Adversarial and Uncertain Reasoning for Adaptive Cyber Defense: Building the Scientific Foundation

Sushil Jajodia
George Mason University

20th European Symposium on Research in Computer Security
Vienna, Austria , September 24, 2015

Outline

- Motivation
  - Current cyber defense landscape & open questions
- Pro-active Defense via Adaptation
  - Adaption Techniques
  - Scientific Challenges
- Research Highlights

Motivation

Today's Cyber Defenses are Static

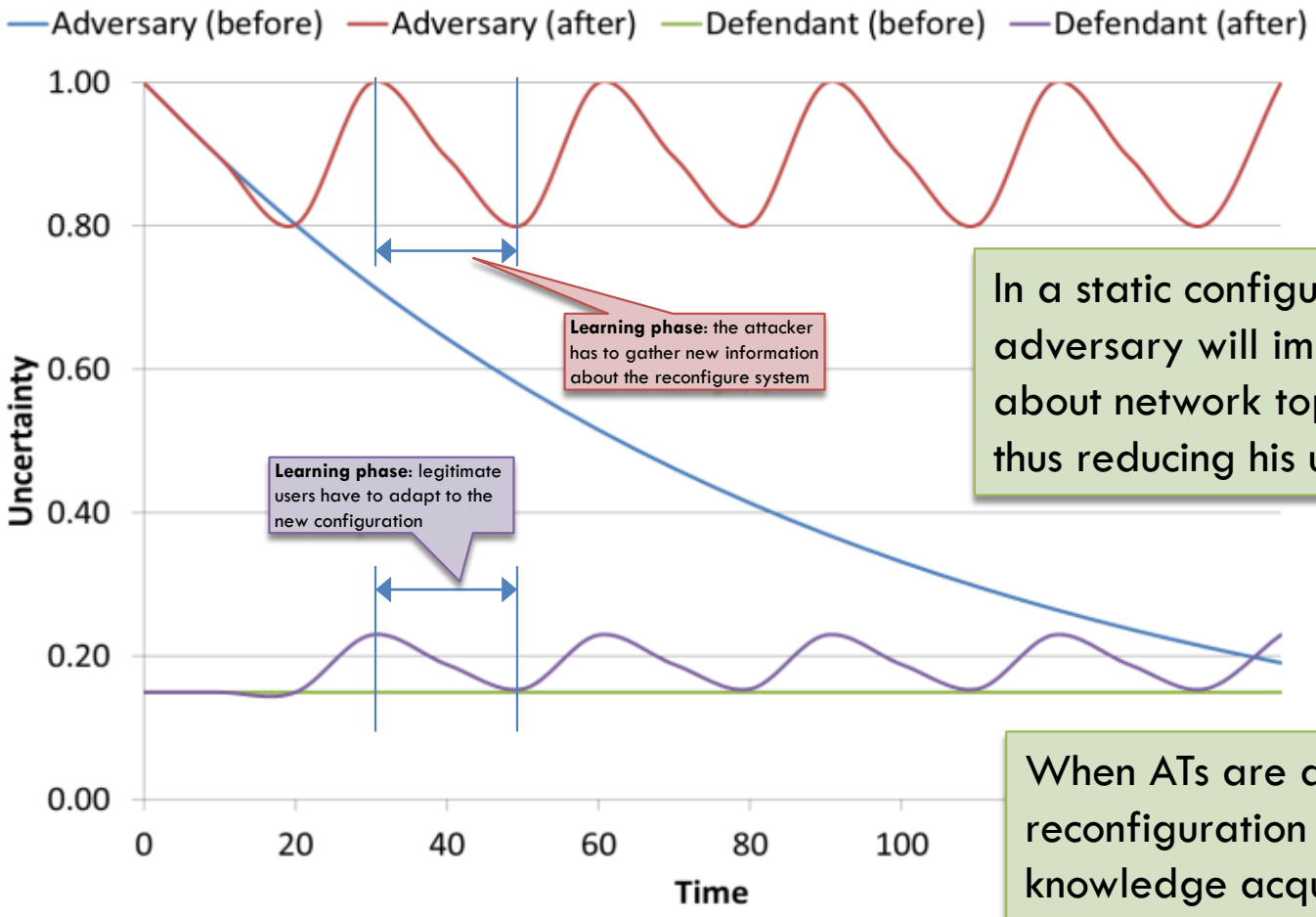
- Today's approach to cyber defense is *governed by slow and deliberative processes* such as
  - Security patch deployment, testing, episodic penetration exercises, and human-in-the-loop monitoring of security events
- Adversaries can greatly benefit from this situation
  - They can *continuously and 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

Security through adaptation: A paradigm shift

- 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

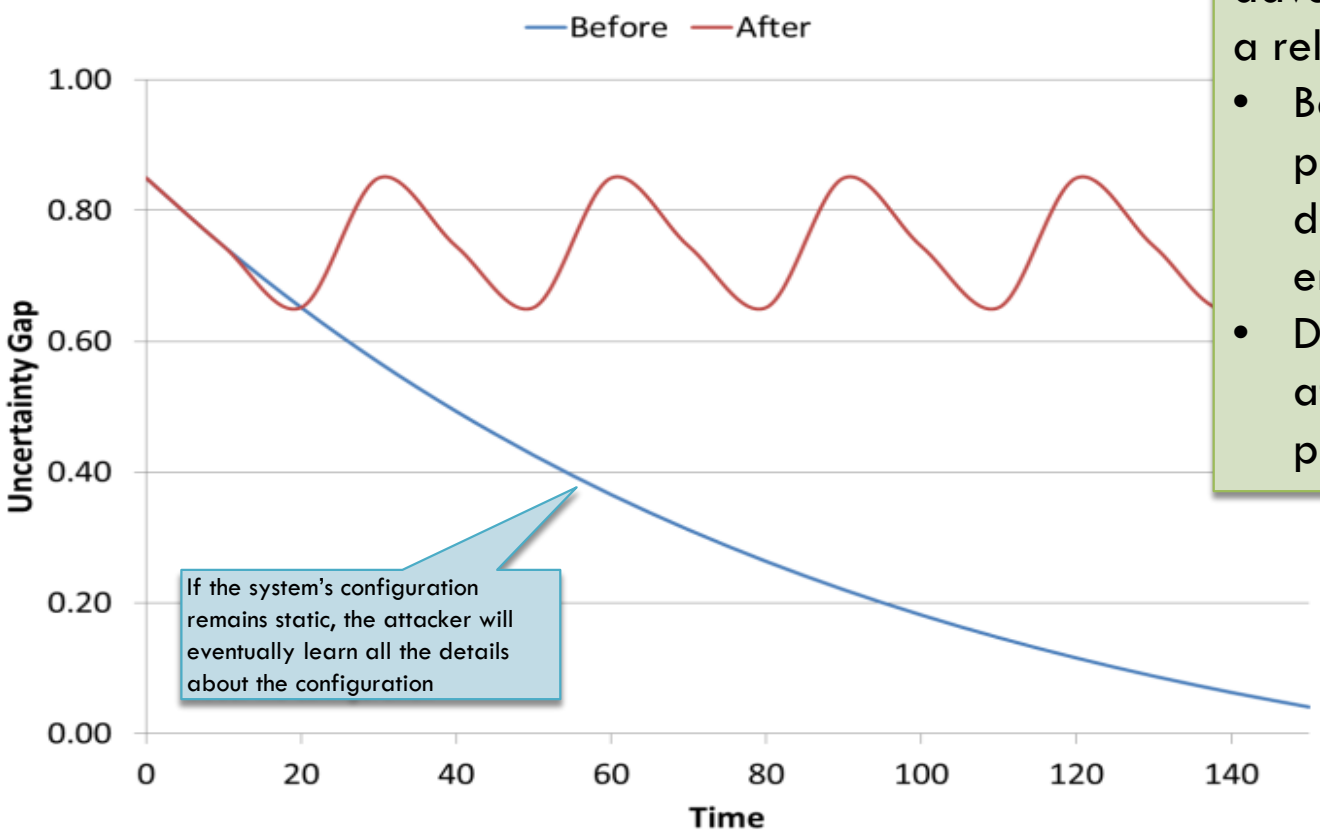
Adversary and Defender Uncertainty



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

Uncertainty Gap



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

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

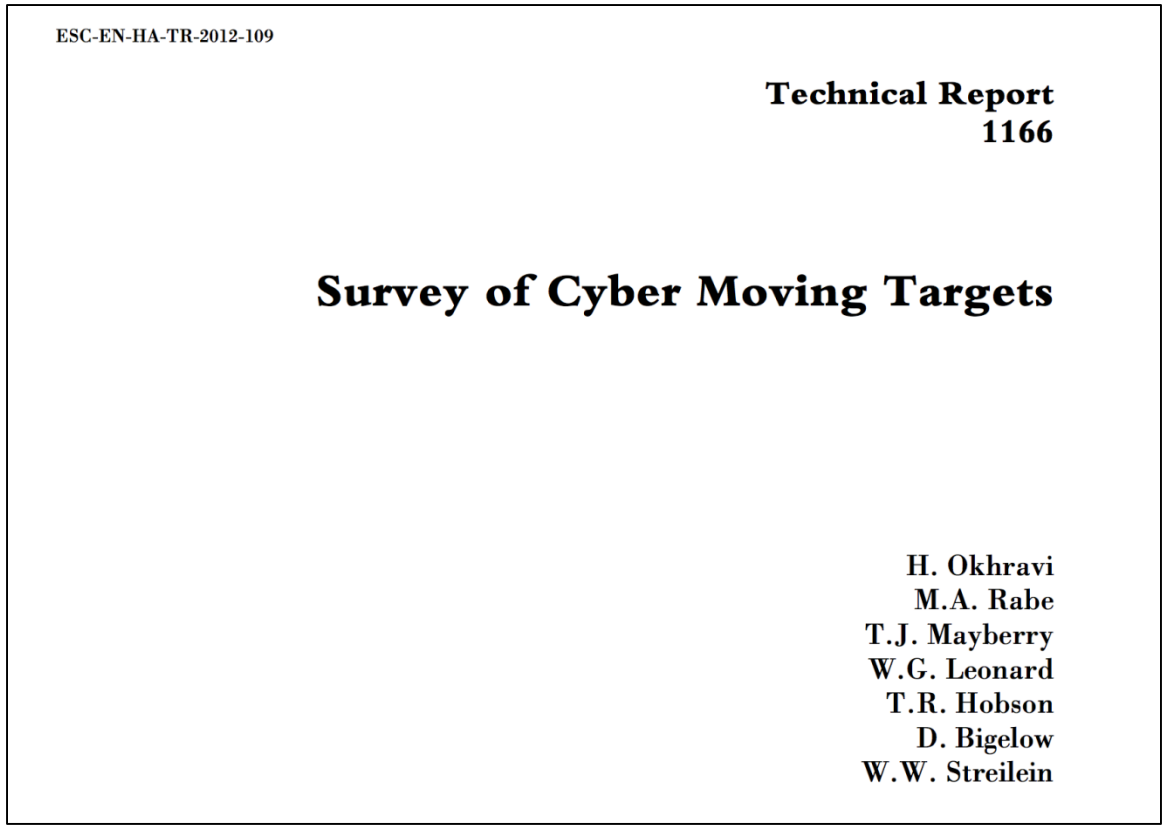
Software-Based Adaptation

- **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…



ESC-EN-HA-TR-2012-109

Technical Report
1166

Survey of Cyber Moving Targets

H. Okhravi
M.A. Rabe
T.J. Mayberry
W.G. Leonard
T.R. Hobson
D. Bigelow
W.W. Streilein

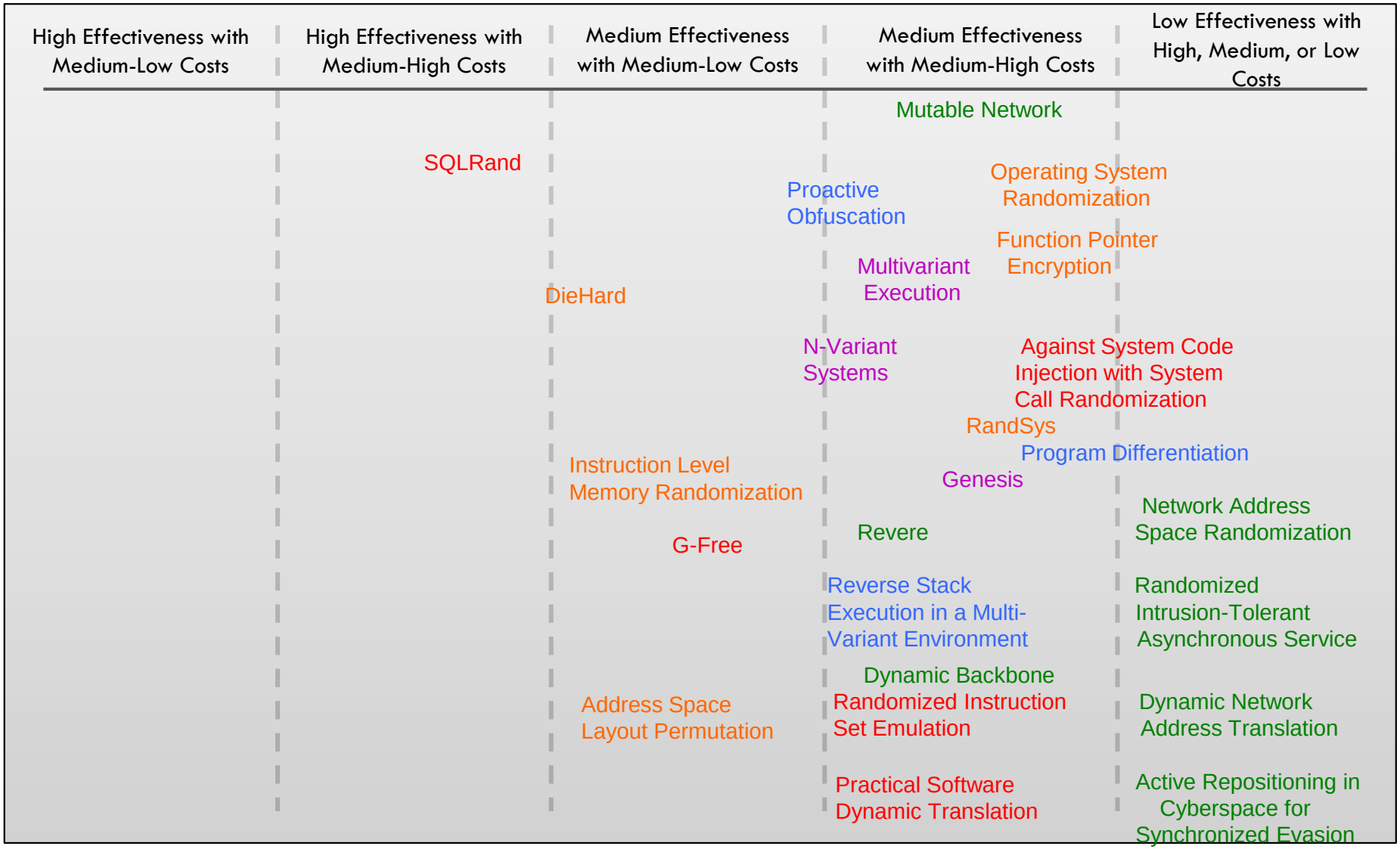
At least 39 documented in this 2013 MIT Lincoln Labs Report

>50 today?

How can we compare them?

Spectrum of Moving Target Defense Techniques

Most Dominant Technique ⟷ Least Dominant Technique



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

Adaptive Cyber Defense (ACD)

Advanced Persistent Threats (APTs) have the time and technology to easily exploit our systems now

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

There are many possible AT options

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

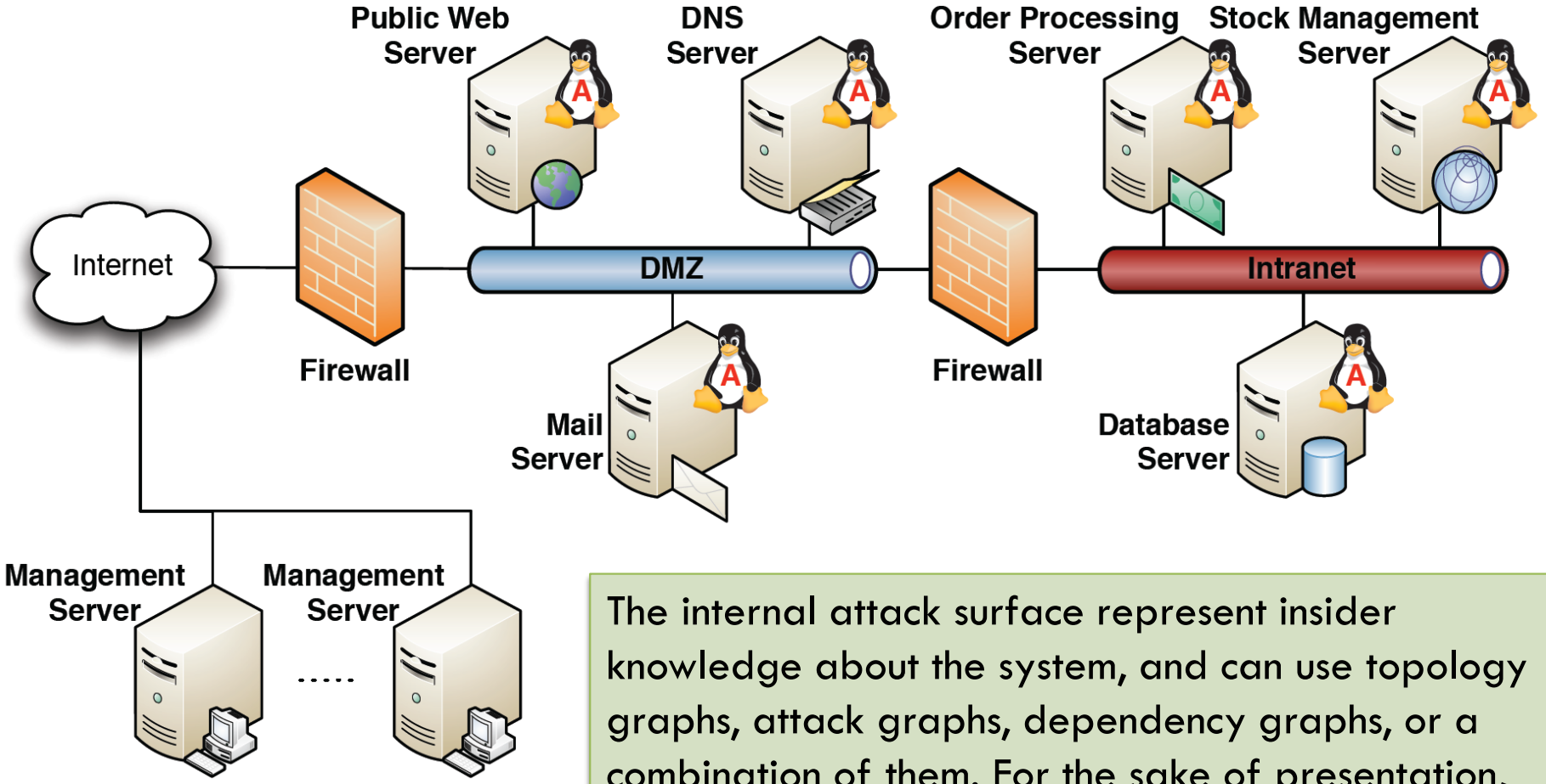
We need to develop a scientific framework for optimizing strategies for deploying adaptation techniques for different attack types, stages and underlying missions

Research Highlights

Novel Adaptive Techniques

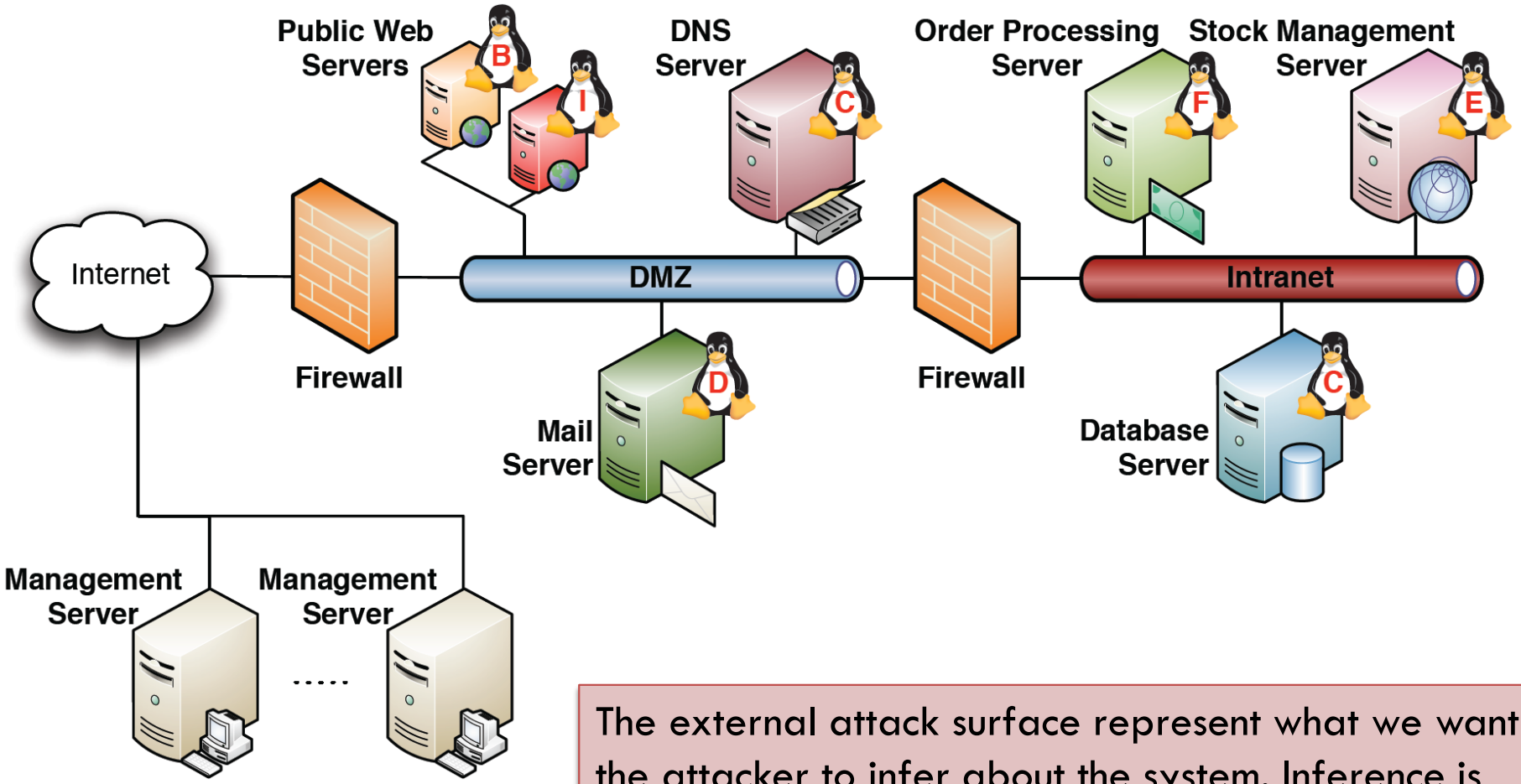
- Manipulating responses to an attacker's probes
  - **Goal:** altering the attacker's perception of a system's attack surface
- Creating distraction clusters
  - **Goal:** controlling the probability that an intruder may reach a certain goal within a specified amount of time
- Increasing diversity
  - **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

Example: Internal Attack Surface



The internal attack surface represent 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.

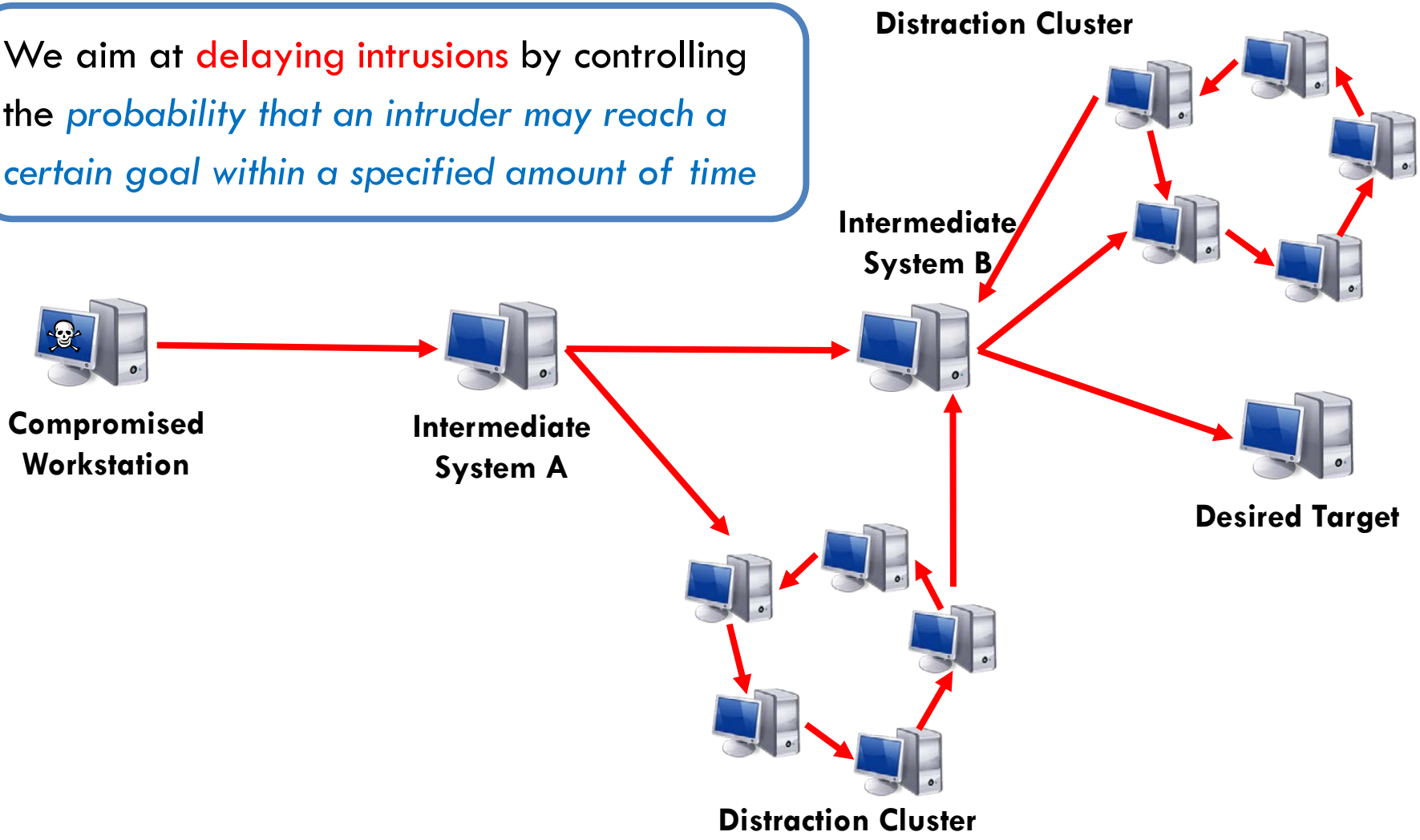
Example: External Attack Surface



The external attack surface represent what we want the attacker to infer about the system. Inference is based on probing and sniffing.

Distraction Clusters

We aim at delaying intrusions by controlling the *probability that an intruder may reach a certain goal within a specified amount of time*



Network diversity

- 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

Optimal Scheduling of Cyber Analysts for Minimizing Risk*

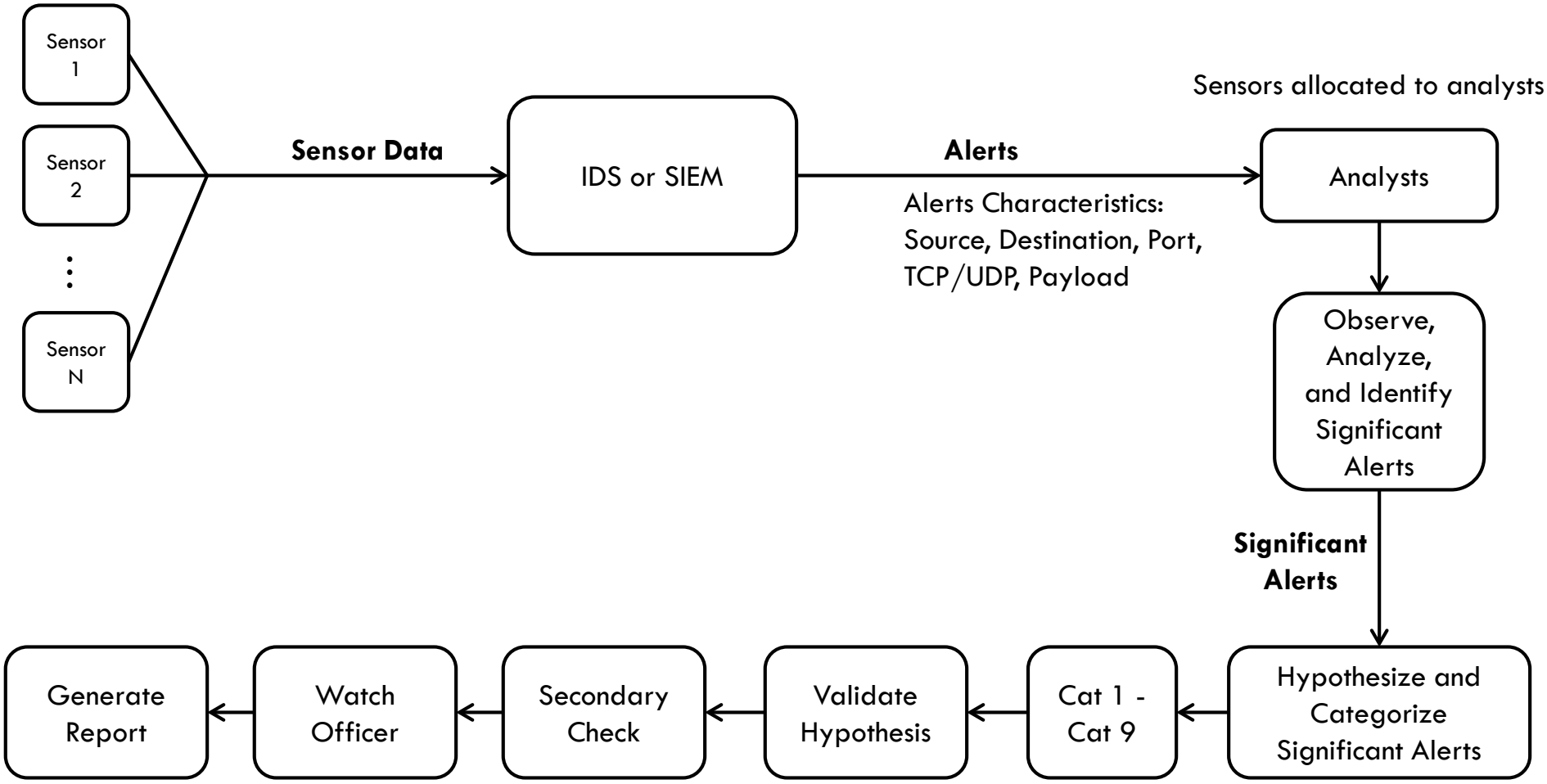
*Joint work with Rajesh Ganesan (GMU), Hasan Cam (ARL), Ankit Shah (GMU)

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

[1] http://www.rand.org/pubs/research_reports/RR430.html
[2] http://www.rand.org/news/press/2014/06/18.html

Process Flow, Definition of Significant Alerts



Significant Alerts = 1% of all Alerts Generated

Categories 1-9

DON CYBER INCIDENT CATEGORY

Cat 1-9	Description
1	**Root Level Intrusion (Incident):** Unauthorized privileged access (administrative or root access) to a DoD system.
2	**User Level Intrusion (Incident):** 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.
3	**Unsuccessful Activity Attempted (Event):** 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.
4	**Denial of Service (DOS) (Incident):** Activity that impairs, impedes, or halts normal functionality of a system or network.
5	**Non-Compliance Activity (Event):** 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.
6	**Reconnaissance (Event):** 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.
7	**Malicious Logic (Incident):** Installation of malicious software (e.g., trojan, backdoor, virus, or worm).
8	**Investigating (Event):** 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.
9	**Explained Anomaly (Event):** 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).

Source: Dept of Navy, Cybersecurity Handbook, page 20

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 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

[1] http://www.rand.org/pubs/research_reports/RR430.html
[2] http://www.rand.org/news/press/2014/06/18.html

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?

Minimizing risk vs. Setting an upper bound on risk

- Direct minimizing risk can be achieved by adjusting
  - the number of analysts,
  - their expertise mix,
  - analyst's shift and days-off scheduling,
  - their sensor assignment,
  - Category of alert – analyst workload – time to analyze (input)
- However, which factor(s) to adjust is hard to determine (requires several simulations)
- Running optimization with risk in the objective function is computationally not viable because the solution space is extremely large for these NP Hard problems.
- Instead, we set up an upper bound on risk and determine the optimal settings of the above factors via optimization using metaheuristics.
- Obtain a set of feasible solutions and pick the best (lowest number of analysts, among them the lowest risk).
- A 0% upper bound can also be set, which constitutes the lowest risk attainable.
- The optimization model provides the flexibility to set any upper bound on risk.

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

Research Objective for Optimization

- Objective: Minimize number of personnel and minimize risk
- Subject to following constraints
  - Maintain risk below the upper bound
  - Ensure ≥ 95% analyst utilization
  - Meet the mix (senior, intermediate, junior) specification 20-40% L3, 40-50% L2, and 30-40% L1
  - Number of sensors per analyst constraint
- Outputs
  - Sensor to analyst allocation
  - Total number of analysts and their mix

Sensor

Analyst	n=1	n=2	n=3
i=1	1	1	0
i=2	0	0	1
i=3	1	0	0

Alert Characteristics

- Sensors generate about 15000 alerts per day
  - All alerts are screened by auto-analysis methods and those that are significant by analysts
  - 1% ~ 150/day ~ avg. 6-7 alerts per hr per sensor are important/have different patterns and requires further investigation by analysts ("significant alerts")
  - Generate alert rate/hr using an arrival probability distribution Poisson (6.5) or Uniform (1,13)
  - Average alert generation rate per hr per sensor can be varied (future work), but for the current model it was kept fixed and equal for all sensors

Analyst Characteristics

- Based on training and experience there are 3 levels of analysts – senior L3, intermediate L2, junior L1
- Over a time interval of one hour,
  - a L1 analyst can handle 5 attacks with simplest actions like blocking an IP address, (Avg 12 min per alert)
  - a L2 analyst can handle 7 or 8 attacks with more complicated actions like blocking a server from an external network (Avg 8 min per alert)
  - a L3 analyst can handle 8+ attacks with the most sophisticated actions (Avg 5 min per alert)
- Alert investigation time could follow a probability distribution – Poisson, normal, triangle, beta

Number of Sensors to Analyst

Constraint - 1

- L3 senior – 4-5 sensors are allocated
- L2 intermediate – 2-3 sensors are allocated
- L1 junior – 1-2 sensors are allocated
- Some overlapping is permissible

- Note: The sensor-to-analyst mapping is an output of optimization

System Requirement Parameters

Constraints – 2 to 4

- Upper bound on Risk - Ex: 10% of the significant alerts are not properly analyzed/unanalyzed
- Analyst Utilization
  - Ensure >95% analyst utilization
- Analyst mix in the organization
  - 20-40% L3, 40-50% L2, and 30-40% L1

Inputs

Inputs that were varied for sensitivity analysis

- Number of sensors -10, 25, 50, 75, 100
- Risk % - 5%, 25%, 45%

Inputs that were maintained fixed for the above studies

- Average alert generation rate using
  - Uniform (0,13) distribution, Mean = 6.5/hr , 6.5*24= 156/day
- Analyst characteristics
  - Average alert investigation rate (time to investigate)
- Number of sensors allocated to analysts

- Optimization was solved using Genetic Algorithm heuristics

Research Findings: Optimization without specifying expertise mix

- Multiple sensors to analyst
- All senior L3 analysts were chosen to minimize personnel
  - No L2 and L1 analysts were selected by optimization
- >95% utilization of analyst time
- At 100% alert coverage (0% Risk), analyst/sensor ratio = 0.7
- At 75% alert coverage (25% Risk), analyst/sensor ratio = 0.5

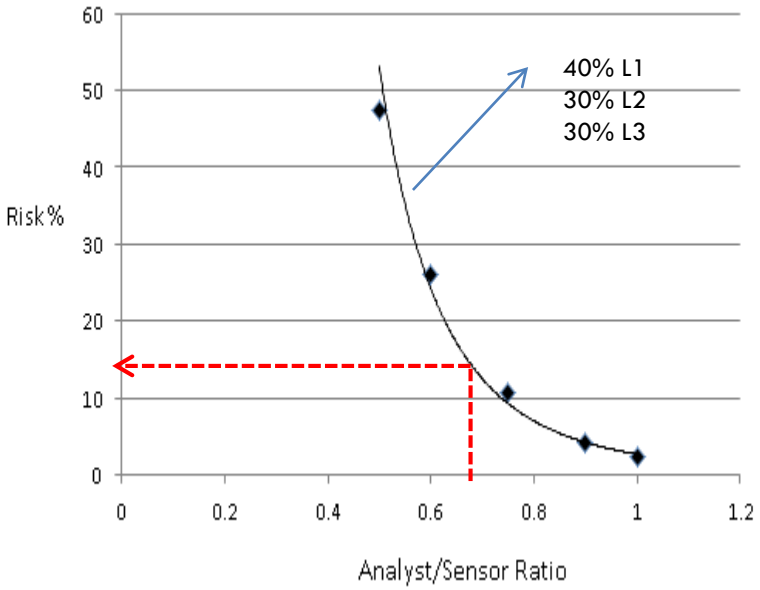
Risk in %	0-5%	25-30%	40-45%
All L3 analysts	7	5	3
Utilization of analyst	>95%	>95%	>95%
Number of sensors	10	10	10
Alert generation rate	U(0,13)	U(0,13)	U(0,13)
Number of sensors per L3 analyst	4-5	4-5	4-5

Research Findings: Optimization specifying expertise mix proportions

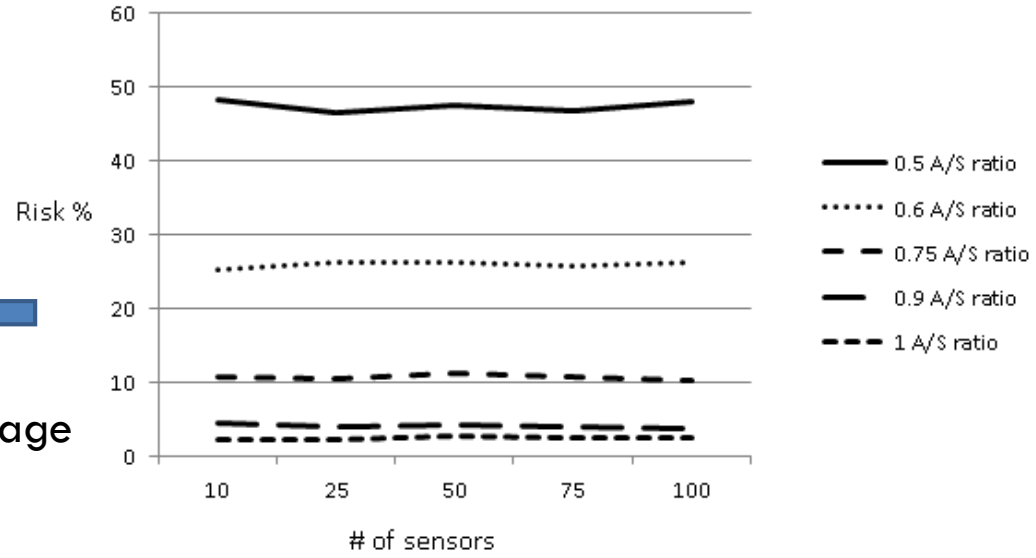
- Multiple sensors to analyst
- Another input – Proportion of L3, L2, and L1 personnel 20-40% L3, 40-50% L2, and 30-40% L1
- >95% utilization of analyst time
- At 100% alert coverage, analyst/sensor ratio = 0.8
- At 75% alert coverage, analyst/sensor ratio = 0.6
- At 55% alert coverage, analyst/sensor ratio = 0.5

Risk in %	0-5%	25-30%	40-45%
Analysts experience mix	2-L1, 3-L2, 3-L3	1-L1, 3-L2, 2-L3	1-L1, 2-L2, 2-L3
Total number of analysts	8	6	5
Utilization of analyst	>95%	>95%	>95%
Number of sensors	10	10	10
Alert generation rate	U(0,13)	U(0,13)	U(0,13)
Number of sensors per L1 analyst	1-2	1-2	1-2
Number of sensors per L2 analyst	2-3	2-3	2-3
Number of sensors per L3 analyst	4-5	4-5	4-5

Main Results



- 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



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

Sample days off Scheduling

- An analyst works 12*6 + 1*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	
2	X	X		X	X	X					X	X			X	X		X	X	X		
3	X	X			X	X					X	X	X		X	X			X	X		
4	X	X				X	X			X			X	X	X	X				X	X	
5	X	X	X				X			X		X		X	X	X	X				X	
6			X	X	X			X	X				X	X			X	X	X			X
7				X	X			X	X	X	X			X				X	X			X
8			X		X	X		X	X		X	X					X		X	X		X
9				X		X	X	X	X			X	X					X		X	X	X
10			X				X	X	X	X			X	X			X				X	X

X – off days

Optimization Recommendations

For an organization that seeks a mix of L3, L2, and L1 analysts

- Use single queue system of alerts in the sensor group
  - When a group of analysts are allocated to a group of sensors by the optimization algorithm, the alerts generated by that group of sensors are arranged in a single queue based on their arrival time-stamp
  - the next available analyst within that group will draw the alerts from the single queue based on a first-in-first-out rule.
- Set proportion of mix L3, L2, L1 level
  - Optimization tends to maximize number of L3 analysts (budget is not considered)
- Do not allocate a sensor only to a junior L1 analyst
  - A junior must be assigned to a sensor that also has a senior L3 person
- All sensors must have at least 1 senior level personnel
- Do not let everyone work on all sensors as an when they become available.
  - The juniors will reduce the overall efficiency of the system.
  - Let optimization decide which junior is paired with a senior and on which sensor.

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

Main Idea behind Dynamic Scheduling

- At the beginning of each 14 day period, determine
  - Static *X*: Number of analysts scheduled to work on a given day (all shifts included and all expertise levels included)
  - Days-off *Y*: Number of analysts who will be on day-off from work on a given day (day-off is guaranteed)

Analyst	14-Days
	X, Y, Z, Y'

- Each day, determine for the next day based on alert estimation
  - Dynamic (on-call) *Z* – Number of analysts scheduled to be on-call (all shifts included and all expertise levels included)
  - Days-off *Y'*: Number of analysts who will on day-off from work

$$X+Y+Z+Y' = W \quad \textit{(total workforce on payroll)}$$

    - Each analyst works 80hrs in 2 weeks. If an analyst comes to work on-call then they will be working more than 80 hrs in 2 weeks.
    - Example: In 14 days, an analyst might work 12*6+8*1 = 80 hrs (7 days of work captured by X matrix), Y matrix guarantees, in advance, 3 days off. The remaining 4 days is split between Z and Y' matrices and is known one day in advance.
  - Each day, a dynamic (on-call) workforce need C will be determined If $0 \leq C \leq Z$ then C will be informed a few hours before their work shift. However, C>Z is allowed and Z will be called in. Optimally design X, Y, Y', and Z

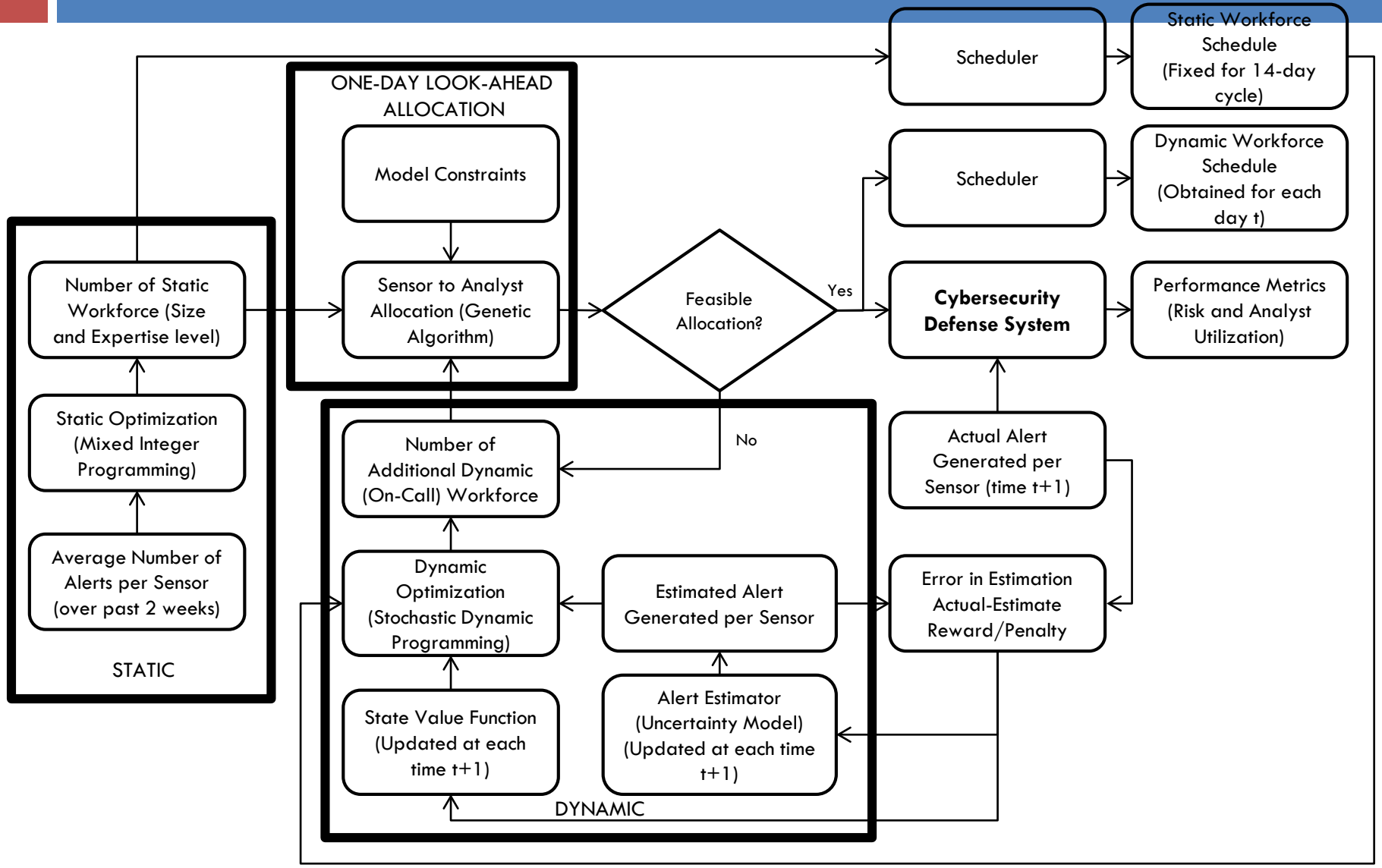
Research Objective for Dynamic Optimization

- Objective: Optimally manage the limited on-call personnel, minimize risk per day, and balance risk over a 14-day period
- Subject to following constraints
  - Ensure ≥ 95% analyst utilization
  - Meet the mix (senior, intermediate, junior) specification 20-40% L3, 40-50% L2, and 30-40% L1
  - Number of sensors per analyst constraint
- Outputs
  - Daily sensor to analyst allocation
  - X, Y, Z, Y' matrices
  - Performance metrics (risk and utilization)

Sensor

Analyst	n=1	n=2	n=3
i=1	1	1	0
i=2	0	0	1
i=3	1	0	0

Optimal Dynamic Scheduling Framework



Research Findings

- 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