Human Biases Meet Cybersecurity of Embedded and Networked Systems

Saurabh Bagchi and Shreyas Sundaram
School of Electrical and Computer Engineering
CERIAS
Purdue University

Vision for Security of Embedded Systems

- Foundations for designing highly secure and resilient networked embedded systems
  - That can achieve mission success
  - Under component failures and sophisticated cyber/physical attacks
- Enable:
  - Systematic and rigorous design principles to build in security and resilience into software code bases of embedded systems
  - Real-time self-diagnostics to detect, identify, and isolate attacks and failures at millisecond level resolution
  - Rational process for deciding on where to spend security budget
  - Self-healing, real-time adaptation, and reconfiguration to achieve mission objectives
Problem Statement

- Many of our critical infrastructures run on large-scale, multi-organizational, interdependent cyberphysical systems (CPS)
- The CPS is subjected to a variety of security threats
  - cyber (e.g., sending malware against a control system)
  - physical (e.g., physically damaging a distribution line)
- Ensuring the security is a complex multi-faceted problem, and requires understanding
  - dynamics of physical systems
  - information exchange and attack propagation in cyber systems
  - human decision making during the design and operation of the coupled system
- Homogeneity in the system eases attack propagation

One Solution Direction: Randomization

- Randomization-based security[3]
  - Randomizes data as well as control to design provably secure systems
  - You cannot acquire one device and reverse engineer it to mount attacks
  - Deals with limited entropy available on embedded devices
  - Bounds degradation in resource usage or performance

Can Randomization Work for Embedded?

- Consider a class of low-end embedded platforms
- Constraints
  - Small memory sizes
  - 1 MB Flash, 128 KB’s of RAM
- Tight constraints on
  - Running time
  - Active power consumed
- Either: single application
  - No kernel/user space separation
- Or: OS with coarse-grained protection
  - Example: Entire thread needs to be provided elevated privileges

Current State of Security on Embedded Applications

Bare-metal Application

- Security Hardware
- Sensitive IO
- IO
- Global Data
- Stack
- Code
- Single (Root) execution domain

- Unused or trivially bypassed
- Always accessible
- Vulnerable to:
  - Stack smashing
  - Code injection
  - Global data corruption
- No ROP defenses
Why is Defense Hard?

- Often single binary image
  - No separation privilege levels (e.g. kernel, user)
- At best large root of trust
  - Much of code runs with elevated privileges
- Systems lack a Memory Management Unit (MMU)
  - Diversification or page-level protection of virtual memory absent
  - Defenses are limited to physical memory space
- Small memory sizes
- Tight run-time constraints: Both on mean overhead and variability

Threat Model and Requirements

- **Threat Model**
  - Arbitrary memory corruption
  - Attacker goals:
    - Take control of execution
    - Corrupt specific global data
  - Does *not* have physical access
- **Requirements**
  - Hardware support for two execution privilege modes
  - Memory Protection Unit (MPU)
    - Hardware that enforces access permissions on physical memory
Our Solution: EPOXY
*Embedded Privilege Overlay across X hardware for Y software*

- LLVM based compiler
- Protects against
  - Code injection
  - Control flow hijacking
  - Data corruption
  - Direct manipulation of IO
- Privilege Overlays
  - Creates two privilege levels
  - Security-sensitive operation done at higher privilege level
  - Static analysis identifies code that requires higher privileges

---

IoT Application After EPOXY

- Security Hardware
- Sensitive IO
- IO
- Global Data
- Stack
- Code

- Privileged Execution
- Unprivileged Execution

- Enabled enforcing DEP
  - Access Restricted
- Access Restricted
- Set to RW-NX
  - Stopping Code Injection
- Set to RX
  - Providing Code Integrity
### Performance Impact

<table>
<thead>
<tr>
<th></th>
<th>SS</th>
<th>PO</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BEEEbs Runtime</strong></td>
<td>Min</td>
<td>Ave</td>
<td>Max</td>
</tr>
<tr>
<td>Min</td>
<td>-7.3%</td>
<td>-3.5%</td>
<td>4.4%</td>
</tr>
<tr>
<td>Ave</td>
<td>-1.3%</td>
<td>0.1%</td>
<td>2.1%</td>
</tr>
<tr>
<td>Max</td>
<td>-11.7%</td>
<td>1.1%</td>
<td>14.2%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>SS</th>
<th>PO</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BEEEbs Power</strong></td>
<td>Min</td>
<td>Ave</td>
<td>Max</td>
</tr>
<tr>
<td>Min</td>
<td>-4.2%</td>
<td>0.2%</td>
<td>7.3%</td>
</tr>
<tr>
<td>Ave</td>
<td>-10.3%</td>
<td>-0.2%</td>
<td>2.8%</td>
</tr>
<tr>
<td>Max</td>
<td>-10.2%</td>
<td>2.5%</td>
<td>17.9%</td>
</tr>
</tbody>
</table>

SS - SafeStack Only, PO - Privilege Overlay Only

---

### What If I Cannot Afford The Performance Impact?

- Modern critical infrastructures have a large number of assets, managed by multiple stakeholders
  - Security depends critically on interdependencies among assets
- We develop a framework for **optimal and strategic** allocation of defense resources in large-scale systems
- Example: SCADA network

---

Attack Graphs to the Rescue

- **Used to**
  - Analyze risk to large-scale embedded system from multi-stage attack
  - Reduction in risk by strategic investments

- **Significant prior work**
  - Bayesian analysis to determine best placement of sensors and response agents

---

Systematic and Rigorous Analysis of Decision-Making for Security

**Key questions:**
- How do we reason systematically and rigorously about the actions of the various defenders and attackers in large-scale interdependent systems?
- What kinds of security outcomes can arise under distributed and decentralized decision-making?
- How do human biases impact the security decisions?

**In the rest of the talk:** bring together ideas from game theory and behavioral economics/psychology to answer the above questions

---

What is Game Theory?

- Consider a scenario with multiple decision-makers ("players")
- Each player has an available set of actions
- Each player gets a benefit that depends on their actions, and the actions of the other players; captured by a utility function

Game Theory:
Given a set of players, a set of actions for each player, and a utility function for each player, analyze/predict the outcomes under selfish decision-making by the players

Example: Prisoner’s Dilemma

- **Players:** Two prisoners
- **Actions:** Remain Quiet / Testify
- **Utilities:** 

<table>
<thead>
<tr>
<th></th>
<th>Remain Quiet</th>
<th>Testify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remain Quiet</td>
<td>5, 5</td>
<td>10, 3</td>
</tr>
<tr>
<td>Testify</td>
<td>3, 10</td>
<td>8, 8</td>
</tr>
</tbody>
</table>

Length of sentence to Player 1 if both players remain quiet
Length of sentence to Player 2 if both players remain quiet
Example: Prisoner’s Dilemma

- No matter what Player 2 does, it is best for Player 1 to testify (and vice versa)
- Outcome: both players testify (and serve 8 years)
- “Optimal” outcome: both players remain quiet (and serve 5 years)
- Selfish decision-making leads to a suboptimal outcome for both players!

```
\begin{tabular}{|c|c|c|}
\hline
\text{Prisoner 1} & \text{Remain Quiet} & \text{Testify} \\
\hline
\text{Remain Quiet} & 5, 5 & 10, 3 \\
\hline
\text{Testify} & 3, 10 & 8, 8 \\
\hline
\end{tabular}
```

Key Concept in Game Theory: Nash Equilibrium

- Consider a set of players, each taking an action
- The set of actions is said to be a Nash Equilibrium if no player can improve their utility by changing their action, when all other players keep playing their original action
- In Prisoner’s Dilemma, both players testifying is a Nash Equilibrium

- Nash equilibrium can be:
  - Pure: each player picks one specific action
  - Mixed: each player randomizes over their actions
Example: A Simple Security Game

Scenario:
- Two players: an attacker and a defender
- There are two targets
- Attacker has to choose whether to attack Target 1 or Target 2
- Defender has to choose whether to defend Target 1 or Target 2
- Defender wins if she chooses the same target as the attacker
- Attacker wins if she chooses a different target from the defender

Security Game: Utilities

- Utility matrix:

<table>
<thead>
<tr>
<th></th>
<th>Attacker</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Target 1</td>
<td>Target 2</td>
</tr>
<tr>
<td>Target 1</td>
<td>1, -1</td>
<td>-1, 1</td>
</tr>
<tr>
<td>Target 2</td>
<td>-1, 1</td>
<td>1, -1</td>
</tr>
</tbody>
</table>

- No Pure Nash Equilibrium in this game: both the attacker and defender must randomize over their actions
- Mixed Nash Equilibrium: Each player picks one of the targets to attack/defend with 50% probability
Behavioral Decision-Making

- Classical game theory assumes that the players (decision-makers) are rational, and take actions to maximize the expected value of the outcomes.
- However, behavioral economics and psychology have shown that humans systematically deviate from “classical” models of decision making.

Prospect Theory

Perceptions of values:
- **Reference dependence**: utility is derived from change in wealth rather than absolute levels of wealth.
- **Diminishing sensitivity**: risk averse in gains and risk seeking in losses.
- **Loss aversion**: disutility due to loss larger than utility due to gain of equal magnitude.

Perceptions of probabilities:
- **Overweighting** of small probabilities
- **Underweighting** of large probabilities
- **Diminishing sensitivity** for mid-range probabilities
Applications to Security: Interdependent Security Games Under Behavioral Probability Weighting

Interdependent Security Games

Players make their security investments in a shared system independently. Probability of attack is a function of investments of all players.

Question
What is the impact of behavioral perceptions of attack probabilities on the security investments?
Interdependent Security Games

- Consider a network consisting of $n$ nodes (e.g., an attack graph)
- Each node has an associated player, who has $1$ to invest in securing their node against attacks
  - Let player $i$’s investment be denoted by $s_i \in [0,1]$
- Probability that a node is successfully attacked is a function of security investments in the neighborhood of that node

Example: Total Effort Game
- Probability that node is successfully attacked depends on average investment in the neighborhood of that node

Optimal Security Investments Under Non-Behavioral Decision-Making

- Utility of each player in the total effort game:
  \[ u_i = -L_i \left( 1 - \frac{s_i + \sum_{j \in N(i)} s_j}{d_i} \right) - s_i \]
  - $L_i$ is the loss experienced by player $i$ due to a successful attack
  - $N(i)$: neighbors of node $i$
  - $d_i$: $1 +$ number of neighbors of node $i$

- Optimal investment by player $i$: \[ s_i^* = \begin{cases} 1, & \text{when } \frac{d_i}{L_i} < 1 \\ 0, & \text{when } \frac{d_i}{L_i} \geq 1 \end{cases} \]
  - “All or nothing” investment strategy
Impact of Behavioral Probability Weighting

**Question**
What happens under behavioral probability weighting?

- Does a pure Nash equilibrium exist under probability weighting?
- How do the investments and security levels at equilibrium depend on the properties of weighting functions?
- How do the investments and security levels at equilibrium depend on the topological properties of the network?

Existence and Properties of Nash equilibrium

**Theorem**
Theorem
There exists a Pure Nash equilibrium (PNE), with player-specific probability weighting functions and cost parameters. Furthermore, in any graph (and with potentially heterogeneous players), the attack probability at each node is always less than 1 at a PNE.

- Recall: Without probability weighting, players invest 0 in certain cases
- Probability weighting eliminates such cases
Does Probability Weighting Lead to More Secure Equilibria?

**Theorem**
Consider a $d$-regular graph. Then there exists a threshold $t$ such that:
- If $d > t$: larger probability weighting leads to a smaller attack probability at equilibrium
- If $d < t$: larger probability weighting leads to a larger attack probability at equilibrium

**Interpretation:**
- Effect of probability weighting most beneficial when the attack probability is high
  - e.g., in networks where each node has many neighbors
- For moderate equilibrium attack probabilities, less probability weighting results in more secure equilibrium.

Expected Fraction of Attacked Vertices

**Question:**
Within the class of graphs with a given number of nodes and edges, which graphs minimize the expected fraction of nodes that are successfully attacked at a Nash equilibrium?

**Definition:**
A quasi-complete graph $QC(n, e)$ with $n$ nodes and $e$ edges is defined via the following construction:
- Use as many edges as possible to build a clique
- Add the remaining edges to a single additional node and connect them to the nodes in the clique

Example: $QC(6,3)$

Example: $QC(6,5)$
Theorem:
- Within the class of graphs with $n$ nodes and $e$ edges, the quasi-complete graph $QC(n, e)$ minimizes the bounds on the expected fraction of successfully attacked vertices at a PNE in the Total Effort game.
- Among all connected graphs on $n$ nodes, the expected fraction of successfully attacked nodes is smallest in the star graph.
- Among all connected graphs with a given number of edges and nodes, the expected fraction of successfully attacked nodes is highest in degree-regular graphs.

Key insight:
Collect edges on as few nodes as possible in order to concentrate attack risks on those nodes.

Ongoing Research

- Extensions to more classes of embedded devices and applications
  - Multiple privilege levels with effective switching among them
  - Handling binary libraries
  - Handling variety of third-party peripherals and their firmware
- Extensions to more general attack graph settings
  - Each defender can manage multiple assets
  - There can be multiple rounds of attack-defense
  - Different stakeholders have different degrees of knowledge about each other
- Preliminary insights:
  - It is possible to enforce multiple privilege levels for security even on low-end embedded devices
  - Behavioral decision-making can cause defenders to invest suboptimally
  - In settings with multiple defenders, behavioral players can benefit the other players
  - Removing restrictions on the locations of security investments can significantly improve system-wide security
Summary

- Current state of work:
  - Developed a suite of protocols specialized to embedded systems for control flow and data integrity protection
  - Examined the impact of behavioral perceptions of values and probabilities on security of interdependent systems and networks
- In interdependent security games:
  - Behavioral probability weighting gives rise to a much richer spectrum of Nash equilibrium than under classical models
  - Misperceptions of probabilities can be helpful for security in dense networks, where the security risk is high
  - Optimal networks to mitigate security risks involve concentrating the edges on as few nodes as possible

Thanks!