Cyberscope

Penetration Test Report

Snork

May 2024

Audited by © cyberscope

Table of Contents

Review

Domain	https://snork.io
Assessment Scope	Landing Page
Initial Report	22 May 2024

Overview

Cyberscope has conducted a comprehensive penetration test on the web application “Snork” hosted at https://snork.io. This report focuses on evaluating the security and performance aspects of the web application. The assessment encompasses various facets of the application, including but not limited to authentication and authorization mechanisms, data handling and storage practices, network security measures, and response to high traffic volumes.

The expansion of blockchain technology has introduced a myriad of innovative applications, each with its own unique security challenges. Snork, as a prime example within the realm of digital currency ecosystems, ensures robust protection of user data and system integrity.

Penetration Assessment Scope

The scope of this assessment extends to identifying vulnerabilities and weaknesses in the application's architecture and functionality, with the aim of providing actionable recommendations to enhance its security posture. The report aims to offer a comprehensive understanding of the application's strengths and areas for improvement, facilitating informed decision-making to mitigate risks, fortify against potential cyber threats, and bolster overall security resilience.

Web Technologies

Technology	Category	Version
Cloudflare Browser Insights	Analytics	N/A
React	JavaScript frameworks	N/A
React Router	JavaScript frameworks	6
HSTS	Security	N/A
Cloudflare Turnstile	Security	N/A
Google Font API	Font scripts	N/A
PWA	Miscellaneous	N/A
LottieFiles	Miscellaneous	N/A
HTTP/3	Miscellaneous	N/A
Cloudflare	CDN	N/A

Findings Breakdown

Critical	1
Medium	2
Minor / Informative	6

Severity	Unresolved	Acknowledged	Resolved	Other
Critical	1	0	0	0
Medium	2	0	0	0
Minor / Informative	6	0	0	0

Diagnostics

Critical Medium Minor / Informative

Severity	Code	Description	Status
●	PSIV	Potential SQL Injection Vulnerability	Unresolved
●	MACH	Missing Anti-Clickjacking Header	Unresolved
●	MCSPH	Missing Content Security Policy (CSP) Header	Unresolved
●	BPC	Best Practices Compliance	Unresolved
●	CSAV	Cookie SameSite Attribute Vulnerability	Unresolved
●	LTC	Latency And Throughput Challenges	Unresolved
●	MSTSH	Missing Strict Transport Security Header	Unresolved
●	MXH	Missing X-Content-Type-Options Header	Unresolved
●	SIUL	Server Instability Under Load	Unresolved

PSIV - Potential SQL Injection Vulnerability

Criticality	Critical
Status	Unresolved

Description

An SQL injection attack consists of insertion or “injection” of an SQL query via the input data from the client to the application. A successful SQL injection exploit can read sensitive data from the database, modify database data, execute administration operations on the database, recover the content of a given file present on the DBMS file system, and in some cases issue commands to the operating system. SQL injection attacks are a type of injection attack, in which SQL commands are injected into data-plane input in order to affect the execution of predefined SQL commands.

The application could potentially be susceptible to a time-based SQL injection attack. This vulnerability allows an attacker to manipulate query execution time by controlling input parameters. For instance, using a crafted parameter value such as `[3929 and exists (SELECT UTL_INADDR.get_host_name('10.0.0.1') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.2') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.3') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.4') from dual union SELECT UTL_INADDR.get_host_name('10.0.0.5') from dual) --]` resulted in a query execution time of 60,045 milliseconds compared to 21 milliseconds for the original query with value [3929].

The following URLs demonstrated this vulnerability:

1. https://snork.io/cdn-cgi/rum

Recommendation

To mitigate the SQL injection vulnerabilities:

- Implement server-side validation and sanitization of all user-supplied input to prevent injection attacks.
- Utilize parameterized queries (e.g., PreparedStatement or CallableStatement in JDBC) to ensure that user input is treated as data rather than executable code.
- Refrain from dynamically constructing SQL queries using string concatenation, as this can inadvertently introduce injection vulnerabilities.
- Assign the least privileged database user role necessary for the application to limit the potential impact of successful injection attacks.
- Enforce strict input validation by implementing allowlists or denylists of permissible characters to mitigate injection risks.

By implementing these recommendations, the application can significantly reduce the risk of SQL injection attacks and enhance its overall security posture. For more detailed guidance on preventing SQL injection vulnerabilities, refer to the OWASP SQL Injection Prevention Cheat Sheet.

MACH - Missing Anti-Clickjacking Header

Criticality	Medium
Status	Unresolved

Description

The absence of an Anti-Clickjacking header exposes the application to potential Clickjacking attacks. Clickjacking is a malicious technique that tricks users into clicking on unintended elements by disguising them as legitimate UI elements. This can lead to unauthorized actions being performed without the user's knowledge or consent. Without proper protection mechanisms in place, attackers can exploit Clickjacking vulnerabilities to perform actions on behalf of users, such as making purchases, changing account settings, or clicking on malicious links.

The response from the following URLs does not include either Content-Security-Policy with 'frame-ancestors' directive or X-Frame-Options to protect against 'ClickJacking' attacks.

1. https://snork.io/
2. https://snork.io/cdn-cgi/challenge-platform/h/b/flow/ov1/1073264910:1716361789:0EGuqSzrLNH046ua5TIa3QkF9kd3fJxxRJ8c7s5mpFw/887b2a4f0feb6f6d/60b639041849c8c

Recommendation

Modern Web browsers support the Content-Security-Policy and X-Frame-Options HTTP headers. The team is advised to ensure one of them is set on all web pages returned by the site/app.

If the team expects the page to be framed only by pages on their server (e.g. it's part of a FRAMESET) then they'll want to use SAMEORIGIN, otherwise, if the team never expects the page to be framed, they should use DENY. Alternatively, the team could consider implementing Content Security Policy's "frame-ancestors" directive.

Reference: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options

MCSPH - Missing Content Security Policy (CSP) Header

Criticality	Medium
Status	Unresolved

Description

Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images, and embeddable objects such as Java applets, ActiveX, audio and video files.

The following URLs are a sample of all the occurrences where a Content Security Policy (CSP) header was not set.

1. https://snork.io/
2. https://snork.io/cdn-cgi/challenge-platform/h/b/flow/ov1/1073264910:1716361789:0EGuqSzrLNH046ua5Tla3QkF9kd3fJxxRJ8c7s5mpFw/887b2a4f0feb6f6d/60b639041849c8c
3. https://snork.io/cdn-cgi/challenge-platform/h/b/flow/ov1/1073264910:1716361789:0EGuqSzrLNH046ua5Tla3QkF9kd3fJxxRJ8c7s5mpFw/887b37e45b846f53/530ff22c1d1405b

Recommendation

To address the absence of Content Security Policy (CSP) headers and enhance the security of the application, the following steps are recommended:

- Verify that your web server, application server, load balancer, or any other relevant components are properly configured to set the Content-Security-Policy header in HTTP responses.
- Define a comprehensive CSP policy tailored to the specific requirements and functionalities of your application. Consider including directives such as default-src, script-src, style-src, img-src, font-src, connect-src, frame-src, media-src, object-src, and sandbox, among others, to restrict content loading from unauthorized sources.
- Utilize CSP reporting mechanisms to monitor policy violations and fine-tune your CSP directives over time based on real-world usage and detected issues.

By implementing a robust Content Security Policy (CSP) and adhering to best practices for CSP configuration and management, you can significantly reduce the risk of XSS attacks, data injection vulnerabilities, and other web security threats, thereby enhancing the overall security posture of your application.

References:

1. Mozilla Developer Network: Introducing Content Security Policy
2. OWASP Content Security Policy Cheat Sheet
3. W3C Content Security Policy Specification

BPC - Best Practices Compliance

Criticality	Minor / Informative
Status	Unresolved

Description

Several issues spanning performance, security, and best practices were identified as part of the assessment. Performance metrics including Largest Contentful Paint indicate subpar performance levels, which could significantly impact user experience and engagement. Additionally, best practices violations, such as console errors, were identified. These findings underscore the importance of addressing these issues promptly to ensure the application's usability, security, and compliance with industry standards.

In summary, the assessment identified the following issues:

- Largest Contentful Paint
- Errors in console

Recommendation

The team is advised to address the identified issues and improve the overall quality of the application. Specifically, the team could ensure compliance with web development best practices by addressing the aforementioned issues. By addressing the identified issues, the application can improve its performance, security posture, and compliance with industry standards, ultimately enhancing user satisfaction and engagement.

CSAV - Cookie SameSite Attribute Vulnerability

Criticality	Minor / Informative
Status	Unresolved

Description

A cookie (cf_clearance) has been set with its SameSite attribute set to “none”, which means that the cookie can be sent as a result of a 'cross-site' request. The SameSite attribute is an effective counter measure to cross-site request forgery, cross-site script inclusion, and timing attacks.

Recommendation

To improve cookie security, the team is advised to ensure that the SameSite attribute is set to either 'lax' or preferably 'strict' for all cookies. This restricts the transmission of cookies in cross-site requests, mitigating the risk of CSRF and other related attacks. By implementing these measures, the team can enhance the security posture of the web application and protect against potential security vulnerabilities.

LTC - Latency And Throughput Challenges

Criticality	Minor / Informative
Status	Unresolved

Description

As part of the rate-limiting test, the web app highlighted concerns regarding latency and throughput, with varying response times across percentiles and an average latency of 5389.81 milliseconds. Additionally, fluctuations in data transfer rates indicate potential bottlenecks or inefficiencies in data processing and transmission, impacting system performance.

Stat	Avg	Stdev	Min
Latency	5389.81 ms	2707.48 ms	17437 ms
Req/Sec	47.5	77.83	1
Bytes/Sec	828 kB	1.36 MB	17.4 kB

Recommendation

To enhance system performance, a comprehensive performance analysis is recommended. This analysis should focus on identifying and addressing latency bottlenecks, such as inefficient database queries, resource-intensive operations, or network congestion. Optimization efforts should target the codebase, database queries, and network configurations to improve response times and enhance overall system throughput, resulting in a smoother user experience and improved system efficiency.

MSTSH - Missing Strict Transport Security Header

Criticality	Minor / Informative
Status	Unresolved

Description

The absence of the HTTP Strict Transport Security (HSTS) header poses a security risk to the application. HSTS is a crucial web security mechanism that instructs compliant web browsers to interact with the server using only secure HTTPS connections, thereby enhancing the overall security of communication between the client and the server. By enforcing HTTPS usage, HSTS helps mitigate various security threats, including man-in-the-middle attacks, network eavesdropping, and protocol downgrade attacks.

The following URLs are a sample of all the occurrences where an HTTP Strict Transport Security (HSTS) header was not set.

1. https://snork.io/cdn-cgi/challenge-platform/h/b/flow/ov1/983258204:1716361759:ocLh6q451_nf_JadtZlN8_UKfOoEURCJBbWFSOcC-Zc/887b1c1d1dc06f5c/59510aecae87864
2. https://snork.io/cdn-cgi/rum

Recommendation

To enhance the security of the application and enforce secure communication over HTTPS, it is essential to ensure that the web server, application server, load balancer, or any other relevant components are configured to enforce Strict Transport Security (HSTS). By configuring the application to enforce Strict Transport Security (HSTS) and following best practices for HSTS implementation, the application can significantly reduce the risk of network-based attacks, protect sensitive data in transit, and enhance overall security posture.

References:

1. https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html
2. https://owasp.org/www-community/Security_Headers
3. http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security

MXH - Missing X-Content-Type-Options Header

Criticality	Minor / Informative
Status	Unresolved

Description

The absence of the X-Content-Type-Options header exposes the application to potential MIME-sniffing attacks, particularly affecting older versions of Internet Explorer and Chrome. This vulnerability allows browsers to interpret response bodies as content types other than the declared type, potentially leading to security breaches and data exposure. Even error pages (e.g., 401, 403, 500) remain susceptible to such attacks, necessitating immediate action to safeguard against injection vulnerabilities.

The following URLs are a sample of all the occurrences where an X-Content-Type-Options header was not set.

1. https://snork.io/cdn-cgi/challenge-platform/h/b/orchestrate/chl_page/v1?ray=887adf4728e9eef8
2. https://snork.io/cdn-cgi/challenge-platform/h/b/orchestrate/chl_page/v1?ray=887adf472efaeee8

Recommendation

To mitigate this risk, the team is advised to ensure that the application or web server configures the Content-Type header accurately and includes the X-Content-Type-Options header set to 'nosniff' for all web pages. Additionally, consider recommending users employ modern, standards-compliant web browsers that either abstain from MIME-sniffing or allow for its suppression via directives from the server or application.

Reference:

https://learn.microsoft.com/en-us/previous-versions/windows/internet-explorer/ie-developer/compatibility/gg622941(v=vs.85) https://owasp.org/www-community/Security_Headers

SIUL - Server Instability Under Load

Criticality	Minor / Informative
Status	Unresolved

Description

The web app highlighted a concerning number of errors (7,664), out of which 7,563 were timeouts during the assessment period, indicating potential challenges with server stability and resource allocation. Such issues can significantly impact user experience and necessitate a deeper investigation into server health and capacity planning.

In summary:

- The conducted test used 3000 concurrent connections in 30 seconds timespan.
- The number of requests that were sent was 13,516 requests in 31.62 seconds.
- The number of connection errors (including timeouts) that occurred was 7,664.
- The number of connection timeouts that occurred was 7,563.

Recommendation

To mitigate these challenges, it is advised to conduct a comprehensive analysis of server logs and infrastructure to pinpoint the underlying causes of errors and timeouts. This analysis should inform the optimization of server configurations, potential resource upgrades, and the implementation of robust error-handling mechanisms. By addressing these areas, disruptions to user access can be minimized, ensuring a smoother and more reliable service experience.

Summary

This report provides a thorough assessment of the web application's security and performance. Through meticulous analysis, the report identifies vulnerabilities and weaknesses in key areas such as data handling and network security. Recommendations are provided to address these issues and enhance the application's resilience against cyber threats.

Overall, the report serves as a valuable resource, offering insights into the application's security posture and actionable recommendations to fortify its defenses. By implementing the suggested measures, the team can strengthen the app’s security foundation and maintain trust among users.

Disclaimer

The information provided in this report does not constitute investment, financial or trading advice and you should not treat any of the document's content as such. This report may not be transmitted, disclosed, referred to or relied upon by any person for any purposes nor may copies be delivered to any other person other than the Company without Cyberscope's prior written consent. This report is not nor should be considered an "endorsement" or "disapproval" of any particular project or team. This report is not nor should be regarded as an indication of the economics or value of any "product" or "asset" created by any team or project that contracts Cyberscope to perform a security assessment. This document does not provide any warranty or guarantee regarding the absolute bug-free nature of the technology analyzed, nor do they provide any indication of the technologies proprietors' business, business model or legal compliance. This report should not be used in any way to make decisions around investment or involvement with any particular project. This report represents an extensive assessment process intending to help our customers increase the quality of their code while reducing the high level of risk presented by cryptographic tokens and blockchain technology.

Blockchain technology and cryptographic assets present a high level of ongoing risk Cyberscope's position is that each company and individual are responsible for their own due diligence and continuous security Cyberscope's goal is to help reduce the attack vectors and the high level of variance associated with utilizing new and consistently changing technologies and in no way claims any guarantee of security or functionality of the technology we agree to analyze. The assessment services provided by Cyberscope are subject to dependencies and are under continuing development. You agree that your access and/or use including but not limited to any services reports and materials will be at your sole risk on an as-is where-is and as-available basis Cryptographic tokens are emergent technologies and carry with them high levels of technical risk and uncertainty. The assessment reports could include false positives false negatives and other unpredictable results. The services may access and depend upon multiple layers of third parties.

About Cyberscope

Cyberscope is a blockchain cybersecurity company that was founded with the vision to make web3.0 a safer place for investors and developers. Since its launch, it has worked with thousands of projects and is estimated to have secured tens of millions of investors' funds.

Cyberscope is one of the leading smart contract audit firms in the crypto space and has built a high-profile network of clients and partners.

The Cyberscope team

https://www.cyberscope.io