



# **Quras Wallet**

Application Penetration Test Report

November 14, 2019



### **Confidentiality Statement**

All information contained in this document is provided in confidence for the sole purpose of adjudication of the document and shall not be published or disclosed wholly or in part to any other party without CertiK's prior permission in writing and shall be held in safe custody. These obligations shall not apply to information that is published or becomes known legitimately from some source other than CertiK.

All transactions are subject to the appropriate CertiK Standard Terms and Conditions.

Certain information given in connection with this proposal is marked "In Commercial Confidence". That information is communicated in confidence, and disclosure of it to any person other than with CertiK's consent will be a breach of confidence actionable on the part of CertiK.

### **Disclaimer**

This document is provided for information purposes only. CertiK accepts no responsibility for any errors or omissions that it may contain.

This document is provided without warranty of any kind, express or implied, including but not limited to the warranties of merchantability, fitness for a particular purpose and non-infringement. In no event shall CertiK be liable for any claim, damages or other liability (either direct or indirect or consequential), whether in an action of contract, tort or otherwise, arising from, out of or in connection with this document or the contents thereof.

This document represents our budgetary price proposal for the solution further described in this herein and is provided for information and evaluation purposes only and is not currently a formal offer capable of acceptance.

## Table of Contents

<b>1. EXECUTIVE SUMMARY .....</b>	<b>3</b>
1.1 SCOPE .....	3
1.2 LIMITATIONS.....	3
1.3 SUMMARY OF RESULTS .....	4
1.4 SUMMARY OF RECOMMENDATIONS .....	4
<b>2. METHODOLOGY.....</b>	<b>5</b>
2.1 COVERAGE AND PRIORITIZATION.....	6
2.2 RECONNAISSANCE.....	6
2.3 APPLICATION MAPPING .....	6
2.4 VULNERABILITY DISCOVERY.....	6
2.5 VULNERABILITY CONFIRMATION .....	6
2.6 IMMEDIATE ESCALATION OF HIGH OR CRITICAL FINDINGS .....	7
2.7 VULNERABILITY CLASSES .....	7
2.8 RISK ASSESSMENT.....	8
<b>3. FINDINGS.....</b>	<b>9</b>
TFM001 – ARBITRARY TEXT INJECTION IN MAILMAN (CVE-2018-13796) .....	9
TFL001 – INSECURE PASSWORD POLICY .....	12
TFL002 – CLICK JACKING: X FRAME OPTIONS HEADER MISSING .....	14
TFL003 – INFORMATION DISCLOSURE - XAMPP AND (PHPINFO()) .....	17
TFI001 – DIRECTORY LISTING.....	19
TFI002 – HTTP STRICT TRANSPORT SECURITY (HSTS) NOT ENFORCED .....	20
TFI003 – SERVER PATH DISCLOSURE .....	22
TFI004 – INFORMATION DISCLOSURE VIA BLOCKAPI.QURASWALLET.ORG:9009/v1/ .....	26
TFI005 – NON-SECURE REQUESTS ARE NOT AUTOMATICALLY UPGRADED TO HTTPS .....	27
<b>4. ENGAGEMENT .....</b>	<b>ERROR! BOOKMARK NOT DEFINED.</b>
4.1 CERTIK CONSULTANTS .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
4.2 OTHER CONTACTS .....	<b>ERROR! BOOKMARK NOT DEFINED.</b>
<b>5. ABOUT CERTIK .....</b>	<b>29</b>

## 1. Executive Summary

Quras engaged CertiK LLC to perform an application penetration test for their application **Quras Wallet**. The main objective of the engagement is to test the overall resiliency of the application to various real-world attacks against the application's controls and functions. And thereby be able to identify its weaknesses and provide recommendations to fix and improve its overall security posture.

After a thorough review of the application, CertiK believes that the Quras Wallet Application is currently at a **MEDIUM** risk level. Given the severity of the vulnerabilities on the application, it is unlikely that the application will be directly compromised.

During the test, CertiK was not able to compromise the target application. However, some significant vulnerabilities were found that could potentially be leveraged to lead to a certain level of compromise later.

The most significant finding was an **Arbitrary Text Injection Vulnerability** on the Mailman function being used by the application. With this, a crafted URL can cause arbitrary text to be displayed on the web page of the affected website. An attacker can leverage this vulnerability to perform a content spoofing attack.

Other weaknesses were also found and are detailed on the Findings section of the report.

It is recommended that Quras immediately work on remediating the findings to raise the security posture of the application.

### 1.1 Scope

At the start of engagement, CertiK worked with Quras to identify the target and set the limits on the scope of the test. A **Black Box** type of testing approach was done where CertiK performed the test with minimal knowledge about the application.

The following details the target scope of the test.

<b>Application Name</b>	Quras Wallet
<b>Version</b>	
<b>Target URL</b>	<a href="https://quraswallet.org/">https://quraswallet.org/</a>
<b>IP Address</b>	162.241.194.39
<b>Environment</b>	Production

### 1.2 Limitations

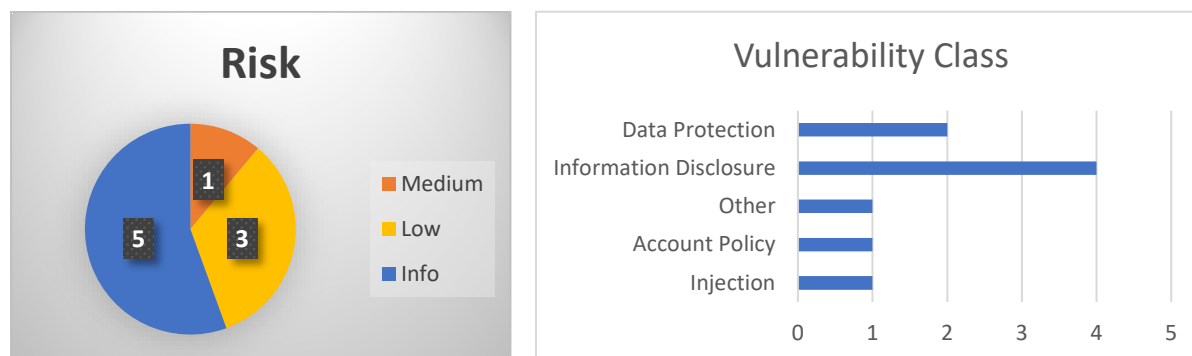
No major limitations were identified during the test. However, Smart Contract functions were not tested as they were not yet available during time of testing.

As a standard practice as well, Denial of Service attacks are out of scope.

Testing was performed during regular hours as well as off hours throughout the course of the test.

### 1.3 Summary of Results

The information below summarizes the vulnerabilities found on the application:



ID	Risk Level	Name	Vulnerability Class	Status
TFM001	Medium	Arbitrary text injection in Mailman (CVE-2018-13796)	Injection	Open
TFL001	Low	Insecure Password Policy	Account Policy	Open
TFL002	Low	Click jacking: X Frame Options Header Missing	Other	Open
TFL003	Low	Information Disclosure - XAMPP and (phpinfo())	Information Disclosure	Open
TFI001	Info	Directory Listing	Information Disclosure	Open
TFI002	Info	HTTP Strict Transport Security (HSTS) Not Enforced	Data Protection	Open
TFI003	Info	Server path disclosure	Information Disclosure	Open
TFI004	Info	Information Disclosure via blockapi.quraswallet.org:9009/v1/	Information Disclosure	Open
TFI005	Info	Non-secure requests are not automatically upgraded to HTTPS	Data Protection	Open

### 1.4 Summary of Recommendations

The information below outlines the major recommendations that CertiK suggests for Quras to take in order to strategically lower the risk level of the application:

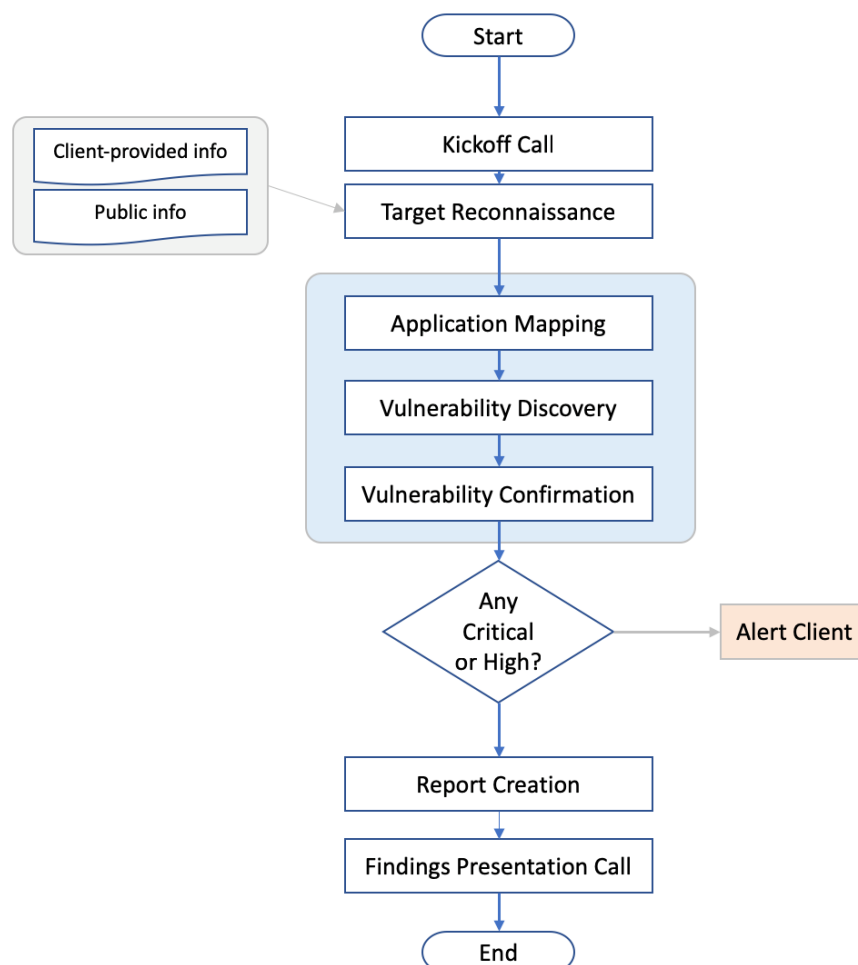
- Hold a Secure Development Training class to attack the root of security problems.
- Perform additional testing once fixes are in place to both validate vulnerabilities have been fixed and that the new functionality did not introduce additional vulnerabilities.
- Continue to perform periodic security assessments of the application source code, server security posture, and network architecture to ensure compliance with the corporate security policies and procedures.

The initiative of Quras to perform these tests shows their appreciation and value for security. However, this is just the initial step and the continuous work to remediate and improve the security posture will be the goal. CertiK is happy to help with any of these issues.

## 2. Methodology

CertiK uses a comprehensive penetration testing methodology which adheres to industry best practices and standards in security assessments including from **OWASP** (Open Web Application Security Project), **NIST**, **PTES** (Penetration Testing Execution Standard).

Below is a flowchart of our assessment process:



## **2.1 Coverage and Prioritization**

As many components as possible will be tested manually. Priority is generally based on three factors: critical security controls, sensitive data, and likelihood of vulnerability.

Critical security controls will always receive the top priority in the test. If a vulnerability is discovered in a critical security control, the entire application is likely to be compromised, resulting in a critical risk to the business. For most applications, critical controls will include the login page, but it could also include major workflows such as the checkout function in an online store.

Second priority is given to application components that handle sensitive data. This is dependent on business priorities, but common examples include payment card data, financial data, or authentication credentials.

Final priority includes areas of the application that are most likely to be vulnerable. This is based on CertiK' experience with similar applications developed using the same technology or with other applications that fit the same business role. For example, large applications will often have older sections that are less likely to utilize modern security techniques.

## **2.2 Reconnaissance**

CertiK gathers information about the target application from various sources depending on the type of test being performed. CertiK obtains whatever information that is possible and appropriate from the client during scoping and supplements it with relevant information that can be gathered from public sources. This helps provide a better overall picture and understanding of the target.

## **2.3 Application Mapping**

CertiK examines the application, reviewing its contents, and mapping out all its functionalities and components. CertiK makes use of different tools and techniques to traverse the entire application and document all input areas and processes. Automated tools are used to scan the application and it is then manually examined for all its parameters and functionalities. With this, CertiK creates and widens the overall attack surface of the target application.

## **2.4 Vulnerability Discovery**

Using the information that is gathered, CertiK comes up with various attack vectors to test against the application. CertiK uses a combination of automated tools and manual techniques to identify vulnerabilities and weaknesses. Industry-recognized testing tools will be used, including Burp Suite, Nikto, Metasploit, and Kali. Furthermore, any controls in place that would inhibit the successful exploitation of a system will be noted.

## **2.5 Vulnerability Confirmation**

After discovering vulnerabilities on the application, CertiK validates the vulnerabilities and assess its overall impact. To validate, CertiK performs a Proof-of-Concept of an attack on

the vulnerability, simulating real world scenarios to prove the risk and overall impact of the vulnerability.

Through CertiK's knowledge and experience on attacks and exploitation techniques, CertiK is able to process all weaknesses and examine how they can be combined to compromise the application. CertiK may use different attack chains, leveraging different weaknesses to escalate and gain a more significant compromise.

To minimize any potential negative impact, vulnerability exploitation was only attempted when it would not adversely affect production applications and systems, and then only to confirm the presence of a specific vulnerability. Any attack with the potential to cause system downtime or seriously impact business continuity was not performed. Vulnerabilities were never exploited to delete or modify data; only read-level access was attempted. If it appeared possible to modify data, this was noted in the list of vulnerabilities below.

## 2.6 Immediate escalation of High or Critical Findings

If critical or high findings are found whereby application elements are compromised, client's key security contacts will be notified immediately.

## 2.7 Vulnerability Classes

Below is the list of common vulnerability classes to be tested.

<b>Data Protection</b>	<ul style="list-style-type: none"><li>• Transport</li><li>• Storage</li></ul>
<b>Information Disclosure</b>	<ul style="list-style-type: none"><li>• Directory Indexing</li><li>• Verbose Error Messages</li><li>• HTML Comments</li><li>• Default Content</li></ul>
<b>Account Policy</b>	<ul style="list-style-type: none"><li>• Default / Weak Passwords</li><li>• Unlimited Login Attempts</li><li>• Password Reset</li><li>• Insufficient Session Expiration</li></ul>
<b>Session Management</b>	<ul style="list-style-type: none"><li>• Session Identifier Prediction</li><li>• Session Hijacking</li><li>• Session Replay</li><li>• Session Fixation/Trapping</li><li>• Cross-Site Request Forgery</li></ul>
<b>Injection</b>	<ul style="list-style-type: none"><li>• SQL Injection</li><li>• Cross-Site Scripting</li><li>• LDAP Injection</li><li>• HTML Injection</li><li>• XML Injection</li><li>• OS Command Injection</li></ul>
<b>Authentication and Authorization</b>	<ul style="list-style-type: none"><li>• Authentication Bypass</li><li>• Authorization Bypass</li></ul>



	<ul style="list-style-type: none"> <li>• Privilege Escalation</li> </ul>
<b>Application Resource Handling</b>	<ul style="list-style-type: none"> <li>• Path Traversal</li> <li>• Predictable Object Identifiers</li> <li>• XML External Entity Expansion</li> <li>• Local &amp; Remote File Inclusion</li> </ul>
<b>Logic Flaws</b>	<ul style="list-style-type: none"> <li>• Abuse of Functionality</li> <li>• Workflow Bypass</li> </ul>
<b>Cryptography</b>	<ul style="list-style-type: none"> <li>• Weak Algorithms</li> </ul>

## 2.8 Risk Assessment

The following risk levels categorize the risk level of issues presented in the report:

Risk Level	CVSS Score	Impact	Exploitability
<b>Critical</b>	9.0-10.0	Root-level or full-system compromise, large-scale data breach	Trivial and straightforward
<b>High</b>	7.0-8.9	Elevated privilege access, significant data loss or downtime	Easy, vulnerability details or exploit code are publicly available, but may need additional attack vectors (e.g., social engineering)
<b>Medium</b>	4.0-6.9	Limited access but can still cause loss of tangible assets, which may violate, harm, or impede the org's mission, reputation, or interests.	Difficult, requires a skilled attacker, needs additional attack vectors, attacker must reside on the same network, requires user privileges
<b>Low</b>	0.1-3.9	Very little impact on an org's business	Extremely difficult, requires local or physical system access
<b>Info</b>	0.0	Discloses information that may be of interest to an attacker.	Not exploitable but rather is a weakness that may be useful to an attacker should a higher risk issue be found that allows for a system exploit

### 3. Findings

The information below lists the vulnerabilities that CertiK have found during the test. Details of the vulnerabilities are provided as well as thorough recommendations on how the vulnerabilities can be fixed.

Findings are arranged according to risk level.

#### TFM001 – Arbitrary text injection in Mailman (CVE-2018-13796)

Risk Level	Medium		
Vulnerability Class	Injection		
Status	Open		
CVSS v3.1 Rating	5.3	CVSS v3.1 Vectors	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N
Location	<a href="https://quraswallet.org/mailman/">https://quraswallet.org/mailman/</a> <a href="https://mail.quraswallet.org/mailman/">https://mail.quraswallet.org/mailman/</a>		
Description			
<ul style="list-style-type: none"><li>• CertiK found that the GNU Mailman version used by the application is vulnerable to Arbitrary text injection. This vulnerability is found on GNU Mailman version 2.1.28 and below.</li><li>• With this vulnerability, a crafted URL can cause arbitrary text to be displayed on the web page of the affected website. An attacker can leverage this vulnerability to perform a content spoofing attack.</li><li>• Content Spoofing is an attack used to trick a user into thinking that fake web site content is legitimate data and is an attack targeting a user made possible by injection vulnerability in a web application. When an application does not properly handle user supplied data, an attacker can supply content to a web application, typically via a parameter value, that is reflected back to the user.</li><li>• The issue was found in <b>Listinfo</b> and <b>Option</b><ul style="list-style-type: none"><li>○ <a href="https://quraswallet.org/mailman/listinfo">https://quraswallet.org/mailman/listinfo</a></li><li>○ <a href="https://quraswallet.org/mailman/options/">https://quraswallet.org/mailman/options/</a></li><li>○ <a href="https://mail.quraswallet.org/mailman/listinfo/">https://mail.quraswallet.org/mailman/listinfo/</a></li><li>○ <a href="https://mail.quraswallet.org/mailman/options/">https://mail.quraswallet.org/mailman/options/</a></li></ul></li></ul>			
Evidence :			
<p>The following evidence shows that a URL with a very long text such as</p> <p><a href="https://quraswallet.org/mailman/listinfo/This_is_a_long_string_with_some_phishing_text">https://quraswallet.org/mailman/listinfo/This_is_a_long_string_with_some_phishing_text</a> <a href="https://quraswallet.org/mailman/options/This_is_a_long_string_with_some_phishing_text">https://quraswallet.org/mailman/options/This_is_a_long_string_with_some_phishing_text</a> <a href="https://mail.quraswallet.org/mailman/listinfo/This_is_a_long_string_with_some_phishing_text">https://mail.quraswallet.org/mailman/listinfo/This_is_a_long_string_with_some_phishing_text</a></p>			

[https://mail.qurасwallet.org/mailman/options/This\\_is\\_a\\_long\\_string\\_with\\_some\\_phishing\\_text](https://mail.qurасwallet.org/mailman/options/This_is_a_long_string_with_some_phishing_text)

Will echo the text injected text in the error response. This can be used to make a potential victim think the phishing text comes from the trusted site

The following shows the sample HTTP Requests/Responses demonstration the vulnerability:

### Request 1:

```
GET /mailman/listinfo/This_is_a_long_string_with_some_phishing_text HTTP/1.1
Host: qurасwallet.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101
Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: roundcube_cookies=enabled; language=en-us;
webmailsession=%3aBQjtKu4P0p78fE7z%2cc5c8dfbaf80df7445ce91f9a19ad9fa0;
timezone=Asia/Shanghai
Upgrade-Insecure-Requests: 1
```

### Response 1:

```
HTTP/1.1 404 Not Found
Date: Wed, 13 Nov 2019 12:21:27 GMT
Server: Apache
Upgrade: h2,h2c
Connection: Upgrade, close
Content-Type: text/html; charset=us-ascii
Content-Length: 1664

<HTML>
<HEAD>
<LINK REL="SHORTCUT ICON" HREF="/img-sys/mm-icon.png">
<META http-equiv="Content-Type" content="text/html; charset=us-ascii">
<TITLE>qurасwallet.org Mailing Lists</TITLE>
<style type="text/css">
div.hidden
{position:absolute;
left:-1000px;
top:auto;
width:1px;
height:1px;
overflow:hidden;}
</style>
</HEAD>
<BODY bgcolor="white"
dir="ltr">

<table WIDTH="100%" BORDER="0">
<tr>
<td COLSPAN="2" BGCOLOR="#99ccff"><center><h2>qurасwallet.org Mailing Lists</h2></center></td>
</tr>
<tr>
<td COLSPAN="2"><font color="ff5060" size="+1">No such list <em>this is a long string with some phishing text</em></font><p>There currently
<a href="http://www.gnu.org/software/mailman/index.html">Mailman</a> mailing lists on qurасwallet.org. To visit the general informat
open a URL similar to this one, but with a '/' and the right
list name appended.
<p>List administrators, you can visit <a href="mailto:admin@qurасwallet.org">the list admin overview page</a> to find the management interface for your list
<p>If you are having trouble using the lists, please contact <a href="mailto:mailman@qurасwallet.org">mailman@qurасwallet.org</a>.<p></
</tr>
</table>
<hr>
<table WIDTH="100%" BORDER="0">
<tr>
<td><br>version 2.1.27</td>
<td></td>
<td></td>
</tr>
```

### Request 2:

```
GET /mailman/options/This_is_a_long_string_with_some_phishing_text HTTP/1.1
Host: guraswallet.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101
Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Cookie: roundcube_cookies=enabled; language=en-us;
webmailsession=%3aBQjtKu4P0p78fE7z%2cc5c8dfbaf80df7445ce91f9a19ad9fa0;
timezone=Asia/Shanghai
Upgrade-Insecure-Requests: 1
```

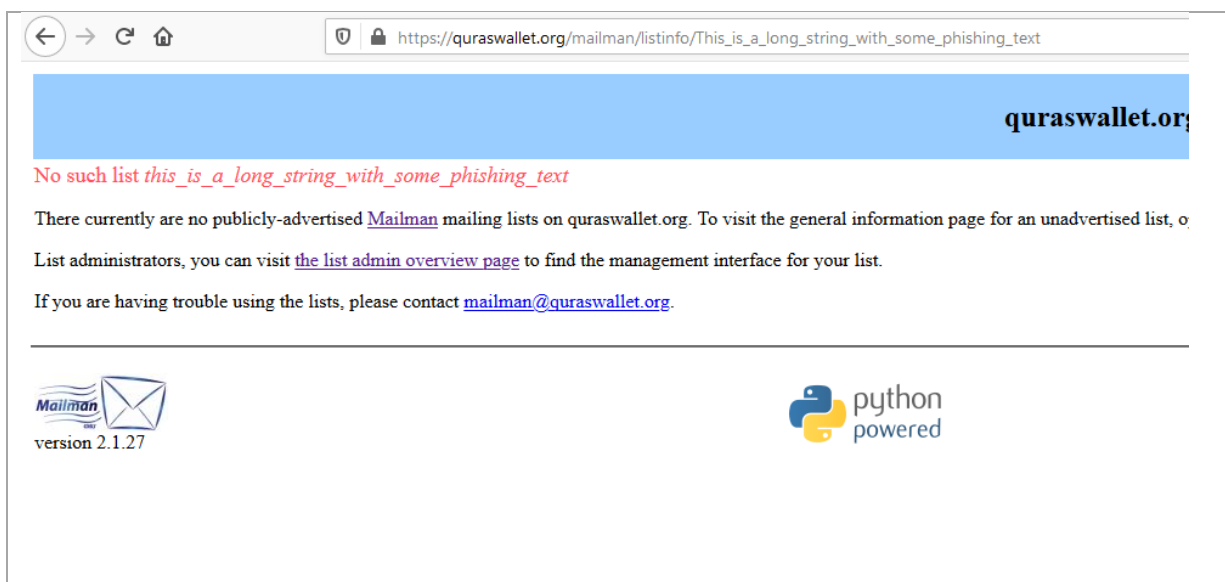
## Response 2:

```
HTTP/1.1 404 Not Found
Date: Wed, 13 Nov 2019 12:21:39 GMT
Server: Apache
Upgrade: h2,h2c
Connection: Upgrade, close
Content-Type: text/html; charset=us-ascii
Content-Length: 908


<HTML>
<HEAD>
<LINK REL="SHORTCUT ICON" HREF="/img-sys/mm-icon.png">
<META http-equiv="Content-Type" content="text/html; charset=us-ascii">
<TITLE>CGI script error</TITLE>
<style type="text/css">
div.hidden
{
position:absolute;
left:-10000px;
top:auto;
width:1px;
height:1px;
overflow:hidden;
}
</style>
</HEAD>
<BODY bgcolor="white"
dir="ltr">
<h3>CGI script error</h3><h3><strong><font color="red" size="+2">Error: </font></strong><em>No such list <em>this is a long string with some phishing text</em></em></h3><hr>
<table WIDTH="100%" BORDER="0">
<tr>
<td><br>version 2.1.27</td>
<td></td>
<td></td>
</tr>
</table>
</BODY>
</HTML>
```

The following screenshots shows evidence that the string is reflected on the website:

## Screenshot 1:



**Screenshot 2:**



Recommendation	<ul style="list-style-type: none"> <li>Never Construct and send Error messages via request parameters and prefer using Messages predefined in a property file.</li> <li>Upgrade GNU mailman 2.1.27 to latest version.</li> </ul>
References	<a href="https://www.owasp.org/index.php/Content_Spoofing">https://www.owasp.org/index.php/Content_Spoofing</a> <a href="https://www.cvedetails.com/cve/CVE-2018-13796/">https://www.cvedetails.com/cve/CVE-2018-13796/</a> <a href="https://vuldb.com/?id.121382">https://vuldb.com/?id.121382</a>

## TFL001 – Insecure Password Policy

Risk Level	Low
Vulnerability Class	Account Policy

Status	Open		
CVSS v3.1 Rating	3.9	CVSS v3.1 Vectors	AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:N/E:X/RL:O/RC:C
Location	<a href="https://quraswallet.org/unlock-your-wallet.html">https://quraswallet.org/unlock-your-wallet.html</a>		

#### Description

- The application or host allowed passwords to be created which
  - Did not have any length restrictions
  - Did not require special characters
  - Did not require mixed-case characters
- Weak passwords can be easily brute forced to gain unauthorized access to the application.
- This may lead to the compromise of customer accounts, disclosure of private information, and access to critical system functions.
- CertiK was able to use the password 'aaa' for any of the provided test accounts.

#### Evidence :

Here are the steps to recreate the finding:

- Create new wallet at <https://quraswallet.org/new-wallet.html>
- Using weak password (e.g. password, pass123 or aaa) user able to create new wallet.
- Client/Server successful login communication showing the submission of 'aaa' as the password:

The following shows actual HTTP request/responses demonstrating the vulnerability:

#### Request:

```
POST /server-php/index.php?c=Wallet&a=uploadFile HTTP/1.1
Host: quraswallet.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=-----115662664831451
Content-Length: 895
Origin: https://quraswallet.org
Connection: close
Referer: https://quraswallet.org/unlock-your-wallet.html
Cookie: roundcube_cookies=enabled; language=en-us; PHPSESSID=3146d75737d04b89afe36356759748c

-----115662664831451
Content-Disposition: form-data; name="file"; filename="DjWT89CNBupvc7M94JtJShaSSZXxCutKAZ.qrs"
Content-Type: application/octet-stream

04ae83d97b5552c809a902d8f27260e0eba5ff5e3559aa9e2608805a7b3e2031fc6b571134c63ae5cf4b89a4935c2f26d144482fa868e1
720339e5185d16d9d573c5b1228fbdfe3ba21f8b2467545a6f214af8225d29bcb186fef24dc010d76e89dd243a7ad147b5535119c91f
cc6ff9a885bba68e08a832918d
-----115662664831451
Content-Disposition: form-data; name="password"

aaa
-----115662664831451
Content-Disposition: form-data; name="sessionId"

-----115662664831451
Content-Disposition: form-data; name="token"

871051778
-----115662664831451--
```

#### Response:

<pre> HTTP/1.1 200 OK Date: Tue, 05 Nov 2019 05:58:36 GMT Server: Apache Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Upgrade: h2,h2c Connection: Upgrade, close Vary: Accept-Encoding Content-Length: 177 Content-Type: text/html; charset=UTF-8  {"success":true,"privateKey":"e60880772fa400c16595a4fa9ff76e5cec26b85e426fa28662c52540987dcce4","ad mAmount":"0"} </pre>	
Recommendation	<ul style="list-style-type: none"> <li>Develop an organizational-wide password policy requiring the use of strong passwords or phrases on all systems. For applications, ensure that all passwords pass through a server-side validation routine that rejects non-compliant passwords.</li> <li>A strong password will be a phrase or sentence that can be easily remembered, and meet the following requirements: <ul style="list-style-type: none"> <li>Be at least 10 characters long</li> <li>Not be restricted to dictionary words or based on the username</li> <li>Contain at least one character from each category: <ul style="list-style-type: none"> <li>Uppercase letters</li> <li>Lowercase letters</li> <li>Numbers</li> <li>Non-alphanumeric characters</li> </ul> </li> </ul> </li> <li>Passwords must be changed periodically. The period should be defined based on the sensitivity of the data being protected.</li> </ul>
References	<a href="https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Implement_Proper_Password_Strength_Controls">https://www.owasp.org/index.php/Authentication_Cheat_Sheet#Implement_Proper_Password_Strength_Controls</a>

## TFL002 – Click jacking: X Frame Options Header Missing

Risk Level	Low		
Vulnerability Class	Other		
Status	Open		
CVSS v3.1 Rating	2.8	CVSS v3.1 Vectors	AV:N/AC:L/PR:N/UI:R/S:U/C:N/I:N/A:N/E:X/RL:O/RC:C
Location	<a href="https://quraswallet.org/unlock-your-wallet.html">https://quraswallet.org/unlock-your-wallet.html</a>		
Description			
<ul style="list-style-type: none"><li>• The server didn't return an X-Frame-Options header which means that this website could be at risk of a <b>Clickjacking attack</b> where an attacker can trick the user to click on a button (or image) which appears to perform a function they may wish to invoke, however in reality overlays an invisible layer (with the real intended website) in front.</li><li>• In addition, the ability to frame pages also increases the risk of successful Cross-Site Scripting (XSS) attacks and exploitation</li></ul>			

- The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

#### Evidence :

The following is a POC code that can be used to demonstrate clickjacking.

```
<html>
  <style>
    iframe {
      width:1000px;
      height:500px;
      position:absolute;
      top:0; left:0;
      filter:alpha(opacity=10); /* in a real attack this would be opacity=0 */
    }
  </style>
  <body>
    <button style="z-index:-1;margin-top:215px;margin-left:270px;width:50px;">Fun!</button>
    <iframe src="https://quraswallet.org/new-wallet.html" width="800" height="400"></iframe>
  </body>
</html>
```

The following shows that there is no X-Frame-Options Header on the response:

**Response:**



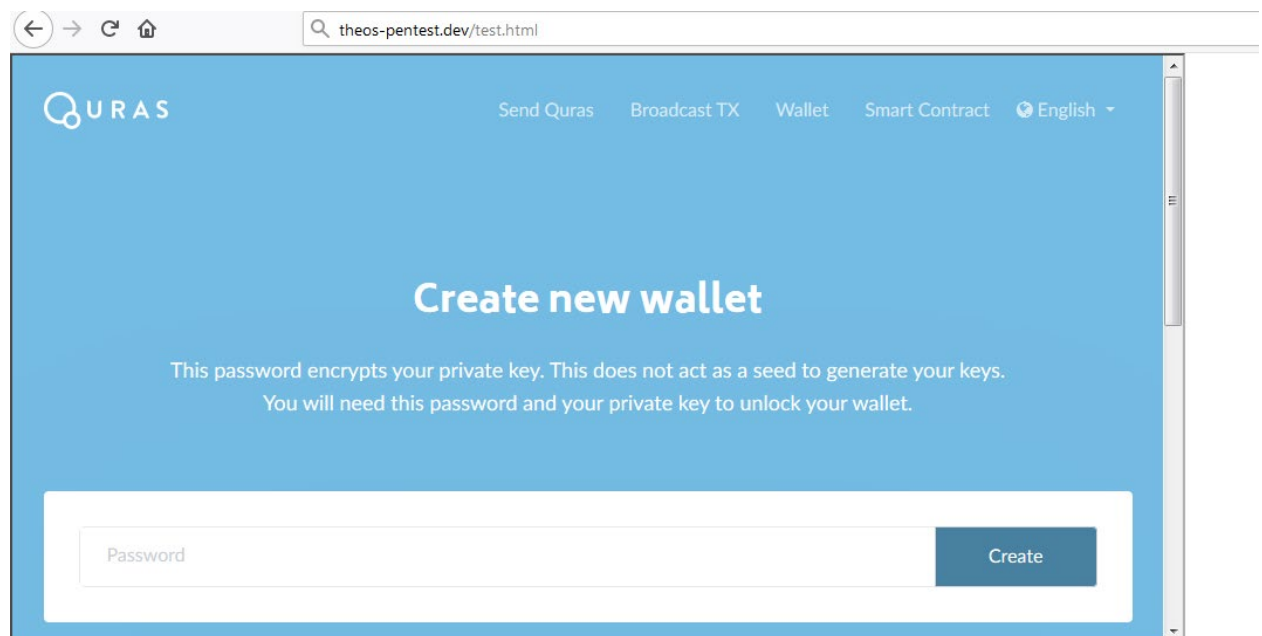
```

HTTP/1.1 200 OK
Date: Tue, 05 Nov 2019 06:13:09 GMT
Server: Apache
Upgrade: h2,h2c
Connection: Upgrade, close
Last-Modified: Wed, 25 Sep 2019 04:34:59 GMT
Accept-Ranges: bytes
Vary: Accept-Encoding
Content-Length: 6858
Content-Type: text/html

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Create Quras Wallet</title>
    <meta HTTP-EQUIV="CACHE-CONTROL" CONTENT="NO-CACHE">
    <meta name="viewport" content="width=device-width, initial-scale=1.0, maxi
    <meta name="keywords" content="quras,org,crypto
currency,cryptocurrency,anonymity,anonymous,smartcontract,IoT,clouds,zksnarks,ring
,wallet,web wallet" />
    <meta name="description" content="Create a new qurawallet, qurawallet.or

```

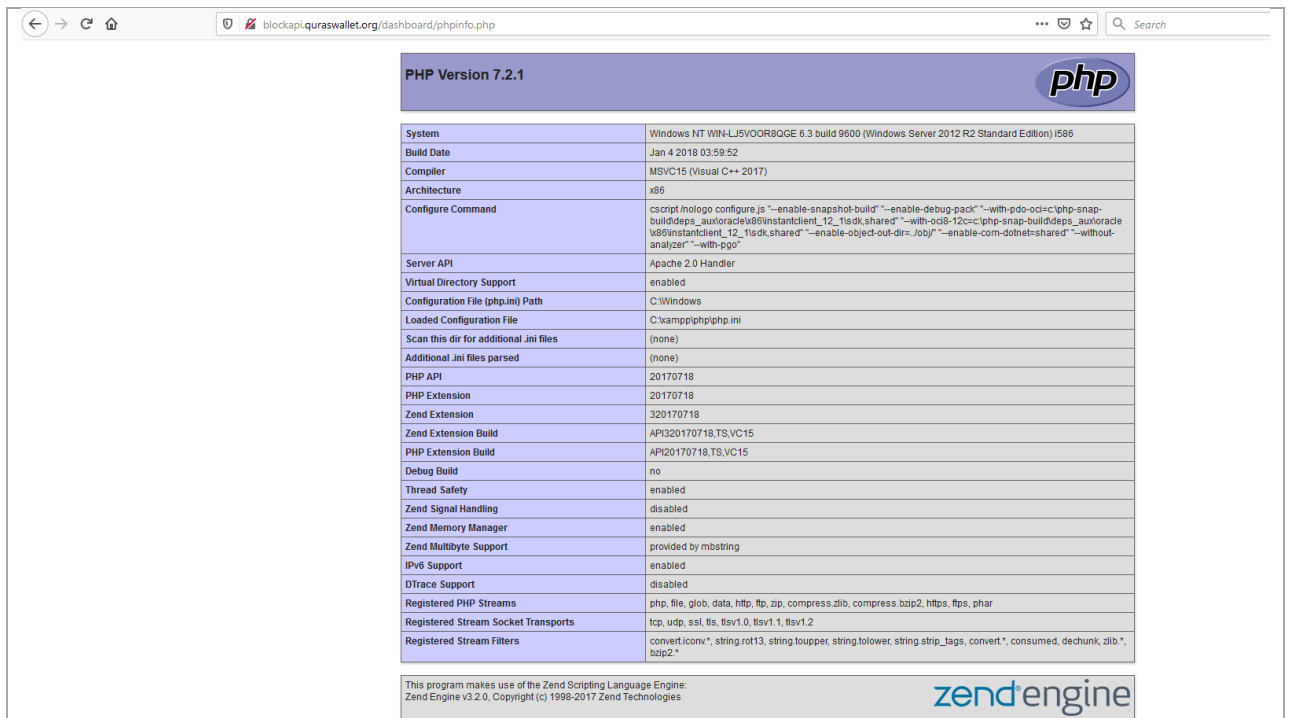
The following shows a screenshot of the site being framed:



Recommendation	<ul style="list-style-type: none"> <li>Configure your web server to include an X-Frame-Options header with the name X-Frame-Options and the value DENY to prevent framing altogether, or the value SAMEORIGIN to allow framing only by pages on the same origin as the response itself.</li> </ul>
References	<a href="https://cwe.mitre.org/data/definitions/693.html">https://cwe.mitre.org/data/definitions/693.html</a> <a href="https://www.owasp.org/index.php/Clickjacking">https://www.owasp.org/index.php/Clickjacking</a>

### TFL003 – Information Disclosure - XAMPP and (phpinfo())

Risk Level	Low		
Vulnerability Class	Information Disclosure		
Status	Open		
CVSS v3.1 Rating	3.4	CVSS v3.1 Vectors	<a href="#">AV:L/AC:L/PR:N/UI:N/S:U/C:L/I:N/A:N/E:U/RL:O/RC:R/CR:X/IR:X/AR:X/MAV:X/MAC:L/MPR:N/MUI:N/MS:X/MC:L/MI:N/MA:N</a>
Location	<a href="http://blockapi.quraswallet.org/dashboard/phpinfo.php">http://blockapi.quraswallet.org/dashboard/phpinfo.php</a>		
Description			
<ul style="list-style-type: none"><li>• CertiK found that the some important information about the application were disclosed on the XAMPP homepage and Phpinfo where were publicly accessible.</li><li>• The information found on XAMPP and phpinfo() can help an attacker gain more information on the web server. An attacker can research known vulnerabilities for that system under review. The attacker can also use this information during the exploitation of other vulnerabilities.</li><li>• Some information that are disclosed include:<ul style="list-style-type: none"><li>○ Exact PHP version.</li><li>○ Exact OS and its version.</li><li>○ Details of the PHP configuration.</li><li>○ Internal IP addresses.</li><li>○ Server environment variables.</li><li>○ Loaded PHP extensions and their configurations.</li></ul></li></ul>			
Evidence :			
<p>The following screenshot shows information disclosed on PHPInfo and XAMPP</p> <p><b>PHPInfo</b></p>			



The screenshot shows the PHP Version 7.2.1 information page. The page title is "PHP Version 7.2.1" and it features the PHP logo. The main content is a table with system and configuration details. The table has two columns: the property name and its value. The properties include System, Build Date, Compiler, Architecture, Configure Command, Server API, Virtual Directory Support, Configuration File (php.ini) Path, Loaded Configuration File, Scan this dir for additional .ini files, Additional .ini files parsed, PHP API, PHP Extension, Zend Extension, Zend Extension Build, PHP Extension Build, Debug Build, Thread Safety, Zend Signal Handling, Zend Memory Manager, Zend Multibyte Support, IPv6 Support, DTrace Support, Registered PHP Streams, Registered Stream Socket Transports, and Registered Stream Filters. The values provide specific details about the installed PHP version and its environment. At the bottom of the page, there is a note about the Zend Engine and the Zend Technologies logo.

System	Windows NT WIN-LJ5VQR8QGE 6.3 build 9600 (Windows Server 2012 R2 Standard Edition) i586
Build Date	Jan 4 2018 03:59:52
Compiler	MSVC15 (Visual C++ 2017)
Architecture	x86
Configure Command	cmd.exe /c: "cd /d C:\php & phpize --enable-snapshot-build --enable-debug-pack --with-pdo-oci=c:\php-snap-build\deps_auroradev\instantclient_12_1\src\shared --with-oci8-12c=c:\php-snap-build\deps_auroradev\instantclient_12_1\src\shared --enable-object-out-dir=.obj --enable-com-donnet-shared --without-analyzer --with-pg"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.TS.VC15
PHP Extension Build	API20170718.TS.VC15
Debug Build	no
Thread Safety	enabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, zip, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

This program makes use of the Zend Scripting Language Engine:  
Zend Engine v3.2.0. Copyright (c) 1998-2017 Zend Technologies

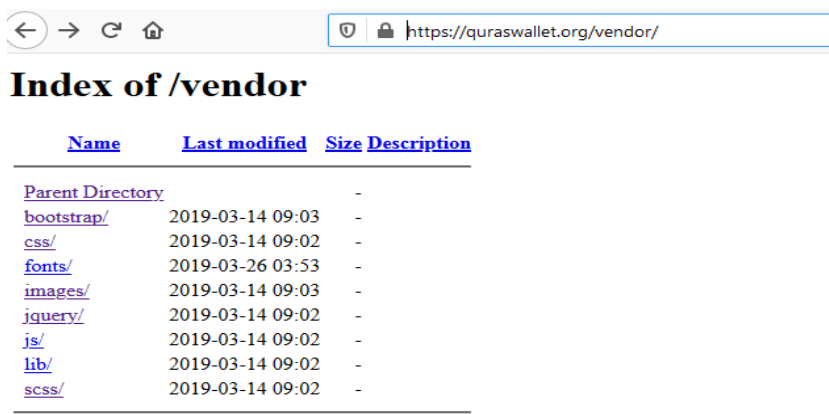
## XAMPP

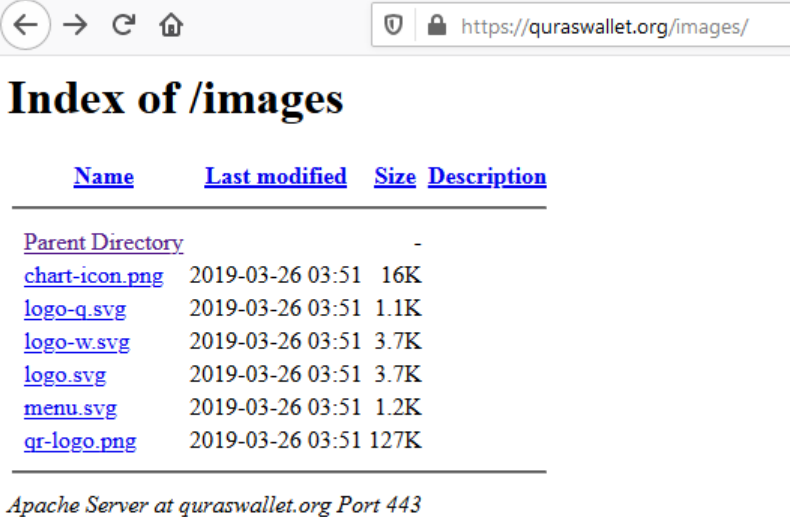
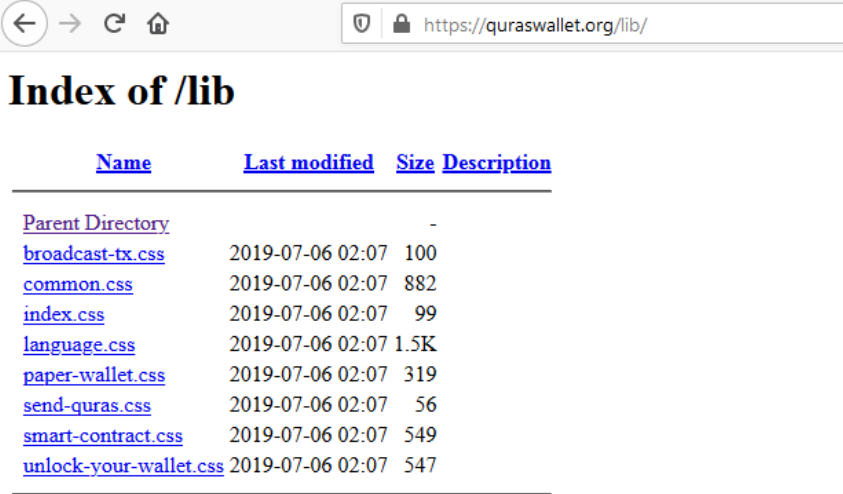


The screenshot shows the XAMPP Apache + MariaDB + PHP + Perl page. The page has a dark blue header with the XAMPP logo and navigation links: Applications, FAQs, HOW-TO Guides, PHPInfo, and phpMyAdmin. The main content area is light yellow and contains the text "Welcome to XAMPP for Windows 7.2.1". Below this, there is a paragraph explaining that the user has successfully installed XAMPP and can start using Apache, MariaDB, PHP, and other components. It also mentions that the user can find more information in the FAQs section or check the HOW-TO Guides. Another paragraph explains that XAMPP is meant only for development purposes and is not secure for production use. It suggests that the user can use WAMP, MAMP, or LAMP for production. A third paragraph instructs the user to start the XAMPP Control Panel to check the server status. A "Community" section mentions that XAMPP has been around for more than 10 years and provides links to the Forums, Mailing List, Facebook, Twitter, and Google+ circles. A "Contribute to XAMPP translation" section provides a link to [translate.apachefriends.org](https://translate.apachefriends.org). At the bottom, there is a small note about translating XAMPP for other community members.

Recommendation	<ul style="list-style-type: none"> <li>Remove the file from production systems or web server (e.g. XAMPP dashboard, phpinfo.php)</li> </ul>
References	<a href="https://www.php.net/manual/en/function.phpinfo.php">https://www.php.net/manual/en/function.phpinfo.php</a> <a href="https://cwe.mitre.org/data/definitions/200.html">https://cwe.mitre.org/data/definitions/200.html</a> <a href="https://vuldb.com/?id.88834">https://vuldb.com/?id.88834</a>

## TFI001 – Directory Listing

Risk Level	Info		
Vulnerability Class	Information Disclosure		
Status	Open		
CVSS v3.1 Rating	N/A	CVSS v3.1 Vectors	N/A
Location	<a href="https://quraswallet.org/vendor/">https://quraswallet.org/vendor/</a> <a href="https://mail.quraswallet.org/vendor/">https://mail.quraswallet.org/vendor/</a>		
Description			
<ul style="list-style-type: none"><li>• CertiK found that some of the directories on the application were susceptible to directory listing.</li><li>• Web servers can be configured to automatically list the contents of directories that do not have an index page present. This can aid an attacker by enabling them to quickly identify the resources at a given path and proceed directly to analyzing and attacking those resources. It particularly increases the exposure of sensitive files within the directory that are not intended to be accessible to users, such as temporary files and crash dumps.</li><li>• Directory listings themselves do not necessarily constitute security vulnerability. Any sensitive resources within the web root should in any case be properly access-controlled, and should not be accessible by an unauthorized party who happens to know or guess the URL. Even when directory listings are disabled, an attacker may guess the location of sensitive files using automated tools.</li></ul>			
Evidence :			
<p>The following screenshot shows successful directory listing:</p> 			

 <p>The screenshot shows a web browser window with the address bar displaying <a href="https://quraswallet.org/images/">https://quraswallet.org/images/</a>. The page title is "Index of /images". Below the title is a table with columns: Name, Last modified, Size, and Description. The table lists several files including chart-icon.png, logo-q.svg, logo-w.svg, logo.svg, menu.svg, and qr-logo.png, all with a last modified date of 2019-03-26 03:51. At the bottom of the screenshot, it says "Apache Server at quraswallet.org Port 443".</p>	
 <p>The screenshot shows a web browser window with the address bar displaying <a href="https://quraswallet.org/lib/">https://quraswallet.org/lib/</a>. The page title is "Index of /lib". Below the title is a table with columns: Name, Last modified, Size, and Description. The table lists several files including broadcast-tx.css, common.css, index.css, language.css, paper-wallet.css, send-quras.css, smart-contract.css, and unlock-your-wallet.css, all with a last modified date of 2019-07-06 02:07. At the bottom of the screenshot, it says "Apache Server at quraswallet.org Port 443".</p>	
Recommendation	<ul style="list-style-type: none"> <li>Configure your web server to prevent directory listings for all paths beneath the web root;</li> <li>Place into each directory a default file (such as index.htm) that the web server will display instead of returning a directory listing.</li> </ul>
References	<a href="https://cwe.mitre.org/data/definitions/538.html">https://cwe.mitre.org/data/definitions/538.html</a> <a href="https://cwe.mitre.org/data/definitions/548.html">https://cwe.mitre.org/data/definitions/548.html</a>

## TFI002 – HTTP Strict Transport Security (HSTS) Not Enforced

Risk Level	Info
Vulnerability Class	Data Protection

Status	Open		
CVSS v3.1 Rating	N/A	CVSS v3.1 Vectors	N/A
Location	<a href="https://quraswallet.org/new-wallet.html">https://quraswallet.org/new-wallet.html</a>		
Description			
<ul style="list-style-type: none"><li>• HTTP Strict Transport Security (HSTS) is an additional response header sent by the web server to instruct the user's browser to only communicate with it over HTTPS.</li><li>• The absence of the Strict-Transport-Security header in response headers can allow eavesdropping, man-in-the-middle and active network attacks.</li><li>• The risk of an attacker intercepting requests and responses, and/or downgrading them from a secure HTTPS to unencrypted HTTP connection is reduced with HSTS.</li><li>• The availability of pages outside secured context can cause legitimate users to believe that the session is secure, and therefore submit private information in clear text.</li><li>• It should be noted that HTTP was found to not be open during the assessment and therefore this finding carries an informational rating as currently there is no risk of this attack taking place. However, future changes to the operation of the application should unsecured content be served could allow an attack scenario.</li></ul>			
Evidence :			
<p>The following HTTP Request/Response pair shows that the server does not enforce "Strict-Transport-Security":</p>			
<p><b>Request:</b></p> <hr/> <pre>POST /server-php/index.php?c=Wallet&amp;a=newWallet HTTP/1.1 Host: quraswallet.org User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0 Accept: application/json, text/javascript, */*; q=0.01 Accept-Language: en-US,en;q=0.5 Accept-Encoding: gzip, deflate Content-Type: application/x-www-form-urlencoded; charset=UTF-8 X-Requested-With: XMLHttpRequest Content-Length: 29 Origin: https://quraswallet.org Connection: close Referer: https://quraswallet.org/new-wallet.html Cookie: roundcube_cookies=enabled; language=en-us; PHPSESSID=3a92e85d102b9dd15fce78ce25faab71  password=aaa&amp;token=1652009339</pre>			
<p><b>Response:</b></p>			

<pre> HTTP/1.1 200 OK Date: Tue, 05 Nov 2019 05:18:48 GMT Server: Apache Expires: Thu, 19 Nov 1981 08:52:00 GMT Cache-Control: no-store, no-cache, must-revalidate Pragma: no-cache Upgrade: h2,h2c Connection: Upgrade, close Content-Length: 16 Content-Type: text/html; charset=UTF-8  {"success":true} </pre>	
Recommendation	<ul style="list-style-type: none"> <li>Consider implementing HTTP Strict Transport Security (HSTS) on the server. HTTP Strict Transport Security (HSTS) is a web security policy mechanism that addresses this issue by informing web browsers to always interact with the site using only secure HTTPS connections.</li> <li>The policy is communicated by the server to the web browser via a HTTPS response header field named "Strict-Transport-Security" which is provided when the user visits the site over HTTPS. Once the server has provided the response, all subsequent requests to the domain from the client will enforce the use of HTTPS.</li> </ul>
References	<a href="https://www.owasp.org/index.php/HTTP_Strict_Transport_Security">https://www.owasp.org/index.php/HTTP_Strict_Transport_Security</a> <a href="http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security">http://en.wikipedia.org/wiki/HTTP_Strict_Transport_Security</a>

### TFI003 – Server path disclosure

Risk Level	Info		
Vulnerability Class	Information Disclosure		
Status	Open		
CVSS v3.1 Rating	N/A	CVSS v3.1 Vectors	N/A
Location	<a href="https://api.guraswallet.org/%NETHOOD/">https://api.guraswallet.org/%NETHOOD/</a> <a href="https://blockapi.guraswallet.org:9009/">https://blockapi.guraswallet.org:9009/</a>		
Description	<ul style="list-style-type: none"> <li>Full Path Disclosure vulnerability enable the attacker to see the path to the webroot/file. Certain vulnerabilities, such as using the <code>load_file()</code> (within a SQL Injection) query to view the page source, require the attacker to have the full path to the file they wish to view.</li> <li>This information can help an attacker identify other vulnerabilities and can be used to conduct further attacks.</li> </ul>		

### Evidence :

The following request/response shows server path disclosed on <https://api.guraswallet.org/%NETHOOD/>

**Request:**

```
GET /%NETHOOD/ HTTP/1.1
Host: api.quraswallet.org
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

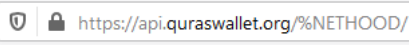

**Response:**

```
HTTP/1.1 400 Bad Request
X-Powered-By: Express
Content-Security-Policy: default-src 'self'
X-Content-Type-Options: nosniff
Content-Type: text/html; charset=utf-8
Content-Length: 1176
Date: Wed, 13 Nov 2019 12:57:10 GMT
Connection: close
```

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Error</title>
</head>
<body>
<pre>URIError: Failed to decode param %39;/%NETHOOD/%39;<br> &nbsp; &nbsp;at
decodeURIComponent (&lt;anonymous&gt;)<br> &nbsp; &nbsp;at decode_param
(/home/ec2-user/quras/node_modules/express/lib/router/layer.js:172:12)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at Layer.match
(/home/ec2-user/quras/node_modules/express/lib/router/layer.js:123:27)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at matchLayer
(/home/ec2-user/quras/node_modules/express/lib/router/index.js:574:18)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at next
(/home/ec2-user/quras/node_modules/express/lib/router/index.js:220:15)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at expressInit
(/home/ec2-user/quras/node_modules/express/lib/middleware/init.js:40:5)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at Layer.handle [as handle_request]
(/home/ec2-user/quras/node_modules/express/lib/router/layer.js:95:5)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at trim_prefix
(/home/ec2-user/quras/node_modules/express/lib/router/index.js:317:13)<br> &nbsp; &nbsp;
&nbsp; &nbsp;at /home/ec2-user/quras/node_modules/express/lib/router/index.js:284:7<br> &nbsp; &nbsp;
&nbsp; &nbsp;at Function.process_params
(/home/ec2-user/quras/node_modules/express/lib/router/index.js:335:12)</pre>
</body>
</html>
```

Screenshot of actual view on the webpage





```
URIError: Failed to decode param '%NETHOOD/'
    at decodeURIComponent (<anonymous>)
    at decode_param (/home/ec2-user/quras/node_modules/express/lib/router/layer.js:172:12)
    at Layer.match (/home/ec2-user/quras/node_modules/express/lib/router/layer.js:123:27)
    at matchLayer (/home/ec2-user/quras/node_modules/express/lib/router/index.js:574:18)
    at next (/home/ec2-user/quras/node_modules/express/lib/router/index.js:220:15)
    at expressInit (/home/ec2-user/quras/node_modules/express/lib/middleware/init.js:40:5)
    at Layer.handle [as handle_request] (/home/ec2-user/quras/node_modules/express/lib/router/layer.js:95:5)
    at trim_prefix (/home/ec2-user/quras/node_modules/express/lib/router/index.js:317:13)
    at /home/ec2-user/quras/node_modules/express/lib/router/index.js:284:7
    at Function.process_params (/home/ec2-user/quras/node_modules/express/lib/router/index.js:335:12)
```

The following request/response shows server path disclosed on <https://blockapi.quraswallet.org:9009/>

**Request:**

```
GET / HTTP/1.1
Host: blockapi.quraswallet.org:9009
User-Agent: Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:70.0) Gecko/20100101 Firefox/70.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

**Response:**

```

HTTP/1.1 404 Not Found
x-powered-by: Express
access-control-allow-origin: *
access-control-allow-headers: X-Requested-With,content-type, Authorization
access-control-allow-methods: GET,PUT,POST,DELETE,OPTIONS
content-type: text/html; charset=utf-8
content-length: 1220
etag: W/"4c4-Qpmo7Ive1N14aS1oiT3kZ86KZMw"
date: Wed, 13 Nov 2019 12:58:39 GMT
connection: close

<!DOCTYPE html><html><head><title></title><link rel="stylesheet"
href="/stylesheets/style.css"></head><body><h1>Not
Found</h1><h2>404</h2><pre>NotFoundError: Not Found
    at C:\works\engine_20191022\quras-api-service\app.js:54:8
    at Layer.handle [as handle_request]
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\layer.js:95:5)
    at trim_prefix
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:317:13)
    at
C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:284:7
    at Function.process_params
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:335:12)
    at next
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:275:10)
    at C:\works\engine_20191022\quras-api-service\app.js:35:3
    at Layer.handle [as handle_request]
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\layer.js:95:5)
    at trim_prefix
(C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:317:13)
    at
C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:284:7</p
re></body></html>

```

← → ↻ 🏠  <https://blockapi.quraswallet.org:9009>

## Not Found

404

```

NotFoundError: Not Found
    at C:\works\engine_20191022\quras-api-service\app.js:54:8
    at Layer.handle [as handle_request] (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\layer.js:95:5)
    at trim_prefix (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:317:13)
    at C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:284:7
    at Function.process_params (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:335:12)
    at next (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:275:10)
    at C:\works\engine_20191022\quras-api-service\app.js:35:3
    at Layer.handle [as handle_request] (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\layer.js:95:5)
    at trim_prefix (C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:317:13)
    at C:\works\engine_20191022\quras-api-service\node_modules\express\lib\router\index.js:284:7

```

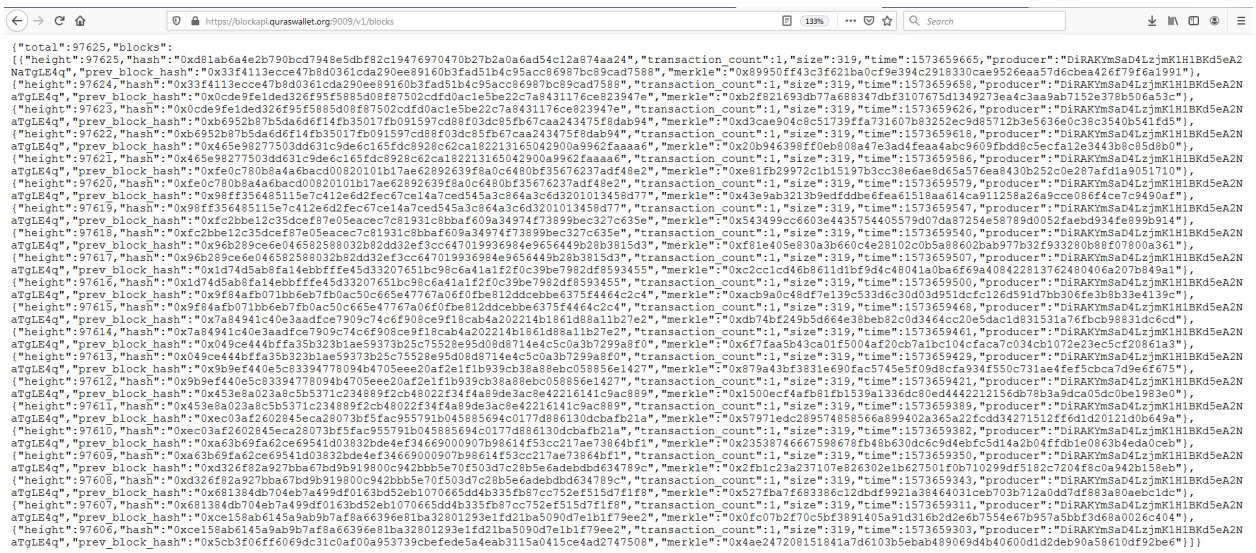
### Recommendation


- Path server disclosure should be disabled in response and remove the sensitive data from the output. (e.g. at decode\_param (/home/ec2-user/quras/node\_modules/express/lib/router/layer.js:172:12))
- Create a custom error message to capture stack errors

### References

[https://www.owasp.org/index.php/Full\\_Path\\_Disclosure](https://www.owasp.org/index.php/Full_Path_Disclosure)  
<https://cwe.mitre.org/data/definitions/200.html>

<https://cwe.mitre.org/data/definitions/209.html>**TFI004 – Information Disclosure via blockapi.quraswallet.org:9009/v1/**

Risk Level	Info
Vulnerability Class	Information Disclosure
Status	Open
CVSS v3.1 Rating	N/A
CVSS v3.1 Vectors	N/A
Location	<a href="https://blockapi.quraswallet.org:9009/v1/blocks">https://blockapi.quraswallet.org:9009/v1/blocks</a> <a href="https://blockapi.quraswallet.org:9009/v1/txs">https://blockapi.quraswallet.org:9009/v1/txs</a>
Description	<ul style="list-style-type: none"><li>Blockchain related information were publicly disclosed on the application.</li><li>This information can help an attacker gain more information on the web server. It allows remote attackers to retrieve sensitive information within the context of the application, via a crafted HTTP request</li><li>An attacker can obtain information such as:<ul style="list-style-type: none"><li>MinerTransaction</li><li>Merkle</li><li>Hash</li><li>transaction_count</li><li>producer</li></ul></li></ul>
Evidence :	<p>Screenshot showing information disclosed on <a href="https://blockapi.quraswallet.org:9009/v1/blocks">https://blockapi.quraswallet.org:9009/v1/blocks</a></p>  <p>Screenshot showing information disclosed on <a href="https://blockapi.quraswallet.org:9009/v1/txs">https://blockapi.quraswallet.org:9009/v1/txs</a></p>

	
<b>Recommendation</b>	<ul style="list-style-type: none"> <li>Remove the <i>txs</i> and <i>blocks</i> from production systems or web server if not needed.</li> <li>Make sure that your web server does not send out response headers or background information that reveal technical details about the backend technology type, version or setup.</li> <li>Always make sure that proper access controls and authorizations are in place in order to disallow access for attackers on all web servers, services and web applications.</li> </ul>
<b>References</b>	<a href="https://cwe.mitre.org/data/definitions/200.html">https://cwe.mitre.org/data/definitions/200.html</a>

## TFI005 – Non-secure requests are not automatically upgraded to HTTPS

Risk Level	Info		
Vulnerability Class	Inadequate Encryption Strength		
Status	Open		
CVSS v3.1 Rating	N/A	CVSS v3.1 Vectors	N/A
Location	<a href="http://quraswallet.org/">http://quraswallet.org/</a>		
Description			
<ul style="list-style-type: none"><li>• Non-secure requests to quraswallet (e.g. <a href="http://quraswallet.org/">http://quraswallet.org/</a>) are not automatically upgraded to HTTPS.</li><li>• The application allows users to connect to it over unencrypted connections. An attacker suitably positioned to view a legitimate user's network traffic could record and monitor their interactions with the application and obtain any information the user supplies. Furthermore, an attacker able to modify traffic could use the application as a platform for attacks against its users and third-party websites.</li><li>• Unencrypted connections have been exploited by ISPs and governments to track users, and to inject adverts and malicious JavaScript. Due to these concerns, web browser vendors are planning to visually flag unencrypted connections as hazardous.</li></ul>			
Evidence :			

## Steps to reproduce

- Using cURL send a HEAD request to <http://quraswallet.org> (e.g. `curl -I http://quraswallet.org`)
- The server does not instruct the client to upgrade the connection to HTTPS.
- The server responds with a 200 OK status code instead of 301 status code with the response header Location set to <http://quraswallet.org>.

```
$ curl -I http://quraswallet.org
% Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
           %         0         0         0           0         0 --:--:-- --:--:-- --:--:--    0HTTP/1.1 200 OK
Date: Wed, 13 Nov 2019 15:16:11 GMT
Server: Apache
Upgrade: h2,h2c
Connection: Upgrade
Last-Modified: Wed, 25 Sep 2019 04:34:59 GMT
Accept-Ranges: bytes
Content-Length: 16525
Vary: Accept-Encoding
Content-Type: text/html
```

## Recommendation

- Non-secure connections need to be upgraded to HTTPS as soon as possible using a permanent redirect.
- Applications should use transport-level encryption (SSL/TLS) to protect all communications passing between the client and the server. The Strict-Transport-Security HTTP header should be used to ensure that clients refuse to access the server over an insecure connection.

## References

<https://cwe.mitre.org/data/definitions/326.html>  
[https://wiki.mozilla.org/Security/Server\\_Side\\_TLS](https://wiki.mozilla.org/Security/Server_Side_TLS)

## 4. About CertiK

CertiK is a blockchain cybersecurity company with the global headquarter in New York City and presence in Beijing, Seattle, Seoul and Tokyo, pioneering the use of cutting-edge technologies, including static & dynamic analysis, Formal Verification, and Penetration Testing. CertiK has received grants from IBM, the Qtum Foundation, and the Ethereum Foundation to support its research of improving security across the blockchain industry. CertiK also contributes to the technical communities and ecosystems by providing guidance, research, and advisory about blockchain and smart contract best practices.

Our Penetration Testing service envisions to empower individuals and businesses to thrive in the new digital security age, especially in the blockchain space.