Red-C

Reclaiming Bus-based Systems During Compromise (Red-C)

Mr. Bernard McShea

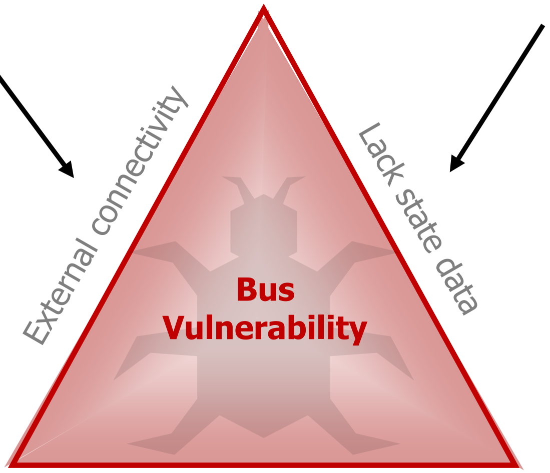
Proposers Day: January 2025

Reclaiming Bus-based Systems During Compromise (Red-C)

Explore algorithms to construct self-healing systems, by retrofitting individual components on a bus to function as forensic sensors that collectively monitor peers to detect, repair, and inoculate on-system during a cyber-attack.

In 2000, widely available internet connectivity enhanced the features of computers and vehicles.

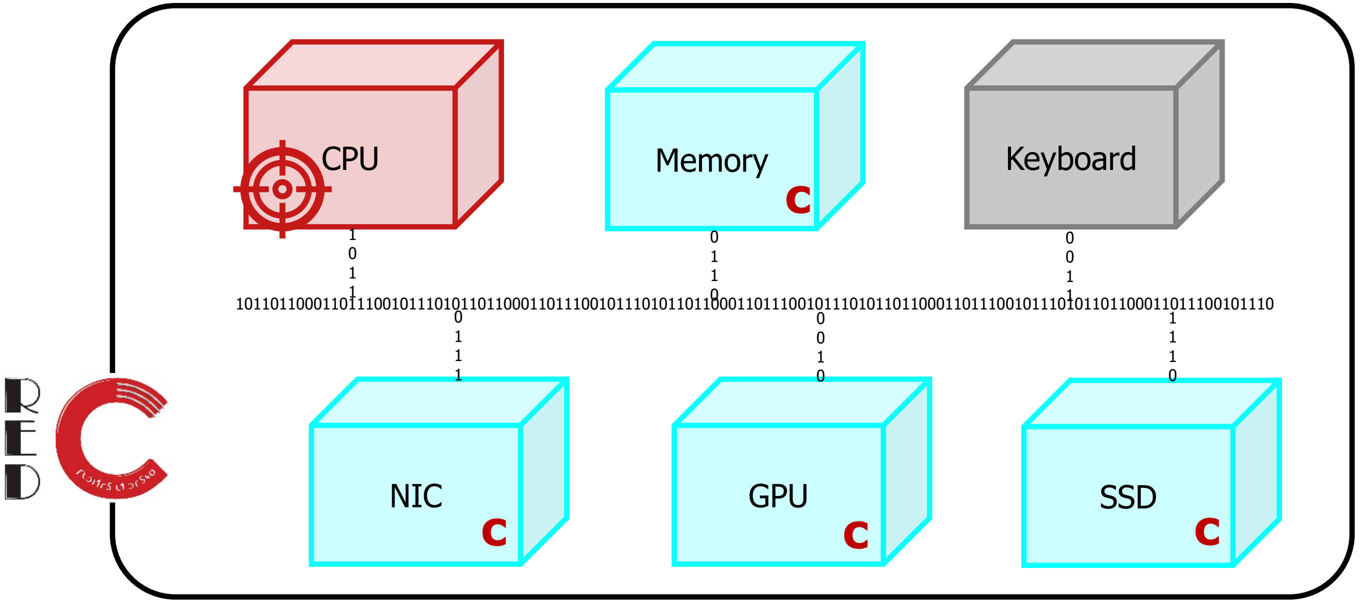
Data on the bus is transactional and does not inform forensics or system restoration.



Implicit trust

When buses were introduced, in computer systems ~1960s and in cars ~1970s, implicit trust was not an issue as physical access was required to connect to the bus.

Modern buses are crossed multiple times by processes on a system to reach numerous specialized components to leverage their resources, thus leaving an ever-growing forensic footprint.



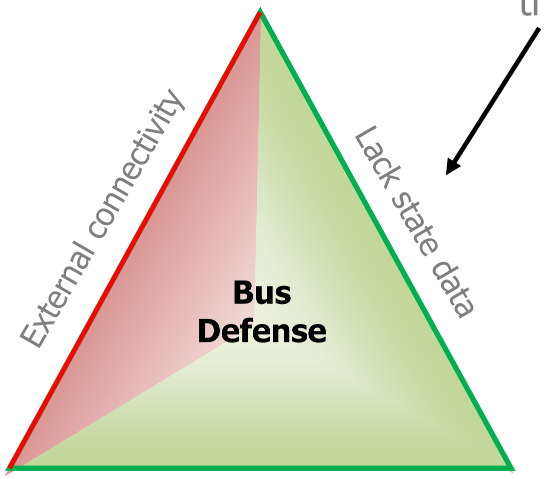
Highlighted in blue are a few of the PCIe components capable of detecting adversary activity during compromise.

GPU – Graphical Processing Unit
CPU – Central Processing Unit
NIC – Network Interface Card
SSD – Solid State Drive

Reclaiming Bus-based Systems During Compromise (Red-C)

Explore algorithms to construct self-healing systems, by retrofitting individual components on a bus to function as forensic sensors that collectively monitor peers to detect, repair, and inoculate on-system during a cyber-attack.

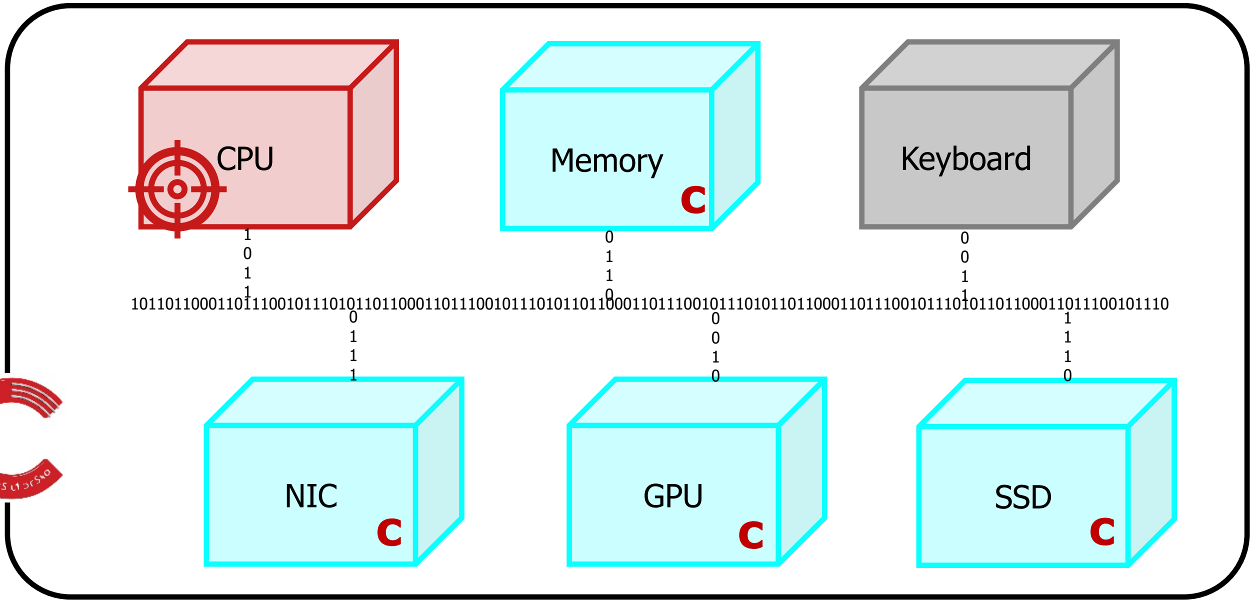
TA1 - Instrumentation will impart forensic and system state data on the bus.



Implicit trust

TA2 - Response will cooperatively detect, repair, and inoculate bus-based components during active cyber-attack on-system in near real-time.

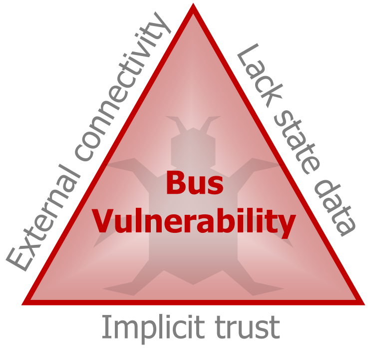
Modern buses are crossed multiple times by processes on a system to reach numerous specialized components to leverage their resources, thus leaving an ever-growing forensic footprint.



Highlighted in blue are a few of the PCIe components capable of detecting adversary activity during compromise.

GPU – Graphical Processing Unit
CPU – Central Processing Unit
NIC – Network Interface Card
SSD – Solid State Drive
TA – Technical Area

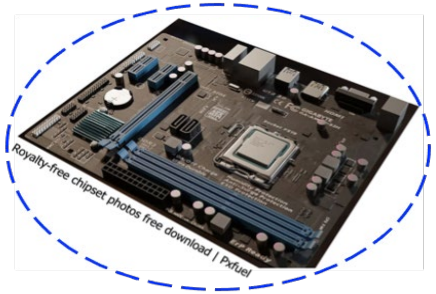
Buses remain vulnerable to cyber attack



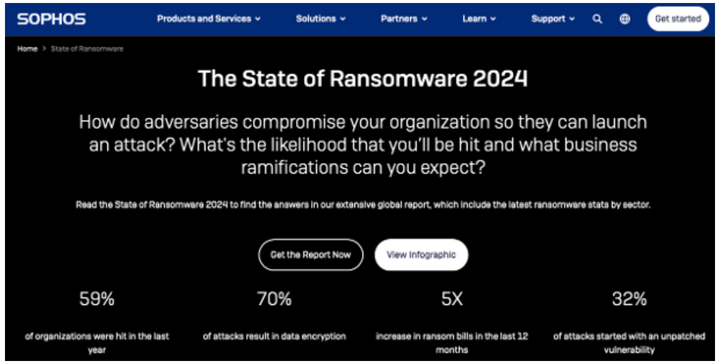
Problem

- Components implicitly trust each other and are externally accessible
- Many components are externally connected, expanding attack surface
- System recovery is hindered by the lack of forensic information available on the bus

PCIe/CXL

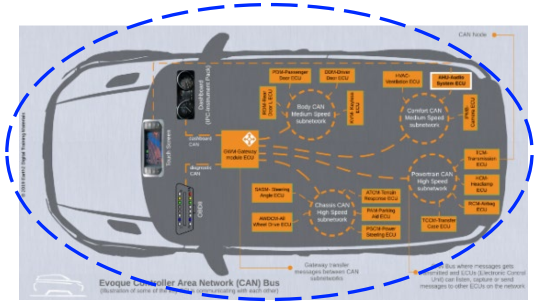


2015: PCIe bus Jellyfish GPU malware key logger [1]

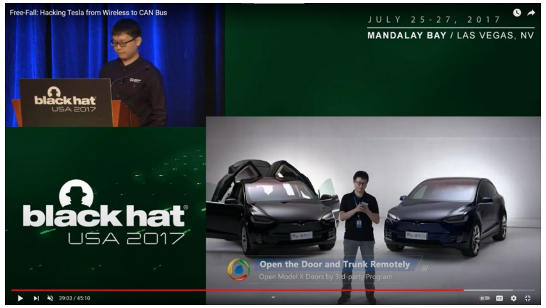


2024: Ransomware attacks continue [2]

CAN



[3]

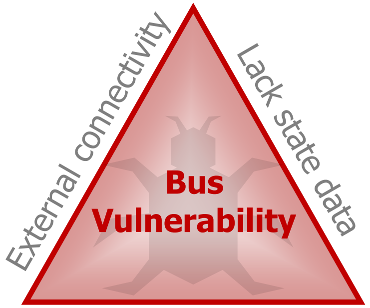


2017: CAN bus attack launched from the internet reached doors, trunk, steering, and brakes [4]

2023: CAN bus physically tapped via the headlight giving thieves full access [5]

GPU – Graphical Processing Unit
CAN – Controller Area Network
PCIe – Peripheral Component Interconnect Express

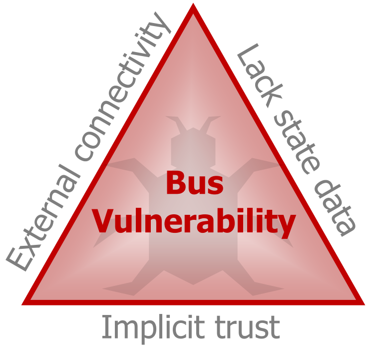
Buses remain vulnerable to cyber attack

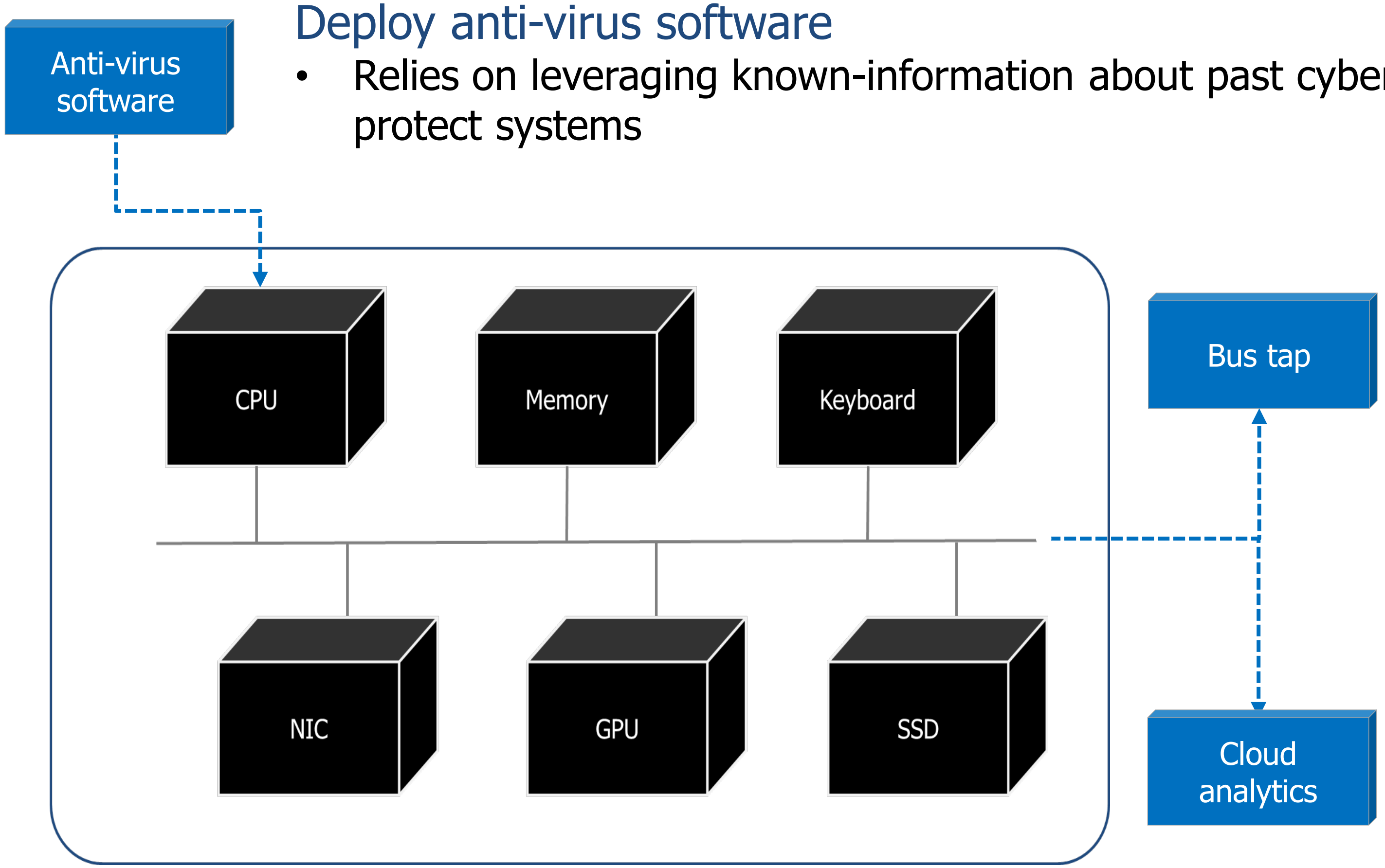


2024: Colorado State proof-of-concept commercial vehicle Electronic Logging Devices (ELD) attack reached steering, brakes, etc. [1]

ELDs are mandated by law to remain powered on.

State-of-the-art of bus protection





Deploy anti-virus software

- Relies on leveraging known-information about past cyber-attacks to protect systems

Analyze traffic on the bus

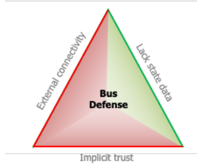
- Adds additional SWAP
- Single point of failure
- Machine learning implementations have known vulnerabilities (e.g., DARPA GARD program)

Analyze traffic in the cloud

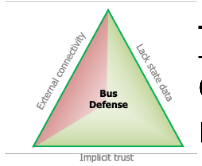
- Creates additional logs to store and secure
- Defense observations are limited to bus traffic
- Detection of a compromise occurs off-system

CAN – Controller Area Network
GPU – Graphical Processing Unit
SWAP – Size, Weight and Power
CPU – Central Processing Unit
NIC – Network Interface Card
SSD – Solid State Drive
GARD – Guaranteeing AI Robustness Against Deception

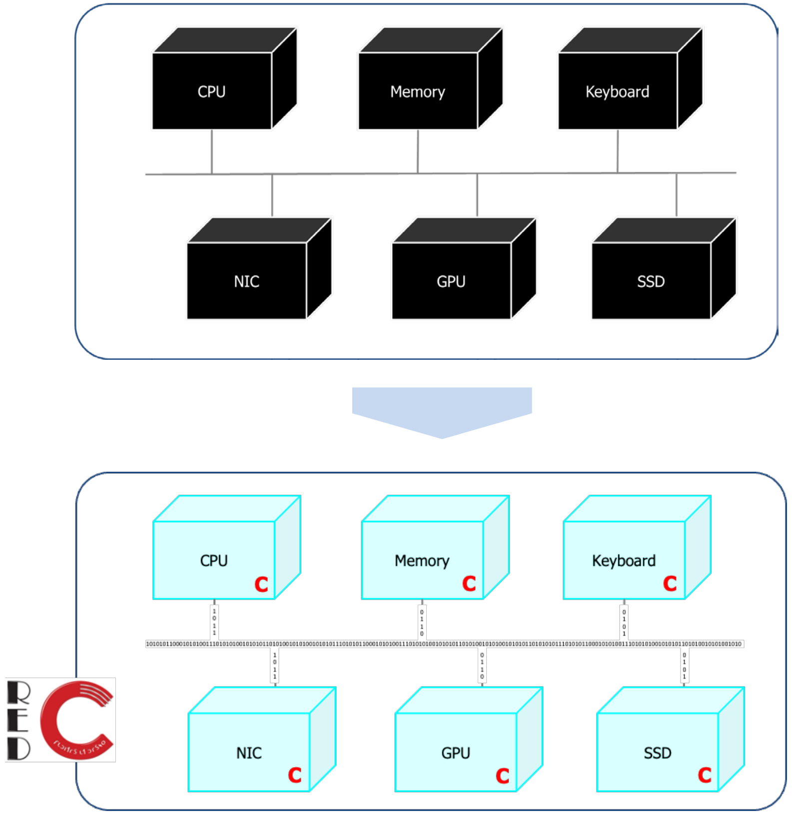
Red-C Approach

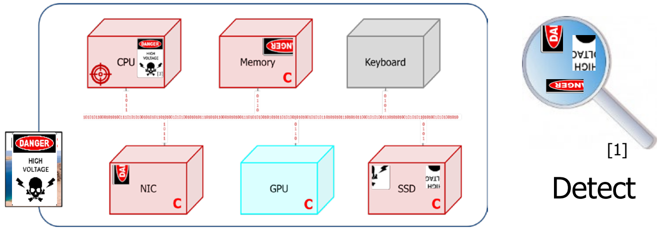


<u>TA1 - Instrumentation</u> will develop fine-resolution sensing via instrumenting a set of critical components to monitor each other cooperatively

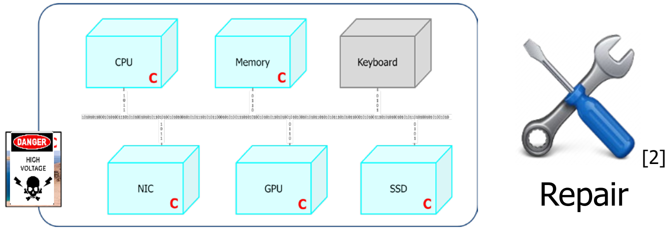


<u>TA2 - Response</u> will develop distributed algorithms for components to cooperatively detect, maximize recovery, and remove the vulnerability used by the attacker on-system

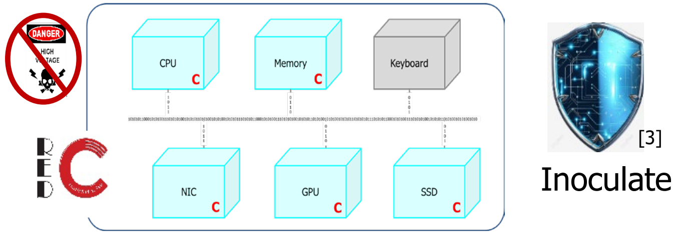




Attack detected via distributed consensuses



Forensic data provided by effected components enables restoration



Red-C data enables on-system code and configuration modification

CAN – Controller Area Network
GPU – Graphical Processing Unit
SWAP – Size, Weight and Power

CPU – Central Processing Unit
NIC – Network Interface Card
SSD – Solid State Drive

TA1 Instrumentation: Develop instrumentation and attestation

Each component in a system must consider what it can see, what it can say, and who it can inform

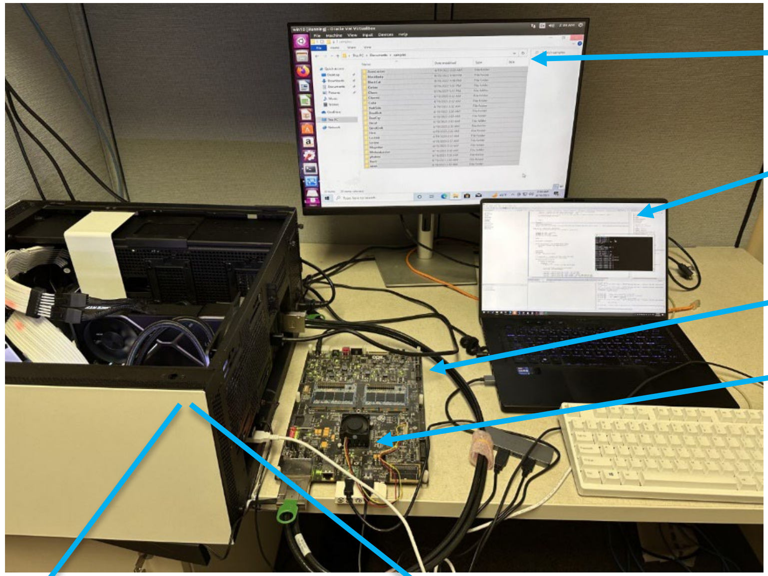
Challenges

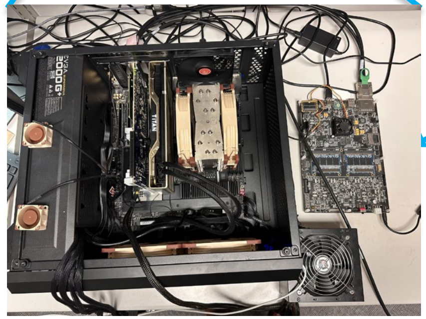
- Instrument firmware as distributed forensic sensors creating unique and immutable signals at the component level
- Coverage of forensic signal for critical bus components
- Components have scarce availability of computation, limited bus bandwidth, and contending with the attacker(s) in the defined system

Potential approaches

- Construct Forensic Observation Vectors (FOV) [1] derived from the forensic data collected on components including; computation, memory, and storage data
- A single FOV would act as a forensic instrument collecting relevant signals to inform detection of part of the cyber-attack kill chain (e.g., MITRE ATT&CK frameworks PCIe [2] and CAN [3])
- Gain trust in some of the components over time via decentralized attestation (e.g., zero-knowledge, encryption)

NYU Seedling Experimental Setup for Collecting Traces





Firmware Instrumented in NYU seedling
CPU, GPU, Keyboard, SATA controller, NIC, and SSD

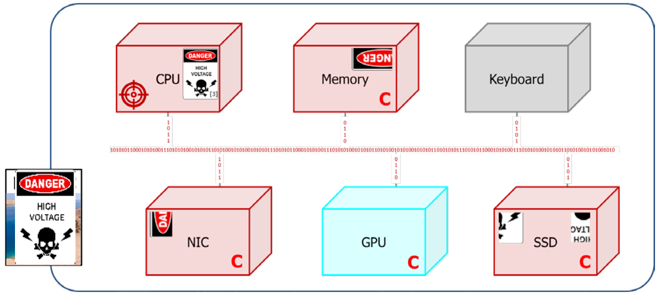
TA2 Response: Explore detection, repair, and inoculation

Challenges

- Detection, repair, and inoculation are interrelated
- Explore distributed consensus algorithms with:
  - scarce availability of computation,
  - limited bus bandwidth, and
  - untrusted fragmented chronology forensic signals
- Recovering a system while the attacker(s) is in one or more components (e.g., preventing active sabotage)
- Automated strategic patch generation to inoculate systems
  - Identifying the root cause of the vulnerability in near real-time
  - Patching code on a running system with limited visibility while ensuring global and local state preservation

Potential approaches

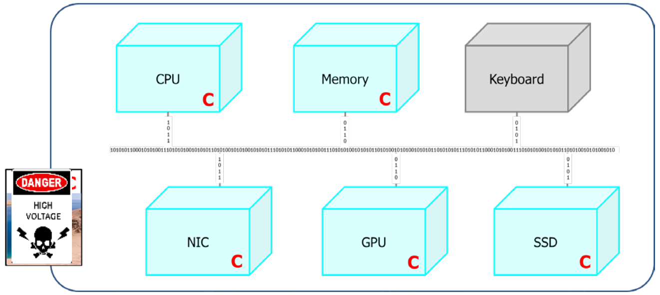
- Cooperative agreement, where each of the components is monitoring its peers to detect and recover from attacks
- Decentralized on-system mitigation leveraging components with disparate computation and memory
- Inform and automate strategic patching, ranging from configuration change to real-time code generation and patching the running system



Attack detected via distributed consensuses

[1]

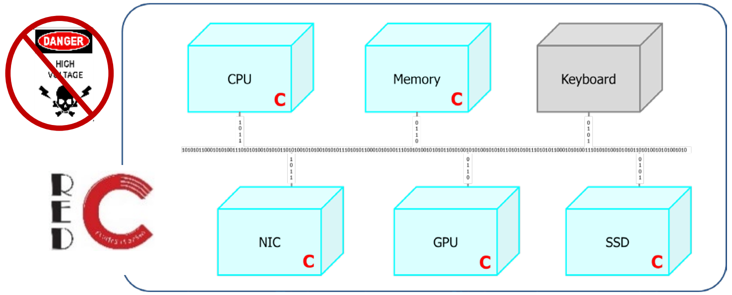
Detect



Forensic data provided by effected components enables restoration

[2]

Repair



Red-C data enables on-system code and configuration modification

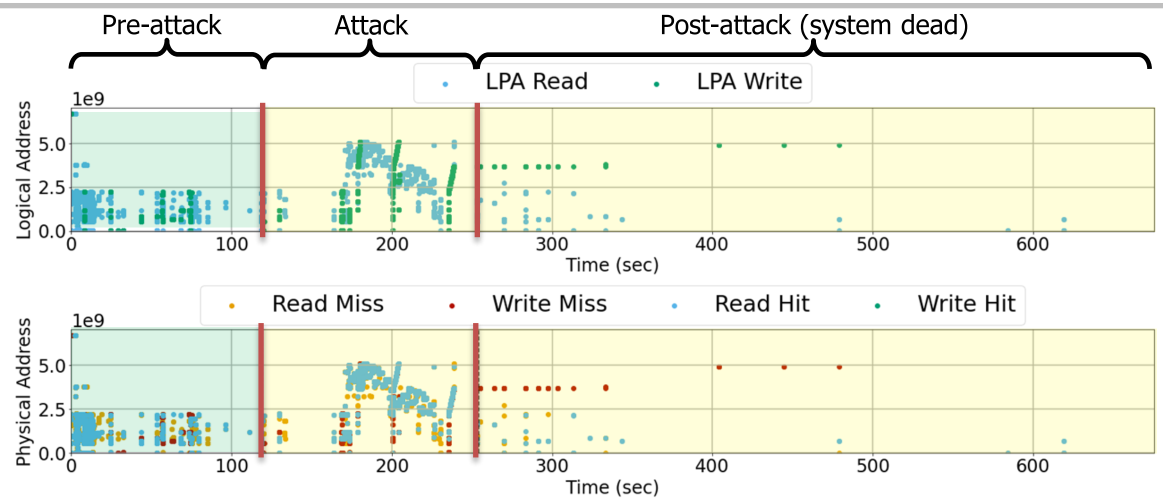
[3]

Inoculate

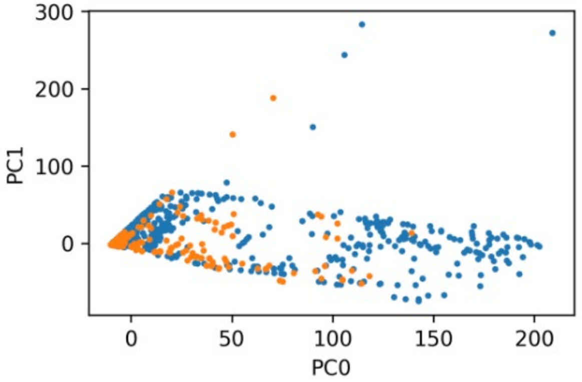
CAN – Controller Area Network
GPU – Graphical Processing Unit
SWAP – Size, Weight and Power

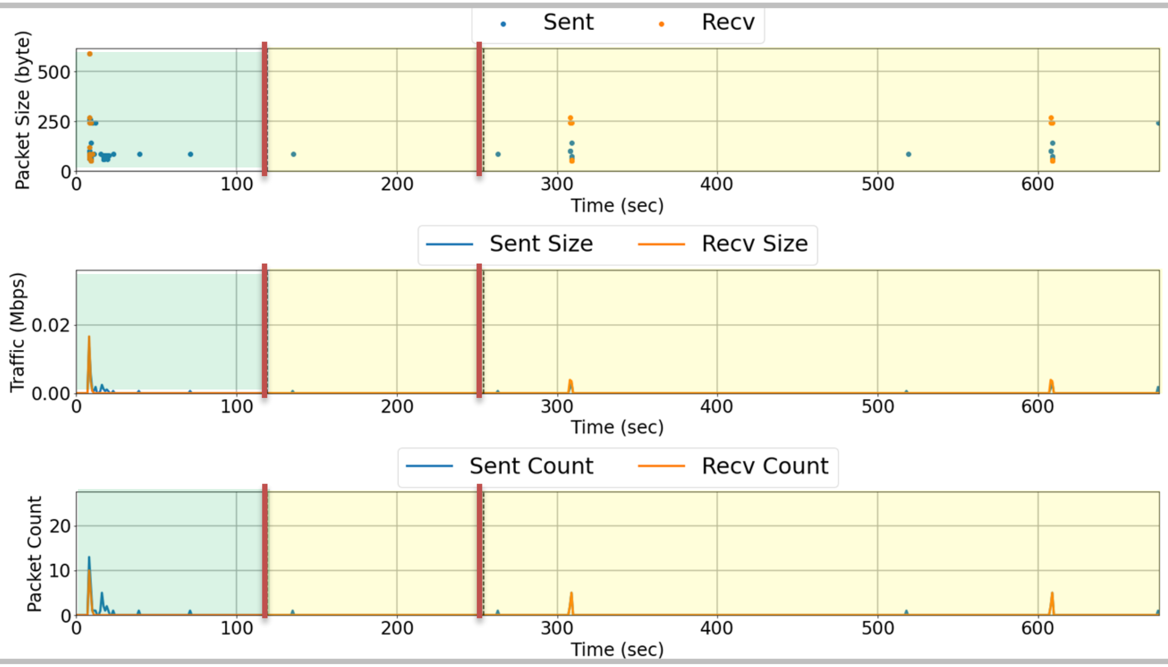
CPU – Central Processing Unit
NIC – Network Interface Card
SSD – Solid State Drive

Detection basis of confidence – improved ransomware identification

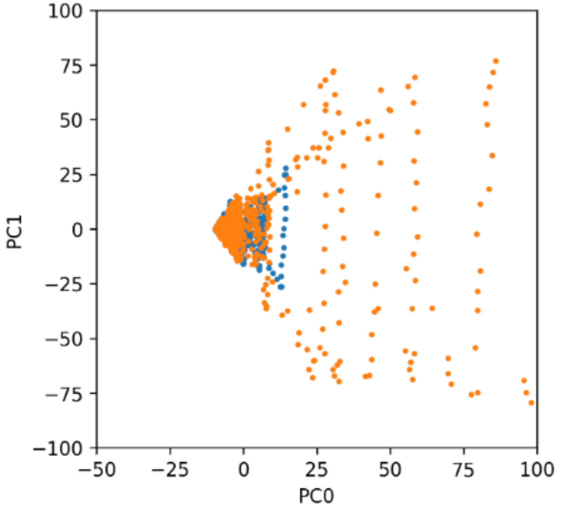


Ransomware detection using storage only



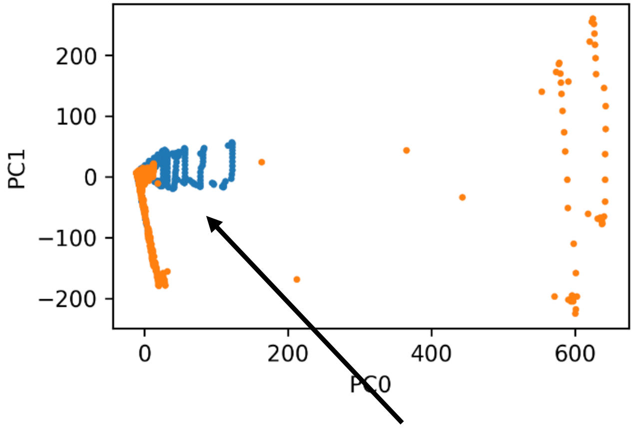


Ransomware detection using network only



Basis of confidence: study of conti ransomware attack

Ransomware detection using both network and storage



Improved clustering

● benign ● ransomware

Aggregated signal amplifies detection

Source for images on this slide: LastAct seedling

Repair basis of confidence – guaranteed 3-day recovery window

Leverage cooperative zero-trust to enable detection, on-system mitigation, and automate strategic patch generation

Basis of confidence

- NYU seedling result for component resource impact for a FOV was a 6% increase in processing and a storage increase of 0.3%
- Inform patch selection via timing analysis (Hsu et al. 2023)
- CAN bus
  - Decentralized cryptographic firmware attestation via Double-ratchet protocol (Khodari et al. 2019)
  - Applying Zero Trust Principles to Distributed Embedded Engine Control Systems. (Pakmehr et al. 2022)
  - Dynamic, Real-Time Analysis, Patching and Protection of Vehicle System Binaries (Brock et. al. 2023)
- PCIe/CXL – NYU and Purdue seedlings
  - FOV sensors, NIC and SSD components, demonstrated collectively they were more effective in ransomware detection than independently
  - 3-day file recovery window guaranteed by the SSD

On-system Mitigation

Guaranteed 3-day recovery window for files after ransomware attack.

Each line represents a distinct SSD workload.

CAN – Controller Area Network
NIC – Network Interface Card
SSD – Solid State Drive

Inoculate basis of confidence – decreased patch generation time

Proposals should detail their range of approaches to inoculation

PCIe defining a minimum set of Intelligent Platform Management Interface (IPMI) APIs which can be used under specific attack detection types to remove effected attack surfaces

DARPA AMP: Inform patch selection via timing analysis (Hsu et al. 2023)

State of the Art

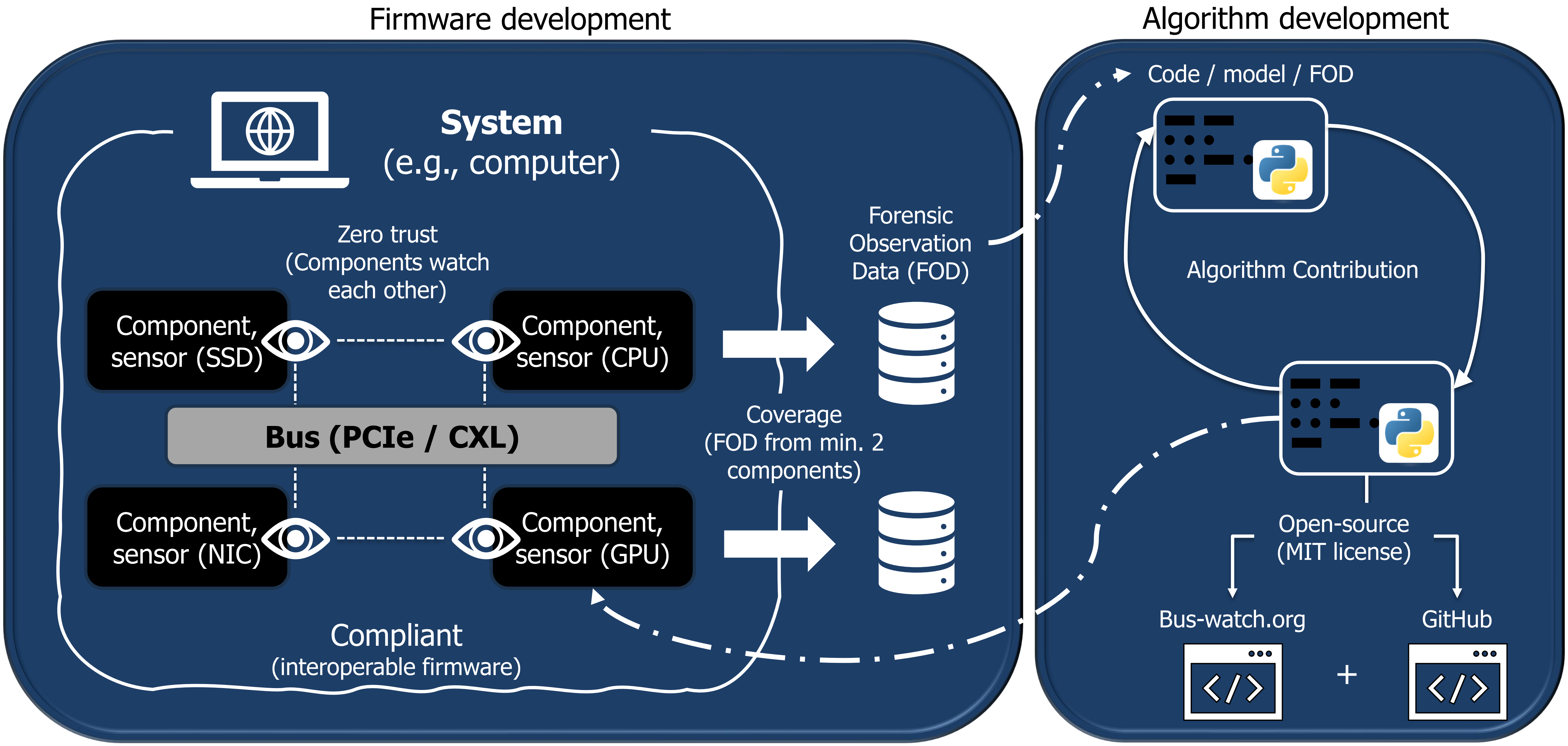
Red-C will push the boundary of real time inoculation

Predetermined configuration changes

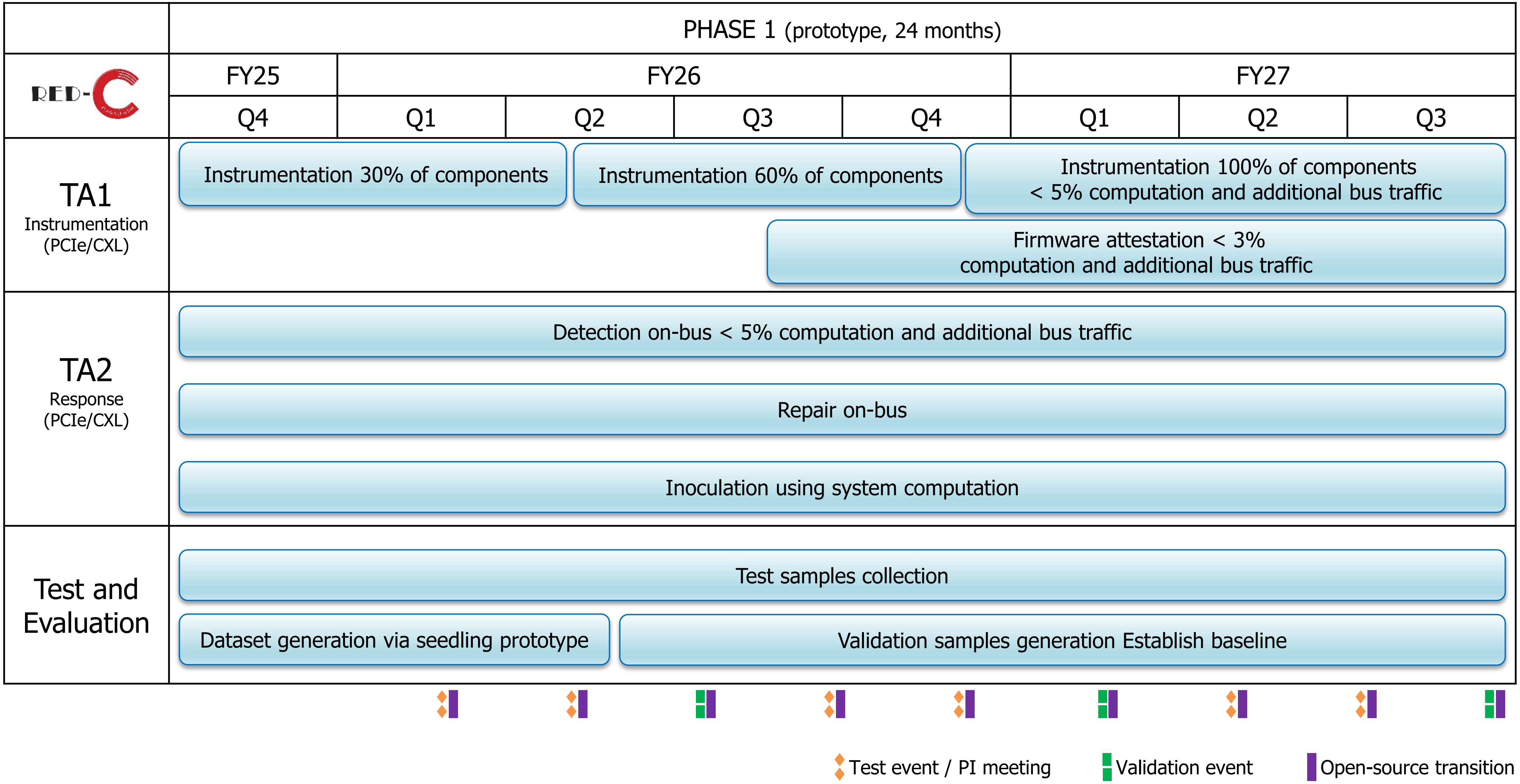
On-system automated code generation and inoculation of the running system

AMP – Assured Micropatching
LLM – Large Language Model

Red-C ecosystem of firmware and algorithms development



Program Schedule



CXL – Computer Express Link
PCIe – Peripheral Component Interconnect Express

Metrics

Metric	Phase 1
Attack detection and recovery time	Laptop PCIe* < 20 sec, < 5 min
Red-C's overhead as % of component and bus usage	Component <13%, Bus < 13%
Accuracy of detection on previously unseen samples	Baseline
Restoration quality	Critical system function* is retained and the attacker's ability to exploit the same vulnerability is removed.
Time to implement Red-C in firmware from model on a new system	Baseline manual translation with standard development workstation

* Critical system function – will be defined for each performer by the T&E team at Kickoff

* PCIe – will be defined after SRO for each bus-based system

www.darpa.mil