Do-It-Yourself Advanced Persistent Threat?

At SilverSky, we love presents…we really do, but we weren't prepared for the one we received in October 2014. A top SilverSky executive received an unsolicited package from China containing a DIY kit for a "White Dual USB 5V 1A 1.5A Power Bank 18650 Battery Charger Box for Phone." True story.

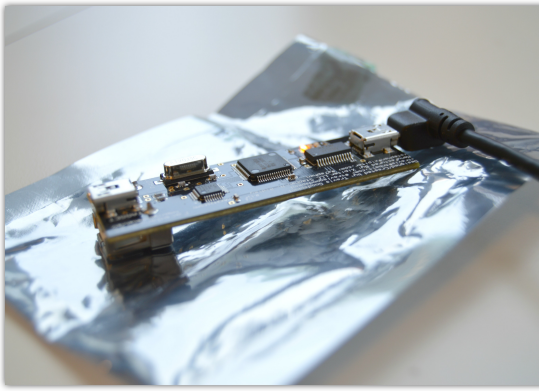
The package was sent to our New York office but listed the phone number of our SOC. There was no explanation besides an eBay card written in broken English, which thanked the purchaser and requested positive feedback on the transaction. The package contained the disassembled pieces of a cellphone power bank (the "Power Bank").

The unsolicited package came shortly after the BadUSB <hyperlink https://srlabs.de/blog/wp-content/uploads/2014/07/SRLabs-BadUSB-BlackHat-v1.pdf> presentation at Black Hat 2014, which renewed security concerns over malicious firmware. Additionally, there have been multiple reports of "production-line malware"<hyperlink http://blogs.microsoft.com/blog/2012/09/13/microsoft-disrupts-the-emerging-nitol-botnet-being-spread-through-an-unsecure-supply-chain/> , literally hardcoded into technology and support software coming out of China. However, if the recent press regarding e-cigarettes is any indication, most security firms are not prepared to investigate at the protocol level.

Understand that when you connect a device via USB, there is a silent handshake between the host (in this case, the Power Bank) and the peripherals (the cellphone), which identifies the device, power exchange, and data flow. All of this happens at the protocol level before you have access to the device.

Essentially, there is a trust relationship that both devices are being honest; however, exploits involving malicious firmware typically take advantage of this trust relationship, gaining access to reserved areas of the system. In our specific case, the Power Bank (acting as the host) will gain access to all areas of the cellphone that a computer can access, including the file system. It's just like connecting your cellphone to a computer. The computer can see application files, photographs, and user data, and with this level of access, there's the potential for the Power Bank to drop a new, malicious file onto the cellphone.

The BAE Applied Intelligence Research Lab maintains close industry ties, keeping up with cutting-edge attack techniques, tools, and methodologies. To investigate the Power Bank, we used Travis Goodspeed's Facedancer21 <hyperlink http://goodfet.sourceforge.net/hardware/facedancer21/> (manufactured by Stephen Ridley at Xipiter), which "allows USB devices to be written in host-side Python, so that one workstation can fuzz-test the USB device drivers of another host." The Facedancer21 is available for purchase here <hyperlink http://int3.cc/products/facedancer21>.



We assembled the Power Bank (without batteries) and tested the unit's power connection. We utilized a USB hardware solution to block the data connection but allow power to flow through called the USB Condom <hyperlink http://syncstop.com/>. The unit powered up and provided a lovely message from China – "FU", which we learned means *full*.

To prep the Facedancer21, you'll need to connect the side labeled "Host" to a Linux machine (recommended) and install the dependencies/packages and flash the firmware. Detailed instructions for flashing the firmware can be found here <hyperlink http://int3.cc/blogs/news/8217777-flashing-the-facedancer21>. Also, make sure your USB cables are *data cables* and not charge-only.

To test our suspicious Power Bank, we configured the Facedancer21 board to emulate a USB mass storage device, the same storage type on a cellphone, by creating a suitable image (which will be emulated by the board) using the commands found in /svn/goodfet/client/make_img.sh.

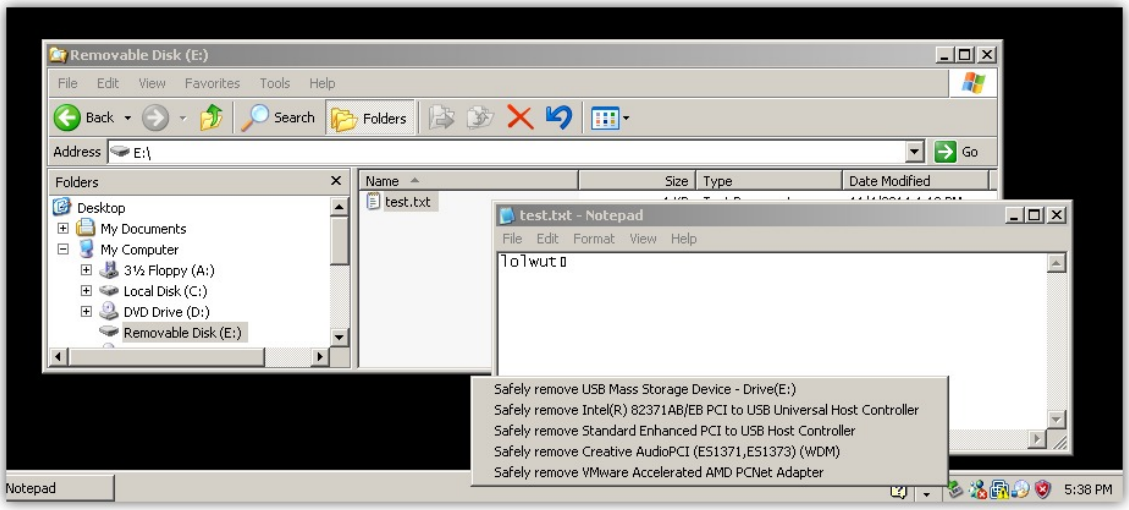
Here, we configure the Facedancer21 to emulate the mass storage device image we created.

```
board=facedancer21 goodfet.maxusbmass /root/usb_images/disk.img
```

We connected the side labeled "Target" to another computer and confirmed that the computer recognized the device as a USB

mass storage device/cellphone (keep in mind that we did not connect an actual USB device or cellphone, we connected the Facedancer21, which we programmed to act like *cellphone data storage*).

We even created a sample file on our “USB Mass Storage Device”.



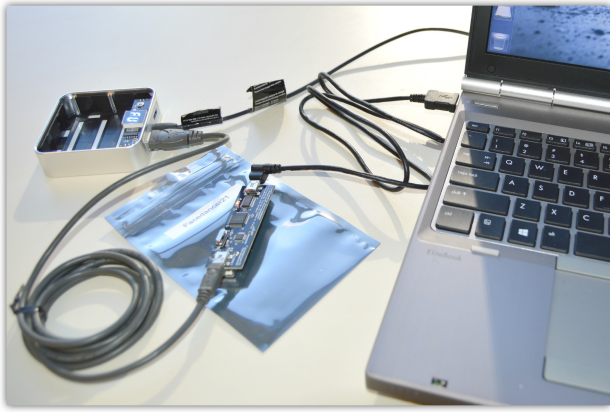

```
root@PC: ~/svn/goodfet/client
Fetching 8 blocks starting at  LBA 2449.
Mode Sense (6) requesting 12 byte Page Code 3f
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0100, wIn
dex=0x0000, wLength=0x0008
Handling a setup packet of bmRequestType=0x00, bRequest=0x05, wValue=0x0006, wIn
dex=0x0000, wLength=0x0000
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0100, wIn
dex=0x0000, wLength=0x0012
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0302, wIn
dex=0x0409, wLength=0x0002
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0302, wIn
dex=0x0409, wLength=0x001c
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0301, wIn
dex=0x0409, wLength=0x0002
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0301, wIn
dex=0x0409, wLength=0x0010
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0303, wIn
dex=0x0409, wLength=0x0002
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0303, wIn
dex=0x0409, wLength=0x0014
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0200, wIn
dex=0x0000, wLength=0x0004
Handling a setup packet of bmRequestType=0x80, bRequest=0x06, wValue=0x0200, wIn
dex=0x0000, wLength=0x0020
Handling a setup packet of bmRequestType=0x00, bRequest=0x09, wValue=0x0001, wIn
dex=0x0000, wLength=0x0000
Handling a setup packet of bmRequestType=0x00, bRequest=0x03, wValue=0x0001, wIn
dex=0x0000, wLength=0x0000
Accepting unknown standard setup request type 03
Handling a setup packet of bmRequestType=0xa1, bRequest=0xfe, wValue=0x0000, wIn
dex=0x0000, wLength=0x0001
Reporting 0 as the maximum LUN.
Mode Sense (6) requesting 8 byte Page Code 3f
Fetching 1 blocks starting at  LBA 0.
Fetching 1 blocks starting at  LBA 0.
Fetching 1 blocks starting at  LBA 0.
Fetching 1 blocks starting at  LBA 0.
```

Additionally, this test provided sample traffic that you might expect from a host device establishing a data connection.

This traffic has been abstracted; however, if you are interested in seeing the detailed protocol handshake, we recommend this write-up on USB Enumeration Code <hyperlink

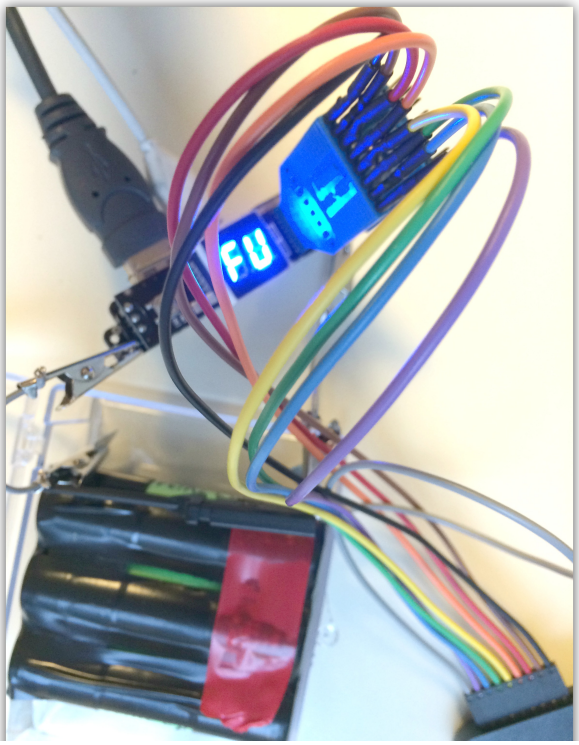
http://pdfserv.maximintegrated.com/en/an/AN3690.pdf>.

Next, we connected the “Target” side of Facedancer21 to the Power Bank. After waiting several minutes, we observed no enumeration whatsoever on either USB port.



```
root@PC:~/svn/goodfet/client# board=facedancer21 goodfet.maxusbmass /root/usb_images/disk.img
/root/svn/goodfet/client/GoodFETMAXUSB.py:15: UserWarning: This library will soon be deprecated in favor of the USB*.py
libraries.
  """This library will soon be deprecated in favor of the USB*.py libraries."""
/usr/local/bin/goodfet.maxusbmass:18: UserWarning: The libraries upon which this program depends will soon be deprecated
 in
favor of the USB*.py libraries.  See facedancer-umass.py (forthcoming) for an
example of this program written using the new libraries.
  example of this program written using the new libraries."""
Opened an image with 204799 blocks.
Connected to MAX342x Rev. 13
Starting a Mass Storage device.
```

According to the USB 2.0 protocol, battery chargers are supposed to enumerate before feeding power; however, some hardware manufacturers will choose not to implement and instead will limit the power draw to prevent an accidental power overload. See Basics of USB Battery Charging <hyperlink http://www.maximintegrated.com/en/app-notes/index.mvp/id/4803> for more information.



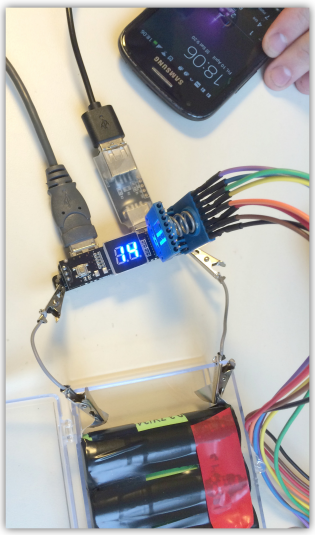
So, is the device JUST a phone charger? Keep in mind that we have no idea where this device was made, why it was made or how it was made. At this point, not only is the software suspect but also the hardware. We decided to do one more round of tests on an unlabeled chip on the circuit board.

For this round of testing, we decided to purchase three 18650 3.7V Li-ion Rechargeable Batteries, so we could test

the Power Bank as it was meant to be used – as a battery-powered charger. We used an 8-pin Saleae Logic Analyzer to monitor the output from the unlabeled chip. We charged the batteries to FULL, continuing to monitor both the unmarked chip and USB port. We noticed that the logic analyzer recorded activity during charging as the LED indicator went from 96, 97, 98, 99, FU. We disconnected the Power Bank from wall power and let it run on batteries. The LED indicator went dark and so did the feedback from the logic analyzer. This led us to believe that the unlabeled chip is the popular 16-pin SOIC LED Driver. Our research revealed that this chip is very common and often ships unlabeled.

We investigated the other chips on the circuit board, identifying a G2116 (power regulator for battery), TP4056 (charger for Li-ion batteries), DW01 (battery protection), and 8205A/B (monitor rechargeable battery/protection).

Finally, we connected a cellphone, using a USB Condom for protection, and verified that the Power Bank does indeed charge a cellphone (big thank you to our fearless lab leader, Brandon Edwards, for donating his personal cellphone for research). The charge was intermittent using battery power, which we attribute to our fabulous setup. Once we connected wall power, the cellphone charged normally.



At no point during our testing did the Power Bank attempt enumeration.

While initial searches for our mysterious Chinese sender provided little information, a few weeks later we linked the sender's contact info to the website sktflyer[dot]com and eventually to an eBay store under the same name.

Not only did we find our specific charger, but also we found additional photos, specifications, and assembly instructions. Also, we learned that skt_flyer is a "Top Seller", though we still don't quite believe the numbers or reviews.

One should note that the stock photo of the circuit board differs subtly from our Power Bank. The "IC" or LED Driver has a different orientation and pinout on our board.

Open-source research on the domain and IP address revealed thousands of additional domains registered by our mysterious sender.

We have our theories as to possible motives, but one thing remains clear, do-it-yourself APT (while possible) seems like way too much work. Although, here at the BAE Applied Intelligence Research Lab, we can't help but plug in random, sketchy tech from China to see what it does.

Feel free to reach out to us if you've done similar research or can recommend other tools/techniques – jody.forness@baesystems.com.