apriorit

Case Study: **Security Testing for iOS Healthcare Application**

Application Description

> *A big healthcare business group created a mobile app for the medical personnel visiting patients at home as well as working at hospitals and other medical institutions. The app is designed to help doctors check in their visits and shifts and timely update information about their patients' state and treatment in the global group data center.*
>
> *Apriorit experts were assigned to perform iOS app penetration testing and vulnerability assessment (supported versions: iOS 9.2+).*

Research Type

> ***The team was requested to perform black box security testing with the source reverse engineering activities. The app was researched on both non-jailbroken and jailbroken devices.***

Important Aspects to Focus on

The app was intended to perform data exchange between the device and data center at one hand and operated with sensitive information and provided access to the stored sensitive information on the other hand. Thus, the number one priority activities were related to the data in motion security testing. Sniffing technique with Fiddler was used. Researchers focused on the ability to get:

1) user credentials and/or session tokens to get access to the protected data center

2) sensitive healthcare data in motion

Another important aspect was data at rest protection, so researchers also focused on getting user credentials and sensitive data portions from the data stored in the device keychains and cache databases.

Meanwhile, other checks for possible attack vectors such as log checks, third-party libraries, various types of injections to the web interaction, source reversing, and other activities obtained normal priority.

Results and Recommendations

Sniffing results showed that although the channel was SSL-protected, the app was not protected from the **man-in-the-middle** attack. Sniffed requests contained access keys, user credentials and details with internal IDs.

To illustrate the risk, Apriorit experts developed a small desktop application, which used sniffed details to access all user-assigned information in the data center. It was discovered that access token expiration policy is configured properly and a token expired in 10 minutes or after logout, but at the same time the

server was sending new tokens to the client in its responses, so that a user did not have to re-enter credentials each 10 minutes. Man-in-the-middle attack allowed the test application to receive new tokens and preserve access to all information.

This risk obviously obtained the highest priority. The recommendation was to use the **SSL pining** technique to make it much more difficult to organize man-in-the-middle attack and avoid transferring user password in plain text.

The Apriorit team also detected several risks when analyzing data at rest protection.

After importing encrypted backup created by iTunes to a mobile data analysis tool, investigators obtained a keychain entry that contained access token. With the existent expiration policy, this security risk was pretty low.

Apriorit experts performed **logical data acquisition** by the iTunes backup protocol from a non-jailbroken device using a professional investigation tool. Besides the encrypted main database, they also obtained the **unencrypted cache database** containing sensitive user information with access keys, token, and credentials. This data would be enough to get access to the user account on the server using an application similar to the one described above.

This risk also got high priority. The recommendation was to exclude the cache database from backup.

The app was successfully installed on **a jailbroken device**. It prompted Apriorit experts to include a normal priority recommendation to the report to add protection from app installation and start on jailbroken devices. **File system acquisition** performed after that once again provided access to the unencrypted cache database.

The app successfully passed other checks.

Second Round Check

After the report was delivered to the client and properly discussed, the client team started to prepare an improved version of the app. This new version was then provided to Apriorit experts to perform the second round check.

First, investigators found that the new app version included protection from being run on a jailbroken device. They still managed to run it on a jailbroken device, but it required some third-party tools to be installed.

The client team introduced the SSL pining technique to protect the app from the man-in-the-middle attack and it worked perfectly on non-jailbroken devices. As it was mentioned above, security testers managed to run the app on a jailbroken device and thus was able to bypass SSL pining.

At the same time, the developers improved device-server communication and user password was no longer sent in plain text. The complex unique access key was used, generated by user credentials and request time. Besides significantly minimizing man-in-the-middle attack output, the latter also protected the server from being accessed by a fake client application.

Neither logical acquisition for a non-jailbroken device nor complete file system acquisition for a jailbroken one provided cache database or any other object containing sensitive data.

Final Recommendations and Results

The only additional recommendation was to add symmetric algorithm encryption for the request/response body to additionally protect communication channel from the man-in-the-middle attack on a jailbroken device.

The whole app vulnerability assessment project starting from the initial research and till the second round of discussions took 80 man-hours.