Security research of the mobile application

Your Application Name

Customer:	Your Company	John Dow john@company.com
Performer:	Valletta Software Development	Mike Vazovsky sales@vallettasoftware.com
Report Date:	10/12/2023	

Valletta
Software
Development

Contents

Terms and Definitions 3

General information about the project and conclusions 4

- Information about the project 4
- General information and recommendations on the results of work 4

Research program and methodology 6

- Methodology for analyzing the security of a mobile Application 6
- The principle of forming the risk level for detected vulnerabilities 7

Analysis of the security of the mobile application 9

- Overall results 9
- **Information on identified vulnerabilities** 9
  - Sensitive data in logs 12
  - Lack of iOS User Input Protection Mechanism 13
  - Insecure format of PIN codes used 14
  - No app lock when running in the background 15
  - No revocation of the user session when exiting the application 16
  - Lack of SSL certificate 17
  - Lack of an obfuscation mechanism for the source code of the application 18

Results of inspections 21

- Android App Signature 21
- iOS app signature 21
- Result of verification of the platform's iOS protection mechanisms 22

Terms and definitions

Investigated application (application)	Your Application Name mobile app for iOS and Android platforms
Keylogger	A type of malware that reads user input
Cache	Intermediate temporary buffer for storing operational information
Obfuscation (source code)	Bringing the source code or executable code of the program to a form that preserves its functionality but complicates the analysis, understanding of operating algorithms, and modification during decompilation
OS	Operating system
Encryption certificate	A digital or paper document confirming the correspondence between the public key and the information identifying the owner of the key
Exploit	A program that an attacker uses to take control of a targeted system
Jailbreak	Getting superuser privileges on the device
Root access	Getting superuser privileges on the device
MiTM	(Man-in-The-Middle) is a type of attack based on listening to network traffic

General information about the project and conclusions

Information about the project

Work on the analysis of the security of the mobile application "Your Application Name" was carried out for the organization "NONAME" by Valletta Software Development
The purpose of the study is a comprehensive independent study of the security of the Your Application Name mobile application for the Android and iOS platforms.

General information and recommendations on the results of work

During the security analysis, vulnerability search techniques were used and simulated attacks were carried out that can be used by real attackers to gain unauthorized access to confidential information. The testing made it possible to identify existing vulnerabilities in the application.

During the security analysis, the following was identified for the platforms:

- Android - 23 vulnerabilities: 1 - critical level, 3 - high level, 5 - medium, and 14 - low severity.
- iOS - 21 vulnerabilities: 1 - critical level, 2 - high level, 5 - medium, and 13 - low risk.

High-risk vulnerabilities pose the greatest threat, and their exploitation can allow a potential attacker to gain access to sensitive user data (for example, the user's wallet).
The rest of the detected vulnerabilities do not pose a significant threat at the moment but may cause severe problems in the future. The overall risk of using these flaws to carry out attacks in most cases is further mitigated by the fact that an attacker would need physical access to the user's device.

For all detected vulnerabilities and deficiencies, the main part of the report provides technical recommendations for their elimination or reduction of potential threats.

Performing these actions will increase the level of protection against the impact of a potential attacker.
The recommendations presented in this report are not intended to be exhaustive due to the nature of the test model. Moreover, the organization of a sufficient level of security of the application can be implemented in various ways.

To increase the level of security of the mobile application, we recommend:

- use mechanisms for detecting work on devices with superuser rights;
- use checklists with secure programming best practices, for example, OWASP Secure Coding Practices Checklist;
- (https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_Checklist). This will help not only to eliminate existing vulnerabilities but also to prevent their occurrence in the future;
- implement processes in the application lifecycle to ensure that vulnerabilities are detected and remedied both at the source code development stage and at the application operation stage. This will reduce potential risks and increase the level of protection;
- re-analyze the security of the application to confirm that these recommendations have been applied correctly and that all detected vulnerabilities have been resolved. Re-analysis can be carried out in the form of random control (checking the correctness of the elimination of vulnerabilities noted in the report) or re-checking all classes of vulnerabilities.

Based on the results obtained, the overall level of security of the "Your Application Name" mobile application for the platform:

- iOS is at an average level
- Android is at an intermediate level

A low level of application security indicates the need to implement a security process at all stages of the development lifecycle. Security analysis should be carried out on a regular basis and include monitoring the elimination of previously identified vulnerabilities.

Research program and methodology

Methodology for analyzing the security of a mobile application

The analysis of the security of the mobile application was carried out on the basis of the adapted OWASP methodology. The work was carried out in accordance with the following methodology:

- Testing for secure data storage
  - Searching for sensitive data in event logs
  - Testing sensitive input fields for disabled keyboard cache
  - Testing for the possibility of leakage of sensitive data through interprocessor interoperability (IPC) mechanisms

- Testing authentication and session management mechanisms
  - Investigation of the "logout" function
  - Testing the password policy
  - Testing to limit authentication attempts
  - Investigation of the correctness of the session timeout

- Data Transmission Security Testing
  - Analyze TLS configuration
  - Analyzing the correctness of X.509 certificate authentication
  - Verifying the Implementation of Pinning's Own Certificate Stores and SSL Security Mechanism
- Testing Build Configuration and Unsafe Programming Practices
  - To verify the debug mode of the application
  - Explore debug meta-information

- Reverse Engineering Resistance Testing
  - Analysis of the mechanisms of obfuscation of the source code of the application

The principle of forming the risk level for detected vulnerabilities

The risk level for all detected vulnerabilities is determined based on the final value in the risk matrix. The matrix is formed on the basis of two independent criteria: Impact and Likelihood, which, by multiplication, determine the value of the final level of risk.

The "Impact" criterion is an approximate indicator of how critical the consequences of exploiting a vulnerability can be: what impact it can have on the technical condition of a system (data) or a group of systems, as well as on the integrity and continuity of business processes. Impact is calculated on a ten-point scale from 0 to 10 depending on how much impact the vulnerability may have. The higher the score that is assigned to the indicator, the higher the negative consequences.

The "Likelihood" criterion is an approximate indicator of how likely it is that a potential attacker has discovered and exploited a vulnerability. There are a number of factors that help determine probability: ease of detection and operation, as well as the required skill level of the attacker. The probability is calculated on a ten-point scale from 0 to 10, depending on how difficult it is to identify the vulnerability and exploit it. The higher the score that is assigned to the indicator, the easier it is to identify the vulnerability and conduct an attack based on it.
The tables provide detailed information on the ranges of values of both criteria.

The range of values of the "Impact" criterion

Magnitude	Designation
from 1 to 3	As a result of exploiting the vulnerability, a small amount of non-critical information may be disclosed, minimal damage to "insignificant" data, or a short-term lack of access to secondary services and systems
from 4 to 7	As a result of exploiting the vulnerability, the attacker is not given full control, but sensitive information may be disclosed, sensitive data may be highly damaged, or prolonged lack of access to secondary and essential services and systems
from 8 to 10	As a result of exploiting the vulnerability, confidential information may be disclosed, all data may be lost, or critical services and systems may be denied access

The range of values of the criterion "Likelihood" (Likelihood)

Magnitude	Designation
from 1 to 3	Detecting a vulnerability will require a detailed analysis of the architecture and configuration of the application and/or system. Operation is significantly complicated and/or may require a lot of time and certain conditions
from 4 to 7	Detecting a vulnerability will require a detailed analysis using high programming skills and knowledge of network interaction. Exploitation of the vulnerability may be difficult due to the peculiarities of the technical implementation
from 8 to 10	No technical skills are required to detect a vulnerability, or a vulnerability is detected by automated tools. The vulnerability is easily exploited or exploited by automated tools

After determining the values of Probability and Impact, a risk matrix is formed and the final level of risk is calculated. Information on the ranking of the risk level is given in the table to the right of the matrix:

Impact	10	10	20	30	40	50	60	70	80	90	100
	9	9	18	27	36	45	54	63	72	81	90
	8	8	16	24	32	40	48	56	64	72	80
	7	7	14	21	28	35	42	49	56	63	70
	6	6	12	18	24	30	36	42	48	54	60
	5	5	10	15	20	25	30	35	40	45	50
	4	4	8	12	16	20	24	28	32	36	40
	3	3	6	9	12	15	18	21	24	27	30
	2	2	4	6	8	10	12	14	16	18	20
	1	1	2	3	4	5	6	7	8	9	10
	0	1	2	3	4	5	6	7	8	9	10
	Likelihood										

Final Risk Value	Risk level
0 — 19	Low
20 — 40	Medium
41 — 72	High
73 — 100	Critical

Analysis of the security of the mobile application

Overall results

<table>
<tr><td>List of works</td><td colspan="2">Validation result</td></tr>
<tr><td colspan="3">Testing for secure data storage</td></tr>
<tr><td rowspan="2">Searching for sensitive data in event logs</td><td>Android</td><td>iOS</td></tr>
<tr><td colspan="2">Vulnerability "Sensitive data in logs" has been detected</td></tr>
<tr><td rowspan="2">Testing sensitive input fields for disabled keyboard cache</td><td>Android</td><td>iOS</td></tr>
<tr><td>Sensitive data is not stored when typing</td><td>Vulnerability "Lack of iOS User Input Protection Mechanism" has been detected</td></tr>
<tr><td rowspan="2">Testing for the possibility of leakage of sensitive data through interprocessor interoperability (IPC) mechanisms</td><td>Android</td><td>iOS</td></tr>
<tr><td colspan="2">No vulnerabilities were found in IPC</td></tr>
<tr><td colspan="3">Testing authentication and session management mechanisms</td></tr>
<tr><td rowspan="2">Investigation of the "logout" function</td><td>Android</td><td>iOS</td></tr>
<tr><td colspan="2">A vulnerability has been identified "Failure to revoke user session when exiting the application"</td></tr>
<tr><td rowspan="2">Testing the password policy</td><td>Android</td><td>iOS</td></tr>
<tr><td colspan="2">An insecure PIN format vulnerability has been identified
A "Not blocking an application when running in the background" vulnerability has been detected</td></tr>
<tr><td rowspan="2">Testing to limit authentication attempts</td><td>Android</td><td>iOS</td></tr>
<tr><td colspan="2">Brute force is limited
After 3 PIN attempts - unlogin
OTP 3 attempts - block for an hour, 9 - ban numbers</td></tr>
</table>

Investigation of the correctness of the session timeout	Android	iOS
	The session timeout is implemented correctly	
Data Transmission Security Testing		
Analyzing the correctness of X.509 certificate authentication	Android	iOS
	A "Missing SSL certificate" vulnerability has been detected	
Verifying the Implementation of Pinning's Own Certificate Stores and SSL Security Mechanism	Android	iOS
	A "Missing SSL certificate" vulnerability has been detected	
Analyze TLS configuration	Android	iOS
	With the help of the testssl.sh utility, the following vulnerabilities and flaws were discovered: TLS 1 TLS 1.1 Triple DES Ciphers / IDEA Obsoleted CBC ciphers (AES, ARIA etc.) TLS 1.2 sig_algs offered: RSA+SHA1 OCSP stapling DNS CAA RR (experimental) Strict Transport Security SWEET32 (CVE-2016-2183, CVE-2016-6329) BEAST (CVE-2011-3389) LUCKY13 (CVE-2013-0169), experimental These vulnerabilities do not pose serious risks, but it is recommended that they be addressed.	
Testing Build Configuration and Unsafe Programming Practices		
To verify the debug mode of the application	Android	iOS
	Debug mode is turned off	
Explore debug meta-information	Android	iOS
	No debug meta information was found	
Reverse Engineering Resistance Testing		
Analysis of the mechanisms of obfuscation of the source code of the application	Android	iOS
	Vulnerability "Lack of Application Source Code Obfuscation Mechanism" has been identified	