Session Lock Authentication Bypass in Wickr iOS App

Published Date	TBD
Vendor	Wickr Inc.
Affected Versions	2.5.2 for iOS
Risk Rating	Medium
CVE Reference	TBD
Author	Sagi Shahar

Description

Wickr version 2.5.2 (iOS) was found to be vulnerable to an authentication bypass weakness on the 'Session Lock' screen. This may enable attackers to access to the user's sensitive information without providing the user's password.

Cause

User's sensitive information is not encrypted after Wickr locks the user's session. Therefore, the user's password is not required to access this sensitive information.

Impact

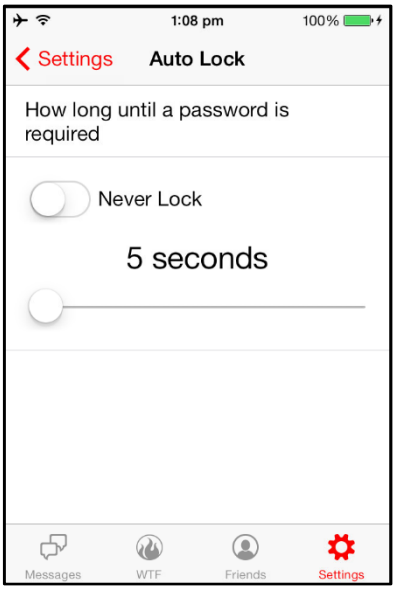
Successful exploitation provides access to the user's sensitive data, contacts and settings.

Interim Workaround

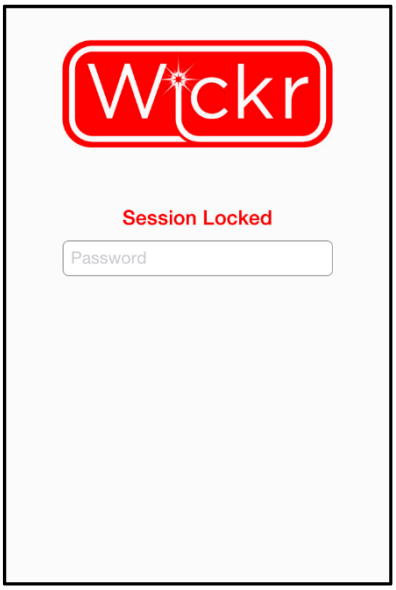
Do not use the lock session functionality and instead log off completely each time the application is not in use.

Technical Details

Wickr has a built-in 'Auto Lock' feature that allows a user to set a time period before they are required to enter their password to the application. By default, the timeout value is 1 hour, however, a user can change that value to 5 seconds, which would appear to be an even more secure option. The screenshot below shows the 'Auto Lock' feature within Wickr's settings:



Once the app is moved into the background and then reopened (after the time set in the 'Auto Lock' functionality has exceeded), the user is required to re-enter their password to access the application. The 'Session Lock' view can be seen in the screenshot below:



'SessionManager' is the class that controls the session lock and implements various methods, for example:

- -(void)sucessfullyResumedSession
- -(BOOL)unlockSessionWithPass:(id)pass

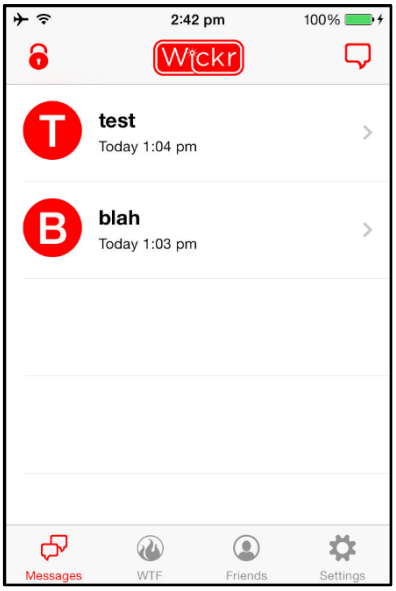
It was observed that the 'sucessfullyResumedSession' method was called after the 'unlockSessionWithPass' method, under the condition that the password was confirmed to be correct. It was also observed that when 'sucessfullyResumedSession' is called, the 'Session Lock' view is removed and the user is granted normal access to the application including the sensitive data it holds.

With this in mind, if a reference to the current 'SessionManager' object is obtained, it is possible to invoke the 'sucessfullyResumedSession' method and therefore bypass the authentication requirement, gaining access to the user's sensitive data.

The figure below demonstrates how the authentication can be bypassed, with the aid of Cycript on a jail broken device:

```
asterisklabs:~ root# cycript -p Wickr
cy# var sm = choose(SessionManager)
[#"<SessionManager: 0x16828e90>"]
cy# [sm[0] sucessfullyResumedSession]
```

The screenshot below shows the access gained from the steps above:



Exploitation of the bug depends on the phone being jail broken and the application running in the background. Even with these dependencies, due to the sensitive nature of the Wickr application, the ability to access the data without supplying a valid password is believed to be risk. In addition, Wickr does not attempt any form of jailbreak detection at all.

A more secure implementation would use the user's password as a component in the decryption key derivation process. This means that if the entered password is invalid, or in our case, not entered at all, then the data remains encrypted. Therefore, any attempts to bypass the 'Lock Screen' will end up unsuccessful since the decryption key could not be derived.

The current execution workflow of the app was observed and shown in a high level in the figure below:

An example of a more secure solution can be implemented as follows: