Probabilistic reasoning with graphical security models

Barbara Kordy

Clermont-Ferrand, January 7, 2016

Digital Confidence seminar
Joint work

Prof. Dr. Marc Pouly
Lucerne University of Applied Sciences and Arts

Dr. Patrick Schweitzer
University of Luxembourg
Probabilistic assessment of security scenarios

security model
ADTree

dependency model
Bayesian network

probabilistic assessment of attack-defense scenarios with dependencies
Outline

1. Attack–defense Trees
2. Probabilistic evaluation
3. Efficiency considerations
4. Wrap Up
### Attack–defense Trees

#### Modeling security scenarios

**Attack–defense tree (ADTree) [JLC’14]**

A tree-like representation of an attack–defense scenario depicting:

- How to attack a system
- How to protect against an attack

- Extend the **industrially recognized** model of attack trees [Schneier’99]
- Integrate
  - **Intuitive** representation features [IJSSE’12, ICISC’12]
  - **Formal** analysis techniques [GameSec’10, SIIS’11, JLC’14]
  - **Software** application ADTool [QEST’13]
Example: ADTree for infecting a computer
Propositional semantics for ADTrees [SIIS’11]

\[ B - \text{the set of non-refined nodes of ADTree } t \]

- \( x \in \{0, 1\}^B \) encodes whether actions from \( B \) succeed or not
  - Action \( A \in B \) succeeds if \( x(A) = 1 \)
  - Action \( A \in B \) does not succeed if \( x(A) = 0 \)

**Boolean function** \( f_t \) **for** \( t \)

\[ f_t: \{0, 1\}^B \rightarrow \{0, 1\} \] associates a Boolean value \( f_t(x) \in \{0, 1\} \)
with each vector \( x \in \{0, 1\}^B \)

\( x \) is called an **attack vector** if \( f_t(x) = 1 \)
ADTrees as Boolean functions

Domain of $f_t$ is composed of the non-refined nodes of $t$

Non-refined OR AND Countermeasure

$A$

$t$

$t'$

$t''$

$A$

$f_t(A) = A$

$f_t = f_{t'} \lor f_{t''}$

$f_t = f_{t'} \land f_{t''}$

$f_t = f_{t'} \land \neg f_{t''}$

Barbara Kordy
Example: Boolean function for infecting a computer

\[ f_t = \left( (X_{EA} \lor X_{US}) \land \neg (X_{IA} \land (X_{RA} \land \neg X_{FA})) \right) \land X_{EV} \]
Example: attack vector

\[ f_t = \left( (X_{EA} \lor X_{US}) \land \neg (X_{IA} \land (X_{RA} \land \neg X_{FA})) \right) \land X_{EV} \]

attack vector

\begin{bmatrix}
1 & 0 & 1 & 0 & 0 & 1
\end{bmatrix}
Knowing the *probabilities* of particular attacks allow us to

- Identify *the most vulnerable components*
- Determine *the strategic points*
- Decide *which protective measures to implement*
Bottom-up evaluation of probability on ADTrees [ICISC’12]

Probability of a disjunctive subtree

Probability of a conjunctive subtree

Probability of a countered subtree

Similarly for subtrees rooted in a defense node
**Probability of a disjunctive subtree**

\[ x + y - xy \]

**Probability of a conjunctive subtree**

**Probability of a countered subtree**
Bottom-up evaluation of probability on ADTrees [ICISC’12]

Probability of a disjunctive subtree

\[ x + y - xy \]

Probability of a conjunctive subtree

\[ xy \]

Probability of a countered subtree

\[ x \]

\[ y \]
Bottom-up evaluation of probability on ADTrees [ICISC’12]

Probability of a disjunctive subtree

\[ x + y - xy \]

Probability of a conjunctive subtree

\[ xy \]

Probability of a countered subtree

\[ x(1 - y) \]
Bottom-up evaluation of probability on ADTrees [ICISC’12]

Probability of a disjunctive subtree

\[
x + y - xy
\]

Probability of a conjunctive subtree

\[
xy
\]

Probability of a countered subtree

\[
x(1 - y)
\]

Similarly for subtrees rooted in a defense node.
Example: probability for infecting a computer

infect computer 0.669375

virus on system 0.74375

execute virus 0.9

antivirus 0.15000000000000002

install antivirus 0.8

run antivirus 0.25

fake antivirus 0.25
Limitations

The bottom-up procedure does not take dependencies between actions into account.

However, in practice

- Installing and running an antivirus
- Distributing and executing a virus

are not independent actions.

Thus, the standard bottom-up evaluation is not suitable for probabilistic assessment of attack–defense trees.
Challenges

1. How to design the appropriate formalism?

2. How to ensure that calculations reflect the reality?

3. How to guarantee the efficiency of the evaluation?
Proposed Framework [INS’16]

security model
ADTree
Proposed Framework [INS’16]

- security model: ADTree
- dependency model: Bayesian network
Proposed Framework [INS’16]

security model
ADTree

dependency model
Bayesian network

probabilistic assessment
of attack–defense scenarios
with dependencies
Probabilistic evaluation

Modeling probability of dependent actions

Bayesian network
A directed, acyclic graph that reflects the conditional interdependencies between variables associated with the nodes of the network

Dependent variables

Conditional probability table for $Y$

\[
\begin{align*}
p(Y = 1|X = 1) &= 0.7 \\
p(Y = 1|X = 0) &= 0.2 \\
p(Y = 0|X = 1) &= 0.3 \\
p(Y = 0|X = 0) &= 0.8
\end{align*}
\]
Constructing Bayesian network $BN_t$ for ADTree $t$

From an ADTree

- $t$ – ADTree
- $\mathcal{B}$ – set of all non-refined nodes of $t$

To a Bayesian network

- Elements of $\mathcal{B}$ are nodes of the Bayesian network $BN_t$
- Relations between actions are depicted by edges in $BN_t$
- Conditional probability tables quantify dependencies between actions
Example: $\text{BN}_t$ for infecting a computer ADTree

\begin{align*}
p(X_{EA} = 1|X_{FA} = 1) &= 0.9 \\
p(X_{EA} = 1|X_{FA} = 0) &= 0.5
\end{align*}

\begin{align*}
p(X_{US} = 1|X_{FA} = 1) &= 0.4 \\
p(X_{US} = 1|X_{FA} = 0) &= 0.5
\end{align*}

\begin{align*}
p(X_{EV} = 1|X_{EA} = 1, X_{US} = 1) &= 0.9 \\
p(X_{EV} = 1|X_{EA} = 1, X_{US} = 0) &= 0.2 \\
p(X_{EV} = 1|X_{EA} = 0, X_{US} = 1) &= 0.8 \\
p(X_{EV} = 1|X_{EA} = 0, X_{US} = 0) &= 0.1
\end{align*}

\begin{align*}
p(X_{IA} = 1) &= 0.6
\end{align*}

\begin{align*}
p(X_{RA} = 1|X_{IA} = 1) &= 0.9 \\
p(X_{RA} = 1|X_{IA} = 0) &= 0.0
\end{align*}
Joint probability distribution for network \( \text{BN}_t \)

\[
p(X_{EA}, X_{US}, X_{IA}, X_{RA}, X_{FA}, X_{EV}) = \\
p(X_{EV}|X_{EA}, X_{US}) \times p(X_{EA}|X_{FA}) \times p(X_{US}|X_{FA}) \times p(X_{FA}) \times p(X_{RA}|X_{IA}) \times p(X_{IA})
\]
Propositional semantics using algebraic operations

- **Non-refined**
  - \( f_t(A) = A \)
  - \( t' \) and \( t'' \)

- **OR**
  - \( f_t = f_{t'} \lor f_{t''} \)
  - \( t' \) and \( t'' \)

- **AND**
  - \( f_t = f_{t'} \land f_{t''} \)
  - \( t' \) and \( t'' \)

- **Countermeasure**
  - \( f_t = f_{t'} \land \neg f_{t''} \)
  - \( t' \) and \( t'' \)
Propositional semantics using algebraic operations

Non-refined

OR

\[ f_t(A) = A \]
\[ id_A \]

AND

\[ f_t = f_{t'} \land f_{t''} \]
\[ f_{t'} \times f_{t''} \]

Countermeasure

\[ f_t = f_{t'} \land \neg f_{t''} \]
\[ f_{t'} \times (1 - f_{t''}) \]
**Probability computation**

$x \in \{0, 1\}^B$ – vector of successful/unsuccessful actions

**Probability of attack vector $x$**

$$f_t(x) \times p(x)$$

**Probability related to ADTree $t$**

$$P(t) = \sum_{x \in \{0,1\}^B} f_t(x) \times p(x)$$

**Probability of the most probable attack vector**

$$P_{\text{max}}(t) = \max_{x \in \{0,1\}^B} f_t(x) \times p(x)$$
Compatibility results

**Theorem**

Probability computations on propositionally equivalent ADTrees yield the same result.

**Observation**

For ADTree $t$ without dependent actions, $P(t)$ coincides with the result of the bottom-up computation.
Efficiency problems

The number of configurations $x$ grows exponentially with the number of involved actions. For large systems, it is therefore not feasible to

- Enumerate all the values of $f_t$
- Enumerate all the values of the joint probability distribution for $BN_t$

$$P(t) = \sum_{x \in \{0,1\}^B} f_t(x) \times p(x)$$

$$P_{\text{max}}(t) = \max_{x \in \{0,1\}^B} f_t(x) \times p(x)$$
Efficiency considerations

security model
ADTree

dependency model
Bayesian network

probabilistic assessment
of attack–defense scenarios
with dependencies
Efficiency considerations

security model
ADTree

dependency model
Bayesian network

constraint
reasoning
fusion

probabilistic assessment
of attack–defense scenarios
with dependencies
Efficiency considerations

Local indicators

\[ f_t = \left( (X_{EA} \lor X_{US}) \land \neg (X_{IA} \land (X_{RA} \land \neg X_{FA})) \right) \land X_{EV} \]

\[ \phi_1(Y_1, X_{EA}, X_{US}) = 1 \text{ exactly if } Y_1 = \max\{X_{EA}, X_{US}\} \]
\[ \phi_2(Y_2, X_{RA}, X_{FA}) = 1 \text{ exactly if } Y_2 = X_{RA} \times (1 - X_{FA}) \]
\[ \phi_3(Y_3, X_{IA}, Y_2) = 1 \text{ exactly if } Y_3 = X_{IA} \times Y_2 \]
\[ \phi_4(Y_4, Y_1, Y_3) = 1 \text{ exactly if } Y_4 = Y_1 \times (1 - Y_3) \]
\[ \phi_5(Y_t, Y_4, X_{EV}) = 1 \text{ exactly if } Y_t = Y_4 \times X_{EV} \]
Global indicator function $\phi_t$ for ADTree $t$

**Domain of $\phi_t$:**
- Non-refined nodes of $t$
- Inner variables of all local indicators

**Global indicator function $\phi_t = \text{product of all local indicators } \phi_i$**

$$\phi_t(Y_1, Y_2, Y_3, Y_4, Y_t, X_{EA}, X_{US}, X_{IA}, X_{RA}, X_{FA}, X_{EV}) =$$

$$\phi_1(Y_1, X_{EA}, X_{US}) \times \phi_2(Y_2, X_{RA}, X_{FA}) \times \phi_3(Y_3, X_{IA}, Y_2) \times$$

$$\phi_4(Y_4, Y_1, Y_3) \times \phi_5(Y_t, Y_4, X_{EV})$$

$\Phi_t$ indicates valid assignments with respect to $f_t$
Important property

Theorem

Consider an ADTree $t$ over the set of non-refined nodes $B$ and the global indicator function $\phi_t$ with the set of inner variables $\mathcal{Y}$.

$$\forall x \in \{0, 1\}^B \exists! y \in \{0, 1\}^\mathcal{Y}, \text{ such that } \phi_t(y, x) = 1$$

Corollary: $\forall x \in \{0, 1\}^B$

$$\max_{y \in \{0, 1\}^\mathcal{Y}} \phi_t(y, x) = \sum_{y \in \{0, 1\}^\mathcal{Y}} \phi_t(y, x) = 1$$
Filtering interesting assignments of $\phi_t$

We are only interested in assignments such that $\phi_t = 1$ and $Y_t = 1$

$$Y_t \times \phi_t(y, x)$$
Expressing $f_t$ with its global indicator

$$\forall x \in \{0, 1\}^B : \max_{y \in \{0, 1\}^Y} \phi_t(y, x) = \sum_{y \in \{0, 1\}^Y} \phi_t(y, x) = 1$$

$$\forall x \in \{0, 1\}^B$$

$$\max_{y \in \{0, 1\}^Y} \left( Y_t \times \phi_t(y, x) \right) = \sum_{y \in \{0, 1\}^Y} \left( Y_t \times \phi_t(y, x) \right) = f_t(x) = \begin{cases} 1, & \text{if } x \text{ is an attack vector} \\ 0, & \text{otherwise} \end{cases}$$
Factorized form for probability formulas

Probability of attack vector $x$

$$f_t(x) \times p(x) = \max_{y \in \{0,1\}^y} \left( Y_t \times \phi_t(y, x) \times p(x) \right)$$

Probability related to ADTree $t$

$$P(t) = \sum_{x \in \{0,1\}^B} f_t(x) \times p(x) = \sum_{(y,x) \in \{0,1\}^{Y \cup B}} \left( Y_t \times \phi_t(y, x) \times p(x) \right)$$

Probability of the most probable attack vector

$$P_{\text{max}}(t) = \max_{x \in \{0,1\}^B} f_t(x) \times p(x) = \max_{(y,x) \in \{0,1\}^{Y \cup B}} \left( Y_t \times \phi_t(y, x) \times p(x) \right)$$
Our framework in the context of semiring theory

- Inference problem over the arithmetic semiring $\langle \mathbb{R}, +, \times \rangle$

  \[
P(t) = \sum_{(y, x) \in \{0, 1\}^Y \cup B} \left( Y_t \times \phi_t(y, x) \times p(x) \right)
  \]

- Inference problem over the product t-norm semiring $\langle [0, 1], \max, \times \rangle$

  \[
P_{\text{max}}(t) = \max_{(y, x) \in \{0, 1\}^Y \cup B} \left( Y_t \times \phi_t(y, x) \times p(x) \right)
  \]
Local computation

Powerful local computation algorithms

- Fusion
- Variable elimination \( \Rightarrow \) **smart distributivity**

<table>
<thead>
<tr>
<th></th>
<th>Complexity bound</th>
<th>Using Nenok tool [IJAIT’10]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Direct computation</td>
<td>( 2^{11} )</td>
<td>3.422sec</td>
</tr>
<tr>
<td>Using fusion</td>
<td>( 2^4 )</td>
<td>0.031sec</td>
</tr>
</tbody>
</table>

Complexity bounded by a **structural parameter** of the problem
Summary
Summary

security model
ADTree
Summary

security model
ADTree

dependency model
Bayesian network
Summary

security model
ADTree

dependency model
Bayesian network

probabilistic assessment of attack–defense scenarios with dependencies
Summary

security model
ADTree

dependency model
Bayesian network

constraint reasoning fusion

probabilistic assessment of attack–defense scenarios with dependencies
Addressing challenges

1. How to design the *appropriate formalism*?

2. How to ensure that calculations *reflect the reality*?

3. How to guarantee the *efficiency* of the evaluation?
Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated

2. How to ensure that calculations **reflect the reality**?

3. How to guarantee the **efficiency** of the evaluation?
Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated

2. How to ensure that calculations **reflect the reality**?
   - Real-life data take dependencies into account
   - Complement ADTree with additional information

3. How to guarantee the **efficiency** of the evaluation?
Addressing challenges

1. How to design the **appropriate formalism**?
   - Used by industry, intuitive & well formalized
   - Security model and dependency network are kept separated

2. How to ensure that calculations **reflect the reality**?
   - Real-life data take dependencies into account
   - Complement ADTree with additional information

3. How to guarantee the **efficiency** of the evaluation?
   - Local computation algorithms
   - Existing software tools, well-known heuristics
Where to take it from here?

- Find the best elimination sequence for Bayesian ADTrees
  - NP-complete in general
  - Prediction is possible for specific families of graphs

- Extend to probability distributions
  - Probability dependent on time

- Interface ADTool [QEST’13] with Nenok
  - Automated probability assessment of large scale scenarios
Take home message

Key to success
- Appropriate technique
  - Components
  - Attributes
- Well-founded model
  - Syntax
  - Semantics
- Erroneous results
- Understand theory
Barbara Kordy, Marc Pouly, and Patrick Schweitzer.  

Barbara Kordy, Sjouke Mauw, Saša Radomirović, and Patrick Schweitzer.  

Barbara Kordy, Ludovic Piètre-Cambacédès, and Patrick Schweitzer.  
DAG-Based Attack and Defense Modeling: Don’t Miss the Forest for the Attack Trees.  

Marc Pouly.  
Nenok - a software architecture for generic inference.  

Barbara Kordy, Sjouke Mauw, and Patrick Schweitzer.  

Barbara Kordy, Marc Pouly, and Patrick Schweitzer.  
References

Barbara Kordy, Marc Pouly, and Patrick Schweitzer.

Barbara Kordy, Piotr Kordy, Sjouke Mauw, and Patrick Schweitzer.

Barbara Kordy, Sjouke Mauw, Matthijs Melissen, and Patrick Schweitzer.
Attack–Defense Trees and Two-Player Binary Zero-Sum Extensive Form Games Are Equivalent.
In Tansu Alpcan, Levente Buttyán, and John S. Baras, editors, Decision and Game Theory for Security (GameSec 2010), volume 6442 of LNCS, pages 245–256. Springer, 2010.

Alessandra Bagnato, Barbara Kordy, Per Håkon Meland, and Patrick Schweitzer.

Bruce Schneier.
Attack Trees.