Security Analysis of the IOTA Trinity Wallet

Comprehensive security analysis of IOTA wallet Desktop app

17.07.2018

Einstufung des Dokumentes: | Classification of document:

Öffentlich | Intern | Vertraulich | Streng vertraulich | Vertrauliche Kundeninformationen

Public | Internal use | Confidential | Strictly confidential | Client confidential

Auf Basis der in diesem Dokument enthalten Informationen zu Arbeitsweisen und Methodologie der accessec erfolgt die Einstufung „vertraulich“. Die Darstellung von kritischen Informationen des Kunden führen darüber hinaus zur Einstufung „vertrauliche Kundeninformationen“.

Based on contained information about operation and methodology from accessec, this document gets the classification “confidential”. The presentation of critical information from customer side lead to the additional classification “client confidential”.

# Table of Contents

1. Introduction .......................................................................................................................... 3  
   1.1 Purpose of this report ......................................................................................................... 3  
   1.2 Reading guide .................................................................................................................... 3  
   1.3 Revision history ................................................................................................................ 3  
2. Goals and Testing methodology ............................................................................................. 4  
   2.1 Goals of the security audit .................................................................................................. 4  
   2.2 Methodology ..................................................................................................................... 4  
3. Detailed Results ..................................................................................................................... 4  
   3.1 Desktop ............................................................................................................................ 4  
      3.1.1 Testing scope and out of scope .................................................................................... 4  
   3.2 Analysis and Issues found .................................................................................................. 6  
      3.2.1 Static Analysis results ................................................................................................. 6  
      3.2.2 Dynamic Analysis results .......................................................................................... 6  
4. Conclusion and recommendations ......................................................................................... 7
1. Introduction

A security assessment for the IOTA "Trinity" wallet project desktop version was carried out by accessec GmbH, between July 2nd and July 14th 2018.

The target of the assessment was to review the current security posture of the Trinity wallet app on Desktop platform. With a “best practice” approach and the expert experience of the accessec security team, accessec’s acta® methodology, Open Web Application Security Project (OWASP) Mobile Security Testing Guide and the SANS Guide for Mobile AppSec Verification were used to set up the test structure.

The assessment itself was carried out by accessec GmbH based on the abbreviated version of the acta® and SANS approaches with focus on Open Web Application Security Project (OWASP) Mobile Security Testing Guide, as only a reduced set of test cases was needed to assess the security of the Trinity wallet.

The assessment involved both static and dynamic analysis of current versions of the wallet.

1.1 Purpose of this report

The IOTA Foundation has provided a basic GUI interface to enable seamless and secure machine-to-machine interactions and transaction on the network. Then, some users have asked for a redesign of the wallet, to deal with some of the issues they face. For that reason, the IOTA Foundation has decided to satisfy the desire of the users by redeveloping the IOTA GUI wallet. However, significant changes have been made to the design over the last couple of months. To ensure that Trinity wallet meets security expectations IOTA hired external security audit company, accessec, to check Android, iOS and desktop Trinity wallet apps for security vulnerabilities before releasing it to the public. This report contains a summary of the findings from the security audit and details how these findings lead to changes in the Trinity wallet applications to mitigate the issues that were identified.

1.2 Reading guide

This document is organized as shown in the list below:

- *Chapter 2* introduces the goals set out for the security audit and gives a high-level overview of the methodology that accessec used to perform the audit.
- *Chapter 3* described the issues identified after the analysis and document the changes that should be made to make them less harmful.
- *Chapter 4* concludes the document with some general recommendations for IOTA Foundation wishing to deploy a more secure Version of Trinity wallet to the public.

1.3 Revision history

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Author</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>12.07.2018</td>
<td>Janis Kinast</td>
<td>Initial Document</td>
</tr>
<tr>
<td>1.0</td>
<td>17.07.2018</td>
<td>Sebastian Rohr</td>
<td>Final</td>
</tr>
</tbody>
</table>
2. Goals and Testing methodology

2.1 Goals of the security audit
To ensure that Trinity wallet meets security expectations IOTA hired external security auditor, accessec, to check Android, iOS and Desktop Trinity wallet apps for security vulnerabilities before releasing it to the public. The first audit was conducted during the month of April. The actual one was performed in May (iOS, Android) and July (Desktop, Windows) 2018 with the goal to help IOTA Foundation to make alterations to the Trinity wallet apps and deal with the small number of vulnerabilities that were identified in the security audit.

2.2 Methodology
Accessec mainly used the Open Web Application Security Project (OWASP) Mobile Security Testing Guide.

3. Detailed Results
The following section describes the results for each domain based on the defined topics for each domain. If there have been no good or bad findings this is indicated by -/-.

3.1 Desktop
During the assessment, the following areas were defined and analyzed as part of the "Security Management":

- Static analysis – an automatic static code analysis of the source Code for the Desktop, combined with a manual expert review of the code if bugs were indicated by automated analysis
- Dynamic analysis – communication and function calling of the installed App have been intercepted at runtime

3.1.1 Testing scope and out of scope
Scope
In the course of this audit the next testing scope has been defined following the MSTG:

1. Static code analysis of the Trinity Wallet. This analysis is done by an automated tool called SonarQube. The results of this analysis are evaluated in reference to the good security programming practices of OWASP.
2. Dynamic code analysis of the Desktop IOTA Wallet. This analysis is done with x64bdg.
3. Network traffic analysis of the Desktop IOTA Wallet. In this phase of the audit the traffic generated by the app at runtime is monitored with Wireshark.

During previous analysis of the Trinity wallet, no differences between the Mac, Linux and Windows versions were found. This re-test was only executed on the Windows platform to validate the issues found in previous versions have been successfully mitigated.
Out of scope

In this audit of the Trinity Wallet app for Desktop the following aspects are out scope:

- Manual code analysis of the application due to the lack of time
- Penetration testing: no active attacks, exploits or code injection attempts to the app have been executed
- Environment dependencies from Electron

Testing environment

The security audit of the Trinity Wallet application for Desktop has been carried out under the following testing conditions:

<table>
<thead>
<tr>
<th>Devices</th>
<th>OS</th>
<th>Applications installed</th>
<th>Characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pc</td>
<td>Win 10</td>
<td>• Wireshark</td>
<td>• X64dbg</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Visual Studio Code</td>
<td>• SonarQube 7.0</td>
</tr>
</tbody>
</table>

Testing tools

For the completion of the security audit of the Desktop IOTA Wallet the following testing tools have been used:

- **SonarQube**: an automated, all-in-one static Code analyzer capable of performing static analysis. It can be used for effective and fast security analysis of Code in different Languages like Java, JavaScript, C, Typescript etc. indicating Code Smalls, Bugs and vulnerabilities (CWE, SANS TOP 25 and OWASP TOP10).
- **Wireshark**: a free and open source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education

Testing subject

The Desktop Trinity Wallet application is programmed with React. This application was tested by the engineer in two different ways:

- **Static analysis**: Source Code
- **Dynamic analysis**: Installed executable


3.2 Analysis and Issues found

3.2.1 Static Analysis results

The static analysis was performed on the source code from the Gitlab repository (iotedledger/trinity-wallet/tree/desktop-next/src/desktop) between the 02nd – 13th of July 2018. The first approach was an automated static analysis with the latest SonarQube version, to spot Bugs and vulnerabilities. The Report indicated that there are some findings regarding code-style but no vulnerabilities in the code. The Files flagged with code-style Errors got manually reviewed and revealed no security issues. As shown by example in the following screenshot.

Tools used

SonarQube 7.0

Notepad++

Visual Studio Code with sonarqube-inject and ESLint

Issues found

<table>
<thead>
<tr>
<th>Finding</th>
<th>File/Library</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>No findings with the static analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

3.2.2 Dynamic Analysis results

The static analysis was performed on the executables from the URL https://desktop-testbuilds-sakawne.iota.works/ with the version 0.1.9. between the 02nd – 13th of July 2018. The application received has been installed and analyzed. Therefore, the x64dbg debugger was attached to the running process to view the output/memory dumps while running through the application while the network traffic was analyzed with Wireshark. All traffic was SSL/TLS encrypted and, in the output/memory dumps no cleartext passwords or seed was found.

Tools

x64dbg

Wireshark
HTTPS

As in the previous audit, a traffic analysis of the IOTA Trinity Wallet has been performed. As recommend all traffic is now secured with SSL/TLS as shown by the following screenshot.

Issues found

<table>
<thead>
<tr>
<th>Finding</th>
<th>File/Library</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>No findings with the dynamic analysis</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4. Conclusion and recommendations

All issues identified during the first analysis of the Trinity wallet have been solved or remediated as far as the analysis has shown. We wish the IOTA Foundation and the Trinity wallet much success and continued growth of the community.

Darmstadt July 17th 2018

Sebastian Rohr
CEO accessec GmbH