deBridge

Backend Penetration Test

Prepared by: **Halborn**

Date of Engagement: **January 16th, 2022 - February 9th, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	01/26/2022	Afaq Abid
0.2	Draft Review	02/11/2022	Gabi Urrutia
1.0	Remediation Plan	05/16/2022	Afaq Abid
1.1	Remediation Plan Review	05/17/2022	Gabi Urrutia

CONTACTS

CONTACT	COMPANY	EMAIL
Rob Behnke	Halborn	Rob.Behnke@halborn.com
Steven Walbroehl	Halborn	Steven.Walbroehl@halborn.com
Gabi Urrutia	Halborn	Gabi.Urrutia@halborn.com
Afaq Abid	Halborn	Afaq.Abid@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

deBridge engaged Halborn to conduct a security audit on their backend APIs. The security assessment was scoped to the backend APIs. Halborn was provided access to the source code of the application, and the testing environment to conduct security testing using tools to scan, detect, validate possible vulnerabilities found in the application and report the findings at the end of the engagement.

Halborn recommends performing further testing to validate extended safety and correctness in context to the whole infrastructure when issues are fixed and new features added.

1.2 AUDIT SUMMARY

The team at Halborn was provided a timeline for the engagement and assigned two full-time security engineers to audit the security of the assets in scope. The engineers are blockchain and smart contract security experts with advanced penetration testing, smart-contract hacking, and deep knowledge of multiple blockchain protocols.

The goals of our security audits are to improve the quality of the systems we review and to target sufficient remediation to help protect users.

In summary, Halborn identified some security risks that were mostly addressed by the deBridge team.

1.3 TEST APPROACH & METHODOLOGY

Halborn performed a combination of manual and automated security testing to balance efficiency, timeliness, practicality, and accuracy regarding the scope of the pentest. While manual testing is recommended to uncover flaws in logic, process and implementation; automated testing techniques assist enhance coverage of the infrastructure and can quickly

identify flaws in it. The following phases and associated tools were used throughout the term of the audit:

- Mapping API Content and Functionality
- API Logical Flaws
- Rate Limitations Tests
- API misconfiguration
- Input Handling
- CloudFlare Bypass
- Fuzzing of all input parameters
- Test for Injection (SQL/JSON/HTML/Command)

RISK METHODOLOGY:

Vulnerabilities or issues observed by Halborn are ranked based on the risk assessment methodology by measuring the **LIKELIHOOD** of a security incident and the **IMPACT** should an incident occur. This framework works for communicating the characteristics and impacts of technology vulnerabilities. The quantitative model ensures repeatable and accurate measurement while enabling users to see the underlying vulnerability characteristics that were used to generate the Risk scores. For every vulnerability, a risk level will be calculated on a scale of 5 to 1 with 5 being the highest likelihood or impact.

RISK SCALE - LIKELIHOOD

5 - Almost certain an incident will occur.
4 - High probability of an incident occurring.
3 - Potential of a security incident in the long term.
2 - Low probability of an incident occurring.
1 - Very unlikely issue will cause an incident.

RISK SCALE - IMPACT

5 - May cause devastating and unrecoverable impact or loss.
4 - May cause a significant level of impact or loss.
3 - May cause a partial impact or loss to many.
2 - May cause temporary impact or loss.

1 - May cause minimal or un-noticeable impact.

The risk level is then calculated using a sum of these two values, creating a value of 10 to 1 with 10 being the highest level of security risk.

10 - CRITICAL
9 - 8 - HIGH
7 - 6 - MEDIUM
5 - 4 - LOW
3 - 1 - VERY LOW AND INFORMATIONAL

1.4 SCOPE

The following APIs and their respective repository is in scope:

- https://testapi101.debridge.finance/
- https://api.debridge.finance/
- http://testnetvalidatorsapi.debridge.io/
- Mainnet Validators API

Repository: Github-Repo

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	2	2	1	3

LIKELIHOOD

IMPACT

			(HAL-02)	
		(HAL-03)	(HAL-01)	
			(HAL-04)	
(HAL-06)	(HAL-05)			
(HAL-07) (HAL-08)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
HAL-01 - DENIAL OF SERVICE	High	SOLVED - 05/17/2022
HAL-02 - OBSOLETE VERSION OF NGINX	High	SOLVED - 05/11/2022
HAL-03 - HARDCODED SECRETS	Medium	SOLVED - 05/11/2022
HAL-04 - LACK OF RESOURCES AND RATE LIMITING	Medium	SOLVED - 05/17/2022
HAL-05 - OUTDATED VERSIONS OF TLS SUPPORTED	Low	SOLVED - 05/16/2022
HAL-06 - MISSING SECURITY HEADERS	Informational	SOLVED - 05/16/2022
HAL-07 - WEAK CREDENTIALS	Informational	NOT APPLICABLE
HAL-08 - NGINX VERSION DISCLOSURE	Informational	SOLVED - 05/11/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) DENIAL OF SERVICE - HIGH

Description:

ClientAPI is vulnerable to DoS attack. It is possible to bring down the service simply by sending multiple requests to the API endpoint. During our assessment, we tested the following endpoint and were able to disable the service.

- https://testapi101.debridge.finance/api/Transactions/getEvents

Proof of concept:

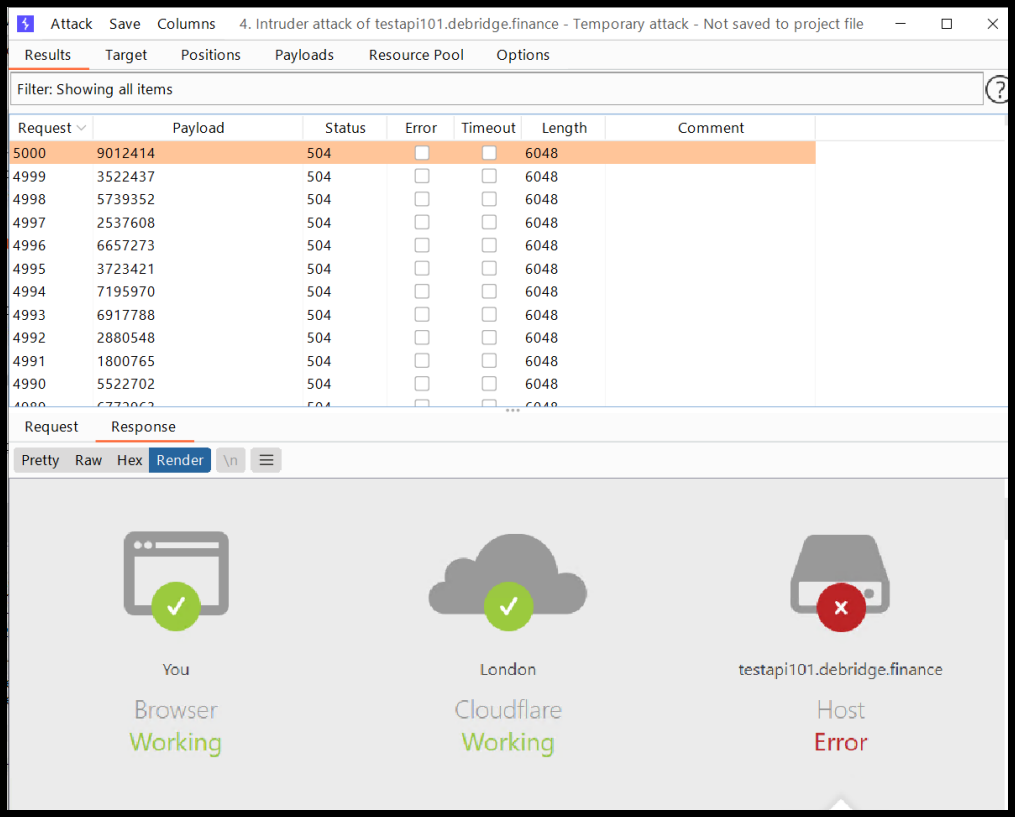

Figure 1: https://testapi101.debridge.finance/

Risk Level:

Likelihood - 4
Impact - 4

Recommendation:

We recommend that you implement Load-Balancers to handle traffic and implement rate limitation as well. We also recommend that you optimize your code/logic and update your server configurations to handle multiple/ambiguous requests.

Remediation Plan:

SOLVED: The deBridge team fixed the issue by implementing the rate limitation. The issue was fixed on the mainnet API (api.debridge.finance) as their testnet is not up-to-date as the protocol is live on mainnet.

3.2 (HAL-02) OBSOLETE VERSION OF NGINX - HIGH

Description:

A service banner response from the remote host indicates an installation of the Nginx HTTP Server at a level indicating the operational HTTP Server on the remote host is outdated, vulnerable to multiple vulnerabilities (Critical, High and Medium).

Reference: Synk: nginx:1.18.0 Vulnerabilities

Screenshots/Videos:

Risk Level:

Likelihood - 4
Impact - 5

Recommendation:

It is recommended to upgrade the Nginx to the latest and most stable version.

Remediation Plan:

SOLVED: The deBridge team fixed the issue by adding the appropriate checks.

3.3 (HAL-03) HARCODED SECRETS - MEDIUM

Description:

During the assessment, we found the Hardcoded Secrets in the source code of the Backend APIs. When sharing public/private code, it is easy to forget your credentials in the code and leave them exposed. Repositories are frequently cloned and forked into new projects, giving new developers access to their full history. Any credentials within the repository history will exist in all new repositories and, due to the storage of credentials in plain text, could lead to possible misuse of the credentials. In case of any violation or potential access to the source code could lead to further exploitation of the infrastructure.

Proof of concept:

- DBCredentials deBridge.Datacollector/appsettings.Development.json:22
- DBCredentials DeBridge.ValidatorAPI/appsettings.json:10
- DBCredentials DeBridge.ClientAPI/appsettings.json:9
- DBCredentials DeBridge.ClientAPI/appsettings.Development.json:11
- PrivateKeys DeBridge.DataCollector/Program - Copy.cs

Risk Level:

Likelihood - 3
Impact - 4

Recommendation:

We recommend that you follow the best security practices:

- Use environment variables to store secrets

- Add sensitive files in .gitignore
- Use encryption to store secrets instead of plain text

Remediation Plan:

SOLVED: The deBridge team removed the Program - Copy.cs file from repository, Moreover, the other credentials were test/example credentials.

3.4 (HAL-04) LACK OF RESOURCES AND RATE LIMITING - MEDIUM

Description:

API requests consume resources such as network, CPU, memory, and storage. This vulnerability occurs when too many requests come in at the same time, and the API does not have enough compute resources to handle those requests.
An attacker could exploit this vulnerability to overload the API by sending more requests than it can handle. As a result, the API becomes unavailable or unresponsive to new requests.

Proof-Of-Concept:

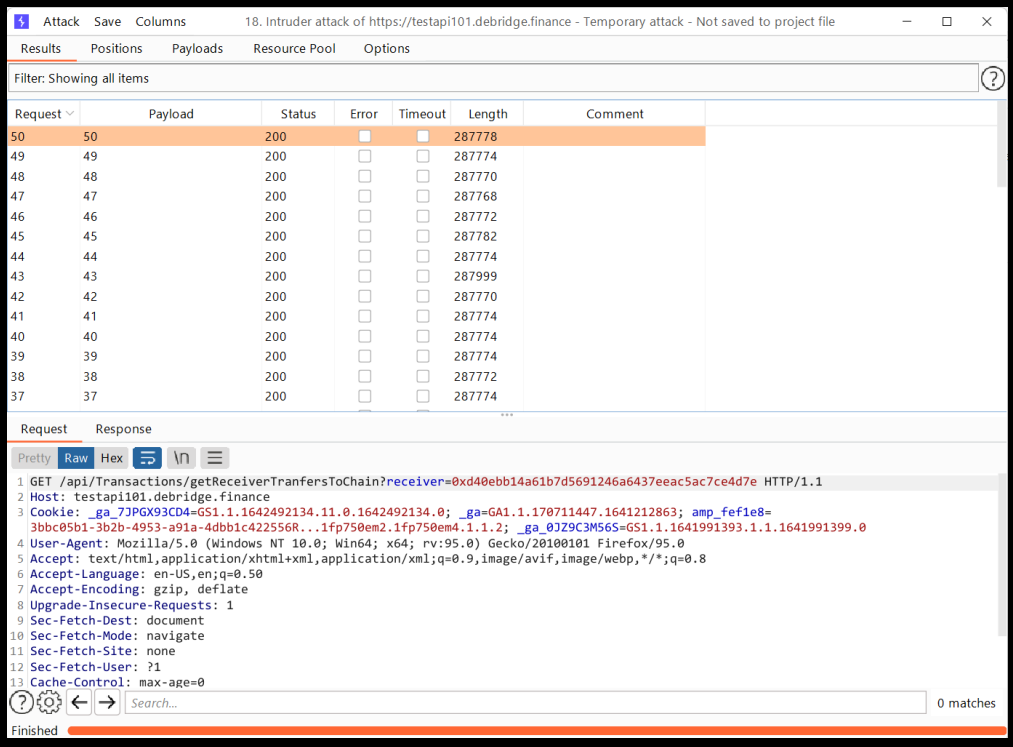

Figure 2: https://testnetvalidatorsapi.debridge.io/

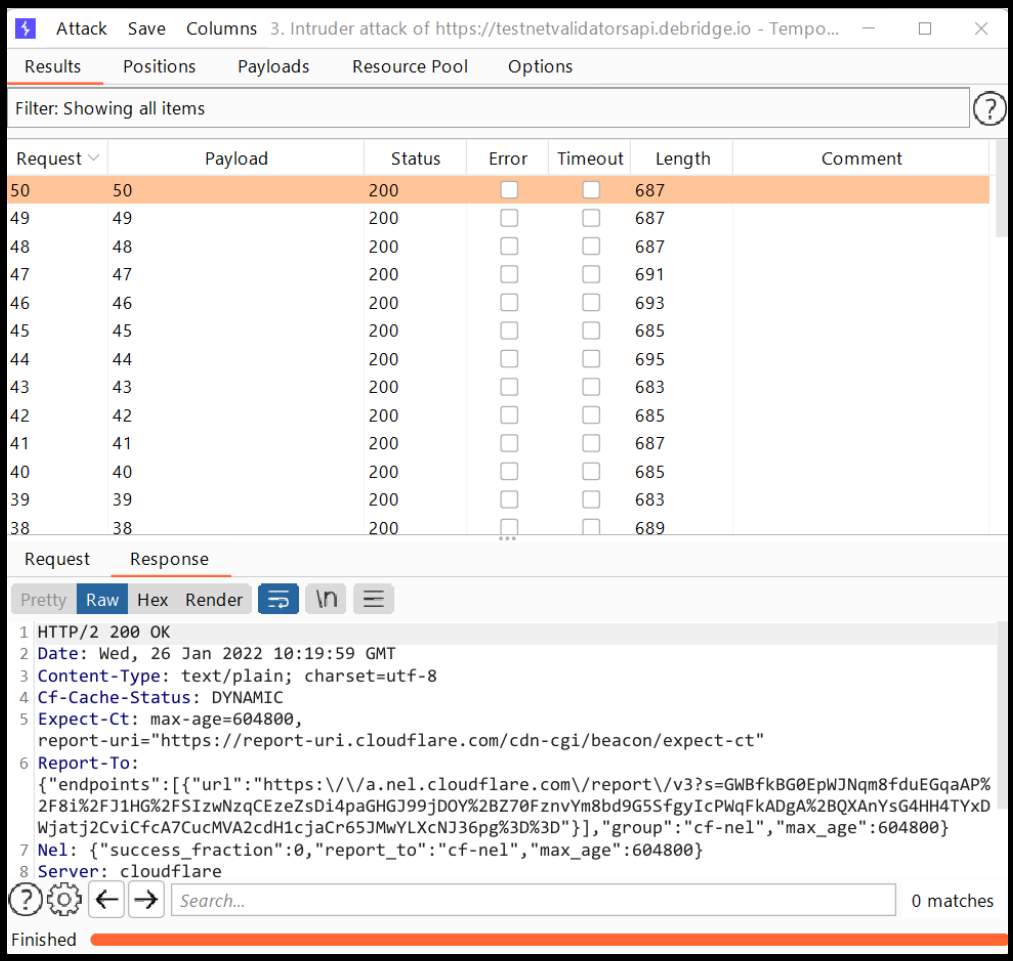

Figure 3: https://testapi101.debridge.finance/

Risk Level:

Likelihood - 4
Impact - 3

Recommendation:

This vulnerability is due to the application accepting requests from users at a given time without performing request limitation checks. We recommend that you follow the following best practices:

- Implement a limit on how often a client can call the API within a defined time frame.
- Notify the client when the limit is exceeded by providing the limit

number and the time the limit will be reset.
- Define and enforce maximum data size on all incoming parameters and payloads, such as the maximum length of strings and the maximum number of elements in arrays.

Reference: CWE-770: Allocation of Resources Without Limits or Throttling

Remediation Plan:

SOLVED: The deBridge team fixed the issue by implementing the rate limitation.

3.5 (HAL-05) OUTDATED VERSIONS OF TLS SUPPORTED - LOW

Description:

We have identified that the scoped URLs are compatible with deprecated versions of TLS v1.0 and TLS v1.1 which are not considered secure and contain numbers of cryptographic design flaws. While attacks that target weak cipher suites or algorithms are complex to execute, with constant progress in computational power, these attacks become easier to pull off over time. As such, it is recommended that you configure the connection to comply with security best practices.

Analysis:

```
Start 2022-02-09 18:21:10        -->> 104.22.12.82:443 (testapi101.debridge.finance) <<--

Further IP addresses:   104.22.13.82 172.67.10.73 2606:4700:10::6816:d52 2606:4700:10::ac43:a49 2606:4700:10::6816:c52
rDNS (104.22.12.82):    --
Service detected:       HTTP


Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   not offered
ALPN/HTTP2 h2, http/1.1 (offered)

Testing cipher categories

NULL ciphers (no encryption)                not offered (OK)
Anonymous NULL Ciphers (no authentication)  not offered (OK)
Export ciphers (w/o ADH+NULL)               not offered (OK)
LOW: 64 Bit + DES, RC[2,4] (w/o export)     not offered (OK)
Triple DES Ciphers / IDEA                   not offered
Obsolete CBC ciphers (AES, ARIA etc.)       offered
Strong encryption (AEAD ciphers)            offered (OK)
```

Figure 4: https://testapi101.debridge.finance/

```
Start 2022-02-09 18:21:16        -->> 104.21.47.129:443 (testnetvalidatorsapi.debridge.io) <<--

Further IP addresses:   172.67.171.38 2606:4700:3032::ac43:ab26 2606:4700:3037::6815:2f81
rDNS (104.21.47.129):   --
Service detected:       HTTP


 Testing protocols via sockets except NPN+ALPN

SSLv2      not offered (OK)
SSLv3      not offered (OK)
TLS 1      offered (deprecated)
TLS 1.1    offered (deprecated)
TLS 1.2    offered (OK)
TLS 1.3    offered (OK): final
NPN/SPDY   not offered
ALPN/HTTP2 h2, http/1.1 (offered)

 Testing cipher categories

NULL ciphers (no encryption)                not offered (OK)
Anonymous NULL Ciphers (no authentication)  not offered (OK)
Export ciphers (w/o ADH+NULL)               not offered (OK)
LOW: 64 Bit + DES, RC[2,4] (w/o export)     not offered (OK)
Triple DES Ciphers / IDEA                   offered
Obsolete CBC ciphers (AES, ARIA etc.)       offered
Strong encryption (AEAD ciphers)            offered (OK)
```

Figure 5: https://testnetvalidatorsapi.debridge.io/

Risk Level:

Likelihood - 2

Impact - 2

Recommendation:

It should be noted that all major browser vendors coordinated to remove support for TLSv1.0 and TLSv1.1 in March 2020; however, these deprecated versions will still be available for outdated browser versions.

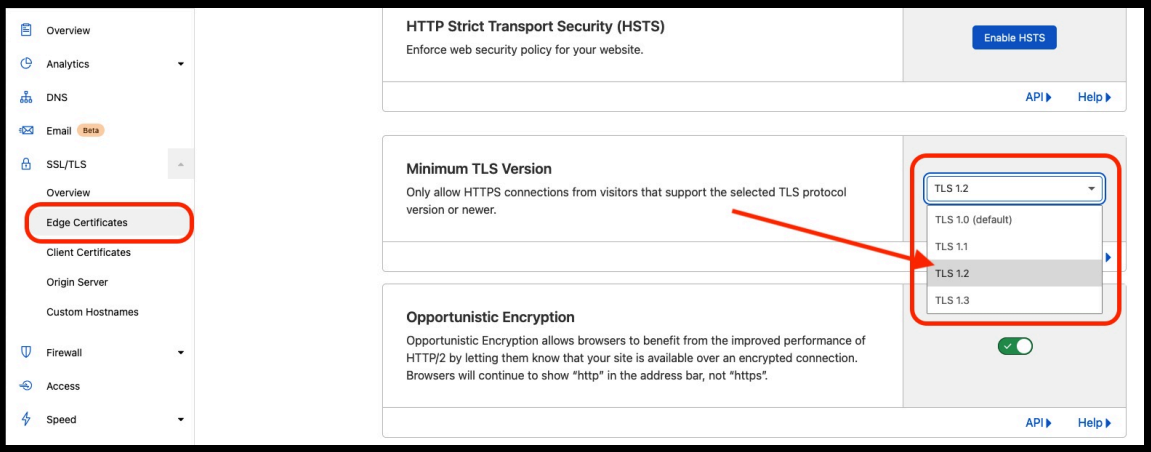

Figure 6: Fix for Cloudflare

It is recommended to adhere to security best practices and use only TLS v1.2, TLS v1.3 as these are considered secure and strong and also disable support for previous versions.

Remediation Plan:

SOLVED: The deBridge team fixed the issue by setting the TLS v1.2 and TLS v1.3 only for communication.

3.6 (HAL-06) MISSING SECURITY HEADERS - INFORMATIONAL

Description:

We have identified that important security Headers are missing from deBridge BackendAPIs. These headers used by the client browser improve the security of end users against common attacks. Although these are just APIs but in an advance scenario, it is still a chance to exploit the vulnerabilities, so it is still recommended to follow the security best practices and implement these headers.

Missing important security headers; **Strict-Transport-Security**, **X-Frame-Options**, **X-Content-Type-Options** and **Content-Security-Policy** response headers.

- Strict-Transport-Security (HSTS) HTTP Strict Transport Security is an important security header to implement, as it makes the browser only communicate over HTTPS. Recommended value "Strict-Transport-Security: max-age=31536000; includeSubDomains".
- X-Frame-Options tells the browser whether you want to allow your site to be framed or not. By preventing a browser from framing your site, you can defend against attacks like clickjacking. Recommended value "X-Frame-Options: SAMEORIGIN".
- X-Content-Type-Options prevents a browser from trying to MIME-sniff the content-type and forces it to stick to the declared content-type. The only valid value for this header is "X-Content-Type-Options: nosniff".
- Content-Security-Policy is an effective measure to protect your site from XSS attacks. By whitelisting approved content sources, you can prevent the browser from loading malicious assets.

Results:

```
HTTP/1.1 405 Method Not Allowed
Allow: POST
CF-Cache-Status: DYNAMIC
CF-RAY: 6dad78c4e9dd75db-LHR
Connection: keep-alive
Content-Length: 0
Date: Wed, 09 Feb 2022 13:33:50 GMT
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
Server: cloudflare
```

Figure 7: https://testapi101.debridge.finance/

```
HTTP/1.1 200 OK
CF-Cache-Status: DYNAMIC
CF-RAY: 6dad7b64bc741e79-AMS
Connection: keep-alive
Content-Encoding: gzip
Content-Type: text/plain; charset=utf-8
Date: Wed, 09 Feb 2022 13:35:37 GMT
Expect-CT: max-age=604800, report-uri="https://report-uri.cloudflare.com/cdn-cgi/beacon/expect-ct"
NEL: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Report-To: {"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=s2ZJXNJmaCDFO5BcpLhkTWK
BVhR%2B1CWLJXp%2BVrW6CAGTBl0s%2Bw5bqP52YgSoR8bM7gaBagr35Q%3D%3D"}],"group":"cf-nel","max_age":604800}
Server: cloudflare
Transfer-Encoding: chunked
alt-svc: h3=":443"; ma=86400, h3-29=":443"; ma=86400
```

Figure 8: http://testnetvalidatorsapi.debridge.io/

Risk Level:

Likelihood - 1

Impact - 2

Recommendation:

It is recommended to define Strict-Transport-Security, X-Content-Type-Options=nosniff, Content-Security-Policy and X-Frame-Options response headers with proper policies.

Reference:

Strict-Transport-Security

X-Content-Type-Options

X-Frame-Options

Content Security Policy
Content-Type

Remediation Plan:

SOLVED: The deBridge team fixed the issue by adding the appropriate checks.

3.7 (HAL-07) WEAK CREDENTIALS - INFORMATIONAL

Description:

We have observed that the credentials found in source code of the Backend-APIs do not follow the best security practices for password. These found credentials are easy to guess/bruteforce, making them easier for an attacker to compromise.

Risk Level:

Likelihood - 1
Impact - 1

Proof of concept:

- DeBridge.ValidatorAPI/appsettings.json:10
- DeBridge.ClientAPI/appsettings.json:9
- DeBridge.ClientAPI/appsettings.Development.json:11

Recommendation:

We recommend that you follow the strong password policy internally. Credentials for infrastructure services should have lowercase, uppercase, numbers, and symbols with a minimum length of 20 characters.

Remediation Plan:

NOT APPLICABLE: The deBridge team stated that the credentials found were only examples/placeholders and these are only for their localhost and internal testing configurations files.

3.8 (HAL-08) NGINX VERSION DISCLOSURE - INFORMATIONAL

Description:

We have identified that the deBridge API reveals the exact version of the nginx server on the error page.

This gives the attacker an idea of the exact version number of the backend technology stack which allows them to design and target attacks specifically designed for this version of the the application.

Screenshot:

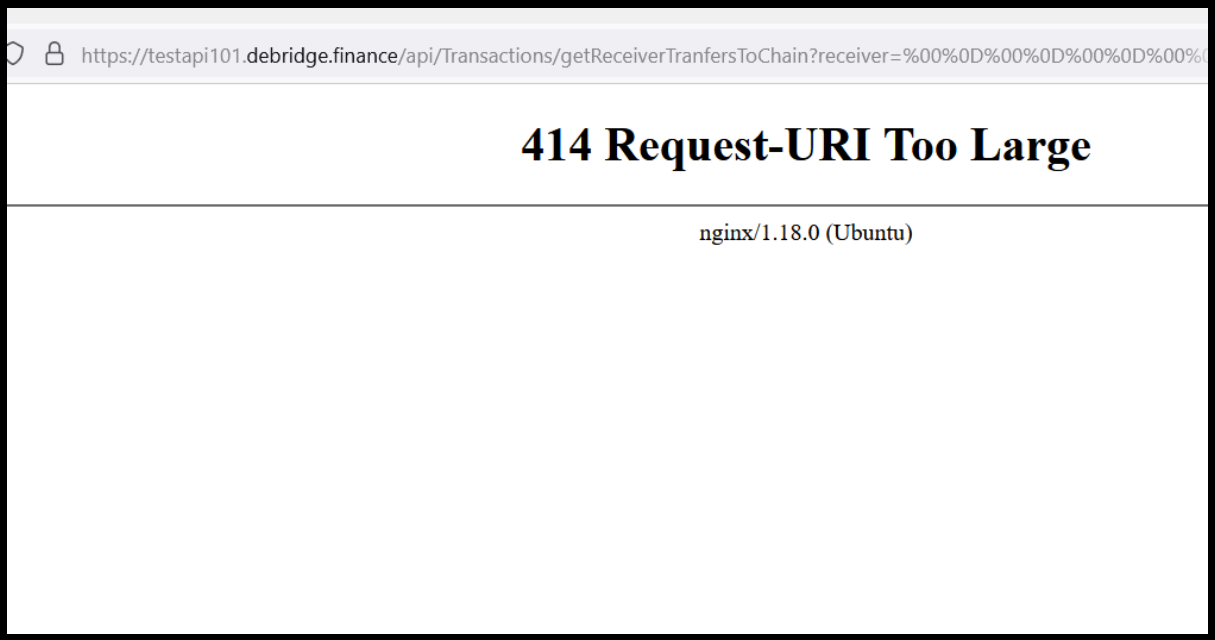

Figure 9: https://testapi101.debridge.finance/

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

By updating nginx.conf with the following, you can remediate this:

Listing 1

```
...
 server_tokens off;
...
```

Remediation Plan:

SOLVED: The deBridge team fixed the issue by adding the appropriate checks.

PERFORMED TESTS

4.1 Fuzzing

Description:

Fuzz testing or Fuzzing is a Black Box software testing technique, which basically consists of a wide range of invalid and unexpected data into an application, then monitoring the application for exceptions and crashes.

Goal:

Fuzz parameters and determine any unexpected crashes against provided input handling to find vulnerabilities.

Result:

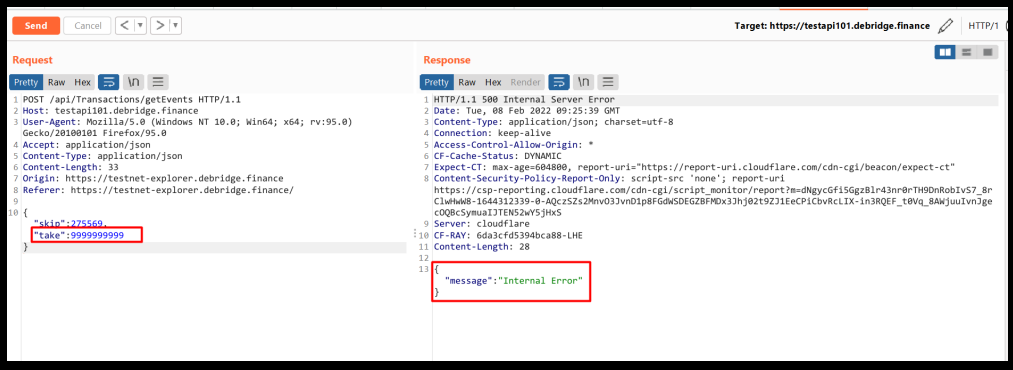

Figure 10: https://testapi101.debridge.finance/api/Transactions/getEvents

We have found an exception at /api/Transactions/getEvents in take parameter, which later lead to DoS after further testing. We recommend you to investigate this further.

4.2 Rate Limitation

Description:

Rate limits define the maximum number of requests that can be sent in a given time frame. By implementation of rate limitation which protect your infrastructure from being overwhelmed by malicious requests and make your infrastructure available for legitimate users.

Goal:

Determine the security measures against the number of requests in a short time frame.

Result:

The vulnerable endpoints already reported above.

THANK YOU FOR CHOOSING

// HALBORN