SEC Consult
ADVISOR FOR YOUR INFORMATION SECURITY

Report 1907974

Source Code Review ProtonMail Android App

for

Proton Technologies AG

conducted by

SEC Consult

Version: 1.2 | **Date:** 2019-11-15
Responsible: SEC Consult | **Author**: SEC Consult
Confidentiality class: Public

Table of Contents

1 Management Summary

The following chapter summarizes the scope and timetable of the code review, the results of the code review, and outlines the measures recommended by SEC Consult.

1.1 Scope and Timetable

During the initial security assessment for Proton Technologies AG, SEC Consult performed a source code review of the ProtonMail client for Android - a secure email app for Android devices, which offers easy-to-use email encryption by seamlessly integrating PGP end-to-end encryption. Objective of the review was to reveal security issues and to offer suggestions for improvement. The focus of the code review was to provide answers to the following questions:

- Is an attacker able to break end-to-end encryption provided by ProtonMail solution?
- Is an attacker able to access data of other customers (cross-tenant access)?
- Is an attacker able to use paid ProtonMail features without an account upgrade?

The initial review was conducted in Q1 2019 and a total effort of 6 days was dedicated to identifying and documenting security issues in the code base of the ProtonMail Android App.

Version 1.12.0 of the application was tested. Full access to the source code was granted and test user credentials of the roles “free”, “plus”, “professional”, and “visionary” were provided.

The following files and documents were made available in the course of the review:

Files	SHA1 Sum
ProtonMail.zip	dbd3dea62c1429fe8489562897dea92406d19855
ProtonMail-Android-1.12.0-playstore-releasePlayStore.apk	9d9a13bfbbda7b99aaab65d9672622e330887bc7
README.md	ae76f59ab629132461256f132db927d110298a69

In September 2019, Proton Technologies AG fixed the identified issues and supplied the fixes to SEC Consult for verification. Goal of the fix verification was to confirm remediation provided by the applied fixes. SEC Consult verified the fixes in October 2019.

1.2 Results

During the initial code review, SEC Consult found one **medium-risk vulnerability** and three **low-risk vulnerabilities** in the reviewed source code and the mobile app.

Although issues with certificate validation have been identified within the encrypted communication between the mobile application and the backend system, the inner layer of end-to-end encryption could not be broken.

No issues were identified, which would provide an attacker unauthorized access to other customers' data without having physical access to the victim's device. An attacker with physical access to a mobile device can obtain user-related information from debug routines, as excessive debug messages contain various user-related information that can be easily accessed by an attacker.

Despite the fact that the app deserializes data without any validation, it was not possible to exploit the issue during the timeframe of the test.

Due to an insecure validation scheme, a mobile user can use advanced ProtonMail features on his mobile device. Therefore, an attacker can use paid ProtonMail features without an account upgrade.

All security issues that were identified in the initial code review were properly fixed or accepted by Proton Technologies AG.

1.3 Disclaimer

At the request of Proton Technology AG, this report has been declassified from strictly confidential to public. While the report was shortened for public release, relevant vulnerability information has been maintained.

In this particular project, a timebox approach was used to define the consulting effort. This means that SEC Consult allotted a prearranged amount of time to identify and document vulnerabilities. Because of this, there is no guarantee that the project has discovered all possible vulnerabilities and risks.

Furthermore, the security check is only an immediate evaluation of the situation at the time the check was performed. An evaluation of future security levels or possible future risks or vulnerabilities may not be derived from it.

2 Vulnerability Summary

This chapter contains all identified vulnerabilities in the reviewed source code of the company Proton Technologies AG.

Risk assessment	Initial no. of vulnerability classes	Current no. of vulnerability classes
Low	3	0
Medium	1	0
High	0	0
Critical	0	0
Total	**4**	**0**

2.1 Total Risk Per System

The following table contains a risk assessment for each system which contained security flaws.

System	Field of application	Initial risk	Current risk
ProtonMail Android App	Mobile	Medium	-
Total	**-**	**Medium**	**-**

2.2 Risk of Each Vulnerability

The following table contains a risk assessment for the discovered vulnerabilities.

Vulnerability	System	Initial risk	Current risk	Page
Potentially Insecure (De)serialization	ProtonMail Android App	Medium	ACCEPTED	7
Account Upgrade Bypass	ProtonMail Android App	Low	ACCEPTED	10
Debug Messages Enabled	ProtonMail Android App	Low	ACCEPTED	11
Missing Certificate Pinning	ProtonMail Android App	Low	FIXED	12
Total	**-**	**Medium**	**-**	**-**

3 Detailed Analysis

This chapter outlines the attacks and found vulnerabilities in detail.

3.1 ProtonMail Android App

3.1.1 General Information

This section describes vulnerabilities found in the ProtonMail Android App.

ProtonMail Android App is designed for Android based mobile devices and provides ProtonMail email capabilities to mobile users. During the timeframe of the audit the ProtonMail Android App version 1.12.0 was tested using a rooted smartphone with Android OS 5.0. The tested Android app is written in Kotlin and Java.

3.1.2 Potentially Insecure (De)serialization - ACCEPTED

Android application has an exposed activity that accepts serialized data. Malicious apps deployed on the same device may call the exposed activity (via IPC calls) and pass malicious serialized data to it. If such untrusted data is deserialized without any validation it may allow a malicious app to escalate privileges, read sensitive data from other apps and even gain remote command execution on the targeted device in the context of the vulnerable app.

CVSS-v3 Base Score: 4.0 (Medium)

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

3.1.2.1 Recheck results

The ProtonMail Android application exposes ComposeMessageActivity intent that can be called by any app, no specific permissions are required:

```
Exported Activities:
    ch.protonmail.android.activities.composeMessage.ComposeMessageActivity
      Parent Activity:
ch.protonmail.android.activities.mailbox.MailboxActivity
      Permission: null
```

By analyzing the source code of the intent, multiple instances with data deserialization were identified:

File: \ProtonMail\app\src\main\java\ch\protonmail\android\activities\composeMessage\ComposeMessageActivity.java

Lines: 498-505

```
[…]
                if (extras.containsKey(EXTRA_MESSAGE_ATTACHMENTS)) {
                    attachmentsList =
extras.getParcelableArrayList(EXTRA_MESSAGE_ATTACHMENTS);
                    if (attachmentsList != null) {
                        for (LocalAttachment localAttachment :
attachmentsList) {
                            localAttachment.setAttachmentId("");
                            localAttachment.setMessageId("");
                        }
                    }
[…]
```

Lines: 817-823

```
[…]
    private void handleSendMultipleFiles(Intent intent) {
        List<Uri> uris =
intent.getExtras().getParcelableArrayList(Intent.EXTRA_STREAM);
        if (uris != null) {
            for (Uri uri : uris) {
                handleSendFileUri(uri);
            }
        }
[…]
```

Lines: 1370-1376

```
[…]
    public void onActivityResult(int requestCode, int resultCode, Intent
data) {
        if (requestCode == ADD_ATTACHMENTS_REQUEST && resultCode ==
```

```
RESULT_OK) {
            askForPermission = false;
            addingMoreAttachments = false;
            ArrayList<LocalAttachment> resultAttachmentList =
data.getParcelableArrayListExtra(AddAttachmentsActivity.EXTRA_ATTACHMENT_LIS
T);
            ArrayList<LocalAttachment> listToSet = resultAttachmentList !=
null ? resultAttachmentList : new ArrayList<>();
            composeMessageViewModel.setAttachmentList(listToSet);
[…]
```

During the timeframe of the audit no mechanisms were identified that would validate the serialized data that is passed to the above deserialization functions – this potentially allows malicious apps to pass serialized data to the intent in order to exploit insecure deserialization.

The issue was not exploited during the timeframe of the tests.

Statement Proton Technologies AG:
To allow the functionality of the ProtonMail Android App to send and receive all desired attachments types, we must accept any kind of files.

3.1.3 Account Upgrade Bypass - ACCEPTED

Specific functions of the mobile application require a user with an unpaid (Free) user account to upgrade to a paid (e.g. Plus) user account in order to be active. However, during the timeframe of the audit several functions were activated using an unpaid (Free) user account thus bypassing the requirement to pay to a service provider.

CVSS-v3 Base Score: 3.7 (Low)

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

3.1.3.1 Recheck results

During the security assessment it was not possible to reproduce the initial scenario.

Statement Proton Technologies AG:
The highlighted functionalities were not restricted on the back end by design in order to enable a good UX for users who upgrade subscription plans, or only cosmetic. As such, we consider it as severity of the lowest level.

3.1.4 Debug Messages Enabled - ACCEPTED

The ProtonMail Android App has debug messages enabled. It is a common practice to add debug routines to the code while developing an application. Often developers forget to remove these debug functions and deploy an application with enabled debugging features. During the audit timeframe it was identified that debug messages contain potentially sensitive data.

CVSS-v3 Base Score: 2.1 (Low)

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

3.1.4.1 Recheck results

The following output is an excerpt of the debug messages produced by the mobile application while sharing the existing contact via the app share function. The log catches the contact name information:

```
$ adb logcat -v threadtime | grep -i protonmail
[…]
09-04 19:49:23.571  2915  2942 D StatusBarManagerService: manageDisableList
userId=0 what=0x0 pkg=Window{857e6e7 u0
ch.protonmail.android/ch.protonmail.android.contacts.details.ContactDetailsA
ctivity} token=android.os.Binder@6986e34 which=1
09-04 19:49:24.423  2915 14038 I ActivityManager: START u0
{act=android.intent.action.SEND typ=text/x-vcard flg=0x1
cmp=org.thoughtcrime.securesms/.ShareActivity clip={text/x-vcard
U:content://ch.protonmail.android.provider/external_files/Android/data/ch.pr
otonmail.android/files/pentest.vcf} (has extras)} from uid 10248 pid 21239
on display 0
09-04 19:49:24.433  2915  2942 D StatusBarManagerService: manageDisableList
userId=0 what=0x0 pkg=Window{857e6e7 u0
ch.protonmail.android/ch.protonmail.android.contacts.details.ContactDetailsA
ctivity} token=android.os.Binder@6986e34 which=1
[…]
```

Statement Proton Technologies AG:
All debug logs from within the Proton applications have been removed; the remaining output is generated by loggers outside of the control of the app itself (e.g. system logs). Information that is shown there is classified as low sensitivity.

3.1.5 Missing Certificate Pinning - FIXED

Certificate Pinning allows mobile applications to verify that they are only connecting to a server over SSL/TLS which he is intended to. Furthermore, it is possible to verify, that the connection between client and server is end-to-end encrypted and not intercepted. This is ensured by embedding a hash of the server's certificate or a hash of the public key directly into the application.

During the process of establishing a connection to the server, the hash of the certificate/public key of the server is obtained and compared against the embedded hash of the certificate(s)/public key(s). If the retrieved hash of the certificate/public key is matching the locally stored hash of the certificate/public key the connection will be established, otherwise the connection will fail.

CVSS-v3 Base Score: 3.7 (Low)

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

3.1.5.1 Recheck results

During the security assessment it was not possible for an attacker to intercept and manipulate the communication between the mobile app and the backend server:

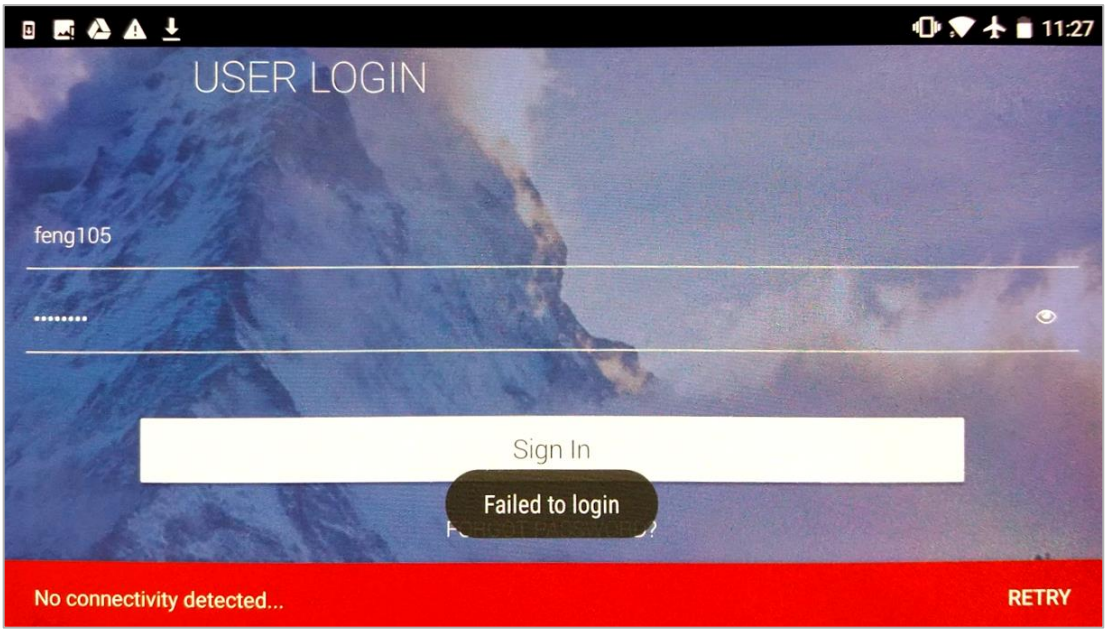

Figure 1. Certificate pinning is in place.

4 Version History

Version	Date	Status/Changes	Created by	Responsible
1.0	2019-03-15	Initial report	SEC Consult	SEC Consult
1.1	2019-10-10	Fix verification	SEC Consult	SEC Consult
1.2	2019-11-15	Public report	SEC Consult	SEC Consult