Assisted design of attack trees

Sophie Pinchinat

IRISA, LogicA team

Joint work with
Maxime Audinot, Barbara Kordy, Didier Vojtisek, Florence Wacheux

in collaboration with DGA Bruz, Lionel van Aertryck

Security Days 2019
Attack trees [Schneier99]

They model how to attack (a system in mind) – attacker’s point of view.

![Diagram of an attack tree with nodes labeled as follows: SAND "main goal", AND "subgoal 1", atomic goal_1, atomic goal_2, OR "subgoal 2", atomic goal_3, atomic goal_4, atomic goal_5.](image-url)
Attack trees [Schneier99]

They model how to attack (a system in mind) – attacker’s point of view.

Easy to read:
- hierarchical decomposition of subgoals
- regroup attacks around common features

Diagram:
```
   goal
  /     \
SAND "main goal" \
 /       \    /     \    /     \    /     \  
AND "subgoal 1" AND atomic goal_3 OR "subgoal 2" \
   /         /     \
atomic goal_1 atomic goal_2 atomic goal_4 atomic goal_5
```
Attack trees [Schneier99]

They model how to attack (a system in mind) – attacker’s point of view.

Easy to read:
- hierarchical decomposition of subgoals
- regroup attacks around common features

Uses in security analysis
- threat assessment
- quantitative analysis
- countermeasures selection
Design process of attack trees: successive refinements

"main goal"
Design process of attack trees: successive refinements

- "main goal" SAND
- "subgoal 1"
- atomic goal \( _3 \)
- "subgoal 2"
Design process of attack trees: successive refinements

Sophie Pinchinat
Design process of attack trees: successive refinements

Done since all leaves are atomic goals
Modeling the system: Transition system over Prop

- Attacker
- Security Agent
- Cam1
- Cam2
- Key
- Combi
- Window
- Door
- Safe
- Briefcase
- is_outside
- cam_1_on
- cam_2_on
- cam_1_on
- cam_2_on
- has_key
- cam_2_on
- has_combi
- cam_1_on

Sophie Pinchinat
Assisted design of attack trees
Modeling the system: Transition system over Prop

- **Security Agent**
  - **Cam1**
  - **Cam2**
  - **Key**
  - **Combi**
  - **Window**
  - **Door**
  - **Briefcase**
  - **Safe**

- **Attacker is outside**
  - **cam1_on**
  - **cam2_on**

- **Enter**
  - **cam1_on**
  - **cam2_on**

- **cam2_on**
  - **has_key**
  - **cam2_on**

- **ccam1_on**
  - **has_combi**
  - **ccam1_on**

...
Modeling the system: Transition system over Prop

is_outside
\[\text{cam}_1 \text{ on}\]
\[\text{cam}_2 \text{ on}\]

\[\text{cam}_1 \text{ on}\]
\[\text{cam}_2 \text{ on}\]

disable\text{ cam}_1

\[\text{cam}_2 \text{ on}\]

\[\text{has key}\]
\[\text{cam}_2 \text{ on}\]

\[\text{has combi}\]
\[\text{cam}_1 \text{ on}\]

...
Modeling the system: Transition system over Prop

is_outside

\[ \text{cam}_1 \text{on} \]
\[ \text{cam}_2 \text{on} \]

\[ \text{cam}_1 \text{on} \]
\[ \text{cam}_2 \text{on} \]

\[ \text{pick_key} \]
\[ \text{has_key} \]
\[ \text{cam}_2 \text{on} \]
\[ \text{has_combi} \]
\[ \text{cam}_1 \text{on} \]

...
Goals and their achievement

Paths in $S = (S, \rightarrow, \models) \models \subseteq S \times \text{Prop}$

$\pi = s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_n \in \text{Paths}(S)$

Goal $\langle \iota, \gamma \rangle$ over $\text{Prop}$

$\pi = s_0 \rightarrow s_1 \rightarrow s_2 \rightarrow s_3$

\[
\begin{align*}
 s_0 & \models \text{is\_outside} \\
 s_1 & \models \text{is\_outside}, \text{has\_key} \\
 s_2 & \models \text{has\_key} \\
 s_3 & \models \text{has\_key}
\end{align*}
\]
Goals and their achievement

Paths in $S = (S, \rightarrow, \models) \models \subseteq S \times \text{Prop}$

$\pi = s_0 \rightarrow s_1 \rightarrow \ldots \rightarrow s_n \in \text{Paths}(S)$

Goal $\langle \iota, \gamma \rangle$ over $\text{Prop}$

- $\iota$ is the precondition,
- $\gamma$ is the postcondition.

$\pi$ achieves $\langle \text{is\_outside}, \text{has\_key} \rangle$
Path semantics

\[ \langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle, \langle \iota_4, \gamma_4 \rangle \]

over

\[ J_\langle \iota, \gamma \rangle \]

\[ K_S = \{ \pi \in \text{Paths}(S) | \pi \text{ achieves atomic goal } \langle \iota, \gamma \rangle \} \]

\[ J \text{ OR } (\tau_1, \ldots, \tau_n) K_S = J_\tau_1 K_S \cup \ldots \cup J_\tau_n K_S \]

\[ J \text{ AND } (\tau_1, \ldots, \tau_n) K_S = \pi_1 \times \ldots \times \pi_n \]

Sophie Pinchinat

Assisted design of attack trees
Path semantics $[\tau]^S \subseteq \text{Paths}(S)$

$\text{AND}$

$\langle \iota_1, \gamma_1 \rangle$

$\langle \iota_2, \gamma_2 \rangle$

$\langle \iota_3, \gamma_3 \rangle$

$\langle \iota_4, \gamma_4 \rangle$

over

start $\xrightarrow{\iota_2} S_0 \xrightarrow{\iota_1, \gamma_4} S_1 \xrightarrow{\gamma_1, \gamma_3} S_2 \xrightarrow{\gamma_2} S_3$
Path semantics $\llbracket \tau \rrbracket^S \subseteq \text{Paths}(S)$

$\llbracket \langle \iota, \gamma \rangle \rrbracket^S = \{ \pi \in \text{Paths}(S) \mid \pi \text{ achieves atomic goal } \langle \iota, \gamma \rangle \}$
Path semantics $\llbracket \tau \rrbracket^S \subseteq \text{Paths}(S)$

- $\llbracket \langle \iota, \gamma \rangle \rrbracket^S = \{ \pi \in \text{Paths}(S) \mid \pi \text{ achieves atomic goal } \langle \iota, \gamma \rangle \}$
- $\llbracket \text{OR}(\tau_1, \ldots, \tau_n) \rrbracket^S = \llbracket \tau_1 \rrbracket^S \cup \ldots \cup \llbracket \tau_n \rrbracket^S$
Path semantics $[\tau]^S \subseteq \text{Paths}(S)$

- $[[\iota, \gamma]]^S = \{\pi \in \text{Paths}(S) | \pi \text{ achieves atomic goal } \langle \iota, \gamma \rangle\}$
- $[[\text{OR}(\tau_1, \ldots, \tau_n)]]^S = [[\tau_1]]^S \cup \ldots \cup [[\tau_n]]^S$
- $[[\text{SAND}(\tau_1, \ldots, \tau_n)]]^S = [[\tau_1]]^S \ldots \ldots [[\tau_n]]^S$

Sophie Pinchinat

Assisted design of attack trees

6 / 15
Path semantics $\left[\tau\right]^S \subseteq \text{Paths}(S)$

- $\left[\langle \iota, \gamma \rangle \right]^S = \{ \pi \in \text{Paths}(S) \mid \pi \text{ achieves atomic goal } \langle \iota, \gamma \rangle \}$
- $\left[\text{OR}(\tau_1, \ldots, \tau_n)\right]^S = \left[\tau_1\right]^S \cup \ldots \cup \left[\tau_n\right]^S$
- $\left[\text{SAND}(\tau_1, \ldots, \tau_n)\right]^S = \left[\tau_1\right]^S \ldots \ldots \left[\tau_n\right]^S$
- $\left[\text{AND}(\tau_1, \ldots, \tau_n)\right]^S = \bigwedge \left(\left[\tau_1\right]^S, \ldots, \left[\tau_n\right]^S\right)$
Sequential composition of paths

$$[\text{SAND}(\tau_1, \ldots, \tau_n)]^S = [\tau_1]^S \cdot \ldots \cdot [\tau_n]^S$$

$$\pi_1 \quad S_0 \xrightarrow{} S_1 \xrightarrow{} S_2$$

$$\pi_2 \quad S_2 \xrightarrow{} S_3 \xrightarrow{} S_4 \xrightarrow{} S_5 \xrightarrow{} S_6$$

$$\pi \quad S_0 \xrightarrow{} S_1 \xrightarrow{} S_2 \xrightarrow{} S_3 \xrightarrow{} S_4 \xrightarrow{} S_5 \xrightarrow{} S_6$$

$$\pi$$ is a sequential composition of $$\pi_1, \pi_2$$
Parallel composition of paths

\[ [\text{AND}(\tau_1, \ldots, \tau_n)]^S = \land([\tau_1]^S, \ldots, [\tau_n]^S) \]

\( \pi \) is parallel composition of \( \pi_1, \pi_2, \pi_3 \)
Exploit the path semantics of attack trees to perform various kinds of analysis.
The attack existence problem $[\tau]^S \neq \emptyset$?

$\tau$

potential ways of attacking

$\pi \in [\tau]^S$ iff $\pi$ is an $\tau$-attack on $S$

$S$

possible attacker behavior
The attack existence problem \([\tau]^S \neq \emptyset\)?
The attack existence problem $[[\tau]]^S \neq \emptyset$?

$$\pi \in [[\tau]]^S \text{ iff } \pi \text{ is an } \tau\text{-attack on } S$$

Decide whether $[[\tau]]^S \neq \emptyset$, in general?
The attack existence problem \([\tau]^S \neq \emptyset\)?

\[ \pi \in [\tau]^S \text{ iff } \pi \text{ is an } \tau\text{-attack on } S \]

**Theorem [Audinot, P., Schwarzentruber, Wacheux in GramSec 2018]**

- Existence of attacks is NP-complete.
- Existence of attacks for AND-free trees is NL-complete.
Analysis of refinements in the tree

\[ \langle \iota, \gamma \rangle \quad \text{vs.} \quad \langle \iota_1, \gamma_1 \rangle \quad \langle \iota_2, \gamma_2 \rangle \quad \langle \iota_3, \gamma_3 \rangle \]
Analysis of refinements in the tree

ideal situation:

\[ [[\langle i, \gamma \rangle]]^S = [\text{OP}(\langle i_1, \gamma_1 \rangle, \langle i_2, \gamma_2 \rangle, \langle i_3, \gamma_3 \rangle)]^S \]
Analysis of refinements in the tree

\[ \langle \iota, \gamma \rangle \text{ vs. } \langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle \]

Under-Match

\[ [\langle \iota, \gamma \rangle]^S \supseteq [\text{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^S \]

\[ \supseteq \text{forgotten attack scenarios} \]
Analysis of refinements in the tree

\[ \langle \iota, \gamma \rangle \] vs. \[ \langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle \]

Over-Match

\[ [\langle \iota, \gamma \rangle]^S \subseteq [\text{OP}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle, \langle \iota_3, \gamma_3 \rangle)]^S \]

\[ \varsubsetneq \text{extra scenarios} \]
### Complexity of the refinement analysis

[Audinot, P., Kordy in ESORICS 2017]

\[ \left[ \text{OP} \left( \langle i_1, \gamma_1 \rangle, \langle i_2, \gamma_2 \rangle, \langle i_3, \gamma_3 \rangle \right) \right]^S \nbowtie \left[ \langle i, \gamma \rangle \right]^S? \]

<table>
<thead>
<tr>
<th></th>
<th>⊆</th>
<th>⊇</th>
<th>⊑</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>SAND</td>
<td>P</td>
<td>P</td>
<td>P</td>
</tr>
<tr>
<td>AND</td>
<td>co-NP-complete</td>
<td>co-NP</td>
<td>co-NP</td>
</tr>
</tbody>
</table>
Witness of refinement property violation

Suppose Under-Match does not hold:

\[ \text{SAND}(\langle \nu_1, \gamma_1 \rangle, \langle \nu_2, \gamma_2 \rangle) \subseteq S \not\subseteq \llbracket \langle \nu, \gamma \rangle \rrbracket^S \]
Witness of refinement property violation

Suppose Under-Match does not hold:
\[
\left[\text{SAND}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle)\right]^S \not\subseteq \left[\langle \iota, \gamma \rangle\right]^S
\]

There is some extra scenario \( \pi \in \text{Paths}(S) \) with

\[
\pi \not\subseteq \left[\langle \iota, \gamma \rangle\right]^S
\]

but

\[
\pi \in \left[\text{SAND}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle)\right]^S
\]
Suppose Under-Match does not hold:
\[
\left[\text{SAND}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle)\right]^S \not\subseteq \left[\langle \iota, \gamma \rangle\right]^S
\]

There is some extra scenario \( \pi \in \text{Paths}(S) \) with
\[
\pi \not\in \left[\langle \iota, \gamma \rangle\right]^S
\]
but
\[
\pi \in \left[\text{SAND}(\langle \iota_1, \gamma_1 \rangle, \langle \iota_2, \gamma_2 \rangle)\right]^S
\]

Counterexample automated generation
By reduction to CTL model-checking witness generation
## Conclusion

State-based attack trees

- Atomic goals are reachability properties
- Path semantics w.r.t. the system model
- Existence of attacks and refinement analysis

[Implementation in ATSyRA Studio](http://atsyra2.irisa.fr)

- DSL for system specification + automated attack generation
- Attack trees editor + Refinements analysis
Conclusion

State-based attack trees
- Atomic goals are reachability properties
- Path semantics w.r.t. the system model
- Existence of attacks and refinement analysis

Assistance to attack tree design
Implementation in ATSyRA Studio (http://atsyra2.irisa.fr)
- DSL for system specification + automated attack generation
- Attack trees editor + Refinements analysis
Main publications

[European Symposium on Research in Computer Security 2017]
- State-based attack trees
- Path semantics
- Refinements analysis

[Graphical Models in Security 2018]
- Path semantics non-emptiness problem

[Computer Security Foundations 2018]
- Trace semantics of attack trees, and automata constructions
- Useful positions for a guided design

Survey (to appear in ACM Computing Surveys)
- Survey on recent research on formal methods for attack trees
## Main publications

<table>
<thead>
<tr>
<th>Publication</th>
<th>Topics</th>
</tr>
</thead>
<tbody>
<tr>
<td>European Symposium on Research in Computer Security 2017</td>
<td>State-based attack trees, Path semantics, Refinements analysis</td>
</tr>
<tr>
<td>Graphical Models in Security 2018</td>
<td>Path semantics non-emptiness problem</td>
</tr>
</tbody>
</table>
Main publications

[European Symposium on Research in Computer Security 2017]
- State-based attack trees
- Path semantics
- Refinements analysis

[Graphical Models in Security 2018]
- Path semantics non-emptiness problem

[Computer Security Foundations 2018]
- Trace semantics of attack trees, and automata constructions
- Useful positions for a guided design
## Main publications

<table>
<thead>
<tr>
<th>Conference/Media</th>
<th>Title/Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>European Symposium on Research in Computer Security 2017</td>
<td>State-based attack trees, Path semantics, Refinements analysis</td>
</tr>
<tr>
<td>Graphical Models in Security 2018</td>
<td>Path semantics non-emptiness problem</td>
</tr>
<tr>
<td>Computer Security Foundations 2018</td>
<td>Trace semantics of attack trees, and automata constructions, Useful positions for a guided design</td>
</tr>
</tbody>
</table>

Survey (to appear in ACM Computing Surveys)  
Survey on recent research on formal methods for attack trees