



On 9 July, Apple released iOS 11.4.1. In addition to minor fixes, the firmware acquired a new function - USB Restricted Mode. It should protect iPhone from hacking through programs like GrayKey, time-limiting the use of Lightning. Unfortunately, a way to circumvent this limitation on iPhone and iPad has already been found.

How USB Restricted Mode Works

The USB Restricted Mode feature has already been introduced in beta versions of iOS, starting with 11.3. In essence, it forces users to enter a password when connecting the device to a PC, Mac or USB device if the iPhone was in a locked state for more than an hour before.

Thus, if you don't unlock your phone for an hour, the protective mechanism will turn on. In theory, this is intended to prevent security services and law enforcement from hacking Apple devices, resorting to the help of companies such as Cellebrite and Grayshift. So, once USB Restricted Mode is activated, any communication via the built-in Lightning port is completely disabled. The only thing that remains available is charging; from the perspective of the computer to which the iPhone is connected with USB Restricted Mode enabled, the device will appear no different from, for example, an external battery.

Although the official release of the new security feature was not expected until the release of iOS 12, as already mentioned above, USB Restricted Mode came out of beta and was presented in iOS 11.4.1.

How the Breach Was Discovered

Previously, ElcomSoft experts had already explored the beta version and acknowledged that it is reliable enough: the phone really did refuse to "communicate" with the computer, even if it rebooted. If you move iPhone to Recovery or DFU mode, it becomes available via the computer, but it was impossible to search for passwords in these modes. The researchers even tried to "restore" the device,

filling it with fresh firmware - but this failed to disable USB Restricted Mode.

All this is still relevant for iOS 11.4.1; however, it's still possible to fool USB Restricted Mode. As it turns out, the countdown timer will be reset before USB Restricted Mode is activated if you connect any untrusted USB-accessory to the iPhone.

According to the entries in the [ElcomSoft blog](#), the company's specialists have already learned to bypass the new function only a few days after the firmware's release.

Cool Technology, But Poorly Implemented

ElcomSoft representatives suggested that Apple made a mistake in the function's implementation which made it possible to fool the device.

To reset the timer, you can use almost any USB-accessory, including the official [Lightning to USB 3 Camera Adapter](#). However, it has already been discovered that the [Apple Lightning to 3.5mm jack adapter](#) is not suitable for these purposes. Researchers continue to test other accessories, including non-original ones, ordered on AliExpress.

IB specialists are now very interested in why USB Restricted Mode is so easily deceived, and whether Apple is going to correct this defect on iOS 11.4.2 and iOS 12. It's also strange that this "bug" successfully survived five beta versions of iOS 11.4.1.

The ElcomSoft experts believe that the problem may lie in the Lightning protocol itself. The fact is that when an iPhone connects to a computer, the devices exchange cryptographic keys before they start trusting each other. When you connect to most of the existing Lightning accessories, however, none of this happens, because many of them are simply unable to make such a key exchange. As a result, while USB Restricted Mode is inactive, iPhone only checks for accessories with MFi certificates and this completes the checks.

The only possible solution to the problem proposed by ElcomSoft analysts is to "teach" the iPhone to remember the accessories it has previously connected with and to trust the timer reset only to them.

Read more about mobile security in our previous posts:

[How to Ensure Secure Data Transmission](#)

[Mobile App Security: Reverse Engineering](#)

[Magora's solution for secure file transfer](#)