Transforming Graphical System Models to Graphical Attack Models

Joint work with Marieta Georgieva Ivanova, René Rydhof Hansen, and Florian Kammüller

Christian W. Probst

Language-Based Technology, DTU Compute

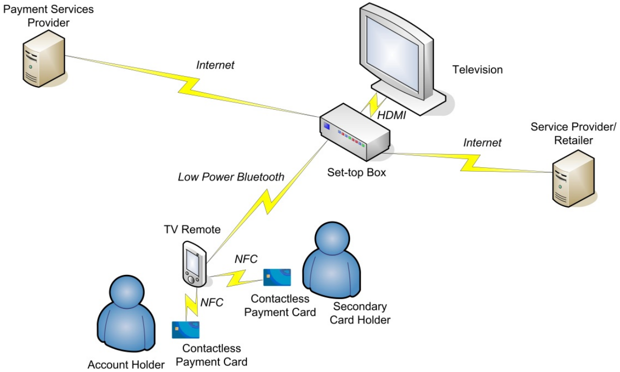
From organisational models to attacks

- System Model
- Analytic approach
- Success based on experience and imagination of the modeller

Attack Attack Attack Attack Attack Attack Attack Attack

- Attack trees
- Descriptive method
- Success based on experience and imagination of the consultant/defender

Example System



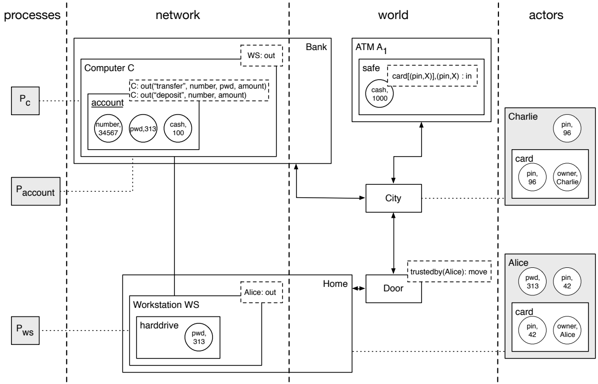
System Model Components

- **Locations** in the organisation linked by directed edges in the graph.
- **Actors** in the modelled organisation.
- **Processes** modelling information sharing or policies.
- **Items** modelling tangible assets in the modelled organisation, for example, access cards, harddrives, etc.
- **Data** modelling intangible assets.

Constraining Actions

- **Policies** regulate access to locations and assets. Policies consist of required credentials and enabled actions.
- **Credentials** are required data, items, or an identity.

Graphical System Model



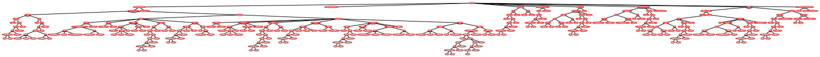
KLAIM: Kernel Language for Agents Interaction and Mobility

- Mobile components
- Communication via tuple spaces
- Distribute/retrieve data and processes
- Localities as first-class citizens
    - Created, communicated, scoping
- Similar ideas have been adapted by industry
- Mostly based on LINDA
    - JavaSpaces by Sun
    - TSpaces by IBM
    - Plus implementations for other programming languages
    - Also used for ubiquitous computing (sTuples) and the Semantic Web (Triple Spaces, Semantic Web Spaces)

Attack Generation is White-box Testing of System Models

- Structured system model for systematic, formal treatment.
- With clearly defined semantics.
- Specification of attacker goals.
- Formal specification of transformation.

Graphical Attack Model



Attack Alternatives

Root node “steal money”

- Hire more skilled attacker.
- Acquire card and access codes.
- Attack set-top box from LAN.
- Make cardholder pay.
- Social-engineer cardholder to make payment.
- Tamper payment data.
- Fake information the cardholder sees on TV.
- Fake set-top box.
- Intercept connection between set-top box and payment provider.

Generating Attack Trees

The General Approach.

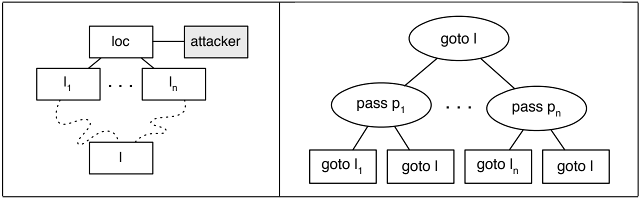
- Identify the policy P to break.
- Identify the required assets to fullfil P.
- Try to obtain these assets.

No Asset Mobility

- Assumes an asset in the system, which an attacker should not be able to obtain.
- Assets are (for now) immobile.
- Apply general approach for all locations of the asset.

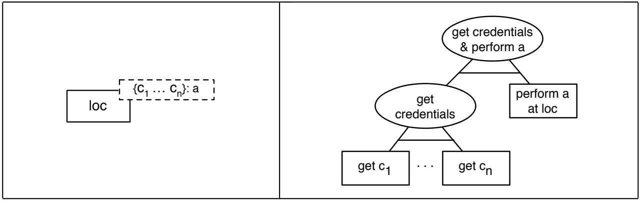
Transforming Locations

- Locations are transformed into disjunction of all paths through the model.
- Recursively invokes attack transformation for the first step and the rest.



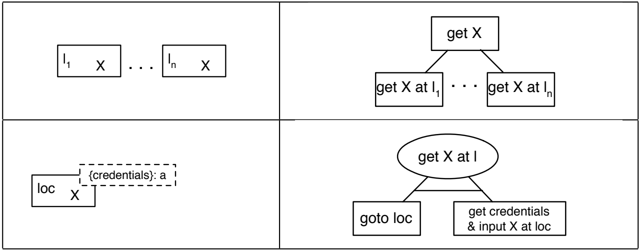
Transforming Policies

- For every policy, missing credentials are identified.
- Recursively invokes attack transformation for missing credentials.



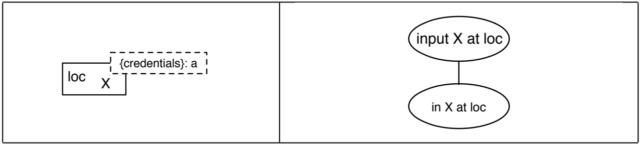
Assets

- Assets can be available at different locations.
- Each location is transformed to a get action.



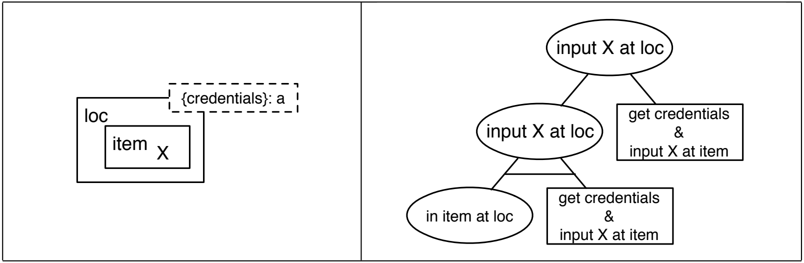
Asset at a Location

- Assets at locations/items is transformed to **in** action.



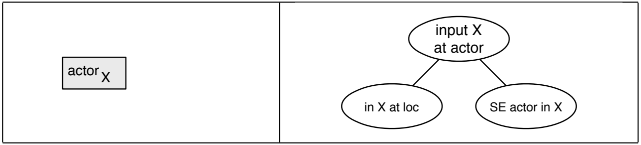
Asset Contained in an Item

- For assets contained in an item, that item is first obtained.
- Then, the transformation is invoked again

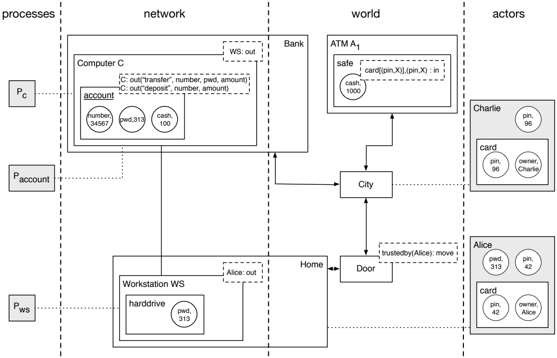


Asset at an Actor

- For assets at actors, social engineering actions are generated.



The IPTV Case Study – Attacker Charlie



goal: get cash

goal: in[C,PIN(C)](cash)

get C, PIN(C)

goal: get Charlies' credentials and perform action

goal: get Alice's credentials and perform action

get card

goto Home

goto Door and get trust

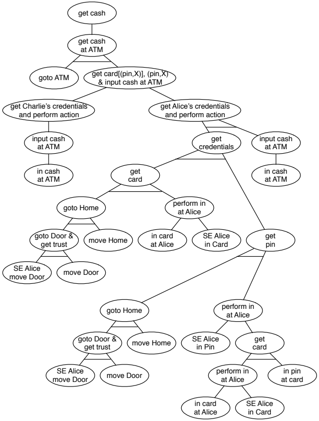
A1: break in, A2: carer, A3: IPTV

move Door

move Home

perform in at Alice

Resulting Attack Model – Charlie



The Problem of Details

Feature creep

- Attack trees will contain many fine-grained details.
- These are very hard to generate from models.
  - Scan wireless connection to obtain access code for card.
  - Requires knowledge about card, communication between set-top box and card, availability of scanner
- Similar to the elephant.
- Can partly be based on libraries, but...

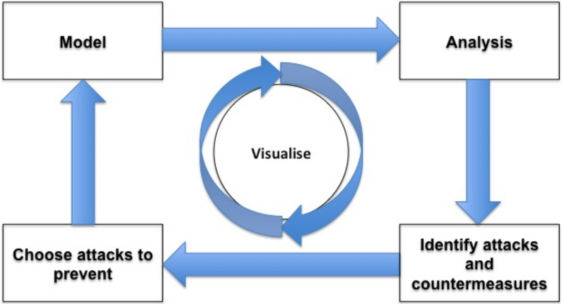
Adding Asset Mobility

- Attackers can make assets move.
- Obtaining assets may be “simpler” at other locations:
  - Less risk of detection.
  - Blame somebody else.
  - Faster attack.
- Attack generation takes all possible asset locations into account.
- There is no free dinner – the resulting attack trees may become huge!

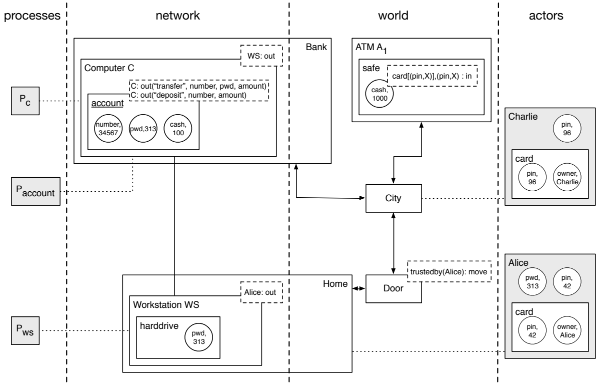
The TRE$_S$PASS Approach to Risk Assessment

- Information security threats to organisations have changed completely over the last decade
- New attacks cleverly exploit multiple organisational vulnerabilities, involving physical security and human behaviour.
- Defenders need to make rapid decisions regarding which attacks to block, as both infrastructure and attacker knowledge change rapidly.

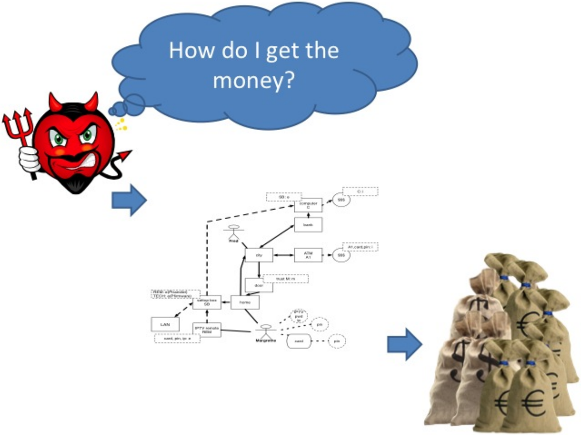
The TRE$_S$PASS Process


Model
Analysis
Visualise
Identify attacks and countermeasures
Choose attacks to prevent

The TRE$_S$PASS Model



The Attack Navigator



The Attack Navigator

- Tool to support prediction, prioritisation, and prevention of complex attack scenarios.
- Also an environment where all tools developed within the project can be viewed, accessed and connected.

Conclusion

- System models provide a systematic way to assess vulnerabilities in organisations...
- ...and can be transformed to attack trees.
- This will enables us to map system components to quantitative results for attack trees.
- Right level of detail is important!