Secure Runtime Auditing of Remote Embedded System Software

Adam Caulfield

Advisor: Dr. Ivan De Oliveira Nunes

Rochester Institute of Technology

CAE-R Research Symposium
September 2023

Embedded devices - Smart Spaces & “Internet of Things”

- Low-end, energy efficient, low cost
- Resource constrained — security
- Execute safety-critical tasks in modern systems
  - Sensor/alarm system
  - Modern medical device
- Must monitor device behavior to determine unexpected/malicious activity

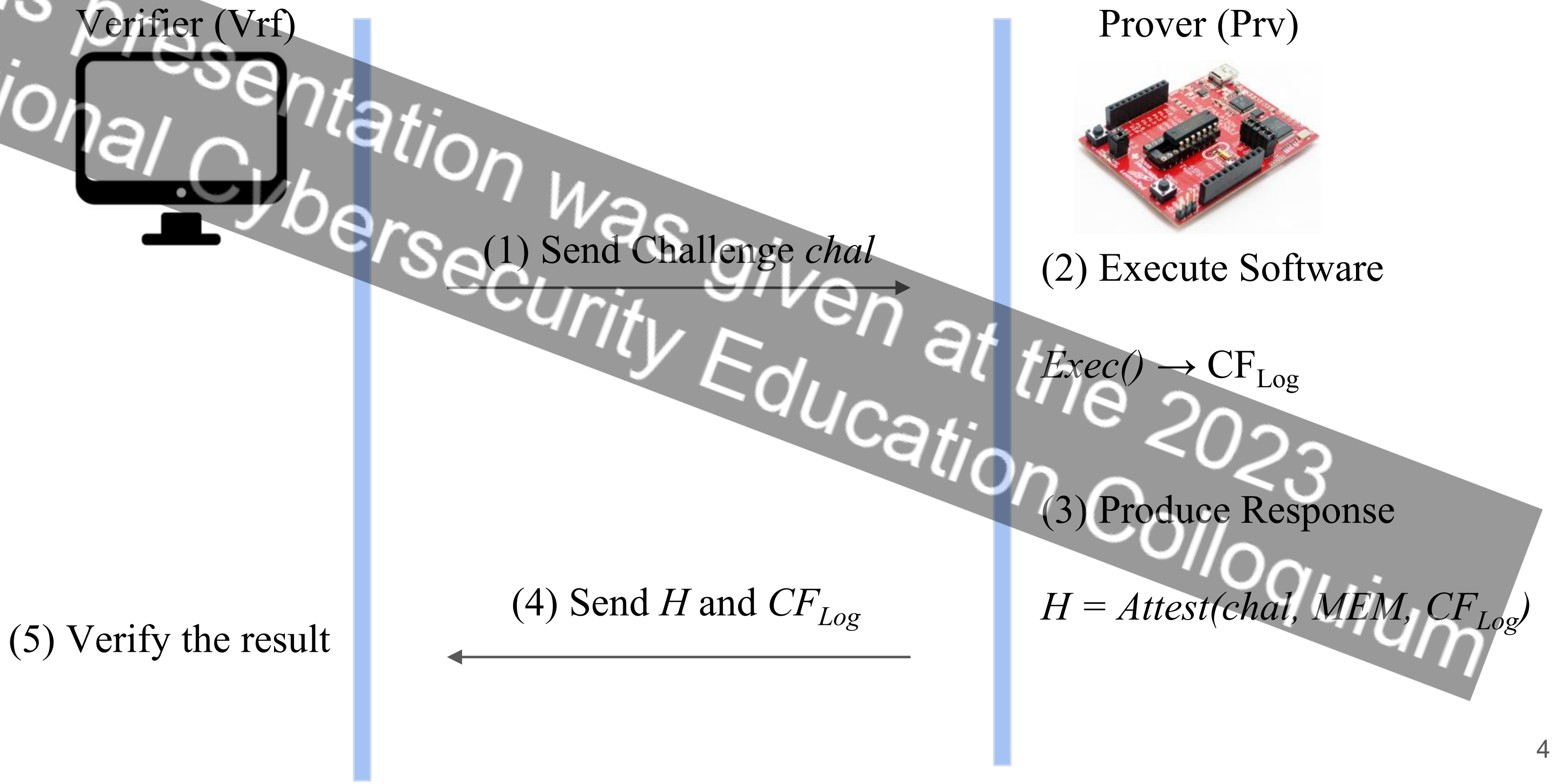
Can we achieve ***runtime auditing*** of a remotely deployed (potentially compromised) MCU?

Desired security guarantees for runtime auditing:

1. Generate authentic/accurate evidence of the exact runtime behavior
2. Deliver the evidence to device operator for further analysis
3. After compromise is detected, provide a means to remotely remediate the source of the compromise

Control Flow Attestation (CFA):

Generate evidence of static and runtime integrity of remote device



Verifier (Vrf)

Prover (Prv)

(1) Send Challenge *chal*

(2) Execute Software

$Exec() \rightarrow \mathrm{CF}_{\mathrm{Log}}$

(3) Produce Response

$H = Attest(chal, MEM, CF_{Log})$

(4) Send H and CF_{Log}

(5) Verify the result

From *Attestation* to *Auditing*

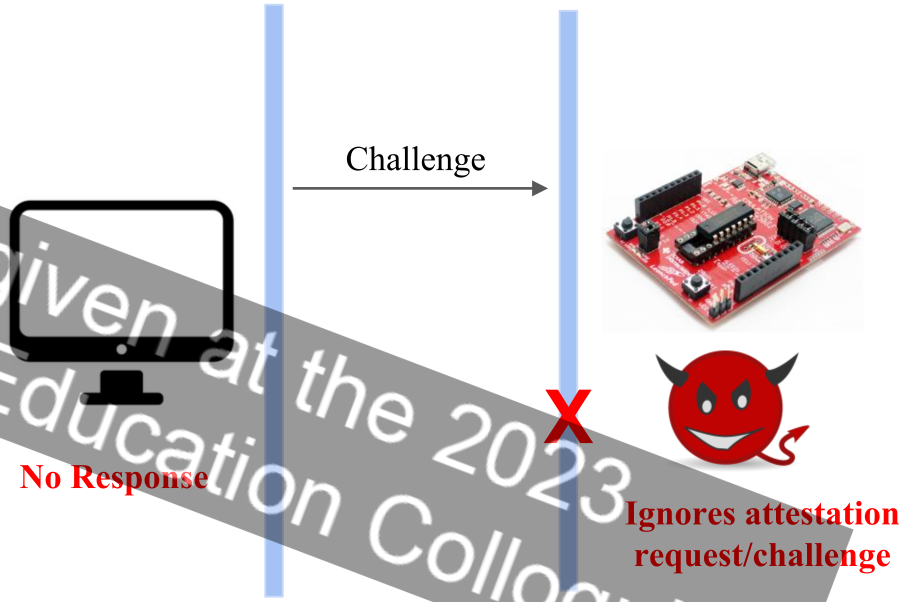
- Attestation is a ***passive*** technique

- No guarantee that Verifier receives the response

- Attestation – ***something is wrong***

- Auditing – ***what is wrong***

- Must physically intervene



The problem…

Current Techniques

✓ Guarantees runtime evidence is accurate/authentic

✗ Cannot guarantee eventually delivery of runtime evidence to Vrf

✗ No ability to remotely intervene after compromise detection

Research Question 1

What exact security features are required to enable runtime auditing under full software compromise?

Research Question 2

How to achieve secure runtime auditing in commodity devices (i.e., without custom hardware support)?

Research Question 3

To what extent does runtime auditing interfere with performance, and how can this be mitigated without giving up on security?

Research Question 4

Can runtime evidence be used to identify (previously unknown) vulnerabilities and pinpoint the root cause of compromises?

Results thus far and next steps…

- ACFA (**USENIX Security 2023**)
    - Secure runtime auditing and compromise remediation for low-end devices (MCUs)
    - Requires hardware modifications…
    - **Check our poster for details!**
    - Paper: https://people.rit.edu/ac7717/acfa.pdf

- Runtime auditing on commodity devices (**ongoing**)
    - Leveraging pre-existent hardware support (e.g., ARM TrustZone M)

- Improving efficiency of runtime auditing schemes (**ongoing**)
    - Complete runtime evidence can be **huge!!!**
    - How to efficiently store and deliver of runtime evidence?