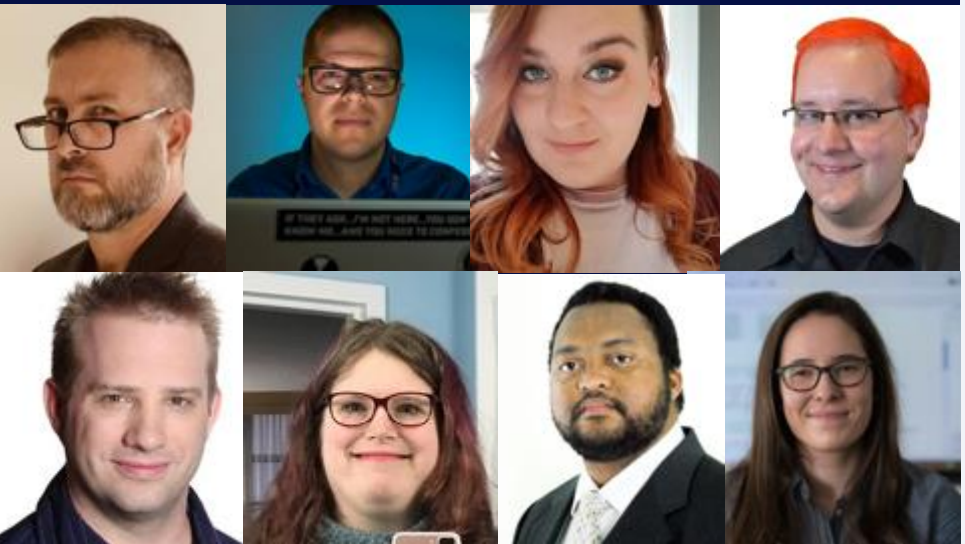
Know your devices: Why penetration testing is an essential security process

X-Force Red

IBM

Who is X-Force Red



X-Force Red is an autonomous team of veteran hackers, within IBM Security, hired to break into organizations and uncover risky vulnerabilities that criminal attackers may use for personal gain.

X-Force Red offers offensive security services which include penetration testing, vulnerability management services, red teaming, code reviews, static analysis and vulnerability assessments.

Their goal is to help security leaders identify and remediate security flaws, covering their entire digital and physical ecosystem.

– 200+ people globally

– Decades of experience – personal and professional

– Researchers, developers, engineers, programmers, business leaders, thought leaders

– Testified before Congress, present at top conferences, built renown hacker communities

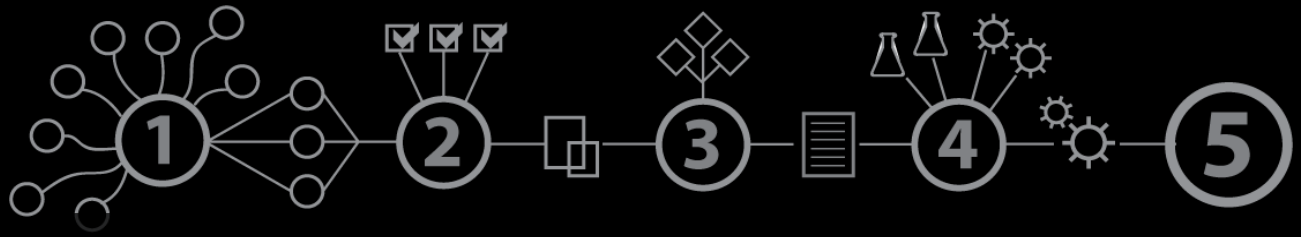
X-Force Red

IoT Security Challenges

Common Security Challenges

- Legacy, unpatched hardware
- Product functionality information left on production device
- Development tooling / logging left on production device
- Default credentials
- Millions of end user devices, lack of visibility
- Engineers are not security experts
  - Reliance on manufacturers to release patches
  - Assumption of security of 3rd party components
  - Complexity of inter-component communications

Testing for IoT – When / Why / How



Define scope and objectives

Data collection and device reconnaissance

Dynamic and static testing for App/OS/HW

Reverse engineering and threat modeling

Report and follow up

- Costs and risks increase throughout production life cycle
  - Better to eliminate issues in initial design
  - Remediation requires push out to installed user base
    - May not be possible if certified device

Meanwhile, there are so many components.

Circuit Board

Interfaces

Processor
Flash
& Logic

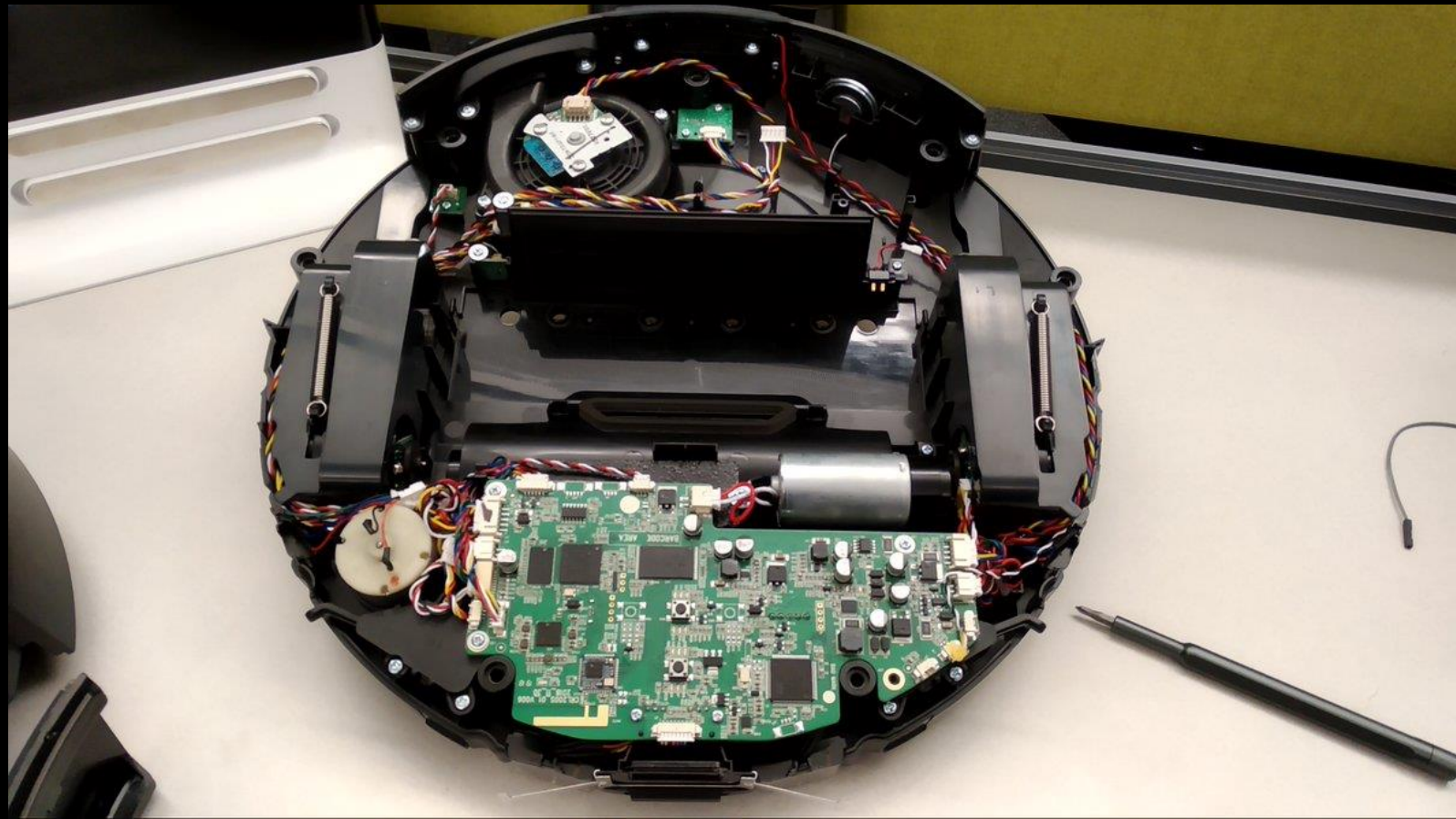
Anti Tamper

WiFi /
Zigbee /
BLE

Flash & RAM

Sensors

Buttons



All can be targeted…

- Architecture and Design
- Electronics (Schematic and PCB)
- Storage Systems
- Cryptography
- Anti-counterfeiting
- Communications (WIFI, Bluetooth, ZigBee etc)
- Software (implementation and SDLC)
- Firmware
- API
- Cloud services
- Build Standards
- Supply Chain

Case Study - Hardware penetration test on a high end IoT device

Objective: Gain access to file system to retrieve hash of root user.

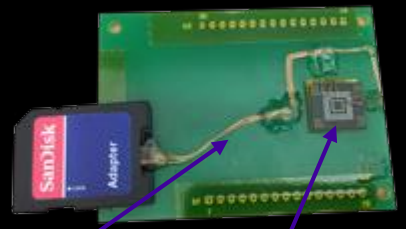
Description: Stringent software security and access controls prevented typical penetration test “privileged escalation” attacks access to file system.

Solution: Rather than continuing a software and network services attack, a fine pitch rework station was used to remove the BGA (ball grid array) 32GB Flash from the IoT device.

Most flash can simulate an SD card interface. As such, a little work under the Microscope resulted in only 7 wires being connected from the flash to a standard SD card adaptor.

This was connected to a standard laptop and the attacker gained unrestricted access to the file system and the desired root user password hash.

Note: Software safeguards were not sufficient.



Only 7 Wires Required

BGA Flash Chip

Case Study - Hardware penetration test on secure phone

Obscurity through to full compromise

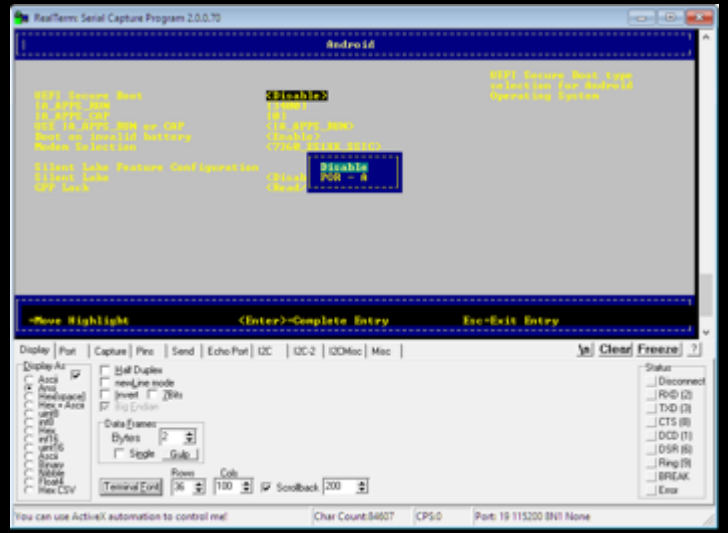
Objective: Full compromise of device.

Description: Operating system proved to be secure and attempts to compromise boot process failed due to robust secure boot controls.

Solution: Examination of the device PCB (circuit board) itself showed a number of unloaded 0402 components. This is often an indicator that “option resistors” have been included in the design. Option resistors are options set through the loading or not loading of certain components on a circuit. Often this includes enabling debugging mode by loading a part to allow developers more access.

Each of the “option resistor” pads were probed during a reboot of the device. The result was as expected and one of the unloaded components was indeed an option resistor that enabled a debug mode of operation.

This debug mode option resistor was checked by the BIOS during very early stages of a reboot, if enabled, it would allow access to a BIOS screen.



With access to the BIOS screen the attacker disabled secure boot and the previous attacks to compromise the boot process were then trivial to apply.

Note: A tiny hidden solder pad was enough to allow a thorough attacker to fully compromise a “secure” mobile phone.

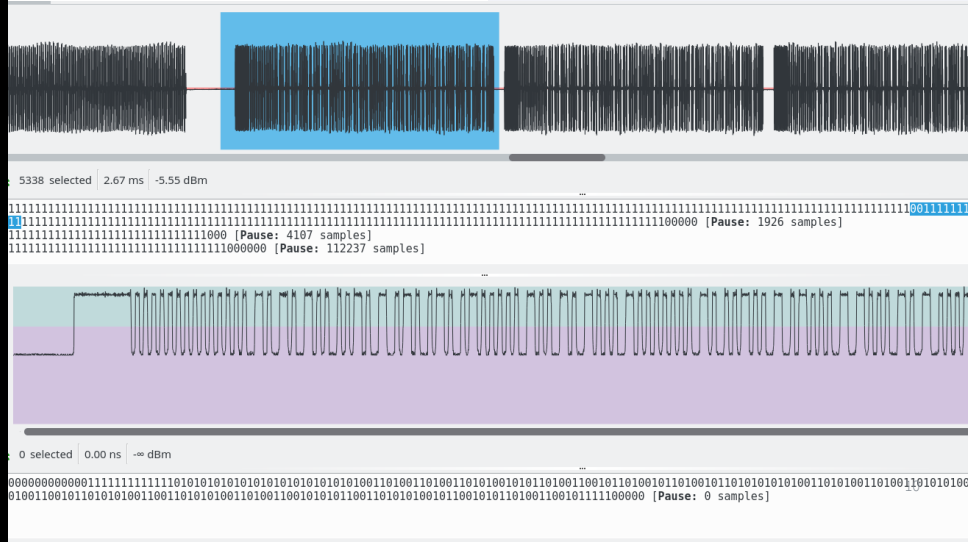
Case Study – Capture & Replay of proprietary encrypted RF signal

Objective: Demodulate, decode and decrypt to enable spoofing.

Description: Unknown RF signal controlled ”push-button” functions on an IoT access control device - Open/Close/Etc.

Solution: Capture signal with SDR. Compare repeated signals to work out packet structure/CRC etc. Identify encrypted payload and apply cryptographic attacks. Re-encode and transmit with COTS SDR.

YardStick One SDR



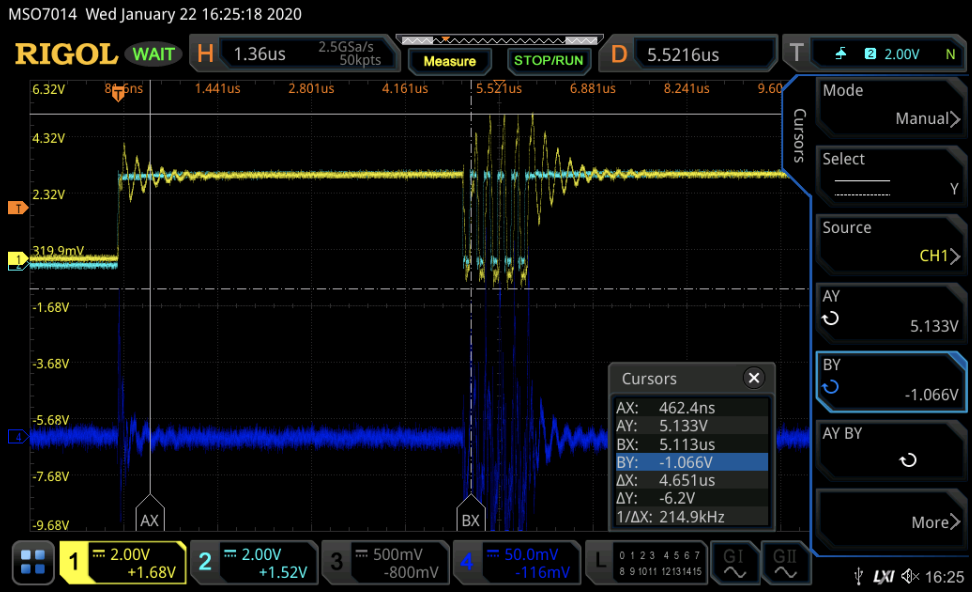
‘Raiden’ Glitching Tool

Hardware

- In-house developed open-source glitching tool – ‘Raiden’
- Based on off-the-shelf FPGA dev board
- Extremely accurate and high speed – 2.5 - 10ns resolution (hardware dependant).
- Selectable glitch voltage levels
- External trigger module
- Device RESET/Power Cycle
- Python API (via serial UART)
- Low Cost
- Simple to use

X-Force Red has developed its own glitching tool for fault injection attacks. Designed to use off the shelf technologies that mirror those that would be available in the real world, this allows us to realistically simulate attacks likely to be used against our clients’ devices.

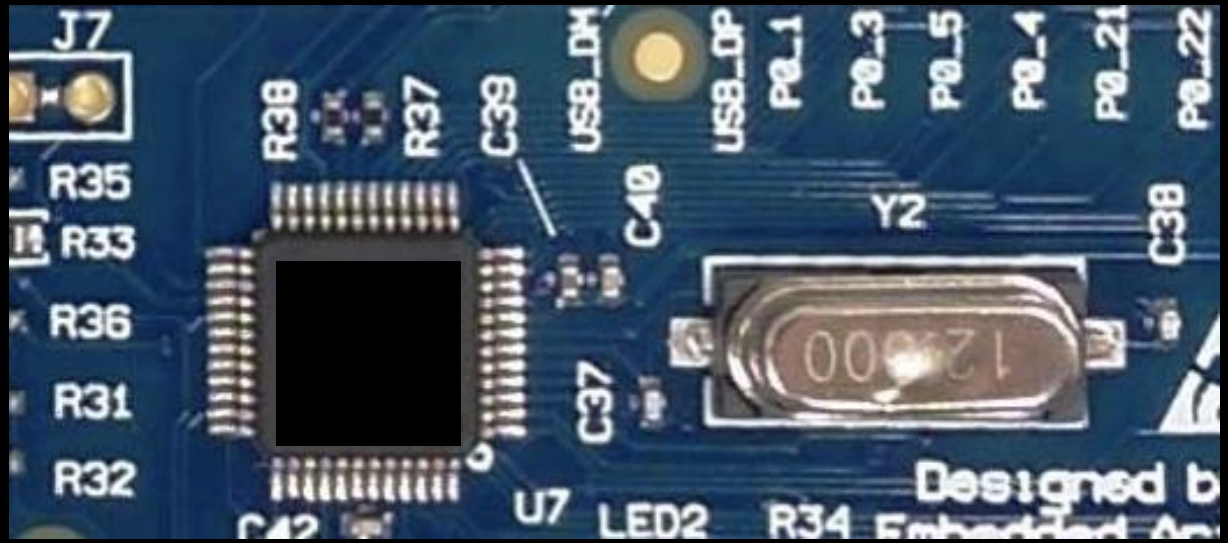
Glitching/fault injection is a very powerful technique for bypassing security checks and causing unexpected behavior in microcontrollers. By carefully controlling the timing and intensity of the faults being induced, an attacker can potentially cause the system to behave in a way that gives them advantage.



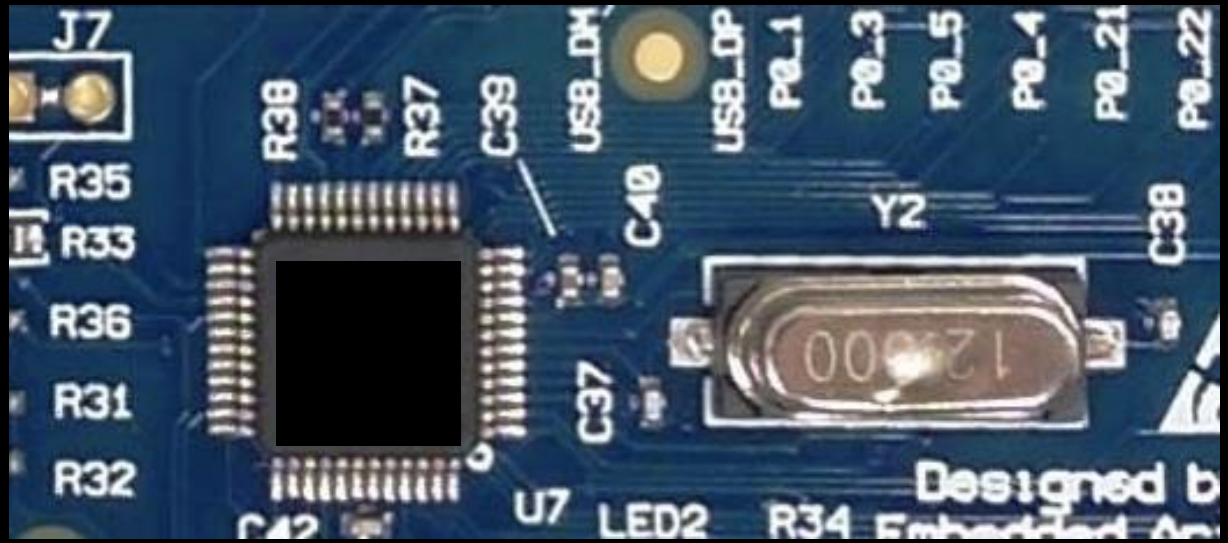
R & D – power analysis and glitching

- Provide some kind of feedback that chip is still alive
- Avoid destroying your only sample
- Confirm desired effect is even possible in a ‘safe’ environment
  - Enable bootloader
  - Enable SWD
  - Bypass read protection
  - Bypass code signing

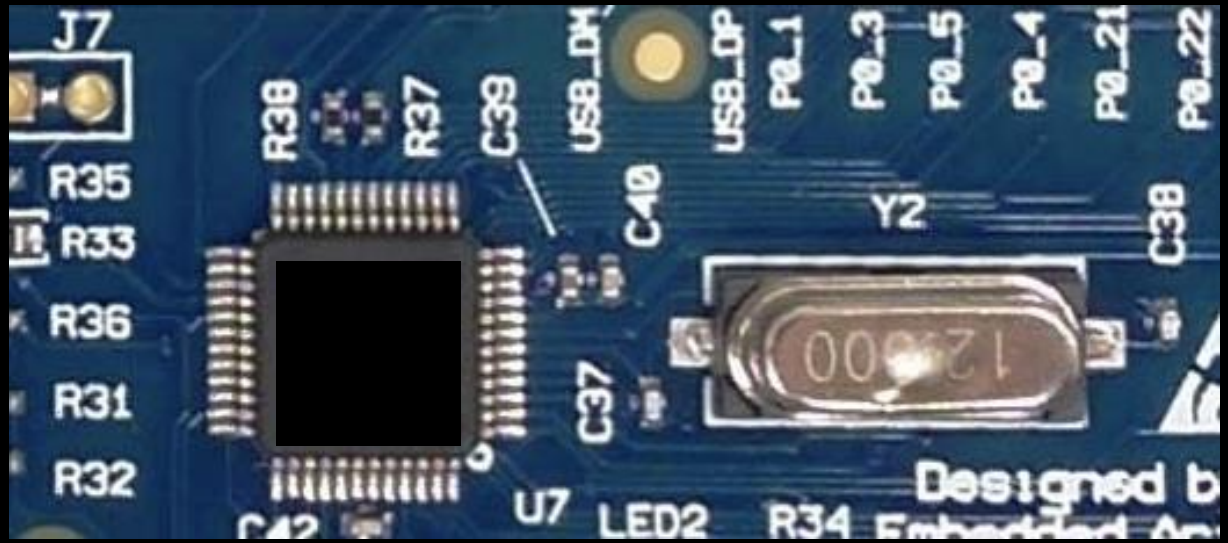
A chip doing nothing, no power…



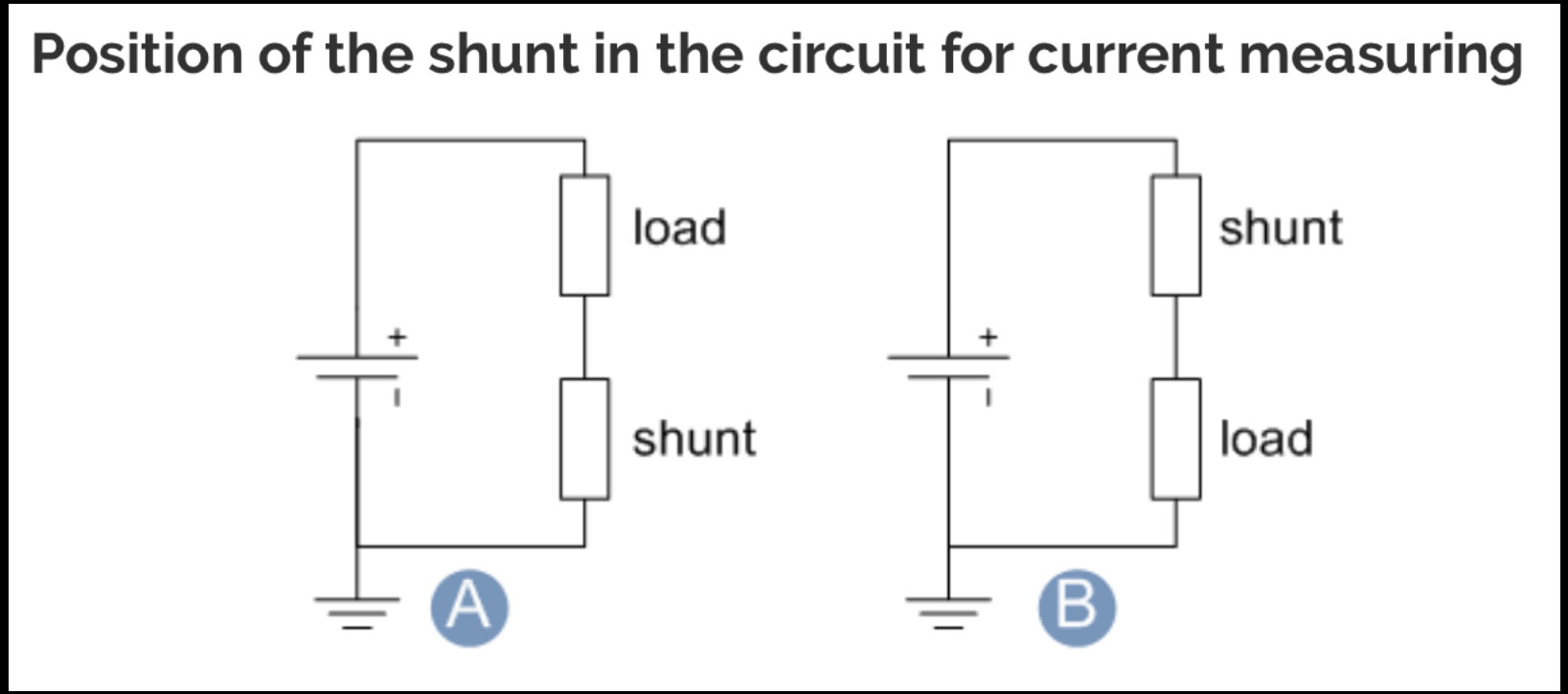
A chip powered up, running user code…



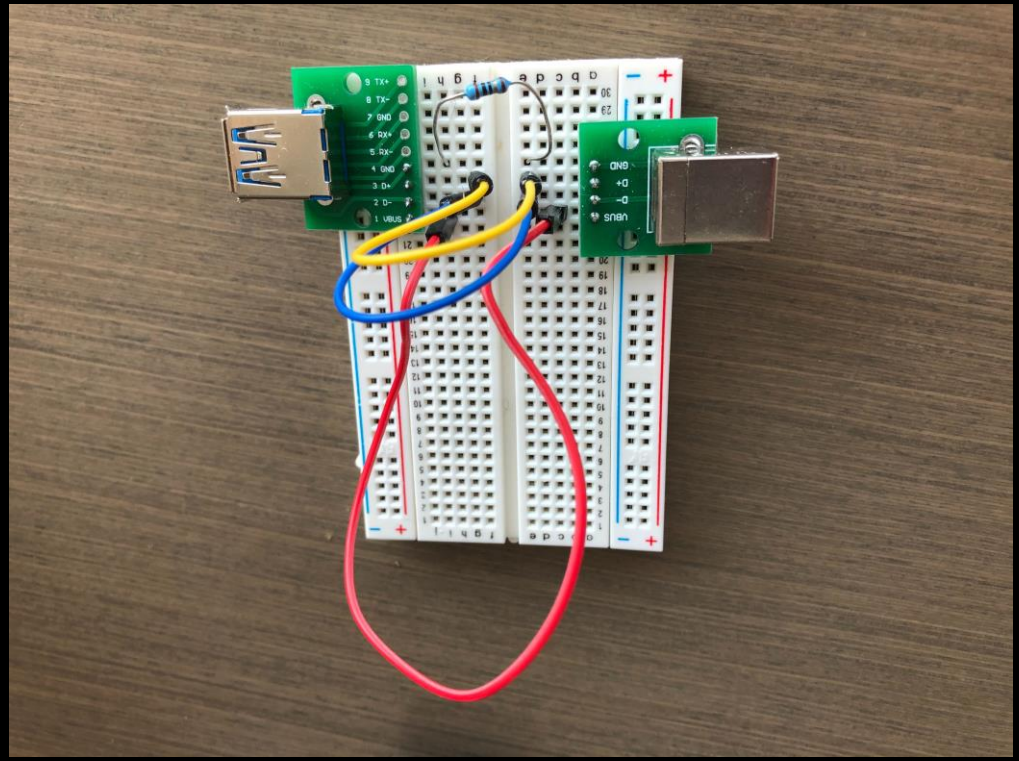
A chip powered up, running bootloader…



The magic shunt resistor!



The magic shunt resistor!



DPA – Differential Power Analysis

- Capture multiple runs
  - Compare power usage
    - Reveal bits in crypto operations
  - Compare timing
    - Reveal how far an operation got – e.g. password check

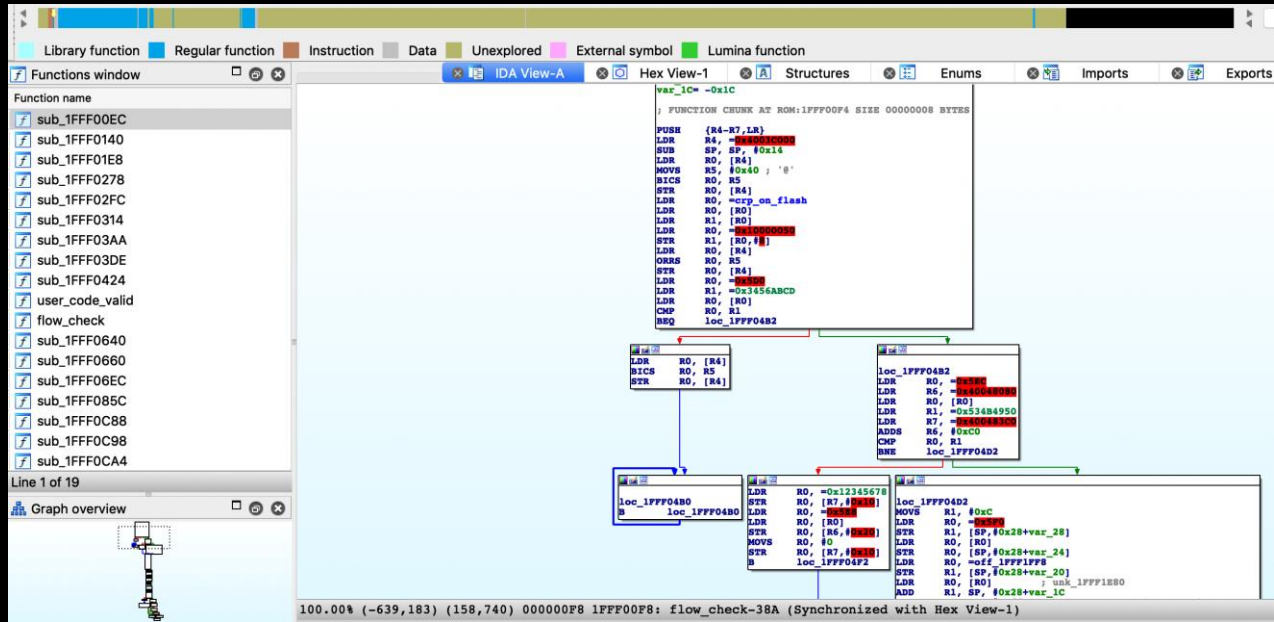
When to glitch?

When to glitch? A common approach

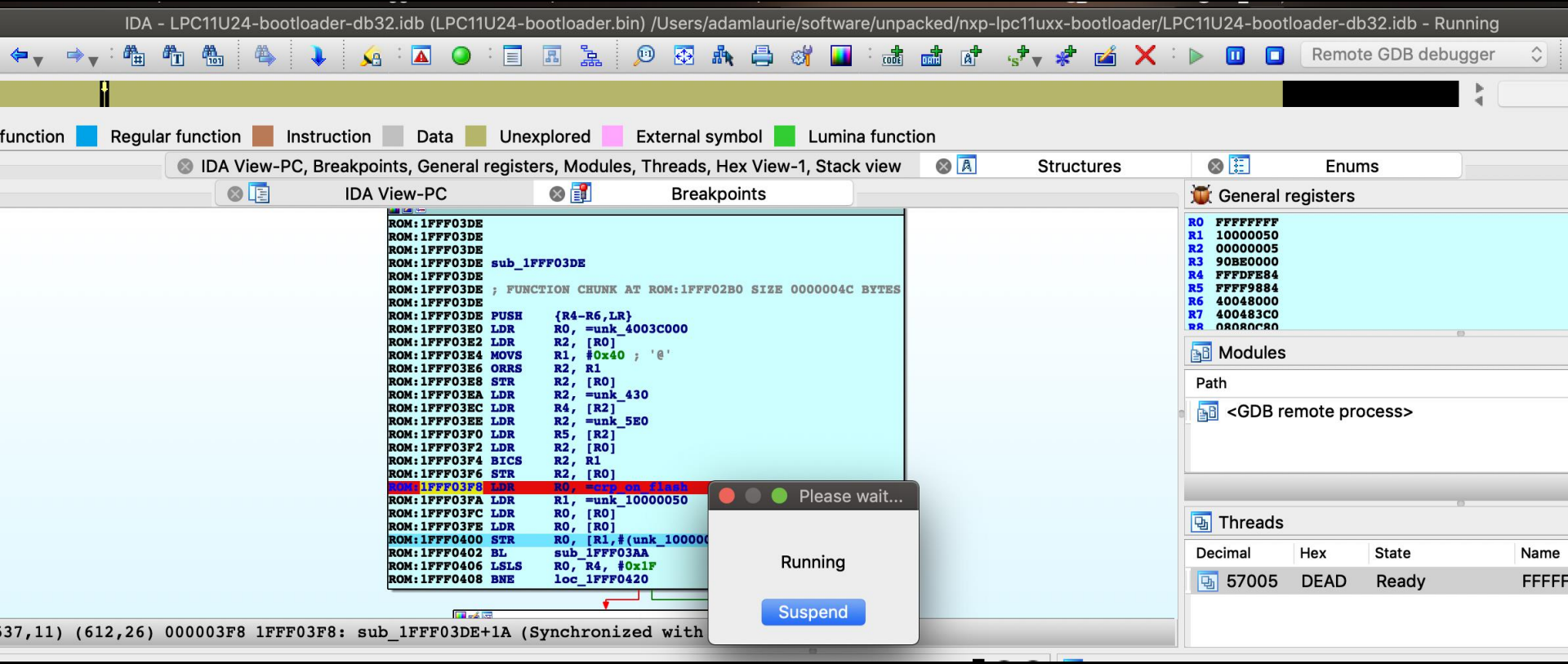
- Capture multiple runs with persistence
  - Switch modes
    - Bootloader vs User code
  - Glitch at point of divergence

- Reconnaissance

  - Extract bootloader
    - SWD – read out ROM
    - Reverse with Ghidra / IDA / Binary Ninja



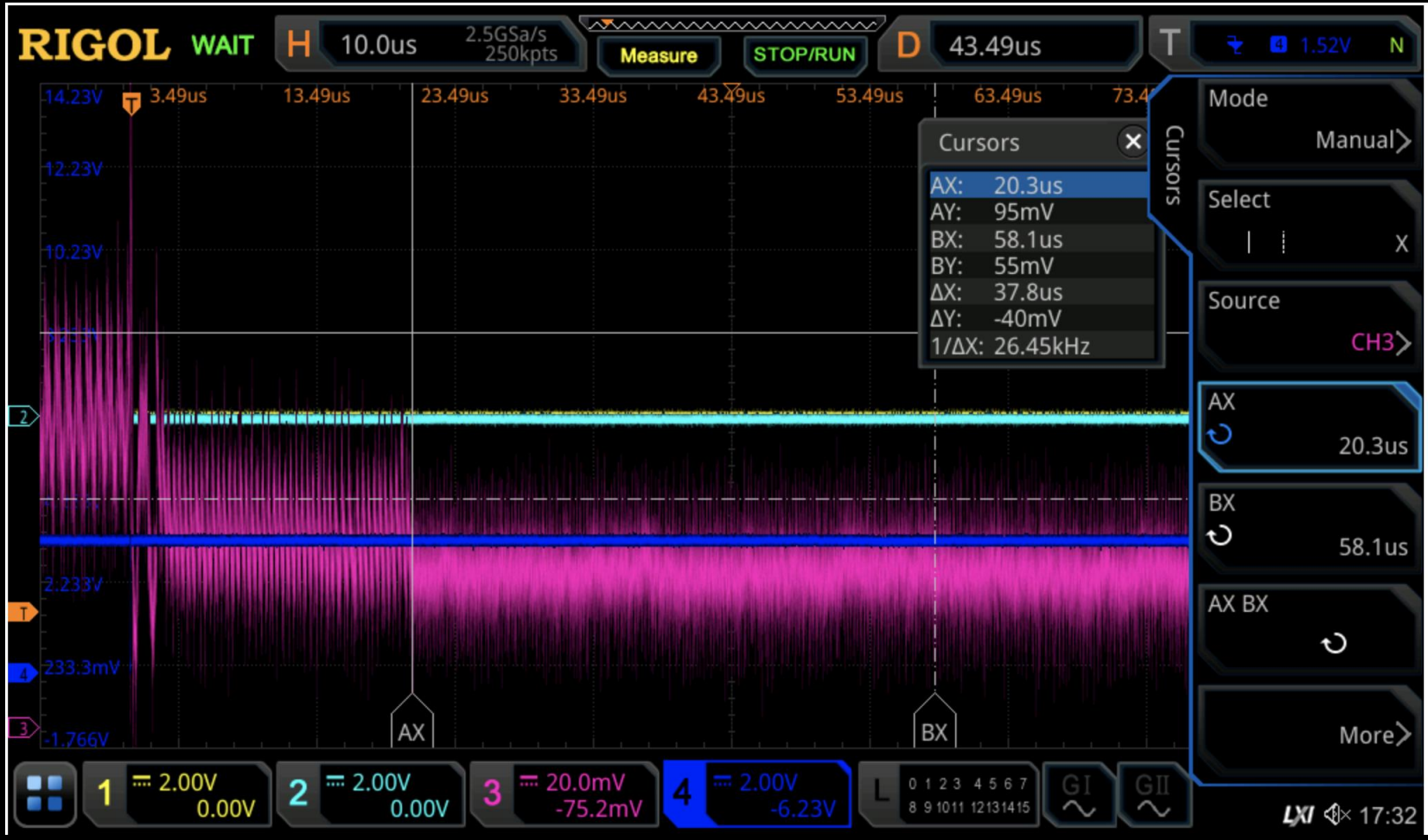
When to glitch? Add precision: Breakpoint



Breakpoint


RIGOL
WAIT
H 10.0us 2.5GSa/s 250kpts
Measure
STOP/RUN
D 43.49us
T 1.52V N
14.23V
3.49us
13.49us
23.49us
33.49us
43.49us
53.49us
63.49us
12.23V
4.233V
2.233V
233.3mV
-1.766V
AX
BX
Cursors
AX: 20.3us
AY: 95mV
BX: 58.1us
BY: 55mV
ΔX: 37.8us
ΔY: -40mV
1/ΔX: 26.45kHz
Cursors
Mode
Manual
Select
X
Source
CH3
AX
20.3us
BX
58.1us
AX BX
More
1 2.00V 0.00V
2 2.00V 0.00V
3 20.0mV -75.2mV
4 2.00V -6.23V
17:31

Breakpoint


RIGOL
WAIT
H 10.0us 2.5GSa/s 250kpts
Measure
STOP/RUN
D 43.49us
T 1.52V N
14.23V
12.23V
10.23V
2.233V
233.3mV
-1.766V
3.49us
13.49us
23.49us
33.49us
43.49us
53.49us
63.49us
AX
BX
Cursors
AX: 20.3us
AY: 95mV
BX: 58.1us
BY: 55mV
ΔX: 37.8us
ΔY: -40mV
1/ΔX: 26.45kHz
Mode
Manual
Select
X
Source
CH3
AX
20.3us
BX
58.1us
AX BX
More
1 2.00V 0.00V
2 2.00V 0.00V
3 20.0mV -75.2mV
4 2.00V -6.23V
17:32

Want to learn more?

– Visit the X-Force Red homepage: https://www.ibm.com/security/services/offensive-security-services

– Read the whitepaper: https://www.ibm.com/downloads/cas/MY6L2O89

– Watch the video: https://www.ibm.com/security/services/iot-testing

Thank you

Follow us on:

ibm.com/xforcered

@xforcered

securityintelligence.com

ibm.com/security/community

xforce.ibmcloud.com

youtube.com/ibmsecurity


IBM