```
Q Search by keywords or tags
Developer Forums
Intermittent -34018 errors from users in the field
            This thread is the place to post if:
               • you're encountering -34018 errors being returned by the Security framework
               • the problem is intermittent, that is, it shows up very occasionally on user devices in the field but is otherwise hard to
                 reproduce
4.7k
            IMPORTANT Error -34018 has other causes. Before posting here, make sure to read the Error -34018
            errSecMissingEntitlement pinned post.
            Apple has been tracking an issue where Security framework APIs can return -34018 on some user devices in the field. The
            typical observed behaviour is:
               • After running successfully for a while, your app starts receiving -34018 from various Security APIs, most commonly the
                 keychain APIs.
               • Those errors persist until the app is terminated and relaunched (or the device is restarted).
               • The problem shows up intermittently on a small percentage of user devices in the field.

    The problem only affects iOS-based platforms (iOS, watchOS, tvOS).

            Apple has been working hard to investigate this issue and believes we have fixed some of its root causes. However, it is a
            complex problem and it's possible that there could be others. If you're still seeing this problem on the latest iOS release, feel
            free to post your experiences to this thread.
            Note This thread is the continuation of an older thread here on DevForums; I've locked that old thread for reasons explain in
            the last post on that thread.
            That thread was itself a continuation of an even older thread on the old DevForums, which became read-only when the old
            DevForums was decommissioned.
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                   Asked 4 years ago by eskimo 

             Security
                                                                                                           Reply to this question
13 Replies
            We raise an exception (/crash) when any unexpected keychain status code is returned. So our crash logs alert us to problems
            with the keychain. Sadly the exception text is not included so I cannot confirm it is error -34018.
            At the moment, we have a couple of crash logs with Keychain problems, including one on iOS 10.0.1. Does that qualify as the
            latest release or should we only be looking for iOS 10.0.2 and higher?
                                                                                                      Posted 4 years ago by tijmen 🗂
               Sadly the exception text is not included so I cannot confirm it is error -34018.
            Bummer. Can you rev your app so that the error code gets included in the log?
               At the moment, we have a couple of crash logs with Keychain problems, including one on iOS 10.0.1. Does
               that qualify as the latest release or should we only be looking for iOS 10.0.2 and higher?
            Yeah. AFAIK 10.0.2 did not include any fixes specific to this issue. However, it's not necessarily relevant without knowing
            what error you got.
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                  Posted 4 years ago by eskimo 

            Sure, but how do I get the error code in the crashlogs? I cannot seem to set an custom exception code for example.
                                                                                                      Posted 4 years ago by tijmen 🗂
               how do I get the error code in the crashlogs?
            There is no supported way to put custom app state into a crash log. However, I don't see how that matters here. This error
            is reported to you as an
             OSStatus
            ; it's your choice whether you crash when you get the error and, if you do, you can log the error somewhere that you can
            report it via your own logging the next time you launch.
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                  Posted 4 years ago by eskimo 

            Hi,
           I'm developing an application in Swft 2.3 that uses this library to access the keychain (version 3.0.16). We use a keychain for
           a group that includes the iOS app, a keyboard extension and an iMessage app
            We are experiencing some problems on testing devices where a really old access token is persisted.
            So this is the flow:
            The user log in successfully -> The user log out -> An old access token (not the latest one) is in the keychain.
            What I found out (just through an experimental way, please correct me if I'm wrong) is that the Security framework provides a
            kind of in-memory cache that has the priority on the stored keychain and where the operation is done even if the stored
            keychain hasn't been modified. That's why the user can login successfully and can use their own brand new access token.
           This problem, by the way, it's occurring only on the access token field, not for the other values stored in the keychain, which
            they are cleaned as well after the user log out.
           So I started to investigate this problem that seems to appers without any logic. It happens on an iPhone 6 Plus iOS 8.4 and
            on an iPhone 6 iOS 10.0.1 (both connected at least once on Xcode and not rebooted).
            While debugging this error on these devices I found out that trying to set a new value or delete the value for the key in
           keychain results in a -34018 OSStatus, while, in the same moment, setting or deleting any other value works perfectly
           Unfortunately I can't give you any other info since I'm still stucked in this problem and we are, sadly saying, moving to a
           shared user default with encrypted values there.
                                                                                                     Posted 4 years ago by luca34 🗂
            This doesn't sound like an exact match for the intermittent problem that's the subject of this thread, but it is an interesting
            failure nevertheless.
               What I found out (just through an experimental way, please correct me if I'm wrong) is that the Security
               framework provides a kind of in-memory cache that has the priority on the stored keychain and where the
               operation is done even if the stored keychain hasn't been modified.
            To be clear, the SecItem APIs always do an IPC to the security daemon, which always results in a database query against the
            true keychain database. So there's no caching there.
           There is, however, caching at other levels. The caches that most commonly trip folks up are:

    the TLS session cache, described in QA1727

    HTTP 1.1 (and now 2) persistent connections

    Foundation network per-session (that is, non-persistent) credential caching

            One or more of these could be in play here; it's hard to say without knowing more about how your credentials are tied to your
            networking.
               It happens on an iPhone 6 Plus iOS 8.4 and on an iPhone 6 iOS 10.0.1 (both connected at least once on
               Xcode and not rebooted).
            To my mind the iOS 8.4 case is uninteresting; during the iOS 9 cycle we definitely fixed bugs that cause the -34018 error
            when using Xcode.
            OTOH, the iOS 10 case is interesting. If you can distill down a reasonable test project that shows this, we would definitely
            appreciate a bug report for that (if you do create a bug, please post your bug number, just for the record).
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                  Posted 4 years ago by eskimo 

           That's quite strange because when I delete the key or set a new value for that key, it changes its value (or it deletes it value)
           until I reopen the app. At that time, in the first line of didFinishLaunchingWithOptions in the AppDelegate I print the value for
           that key and it's again the old value.
            We are not using anything complex: we store the api token received from the login in the keychain and we use this api token
            for each request. The api token it's a string (see JSON Web Token)
           I'll try to investigate more the problem on iOS 10.
                                                                                                     Posted 4 years ago by luca34 🗂
            My app got a -34018 OSStatus calling SecItemCopyMatching when resuming from facebook login process.
           I was using Xcode 7.3 to run a debug build on IOS 8.3 Iphone 7, 1, not 100% reproducible but I have seen it happened a few
            times and our QA have reported a few occurrences before.
           Key chain access calls working fine after the app was launched, read and write returns successfully.
            Then Facebook login is trigger and the app was pushed to background, after the Facebook login is done and returned to the
            app, the keychain read failed with -34018 and keep failing all the calls until the app is terminated.
            I assume the bug was triggered when
            - Resuming the app
            - System resources may be low(?)
                                                                                               Posted 4 years ago by schang1967 🗂
               I was using Xcode 7.3 to run a debug build on iOS 8.3 ...
            As I mentioned to luca34, the iOS 8 case is uninteresting because we definitely fixed bugs related to this during the iOS 9
           cycle.
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                  Posted 4 years ago by eskimo 

            We're seeing an issue where calls to SecKeyCreateRandomKey for generating a key into the SE hang. Below is how we call it
           (on an iOS 10 device; the kSecAttrApplicationTag attribute points to some NSData object containing a tag derived from a
            string.)
            This happened to us in the lab once. This is accompanied by log messages about access permissions from keychain --
            however note that the call hangs. When we saw this in the lab, we were able to make this call right after application launch, in
            the application delegate's application:didFinishLaunchingWithOptions: call -- hangs again. Seems like once a device is in that
           state, it's persistent. The only way we were able to recover was to reboot the device.
            We now get reports on this from customers in the field.
           Any pointers will be appreciated.
            -----
           CFErrorRef error = NULL;
            SecAccessControlRef access = NULL;
            access = SecAccessControlCreateWithFlags(kCFAllocatorDefault,
                                       kSecAttrAccessibleWhenUnlockedThisDeviceOnly,
                                       kSecAccessControlPrivateKeyUsage,
                                       &error); // Ignore error
            NSDictionary* attributes =
                                             kSecAttrKeyTypeECSECPrimeRandom,
              @{ (id)kSecAttrKeyType:
                (id)kSecAttrKeySizeInBits:
                                             @256,
                (id)kSecAttrTokenID:
                                           (id)kSecAttrTokenIDSecureEnclave,
                (id)kSecAttrAccessControl: (__bridge id)access,
                (id)kSecPrivateKeyAttrs:
                  @{ (id)kSecAttrlsPermanent: @YES,
                    (id)kSecAttrApplicationTag: ....
                SecKeyCreateRandomKey((__bridge CFDictionaryRef)attributes,
                                     &error);
                                                                                                     Posted 3 years ago by eldanb 🗂
            The problem, as you've described it, is unlikely to be related to your code. My recommendation here is that you file a bug,
            including a sysdiagnose log taken shortly after you reproduced the problem.
            Note For more info on sysdiagnose logs, see our Bug Reporting > Profiles and Logs page.
            Please post your bug number, just for the record.
            Share and Enjoy — Quinn "The Eskimo!" Apple Developer Relations, Developer Technical Support, Core OS/Hardware
             let myEmail = "eskimo" + "1" + "@apple.com"
                                                                                                  Posted 3 years ago by eskimo 

            Thank you for this response.
           We have previously filed a report on this -- nr. 35622508 and we were informed that it is a duplicate of 33714117.
           Unfortunately that is the only information we have -- no insight on why this is happening or what's a suitable workaround.
            Are you able to shed more light on this?
                                                                                                     Posted 3 years ago by eldanb 🗂
            Hi,
           it looks like to me that this problem still exists. I am experiencing same issues occasionally, and I am getting out of ideas how
           to resolve. App is using keychan for storing access and refresh token, and keychain returns this kind of error on random, as I
            can see. On some devices works without any issues, and on some fails, from time to time, can't figure out any pattern. Using
            latest xcode, and it doesn't depends on iOS version. If something is wrong with keychain group, entitlements, or code I
            assueme that it will fail more often, since implementation is straightforward. App is presenting welcome screen, and waits for
            tokens to be refreshed, or it is refreshed at runtime if server request fails, and we can remove memory pressure as potentialy
           reason (I read somewhere that can cause this). Error is retuned no meeter if we try to retreive or store value.
           Any advice would be appreciated.
           Edit: I tried to force memoryWarning, simulate memory preassure with allocating memory and data is retreived properly.
           Since I am logging errors into file, I just received log file from client device with message: `Keychain loading error for key
           refreshToken. Keychain error status: -34018 App is using 34.21953582763672MB MB. Device memory status is: Used
           memory: 2350.40625 MB; free memory: 184.625 MB.`
                                                                                                      Posted 1 year ago by IgorCR 🗂
This site contains user submitted content, comments and opinions and is for informational purposes only. Apple disclaims any and all liability for the acts, omissions and conduct
of any third parties in connection with or related to your use of the site. All postings and use of the content on this site are subject to the Apple Developer Forums Participation
Agreement.
Developer
                Apple Developer Forums
                                                                                Distribute
                                                                                                            Support
Discover
                           Design
                                                      Develop
macOS
                           Human Interface Guidelines
                                                      Xcode
                                                                                Developer Program
                                                                                                           Articles
iOS
                                                      Swift
                                                                                App Store
                                                                                                           Developer Forums
                           Resources
```

© Developer

Discover

Design

Develop

Distribute

Account

watchOS Videos Swift Playgrounds Feedback & Bug Reporting App Review

Mac Software

Apps for Business

Safari Extensions

Marketing Resources

Trademark Licensing

System Status

Certificates, Identifiers &

App Store Connect

Contact Us

Account

Profiles

TestFlight

Videos

Downloads

Documentation

Apple Design Awards

Internationalization

Copyright © 2021 Apple Inc. All rights reserved. Terms of Use Privacy Policy License Agreements

Fonts

To view the latest developer news, visit News and Updates.

Accessibility

tvOS

Games

Business

WWDC

Education

Safari and Web