informational | Ethernet Card Manufacturer Detection | Each ethernet MAC address starts with a 24-bit Organizationally Unique Identifier (OUI) which can be identified and spoofed. | Limit exposure of MAC address information by segmenting networks and using firewalls to restrict unauthorized access to Layer 2 discovery protocols. | 1 |
| Informational | Ethernet MAC Addresses | MAC addresses were discovered from both remote probing of the host (e.g. SNMP and NetBIOS) and from running local checks (e.g. ifconfig). | Restrict access to protocols like SNMP and NetBIOS from untrusted networks and disable unnecessary services to reduce MAC address exposure. | 1 |
| Informational | jQuery Detection | jQuery was detected on the remote host. | Update jQuery to the newest version. | 1 |
| Informational | Nessus SYN scanner | A SYN 'half-open' port scanner was used. As a result, TCP port 80 was found to be open. | Protect targets with an IP filter. | 1 |
| Informational | OS Fingerprints Detected | Using a combination of remote probes, it was possible to gather one or more fingerprints from the remote system. | Restrict or filter unnecessary network services and configure firewall rules to reduce OS fingerprinting exposure. | 1 |
| Informational | Patch Report | The remote host is missing one or more security patches. | Install the missing patches: jQuery to version 3.5.0 or later. | 1 |
| Informational | Service Detection | The scanner was able to identify the remote service by its banner. | Configure services to limit banner disclosure or use banner obfuscation to reduce information disclosure. | 1 |
| Informational | TCP/IP Timestamps Supported | The remote host implements TCP timestamps, as defined by RFC1323 which can allow the uptime of the remote host to be computed. | Disable TCP timestamps at the OS level to prevent remote uptime enumeration. | 1 |
| Informational | Traceroute Information | Traceroute information can be viewed. | Restrict ICMP and traceroute-related traffic using firewall rules to limit network path exposure. | 1 |
| Informational | Web Site Cross-Domain Policy File Detection | The remote web server contains a cross-domain policy file. | Review the contents of the policy file carefully. | 1 |

The medium and low vulnerabilities pose legitimate risks that require action to be taken to minimize attack surfaces. The informational findings are not directly exploitable but can be used to reveal weaknesses and aid an attacker during exploitation.

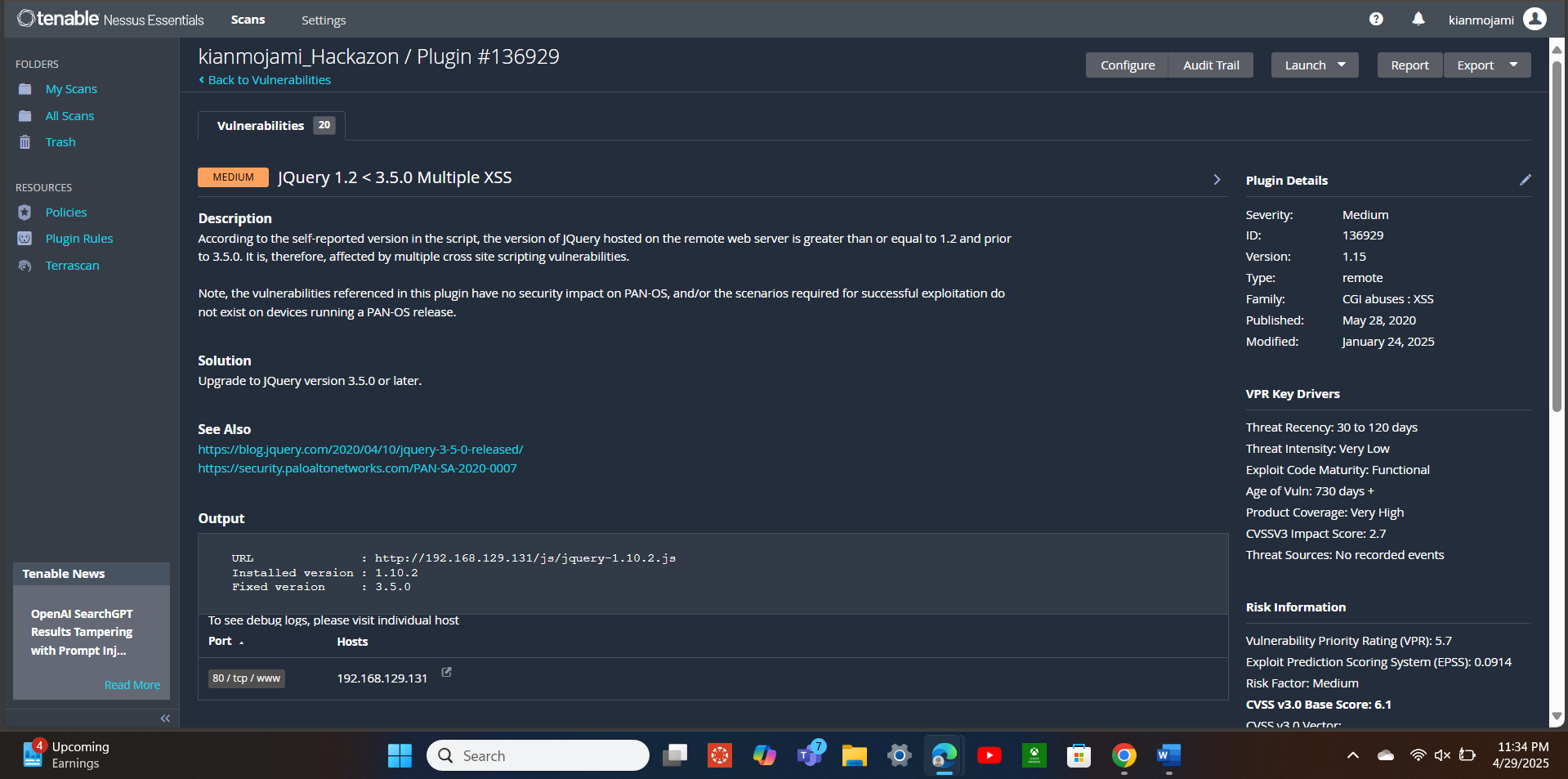


Figure 2: Medium severity vulnerability - jQuery 1.2 < 3.5.0 Multiple XSS

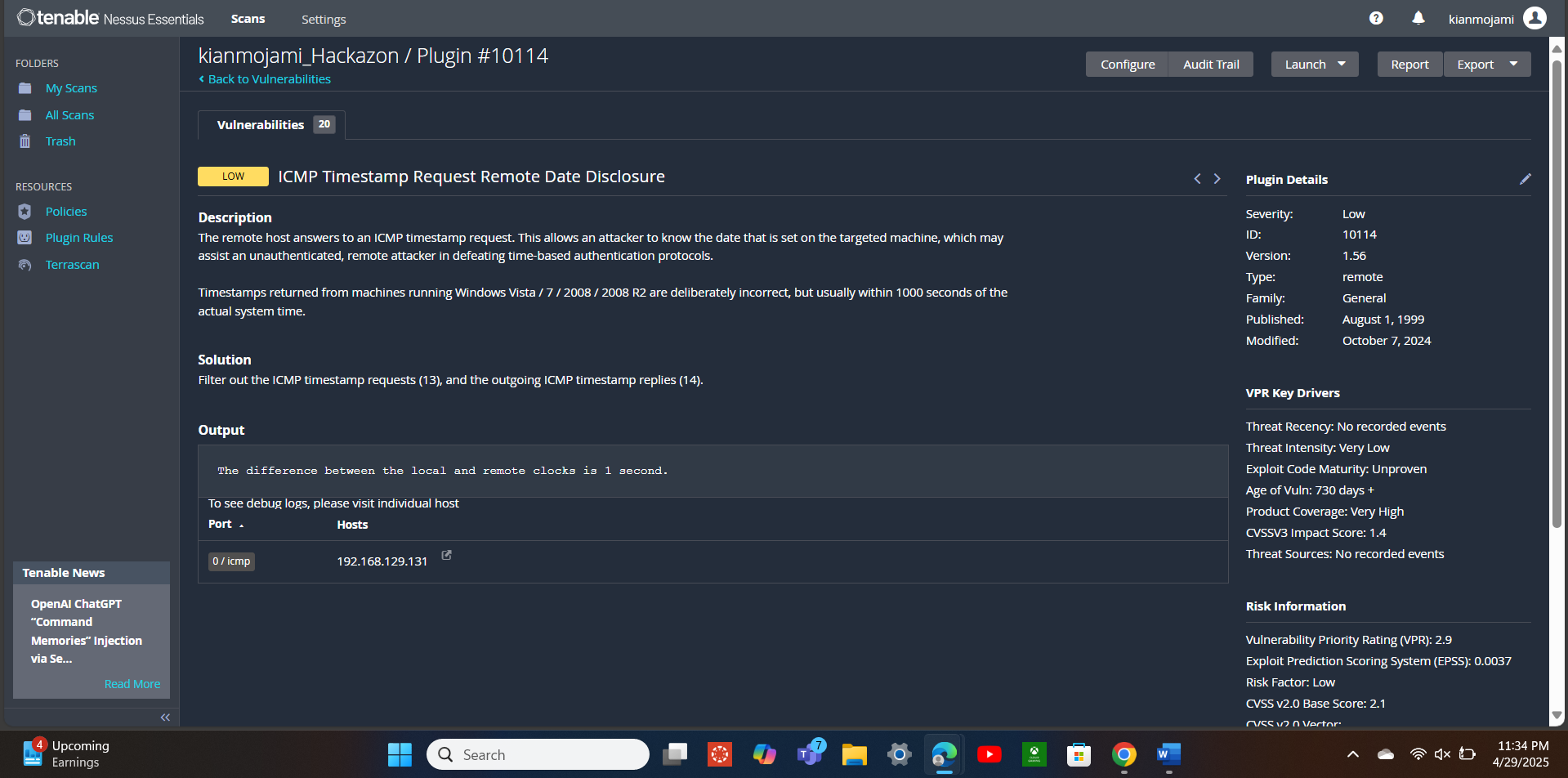


Figure 3: Low severity vulnerability - ICMP Timestamp Request Remote Date Disclosure

## Recommendations

Addressing the medium and low-severity vulnerabilities is a priority as they provide the most immediate threat to the application. To address jQuery being susceptible to cross-site scripting attacks because of its version being between 1.2 and 3.5.0, it should be updated to version 3.5.o or later. Doing this will allow the program to safely run on the application without making it vulnerable to cross-site scripting attacks. The other vulnerability that should be remediated immediately is the host answering ICMP timestamp requests. This vulnerability can allow an attacker to know the time and date that is set on the machine which can help against time-based authentication. Blocking ICMP type 13 requests and type 14 replies stops users from being able to request the time and date and stops the server from responding to these requests respectively.

Implementing hardening techniques to address the informational vulnerabilities that do not pose immediate threats but can provide useful information will mitigate their risks. Using an industry-standard such as CIS Benchmarks can provide a good baseline to harden devices. Disabling unnecessary ports and services, and applying a Deny by Default, Allow by Exception policy will stop unneeded and unwanted traffic from entering the network. Additionally, applying patches and updates regularly will ensure every program is up to date and doesn’t pose any security risk. Banner information disclosure should be limited to services as it can provide useful information to an attacker.

Not only is Nessus a great vulnerability scanner, but OpenVAS is a free, open-source scanner that, like Nessus, can be used to identify vulnerabilities and weaknesses in the system. Burp Suite is another tool that is suitable for web application testing to identify issues in application code like logic flaws and input validation. Wireshark is effective for inspecting traffic to the web application and detecting suspicious behavior in real time. Wazuh is another free, open-source threat detection platform that can provide host-based detection of suspicious behavior like rootkits on systems. Furthermore, Nmap can be utilized for identifying open ports and running services on the application which helps assess which ones are unnecessary for operating the web application.