Attack trees: meaning, analysis, and correctness

Barbara Fila (Kordy)
http://people.irisa.fr/Barbara.Kordy/

GdR Security, WG FM, January 2020

It's been already 20 years!

Schneier on Security

Blog | Newsletter | Books | Essays | News | Talks | **Academic** | About Me

Academic >

Attack Trees

B. Schneier

Dr. Dobb's Journal, December 1999.

Modeling security threats

By Bruce Schneier

Few people truly understand computer security, as illustrated by computer-security company marketing literature that touts "hacker proof software," "triple-DES security," and the like. In truth, unbreakable security is broken all the time, often in ways its designers never imagined. Seemingly strong cryptography gets broken, too. Attacks thought to be beyond the ability of mortal men become commonplace. And as newspapers report security bug after security bug, it becomes increasingly clear that the term "security" doesn't have meaning unless also you know things like "Secure from whom?" or "Secure for how long?"

Search

Powered by DuckDuckGo

Go

blog essays whole site

Subscribe

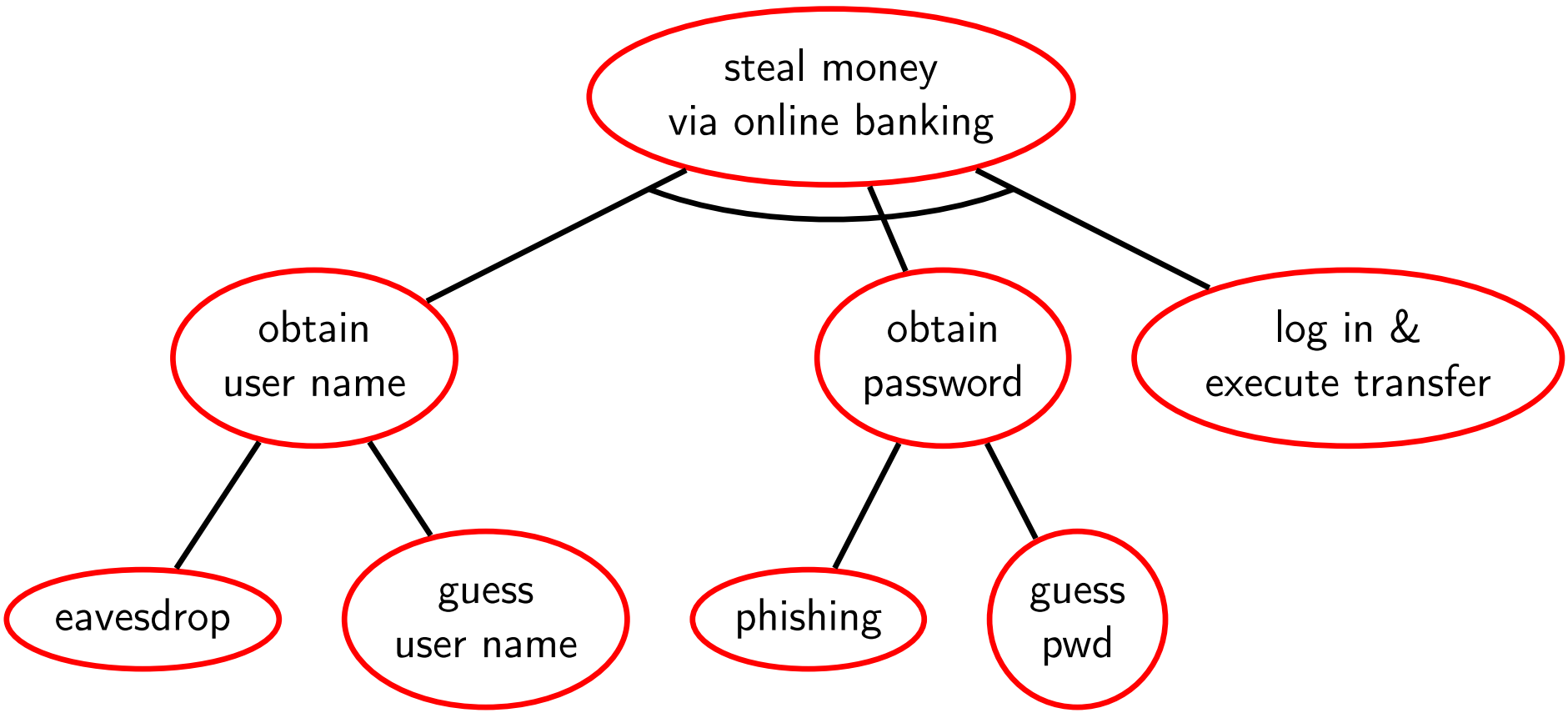
About Bruce Schneier

Outline

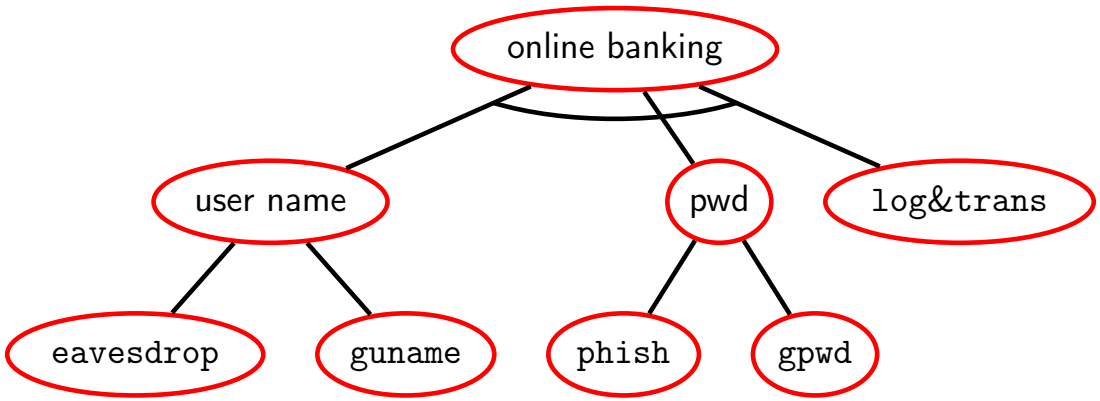
1. Attack trees
2. Repeated labels
3. State-based attack trees
4. There is much more going on

Outline

An attack tree



Attacks



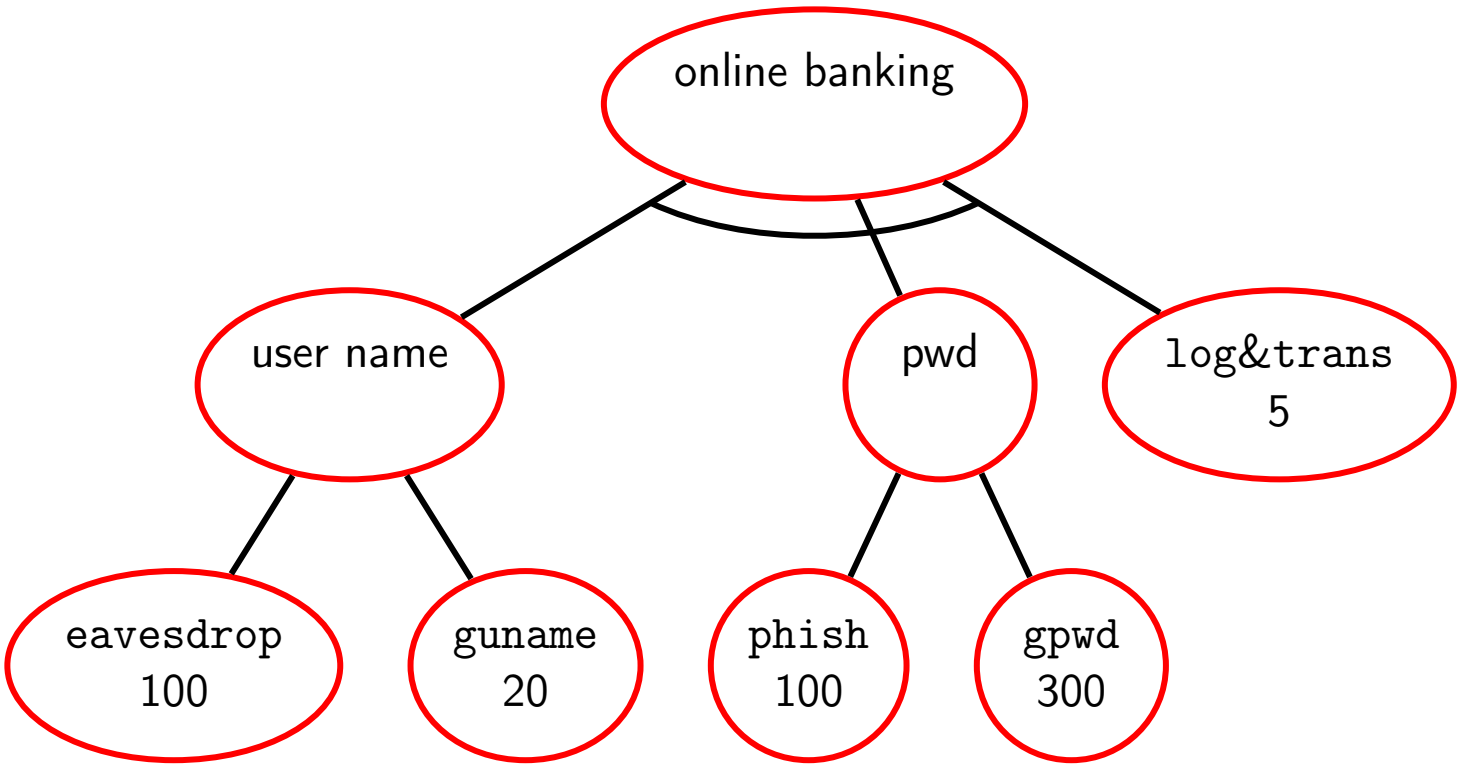
{eavesdrop, phish, log&trans}

{guname, phish, log&trans}

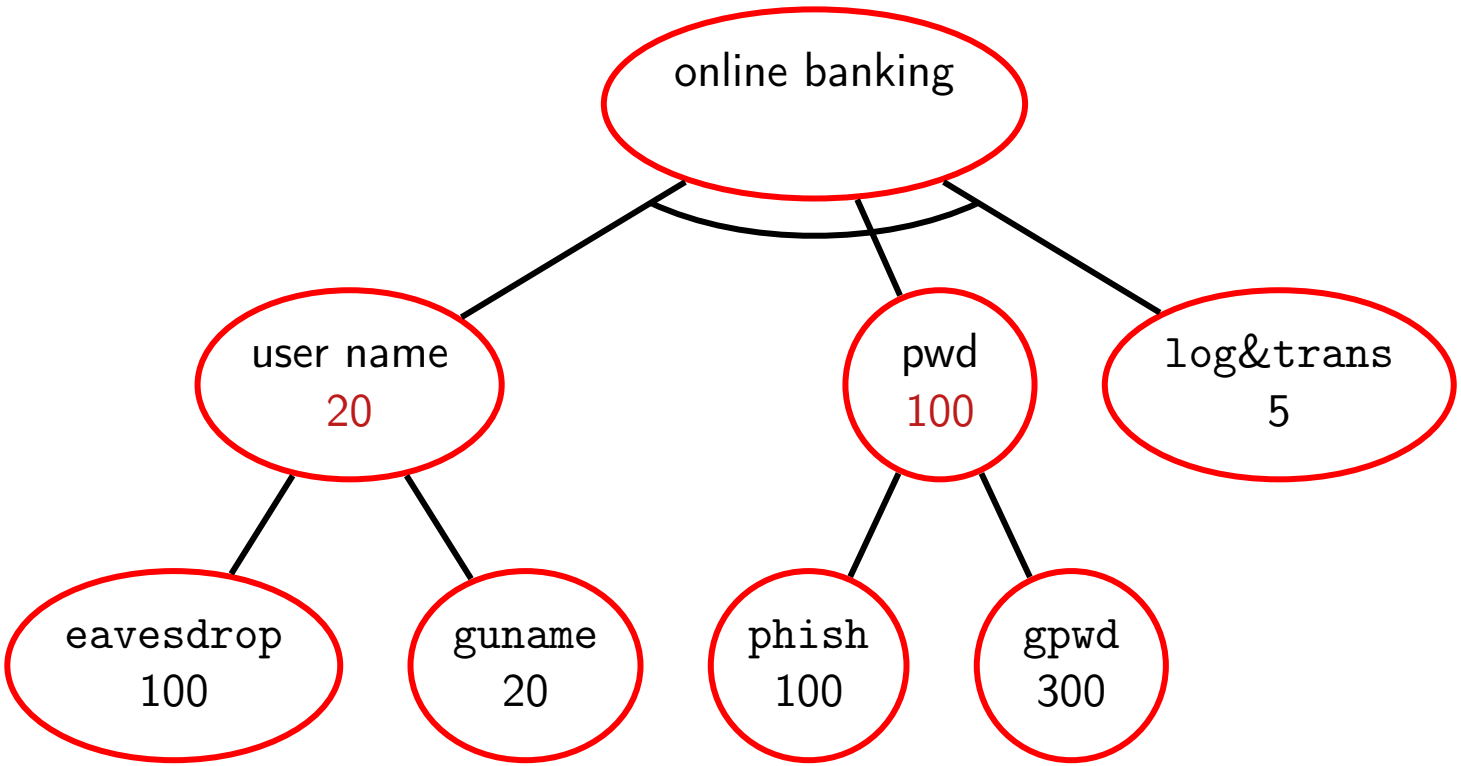
{eavesdrop, gpwd, log&trans}

{guname, gpwd, log&trans}

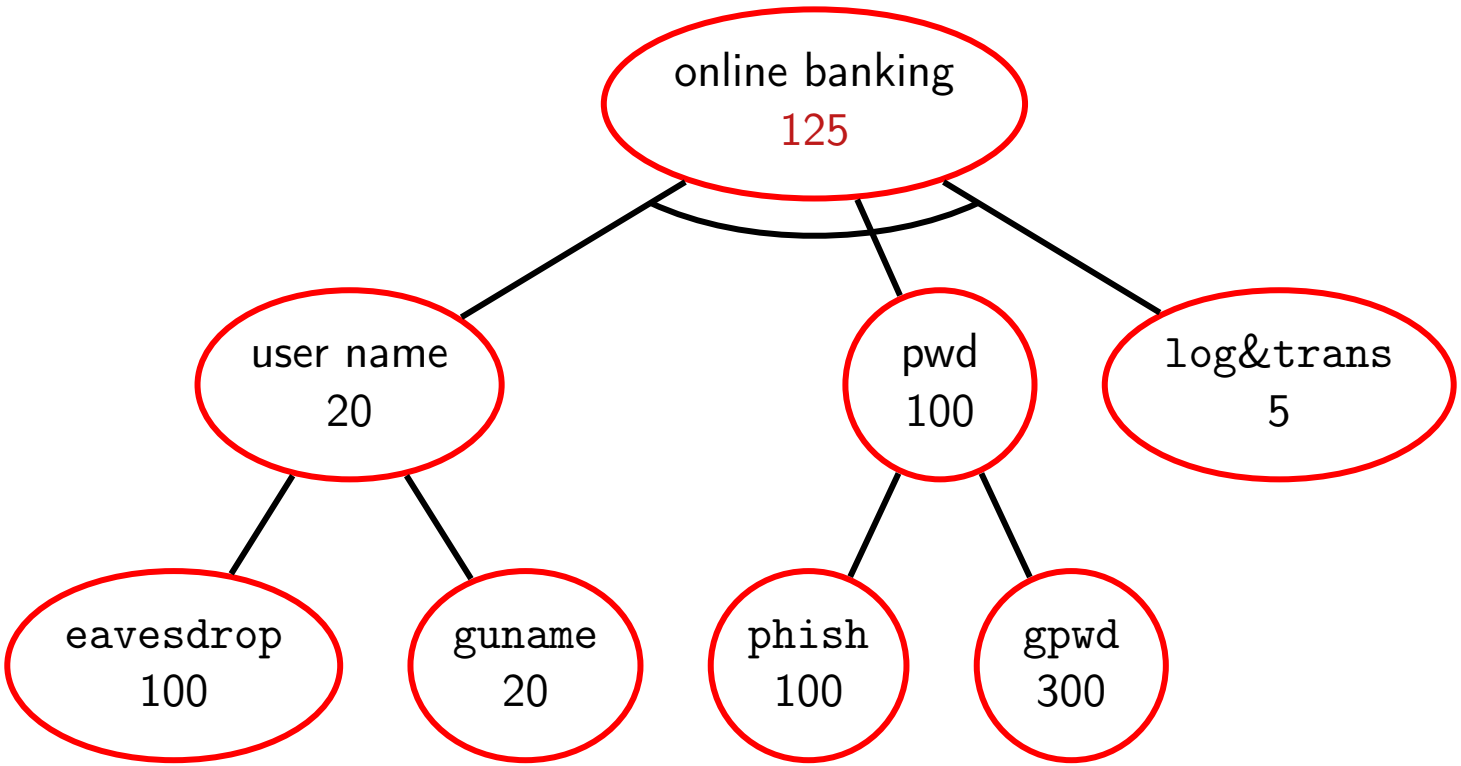
Bottom-up for `min time`



Bottom-up for `min time`



Bottom-up for `min time`



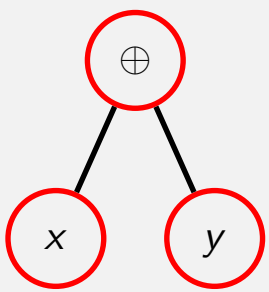
Bottom-up procedure formally

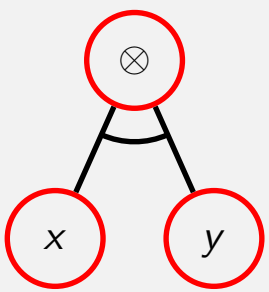
Basic assignment

Values assigned to the leaf nodes

Attribute domain

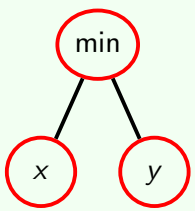
Algebraic structure defining the propagation rules $A = (D, \oplus, \otimes)$

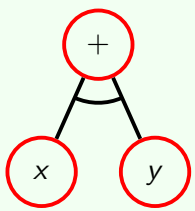




Example (Minimal time of attacking)

$$A_{\texttt{min\_time}} = (\mathbb{N}, \min, +)$$





Outline

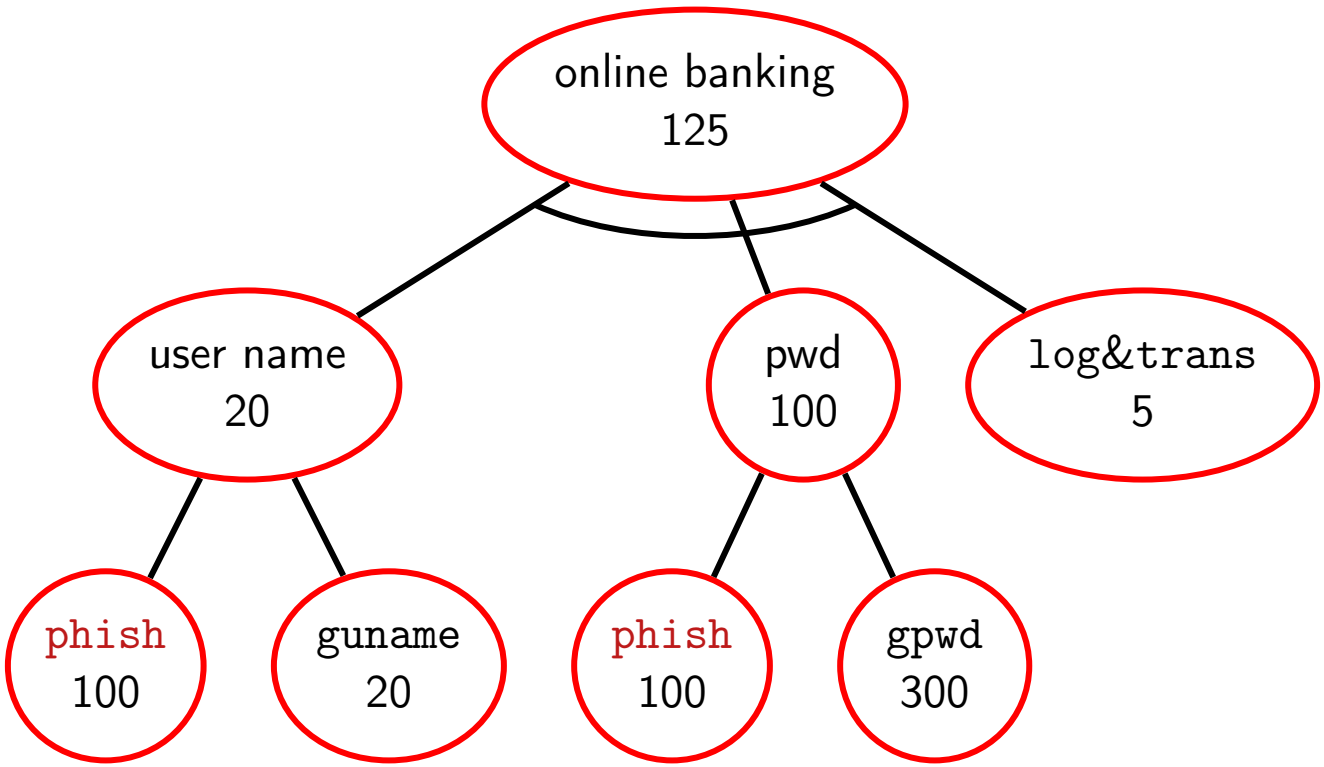
Reference

Wojciech Wideł and Barbara Kordy (Fila)

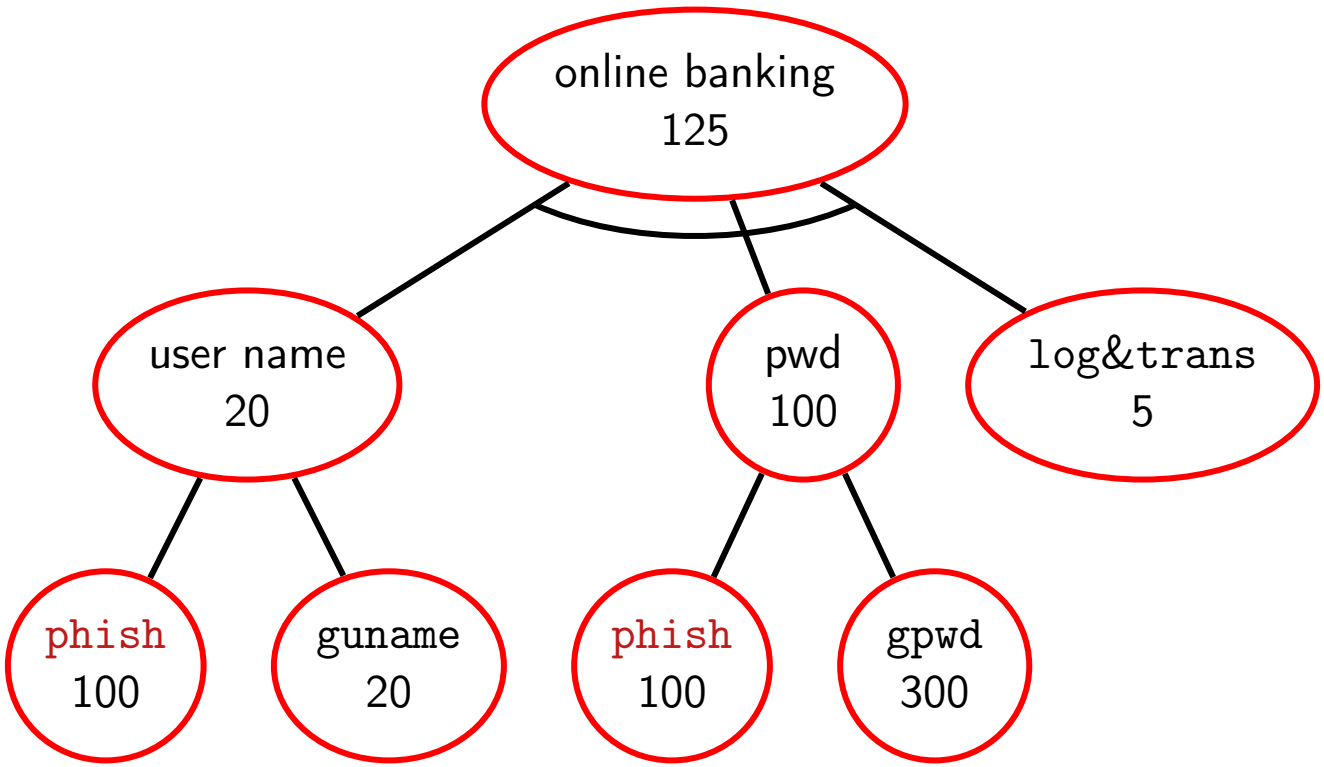
On quantitative analysis of attack-defense trees with repeated labels

POST 2018

A well-known problem



A well-known problem

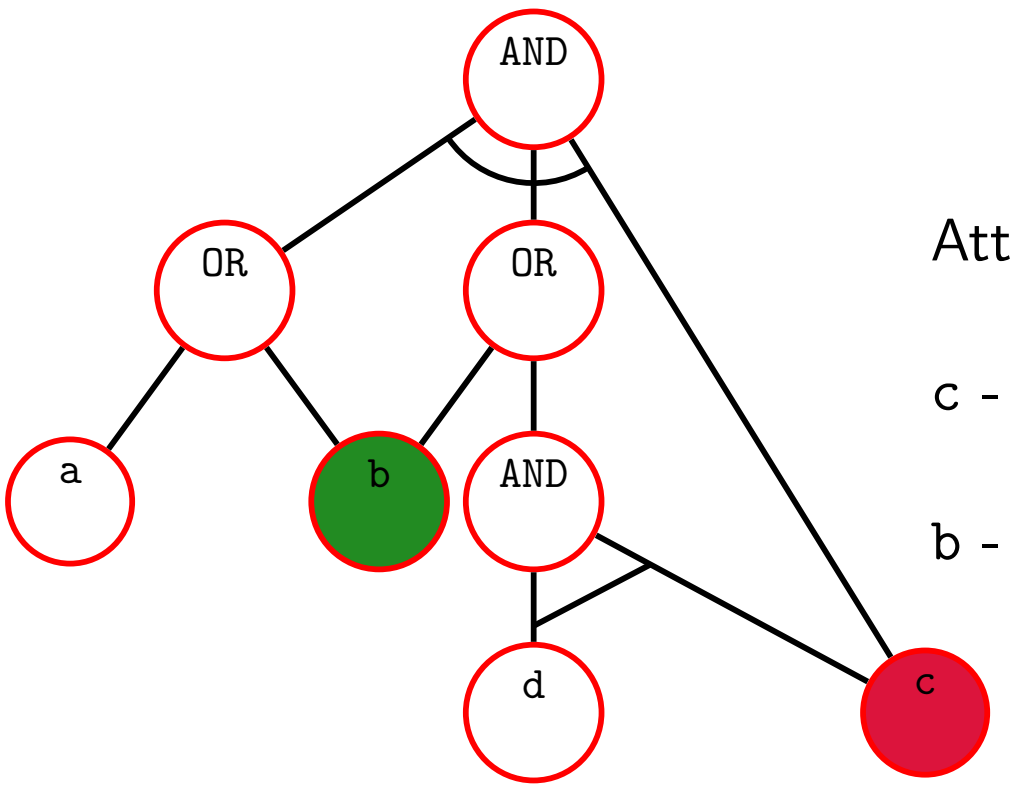


Overestimated!

Nodes with the same labels

Clones

Nodes representing the same instance of an action



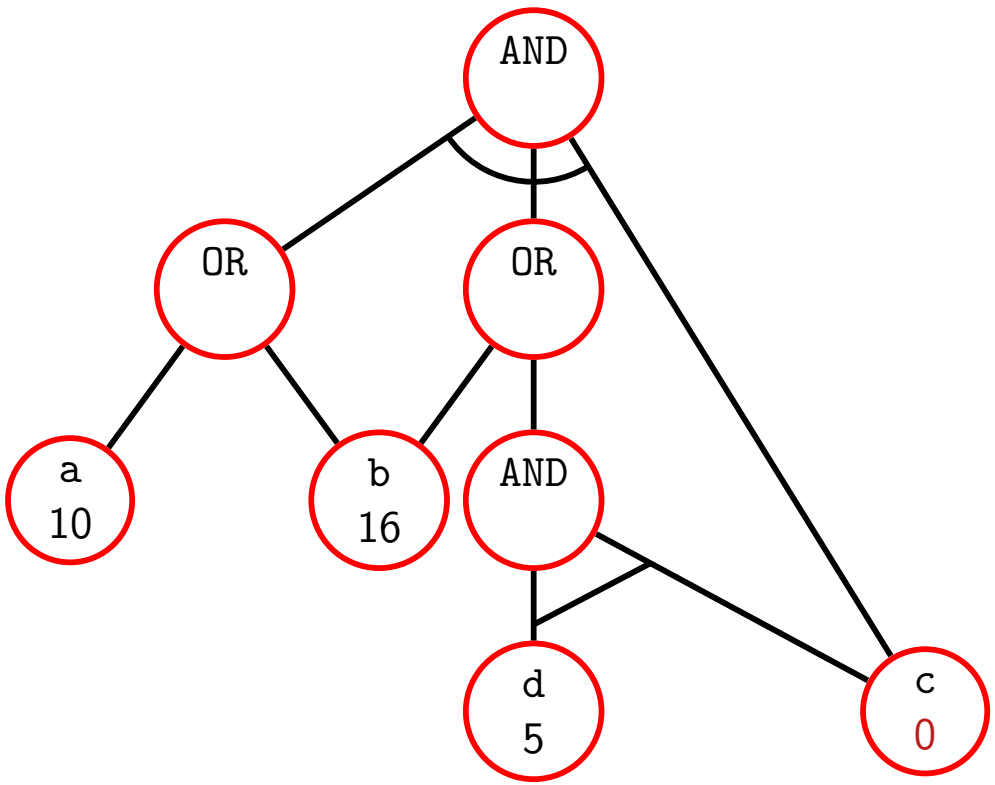
Attacks: $\{b, c\}, \{a, c, d\}$

c - necessary clone

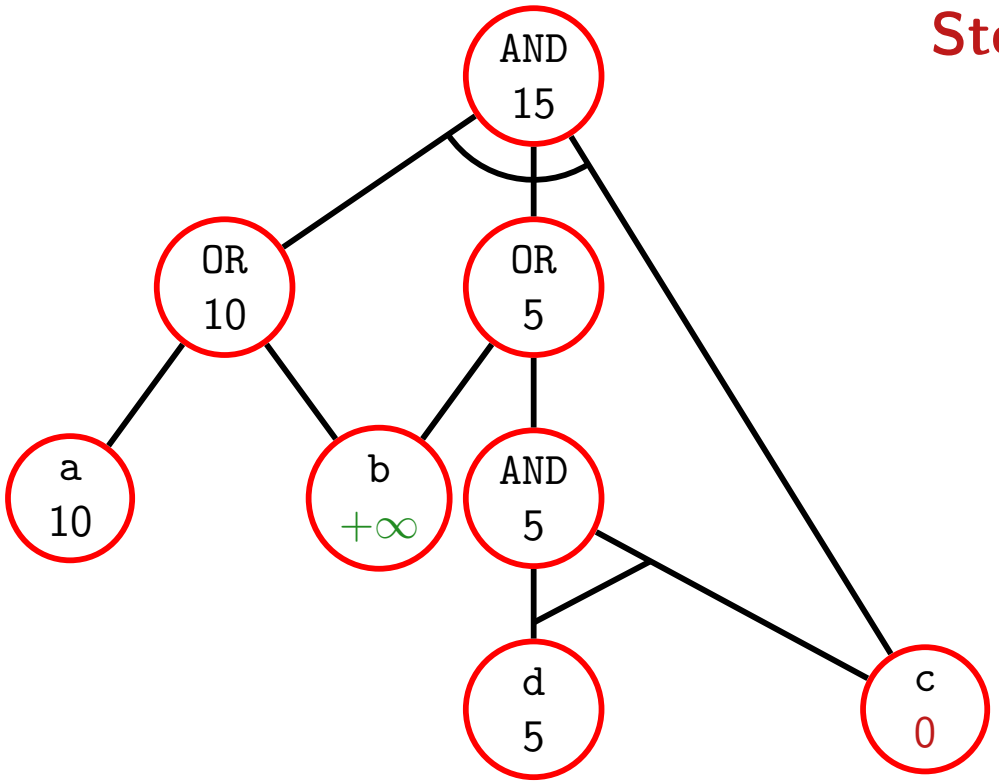
b - optional clone

Neutralize necessary clones

Step$_0$: $\texttt{time}'(\texttt{c}) := 0$



Play with the values of optional clones

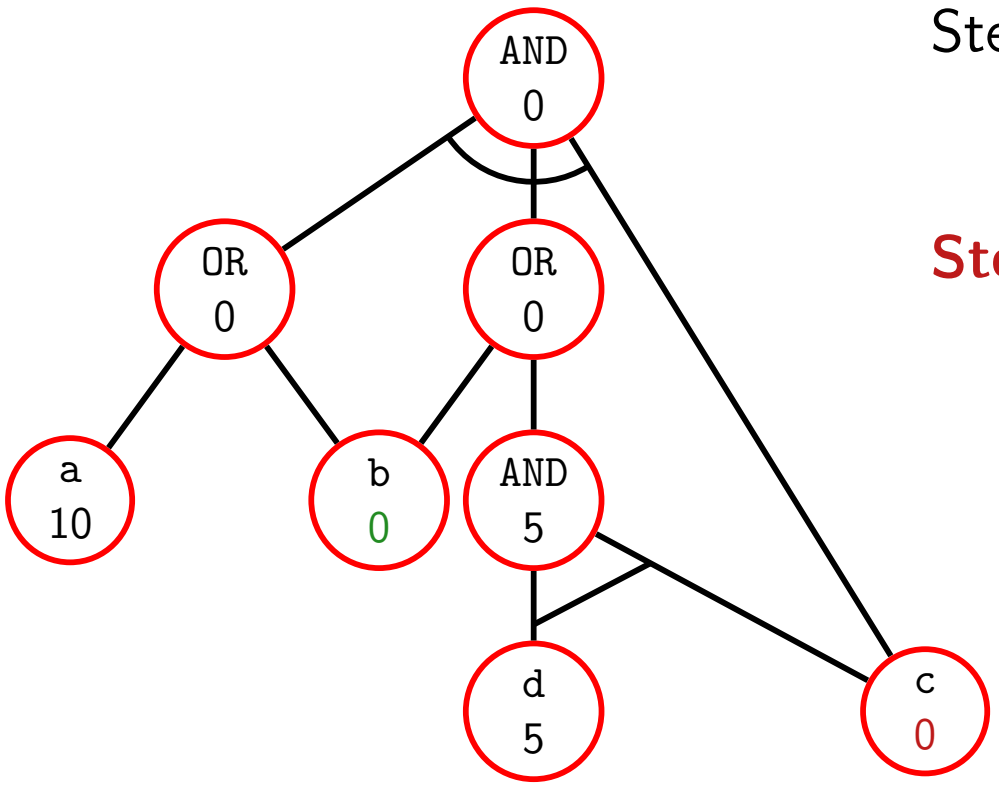


Step_0: $\texttt{time}'(\texttt{c}) := 0$

Step_1: $\texttt{time}'(\texttt{b}) := +\infty$

$\texttt{res}_1 = 15$

Play with the values of optional clones



Step_0: $\texttt{time}'(\texttt{c}) := 0$

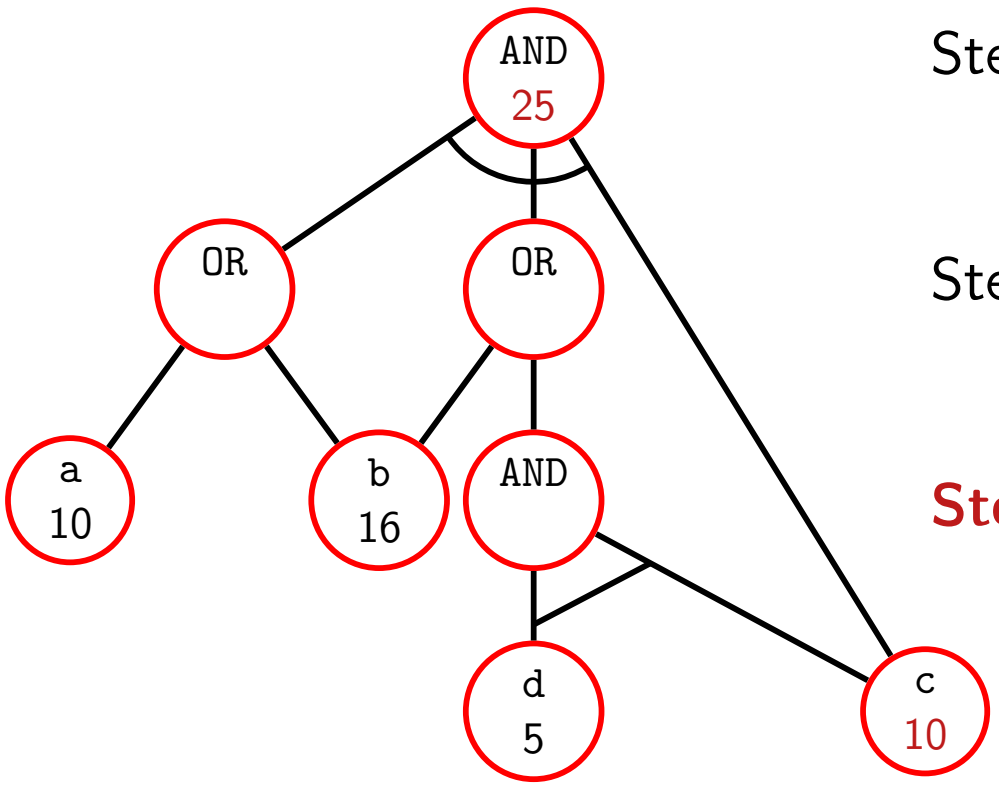
Step_1: $\texttt{time}'(\texttt{b}) := +\infty$

$\texttt{res}_1 = 15$

Step_2: $\texttt{time}'(\texttt{b}) := 0$

$\texttt{res}_2 = 0 + \texttt{time}(\texttt{b}) = 16$

Combine the results



Step_0: $\texttt{time}'(\texttt{c}) := 0$

Step_1: $\texttt{time}'(\texttt{b}) := +\infty$

$\texttt{res}_1 = 15$

Step_2: $\texttt{time}'(\texttt{b}) := 0$

$\texttt{res}_2 = 16$

Step_3: $\texttt{res} = \min\{15, 16\} + 10 = 25$

Algorithm for minimal time

Input: Attack tree t, $(\mathbb{R}^+, \min, +)$, basic assignment of $\mathtt{time}$
Output: Minimal time of attacking

1: $\mathcal{C}_N \leftarrow$ necessary clones
2: $\mathcal{C}_O \leftarrow$ optional clones
3: $\mathtt{time}'(\mathtt{b}) \leftarrow 0$ for $\mathtt{b} \in \mathcal{C}_N$
4: **for** every subset $\mathcal{C} \subseteq \mathcal{C}_O$ **do**
5: $\quad \mathtt{time}'(\mathtt{b}) \leftarrow +\infty$ for every $\mathtt{b} \in \mathcal{C}$
6: $\quad \mathtt{time}'(\mathtt{b}) \leftarrow 0$ for every $\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}$
7: $\quad r_{\mathcal{C}} \leftarrow \mathtt{time}_B(t, \mathtt{time}') + \sum_{\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}} \mathtt{time}(\mathtt{b})$
8: **end for**
9: **return** $\min_{\mathcal{C} \subseteq \mathcal{C}_O} r_{\mathcal{C}} + (\sum_{\mathtt{b} \in \mathcal{C}_N} \mathtt{time}(\mathtt{b}))$

Algorithm for $\texttt{minimal time}$

Input: Attack tree t, $(\mathbb{R}^+, \min, +)$, basic assignment of $\texttt{time}$
Output: Minimal time of attacking

1: $\mathcal{C}_N \leftarrow$ necessary clones
2: $\mathcal{C}_O \leftarrow$ optional clones
3: $\texttt{time}'(\texttt{b}) \leftarrow 0$ for $\texttt{b} \in \mathcal{C}_N$ $//0 = e_+$
4: **for** every subset $\mathcal{C} \subseteq \mathcal{C}_O$ **do**
5: $\quad \texttt{time}'(\texttt{b}) \leftarrow +\infty$ for every $\texttt{b} \in \mathcal{C}$ $//+\infty = a_+ = e_{\min}$
6: $\quad \texttt{time}'(\texttt{b}) \leftarrow 0$ for every $\texttt{b} \in \mathcal{C}_O \setminus \mathcal{C}$
7: $\quad r_{\mathcal{C}} \leftarrow \texttt{time}_B(t, \texttt{time}') + \sum_{\texttt{b} \in \mathcal{C}_O \setminus \mathcal{C}} \texttt{time}(\texttt{b})$
8: **end for**
9: **return** $\min_{\mathcal{C} \subseteq \mathcal{C}_O} r_{\mathcal{C}} + (\sum_{\texttt{b} \in \mathcal{C}_N} \texttt{time}(\texttt{b}))$

Algorithm in the general case

Input: Attack tree t, non-increasing attribute domain $(D, \oplus, \otimes)$, basic assignment for attribute α

Output: $A(t, \alpha)$

1: $\mathcal{C}_N \leftarrow$ necessary clones
2: $\mathcal{C}_O \leftarrow$ optional clones
3: $\alpha'(\mathtt{b}) \leftarrow e_\otimes$ for $\mathtt{b} \in \mathcal{C}_N$
4: **for** every subset $\mathcal{C} \subseteq \mathcal{C}_O$ **do**
5: $\quad \alpha'(\mathtt{b}) \leftarrow a_\otimes$ for every $\mathtt{b} \in \mathcal{C}$
6: $\quad \alpha'(\mathtt{b}) \leftarrow e_\otimes$ for every $\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}$
7: $\quad r_\mathcal{C} \leftarrow \alpha_B(t, \alpha') \otimes \bigotimes_{\mathtt{b} \in \mathcal{C}_O \setminus \mathcal{C}} \alpha(\mathtt{b})$
8: **end for**
9: **return** $\bigoplus_{\mathcal{C} \subseteq \mathcal{C}_O} r_\mathcal{C} \otimes (\bigotimes_{\mathtt{b} \in \mathcal{C}_N} \alpha(\mathtt{b}))$

Non-increasing attribute domain

Commutative idempotent semiring

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

- $\oplus$ is **idempotent**
- $\oplus$ and $\otimes$ are **associative** and **commutative**
- $\otimes$ **distributes** over $\oplus$
- absorbing element wrt $\otimes$ is equal to the neutral element wrt $\oplus$
  $a_{\otimes} = e_{\oplus}$

Canonical partial order $\preceq$ in an idempotent semiring: $x \preceq y$ iff $x \oplus y = y$

Non-increasing attribute domain

An attribute domain $(D, \oplus, \otimes)$ where

- $(D, \oplus, \otimes)$ is a commutative idempotent semiring
- $x \otimes y \preceq y$ (doing less is better)

Interesting attribute domains

Example (`minimal time`)

$(\mathbb{N}, \min, +)$

Example (`maximal probability`)

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

Example (`satisfiability`)

$(\{\mathsf{T}, \mathsf{F}\}, \vee, \wedge)$

Example (`required skills level`)

$(\mathbb{N}, \min, \max)$

Outline

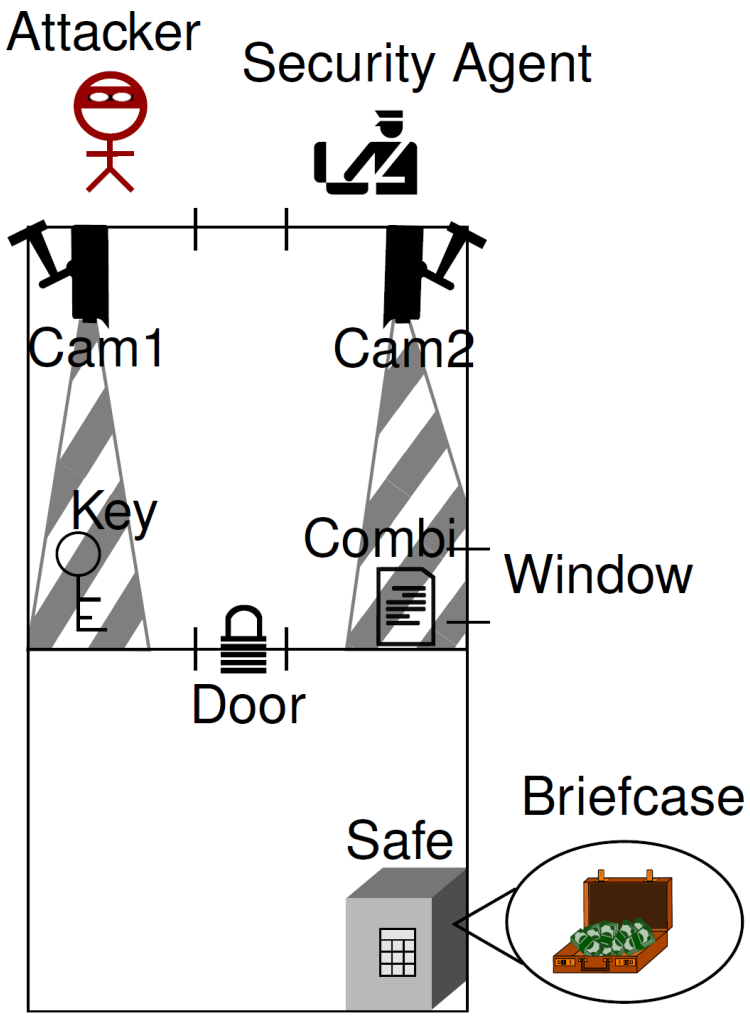
Reference

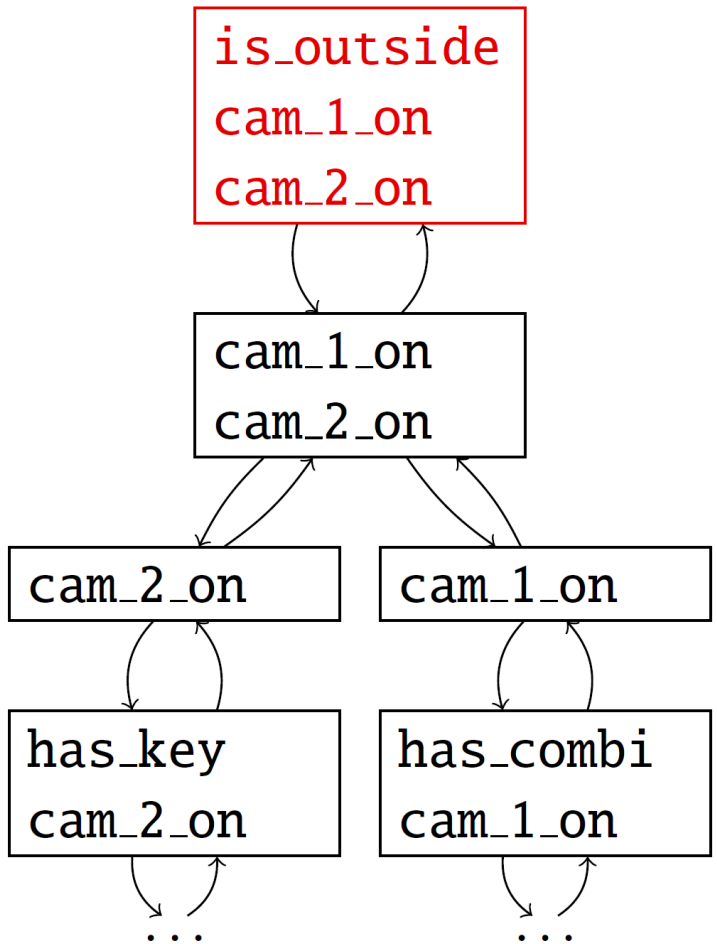
Maxime Audinot, Sophie Pinchinat, and Barbara Kordy (Fila)

Is My Attack Tree Correct?

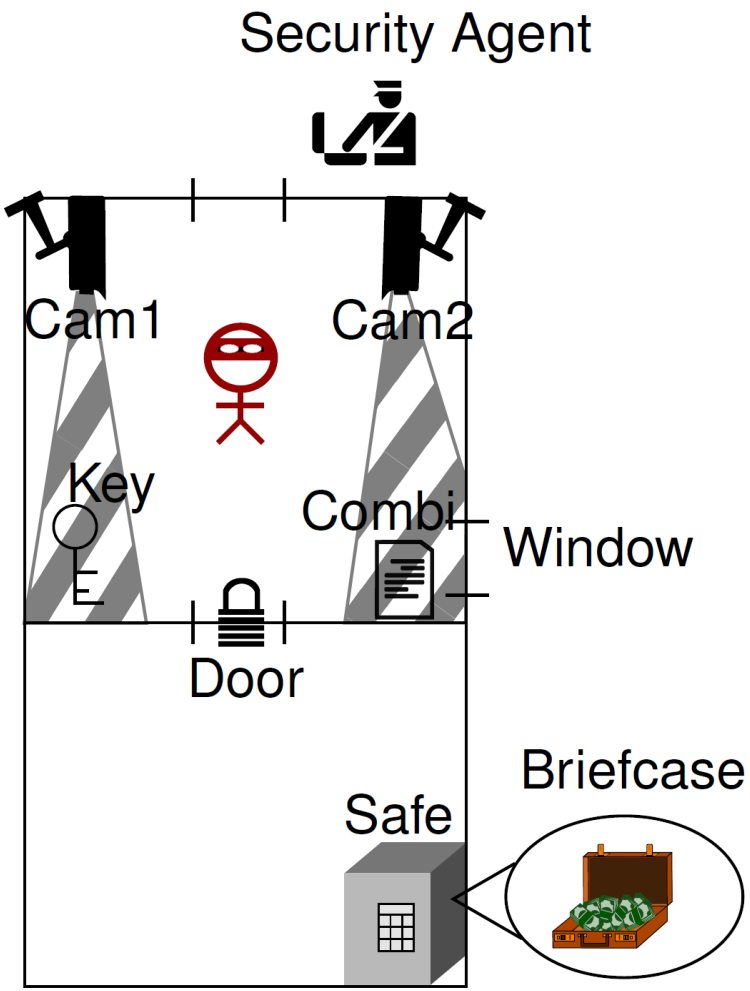
ESORICS 2017

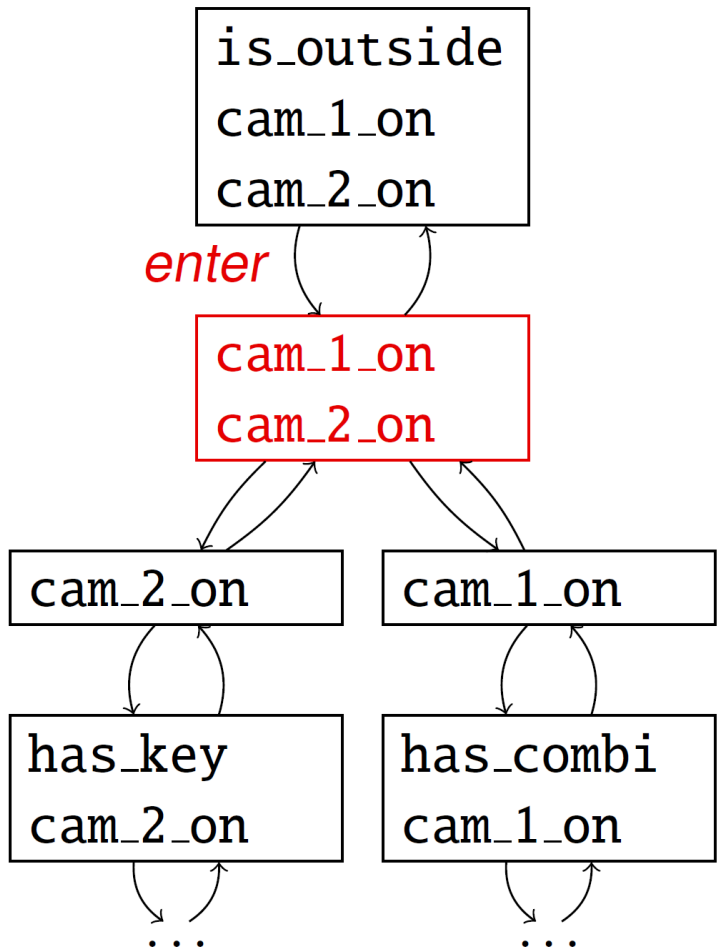
Modeling the system



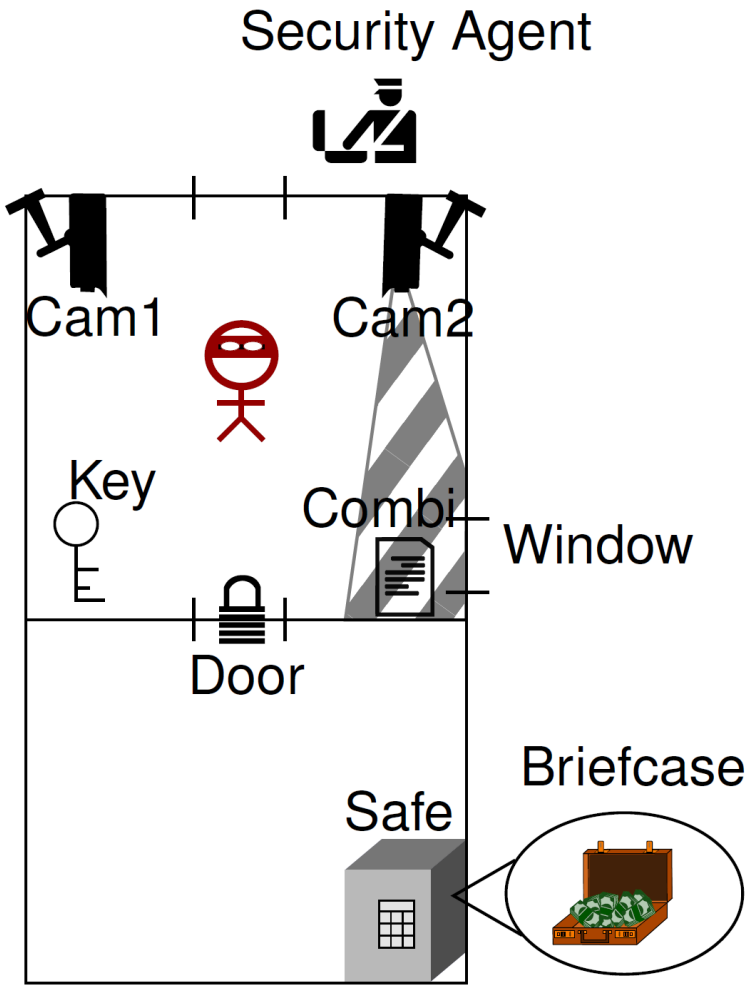


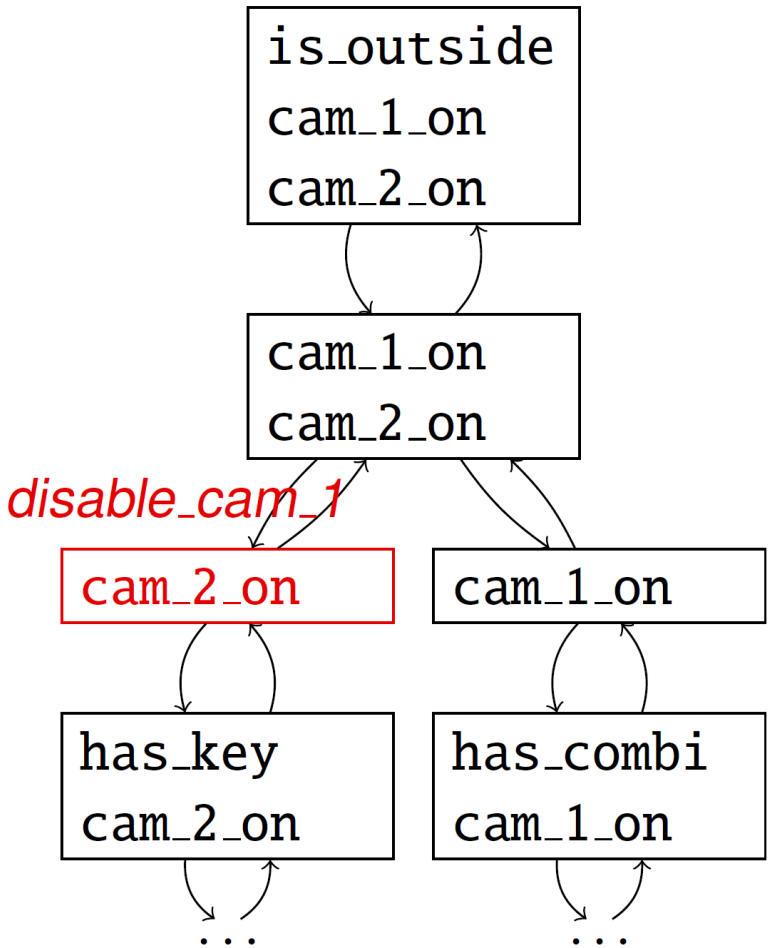
Modeling the system



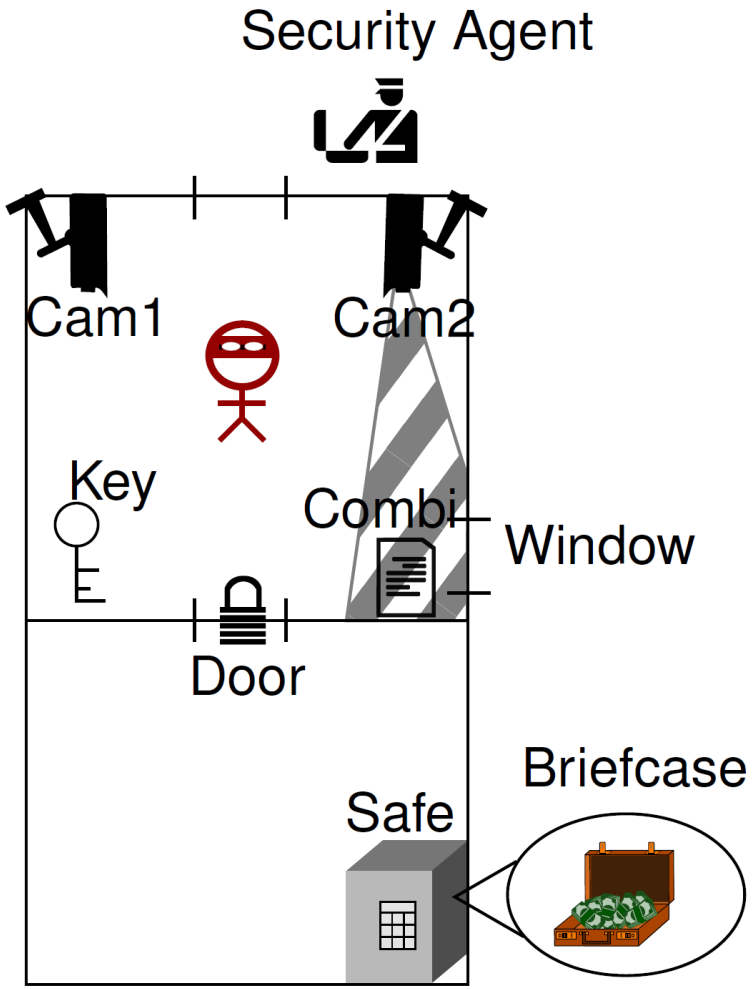


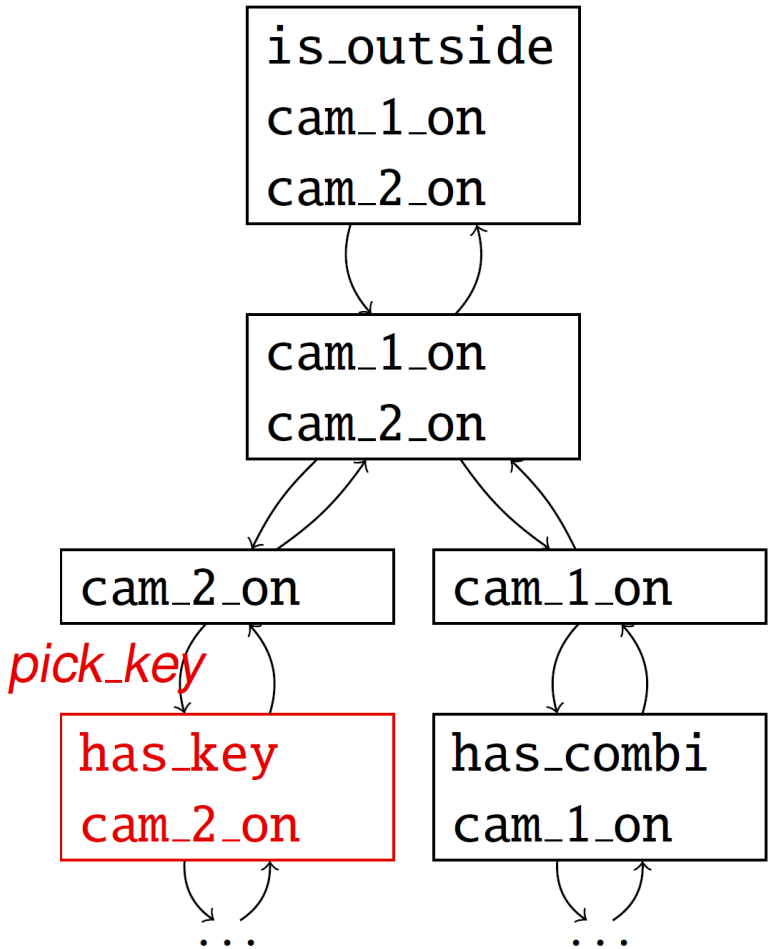
Modeling the system





Modeling the system





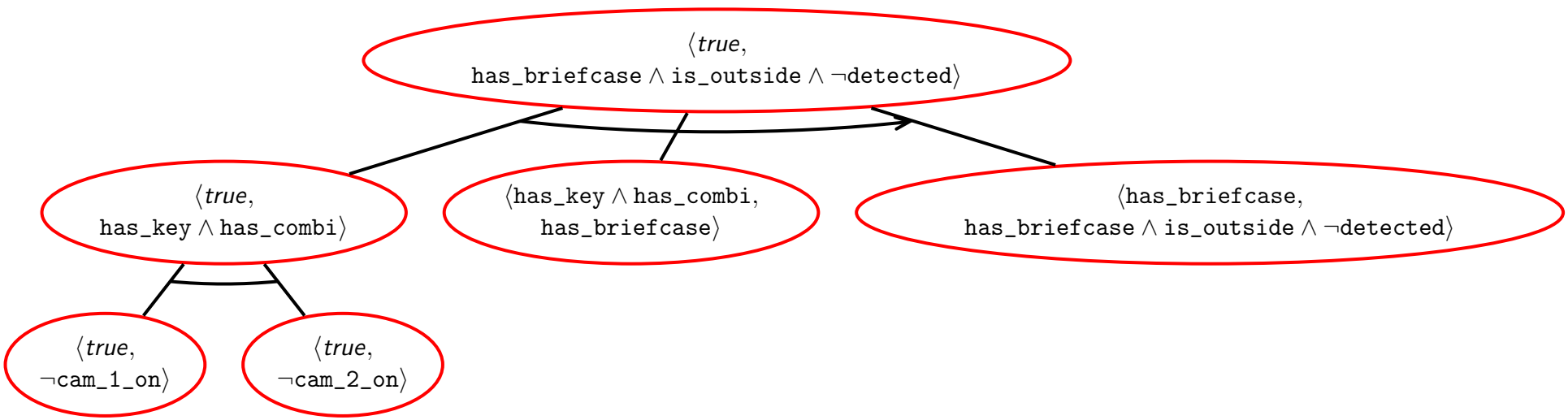
State-based attack tree

Goal $\langle \iota, \gamma \rangle$

- ι – precondition
- γ – postcondition

State-based attack tree grammar

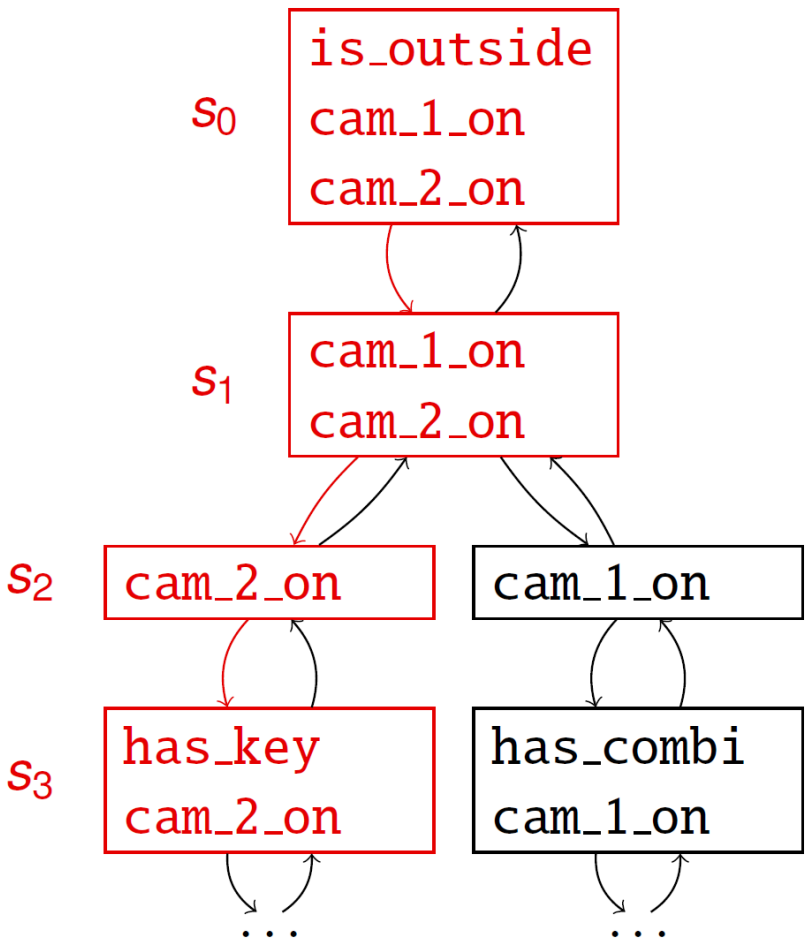
$\tau ::= \langle \iota, \gamma \rangle \mid \mathtt{OP}(\tau_1, \ldots, \tau_n)$
where $\mathtt{OP} \in \{\mathtt{OR}, \mathtt{AND}, \mathtt{SAND}\}$



Paths in transition systems

Paths in the system *Sys*

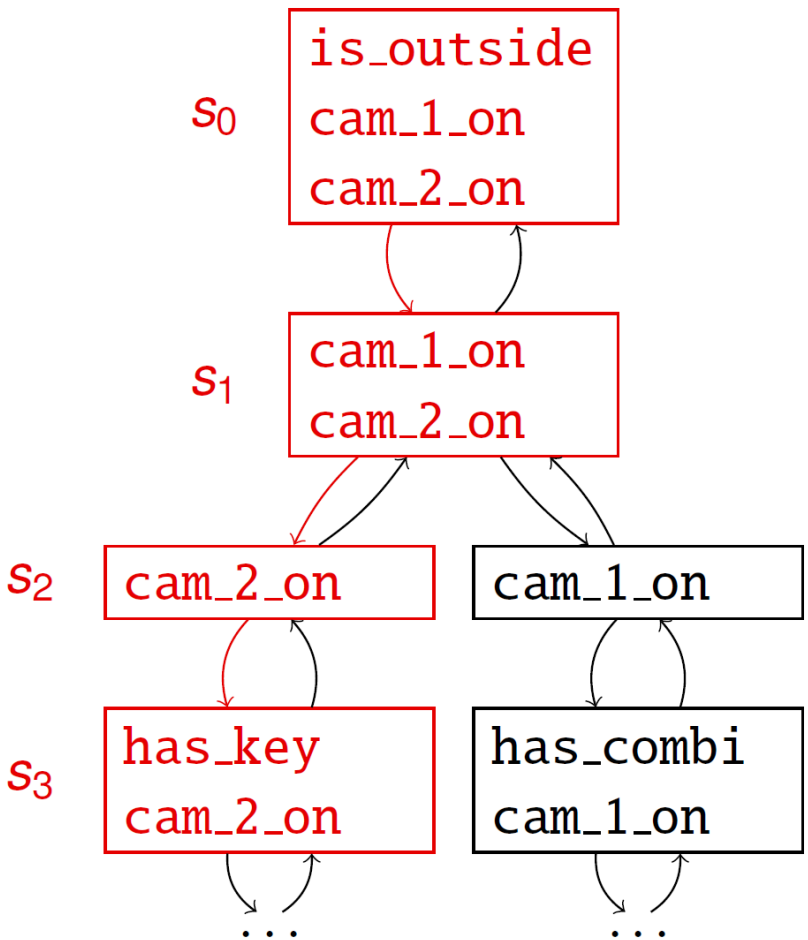
A path is a sequence of states $\pi = s_0 \dots s_n$ with $s_i \to s_{i+1}$ for all i.



Paths in transition systems

Paths in the system *Sys*

A path is a sequence of states $\pi = s_0 \dots s_n$ with $s_i \to s_{i+1}$ for all i.



Goal $\langle \iota, \gamma \rangle$ over Prop

- ι – precondition
- γ – postcondition

$\pi = s_0 s_1 s_2 s_3$

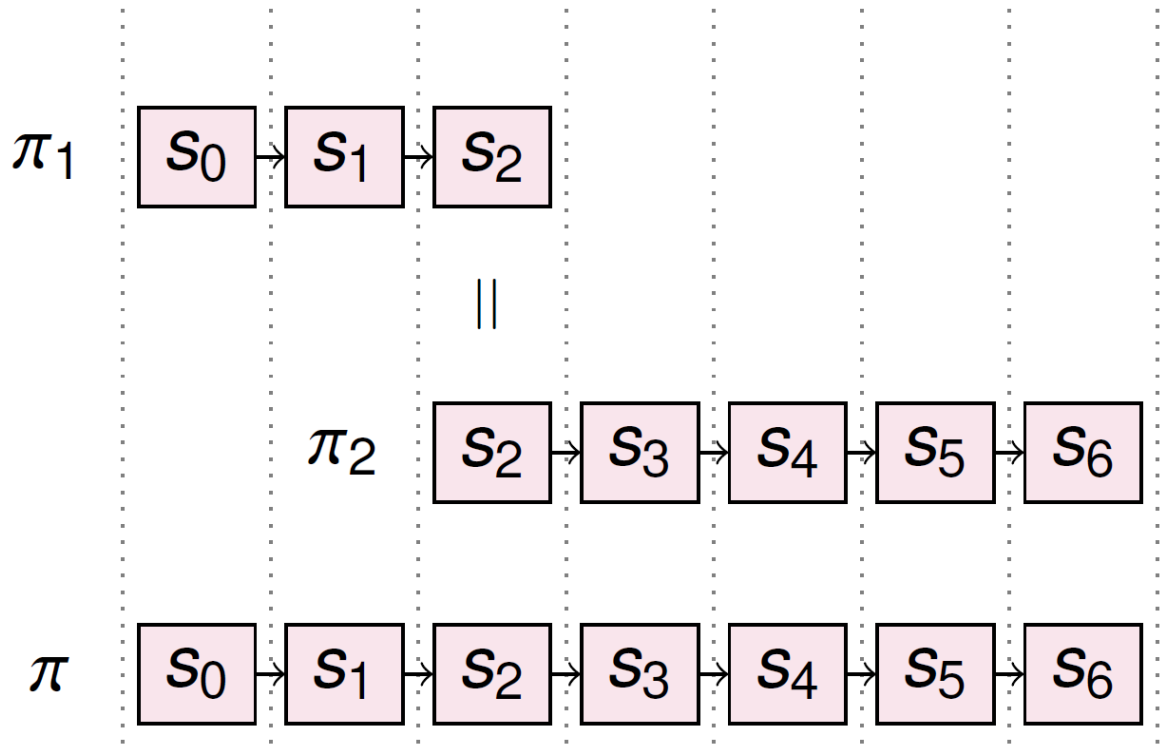
$s_0 \models \texttt{is\_outside}$

$s_3 \models \texttt{has\_key}$

π achieves $\langle \texttt{is\_outside}, \texttt{has\_key} \rangle$

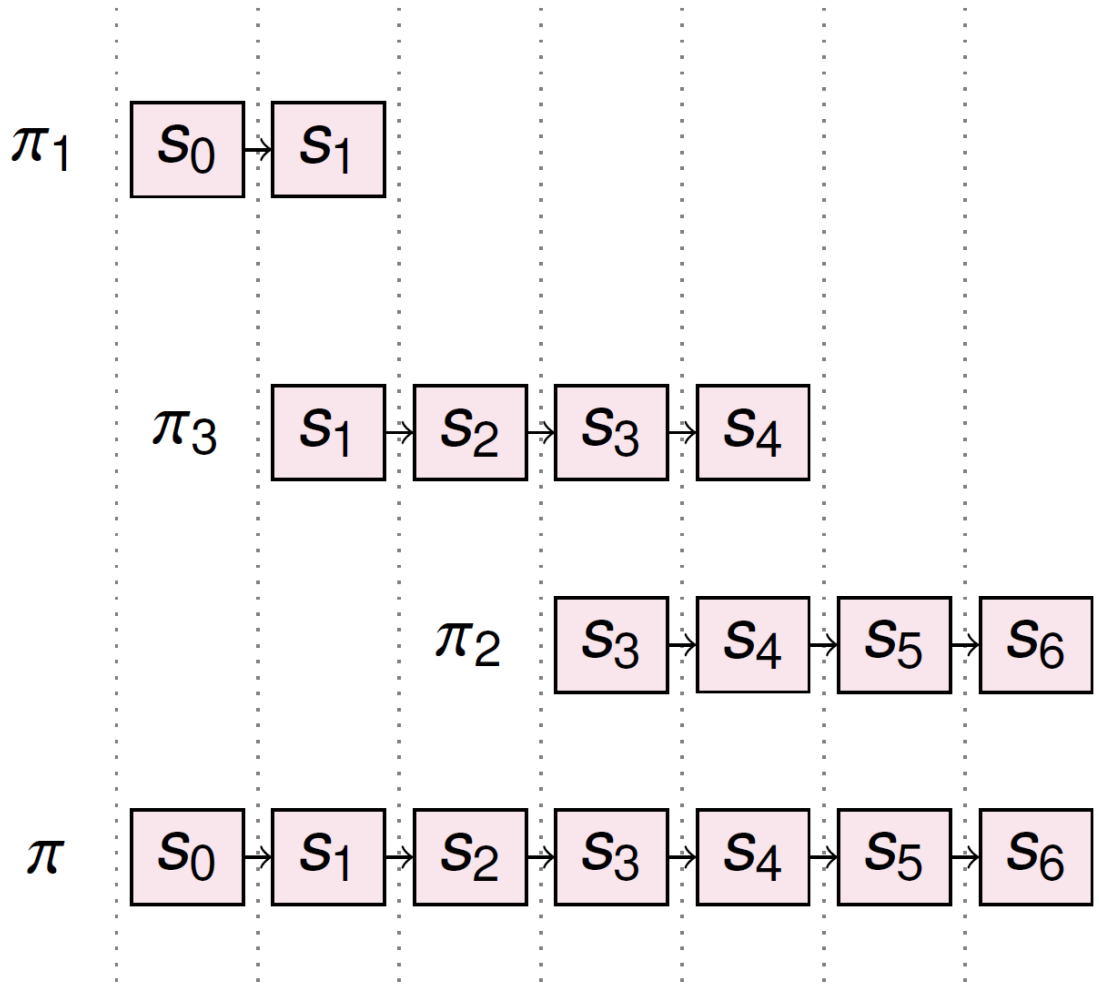
Sequential composition of paths •

$$\pi = \pi_1 \bullet \pi_2$$



Parallel composition of paths $\mathbb{M}$

$$\pi = \mathbb{M}(\pi_1, \pi_2, \pi_3)$$



Path semantics $[\tau]^{Sys}$

State-based attack trees formalized with sets of paths

Let *Sys* be a system.

The path semantics $[\cdot]^{Sys}$ is a set of paths in *Sys*, constructed as follows

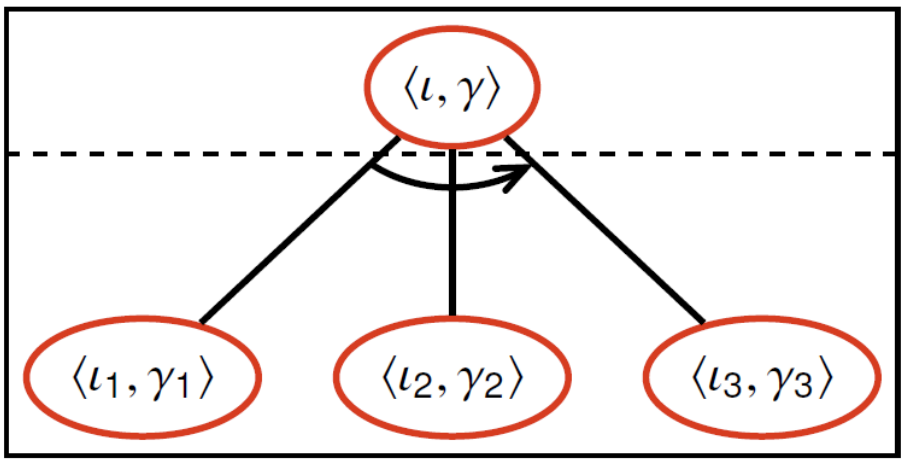
- $[\langle \iota, \gamma \rangle]^{Sys} = \{\pi \mid \pi \text{ achieves } \langle \iota, \gamma \rangle \text{ in } Sys\}$
- $[\mathtt{OR}(\tau_1, \ldots, \tau_n)]^{Sys} = [\tau_1]^{Sys} \cup \cdots \cup [\tau_n]^{Sys}$
- $[\mathtt{SAND}(\tau_1, \ldots, \tau_n)]^{Sys} = [\tau_1]^{Sys} \bullet \cdots \bullet [\tau_n]^{Sys}$
- $[\mathtt{AND}(\tau_1, \ldots, \tau_n)]^{Sys} = \mathbb{M}([\tau_1]^{Sys}, \ldots, [\tau_n]^{Sys})$

Analysis of state-based attack trees

How can we exploit the path semantics to analyze state-based attack trees?

Refinement quality problem

Labels of intermediate nodes represent the history of refinement



$$\langle \iota, \gamma \rangle$$

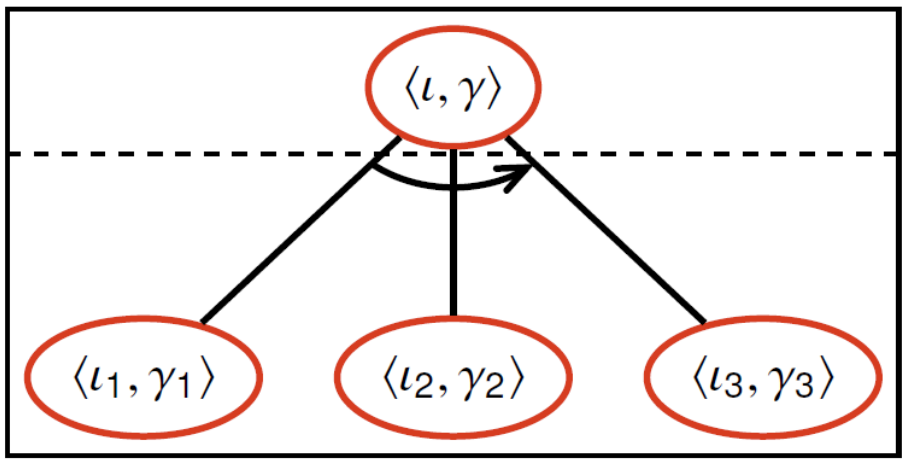
$$\downarrow \text{refinement}$$

$$\mathtt{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)$$

Objective

How well has a node been refined with respect to a given system?

Meet



$\exists \pi$ in *Sys*, such that

$$\pi \in [\langle \iota, \gamma \rangle]^{Sys}$$

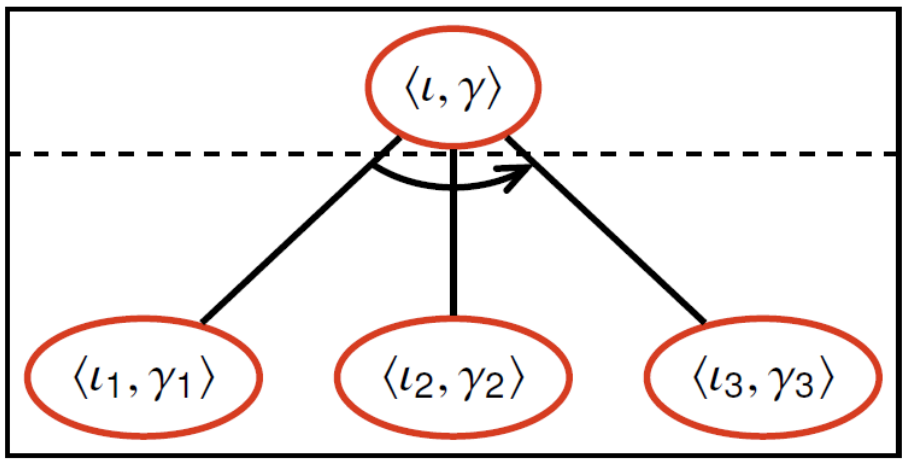
$$\pi \in [\mathtt{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^{Sys}$$

Meet property

There exists a common path achieving the node's goal and its refinement

$$[\langle \iota, \gamma \rangle]^{Sys} \cap [\mathtt{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^{Sys} \neq \emptyset$$

Match



$$[\langle \iota, \gamma \rangle]^{Sys}$$

$$\| \quad \text{Match}$$

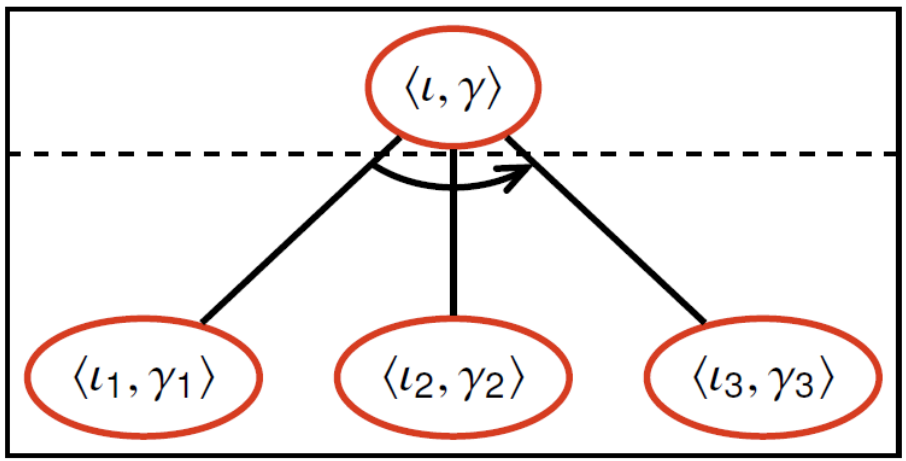
$$[\mathtt{OP}(\langle \iota 1, \gamma 1 \rangle, \langle \iota 2, \gamma 2 \rangle, \langle \iota 3, \gamma 3 \rangle)]^{Sys}$$

Match property

Ideal situation – the node and its refinement model the same set of paths

$$[\langle \iota, \gamma \rangle]^{Sys} = [\mathtt{OP}(\langle \iota 1, \gamma 1 \rangle, \langle \iota 2, \gamma 2 \rangle, \langle \iota 3, \gamma 3 \rangle)]^{Sys}$$

Under-Match



$$[\langle \iota, \gamma \rangle]^{Sys}$$

$$\cup\!\!\!\!\!\!\backslash \quad \text{Under-Match}$$

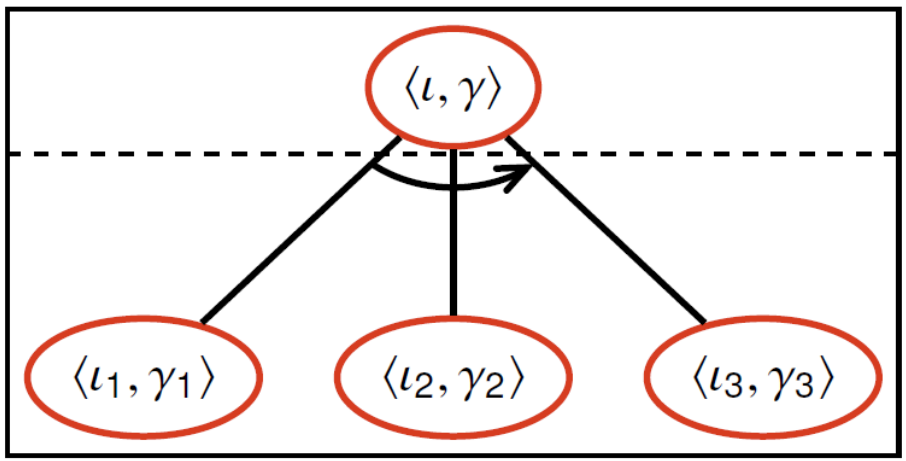
$$[\mathtt{OP}(\langle \iota 1, \gamma 1 \rangle, \langle \iota 2, \gamma 2 \rangle, \langle \iota 3, \gamma 3 \rangle)]^{Sys}$$

Under-Match property

Forgotten attack scenarios – refinement models less paths

$$[\langle \iota, \gamma \rangle]^{Sys} \supsetneq [\mathtt{OP}(\langle \iota 1, \gamma 1 \rangle, \langle \iota 2, \gamma 2 \rangle, \langle \iota 3, \gamma 3 \rangle)]^{Sys}$$

Over-Match



$$[\langle \iota, \gamma \rangle]^{Sys}$$

$$\cap\!\!\!\!\!\!\not\quad \text{Over-Match}$$

$$[\mathtt{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^{Sys}$$

Over-Match property

Extra attack scenarios – refinement models more paths

$$[\langle \iota, \gamma \rangle]^{Sys} \subsetneq [\mathtt{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^{Sys}$$

Complexity

	Meet	Under-Match	Over-Match	Match
OR	P	P	P	P
SAND	P	P	P	P
AND	NP-c	co-NP-c	co-NP	co-NP

High complexity is induced by the AND refinement (due to $\mathbb{M}$)

Witness of refinement property violation

Witness path generation by reduction to CTL model checking

Support for attack tree design

ATSyRA: Attack tree synthesis and risk analysis

ATSyRA studio tool http://atsyra2.irisa.fr/

- DSL for system specification
- Automated attack generation
- Attack tree refinement analysis

Outline

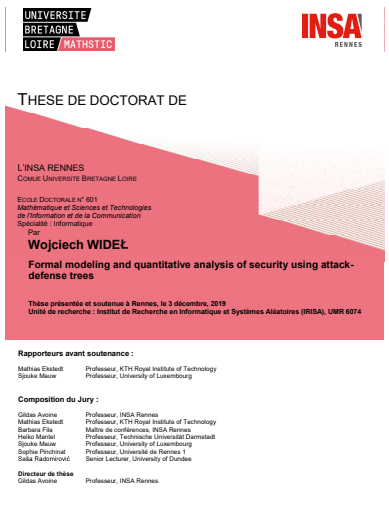
Surveys

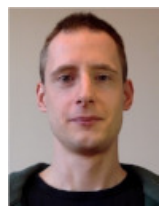
- Wojciech Wideł, Maxime Audinot, Barbara Fila, and Sophie Pinchinat. *Beyond 2014: formal methods for attack tree-based security modeling.* ACM Comput. Surv., 52(4):75:1–75:36, 2019.

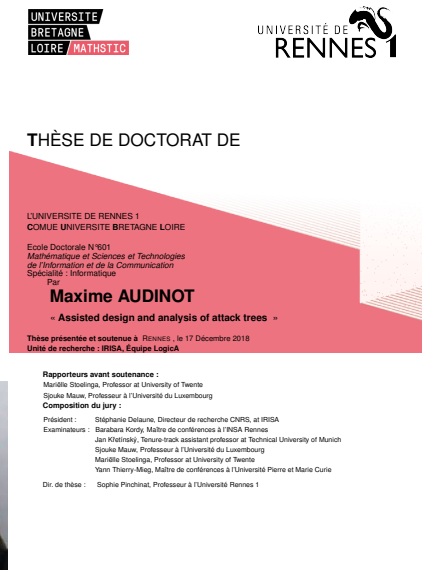
- Barbara Kordy, Ludovic Pièvre-Cambacédès, and Patrick Schweitzer. *Dag-based attack and defense modeling: Don't miss the forest for the attack trees.* Computer Science Review, 13–14:1–38, 2014.

Electronic versions available on
http://people.irisa.fr/Barbara.Kordy/publications.php

Ph.D. theses on foundations for attack trees







The most important open research problem

Automated generation of large attack trees

GraMSec http://www.gramsec.uni.lu/

The unique dissemination event for researchers and practitioners designing and using graphical approaches for modeling and analysis of security

Co-located with CSF

Do you have any questions?

Thank you for your attention