

Balancing a Balanced Budget: making intelligent security budget allocations

Adam Anthony
December 5, 2005
University of Maryland,
Baltimore County

Making the most out of your money

- Number of nodes N such that $P(\text{successful attack}) < t$
- Sum of losses expected from nodes such that $P(s/a) \geq t$
- For nodes whose $P(s/a) \geq t$, the number of nodes that can be compromised in an attack
- The cost effectiveness of spending on a node is $\sum (P(s/a) - P(s/a)') * L$

Outline

- Motivation
- Related work
- New and Significant
- Formal Description
- Overview of (Aspnes, 2004)
- Problems with Assumptions
- Alterations to (Aspnes, 2004)
- Example
- Discussion
- Results
- Open Problems
- Conclusion

Motivation

- Economic vs. Technological
- Remove the guesswork
- Increase the value of every security dollar spent

Computer Security Metrics

- “If you haven't calibrated the model with measurement, only one thing is certain: You will either overspend or under-protect.” (Geer 2003)
- Metrics Remove FUD (Fear, Uncertainty, Doubt)
- Host-level
- Network level

Related Work

- (Hamilton 2002)
 - Problems in applying game theory to information security
- (Kephart 1993)
 - Epidemiologically based security decisions
- (Nikoletseas 2003)
 - Probability of failure model
- (Aspnes 2004)
 - Game-theoretic Graph-based algorithm

Innovation

- Novel: Counter-intuitive results showing an effect of overspending on a node
- New: First research to attempt a budget-oriented approach to making security decisions
- New: First research to define a network-wide security metric
- Significance: the ability to measure the security of a network allows network administrators to choose new protections wisely, maximizing their spending potential

Formal Description

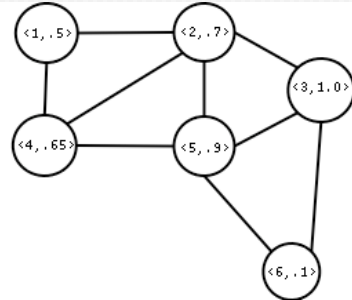
- System Administrator VS. Hacker
- One Round
- System Admin: allocate a finite budget B throughout his network
- Hacker: randomly choose an entry point and attack



Formal Description, cont.

- Spending affects the likelihood of the hacker's success
- Hacker stops when
 - He fails on all paths
 - All nodes are compromised
- System Administrator is successful by minimizing the number of infected nodes

Example Graph



Aspnes, Chang and Yampolskiy's Model

- Represent a network as an undirected graph $G = (V, E)$
- Create a strategy vector \mathbf{a}
- Let $a_i = \text{Prob}[\text{Node } i \text{ installs anti-virus}]$
- Pure strategy implies all components of \mathbf{a} are either 1 or 0
- Mixed strategies allow for values $[0, 1]$
- Common C (cost of software) for all nodes
- Common L (cost of infection) for all nodes

Bad Assumptions

- Every infected node infects all unprotected neighbors
- Costs are all the same
- The virus imposes no costs on protected nodes
- Nodes can form a strategy based on other nodes

Alterations to (Aspnes,2004)

- Let C and L be variable
 - $C = \sum c_i$ where each c_i is a cost of an individual security resource
 - L can be higher for more important machines
- Let $B = \sum C_i$ represent the budget
- Let a_i represent the probability that an attack on node i will succeed
- Let $a_i = f(C_i)$ where C_i is the sum of the costs of resources installed to protect node i

More Alterations

The cost of a mixed strategy $\vec{a} \in [0,1]^n$ to node i is

$$\text{cost}_i(\vec{a}) = C_i + L \cdot p_i(\vec{a})$$

where $p_i(\vec{a})$ is the probability of node i being infected given \vec{a}

Calculating p_i

- $p_i(\mathbf{a}) = \max_j (a_j/n, a_j/n * \sum R_{ijp}(\mathbf{a}))$ for all j , where the summation is over all paths p and:

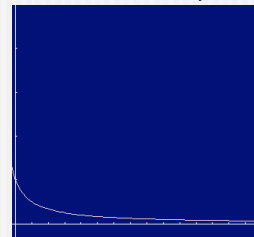
$$R_{ijp} = \left(\prod_p a_p \right)$$

where p is a set of nodes on a path from i to j

- *note: 1. Node j not included in any set p ,
2. No path from i to j is cyclical

A potential function for a_i

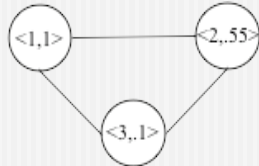
$$f(C_i) = 1 - \frac{C_i}{C_i + 1}$$



Example

- $a_i = 1 - \frac{C_i}{C_i + 1}$

- $G = (V, E)$
 - $V = \{1, 2, 3\}$
 - $E = \{(1, 2), (1, 3), (2, 3)\}$
- $S = \{C_1, C_2, C_3\}$



$$p_3(<0, .55, .1>) = \max\left(\frac{.1}{3}, \frac{.55}{3} * (.1 + .1), \frac{1}{3} * (.1 + .55 * .1)\right)$$

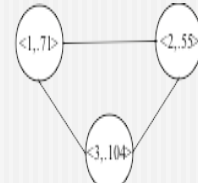
$$= \max(0.033, 0.037, 0.052)$$

$$= 0.052$$

Example Variation

- $a_i = 1 - \frac{C_i}{C_i + 1}$

- $G = (V, E)$
 - $V = \{1, 2, 3\}$
 - $E = \{(1, 2), (1, 3), (2, 3)\}$
- $S = \{C_1, C_2, C_3\}$



$$p_3(<.71, .55, .104>) = \max\left(\frac{.104}{3}, \frac{.55}{3} * (.71 * .104 + .104), \frac{.71}{3} * (.104 + .55 * .104)\right)$$

$$= \max(0.0347, 0.033, 0.038)$$

$$= 0.038$$

The Good Sibling's Paradox

- Imbalance in Security Decisions
- Ignoring machines in favor of others



Results

- Number of nodes $p_i(\mathbf{a}) < t$
- If a node exists s.t. $p_i(\mathbf{a}) \geq t$, then measure the number of nodes reachable from i with a probability $\leq t$
- as $\sum L$ where $L = \{L_i \mid p_i(\mathbf{a}) > t\}$
- cost effectiveness = $\sum(p_i(\mathbf{a}) - p_i(\mathbf{a}')) * L_i$

Finding the most cost effective decision

- Answer the question of where to spend one more unit of a security resource
- Maximize $\sum (p_i(\mathbf{a}) - p_i(\mathbf{a}')) * L_i$
- Calculating $p_i(\mathbf{a})$ is hard
- Re-calculating $p_i(\mathbf{a})$ for all nodes N times
- Dynamic programming may reduce computations
- Computations can also be done in parallel
- Distributed AI agents

Open Problems

- Determining a good function for producing a_i values
- Fast algorithm for finding the optimal node for cost-effective spending
- What if the hacker can choose his entry point?
- What about multiple rounds?

Conclusion

- Model is adaptable
- Model demonstrates a common paradox in security spending
- Progress in host-level security metrics will make this metric a useful one
- Security professionals would have a resource that complements their intuition