PENETRATION TESTING REPORT

White Box Penetration Test of Pera Web Wallet Application

VPQ-20220568

Pera Wallet, LDA

13th January 2023

VANTAGEPOINT

Security at the Speed of Development

CREST

TABLE OF CONTENTS

1. EXECUTIVE SUMMARY

OVERVIEW

Vantage Point Security Pte Ltd was engaged by Pera Wallet, LDA to conduct an independent security assessment of Pera Web Wallet to identify security vulnerabilities, weaknesses, and any instances of non-compliance to best practices or regulatory requirements. Testing commenced on the 16th of November 2022 and was completed on the 12th of December 2022. Testing was conducted by Vantage Point Security Singapore.

Penetration testing was conducted from the perspective of an authenticated and unauthenticated user on the Pera Web Wallet staging environment in a White Box approach. During this review there was complete information provided to Vantage Point consultants including complete source code prior to the project commencing.

White Box testing provides the highest degree of security assurance. White Box testing is conducted as both an authenticated user and as an unauthenticated user of the target system, and access to the complete applications source code is supplied.

Vantage Point performed this review by first modelling the software stack used by the Pera Web Wallet application against the required regulatory standards and security best practices. The unique technical and environmental conditions of the application were taken into consideration to generate a methodology and a list of specific test-cases that meet each item of the required regulatory standards. This review was conducted in accordance with the OWASP Web Application Testing Guide, and each item contained within was manually validated by Vantage Point consultants.

The assessment on Pera Web Wallet consist of source code review, network VAPT and web application penetration testing. While there is no major security finding discovered in the source code, an observational issue was raised pertaining to the use of cross domain script. The network assessment of the application found that TLS1.1 protocol and few other weak cipher suites were in used. It is recommended to disable weak protocol and cipher suites to further increase the security posture of the application.

On web application penetration testing, it was discovered that wallet password prompt can be bypassed to access the wallet information by modifying the Javascript used by the application. As one of the possible workarounds, it is advisable to encrypt wallet information before correct password is supplied.

The outcome of this penetration testing engagement is provided as a detailed technical report that provides the system's owners a full description of the vulnerabilities identified, the associated risk rating for each vulnerability, and detailed recommendations that will resolve the underlying technical issue.

VULNERABILITY OVERVIEW

Severity	Count	Open	Closed
Critical	**0**	**0**	**0**
High	**0**	**0**	**0**
Medium	**1**	**0**	**1**
Low	**1**	**0**	**1**
Observational	**3**	**0**	**3**
Summary	**5**	**0**	**5**

Vulnerability Risk Score

All vulnerabilities found by Vantage Point will receive and individual risk rating based on the following four categories. The risk calculation will be performed with the official CWE CVSS risk calculator.

CRITICAL COMPONENT RISK SCORE **CVSS score between 9.0 - 10.0**

Critical severity findings relate to an issue, which requires immediate attention and should be given the highest priority by the business as it will critically impact business interest critically.

HIGH COMPONENT RISK SCORE **CVSS score between 7.0 – 8.9**

HIGH severity findings relate to an issue, which requires immediate attention and should be given the highest priority by the business.

MEDIUM COMPONENT RISK SCORE **CVSS score between 4.0 - 6.9**

A MEDIUM severity finding relates to an issue, which has the potential to present a serious risk to the business.

LOW COMPONENT RISK SCORE **CVSS score between 0.0 – 3.9**

LOW severity findings contradict security best practice and have minimal impact on the project or business.

OBSERVATIONAL **CVSS score 0.0**

Observational findings relate primarily to non-compliance issues, security best practices or are considered an additional security feature that would increase the security stance of the environment. Observational findings do not have a CVSS Score.

2. PROJECT DETAILS

SCOPE

Application Name	Pera Web Wallet
Testing Start Date	16th November 2022
Testing Finish Date	12th December 2022
Target URL	https://staging.web.perawallet.app https://node-testnet.chain.perawallet.app https://testnet.staging.api.perawallet.app
SVN / GIT Revision Number	Pera-web-wallet: f87642b2c0f696fdd5dda8238af78f692a109a7f
Items Completed	Vantage Point completed the agreed Security assessment White Box VAPT of Web Application for Pera Wallet, including Source Code Review • https://staging.web.perawallet.app Grey Box VAPT of Pera Wallet APIs (4 API endpoints) – Let us know if there are any more other than the identified. • https://testnet.staging.api.perawallet.app/v1/accounts/multiple-overview/ • https://testnet.staging.api.perawallet.app/v1/accounts/search/ • https://testnet.staging.api.perawallet.app/v1/accounts/multiple-overview/?last_known_round=24631966 • https://testnet.staging.api.perawallet.app/v1/accounts/UCUCSLU24F4T46VLISZIC62XGO4ADJSIQF5R7L7JPTGGK2RHQIIED6HHNQ/assets/?include_algo=true • https://testnet.staging.api.perawallet.app/v1/currencies/USD/ Network VAPT for Pera Wallet Algorand Node (1 IP or if you have multiple instances for HA, let us know as well) • https://node-testnet.chain.perawallet.app

Component	Review Type	Status
Web Application	Penetration Testing	Completed
Web Application	Regression Testing	Completed

VERSION HISTORY

Date	Version	Release Name
13th December 2022	v0.1	Draft
13th December 2022	v0.2	QA Release
13th December 2022	v1.0	Final
13th January 2023	v1.1	Retest

3. Risk Assessment

This chapter contains an overview of the vulnerabilities discovered during the project. The vulnerabilities are sorted based on the CVSSv3 scoring categories CRITICAL, HIGH, MEDIUM and LOW. The category OBSERVATIONAL refers to vulnerabilities that have no risk score and therefore have no immediate impact on the system.

Overview of Components and Their Vulnerabilities

1. White Box Penetration Test of Pera Web Wallet Application		MEDIUM RISK 
1.1. Authentication Bypass	Closed	MEDIUM RISK 5.9 
1.2. SSL/TLS Cipher Issues - Weak Ciphers Supported	Closed	LOW RISK 3.7 
1.3. Local Database can be Duplicated	Closed	OBSERVATIONAL 
1.4. Misconfigured/Overly Permissive Cross-Origin Resource Sharing (CORS)	Closed	OBSERVATIONAL 
1.5. Use of Cross-Domain Script	Closed	OBSERVATIONAL 

4. Detailed Description of Vulnerabilities

2. White Box Penetration Test of Pera Web Wallet Application — MEDIUM RISK

1.1. Authentication Bypass

CVSSv3 Score: 5.9 — **MEDIUM RISK**

VULNERABILITY TRACKING

STATUS: **Closed**

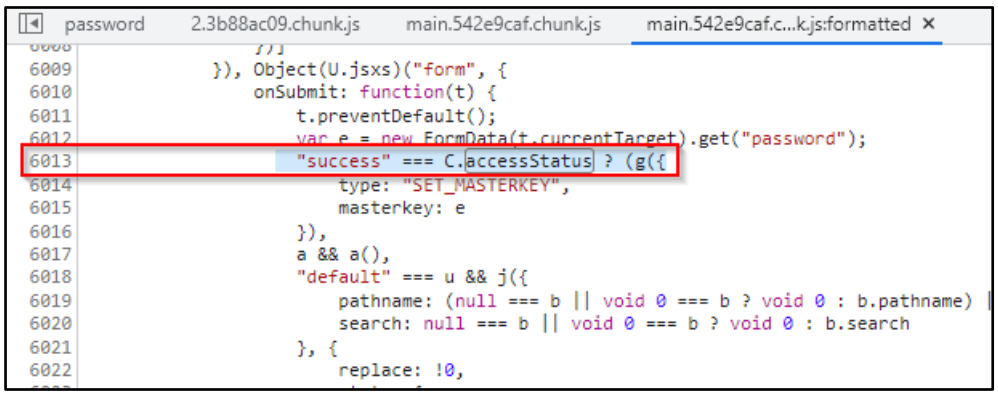
BACKGROUND

Authentication is the process of attempting to verify the digital identity of the sender of a communication. A common example is the log on process. It is a crucial part of an application as it protects the whole application from unauthorised access. The mechanism and flow of authentication must be robust and leave no application logic flaws.

However, the web wallet application relay entirely on client-side validation. Hence, it was possible to modify the application logic by changing the Javascript to bypass the login prompt.

DESCRIPTION

During the assessment, the Javascript used by the web wallet was analysed by setting up a breakpoint in web browser to learn and understand the application logic. It was found that the code below would allow user to access the wallet if the access status is 'success'.

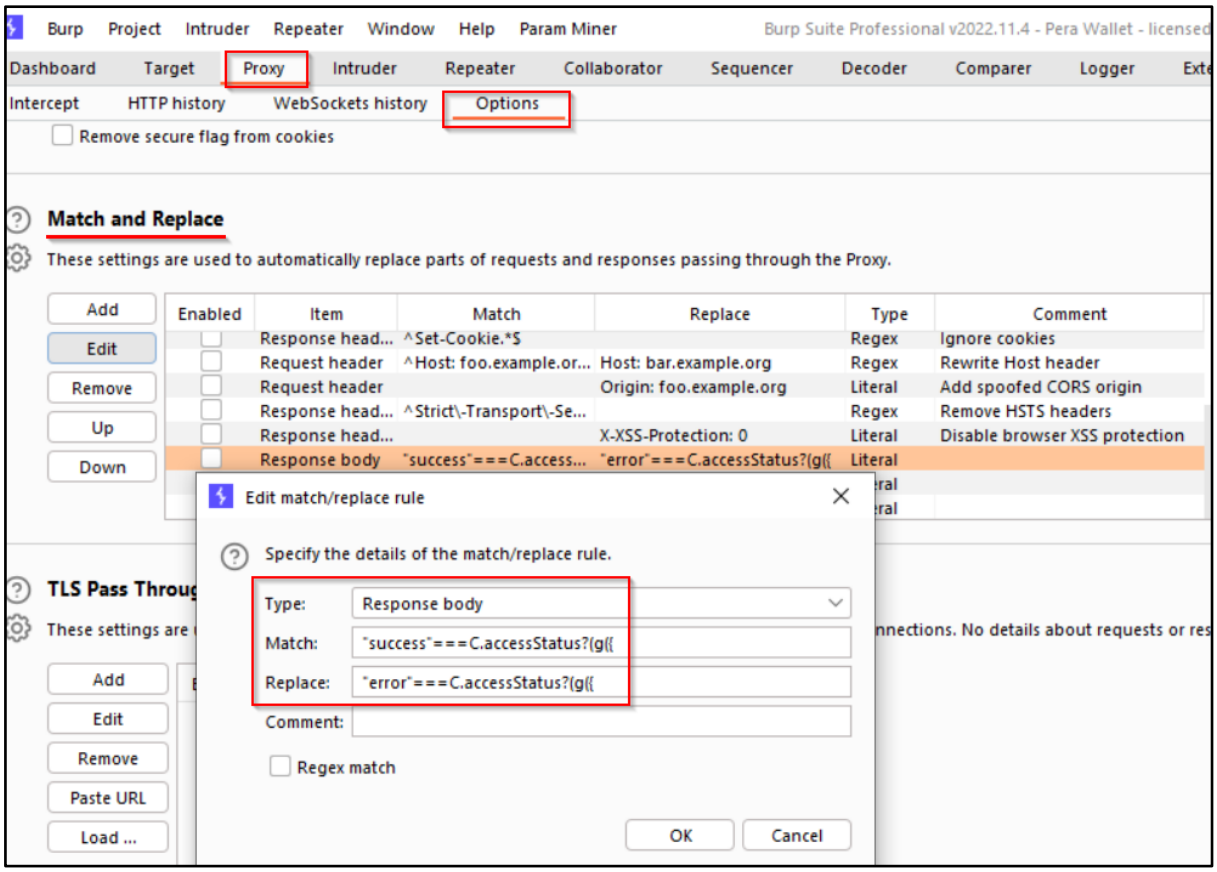


In order to get access status as 'success' user need to enter the correct password otherwise, the access status would be 'error'. This logic can be abused by changing the flow to allow user to access the wallet when access status is 'error'. In another words, allowing user to access the wallet by entering wrong password and doing so would render the password prompt useless. This could be achieved by setting up a HTTP proxy tool in between client (web browser) and server (application server) so the Javascript can be modified from the proxy tool before being passed to the web browser. To demonstrate:

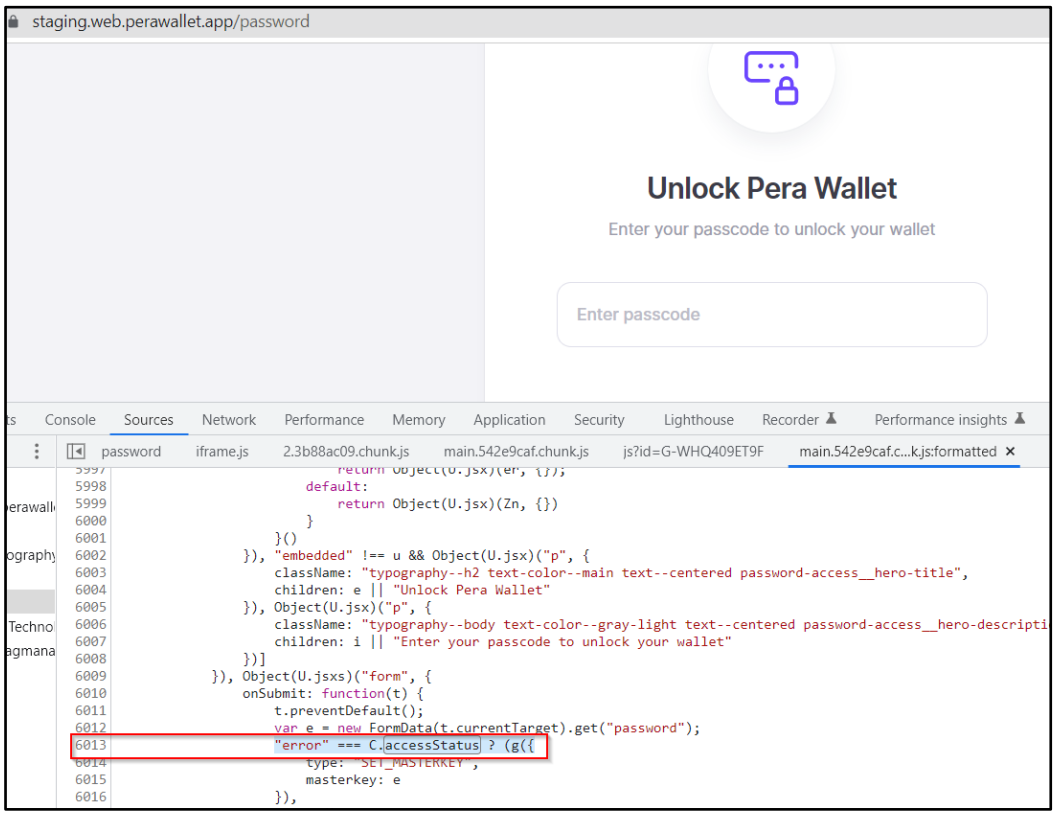
1. Enable proxy from the web browser in used (Chrome).

2. On proxy tool (Burp Suite), configure 'Match and Replace' rule to automatically modify the Javascript once it web browser request it.

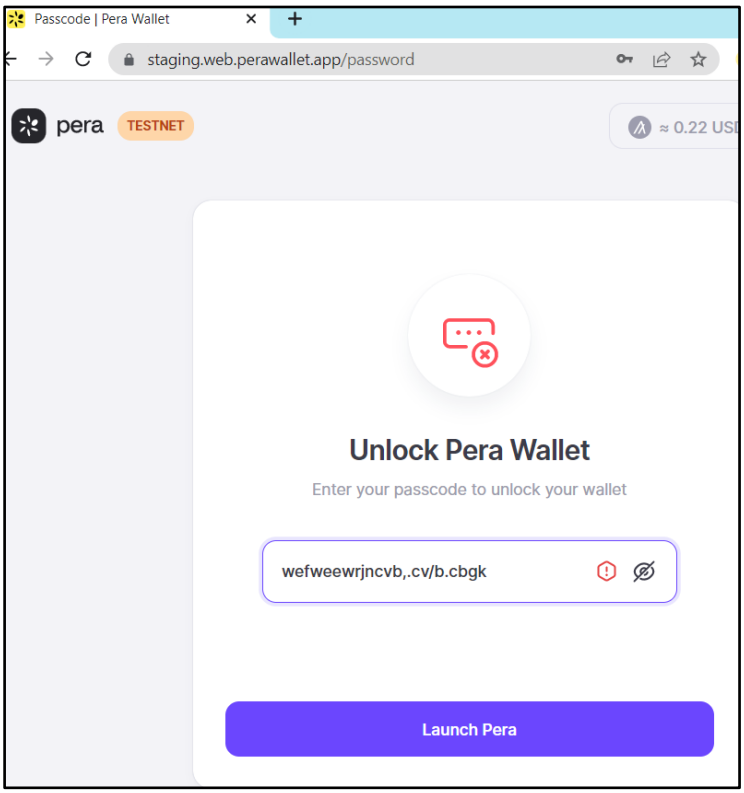
```
Match: "success"===C.accessStatus?(g({
Replace: "error"===C.accessStatus?(g({
```



3. Hard refresh the web wallet application from the browser and ensure that the script has been modified as shown below.

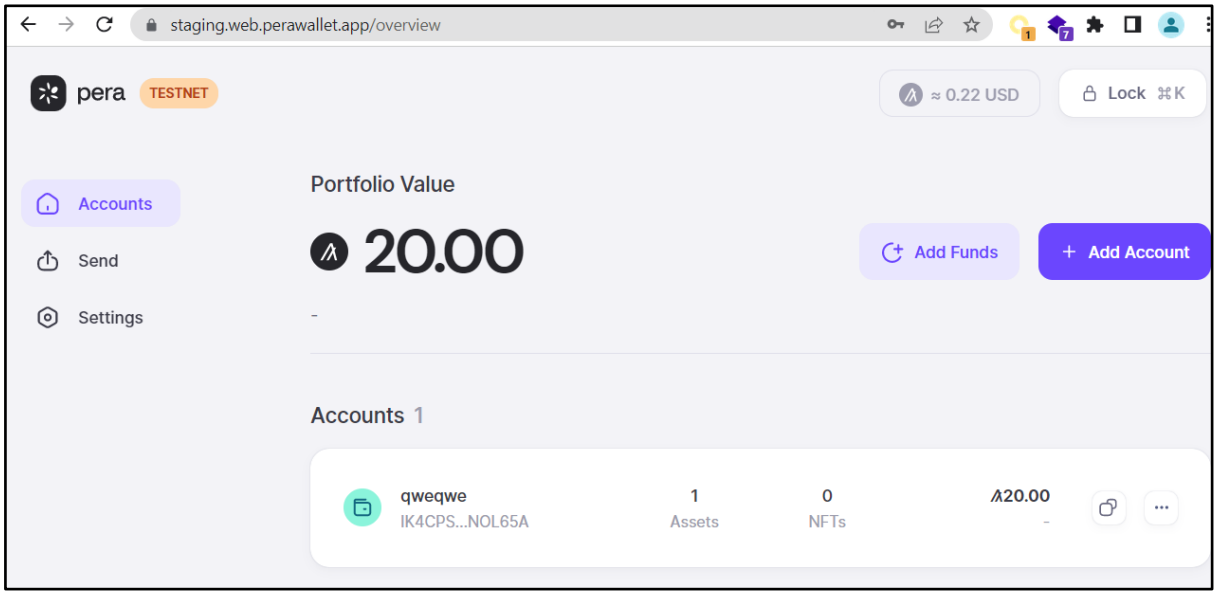


With the modified Javascript, it was possible to access the web wallet by entering any password



However, using this method, the only actions user can perform were limited to:

- Viewing wallet information
- Add Account
- Remove Account



RECOMMENDATION

It is advisable to encrypt wallet information such as wallet address, balance, hashedmasterkey and pk. And with the inserted password, the application performs the decryption using the user supplied password as the decryption key. If it is not the correct password, the decryption will fail, and wallet information cannot be retrieved. Only with correct password will the decryption be successful, and user can access the wallet application.

REGRESSION TESTING COMMENT

12th January 2023

It was observed that the login mechanism has been updated. The 'masterkey' value is taking in user supplied password and use that value to directly decrypt the encrypted sessions and accounts information.

```
async function syncIDB(masterkey: string) {
  if (hashPassword(masterkey) !== hashedMasterkey) {
    throw new Error("Incorrect password.");
  }

  const accounts = await appDBManager.decryptTableEntries(
    "accounts",
    masterkey
  )("address");
  const sessions = await appDBManager.decryptTableEntries("sessions", masterkey)("url");

  dispatchAppState({type: "SYNC_IDB", payload: {accounts, sessions}});
  dispatchAppState({type: "SET_MASTERKEY", masterkey});
}
```

If the supplied password is wrong, decryption will fail, and user will not be able to access their wallet. Since the login mechanism is no longer comparing the login status, it is not possible to bypass the login mechanism. Hence, the issue is resolved.

```
public decryptTableEntries<T extends keyof Model, K extends keyof Model[T]>(
  table: T,
  encryptionKey: string
): (primaryKey: K) => Promise<Model[T]> {
  return (primaryKey: K) =>
    new Promise(async (resolve, reject) => {
      try {
        const entries = await this.getAllValues(table, {isEncrypted: true});

        const decryptedEntries = await Promise.all(
          entries.map((entry) => decryptSK(entry, encryptionKey, {stringify: true}))
        );

        resolve(
          generateKeyMapFromArray(
            decryptedEntries.map((decryptedEntry) =>
              JSON.parse(decryptedEntry, (key, value) => {
                // encrypted values are JSON.stringified
                // reviver function is needed to parse stringified dates
                if (key === "date") {
                  return new Date(value);
                }

                return value;
              })
            ),
            primaryKey
          )
        );
      } catch (error) {
        reject(error);
      }
    });
}
```

CVSS RISK RATING

CVSS v3 Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:L

VULNERABILITY REFERENCES

CWE:
https://cwe.mitre.org/data/definitions/287.html

OWASP:
https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A2-Broken_Authentication

1.2. SSL/TLS Cipher Issues - Weak Ciphers Supported

CVSSv3 Score: 3.7 **LOW RISK** 

VULNERABILITY TRACKING

STATUS: **Closed**

BACKGROUND

Secure Socket Layer (SSL) or Transport Layer Security (TLS) are designed to secure the transfer of data between client and server through authentication, encryption, and integrity protection. To achieve that, SSL/TLS uses one or more cipher suites. A cipher suite is a combination of authentication, encryption, and message authentication code (MAC) algorithms. Insecure SSL/TLS cipher suite reduce the efficiency of the transport layer encryption and increase the risk of successful man-in-the-middle attacks against the affected services.

It was noted that the affected hosts had following weak SSL/TLS cipher suites enabled as highlighted below.

- Cipher suites with SHA1
- Cipher suites with CBC
- TLS 1.1 cipher suites

DESCRIPTION

Instance 1

Affected Host

- https://staging.web.perawallet.app
- https://testnet.staging.api.perawallet.app

Supported SSL Ciphers

```
* SSL 2.0 Cipher Suites:
 Attempted to connect using 7 cipher suites; the server rejected all cipher
suites.

* SSL 3.0 Cipher Suites:
 Attempted to connect using 80 cipher suites; the server rejected all cipher
suites.

* TLS 1.0 Cipher Suites:
 Attempted to connect using 80 cipher suites; the server rejected all cipher
suites.

* TLS 1.1 Cipher Suites:
 Attempted to connect using 80 cipher suites.
The server accepted the following 4 cipher suites:
 TLS_RSA_WITH_AES_256_CBC_SHA 256
```

```
 TLS_RSA_WITH_AES_128_CBC_SHA 128
 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA 128 ECDH: prime256v1 (256 bits)
The group of cipher suites supported by the server has the following properties:
 Forward Secrecy OK - Supported
 Legacy RC4 Algorithm OK - Not Supported

 * TLS 1.2 Cipher Suites:
 Attempted to connect using 156 cipher suites.
The server accepted the following 18 cipher suites:
 TLS_RSA_WITH_AES_256_GCM_SHA384 256
 TLS_RSA_WITH_AES_256_CBC_SHA256 256
 TLS_RSA_WITH_AES_256_CBC_SHA 256
 TLS_RSA_WITH_AES_128_GCM_SHA256 128
 TLS_RSA_WITH_AES_128_CBC_SHA256 128
 TLS_RSA_WITH_AES_128_CBC_SHA 128
 TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 256 ECDH: X25519 (253 bits)
 TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 128 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 128 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA 128 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256 256 ECDH: X25519 (253 bits)
 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 256 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 128 ECDH: prime256v1 (256 bits)
 TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 128 ECDH: prime256v1 (256 bits)
The group of cipher suites supported by the server has the following properties:
 Forward Secrecy OK - Supported
 Legacy RC4 Algorithm OK - Not Supported

 * TLS 1.3 Cipher Suites:
 Attempted to connect using 5 cipher suites.
The server accepted the following 3 cipher suites:
 TLS_CHACHA20_POLY1305_SHA256 256 ECDH: X25519 (253 bits)
 TLS_AES_256_GCM_SHA384 256 ECDH: X25519 (253 bits)
 TLS_AES_128_GCM_SHA256 128 ECDH: X25519 (253 bits)
```

RECOMMENDATION

Based on the identified insecure SSL/TLS cipher suites, it is highly recommended to disable the use of weak cipher suites. Preferred cipher suites that provide perfect forward secrecy are as follow:

- Configure TLS 1.2 to use Elliptic Curve Diffie-Hellman (ECDH) key exchange algorithm and DHE as a fallback algorithm. If possible, refrain from using RSA key exchange.
- Configure TLS 1.3 to use Ephemeral Diffie-Hellman (EDH or DHE) key exchange protocol.

REGRESSION TESTING COMMENT

12th January 2023

The issue has been resolved as TLS1.1 and weak SSL/TLS cipher suites have been removed. Below are the protocol and cipher suites used by the application.

```
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-12 10:07 Ulaanbaatar Standard Time
Nmap scan report for staging.web.perawallet.app (172.67.75.196)
Host is up (0.011s latency).
Other addresses for staging.web.perawallet.app (not scanned): 2606:4700:20::681a:dce 260
:4700:20::ac43:4bc4 2606:4700:20::681a:cce 104.26.13.206 104.26.12.206

PORT    STATE SERVICE  VERSION
443/tcp open  ssl/http Cloudflare http proxy
|_http-server-header: cloudflare
| ssl-enum-ciphers:
|   TLSv1.2:
|     ciphers:
|       TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256-draft (ecdh_x25519) - A
|       TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|     compressors:
|       NULL
|     cipher preference: server
|   TLSv1.3:
|     ciphers:
|       TLS_AKE_WITH_AES_128_GCM_SHA256 (ecdh_x25519) - A
|       TLS_AKE_WITH_AES_256_GCM_SHA384 (ecdh_x25519) - A
|       TLS_AKE_WITH_CHACHA20_POLY1305_SHA256 (ecdh_x25519) - A
|     cipher preference: client
|_  least strength: A
```

CVSS RISK RATING

CVSS v3 Vector: CVSS:3.0/AV:A/AC:H/PR:N/UI:R/S:U/C:L/I:L/A:N

VULNERABILITY REFERENCES

OWASP:
https://www.owasp.org/index.php/Testing_for_Weak_SSL/TLS_Ciphers,_Insufficient_Transport_Layer_Protection_%28OTG-CRYPST-001%29

1.3. Local Database can be Duplicated

OBSERVATIONAL

VULNERABILITY TRACKING

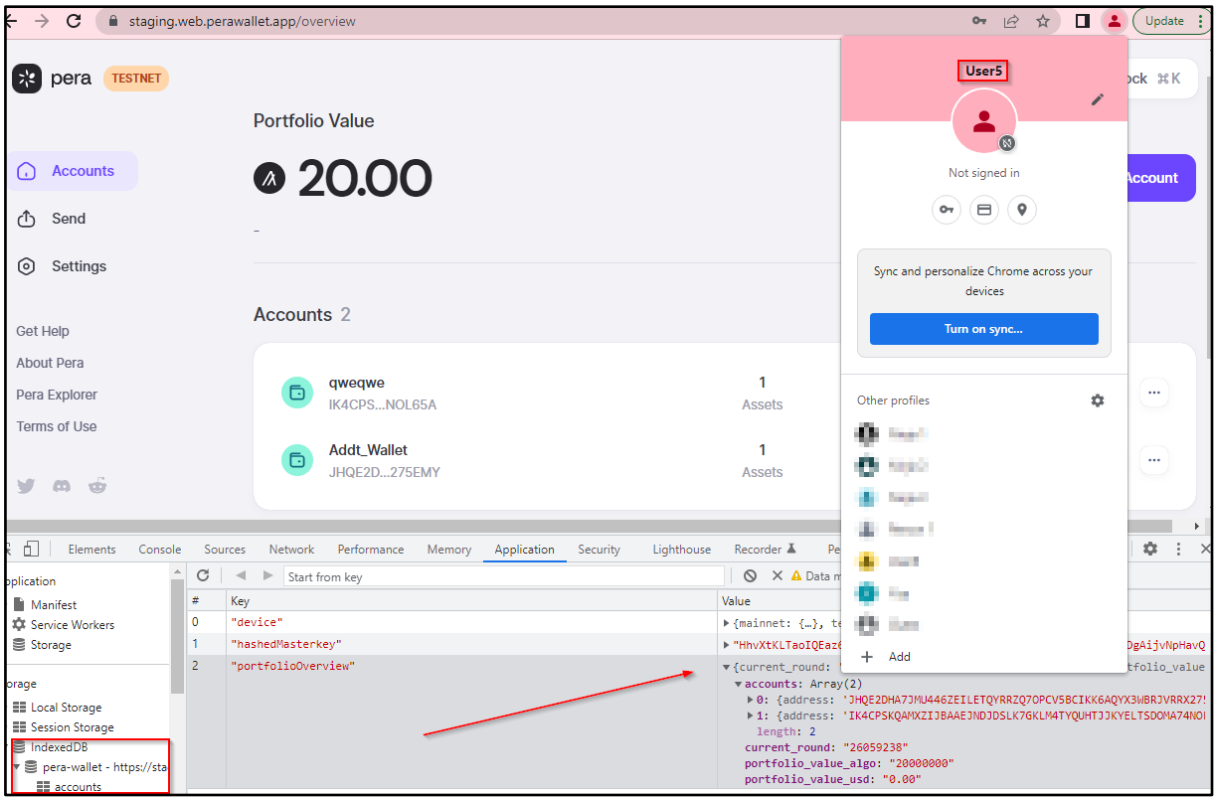
STATUS: **Closed**

BACKGROUND

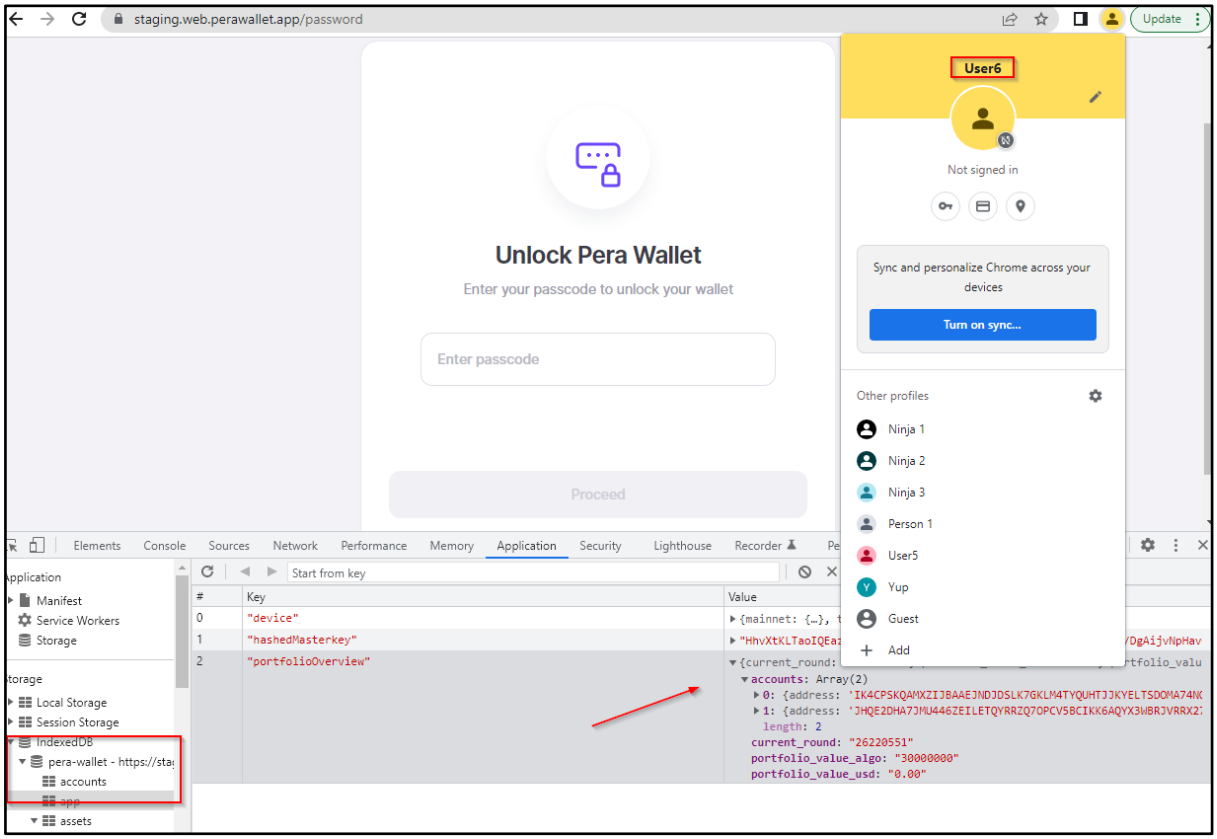
It was noted that the web wallet uses browser IndexedDB to store user's wallet information. However, browser IndexedDB directory can be copied to another browser and without entering any password, the web wallet information can be viewed by whoever copy the IndexedDB. In the event where malicious user has gotten access to victim's laptop, he/she would be able to steal the IndexedDB and access it from their browser.

DESCRIPTION

In Chrome browser, the location of IndexedDB for 'User 5' is stored in `C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Profile 5\IndexedDB\https_staging.web.perawallet.app_0.indexeddb.leveldb`



By copying the `https_staging.web.perawallet.app_0.indexeddb.leveldb` directory to new 'User 6' IndexedDB at `C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Profile 6\IndexedDB`, the wallet information can be viewed



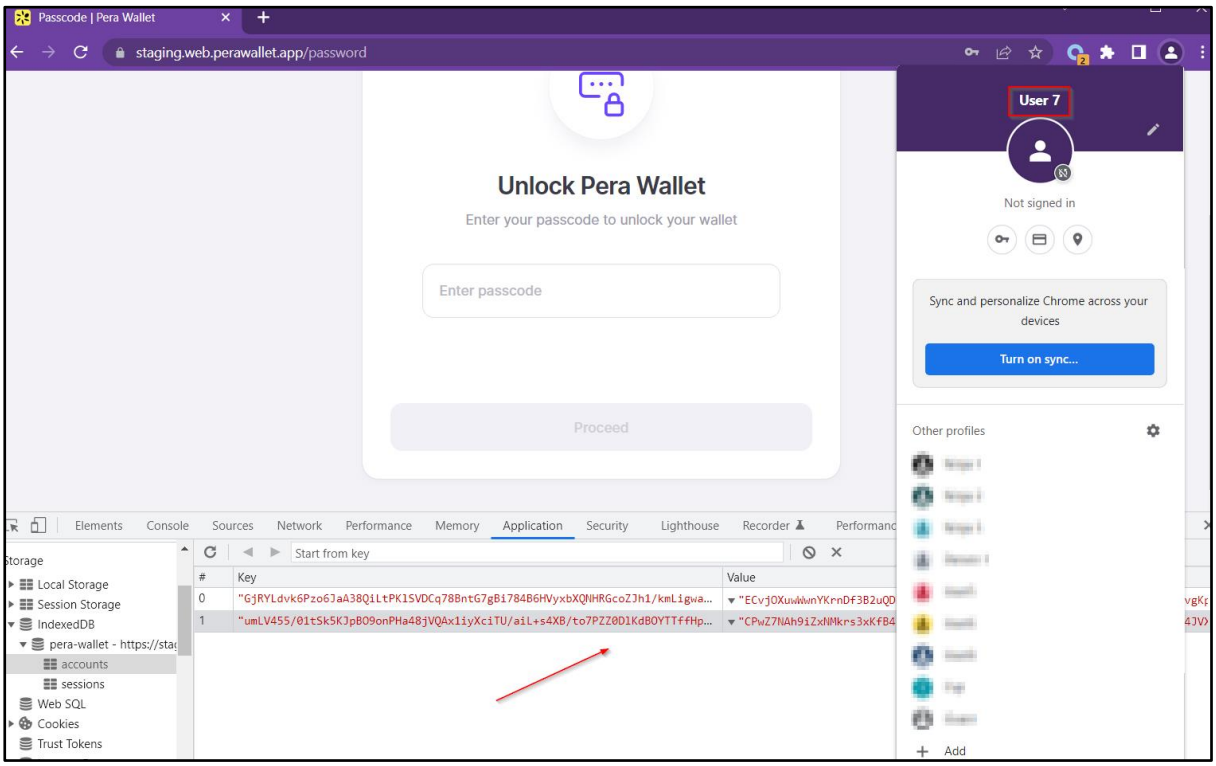
RECOMMENDATION

It is recommended to secure the IndexedDB by adding additional check if the IndexedDB belongs to the originated web browser. Otherwise, the web application should prompt user to create new wallet. Alternatively, encrypt all the data in IndexedDB and without entering the correct password, the data should not be in readable format. This method will slow down the attacker in getting the wallet information by just copying the IndexedDB directory.

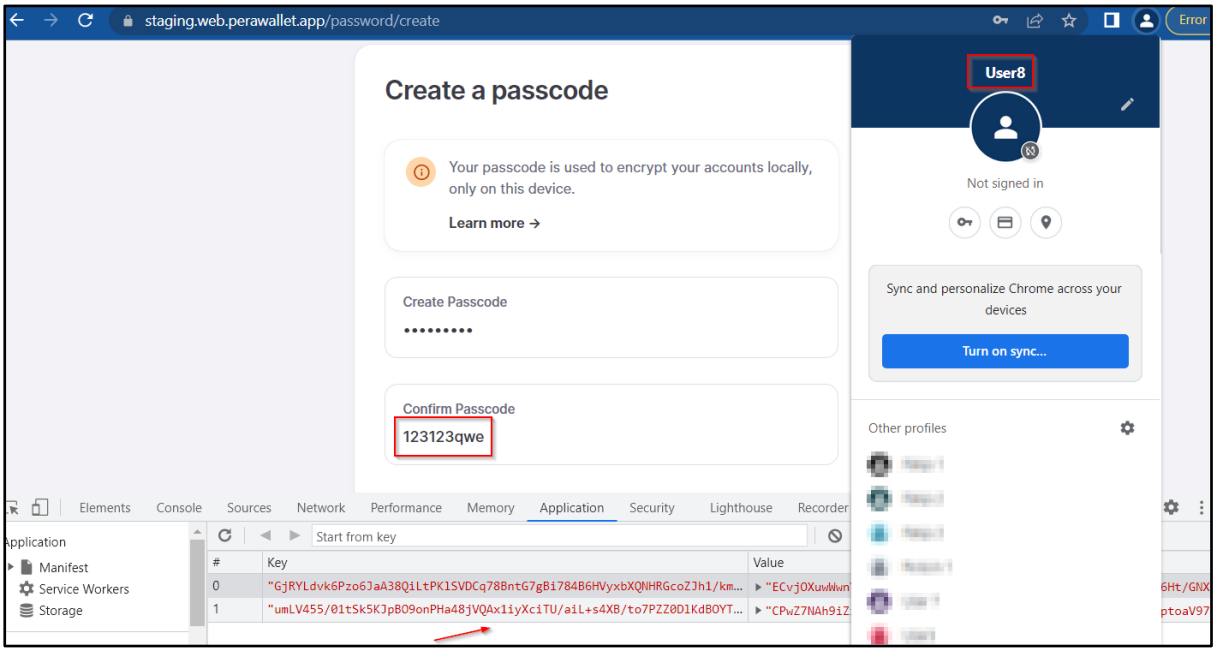
REGRESSION TESTING COMMENT

13th January 2022

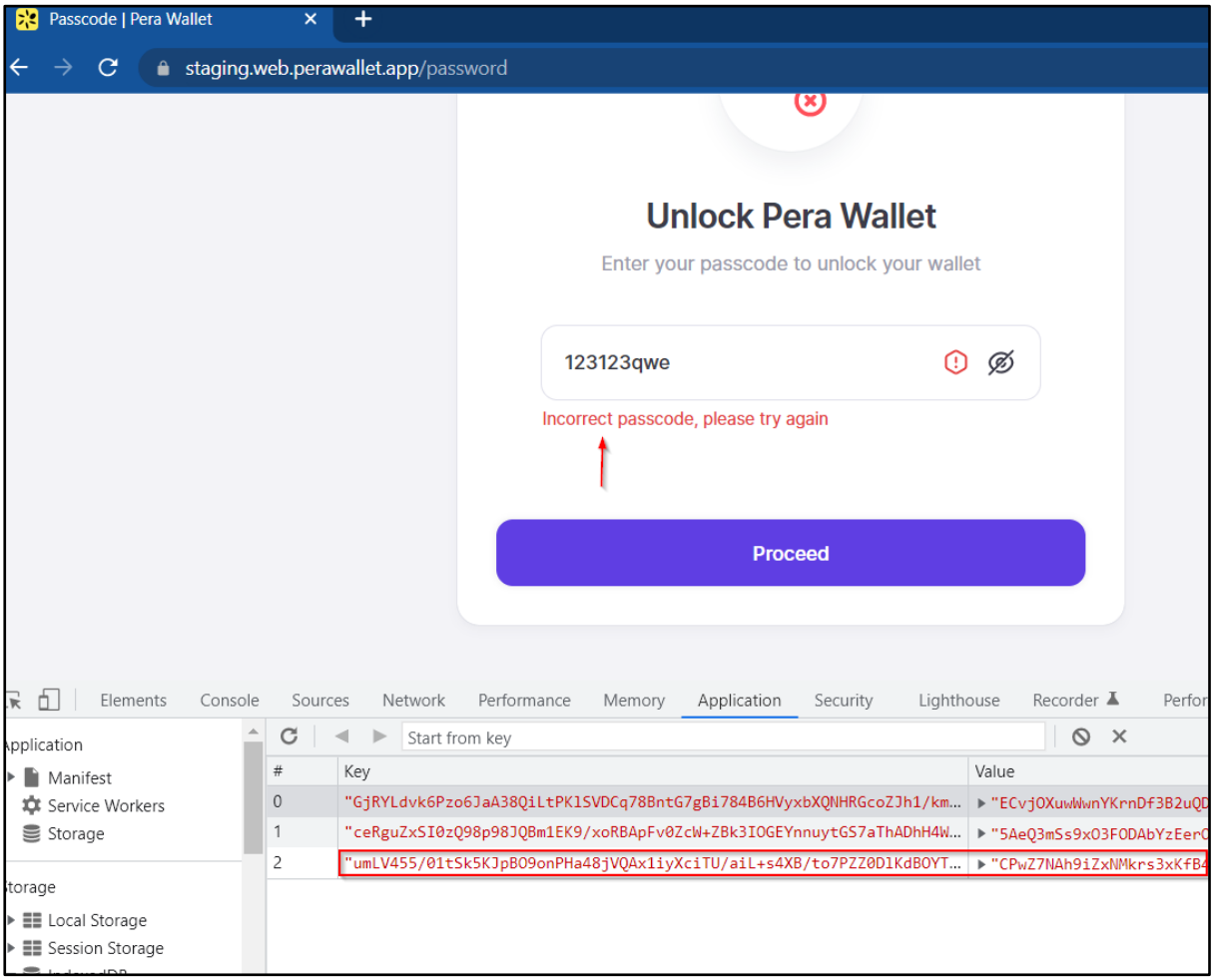
The issue has been fixed as it is no longer possible to view wallet information by copying IndexedDB file to different browser. Wallet information is encrypted in IndexedDB before and after login.



Copying the IndexedDB file from C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Profile 7\IndexedDB\https_staging.web.perawallet.app_0.indexeddb.leveldb to C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Profile 8\IndexedDB\https_staging.web.perawallet.app_0.indexeddb.leveldb will prompt the browser of User 8 to setup new passcode and create new account as shown below.



Notice the encrypted 'Key' and 'Value' of User 8 was the same as User 7. By creating new account with new passcode, the web wallet will add new account into the copied IndexedDB file of User 7. As a result, the web wallet for User 8 was unusable as 2 passcodes exist within the wallet. Entering the passcode set above '123123qwe' resulted in wrong passcode as shown below.



CVSS RISK RATING

CVSS v3 Vector: CVSS:3.0/AV:L/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

1.4. Misconfigured/Overly-Permissive Cross-Origin Resource Sharing (CORS)

OBSERVATIONAL

VULNERABILITY TRACKING

STATUS: **Closed**

BACKGROUND

The application was found to return the "Access-Control-Allow-Origin" header, which allows requests from unauthorised domains to subsequently view the responses that may contain confidential information.

In a typical Cross Site Request Forgery attack, the response cannot be viewed due to the request coming from a different domain. The "Access-Control-Allow-Origin" header determines which domains can overcome the same origin policy restrictions, enabling the retrieval of the user's confidential information and the ability to bypass CSRF tokens.

DESCRIPTION

It was observed both domains below allowed Cross-Origin request from any domains. It indicates that any domain can send a request and the server would return the data.

Domain: testnet.staging.api.perawallet.app

Request

Pretty Raw Hex

```
POST /v1/accounts/multiple-overview/ HTTP/2
Host: testnet.staging.api.perawallet.app
Content-Length: 137
Sec-Ch-Ua: "Google Chrome";v="107",
"Chromium";v="107", "Not=A?Brand";v="24"
Accept: application/json
Content-Type: application/json
Sec-Ch-Ua-Mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
 x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/107.0.0.0 Safari/537.36
X-Api-Key:
pera-web-staging-U-jZ3m-LR6-ed-7fLTmekDI0-95N5jU
X
Sec-Ch-Ua-Platform: "Windows"
Origin: https://malicious-site.com
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9

{
  "account_addresses":[
    "JKMB722PD6HMV5AXTKALXBXMWXVEZ743RRKTELJC6W3
    QNUVQJKJ5OVTGUY"
  ],
  "last_known_round":"25702529",
  "exclude_opt_ins":true
}
```

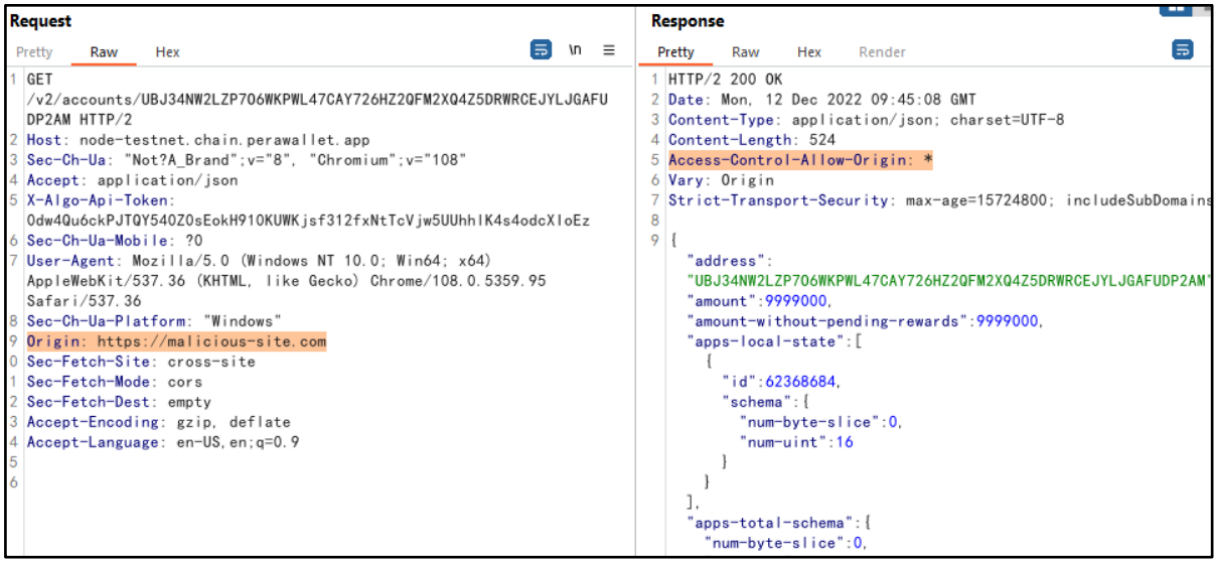
Response

Pretty Raw Hex Render

```
Content-Language: en
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: X-API-Key, algorand-network, accept, accept-enco
authorization, content-type, dnt, origin, user-agent, x-csrftoken, x-requested
x-device-auth-token
Access-Control-Max-Age: 3600
Access-Control-Allow-Methods: GET, POST, PUT, PATCH, DELETE, OPTIONS
Cf-Cache-Status: DYNAMIC
Report-To:
{"endpoints":[{"url":"https:\/\/a.nel.cloudflare.com\/report\/v3?s=iRe3hMqOeaP
yuC%2BZOS3%2FPPcl0OU16qWLTKZ01DVjZsHChKVRudyj7zdrpQ4dQGpzJuDzQWPuladrolBiOCXC5
RCIlJ1ym4QVLy6%2BDM9x4UDMJfyw9PdHS4FbMUiwMantcFKVrv8PwrPSRHI%3D"}],"group":"cf
max_age":604800}
Nel: {"success_fraction":0,"report_to":"cf-nel","max_age":604800}
Strict-Transport-Security: max-age=2592000; includeSubDomains; preload
Server: cloudflare
Cf-Ray: 7785886968916bb8-SIN

{
  "current_round":"26203886",
  "portfolio_value_usd":"0.00",
  "portfolio_value_algo":"21000000",
  "accounts":[
    {
      "address":"JKMB722PD6HMV5AXTKALXBXMWXVEZ743RRKTELJC6W3QNUVQJKJ5OVTGUY",
      "names":[
      ],
      "name":null,
      "total_usd_value":"0.00",
      "total_algo_value":"21000000",
      "standard_asset_count":1,
      "collectible_count":0
```

Domain: node-testnet.chain.perawallet.app



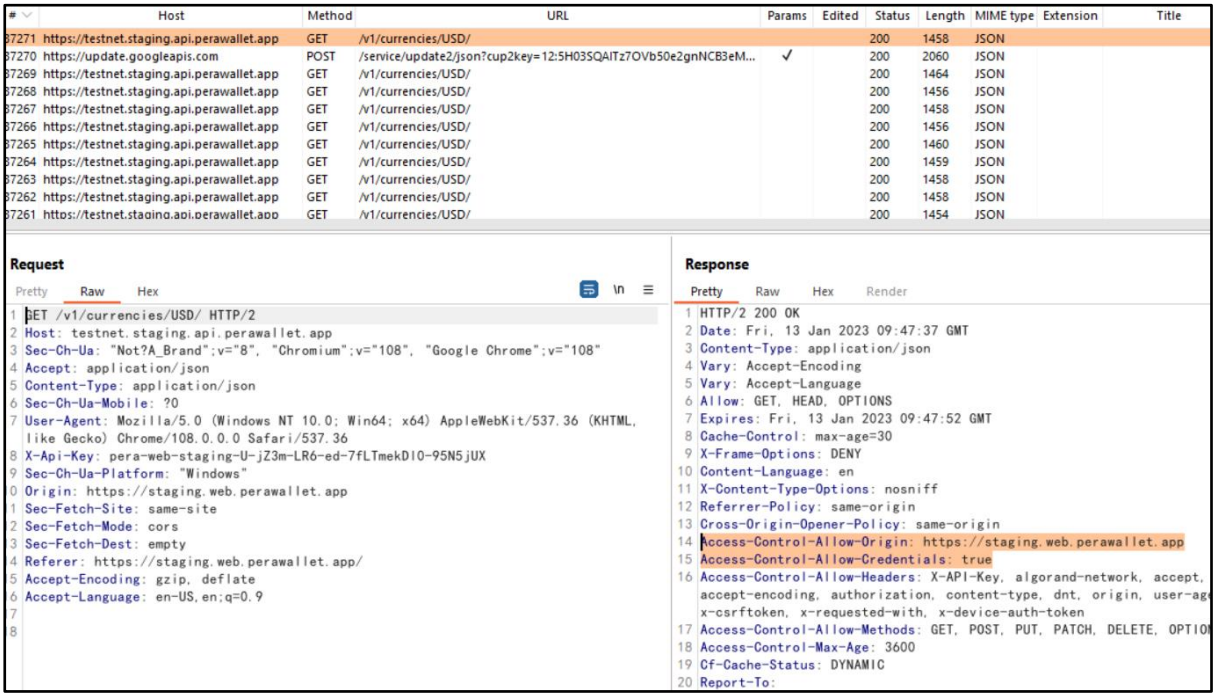
RECOMMENDATION

- Use the Access-Control-Allow-Origin header only on chosen URLs that need to be accessed Cross-Domain. Do not use the header for the entire domain. Allow only trusted domains in the Access-Control-Allow-Origin header.
- White-list domains and do not use the "*" wildcard nor blindly return the Origin header content without any validation.

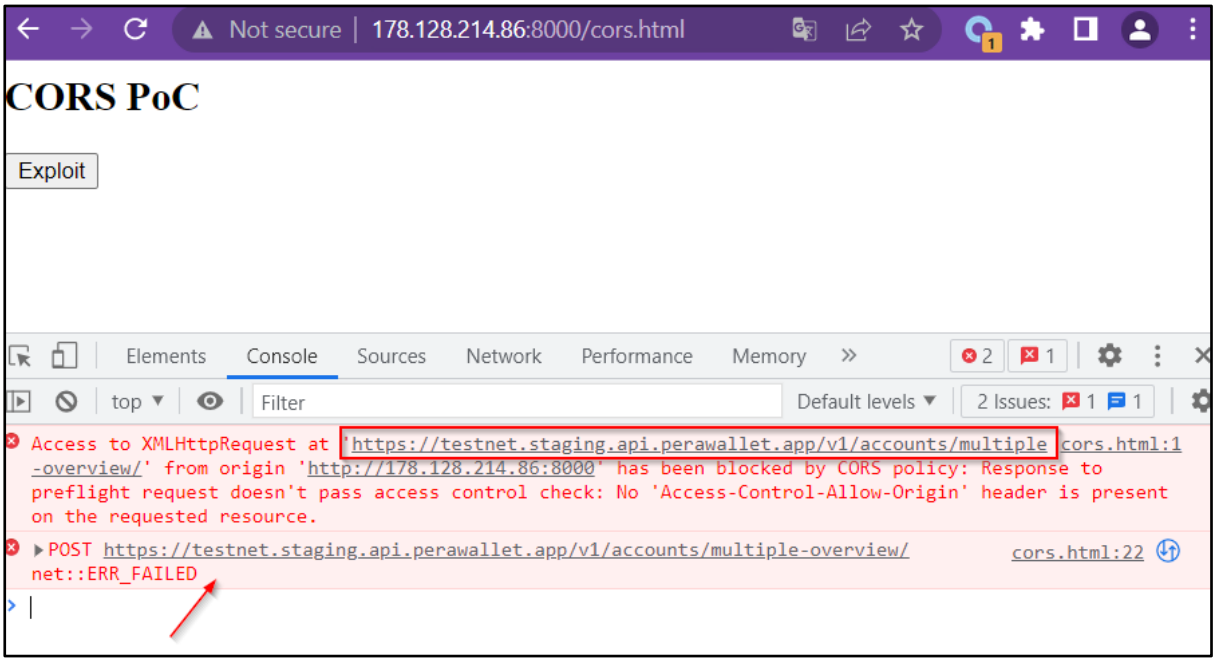
REGRESSION TESTING COMMENT

13th January 2023

The issue has been resolved as the 'Access-Control-Allow-Origin' header has been configured correctly as shown below.



Testing it via web browser returned following error.



CVSS RISK RATING

CVSS v3 Vector: CVSS:3.0/AV:N/AC:H/PR:L/UI:R/S:U/C:N/I:N/A:N

VULNERABILITY REFERENCES

CWE:
http://cwe.mitre.org/data/definitions/749.html

OWASP:
A5 - Security Misconfiguration

1.5. Use of Cross-Domain Script

OBSERVATIONAL

VULNERABILITY TRACKING

STATUS: **Closed**

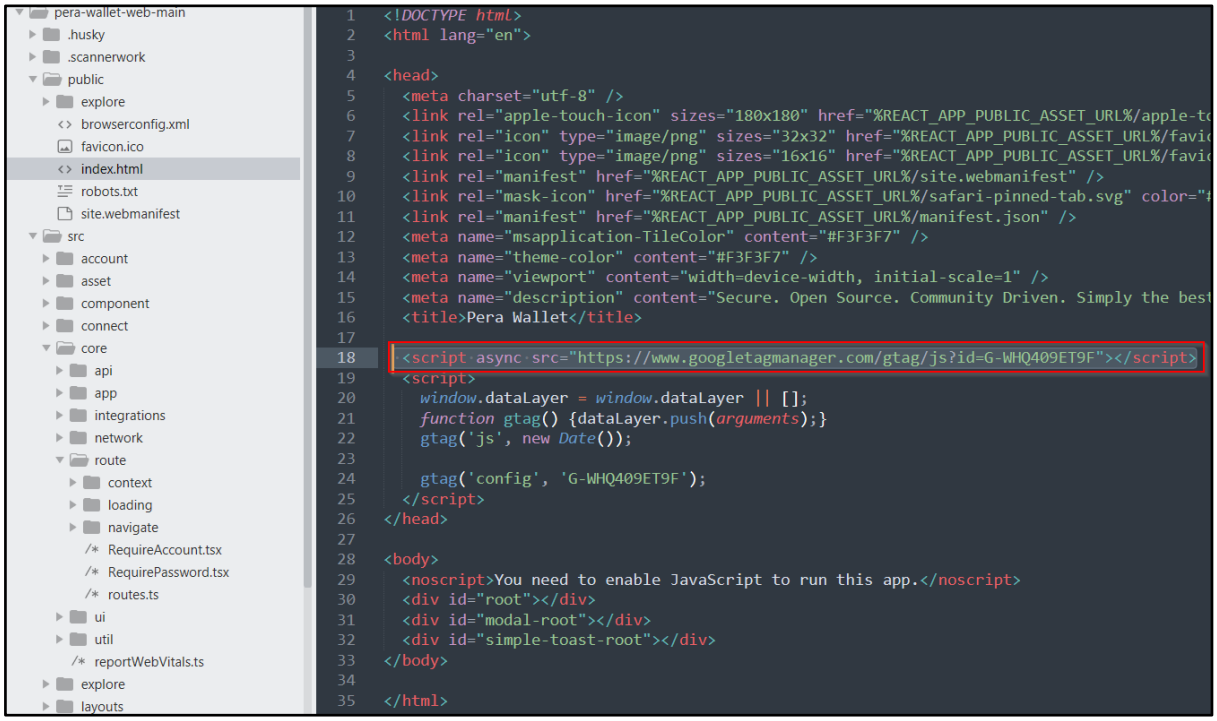
BACKGROUND

When a web application uses third-party resources/assets such as scripts, the script is executed by the browser within the security context of the web application. This means that the script can perform any actions to the web application such as accessing application data and execute actions within the context of the current user.

DESCRIPTION

It was observed that the web wallet application does not use Subresource Integrity check when using external domain's script. This can be seen from one of the web requests as shown below:

It can be seen also from the source code /public/index.html.



RECOMMENDATION

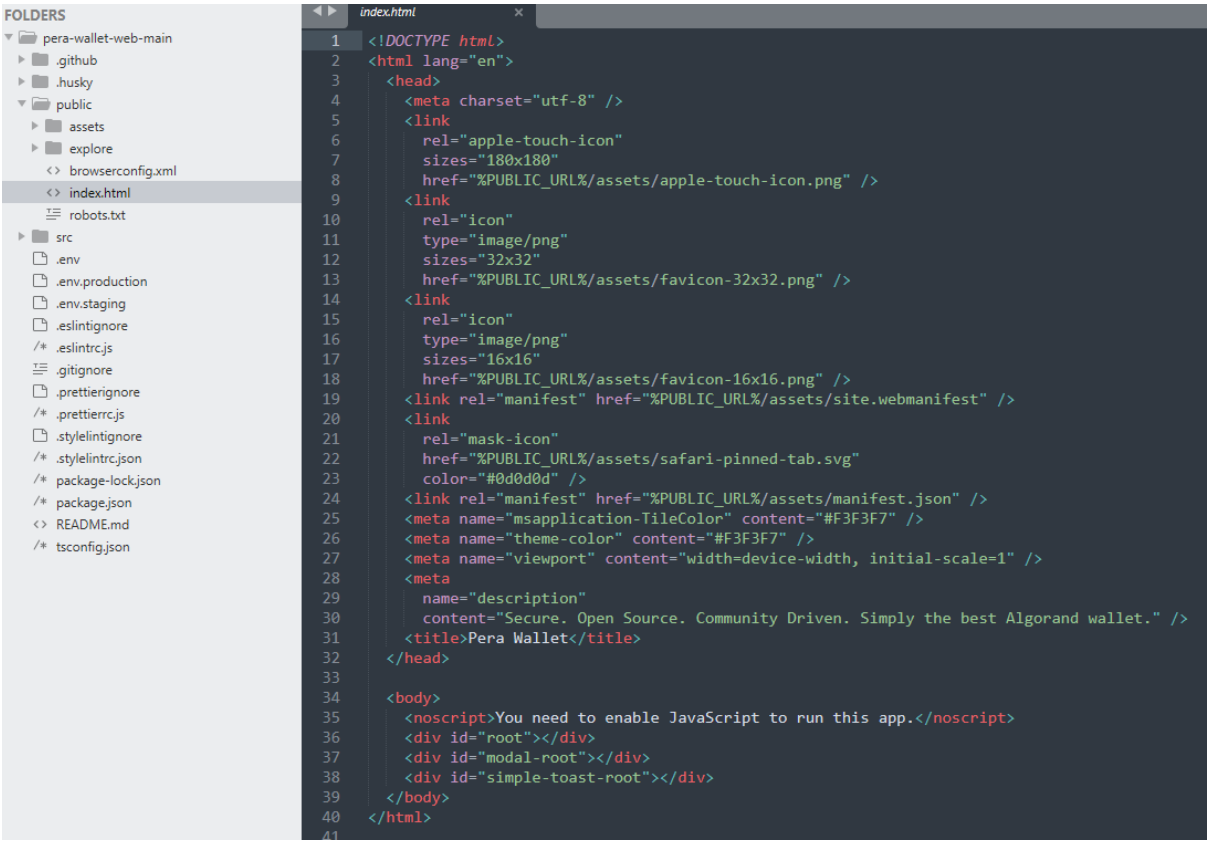
Review the need to use third-party scripts and if it is needed, it is advisable to make a local copy of the static third-party script. Otherwise, consider using Subresource Integrity on the static script so the web browser will verify it.

```
<script src="https://cdn.example.com/app.js" integrity="sha384-
+/M6kredJcxdsqkczBUjMLvqyHb1K/JThDXWsBVxMEeZHEaMKEOEct339VItX1zB"></script>
```

REGRESSION TESTING COMMENT

12th January 2023

The issue has been resolved as third-party scripts has been removed from the web wallet application.



CVSS RISK RATING

CVSS v3 Vector: CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N

VULNERABILITY REFERENCES

CWE:
https://cwe.mitre.org/data/definitions/829.html

Others:
https://www.w3.org/TR/SRI/

5. APPENDIX

DISCLAIMER

The material contained in this document is confidential and only for use by the company receiving this information from Vantage Point Security Pte. Ltd. (Vantage Point). The material will be held in the strictest confidence by the recipients and will not be used, in whole or in part, for any purpose other than the purpose for which it is provided without prior written consent by Vantage Point. The recipient assumes responsibility for further distribution of this document. In no event shall Vantage Point be liable to anyone for direct, special, incidental, collateral or consequential damages arising out of the use of this material, to the maximum extent permitted under law.

The security testing team made every effort to cover the systems in the test scope as effectively and completely as possible given the time budget available. There is however no guarantee that all existing vulnerabilities have been discovered. Furthermore, the security assessment applies to a snapshot of the current state at the examination time.

RISK RATING

Today's IT landscape is changing and evolving at an increasingly fast pace. In order keep up, develop and maintain a strong security posture it is crucial to manage IT Security successfully. An important part of that is to understand potential threats to IT systems and to actively test security controls to see if they are effective. An active security testing process brings up the challenge of managing vulnerabilities in an efficient manner. To do that Vantage Point uses the CVSS risk scoring system and it has several key advantages over proprietary risk scoring systems.

Common Vulnerability Scoring System

CVSS is an open standard designed to allow organizations to understand the criticality of security vulnerabilities and assess the priority given to security patching. It is platform and technology independent; in practice, CVSS scores can be used to rate security vulnerabilities (to get an indication of their relative severity) affecting a very wide range of software products: operating systems, web and legacy applications, security products (firewalls, antivirus software, etc.), databases, etc. Vantage Point uses CVSS scoring to be more transparent and allow organizations to compare and process vulnerabilities found by the security consultants of Vantage Point with vulnerabilities from other sources. A unified vulnerability scoring system within an organization allows better understanding of vulnerability risks and consequently provides a better basis for decisions to improve information security.

Component risk score

A component is typically identified by an IP address or domain name and usually refers to a physical or virtual computer unit such as personal computer, a mobile device or a server. All systems within the scope, containing at least one security weakness, receive a system risk score, which is directly derived from the vulnerability with the highest CVSS score.