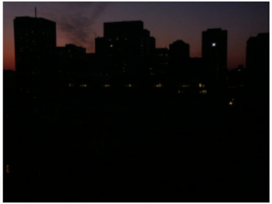
Formal modeling and quantitative analysis of security using attack–defense trees

Wojciech Wideł

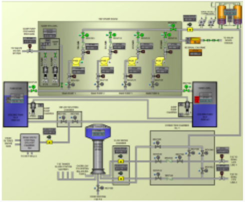
INSA Rennes, IRISA
Supervisor: Barbara Fila
Thesis director: Gildas Avoine

Motivation

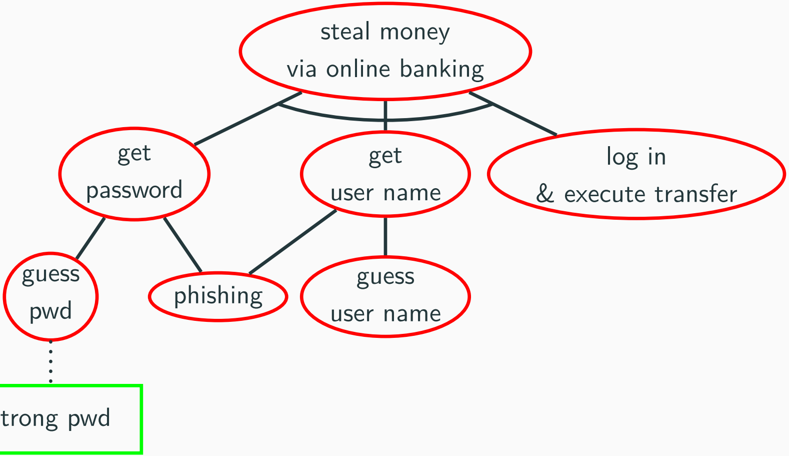
- Need for security
- Prepare for the worst
  - by speculating about possible **attacks** and their **likelihoods**
  - by speculating about possible **countermeasures** and their influence on security

Practical challenges

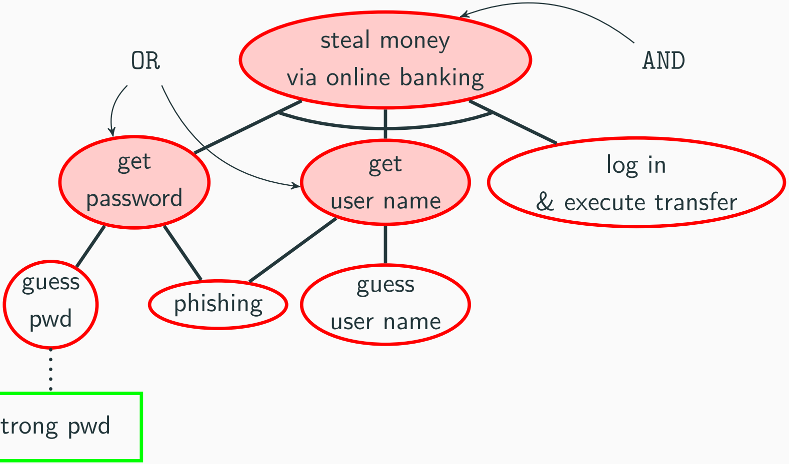


- Complex systems
- Complex threat landscape

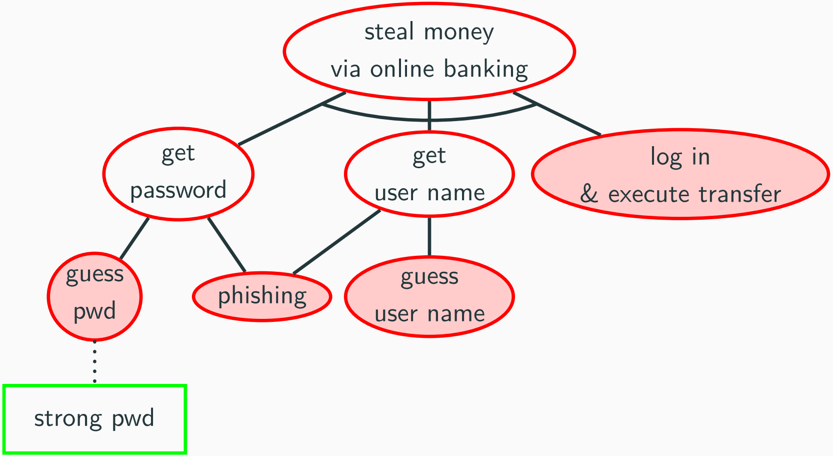
Security modeling with attack–defense trees



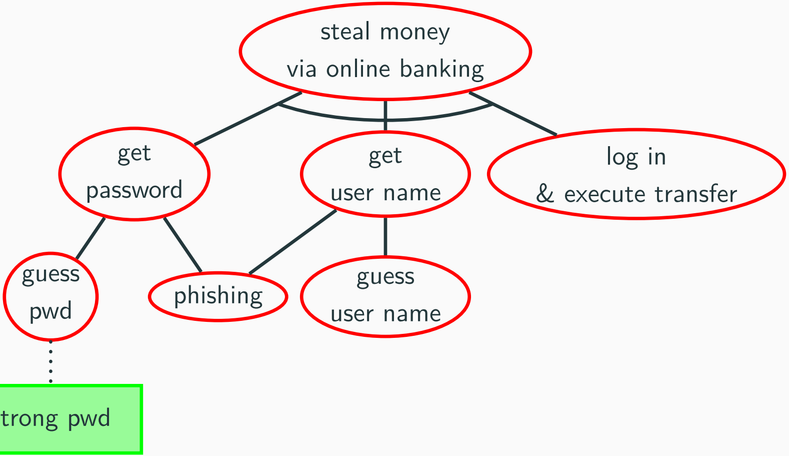
Attacker goals



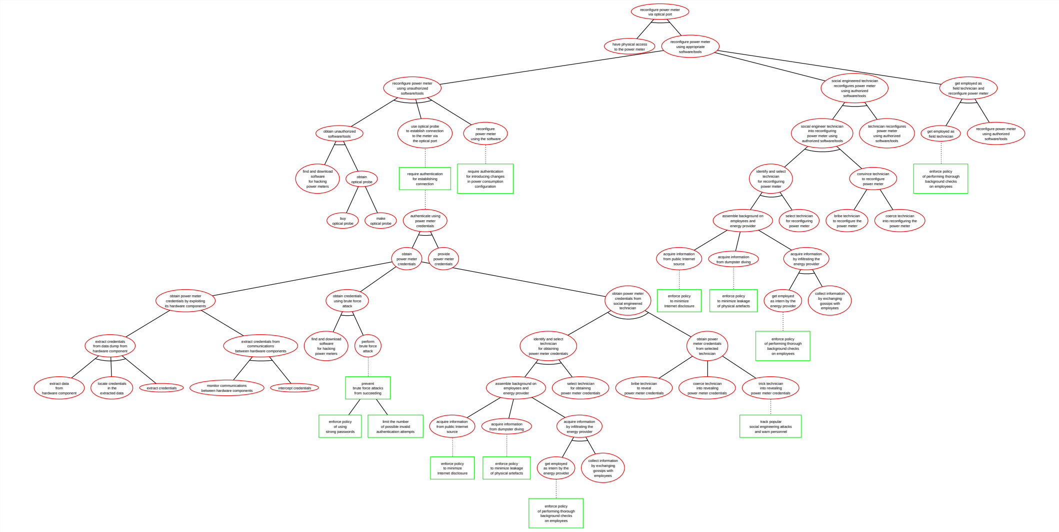
Basic actions



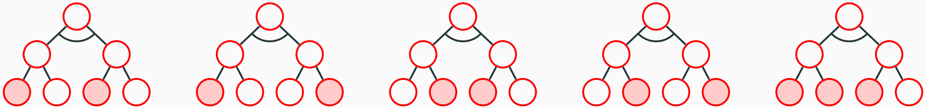
Countermeasures



A bigger example

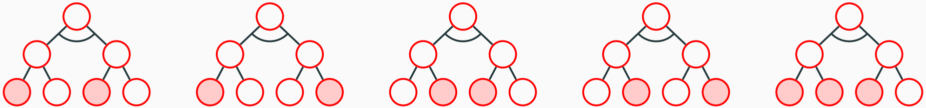


The fundamental questions



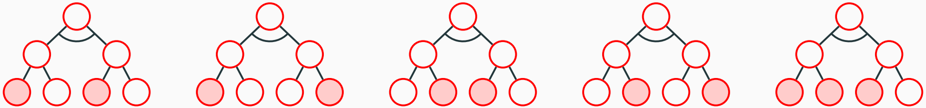
- Which attacks are the **most likely** to occur?

The fundamental questions



- What are the **optimal attacks**?

The fundamental questions



- What are the **optimal attacks**?
- How to **counter these attacks**?

Problem 1: Repeated basic actions (clones)

Problem 1: Repeated basic actions (clones)

Research question 1

How to determine optimal attacks efficiently in the presence of clones?

Problem 2: Multi-parameter analysis

- Attacks requiring **high skills** can be **disastrous**.
- Attacks **easy** to mount might have **low success probability**.

Research question 2

How to determine efficiently attacks optimal w.r.t. to multiple parameters?

Problem 3: Optimal security

- **Limited resources** (time, money, etc.)
- **Big pool** of available countermeasures
- **Dependencies** between countermeasures and potential attacks
- Need to **prioritize**

Problem 3: Optimal security

- **Limited resources** (time, money, etc.)
- **Big pool** of available countermeasures
- **Dependencies** between countermeasures and potential attacks
- Need to **prioritize**

Research question 3

How to determine efficiently sets of optimal countermeasures?

Research questions

Research question 1

How to determine optimal attacks efficiently in the presence of clones?

Research question 2

How to determine efficiently attacks optimal w.r.t. to multiple parameters?

Research question 3

How to determine efficiently sets of optimal countermeasures?

Contributions

- Theoretical developments
  - Analysis of attacks in the presence of **clones** (POST'18)
  - **Multi-parameter analysis** of attacks (CSF'19)
  - Selection of **optimal sets of countermeasures** (iFM'17, paper under submission)
- Practical contributions
  - **Overview** of recent developments in the field (ACM Comput. Surv. 2019)
  - **Tool support** and realistic **case study** (GraMSec'19)

Focus of this talk

- Theoretical developments
  - Analysis of attacks in the presence of **clones** (POST'18)
  - **Multi-parameter analysis** of attacks (CSF'19)
  - Selection of **optimal sets of countermeasures** (iFM'17, paper under submission)
- Practical contributions
  - **Overview** of recent developments in the field (ACM Comput. Surv. 2019)
  - **Tool support** and realistic **case study** (GraMSec'19)

Outline of the rest of the talk

Analysis of attacks in the presence of clones

Multi-parameter analysis of attacks

Other contributions

Future work

Outline of the rest of the talk

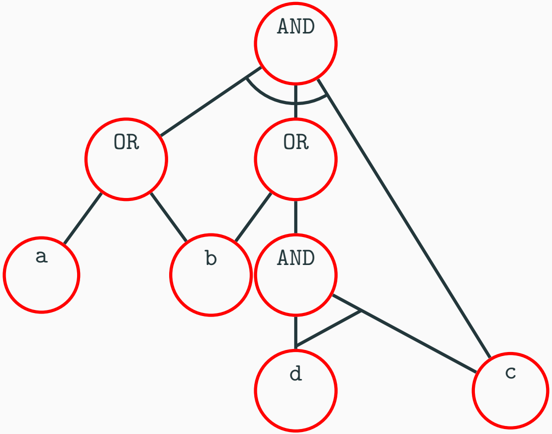
Analysis of attacks in the presence of clones

Multi-parameter analysis of attacks

Other contributions

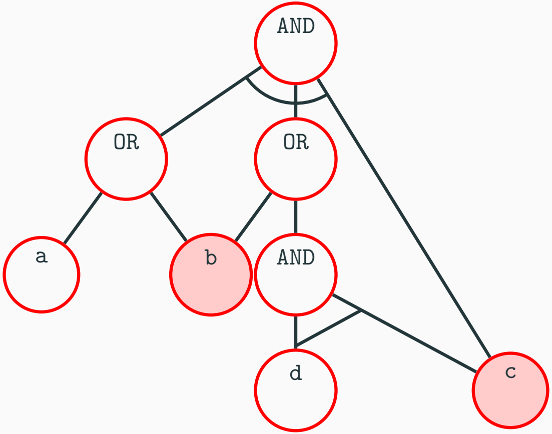
Future work

Example: minimal cost of attack in attack trees



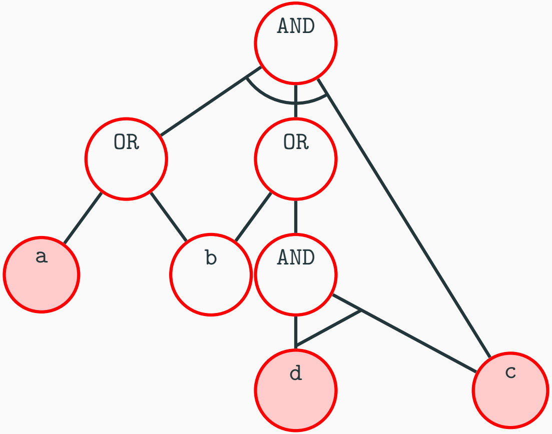
Attack = minimal set of actions achieving the root goal

Attacks



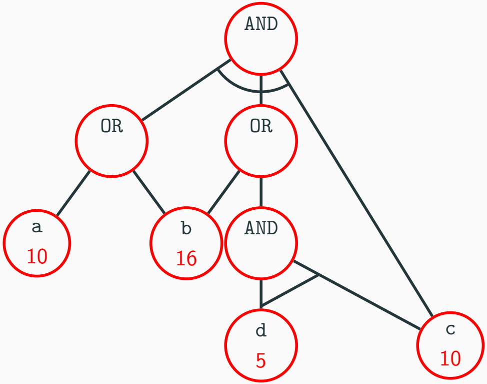
$\{b, c\}$

Attacks



$\{b, c\}, \{a, c, d\}$

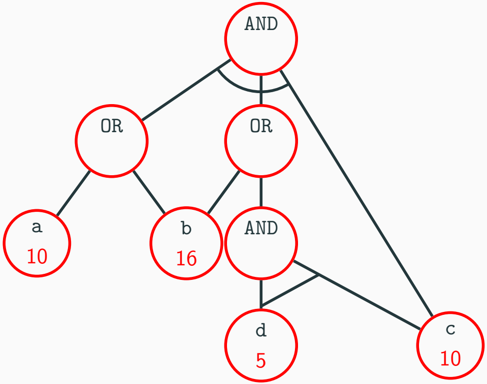
Minimal cost via an extraction of attacks



$\{\mathtt{b}, \mathtt{c}\}, \{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$

$\min\{16 + 10, 10 + 10 + 5\} = 25$

Minimal cost via an extraction of attacks



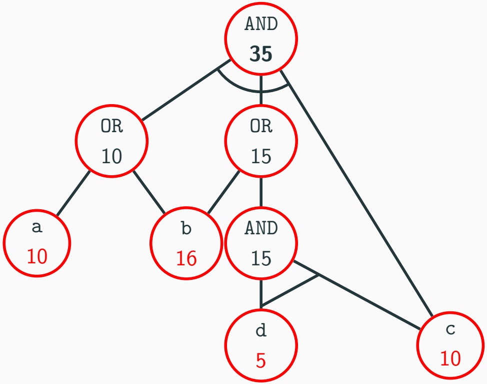
$\{b, c\}, \{a, c, d\}$

$\min\{16 + 10, 10 + 10 + 5\} = 25$

pros: intuitively desired result

cons: slow

Minimal cost via the bottom-up procedure

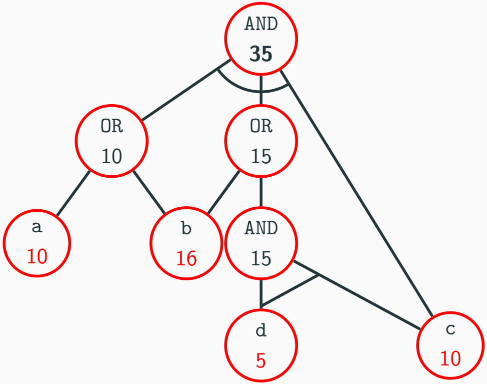


costs propagated up to the root

OR: min

AND: $+$

Minimal cost via the bottom-up procedure



costs propagated up to the root

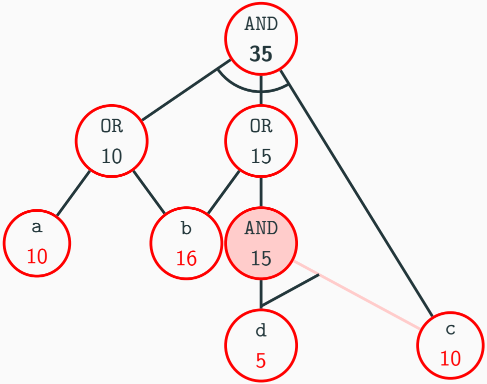
OR: min

AND: $+$

pros: fast

cons: incorrect result in the presence of clones

Minimal cost via the bottom-up procedure



costs propagated up to the root

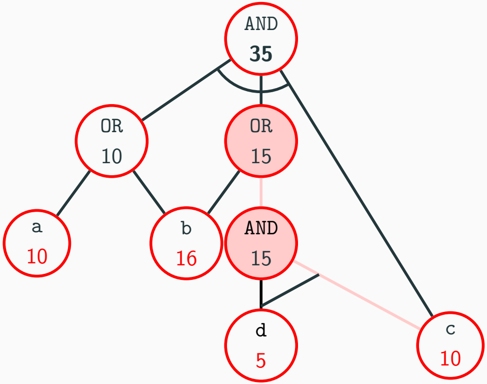
`OR`: min

`AND`: $+$

pros: fast

cons: incorrect result in the presence of clones

Minimal cost via the bottom-up procedure



costs propagated up to the root

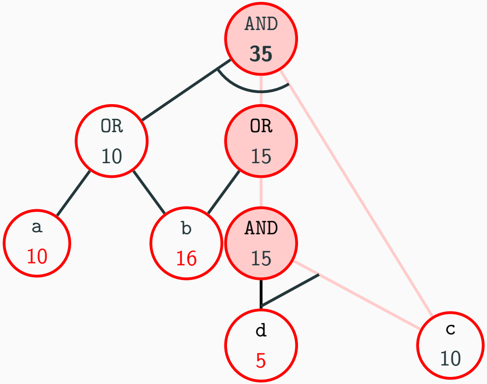
OR: min

AND: $+$

pros: fast

cons: incorrect result in the presence of clones

Minimal cost via the bottom-up procedure



costs propagated up to the root

OR: min

AND: $+$

pros: fast

cons: incorrect result in the presence of clones

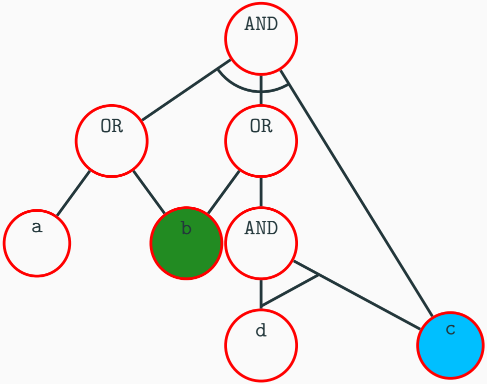
Combining the two methods

- Attacks extraction: **correct**, slow
- Bottom-up: **fast**, incorrect in the presence of clones

Research question 1

How to determine optimal attacks efficiently in the presence of clones?

Necessary and optional clones



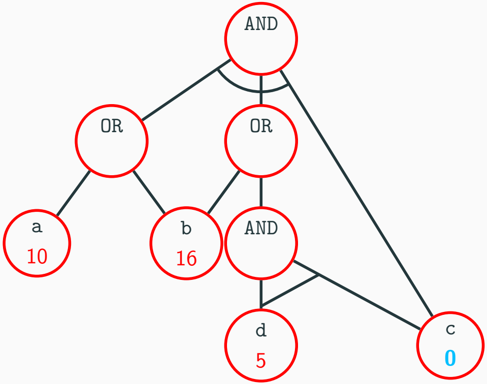
Attacks: $\{b, c\}, \{a, c, d\}$

c - **necessary clone**

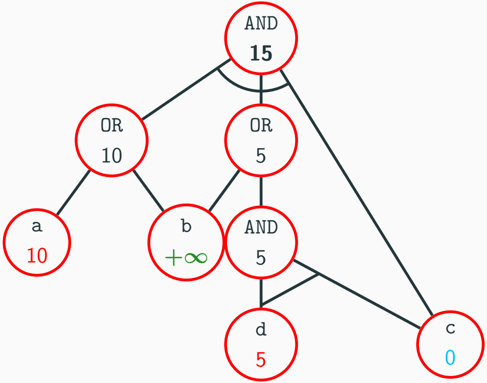
b - **optional clone**

Neutralize necessary clones

Step 1: $\texttt{cost}'(\texttt{c}) := \mathbf{0}$



Play with optional clones

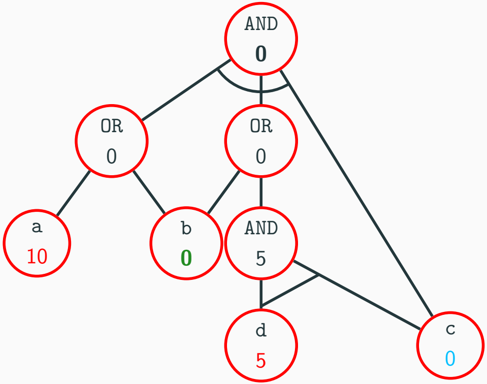


Step 1: $\mathtt{cost}'(\mathtt{c}) := 0$

Step 2.1: $\mathtt{cost}'(\mathtt{b}) := +\infty$

$\mathtt{res}_1 := \mathbf{15}$

Play with optional clones



Step 1: $\texttt{cost}'(\texttt{c}) := 0$

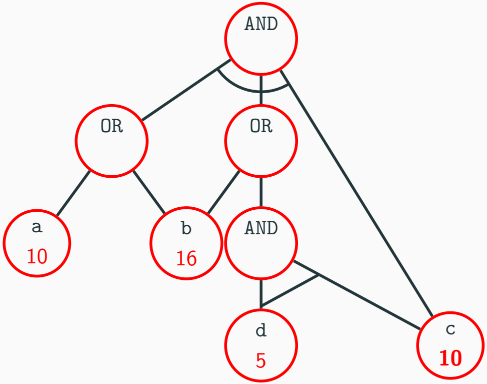
Step 2.1: $\texttt{cost}'(\texttt{b}) := +\infty$

$\texttt{res}_1 := 15$

Step 2.2: $\texttt{cost}'(\texttt{b}) := \mathbf{0}$

$\texttt{res}_2 := \mathbf{0} + \texttt{cost}(\texttt{b}) = 16$

Combine the results



Step 1: $\texttt{cost}'(\texttt{c}) := 0$

Step 2.1: $\texttt{cost}'(\texttt{b}) := +\infty$

$\texttt{res}_1 := 15$

Step 2.2: $\texttt{cost}'(\texttt{b}) := 0$

$\texttt{res}_2 := 16$

Step 3:

$\texttt{res} := \min\{15, 16\} + 10 = 25$

Algorithm for cost on attack trees

Input: Attack tree T, $(\overline{\mathbb{R}}^+, \min, +)$, $\mathtt{cost} : \mathbb{B} \to \overline{\mathbb{R}}^+$

Output: $\mathtt{Cost}(T, \mathtt{cost})$

$\mathcal{C}_N \leftarrow$ necessary clones

$\mathcal{C}_O \leftarrow$ optional clones

$\mathtt{cost}'(\mathtt{b}) \leftarrow 0$ for $\mathtt{b} \in \mathcal{C}_N$ //neutral for $+$

for every subset $\mathcal{C} \subseteq \mathcal{C}_O$ **do**

  $\mathtt{cost}'(\mathtt{b}) \leftarrow +\infty$ for every $\mathtt{b} \in \mathcal{C}$ //absorbing for $+$, neutral for min

  $\mathtt{cost}'(\mathtt{b}) \leftarrow 0$ for every $\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}$

  $r_{\mathcal{C}} \leftarrow \mathtt{cost}_{BU}(T, \mathtt{cost}') + \sum_{\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}} \mathtt{cost}(\mathtt{b})$

end for

return $\min\limits_{\mathcal{C} \subseteq \mathcal{C}_O} r_{\mathcal{C}} + \left(\sum_{\mathtt{b} \in \mathcal{C}_N} \mathtt{cost}(\mathtt{b})\right)$

Examples of attributes of interest

Minimal cost

$(\overline{\mathbb{R}}^{+}, \min, +)$

Examples of attributes of interest

Minimal cost

$(\overline{\mathbb{R}}^{+}, \mathsf{min}, +)$

Maximal success probability

$([0, 1], \mathsf{max}, \cdot)$

Examples of attributes of interest

Minimal cost

$(\overline{\mathbb{R}}^{+}, \min, +)$

Maximal success probability

$([0,1], \max, \cdot)$

Minimal skill level

$(\mathbb{N} \cup \{+\infty\}, \min, \max)$

Need for special equipment

$(\{0,1\}, \min, \max)$

Attributes modeled with semirings

Minimal cost

$(\mathbb{R}^+, \min, +)$

Maximal s

$([0,1], \max,$

Minimal sk

$(\mathbb{N} \cup \{+\infty\}$

Need for s

$(\{0,1\}, \min$

Commutative idempotent semiring

An algebraic structure $(D, \oplus, \odot)$, where

- $\oplus$ is **idempotent**
- $\oplus$ and $\odot$ are **associative** and **commutative**
- $\odot$ **distributes** over $\oplus$
- with $\mathbf{1}$ and $\mathbf{0}$

When less is better

Non-increasing attribute domain

An attribute domain $(D, \oplus, \odot)$ s.t.

- $(D, \oplus, \odot)$ - commutative idempotent semiring
- $c \oplus (c \odot d) = c$ for $c, d \in D$

When less is better

Non-increasing attribute domain

An attribute domain $(D, \oplus, \odot)$ s.t.

- $(D, \oplus, \odot)$ - commutative idempotent semiring
- $c \oplus (c \odot d) = c$ for $c, d \in D$ $\approx$ **doing less is better**

When less is better

Non-increasing attribute domain

An attribute domain $(D, \oplus, \odot)$ s.t.

- $(D, \oplus, \odot)$ - commutative idempotent semiring
- $c \oplus (c \odot d) = c$ for $c, d \in D$ $\quad\approx$ **doing less is better**

Minimal cost, $(\overline{\mathbb{R}}^{+}, \min, +)$

$\min\{x, x + y\} = x$

Maximal success probability, $([0, 1], \max, \cdot)$

$\max\{x, x \cdot y\} = x$

Algorithm in the general case

Input: Attack tree T, non-increasing attribute domain $(D, \oplus, \odot)$, $\alpha\colon \mathbb{B} \to D$

Output: $A(T, \alpha)$

```
C_N ← necessary clones
C_O ← optional clones
α'(b) ← 1 for b ∈ C_N                                  //neutral for ⊙
for every subset C ⊆ C_O do
    α'(b) ← 0 for every b ∈ C                          //absorbing for ⊙, neutral for ⊕
    α'(b) ← 1 for every b ∈ C_O \ C
    r_C ← α_B(T, α') ⊙ ⨀_{b∈C_O\C} α(b)
end for
return ⨁_{C⊆C_O} r_C ⊙ (⨀_{b∈C_N} α(b))
```

$$\mathcal{C}_N \leftarrow \text{necessary clones}$$
$$\mathcal{C}_O \leftarrow \text{optional clones}$$
$$\alpha'(\mathtt{b}) \leftarrow \mathbf{1} \text{ for } \mathtt{b} \in \mathcal{C}_N \qquad \text{//neutral for } \odot$$
for every subset $\mathcal{C} \subseteq \mathcal{C}_O$ **do**
$$\alpha'(\mathtt{b}) \leftarrow \mathbf{0} \text{ for every } \mathtt{b} \in \mathcal{C} \qquad \text{//absorbing for } \odot\text{, neutral for } \oplus$$
$$\alpha'(\mathtt{b}) \leftarrow \mathbf{1} \text{ for every } \mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}$$
$$r_{\mathcal{C}} \leftarrow \alpha_B(T, \alpha') \odot \bigodot_{\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}} \alpha(\mathtt{b})$$
end for

return $\displaystyle\bigoplus_{\mathcal{C} \subseteq \mathcal{C}_O} r_{\mathcal{C}} \odot \Big(\bigodot_{\mathtt{b} \in \mathcal{C}_N} \alpha(\mathtt{b})\Big)$

Algorithm in the general case

Input: Attack tree T, non-increasing attribute domain $(D, \oplus, \odot)$, $\alpha : \mathbb{B} \to D$

Output: $A(T, \alpha)$

$\mathcal{C}_N \leftarrow$ necessary clones
$\mathcal{C}_O \leftarrow$ optional clones
c … or $\odot$
f
Theorem

The algorithm returns correct results for non-increasing attribute domains.

  $\alpha'(\mathrm{b}) \leftarrow \mathbf{0}$ for every $\mathrm{b} \in \mathcal{C}$ //absorbing for $\odot$, neutral for $\oplus$
  $\alpha'(\mathrm{b}) \leftarrow \mathbf{1}$ for every $\mathrm{b} \in \mathcal{C}_O \setminus \mathcal{C}$
  $r_{\mathcal{C}} \leftarrow \alpha_B(T, \alpha') \odot \bigodot_{\mathrm{b} \in \mathcal{C}_O \setminus \mathcal{C}} \alpha(\mathrm{b})$
end for
return $\bigoplus_{\mathcal{C} \subseteq \mathcal{C}_O} r_{\mathcal{C}} \odot \left(\bigodot_{\mathrm{b} \in \mathcal{C}_N} \alpha(\mathrm{b})\right)$

Is the problem solved actually difficult?

Weighted Monotone Satisfiability Problem [Buldas 2012]

Given

- ϕ – monotone propositional formula (only $\vee$ and $\wedge$) over X,
- $w\colon X \to \mathbb{R}_{\geq 0}$ – weight function,
- t – threshold value,

decide whether

$$\min\{w(x_1) + \ldots + w(x_k)\colon x_1 \wedge \ldots \wedge x_k \models \phi\} \leq t.$$

Is the problem solved actually difficult?

Weighted Monotone Satisfiability Problem [Buldas 2012]

Given

- ϕ – monotone propositional formula (only $\vee$ and $\wedge$) over X,
- $w\colon X \to \mathbb{R}_{\geq 0}$ – weight function,
- t – threshold value,

decide whether

$$\min\{w(x_1) + \ldots + w(x_k) \colon x_1 \wedge \ldots \wedge x_k \models \phi\} \leq t.$$

$\Rightarrow$ in the case of cost **better to use mathematical programming**

Outline of the rest of the talk

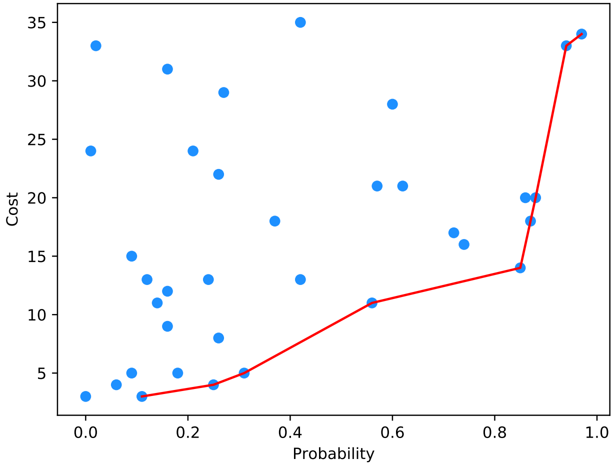
Analysis of attacks in the presence of clones

Multi-parameter analysis of attacks

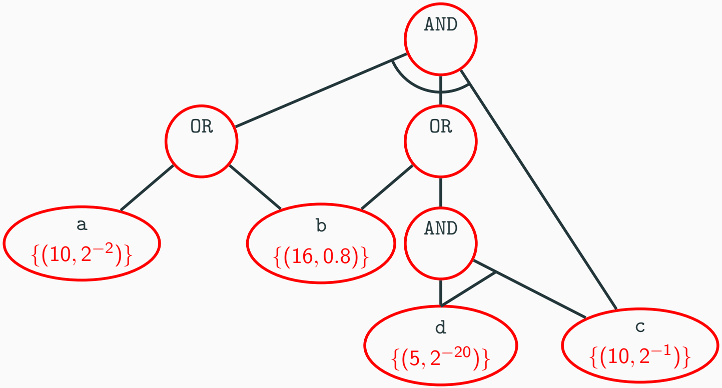
Other contributions

Future work

Pareto frontier (PF) for cost and probability



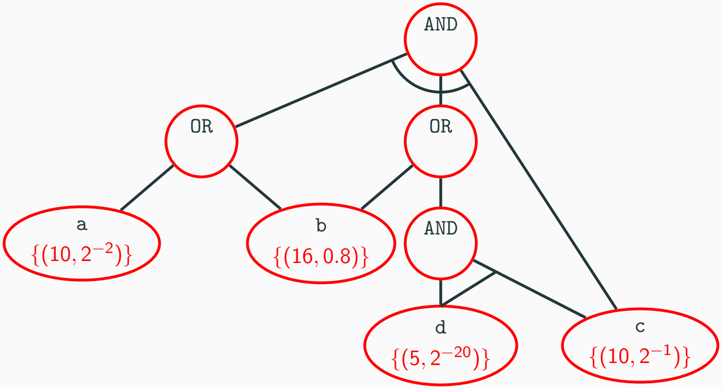
Pareto optimal attacks w.r.t. cost and probability



$\{\mathtt{b}, \mathtt{c}\} : \{(16 + 10, 0.8 \cdot 2^{-1})\}$

$\{\mathtt{a}, \mathtt{c}, \mathtt{d}\} : \{(10 + 10 + 5, 2^{-2} \cdot 2^{-1} \cdot 2^{-20})\}$

Pareto optimal attacks w.r.t. cost and probability



$\{\mathtt{b}, \mathtt{c}\}$: $\{(16 + 10, 0.8 \cdot 2^{-1})\}$

$\{\mathtt{a}, \mathtt{c}, \mathtt{d}\}$: $\{(10 + 10 + 5, 2^{-2} \cdot 2^{-1} \cdot 2^{-20})\}$

values: $\{(26, 0.4), (25, 2^{-23})\}$

PF: $\{(26, 0.4), (25, 2^{-23})\}$

Pareto attribute domain for cost and probability

- Domains for cost and probability: $(\overline{\mathbb{R}}^+, \min, +), ([0,1], \max, \cdot)$
- $\mathbf{d} = (d_c, d_p),\ \mathbf{d}' = (d'_c, d'_p) \in \mathbb{N} \times [0,1]$
- $D, D' \subseteq \mathbb{N} \times [0,1]$

$$\begin{aligned} \mathbf{d} \odot \mathbf{d}' &:= (d_c + d'_c, d_p \cdot d'_p) \\ D \odot D' &:= \{\mathbf{d} \odot \mathbf{d}' : \mathbf{d} \in D, \mathbf{d}' \in D'\} \\ D \widehat{\odot} D' &:= \mathrm{PF}(D \odot D') && \text{// Pareto frontier} \\ D \widehat{\oplus} D' &:= \mathrm{PF}(D \cup D') && \text{// Pareto frontier} \end{aligned}$$

- $\left(P(\overline{\mathbb{R}}^+ \times [0,1]), \widehat{\oplus}, \widehat{\odot}\right)$ // $P(X)$ = Pareto optimal subsets of X

Pareto attribute domain: general construction

- Attribute domains: $(D_1, \oplus_1, \odot_1), \ldots, (D_m, \oplus_m, \odot_m)$
- Operations for $\mathbf{d}, \mathbf{d}' \in D_1 \otimes \ldots \otimes D_m$ and $D, D' \subseteq D_1 \otimes \ldots \otimes D_m$:

$$\begin{aligned}
\mathbf{d} \odot \mathbf{d}' &:= (d_1 \oplus_1 d_1', \ldots, d_m \oplus_m d_m') \\
D \odot D' &:= \{\mathbf{d} \odot \mathbf{d}' \colon \mathbf{d} \in D, \mathbf{d}' \in D'\} \\
D \widehat{\odot} D' &:= \mathrm{PF}(D \odot D') \\
D \widehat{\oplus} D' &:= \mathrm{PF}(D \cup D')
\end{aligned}$$

Pareto attribute domain: general construction

- Attribute domains: $(D_1, \oplus_1, \odot_1), \ldots, (D_m, \oplus_m, \odot_m)$
- Operations for $\mathbf{d}, \mathbf{d}' \in D_1 \otimes \ldots \otimes D_m$ and $D, D' \subseteq D_1 \otimes \ldots \otimes D_m$:

$$\mathbf{d} \odot \mathbf{d}' := (d_1 \oplus_1 d_1', \ldots, d_m \oplus_m d_m')$$

$$D \odot D' := \{\mathbf{d} \odot \mathbf{d}' : \mathbf{d} \in D, \mathbf{d}' \in D'\}$$

$$D \widehat{\odot} D' := \mathrm{PF}(D \odot D')$$

$$D \widehat{\oplus} D' := \mathrm{PF}(D \cup D')$$

Pareto attribute domain

Let $(D_i, \oplus_i, \odot_i)$,for $i \in \{1, \ldots, m\}$, be **commutative idempotent semirings**.
The algebraic structure $(P(D_1 \otimes \ldots \otimes D_m), \widehat{\oplus}, \widehat{\odot})$ is the **Pareto attribute domain induced by** $(D_i, \oplus_i, \odot_i)$.

Properties of Pareto attribute domains

Theorem 1

Pareto attribute domains are commutative idempotent semirings.

Properties of Pareto attribute domains

Theorem 1

Pareto attribute domains are commutative idempotent semirings.

Theorem 2

Pareto attribute domain induced by non-increasing attribute domains is itself non-increasing.

$\Rightarrow$ the algorithm presented earlier can be applied

Pareto domains – takeaways

- **General framework** for multi-parameter analysis of security
  - suitable for attributes modeled with **semirings**
  - suitable for **any number** of such attributes
- Developed for **attack–defense trees**
- Applicable for **trees containing clones**

Outline of the rest of the talk

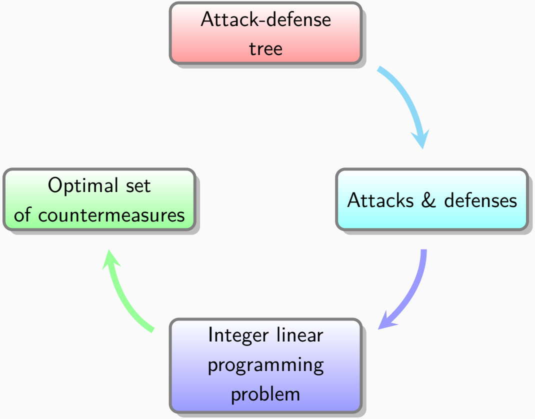
Analysis of attacks in the presence of clones

Multi-parameter analysis of attacks

Other contributions

Future work

Overview of our framework for selection of optimal countermeasures



The general integer linear programming problem

Optimization goal: maximize $F(x_1, \dots, x_p, f_1, \dots, f_m, z_1, \dots, z_n)$

Subject to:
$$\sum_{k=1}^{p} \mathtt{cost}(\mathtt{b}_k) x_k \leq \mathcal{B}$$

$$f_j \geq \frac{\sum_{k=1}^{p} A_{kj}(1 - x_k)}{p}, \ 1 \leq j \leq m$$

$$f_j \leq \sum_{k=1}^{p} A_{kj}(1 - x_k), \ 1 \leq j \leq m$$

$$z_i \geq 1 + \sum_{j=1}^{m} B_{ij}(f_j - 1), \ 1 \leq i \leq n$$

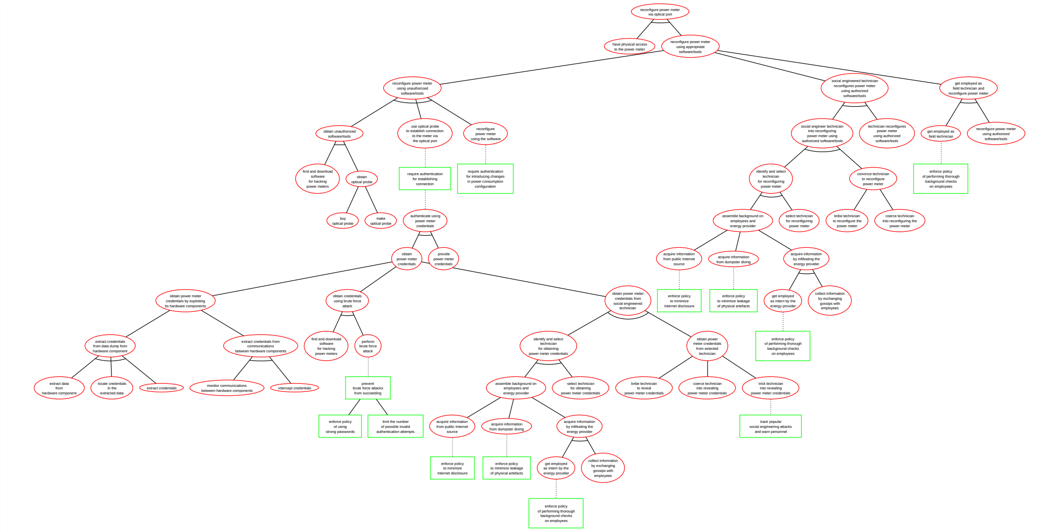
$$z_i \leq \frac{\sum_{j=1}^{m} B_{ij} f_j}{\sum_{j=1}^{m} B_{ij}}, \ 1 \leq i \leq n$$

$$x_k \in \{0, 1\}, f_j \in \{0, 1\}, z_i \in \{0, 1\}$$

Demonstration of usefuleness

- A **realistic case study** of electricity theft scenario and **tool demonstration**: B. Fila and W. Wideł. *Attack–defense trees for abusing optical power meters: A case study and the `OSEAD` tool experience report*. GraMSec'19.

Practical validation



Overview of the field

- A **detailed description and comparison** of approx. 30 **selected** recent papers on attack–defense trees:
  W. Wideł, M. Audinot, B. Fila and S. Pinchinat. *Beyond 2014: Formal methods for attack tree-based security modeling*. ACM Computing Surveys, 2019.

Outline of the rest of the talk

Analysis of attacks in the presence of clones

Multi-parameter analysis of attacks

Other contributions

Future work

Future work

- Focus on **automatization of models creation**
- Take **additional dependencies** into account
- Work on attribute domains **other than the non-increasing** ones
- Improve **efficiency** of the methods for **countermeasures selection**

Publications

- B. Kordy and W. Wideł. *Exploiting attack–defense trees to find an optimal set of countermeasures*. Under submission.
- B. Fila and W. Wideł. *Attack–defense trees for abusing optical power meters: A case study and the `OSEAD` tool experience report*. In proc. of GraMSec 2019.
- B. Fila and W. Wideł. *Efficient Attack–Defense Tree Analysis using Pareto Attribute Domains*. In proc. of CSF 2019.
- W. Wideł, M. Audinot, B. Fila and S. Pinchinat. *Beyond 2014: Formal methods for attack tree-based security modeling*. ACM Computing Surveys, 2019.
- B. Kordy and W. Wideł. *On quantitative analysis of attack–defense trees with repeated labels*. In proc. of POST 2018.
- B. Kordy and W. Wideł. *How well can I secure my system?* In proc. of iFM 2017.

Our place in the research area

- Analysis with clones
  - clones are not liked: **[Aslanyan 2015]**, **[Muller 2016]**
  - clones tend to be overlooked
  - we solve WMSAT of **[Buldas 2012]**, **[Buldas 2017]**, determining the cause for its difficulty
- Pareto-based analysis
  - more general than **[Aslanyan 2015]**, works with clones
  - faster than **[Kumar 2015]**, takes probability into account
- Selection of countermeasures
  - more complex settings than in **[Muller 2016]**, **[Roy 2017]**, **[Sendi 2018]**
  - more efficient than **[Aslanyan 2016]**