IOActive Labs: Breaking Embedded Devices

Mike Davis
Joshua Hammond
Thomas Kilbride
Daniel Schaffner

IOActive®

IOActive is the only global security consultancy with a state-of-the-art hardware lab and deep expertise spanning hardware, software and wetware services.

Talk Breakdown

- Who are we?
- The challenges we face
- How we are approaching things differently
- Then the fun part: Examples of wins
  - Segway
  - ATM vulnerabilities
  - Skimmer Research

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Mission: Who Are We?

The mission of our embedded labs: are to provide cutting-edge hardware security capabilities, conduct research to make the world more secure, and train the next generation of hardware security consultants.

IOActive, Inc. Copyright ©2017. All Rights Reserved.

A Few More Basics About Us

- **Been around since 1998**
- **Independently owned**
- **No outside funding**
- **Free to determine our own research paths**

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Challenges: Building a Space to Break Things

- Balancing three missions
  - Training
  - Research
  - Billable Work

- ROI on Non-Billable Work
  - 20% Ideal Increasingly difficult for teams to maintain
  - Structure of independent research choices
  - Short term needs vs long term requirements

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Thinking About Different Solutions:

- Clear Goals
  - What skills do you really need?
  - Which markets the company be moving into?
  - What are your Revenue / Product requirements?

- Define
  - Personnel passionate about a particular area
  - What wins look like
  - Understand a reasonable timeline
  - A path forward
  - Plans to advantage of chaos

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Summary

- It is possible to build research models to push a team forward inside of incredibly ‘busy’ environments

- Unpredictability will happen reliably – plan to take advantage

- Understand what will move your organization forward broadly and then you can use that to allow passionate individuals the ability to make tactical decisions

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

ATM Research

Research Outline

- Physical Bypass with a metal tool
- Security model of the (AFD)
- Opening up the Safe
  - Literally
  - Also, reversing firmware
- The protocol problem
- Being a professional: disclosure and such

IOActive, Inc. Copyright ©2017. All Rights Reserved.

ATM MAP

Upper Cabinet
- Handles user interaction
- Reads card
- Contacts bank

The Safe
- Only talks to Upper Cabinet
- Thick steel plates
- Contains funds

IOActive, Inc. Copyright ©2017. All Rights Reserved.

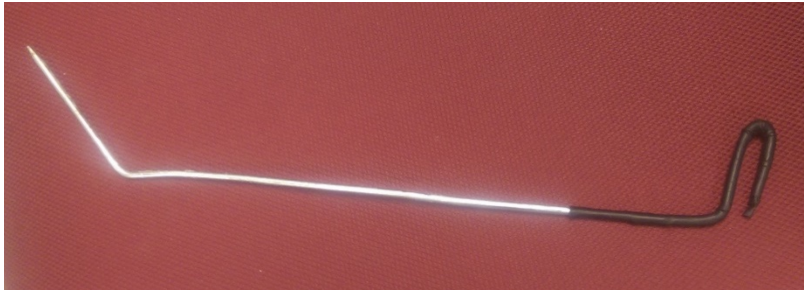
Physical Bypass

- Lock bar added to prevent accidentally leaving the unit open
- A finely crafted tool (metal rod) can be poked through a speaker hole and pop open the lock bar
- The cabinet opens up and we can see the guts of the ATM
- So… that's it right?

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Physical Bypass Pictures



IOActive, Inc. Copyright ©2017. All Rights Reserved.

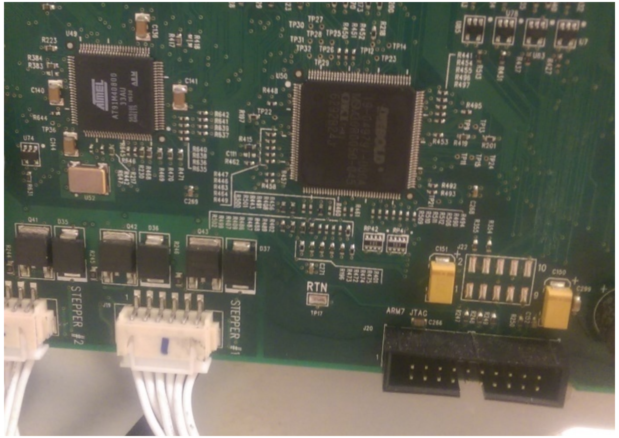
AFD Security Model - The safe is safe, right?

- The claim: access to the upper cabinet != access to the safe
- The safe has a controller inside of it to authenticate the access from the upper cabinet
- Safe is connected to the upper cabinet through a USB
- For us… game on

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Opening up the ATM

- Time to pop open the safe and take a look under the hood
- Lots of belts, slides, things that will totally break your fingers
- Also a whole lot of dust
- Giant controller board with fun Atmel processor
- Oh look, JTAG



IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive

Time to get to Work

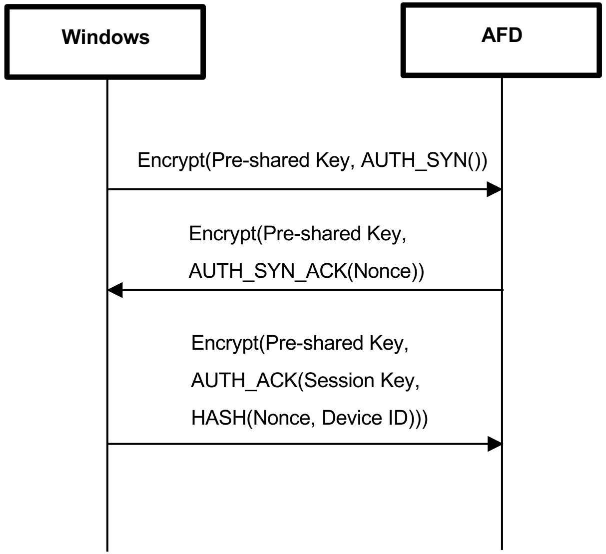
- USB Captures
  - Looks like it's mostly encrypted
  - Good sample for initializing the USB controller
- Pulling the Firmware
  - Find message types
  - Reverse out message structure: length fields and hashes and such
- Hooray for Debugging
  - Trace messages through execution, skip indirect C++ calls
  - Break after messages decrypt, pull sample plaintext

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

The Problem Protocol: How it's Supposed to Work

- Upper Cabinet encrypts a Hello message with a pre-shared AES key
- Safe responds with an encrypted acknowledgment
    - Contains a fun nonce
- Upper Cabinet sends a final packet encrypted with the pre-shared key
    - Contains secondary AES key for the session
    - Contains hash with the nonce and device-specific ID

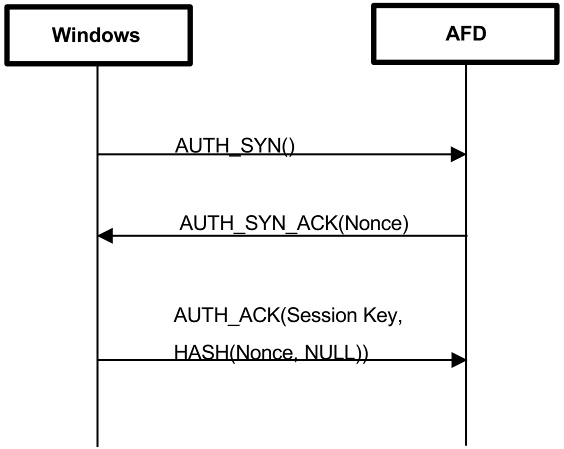


IOActive

IOActive, Inc. Copyright ©2017. All Rights Reserved.

The Problem Protocol: How it Actually works

- Turns out encryption is entirely optional
  - A header flag indicates if the message is encrypted
  - If the safe receives an unencrypted message, it responds unencrypted
- Safe sends back the nonce in the clear
- We get to respond with our own session key
  - We still don't know the device ID to finish the hash though
  - Good thing that's optional too



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Mitigation

- Only allow authentication messages if they’re encrypted
- Always check the device ID

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Version Updates Process

- 3/24/17: IOActive Follows up to find Affected Firmware Versions
- …
- Today: Still waiting on those versions…

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive

Summary

- We can get into the Upper Cabinet with a metal rod
- We can pull out the USB and connect to the safe
- We can authenticate to the safe and dispense currency
- Diebold wouldn’t tell us what it affects and if there are fixes available

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Hacking a Segway

Outline

- Finding Areas of Interest
- Hardware Reverse Engineering
- Bluetooth Sniffing/Protocol Analysis
- Exploiting Firmware Update Processes
- Modifying and uploading malicious firmware

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Identifying Areas of Interest

- Bluetooth Authentication
- Mobile Application
- Firmware Update Process
- Firmware Verification
- Safety Systems

IOActive, Inc. Copyright ©2017. All Rights Reserved.

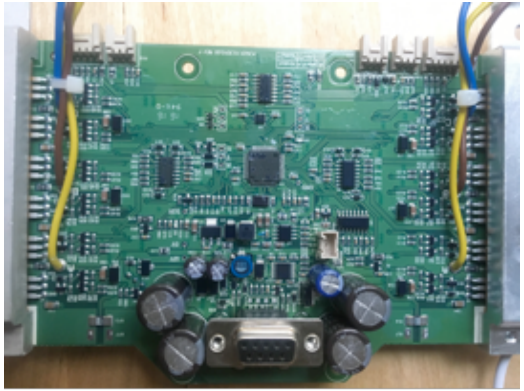
Hardware Reverse Engineering

- Gather information on every processor onboard
- Look for open Interfaces and try to connect to those interfaces
- Attempt to find open programming interfaces

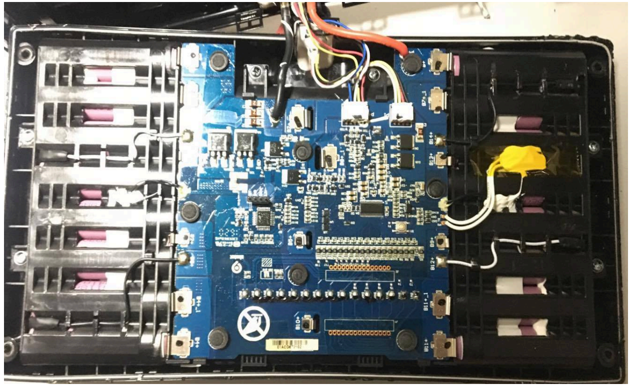
IOActive, Inc. Copyright ©2017. All Rights Reserved.

Internal Photographs

Driver Board



Battery Management System (BMS)



Bluetooth Module

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

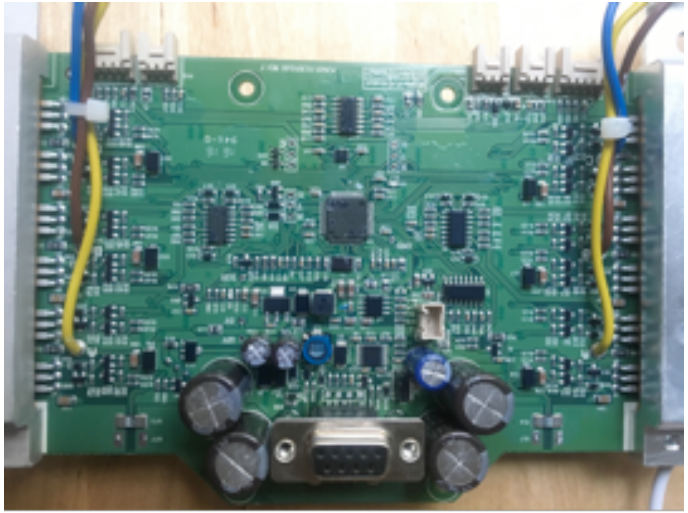
Hardware Documentation

- Main CPU was ARM Cortex-M3 (STM32F103 64-LQFP Package)
- Bluetooth Module is from Nordic Semiconductor
- Battery Management System has an STM8 Processor (No IDA Plugin??)

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Reverse Engineering the "Driver Board"

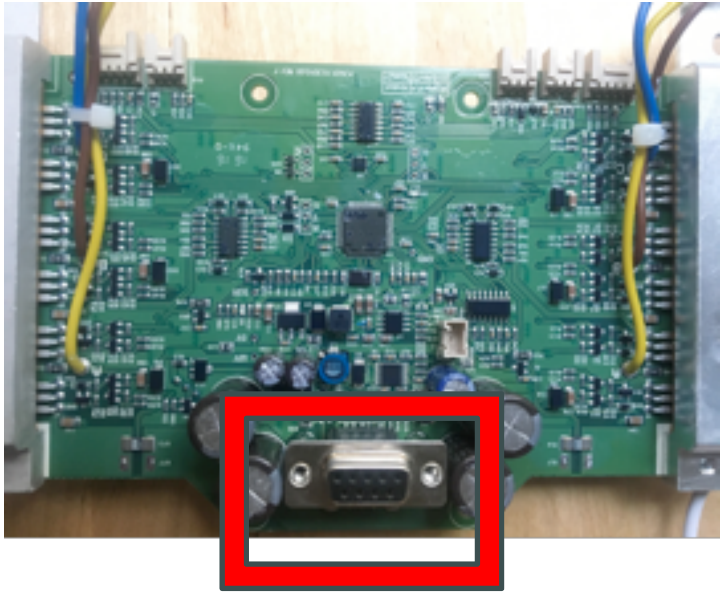
- Looking for headers which might reveal a console
- Use "Successive Probe-and-Pray™" Technique.
- Basically guess-and-check with a logic analyzer until you find something.



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Analyzing Internal Communications

- Looking for undocumented interfaces such as serial communications
  - Headers labeled R, T, G (Rx, Tx, GND) didn't show anything
  - Programming interface (SWD) locked
- Captured serial communications from one header which connects to the Bluetooth module (highlighted on right).

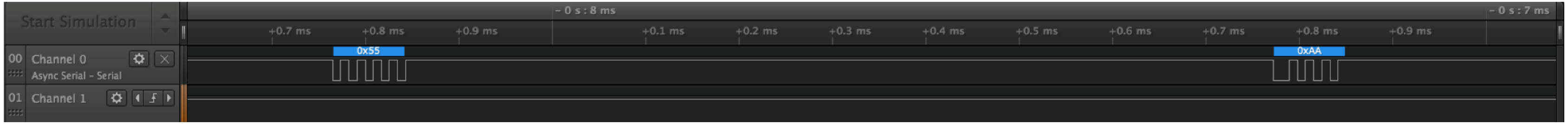


IOActive

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Serial Captures

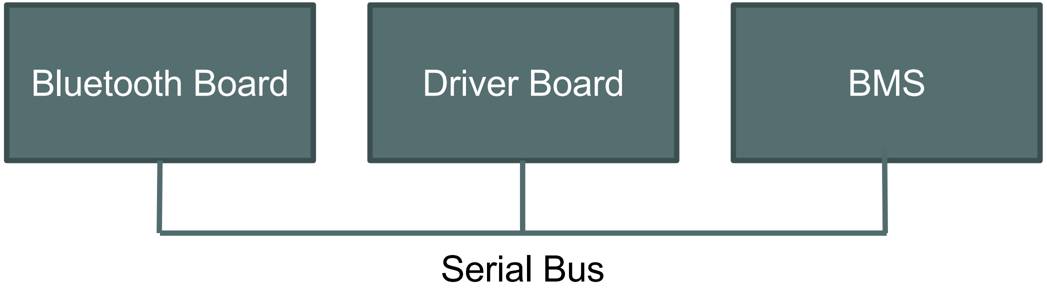
- Using a logic analyzer we recovered the following serial communications
- This appears to be a binary protocol which we will investigate later



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Internal Communications Analysis

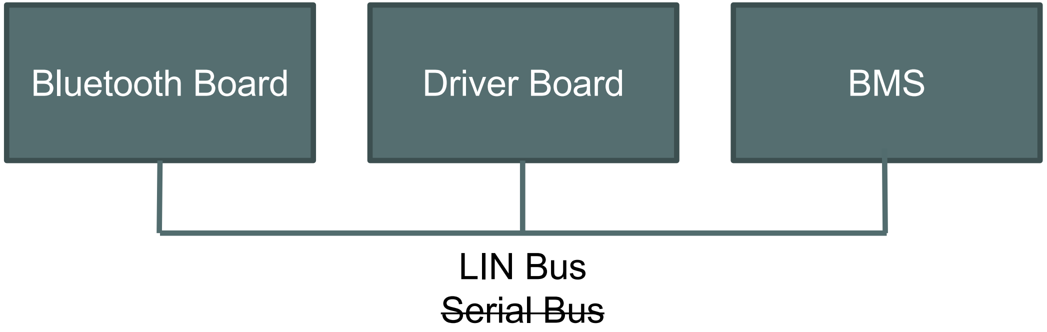
- The serial bus connects the onboard processors together
- RS232 serial is typical one device with one master
- This must not be standard RS232 because this is a multi-slave environment



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Internal Communications Analysis

- Local Interconnect Network (LIN) architecture is a multi-slave ”RS232-like” equivalent
- Performs Slave Selection by address
- Has a sync field which starts with 0x55



IOActive, Inc. Copyright ©2017. All Rights Reserved.

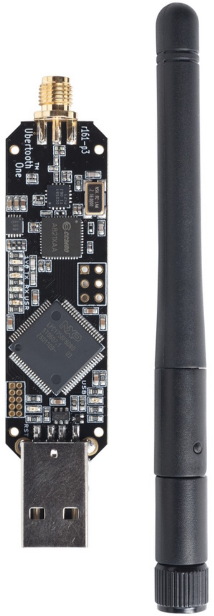
Summary up to this point

- What we know
  - Architectures of the chips
  - The bus used for internal communications on the hoverboard
- What we want to know
  - How does this device communicate to the outside world?

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Bluetooth Protocol Analysis

- Goal: Capture/Decode communication between the app and hoverboard.
- Second Goal: Determine how BT traffic is used in onboard busses
- Stretch Goal: See if we can circumvent any security controls.
- Tools used to analyze full chain of communication.
  - Wireshark (TCP/IP Captures)
  - Ubertooth One

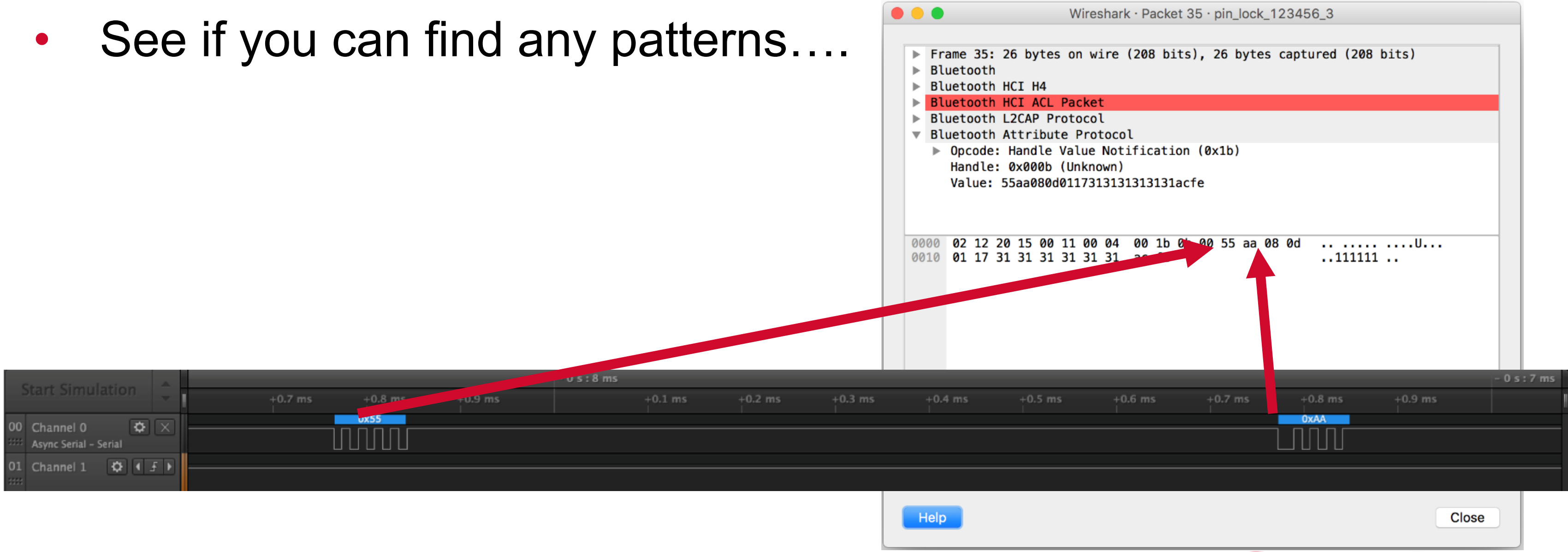


IOActive®

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Bluetooth Protocol Analysis

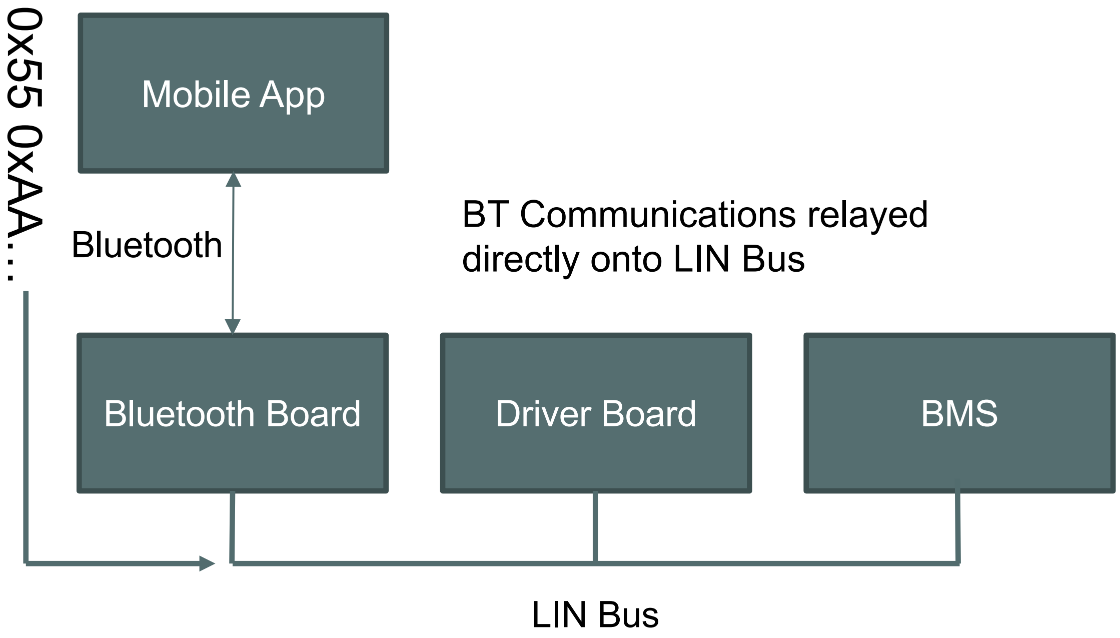
- Curious about security controls we want to understand how those function.
- 0x55 LIN Sync. Bits seen over Bluetooth
- See if you can find any patterns….



IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Communication Chain Breakdown

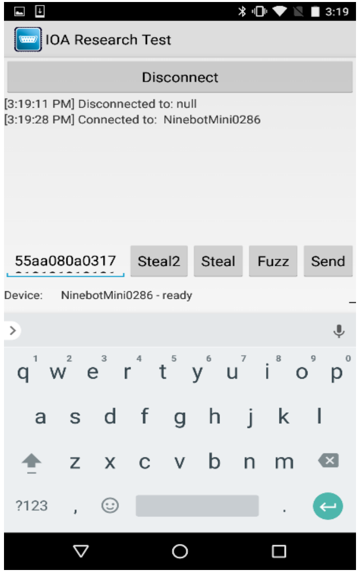


IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive

Bluetooth Protocol Analysis

- If we can sniff the packet to set the PIN, do we need to know the old PIN to set a new PIN?
  - Nope!
- Complete authentication bypass!



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Impacts of Segway Authentication Bypass

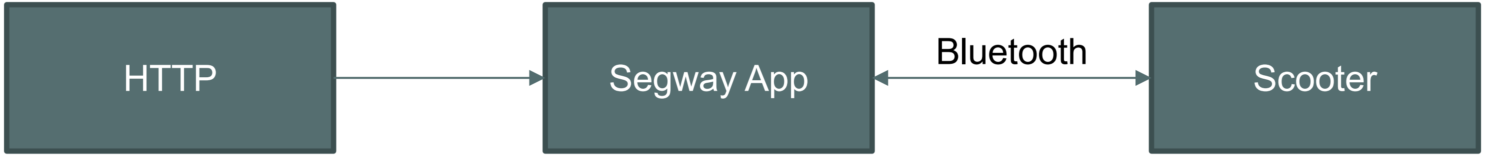
- Impact of this authentication Bypass is full access to Segway/Ninebot Rider Application
  - Set Security PIN
  - Set LED colors
  - Turn the motors off
  - Remote Control
  - Perform Firmware Updates

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive

Firmware Update System

- Originally firmware updates were served over HTTP (unencrypted)
- The Segway/Ninebot app downloads a JSON object and checks if an update is available
- If there is an update available, the app downloads it and then sends it to the segway over bluetooth



IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive

Exploiting the Update System

- Images are served over an unverified HTTP connection.
  - An attacker could perform a DNS Spoofing attack to serve updates
- Can we apply arbitrary updates to the hoverboard?
  - Yes! We tested this by sending a modified update to the device, it accepted and applied the update without any issues.

IOActive, Inc. Copyright ©2017. All Rights Reserved.

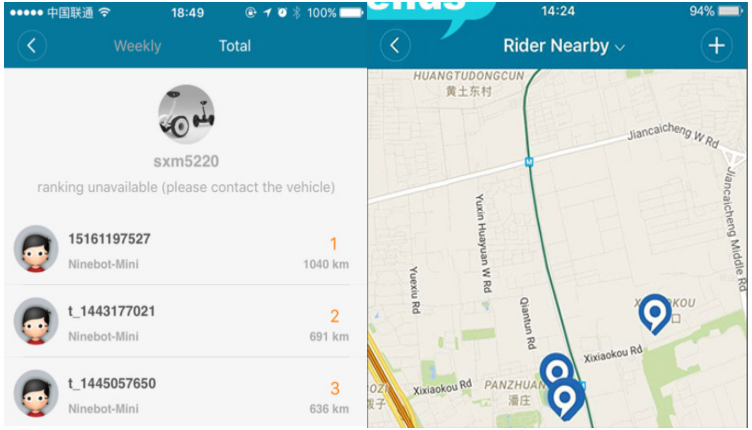
Summary up to this point

- Exploits that we have
  - Bluetooth PIN Authentication Bypass
  - Firmware updates served over HTTP
  - We can upload any firmware that we want to the device
    - A.k.a. Unsigned firmware updates
- What we want to know
  - How do we exploit this make someone faceplant?
  - How do we deploy this exploit in the field?

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Finding targets for deployment

- The hard work was done already…the app tracks your location and periodically uploads coordinates to Ninebot.
- Anyone with the app can see this in an easy to use map!
- The latest release of the Ninebot app has removed this feature



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Safety System Bypasses / Firmware RE

- Normally, the hoverboard will not turn off if there is a rider standing on it.
- To cause someone to faceplant we must bypass the checks that insure there is not a rider onboard with a firmware update, then we need to use bluetooth to turn it off.

IOActive, Inc. Copyright ©2017. All Rights Reserved.

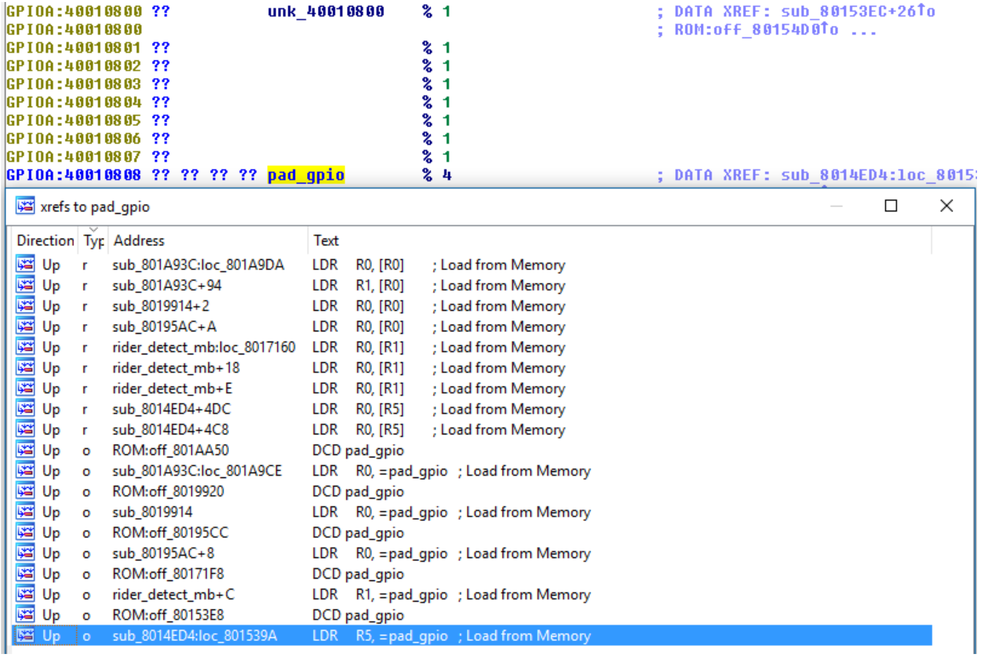
Firmware Reverse Engineering (RE)

- Using a multimeter we need to determine where the rider detection switch connects to the Driver Board
- Knowing the pin it's connected to allows us to reference datasheets to determine where this GPIO exists in the processor's Memory-Mapped Input / Output (MMIO).

IOActive, Inc. Copyright ©2017. All Rights Reserved.

Analyzing all XREF's in firmware

- Look at all places where the switch/GPIO state is evaluated using IDA Pro.



IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Bypassing Safety checks

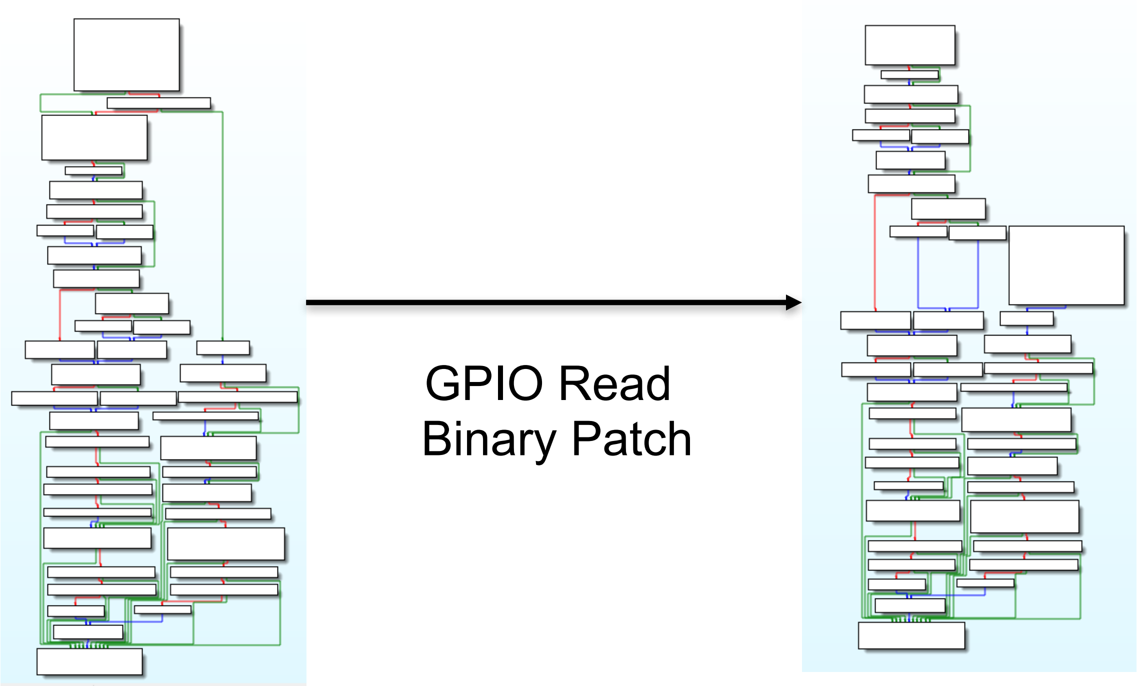
- The hoverboard appears to set a global variable when there is a rider onboard
- If we bypass this check with some Assembly Fu (Josh Hammond et al.) the hoverboard will no longer be able to check if it is safe to turn off
- INSERT Image of IDA-View differences between original FW and bypassed FW

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Bypassing Safety Check Bypass

- Again, Josh Hammond (ATM dude) and others helped a lot here with the fiddly bits. Thanks!



IOActive, Inc. Copyright ©2017. All Rights Reserved.

Conclusion

- Since PIN authentication was not verified before executing commands, I could perform privileged actions without first authenticating
  - Remote Control
  - Firmware Updates
  - PIN Changes, etc.
- Since updates were served over HTTP, I was able to easily force the application to update to malicious firmware

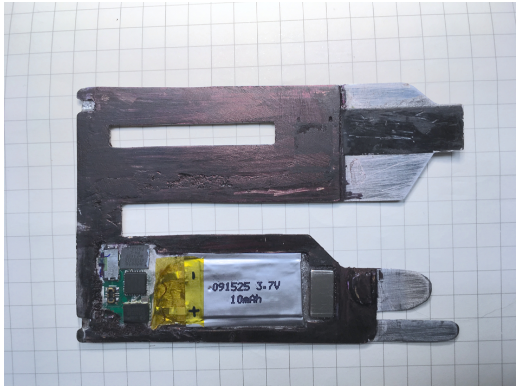
IOActive, Inc. Copyright ©2017. All Rights Reserved.

Potential Mitigations

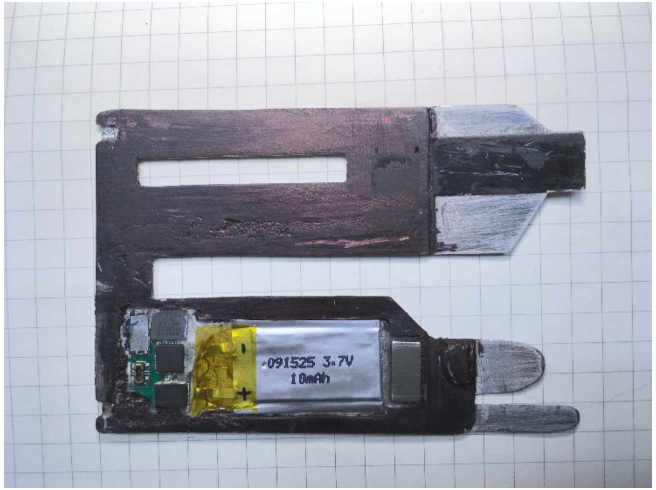
- Use verify bluetooth PIN authentication to prevent someone from gaining unauthorized access or executing arbitrary commands
- Check firmware updates cryptographically for integrity and vaidity
- Use encryption to prevent someone from sniffing credentials
- Mitigate MitM attacks by enforcing transport security (HTTPS) to send firmware updates with pinned certificates.
- Don’t expose rider locations to the public.

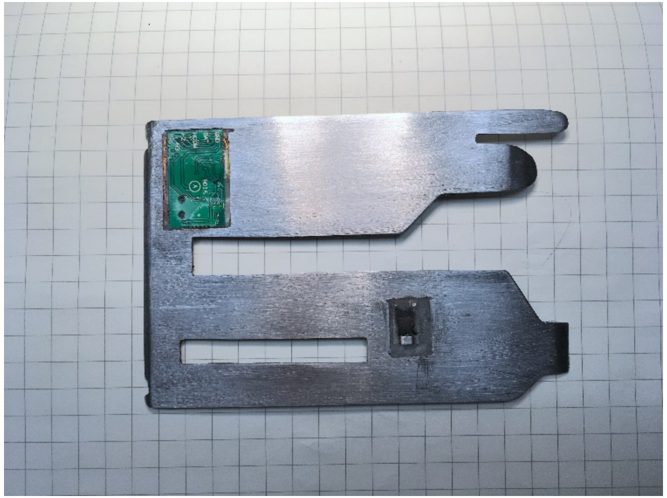
IOActive, Inc. Copyright ©2017. All Rights Reserved.

Skimmer Research



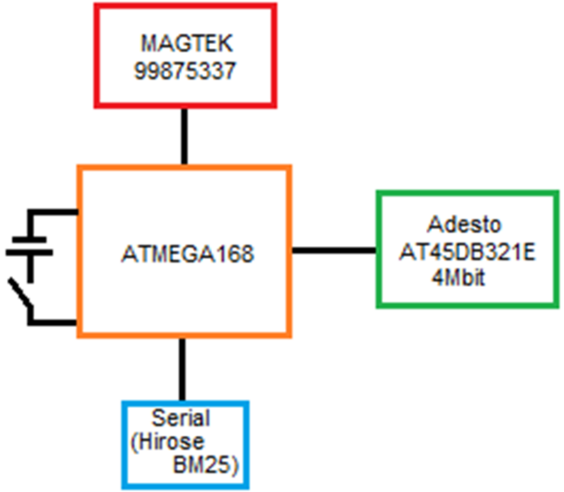
Skimmer Detail 1

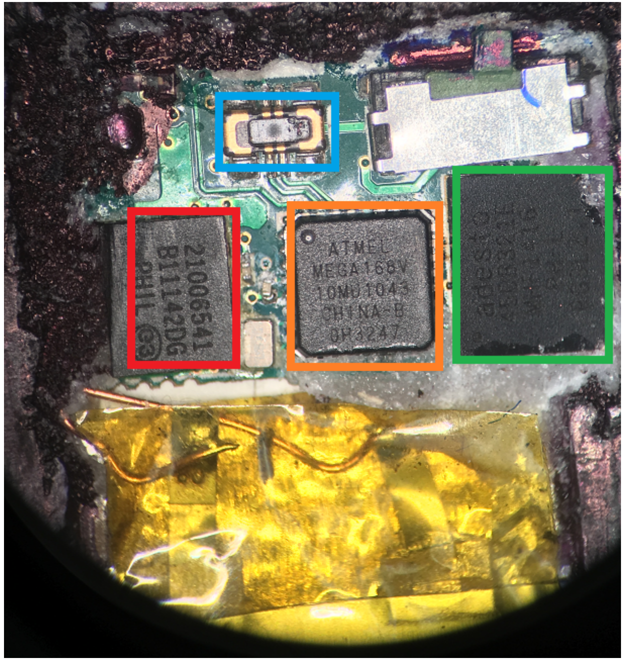




IOActive, Inc. Copyright ©2017. All Rights Reserved.

Skimmer Detail 2





IOActive, Inc. Copyright ©2017. All Rights Reserved.

Summary of Reader

IOActive, Inc. Copyright ©2017. All Rights Reserved.

IOActive®

Thank You

Email:
@IOActive