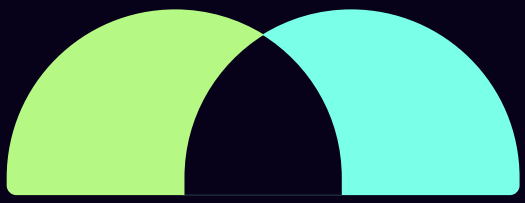


ShimmerBridge

Penetration Test Report

zokyo

Jan 19th 2024 | v. 1.0

Table of Contents

AUDITING STRATEGY AND TECHNIQUES APPLIED

The source code was taken from the Shimmer repository:

Repository: https://github.com/zokyo-sec/Shimmer-Bridge
Repository: https://github.com/LayerZero-Labs-External/x-shimmer-bridge
Project URL: https://shimmerbridge.org

The team at Zokyo was provided nearly two week for the engagement and assigned a full time security engineer to audit the security of the Shimmer code. The security engineer is a blockchain and smart-contract security expert with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The purpose of this audit is to achieve the following:

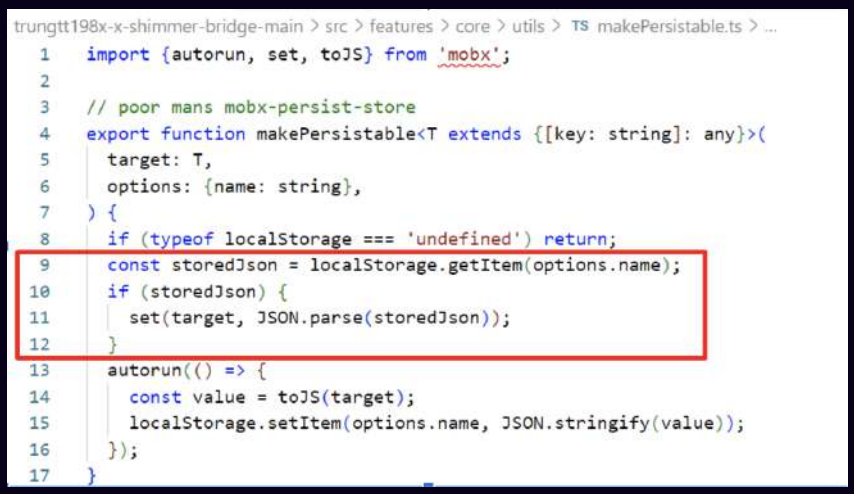
- Tested for JSON Injection.
- Tested for Path Traversal.
- Tested for Injection Vulnerability.
- Tested for Vulnerable Dependencies.
- Tested for Cross Site Scripting.
- Tested for Cryptographic Weakness.
- Tested for SSRF.
- Tested for Fuzzing.
- Scanned the repository with scanners like Graudit, Contrast, Betterscan, Bearer cli, Agnitio, Fortify and Insider Cli
- Scanned the website with scanners like OWASP Zap, BBOT, RepCheck, Realm, etc.
- Analyzed the scan reports for potential vulnerabilities.

FINDINGS

HARDCODED API KEY

Severity	Medium
Status	FIXED
	The repository stores hardcoded API key within the repository's source code. This insecure practice leaves these keys exposed and easily accessible to anyone. Attackers can exploit this vulnerability by extracting these keys, potentially compromising the security of external services, systems, or resources connected to the API.
Assets	https://github.com/LayerZero-Labs-External/x-shimmer-bridge
Evidence	
Recommendation	• Store sensitive information like API keys in environment variables instead of hardcoding them directly into the code. • Utilize a dedicated secrets management tool or service provided by your platform (like AWS Secrets Manager, HashiCorp Vault, or Azure Key Vault). • Restrict access to API keys by implementing appropriate access controls. Only grant necessary permissions to specific users or services, minimizing the risk of unauthorized access.

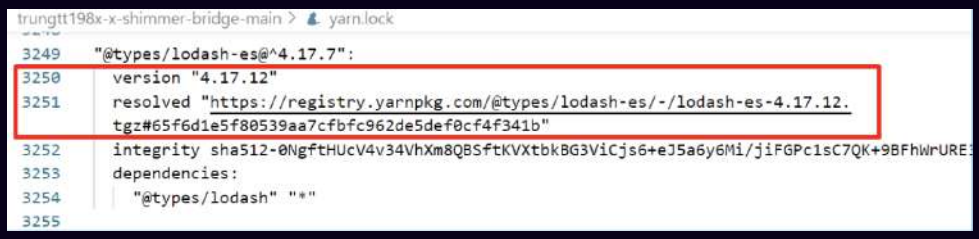
POSSIBLE JSON INJECTION

Severity	Medium
Status	FIXED
	Injecting malicious or malformed JSON payloads could lead to severe consequences, including unhandled exceptions that disrupt the application's normal operations, potentially causing system instability or crashes. Moreover, the absence of validation opens doors to unauthorized access or data manipulation, enabling attackers to leverage crafted JSON content to exploit the system.
Assets	https://github.com/LayerZero-Labs-External/x-shimmer-bridge
Evidence	Open the path and observe the storedJson constant takes value from localhost and passes it to set() function without validation inside JSON.parse. This might raise unhandled exception or JSON Injection when passing maliciously crafted JSON. 
Recommendation	• Implement strict input validation on any JSON data received by the application. • Instead of using JSON.parse directly, consider using safer JSON parsing methods or libraries that provide additional security features.

trungtt198x-x-shimmer-bridge-main > src > features > core > utils > TS makePersistable.ts > ...

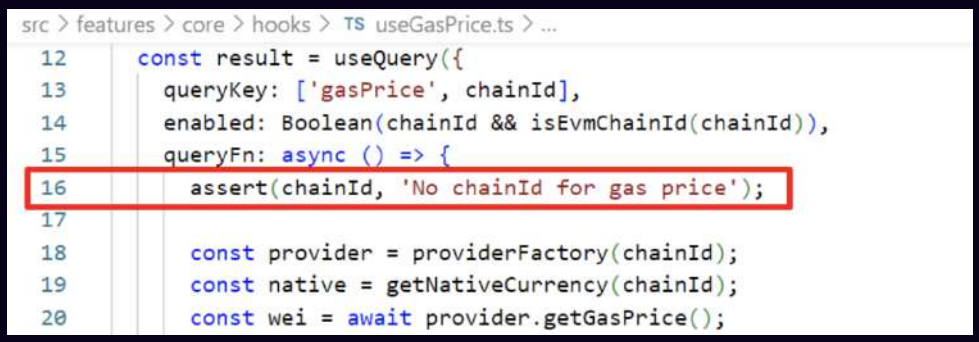
```
1  import {autorun, set, toJS} from 'mobx';
2
3  // poor mans mobx-persist-store
4  export function makePersistable<T extends {[key: string]: any}>(
5    target: T,
6    options: {name: string},
7  ) {
8    if (typeof localStorage === 'undefined') return;
9    const storedJson = localStorage.getItem(options.name);
10   if (storedJson) {
11     set(target, JSON.parse(storedJson));
12   }
13   autorun(() => {
14     const value = toJS(target);
15     localStorage.setItem(options.name, JSON.stringify(value));
16   });
17 }
```

USING COMPONENTS WITH KNOWN VULNERABILITY

Severity	Low
Status	FIXED
	Software components or libraries used in a program may have known security vulnerabilities. These vulnerabilities may include weaknesses, bugs, or flaws that have been identified and documented in publicly available databases or guidelines. If these components are not updated to patched versions, they can be used by attackers to gain unauthorized access, execute arbitrary code, or compromise system integrity.
Assets	https://github.com/LayerZero-Labs-External/x-shimmer-bridge
Evidence	Open the yarn.lock file and observe the application uses lodash 4.17.12 which has known vulnerabilities like CVE-2021-23337, CVE-2020-8203, CVE-2020-28500 
Recommendation	1. Conduct regular vulnerability assessments and scans using reputable security tools to identify potential vulnerabilities within your software components and libraries. 2. Update the used components to latest version.

USE OF RISKY 'ASSERT'

Severity	Low
Status	FIXED
	The use of `assert` signifies a critical check for conditions assumed to always hold true. Yet, if an `assert` condition fails, it abruptly halts contract execution, risking a loss of funds and leaving the contract in an inconsistent state. This vulnerability can be exploited maliciously, potentially leading to a Denial of Service (DoS) attack.
Assets	https://github.com/LayerZero-Labs-External/x-shimmer-bridge
Evidence	Paths with assert function: \trungtt198x-x-shimmer-bridge-main\src\bootstrap.ts: 163 \trungtt198x-x-shimmer-bridge-main\src\features\bridge\stores\bridgeStore.tsx: 564-579, 694-696, 731-734 \trungtt198x-x-shimmer-bridge-main\src\features\core\hooks\useGasPrice.ts: 16 \trungtt198x-x-shimmer-bridge-main\src\features\core\stores\airdropStore.tsx: 23 \trungtt198x-x-shimmer-bridge-main\src\features\core\stores\tokenStore.tsx: 23, 34 \trungtt198x-x-shimmer-bridge-main\src\features\gas\stores\gasDropStore.tsx: 237-239, 247-249, 259-261, 263, 271, 490, 494-499, 587, 597-600 \trungtt198x-x-shimmer-bridge-main\src\features\onft\hooks\useAssetMetadata.ts: 18, 19 \trungtt198x-x-shimmer-bridge-main\src\features\onft\stores\onftStore.tsx: 390-400, 403, 529, 535



Recommendation	1. Use Carefully: assert should be used judiciously for critical conditions that should never be false, like invariant checks. For conditions that could occur in regular execution (such as validating user inputs), consider using require statements instead. 2. Input Validation and Error Handling: Perform thorough input validation using require or custom checks to handle potential issues gracefully. Implement robust error handling mechanisms to recover from failed conditions without leaving the contract in an inconsistent state.

MISSING SECURITY HEADERS

Severity	Low
Status	FIXED
	The lack of security headers in the system indicates that the necessary measures have not been taken to protect against potential attacks and threats. Usually, security headers act as the first line of defense, and their absence can allow attackers to use various methods, such as code injection attacks, information interception, or hacking of system processes.
Assets	https://shimmerbridge.org/
Evidence	
Request:	GET /api/trpc/tokens?batch=1&input=%7B%7D HTTP/2 Host: shimmerbridge.org Sec-Ch-Ua: "Not_A Brand";v="8", "Chromium";v="120" Sec-Ch-Ua-Platform: "Windows" Sec-Ch-Ua-Mobile: ?0 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.6099.71 Safari/537.36 Content-Type: application/json Accept: */* Origin: google.com: Sec-Fetch-Site: same-origin Sec-Fetch-Mode: cors Sec-Fetch-Dest: empty Referer: https://shimmerbridge.org/bridge Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9 Priority: u=1, i

MISSING SECURITY HEADERS

Response:

```
HTTP/2 200 OK
Date: Tue, 26 Dec 2023 18:22:53 GMT
Content-Type: application/json
Vary: Accept-Encoding
Content-Security-Policy: frame-ancestors 'self' https://*.stargate.finance
https://stargate.finance https://verify.walletconnect.org/
Cf-Cache-Status: DYNAMIC
Report-To: {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?
s=098uBUZC3Pej9QHvVcNfCIWN9sju%2BvVBpQc64hAqO0aK%2Bxq0A6Ts
C0×3pO%2BpewcWkOcLVrfgCPfhSch0CHFdRg00NY%2FIveTIRttanhFHbnf
E75y%2BTAjexfLUF%2BYt%2F3qIUhuVYA%3D%3D"}],"group":"cf-
nel","max_age":604800}
Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Server: cloudflare
Cf-Ray: 83bb5a0c4a83f33f-BOM
Alt-Svc: h3=":443"; ma=86400
```

Recommendation:

- Configure and implement essential security headers. Each header serves a specific purpose in enhancing security.
- After implementing headers, thoroughly test the application to ensure the headers are correctly applied and do not disrupt the functionality.

We are grateful for the opportunity to work with the ShimmerBridge.

The statements made in this document should not be interpreted as an investment or legal advice, nor should its authors be held accountable for the decisions made based on them.

Zokyo Security recommends the ShimmerBridge put in place a bug bounty program to encourage further analysis of the application by third parties.

zokyo