Modelling and Simulating Systems Security Policy

Tristan Caulfield    David Pym
Department of Computer Science
University College London
Security managers must make decisions about policy while largely uncertain of the outcome.

Outcome influenced by many factors, including technology, organisational culture, difficulty of compliance, etc.

Need to think about how all of these factors interact when making a decision about security policy.

Solution

A methodology for creating models that capture these interactions and allow security managers to explore the effects and consequences of different policy choices.
Framework and Methodology

- Capable of modelling the interactions of policy, technology, and human decisions.

- **Compositional**: more complex systems can be expressed as combinations of smaller, complete models.
Distributed Systems Modelling

- Uses concepts of Locations, Resources and Processes.
- Processes manipulate resources in locations.
- Mathematical structures for all of these components.
  - Locations: directed graph
  - Resources: ordered monoids
  - Processes: process algebra
- These sit inside an environment which models the world outside of the system. Events in the environment are drawn from probability distributions.
Processes

- Processes constructed from basic actions
  - Allows for parallel or concurrent execution, choice, etc.

L, R, E \xrightarrow{a} L', R', E'

Process $E$ with access to resources $R$ at location $L$ can evolve by performing action $a$ to become the process $E'$ with access to resources $R'$ at location $L'$. 
Formalising a Model

Create a formal definition of a model and an operator for composition.

<table>
<thead>
<tr>
<th>A model ( M = (G(\mathcal{V}[\mathcal{R}], \mathcal{E}), \mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{I}) ) is a tuple that consists of:</th>
</tr>
</thead>
<tbody>
<tr>
<td>- a location graph ( G ), with vertices labelled by resources</td>
</tr>
<tr>
<td>- a set of actions ( \mathcal{A} )</td>
</tr>
<tr>
<td>- a set of processes ( \mathcal{P} ),</td>
</tr>
<tr>
<td>- a set of located actions ( \mathcal{L} ), where ( l \in \mathcal{L} = (a \in \mathcal{A}, v \in \mathcal{V}) )</td>
</tr>
<tr>
<td>- a set of interfaces ( \mathcal{I} ), where ( i \in \mathcal{I} = (In \subseteq \mathcal{V}, Out \subseteq \mathcal{V}, L \subseteq \mathcal{L}) )</td>
</tr>
</tbody>
</table>

The environment associates located actions with probability distributions: \( Env : \mathcal{L} \rightarrow \text{ProbDist} \).
Composition Operator

Two models are composed using specific interfaces: $M_{1_{i,j}} |_{l_{2,k}} M_{2}$

- Models are composed together using specific interfaces.
- When uncomposed, some processes at input locations are started by the environment.
- When composed, these processes occur as consequences of the other model.
Framework that implements these mathematical concepts.
The framework is written in the Julia language.
Designed for scientific computing.
Nice features for doing process-based simulation.

```julia
function example_process(proc :: Process)
    # do stuff...
    hold(proc, 5hours) # wait a while
    # etc.
end
```
Model: Tailgating

- Looks at employee behaviour when entering the office.
- Choice: tailgate or queue at reception.
- Choice: intervene or ignore when tailgating is observed.
- Params: security guard, number of receptionists
- Attackers always try to tailgate.
Model: Document Sharing

- How documents are shared within the office.
- Choice: by email, physical media, or shared drive.
- Attackers walking around the office look for physical media.
Devices can be lost on public transportation.

Very simple model.

Lost devices may contain confidential documents.
Creating Locations

#define locations

local loc_outside :: Location
local loc_atrium :: Location
loc_lobby = Location("Lobby")
loc_entry = Location("Entryway")
loc_reception = Location("Reception")

#link locations

link(loc_lobby, loc_entry)
link(loc_reception, loc_lobby)

The other locations are part of interfaces, and will be linked when the model is executed.
# Create a door

entry_door :: Door = Door(loc_lobby)
distrib(entry_door, loc_lobby)

# ...

# Later on, agents must access the door before
# moving from the foyer to the hallway
access(proc, agent, entry_door)
move(proc, agent, loc_lobby, loc_entry)

- Door can be changed for a turnstile, etc.
Agents are implemented as a bundle of a resource, a process, and locations.

Locations are for things the agent is holding, remembering, etc.

The process moves the agent’s resource around the model’s locations.
Agents make decisions at specific choice points. Based on preferences and the state of the model. Preferences drawn from distributions. Agents select choice $C$ that maximises:

$$\text{val}(C) = p_{\text{sec}} f_{\text{sec}}(C) + p_{\text{prod}} f_{\text{prod}}(C)$$
Managers care about various attributes.
Each attribute has a target value.
Attributes have different weights.
Care about deviations from target values differently for each attribute.

\[ Utility = \sum_{a \in A} w_a f_a \left( v_a - \bar{v}_a \right) \]
Agents make a series of decisions between different choices, each of which has a value to the manager:

\[ D = C_1^{\lambda_1} C_2^{\lambda_2} \cdots C_n^{\lambda_n} \]

Choice selected during model execution has \( \lambda = 1 \), others \( \lambda = 0 \).

Can also calculate the expected value:

\[ E[D] = \lambda_1 \ln(C_1) + \lambda_2 \ln(C_2) + \cdots + \lambda_n \ln(C_n) \]

\( \lambda_1, \ldots, \lambda_n \) are now probabilities and sum to 1.
Security manager concerned about employees accessing documents on the globally shared folder.

Trying to decide between two alternative policies: Email or Portable Media.

Also concerned about the number of documents attackers might find on media.

Global share is the most productive option for employees, followed by email, and then portable media.
Agents’ Preferences

- Two companies with different attitudes towards security

These values can come from collected data.
High security company cares a lot more documents being shared
- Low target for documents shared
- Exponential decrease in utility when target is missed

<table>
<thead>
<tr>
<th>Company</th>
<th>a</th>
<th>w_a</th>
<th>v̅_a</th>
<th>f_a(x)</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Sec</td>
<td>Share</td>
<td>0.25</td>
<td>20</td>
<td>- (e^{0.01x} + 0.01x - 1)</td>
</tr>
<tr>
<td></td>
<td>Found</td>
<td>1</td>
<td>0</td>
<td>-e^x + 1</td>
</tr>
<tr>
<td>Low Sec</td>
<td>Share</td>
<td>0.1</td>
<td>200</td>
<td>.003x</td>
</tr>
<tr>
<td></td>
<td>Found</td>
<td>0.3</td>
<td>0</td>
<td>-e^x + 1</td>
</tr>
</tbody>
</table>
From the Document Sharing Model
- 10,000 simulation runs
- Neutral policy is current state.

<table>
<thead>
<tr>
<th>Company</th>
<th>Policy</th>
<th>Email</th>
<th>Share</th>
<th>Media</th>
<th>Found</th>
<th>$E[U]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Sec</td>
<td>Email</td>
<td>151.73</td>
<td>7.75</td>
<td>0.00</td>
<td>0.00</td>
<td>0.059</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>0.00</td>
<td>167.32</td>
<td>0.00</td>
<td>0.00</td>
<td>-1.209</td>
</tr>
<tr>
<td></td>
<td>Media</td>
<td>0.00</td>
<td>54.42</td>
<td>105.33</td>
<td>2.43</td>
<td>-10.543</td>
</tr>
<tr>
<td>Low Sec</td>
<td>None</td>
<td>0.00</td>
<td>167.29</td>
<td>0.00</td>
<td>0.00</td>
<td>-0.010</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>54.87</td>
<td>107.96</td>
<td>0.00</td>
<td>0.00</td>
<td>-0.028</td>
</tr>
<tr>
<td></td>
<td>Media</td>
<td>0.00</td>
<td>154.15</td>
<td>12.24</td>
<td>0.07</td>
<td>-0.035</td>
</tr>
</tbody>
</table>
Expected utility calculated from stochastic processes $\Phi_a$.

- In theory, this can be calculated using queuing theory and knowledge of the model, distributions of environment events, etc.
- In practice, this is very difficult.
- Instead, we use lots of model executions to approximate the value.

$$E[Utility] = E \left[ \sum_{a \in A} w_a f_a (v_a(\Phi_a) - \bar{v}_a) \right]$$
m1 = create_device_loss_model()
m2 = create_tailgating_model()
m3 = create_document_sharing_model()
m4 = compose(m1, "Outside", m2, "Outside")
m = compose(m3, "Atrium", m4, "Atrium")
Now with additional attributes from the other models:
- High Sec cares about data loss
- Low Sec cares about wasted time

<table>
<thead>
<tr>
<th>Company</th>
<th>a</th>
<th>$w_a$</th>
<th>$\bar{v}_a$</th>
<th>$f_a(x)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>High Sec</td>
<td>Share</td>
<td>0.25</td>
<td>20</td>
<td>$-(e^{0.01x} + 0.01x - 1)$</td>
</tr>
<tr>
<td></td>
<td>Found</td>
<td>1</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td></td>
<td>Lost</td>
<td>1.1</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td></td>
<td>Rcpt Time</td>
<td>0.01</td>
<td>3000</td>
<td>$-.002x$</td>
</tr>
<tr>
<td></td>
<td>Tgate</td>
<td>1.5</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td>Low Sec</td>
<td>Share</td>
<td>0.1</td>
<td>200</td>
<td>$.003x</td>
</tr>
<tr>
<td></td>
<td>Found</td>
<td>0.3</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td></td>
<td>Lost</td>
<td>0.4</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td></td>
<td>Rcpt Time</td>
<td>0.4</td>
<td>2000</td>
<td>$-.005x$</td>
</tr>
<tr>
<td></td>
<td>Tgate</td>
<td>0.2</td>
<td>0</td>
<td>$-e^x + 1$</td>
</tr>
<tr>
<td>Company</td>
<td>Policy</td>
<td>#Grd</td>
<td>#Rcpt</td>
<td>Email</td>
</tr>
<tr>
<td>---------</td>
<td>--------</td>
<td>------</td>
<td>-------</td>
<td>--------</td>
</tr>
<tr>
<td>High Sec</td>
<td>Email</td>
<td>2</td>
<td>2</td>
<td>174.20</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>2</td>
<td>1</td>
<td>174.60</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>1</td>
<td>2</td>
<td>173.92</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>1</td>
<td>1</td>
<td>173.67</td>
</tr>
<tr>
<td></td>
<td>Media</td>
<td>2</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td>Low Sec</td>
<td>None</td>
<td>1</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Media</td>
<td>1</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Email</td>
<td>1</td>
<td>2</td>
<td>67.21</td>
</tr>
<tr>
<td></td>
<td>None</td>
<td>2</td>
<td>2</td>
<td>0.00</td>
</tr>
<tr>
<td></td>
<td>Media</td>
<td>1</td>
<td>1</td>
<td>0.00</td>
</tr>
</tbody>
</table>
Conclusions

- These models are just examples.
- The model can be about whatever is needed.
- Currently working on models that focus more on information security
  - Servers, services, firewalls, the flow of data, etc.
- Future work includes better models of decision-making
  - Different models of discounting
  - Herding behaviour
Thanks!

- Tristan Caulfield
- t.caulfield@ucl.ac.uk
Run models on the cluster!

```julia
using ClusterManagers
addprocs(SGEManager(100, ""),
    exename="/home/tcaulfie/julia/bin/julia",
    exeflags="--worker",
    qsubopts=["-l", "h_vmem=1.0G,tmem=1.0G,h_rt=2:00:00"])
ret = @parallel (combine) for i = 1 : 10000 # ....
```

Output info for visualisations

```julia
@jslog(LOG_MAX, proc.simulation, Dict{Any,Any}(
    "time" => now(proc.simulation), "type" => "event",
    "id" => string(object_id(agent)),
    "value" => "Observed tailgating."
))
```
qlen = length(get_store(loc_reception).resources)
qtime = (mean(dist_reception_time) / num_receptionists) * qlen

lateness = time_of_day(now(proc) + qtime) - time_of_day(expected_arrival_time)

tailgate_prod = 1.2
if lateness > 0
    tailgate_prod = min(1.2, 1.2 + (lateness / 5minutes))
end

choices = Dict( :reception => [1.1, 1.6],
                :tailgate => [tailgate_prod, 1.2])
choice = choose(agent, choices)