Insider Attacks

Formal Models and Analyses

Christian W. Probst

Language-Based Technology, DTU Informatics

Aalborg University

About me

- Associate Professor at The Technical University of Denmark in the section of Language-Based Technology
- M.Sc. in Computer Science and Ph.D. in Engineering from Universität des Saarlandes, Germany
- Research Interests
  - Optimizing Compilers and Static Analysis
  - System Security
  - and especially Insider Threats

Insider Attacks

Example 1: The Hard Disk Example: Naive user and absent policy

On April 5, 2003, Banner Therapy employee Christina Binney was discharged from her position for "misconduct", and instructed not to return to the office. Christina Binney was also a co-founder of Banner Therapy.
The company claimed she impermissibly removed a hard drive from her work computer and took it home over the weekend to prepare for a client meeting. The company claimed that the disk drive removal crippled Banner's operations and placed vital company data at risk. Binney explained that a Banner customer requested a meeting on a Friday for the following Monday morning. To prepare, she chose to physically remove the entire hard drive from her work computer, rather than take time to transfer the files to a disk.
At the time, Banner Therapy had neither company policy about taking work equipment home nor established computing protocols. When Binney attempted to return to work on Monday, she was denied access; this inability to enter the workplace prevented her from returning the hard drive as she claimed she intended to do.

Example 2: The Email Example: Ordinary user generates an extraordinary amount of email

In early October 2007, Alex Greene wanted to update his subscription to a Department of Homeland Security intelligence bulletin. In doing so, he mistakenly hit “reply all”, and his request arrived in the electronic mailboxes of several thousand government and private sector security specialists. The result was what commentators described as a mini distributed denial of service attack. There were more than 2.2 million emails pinging among approximately 7,500 recipients before the email server was forced to shut down.
The information contained in the bulletin is unclassified, but nevertheless, the decision to respond inadvertently compromised classified contact and departmental information. Individual subscribers with security classifications remained anonymous until they also hit reply, responding from work accounts that included automatically generated signatures.

Example 3: The Trade Secret Example: Malicious user steals trade secrets

On June 16, 2007, FBI agents arrested a pair of engineers, who both had worked for NetLogic Microsystems (NLM) until July 2003. The two men used money from mainland China to create and incorporate a company for the sole purpose of exploiting the secrets they stole.

Lee and Ge downloaded sensitive NLM documents onto their home computers. NLM data sheets are "top-level confidential technical descriptions of their products", including information described in enough specificity to enable someone to produce the technology. Together, the men accumulated the information needed to design and produce their own lines of microprocessors and microchips. To finance the business, the men contacted Beijing FBNI Electronic Technology Development Company Ltd, and entered into an agreement to develop and sell microprocessor chips. Both men were able to access proprietary information without exceeding their individual authorizations. Investigators uncovered evidence that the venture capitalist had ties to the Chinese government and military.

Example 4: The Tax Fraud Example: Perimeter definition and system design

Harriette Walters and others are accused for perpetrating the biggest fraud in Washington's history. Until her arrest, "Walters was a 26-year tax employee known as a problem solver with a knack for finding solutions by using the department's antiquated and balky computers or finding a way around them." She allegedly used her position to produce fake checks for bogus refunds with fictitious names; the total is said to exceed (USD) $50 million.

The scheme involved Washington's new Integrated Tax System. During design phase, Walters "contributed to the decision that her unit, which handled real estate tax refunds, be left out of it." At the time, the decision seemed to make sense for cost reasons.

The scheme exploited several loopholes: each check was under the $40,000 threshold for requiring a supervisor's approval, and no action was taken to cancel the first check or confirm that it had not already been cashed.

Discussion

???

- What is an insider?
- What could important points to consider be?

What Is the Problem?

- We depend increasingly upon complex information systems
- Focus on the vulnerability to
  - Computer crime
  - Security attacks

[RAND Report, 2004]

"The insider threat is perhaps the greatest threat to [society, information system, ...]"

What Is the Problem?

Securing against the Inside

- Protect against attacks from an insider
- Insider has
  - Better knowledge/information
  - Better access
- Hard or impossible to distinguish from admissible actions
- Little research on analysing systems

Another Aspect

- Criminal investigations increasingly contain a digital component
  - Or rather, the number of cases *without* a digital component are decreasing rapidly—and most cases contain *both*
- Increasingly investigations span several teams or consist of a combinations of individual cases
  - Where boundaries might change dynamically.
  - But the individual investigations can benefit from each other, or
  - Should be combined.
- We need an approach that allows to
  - Combine the physical and digital domain to allow analysis of
    - Interactions between both domains, and
    - Dependencies between actions in different domains.
  - Combine models from different sources.

Insider and Insider Threat

Insider

An insider with respect to rules R is a user who may take an action that would violate some set of rules R in the security policy, were the user not trusted.
The insider is trusted to take the action only when appropriate, as determined by the insiders discretion.

[Matt Bishop, NSPW Panel, 2005]

Insider and Insider Threat

Insider

An insider with respect to rules R is a user who may take an action that would violate some set of rules R in the security policy, were the user not trusted.
The insider is trusted to take the action only when appropriate, as determined by the insiders discretion.

[Matt Bishop, NSPW Panel, 2005]

Insider and Insider Threat

Insider

An insider with respect to rules R is a user who may take an action that would violate some set of rules R in the security policy, were the user not trusted.
The insider is trusted to take the action only when appropriate, as determined by the insiders discretion.

[Matt Bishop, NSPW Panel, 2005]

Insider Threat

The insider threat is the threat that an insider may abuse his discretion by taking actions that would violate the security policy when such actions are not warranted.

Discussion

???

- What do you think of this definition?

Insider–A Definition

Insider (Dagstuhl 2008)

An insider is a person that has been legitimately empowered with the right to access, represent, or decide about one or more assets of the organization's structure.

Discussion

???

- If we want to protect "systems" against Insider Attacks, what are the properties we should be interested in?
- And, what are the "systems" we are interested in?

The System Model

Goal

- Model real-world systems
  - Relevant properties
  - Relevant actions
- Map system model to an analysable formalism
- Apply static analyses
- (next step—add non-technical properties)

The Extendable System Model (ExaSym)

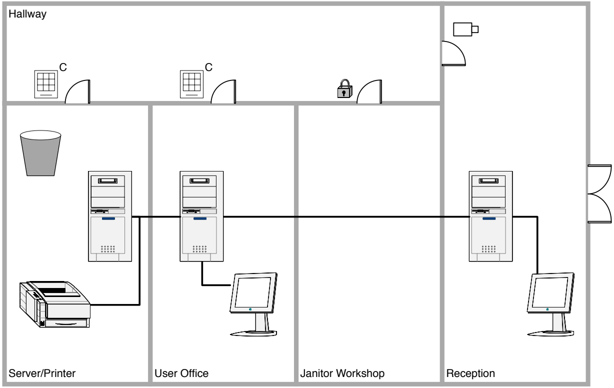
System Model

- Directed graph
- Models all locations that can be
  - Accessed
  - Store data
- Models all entities that can move in the system

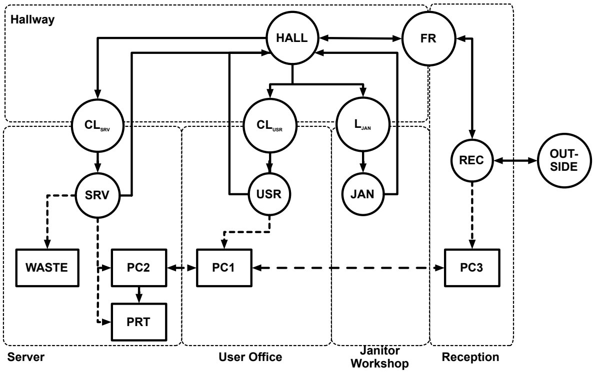
Analyses

- Effect of a given sequence of actions
- Reachability on the system graph
- Match observed actions to system model

Example System



Abstracted Example System



System Model

Infrastructure

- Locations, Connections
  - Directed graph
- Domains
  - Group locations that share accessibility
- The infrastructure can be extended to include *all* relevant infrastructure

System Model

Data and Actors

- Data
  - Models any kind of object in the system
- Actors (or Processes)
  - Model mobile entities
  - Can perform actions along edges
  - Bound to a certain domain
- Actions
  - Model input/output of data, evaluating code, and movement

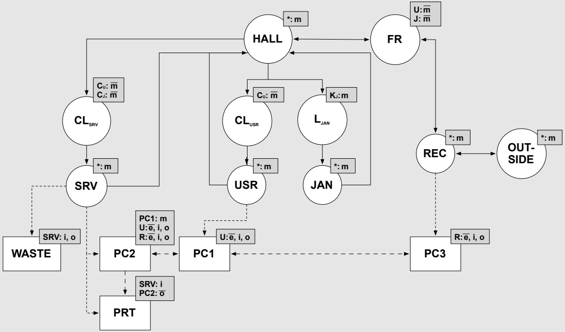
System Model

Extensions

- Access Control
  - Limit access to locations
  - Data items used as keys
  - Can be based on knowledge/identity/location of actor
- Encryption/Decryption of data
  - Data used as encryption keys
  - Similar limitations as for AC
- Logging of actions
  - Extend system to trace certain actions in the system

Example System

System Graph



acKLAIM

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)

acKlaim Syntax

Localities

$$\begin{array}{lcll} \ell & ::= & l & \text{locality} \\ & | & u & \text{locality variable} \end{array}$$

Nets

$$\begin{array}{lcll} N & ::= & l ::^{\delta} [P]^{\langle n,\kappa \rangle} & \text{process} \\ & | & l ::^{\delta} \langle et \rangle & \text{located tuple} \\ & | & N_1 \parallel N_2 & \text{net composition} \end{array}$$

Processes

$$\begin{array}{lcll} P & ::= & \mathbf{nil} & \text{nil process} \\ & | & a.P & \text{action prefixing} \\ & | & P_1 \mid P_2 & \text{parallel composition} \\ & | & A & \text{process invocation} \end{array}$$

Actions

$$\begin{array}{lcll} a & ::= & \mathbf{out}\,(t)\,@\ell & \text{output} \\ & | & \mathbf{in}\,(T)\,@\ell & \text{input} \\ & | & \mathbf{read}\,(T)\,@\ell & \text{read} \\ & | & \mathbf{eval}\,(P)\,@\ell & \text{remote exec} \\ & | & \mathbf{move}\,(\ell) & \text{re-locate} \end{array}$$

Process Declaration

$$\begin{array}{lcll} D & ::= & A \triangleq P & \text{process decl.} \end{array}$$

acKlaim Syntax

Localities

ℓ	$::=$	l	locality
	$\mid$	u	locality variable

Nets

N	$::=$	$l ::^{\delta} [P]^{\langle n, \kappa \rangle}$	process
	$\mid$	$l ::^{\delta} \langle et \rangle$	located tuple
	$\mid$	$N_1 \parallel N_2$	net composition

Processes

P	$::=$	**nil**	nil process
	$\mid$	$a.P$	action prefixing
	$\mid$	$P_1 \mid P_2$	parallel composition
	$\mid$	A	process invocation

Actions

a	$::=$	**out** $(t)\,@\ell$	output
	$\mid$	**in** $(T)\,@\ell$	input
	$\mid$	**read** $(T)\,@\ell$	read
	$\mid$	**eval** $(P)\,@\ell$	remote exec
	$\mid$	**move** (ℓ)	re-locate

Process Declaration

D	$::=$	$A \triangleq P$	process decl.

Tuples and Templates

Tuples

$t \quad ::= \quad \ell \mid \ell, t$ tuples

Evaluated Tuples

$et \quad ::= \quad l \mid l, et$ evaluated tuple

Templates

$T \quad ::= \quad F \mid F, T$ templates

$F \quad ::= \quad \ell \mid !u$ template fields

Discussion

???

- Is acKlaim powerful enough to model “real-world” scenarios?

Semantics

grant

$$\mathrm{grant}(n, l, \kappa, a, l') = \begin{cases} \texttt{true} & \text{if } a \in \delta_{l'}(n) \vee a \in \delta_{l'}(l) \vee \\ & \quad \exists k \in \kappa : a \in \delta_{l'}(k) \\ \texttt{false} & \text{otherwise} \end{cases}$$

$\rightsquigarrow$, $\succ$

$$\frac{\exists (l, l') \in \mathsf{Con} : \mathrm{grant}(n, l, \kappa, \mathbf{e}, l') \wedge \langle \mathcal{I}, n, \kappa \rangle \succ (l', t)}{\langle \mathcal{I}, n, \kappa \rangle \succ (l, t)}$$

$$\frac{\mathrm{grant}(n, l, \kappa, a, t) \wedge \langle \mathcal{I}, n, \kappa \rangle \succ (l, t)}{\langle \mathcal{I}, n, \kappa \rangle \rightsquigarrow (l, t, a)}$$

Semantics

in,out,read

$$\frac{[\![t]\!] = et \qquad \boxed{\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, l', \mathbf{o})}}{l ::^{\delta} [\mathbf{out}\,(t)\,@l'.P]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} [P']^{\langle n',\kappa'\rangle} \succ\!\!\longrightarrow_{\mathcal{I}} l ::^{\delta} [P]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} [P']^{\langle n',\kappa'\rangle} \parallel l' ::^{\delta'} \langle et\rangle}$$

$$\frac{\mathrm{match}([\![T]\!], et) = \sigma \qquad \boxed{\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, l', \mathbf{i})}}{l ::^{\delta} [\mathbf{in}\,(T)\,@l'.P]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} \langle et\rangle \succ\!\!\longrightarrow_{\mathcal{I}} l ::^{\delta} [P\sigma]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} \mathbf{nil}}$$

$$\frac{\mathrm{match}([\![T]\!], et) = \sigma \qquad \boxed{\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, l', \mathbf{r})}}{l ::^{\delta} [\mathbf{read}\,(T)\,@l'.P]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} \langle et\rangle \succ\!\!\longrightarrow_{\mathcal{I}} l ::^{\delta} [P\sigma]^{\langle n,\kappa\rangle} \parallel l' ::^{\delta'} \langle et\rangle}$$

Semantics

eval,move

$$\frac{\boxed{\langle \mathcal{I}, n, \kappa \rangle \rightsquigarrow (l, l', \mathbf{e})} \wedge l' \text{exists}}{l ::^{\delta} [\mathbf{eval}\,(Q)\,@l'.P]^{\langle n,\kappa \rangle} \rightarrowtail_{\mathcal{I}} l ::^{\delta} [P]^{\langle n,\kappa \rangle} \parallel l' ::^{\delta'} [Q]^{\langle n,\kappa \rangle}}$$

$$\frac{\boxed{\langle \mathcal{I}, n, \kappa \rangle \rightsquigarrow (l, l', \mathbf{m})} \wedge l' \text{exists}}{l ::^{\delta} [\mathbf{move}\,(l')\,.P]^{\langle n,\kappa \rangle} \rightarrowtail_{\mathcal{I}} l ::^{\delta} \mathbf{0} \parallel l' ::^{\delta'} [P]^{\langle n,\kappa \rangle}}$$

Discussion

???

- How can we add logging to this system model?

Semantics

logging

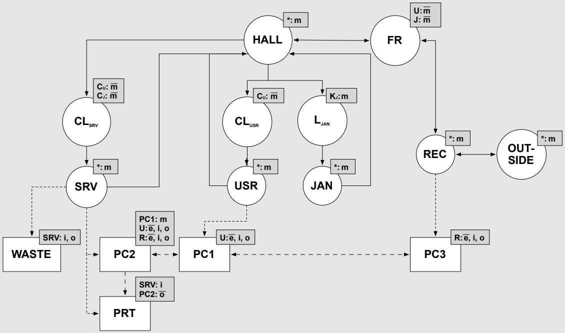
- Extend semantics with a global Lamport clock $\mathcal{T}$, and
- A logging component $\mathcal{L}$
- Whenever a logged action is performed, update $\mathcal{L}$

out

$$\frac{[\![t]\!] = et \qquad \boxed{\langle \mathcal{I}, n, \kappa \rangle \rightsquigarrow (l, l', \mathbf{o})} \qquad t = \mathcal{T} \qquad \boxed{\mathcal{L}' = \mathcal{L}[t \mapsto (x, l', \bar{\mathbf{o}})]}}{\begin{array}{r}\mathcal{L}, \mathcal{T} \vdash l ::^{\delta} [\mathbf{out}\,(t)\,@l'.P]^{\langle n,\kappa \rangle} \parallel l' ::^{\delta'} [P']^{\langle n',\kappa' \rangle} \rightarrowtail_{\mathcal{I}} \\ \mathcal{L}', \mathcal{T}' \vdash l ::^{\delta} [P]^{\langle n,\kappa \rangle} \parallel l' ::^{\delta'} [P']^{\langle n',\kappa' \rangle} \parallel l' ::^{\delta'} \langle et \rangle\end{array}}$$

Example System revisited

System Graph



Example System revisited

Abstract System

$$
\begin{array}{ll}
\mathrm{OUTSIDE} ::^{\langle \star \mapsto \mathbf{m} \rangle} \mathbf{nil} & \parallel \\
\mathrm{REC} ::^{\langle \star \mapsto \mathbf{m} \rangle} \mathbf{nil} & \parallel \; \mathrm{PC3} ::^{\langle \mathrm{R} \mapsto \bar{\mathbf{e}}, \mathbf{i}, \mathbf{o} \rangle} \mathbf{nil} \quad \parallel \\
\mathrm{FR} ::^{\langle \mathrm{U} \mapsto \overline{\mathbf{m}}, \mathrm{J} \mapsto \overline{\mathbf{m}} \rangle} \mathbf{nil} & \parallel \; \mathrm{HALL} ::^{\langle \star \mapsto \mathbf{m} \rangle} \mathbf{nil} \quad \parallel \\
\mathrm{L_{JAN}} ::^{\langle \mathrm{k_J} \mapsto \overline{\mathbf{m}} \rangle} \mathbf{nil} & \parallel \; \mathrm{JAN} ::^{\langle \star \mapsto \mathbf{m} \rangle} J \quad \parallel \\
\mathrm{CL_{USR}} ::^{\langle \mathrm{c_U} \mapsto \overline{\mathbf{m}} \rangle} \mathbf{nil} & \parallel \; \mathrm{USR} ::^{\langle \star \mapsto \mathbf{m} \rangle} U \quad \parallel \\
\mathrm{PC1} ::^{\langle \mathrm{U} \mapsto \bar{\mathbf{e}}, \mathbf{i}, \mathbf{o} \rangle} \mathbf{nil} & \parallel \\
\mathrm{CL_{SRV}} ::^{\langle \mathrm{c_U} \mapsto \overline{\mathbf{m}}, \mathrm{c_J} \mapsto \overline{\mathbf{m}} \rangle} \mathbf{nil} & \parallel \; \mathrm{SRV} ::^{\langle \star \mapsto \mathbf{m} \rangle} \mathbf{nil} \quad \parallel \\
\mathrm{WASTE} ::^{\langle \mathrm{SRV} \mapsto \mathbf{i}, \mathbf{o} \rangle} \langle\rangle & \parallel \; \mathrm{PRT} ::^{\langle \mathrm{SRV} \mapsto \mathbf{i}, \mathrm{PC2} \mapsto \bar{\mathbf{o}} \rangle} \langle\rangle \quad \parallel \\
\mathrm{PC2} ::^{\langle \mathrm{U} \mapsto \bar{\mathbf{e}}, \mathbf{i}, \mathbf{o} \rangle} \mathbf{nil} &
\end{array}
$$

Discussion

???

- What can we use these models for?

Analysing System Models

Analysing System Models

Reachability

- Which places can actors reach?
- Which data can they access?
- a-priori analysis for vulnerabilities

Matching logged sequences

- What might have happened unnoticed?
- a-posteriori analysis of logged actions

Analysing System Models

Effect of processes

- Compute effects of given processes
- Based on Flow Logic specification

Online-analysis of logged events

- Surveillance of systems
- Prediction of location of actors

Flow Logic Analysis

Nets

$$
\begin{array}{lll}
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{N}} l ::^{\delta} [P]^{\langle n,\kappa\rangle} & \text{iff} & (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{\lfloor l\rfloor,n,\kappa} P \\
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{N}} l ::^{\delta} \langle et\rangle & \text{iff} & \langle et\rangle \in \hat{T}(\lfloor l\rfloor) \\
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{N}} N_1 \parallel N_2 & \text{iff} & (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{N}} N_1 \wedge (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{N}} N_2
\end{array}
$$

Processes

$$
\begin{array}{lll}
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} \mathbf{nil} & \text{iff} & \mathit{true} \\
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} P_1 \mid P_2 & \text{iff} & (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} P_1 \wedge (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} P_2 \\
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} A & \text{iff} & (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} P \quad \text{if } A \triangleq P \\
(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} a.P & \text{iff} & (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{A}}^{l,n,\kappa} a \wedge (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{l,n,\kappa} P
\end{array}
$$

Flow Logic Analysis

Actions

$$(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{A}}^{l,n,\kappa} \textbf{out}\,(t)\,@\ell'$$
$$\quad\text{iff}\quad \forall \hat{l} \in \hat{\sigma}(\ell') \colon\; (\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, \hat{l}, \mathbf{o}) \Rightarrow \hat{\sigma}[\![t]\!] \subseteq \hat{T}(\hat{l}))$$

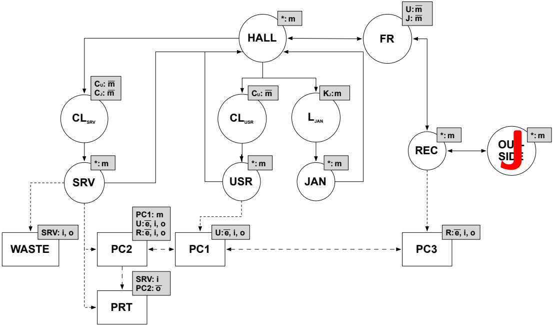
$$(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{A}}^{l,n,\kappa} \textbf{in}\,(T)\,@\ell'$$
$$\quad\text{iff}\quad \forall \hat{l} \in \hat{\sigma}(\ell') \colon\; (\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, \hat{l}, \mathbf{i}) \Rightarrow \hat{\sigma} \models_1 T : \hat{T}(\hat{l}) \triangleright \hat{W}_\bullet)$$

$$(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{A}}^{l,n,\kappa} \textbf{read}\,(T)\,@\ell'$$
$$\quad\text{iff}\quad \forall \hat{l} \in \hat{\sigma}(\ell') \colon\; (\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, \hat{l}, \mathbf{r}) \Rightarrow \hat{\sigma} \models_1 T : \hat{T}(\hat{l}) \triangleright \hat{W}_\bullet)$$

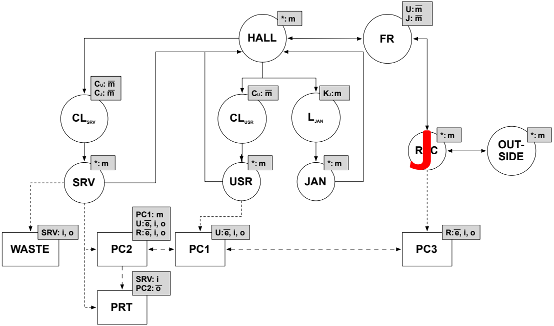
$$(\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{A}}^{l,n,\kappa} \textbf{eval}\,(Q)\,@\ell'$$
$$\quad\text{iff}\quad \forall \hat{l} \in \hat{\sigma}(\ell') : (\langle \mathcal{I}, n, \kappa\rangle \rightsquigarrow (l, \hat{l}, \mathbf{e}) \Rightarrow (\hat{T},\hat{\sigma},\mathcal{I}) \models_{\mathsf{P}}^{\hat{l},n,\kappa} Q)$$

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)

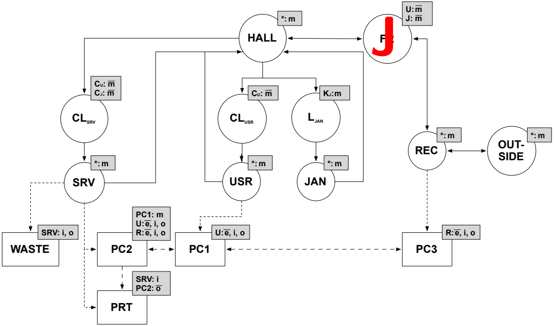


$\mathbf{move}(\mathrm{REC}).\mathbf{move}(\mathrm{FR}).\mathbf{move}(\mathrm{HALL}).\mathbf{move}(\mathrm{CL}_{\mathrm{SRV}}).\mathbf{move}(\mathrm{SRV}).$
$\mathbf{in}(!r)@\mathrm{PRT}.\mathbf{move}(\mathrm{HALL}).\mathbf{move}(\mathrm{FR}).\mathbf{move}(\mathrm{REC})\mathbf{move}(\mathrm{OUTSIDE})$



move(REC).**move**(FR).**move**(HALL).**move**(CL_{SRV}).**move**(SRV).

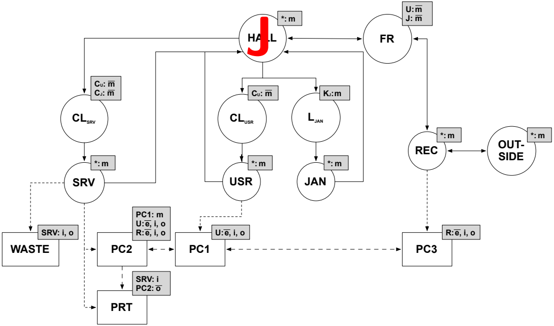
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m)

$\textbf{move}(\mathrm{REC}).\textbf{move}(\mathrm{FR}).\textbf{move}(\mathrm{HALL}).\textbf{move}(\mathrm{CL_{SRV}}).\textbf{move}(\mathrm{SRV}).$

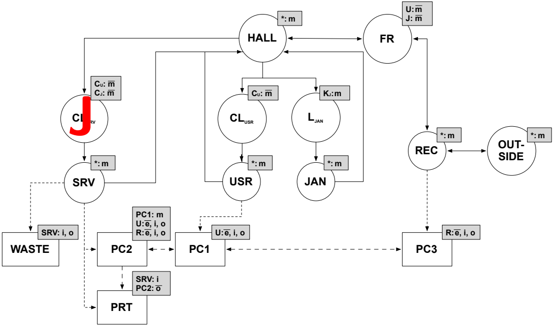
$\textbf{in}(!r)@\mathrm{PRT}.\textbf{move}(\mathrm{HALL}).\textbf{move}(\mathrm{FR}).\textbf{move}(\mathrm{REC})\textbf{move}(\mathrm{OUTSIDE})$



(2,J,FR,m)

move(REC).**move**(FR).**move**(HALL).**move**(CL$_{\text{SRV}}$).**move**(SRV).

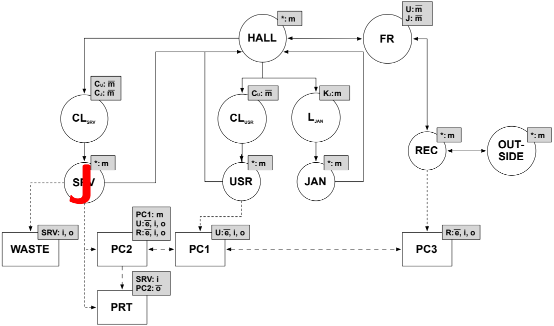
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m) (4,C$_j$,CL$_{\text{SRV}}$,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL}_{\mathrm{SRV}}$).**move**(SRV).

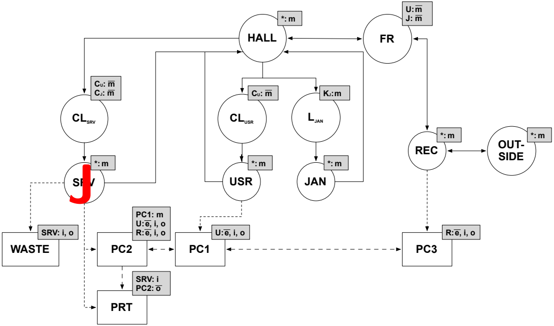
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m) (4,C_j,$\mathrm{CL}_{\mathrm{SRV}}$,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

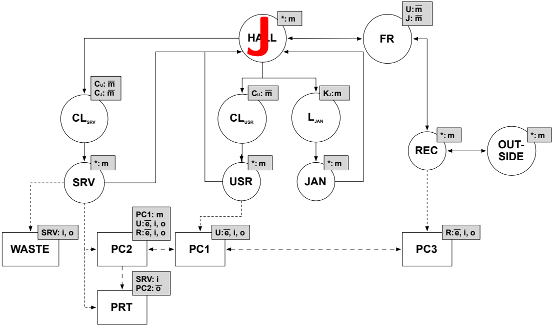
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m) (4,C_j,$\mathrm{CL_{SRV}}$,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

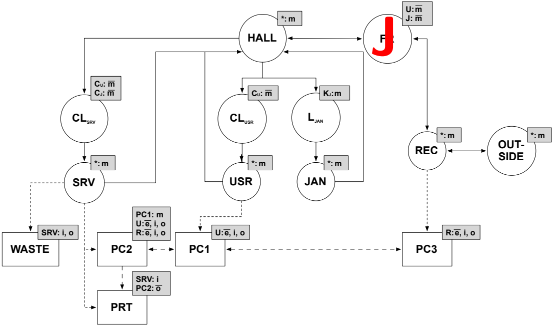
in(!r)@PRT.**move(HALL)**.**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m)   (4,C_j,$\mathrm{CL_{SRV}}$,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

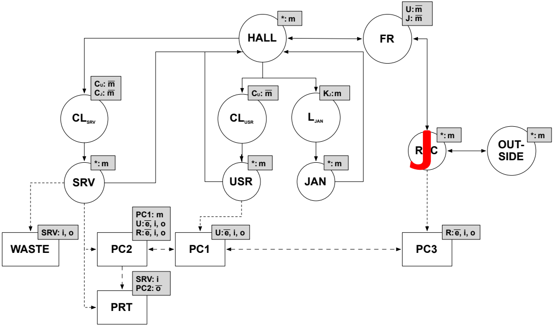
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m)	(4,C_j,$\mathrm{CL_{SRV}}$,m)(7,J,FR,m)

$\mathbf{move}(\mathrm{REC}).\mathbf{move}(\mathrm{FR}).\mathbf{move}(\mathrm{HALL}).\mathbf{move}(\mathrm{CL_{SRV}}).\mathbf{move}(\mathrm{SRV}).$

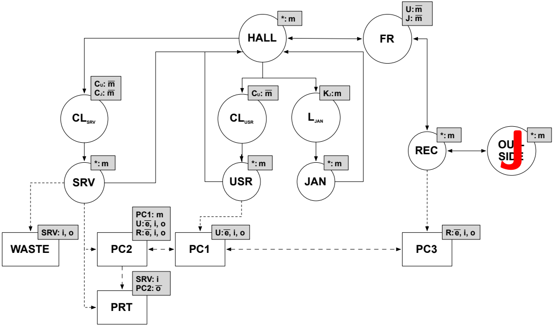
$\mathbf{in}(!r)@\mathrm{PRT}.\mathbf{move}(\mathrm{HALL}).\mathbf{move}(\mathrm{FR}).\mathbf{move}(\mathrm{REC})\mathbf{move}(\mathrm{OUTSIDE})$



(2,J,FR,m) (4,C_j,$\mathrm{CL_{SRV}}$,m)(7,J,FR,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

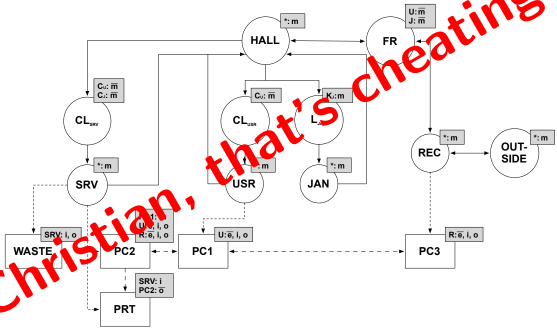
in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move(OUTSIDE)**



(2,J,FR,m) (4,C_j,$\mathrm{CL_{SRV}}$,m)(7,J,FR,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)

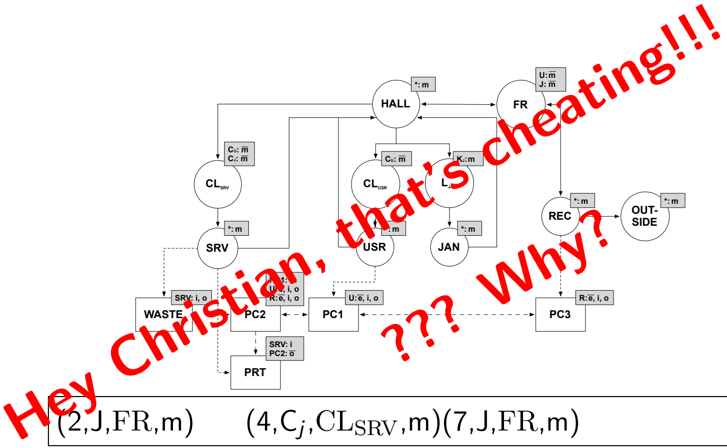


Hey Christian, that's cheating!!!

(2,J,FR,m)　　(4,C_j,$\mathrm{CL_{SRV}}$,m)(7,J,FR,m)

move(REC).**move**(FR).**move**(HALL).**move**($\mathrm{CL_{SRV}}$).**move**(SRV).

in(!r)@PRT.**move**(HALL).**move**(FR).**move**(REC)**move**(OUTSIDE)



(2,J,FR,m)   (4,C_j,$\mathrm{CL_{SRV}}$,m)(7,J,FR,m)

Analysing Processes

That's cheating

- How do we observe $\textbf{in}\,(!r)\,@\mathrm{PRT}$?
- Which actions are performed by processes in the system?

Observables

- "What observables can be obtained at all stages of the process." [RAND Report, 2004]
- Automatically logged actions
- Manually logged after detection

Beyond Analysing Processes

Location/Action equivalence

- What if we do **not** know the processes/actions a priori?

$\Rightarrow$ limited to what we can observe

- Unobservable actions are **equivalent** wrt the system
- Match **observed** actions to **possible** actions
- Especially **log-equivalence**

Alternative Analyses

- Reachability
- Log-Trace Reachability

Reachability

Which places can an Insider reach?

- Assume given actor with given knowledge
- Determine who can access which locations/data

Result

- Are actors able to reach locations that should not be at?
  - Where are additional policies required?
- Which actors reach which locations/data?
  - Who should (be trusted to) retrieve data from a given location?

Log-based Analysis

Which actions cause a logged sequence?

- Assume log sequence as extracted from logging system
- Determine what might have happened unnoticed

Result

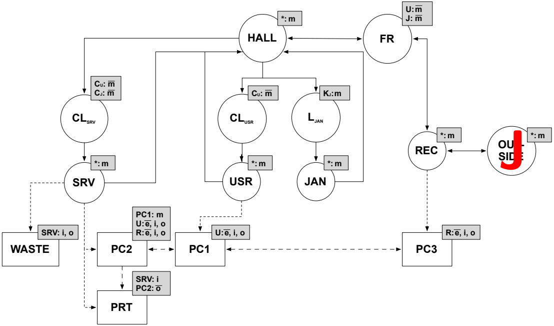
- Which actions could go unnoticed?
  - Where is additional logging required?
- Which actors reach which locations?
  - Who should retrieve data from a given location?

Log-based Analysis

Log-Trace Reachability

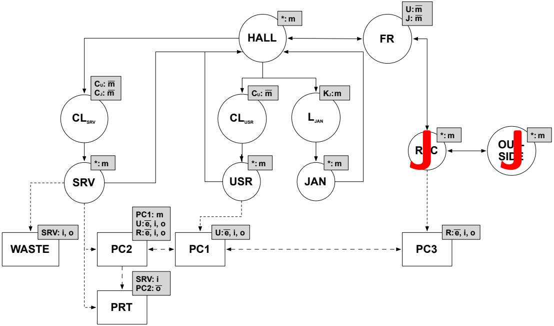
while log sequence not empty **do**
  equivalent()
  pick next (*time*, *reason*, *to*, *action*) from log sequence
  if *reason* is an actor **then**
    actor *reason* must be at a location connected to *to*
    remove that actor from all other locations
  else if *reason* is a key **then**
    possible actors are all actors who might be at *to* and know the key *reason*
    if only one actor who might be at *to* knows the key *reason* **then**
      remove that actor from all other locations
    end if
  else if *reason* is a location **then**
    potential actors are all actors who might access *to*
    if only one actor can access *to* **then**
      remove that actor from all other locations
    end if
  end if
  for all potential actors *n* **do**
    simulate effect of *n* performing action *action*
  end for
end while
equivalent()

Log-based Analysis of the Example



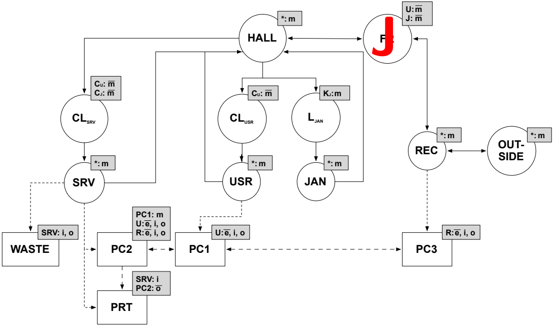
$$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})$$

Log-based Analysis of the Example



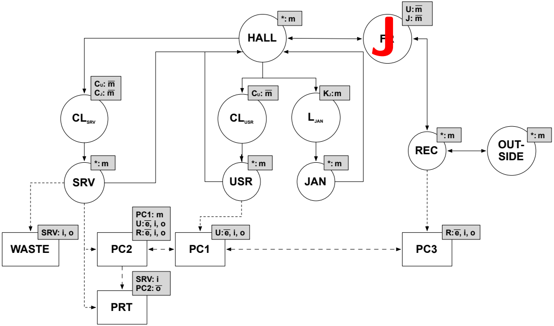
$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})$

Log-based Analysis of the Example



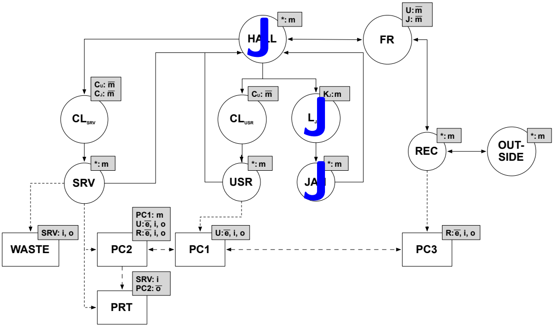
(1,J,$\mathrm{FR}_{\mathrm{ENT}}$,m)(4,$C_j$,$\mathrm{CL}_{\mathrm{SRV}}$,m)(7,J,$\mathrm{FR}_{\mathrm{EXIT}}$,m)

Log-based Analysis of the Example



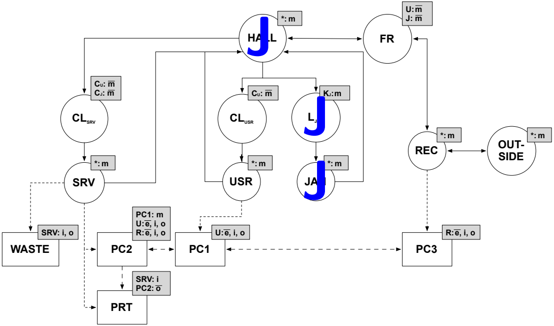
$(1, J, \mathrm{FR}_{\mathrm{ENT}}, m)(4, C_j, \mathrm{CL}_{\mathrm{SRV}}, m)(7, J, \mathrm{FR}_{\mathrm{EXIT}}, m)$

Log-based Analysis of the Example



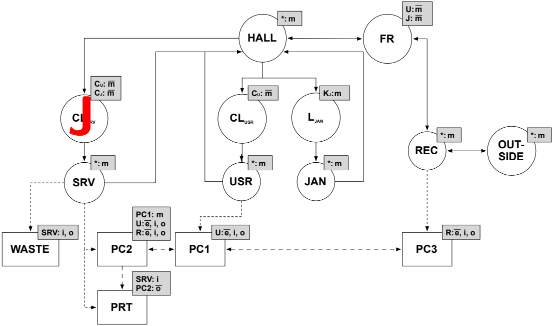
$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})$

Log-based Analysis of the Example



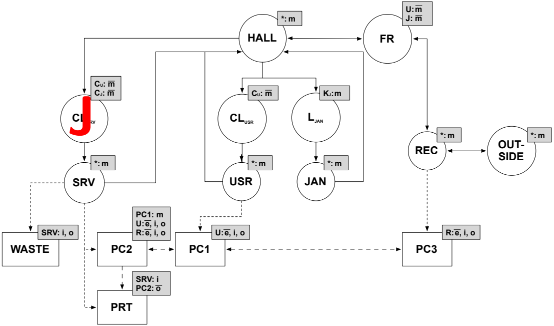
$(1,\mathrm{J},\mathrm{FR_{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL_{SRV}},\mathrm{m})(7,\mathrm{J},\mathrm{FR_{EXIT}},\mathrm{m})$

Log-based Analysis of the Example



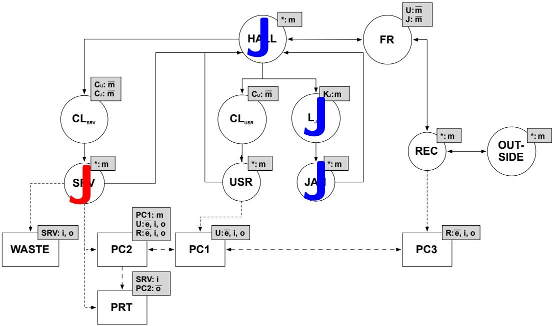
$$(1,J,\mathrm{FR}_{\mathrm{ENT}},m)(4,C_j,\mathrm{CL}_{\mathrm{SRV}},m)(7,J,\mathrm{FR}_{\mathrm{EXIT}},m)$$

Log-based Analysis of the Example



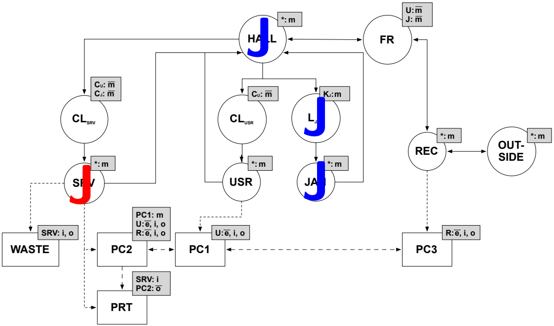
$$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})\color{blue}{(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})}\color{black}{(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})}$$

Log-based Analysis of the Example



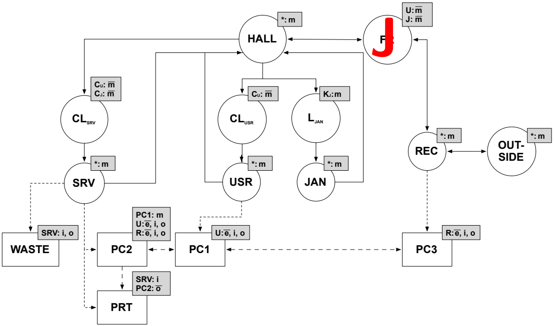
$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})$

Log-based Analysis of the Example



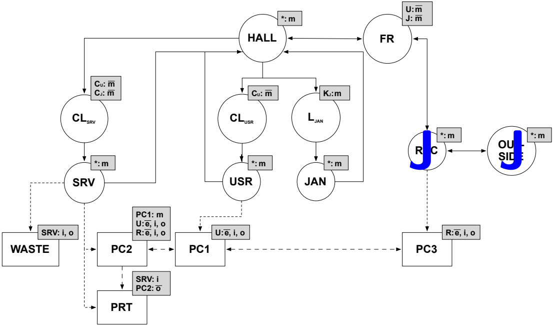
$(1,J,FR_{ENT},m)(4,C_j,CL_{SRV},m)(7,J,FR_{EXIT},m)$

Log-based Analysis of the Example



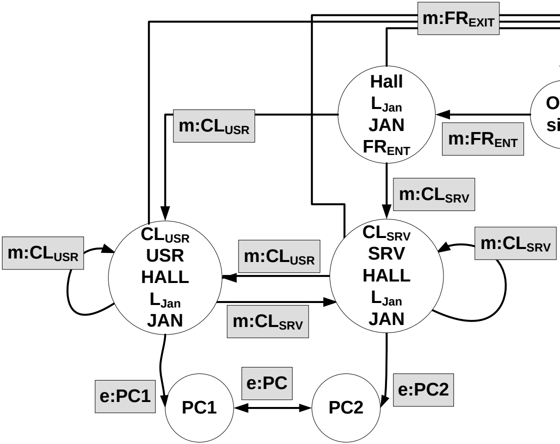
$(1,J,\mathrm{FR}_{\mathrm{ENT}},m)(4,C_j,\mathrm{CL}_{\mathrm{SRV}},m)(7,J,\mathrm{FR}_{\mathrm{EXIT}},m)$

Log-based Analysis of the Example



$$(1,\mathrm{J},\mathrm{FR}_{\mathrm{ENT}},\mathrm{m})(4,\mathrm{C}_j,\mathrm{CL}_{\mathrm{SRV}},\mathrm{m})\color{blue}{(7,\mathrm{J},\mathrm{FR}_{\mathrm{EXIT}},\mathrm{m})}$$

Online Log Surveillance



Required Credentials

What is needed to get from A to B?

- Based on access control restrictions
- Determine who can access which locations/data

Result

- Set of credentials required to reach a certain location
  - Who had possibility to perform a certain action?
  - Can be combined with results of other analyses
- Which actors can reach which locations/data?
  - Who should (be trusted to) retrieve data from a given location?
  - Which credentials does the person in charge lack?

Required Credentials

extractCredentials

```
/* return the credentials needed to move to node n */
```

credentials $= \emptyset$
acl $=$ access control list at n
for all restrictions $(factor, actions) \in acl$ **do**
    for all actions a in *actions* **do**
        if a is *move* **then**
            if *factor* is an actor or a key **then**
                $credentials = credentials \cup \{factor\}$
            end if
        end if
    end for
end for

Required Credentials

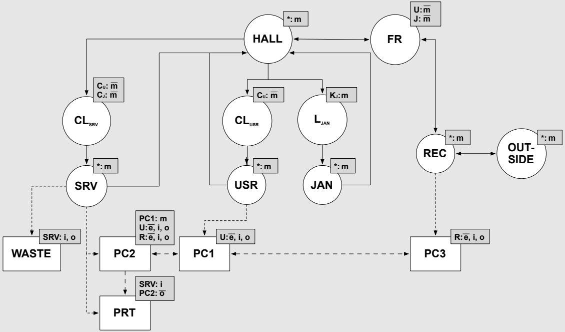
/* return the credentials needed to get from *from* to *to* */

for all paths $p = n_1, \ldots, n_k$ from *from* to *to* **do**
  $IDs_p = AllIDs$
  $result_p = \emptyset^k$
  for all nodes $n_i = n_2, \ldots, n_k$ **do**
    $result_{n_i} = extractCredentials(n_i) \setminus (AllIDs \setminus IDs_p)$
    $IDs_p = IDs_p \cap result_{n_i}$
    if $result_{n_i} = \emptyset$ **then**
      $result_{p,i} = \perp$; continue
    else
      $result_{p,i} = result_{n_i} \setminus \bigcup_{1<j<=i} result_{p,j}$
    end if
  end for
end for
return the computed tuples $result_p$ and IDs_p

Behaviour

What is missing?

Example System



What is missing?

Shortcomings

- Analyses over-approximate
  - All performable actions are performed
  - Needed to guarantee correctness of the result
- Whenever an actor can loose something...
- ...he will do so

What is missing?

Adding behaviour

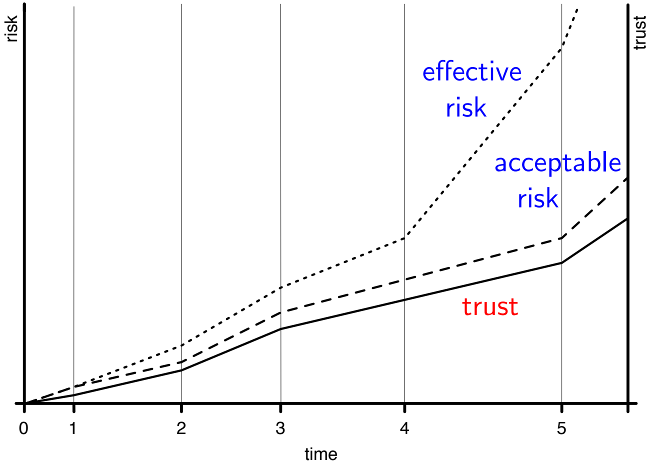
- What the analyses should do is
  - Annotate items with a probability of a certain action being performed on them
  - This may depend on many factors
    - Time of day, mood, item, location, ...
- Compute probabilities that a certain data item is lost, or that a certain action can be performed

Limitations

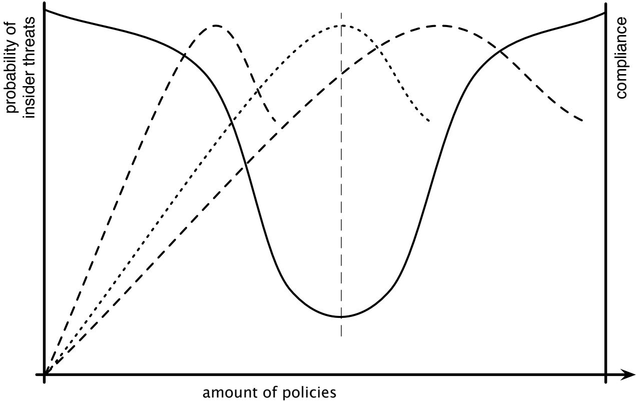
Kind of Threats

- Precision of analysis results depends on access rights of actors
- What about CEOs, cleaning personal?
  - Either full access to locations, or full access to everything
  - Handling data exchange results in 100% insider threat vulnerability
- High-level insiders really pose a risk that we can not deal with
- Or insiders that acquire more and more knowledge about the system and its workings
  - On the other hand, this learning is what we want or expect

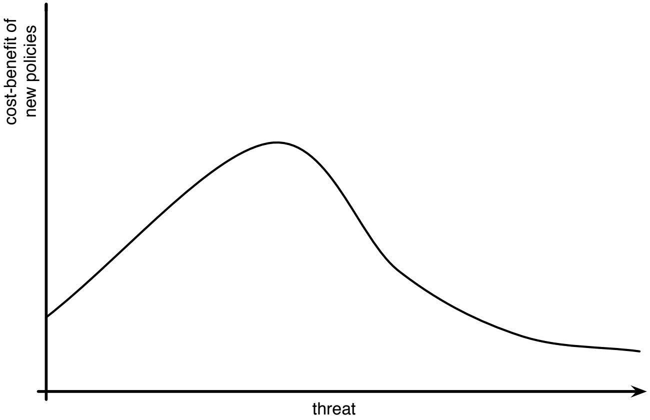
Trust vs. Risk



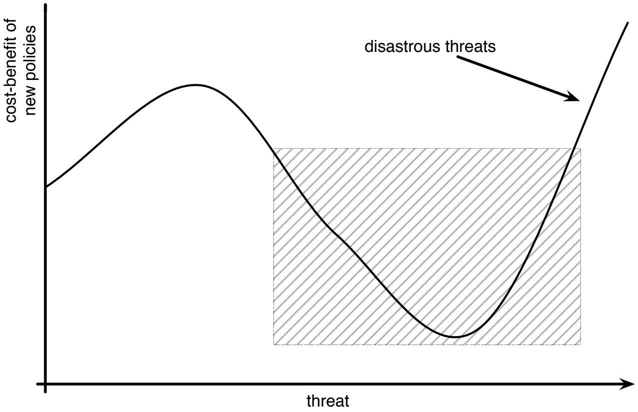
Insider Threats, Compliance, and Policies



Cost Benefit of Policies against Threats



Cost Benefit of Policies against Threats



Conclusions

Our Approach

- System model
  - Captures relevant system properties, easily extendable
  - Abstract model generated from system models
- Analyses
  - System design and Audit, not insider-specific
- Limitations when dealing with high-level threats

Conclusions

Future/Current Work

- Add “soft” properties
    - Trust, probability to loose/reveal data
- Link to model checking, integrate log-trace analysis into Flow Logic analysis
- Generate and rank attacks from system models

Conclusions

Partners in Crime

- René Rydhof Hansen (Aalborg University), Flemming Nielson
- Pieter Hartel, Wolter Pieters, Trajce Dimkov (Universiteit Twente)
- Matt Bishop (University of California, Davis), Jeffrey Hunker (Carnegie Mellon University), Dieter Gollmann (TU Hamburg-Harburg), Lizzie Coles-Kemp (Royal Holloway)

Conclusions

Publications

- Christian W. Probst, Jeffrey Hunker, Dieter Gollmann, Matt Bishop (editors): *Insider Threats in Cybersecurity*, Springer, 2010.
- Christian W. Probst, René Rydhof Hansen: *An Extensible Analysable System Model*. In Information Security Technical Report, Elsevier.
- Christian W. Probst and Jeffrey Hunker: *The Risk of Risk Analysis, and its relation to the Economics of Insider Threats*. In Proceedings of The Eighth Workshop on the Economics of Information Security (WEIS 2009).

Conclusions

Publications

- Christian W. Probst, René Rydhof Hansen: *Fluid Information Systems*. In Proceedings of the New Security Paradigms Workshop (NSPW 2009).
- Christian W. Probst, René Rydhof Hansen: *Analysing Access Control Specifications*. In Proceedings of the Fourth International IEEE Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE 2009).
- Christian W. Probst, René Rydhof Hansen, Flemming Nielson: *Where can an Insider attack?* In Proceedings of The Fourth Workshop on Formal Aspects in Security and Trust (FAST 2006).