SEC Consult
ADVISOR FOR YOUR INFORMATION SECURITY

Report 1907974

Source Code Review ProtonMail iOS Mail App

for

Proton Technologies AG

conducted by

SEC Consult

Version: 1.2 | **Date:** 2019-10-29
Responsible: SEC Consult | **Author**: SEC Consult
Confidentiality class: Public

Table of Contents

Table of Contents **2**
1 Management Summary **3**
1.1 Scope and Timetable 3
1.2 Results 4
1.3 Disclaimer 4
2 Vulnerability Summary **5**
2.1 Total Risk Per System 5
2.2 Risk of Each Vulnerability 6
3 Detailed Analysis **7**
3.1 ProtonMail iOS Mail App 7
3.1.1 General Information 7
3.1.2 Account Upgrade Bypass - ACCEPTED 7
3.1.3 Hardcoded Credentials - FIXED 8
3.1.4 Debug Messages Enabled - PARTIALLY FIXED & ACCEPTED 9
3.1.5 Data Protection Class Is Not Specified - FIXED 10
3.1.6 Secure Backgrounding Is Not Fully Implemented - FIXED 11
3.1.7 Strongest Keychain Data Protection Class Not in Use - FIXED 13
3.1.8 Missing Certificate Pinning - FIXED 14
4 Version History **15**

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 iOS - a secure email app for iPhone and iPad, 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 iOS Mail App.

Version 1.11.12 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
Accounts.md	7723e8b7f097db66ceec1bfc0e4e7b0599d8c726
Before everything.md	d046ae71274063bd2ea6e0360ee15d563872f683
protonmail_ios-audit-develp.zip	6e754227324a074cd4cf94464c5d499eab99b05a
ProtonMailDev.ipa	8c285b13aa84b9453cbff21780923c6caedf97b0
README.md	79a5a990d0e68b8ca975267f00d4c82b227f5e39
Release notes.md	083a6f76d29eae3a071addfa9819d78b2b602649

In September 2019, Proton Technologies 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 seven **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. Additionally, it was found that the application did not take full advantage of security mechanisms provided by iOS.

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	7	0
Medium	0	0
High	0	0
Critical	0	0
Total	**7**	**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 iOS App	Mobile	Low	-
Total	**-**	**Low**	**-**

2.2 Risk of Each Vulnerability

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

Vulnerability	System	Initial risk	Current risk	Page
Account Upgrade Bypass	ProtonMail iOS App	Low	ACCEPTED	7
Hardcoded Credentials	ProtonMail iOS App	Low	FIXED	8
Debug Messages Enabled	ProtonMail iOS App	Low	PARTIALLY FIXED & ACCEPTED	9
Data Protection Class Is Not Specified	ProtonMail iOS App	Low	FIXED	10
Secure Backgrounding Is Not Fully Implemented	ProtonMail iOS App	Low	FIXED	11
Strongest Keychain Data Protection Class Not in Use	ProtonMail iOS App	Low	FIXED	13
Missing Certificate Pinning	ProtonMail iOS App	Low	FIXED	14
Total	**-**	**Low**	**-**	**-**

3 Detailed Analysis

This chapter outlines the attacks and found vulnerabilities in detail.

3.1 ProtonMail iOS Mail App

3.1.1 General Information

This section describes vulnerabilities found in the ProtonMail iOS Mail App.

ProtonMail iOS Mail App is designed for iOS based mobile devices and provides ProtonMail email capabilities to mobile users. During the timeframe of the review, the ProtonMail iOS Mail App version 1.11.12 was tested using a Jailbroken iPhone with iOS 12.4. The tested iOS app is written in Swift and Objective-C.

3.1.2 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 review 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

3.1.2.1 Recheck results

The issue remained unchanged. However, the risk is accepted by Proton Technologies AG

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.

3.1.3 Hardcoded Credentials - FIXED

The application source code files contain hardcoded credentials. This could potentially allow an attacker to bypass the authentication provider that has been configured by the software administrator. Usually, the existence of hardcoded credentials is not known to administrators. Once this security issue has been detected, it's not always trivial to mitigate it as the affected software may come in a binary form, so temporary solutions such as entirely disabling the affected software are involved.

CVSS-v3 Base Score: 2.9 (Low)

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

3.1.3.1 Recheck results

During the initial review the hardcoded credentials were found in the source file `Analytics.swift`. The following code fragment (same as in the initial code review) was used to verify if the issue is fixed:

Path: protonmail_ios-audit-develp/ProtonMail/ProtonMail/Utilities/Analytics.swift (fix verification version)

Lines: 36-42

```
[…]
    private var sentryEndpoint: String {
        #if Enterprise
            return ObfuscatedConstants.Sentry.enterprise
        #else
            return ObfuscatedConstants.Sentry.live
        #endif
    }
[…]
```

Furthermore, these credentials were not found in other parts of the source tree.

3.1.4 Debug Messages Enabled - PARTIALLY FIXED & ACCEPTED

The iOS 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 review timeframe it was identified that debug messages contain sensitive data.

CVSS-v3 Base Score: 2.9 (Low)

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

3.1.4.1 Recheck results

All logs within the mobile app have been disabled. However, the following output by system level logs show an excerpt of the debug messages containing data from the application keychain and email communication.

```
Feb  7 14:43:31 Sectest-iPhone apsd(PersistentConnection)[90] <Notice>:
2019-02-07 14:43:31 +0200 apsd[90]: <APSPushHistory: 0x10031c3e0>
receivedPushWithTopic com.protonmail.protonmail token <a0944226 b6f84a81
65630d14 fc99ec3c 0f7051cd b2806842 ca44a407 1670e194> payload <7b226170
73223a7b 22616c65 7274223a 224e6577 206d6573 73616765 20726563 65697665
64222c22 62616467 65223a31 382c226d 75746162 6c652d63 6f6e7465 6e74223a
317d2c22 656e6372 79707465 644d6573 73616765 223a222d 2d2d2d2d 42454749
4e205047 50204d45 53534147 452d2d2d 2d2d5c6e 56657273 696f6e3a 2050726f
746f6e4d 61696c5c 6e436f6d 6d656e74 3a206874 7470733a 5c2f5c2f 70726f74
6f6e6d61 696c2e63 6f6d5c6e 5c6e7763 424d4134 592b5c2f 38705173 57594c41
51674167 695a584f 74696f71 6b4a6671 4e383959 46527265 4751446b 496a3562
70415346 72594967 4e47635c 6e5a4631 62354a61 6a736c6f 30477977 39725853
6d464e41 58542b7a 64544a69 59657942 6c525277 7a445155 4f434d6d 6b76686c
65654f6a 516b6e6e 5c2f5c6e 47594a5c 2f624331 736d6864 724d4449 6e63332b
6e4a5a4d 75626a46 67704c67 73586938 4b316339 4f716578 696c5163 37497031
49687478 35<\M-b\M^@\M-&>

Feb  7 14:43:31 Sectest-iPhone apsd(PersistentConnection)[90] <Notice>:
2019-02-07 14:43:31 +0200 apsd[90]: <APSCourier: 0x10030d840>: Received
message for enabled topic 'com.protonmail.protonmail' with payload '{

    encryptedMessage = "-----BEGIN PGP MESSAGE-----\134nVersion:
ProtonMail\134nComment:
https://protonmail.com\134n\134nwcBMA4Y+/8pQsWYLAQgAgiZXOtioqkJfqN89YFRreGQD
kIj5bpASFrYIgNGc\134nZF1b5Jajslo0Gyw9rXSmFNAXT+zdTJiYeyBlRRwzDQUOCMmkvhleeOj
Qknn/\134nGYJ/bC1smhdrMDInc3+nJZMubjFgpLgsXi8K1c9OqexilQc7Ip1Ihtx5SWLb\134nF
tFH/NRBx2jxUv71GCPki5vw92OIg7Hh+g0YavJMNQizTjihaghEGkk2Pxak\134niIxLeKNl13ii
2mCxLMn95IHUrUknrYOg5J70xB75MBptEupD7Ds7HE1d/rdd\134nxOYB+1yBlN2YiaCaZsK6mpP
OkBnw8PXqZxsRs0raFiA6uFwexRXIEPRQfZ7b\134natLBzQEJANjYzb/9dd5lZ2462j3HAEmPFu
+CdRaShgowXijEQLLrLvhwQJ9c\134n/WKI7lMMXERc0NpO8FIkJawsuTCKiWbGxbuBBBGgQ80f1
T/Hjf4yrjY1hCaX\134nAahyI7zVIst5+dkDD0fXWSMJA4PTE9dN7pA7xewF0bIryAE1ERMWTmX+
dmy6\134nkiMdiO1fsAfWrKcYHJz9kpRWfHAZSfSMlLp/W0wI<\M-b\M^@\M-&>
```

3.1.5 Data Protection Class Is Not Specified - FIXED

The iOS application does not explicitly set a data protection class for its files. If a protection class is not set, the default one (`NSFileProtectionCompleteUntilFirstUserAuthentication`) is used. The default data protection class ensures sufficient level of security in regards of file encryption. However, it's recommended for mobile apps dealing with sensitive information to take the full advantage of security mechanisms provided by iOS.

CVSS-v3 Base Score: 2.9 (Low)

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

3.1.5.1 Recheck results

During the recheck it was identified that the mobile iOS app sets the data protection class for its files to the slightly more restricted value `NSDataWritingFileProtectionCompleteUntilFirstUserAuthentication` as the application needs to access data in the background while the device is locked for usability reasons:

Path: protonmail_ios-audit-develp/ProtonMail/Pods/TrustKit/TrustKit/Reporting/TSKBackgroundReporter.m (fix verification version)

Lines: 183-189

```
[…]
    // Write the JSON report data to the temporary file
    NSError *error;
    NSUInteger writeOptions = NSDataWritingAtomic;
#if TARGET_OS_IPHONE
    // Ensure the report is accessible when locked on iOS, in case the App has the NSFileProtectionComplete entitlement
    writeOptions = writeOptions | NSDataWritingFileProtectionCompleteUntilFirstUserAuthentication;
#endif
[…]
```

3.1.6 Secure Backgrounding Is Not Fully Implemented - FIXED

The way iOS handles application snapshots could result in a privacy leak if secure backgrounding is not implemented in the mobile application. When a mobile device is sent to sleep by pressing a Power button a snapshot of the application is taken and stored in the Snapshots directory. Any sensitive information that was seen on the screen before entering the background is written to the file system in clear text.

CVSS-v3 Base Score: 2.9 (Low)

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

3.1.6.1 Recheck results

While using the mobile application a Power/Screen lock button was pressed to put the mobile device to sleep. A snapshot of the application was created in the file system.

```
/private/var/mobile/Containers/Data/Application/<UUID>/Library/Caches/Snapsh
ots/com.protonmail.protonmail
```

During the recheck no privacy leaks were detected as demonstrated in Figure 1:

```
# cd /private/var/mobile/Containers/Data/Application/A0CC77C3-F4A3-4E2A-
9981-30B4175ABB8B/Library/Caches/Snapshots/com.protonmail.protonmail

# ls -la

total 1032

drwxr-xr-x 6 mobile mobile    192 Sep 10 03:13 ./

drwxr-xr-x 3 mobile mobile     96 Sep  6 17:28 ../

-rw-r--r-- 1 mobile mobile 339015 Sep  6 17:28 57DE8FF8-CB81-4770-B837-
0BB9730ADD29\@2x.ktx

-rw-r--r-- 1 mobile mobile 339015 Sep 10 03:13 5EABAE8C-F840-4BFA-8CDA-
BC9719B9E235\@2x.ktx

-rw-r--r-- 1 mobile mobile 373887 Sep  6 17:28 E7C5917B-7F60-4773-B40E-
A7339E6451F5\@2x.ktx

drwxr-xr-x 3 mobile mobile     96 Sep 10 03:13 downscaled/

# cd downscaled/

# ls -la

total 240

drwxr-xr-x 3 mobile mobile     96 Sep 10 03:13 ./

drwxr-xr-x 6 mobile mobile    192 Sep 10 03:13 ../

-rw-r--r-- 1 mobile mobile 241981 Sep 10 03:13 CAD2D826-15D1-40FF-84AB-
3B5424C5D08C\@2x.ktx
```

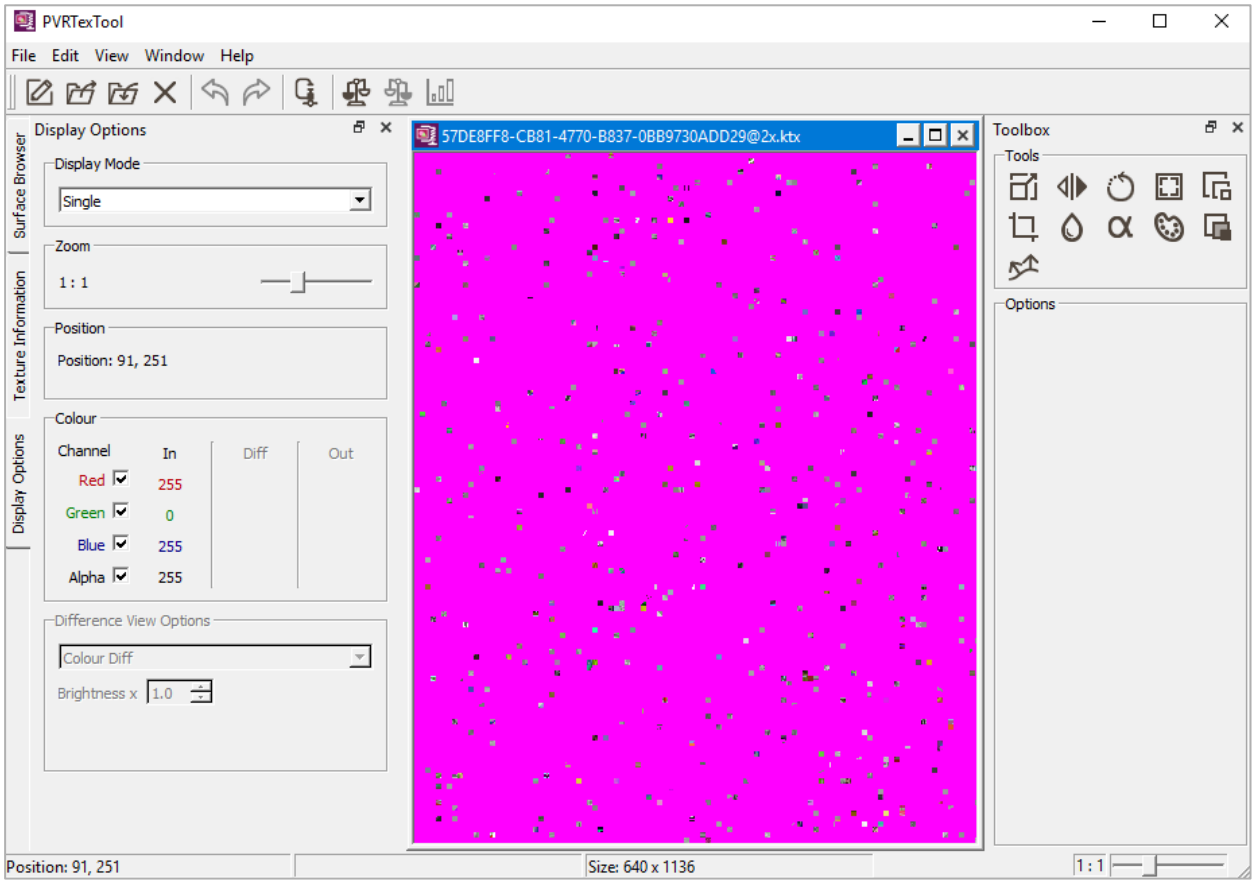

Figure 1. No sensitive information is written to the file system in clear text.

3.1.7 Strongest Keychain Data Protection Class Not in Use - FIXED

The iOS keychain provides a secure way to store sensitive data such as user credentials. Keychain data is protected using Data Protection classes. Depending on the sensitivity of data, different Data Protection classes can be applied. The `kSecAttrAccessibleWhenUnlockedThisDeviceOnly` Data Protection Class is the strongest one. This class ensures that keychain data is only accessible when a device is unlocked. It also prevents data migration to other devices using backups.

CVSS-v3 Base Score: 2.9 (Low)

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

3.1.7.1 Recheck results

The following ProtonMail keychain items are protected using the Data Protection class `kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly` as the application needs to access data in the background while the device is locked for usability reasons:

Path: protonmail_ios-audit-develp/ProtonMail/Keymaker/Keychain.swift (fix verification version)

Lines: 34-43

```
[…]
open class Keychain {
    internal enum Accessibility {
        case afterFirstUnlockThisDeviceOnly

        var cfString: CFString {
            switch self {
                case .afterFirstUnlockThisDeviceOnly: return
kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly
[…]
```

Lines: 65-71

```
[…]
    public init(service: String, accessGroup: String) {
        self.service = service
        self.accessGroup = accessGroup

        self.accessibility = .afterFirstUnlockThisDeviceOnly
        self.authenticationPolicy = .none
    }
[…]
```

3.1.8 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

3.1.8.1 Recheck results

During the recheck it was not possible for an attacker to intercept and manipulate the communication between the mobile app and the backend server as shown in Figure 2:

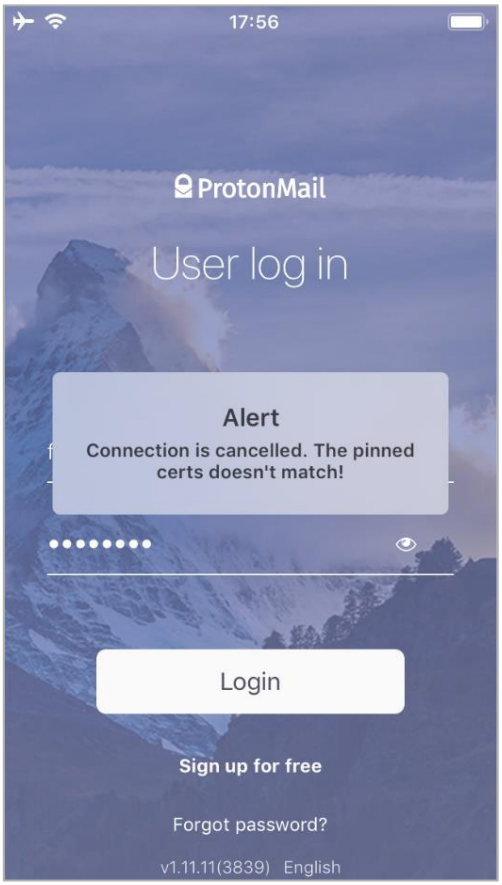

Figure 2. Certificate pinning is in place.

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	2019-10-29	Public report	SEC Consult	SEC Consult