Multiplexed Wired Attack Surfaces

Michael Ossmann <mike@ossmann.com>
Kyle Osborn <kos@kos.io>

1. Introduction

For as long as electrical connectors have existed, engineers have found alternative uses for them. Power may be supplied through a connector intended only for data. Serial interfaces may operate over connectors designed for voice communication. Today there are many cases where a single connector is used for multiple functions on the same device. We investigate such cases, particularly on smart phones and tablet computers, calling attention to attack surfaces not previously studied.

Additionally we examine particular devices featuring multiplexed wired interfaces where a typical user would not be aware of the available multiplexed functions, and we demonstrate an attack against such a function that permits unauthorized access to the device and to user data.

2. Historical Examples

Power over Ethernet (PoE) is a method for delivering electrical power over a twisted pair Ethernet cable originally intended to carry only data. Various schemes have been implemented, most of which apply Direct Current (DC) to the common-mode voltage of one differential pair vs. another. (The exceptions are older schemes that carry power only on wires unused by 10BASE-T and 100BASE-TX.) PoE coexists with Ethernet data communications on the same cable with the data carried differentially over one or more pairs.

Digital Subscriber Line (DSL) provides data communication over telephone lines intended for voice communication. DSL employs Frequency Division Multiplexing (FDM), using higher frequencies for data while lower frequencies may carry voice communication simultaneously.

The Apple iPod Shuffle is one of several portable MP3 players that implement USB over a four-conductor (TRRS) headphone jack that is also used for audio. The product comes with a special USB cable [1] with USB A male on one end and TRRS male on the other. The iPod Shuffle is an early example of a portable device that is able to connect to multiple communication media with the same connector and performs automatic detection to determine which medium is connected.

In all three of these cases, a naive user of the technology may be unaware of the attack surface presented by the multiplexed wired connection. A user of a PoE device might recognize the Ethernet attack surface but might not be aware that the Ethernet connection could be attacked to deny, glitch, or measure power. A user of a telephone may not recognize that the telephone connection carries an entire home's Internet connection. A user of an MP3 player might not be aware that a headphone jack could be used for USB data instead of audio.

3. Multiplexer Integrated Circuits

Many Integrated Circuits (ICs) that are able to switch between various functions of an electrical connector are available on the electronic components market. Several are capable of "accessory detection" and automatically switch to the appropriate function for the type of device or medium connected. One such device known to be used in several smart phones is the Fairchild Semiconductor FSA9280A [2].

The FSA9280A or a similar multiplexer IC for USB is placed by the designer of the device between the USB connector and the USB controller.

In normal operation it provides a single USB connection between the controller and connector. If the multiplexer IC detects a particular electrical resistance between certain pins on the USB connector, however, it determines that a corresponding accessory is connected. In this case it breaks the USB connection and switches to an alternative function.

We find that this type of multiplexer IC is common in smart phones and tablets. While a typical user of such a device would likely presume that the connector may be used only for USB data or power, we enable additional multiplexed functions by triggering accessory detection.

To activate accessory detection, we connect a particular electrical resistance between the ID and GND pins of the target device's USB connector. The ID pin, available only on USB Micro and Mini connectors, is used for USB On-The-Go (OTG), enabling a USB device to operate in either host mode or device mode. Originally intended as a binary input for OTG, manufacturers of USB devices have since implemented checks for multiple levels of resistance between the ID pin and GND. This extends the range of possible values from two to three or more. The first known application of this type of detection is described in the USB Car Kit specification [3].

Modern multiplexer ICs are capable of detecting many different resistances. The FSA9280A data sheet includes a table of 32 distinct values. Of particular security interest are various "factory mode" and UART functions, most of which appear to be intended for use with specialized test jigs used for test, development, or device programming. One such test jig is the Samsung Anyway S102 [4].

While the phone modding community is aware of many of these special functions and test jigs [5], we observe that the information security community is not and that few, if any, of the functions exposed by multiplexer ICs on particular devices have been studied as attack surfaces.

4. Getting a Shell over a USB Connector Without using USB

The Galaxy Nexus (GT-I9250M) features USB accessory detection and implements a TTL UART over the USB connector when a resistance of 150k ohms is applied between the ID and GND pins. A TTL UART is a serial interface similar to RS-232 but operating at TTL logic levels. The Galaxy Nexus UART operates at 0 to 5 V whereas many phones and tablets likely operate at lower voltages. We use a male USB Micro connector [6] connected to our own TTL UART [7] and 150k ohm resistor to communicate with the Galaxy Nexus over this interface.

On the first Galaxy Nexus tested, we found that this gave us access to the FIQ debugger [8] compiled in read-only mode. The FIQ debugger's "console" command dropped us into a shell as the "shell" user (unprivileged). We found that we could use a command from this shell to enable adb [9], overriding the device owner's preference. By disconnecting our UART and connecting a USB host running adb, we gained full access to the device at the privilege level of the device's owner.

The Galaxy Nexus was running CyanogenMod [10]. Subsequent tests of devices of the same model running different operating systems yielded different results. We were unable to gain access to the FIQ debugger on a unit with an operating system provided by Verizon. On a unit with an operating system provided by Sprint, we were able to access the FIQ debugger, but the console command was missing or disabled. We suspect that such variations may appear even between software releases from the same provider.

On a Samsung Galaxy S III running CyanogenMod, we used the same method to access the FIQ debugger and found that the console command gave us a root shell. The discrepancies between results on each target device indicate that the attack surface varies considerably by installed software in addition to the variations presented by different hardware platforms. (There are 13 distinct models called "Samsung Galaxy S III" [11].)

5. Multiplexed Audio Connectors

Many portable devices include headphone jacks that provide special functions in addition to audio. For example, Calypso phones popular with GSM hackers implement a TTL UART over the headphone jack that can be used to access special functions [12].

Using a custom UART cable similar to our implementation for USB connectors, it is possible to access a TTL UART on the headphone jack of the Nexus 4 [13]. Depending on the software installed on the target device, it might be possible to use such a cable to access more than debug output.

6. Future Work

To explore the great variation of multiplexed wired interfaces presented by different target hardware and software, we are developing a custom hardware platform designed specifically for automated probing of these attack surfaces. We hope that this platform can be used by the information security community to test a wide variety of target devices. We observe that multiplexed wired interfaces are present on more than just phones and tablets [14] and hope that automatic probing will enable the extension of this research.

A. References

[1] http://store.apple.com/us/product/MC003ZM/A/apple-ipod-shuffle-usb-cable
[2] http://www.fairchildsemi.com/pf/FS/FSA9280A.html
[3] ANSI/CEA-936-A USB Car Kit Specification (defunct)
[4] http://www.samsungparts.com/PartsList.aspx?Catalog=Parts_and_Accessories&PartSearch=gh99-36900b
[5] http://forum.xda-developers.com/showthread.php?t=1629359
[6] https://www.sparkfun.com/products/10031
[7] https://www.sparkfun.com/products/718
[8] https://android.googlesource.com/kernel/common.git/+/a82e9f5a7ee65687bda08d70256983fdade2d0d2/arch/arm/common/fiq_debugger.c
[9] http://developer.android.com/tools/help/adb.html
[10] http://www.cyanogenmod.org/
[11] http://en.wikipedia.org/wiki/Samsung_Galaxy_S_III
[12] http://bb.osmocom.org/trac/wiki/Hardware/SerialCable
[13] http://blog.accuvantlabs.com/blog/jdryan/building-nexus-4-uart-debug-cable
[14] https://antibore.wordpress.com/2013/04/30/diy-remote-shutter-release-for-samsung-nx20-nx210-and-nx1000-cameras/