SAND2016-10581PE

Exceptional service in the national interest

Sandia National Laboratories

Trusted System Development and Research Challenges

Oct 19, 2016

Brandon Eames, PhD
Technical Staff
Sandia National Laboratories
bkeames@sandia.gov

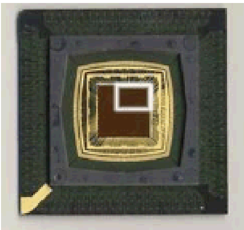
U.S. DEPARTMENT OF ENERGY NNSA National Nuclear Security Administration

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Trust in Microelectronics Based Systems

Sandia National Laboratories



- The US Government and industries develop microelectronics-based systems for ensuring safety and security
  - Military systems, satellites, cyber infrastructure, critical infrastructure (e.g. power grid), etc.

- Can adversaries manipulate these systems as they are developed? What would the impact be?
- Can these systems be ***trusted*** to perform their intended function?

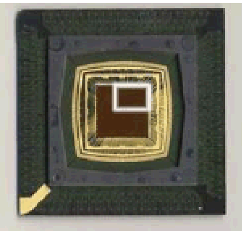
How vulnerable are systems to development time manipulation?

Trust in Microelectronics Based Systems

Sandia National Laboratories

- The USG develops microelectronics-based systems for ensuring safety and security
- Can these systems be ***trusted*** to perform their intended function?



- Trust is a system-level problem, and must consider the entire ***system-development lifecycle***

System Development Lifecycle

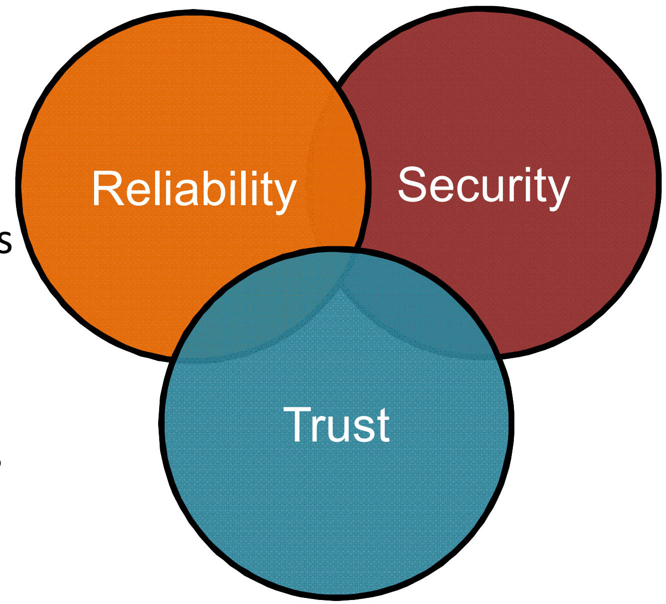
Concept	Design	Implementation	Manufacturing	Test	Deployment	Maintenance

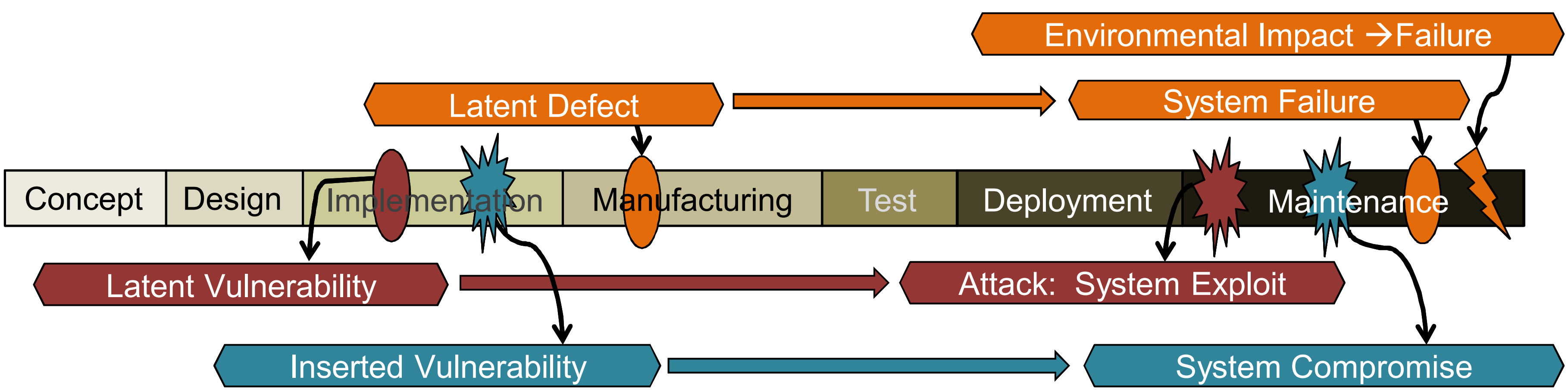
Reliability vs. Security vs. Trust

- **Reliability**: The probability that an item will perform a required function under stated conditions for a stated period of time
  - Premature System Failure → Design for Reliability

$$R \approx 1 - \left[\left(1 - \prod_{i=1}^{5}(1-J_i)\right) + \sum_{i=1}^{7} K_i^2 + 2K_2\left(K_3 + \sum_{i=5}^{7} K_i\right)\right]$$

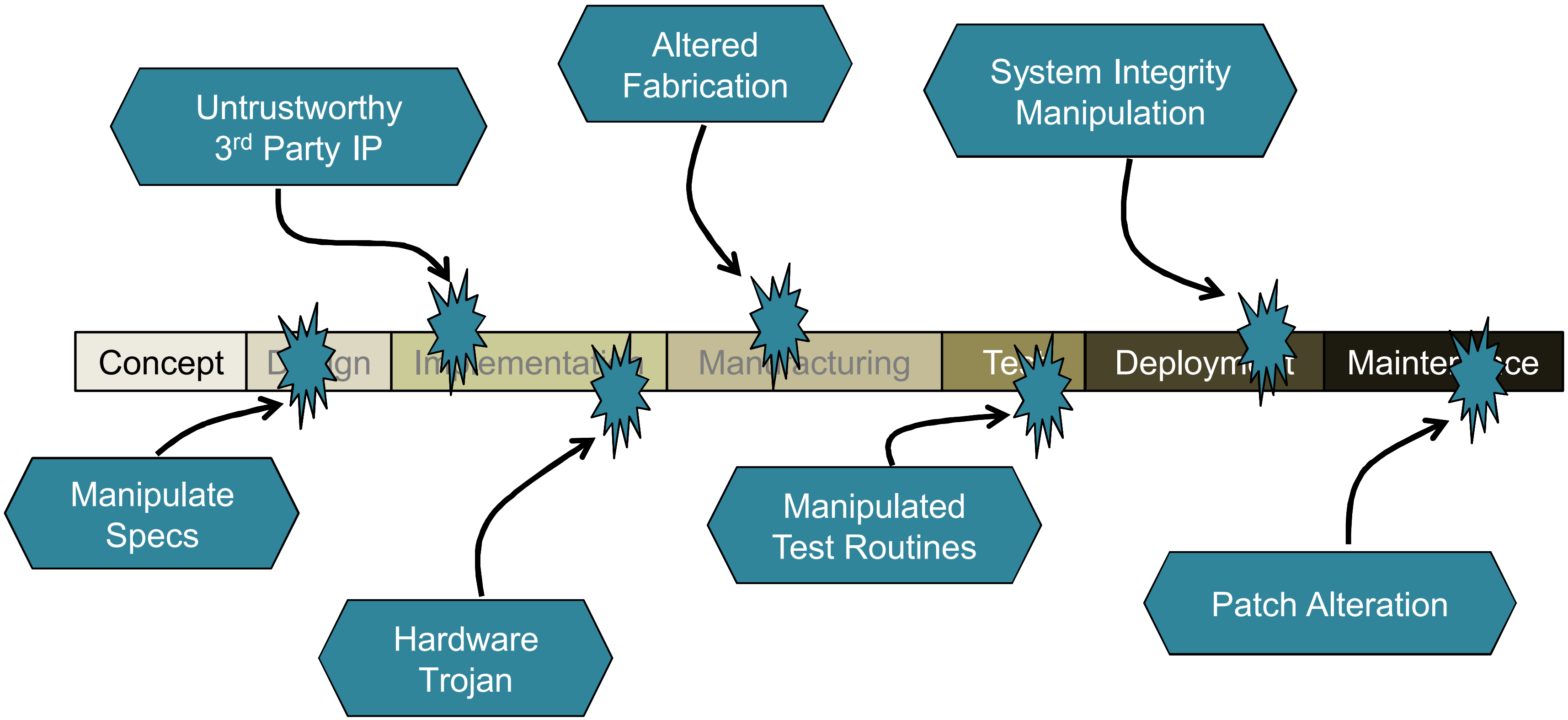
- **Security**: The protection of systems from theft or damage …, as well as from disruption … of the services they provide.
  - System Exploitation → Design for Security
- **Trust**: The confidence in … secur[ing] national security systems by assessing the integrity of the people and processes used to design, generate, manufacture, and distribute … [systems]
  - System Compromise → **Design for Trust**





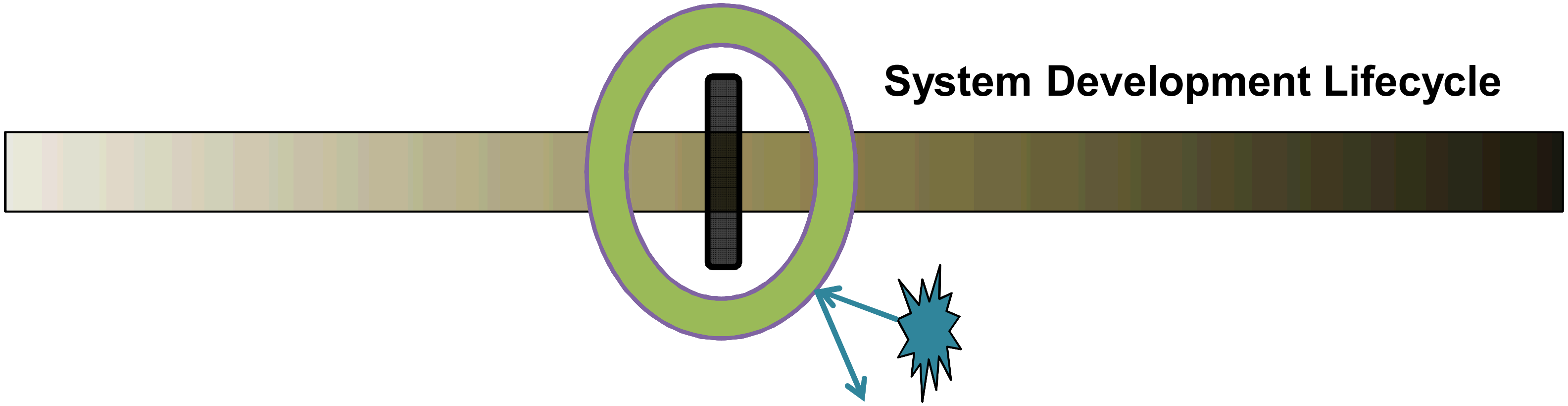
Where Trust Breaks Down

Sandia National Laboratories



Adversaries can potentially manipulate development at any point

Current Approach to Trusted System Development

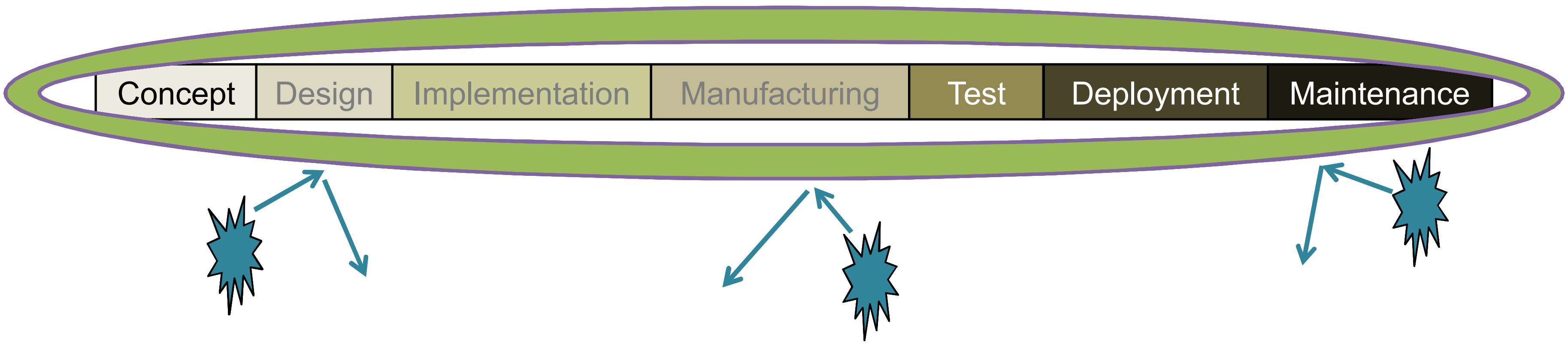


Isolate Development Process to Prevent Attacks

- Keep the attacker from manipulating the system / development process
- Process-based approaches: control information flow, control supply chain, government-owned manufacturing etc.

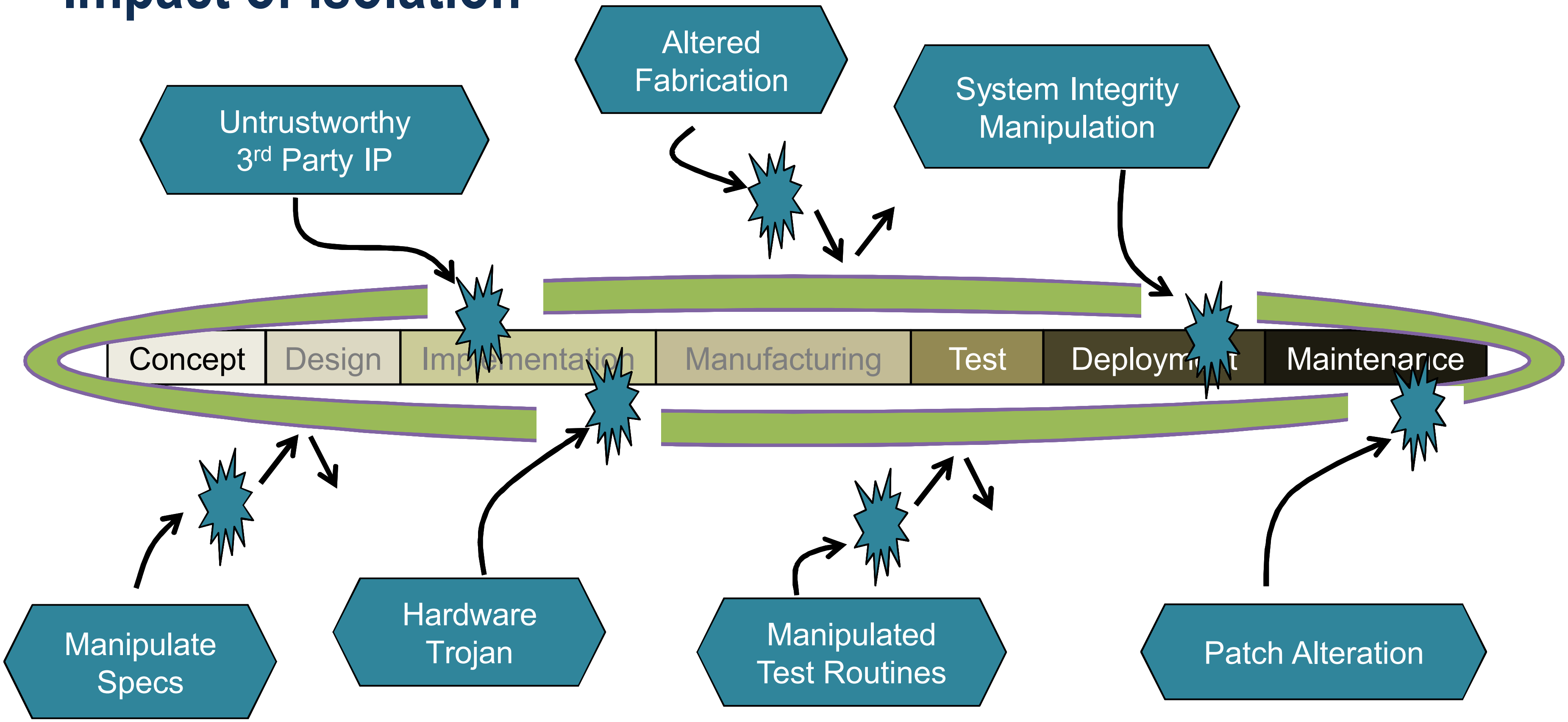
- Examples:
  - Trusted Foundry Program: Certification process to establish domestic, isolated microelectronics fabrication
    - Ensure integrity, availability of microelectronics fabrication
  - Isolated computer networks
  - Vetted design teams

Impact of Isolation



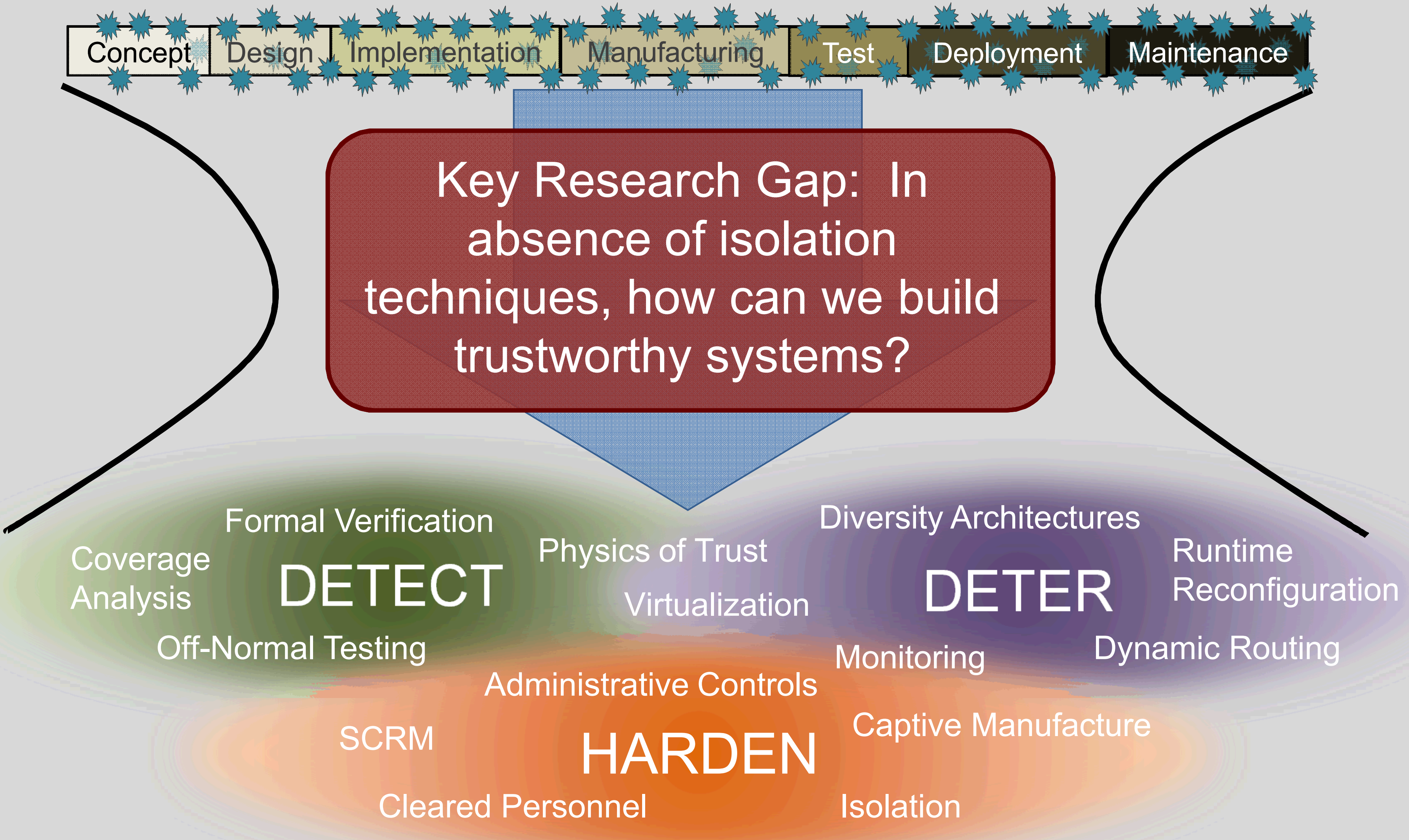
- Isolation can be highly effective as an adversarial deterrent
- Can we fully isolate the complete system development lifecycle?
  - Captive fabrication (trusted foundry) addresses only one aspect of the development process
  - Completely isolated development processes are VERY expensive
    - Consider cost of leading edge microelectronics fabrication facility
  - Systems use COTS components, development tools
  - Insider threat?

Impact of Isolation



- Currently identified isolation techniques can be highly effective at deterring many paths of adversary access
- ***Gaps Remain: Practicality of real system development precludes complete isolation***

Trust Research



The Challenge With Trust

Is my system Trustworthy?

How much "trust" do I need?

Cost vs. Benefit of trust?

Trust vs. SWaP?

What should I really be worried about?

What mitigations make sense?

How do I engineer Trustworthy systems?

Is Trusted manufacturing enough?

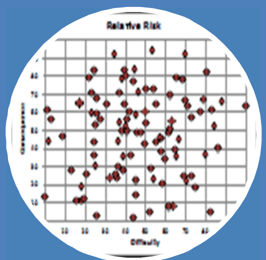
When are we done?

?

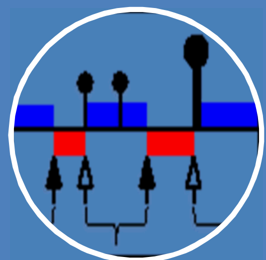
Why is Sandia Interested in Trust Research?

- National research problem with significant scope and complexity
- USG concern for trust codified in current policies, but
  - No approach exists forderiving, addressing quantitative trust requirements
- Research heritage studying advanced persistent threats (APTs)
- *Lack of comprehensive, cohesive solution for analyzing and developing trustworthy systems*

USG Trust Policies:
- DoDI 5200.44
- DOE 452.1E, 452.4C
- NAP-24A, attachment 4

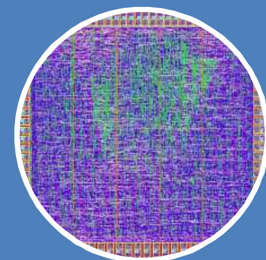


LDRD: Risk Assessment Methodology (ILS): Expert-based Risk Evauation

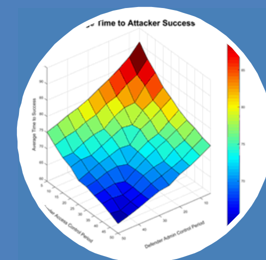


Moving Target Defense in Cyber Systems: Game Theory Analysis / PLADD

LDRD: Supply Chain Analytics: Modeling Development Attacks



Trust in FPGAs Studies: Process Models, Trust Assessment, Attack Graphs



LDRD: FTA (FY16) Fundamental Trust Analysis: Game Theory for Trust

RECENT/ONGOING TRUST RESEARCH AT SANDIA

Risk Analysis: Identify Areas of Highest Concern

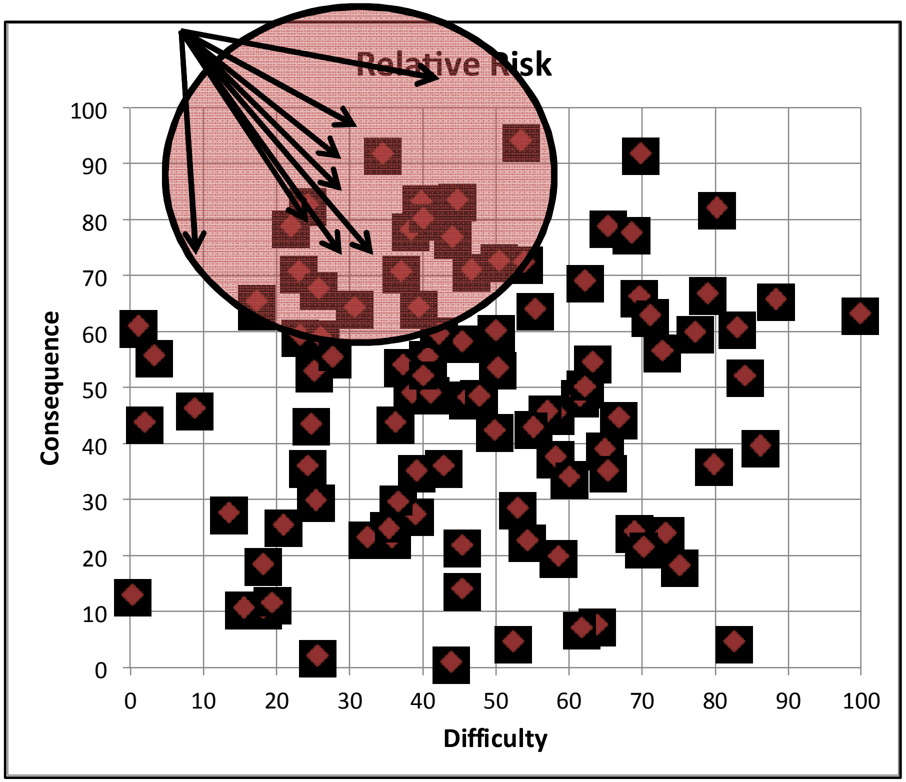
Identify Potential Risks

↓

Evaluate Attack Difficulty

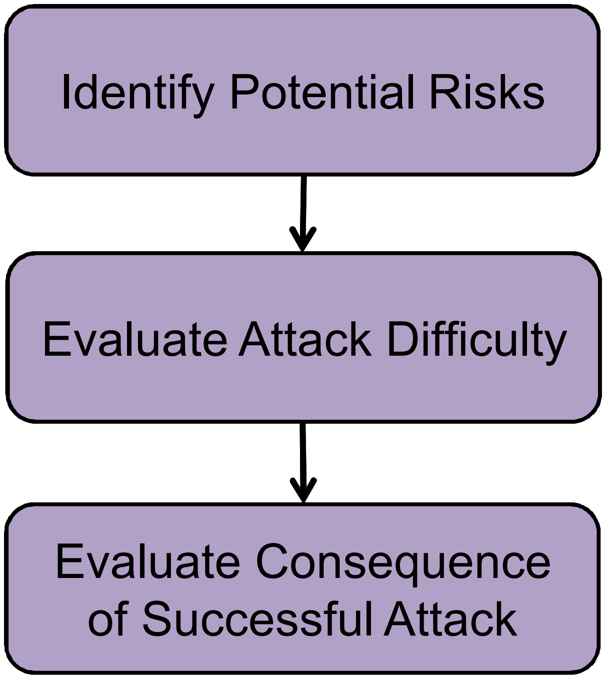
↓

Evaluate Consequence of Successful Attack

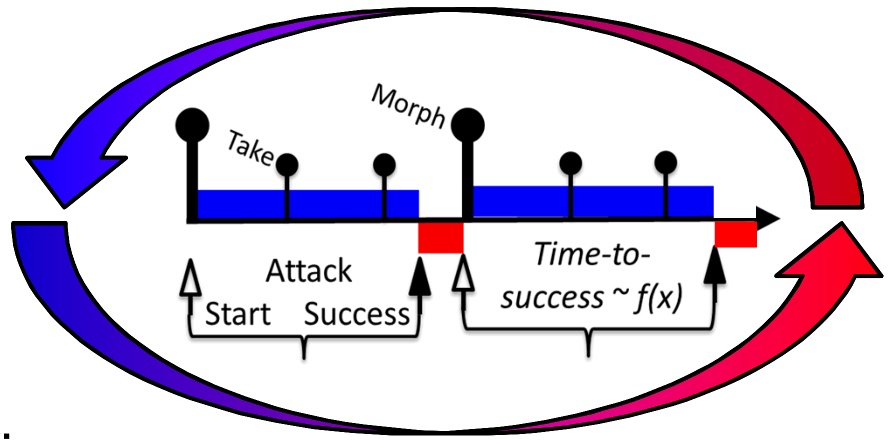


Relative Risk Assessment

- Assessing Difficulty
  - Subject Matter Expert (SME)-based evaluation
  - Domain-agnostic rubric for supporting assessment
  - 13 different dimensions for difficulty assessment
    - E.g. Size of outsider team, level of stealth required, complexity of attack
    - SMEs assign 1-5 ranking in each category
- Challenge: subjective analysis
- Repeatability?
- Science-based assessment?



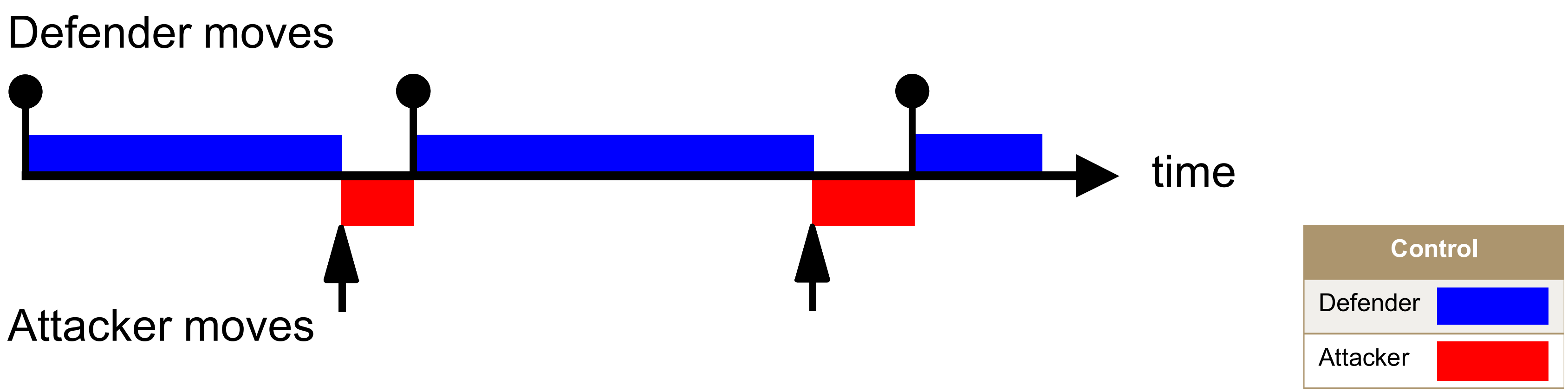
Game Theoretic Analysis: Why?



- Game Theory:
  - "The study of ***mathematical models of conflict and cooperation*** between intelligent, rational decision-makers"[1]
  - Initially developed by von Neumann and Morgenstern in 1944
  - Nobel Prizes awarded for work on game theory: 2014, 2007, 2005, 1996, 1995, 1994, 1972, 1970
- Why Game Theory for Trust?
  - Trust is concerned with the **risk of potential interaction** between **adversaries and system developers** and development processes
  - Game Theory allows explicit representation and **evaluation of dynamic interaction** between attacker and defender

1. Myerson, B. "Game Theory: Analysis of Conflict". Harvard University Press

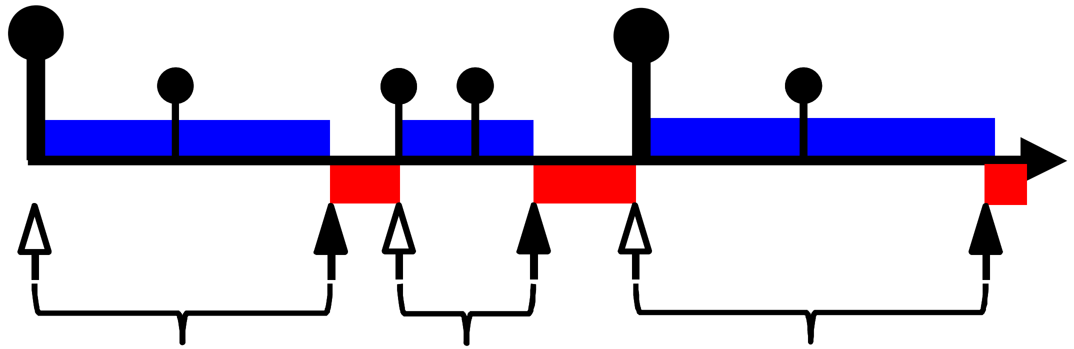
FlipIt: A Game Theoretic Model to Investigate Cyber Defense Effectiveness



FlipIt Constructs

- Two players (defender and attacker)
- A single contested resource
- Player moves seize the resource
- Moves incur a cost
- Strategy consists of move timing
- Single defender move (take)
- Limited player information
- Utility = Control Time - Cost

Probabilistic, Learning Attacker, Dynamic Defender (PLADD) Model

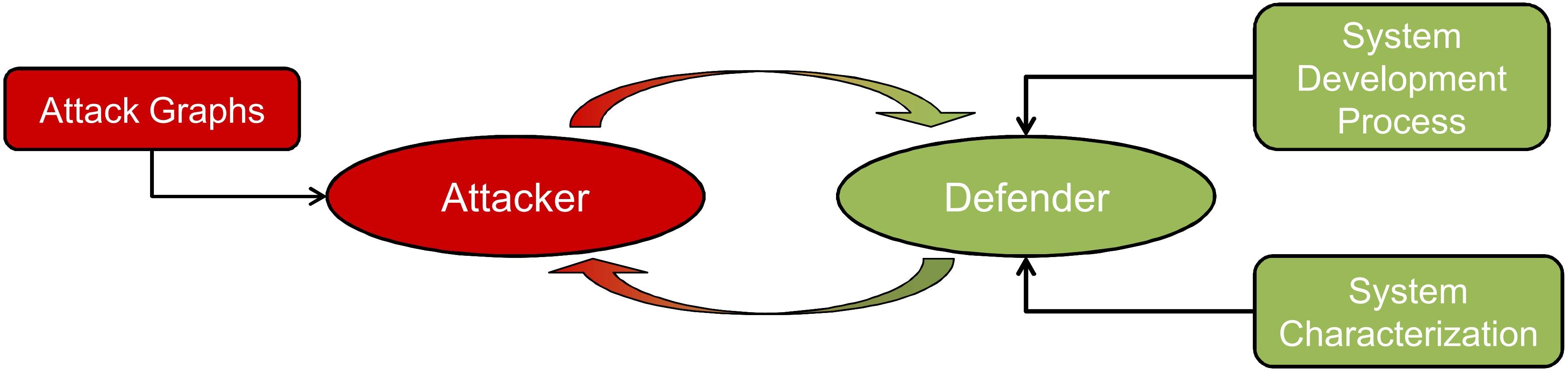


PLADD Model for Analysis

- Represent Attacker-Defender interaction as contention for a single resource
- Defender executes periodic actions
  - Each action wrests control from attacker
- Attacker actions wrest control from defender, after a random period of time
- Attack cost: fixed to initiate + variable cost proportional to time-to-success
- As attacker repeats attacks, they become more efficient.
- Special defender "morph" move resets attacker learning
- Goal: determine defender strategies that drive attacker costs to be prohibitive

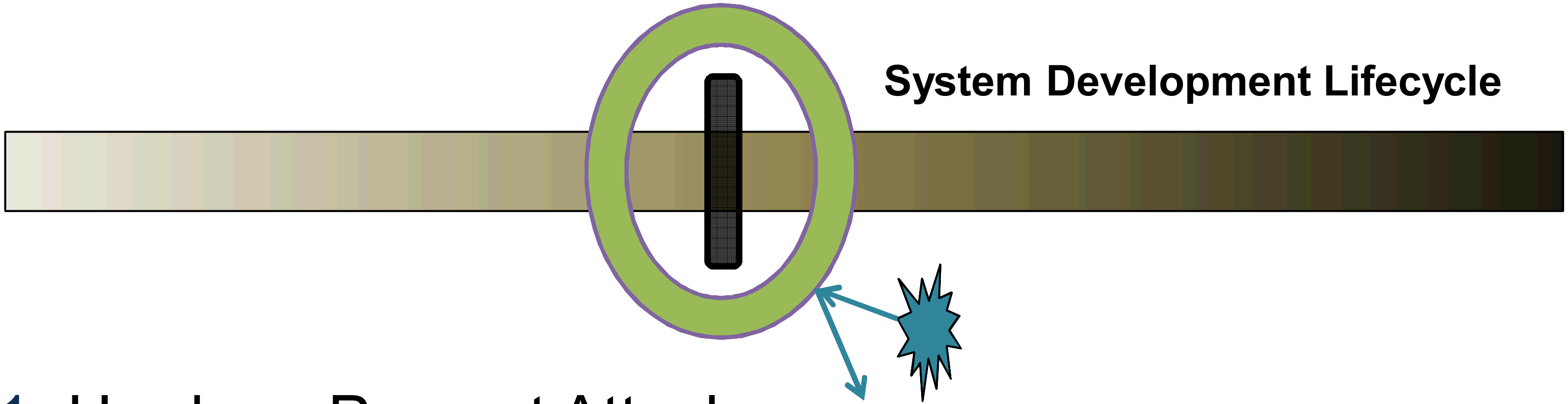
Fundamental Trust Analysis

- Amalgamation of *game theory* with *relative risk assessment* to model full lifecycle trust concerns, and objectively evaluate system trustworthiness
  - Incorporate game theory, risk assessment, resiliency analysis, optimization and supply chain analytics
  - Apply PLADD to trust analysis
- Goal: Empower decision makers to make quantitative, science-based tradeoff decisions about trust



Approaches to Trusted System Development

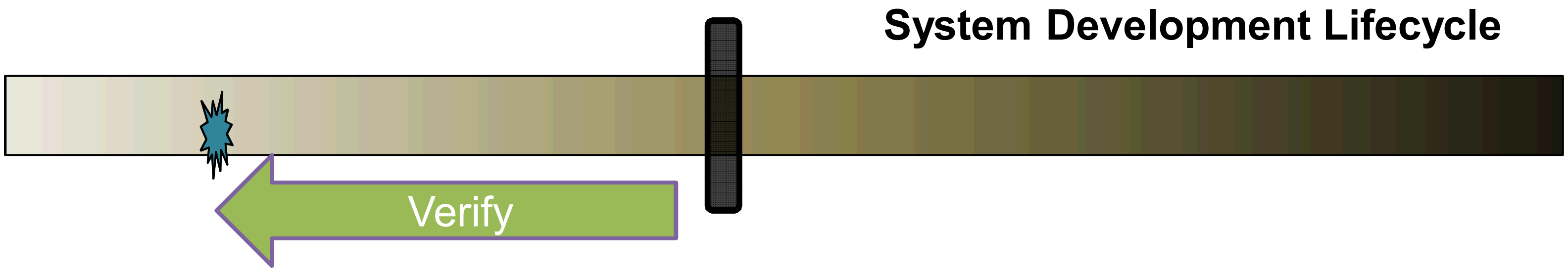
- At a given point in the development lifecycle, what can we do to address trust?



1. Harden: Prevent Attacks
   - Strengthen development processes to prevent /mitigate attack vectors
   - Isolate development networks, better materials, closed environments

Approaches to Trusted System Development

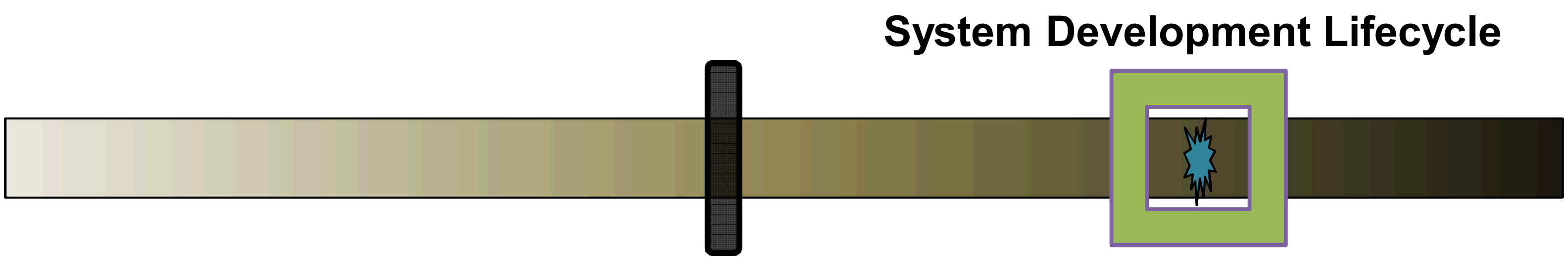
- At a given point in the development lifecycle, what can we do to address trust?



1. Harden: Prevent Attacks
2. Detect: Uncover previously deployed attacks
   - Trigger covert/stealthy attacks / killswitches
   - Discover manipulations of reliability

Approaches to Trusted System Development

- At a given point in the development lifecycle, what can we do to address trust?



1. Harden: Prevent Attacks
2. Detect: Uncover previously deployed attacks
3. Deter: Integrate processes and structures that survive attack
   - For attacks that survive hardening, detection
   - Construct structures that are resilient to attack efforts
   - Can include *development process resiliency*
   - Can include *system resiliency*

Open Discussion

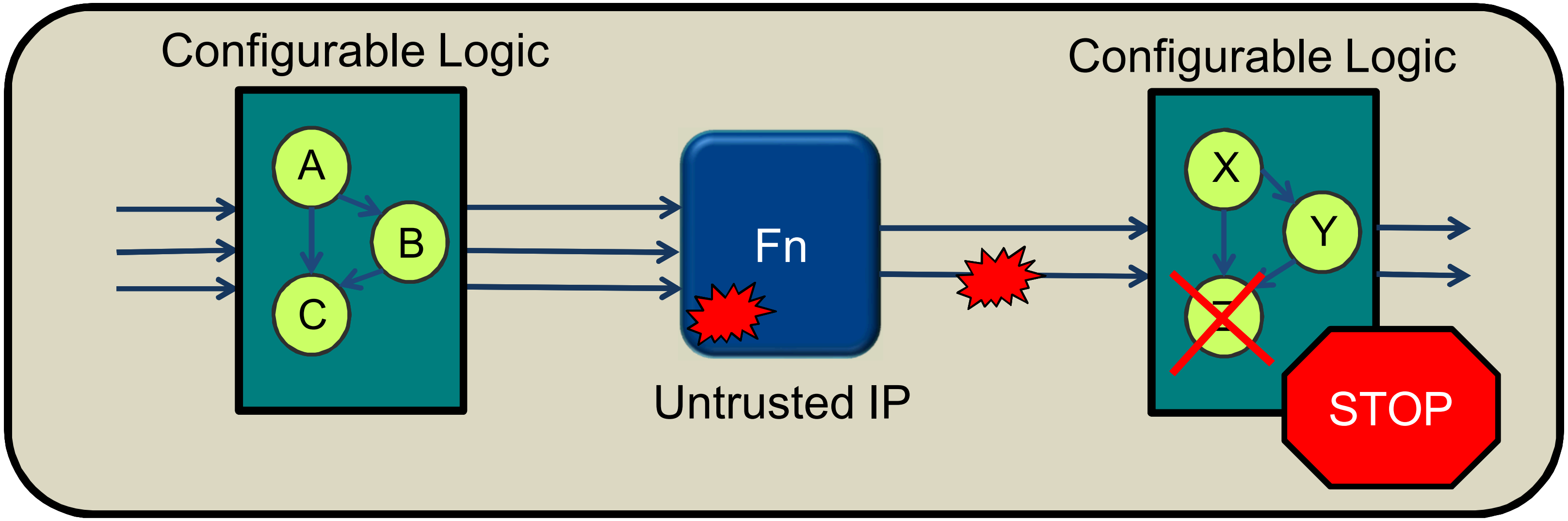
- What specific techniques can be developed to support Hardening, Detection and Deterrence for various types of systems?
  - Specific systems require unique solutions
- "Let the Punishment Fit the Crime" – How to quantify risk and determine the highest areas of risk? How to best address those risks?
- For each potential mitigation, how to predict quantitatively its effectiveness and impact on risk?
- How do we validate risk evaluation and risk reduction?
  - We will never have large data sets characterizing observations of attacks, and the effectiveness of deployed mitigations. Should we just ask the hackers….?

Summary

- Trust is a complex issue.   How do we prevent adversarial manipulation during development?
- Every system the US government and industry develops is faced with the challenge trust, and developers must determine how best to address risks
- Risks are inherently system-specific and must be addressed with knowledge of how the system is developed
- Research needed to address:
  - Risk quantification
  - How to mitigate risks for different systems (Harden, Detect, Deter)
  - Validation, and quantification/prediction of mitigation impact

Backup Slides

Example Detect Strategy: Hardware Isolation



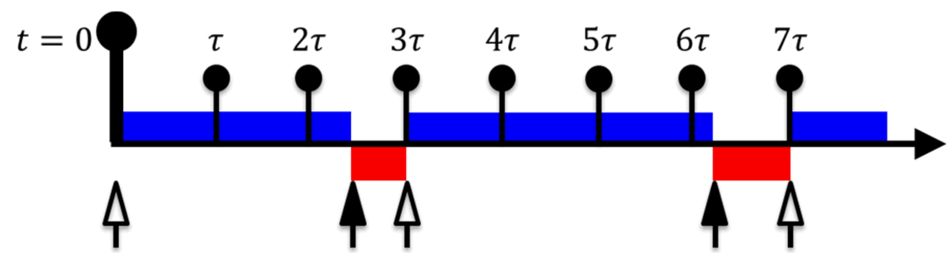
- Dynamically monitor, verify untrusted circuit behavior
  - Specify and monitor for behaviors that result in trust failures
- Configurable logic: expose attacker to uncertainty
- Monitoring coverage vs. required monitoring resources
  - Formal methods-based analysis required for derivation of monitoring logic, offline proof that coverage is sufficient

Allow Use of IP of Unknown Provenance in Trusted Systems

PLADD Mathematical Formulation

- Utility

$$u(x,S) = -\alpha - \beta x + \left(\min_{t_i \in S}\left(t_i : t_i \geq x\right) - x \right)$$

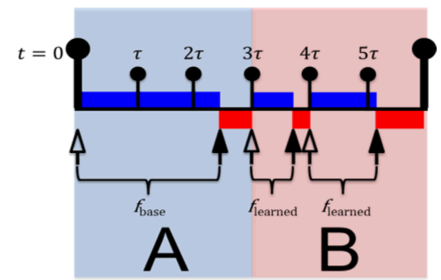


- Infinite time horizon

$$S = \{t_0, t_1, \cdots\}$$

$$E\left[u(X,S)\right] = -\alpha - \beta \int_0^\infty x f(x)\,dx + \int_0^\infty \left(\min_{t_i \in S}\left(t_i : t_i \geq x\right) - x \right) f(x)\,dx$$

- Finite time horizon



$$S = \{t_o, t_1, \cdots, t_{N+1}\}$$

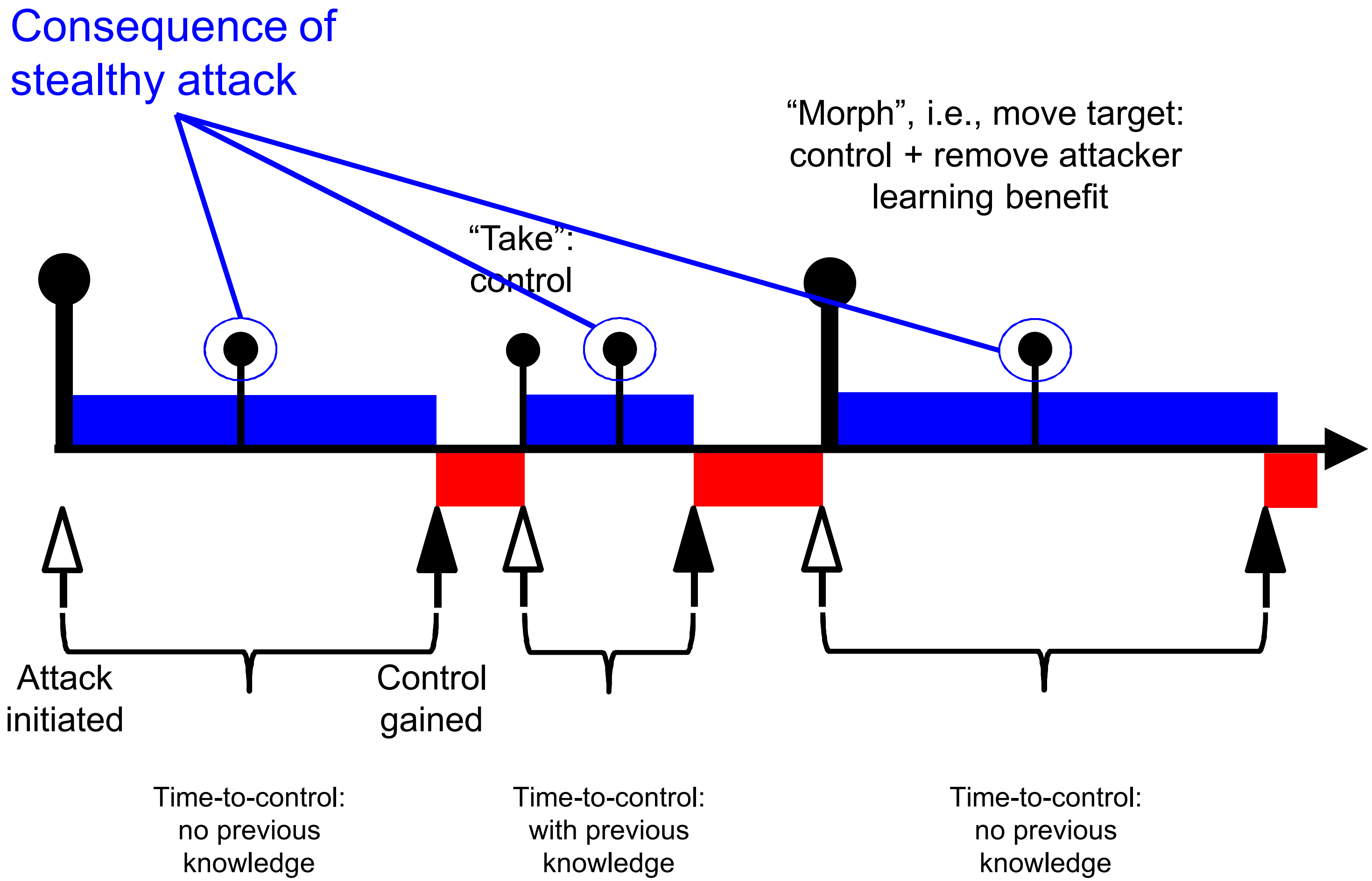
$$E_{N+1}\left[u(X,S)\right] = 0$$

$$E_j\left[u(X,S)\right] = -\alpha - \beta\left(t_{N+1} - t_j\right) \int_{\tau_{N+1}}^\infty f_{learned}(x - t_j)\,dx + \sum_{i=j+1}^{N+1} \int_{t_{i-1}}^{t_i} f_{learned}(x - t_j)\left(t_i - x - \beta(x - t_j) + E_i\left[u(X,S)\right]\right)dx$$

$$E\left[u(X,S)\right] = -\alpha - \beta t_{N+1} \int_{t_{N+1}}^\infty f_{base}(x)\,dx + \sum_{j=1}^{N+1} \int_{t_{j-1}}^{t_j} f_{base}(x)\left(t_j - x - \beta x + E_j\left[u(X,S)\right]\right)dx$$

Mathematics-based analysis of attacker utility

Probabilistic, Learning Attacker, Dynamic Defender (PLADD) Model



Pillars of Microelectronics-Based System Development