ADVERSARIAL AND UNCERTAIN REASONING FOR ADAPTIVE CYBER DEFENSE: BUILDING THE SCIENTIFIC FOUNDATION

Sushil Jajodia
George Mason University
Outline

- Motivation
  - Current cyber defense landscape & open questions
- Pro-active Defense via Adaptation
  - Adaptation Techniques
  - Scientific Challenges
- Research Highlights
Motivation
Today’s Cyber Defenses are Static

- Today’s approach to cyber defense is governed by slow and deliberative processes such as:
  - Security patch deployment, testing, episodic penetration exercises, and human-in-the-loop monitoring of security events.

- Adversaries can greatly benefit from this situation:
  - They can continuously and systematically probe targeted networks with the confidence that those networks will change slowly if at all.
  - They have the time to engineer reliable exploits and pre-plan their attacks.

- Additionally, once an attack succeeds, adversaries persist for long times inside compromised networks and hosts:
  - Hosts, networks, software, and services do not reconfigure, adapt, or regenerate except in deterministic ways to support maintenance and uptime requirements.
Pro-active Defense via Adaptation
Adaptation Techniques (AT) consist of engineering systems that have **homogeneous functionalities** but **randomized manifestations**

- These techniques make networked information systems less **homogeneous and less predictable**
- **Examples**: Moving Target Defenses (MTD), artificial diversity, and bio-inspired defenses

**Homogeneous functionality** allows **authorized use** of networks and services in predictable, standardized ways

**Randomized manifestations** make it difficult for attackers to engineer exploits remotely, or reuse the same exploit for successful attacks against a multiplicity of hosts
Adversary and Defender Uncertainty

In a static configuration, over time, the adversary will improve his knowledge about network topology and configuration, thus reducing his uncertainty.

When ATs are deployed, each system reconfiguration will invalidate previous knowledge acquired by adversaries, thus restoring their uncertainty to higher levels.

Learning phase: legitimate users have to adapt to the new configuration.

Learning phase: the attacker has to gather new information about the reconfigure system.
ATs enable us to maintain the information gap between adversaries and defenders at a relatively constant level

- Before deploying the proposed mechanisms, the defender’s advantage is eroded over time
- Dynamically changing the attack surface ensures a persistent advantage

If the system's configuration remains static, the attacker will eventually learn all the details about the configuration.
AT Benefits

- Increase complexity, cost, and uncertainty for attackers
- Limit exposure of vulnerabilities and opportunities for attack
- Increase system resiliency against known and unknown threats
- Offer probabilistic protection despite exposed vulnerabilities, as long as the vulnerabilities are not predictable by the adversary at the time of attack
Software-Based Adaptation

- **Address Space Layout Randomization (ASLR)**
  - Randomizes the locations of objects in memory, so that attacks depending on knowledge of the address of specific objects will fail.

- **Instruction Set Randomization (ISR)**
  - A technique for preventing code injection attacks by randomly altering the instructions used by a host machine or application.

- **Compiler-based Software Diversity**
  - When translating high-level source code to low-level machine code, the compiler diversifies the machine code on different targets, so that vulnerability exploits working on one target may not work on other targets.
Network-Based Adaptation

- ID randomization
- Generation of arbitrary external attack surfaces
- VM-based dynamic virtualized network
- Phantom servers to mitigate insider and external attacks
- Proxy moving and shuffling to detect insider attacks

Overall, these techniques aim at giving the attacker a view of the target system that is significantly different from what the system actually is
But there are Many ACD Ideas…

At least 39 documented in this 2013 MIT Lincoln Labs Report

>50 today?

How can we compare them?
<table>
<thead>
<tr>
<th>Most Dominant Technique</th>
<th>Least Dominant Technique</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Effectiveness with Medium-Low Costs</td>
<td>Low Effectiveness with High, Medium, or Low Costs</td>
</tr>
<tr>
<td>High Effectiveness with Medium-High Costs</td>
<td></td>
</tr>
<tr>
<td>Medium Effectiveness with Medium-Low Costs</td>
<td>Medium Effectiveness with Medium-Low Costs</td>
</tr>
<tr>
<td>Medium Effectiveness with Medium-High Costs</td>
<td>Medium Effectiveness with Medium-High Costs</td>
</tr>
<tr>
<td>Low Effectiveness with High, Medium, or Low Costs</td>
<td>Low Effectiveness with High, Medium, or Low Costs</td>
</tr>
</tbody>
</table>

**Spectrum of Moving Target Defense Techniques**

- **SQLRand**
- **Proactive Obfuscation**
- **Operating System Randomization**
- **Function Pointer Encryption**
- **Multivariant Execution**
- **Program Differentiation**
- **Against System Code Injection with System Call Randomization**
- **RandSys**
- **Genesis**
- **Network Address Space Randomization**
- **Revere**
- **RandSys Program Differentiation**
- **Randomized Intrusion-Tolerant Asynchronous Service**
- **G-Free**
- **Reverse Stack Execution in a Multi-Variant Environment**
- **Dynamic Backbone Randomized Instruction Set Emulation**
- **Dynamic Network Address Translation**
- **Address Space Layout Permutation**
- **Practical Software Dynamic Translation**
- **Active Repositioning in Cyberspace for Synchronized Evasion**
- **Instruction Level Memory Randomization**
- **Dynamic Runtime Environment: Address Space Layout Randomization**
- **Dynamic Runtime Environment: Instruction Set Randomization**
- **Dynamic Software**
- **Dynamic Networks**
- **Dynamic Platforms**

*Source: Kate Ferris, George Cybenko*
Limitations of Current Approaches

- The contexts in which ATs are useful and their added cost (in terms of performance and maintainability) to the defenders can vary significantly
  - Most ATs aim at preventing a specific type of attack

- The focus of existing approaches is on developing new techniques, not on understanding overall operational costs, when they are most useful, and what their possible interrelationships might be

- While each AT might have some engineering rigor, the overall discipline is largely ad hoc when it comes to understanding the totality of AT methods and their optimized application

- AT approaches assume non-adversarial, environments
Adaptive Cyber Defense (ACD)

- We need to understand
  - the overall operational costs of these techniques
  - when they are most useful
  - their possible inter-relationships
- Propose new classes of techniques that force adversaries to continually re-assess and re-plan their cyber operations
- Present adversaries with optimally changing attack surfaces and system configurations
Advanced Persistent Threats (APTs) have the time and technology to easily exploit our systems now.

<table>
<thead>
<tr>
<th>Attack Phase</th>
<th>Reconnaissance</th>
<th>Access</th>
<th>Persistence</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identify the attack surface</td>
<td>Compromise a targeted component</td>
<td>Maintain presence and exploitation</td>
</tr>
<tr>
<td>Possible Adaptation Techniques (AT)</td>
<td>Randomized network addressing and layout; Obfuscated OS types and services.</td>
<td>Randomized instruction set and memory layout; Just-in-time compiling and decryption.</td>
<td>Dynamic virtualization; Workload and service migration; System regeneration.</td>
</tr>
</tbody>
</table>

Adaptation techniques are typically aimed at defeating different stages of possible attacks.

We need to develop a scientific framework for optimizing strategies for deploying adaptation techniques for different attack types, stages and underlying missions.
Novel Adaptive Techniques

- Manipulating responses to an attacker’s probes
  - **Goal:** altering the attacker’s perception of a system’s attack surface

- Creating distraction clusters
  - **Goal:** controlling the probability that an intruder may reach a certain goal within a specified amount of time

- Increasing diversity
  - **Goal:** increasing the complexity and cost for attackers by increasing the diversity of resources along certain attack paths
    - Different metrics are proposed to measure diversity
The internal attack surface represents insider knowledge about the system, and can use topology graphs, attack graphs, dependency graphs, or a combination of them. For the sake of presentation, this example only shows topology information.
The external attack surface represent what we want the attacker to infer about the system. Inference is based on probing and sniffing.
We aim at delaying intrusions by controlling the probability that an intruder may reach a certain goal within a specified amount of time.
We take the first step towards formally modeling network diversity as a security metric.

- We propose a network diversity function based on well-known mathematical models of biodiversity in ecology.
- We design a network diversity metric based on the least attacking effort.
- We design a probabilistic network diversity metric to reflect the average attacking effort.
- We evaluate the metrics and algorithms through simulation.

The modeling effort helps understand diversity and enables quantitative hardening approaches.
Solving Real-world Problems

- Adversarial defense of enterprise systems
  - Pareto-optimal solutions that allow defenders to simultaneously maximize productivity and minimize the cost of patching
- Optimal scheduling of cyber analysts
  - Given limited resources, the analyst workforce must be optimally managed for minimizing risk
The attacker starts probing and is somehow redirected to the honeypot (VLAN, IPS and so on).

Logging the activities

The attacker checks for other systems

The attacker realizes that the system is a honeypot
A Different Approach

The attacker sees directly the Production System.

Option 1: He thinks that the system is a Honeypot, look for other systems.

Option 2: The attacker keeps interacting with the system.

Production System: Client/Server/Honeypot/Network Component

Attacker

Joint work with Prof Luigi Mancini, U of Rome
Evaluation of our Approach

31 last year MSc students

3-layer experiment:
- L1 - No AHEAD deployed
- L2 - AHEAD on one machine
- L3 - AHEAD on both machines

Goal: root privilege in L3 machine
L3 machines and L1 machines had same vulnerable service
# Results

<table>
<thead>
<tr>
<th>Layer</th>
<th>Machine</th>
<th>Success %</th>
<th>Time to Success</th>
<th>Traffic (GB)</th>
<th>Avg. Individual Traffic</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td></td>
<td>90.32%</td>
<td>1h 9m 36s</td>
<td>21.23</td>
<td>0.68</td>
</tr>
<tr>
<td></td>
<td>Prod. System 1</td>
<td>5.34%</td>
<td></td>
<td>7.4305</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td>Prod. System 2</td>
<td>84.98%</td>
<td></td>
<td>13.7995</td>
<td>0.44</td>
</tr>
<tr>
<td>L2</td>
<td></td>
<td>61%</td>
<td>14h 37m 26s</td>
<td>78.88</td>
<td>2.82</td>
</tr>
<tr>
<td></td>
<td>Prod. System 3</td>
<td>61%</td>
<td>14h 37m 26s</td>
<td>52.0608</td>
<td>1.86</td>
</tr>
<tr>
<td></td>
<td>Prod. System + AHEAD</td>
<td>0%</td>
<td>∞</td>
<td>26.82</td>
<td>0.96</td>
</tr>
<tr>
<td>L3</td>
<td></td>
<td>6%</td>
<td>48h 25m 42s</td>
<td>54.89</td>
<td>2.89</td>
</tr>
<tr>
<td></td>
<td>Prod. System1 + AHEAD</td>
<td>0%</td>
<td>∞</td>
<td>23.6027</td>
<td>1.24</td>
</tr>
<tr>
<td></td>
<td>Prod. System2 + AHEAD</td>
<td>6%</td>
<td>48h 25m 42s</td>
<td>31.29</td>
<td>1.65</td>
</tr>
</tbody>
</table>
Optimal Scheduling of Cyber Analysts for Minimizing Risk

*Joint work with Rajesh Ganesan (GMU), Ankit Shah (GMU), Hasan Cam (ARL)
Statement of Need

- Cybersecurity threats are on the rise
- Demand for Cybersecurity analysts outpaces supply [1] [2]
- Given limited resources (personnel), the analyst workforce must be optimally managed
- Given the current/projected number of alerts it is also necessary to know the optimal workforce size

Process Flow, Definition of Significant Alerts

Significant Alerts = 1% of all Alerts Generated

August 19, 2017

IEEE 5G Summit
# DON Cyber Incident Category

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Root Level Intrusion (Incident):</strong> Unauthorized privileged access (administrative or root access) to a DoD system.</td>
</tr>
<tr>
<td>2</td>
<td><strong>User Level Intrusion (Incident):</strong> Unauthorized non-privileged access (user-level permissions) to a DoD system. Automated tools, targeted exploits, or self-propagating malicious logic may also attain these privileges.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Unsuccessful Activity Attempted (Event):</strong> Attempt to gain unauthorized access to the system, which is defeated by normal defensive mechanisms. Attempt fails to gain access to the system (i.e., attacker attempt valid or potentially valid username and password combinations) and the activity cannot be characterized as exploratory scanning. Can include reporting of quarantined malicious code.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Denial of Service (DOS) (Incident):</strong> Activity that impairs, impedes, or halts normal functionality of a system or network.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Non-Compliance Activity (Event):</strong> This category is used for activity that, due to DoD actions (either configuration or usage) makes DoD systems potentially vulnerable (e.g., missing security patches, connections across security domains, installation of vulnerable applications, etc.). In all cases, this category is not used if an actual compromise has occurred. Information that fits this category is the result of non-compliant or improper configuration changes or handling by authorized users.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Reconnaissance (Event):</strong> An activity (scan/probe) that seeks to identify a computer, an open port, an open service, or any combination for later exploit. This activity does not directly result in a compromise.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Malicious Logic (Incident):</strong> Installation of malicious software (e.g., trojan, backdoor, virus, or worm).</td>
</tr>
<tr>
<td>8</td>
<td><strong>Investigating (Event):</strong> Events that are potentially malicious or anomalous activity deemed suspicious and warrants, or is undergoing, further review. No event will be closed out as a Category 8. Category 8 will be re-categorized to appropriate Category 1-7 or 9 prior to closure.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Explained Anomaly (Event):</strong> Events that are initially suspected as being malicious but after investigation are determined not to fit the criteria for any of the other categories (e.g., system malfunction or false positive).</td>
</tr>
</tbody>
</table>
Cybersecurity threats are on the rise

Demand for Cybersecurity analysts outpaces supply

[1] [2]

Given limited resources (personnel), the analyst workforce must be optimally managed for minimizing today’s risk

Given the current/projected number of alerts it is also necessary to know the optimal workforce size to keep risk under a certain threshold

Definition of Risk

- Alert Coverage is defined as the % of the significant alerts (1% of the total alerts) that are thoroughly investigated in a work-shift by analysts and the remainder (forms the Risk) is not properly analyzed or unanalyzed because of:
  - Sub-optimal shift scheduling
  - Not enough personnel in the organization
  - Lack of time (excessive analyst workload)
  - Not having the right mix of expertise in the shift in which the alert occurs

- Risk % = 100 – Alert Coverage %

Note: From this slide onward, the term alert refers to significant alerts only.
Requirements

- The cybersecurity analyst scheduling system
  - Shall ensure that an optimal number of staff is available to meet the demand to analyze alerts
  - Shall ensure that a right mix of analysts are staffed at any given point in time
  - Shall ensure that risks due to threats are maintained below a pre-determined threshold
  - Shall ensure that weekday, weekend, and holiday schedules are drawn such that it conforms to the working hours/leave policy
Problem Description

Risk is proportional to Analyst Characteristics

1. Alert generation rate
2. the number of analysts,
3. their expertise mix,
4. analyst’s shift and days-off scheduling,
5. their sensor assignment,
6. Category of alert – analyst workload – time to analyze (input)

Two types of problems to solve:

Simulation: Given all of the above, what level of risk is the organization operating at?
Optimization: Given an upper bound on risk, what are the optimal settings for 1-5?
Algorithm Contributions

Optimization Algorithm

- Mixed Integer Programming solved using Genetic Algorithm
- Outputs
  - the number of analysts,
  - their expertise mix,
  - their sensor-to-analyst assignment

Scheduling Algorithm

- Integer programming and a heuristic approach
- Output
  - Analyst shift and days-off scheduling

Simulation Algorithm

- Validates optimization
- A tool can be used as a stand-alone algorithm to measure the current risk performance of the organization for a given set of inputs
Main Results

For a given analyst/sensor ratio, risk is independent of the # of sensors, when the average alert arrival and average service rates remain the same.

- Risk% varies non-linearly with analyst/sensor (A/S) ratio
- Plot is useful for hiring decisions
- Assumption: All sensors have the same average alert generation rate, and it remains fixed
Sample days off Scheduling

- An analyst works $12 \times 6 + 1 \times 8 = 80$ hrs in 2 weeks (7 out of every 14 days from Sun to Sat)
- Gets every other weekend off
- Works no more than 5 consecutive days in a 14 day period

Output of the days-off scheduling algorithm or 10 analysts

| Day | S | S | M | T | W | T | F | S | S | M | T | W | T | F | S | S | M | T | W | T | F | S |
| 1   | X | X | X | X |   | X |   | X | X |   | X | X | X | X | X |   | X | X | X | X | X | X |
| 2   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 3   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 4   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 5   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 6   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 7   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 8   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 9   | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |
| 10  | X | X | X | X | X |   | X | X | X | X | X | X | X | X | X |   | X | X | X | X | X | X |

$X$ – off days
Need for Dynamic Scheduling

- Static optimization and scheduling assumes
  - Same average alert generation rates for all sensors, which is drawn from a Uniform distribution.
- What if there are world events or zero-day attacks that could trigger an increase in analyst workload
- What if there are varying alert generation rates per sensor per hour
  - Causes uncertainty in future alert workload to be investigated
    - Workload uncertainty makes it difficult for managing personnel scheduling
      - How many analysts at each level of expertise must report to work?
      - Do we have the flexibility in the schedule to adapt to day to-day changing analyst needs
Alert estimation is critical for a successful implementation of the dynamic optimization model.

The average alert generation rate must be handled by a static workforce (X matrix).

Dynamic optimization is capable of adapting to changes in alert generation because the alert estimation model is updated daily and the model learns to bring in adequate on-call personnel by simulating several alert generation rates.

If estimation accuracy is good then risk is minimized and balanced between the 14-days.
Questions?

Sushil Jajodia

jajodia@gmu.edu

http://csis.gmu.edu/jajodia