MouseJack: Injecting Keystrokes into Wireless Mice

Marc Newlin | marc@bastille.net | @marcnewlin
Marc Newlin

Security Researcher @ Bastille Networks
(((Mouse|Key)Jack|KeySniffer))

- **Wireless mice and keyboards**
  - 16 vendors
  - proprietary protocols (non-Bluetooth)
  - 4 families of transceivers
- **16 vulnerabilities**
  - keystroke sniffing
  - keystroke injection
  - many are unpatchable
Types of vulnerabilities

- **Keystroke Injection**
  - Unencrypted, targeting mice
  - Unencrypted, targeting keyboards
  - Encrypted, targeting keyboards

- **Keystroke Sniffing**
  - Unencrypted keyboards

- **Forced Pairing**
  - Logitech Unifying dongles
  - Keyboard disguised as mouse

- **Malicious macro programming**
  - Delayed keystroke injection

- **Denial of service**
  - Crash USB dongle firmware
Turns out everybody makes vulnerable devices...
Prior Research

Thorsten Schroeder and Max Moser (2010)

- “Practical Exploitation of Modern Wireless Devices” (KeyKeriki)
- Research into XOR encrypted Microsoft wireless keyboards

Travis Goodspeed (2011)

- “Promiscuity is the nRF24L01+’s Duty”
- Research into nRF24L pseudo-promiscuous mode functionality

Samy Kamkar (2015)

- KeySweeper
- Microsoft XOR encrypted wireless keyboard sniffer
How do mice and keyboards work?
Peripherals send user input to dongle
Dongle sends user input to computer
An attacker can talk to your dongle...
or eavesdrop on your unencrypted keyboard
Background and Motivation
"Since the displacements of a mouse would not give any useful information to a hacker, the mouse reports are not encrypted."

- Logitech (2009)
Initial Logitech mouse research

- USRP B210 SDR
- Logitech M510 mouse
- GNU Radio decoder
- Good for passive RX
- USB and CPU latency make two way communications tricky
Burning Man to the rescue! (duh)
NES controller internals

- Arduino Nano
- DC boost converter
- nRF24L01+
- vibration motor
- WS2812B LED
Logitech mouse hijacking NES controller
“Village Adventure”  
by Marc Newlin

IoT Village

a Logitech mouse clicker

did not like the hax
NES Controller v2 (now with more things!)
NES controller v2 internals

- Teensy 3.1
- 5x nRF24L01+ radios
- 1x WS2812B RGB LED
- 500mAh LiPo battery
- microSD card reader
- OLED display
OSK attack @ ToorCon

- Windows 8.1/10
- Deterministically launch split OSK
- Keys are at known offsets from screen corners, assuming default DPI
- Slow, very slow
Discovering that first vulnerability

- Logitech Unifying keyboards
- Unencrypted keystroke injection
- Is it really that easy?
I’ll take one of each, please...
Research Process
Gather OSINT and implement SDR decoder

- **FCC test reports**
  - Frequencies
  - Modulation (sometimes)
- **RFIC documentation**
  - Physical layer configuration
  - Packet formats
- **The Google**
  - “How hack mice?”
  - “Why keyboard not encrypt?”
- **SDR decoder**
  - GNU Radio
  - USRP B210
  - 2.4GHz ISM band
  - 500kHz, 1MHz, 2MHz GFSK

**Caution**

Please use the Transceiver in human house only and keep away water.

Children don’t to install the Transceiver.
Build out a protocol model

1. Generate some ARFz
   a. Move the mouse, click some buttons
   b. Type on the keyboard

2. What data is sent over the air, and when?
   a. Infer payload structures
   b. Observe protocol behavior (channel hopping, ACKs, crypto, etc)
Look for low hanging fruit

- Wireless mice
  - All tested mice are unencrypted
  - Does it transmit keystrokes?
  - Does it send raw HID data?

- Wireless keyboards
  - Is the keyboard unencrypted?
  - Is it replay vulnerable?
Fuzzing (poke it and see what breaks)

- **usbmon / wireshark**
  - USB sniffing to see what the dongle sends to the host computer

- **xinput / magic sysrq**
  - Disable xinput processing of target keyboards and mice
  - Disable magic sysrq to avoid those pesky unintended hard reboots

- **fuzzer**
  - NES controller, and later custom nRF24LU1+ firmware
Nordic Semiconductor nRF24L
Nordic Semiconductor nRF24L Family

- 2.4GHz GFSK transceivers
- 250kbps, 1Mbps, 2Mbps data rates
- 0-32 byte payloads, 8 or 16 bit CRC
- Vendor defined mouse/keyboard protocols

<table>
<thead>
<tr>
<th>Transceiver</th>
<th>8051 MCU</th>
<th>128-bit AES</th>
<th>USB</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>nRF24LE1</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Flash</td>
</tr>
<tr>
<td>nRF24LE1 OTP</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>OTP (no firmware updates)</td>
</tr>
<tr>
<td>nRF24LU1+</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Flash</td>
</tr>
<tr>
<td>nRF24LU1+ OTP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>OTP (no firmware updates)</td>
</tr>
</tbody>
</table>
nRF24L Enhanced Shockburst

**MAC Layer Functionality**
- Automatic ACKs
- Automatic retransmit

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Address</th>
<th>Packet Control Field</th>
<th>Payload</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>3-5 bytes</td>
<td>9 bits</td>
<td>0-32 bytes</td>
<td>1-2 bytes</td>
</tr>
</tbody>
</table>

Payload Length: 6 bits
PID: 2 bits
No ACK: 1 bits

Enhanced Shockburst Packet Format
Common nRF24L Configuration

- "Standardized" properties
  - 2 Mbps data rate
  - 5 byte address length
  - 2 byte CRC
  - Automatic ACKs
  - Automatic retransmit

- Vendor specific properties
  - RF channels
  - Payload lengths
Logitech Unifying
Logitech Unifying

- **Universal pairing**
  - Any mouse or keyboard can pair with any dongle

- **Firmware update support**
  - Dongles support firmware updates
  - Most mice/keyboards do not

- **Transceivers**
  - nRF24LU1+ / nRF24LE1 (most common)
  - TI-CC2544 / TI-CC2543 (higher end)
  - All OTA compatible

- **Encryption**
  - Mice are unencrypted
  - Keyboard multimedia keys are unencrypted
  - Regular keyboard keys are encrypted with 128-bit AES
  - Key generation during pairing

- **Some Dell products are really Unifying**
  - Dell KM714
  - Likely others
Logitech Unifying Base Packet Format

<table>
<thead>
<tr>
<th>Preamble</th>
<th>Address</th>
<th>PCF</th>
<th>Enhanced Shockburst Payload</th>
<th>CRC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>5 bytes</td>
<td>9 bits</td>
<td>5, 10, or 22 bytes</td>
<td>2 bytes</td>
</tr>
</tbody>
</table>

Unifying Payload

- 4, 9, or 21 bytes

Checksum

- 1 byte

(Logitech Unifying Packet Format)

- 5, 10, and 22 byte payloads
- 1 byte payload checksum
Logitech Unifying Addressing

- Lowest octet is device ID
  - Defaults to 07 from the factory
- Device ID increments when you pair a new device
  - Re-pairing a device doesn’t change its ID
- Device ID 00 is reserved for the dongle

<table>
<thead>
<tr>
<th>Example RF Addressing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dongle serial number</td>
</tr>
<tr>
<td>Dongle RF address</td>
</tr>
<tr>
<td>Paired device 1 RF address</td>
</tr>
<tr>
<td>Paired device 2 RF address</td>
</tr>
<tr>
<td>Paired device 3 RF address</td>
</tr>
</tbody>
</table>
Logitech Unifying Payload Addressing

<table>
<thead>
<tr>
<th>RF Address</th>
<th>Payload</th>
<th>Addressing Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>11:22:33:44:07 (Dongle Address)</td>
<td>00:XX:XX:XX:XX</td>
<td>Transmit to the address of a paired mouse and ignore the device index field</td>
</tr>
<tr>
<td>11:22:33:44:00 (Mouse Address)</td>
<td>07:XX:XX:XX:XX</td>
<td>Transmit payload to the dongle address and use the device index field</td>
</tr>
</tbody>
</table>

(Logitech Unifying Payload Format)
ACK Payloads (Dongle to Peripheral Cmds)

- Keystroke
- Ack
  - Normal Ack
  - Ack Payload
  - Ack + CMD
Logitech Unifying ACK Payload Example

Logitech Unifying Dynamic Keepalives

- Keepalives are used to detect poor channel conditions
- Missed a keepalive? Change channels
- Mouse/keyboard dynamically sets keepalive interval
- Short interval when active, long interval when idle

<table>
<thead>
<tr>
<th>Unused</th>
<th>Frame Type (0x4F)</th>
<th>Unused</th>
<th>Timeout</th>
<th>Unused</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>4 bytes</td>
<td>1 byte</td>
</tr>
</tbody>
</table>

(Logitech Set Keepalive Timeout Payload)

<table>
<thead>
<tr>
<th>Unused</th>
<th>Frame Type (0x40)</th>
<th>Timeout</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 byte</td>
<td>1 byte</td>
<td>2 bytes</td>
<td>1 byte</td>
</tr>
</tbody>
</table>

(Logitech Keepalive Payload)
Logitech Unifying Keepalives - Example

[20.173] 4C:29:9D:C6:09 00:C2:00:00:01:00:00:00:00:3D // mouse movement (implicitly sets keepalive interval to 8ms)
[20.181] 4C:29:9D:C6:09 00:4F:00:00:6E:00:00:00:00:43 // no movement after 8ms, set keepalive interval to 110ms
[20.189] 4C:29:9D:C6:09 00:C2:00:00:01:00:00:00:00:3D
[20.196] 4C:29:9D:C6:09 00:C2:00:00:01:00:00:00:00:3D
...
[20.282] 4C:29:9D:C6:09 00:C2:00:00:00:E0:FF:00:00:5F
[20.289] 4C:29:9D:C6:09 00:C2:00:00:00:F0:FF:00:00:4F
[20.297] 4C:29:9D:C6:09 00:4F:00:00:6E:00:00:00:00:43 // no movement after 8ms, set keepalive interval to 110ms
[20.305] 4C:29:9D:C6:09 00:40:00:6E:52 // keepalive at 110ms interval
[20.390] 4C:29:9D:C6:09 00:40:00:6E:52
[20.483] 4C:29:9D:C6:09 00:40:00:6E:52
...
[25.377] 4C:29:9D:C6:09 00:40:00:6E:52
[25.470] 4C:29:9D:C6:09 00:40:00:6E:52
[25.563] 4C:29:9D:C6:09 00:4F:00:00:04:B0:00:00:00:FD // after 5 seconds idle, increase keepalive interval to 1200ms
[25.571] 4C:29:9D:C6:09 00:40:04:B0:0C // keepalive at 1200ms interval
[26.533] 4C:29:9D:C6:09 00:40:04:B0:0C
[27.486] 4C:29:9D:C6:09 00:40:04:B0:0C
[28.439] 4C:29:9D:C6:09 00:40:04:B0:0C
[29.392] 4C:29:9D:C6:09 00:40:04:B0:0C
[30.345] 4C:29:9D:C6:09 00:40:04:B0:0C
Logitech Unifying Pairing

1. Unifying software tells the dongle to enter pairing mode
2. Dongle listens to pairing requests on address BB:0A:DC:A5:75
3. Dongle times out if pairing doesn’t happen in 30-60 seconds
4. Device type and properties are sent during pairing
Logitech Unifying Device Power-on Behavior

Dongle
Where you at?

[crickets chirping]

Yo, anybody want to pair??
Vulnerabilities
Encrypted Protocols

Unencrypted Injection

YOU BUILT A PROPRIETARY ENCRYPTED PROTOCOL?

YOU MUST BE REALLY SMART
Unencrypted Injection Targeting Keyboard
(Logitech Unifying, Dell KM714)
Forced Pairing
(Logitech Unifying, Dell KM714)
Disguise Mouse as Keyboard
(Logitech Unifying, Dell KM714)
Logitech Response, Round 1

- Vendor notified on 11-24-2015
- Public disclosure on 02-23-2016
- Firmware update released on 02-23-2016
  - Fixed forced pairing
  - Partially fixed unencrypted keystroke injection
  - Also applies to Dell KM714
Bypassing The Fix

- Use Linux
- Use OSX
- Install Logitech Setpoint

lol, wut?
Logitech Response, Round 2a

- Vendor notified on 04-27-2016
- Public disclosure on 07-26-2016
- Firmware update released on 07-26-2016
  - Fixed unencrypted keystroke injection
  - Also applies to Dell KM714
Logitech G900 Chaos Spectrum

- "Professional Grade Wireless" gaming mouse ($150!!!)
  - Tuned (and power hungry) version of Unifying
  - Shorter ACK timeouts
  - 8 channels vs. 24 with Unifying
  - No pairing support
  - USB connection to charge or use as a wired mouse
  - TI-CC2544/TI-CC2543 offers more TX power than nRF24L
  - Vulnerable to unencrypted keystroke injection!
Logitech G900 Macros

- Keystroke macros are programmed into the mouse
- Macros can be programmed wirelessly, by an attacker
- Timing delays can be inserted between keystrokes
Logitech Response, Round 2b

- Vendor notified on 04-27-2016
- Public disclosure on 07-26-2016
- Firmware update released on 07-26-2016
  - Fixed unencrypted keystroke injection (G900 gaming mouse)
Unencrypted Injection Targeting Mouse
(AmazonBasics, Dell KM632, Lenovo 500, Microsoft)
Microsoft Sculpt Ergonomic Mouse

Press to send unencrypted keystroke packet
Amazon response, round 1

- Unencrypted keystroke injection into mouse dongle
- Vendor notified on 11-24-2015
- Public disclosure on 02-23-2016
- No vendor response
Dell response, round 1

- Unencrypted keystroke injection into mouse dongle
- Vendor notified on 11-24-2015
- Public disclosure on 02-23-2016
- Dell fixed the firmware and sent an updated version to test, but firmware updates are not possible on existing devices
Lenovo response, round 1

- Unencrypted keystroke injection into mouse dongle
- Vendor notified on 11-24-2015
- Public disclosure on 02-23-2016
- Lenovo fixed the firmware and sent an updated version to test, but firmware updates are not possible on existing devices
Microsoft response

- Unencrypted keystroke injection into mouse dongle
- Vendor notified on 11-24-2015
- Public disclosure on 02-23-2016
- Microsoft released Windows update on 04-22-2016
  - Works on client versions of Windows (no server support)
  - Addresses mice, but not mouse/keyboard sets
  - No fix for Linux or OSX
  - No firmware update support
DDoS: Dongle Denial of Service

(Lenovo Ultraslim, Ultraslim Plus, N700)
Encrypted Protocols

Encrypted Injection

COUNTER MODE AES

DOES NOT WORK THAT WAY
Counter Mode AES

- nonce
- ctr

AES encrypt

- ciphertext

Data XOR

- ciphertext
- output
Repeated Counters? Give ‘em here!

keystroke 00001
keystroke 00001
keystroke 00001
keystroke 00001
keystroke 00001
Encrypted Keystroke Packets

- **Key Down**: 00001
- **Encrypted**: Incremented Counter
- **Key Up**: 00002
USB HID Key Up Ciphertext

"A" key down

key up

XOR'd w/ciphertext = ciphertext
Encrypted Keystroke Injection Devices

- Logitech Unifying keyboards (including Dell KM714)
- Dell KM632
- Lenovo Ultraslim
- AmazonBasics Wireless Keyboard
- HP Wireless Elite V2 Keyboard
Encrypted Keystroke Injection Responses

- Vendor notified on 04-27-2016
- Public disclosure on 07-26-2016
- Logitech *is working on a fix*
- Lenovo *is working on a fix*
- Dell updated the firmware and set us a fixed unit to verify, but firmware updates are not possible in the field
- No response from Amazon
- No acknowledgement of the vulnerability from HP
Unencrypted Protocols
Unencrypted Transceivers (KeySniffer)

- **MOSART Semiconductor (undocumented)**
  - 1Mbps or 375kbps GFSK
  - Single channel
  - No encryption

- **Signia SGN6210 (sparsely documented)**
  - 1Mbps GFSK
  - Frequency hopping
  - No encryption

- **GE/Jasco mystery transceiver (no idea what this thing is)**
  - 500kbps GFSK
  - Frequency hopping
  - No encryption
Unencrypted Devices - MOSART

- Anker Ultra Slim 2.4GHz Wireless Compact Keyboard
- EagleTec K104 / KS04 2.4 GHz Wireless Combo keyboard
- HP Wireless Classic Desktop wireless keyboard
- Insignia Wireless Keyboard NS-PNC5011
- Kensington Pro Fit Wireless Keyboard
- RadioShack Slim 2.4GHz Wireless Keyboard
- ShhhMouse Wireless Silent Mouse (injection only)
- HDE Slim Wireless Optical Mouse (injection only)
Unencrypted Devices - non-MOSART

- GE/Jasco 98614 Wireless Keyboard and Mouse
- Gigabyte K7600 Wireless Keyboard and Mouse
- Toshiba PA3871U-1ETB Wireless Keyboard
Dongle Sync Packets

“Keyboard, I'm over here!!”

↑

Every 8 or 16 milliseconds
Sniffing Multiple Keyboards
KeySniffer Vendor Responses

- **Anker** will exchange vulnerable keyboards for Bluetooth models through 08-30-2016

- **Kensington** claims to have a new AES encrypted version of the Pro Fit wireless keyboard
  - I have not seen or tested this device
  - FCC docs don’t show any new keyboards

- **Insignia** told reporters that its keyboards are encrypted, however the vulnerable model is unencrypted

- **GE/Jasco** is no longer making wireless keyboards/mice
White-label Hardware, White-Label Vulnerabilities
Vendor vs OEM: Hewlett-Packard / ACROX

HP Wireless Classic Desktop
- Added HP logo
- Modified side button style

ACROX KBJ+G1G
- OEM keyboard
Vendor vs OEM: AmazonBasics / Chicony

AmazonBasics Wireless Keyboard/Mouse

- Added AmazonBasics logos
- Dell KM632 (made by Chicony) has the same vulns as AmazonBasics

Chicony WUG1213

- OEM keyboard/mouse set
Vendor vs OEM: RadioShack / Siliten

RadioShack Wireless Keyboard
- Added RadioShack logo
- Small styling changes

Siliten DK/M-9091RL
- OEM keyboard
Attack Hardware
CrazyRadio PA and Open Sourced Firmware

- nRF24LU1+ w/ LNA/PA
- 225 meter injection w/ yagi
- Open source hardware
- Part of Crazyflie project
- Bastille Research firmware:
- https://github.com/BastilleResearch/mousejack/
Research Firmware on Logitech Dongles

- **Unifying dongles support firmware updates**
  - Logitech bootloader doesn’t do signature validation
  - Any firmware image that passes CRC is accepted

- **Flash the Bastille Research firmware onto a Logitech dongle:**
  - `sudo make logitech_install`

- **Cheap and available**
  - ~$10 vs ~$30 for the CrazyRadio PA
  - CrazyRadio PA harder to find after MouseJack release
  - Unifying dongles are widely available
Android App

- Device discovery and classification
  - Logitech devices
  - Microsoft devices
- Dongle firmware flashing support
  - CrazyRadio dongles
  - Logitech dongles
Demo Time!

YO DAWG, I HEARD YOU LIKE LOGITECH DONGLES

SO I ATTACKED YOUR LOGITECH DONGLE WITH A LOGITECH DONGLE
Questions?

Marc Newlin
marc@bastille.net
@marcnewlin