

Vulnerability Audit and Assessment Results and Executive Summary

Executive Summary

Overview

The objective of conducting a vulnerability audit and assessment on the Zero Bank website (<http://zero.webappsecurity.com>) was to detect potential security weaknesses, evaluate their implications, and develop methods to reduce risks. This process is crucial for ensuring the safety of sensitive customer data, adhering to relevant security standards, and protecting against potential cyberattacks. Given the growing prevalence of cyber threats and the critical nature of financial data, maintaining a strong security posture is essential for a Zero Bank to safeguard its operations and maintain customer trust.

The assessment methodology involved a thorough examination of the Zero Bank website, using a combination of automated and manual testing methodologies. The approach was guided by established frameworks including the OWASP Testing Guide and NIST SP 800-115 (Kumar & Gandhi, 2023). Key activities included:

- Remote and local assessments
- Automated scanning for initial vulnerability identification
- Manual testing for detailed analysis and verification
- Evaluation against security standards such as GDPR and PCI DSS
(Machado et al., 2024; GDPR, 2016; PCI, 2012)

Methodology

Approach

The assessment was carried out in two main phases:

- **Automated Scanning:** Using industry-standard tools to perform an initial vulnerability scan. This phase aimed to quickly identify common vulnerabilities and establish a baseline for further manual analysis.
- **Manual Testing:** A thorough inspection by security specialists to confirm the results of the automated scans and uncover any new vulnerabilities that were not discovered by automated techniques.

Tools Used

- **OWASP ZAP:** for identifying vulnerabilities in web applications.
- **Burp Suite:** for performing security testing on web applications.
- **Nmap:** for network discovery and security auditing.
- **Nikto:** for scanning web servers.
- **OpenVAS:** for vulnerability assessment and management.
- **Nessus:** for comprehensive vulnerability scanning.

Techniques

- **Vulnerability Scanning:** Automated tools were used to scan the website for known vulnerabilities such as SQL Injection, XSS, and CSRF.

- **Penetration Testing:** Manual techniques were applied to exploit identified vulnerabilities and assess their impact.
- **Configuration Review:** Manual review of server and application configurations to identify security misconfigurations.
- **Compliance Checks:** Evaluation against GDPR and PCI DSS standards to ensure regulatory compliance.

Limitations Encountered

- **Limited Access to Internal Systems:** Access constraints hindered the ability to perform a comprehensive evaluation of some internal systems.
- **Potential for False Positives/Negatives:** Automated tools may not always accurately identify vulnerabilities, necessitating manual verification.
- **Assumptions:** The assessment assumed full cooperation from the IT team and access to all necessary resources and documentation.

Summary Findings

The audit identified several critical and high-priority vulnerabilities within the Zero Bank website. The main issues found included:

Table 1. Critical and high-priority vulnerabilities.

Vulnerability	Description	Risk level
SQL Injection	Detected in the login and search functionalities, allowing attackers to manipulate the database.	High
Cross-Site Scripting (XSS)	Multiple instances are found in user input fields, posing a risk of unauthorized script execution.	High
Cross-Site Request Forgery (CSRF)	Identified in the transaction processing module, which could allow attackers to perform actions on behalf of authenticated users.	Medium
Security Misconfiguration	Inadequate security settings in web servers and applications, exposing sensitive data.	Medium
Unencrypted Data Transmission	Detected in some data transmission processes, risking exposure of sensitive information.	High

Scanning Results

The automated scanning results provided an initial list of potential vulnerabilities, which were then verified and expanded upon through manual testing. Key findings from the automated scans included:

Table 2. Automated scanning summary.

Vulnerability	Instances detected	Severity
SQL Injection	5	High
Cross-Site Scripting (XSS)	12	High
Cross-Site Request Forgery (CSRF)	8	Medium
Security Misconfiguration	15	Medium
Unencrypted Data Transmission	4	High

Risk Assessment

Each detected vulnerability was evaluated for its potential effect and likelihood of exploitation. The risk assessment matrix is presented below:

Table 3. Risk assessment matrix.

Vulnerability	Impact	Likelihood	Risk level
SQL Injection	High	High	Critical
Cross-Site Scripting (XSS)	High	High	Critical
Cross-Site Request Forgery (CSRF)	Medium	Medium	High
Security Misconfiguration	Medium	High	High
Unencrypted Data Transmission	High	High	Critical

Findings

Main findings of vulnerability presented below:

Table 4. Vulnerabilities by host.

24	34	28	7	21
CRITICAL	HIGH	MEDIUM	LOW	INFO

Table 5. Apache Tomcat 7.0.0 < 7.0.72 Multiple Vulnerabilities.



Risk Level	URL	Evidence
 Critical	http://zero.webappsecurity.com	Plugin ID 197818: Apache Tomcat 7.0.0 < 7.0.72 Multiple Vulnerabilities: Response does not include the HTTP Content-Security-Policy security header or meta tag, which helps mitigate the risk of certain types of attacks, such as cross-site scripting (XSS) and data injection attacks. Adding this header can significantly enhance the security posture of the web application.
DETAILS Vulnerability description This plugin identifies multiple vulnerabilities in Apache Tomcat versions prior to 7.0.72. These vulnerabilities can include issues such as improper request handling, session fixation, and information disclosure. Specific vulnerabilities addressed in this version include CVE-2016-1240, CVE-2016-3092, and CVE-2016-0706. Risk description An attacker exploiting these vulnerabilities can potentially perform actions such as unauthorized access to sensitive information, session hijacking, or execution of arbitrary code. These actions could compromise the confidentiality, integrity, and availability of the server and its data. Recommendation Update Apache Tomcat to version 7.0.72 or later. Regularly monitor for and apply security patches to protect against new and existing vulnerabilities. Ensure secure configurations and follow best practices for server security.		

Table 6. Apache Tomcat 7.0.0 < 7.0.100 Multiple Vulnerabilities.

Risk Level	URL	Evidence
 Critical	http://zero.webappsecurity.com	Plugin ID 197843: Apache Tomcat 7.0.0 < 7.0.100 Multiple Vulnerabilities: Response does not include the HTTP Content-Security-Policy security header or meta tag, which is essential in preventing various types of attacks by specifying which resources can be loaded and executed on the webpage.

DETAILS
Vulnerability description <p>This plugin identifies multiple vulnerabilities in Apache Tomcat versions before 7.0.100. These vulnerabilities can include issues with <u>WebSocket</u> implementation, improper input validation, and insecure default configurations. Specific vulnerabilities addressed in this version include CVE-2018-1304, CVE-2018-1305, and CVE-2018-8037.</p>
Risk description <p>Exploiting these vulnerabilities could allow attackers to execute arbitrary code, disclose sensitive information, or cause a denial of service. The impact of these vulnerabilities can vary but generally pose significant risk to the server's security posture.</p>
Recommendation <p>Upgrade Apache Tomcat to version 7.0.100 or later. Implement a robust security update and monitoring process to quickly address any future vulnerabilities. Additionally, review and enhance server security configurations.</p>

Table 7. Apache Tomcat 7.0.0 < 7.0.73 Multiple Vulnerabilities.



Risk Level	URL	Evidence
 Critical	http://zero.webappsecurity.com	Plugin ID 197848: Apache Tomcat 7.0.0 < 7.0.73 Multiple Vulnerabilities: Response does not include the HTTP Content-Security-Policy security header or meta tag. This header helps protect the web application from certain types of attacks, including XSS and data injection attacks, by controlling the resources the browser is allowed to load for the page.
DETAILS		
Vulnerability description <p>This plugin highlights multiple vulnerabilities in Apache Tomcat versions prior to 7.0.73. The vulnerabilities may include flaws related to resource handling, input validation, and error processing. Specific vulnerabilities addressed in this version include CVE-2016-5018, CVE-2016-6794, and CVE-2016-6796.</p>		
Risk description <p>Attackers could exploit these vulnerabilities to perform unauthorized actions, such as accessing or modifying sensitive data, executing arbitrary commands, or disrupting services. The overall risk to the server's security and operational integrity is significant.</p>		
Recommendation <p>Ensure Apache Tomcat is updated to at least version 7.0.73. Continuously apply security updates and patches. Regularly review and secure server configurations to minimize vulnerabilities.</p>		

Table 8. Communication is not secure.

Risk Level	URL	Evidence
 Medium	http://zero.webappsecurity.com	Communication is made over unsecure, unencrypted HTTP.
DETAILS		
Vulnerability description		

Noticed that the communication between the web browser and the server is done using the HTTP protocol, which transmits data unencrypted over the network.

Risk description

The risk is that an attacker who manages to intercept the communication at the network level can read and modify the data transmitted (including passwords, secret tokens, credit card information and other sensitive data).

Recommendation

Recommended to reconfigure the web server to use HTTPS - which encrypts the communication between the web browser and the server.

Table 9. Vulnerabilities found for server-side software.

Risk Level	CVSS	CVE	Summary	Affected software
● Medium	4.3	CVE-2012-6708	jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.	jquery 1.8.2
● Medium	4.3	CVE-2015-9251	jQuery before 3.0.0 is vulnerable to Cross-site Scripting (XSS) attacks when a cross-domain Ajax request is performed without the dataType option, causing text/javascript responses to be executed.	jquery 1.8.2
● Medium	4.3	CVE-2019-11358	jQuery before 3.4.0, as used in Drupal,Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution. If an unsanitized source object contained an enumerable __proto__ property, it could extend the native Object.prototype.	jquery 1.8.2
● Medium	4.3	CVE-2020-11023	In jQuery versions greater than or equal to 1.0.3 and before 3.5.0, passing HTML containing <option> elements from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	jquery 1.8.2
● Medium	4.3	CVE-2020-11022	In jQuery versions greater than or equal to 1.2 and before 3.5.0, passing HTML from untrusted sources - even after sanitizing it - to one of jQuery's DOM manipulation methods (i.e. .html(), .append(), and others) may execute untrusted code. This problem is patched in jQuery 3.5.0.	jquery 1.8.2

<p>DETAILS</p> <p>Vulnerability description</p> <p>Noticed known vulnerabilities in the target application. They are usually related to outdated systems and expose the affected applications to the risk of unauthorized access to confidential data and possibly denial of service attacks.</p> <p>Risk description</p> <p>The risk is that an attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.</p> <p>Recommendation</p> <p>Recommended to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.</p>

Table 10. Missing security header: Referrer-Policy.

Risk Level	URL	Evidence
● Low	http://zero.webappsecurity.com	Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with name 'referrer' is not present in the response.
<p>DETAILS</p> <p>Vulnerability description</p> <p>Noticed that the target application's server responses lack the Referrer-Policy HTTP header, which controls how much referrer information the browser will send with each request originated from the current web application.</p> <p>Risk description</p> <p>The risk is that if a user visits a web page (e.g. "http://example.com/pricing/") and clicks on a link from that page going to e.g. "https://www.google.com", the browser will send to Google the full originating URL in the Referer header, assuming the Referrer-Policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.</p> <p>Recommendation</p> <p>The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value no-referrer of this header instructs the browser to omit the Referer header entirely.</p>		

Table 11. Missing security header: Content-Security-Policy.

Risk Level	URL	Evidence
● Low	http://zero.webappsecurity.com	Response does not include the HTTP Content-Security-Policy security header or meta tag.
<p>DETAILS</p> <p>Vulnerability description</p> <p>Noticed that the target application lacks the Content-Security-Policy (CSP) header in its HTTP responses. The CSP header is a security measure that instructs web browsers to enforce specific security rules, effectively preventing the exploitation of Cross-Site Scripting (XSS) vulnerabilities.</p> <p>Risk description</p>		

The risk is that if the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

Recommendation

Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

Table 12. Server software and technology found.

Risk Level	Software / Version	Category
● Low	Apache Tomcat	Web servers
● Low	Bootstrap	UI frameworks
● Low	Java	Programming languages
● Low	jQuery 1.8.2	JavaScript libraries
● Low	Font Awesome	Font scripts
DETAILS		
Vulnerability description		
Noticed that server software and technology details are exposed, potentially aiding attackers in tailoring specific exploits against identified systems and versions.		
Risk description		
The risk is that an attacker could use this information to mount specific attacks against the identified software type and version.		
Recommendation		
Recommended to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.		

Table 13. HTTP OPTIONS enabled.

Risk Level	URL	Method	Summary
● Info	http://zero.webappsecurity.com	OPTIONS	We did a HTTP OPTIONS request. The server responded with a 200 status code and the header: Allow: GET, HEAD, POST, PUT, DELETE, TRACE, OPTIONS, PATCH
DETAILS			
Vulnerability description			
We have noticed that the webserver responded with an Allow HTTP header when an OPTIONS HTTP request was sent. This method responds to requests by providing information about the methods available for the target resource.			
Risk description			
The only risk this might present nowadays is revealing debug HTTP methods that can be used on the server. This can present a danger if any of those methods can lead to sensitive information, like authentication information, secret keys.			

Recommendation

We recommend that you check for unused HTTP methods or even better, disable the OPTIONS method. This can be done using your webserver configuration.

Table 14. Security.txt file is missing.

Risk Level	URL
● Info	http://zero.webappsecurity.com
DETAILS	
Vulnerability description	
Noticed that the server is missing the security.txt file, which is considered a good practice for web security. It provides a standardized way for security researchers and the public to report security vulnerabilities or concerns by outlining the preferred method of contact and reporting procedures.	
Risk description	
There is no particular risk in not having a security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.	
Recommendation	
Recommended to implement the security.txt file according to the standard, in order to allow researchers or users report any security issues they find, improving the defensive mechanisms of your server.	

Table 15. Other informational findings.

Risk Level	Finding
● Info	Website is accessible.
● Info	Nothing was found for client access policies.
● Info	Nothing was found for robots.txt file.
● Info	Nothing was found for use of untrusted certificates.
● Info	Nothing was found for enabled HTTP debug methods.
● Info	Nothing was found for directory listing.
● Info	Nothing was found for missing HTTP header - Strict-Transport-Security.
● Info	Nothing was found for domain too loose set for cookies.
● Info	Nothing was found for HttpOnly flag of cookie.
● Info	Nothing was found for Secure flag of cookie.
● Info	Nothing was found for unsafe HTTP header Content Security Policy.

Evaluation Against Security Standards

GDPR Compliance

The General Data Protection Regulation (GDPR) requires rigorous measures to protect the personal data of EU residents. Data encryption, access limits, and regular security audits are among the key GDPR obligations. Our evaluation revealed that the Zero Bank website partially complies with GDPR but needs significant improvements (Machado et al., 2024).

Table 16. Communication is not secure.

GDPR requirement	Current Status	Compliance level
Data encryption	Some data transmission processes found unencrypted.	Partial
Access controls	Basic access controls in place, but no Multi-functional authentication.	Partial
Regular assessments	Need for more frequent security assessments.	Inadequate
Cookie banner	Cookie banner is not displayed on website. Displaying a cookie banner on users' first visit is required by laws like GDPR and the EU Cookie Law. The banner should have a "Reject" and "Accept" buttons.	Inadequate
Privacy controls and Cookie	It was not gathered enough information to determine whether or not you are providing your users with the option to reopen the cookie banner and change their cookie preferences.	Inadequate
Consent records	It was not gathered enough information to determine whether or not you are keeping a record of your users' consent preferences.	Inadequate
Transparency and consent framework	If Zero Bank has users from the EU and UK, bank may want to activate the IAB Transparency and Consent Framework to secure your ad revenue and enhance transparency for your end users.	Inadequate
Cookie policy	It seems Zero Bank is correctly displaying the type, storage duration, and purposes for which cookies are installed. However, Cookie policy displayed in other domain.	Partial
Privacy policy	It seems Zero Bank is currently providing users with an up-to-date privacy policy. However, Privacy policy displayed in other domain.	Partial
Terms & Conditions document	It was not gathered enough information to determine whether or not Zero Bank is providing a link to terms and condition.	Inadequate

European ODR platform	Zero Bank is missing a direct link to the European Online Dispute Resolution (ODR) platform.	Inadequate
-----------------------	----------------------------------------------------------------------------------------------	------------

PCI DSS Compliance

The Payment Card Industry Data Security Standard (PCI DSS) specifies security procedures for securing cardholder data (PCI 2012). The key requirements include:

Table 17. Communication is not secure.

PCI DSS requirement	Current Status	Compliance level
Secure network	Robust firewalls and secure network configurations.	Inadequate
Protection of cardholder data	Inadequate encryption of data transmission and storage.	Partial
Access controls	Basic controls need enhanced measures.	Partial

Conclusions

The vulnerability audit of the Zero Bank website revealed critical security weaknesses requiring immediate attention to protect customer data and ensure compliance. The primary conclusions are:

- **Immediate Action Needed:** Critical vulnerabilities such as SQL Injection and Cross-Site Scripting (XSS) need urgent remediation to prevent data breaches and unauthorized transactions.
- **Partial Compliance:** The website shows partial compliance with GDPR and PCI DSS standards, necessitating enhanced encryption and access controls for full compliance.
- **Ongoing Assessments:** Regular vulnerability scans, penetration testing, and configuration reviews are essential to keep pace with evolving cyber threats (Ewuga et al., 2024).

- **Policy and Training Enhancements:** Security policies must be updated, and regular employee training is crucial to mitigate risks associated with human error.
- **Improved Data Protection:** Encrypting data at rest and in transit enhances consumer information security.
- **Stronger Access Controls:** Using multi-factor authentication (MFA) and role-based access control (RBAC) can reduce unauthorised access.
- **Compliance Updates:** Regular policy reviews and modifications are required to be compliance with changing standards and requirements.

In summary, addressing critical vulnerabilities, achieving full compliance, conducting ongoing assessments, and enhancing data protection and access controls will strengthen Zero Bank's security. These steps are vital for protecting operations, safeguarding customer data, and maintaining trust in the digital financial landscape.

Recommendations

Based on the findings and conclusions, the following recommendations are proposed, prioritized by business needs and risk levels:

1. Mitigate critical vulnerabilities

Action	Description	Priority
SQL Injection	Implement parameterized queries and input validation.	High
Cross-Site Scripting (XSS)	Sanitize all user inputs and utilize Content Security Policy (CSP).	High
Cross-Site Request Forgery (CSRF)	Implement CSRF tokens in forms.	High

To prevent SQL injection, parameterized queries will regard inputs as data rather than executable commands, preserving the database from manipulation (Dawodu et al.,

2023). For XSS, sanitizing user inputs and employing CSP will prevent unauthorized script execution by ensuring that only trusted scripts run on the webpage. Implementing CSRF tokens will protect against CSRF attacks by ensuring that every form submission is legitimate and comes from an authenticated user session.

2. Enhance Data Protection Measures

Action	Description	Priority
Data encryption	Ensure all data transmissions are encrypted using TLS.	High
Secure storage	Encrypt sensitive data at rest.	High

Ensuring data encryption for all transmissions using TLS will protect sensitive information from being intercepted during transit. Encrypting data at rest will add another layer of protection, ensuring that sensitive information remains secure even if unauthorized access is gained to the storage system (Machado et al., 2024).

3. Strengthen Access Controls

Action	Description	Priority
Multi-factor authentication (MFA)	Implement MFA to add an extra layer of security for user authentication.	High
Role-based access control (RBAC)	Define and enforce RBAC to limit access based on user roles and responsibilities.	Medium

Implementing MFA will force users to give two or more verification factors in order to obtain access, considerably increasing security. RBAC ensures that users only have access to the information and resources required for their responsibilities, reducing the risk of unauthorised access (Ahmad et al., 2024).

4. Regular Security Assessments and Updates

Action	Description	Priority
Regular vulnerability scans	Conduct monthly vulnerability scans to identify and mitigate new threats.	Medium
Penetration testing	Perform annual penetration tests to evaluate the effectiveness of security measures.	Medium

Regular vulnerability scans will help identify and address new security threats promptly. Annual penetration testing will provide a comprehensive assessment of the website's security posture and the effectiveness of implemented security measures (Kumar & Gandhi, 2023).

5. Compliance and Policy Updates

Action	Description	Priority
Policy updates	Update security policies and procedures to align with GDPR and PCI DSS requirements.	Medium
Employee training	Provide regular training to employees on security best practices and compliance requirements.	Medium

Updating security policies and processes to comply with GDPR and PCI DSS ensures that the website meets current regulatory and industry standards. Employees should get regular training on security best practices to assist avoid human error and develop a security-conscious culture inside the organisation (Oyewole et al., 2024).

Summary of Limitations and Assumptions

Limitations

- **Limited Access to Internal Systems:** Access constraints hindered the ability to perform a comprehensive evaluation of some internal systems.

- **Potential for False Positives/Negatives:** Automated tools may not always accurately identify vulnerabilities, necessitating manual verification.

Assumptions

- **Full Cooperation from the IT Team:** The effectiveness of the assessment relied on the assumption that the IT team would provide the necessary support and access.
- **Access to Necessary Resources and Documentation:** It was assumed that all relevant resources and documentation would be available to conduct a thorough assessment.

Reference list

- Ahmad, A., Chaturvedi, S., Im, A., et al. (2024). "Cybersecurity in Online Banking Applications: A Comprehensive Review." *Journal of Information Security*.
- Dawodu, J., Shaji, M. (2023). "Emerging Threats in Online Banking: An Analytical Perspective." *Cybersecurity Journal*.
- GDPR (2016) Regulation - 2016/679 - en - GDPR - EUR-lex, EUR. Available at: <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=celex%3A32016R0679> (Accessed: 22 July 2024).
- Ewuga, J., Petranović, T., Žarić, R. (2024). "Information Security Management Systems: Compliance and Implementation." *Information Security Journal*.
- Kumar, R., Gandhi, S. (2023). "OWASP Testing Guide: Best Practices for Web Application Security Testing." *Cybersecurity Research Journal*.
- Machado, L., Singh, P., et al. (2024). "GDPR Compliance in Financial Institutions: Challenges and Strategies." *Journal of Data Protection and Privacy*.
- PCI Security Standards Council. (2012). "Payment Card Industry Data Security Standard: Requirements and Security Assessment Procedures."
- Oyewole, A., Dawodu, J., et al. (2024). "Cybersecurity in the Banking Sector: Trends and Best Practices." *Journal of Financial Technology*.

Appendix A - Vulnerabilities by Host

Severity	CVSS v3.0	VPR Score	Plugin	Name
CRITICAL	9.8	6.7	57603	Apache 2.2.x < 2.2.13 APR apr_palloc Heap Overflow
CRITICAL	9.8	9.0	45004	Apache 2.2.x < 2.2.15 Multiple Vulnerabilities
CRITICAL	9.8	6.7	100995	Apache 2.2.x < 2.2.33-dev / 2.4.x < 2.4.26 Multiple Vulnerabilities
CRITICAL	9.8	6.7	101787	Apache 2.2.x < 2.2.34 Multiple Vulnerabilities
CRITICAL	9.8	6.7	158900	Apache 2.4.x < 2.4.53 Multiple Vulnerabilities
CRITICAL	9.8	5.9	193421	Apache 2.4.x < 2.4.54 Authentication Bypass
CRITICAL	9.8	6.7	172186	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities
CRITICAL	9.8	6.7	153584	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	9.8	9.0	197843	Apache Tomcat 7.0.0 < 7.0.100 multiple vulnerabilities
CRITICAL	9.8	7.4	197848	Apache Tomcat 7.0.0 < 7.0.73 multiple vulnerabilities
CRITICAL	9.8	6.7	111066	Apache Tomcat 7.0.0 < 7.0.89
CRITICAL	9.8	5.9	17760	OpenSSL 0.9.8 < 0.9.8f Multiple Vulnerabilities
CRITICAL	9.8	6.7	45039	OpenSSL 0.9.8 < 0.9.8m Multiple Vulnerabilities
CRITICAL	9.8	5.9	200190	OpenSSL 0.9.8 < 0.9.8p Vulnerability
CRITICAL	9.8	5.9	57459	OpenSSL 0.9.8 < 0.9.8s Multiple Vulnerabilities
CRITICAL	9.8	6.7	58799	OpenSSL 0.9.8 < 0.9.8v Vulnerability
CRITICAL	9.8	3.6	78552	OpenSSL 0.9.8 < 0.9.8zc Multiple Vulnerabilities
CRITICAL	9.1	5.2	161948	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
CRITICAL	9.1	6.0	197818	Apache Tomcat 7.0.0 < 7.0.72 multiple vulnerabilities
CRITICAL	9.1	5.2	121120	Apache Tomcat 7.0.0 < 7.0.76
CRITICAL	9.0	6.5	170113	Apache 2.4.x < 2.4.55 Multiple Vulnerabilities
CRITICAL	9.0	8.1	153583	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	10.0	-	171356	Apache HTTP Server SEoL (2.1.x <= x <= 2.2.x)
CRITICAL	10.0	-	171351	Apache Tomcat SEoL (7.0.x)
HIGH	8.2	6.4	40467	Apache 2.2.x < 2.2.12 Multiple Vulnerabilities
HIGH	8.1	9.0	103329	Apache Tomcat 7.0.0 < 7.0.81 multiple vulnerabilities
HIGH	8.1	9.2	103782	Apache Tomcat 7.0.0 < 7.0.82
HIGH	8.1	8.4	124064	Apache Tomcat 7.0.0 < 7.0.94 multiple vulnerabilities
HIGH	7.5	3.6	193422	Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability
HIGH	7.5	3.6	193423	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
HIGH	7.5	3.6	193424	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)
HIGH	7.5	4.4	183391	Apache 2.4.x < 2.4.58 Multiple Vulnerabilities
HIGH	7.5	4.4	193419	Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)

HIGH	7.5	5.2	192923	Apache 2.4.x < 2.4.59 Multiple Vulnerabilities
HIGH	7.5	6.1	201532	Apache 2.4.x < 2.4.61
HIGH	7.5	3.6	197823	Apache Tomcat 7.0.0 < 7.0.75
HIGH	7.5	3.6	197820	Apache Tomcat 7.0.0 < 7.0.77
HIGH	7.5	4.4	197831	Apache Tomcat 7.0.0 < 7.0.78
HIGH	7.5	6.7	197838	Apache Tomcat 7.0.0 < 7.0.99 multiple vulnerabilities
HIGH	7.5	4.4	197826	Apache Tomcat 7.0.25 < 7.0.90
HIGH	7.5	3.6	138851	Apache Tomcat 7.0.27 < 7.0.105
HIGH	7.5	3.6	121121	Apache Tomcat 7.0.28 < 7.0.88
HIGH	7.5	5.1	17761	OpenSSL 0.9.8 < 0.9.8i Vulnerability
HIGH	7.5	3.6	17763	OpenSSL 0.9.8 < 0.9.8k Multiple Vulnerabilities
HIGH	7.5	4.4	45359	OpenSSL 0.9.8 < 0.9.8n Multiple Vulnerabilities
HIGH	7.5	3.6	200204	OpenSSL 0.9.8 < 0.9.8q Vulnerability
HIGH	7.5	3.6	58564	OpenSSL 0.9.8 < 0.9.8u Vulnerability
HIGH	7.5	3.6	59076	OpenSSL 0.9.8 < 0.9.8x Vulnerability
HIGH	7.5	3.6	64532	OpenSSL 0.9.8 < 0.9.8y Multiple Vulnerabilities
HIGH	7.5	3.6	77086	OpenSSL 0.9.8 < 0.9.8zb Multiple Vulnerabilities
HIGH	7.5	5.9	84151	OpenSSL 0.9.8 < 0.9.8zg Multiple Vulnerabilities
HIGH	7.4	7.7	74363	OpenSSL 0.9.8 < 0.9.8za Multiple Vulnerabilities
HIGH	7.3	6.7	42052	Apache 2.2.x < 2.2.14 Multiple Vulnerabilities
HIGH	7.3	6.7	77531	Apache 2.2.x < 2.2.28 Multiple Vulnerabilities
HIGH	7.0	5.9	62101	Apache 2.2.x < 2.2.23 Multiple Vulnerabilities
HIGH	7.0	6.7	136770	Apache Tomcat 7.0.0 < 7.0.104
HIGH	7.0	6.7	147163	Apache Tomcat 7.0.0 < 7.0.108 multiple vulnerabilities
HIGH	7.6*	5.9	17766	OpenSSL < 0.9.8p / 1.0.0b Buffer Overflow
MEDIUM	6.5	3.3	17696	Apache HTTP Server 403 Error Page UTF-7 Encoded XSS
MEDIUM	6.5	4.4	106975	Apache Tomcat 7.0.0 < 7.0.85 multiple vulnerabilities
MEDIUM	6.5	-	142960	HSTS Missing From HTTPS Server (RFC 6797)
MEDIUM	6.1	5.7	136929	jQuery 1.2 < 3.5.0 Multiple XSS
MEDIUM	6.1	3.0	17762	OpenSSL 0.9.8 < 0.9.8j Vulnerability
MEDIUM	5.9	3.6	148405	Apache Tomcat 7.0.0 < 7.0.107
MEDIUM	5.9	5.2	82030	OpenSSL 0.9.8 < 0.9.8zf Multiple Vulnerabilities
MEDIUM	5.6	3.4	68915	Apache 2.2.x < 2.2.25 Multiple Vulnerabilities
MEDIUM	5.3	3.6	48205	Apache 2.2.x < 2.2.16 Multiple Vulnerabilities

MEDIUM	5.3	4.4	50070	Apache 2.2.x < 2.2.17 Multiple Vulnerabilities
MEDIUM	5.3	2.2	53896	Apache 2.2.x < 2.2.18 APR apr_fnmatch DoS
MEDIUM	5.3	2.2	56216	Apache 2.2.x < 2.2.21 mod_proxy_ajp DoS
MEDIUM	5.3	6.6	57791	Apache 2.2.x < 2.2.22 Multiple Vulnerabilities
MEDIUM	5.3	3.0	64912	Apache 2.2.x < 2.2.24 Multiple XSS Vulnerabilities
MEDIUM	5.3	1.4	73405	Apache 2.2.x < 2.2.27 Multiple Vulnerabilities
MEDIUM	5.3	3.8	31118	Apache 2.2.x < 2.2.8 Multiple Vulnerabilities (XSS, DoS)
MEDIUM	5.3	4.2	33477	Apache 2.2.x < 2.2.9 Multiple Vulnerabilities (DoS, XSS)
MEDIUM	5.3	1.4	193420	Apache 2.4.x < 2.4.54 Out-Of-Bounds Read (CVE-2022-28330)
MEDIUM	5.3	1.4	88098	Apache Server ETag Header Information Disclosure
MEDIUM	5.3	-	12085	Apache Tomcat Default Files
MEDIUM	5.3	4.0	11213	HTTP TRACE / TRACK Methods Allowed
MEDIUM	5.3	4.2	80566	OpenSSL 0.9.8 < 0.9.8zd Multiple Vulnerabilities
MEDIUM	5.3	2.2	87219	OpenSSL 0.9.8 < 0.9.8zh Vulnerability
MEDIUM	4.3	2.2	118035	Apache Tomcat 7.0.23 < 7.0.91
MEDIUM	4.3	1.4	102587	Apache Tomcat 7.0.41 < 7.0.79
MEDIUM	5.1*	5.9	17765	OpenSSL < 0.9.8l Multiple Vulnerabilities
MEDIUM	4.3*	4.2	17767	OpenSSL < 0.9.8p / 1.0.0e Double Free Vulnerability
MEDIUM	4.3*	-	85582	Web Application Potentially Vulnerable to Clickjacking
LOW	2.6*	-	34850	Web Server Uses Basic Authentication Without HTTPS
INFO	N/A	-	48204	Apache HTTP Server Version
INFO	N/A	-	39446	Apache Tomcat Detection
INFO	N/A	-	47830	CGI Generic Injectable Parameter
INFO	N/A	-	33817	CGI Generic Tests Load Estimation (all tests)
INFO	N/A	-	49704	External URLs
INFO	N/A	-	84502	HSTS Missing From HTTPS Server
INFO	N/A	-	43111	HTTP Methods Allowed (per directory)
INFO	N/A	-	10107	HTTP Server Type and Version
INFO	N/A	-	24260	HyperText Transfer Protocol (HTTP) Information
INFO	N/A	-	106658	jQuery Detection
INFO	N/A	-	50344	Missing or Permissive Content-Security-Policy frame-ancestors HTTP Response Header
INFO	N/A	-	50345	Missing or Permissive X-Frame-Options HTTP Response Header
INFO	N/A	-	11219	Nessus SYN scanner
INFO	N/A	-	19506	Nessus Scan Information

INFO	N/A	-	57323	OpenSSL Version Detection
INFO	N/A	-	66334	Patch Report
INFO	N/A	-	40665	Protected Web Page Detection
INFO	N/A	-	40773	Web Application Potentially Sensitive CGI Parameter Detection
INFO	N/A	-	91815	Web Application Sitemap
INFO	N/A	-	11032	Web Server Directory Enumeration
INFO	N/A	-	10662	Web mirroring