2022 eCTF Kickoff

Jake Grycel

1/26/2022

Outline

1. Welcome
2. Competition Overview
3. Challenge Overview
4. Requirements
5. Attack Deployment
6. Flags

Outline

1. **Welcome**
2. Competition Overview
3. Challenge Overview
4. Requirements
5. Attack Deployment
6. Flags

Participating Schools

AMRITA VISHWA VIDYAPEETHAM

CAPITOL Technology University

Carnegie Mellon University

DAKOTA STATE UNIVERSITY

DACC DELAWARE AREA CAREER CENTER

FIU FLORIDA INTERNATIONAL UNIVERSITY

INDIANA TECH

Massachusetts Institute of Technology

MICHIGAN STATE UNIVERSITY

MORGAN STATE UNIVERSITY

Nonsuch High School for Girls

NORFOLK STATE UNIVERSITY

Northeastern

NOVA SOUTHEASTERN UNIVERSITY NSU Florida

The Ohio State University

SINGAPORE MANAGEMENT UNIVERSITY

TEXAS A&M UNIVERSITY

Tufts UNIVERSITY

University at Buffalo

UCIrvine

University of Colorado Boulder

UCONN UNIVERSITY OF CONNECTICUT

University of Dayton

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

UMass Amherst

University of New Hampshire

UT DALLAS The University of Texas at Dallas

UNIVERSITY OF WYOMING

VSU VIRGINIA STATE

VIRGINIA TECH

WPI

MITRE

Organizers

MITRE

MITRE is a not-for-profit organization that operates research and development centers sponsored by the federal government. MITRE works with industry and academia to apply science, technology, and systems engineering that enables the government and the private sector to make better decisions. Learn more at www.mitre.org

Follow us on Twitter @MITRECorp

RIVERSIDE RESEARCH

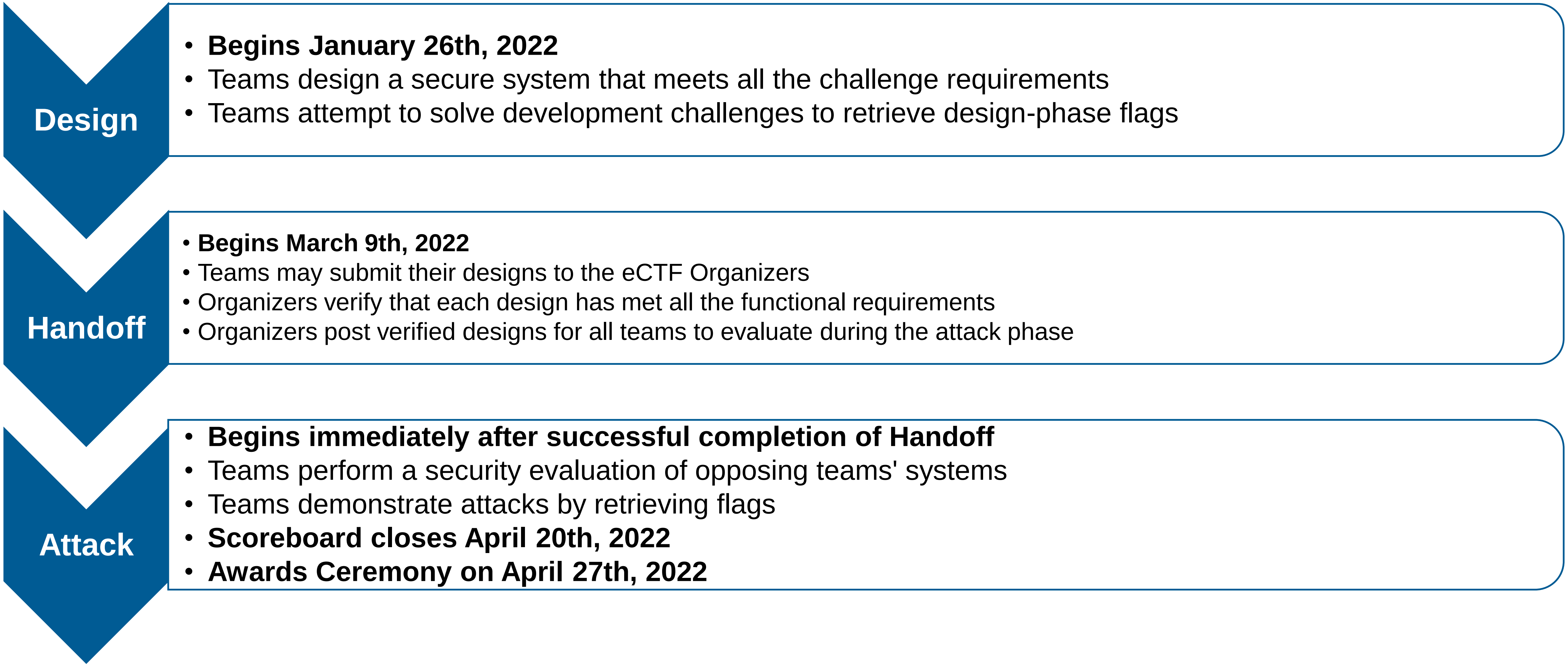
Riverside Research is a not-for-profit organization advancing scientific research in the interest of National Security. Through the company's Open Innovation Center (OIC), it invests in multi-disciplinary research and development and encourages collaboration to accelerate innovation and advance science. Research areas include: AI/ML, Trusted and Resilient Systems, Optics, Electromagnetics, Commercial ISR, and Collection Planning. Learn more at www.riversideresearch.org.

Follow us on Twitter @RiversideRsch

Outline

1. Welcome
2. **Competition Overview**
3. Challenge Overview
4. Requirements
5. Attack Deployment
6. Flags

Competition Overview



Design

- **Begins January 26th, 2022**
- Teams design a secure system that meets all the challenge requirements
- Teams attempt to solve development challenges to retrieve design-phase flags

Handoff

- **Begins March 9th, 2022**
- Teams may submit their designs to the eCTF Organizers
- Organizers verify that each design has met all the functional requirements
- Organizers post verified designs for all teams to evaluate during the attack phase

Attack

- **Begins immediately after successful completion of Handoff**
- Teams perform a security evaluation of opposing teams' systems
- Teams demonstrate attacks by retrieving flags
- **Scoreboard closes April 20th, 2022**
- **Awards Ceremony on April 27th, 2022**

New Features

- **Emulated and physical hardware**
- **Design Phase Points**
  - Side-Channel Analysis (SCA) Challenge
- **Automated Testing Service**
- **Hardware Trojans**

Prizes and Competition Qualification Requirements

- **This year the eCTF will award $5000 in prizes to the winning teams**
  - 1st Place: $2000
  - 2nd Place: $1000
  - 3rd Place: $500
  - Special Awards: $1500 (may be split among multiple teams)
- **Any student can compete in the eCTF, but to receive prize money you must meet certain eligibility requirements**
  - Check our website (ectf.mitre.org) for award eligibility terms
- **Several policies and processes have been put in place to ensure fairness**
  - All questions and requests for help are taken on a first-come-first-serve basis
  - Write-ups are anonymized before judging
  - To specifically address and mitigate any potential unfair advantage for participants that have interned at MITRE or Riverside Research:
    - Competition organizers are “firewalled” from current intern participants – no discussions allowed outside of official channels
    - The challenge requirements were changed significantly from the summer version that we run each year with interns

Outline

1. Welcome
2. Competition Overview
3. **Challenge Overview**
4. Requirements
5. Attack Deployment
6. Flags

The Scenario

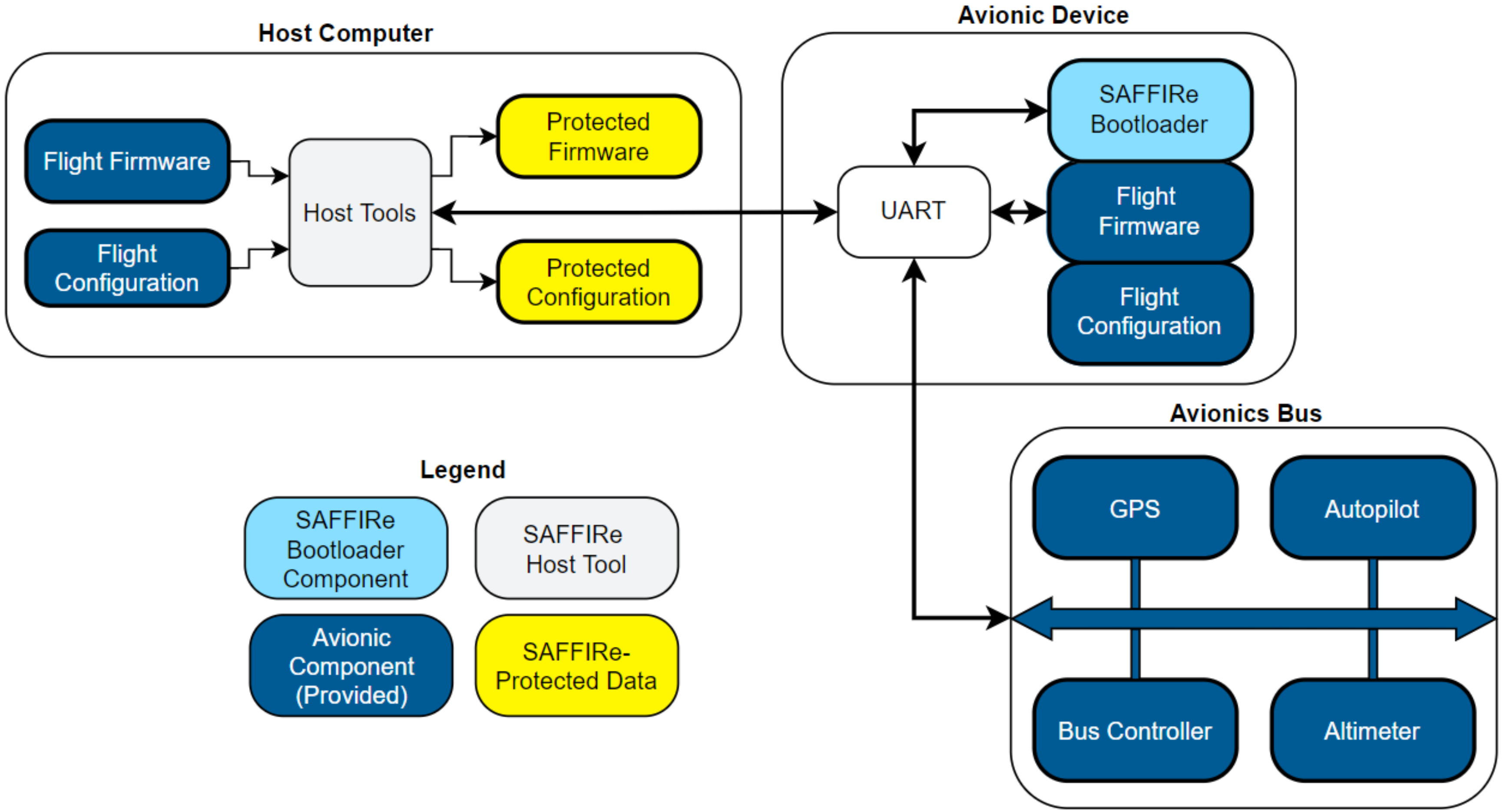
- **Your team is tasked with designing and implementing a secure firmware update system for an avionic device**

- **This system is called…**
  - …The Secure Avionics Flight Firmware Installation Routine, or SAFFIRe!

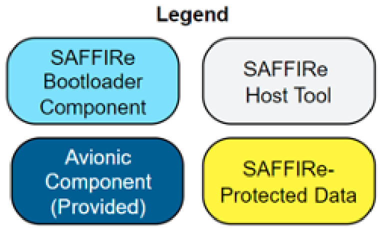
- **The Goal:**
  - Securely install new firmware and flight configurations on the device…
  - …in the face of physical attacks and…
  - …in the face of supply-chain security threats!

Avionic System

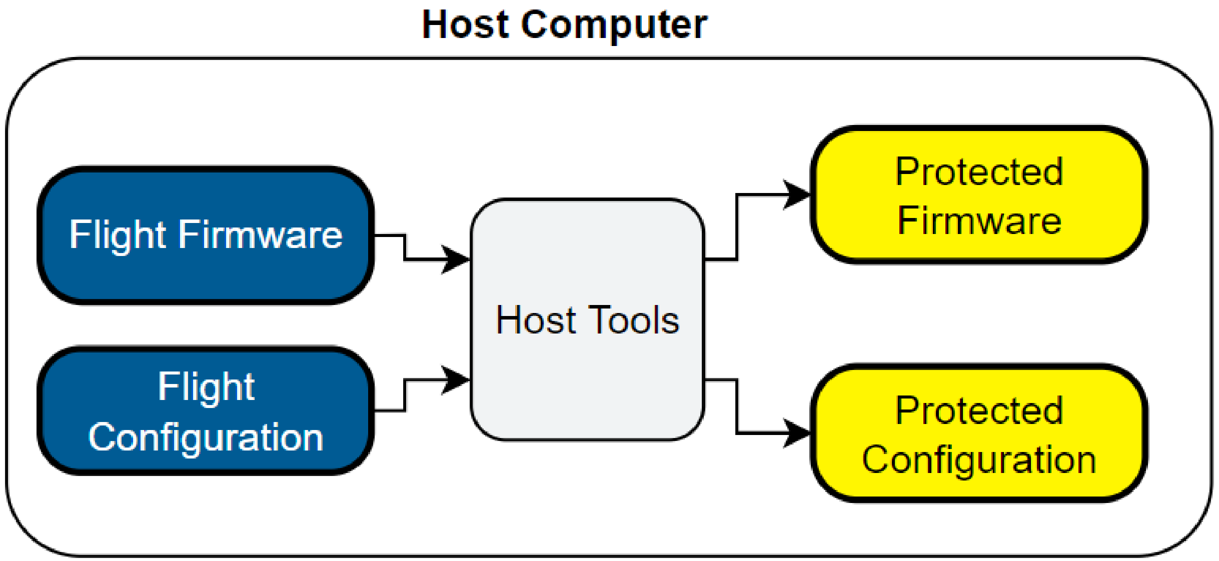
- **The avionic system has three main components**
  - Host Computer
  - Avionic Device
  - Avionics Bus



Host Computer



- **The host computer runs SAFFIRe host tools**
  - Protects avionic flight firmware and configurations
  - Loads firmware and configuration updates into the avionic device
  - Requests data back from the avionic device
- **CPU**
  - Runs a general-purpose host OS



Legend

SAFFIRe Bootloader Component

SAFFIRe Host Tool

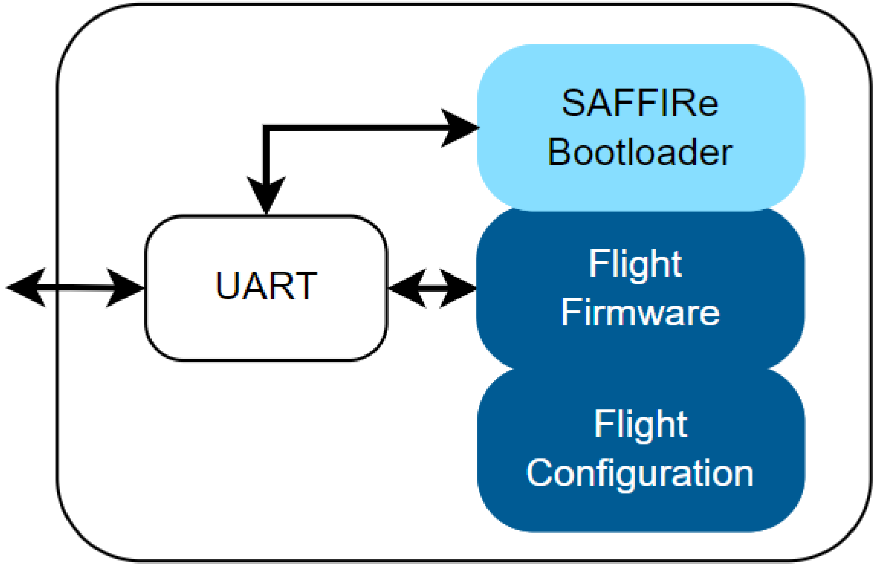
Avionic Component (Provided)

SAFFIRe-Protected Data

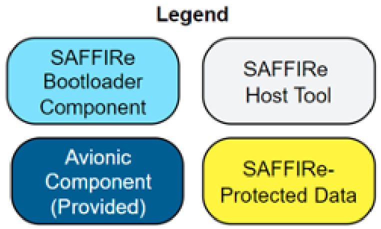
Avionic Device

- **The avionic device runs the SAFFIRe bootloader and avionic firmware**
  - The SAFFIRe bootloader is only responsible for installing flight firmware and configurations created by a secure host
  - The avionic firmware runs during aircraft flight and communicates important information over the avionics bus
- **CPU**
  - Embedded Arm Cortex-M4 microcontroller

Avionic Device



Aircraft



Legend

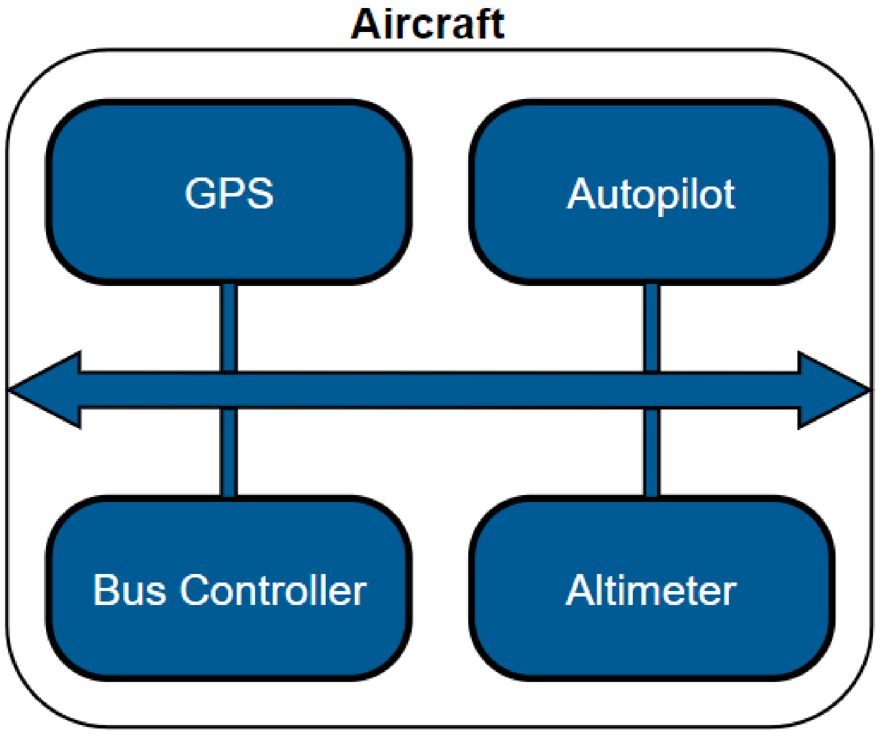
SAFFIRe Bootloader Component

SAFFIRe Host Tool

Avionic Component (Provided)

SAFFIRe-Protected Data

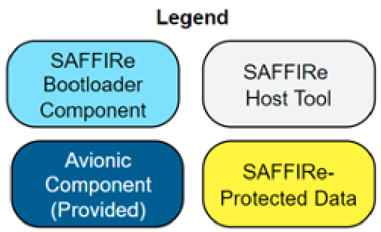
- **The aircraft contains an avionics bus with various sensors and controllers**
  - Reliable and correct operation of all avionic devices is critical for safe flight!
- This component is provided for you and cannot be modified



Aircraft

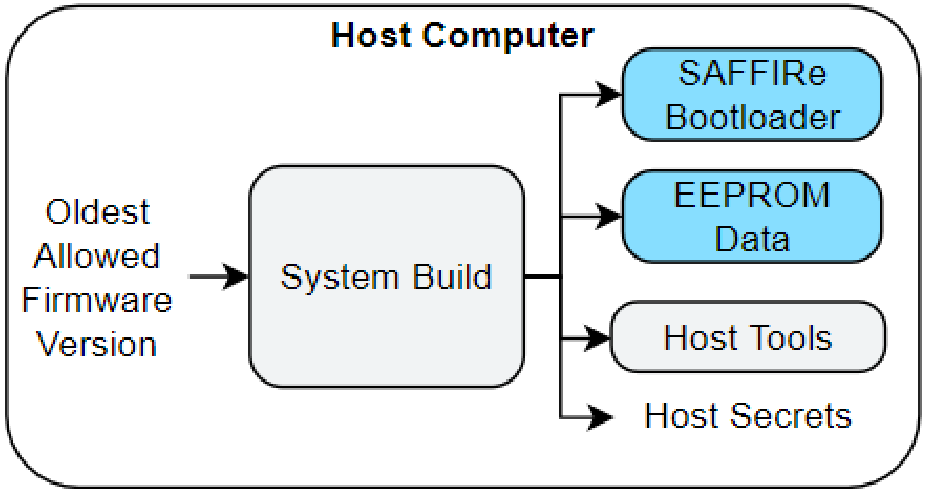
Outline

1. Welcome
2. Competition Overview
3. Challenge Overview
4. **Requirements**
5. Attack Deployment
6. Flags



SAFFIRe Build Requirements

- Generate system secrets
- Create a host tool package
- Compile the bootloader
  - The bootloader may use an EEPROM initialization file

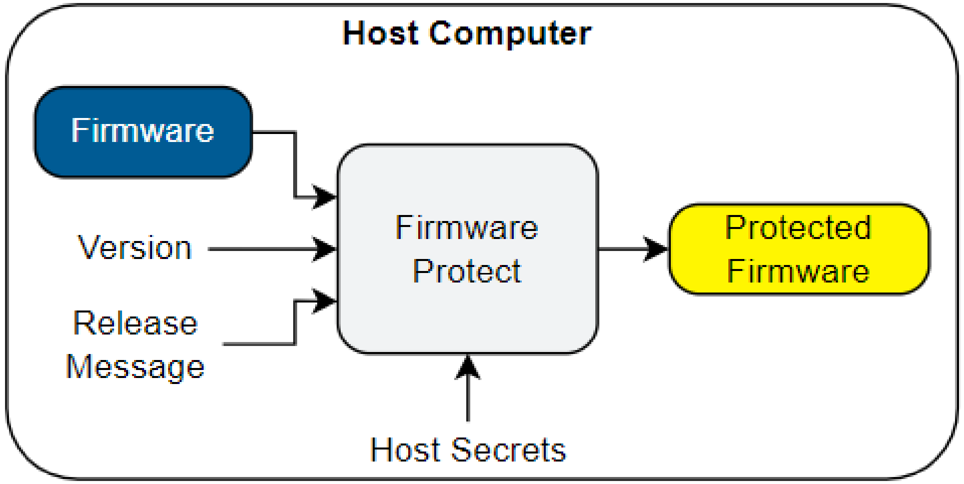


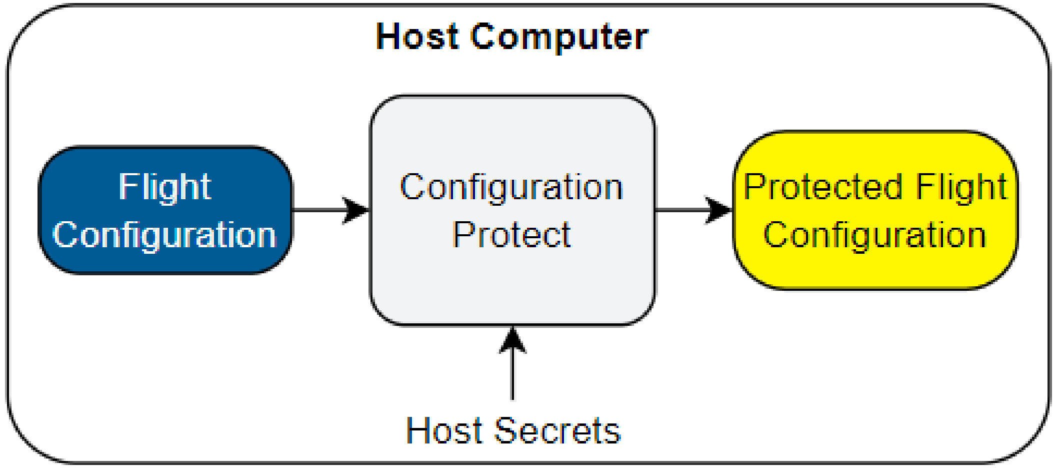
Firmware / Configuration Protection Requirements

Legend

- SAFFIRe Bootloader Component
- SAFFIRe Host Tool
- Avionic Component (Provided)
- SAFFIRe-Protected Data

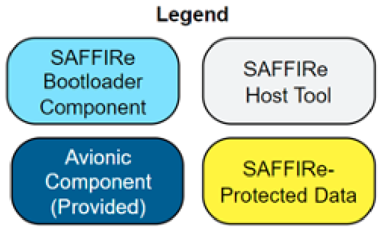
- Host tools must create images containing *protected firmware* and *protected flight configurations*
  - *Protected Firmware* images should contain a version number and release message
  - *Protected Flight Configurations* are packaged standalone (i.e., they do not contain a version number or release message)



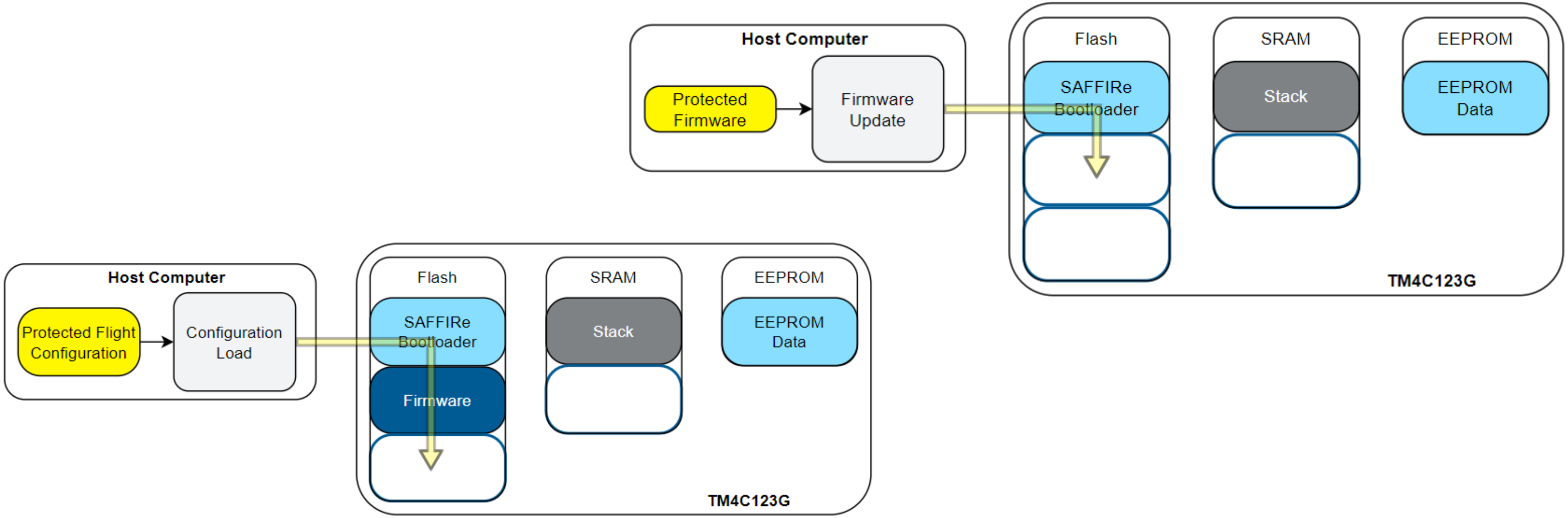


MITRE

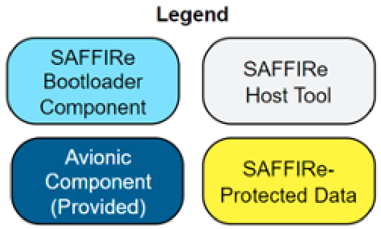
Firmware / Configuration Update Requirements



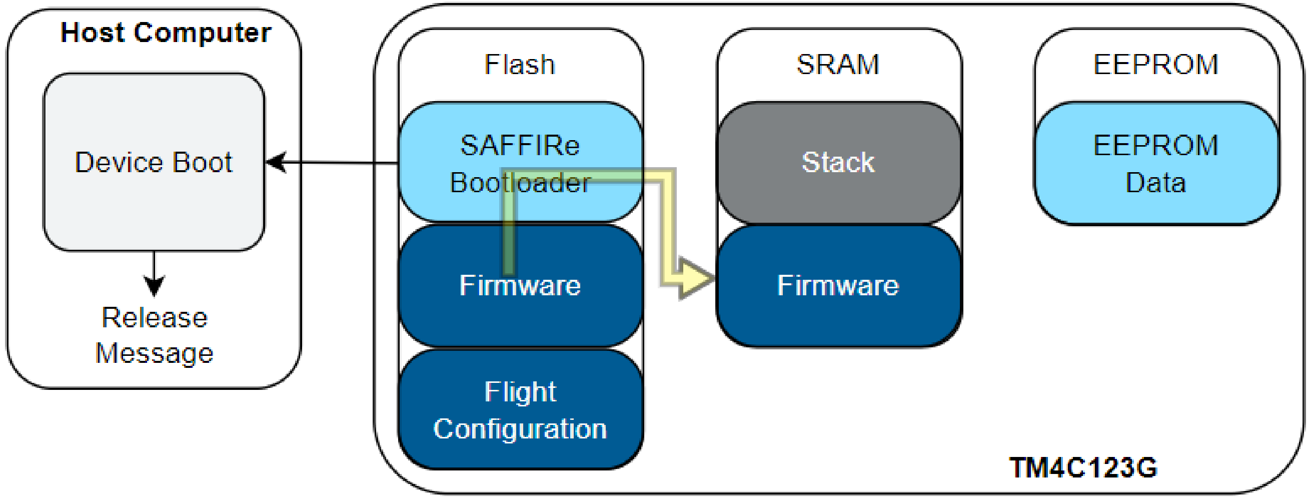
- Host tools must send protected images to the bootloader
- The bootloader must install images so they can be executed later



Device Boot Requirements



- The bootloader loads the firmware and configuration images to run attached to the avionics bus
  - Firmware placed at the end of SRAM, Configuration at the end of Flash
- The bootloader returns the release message to the host
- Finally, the bootloader executes the firmware

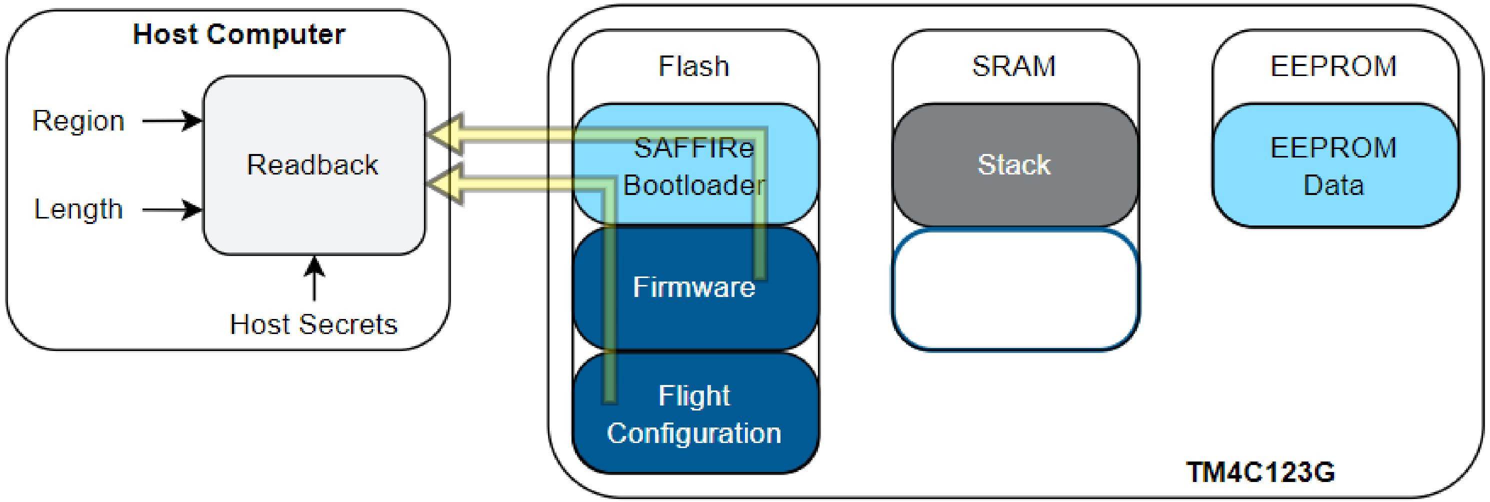


Readback Requirements

Legend

- SAFFIRe Bootloader Component
- SAFFIRe Host Tool
- Avionic Component (Provided)
- SAFFIRe-Protected Data

- The host tools request data from either the installed firmware or configuration
- The bootloader returns the amount of data requested



SAFFIRe Security Requirements

- **Confidentiality**
  - Avionic firmware and configurations should not be readable by anyone other than the intended SAFFIRe bootloader
- **Device Integrity and Authenticity**
  - The SAFFIRe bootloader should only install and boot firmware and configuration images that were created by a secure host computer
- **Firmware Versioning**
  - The SAFFIRe bootloader should only install current or newer firmware images, and should never install an old firmware version
- **Readback Authentication**
  - The SAFFIRe bootloader should only return installed image data to an authentic host with access to the host secrets

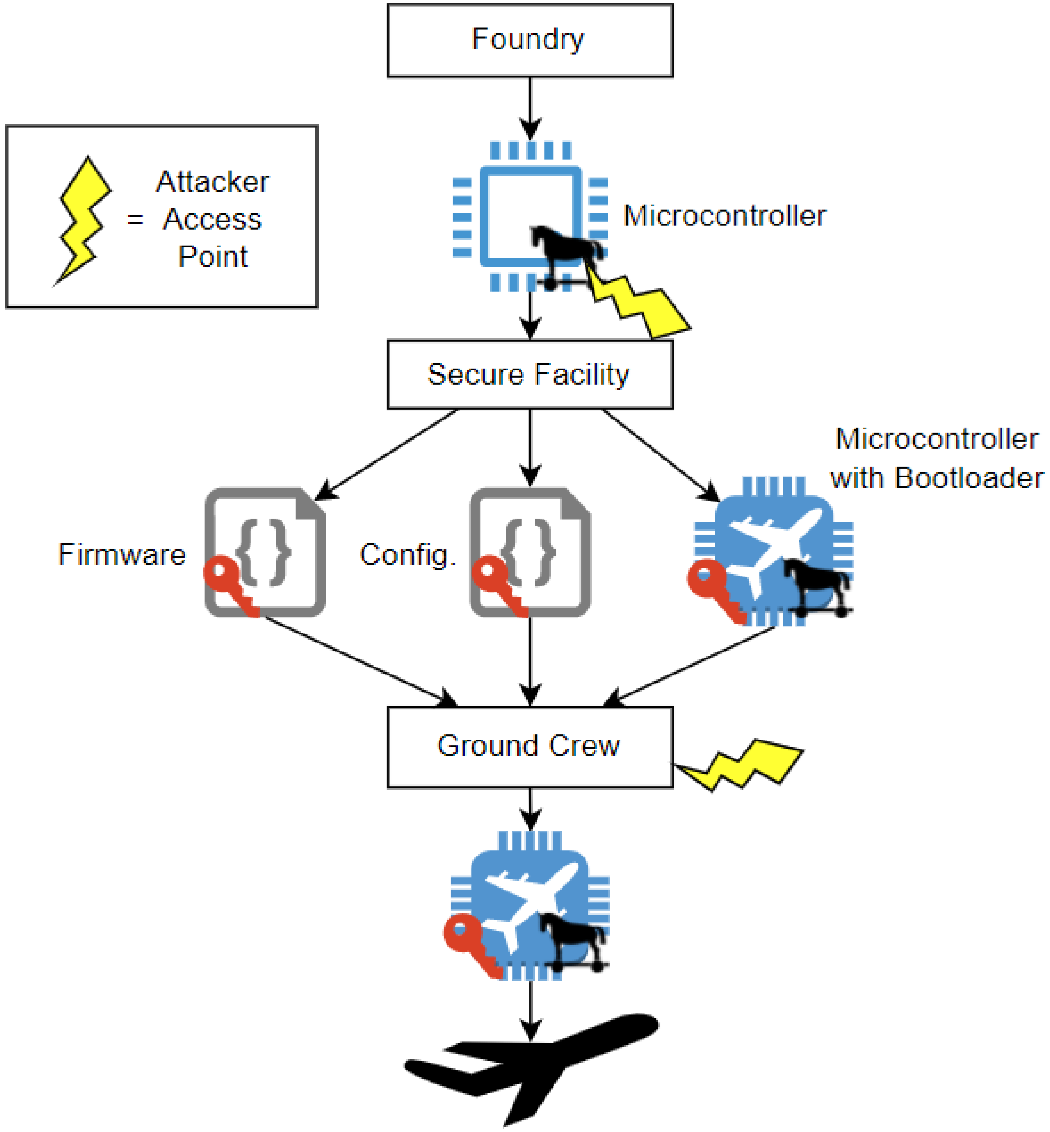
Outline

1. Welcome
2. Competition Overview
3. Challenge Overview
4. Requirements
5. **Attack Deployment**
6. Flags

Attack Phase Deployment

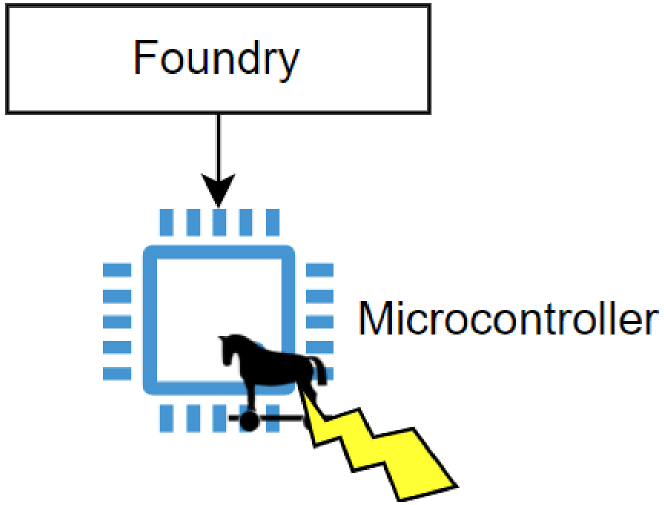
- This is the context in which your SAFFIRe system will be used

- Your team does not have to implement any of this flow



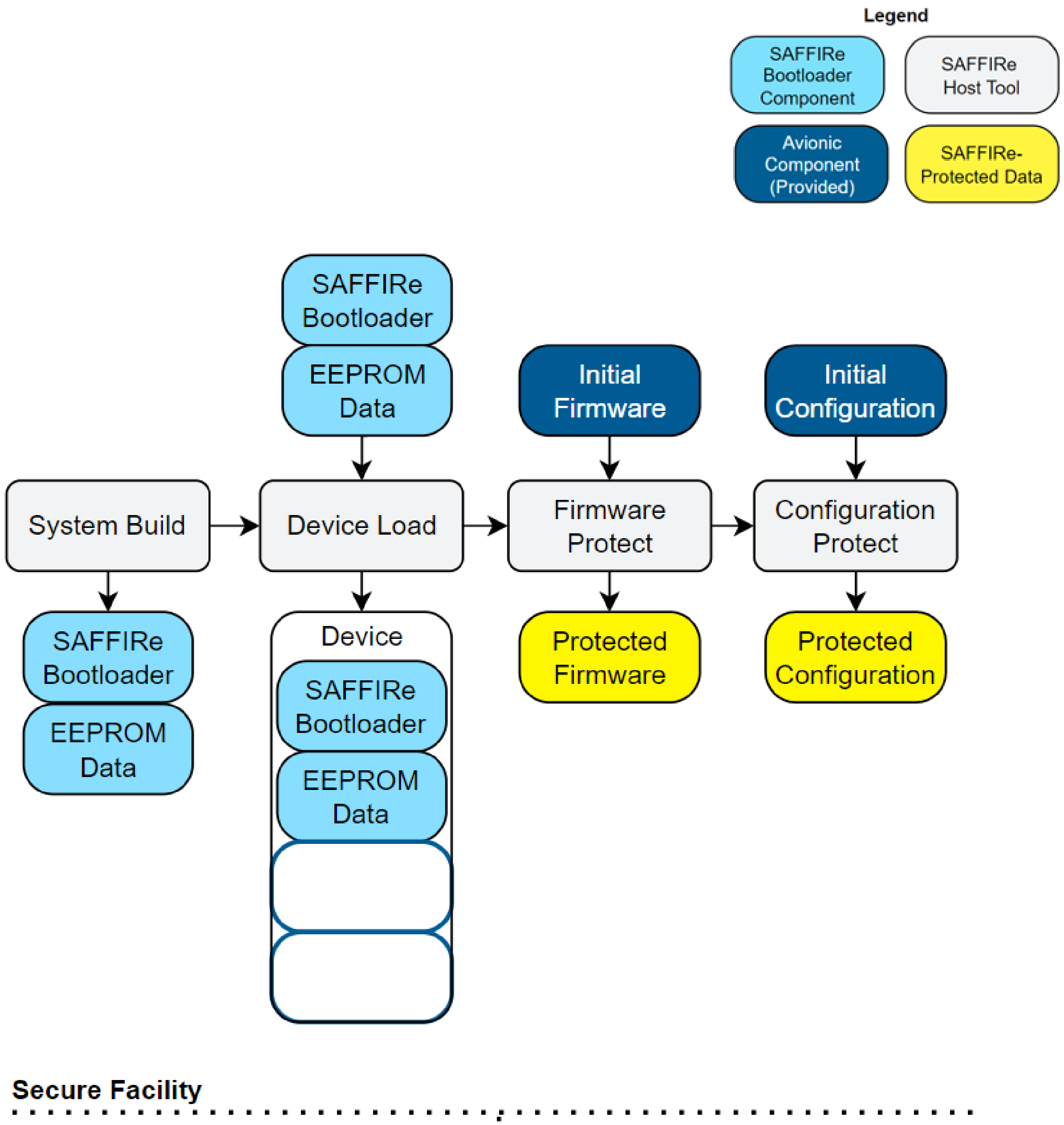
Step 1: Device Fabrication

- **The avionic device microcontroller is fabricated at an untrusted foundry**
  - Adversaries at the foundry may insert a hardware trojan into the Flash memory controller
- **The device is shipped out for commercial purchasing**
  - Your company buys it off-the-shelf



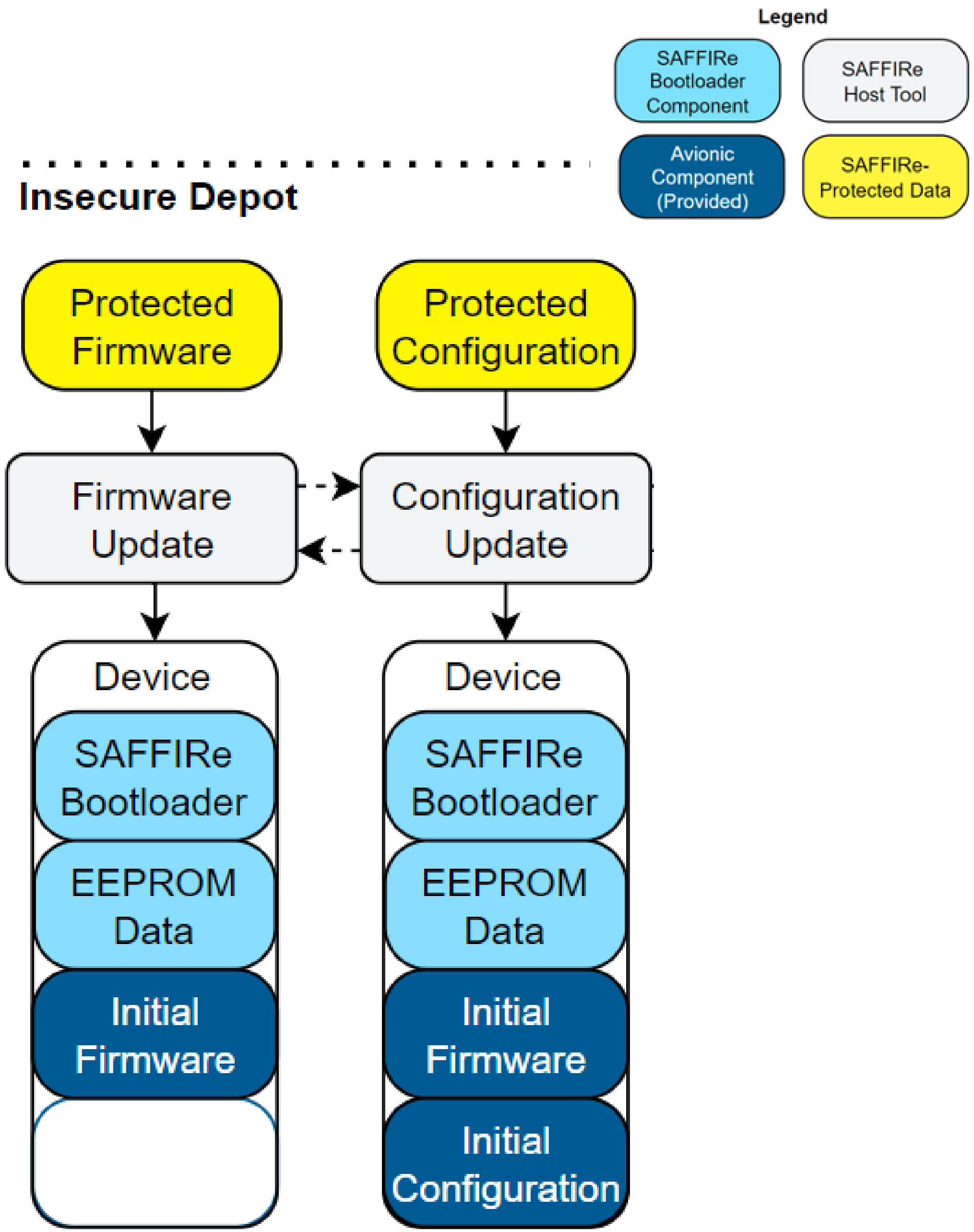
Step 2: SAFFIRe Creation

- **A SAFFIRe system is built in your company's secure facility**
- **Your company loads the SAFFIRe bootloader into the device and creates protected images**
  - Multiple firmware images are protected
  - Multiple configuration images are protected
- **Your company may install initial firmware and configurations on the device**



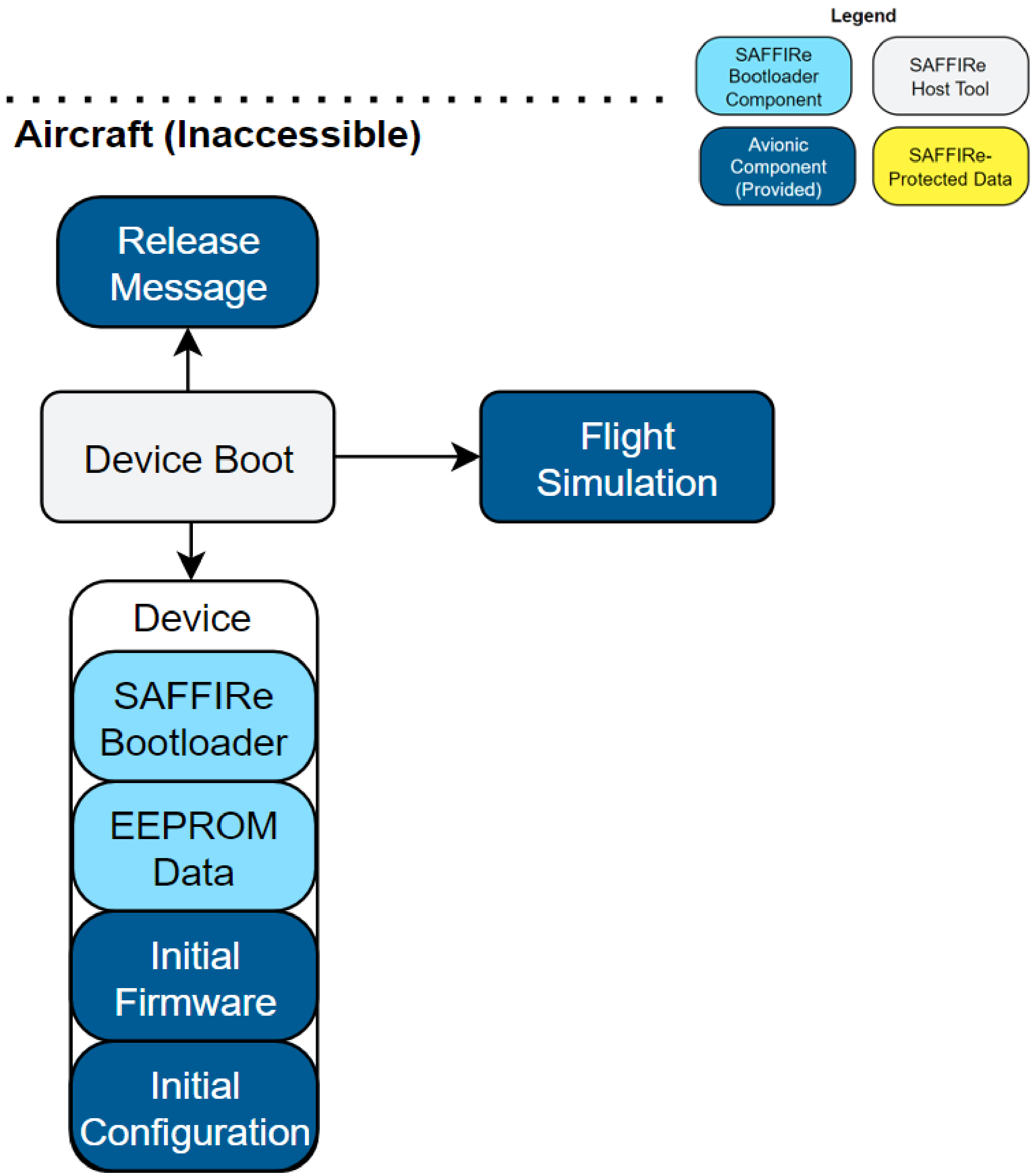
Step 3: Arrival at the Depot

- **The loaded device is shipped to the aircraft depot**
  - Protected images are sent with it
- **A disgruntled employee gets physical access to the device**
  - They may attempt to extract secret information from the device, or install malicious firmware and configurations on the device



Step 4: Aircraft Launch

- **Attacker loses access**
- **Device is placed on the aircraft**
- **The aircraft requests the bootloader to boot the system**
  - If the bootloader refuses, the aircraft does not take off
- **The aircraft takes off and runs the flight**



Attack Phase Avionic Device

- **The avionic device will play the role of the aircraft navigation computer. The firmware (which will be provided to you) does the following:**
  - Read the current location from the GPS
  - Calculate the correct heading based on the destination coordinates in the flight configuration
  - Send the heading to the autopilot

- **The rest of the avionics bus and aircraft is simulated**
  - GPS, Autopilot
  - An altimeter reports the altitude to the autopilot
  - A bus controller informs bus devices of who can send privileged commands, like *start* and *shutdown*

Aircraft Exceptions

- **Flight Abort**
  - If the pilot detects that devices are sending incorrect data, they turn the aircraft around and abort the flight
  - Example: An incorrect flight path is set

- **Aircraft Crash**
  - If critical safety features are disabled at the bus level, the aircraft will be uncontrollable and crash
  - Example: The altimeter is forced to shut down

What Attackers Will Receive (For Each Target System)

- All source code (with the .git directory removed)
- The most recent documentation for the target system
- A protected SAFFIRe bootloader image to load onto their physical hardware
- Access to an emulator with the SAFFIRe bootloader installed
  - Data and reset interfaces
  - Emulated side-channel probe
  - Optional: Trojan running in the microcontroller
- Protected firmware and configuration images

UNDERSTAND WHAT THE ATTACKERS HAVE ACCESS TO

Outline

1. Welcome
2. Competition Overview
3. Challenge Overview
4. Requirements
5. Attack Deployment
6. **Flags**

Design Phase Flags

- **Encourage teams to stay on track during development**

Milestone	Description	Target Date	Deadline Date
Read Rules	If you read all the rules, you'll know	1/25	2/2
Boot Reference	Provision and boot the example SAFFIRe design to receive a flag (see the README)	1/28	2/9
Use Debugger	Use the GDB target in the top SAFFIRe script to step through a binary and retrieve a flag. See the reference design for details.	1/31	2/16
Design Document	Submit an initial design document containing high-level descriptions of how each host tool and bootloader function will work in your system.	2/9	2/23
Bug Bounty	Find and fix bugs in the reference design	N/A	Handoff

Reverse Engineering Challenge

- **One firmware binary to RE**
  - RE Binary
  - Determine correct input
  - Run binary and send input to dispense flag
  - Reveals the bus ID of the altimeter
- **Avionic bus interface source code**
  - RE bus interface logic present in every avionic device
  - Develop custom code to exploit bus vulnerabilities and shut down a device
  - Submit a firmware binary that forces the altimeter to shutdown
  - Flag awarded upon aircraft crash

Side-Channel Analysis Challenge

- **Collect side-channel traces from an emulated device running an AES operation**

- **Three steps to the challenge**
  1. Collect side-channel traces that cover an AES operation
  2. Filter and align multiple traces
  3. Recover the cipher key (DPA recommended)

- **Teams submit trace plots according to a specific format**
  - Teams submit the key for the last challenge
  - Teams awarded flags upon verification of trace collection

Attack Phase Flags

Flag Name	Capturing this flag proves that you can compromise…	To Submit this Flag	Requires Aircraft Simulation
Confidentiality Flags			
IP Extraction	Firmware Confidentiality	Extract the flag by reading any protected firmware image	No
Flight Extraction	Configuration Confidentiality	Extract the flag by reading any protected flight configuration	No
Rollback Flags			
Firmware Rollback	Firmware Versioning	Install and boot an old firmware image	No
Integrity Flags			
Data Extraction	Firmware Integrity / Bootloader Execution	Read out data from EEPROM of device	No
Flight Abort	Firmware and Configuration Integrity	Boot a corrupted firmware or configuration that makes the aircraft deviate from the flight plan	Yes
Aircraft Crash	Firmware Integrity	Boot a malicious firmware that exploits bus vulnerabilities to shutdown aircraft safety features, crashing the aircraft	Yes

MITRE

Words of Advice

- **Start development early**
  - Verify functionality as you go
  - Get comfortable with the build process – familiarize yourself with how the main components work
- **Always think like an attacker**
  - Especially with the hardware trojan threat!
- **Be creative yet elegant with countermeasures**
- **Use Slack for help**
  - Use #tech-support !
- **Understand what attackers have access to**

Next Steps

- **Read the rules and the technical specifications**
- **Start designing your system**
- **Get access to the development server**
- **Get the reference system running**
- **Begin development**

Jake Grycel

jgrycel@mitre.org

ectf@mitre.org

https://www.linkedin.com/groups/12371545/

SOLVING PROBLEMS
FOR A SAFER WORLD®