The bug was fixed on iOS 13.2, CVE number is unknown. I originally intended to use the iPhone rjb for TianfuCup (of course, only the vulnerability ^^), but I regret that it was fixed more than ten days before the game, but the cause of the vulnerability is very simple. Interesting. I will write a post later about the vulnerability in Safari. The vulnerability is a recently added syscall: kqueue_workloop_ctl, which then calls the underlying function kqueue_workloop_ctl_internal. There is no MACF check on the vulnerability path, which means it can be used for any sandboxed privilege, including Safari. There are two problems in the kqueue_workloop_ctl_internal function. The first problem is as follows: We can see that if the TRP_RELEASED flag is not set, kqueue_release will be called twice, and a total of two reference counts will be subtracted. If it is set, it means that the kq has been released, and only one call is subtracted. This reference count is added to the kevent_get_kq function. At first glance, there is no problem, you can avoid the occurrence of race, and you can avoid releasing the same kq multiple times, resulting in over release. But the key to the problem is that the flag is set on a stack variable instead of the heap variable. In other words, no matter how to set a flag will not be true.So the cause of this problem is simple, but it's hard to find, because the code doesn't seem to have any problems, if you don't see the source of the |trp| variable above, there are two ways to trigger the poc. The first is to use race, which causes overrelease. The second is to hang kq on another object. First, add the reference count to one, and then release it multiple times through this vulnerability. Reduce the object reference count to 0 and release it. UaF is generated by pointers to other objects. Here my poc is simpler, it is triggered by race. If you want to write exploit, the second trigger method is more reliable: The patch for the vulnerability is very simple, that is, the flag must be set to the heap variable, otherwise it will not produce any effect: As for the second question, it exists in kevent_get_kq and is left to the reader to discover.