Attack Trees: semi-adaptive model

Aivo Jürgenson[2,3] Jan Willemson[1]

[1]Cybernetica, Tartu, Estonia

[2]Tallinn University of Technology, Tallinn, Estonia

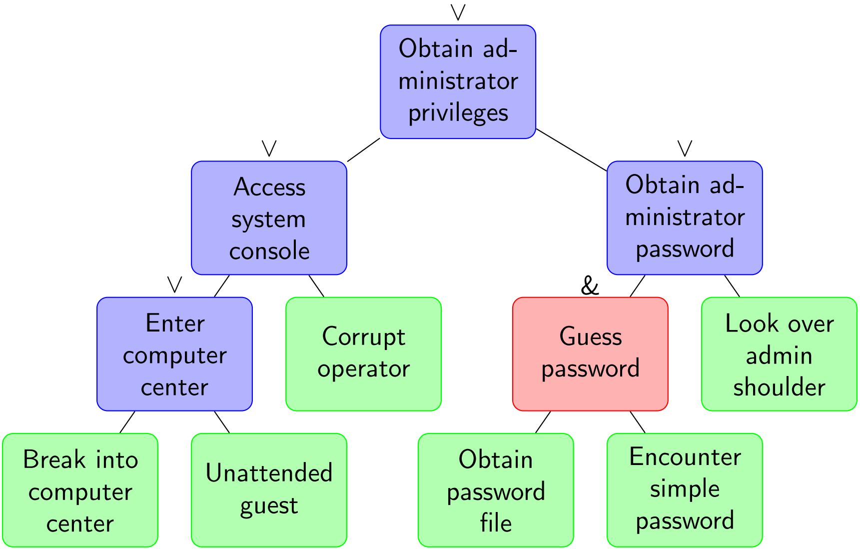
[3]Elion Enterprises Ltd, Tallinn, Estonia

1st February 2009

Outline of the talk

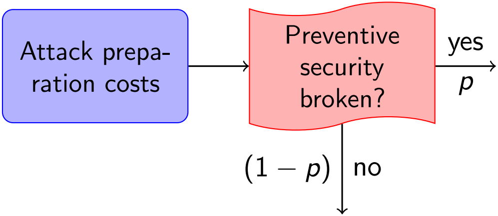
1. Introduction to multi-parameter attack trees
2. Semi-adaptive model
3. Semi-adaptive blocking model
4. Results and Questions

Attack trees (J. D. Weiss 1991, B. Schneier 1999)

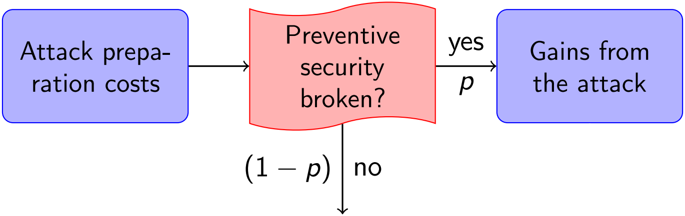


Attacker financial game (A. Buldas et al. 2006)

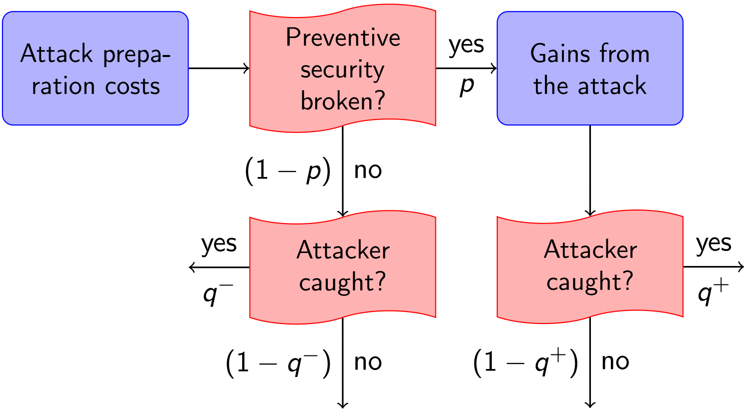
Attacker financial game (A. Buldas et al. 2006)



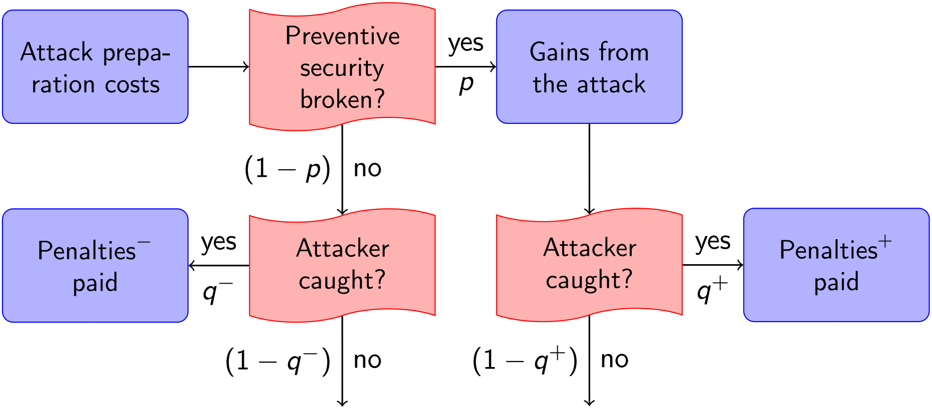
Attacker financial game (A. Buldas et al. 2006)



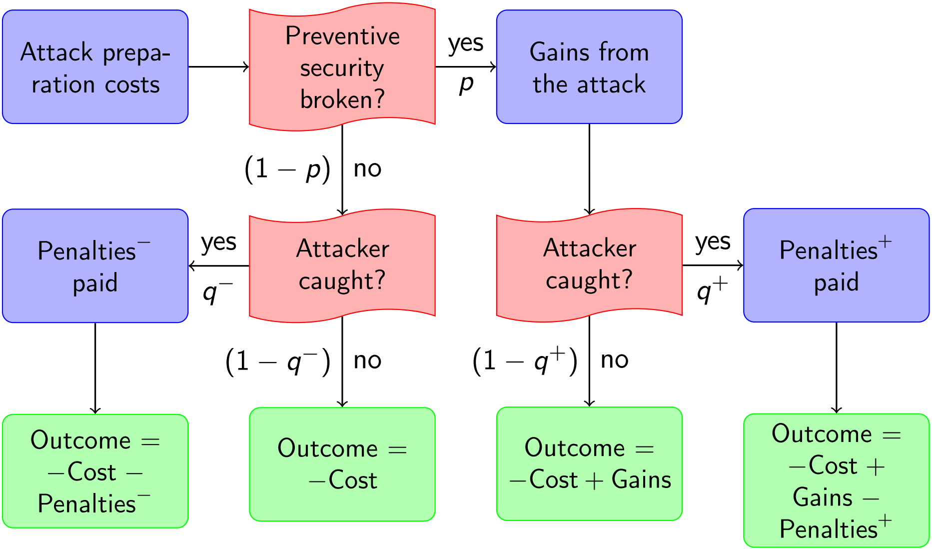
Attacker financial game (A. Buldas et al. 2006)



Attacker financial game (A. Buldas et al. 2006)



Attacker financial game (A. Buldas et al. 2006)



Multi-parameter Attack Trees (A. Buldas et al., 2006)

- Gains – the value gained from the successful attack
- Cost_i – the cost of the elementary attack, p_i – success probability
- $\pi_i^- = q_i^- \cdot \text{Penalty}_\text{i}^-$ – the expected penalty, unsuccessful attack
- $\pi_i^+ = q_i^+ \cdot \text{Penalty}_\text{i}^+$ – the expected penalty, successful attack

Multi-parameter Attack Trees (A. Buldas et al., 2006)

- Gains – the value gained from the successful attack
- Cost_i – the cost of the elementary attack, p_i – success probability
- $\pi_i^- = q_i^- \cdot \text{Penalty}_\text{i}^-$ – the expected penalty, unsuccessful attack
- $\pi_i^+ = q_i^+ \cdot \text{Penalty}_\text{i}^+$ – the expected penalty, successful attack

$$(\text{Cost}, p, \pi^+, \pi^-) = \begin{cases} (\text{Cost}_1, p_1, \pi_1^+, \pi_1^-), & \text{if } \text{Outcome}_1 > \text{Outcome}_2 \\ (\text{Cost}_2, p_2, \pi_2^+, \pi_2^-), & \text{if } \text{Outcome}_1 \leq \text{Outcome}_2 \end{cases}$$

$$\text{Outcome}_i = p_i \cdot \text{Gains} - \text{Cost}_i - p_i \cdot \pi_i^+ - (1 - p_i) \cdot \pi_i^-$$

Multi-parameter Attack Trees (A. Buldas et al., 2006)

- Gains – the value gained from the successful attack
- Cost_i – the cost of the elementary attack, p_i – success probability
- $\pi_i^- = q_i^- \cdot \mathrm{Penalty}_\mathrm{i}^-$ – the expected penalty, unsuccessful attack
- $\pi_i^+ = q_i^+ \cdot \mathrm{Penalty}_\mathrm{i}^+$ – the expected penalty, successful attack

$$(\mathrm{Cost}, p, \pi^+, \pi^-) = \begin{cases} (\mathrm{Cost}_1, p_1, \pi_1^+, \pi_1^-), & \text{if } \mathrm{Outcome}_1 > \mathrm{Outcome}_2 \\ (\mathrm{Cost}_2, p_2, \pi_2^+, \pi_2^-), & \text{if } \mathrm{Outcome}_1 \leq \mathrm{Outcome}_2 \end{cases}$$

$$\mathrm{Outcome}_i = p_i \cdot \mathrm{Gains} - \mathrm{Cost}_i - p_i \cdot \pi_i^+ - (1 - p_i) \cdot \pi_i^-$$

$$\begin{aligned} \mathrm{Cost} &= \mathrm{Cost}_1 + \mathrm{Cost}_2, \quad p = p_1 \cdot p_2, \quad \pi^+ = \pi_1^+ + \pi_2^+, \\ \pi^- &= \frac{p_1(1-p_2)(\pi_1^+ + \pi_2^-) + (1-p_1)p_2(\pi_1^- + \pi_2^+)}{1 - p_1 p_2} + \\ &\quad + \frac{(1-p_1)(1-p_2)(\pi_1^- + \pi_2^-)}{1 - p_1 p_2} \end{aligned}$$

Attacker *adaptiveness*

- Current models all assume that all attacks take place simultanously, in the same time.
- In the real life, attacker has the option to choose different strategy during the execution of attack tree, after some elementary attack succeeds, or fails.

Attacker *adaptiveness*

- Current models all assume that all attacks take place simultanously, in the same time.
- In the real life, attacker has the option to choose different strategy during the execution of attack tree, after some elementary attack succeeds, or fails.
- Full-adaptive model
  - attacker can choose any not-used attack for the next step,
  - rather complicated to analyze, we will not go there.

Attacker *adaptiveness*

- Current models all assume that all attacks take place simultanously, in the same time.
- In the real life, attacker has the option to choose different strategy during the execution of attack tree, after some elementary attack succeeds, or fails.
- Full-adaptive model
  - attacker can choose any not-used attack for the next step,
  - rather complicated to analyze, we will not go there.
- Semi-adaptive model
  - attacker fixes the order of the attacks,
  - attacker has the option to skip some attacks from the previously fixed order.

Semi-adaptive model

Simplified attacker actions:

- Create the attack tree $\mathcal{F}$ with the set of elementary attacks $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.

Semi-adaptive model

Simplified attacker actions:

- Create the attack tree $\mathcal{F}$ with the set of elementary attacks $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.
- Choose subset $S \subseteq \mathcal{X}$ and create the subtree, i.e. choose one possible way of realizing the attack tree $\mathcal{F}$.

Semi-adaptive model

Simplified attacker actions:

- Create the attack tree $\mathcal{F}$ with the set of elementary attacks $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.
- Choose subset $S \subseteq \mathcal{X}$ and create the subtree, i.e. choose one possible way of realizing the attack tree $\mathcal{F}$.
- Choose the permutation α for the subset S, i.e. choose the order of the attacks, eq $\alpha = \{X_2, X_3, X_1\}$.

Semi-adaptive model

Simplified attacker actions:

- Create the attack tree $\mathcal{F}$ with the set of elementary attacks $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.
- Choose subset $S \subseteq \mathcal{X}$ and create the subtree, i.e. choose one possible way of realizing the attack tree $\mathcal{F}$.
- Choose the permutation α for the subset S, i.e. choose the order of the attacks, eq $\alpha = \{X_2, X_3, X_1\}$.
- Evaluate the outcome of the subtree S and permutation α.

Semi-adaptive model

Simplified attacker actions:

- Create the attack tree $\mathcal{F}$ with the set of elementary attacks $\mathcal{X} = \{X_1, X_2, \ldots, X_n\}$.
- Choose subset $S \subseteq \mathcal{X}$ and create the subtree, i.e. choose one possible way of realizing the attack tree $\mathcal{F}$.
- Choose the permutation α for the subset S, i.e. choose the order of the attacks, eq $\alpha = \{X_2, X_3, X_1\}$.
- Evaluate the outcome of the subtree S and permutation α.
- Choose the maximum outcome for all different combinations of permuations α and subtrees S.

Evaluating the outcome of attack tree

$$\mathsf{Outcome}_{semiadaptive} = \max\{\mathsf{Outcome}_\alpha : S \subseteq \mathcal{X}, \mathcal{F}(S := \mathsf{true}) = \mathsf{true}, \alpha\}$$

$$\mathsf{Outcome}_\alpha = p_\alpha \cdot \mathsf{Gains} - \sum_{i=1}^{n} p_{\alpha,i} \cdot \mathsf{Expenses}_i$$

Evaluating the outcome of attack tree

$$\text{Outcome}_{semiadaptive} = \max\{\text{Outcome}_\alpha : S \subseteq \mathcal{X}, \mathcal{F}(S := \text{true}) = \text{true}, \alpha\}$$

$$\text{Outcome}_\alpha = p_\alpha \cdot \text{Gains} - \sum_{i=1}^{n} p_{\alpha,i} \cdot \text{Expenses}_i$$

Theorem:

$$\text{Outcome}_{\text{semiadaptive}} \geq \text{Outcome}_{\text{JW08}} \geq \text{Outcome}_{\text{Buldas06}}$$

Algorithm 1: Evaluating the outcome of permutation α

Data: Variables A, variable counter i, path probability p

Result: *sum* - outcome of the permutation α

$sum := 0$;

if *evaluating $\mathcal{F}(A)$ and in the path from leaf $\mathcal{X}_{\alpha(i)}$ to root of the tree, some node will get value t or f* **then**

  compute_outcome$(A, i+1, p)$;**return** *sum*;

$A[\alpha(i)] :=$ t; **if** $\mathcal{F}(A) =$ t **then**

  $sum := sum + p \cdot p_{\alpha(i)} \cdot \left[\text{Gains} - \sum_{j \in A}(\text{Cost}_j + \pi_i^j)\right]$;

else

  compute_outcome $\left(A, i+1, p \cdot p_{\alpha(i)}\right)$;

$A[\alpha(i)] :=$ f; **if** $\mathcal{F}(A) =$ f **then**

  $sum := sum + p \cdot (1 - p_{\alpha(i)}) \cdot \left[- \sum_{j \in A}(\text{Cost}_j + \pi_i^j)\right]$;

else

  compute_outcome $\left(A, i+1, p \cdot \left(1 - p_{\alpha(i)}\right)\right)$;

return *sum*;

Algorithm 2: Evaluating the probability $p_{\alpha(i)}$

Data: Variables $\{X_1, \ldots, X_n\}$, permutation α

Result: $p_{\alpha,i}$ - probability of the permutation α

forall *node* Z *in* $\{X_1, \ldots, X_n\}$ **do**

  $Z.t := 0$; $Z.f := 0$;

for $i := 1$ **to** n **do**

  Find the path $(Y_0, Y_1, \ldots, Y_m)$ from the root Y_0 to leaf $Y_m = X_{\alpha(i)}$;

  $p_{\alpha,\alpha(i)} = \prod_{i=1}^{m} (1 - Z_j.a)$;

  (where Z_j is the second subnode of the node Y_{j-1} after the node Y_j and $a = \begin{cases} \text{t} & \text{if } Y_{j-1} \text{ is OR-node} \\ \text{f} & \text{if } Y_{j-1} \text{ is AND-node} \end{cases}$);

  $X_{\alpha(i)}.t := p_{\alpha(i)}$;

  $X_{\alpha(i)}.f := 1 - p_{\alpha(i)}$;

  Update the parameters for the nodes $\{Y_{m-1}, Y_{m-2}, \ldots, Y_0\}$;

return $p_{\alpha,i}$;

Computational complexity

- Computing the outcome of permutation (Algorithm 1) has exponential complexity.
- Computing the probability $p_{\alpha,i}$ (Algorithm 2) is efficient.
- All together, for finding the best outcome, we have something in the order of
$$O(2^n \cdot n! \cdot n^2)$$

Semi-adaptive blocking model

- We also consider elementary attacks, which block the whole attack tree, when they fail.
- The real life analogue for capturing the attacker, imprisonment or death penalty.
- Algorithms 1 and 2 require only a slight change.
- However, the complexity for computing $p_{\alpha,\alpha(i)}$ becomes also exponential and therefore the model is even more difficult to compute.

Results and Questions

Results:

- We have yet another way to compute the outcome of the attack tree, which yields even bigger outcomes.
- The model unfortunately has exponential complexity, again.

Results and Questions

Results:

- We have yet another way to compute the outcome of the attack tree, which yields even bigger outcomes.
- The model unfortunately has exponential complexity, again.

Questions:

- Applying theorems from the last article (Jürgenson and Willemson, 2008) to this model as well and optimizing the computions?
- Applying genetic programming concepts to attack trees and outcome computions?
- Learning Bayesian networks to come up with other interesting models?