A Quantitative Approach to Security Monitor Deployment

Uttam Thakore

PI: William H. Sanders

Problem

- Intrusion detection requires adequate monitoring
  - Monitors must collect sufficient information about intrusions
  - Should support intrusion detection even when monitors are taken out
- Current approaches to monitor deployment do not allow for deployment based on intrusion detection goals
  - Tough to compare different deployment approaches

Another problem

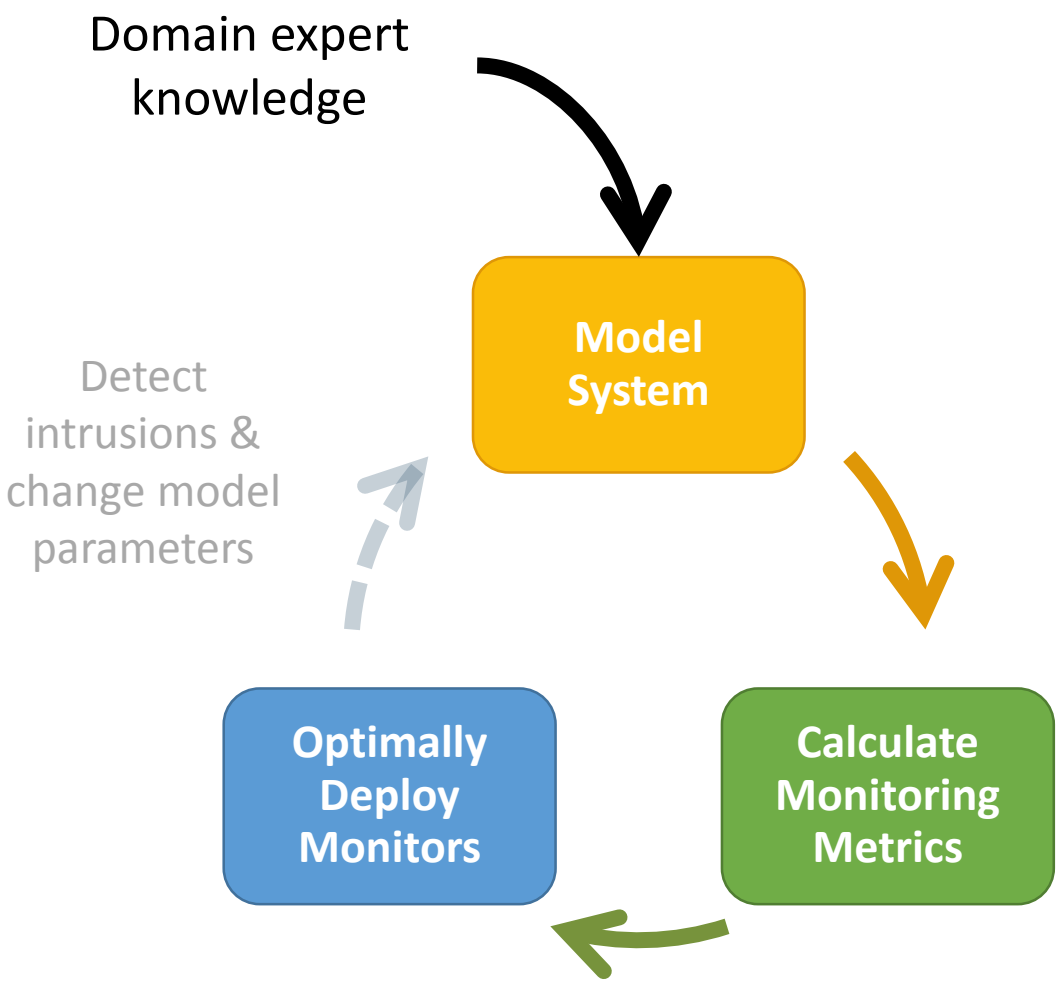
- Monitoring is expensive!
  - Price of monitors
    - NIKSUN NetDetector/NetVCR 2005 – DPI, analytics, and alerting appliance
      Price: $10,000 to $100,000[1]
  - Data storage costs
    - Large enterprises can generate hundreds of GBs of logs *per day*
  - Cost of data analysis and response
    - Average salary of a network security administrator: $65,000[2]
    - Salary of a computer forensic analyst: upwards of $80,000[3]

[1] Jerry Shenk, "NetDetector/NetVCR 2005 Traffic Analyzer," SANS Institute, Whitepaper, Aug. 2007.
[2] "Salary: Network Security Administrator," *Glassdoor*. [Online]. Available: http://www.glassdoor.com/Salaries/network-security-administrator-salary-SRCH_KO0,30.htm.
[3] "Salary: Computer Forensic Analyst," *Glassdoor*. [Online]. Available: http://www.glassdoor.com/Salaries/computer-forensic-analyst-salary-SRCH_KO0,25.htm.

Our contribution

- A **cost-effective** methodology for monitor deployment that **meets intrusion detection goals**

Our approach



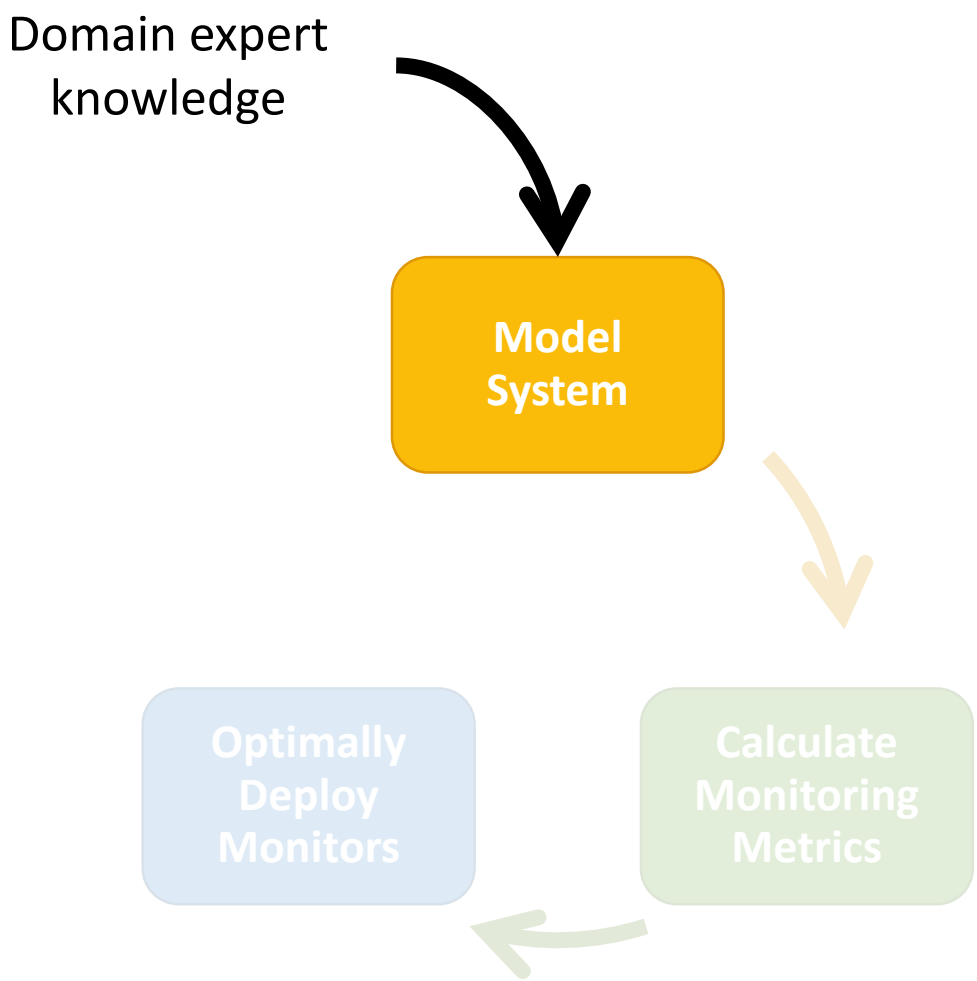
Guiding principles

- Monitors and computing assets can be compromised
  - Monitor compromise can affect ability to detect intrusions
- Monitors should be deployed with redundancy such that the effect of compromise is mitigated

Simplifying assumptions

- Monitors are independent
- Monitors and assets are not currently compromised, but could be compromised in the future
- Monitor costs are independent and additive
  - I.e., cost of deploying *n* monitors = Σ cost of deploying each monitor
- Intrusion detection is ideal

Outline

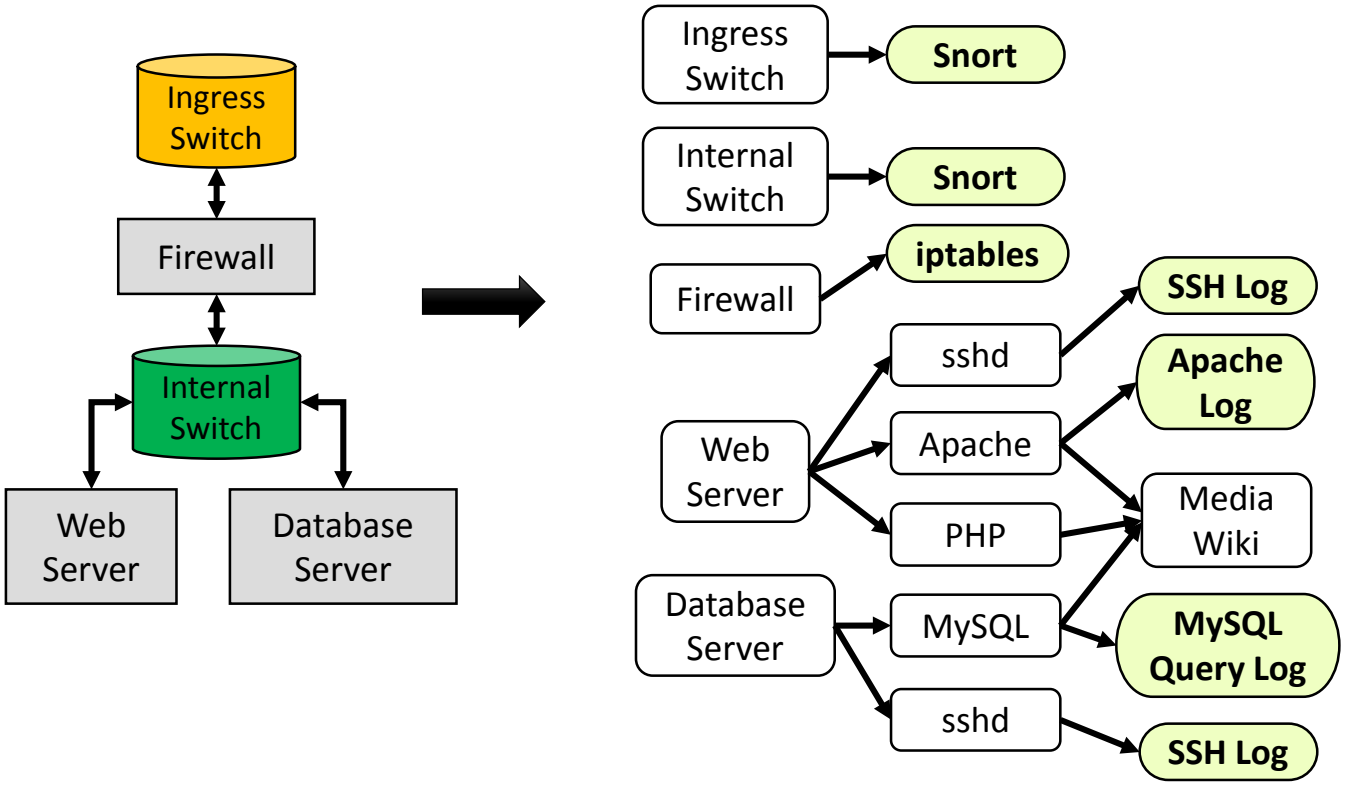

Domain expert knowledge
Model System
Calculate Monitoring Metrics
Optimally Deploy Monitors

Model: System model

- **Assets**: computing components in the system to protect
  - E.g., host machines, servers, virtual machines, applications, network hardware
- **Monitors**: sensors that can be deployed in the system to provide information about intrusions
  - E.g., Snort, Bro, sshd log, Apache request log, MySQL query log
- **Asset-asset dependence**: directed physical dependency between assets, such that correct operation of the second asset relies on correct operation of the first
  - E.g,. hypervisor → tenant VMs
- **Monitor-asset dependence**: dependence relationship between monitors and assets, where compromise of the asset would result in compromise of the monitor
  - E.g., Apache server process → Apache request log

Model: System model illustration

- **Assets**
- **Asset-asset dependence**
- **Monitors**
- **Monitor-asset dependence**



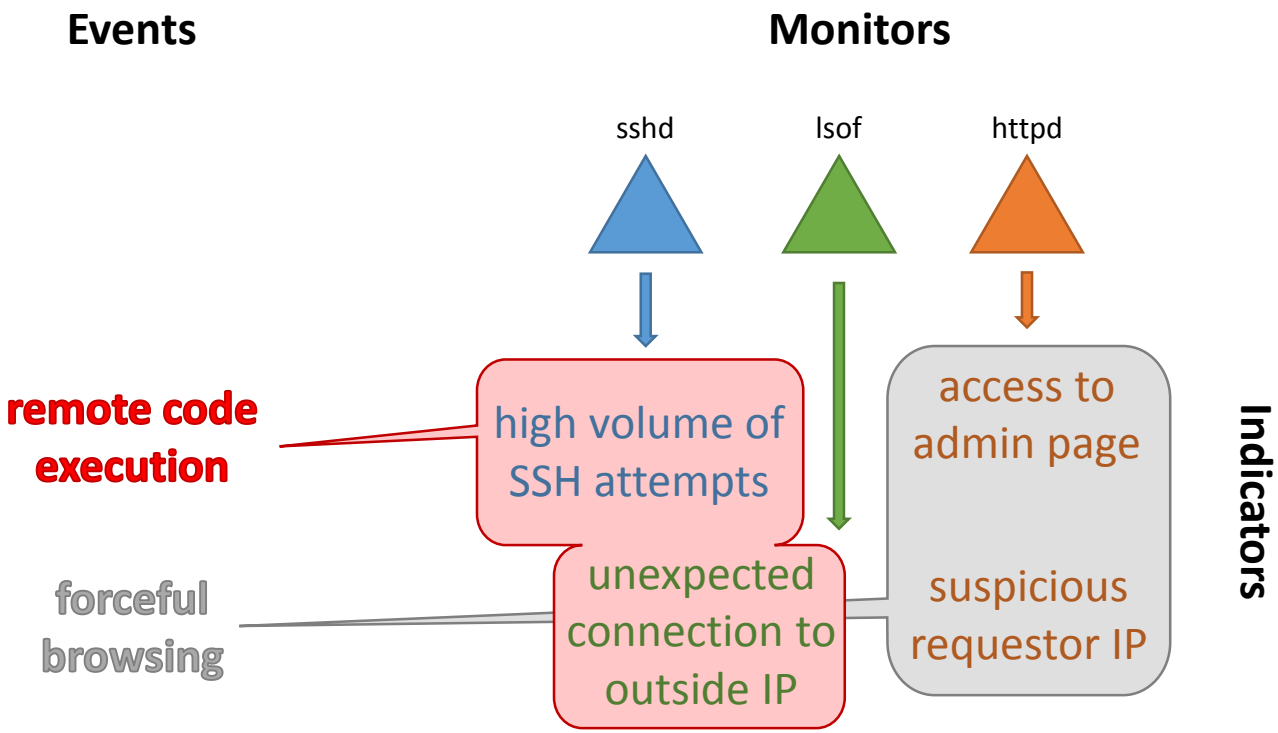
Model: System model (cont'd)

- **Truthfulness of monitors:** what proportion of a monitor's output can be trusted
  - Measure of compromisability of monitors or the assets on which they depend
- **Resource costs of assets:** monetary cost of dedicating computing resources on an asset to monitoring
  Resource types:
  - CPU utilization
  - Memory utilization
  - Disk storage
  - Network communication

Model: Data model

- **Indicators**: primitives representing semantic information provided by monitors about events in the system
  - Generated by monitors
- **Events**: occurrences in the system that are symptomatic of an attack or intrusion or are attacks themselves
  - Events can be detected by observing sets of indicators (similar to an IDS signature)
- **Detectability**: an event is *detectable* if at least one of its indicator sets is observable given the set of deployed monitors

Model: Data model illustration



- Events can map to multiple sets of indicators

Model accounts for coordinated intrusion detection and digital forensics

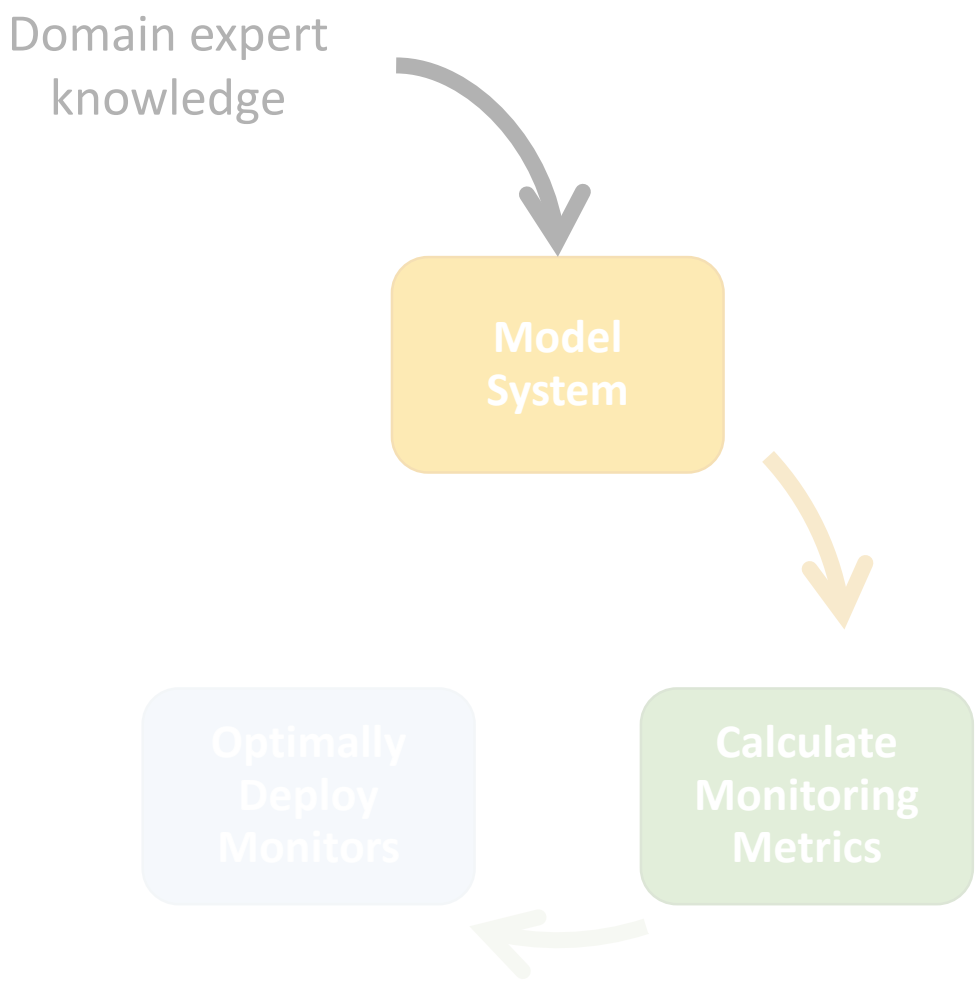
Model: Terminology

- V = set of assets
- M = set of deployable monitors
- E_S = set of asset-asset dependence relationships
- E_M = set of monitor-asset dependence relationships

Model: Terminology

- © = set of events $Á_i$ that we wish to detect
- I = set of observable indicators
- ® (m) = set of indicators that monitor m can generate
- ¯ (Á) = set of indicator sets that can be used to detect event Á
- ±(Á, M_d) = 1 iff an event is detectable given the monitors in M_d

Outline


Domain expert knowledge
Model System
Calculate Monitoring Metrics
Optimally Deploy Monitors

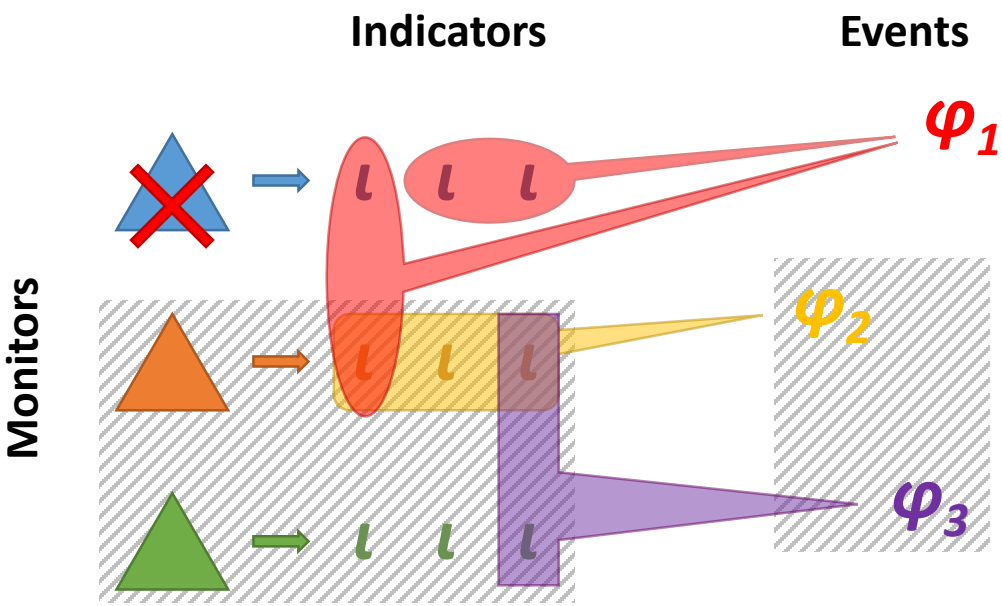
Monitor deployment metrics

- **Goal of metrics**: *quantify* utility and cost of monitors in supporting intrusion detection
- Three monitor utility metrics:
  - Coverage
  - Redundancy
  - Confidence
- One cost metric:
  - Monitor cost

- Developing other metrics for specific analyses

Metrics: Coverage

- **Definition:** overall fraction of events of importance that can be detected given a set of monitors

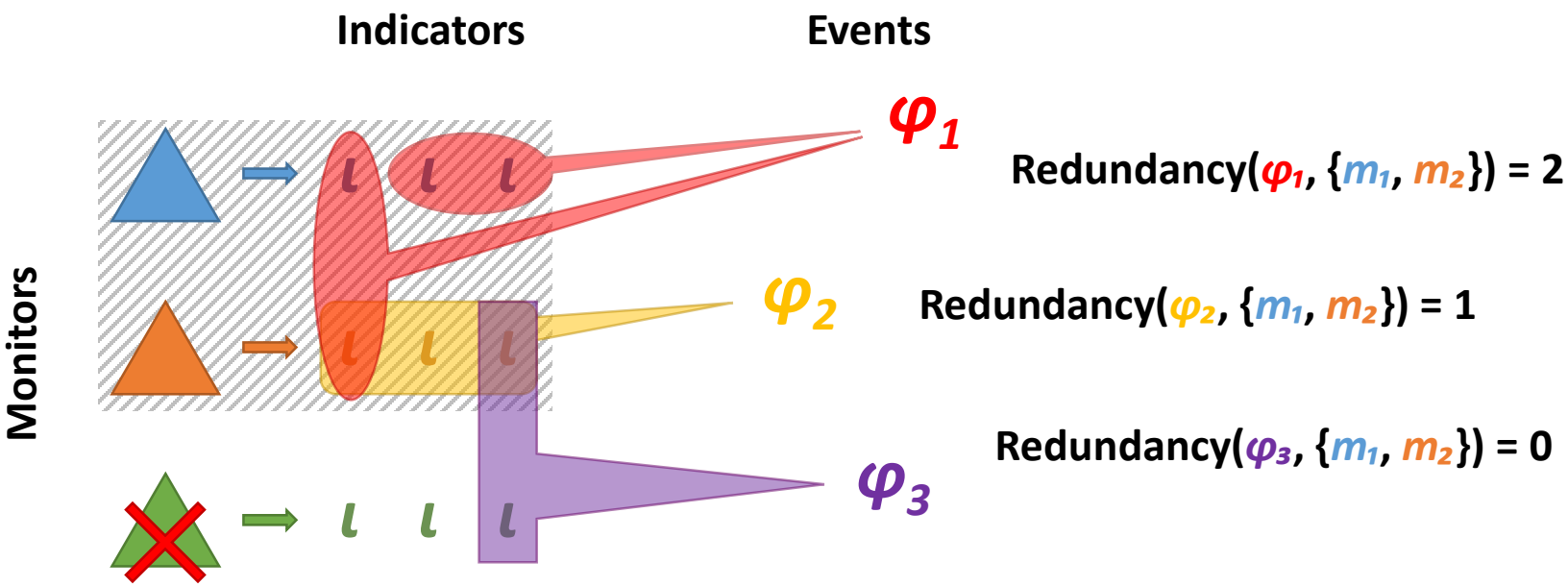


Coverage({φ_1, φ_2, φ_3}, {m_2, m_3}) = 67%

Coverage(©, M_d) = j{Á : ±(Á, M_d) = 1}j ¥ j©j

Metrics: Redundancy

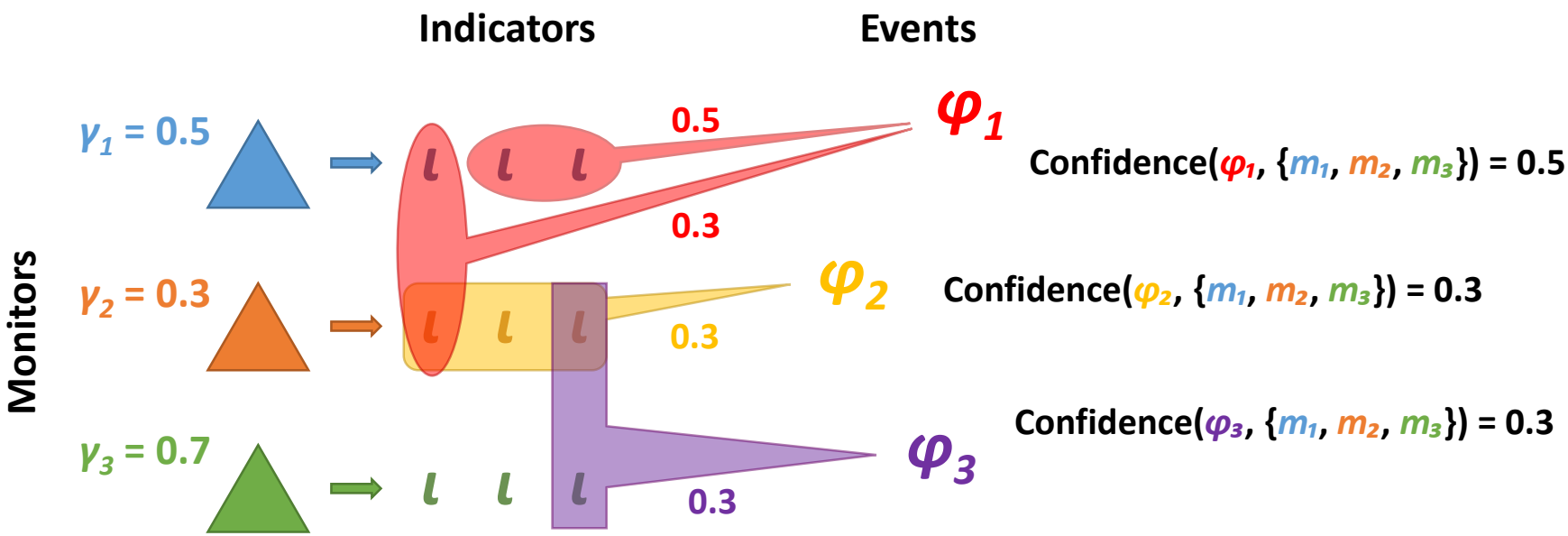
- **Definition:** the number of ways an event can be detected given a set of monitors



Redundancy(Á, M d) = ∑¾: ¾2 ¯(Á), ±(¾ M _d) = 1 min¶2 ¾j{m : m 2 M d, ¶2 ®(m)}j

Metrics: Confidence

- **Definition**: belief in the ability of the monitors to detect events accurately, even when monitors are compromised



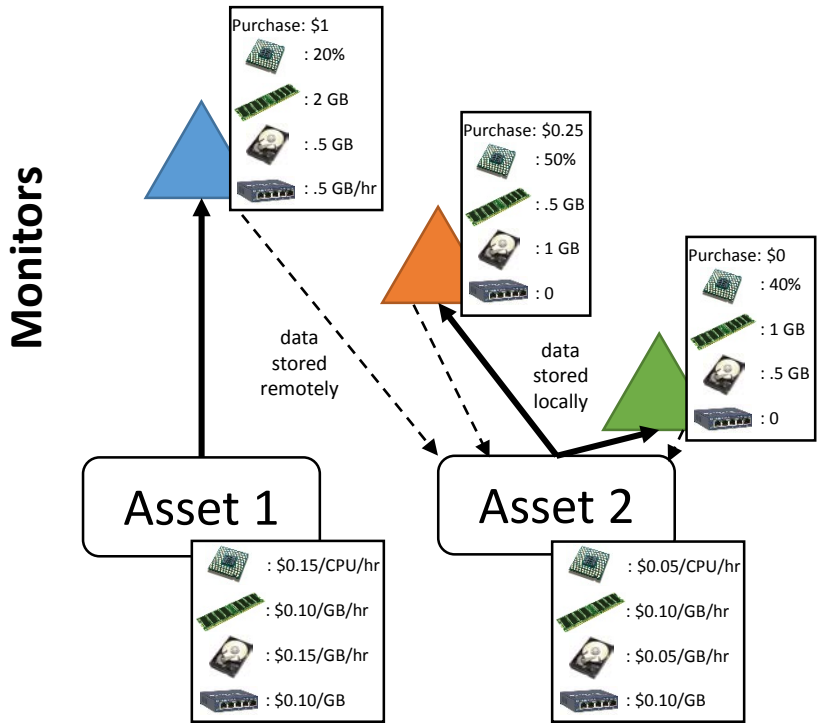
Confidence(Á, M d) = max¾2 ¯(Á) min¶2 ¾ ° I (¶, M d)

Metrics: Cost model

- Each asset has resource costs per unit of use per unit time for each resource type:
  - CPU utilization (e.g., per CPU core per hour)
  - Memory utilization (e.g, per GB per hour)
  - Disk storage (e.g., per TB per hour)
  - Network communication (e.g., per GB)
- Each monitor has:
  - A long-term average resource consumption rate for each resource type
  - An amortized purchase price and recurring maintenance cost per unit time
  - A monitor-asset relationship that describes where monitor data is stored
    - E.g., remote logging

Metrics: Cost

- **Definition:** overall value of the computing resources consumed by monitors that are deployed in the system



Cost(m_1) = \$1.86/hr

Cost(m_2) = \$1.53/hr

Cost(m_3) = \$0.72/hr

Cost({m_1, m_2, m_3}) = \$4.11/hr

$Cost(M_d) = \sum_{m \, 2 \, M\_d}$ (product sum of resource utilization costs + purchase/maintenance costs)

Outline

Domain expert knowledge
Model System
Calculate Monitoring Metrics
Optimally Deploy Monitors

Optimal deployment methodology

- **Goal**: to be able to use methodology to answer a variety of monitor deployment questions
  - What is the minimum set of monitors that can detect a given attack/set of attacks?
  - Under cost constraints, what is the best set of monitors to deploy to maximize detection of a set of attacks?
  - Under cost constraints, what set of monitors will maximize my ability to detect a high-priority attack?

Optimal deployment methodology

- 0-1 integer nonlinear programming problem
- Monitors are input variables
- Utility metrics → objective functions to maximize
  - Parameterized by user-specified weight parameters w
- Cost metric → cost function to minimize

Given:

System model, data model

8 Á 2 ©, $w_{Redundancy_Á}$, $w_{Confidence_Á}$ 2 R

$w_{Coverage}$ 2 R

maxCost 2 R

Maximize:

$\sum_{Á\ 2\ ©}$ ($w_{Redundancy_Á}$ Redundancy(Á, M_d + $w_{Confidence_Á}$ Confidence(Á, M_d)) + $w_{Coverage}$ Coverage(©, M_d)

Subject to:

Cost(M_d) · maxCost

Monitor deployment vector M_d 2 $\{0,1\}^n$

Evaluation

- Using CAPEC attack taxonomy to define representative attack set on web service architecture
- Indicators defined per monitor
  - E.g., SSH logs:
    - Invalid username used
    - Authentication failures > threshold
    - Access from unexpected geo-region
- Defining event-indicator mapping by running attacks individually and observing indicators generated by monitors
- Demonstrating utility of model for various intrusion detection scenarios

Conclusions

- We define a monitor deployment model that allows us to:
  - Model monitors and intrusion detection requirements
  - Compute utility and cost metrics for monitors
- We define a methodology for deploying monitors that allows us to compute optimal monitor deployment under constraints
- Our optimization problem is very expressive

Ongoing and Future Work

- Evaluation
- Metrics that establish the effect of monitor compromise on detection ability on a per-monitor basis
- Accounting for probabilistic fusion algorithms
- Dealing with unknown attacks
- Quantification of dependence of monitors
  - Effect on intrusion detection
- Completion of the deployment loop
  - Accounting for uncertainty in system state

Backup Slides

Evaluation: Experiment