[REDACTED]

Vulnerability assessment and penetration testing

Abstract

This report documents the findings for the Web and Mobile applications Security assessment of the [REDACTED]

Danish Tariq

danishtariqar@gmail.com

Table of Contents

Executive Summary

Overview

This activity is engaged to conduct a Security Assessment of [REDACTED] [REDACTED]'s web wallet and mobile applications. The purpose of the engagement was to utilize active exploitation techniques in order to evaluate the security of the application against best practice criteria and to validate its security mechanisms and identify application level vulnerabilities.

Scope

I defined the following components as in scope
https://www.[REDACTED].io
https://wallet.[REDACTED].io
https://keys.[REDACTED].io
https://mail.[REDACTED].io
[REDACTED] mobile applications

Out of scope

Any third party services like Nodes (Uptimerobot), Bitport, Paperwallet were out of scope.

Web and Mobile Applications Security Assessment

The assessment provides a point-in-time security analysis and resultant recommendations for improving the security of the application and its environment and consisted of the following activities:

Owasp top 10

Everything is audited and everything is analyzed with respect to top financial system security standards.

Information gathering

Information Gathering techniques were used in conjunction with a review of application and support system documentation in order to gain a deep and thorough understanding of how the application works, what its purpose is and how it has been implemented.

Reconnaissance

Reconnaissance involved performing active assessment techniques in order to fingerprint the technologies and versions of software in use as well as mapping the available functionality of the application.

Communications Security and Cryptography

Communications Security and Cryptography implementations were analyzed in order to ensure that cryptography is appropriately used to protect the confidentiality and integrity of sensitive user data. Cryptographic algorithms, ciphers, key lengths and storage strategies were assessed to ascertain their effectiveness to withstand cryptanalysis attack.

Authentication Mechanisms

Authentication Mechanisms were examined to determine the effectiveness and resilience to subversion techniques.

Session Management

Session Management implementations were assessed and attempts were made to violate session state to become another valid user or to escalate privileges and Authorisation Access Controls that enforce authorisation levels for the application were analyzed in detail to assess the user segregation methods employed and to validate their effectiveness.

Audit

Mobile application

File Information

Name	com-[REDACTED]-wallet-android1538107200.apk
Size	12.13MB
MD5	0a8a0c5823be2c52f8dec5cf40e68963
SHA1	67a78db796edd8103aedc55f478d93220e7c8a3c
SHA256	a4af8289c5d9cee75acd0302014180fca74893b2940d2e7e442490a4542fe7e7

App Information

Package Name	com.[REDACTED].wallet.android				
Main Activity	com.[REDACTED].wallet.android.MainActivity				
Target SDK	26	Min SDK	16	Max SDK	-
Android Version Name	2.0.1				
Android Version Code	27				

Components

Activities

com.[REDACTED].wallet.android.MainActivity
com.facebook.react.devsupport.DevSettingsActivity
com.google.android.gms.common.api.GoogleApiActivity
com.yalantis.ucrop.UCropActivity

Services

io.invertase.firebase.messaging.RNFirebaseMessagingService
io.invertase.firebase.messaging.RNFirebaseInstanceIdService
io.invertase.firebase.messaging.RNFirebaseBackgroundMessagingService
com.google.firebase.messaging.FirebaseMessagingService
com.google.firebase.components.ComponentDiscoveryService
com.google.android.gms.measurement.AppMeasurementService
com.google.android.gms.measurement.AppMeasurementJobService
com.google.firebase.iid.FirebaseInstanceIdService

Broadcast Receivers

com.google.android.gms.measurement.AppMeasurementReceiver
com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver
com.google.firebase.iid.FirebaseInstanceIdReceiver

Content Providers

android.support.v4.content.FileProvider
com.google.firebase.provider.FirebaseInitProvider

Manifest Analysis

ISSUE	SEVERITY	DESCRIPTION
Service (io.invertase.firebase.messaging.RNFirebaseMessagingService) is not Protected. An intent-filter exists.	High	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Service is explicitly exported.
Service (io.invertase.firebase.messaging.RNFirebaseInstanceIdService) is not Protected. An intent-filter exists.	High	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device. The presence of intent-filter indicates that the Service is explicitly exported.
Service (com.google.firebase.messaging.FirebaseMessagingService) is not Protected. [android:exported=true]	High	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.
Broadcast Receiver (com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver) is Protected by a permission. Permission: android.permission.INSTALL_PACKAGES [android:exported=true]	Info	A Broadcast Receiver is found to be exported, but is protected by permission.
Broadcast Receiver (com.google.firebase.iid.FirebaseInstanceIdReceiver) is Protected by a permission. Permission: com.google.android.c2dm.permission.SEND [android:exported=true]	Info	A Broadcast Receiver is found to be exported, but is protected by permission.
Service (com.google.firebase.iid.FirebaseInstanceIdService) is not Protected. [android:exported=true]	High	A Service is found to be shared with other apps on the device therefore leaving it accessible to any other application on the device.

VULNERABILITY FINDINGS

App Certificate Pinning Not Enforced

Issue detail

SSL Pinning makes sure that the client checks the server's certificate against a known copy of that certificate. If the server's SSL certificate is bundled inside the application, then an SSL request first validates that the server's certificate exactly matches the bundle's certificate.
When pinning is used, even when using a proxy with an imported certificate, an attacker will not be able to see sensitive data. This occurs because the imported certificate does not match the certificate that was bundle/pinned with the application.

Remediation

Bundle the server's SSL certificate with the application and enforce SSL pinning to ensure that the certificate is validated before establishing a secure connection. This will make it more difficult for a successful man-in-the middle attack to occur.

Testing Process

A proxy certificate was used to tunnel the application data through a monitoring proxy. This was successful because app certificate pinning was not enable. The following is a recorded login sequence from a user accessing the app while there was an active man-in-the middle between the app and server:

[Redacted screenshot here]

Improper user authentication and authorization

Issue detail

The app does not properly authenticate and/or authorize user, since authentication tokens can be viewed as plain-text by using a proxy to intercept session tokens. These tokens can then be re-used to query the API.

Testing Process

The HTTP request header below shows the request intercepted by the proxy.

HTTP Request

POST /api/createBtcWallet HTTP/1.1
Host: keys.[REDACTED].io
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:64.0) Gecko/20100101 Firefox/64.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://wallet.[REDACTED].io/home.html
Content-Type: application/json; charset=utf-8
authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6ImFocHRhaHB0YWhwdEBnbWFpbC5jb20iLCJyb2xlIjoidXNlciIsImlhdCI6MTU0ODc4NjY5NCwiZXhwIjoxNTQ5NTA2Njk0fQ.GE5aFa8bxQOjCqzmUsbzaEMLUTWEBrj3qDZ-NEnsP_s
fl_auth_version: 4
Content-Length: 289
Origin: https://wallet.[REDACTED].io
Connection: close

{"sessionToken":"eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6ImFocHRhaHB0YWhwdEBnbWFpbC5jb20iLCJyb2xlIjoidXNlciIsImlhdCI6MTU0ODc4NjY5NCwiZXhwIjoxNTQ5NTA2Njk0fQ.GE5aFa8bxQOjCqzmUsbzaEMLUTWEBrj3qDZ-NEnsP_s","publicKey":"q2Y+k6cYMHSMYc7VEIO6MMimTJHJHMRDTA+OehXNTHo=","appId":"[REDACTED]"}

Remediation

Pin the SSL certificate so that plain-text tokens cannot be viewed.

Compromise of user account

Issue detail

The app's request and response can be intercepted using burp's proxy feature. Due to which usernames and passwords can be intercepted in plain-text.

Testing Process

Username and password can be sniffed enabled proxy as shown in the screenshot below.

[Redacted screenshot here]

HTTP Request

POST /api/login HTTP/1.1
accept: application/json
Content-Type: application/x-www-form-urlencoded; charset=utf-8
Content-Length: 452
Host: mkeys.[REDACTED].io
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.11.0

email=ahptahptahpt%40gmail.com&password=Password123&g_recaptcha_response=03AF6jDqV51ivBR1NbvxdZNrwVtd5pEKwT7Bz6AySSVFeP6V2H5JGbi0sqvUve2T3TB2MLE5PGU4W3YgmvaWom_op2f2ks3gqn0SuKDy8C6FHdX1tSF5qOxmYHA17zAOPs7KfAB1nmtGw0-XXIxU01CpfwlXUpGfkKU09D1qh-M8CEukO8OBdrb9FIVIvdlNUd2L1A1UwtZX4XwoEichLA38V3Mx9LbHpsQioEHoN4VOK1iOlBkfvW-RqS6H3qsMNteSQI0Em1ZeofuxcG45mG0mwklVvRhpdfpRUYAYWQbkYRo9pxUce_dsnmuIwAAswq3GQ-LasLMzZUqS_ZMtGNXod6SeYorJ1IfA&appversion=2.0.1&res=app

Remediation

Pin the SSL certificate to accept from and send requests to valid host endpoints.

No Root Check On Android Application

Issue detail

No detection is implemented in the application to determine whether the Android device has been ‘rooted’. This allows a user to modify an app on a rooted Android device, inducing behaviors that otherwise would not occur. Examples include removing or overwriting critical functions, and more. This would leave the application more vulnerable and allow an attacker to more easily exploit it.

Remediation

Implement ‘root’ detection before beginning the runtime of your application, such as looking for the presence of files and packages specific to a ‘rooted’ device.

Testing Process

This was discovered by running the app on both normal and rooted Android devices.

User Profile Name can be changed using same token

Issue detail

The User Profile Name can be changed using the same request repeatedly. Once an attacker has valid token he can use it to change user's Profile Name again and again.

Testing Process

Capturing the user profile Name change HTTP request will allow to use it to change Profile Name number of times with same request captured as showed in the screenshot below:

[Redacted screenshot here]

Figure 1The Profile Name Before

[Redacted screenshot here]

Figure 2 Change Profile Name to Danish with captured request

[Redacted screenshot here]

Figure 3Logout and Login again to see Profile Name changed to Danish

HTTP Request

POST /api/update-profile HTTP/1.1
authorization:
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJlbWFpbCI6ImFocHRhaHB0YWhwdEBnbWFpbC5jb20iLCJyb2xlIjoidXNlciIsImlhd
CI6MTU0ODc4NzQ4NSwiZXhwIjoxNTQ5NTA3NDg1fQ.NPuPh-dhoGwNYGbhQBP-cdPXDwsWZL70aY5T3FmjScI
fl_auth_version: 4
Content-Type: application/json; charset=utf-8
Content-Length: 66
Host: keysstg.[REDACTED].io
Connection: close
Accept-Encoding: gzip, deflate
Cookie: __cfduid=deb6d1d8713be31eb9810ff606d945e211548787435
User-Agent: okhttp/3.8.1

{"display_name":"Danish","appversion":"1.9.6-stage.9","res":"app"}

Web applications

Session Not Expired On Logout

Session management vulnerability in [REDACTED]'s website. i.e. user's session is not expiring immediately after the logout. An attacker can get the user's session cookies by using Session Spoofer, Cookie Staler etc. and thus, can get the access to the user account.

Steps To Reproduce:

-Login into your wallet.

-Capture any request. For example Account Settings using Burp Proxy.

-Logout from the wallet.

-Replay the request captured in step 2 and notice it displays the proper response.

1- Log in to your web wallet and change your name in settings (intercept the request and response with some proxy like Burp suite)

[Redacted screenshot here]

You will get this response (Proving that our request is fulfilled)

[Redacted screenshot here]

2- Now log out from the wallet and send the request we copied earlier (after changing the name parameter)

Request and response even after logging out :

Request is being responded and session is not expired even after logging out from the wallet

[Redacted screenshot here]

For financial platforms like [REDACTED], this type of flaw is not tolerable and should be patched immediately as it can be harmful for [REDACTED] users.

Remedy:

Proper mechanism should be implemented through which session should be expired immediately after logging out. More @ https://www.owasp.org/index.php?title=Broken_Authentication_and_Session_Management&setlang=en

Old version of Zimbra being used

[REDACTED] is using open source email solution called Zimbra @ mail.[REDACTED].io .
Latest stable version is Zimbra is 8.8.11 but [REDACTED] is using 8.7.0 which can be checked when someone from [REDACTED] send the email.

Proof of concept :-

[Redacted screenshot here]

This version is vulnerable to almost every vulnerability which can defame [REDACTED]'s integrity.

https://wiki.zimbra.com/wiki/Zimbra_Security_Advisories (Official security advisories)

Bug#	Summary	CVE-ID	CVSS Score	Zimbra Rating	Fix Release or Patch Version	Reporter
109093	XXE CWE-611	CVE-2018-20160	6.4	Major	8.8.9 Patch9 8.8.10 Patch5 8.8.11 Patch1	An Trinh
109017	Non-Persistent XSS CWE-79	CVE-2018-14013	4.3	Minor	8.8.9 Patch9 8.8.10 Patch5 8.8.11	Issam Rabhi of Sysdream
109020	Persistent XSS CWE-79	CVE-2018-18631	5.0	Major	8.7.11 Patch7 8.8.9 Patch7 8.8.10 Patch2	Netragard
109018	Non-Persistent CWE-79	CVE-2018-14013	2.6	Minor	8.7.11 Patch7 8.8.9 Patch6 8.8.10 Patch1	Issam Rabhi of Sysdream
109021	Limited Content Spoofing CWE-345	CVE-2018-17938	4.3	Minor	8.8.10	Sumit Sahoo
109012	Account Enumeration CWE-203	CVE-2018-15131	5.0	Major	8.7.11 Patch6 8.8.8 Patch9 8.8.9 Patch3	Danielle Deible
108970	Persistent XSS CWE-79	CVE-2018-14425	3.5	Minor	8.8.8 Patch7 8.8.9 Patch1	Diego Di Nardo
108902	Persistent XSS CWE-79	CVE-2018-10939	3.5	Minor	8.6.0 Patch11 8.7.11 Patch4 8.8.8 Patch4	Diego Di Nardo
108963	Verbose Error Messages CWE-209	CVE-2018-10950	3.5	Minor	8.7.11 Patch3 8.8.8	Netragard
108962	Account Enumeration CWE-203	CVE-2018-10949	5.0	Major	8.7.11 Patch3 8.8.8	Netragard
108894	Persistent XSS CWE-199	CVE-2018-	3.6	Minor	8.6.0 Patch10	Netragard

Remedy is to just update this Zimbra.

Timeline of previously reported and patched vulnerabilities

- Unrestricted file upload leading to XSS and HTML Injection.
- Out-dated version of Wordpress causing severe possible attacks and vulnerabilities
- Host header attack and web cache poisoning in forget password page
- Security flaws due to missing headers
- Insecure direct object reference
- IDOR in web wallet
- Privilege escalation of signing up with any email without registration
- Bypassing security questions when changing password
- Attacker can use any phone number (without verifying) and can get it verified by bypassing (privilege escalation)
- Sign up token can be hijacked via request during sign up poisoning
- MITM attack