# Spy Mouse

written by Mert SARICA | 1 December 2017

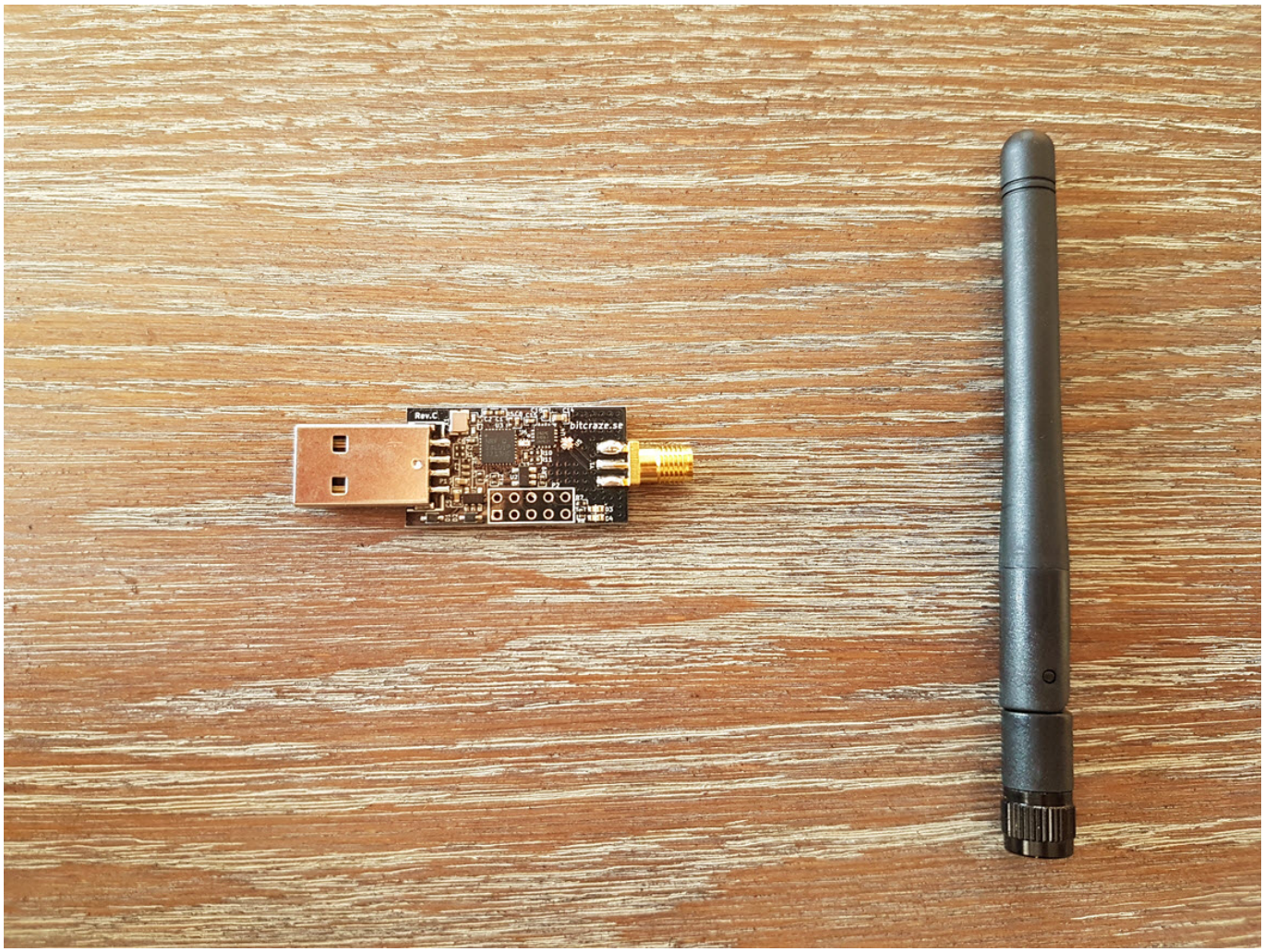
In 2015, I came across a product set that was on sale in an electronics store. If you bought Kaspersky Internet Security software, you would receive a Microsoft Sculpt Mobile model wireless mouse as a gift. I bought it without thinking, because I needed a new mouse, and I've been using it lovingly for years, but it never occurred to me that this wireless mouse, due to the vulnerability it had, could turn into a spy that could work behind my back. :)

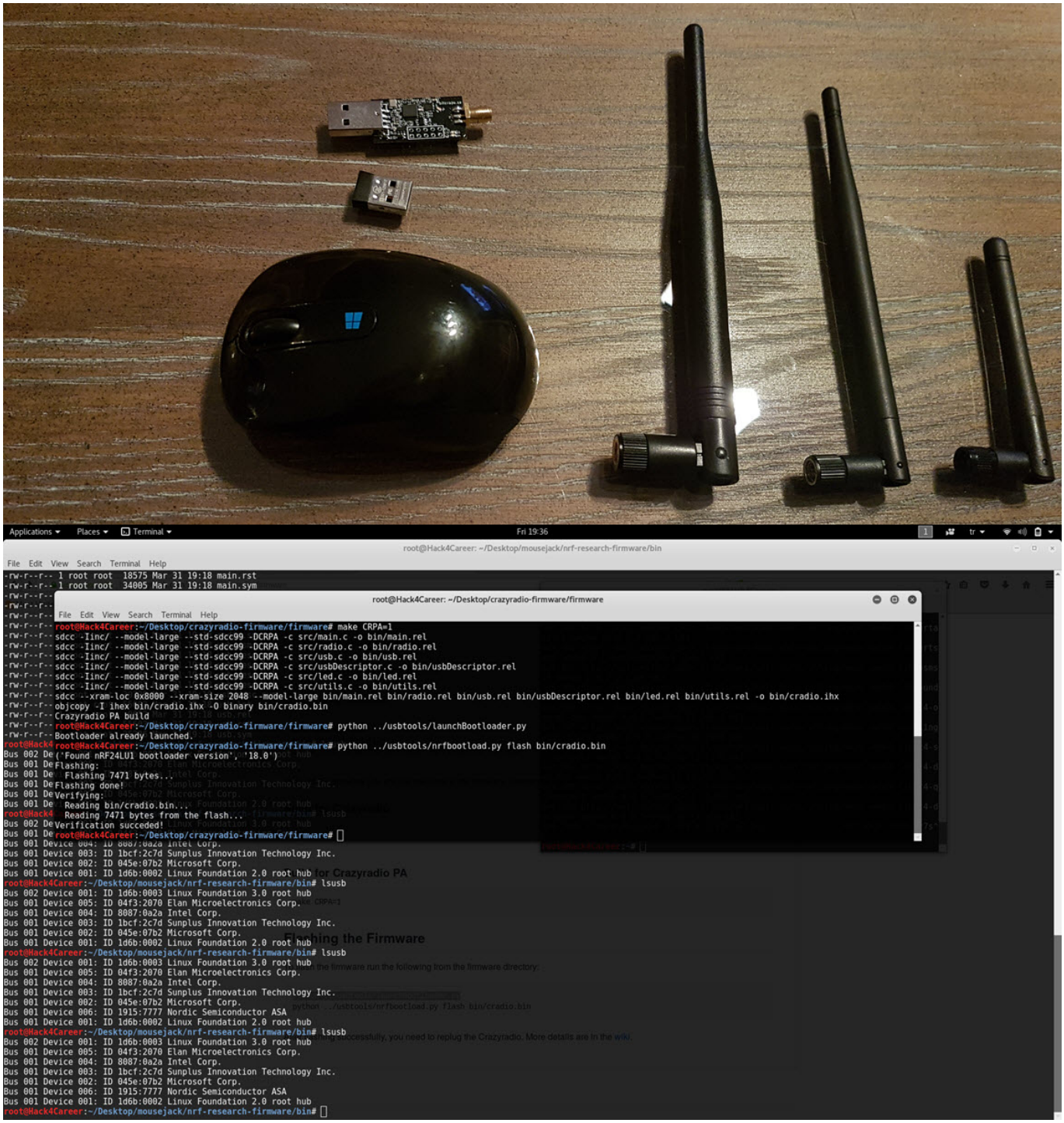
If we take a brief look at the research on RF communication of wireless keyboards and mice that are not Bluetooth, in 2007, Max Moser discovered that wireless keyboards (Microsoft and Logitech) that communicate on the 27 MHz band can be easily monitored remotely, and this caused a stir in the security world. In 2009, Max Moser and Thorsten Schroeder announced the KeyKeriki device, which they developed to listen to wireless keyboards. In 2010, they announced the KeyKeriki v2.0, which can also listen to keyboards communicating on the 2.4 GHz band and equipped with the Nordic Semiconductor NRF24XXX chip. In 2011, Travis Goodspeed showed that the ~5TL value nRF24L01+ chip can be used in promiscuous mode to easily monitor (sniff) packets sent by NRF24XXX chips on the 2.4 GHz band. In 2015, Samy Kamkar showed the world how Microsoft keyboard keys could be instantly and practically stolen using the Arduino-based KeySweeper device.

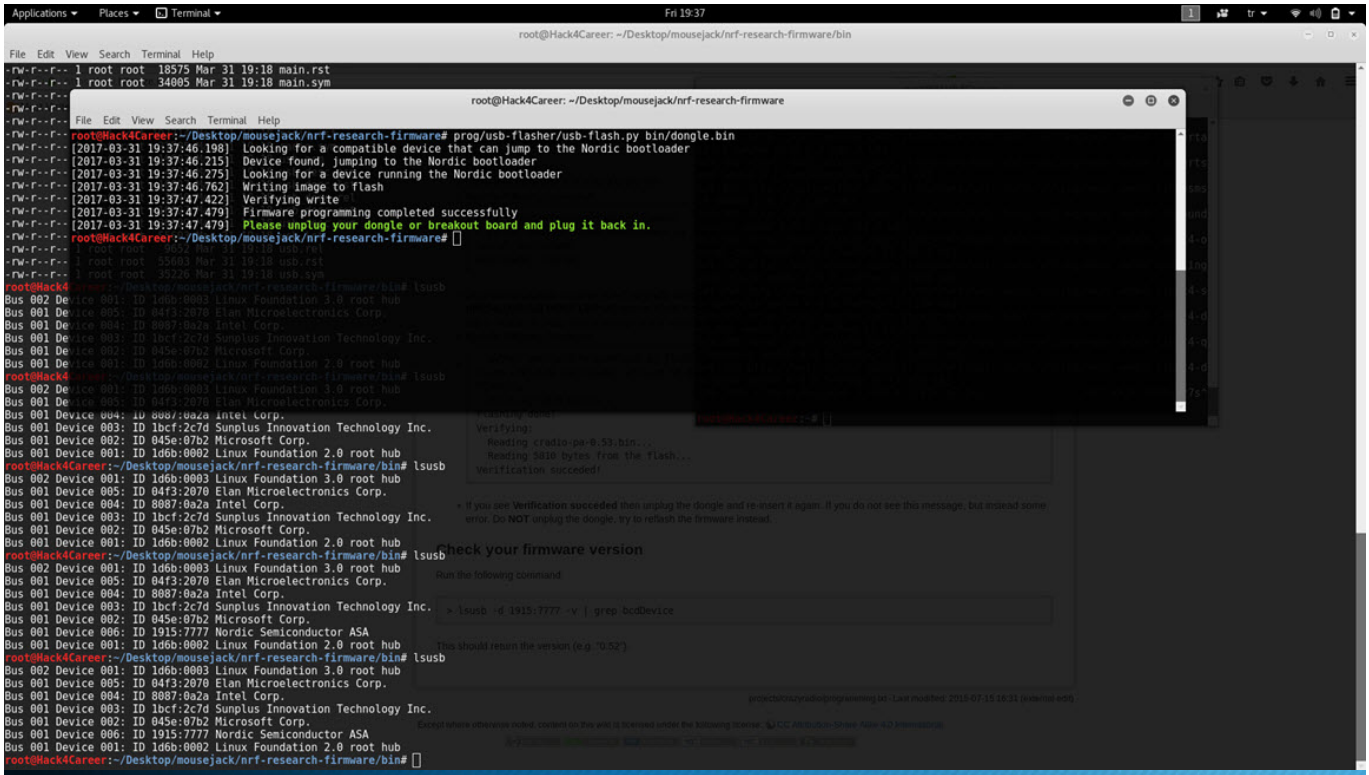
Over the years, these research and studies have led to wireless keyboards (2.4 GHz ISM) and computers using RF communication being encrypted with strong algorithms by manufacturers (with exceptions) to prevent malicious individuals from monitoring key information. While manufacturers have made efforts to make wireless keyboards secure, wireless mice have been left behind. After all, what use could a malicious person have for monitoring the movements and button presses (right, left, middle) of a mouse? The truth is, it is not that simple. In 2015, Bastille firm revealed a research called MouseJack and a method that affected numerous manufacturers (video). The MouseJack method uses a USB receiver that is connected to a computer as a mouse receiver to send wireless mouse movements and pressed buttons as keyboard keystroke data in the Ducky Script format like in my article called Bad USB. This allows even if you use a laptop, you don't have to use a wireless keyboard even if you use a wireless mouse, if you step away from

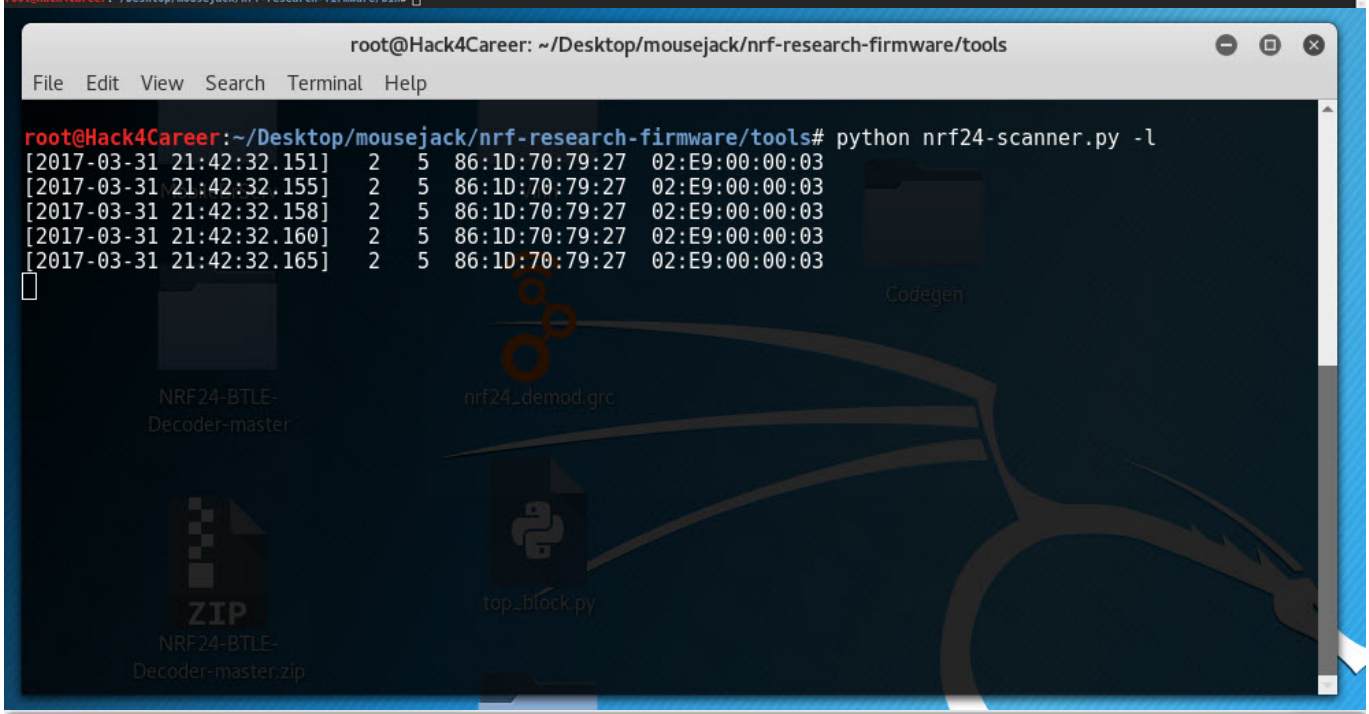
your computer for a short time, a malicious person can wirelessly send keystroke data to the USB receiver connected to your computer as if it were sent from a wireless mouse!

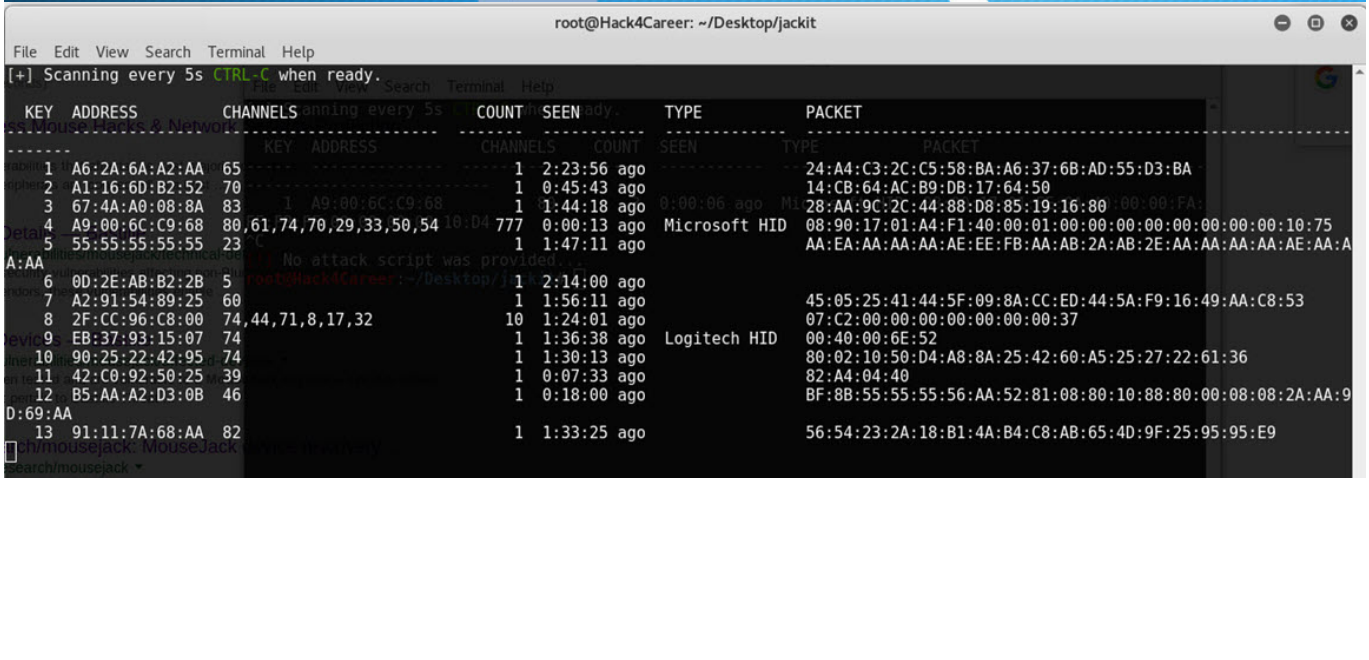
As someone who uses a wireless Microsoft mouse, I immediately set out to determine if the MouseJack method would affect my mouse and decided to purchase the CrazyRadio PA USB device as specified on Bastille's MouseJack GitHub page. After compiling the nrf-research-firmware firmware and uploading it to the CrazyRadio PA (bin/dongle.bin), I saw that the tools on Bastille's GitHub page allow detection and tracking of packets from nRF24L01+ devices around. I decided to do a small research on GitHub as Bastille only share the tool that can send keyboard keystroke data with manufacturers, and soon I came across the jacjackitkit tool that also allows sending keyboard keystroke data.



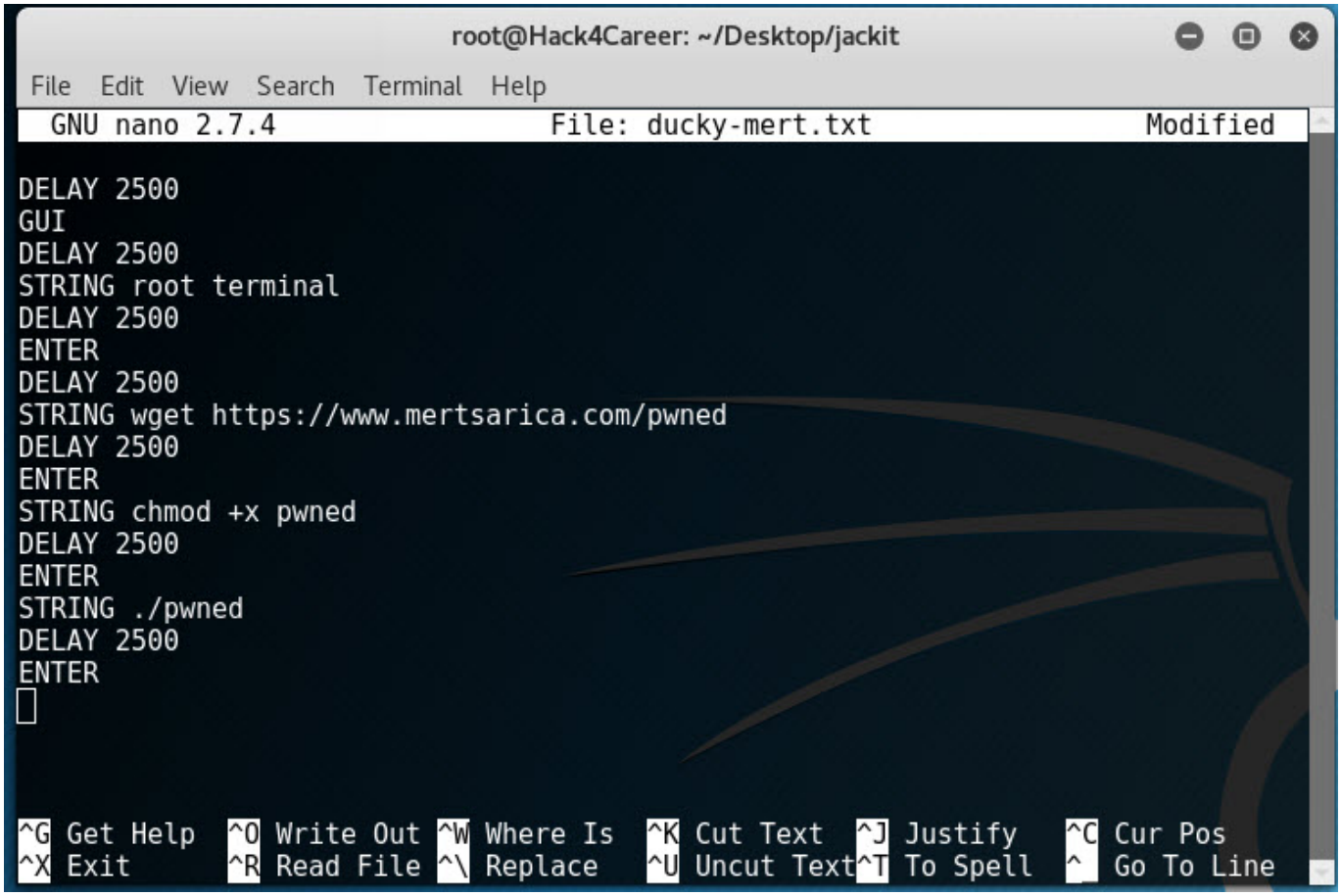

Applications
Places
Terminal
Fri 19:36
root@Hack4Career: ~/Desktop/mousejack/nrf-research-firmware/bin
File Edit View Search Terminal Help
-rw-r--r-- 1 root root 18575 Mar 31 19:18 main.rst
-rw-r--r-- 1 root root 34005 Mar 31 19:18 main.sym
root@Hack4Career: ~/Desktop/crazyradio-firmware/firmware
File Edit View Search Terminal Help
root@Hack4Career:~/Desktop/crazyradio-firmware/firmware# make CRPA=1
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/main.c -o bin/main.rel
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/radio.c -o bin/radio.rel
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/usb.c -o bin/usb.rel
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/usbDescriptor.c -o bin/usbDescriptor.rel
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/led.c -o bin/led.rel
sdcc -Iinc/ --model-large --std-sdcc99 -DCRPA -c src/utils.c -o bin/utils.rel
sdcc --xram-loc 0x8000 --xram-size 2048 --model-large bin/main.rel bin/radio.rel bin/usb.rel bin/usbDescriptor.rel bin/led.rel bin/utils.rel -o bin/cradio.ihx
objcopy -I ihex bin/cradio.ihx -O binary bin/cradio.bin
Crazyradio PA build
root@Hack4Career:~/Desktop/crazyradio-firmware/firmware# python ../usbtools/launchBootloader.py
Bootloader already launched.
root@Hack4Career:~/Desktop/crazyradio-firmware/firmware# python ../usbtools/nrfbootload.py flash bin/cradio.bin
('Found nRF24LU1 bootloader version', '18.0')
Flashing:
Flashing 7471 bytes...
Flashing done!
Verifying:
Reading bin/cradio.bin...
Reading 7471 bytes from the flash...
Verification succeded!
root@Hack4Career:~/Desktop/crazyradio-firmware/firmware#
Bus 001 Device 004: ID 8087:0a2a Intel Corp.
Bus 001 Device 003: ID 1bcf:2c7d Sunplus Innovation Technology Inc.
Bus 001 Device 002: ID 045e:07b2 Microsoft Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware/bin# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 04f3:2070 Elan Microelectronics Corp.
Bus 001 Device 004: ID 8087:0a2a Intel Corp.
Bus 001 Device 003: ID 1bcf:2c7d Sunplus Innovation Technology Inc.
Bus 001 Device 002: ID 045e:07b2 Microsoft Corp.
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware/bin# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 04f3:2070 Elan Microelectronics Corp.
Bus 001 Device 004: ID 8087:0a2a Intel Corp.
Bus 001 Device 003: ID 1bcf:2c7d Sunplus Innovation Technology Inc.
Bus 001 Device 002: ID 045e:07b2 Microsoft Corp.
Bus 001 Device 006: ID 1915:7777 Nordic Semiconductor ASA
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware/bin# lsusb
Bus 002 Device 001: ID 1d6b:0003 Linux Foundation 3.0 root hub
Bus 001 Device 005: ID 04f3:2070 Elan Microelectronics Corp.
Bus 001 Device 004: ID 8087:0a2a Intel Corp.
Bus 001 Device 003: ID 1bcf:2c7d Sunplus Innovation Technology Inc.
Bus 001 Device 002: ID 045e:07b2 Microsoft Corp.
Bus 001 Device 006: ID 1915:7777 Nordic Semiconductor ASA
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware/bin#


root@Hack4Career: ~/Desktop/mousejack/nrf-research-firmware
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware# prog/usb-flasher/usb-flash.py bin/dongle.bin
[2017-03-31 19:37:46.198] Looking for a compatible device that can jump to the Nordic bootloader
[2017-03-31 19:37:46.215] Device found, jumping to the Nordic bootloader
[2017-03-31 19:37:46.275] Looking for a device running the Nordic bootloader
[2017-03-31 19:37:46.762] Writing image to flash
[2017-03-31 19:37:47.422] Verifying write
[2017-03-31 19:37:47.479] Firmware programming completed successfully
[2017-03-31 19:37:47.479] Please unplug your dongle or breakout board and plug it back in.
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware#


root@Hack4Career: ~/Desktop/mousejack/nrf-research-firmware/tools
File Edit View Search Terminal Help
root@Hack4Career:~/Desktop/mousejack/nrf-research-firmware/tools# python nrf24-scanner.py -l
[2017-03-31 21:42:32.151] 2 5 86:1D:70:79:27 02:E9:00:00:03
[2017-03-31 21:42:32.155] 2 5 86:1D:70:79:27 02:E9:00:00:03
[2017-03-31 21:42:32.158] 2 5 86:1D:70:79:27 02:E9:00:00:03
[2017-03-31 21:42:32.160] 2 5 86:1D:70:79:27 02:E9:00:00:03
[2017-03-31 21:42:32.165] 2 5 86:1D:70:79:27 02:E9:00:00:03


root@Hack4Career: ~/Desktop/jackit
File Edit View Search Terminal Help
[+] Scanning every 5s CTRL-C when ready.
KEY ADDRESS CHANNELS COUNT SEEN TYPE PACKET
1 A6:2A:6A:A2:AA 65 1 2:23:56 ago 24:A4:C3:2C:C5:58:BA:A6:37:6B:AD:55:D3:BA
2 A1:16:6D:B2:52 70 1 0:45:43 ago 14:CB:64:AC:B9:DB:17:64:50
3 67:4A:A0:08:8A 83 1 1:44:18 ago 28:AA:9C:2C:44:88:D8:85:19:16:80
4 A9:00:6C:C9:68 80,61,74,70,29,33,50,54 777 0:00:13 ago Microsoft HID 08:90:17:01:A4:F1:40:00:01:00:00:00:00:00:00:00:00:10:75
5 55:55:55:55:55 23 1 1:47:11 ago AA:EA:AA:AA:AA:AE:EE:FB:AA:AB:2A:AB:2E:AA:AA:AA:AA:AE:AA:A
A:AA
6 0D:2E:AB:B2:2B 5 1 2:14:00 ago
7 A2:91:54:89:25 60 1 1:56:11 ago 45:05:25:41:44:5F:09:8A:CC:ED:44:5A:F9:16:49:AA:C8:53
8 2F:CC:96:C8:00 74,44,71,8,17,32 10 1:24:01 ago 07:C2:00:00:00:00:00:00:00:37
9 EB:37:93:15:07 74 1 1:36:38 ago Logitech HID 00:40:00:6E:52
10 90:25:22:42:95 74 1 1:30:13 ago 80:02:10:50:D4:A8:8A:25:42:60:A5:25:27:22:61:36
11 42:C0:92:50:25 39 1 0:07:33 ago 82:A4:04:40
12 B5:AA:A2:D3:0B 46 1 0:18:00 ago BF:8B:55:55:55:56:AA:52:81:08:80:10:88:80:00:08:08:2A:AA:9
D:69:AA
13 91:11:7A:68:AA 82 1 1:33:25 ago 56:54:23:2A:18:B1:4A:B4:C8:AB:65:4D:9F:25:95:95:E9

After installing the Jackit tool on Kali, I immediately began sending keyboard keystroke data prepared in Ducky Script format using the Crazyradio PA. After a short period of time, a root terminal opened on Kali, the pwned file was downloaded from https://www.mertsarica.com using wget and executed.




After this research, I regretfully threw away my Microsoft brand wireless mouse and headed to an electronics store to buy a more secure wireless mouse. I hope that this research I did for my physical security and security awareness will be useful for those who use wireless keyboard and mice. I wish you all safe days and look forward to seeing you in my next article.