

Security Audit of app.vaulty.fi (<http://app.vaulty.fi>)

Conclusion



Audit was done by the "Web3Go" team <https://web3go.tech/> (<https://web3go.tech/>)
by Vladimir Smelov vladimirfol@gmail.com (<mailto:vladimirfol@gmail.com>)
<https://www.linkedin.com/in/vladimir-smelov-25021669/> (<https://www.linkedin.com/in/vladimir-smelov-25021669/>).

In the final contract were not found:

- Backdoors for investor funds withdrawal by anyone.
- Bugs allowing to steal money from the contract.
- Other security problems.

Obvious errors or backdoors were not found in the contract.

The client was acknowledged about all security notes below.



Scope

<https://app.vaulty.fi/> (<https://app.vaulty.fi/>).

20AUG2021

Methodology

1. Make a list of all the WebApp endpoints.
2. Make the structure of the WebApp and find out places where the data in the database changed (if exists), probably the API only gives views and caches and all work with SmartContract is done directly.

3. Research opportunities to DDOS the WebApp.
4. Check that admin panel is hidden and access is restricted.
5. Make a list of versions of nginx, db, admin panel and find exploits for them.
6. Push every button on the WebApp and check that everything works as expected. Check the data transfered to and from the server.
7. Try to access prohibited pages on the WebSite (if exists).
8. Scan ports.
9. Discuss recent privacy restrictions such as the GDPR coming into play
10. Discuss anomaly detection system.

Result

1) The Application endpoints

- <https://api.trongrid.io/wallet/getnodeinfo> (<https://api.trongrid.io/wallet/getnodeinfo>).
- <https://sun.tronex.io/wallet/getnodeinfo> (<https://sun.tronex.io/wallet/getnodeinfo>).
get info about blockchain node
- <https://55vvs1ddm4.execute-api.eu-central-1.amazonaws.com/default/getVaults>
(<https://55vvs1ddm4.execute-api.eu-central-1.amazonaws.com/default/getVaults>).
get cached info about vault
- <https://www.google-analytics.com/j/collect> (<https://www.google-analytics.com/j/collect>).
marketing statistics stuff

2) Make the structure of the WebApp

It's a single-page WebApp to access functionality of Vaulty Smart-Contract.

It takes the cache snapshot from the backend with information about reward pools and vaults and provide GUI to access SmartContract methods through one of web3 provider.

3) Research opportunities to DDoS the WebApp.

The WebSite is hosted on the AWS, so it requires big and complex botnet network to do a DDoS. We have contacted several hacker groups in DarkWeb and they all said that the cost to keep such page under DDoS attack is about ~1500\$/day.

Maybe it is related to some new deployments. But sometimes the WebSite is not accessible for plenty of minutes.

4) Check that admin panel is hidden and access is restricted.

Admin panel is not found.

5) Make a list of versions of nginx, db, admin panel and find exploits for them.

No access to such services found.

6) Push every button on the WebApp and check that everything works as expected. Check the data transfered to and from the server.

Staking and claiming works.

No data is transfered between backend and webapp, because all interaction with the smartcontract is done directly via web3 provider (metamask in my case).

7) Try to access prohibited pages on the WebSite (if exists).

No such pages found.

8) Scan ports.

55vvs1ddm4.execute-api.eu-central-1.amazonaws.com (<http://55vvs1ddm4.execute-api.eu-central-1.amazonaws.com>),

18.185.198.242

only 443 port is open (HTTPS)

app.vaulty.fi (<http://app.vaulty.fi>).

server-99-86-253-103.lhr3.r.cloudfront.net (<http://server-99-86-253-103.lhr3.r.cloudfront.net>).

99.86.253.103

port 80 open http Amazon CloudFront httpd

port 443 open https Amazon CloudFront httpd

9) Discuss recent privacy restrictions such as the GDPR coming into play

app.vaulty.fi (<http://app.vaulty.fi>) does not store cookies.

But cookies are collected by vaulty.fi (<http://vaulty.fi>).

COOKIE MANAGER V2.1 BETA

Domain: .vaulty.fi

Path: /

Name: _ga_66KRMFJR62

Store ID: 0

Value: GS1.1.1630587944.4.1.1630592364.0

Expires: 09/02/2022 05:19 PM

Same Site: Set to none, lax, strict or leave empty

☐ Session

☐ Host-Only

☐ Http-Only

☐ Secure

Set / Create New

4737 cookies from 1239 domains

it's better to add warning about using the cookies to be GDPR compliant.

10) Discuss anomaly detection system.

Too frequent requests should be filtered out with amazon by itself.

No POST requests are send, that's why anomaly detection system is not required.

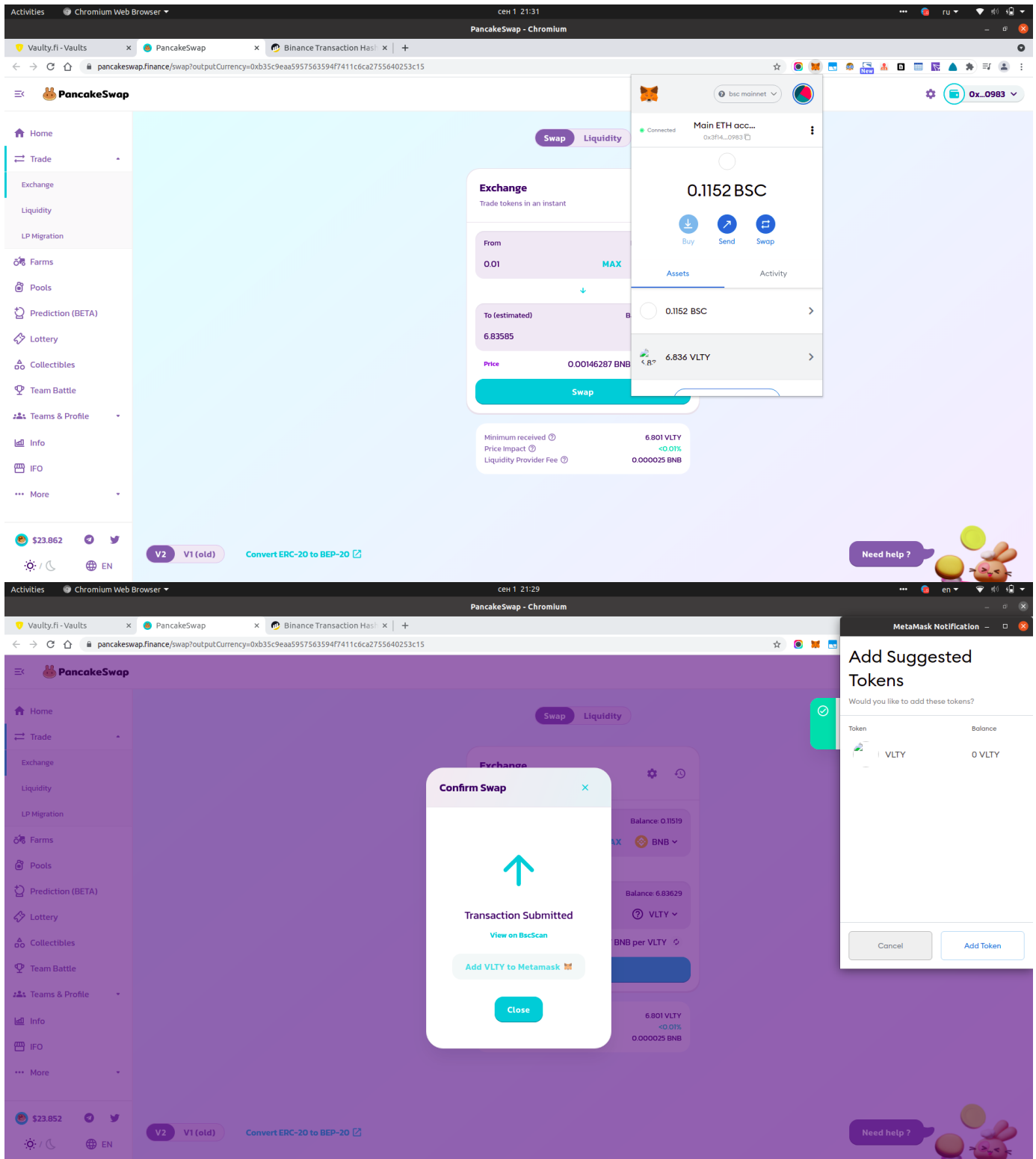
But you can think about usage of Amazon CloudWatch alarms.

Other notes.

1. VLTY token icon does not load.

When add VLTY token to MetaMask

See screenshots:



2. Event disappears before transaction confirmed.

It's confusing that when I do any transaction I see the notification in the bottom left side of the page but it disappears before the transaction is confirmed.

3. Visible x-api-key.

Just to confirm x-api-key is visible.

The other Amazon auth method is also possible and provide smaller possibilities for an attacker.

getVaults

getnodeinfo

getnodeinfo

getnodeinfo

5 / 48 requests

86.9 kB / 1.7 MB transferred

85.7 kB / 6.4 MB resources

Referrer Policy: strict-origin-when-cross-origin

▼ Response Headers

access-control-allow-headers: Content-Type, X-API-Key, Authorisation, *

access-control-allow-methods: OPTIONS,GET

access-control-allow-origin: *

content-length: 33522

content-type: application/json

date: Thu, 02 Sep 2021 14:34:16 GMT

x-amz-apigw-id: FCgUQECIliAFc-w=

x-amzn-requestid: 1a9ddf63-2532-4752-bcd9-c7b0e314db4a

x-amzn-trace-id: Root=1-6130e0e7-1ceeb9a37d8c14c27bf38704;Sampled=0

▼ Request Headers

:authority: 55vvs1ddm4.execute-api.eu-central-1.amazonaws.com

:method: GET

:path: /default/getVaults

:scheme: https

accept: application/json, text/plain, */*

accept-encoding: gzip, deflate, br

accept-language: en-US,en;q=0.9,ru-RU;q=0.8,ru;q=0.7

origin: https://app.vaulty.fi

referer: https://app.vaulty.fi/

sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="92"

sec-ch-ua-mobile: ?0

sec-fetch-dest: empty

sec-fetch-mode: cors

sec-fetch-site: cross-site

user-agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36

x-api-key: Zkrw8qxN29ai0qhPhSntF4iUmlwJ2wysaYED6QRC

Automated tool report

https://hackmd.io/oEqWOvPNSLC-OH-XivghCw?view

6/6

Vulnerability Scan Results

Summary

Overall risk level:

Low

Risk ratings:

High:

0

Medium:

0

Low:

8

Info:

66

Scan information:

This is an aggregated report from 7 scans.

Start time: 2021-09-02 17:09:53 UTC+03

Finish time: 2021-09-02 17:34:23 UTC+03

Findings

1. Target: <https://55vvs1ddm4.execute-api.eu-central-1.amazonaws.com/default/getVaults>

Website is accessible.

Nothing was found for website technologies.

Nothing was found for vulnerabilities of server-side software.

Nothing was found for client access policies.

Nothing was found for robots.txt file.

Nothing was found for outdated JavaScript libraries.

Nothing was found for use of untrusted certificates.

Nothing was found for administration consoles.

Nothing was found for software identification.

Nothing was found for information disclosure.

Nothing was found for sensitive files.

Nothing was found for interesting files.

Spider results: 0 dynamic URLs of total 1 URLs crawled

Nothing was found for Cross-Site Scripting.

Nothing was found for SQL Injection.

🚩 Nothing was found for File Inclusion.

🚩 Nothing was found for OS Command Injection.

🚩 Nothing was found for password autocomplete in browser.

🚩 Nothing was found for Secure flag of cookie..

🚩 Nothing was found for incomplete or no Cache-Control and Pragma HTTP header set.

🚩 Nothing was found for script passive scan rules.

🚩 Nothing was found for private IP disclosure.

🚩 Nothing was found for mixed content.

🚩 Nothing was found for session ID in URL rewrite.

🚩 Nothing was found for application error disclosure.

🚩 Nothing was found for missing HTTP header - X-Content-Type.

🚩 Nothing was found for stats passive scan rules.

🚩 Nothing was found for HttpOnly flag of cookie..

🚩 Nothing was found for missing HTTP header - X-XSS-Protection.

🚩 Nothing was found for domain too loose set for cookies.

🚩 Nothing was found for missing HTTP header - X-Frame-Options.

🚩 Nothing was found for missing HTTP header - Content Security Policy.

🚩 Nothing was found for missing HTTP header - Strict-Transport-Security.

🚩 Nothing was found for missing HTTP header - Referrer.

🚩 Nothing was found for directory listing.

🚩 Nothing was found for secure communication.

🚩 Nothing was found for missing HTTP header - X-Content-Type-Options.

2. Target: <https://vaulty.fi/>

Missing security header: Strict-Transport-Security

URL	Evidence
https://vaulty.fi/	Response headers do not include the HTTP Strict-Transport-Security header

Details

Risk description:

The HTTP Strict-Transport-Security header instructs the browser to initiate only secure (HTTPS) connections to the web server and deny any unencrypted HTTP connection attempts. Lack of this header permits an attacker to force a victim user to initiate a clear-text HTTP connection to the server, thus opening the possibility to eavesdrop on the network traffic and extract sensitive information (e.g. session cookies).

Recommendation:

The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

`Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]`

The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check.

The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

Classification:

CWE : [CWE-693](#)

OWASP Top 10 - 2013 : [A5 - Security Misconfiguration](#)

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

Missing security header: Content-Security-Policy

URL	Evidence
https://vaulty.fi/	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:

The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). 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.

Read more about CSP:

https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy>

Classification:

CWE : [CWE-693](#)

OWASP Top 10 - 2013 : [A5 - Security Misconfiguration](#)

OWASP Top 10 - 2017 : [A6 - Security Misconfiguration](#)

Missing security header: X-Frame-Options

URL	Evidence
https://vaulty.fi/	Response headers do not include the HTTP X-Frame-Options security header

Details

Risk description:

Because the `X-Frame-Options` header is not sent by the server, an attacker could embed this website into an iframe of a third party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user's consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:

<https://owasp.org/www-community/attacks/Clickjacking>

Recommendation:

We recommend you to add the **X-Frame-Options** HTTP header with the values **DENY** or **SAMEORIGIN** to every page that you want to be protected against Clickjacking attacks.

More information about this issue:

https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

Classification:

CWE : **CWE-693**

OWASP Top 10 - 2013 : **A5 - Security Misconfiguration**

OWASP Top 10 - 2017 : **A6 - Security Misconfiguration**

Missing security header: X-XSS-Protection

URL	Evidence
https://vaulty.fi/	Response headers do not include the HTTP X-XSS-Protection security header

Details

Risk description:

The **X-XSS-Protection** HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

Recommendation:

We recommend setting the X-XSS-Protection header to **X-XSS-Protection: 1; mode=block**.

More information about this issue:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

Classification:

CWE : **CWE-693**

OWASP Top 10 - 2013 : **A5 - Security Misconfiguration**

OWASP Top 10 - 2017 : **A6 - Security Misconfiguration**

Missing security header: X-Content-Type-Options

URL	Evidence
https://vaulty.fi/	Response headers do not include the X-Content-Type-Options HTTP security header

Details

Risk description:

The HTTP header **X-Content-Type-Options** is addressed to the Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

Recommendation:

We recommend setting the X-Content-Type-Options header such as **X-Content-Type-Options: nosniff**.

More information about this issue:

<https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options>.

Classification:

CWE : **CWE-693**

OWASP Top 10 - 2013 : **A5 - Security Misconfiguration**

OWASP Top 10 - 2017 : **A6 - Security Misconfiguration**

Missing security header: Referrer-Policy

URL	Evidence
https://vaulty.fi/	Response headers do not include the Referrer-Policy HTTP security header

Details

Risk description:

The Referrer-Policy HTTP header controls how much referrer information the browser will send with each request originated from the current web application.

For instance, if a user visits the web page "http://example.com/pricing/" and it 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.

Recommendation:

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.

Read more:

https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns



Classification:

CWE : **CWE-693**

OWASP Top 10 - 2013 : **A5 - Security Misconfiguration**

OWASP Top 10 - 2017 : **A6 - Security Misconfiguration**

Server software and technology found

Software / Version	Category
 Amazon Cloudfront	CDN
 Amazon S3	Miscellaneous
 Google Analytics UA	Analytics
 Google Font API	Font Scripts
 Google Tag Manager	Tag Managers

Details

Risk description:

An attacker could use this information to mount specific attacks against the identified software type and version.

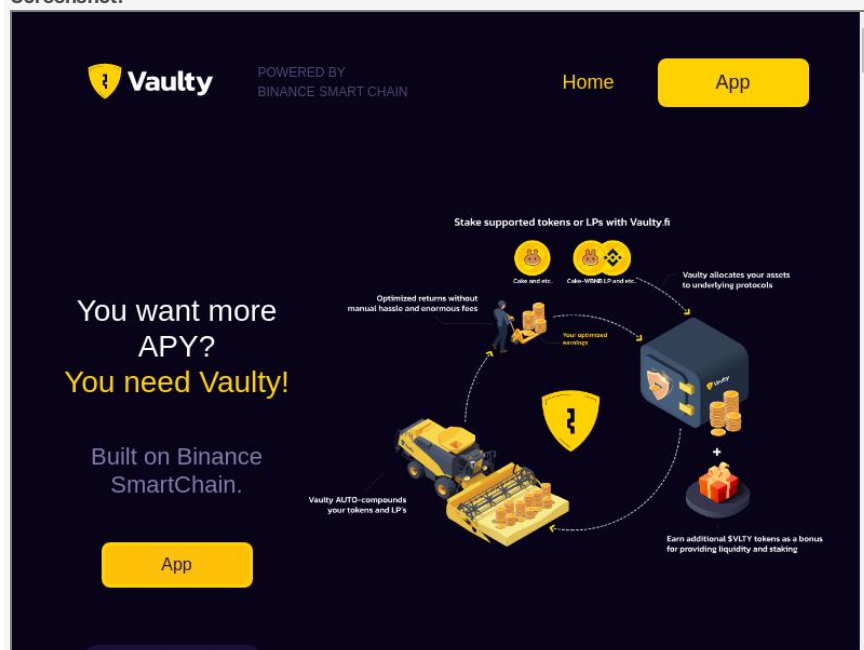
Recommendation:

We recommend you to eliminate the information which permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

More information about this issue:

https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html.

Screenshot:



Classification:

Incomplete or No Cache-control and Pragma HTTP Header Set

Affected items	Evidence
https://vaulty.fi/	no-cache, s-maxage=2

▼ Details

Risk description:

The cache-control and pragma HTTP header have not been set properly or are missing allowing the browser and proxies to cache content.

Recommendation:

Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate; and that the pragma HTTP header is set with no-cache.

https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html#web-content-caching

Classification:

CWE : [CWE-693](#)

OWASP Top 10 - 2013 : A5 - Security Misconfiguration

OWASP Top 10 - 2017 : A6 - Security Misconfiguration

 Website is accessible.

 Nothing was found for vulnerabilities of server-side software.

 Nothing was found for client access policies.

 Nothing was found for robots.txt file.

 Nothing was found for outdated JavaScript libraries.

 Nothing was found for use of untrusted certificates.

 Nothing was found for administration consoles.

 Nothing was found for software identification.

 Nothing was found for information disclosure.

 Nothing was found for sensitive files.

 Nothing was found for interesting files.

 Spider results: 0 dynamic URLs of total 5 URLs crawled

 Nothing was found for Cross-Site Scripting.

 Nothing was found for SQL Injection.

 Nothing was found for File Inclusion.

🚩 Nothing was found for OS Command Injection.

🚩 Nothing was found for password autocomplete in browser.

🚩 Nothing was found for Secure flag of cookie..

🚩 Nothing was found for script passive scan rules.

🚩 Nothing was found for private IP disclosure.

🚩 Nothing was found for mixed content.

🚩 Nothing was found for session ID in URL rewrite.

🚩 Nothing was found for application error disclosure.

🚩 Nothing was found for missing HTTP header - X-Content-Type.

🚩 Nothing was found for stats passive scan rules.

🚩 Nothing was found for HttpOnly flag of cookie..

🚩 Nothing was found for domain too loose set for cookies.

🚩 Nothing was found for directory listing.

🚩 Nothing was found for secure communication.

Discovery Scan Results

1. Sniper: Auto-Exploiter (https://app.vaulty.fi/)

No exploit was successful

[*] Running TCP port scan against app.vaulty.fi on ports 1-60000...

[-] No open ports found

[*] Fingerprinting http service https://app.vaulty.fi (port 443)...

[+] Fingerprint results:
Server type: AmazonS3
App title: Vaulty

[*] Looking for compatible exploits on port 443...

[+] Found 2 possible compatible exploits:
- Node.js Systeminformation Command Injection (CVE-2021-21315)
- FortiOS SSL VPN (CVE-2018-13379)

[*] Checking if https://app.vaulty.fi is vulnerable to Node.js Systeminformation Command Injection (CVE-2021-21315)...

[-] Target is not vulnerable

[*] Checking if https://app.vaulty.fi is vulnerable to FortiOS SSL VPN (CVE-2018-13379)...

[-] Target is not vulnerable

2. URL Fuzzer (https://app.vaulty.fi/)

Found 1 items

Name	HTTP Code	HTTP Reason	Page Size (KB)
robots.txt	200	OK	0.067

3. Whois Lookup (vaulty.fi)

Scan result

domain.....: vaulty.fi
status.....: Registered
created.....: 7.6.2021 13:33:59
expires.....: 7.6.2026 13:33:59
available.....: 7.7.2026 13:33:59
RegistryLock.....: no

Nameservers

nserver.....: ns-177-b.gandi.net [OK]
nserver.....: ns-80-a.gandi.net [OK]
nserver.....: ns-97-c.gandi.net [OK]

DNSSEC

dnssec.....: no

Holder

holder.....: Private person

Registrar

registrar.....: Gandi SAS
www.....: www.gandi.net

>>> Last update of WHOIS database: 2.9.2021 17:03:21 (EET) <<<

Copyright (c) Finnish Transport and Communications Agency Traficom

4. Find Subdomains (app.vaulty.fi)

Found 1 subdomains

Subdomain	IP address	OS	Server	Technology	Web Platform	Page Title
app.vaulty.fi	13.224.193.101		AmazonS3			Vaulty

Tool configuration details

The following tools were run to obtain the findings above:

- Whois Lookup – Find Domain Name, IP Address

Scan parameters

Target: vaulty.fi

Scan information

Start time: 2021-09-02 18:10:23 UTC+03
Finish time: 2021-09-02 18:10:25 UTC+03
Scan duration: 2 sec
Tests performed: 1/1
Scan status: Finished

- Sniper - Automatic Exploiter

Scan parameters

Target: https://app.vaulty.fi/
Ports to scan: 1-60000

Scan information

Start time: 2021-09-02 18:08:02 UTC+03
Finish time: 2021-09-02 18:08:05 UTC+03
Scan duration: 3 sec
Tests performed: 1/1
Scan status: Finished

- URL Fuzzer - Discover hidden files and directories

Scan parameters

URL: https://app.vaulty.fi/FUZZ
Method: GET
POST Data:
Fuzz types: Configuration files, Archives, Database files, Logs, Backups files, Web files
Custom extensions:
Options: Dynamic wordlist, Mutate found files
Wordlist: URL Fuzzer (default) (1266 words)
Custom headers:
Number of threads: 7
Request Timeout: 4
Delay between requests: 0
Maximum number of retries for a request: 3
Retry delays factor: 1
Force retry on HTTP codes:
Response filter: Auto
Match HTTP codes: All
Match response size: All
Match response content: All
Ignore HTTP codes: None
Ignore response size: None
Ignore response content: None

Scan information

Start time: 2021-09-02 17:35:00 UTC+03
Finish time: 2021-09-02 18:07:39 UTC+03
Scan duration: 32 min, 39 sec
Tests performed: 1/1
Scan status: Finished

- Website Vulnerability Scanner

Scan parameters

Website URL: https://55vvs1ddm4.execute-api.eu-central-1.amazonaws.com/default/getVaults
Scan type: Full
Authentication: False

Scan information

Start time: 2021-09-02 17:29:07 UTC+03
Finish time: 2021-09-02 17:34:23 UTC+03
Scan duration: 5 min, 16 sec
Tests performed: 37/37
Scan status: Finished

- Website Vulnerability Scanner

Scan parameters

Website URL: https://app.vaulty.fi/
Scan type: Full
Authentication: False

Scan information

Start time: 2021-09-02 17:26:59 UTC+03
Finish time: 2021-09-02 17:27:27 UTC+03
Scan duration: 28 sec
Tests performed: 12/13
Scan status: Started

- Find Subdomains

Scan parameters

Domain:	app.vaulty.fi
DNS records (NS, MX, TXT, AXFR):	On
DNS enumeration:	On
Certificate Transparency Logs:	On
External APIs:	On
Bing search:	On
Google search:	On
HTML links search :	On
SSL search:	On
Reverse DNS search:	On
Smart DNS search:	On
IP information:	False
Web technologies:	True

Scan information

Start time:	2021-09-02 17:25:02 UTC+03
Finish time:	2021-09-02 17:25:16 UTC+03
Scan duration:	14 sec
Tests performed:	1/1
Scan status:	Finished

- Website Vulnerability Scanner

Scan parameters

Website URL:	https://vaulty.fi/
Scan type:	Full
Authentication:	False

Scan information

Start time:	2021-09-02 17:09:53 UTC+03
Finish time:	2021-09-02 17:24:17 UTC+03
Scan duration:	14 min, 24 sec
Tests performed:	37/37
Scan status:	Finished