SIGPwny

Embedded FA2025 • 2025-09-08

Intro to Embedded Team

Minh and Jake

Minh Duong

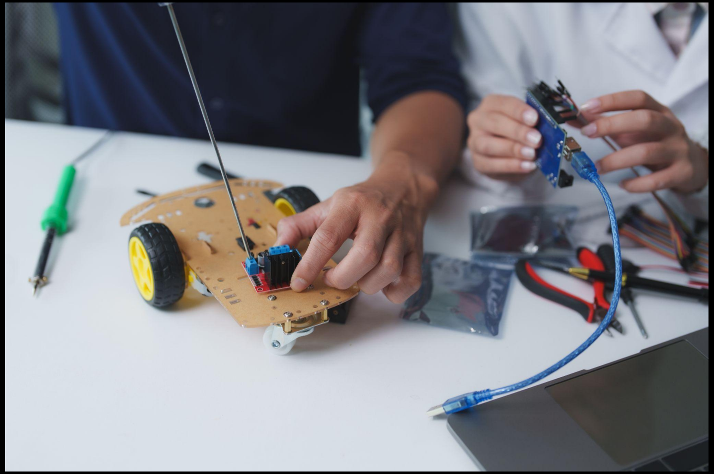
- Previous SIGPwny President
- BS-MCS program (last semester)
- Involved in SIGPwny's eCTF team since 2023
- Fun fact: I rickrolled my high school by hacking the projectors

Jake Mayer

- SIGPwny Admin & Embedded co-lead for '24 & '25
- CS, Math major
- Fun fact: I was rejected from UIUC

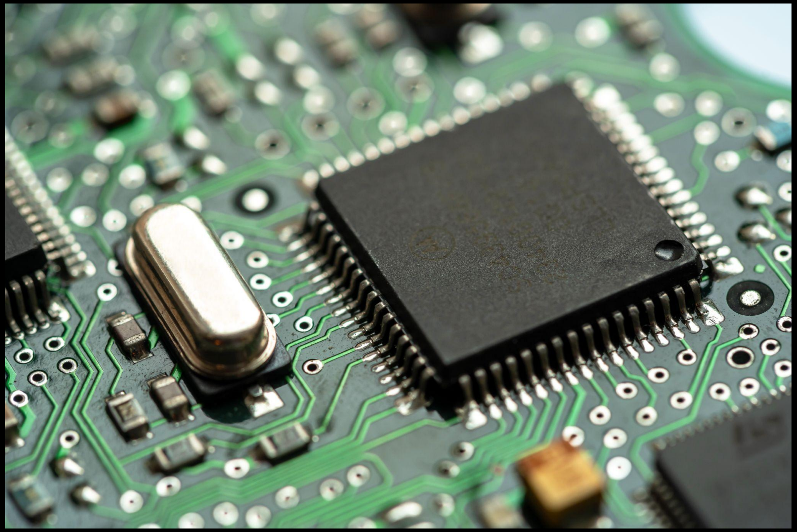
What is an Embedded System?

- Wikipedia: An **embedded system** is a specialized computer system—a combination of a computer processor, computer memory, and input/output peripheral devices—that has a dedicated function within a larger mechanical or electronic system.
- Examples:
  - Voting machines
  - Cars (entertainment system, vehicle controls)
  - Traffic lights
  - Security cameras
  - Insulin pump
  - Missiles
  - Any more?



What is an Embedded System?

- Common characteristics:
  - Part of a larger device
  - Interfaces with sensors
  - Actuates mechanisms
  - Often low-power, resource constrained (microcontrollers)



What is Embedded Team?

- We focus on security of embedded systems and IoT devices
- We are especially interested in the intersection of **hardware** and **software**

- We play competitions related to embedded security
- Sometimes, we do research or work on large development projects

Siebel School of Computing and Data Science
Siebel School of Computing and Data Science
GRAINGER ENGINEERING

Why join Embedded Team?

1. It’s one of the best ways to learn about programming low level systems.
    a. Embedded systems are tinier and usually much less complex than general computing systems, making it easier to build something from scratch

2. Learn how to apply the “adversarial mindset” in a completely new domain/field

3. Help us beat **Purdue** and **CMU**

Our Lab (Before)

Our Lab (After)

Planned Meetings

- Embedded 101: Fundamentals
- Embedded 102: Microcontroller Programming
- Embedded 103: Breadboarding and Hardware

- Secure Protocol Design
- Software Security
- Attacking “Secure” Protocols

- Side-Channel Attacks
- Firmware Security
- Fault Injection and Voltage Glitching
- Embedded Rust

CSAW Embedded Security Challenge

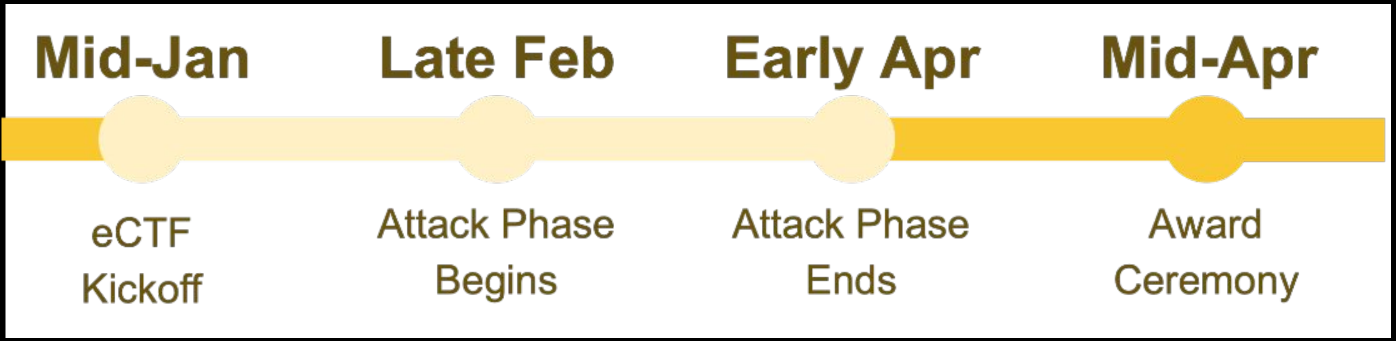
TODO: Talk about at the end

MITRE eCTF 2026

Starts in January 2026

What is eCTF?

- **100+** high school and university teams participating
- Build a secure embedded device fitting the organizer's (MITRE) functional and security requirements
- Organizers verify functional requirements
- Teams earn points by breaking security requirements of competitors



Mid-Jan — eCTF Kickoff

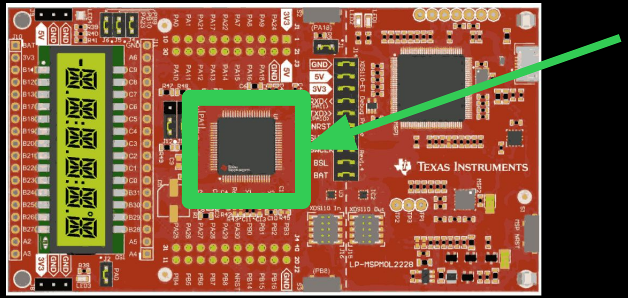
Late Feb — Attack Phase Begins

Early Apr — Attack Phase Ends

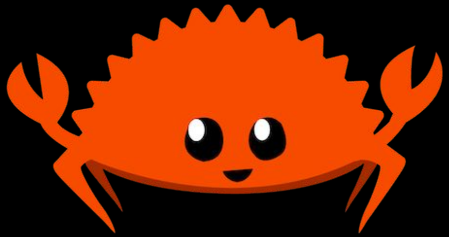
Mid-Apr — Award Ceremony

eCTF Design Phase

- **Goal**: convert organizer's requirements into a secure design
- First, specify protocol and cryptographic methods
  - Defense in depth: redundant security measures hedge against vulnerabilities
  - Carefully audit abstract design against requirements
- Next, implement the design
  - Many teams use **C**, we choose **Rust** for its safety
  - The concrete design must be secure in software and **hardware**
  - Other teams will attack your design and have hardware access

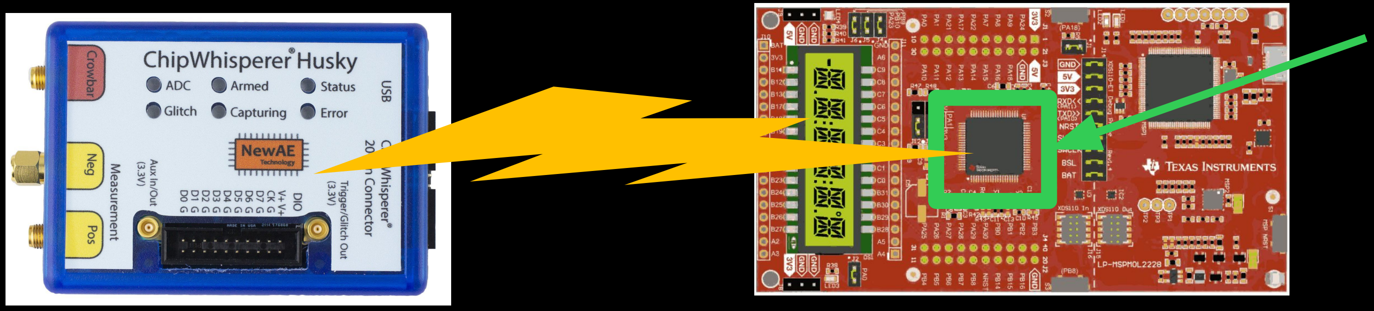


Your code runs here



eCTF Attack Phase

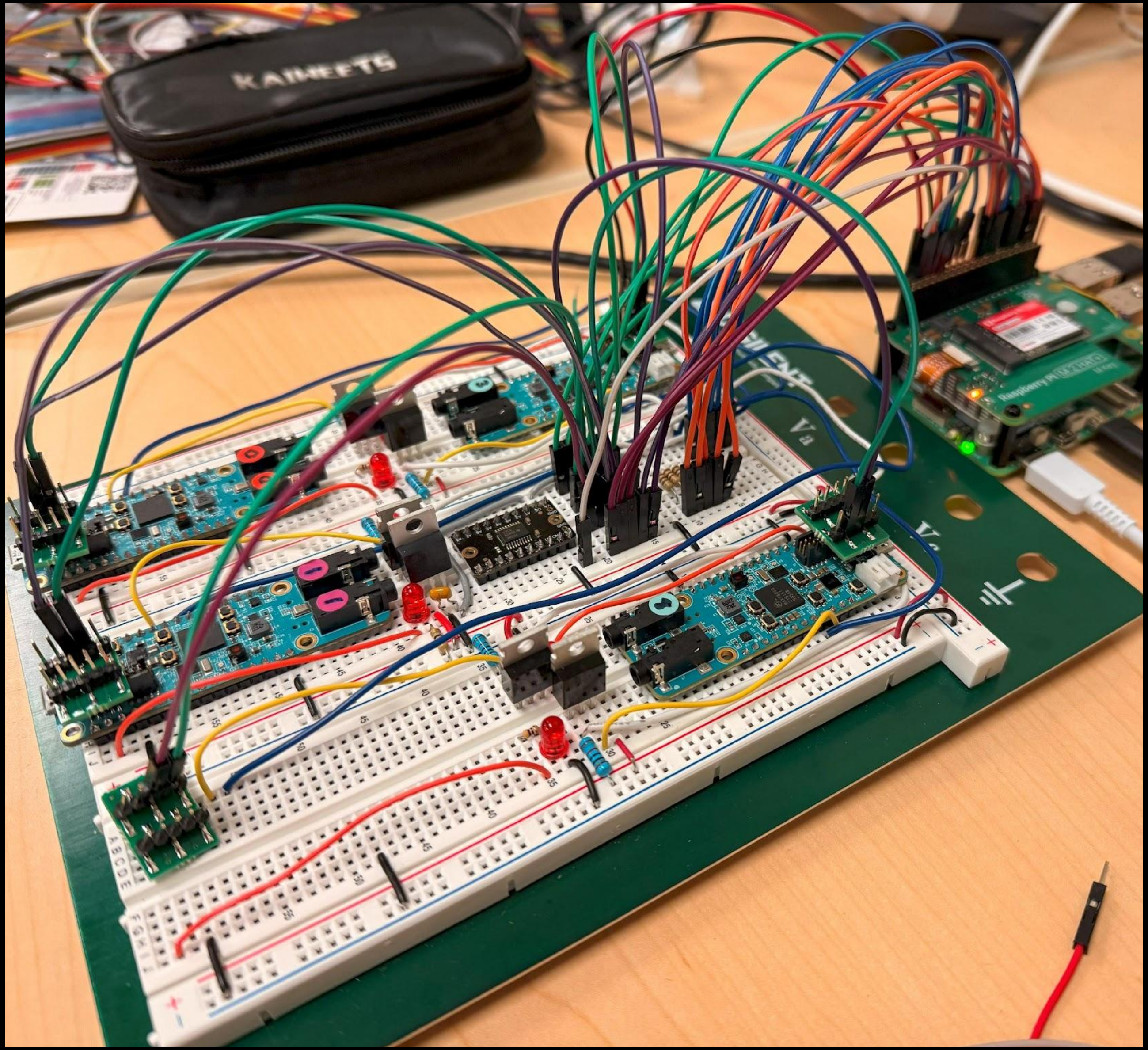
- **Goal**: break the security requirements
  - Points are awarded roughly per each requirement compromised
  - A single vulnerability may compromise many or all requirements
- Some creativity is required: a lot is in scope
- Typical C unsafety (buffer overflow, integer overflow)
- Bad crypto (padding oracle, ECB use, CBC manipulation)
- Timing side-channel (`memcmp`)
- Hardware attacks (fault injection, power analysis)

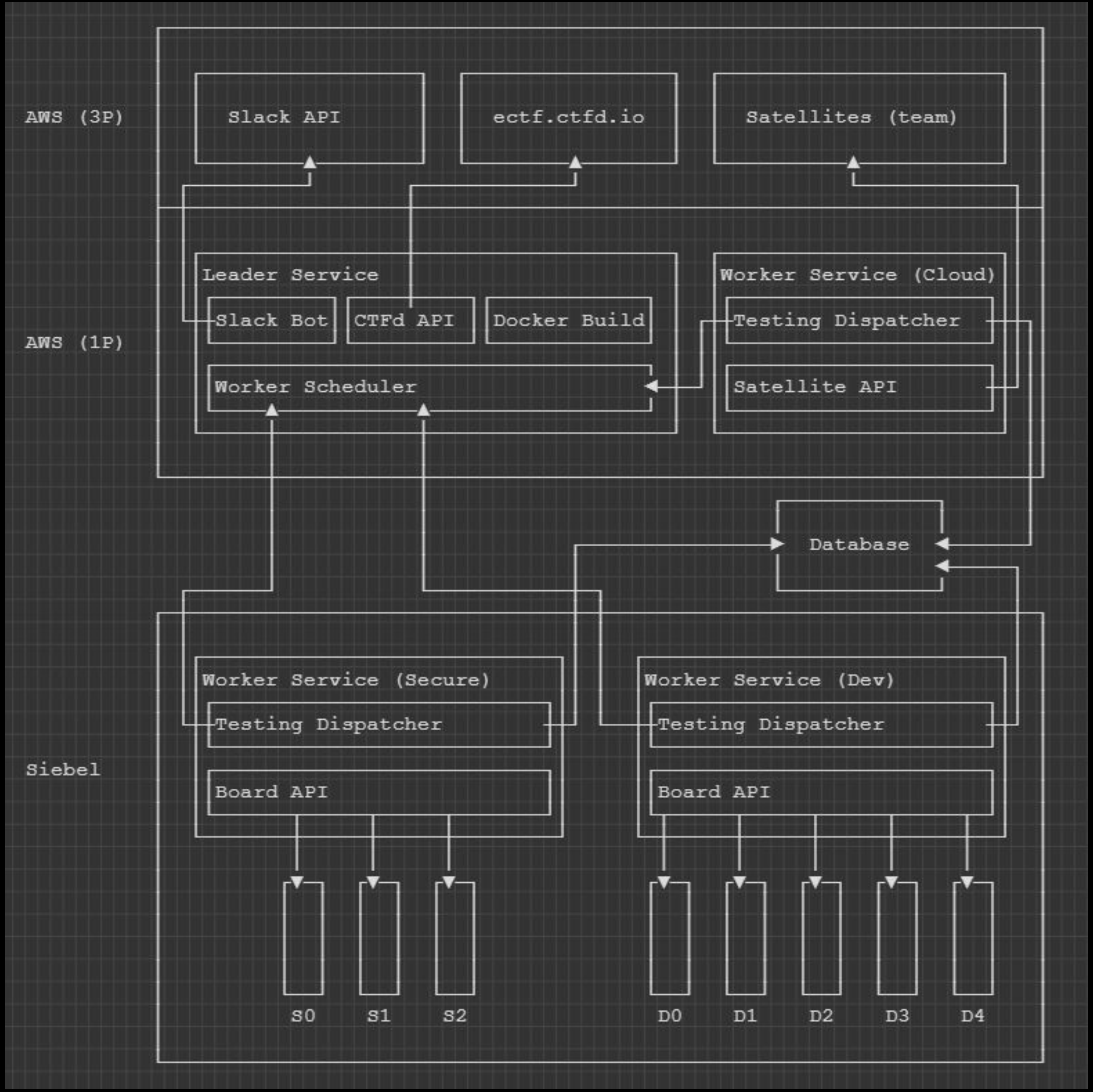


Other teams' code runs here too

Previous Competitions

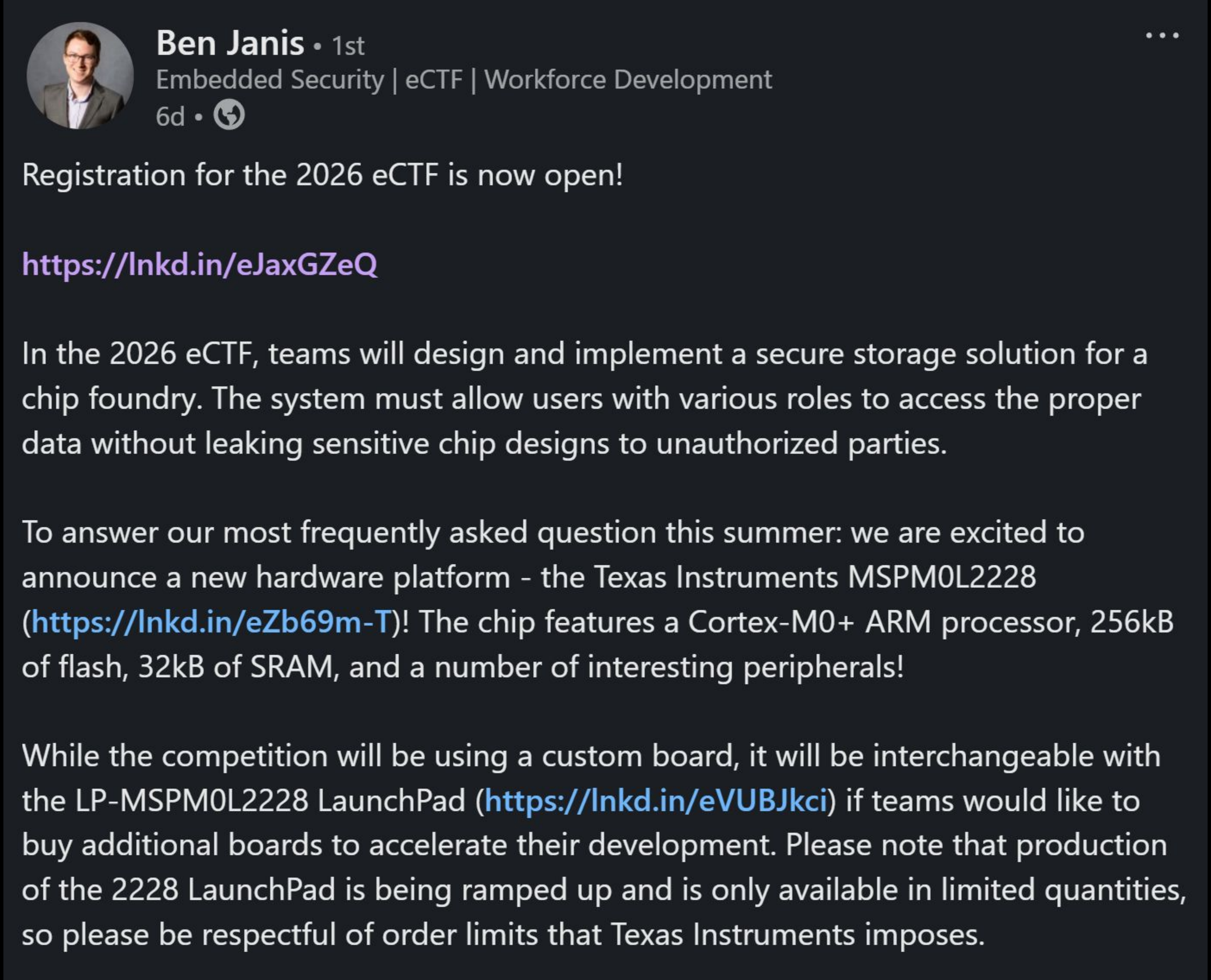
- **2023**: Car lock system
    - Key fob securely unlocks the car
    - Car DLC: upgrades enabled by manufacturer-provided package
- **2024**: Medical device
    - Modular system that authenticates components
    - Supports component replacement
    - Secure communication between parts
- **2025**: Satellite TV
    - Several channels which broadcast frames publicly
    - Boxes may decode frames only for a channel and time which is encoded in a subscription package


KAIWEETS


AWS (3P)
Slack API
ectf.ctfd.io
Satellites (team)
AWS (1P)
Leader Service
Slack Bot
CTFd API
Docker Build
Worker Scheduler
Worker Service (Cloud)
Testing Dispatcher
Satellite API
Database
Siebel
Worker Service (Secure)
Testing Dispatcher
Board API
Worker Service (Dev)
Testing Dispatcher
Board API
S0
S1
S2
D0
D1
D2
D3
D4

2026 Competition

“In the 2026 eCTF, teams will design and implement a secure storage solution for a chip foundry. The system must allow users with various roles to access the proper data without leaking sensitive chip designs to unauthorized parties.”



Ben Janis • 1st

Embedded Security | eCTF | Workforce Development

6d •

Registration for the 2026 eCTF is now open!

https://lnkd.in/eJaxGZeQ

In the 2026 eCTF, teams will design and implement a secure storage solution for a chip foundry. The system must allow users with various roles to access the proper data without leaking sensitive chip designs to unauthorized parties.

To answer our most frequently asked question this summer: we are excited to announce a new hardware platform - the Texas Instruments MSPM0L2228 (https://lnkd.in/eZb69m-T)! The chip features a Cortex-M0+ ARM processor, 256kB of flash, 32kB of SRAM, and a number of interesting peripherals!

While the competition will be using a custom board, it will be interchangeable with the LP-MSPM0L2228 LaunchPad (https://lnkd.in/eVUBJkci) if teams would like to buy additional boards to accelerate their development. Please note that production of the 2228 LaunchPad is being ramped up and is only available in limited quantities, so please be respectful of order limits that Texas Instruments imposes.

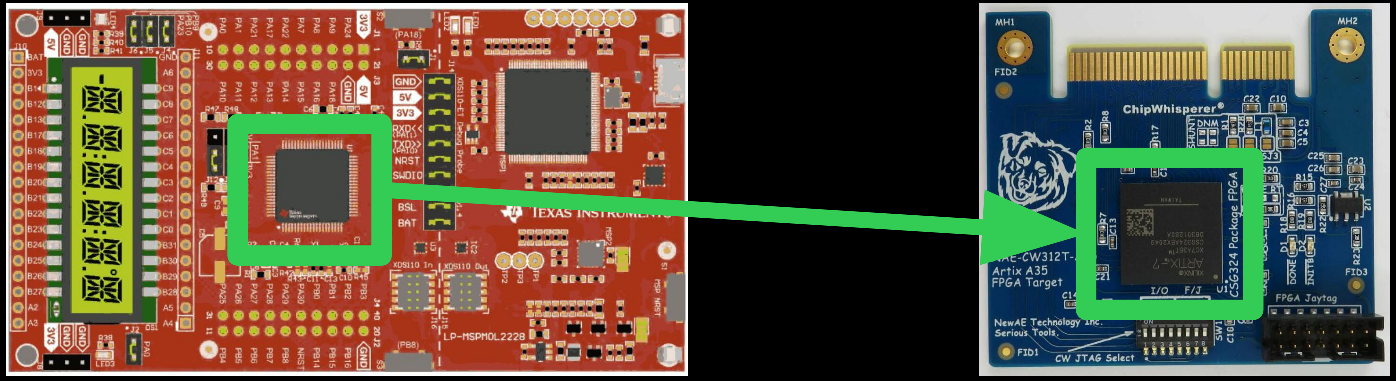
MSP	M0	L	2228
Mixed Signal Processor	Arm Cortex M0+	Low Power	Part Number

Project: MSPM0L2228 Development

- Because the chip is new (for eCTF and also to the public generally), there may be limited support for development on embedded Rust
- There is active development by the Embassy-rs team on a hardware abstraction layer for MSPM0 MCUs
  - https://github.com/mspm0-rs
  - We have yet to confirm if it is ready to use
- Alternatively: build our own!
- We’ve done it before for ADI’s MAX78000 (used in 2023, 2024)
  - https://github.com/sigpwny/max7800x-hal

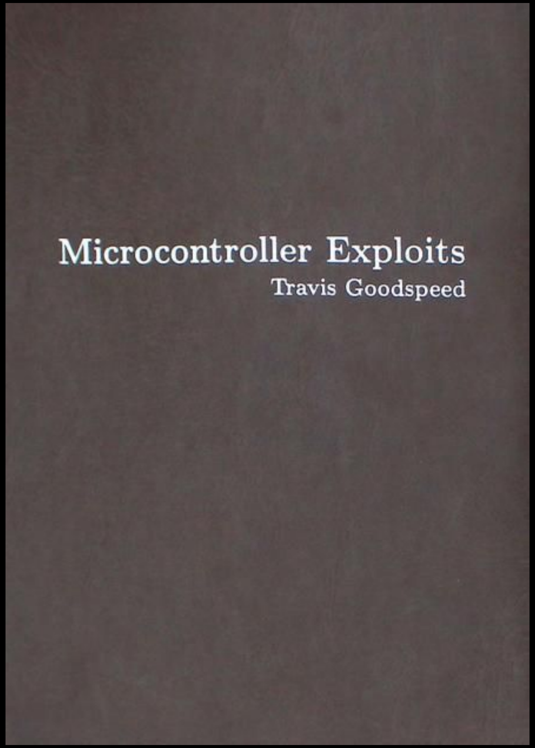
Project: MSPM0L2228 Glitching

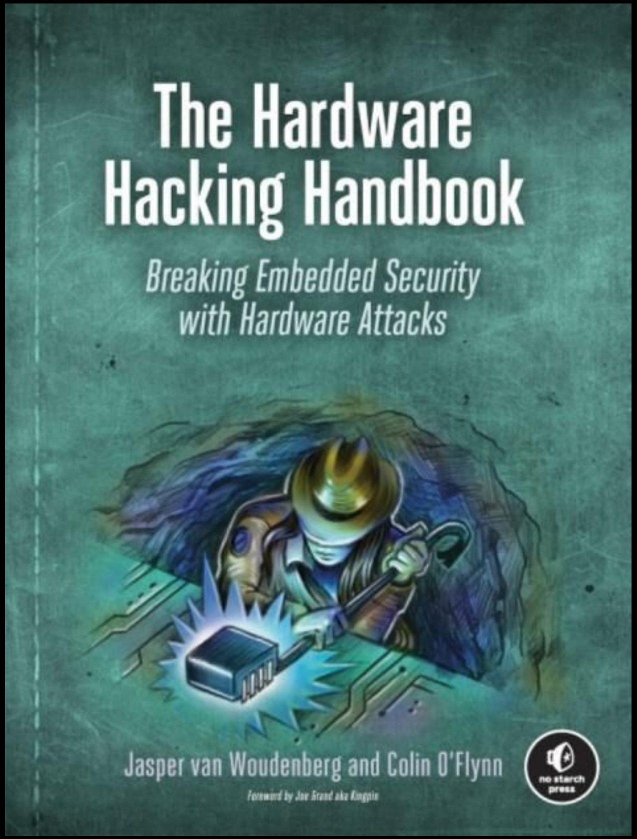
- **Goal**: Execute a glitching attack against the MSPM0L2228 and document glitch parameters.
- Design a custom CW312-shaped board to attack the MSPM0L2228
- Perform glitching attacks against the custom board.
- Practice chip transfers to the CW312 board.
- Stretch goal: Perform side-channel analysis on common embedded C cryptography libraries (AES on wolfcrypt and/or mbedtls) using the custom board.



Cool Readings

- “Microcontroller Exploits” by Travis Goodspeed
- “The Hardware Hacking Handbook” by Jasper van Woudenberg and Colin O'Flynn




The Hardware Hacking Handbook
Breaking Embedded Security with Hardware Attacks
Jasper van Woudenberg and Colin O'Flynn
Foreword by Joe Grand aka Kingpin
no starch press

Other Resources

- MITRE EMB3D framework: https://emb3d.mitre.org/
    - Threat modeling framework for embedded systems
    - Think of it as a “checklist” to see what areas you need to consider for security

Next Meetings

2025-09-15 • Next Monday

- Embedded 101: Fundamentals
- Learn about the components of an embedded system and why security matters!

Meeting content can be found at sigpwny.com/meetings.