From Attack Trees to Attack-Defense Trees with TTool-AI

Alan Birchler De Allende, Bastien Sultan, Ludovic Apvrille
birchler@eurecom.fr, {bastien.sultan, ludovic.apvrille}@telecom-paris.fr

MDE Intelligence 2024, Linz

Outline

Introduction

Our contribution in a nutshell...

Given an attack tree (provided in graphical format), draws an attack-defense tree including countermeasures mitigating the attack leaves of the input tree.
It is intended to be an **assistant** rather than an autonomous generator.

Introduction

@Bastien: Please, draw an attack tree.

Introduction

@Bastien: Please, draw an attack tree.
@GPT:

Introduction

@Bastien: Please, draw an attack tree.
@GPT:

@Bastien: And an attack-defense tree?

Introduction

@Bastien: Please, draw an attack tree.
@GPT:

@Bastien: And an attack-defense tree?
@GPT:

Introduction

Our contribution in a nutshell...

Given an attack tree (provided in graphical format), draws an attack-defense tree including countermeasures mitigating the attack leaves of the input tree.
It is intended to be an **assistant** rather than an autonomous generator.

... and in depth

- Builds upon TTool-AI framework, adapting its LLM interaction mechanisms:
  - Initial knowledge injection
  - Automated response analysis and feedback loop

Introduction

Our contribution in a nutshell...

Given an attack tree (provided in graphical format), draws an attack-defense tree including countermeasures mitigating the attack leaves of the input tree.
It is intended to be an **assistant** rather than an autonomous generator.

... and in depth

- Builds upon TTool-AI framework, adapting its LLM interaction mechanisms:
  - Initial knowledge injection
  - Automated response analysis and feedback loop
- A preceding NLP stage to help the LLM with suggesting more effective countermeasures

TELECOM Paris

IP PARIS

Introduction

Our contribution in a nutshell...

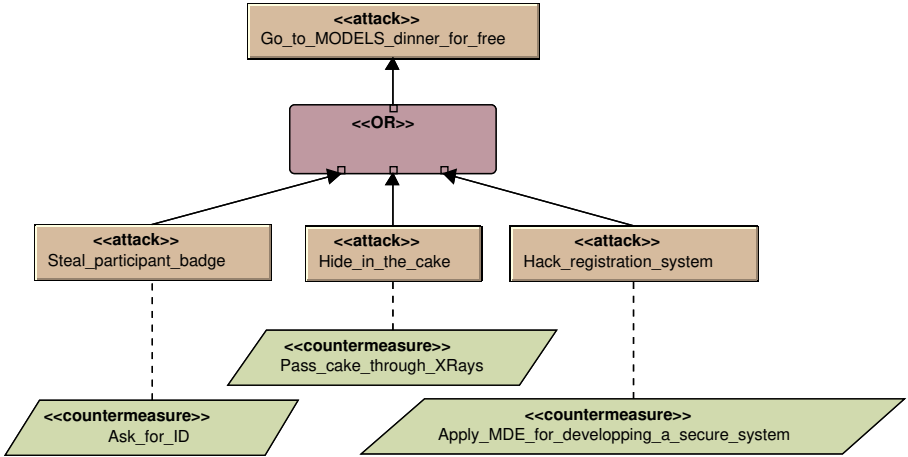
Given an attack tree (provided in graphical format), draws an attack-defense tree including countermeasures mitigating the attack leaves of the input tree.
It is intended to be an **assistant** rather than an autonomous generator.

... and in depth

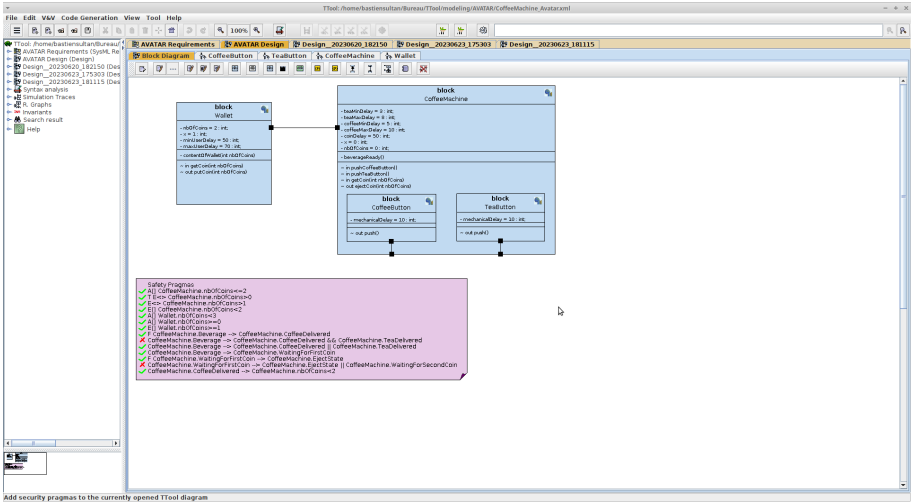
- Builds upon TTool-AI framework, adapting its LLM interaction mechanisms:
  - Initial knowledge injection
  - Automated response analysis and feedback loop
- A preceding NLP stage to help the LLM with suggesting more effective countermeasures
- A one-click approach, while allowing users to view the interactions between the tool and the LLM, and intervene by adding their own knowledge and constraints

Outline

Some context – Attack-defense trees

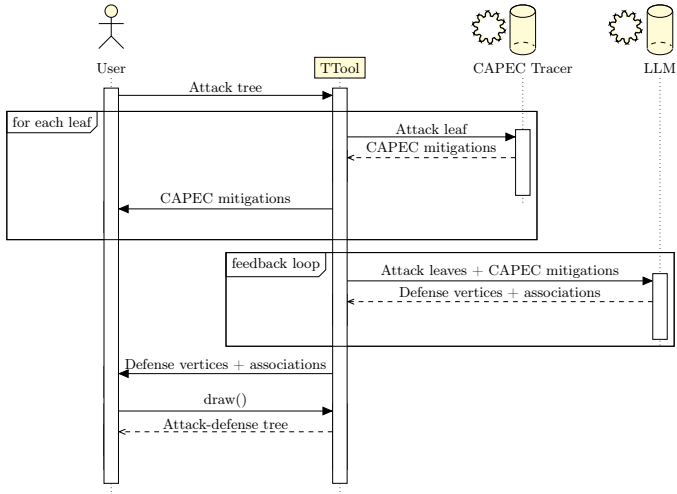


Some context – TTool

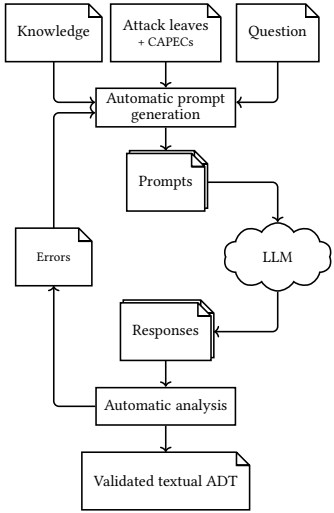


Outline

Contribution overview



TTool-AI interaction



Automatic prompt generation

Knowledge

```
When you are asked to identify mitigations for a provided list of attack
     steps, return them as a JSON specification formatted as follows:
"mitigations": [{\"name\":  \"NameOfMitigation\", \"description\": \"The
     description of the mitigation and how it prevents the attack.\"}, \
    "attacksteps\": [\"TheNameOfAProvidedAttackStep\" ...]} ...]}
# Respect: All words in \"name\" must be conjoined together.
# Respect: There must be no more than forty characters in \"name\".
# Respect: For each word in \"name\", its first letter must be
    capitalized.
[...]
```

Automatic prompt generation

Question

```
Using the specified JSON format, identify a list of possible mitigations
     that would prevent an attacker from using most, if not all, of the
    provided attack steps to further advance the attacker's attack
    scenario. Each mitigation should be associated with at least one
    attack step from the provided attack step list. If applicable, use
    the provided list of possible countermeasures as support for
    identifying mitigations. Do respect the JSON format, and provide
    only JSON (no explanation before or after).
```

Automatic analysis

Data: String r (initial LLM response), int *remainingIterations*
Result: String r_i (improved LLM response)

String *query* $\leftarrow \epsilon$
$err_{json} \leftarrow setOfJSONFormatErrorsIn(r)$
$err_{const} \leftarrow setOfConstraintsErrorsIn(r)$
while $(err_{json}, err_{const}) \neq (\emptyset, \emptyset) \vee remainingIterations > 0$ **do**
    query $\leftarrow$ "In the previous response" $\cdot\ r\ \cdot$ "the following errors were found, do correct them:"
    foreach $e \in err_{json}$ **do**
        query $\leftarrow$ *query* $\cdot\ e$
    end
    foreach $e \in err_{const}$ **do**
        query $\leftarrow$ *query* $\cdot\ e$
    end
    sendToLLM(*query*)
    $r \leftarrow$ *NewLLMResponse*
    $err_{json} \leftarrow setOfJSONFormatErrorsIn(r)$
    $err_{const} \leftarrow setOfConstraintsErrorsIn(r)$
    remainingIterations $\leftarrow$ *remainingIterations* $- 1$
end
$r_i \leftarrow r$

Outline

Experimental insights

Setup

- Three input attack-tree diagrams, related to three different specifications (a CPU, a mobile application, a cloud service infrastructure)
- Generation of attack-defense trees on this basis, by a security engineer and by our framework
- Subsequent grading of the ADTs (syntactic correctness, number of countermeasure/attack leaves associations, ...)

Experimental insights

(a) Cloud service case study

Creator	Complexity	Semantic correctness	Completeness	Generation time (s)
Engineer	8	1.00	0.42	3600
Tool	10	0.70	0.37	633

(b) CPU case study

Creator	Complexity	Semantic correctness	Completeness	Generation time (s)
Engineer	5	1.00	0.50	3600
Tool	9	0.44	0.31	882

(c) Social network application case study

Creator	Complexity	Semantic correctness	Completeness	Generation time (s)
Engineer	6	1.00	0.46	3600
Tool	17	0.41	0.47	912

Outline

Conclusions

In a nutshell

- LLM-based tool integrated into TTool, an open-source SysML toolkit
- **One-click generation**
- **Rapid first ADT construction**, serving as a tool for quickly generating a draft that can be further refined.

Future works

- Need for further evaluations
- Generalize the countermeasures associations to higher abstraction level nodes
- Improve the generated prompts, integrate other LLMs and countermeasures databases

Questions?

Latest version of TTool includes TTool-AI, ADT generation, nice tree pictures, ...and much more! `ttool.telecom-paris.fr`