An Architectural Perspective on Securing Artificial Intelligence/Machine Learning (AI/ML) Enabled Systems

David McClurg
Cyber Architect
BAE Systems, San Diego CA 92127

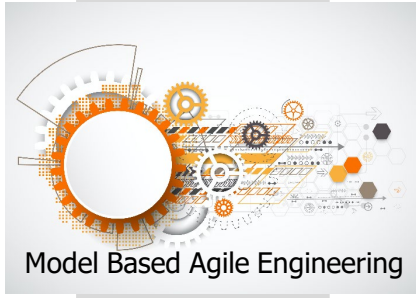
Dr. Mark Vriesenga
Chief Engineer, Information Analytics
BAE Systems, San Diego CA 92127

BAE SYSTEMS

Presenter Bio

- David McClurg
- Cyber Architect Sr
  - ISSE for on prem cloud environment
  - Application Security and Pentesting for commercial software.
- Prior Experience
  - Numerous IT positions over the last 10 years
- Bachelor of Science, Cybersecurity, McKendree University
  - Associate in Applied Science degree , Cybersecurity, John A. Logan College (John A. Logan College)
  - Security +
  - CISSP
- One of the BAE Systems Hacking club Leads

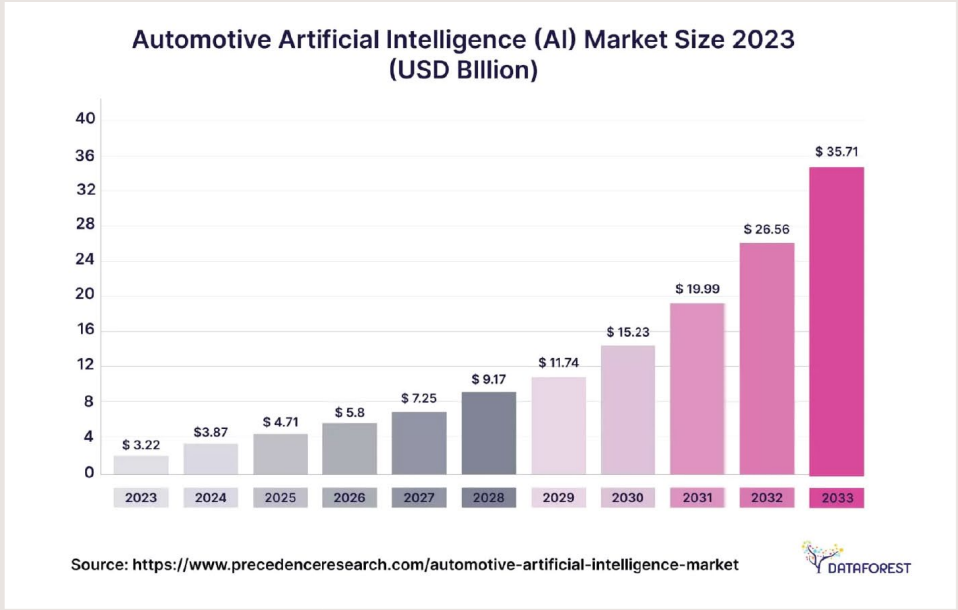
Presenter Bio



- Dr. Mark Vriesenga
  - Chief Engineer, Information Analytics
  - Global Engineering Fellow (Agile EaSA, Offensive Cyber Technology)
  - OSWP, OSCP OSCE, OSEE, CEH/CNDA, CISSP, Safe Agilest, Certified Enterprise Architect
- Background & Experience
  - 29 years at BAE Systems
    - Algorithm Developer
    - Chief Engineer, Advanced Programs
    - Business Development
    - Strategic Program Capture
    - I&S University Founder
    - Cyber Resilience Capability Group
    - Deputy Director, FAST LABS Cyber Technology
    - Model-Based Agile Engineering Capability Group (MBAE CG)
  - Offensive & Defensive Cyber Security (15 years as a SME)

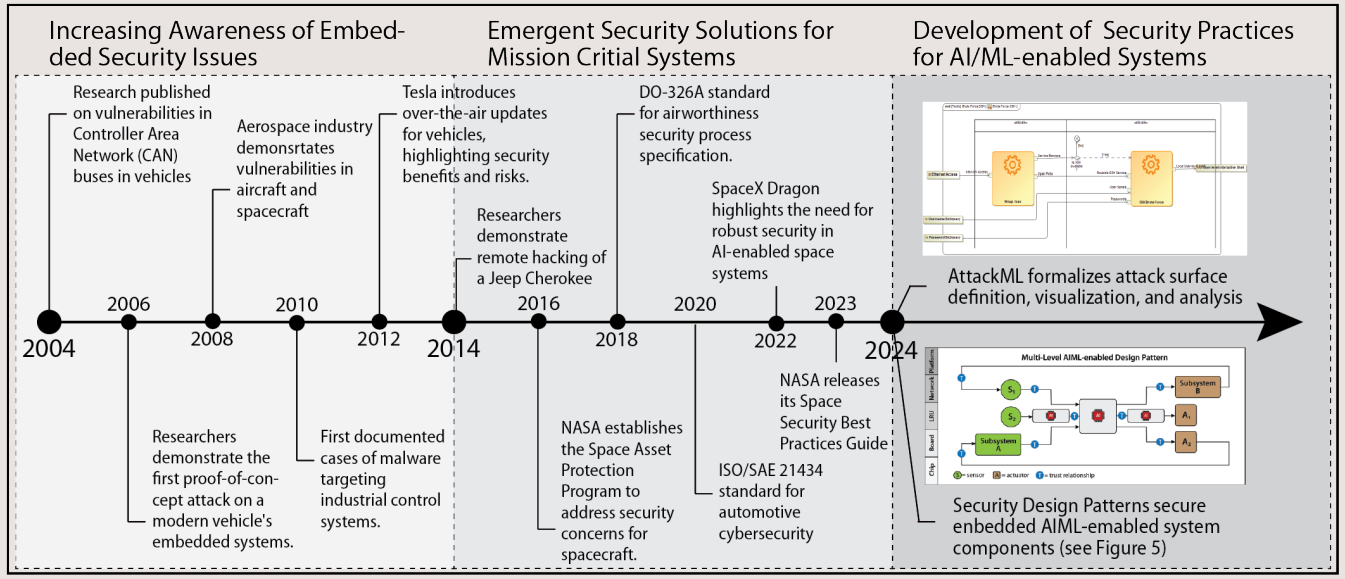
Understanding the Security Challenge

- As systems engineers, we face significant security challenges when integrating AI/ML-enabled components into complex, dynamic architectures.
- These architectures, increasingly prevalent across ground, air, and space domains, require new security considerations.
- For example, modern electric vehicles (EVs) provide AI/ML-enabled autonomy and decision-making capabilities, including self-driving and self-parking functions.
- With these advanced functions comes the potential for adversaries to disrupt operations through cyber-attacks targeting AI/ML-enabled system elements.



History of Prior Work

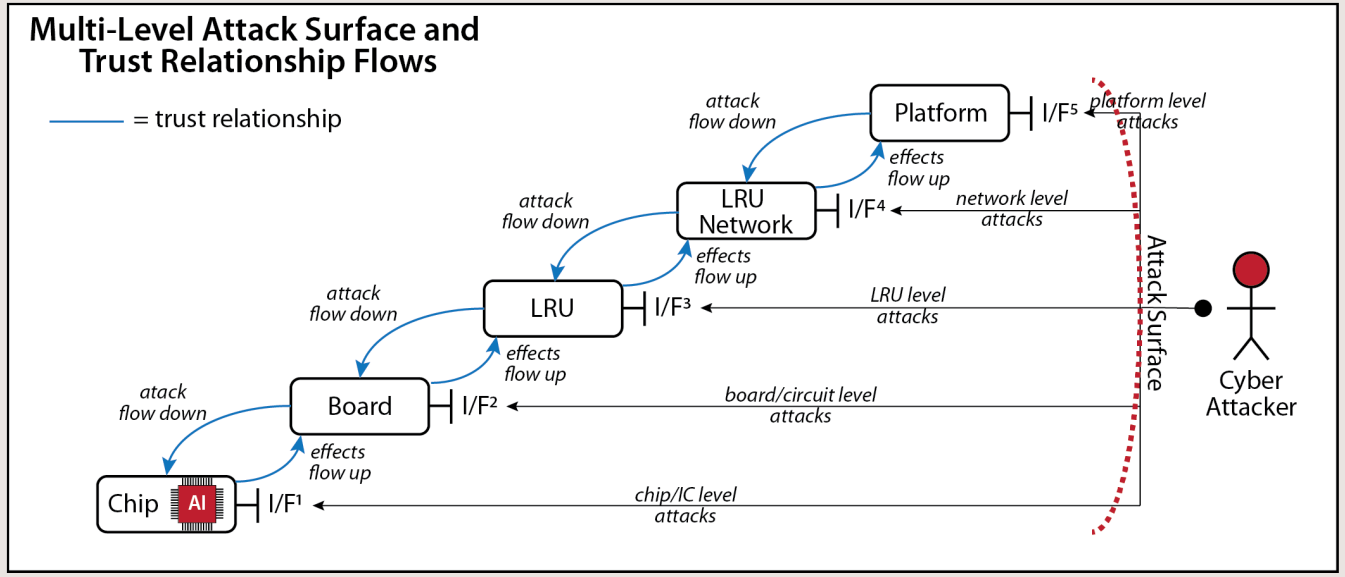
- The history of securing cyber-physical systems is over 20 years old and was triggered by the integration of microcontrollers, software, and networked components into product architectures.
- As cyber-physical systems became more complex, the security challenges grew, leading to significant research efforts to safeguard these product architectures.
- The topic of securing embedded systems is gaining momentum again with the rapid adoption of AI/ML-enabled system elements, which bring both advanced capabilities and expanded attack surfaces.



BAE SYSTEMS

Architectural Framework for Security Analysis

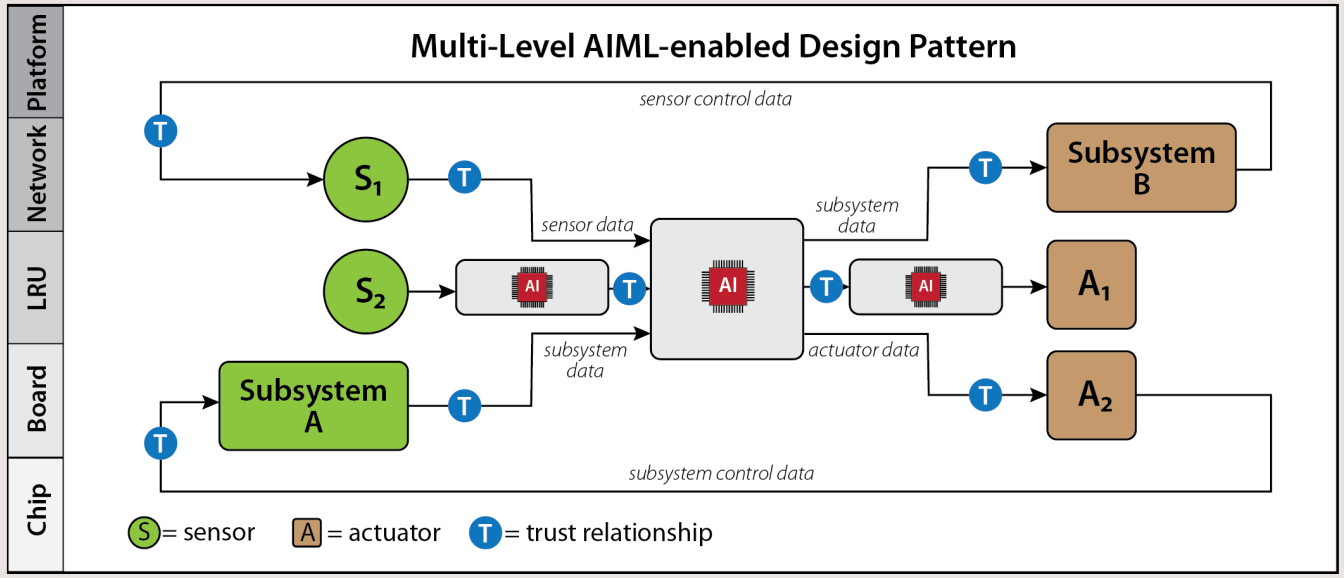
- As a rule, cyber attackers exploit trust relationships between system elements to access critical mission functions and to perform cyber attacks.
- In both commercial and military platforms, trust relationships connect architectural scales
  - Platform ←→Network ←→LRU ←→Board ←→Chip
- In this framework
  - Attacks flow from level to level along trust relationships
  - AI/ML exploitation occurs at the chip (software)
  - Effects flow upward to the targeted system element
- Notably, the union of AI/ML attacks across architecture scales forms the new and expanded attack surface



BAE SYSTEMS

Design Pattern for Developing Solutions

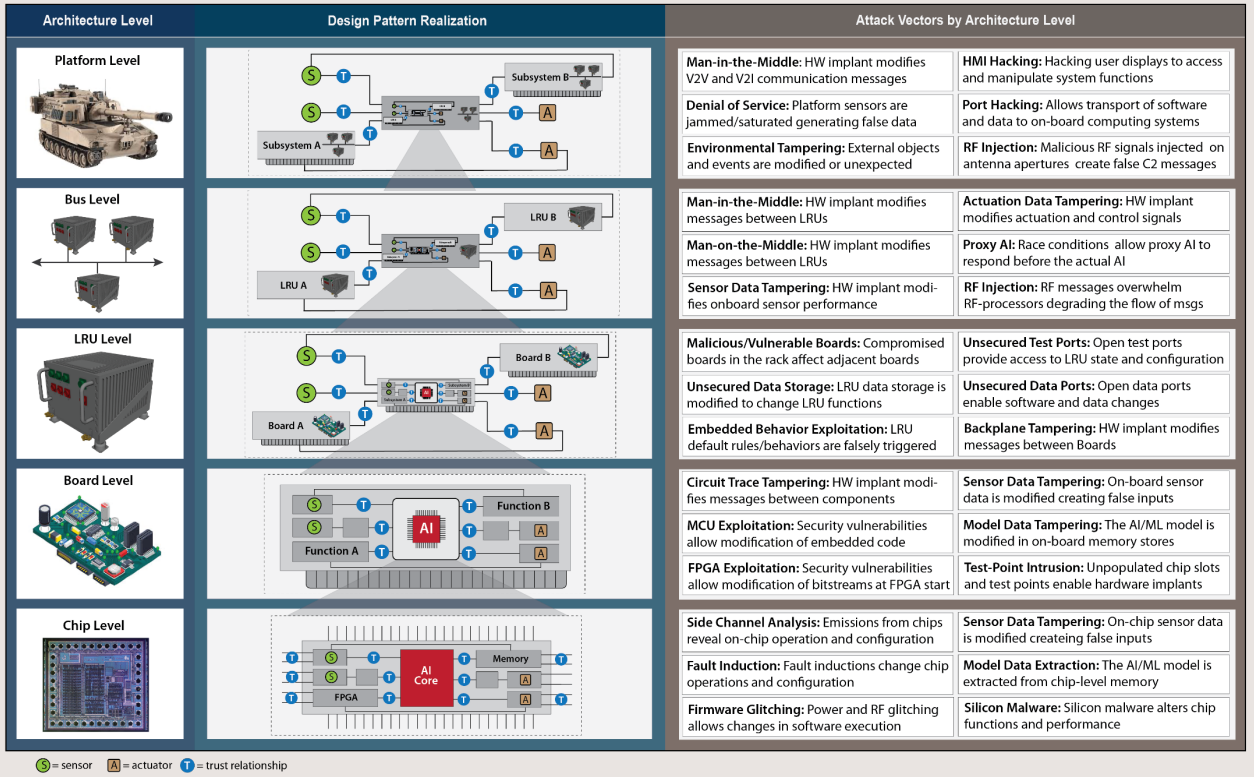
- Analysis of AI/ML-enabled security issues across the five framework scales produces a common design pattern for assessing security issues.
- In this design pattern
  - Sensors (S1, S2) provide sensor data from the onboard (internal) and offboard (external) environments.
  - AI/ML-enabled system components perform data conditioning, sensor data integration, decision-making, and actuator data conditioning.
  - Actuators (A1, A2) receive commands from the AI/ML and generate control signals for connected elements.
- It's important to note that most connections are unsecured trust relationships between components in modern systems.



Unsecured trust relationships provide a potential avenue for attack and should be a key focus in our security measures.

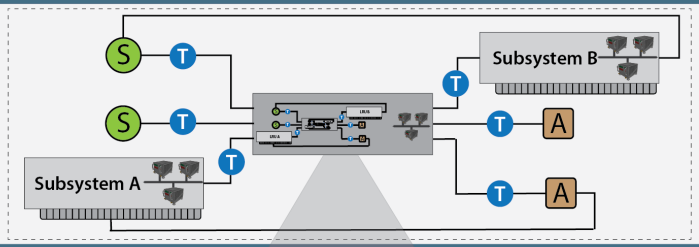
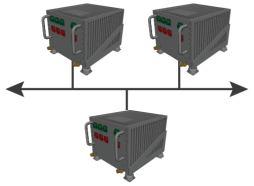
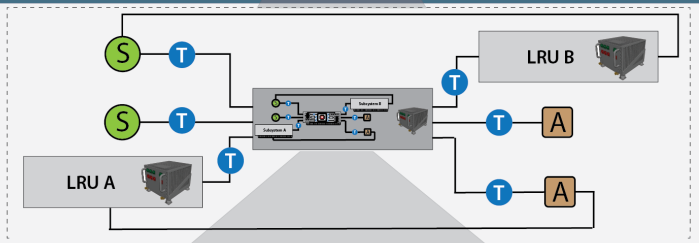
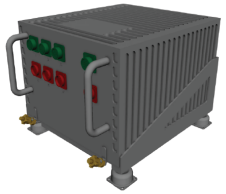
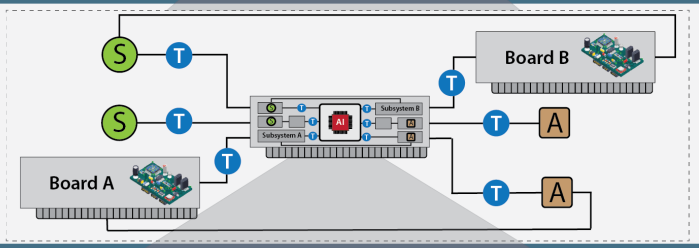
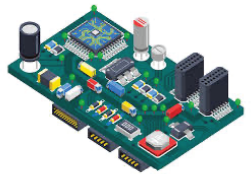
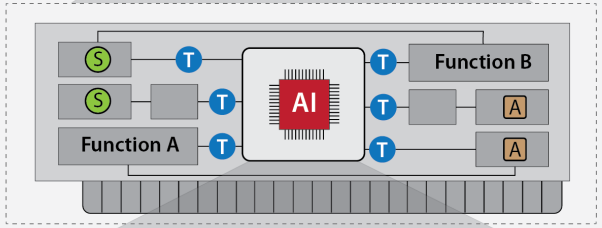
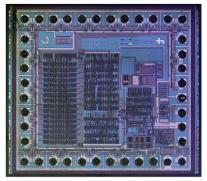
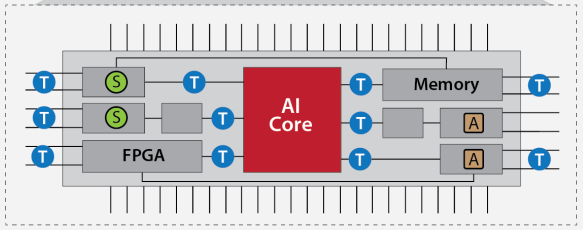
Cross Layer Attack Framework for on AI/ML-Enabled Systems

- The cross-layer attack framework identifies cyber attacks affecting AI/ML-enabled components at each framework level.
- In many cases, the attacks focus on manipulating or modifying sensor data inputs to the AI/ML algorithm, causing it to generate outputs determined by the attacker's intent.
- These erroneous inputs change system performance as their effects propagate to actuators and subsystems throughout the system architecture.



(See next slide)

BAE SYSTEMS

Architecture Level	Design Pattern Realization	Attack Vectors by Architecture Level	
Platform Level 		**Man-in-the-Middle**: HW implant modifies V2V and V2I communication messages	**HMI Hacking:** Hacking user displays to access and manipulate system functions
		Denial of Service: Platform sensors are jammed/saturated generating false data	**Port Hacking:** Allows transport of software and data to on-board computing systems
		Environmental Tampering: External objects and events are modified or unexpected	**RF Injection:** Malicious RF signals injected on antenna apertures create false C2 messages
Bus Level 		**Man-in-the-Middle:** HW implant modifies messages between LRUs	**Actuation Data Tampering:** HW implant modifies actuation and control signals
		Man-on-the-Middle: HW implant modifies messages between LRUs	**Proxy AI:** Race conditions allow proxy AI to respond before the actual AI
		Sensor Data Tampering: HW implant modifies onboard sensor performance	**RF Injection:** RF messages overwhelm RF-processors degrading the flow of msgs
LRU Level 		**Malicious/Vulnerable Boards:** Compromised boards in the rack affect adjacent boards	**Unsecured Test Ports:** Open test ports provide access to LRU state and configuration
		Unsecured Data Storage: LRU data storage is modified to change LRU functions	**Unsecured Data Ports:** Open data ports enable software and data changes
		Embedded Behavior Exploitation: LRU default rules/behaviors are falsely triggered	**Backplane Tampering:** HW implant modifies messages between Boards
Board Level 		**Circuit Trace Tampering:** HW implant modifies messages between components	**Sensor Data Tampering:** On-board sensor data is modified creating false inputs
		MCU Exploitation: Security vulnerabilities allow modification of embedded code	**Model Data Tampering:** The AI/ML model is modified in on-board memory stores
		FPGA Exploitation: Security vulnerabilities allow modification of bitstreams at FPGA start	**Test-Point Intrusion:** Unpopulated chip slots and test points enable hardware implants
Chip Level 		**Side Channel Analysis:** Emissions from chips reveal on-chip operation and configuration	**Sensor Data Tampering:** On-chip sensor data is modified createing false inputs
		Fault Induction: Fault inductions change chip operations and configuration	**Model Data Extraction:** The AI/ML model is extracted from chip-level memory
		Firmware Glitching: Power and RF glitching allows changes in software execution	**Silicon Malware:** Silicon malware alters chip functions and performance

S = sensor A = actuator T = trust relationship

Cross Layer Attacks on AI/ML-Enabled Systems

Man in the Middle (MITM): MITM attacks intercept and modify communications between system elements connected by an unsecured trust relationship.

For example, at the Bus Level, hardware implants intercept and modify CAN bus messages, allowing the attacker to influence the system's operational state.

Sensor Data Tampering : By altering sensor data, the attacker can feed false information into the system, leading to incorrect assessments of the platform's environment or operations.

For Example, at the Bus level, tampering with incoming sensor data to AIML, which is at the Chip level, falsifies the state of the environment to achieve an attacker's objective.

Denial of Service (DoS): DoS attacks overload sensors to produce false readings or to disable the sensor and deny state data to platform controllers.

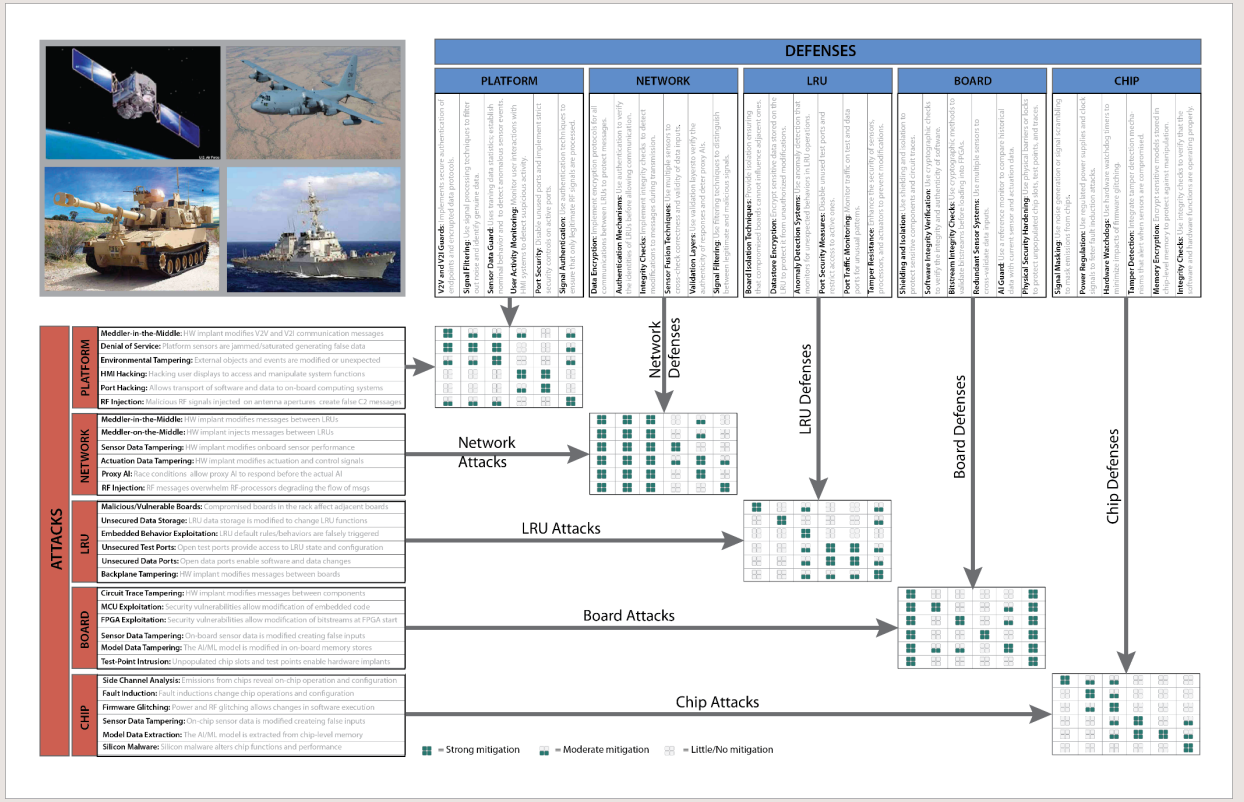
For example, at the Platform Level, spamming AIML with "Chaff Data" to attack the sensors. Kinetic attacks on sensors at Platform and Bus levels cause degradation of AIML functions.

Actuator Data Tampering: By altering AI/ML-provided actuation signals, the attacker can change the behavior of critical systems such as steering, braking, or propulsion, potentially causing malfunctions or dangerous conditions.

For example, at the Bus level, Proxy AI abuses the trust relationship of signals that would be coming from the AI Chip to change physical state (i.e., braking)

Cross Layer Defenses for AI/ML-Enabled Systems

- If the trust relationships are open and connected, each previously identified attack may affect the operation of AI/ML-enabled system elements, thereby allowing the attacker to influence the system.
- In such cases, it is crucial to secure trust relationships by implementing security measures and protocols that protect the information flows leading to AI/ML-enabled system elements.
- This protection is achieved using security controls, some of which are described in the Risk Management Framework , and others are unique to the architecture level.



(See next slide)

BAE SYSTEMS

DEFENSES

PLATFORM

- **V2V and V2I Guards:** implements secure authentication of endpoints and encrypted data protocols.
- **Signal Filtering:** Use signal processing techniques to filter out noise and identify genuine data.
- **Sensor Data Guard:** Uses traning data statistics; establish nominal behavior and to detect anomalous sensor events.
- **User Activity Monitoring:** Monitor user interactions with HMI systems to detect suspicious activity.
- **Port Security:** Disable unused ports and implement strict security controls on active ports.
- **Signal Authentication:** Use authentication techniques to ensure that only legitimate RF signals are processed.

NETWORK

- **Data Encryption:** Implement encryption protocols for all communications between LRUs to protect messages.
- **Authentication Mechanisms:** Use authentication to verify the identities of LRUs before allowing communication.
- **Integrity Checks:** Implement integrity checks to detect modifications to messages during transmission.
- **Sensor Fusion Techniques:** Use multiple sensors to cross-check correctness and validity of data inputs.
- **Validation Layers:** Use validation layersto verify the authenticity of responses and deter proxy AIs.
- **Signal Filtering:** Use filtering techniques to distinguish between legitimate and malicious signals.

LRU

- **Board Isolation Techniques:** Provide isolation ensuring that compromised boards cannot influence adjacent ones.
- **Datastore Encryption:** Encrypt sensitive data stored on the LRU to protect it from unauthorized modifications.
- **Anomaly Detection Systems:** Use anomaly detection that monitors for unexpected behaviors in LRU operations.
- **Port Security Measures:** Disable unused test ports and restrict access to active ones.
- **Port Traffic Monitoring:** Monitor traffic on test and data ports for unusual patterns.
- **Tamper Resistance:** Enhance the security of sensors, processors, and actuators to prevent modifications.

BOARD

- **Shielding and Isolation:** Use shielding and isolation to protect sensitive components and circuit traces.
- **Software Integrity Verification:** Use cryptographic checks to verify the integrity and authenticity of software.
- **Bitstream Integrity Checks:** Use cryptographic methods to validate bitstreams before loading into FPGAs.
- **Redundant Sensor Systems:** Use multiple sensors to cross-validate data inputs.
- **AI Guard:** Use a reference monitor to compare historical data with current sensor and actuation data.
- **Physical Security Hardening:** Use physical barriers or locks to protect unpopulated chip slots, test points, and traces.

CHIP

- **Signal Masking:** Use noise generation or signal scrambling to mask emissions from chips.
- **Power Regulation:** Use regulated power supplies and clock signals to feter fault induction attacks.
- **Hardware Watchdogs:** Use hardware watchdog timers to minimize impacts of firmware glitching.
- **Tamper Detection:** Integrate tamper detection mechanisms that alert when sensors are compromised.
- **Memory Encryption:** Encrypt sensitive models stored in chip-level memory to protect against manipulation.
- **Integrity Checks:** Use integrity checks to verify that the software and hardware functions are operating properly.

ATTACKS

PLATFORM

- **Meddler-in-the-Middle:** HW implant modifies V2V and V2I communication messages
- **Denial of Service:** Platform sensors are jammed/saturated generating false data
- **Environmental Tampering:** External objects and events are modified or unexpected
- **HMI Hacking:** Hacking user displays to access and manipulate system functions
- **Port Hacking:** Allows transport of software and data to on-board computing systems
- **RF Injection:** Malicious RF signals injected on antenna apertures create false C2 messages

NETWORK

- **Meddler-in-the-Middle:** HW implant modifies messages between LRUs
- **Meddler-on-the-Middle:** HW implant injects messages between LRUs
- **Sensor Data Tampering:** HW implant modifies onboard sensor performance
- **Actuation Data Tampering:** HW implant modifies actuation and control signals
- **Proxy AI:** Race conditions allow proxy AI to respond before the actual AI
- **RF Injection:** RF messages overwhelm RF-processors degrading the flow of msgs

LRU

- **Malicious/Vulnerable Boards:** Compromised boards in the rack affect adjacent boards
- **Unsecured Data Storage:** LRU data storage is modified to change LRU functions
- **Embedded Behavior Exploitation:** LRU default rules/behaviors are falsely triggered
- **Unsecured Test Ports:** Open test ports provide access to LRU state and configuration
- **Unsecured Data Ports:** Open data ports enable software and data changes
- **Backplane Tampering:** HW implant modifies messages between boards

BOARD

- **Circuit Trace Tampering:** HW implant modifies messages between components
- **MCU Exploitation:** Security vulnerabilities allow modification of embedded code
- **FPGA Exploitation:** Security vulnerabilities allow modification of bitstreams at FPGA start
- **Sensor Data Tampering:** On-board sensor data is modified creating false inputs
- **Model Data Tampering:** The AI/ML model is modified in on-board memory stores
- **Test-Point Intrusion:** Unpopulated chip slots and test points enable hardware implants

CHIP

- **Side Channel Analysis:** Emissions from chips reveal on-chip operation and configuration
- **Fault Induction:** Fault inductions change chip operations and configuration
- **Firmware Glitching:** Power and RF glitching allows changes in software execution
- **Sensor Data Tampering:** On-chip sensor data is modified createing false inputs
- **Model Data Extraction:** The AI/ML model is extracted from chip-level memory
- **Silicon Malware:** Silicon malware alters chip functions and performance

Network Defenses

LRU Defenses

Board Defenses

Chip Defenses

Network Attacks

LRU Attacks

Board Attacks

Chip Attacks

= Strong mitigation = Moderate mitigation = Little/No mitigation

Cross Layer Defenses for AI/ML-Enabled Systems

Encryption: Implementation of encryption for message communication occurs at multiple layers since attacks originate at any layer of the system design.

For Example, Attacker conducts MitM to achieve secondary effect, this fails as MitM is unable to spoof trustworthy messages.

Guards: Reference monitor to compare historical data with current sensor and actuation data.

For Example, MitM attacks at the Bus layer produce "Chaff" data to blind Platform AI, resulting in a DoS attack. AI Guards can reject input data reaching the AI core, breaking the MiTM attack.

Integrity checks: All platform levels have methods for performing integrity checks of the platform's data flows, software, messages, and physical configuration.

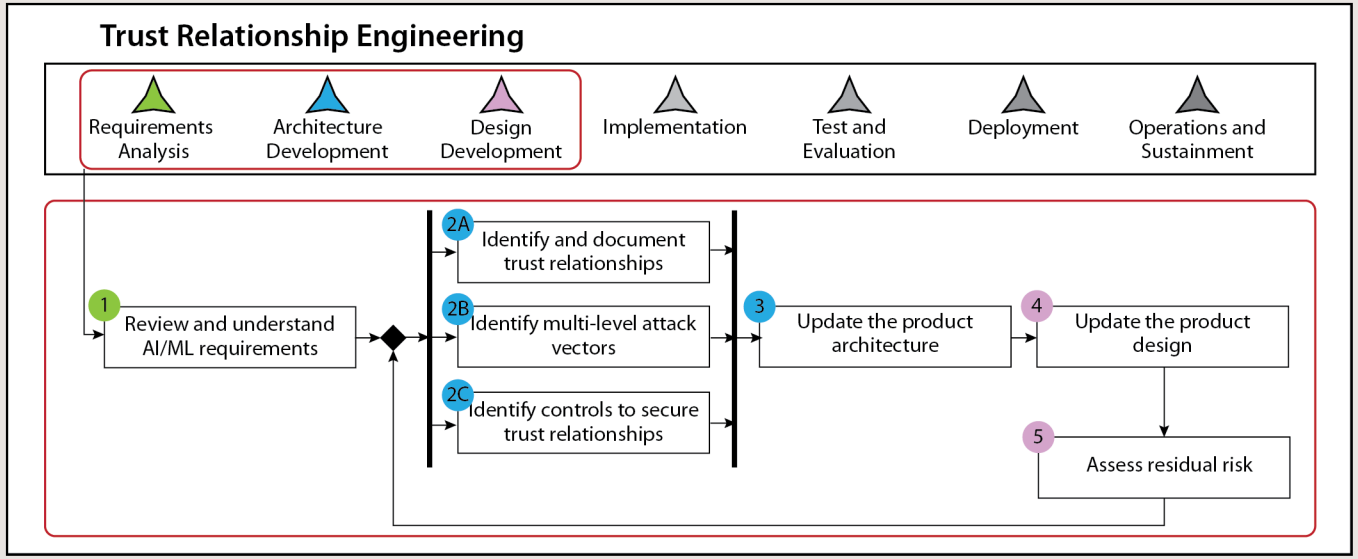
For Example, As a processing board boots, it validates its software and bitstream data, ensuring that the onboard memory stored has not been corrupted.

Authentication: Device-to-device authentication at the end-point of each trust relationship validates the identity of components on the communication pathway.

For example, a proxy AI operating at the bus level may exploit race conditions and respond ahead of the AI platform. Point-to-point authentication closes off the race condition, thereby breaking the attack chain.

Application Process

- The application process augments existing engineering practices, which are broadly applicable across product developers.
- The heart of the approach is identifying inappropriate trust relationships that allow attackers to move horizontally and vertically within a system architecture.
  - Attacks → Trust Relationships → Defenses
- Defensive controls are mapped to trust relationships, focusing on breaking the attack vector.

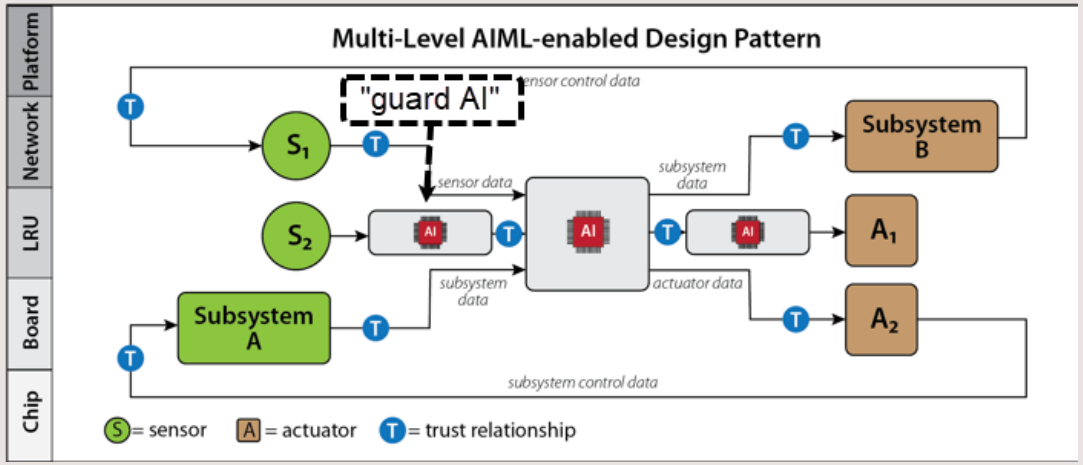


NOTE: This approach to identifying and managing security trust relationships is also the correct way to implement Zero Trust and fundamentally changes how we approach embedded product security!

Real-World Applications

- BAE Systems Cyber Resilience Capability Group(CRCG) applies its Cyber Systems Engineering(CSE) methodology to harden our electronic warfare products against cyber attacks.
- The described capability for securing AI/ML-enabled systems is the next increment of CSE capability.
- Using trust engineering, a hierarchical analysis framework, and a common design pattern, we tailor specific attacks and defense appropriate to each products to secure AI/ML-enabled components.

- Using our research results, our next step is to explore the implementation of a "guard AI" to secure the trust relationship for signals entering and leaving the AI/ML-enabled systems elements and providing maximum security to the platform and users.



BAE SYSTEMS

Questions?

BAE SYSTEMS