// HALBORN

Beanstalk - UI

WebApp Pentest

Prepared by: **Halborn**

Date of Engagement: **September 5th, 2022 - September 23rd, 2022**

Visit: **Halborn.com**

DOCUMENT REVISION HISTORY

VERSION	MODIFICATION	DATE	AUTHOR
0.1	Document Creation	09/21/2022	Afaq Abid
0.2	Document Updates	09/26/2022	Afaq Abid
0.3	Draft Review	09/28/2022	Gabi Urrutia
1.0	Remediation Plan	10/14/2022	Afaq Abid
1.1	Remediation Plan Review	10/18/2022	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
Constantin Casmir	Halborn	Constantin.Casmir@halborn.com
Afaq Abid	Halborn	Afaq.Abid@halborn.com

EXECUTIVE OVERVIEW

1.1 INTRODUCTION

Beanstalk engaged Halborn to conduct a security assessment on their frontend UI beginning on September 5th, 2022 and ending on September 23rd, 2022. The security assessment was scoped to the web application. 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 a full-time security engineer to perform assessment on the Beanstalk web application. The security engineer is a blockchain, web application and smart-contract security expert with advanced penetration testing, smart-contract hacking, and in-depth knowledge of multiple blockchain protocols.

The goals of the security audit are to improve the quality of systems by testing this as a blackbox and whitebox approach.

In summary, Halborn found secrets in the history of git and hardcoded API keys in the repository. Moreover, there was no rate limitation, and missing important security headers were also identified. An informational issue was also reported, further improving the security posture of the application. Beanstalk fixed almost all of those issues, except a security header that required more testing before implementation.

1.3 TEST APPROACH & METHODOLOGY

Halborn followed the Blackbox and Whitebox methodology and performed a combination of manual and automated security testing with both 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 and can quickly identify infrastructure failures. The following phases were followed, but not limited throughout the term of the audit:

- Mapping Application Content and Functionality
- Application Logic Flaws
- Rate Limitations Tests
- API misconfiguration
- Brute Force Attempts
- Input Handling
- CloudFlare Bypass
- Fuzzing of all input parameters
- Test for Injection (SQL/JSON/HTML/Command)
- Known vulnerabilities in 3rd party / OSS dependencies.

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

IN-SCOPE:
The following URL, and its respective repository, was in scope:

- https://Beanstalk-ui-audit-b5305c.netlify.app/
- commit: b5305c5bd7e028bac9abd8e9129a85fd232c3e5e

OUT-OF-SCOPE:
- External libraries

2. ASSESSMENT SUMMARY & FINDINGS OVERVIEW

CRITICAL	HIGH	MEDIUM	LOW	INFORMATIONAL
0	0	2	1	2

LIKELIHOOD

IMPACT

		(HAL-01)	(HAL-02)	
	(HAL-03)			
(HAL-04) (HAL-05)				

EXECUTIVE OVERVIEW

SECURITY ANALYSIS	RISK LEVEL	REMEDIATION DATE
SECRETS IN GIT HISTORY	Medium	SOLVED - 10/14/2022
LACK OF RATE LIMITING	Medium	SOLVED - 10/14/2022
LACK X-FRAME-OPTIONS HEADER	Low	PARTIALLY SOLVED - 10/14/2022
OUTDATED SOFTWARE IN-USE	Informational	SOLVED - 10/14/2022
HARCODED SECRETS	Informational	SOLVED - 10/14/2022

FINDINGS & TECH DETAILS

3.1 (HAL-01) SECRETS IN GIT HISTORY - MEDIUM

Description:

The git commit history of the Beanstalk repository contains secrets. Sharing public/private code is easy to forget your secrets 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 secrets within the repository history will exist in all new repositories, and due to the storage of the credentials in plain text, could lead to possible misuse of the credentials. In case of any breach or potential access to the source code, it could lead to further exploitation of the infrastructure.

Proof of concept:

The results then contain the commit ID along with any matching secrets in the git commit history.

Repo : Beanstalk-UI

Listing 1: Beanstalk-UI (Lines 7,25,43,61,79,97,115,133,151,169,187)

```
{
  "Description": "Generic API Key",
  "StartLine": 16,
  "EndLine": 16,
  "StartColumn": 15,
  "EndColumn": 56,
  "Match": "apiKey: 'f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs'",
  "Secret": "f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs",
  "File": "src/util/Curve.ts",
  "Commit": "7ae74c4512ea66a17c9cc9398c534853f45ba6ec",
  "Entropy": 4.6875,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-06-21T23:41:46Z",
  "Message": "wip: curve routing; support Astro and Cujo nodes",
```

```
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 32,
  "EndLine": 32,
  "StartColumn": 10,
  "EndColumn": 51,
  "Match": "apiKey: 'f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs'",
  "Secret": "f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs",
  "File": "src/util/Curve.ts",
  "Commit": "7ae74c4512ea66a17c9cc9398c534853f45ba6ec",
  "Entropy": 4.6875,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-06-21T23:41:46Z",
  "Message": "wip: curve routing; support Astro and Cujo nodes",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 4,
  "EndLine": 4,
  "StartColumn": 34,
  "EndColumn": 75,
  "Match": "apiKey: 'f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs'",
  "Secret": "f06l9TnsZyxvF0JaPzjoWQ_6baS5hEQs",
  "File": "src/util/Curve.ts",
  "Commit": "31c75e840e30aff36fea99d48f6005e8a3ec7384",
  "Entropy": 4.6875,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-06-18T23:25:09Z",
  "Message": "add Curve router test",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 3,
  "EndLine": 3,
  "StartColumn": 14,
```

```
  "EndColumn": 53,
  "Match": "KEY = '23db69ab62394f4eb41db6a21853402c'",
  "Secret": "23db69ab62394f4eb41db6a21853402c",
  "File": "src/constants/rpc/infura.ts",
  "Commit": "b3ca2c5f2847491329b11496fbfe2d2e8210218b",
  "Entropy": 3.7028196,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-03-31T20:35:45Z",
  "Message": "add alchemy; annotate chain calls",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 85,
  "EndLine": 85,
  "StartColumn": 24,
  "EndColumn": 75,
  "Match": "API_KEY = 'MLpZllgINkSrNFn4XySbgx2r4bzAv95zlzEofKWJ'",
  "Secret": "MLpZllgINkSrNFn4XySbgx2r4bzAv95zlzEofKWJ",
  "File": "src/constants/values.ts",
  "Commit": "c76d546237ef7132910bc565b49783dd42ebf11e",
  "Entropy": 4.7841835,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-03-22T22:24:51Z",
  "Message": "test: remove extra bundled themes",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 87,
  "EndLine": 87,
  "StartColumn": 25,
  "EndColumn": 68,
  "Match": "API_KEY = '3e668b7b056a45ac9980fd8064f9d51d'",
  "Secret": "3e668b7b056a45ac9980fd8064f9d51d",
  "File": "src/constants/values.ts",
  "Commit": "c76d546237ef7132910bc565b49783dd42ebf11e",
  "Entropy": 3.7570488,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
```

```
  "Date": "2022-03-22T22:24:51Z",
  "Message": "test: remove extra bundled themes",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 3,
  "EndLine": 3,
  "StartColumn": 14,
  "EndColumn": 53,
  "Match": "KEY = '23db69ab62394f4eb41db6a21853402c'",
  "Secret": "23db69ab62394f4eb41db6a21853402c",
  "File": "src/constants/infura.ts",
  "Commit": "8cb2f2075c5666942ff5db020a07e9f9aa3fed5d",
  "Entropy": 3.7028196,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-03-22T00:30:44Z",
  "Message": "lint + no wallet connection flow",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 3,
  "EndLine": 3,
  "StartColumn": 15,
  "EndColumn": 54,
  "Match": "KEY = \"23db69ab62394f4eb41db6a21853402c\"",
  "Secret": "23db69ab62394f4eb41db6a21853402c",
  "File": "src/constants/infura.ts",
  "Commit": "daf481ea7dbd667b9384eac10fa5a89ac48ce306",
  "Entropy": 3.7028196,
  "Author": "Silo Chad",
  "Email": "silochad@protonmail.com",
  "Date": "2022-03-21T20:07:50Z",
  "Message": "use chain provided by wallet",
  "Tags": [],
  "RuleID": "generic-api-key"
 },
 {
  "Description": "Generic API Key",
  "StartLine": 73,
```

```
148   "EndLine": 73,
149   "StartColumn": 21,
150   "EndColumn": 64,
151   "Match": "API_KEY = '3e668b7b056a45ac9980fd8064f9d51d'",
152   "Secret": "3e668b7b056a45ac9980fd8064f9d51d",
153   "File": "src/constants/values.ts",
154   "Commit": "865c20b55cd5ec179460ef4fcb1adb757cb422d6",
155   "Entropy": 3.7570488,
156   "Author": "Silo Chad",
157   "Email": "silochad@protonmail.com",
158   "Date": "2022-03-02T18:35:05Z",
159   "Message": "Ropsten infura url, add small subset of tokens to
 ↳ ropsten",
160   "Tags": [],
161   "RuleID": "generic-api-key"
162  },
163  {
164   "Description": "Generic API Key",
165   "StartLine": 67,
166   "EndLine": 67,
167   "StartColumn": 21,
168   "EndColumn": 72,
169   "Match": "API_KEY = 'MLpZllgINkSrNFn4XySbgx2r4bzAv95zlzEofKWJ'",
170   "Secret": "MLpZllgINkSrNFn4XySbgx2r4bzAv95zlzEofKWJ",
171   "File": "src/constants/values.ts",
172   "Commit": "54e85f023ea45f24a60f5fbe194fac83c1b58a55",
173   "Entropy": 4.7841835,
174   "Author": "Beanstalk Farms",
175   "Email": "88561107+BeanstalkFarms@users.noreply.github.com",
176   "Date": "2021-12-26T01:04:51Z",
177   "Message": "Merge pull request #70 from BeanstalkFarms/
 ↳ development\n\nDevelopment -\u003e Master\n\ncommit 4
 ↳ dc34e92a8ce81b00d591f079ac5046425a0e2c0\nAuthor: Beanstalk Farms \
 ↳ u003c88561107+BeanstalkFarms@users.noreply.github.com\u003e\nDate:
 ↳    Sat Dec 25 19:57:48 2021 -0500\n\nDisplay winter NFTs (#71)",
178   "Tags": [],
179   "RuleID": "generic-api-key"
180  },
181  {
182   "Description": "Generic API Key",
183   "StartLine": 57,
184   "EndLine": 57,
185   "StartColumn": 12,
186   "EndColumn": 57,
```

```
  "Match": "API_KEY = 'NU3WFYG5RBQHP6KIJKWVGCKAHK9PFUAC8D'",
  "Secret": "NU3WFYG5RBQHP6KIJKWVGCKAHK9PFUAC8D",
  "File": "src/util/LedgerUtilities.tsx",
  "Commit": "d77eb5f6f889bdda58a398dc8a2a515ad1c9cbbc",
  "Entropy": 4.3815804,
  "Author": "Fernando Ribeiro Aguilar",
  "Email": "fernando.aguilar@hotmail.com.br",
  "Date": "2021-12-02T11:02:53Z",
  "Message": "Add eth price",
  "Tags": [],
  "RuleID": "generic-api-key"
 }
]
```


Risk Level:

Likelihood - 3
Impact - 3

Recommendation:

It is recommended to revoke all secrets that are currently in the git history of your repositories. For any future commits, it is recommended to follow the security best practices:

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

Reference:

GitHub Encrypted Secrets

Remediation Plan:

SOLVED: This issue was fixed, and all keys have been rotated out of production, Beanstalk has removed the Alchemy/Infura keys from the respective services.

3.2 (HAL-02) LACK OF RATE LIMITING - MEDIUM

Description:

API requests consume resources such as network, CPU, memory, and storage. This vulnerability occurs when too many requests arrive simultaneously, 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:

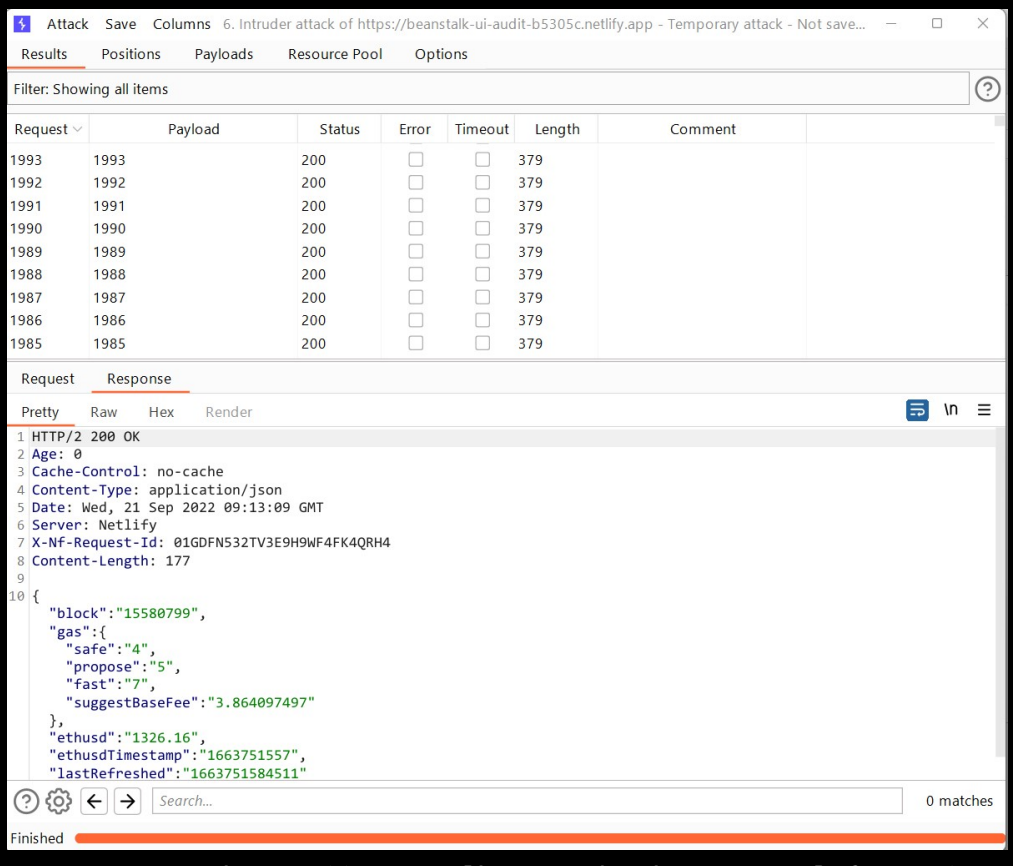


Figure 1: https://Beanstalk-ui-audit-b5305c.netlify.app/

Note: More than 200 requests were sent in a short period of time.

Risk Level:

Likelihood - 4
Impact - 3

Recommendation:

This vulnerability is due to the application accepting user requests at a given time without performing request throttling checks. The following best practices are recommended:

- Implement a limit on how often a client can call the API within a defined timeframe.
- 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 Beanstalk team fixed this issue by adding the rate limiting to their application.

3.3 (HAL-03) LACK X-FRAME-OPTIONS HEADER - LOW

Description:

Important security Headers are missing from the Web App. These headers used by the client browser improve the security of end users against common attacks.

Missing important security headers;
X-Content-Type-Options Content-Security-Policy response headers
X-Frame-Options

- 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 sources of approved content, it is possible to prevent the browser from loading malicious assets.
- X-Frame-Options header is not present in the response of the scoped websites, which makes them vulnerable to the Clickjacking attack.Clickjacking is a malicious technique to trick a web user into clicking something other than what the user perceives them to be doing, potentially revealing sensitive information or taking control of their computer while clicking seemingly innocuous web pages.

Proof of concept:

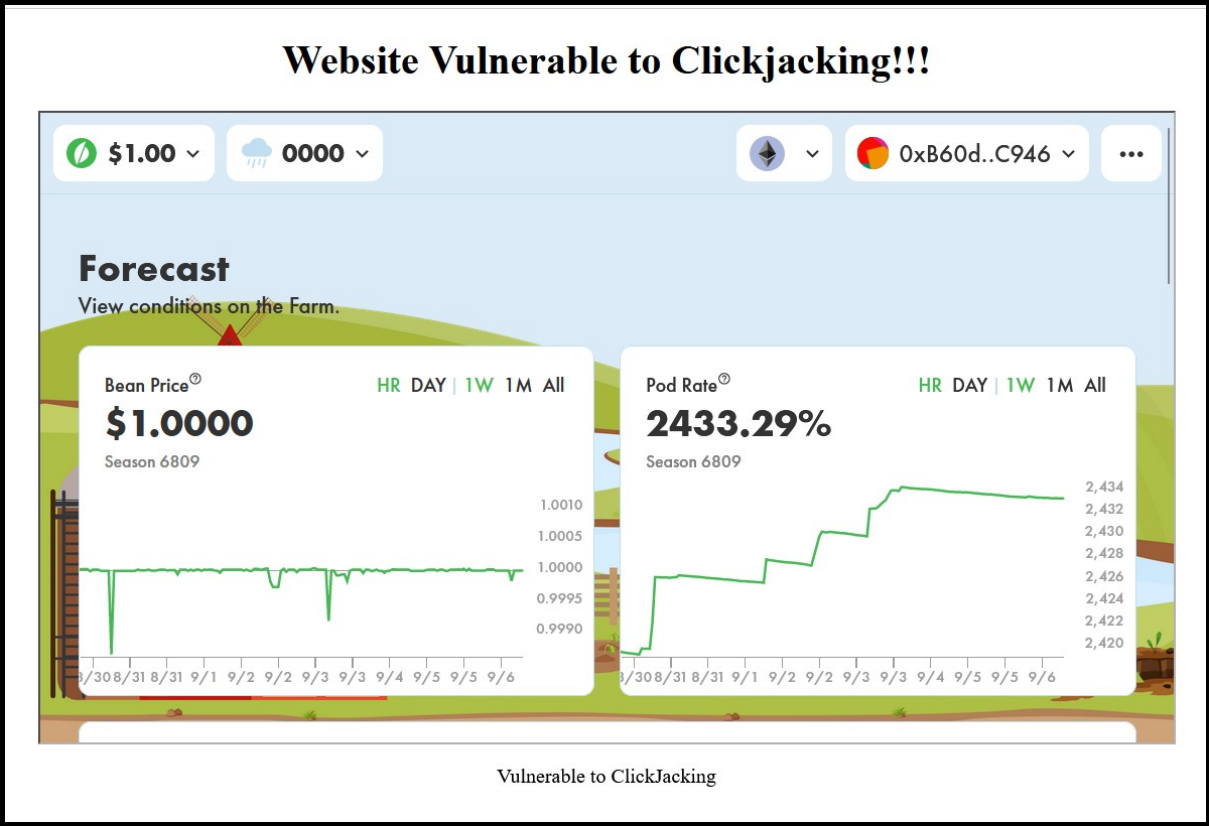


Figure 2: https://Beanstalk-ui-audit-b5305c.netlify.app/

```
box@box-vm:~$ curl -I https://beanstalk-ui-audit-b5305c.netlify.app/
HTTP/2 200
age: 0
cache-control: public, max-age=0, must-revalidate
content-type: text/html; charset=UTF-8
date: Tue, 20 Sep 2022 09:38:26 GMT
etag: "16d1ce8d1373d1f0655d62fc6f5de90a-ssl"
server: Netlify
strict-transport-security: max-age=31536000; includeSubDomains; preload
x-nf-request-id: 01GDD46P6RADHQY0M6YNX6T3F5
content-length: 1594
```

Figure 3: https://Beanstalk-ui-audit-b5305c.netlify.app/

Risk Level:

Likelihood - 2

Impact - 2

Recommendation:

It is recommended to set the X-Frame-Options header to DENY OR SAMEORIGIN to avoid this. The SAMEORIGIN value to allow framing only by pages on the same origin as the response itself. Note that the SAMEORIGIN header can be partially bypassed if the application itself can be made to frame untrusted websites. In addition, it is recommended to define X-Content-Type-Options =nosniff, Content-Security-Policy and X-Frame-Options response headers with the appropriate policies.

Reference:

X-Frame-Options
X-Content-Type-Options
Content Security Policy
Content-Type
Clickjacking Defense - OWASP

Remediation Plan:

PARTIALLY SOLVED: The Beanstalk team partially solved this issue by adding the appropriate headers and respective policies. The other reported header will require further testing and will be fixed later.

3.4 (HAL-04) OUTDATED SOFTWARE IN-USE - INFORMATIONAL

Description:

The version of React was out of date. Since this is an outdated version of the software, it may be vulnerable to attacks in the future.

Proof of concept:

Figure 4: https://Beanstalk-ui-audit-b5305c.netlify.app/

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

As this vulnerability has no direct impact on the application, but as security best practices, it is still recommended to update to the latest version.

Reference:

React Latest Releases

Remediation Plan:

SOLVED: The Beanstalk team fixed this issue by upgrading to the latest version 18.2.0.

3.5 (HAL-05) HARCODED SECRETS - INFORMATIONAL

Description:

Hardcoded API keys were found for development and production Beanstalk-UI repository. 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 credentials. In case of any breach or potential access to the source code, it could lead to further exploitation of the infrastructure.

Proof of concept:

.env.development

.env.production

Risk Level:

Likelihood - 1
Impact - 1

Recommendation:

It is recommended to follow security best practices and use environment variables to store secrets, as well as adding sensitive files in .gitignore.

Reference:

GitHub Encrypted Secrets

Remediation Plan:

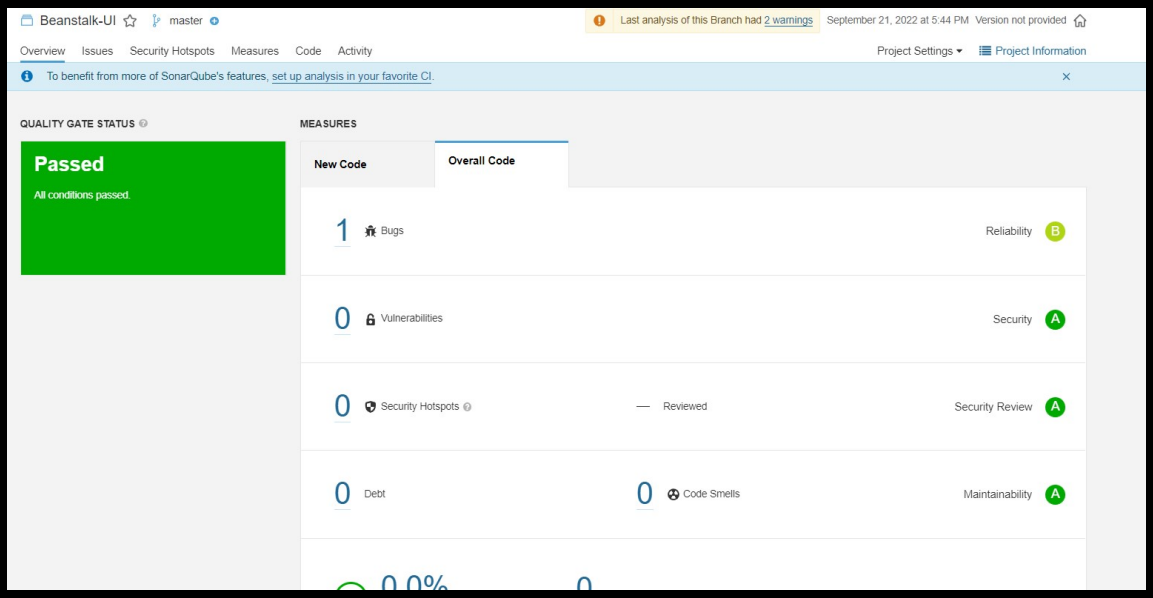
SOLVED: The Beanstalk team fixed this issue by adding a whitelist approach on these keys.

AUTOMATED TESTING

4.1 Description

Halborn used automated testing techniques to improve the coverage of certain areas of the scope repositories. sonarQube was used to analyze and find security issues. This tool used to help with detection of well-known security issues, and to identify low-hanging fruits of this engagement at hand.

SonarQube Results:



- No vulnerabilities were found by sonarQube.

THANK YOU FOR CHOOSING

// HALBORN