Hybrid Symbiotic Simulation and Security Compliance Tracking for Decision Support in Cooperating Cyber-Physical Systems

Roland Rieke

`http://rieke.link`

St. Petersburg, November 2015

Fraunhofer Institute for Secure Information Technology SIT

Fraunhofer

- Europe's largest nonprofit organization (24 000 employees) for applied research
- 66 institutes and research institutions across Germany
- Institutional funding (30%), contract research for industry and government (70%)

Fraunhofer SIT (Darmstadt+Birlinghoven)

- Largest IT Security Research Center in Germany (163 Employees, 78 scientific)
- Directorate: Prof. Dr. Michael Waidner `www.sit.fraunhofer.de`

Department Cyber-Physical Systems – (Head: Dr. Christoph Krauß)

Automotive Security: Car2X-Security, ECU and EE-System Security, HSM analysis

Smart Grid / Smart Metering Security: Secure Smart Meter (Gateways)

Trustworthy Platforms: Embedded OS, Microkernel, Virtualization, TSS

Mobile Services: Smartphone-based Access Control, Mobile Payment

(Formal) Security Analyses, Tests: Penetration Tests, Side Channel and Fault Attacks

Overview

- Introduction
- Cooperating Cyber-Physical Systems
- Symbiotic Simulation
- Behavior Conformance Tracking
- Security Compliance Tracking
- Security Strategy Management
- Conclusion

The first question in any scientific research is its subject matter: What are we studying ? The most general answer is a certain kind of system.
— Sunny Y. Auyang

Introduction

Information Security

Information security

“*The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.* ”

NIST, 2013 (similar: ISO27001, rfc4949)

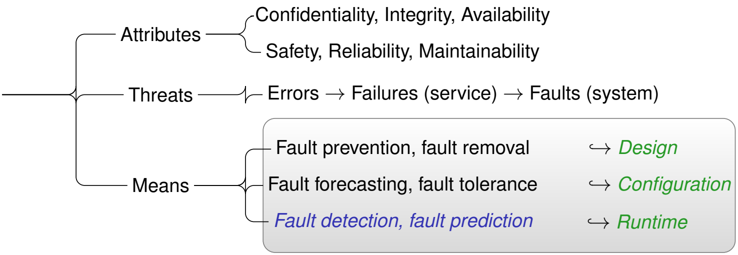
Information Security

Information security

"*The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability.* "

NIST, 2013 (similar: ISO27001, rfc4949)

Dependability and security tree



Attributes

Confidentiality, Integrity, Availability

Safety, Reliability, Maintainability

Threats

Errors → Failures (service) → Faults (system)

Means

Fault prevention, fault removal ↪ *Design*

Fault forecasting, fault tolerance ↪ *Configuration*

Fault detection, fault prediction ↪ *Runtime*

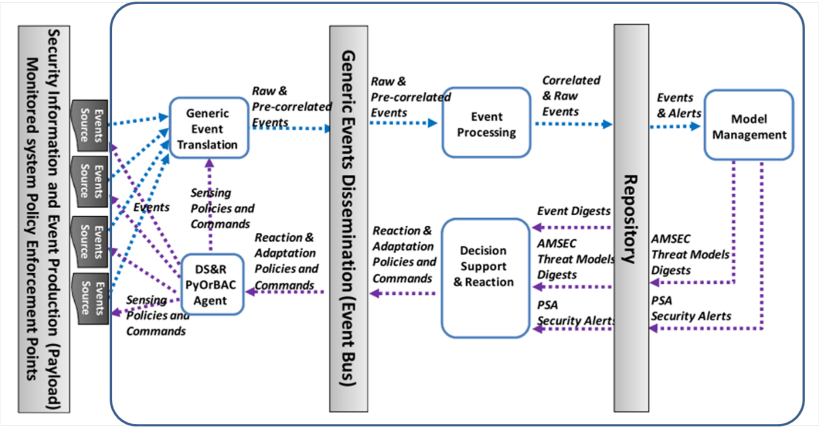
based on Avizienis, Laprie, Randell, and Landwehr, 2004 (scholar 3289)
" Basic Concepts and Taxonomy of Dependable and Secure Computing"

Motivation: Security @ runtime

> “*Even when applying current best practices and technologies to secure software and IT landscapes,*
> *it would be inappropriate to assume that there are no remaining vulnerabilities and that there will be no attempts to exploit them.*
>
> *Hence, complementing security technology and management with*
> *means to detect and monitor vulnerabilities and attacks*
> *is an essential element in a comprehensive security strategy.*”

Volkmar Lotz, ARES 2014

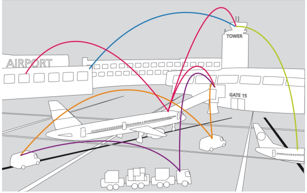
Background: The MASSIF distributed SIEM approach



Security Information and Event Production (Payload)
Monitored system Policy Enforcement Points
Events Source
Events Source
Events Source
Events Source
Events
Generic Event Translation
Raw & Pre-correlated Events
Sensing Policies and Commands
DS&R PyOrBAC Agent
Reaction & Adaptation Policies and Commands
Sensing Policies and Commands
Generic Events Dissemination (Event Bus)
Raw & Pre-correlated Events
Event Processing
Correlated & Raw Events
Reaction & Adaptation Policies and Commands
Decision Support & Reaction
Event Digests
AMSEC Threat Models Digests
PSA Security Alerts
Repository
Events & Alerts
Model Management
AMSEC Threat Models Digests
PSA Security Alerts

Source: MASSIF project `http://www.massif-project.eu/`

Cooperating Cyber-Physical Systems



Introduction

Topic 1: Cooperating Cyber-Physical Systems

- Security of Cooperating Cyber-Physical Systems
- CPS Interfaces

Topic 2: Symbiotic Simulation

Topic 3: Behavior Conformance Tracking

Topic 4: Security Compliance Tracking

Topic 5: Security Strategy Management

TCONC

Security of Cooperating CPS

Cooperating Cyber-Physical Systems (CPS) are systems of systems that collaborate for a common purpose.

- Systems in the physical world are linked to the cyber world by elements such as sensors, which capture data from the physical world that provides an abstraction of the state of the physical world for processing in the cyber world.
- Analysis of this information may lead to decisions in the cyber world.
- These, in turn, influence the physical world either directly, e.g., by actuator elements, or indirectly, e.g., by visualizing information for human actuators.

Security of Cooperating CPS

Cooperating Cyber-Physical Systems (CPS) are systems of systems that collaborate for a common purpose.

- Systems in the physical world are linked to the cyber world by elements such as sensors, which capture data from the physical world that provides an abstraction of the state of the physical world for processing in the cyber world.
- Analysis of this information may lead to decisions in the cyber world.
- These, in turn, influence the physical world either directly, e.g., by actuator elements, or indirectly, e.g., by visualizing information for human actuators.

> "*Vehicular ad hoc networks will enable vehicles to act autonomously . . .*
> *this technology presents major challenges in the secure design of the involved systems and protocols.*"
>
> Gerlach, 2005

> "*Future distributed air traffic management systems need to . . .*
> *collaborate for a common purpose (e.g. smooth running of an airport) . . .*
> *ensure continual update and improvement to security.*"
>
> Hawley et al, 2013

These systems must not only be secure, they must be *demonstrably* so.

Cyber-physical interface

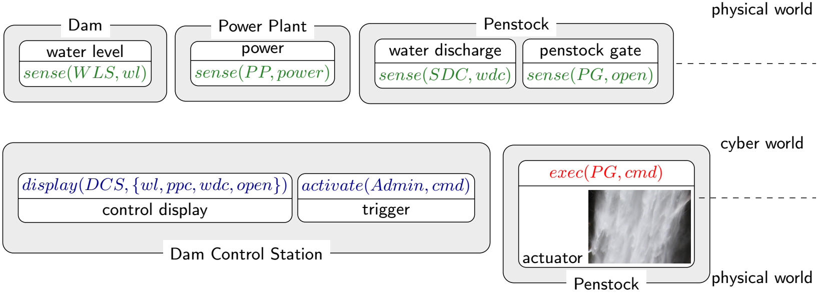
Sensors

- physical (e.g. temperature, no. people in control room)
- cyber (e.g. intrusion detection, OS system call, login uid/role)

Cyber Observable eXpression: standard for cyber events/properties.

Actuators

- physical (e.g. open gate, traffic light, spinning reserve)
- cyber (e.g. firewall policy, resources: amount of available money, power, bandwidth)



physical world

Dam

water level

$sense(WLS, wl)$

Power Plant

power

$sense(PP, power)$

Penstock

water discharge

$sense(SDC, wdc)$

penstock gate

$sense(PG, open)$

cyber world

$display(DCS, \{wl, ppc, wdc, open\})$

control display

$activate(Admin, cmd)$

trigger

Dam Control Station

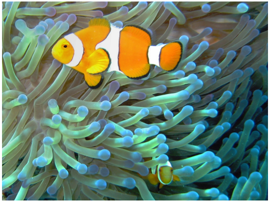
$exec(PG, cmd)$

actuator

Penstock

physical world

Symbiotic Simulation



Introduction

Topic 1: Cooperating Cyber-Physical Systems

Topic 2: Symbiotic Simulation

- Introduction to Symbiotic Simulation
- Trustworthy CPS interfaces
- What-if - Security of system configurations
- Simulation results analysis

Topic 3: Behavior Conformance Tracking

Topic 4: Security Compliance Tracking

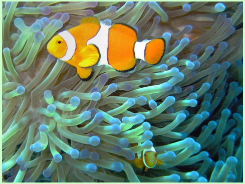
Topic 5: Security Strategy Management

Conclusion

Symbiotic Simulation Systems

Symbiosis (biol.)

- In biology, symbiosis is a persistent (mutualistic) interaction between different biological species.
- Symbiosis is also involved in interdependent co-evolution of species.



Symbiotic Simulation

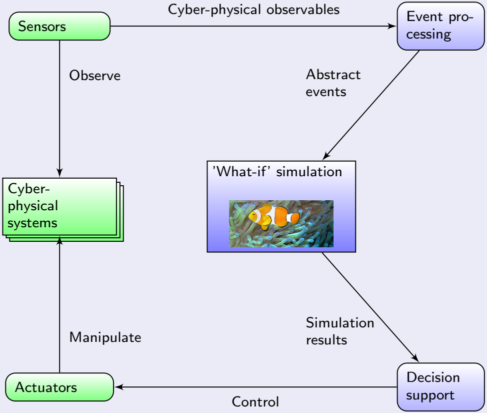
- a continuous on-line simulation interacts in real-time with a physical system in a mutually beneficial way
- the physical system benefits from optimization obtained from the analysis of simulation experiments
- the simulation benefits from the validation of its simulation outputs

Richard Fujimoto et al.
Grand Challenges for Modeling and Simulation. In *Dagstuhl Seminar 02351*, 2002.

Heiko Aydt et al.
Symbiotic Simulation Systems: An Extended Definition Motivated by Symbiosis in Biology. In *22nd PADS Workshop*, 2008.

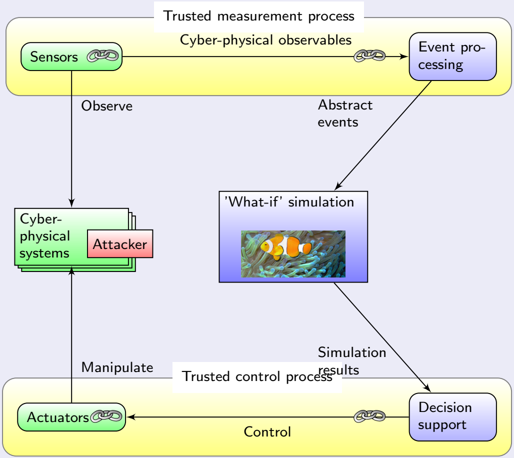
Symbiotic Simulation of CPS

Continuous on-line simulation interacts in real-time with a CPS



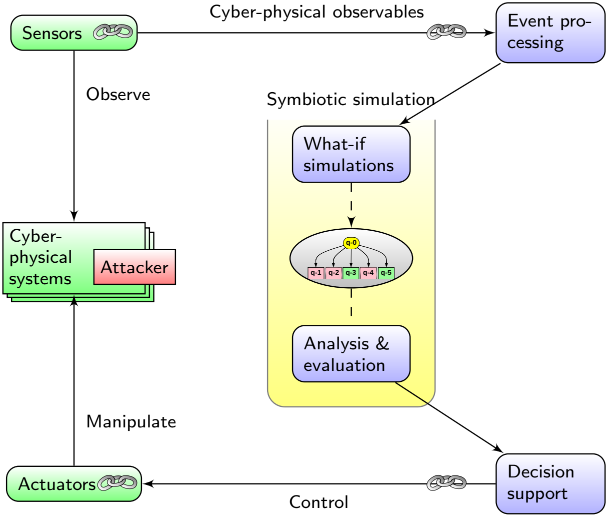
Trustworthy CPS interfaces

Integrity of measurement system needed for sensor value authenticity



Andre Rein, Roland Rieke, Michael Jäger, Nicolai Kuntze, and Luigi Coppolino,
Trust Establishment in Cooperating Cyber-Physical Systems,
The first Conference on Cybersecurity of Industrial Control Systems (CyberICS 2015)

Symbiotic Simulation Architecture



Sensors

Cyber-physical observables

Event processing

Observe

Symbiotic simulation

What-if simulations

Analysis & evaluation

Cyber-physical systems

Attacker

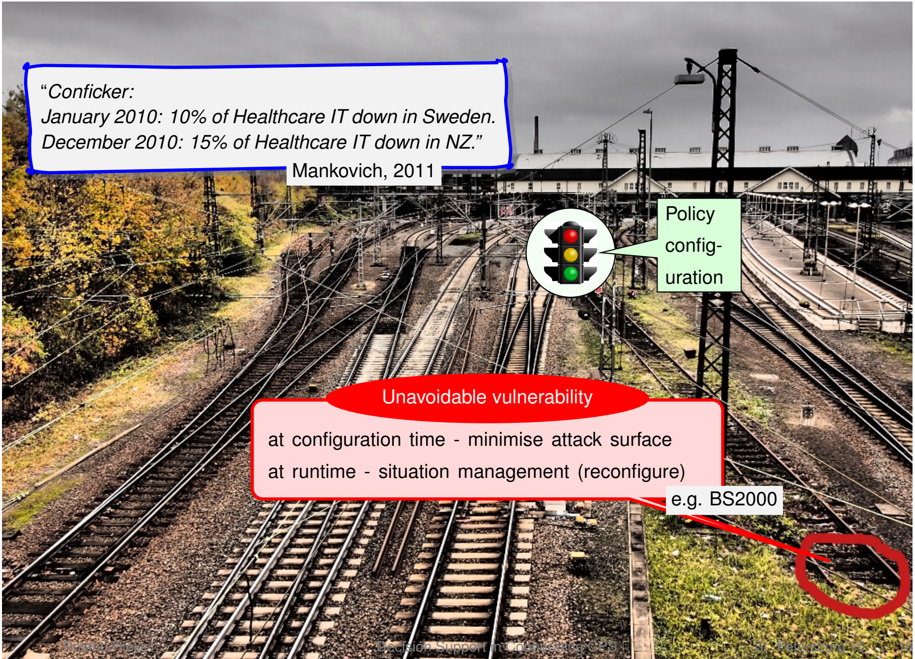
Manipulate

Actuators

Control

Decision support

What-if – Safety & Security of CPS Configurations



“Conficker:
January 2010: 10% of Healthcare IT down in Sweden.
December 2010: 15% of Healthcare IT down in NZ.”

Mankovich, 2011

Policy configuration

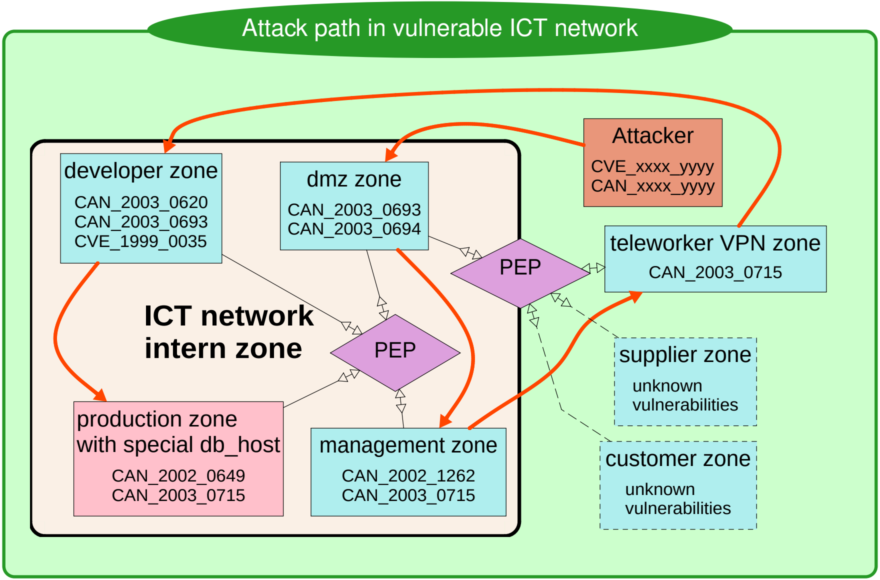
Unavoidable vulnerability

at configuration time - minimise attack surface

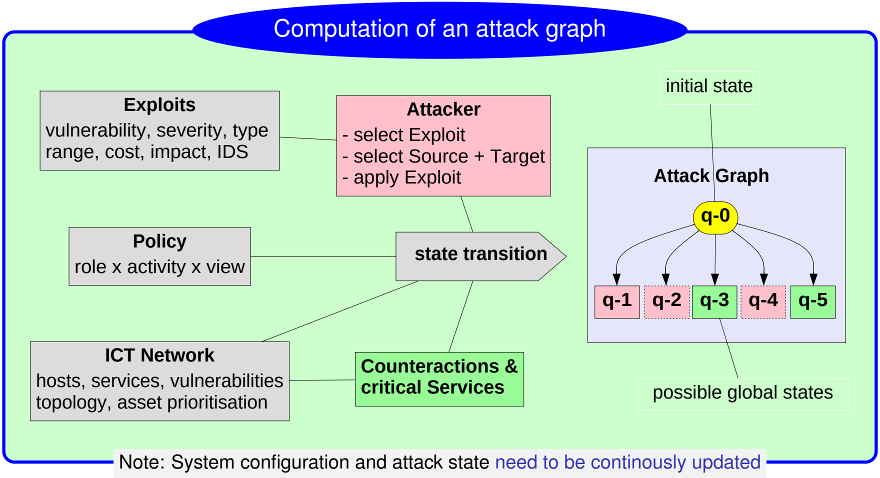
at runtime - situation management (reconfigure)

e.g. BS2000

What-if – Simulation of Attack Paths



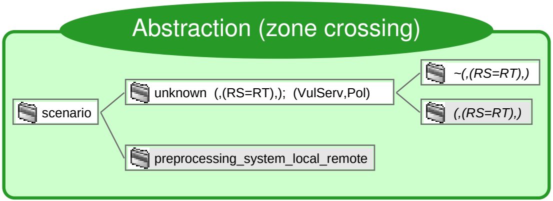
Simulation Model of Relevant Components of the CPS



Note: System configuration and attack state need to be continously updated

Igor Kotenko and Andrey Chechulin,
Attack Modeling and Security Evaluation in SIEM Systems,
International Transactions on Systems Science and Applications, SIWN Press 2012.

Analysis & Evaluation of Simulation Results



Abstraction (zone crossing)

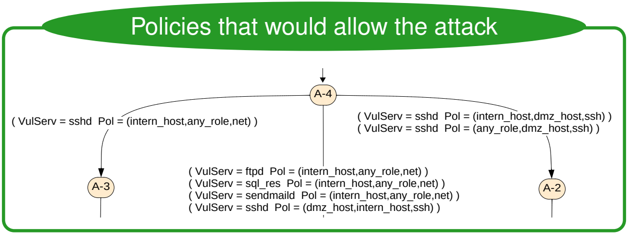
scenario

unknown (,(RS=RT),); (VulServ,Pol)

~(,(RS=RT),)

(,(RS=RT),)

preprocessing_system_local_remote



Policies that would allow the attack

A-4

(VulServ = sshd Pol = (intern_host,any_role,net))

(VulServ = sshd Pol = (intern_host,dmz_host,ssh))
(VulServ = sshd Pol = (any_role,dmz_host,ssh))

A-3

(VulServ = ftpd Pol = (intern_host,any_role,net))
(VulServ = sql_res Pol = (intern_host,any_role,net))
(VulServ = sendmaild Pol = (intern_host,any_role,net))
(VulServ = sshd Pol = (dmz_host,intern_host,ssh))

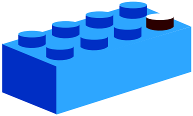
A-2

Roland Rieke,

Abstraction-based analysis of known and unknown vulnerabilities of critical information infrastructures,

International Journal of System of Systems Engineering (IJSSE), InderScience 2008.

Behavior Conformance Tracking



Introduction

Topic 1: Cooperating Cyber-Physical Systems

Topic 2: Symbiotic Simulation

Topic 3: Behavior Conformance Tracking

- Introduction to Behavior Conformance Tracking
- Event model & event processing
- Behavior anomaly detection
- Conformance tracking algorithm

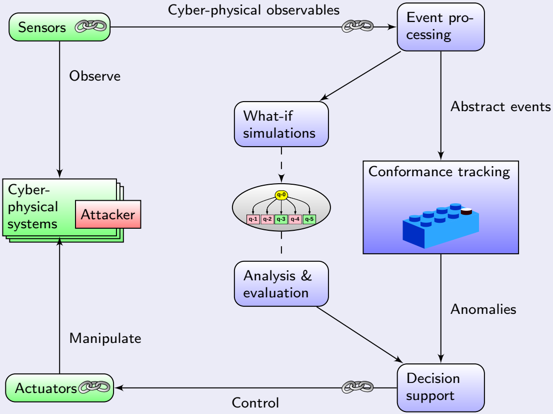
Topic 4: Security Compliance Tracking

Topic 5: Security Strategy Management

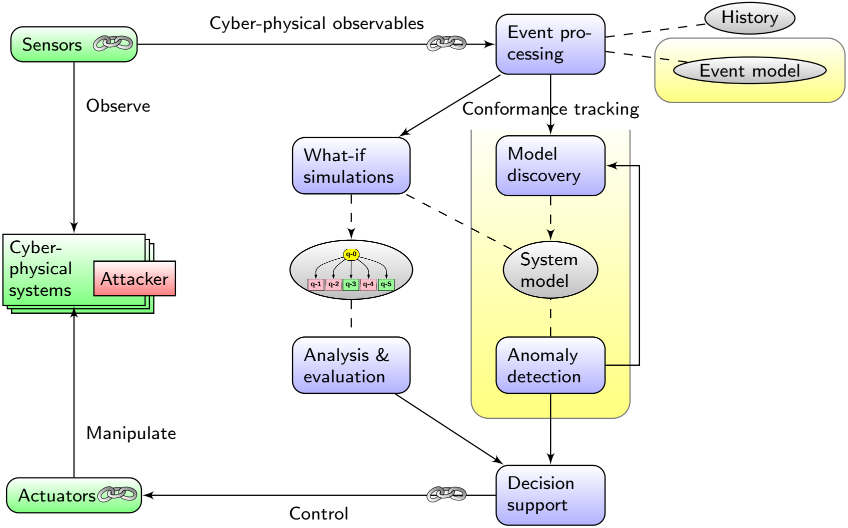
Conclusion

Conformance Tracking

Conformance tracking is the capability to detect deviations of observed events from expected events in the current state



Conformance Tracking Architecture



Sensors

Cyber-physical observables

Event processing

History

Event model

Observe

Conformance tracking

What-if simulations

Model discovery

Cyber-physical systems

Attacker

System model

Analysis & evaluation

Anomaly detection

Manipulate

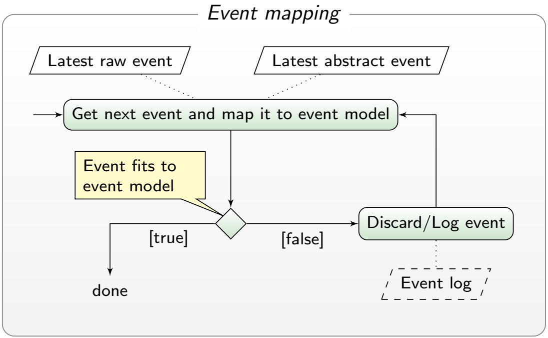
Actuators

Decision support

Control

Event Model & Event Processing

Formally, it is assumed that an event represents a letter of the alphabet that denotes the possible actions in the observed, simulated system.



Event mapping

Latest raw event

Latest abstract event

Get next event and map it to event model

Event fits to event model

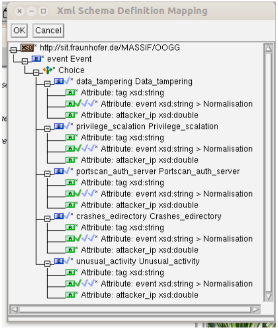
[true]

[false]

Discard/Log event

done

Event log



```
* http://sit.fraunhofer.de/MASSIF/OOGG
  * event Event
    * Choice
      * data_tampering Data_tampering
        * Attribute: tag xsd:string
        * Attribute: event xsd:string > Normalisation
        * Attribute: attacker_ip xsd:double
      * privilege_scalation Privilege_scalation
        * Attribute: tag xsd:string
        * Attribute: event xsd:string > Normalisation
        * Attribute: attacker_ip xsd:double
      * portscan_auth_server Portscan_auth_server
        * Attribute: tag xsd:string
        * Attribute: event xsd:string > Normalisation
        * Attribute: attacker_ip xsd:double
      * crashes_edirectory Crashes_edirectory
        * Attribute: tag xsd:string
        * Attribute: event xsd:string > Normalisation
        * Attribute: attacker_ip xsd:double
      * unusual_activity Unusual_activity
        * Attribute: tag xsd:string
        * Attribute: event xsd:string > Normalisation
        * Attribute: attacker_ip xsd:double
```

In order to avoid state space explosion problems, the coarsest abstraction *h* that still contains all security relevant information should be used.

Complexity Reduction by Process Instance Projection

- Mendling: Metrics for Process Models: Average 3.56 connections

Definition (process instance projection)

Let P denote a finite set of process instances i of some process with $i \in P$ and let Σ_i denote pairwise disjoint copies of Σ. The elements of Σ_i are denoted by e_i and $\Sigma_P := \bigcup_{i \in P} \Sigma_i$. The index i describes the bijection $e \leftrightarrow e_i$ for $e \in \Sigma$ and $e_i \in \Sigma_i$. Now the projection π identifies events from a specific process instance i.

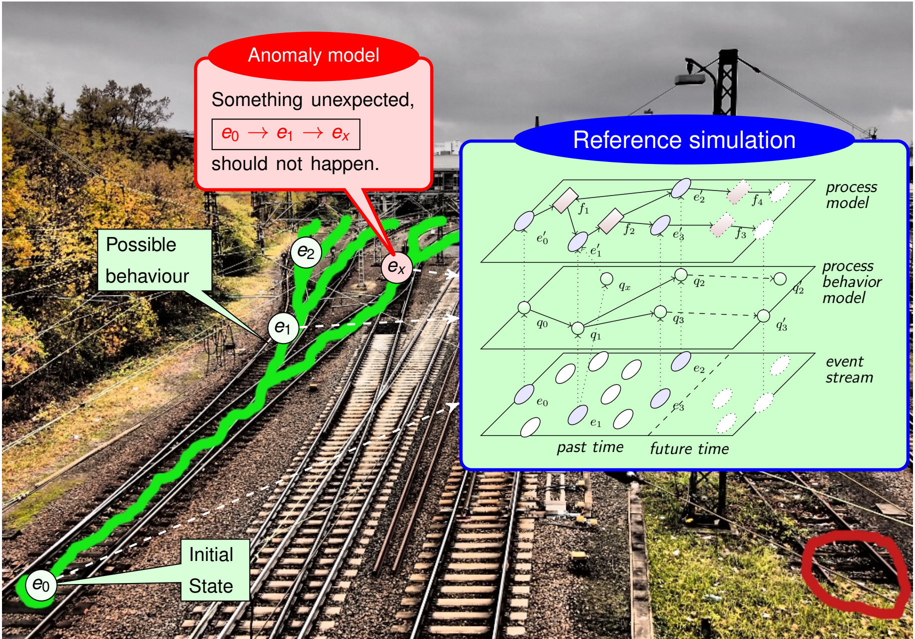
For $i \in P$, let $\pi_i^P : \Sigma_P^* \to \Sigma^*$ with $\pi_i^P(e_r) = \begin{cases} e \mid & e_r \in \Sigma_i \\ \varepsilon \mid & e_r \in \Sigma_P \setminus \Sigma_i \end{cases}$.

Note: In process-unaware systems, the assumption about pairwise disjoint alphabets is not always valid. Sometimes, a set of attributes identifies the process instance.

Wil M. P. van der Aalst
Process Mining: Discovery, Conformance and Enhancement of Business Processes
Springer 2011.

Jan Mendling
Metrics for Process Models
Springer LNBIP Vol. 6, 2008.

Behaviour anomaly detection



Anomaly model

Something unexpected,

$e_0 \to e_1 \to e_x$

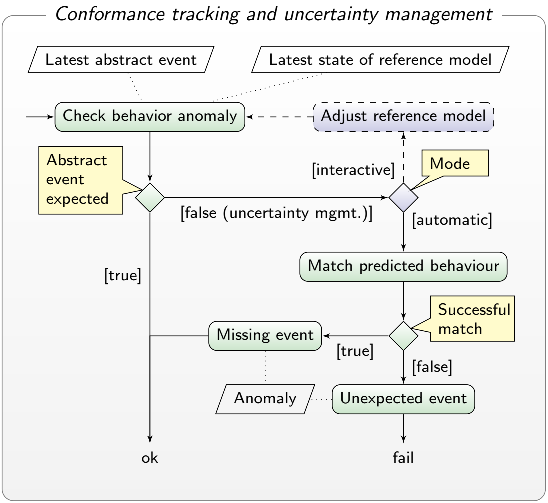
should not happen.

Reference simulation

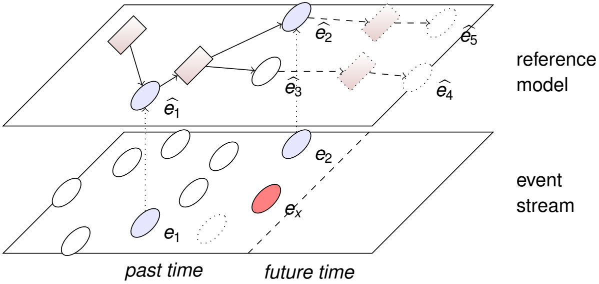
Possible behaviour

Initial State

Conformance Tracking Algorithm

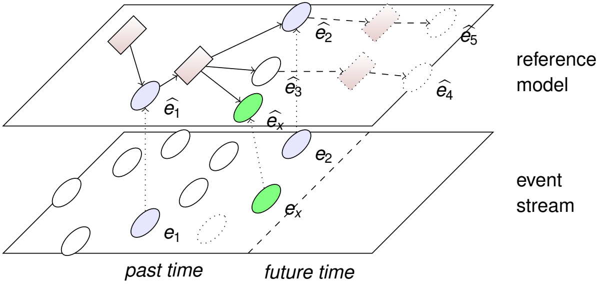


Model Evolution – Unknown Events



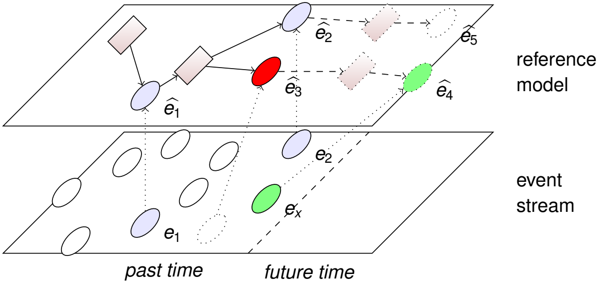
- event e_x is received (and fits to the event model)
- it is not part of the model behavior (in scope of the analysis)

Model Evolution – Unknown Events



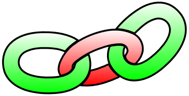
- event e_x is received (and fits to the event model)
- it is not part of the model behavior (in scope of the analysis)
  - possible adjustment: user inserts a new abstract event $\widehat{e_x} = h(\pi_i^P(e_x))$ to the reference model along with a connecting edge

Model Evolution – Unknown Events



- event e_x is received (and fits to the event model)
- it is not part of the model behavior (in scope of the analysis)
  - possible adjustment: user inserts a new abstract event $\widehat{e_x} = h(\pi_i^P(e_x))$ to the reference model along with a connecting edge
  - match predicted behavior: an abstract event $\widehat{e_4} = h(\pi_i^P(e_x))$ is part of a possible continuation where e with $\widehat{e_3} = h(\pi_i^P(e))$ has been missed

Security Compliance Tracking



Introduction

Topic 1: Cooperating Cyber-Physical Systems

Topic 2: Symbiotic Simulation

Topic 3: Behavior Conformance Tracking

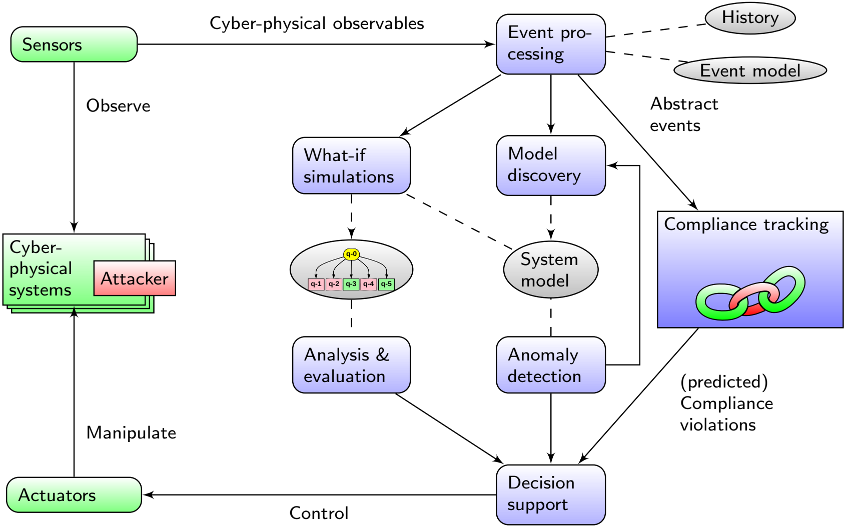
Topic 4: Security Compliance Tracking

- Introduction to Security Compliance Tracking
- Prediction of close future critical states
- Example for security reasoning at runtime
- Security requirements elicitation

Topic 5: Security Strategy Management

Conclusion

Compliance Tracking



Sensors

Cyber-physical observables

Event processing

History

Event model

Observe

What-if simulations

Model discovery

Abstract events

Compliance tracking

Cyber-physical systems

Attacker

System model

Analysis & evaluation

Anomaly detection

(predicted) Compliance violations

Manipulate

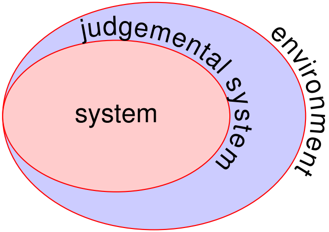
Actuators

Decision support

Control

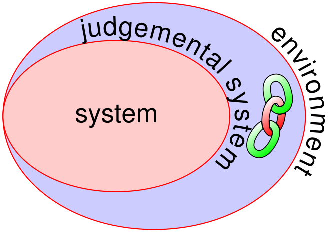
Security compliance

> “*Systems Come in Threes!*
> *. . . a judgemental system, is involved in determining whether any particular activity (or inactivity) of a system in a given environment constitutes or would constitute - from its viewpoint - a failure.*”
>
> Brian Randell, IFIP WG 10.4, 2007



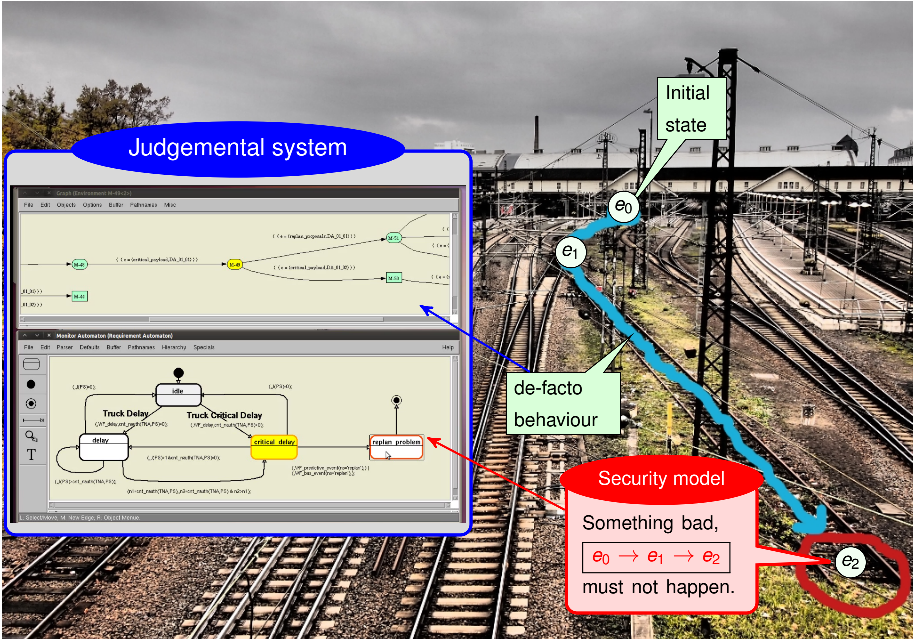
Security compliance

> "*Systems Come in Threes!*
> *. . . a judgemental system, is involved in determining whether any particular activity (or inactivity) of a system in a given environment constitutes or would constitute - from its viewpoint - a failure.*"
>
> Brian Randell, IFIP WG 10.4, 2007



Security compliance tracking is the capability to apply a security model at runtime in order to identify violations of security requirements.

Security & safety compliance check at runtime


Graph (Environment M-49<2>)
File Edit Objects Options Buffer Pathnames Misc
⟨ ⟨ e = ⟨replan_proposals,DA_01_01⟩ ⟩ ⟩
⟨ ⟨ e = ⟨critical_payload,DA_01_01⟩ ⟩ ⟩
⟨ ⟨ e = ⟨critical_payload,DA_01_02⟩ ⟩ ⟩
_01_01⟩ ⟩ ⟩
_01_02⟩ ⟩ ⟩
M-48
M-49
M-51
M-50
M-44
Monitor Automaton (Requirement Automaton)
File Edit Parser Defaults Buffer Pathnames Hierarchy Specials Help
idle
delay
critical delay
replan problem
Truck Delay
Truck Critical Delay
(,,I(PS)=0);
(,,I(PS)=0);
(,WF_delay,cnt_nauth(TNA,PS)=0);
(,WF_delay,cnt_nauth(TNA,PS)>0);
(,,I(PS)>1&cnt_nauth(TNA,PS)=0);
(,,I(PS)>cnt_nauth(TNA,PS));
(n1=cnt_nauth(TNA,PS),,n2=cnt_nauth(TNA,PS) & n2>n1);
(,WF_predictive_event(ns='replan'),) |
(,WF_bus_event(ns='replan'),);
L: Select/Move; M: New Edge; R: Object Menue.
e_0
e_1
e_2

Judgemental system

Initial state

de-facto behaviour

Security model

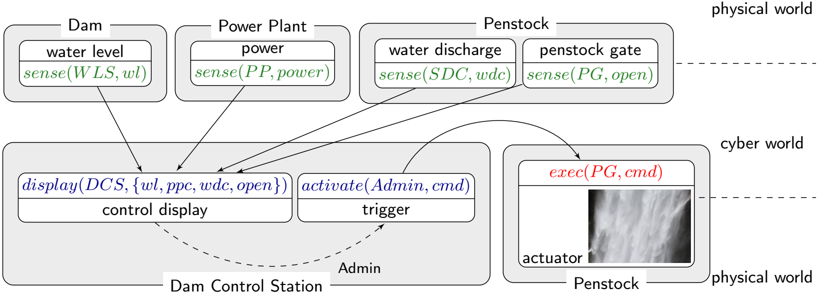
Something bad,

$\boxed{e_0 \rightarrow e_1 \rightarrow e_2}$

must not happen.

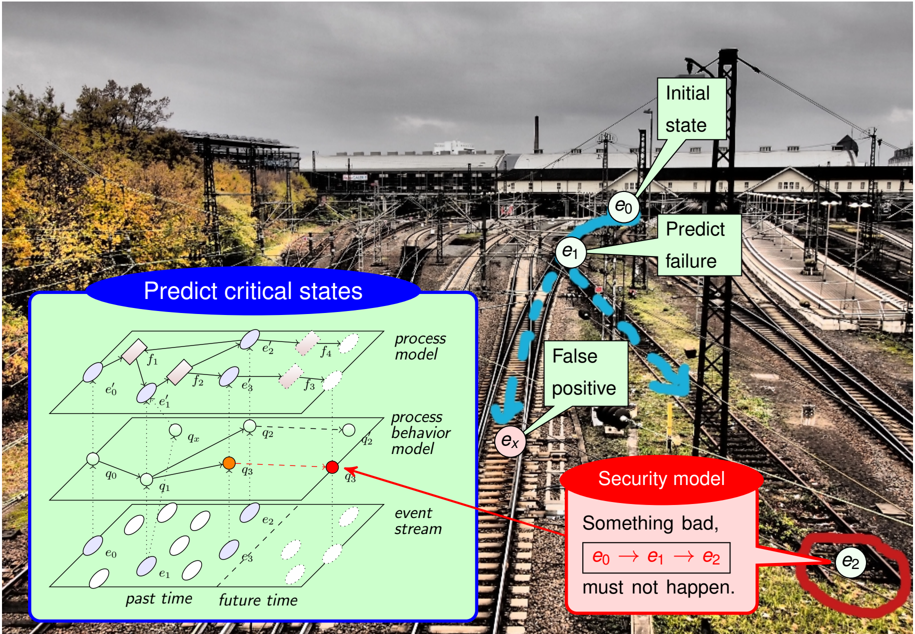
Example: Functional dependencies between sensors, control station components, and actuators

- Dam administrator decisions depend on the displayed measurements
- Control display values are derived from the sensor measurements.
- The overall function of the system requires authenticity of measurement values for several critical sensors.



Andreas Fuchs, Roland Rieke.
Identification of Security Requirements in Systems of Systems by Functional Security Analysis.
In *Architecting Dependable Systems VII, 2010*, Springer LNCS 6420.

Prediction of close future critical states



Initial state

Predict failure

False positive

Predict critical states

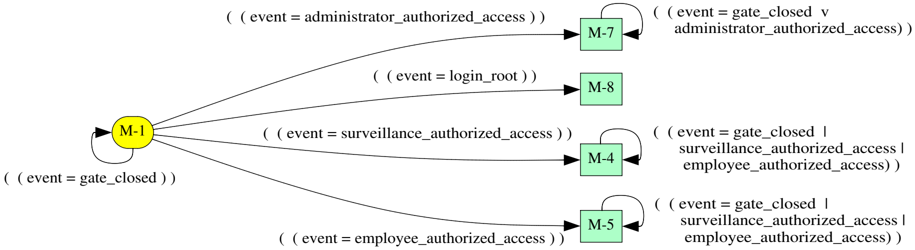
Security model

Something bad,

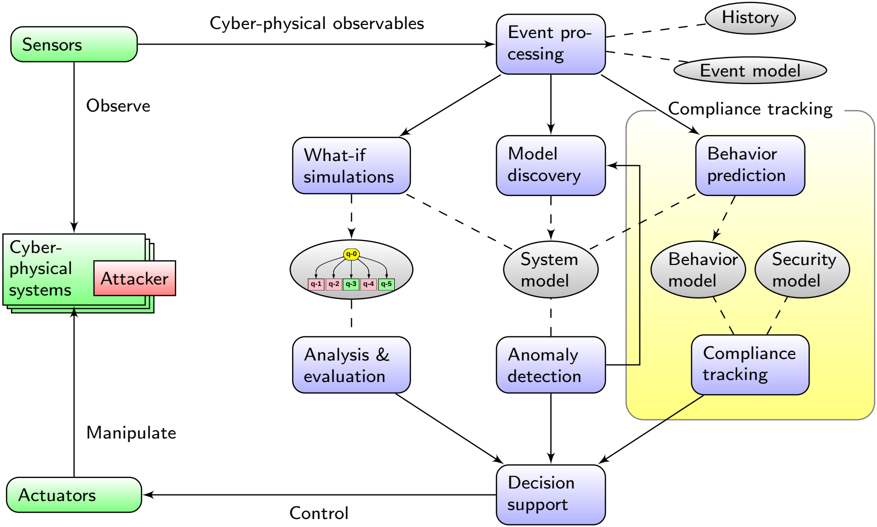
$e_0 \rightarrow e_1 \rightarrow e_2$

must not happen.

Computation of Possible Future Behavior


((event = administrator_authorized_access))
M-7
((event = gate_closed v
administrator_authorized_access))
((event = login_root))
M-8
M-1
((event = surveillance_authorized_access))
M-4
((event = gate_closed |
surveillance_authorized_access |
employee_authorized_access))
((event = gate_closed))
M-5
((event = gate_closed |
surveillance_authorized_access |
employee_authorized_access))
((event = employee_authorized_access))

Compliance Tracking Architecture



Sensors

Cyber-physical observables

Event processing

History

Event model

Observe

Compliance tracking

What-if simulations

Model discovery

Behavior prediction

Cyber-physical systems

Attacker

System model

Behavior model

Security model

Analysis & evaluation

Anomaly detection

Compliance tracking

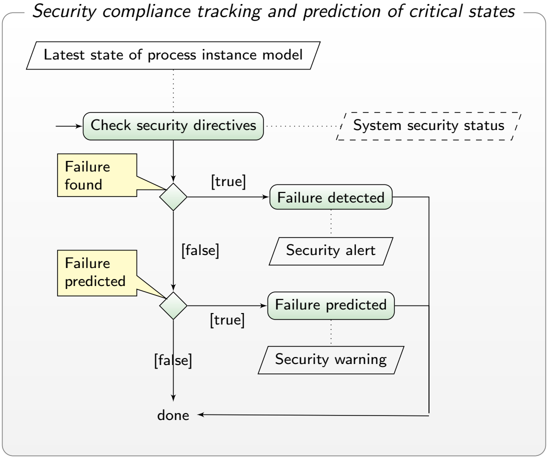
Manipulate

Actuators

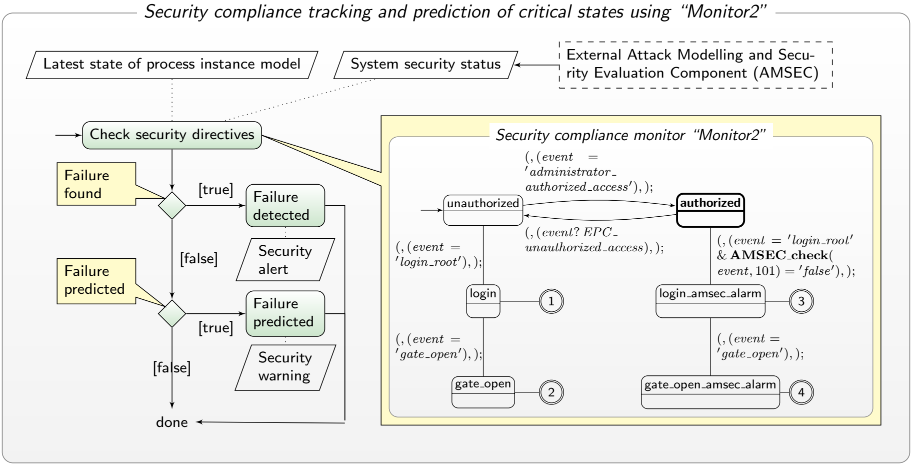
Decision support

Control

Security Compliance Tracking and Prediction of Critical States



Hybrid Symbiotic Simulation (PSA & AMSEC)



Security compliance tracking and prediction of critical states using "Monitor2"

Latest state of process instance model

System security status

External Attack Modelling and Security Evaluation Component (AMSEC)

Check security directives

Failure found

[true]

Failure detected

[false]

Security alert

Failure predicted

[true]

Failure predicted

Security warning

[false]

done

Security compliance monitor "Monitor2"

$(,(event\ =\ 'administrator\_authorized\_access'),);$

unauthorized

authorized

$(,(event?\ EPC\_unauthorized\_access),);$

$(,(event\ =\ 'login\_root'),);$

$(,(event\ =\ 'login\_root'\ \&\ \mathbf{AMSEC\_check}(event,101)\ =\ 'false'),);$

login

1

login_amsec_alarm

3

$(,(event\ =\ 'gate\_open'),);$

$(,(event\ =\ 'gate\_open'),);$

gate_open

2

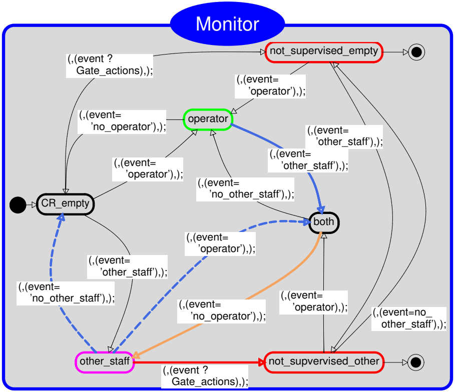
gate_open_amsec_alarm

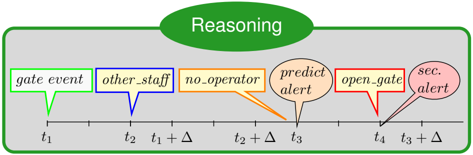
4

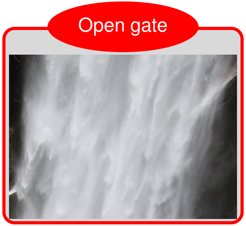
Roland Rieke, Maria Zhdanova, and Jürgen Repp.
Security Compliance Tracking of Processes in Networked Cooperating Systems.
In *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2015.*

Validate security compliance at runtime

↪ Process models identify (close-future) violations of sec. requirements


Monitor
not_supervised_empty
(,(event ? Gate_actions),);
(,(event= 'operator'),);
operator
(,(event= 'no_operator'),);
(,(event= 'other_staff'),);
(,(event= 'other_staff'),);
(,(event= 'operator'),);
(,(event= 'no_other_staff'),);
CR_empty
both
(,(event= 'operator'),);
(,(event= 'other_staff'),);
(,(event= 'no_other_staff'),);
(,(event= 'no_operator'),);
(,(event= 'operator),);
(,(event=no_ other_staff'),);
other_staff
not_supvervised_other
(,(event ? Gate_actions),);

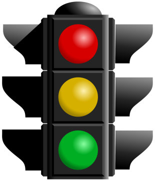




$\Sigma_P := \Sigma_1$

Roland Rieke, Jürgen Repp, Maria Zhdanova, and Jörn Eichler.
Monitoring security compliance of critical processes.
In *Parallel, Distributed and Network-Based Processing (PDP), 2014*, IEEE.

Security Strategy Management



Introduction

Topic 1: Cooperating Cyber-Physical Systems

Topic 2: Symbiotic Simulation

Topic 3: Behavior Conformance Tracking

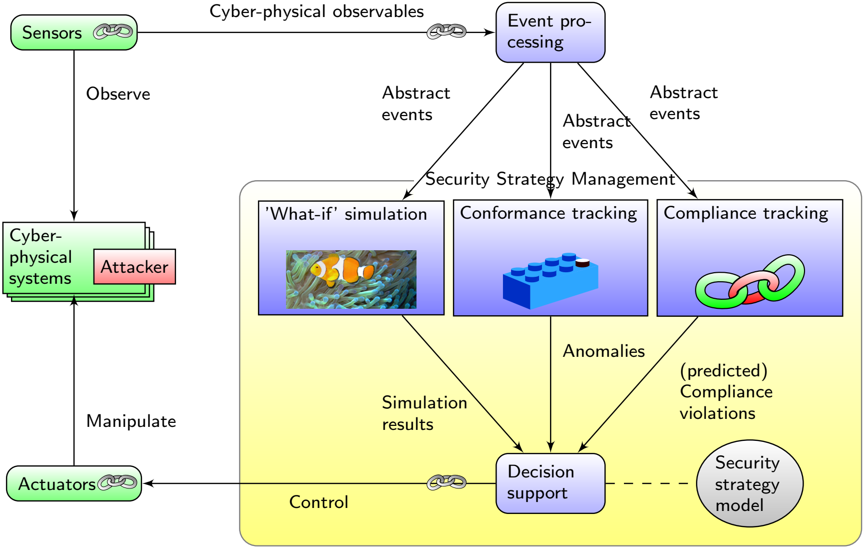
Topic 4: Security Compliance Tracking

Topic 5: Security Strategy Management

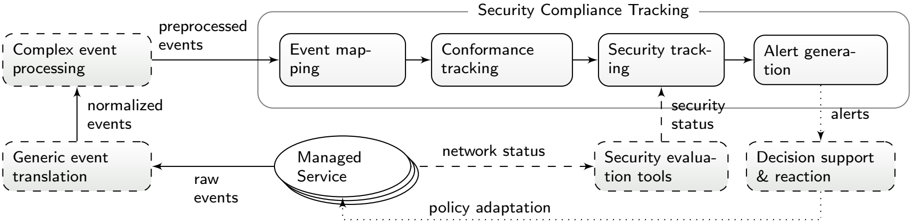
- Decision Support for Security Strategy Management
- Prediction Security Analysis Prototype
- Security Strategy Meta Model
- Usability & Performance

Conclusion

Decision Support for Security Strategy Management



PSA Prototype Implementation



Distinctive features of the PSA@R approach

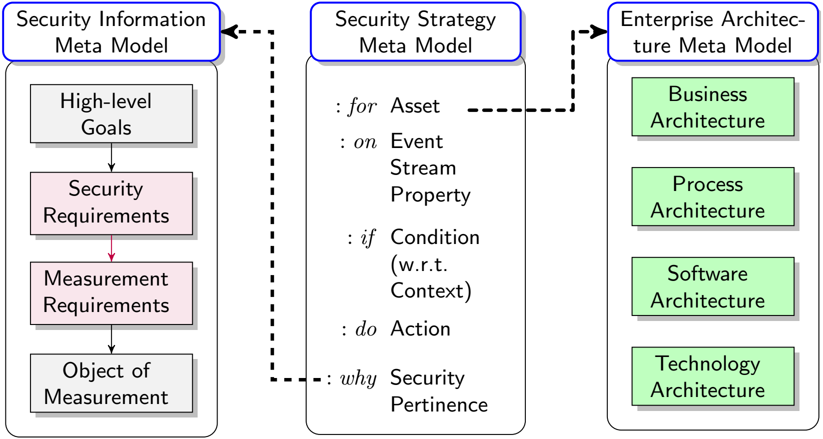
1. Security strategy meta model
2. Systematic security requirements elicitation & monitoring
3. Behaviour anomaly detection
4. Security and safety compliance check at runtime
5. Prediction of close future critical states

Alert Model for Decision Support and Reaction

```
<IDMEF-Message>
  <Analyzer analyzerid="0" name="PSA"
    manufacturer="http://www.sit.fraunhofer.de"
    model="PSA" version="3.0.916" class="Concentrator" ostype="Linux"
    osversion="3.1.10-1.19-desktop">
     <Node category="unknown"><name>tux</name></Node>
     <Process><name>psa</name><pid>19302</pid><path>/local/acl90/clim</path>
     </Process>
  </Analyzer>
  <CreateTime ntpstamp="0xd563683d.0x00000000">2013-06-12T23:35:57+02:00
  </CreateTime>
  <Classification text="Monitor␣Automaton"/>
  <AdditionalData type="xsd:string" meaning="MonitorState">gate_open
  </AdditionalData>
  <AdditionalData type="xsd:string" meaning="Predicted">true</AdditionalData>
</IDMEF-Message>
```

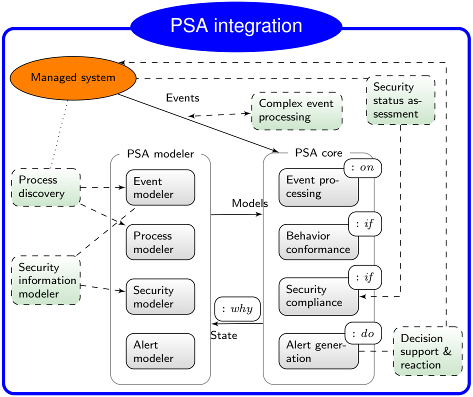
- PSA alert generation applies a mapping from state transitions of the security analysis model onto security events, such as warnings or (predictive) alerts.
- PSA supports Intrusion Detection Message Exchange Format (IDMEF), OSSIM, and the MASSIF format.
- Security alerts are enriched by the information needed for further processing and fed into the runtime environment for delivery to DSR systems.

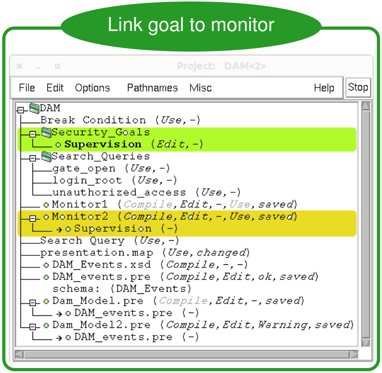
Security Strategy Meta Model



Roland Rieke, Maria Zhdanova and Jürgen Repp,
Security and Business Situational Awareness
CSP Innovation Forum 2015, Springer CCIS 530.

Security pertinence





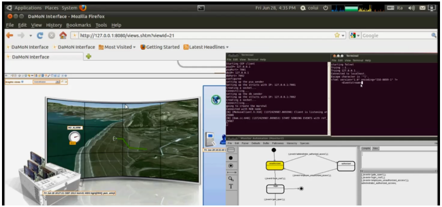
```
DAM
  Break Condition (Use,-)
  Security_Goals
    o Supervision (Edit,-)
  Search_Queries
    gate_open (Use,-)
    login_root (Use,-)
    unauthorized_access (Use,-)
  o Monitor1 (Compile,Edit,-,Use,saved)
  o Monitor2 (Compile,Edit,-,Use,saved)
    → o Supervision (-)
  Search Query (Use,-)
  presentation.map (Use,changed)
  o DAM_Events.xsd (Compile,-,-)
  o DAM_events.pre (Compile,Edit,ok,saved)
    schema: (DAM_Events)
  o Dam_Model.pre (Compile,Edit,-,saved)
    → o DAM_events.pre (-)
  o Dam_Model2.pre (Compile,Edit,Warning,saved)
    → o DAM_events.pre (-)
```

- PSA has a capability to determine how every observed alert refers to (high-level) security goals set for a monitored system
- A backward reference of a security monitor (e.g., "Monitor2") to the originating security goal (e.g. "Supervision"), is provided
- Goal is a textual explanation (e.g., "All critical actions have do be supervised")
- Should be refined to a structured representation of the security directive

Usability in Large Industrial Scenarios

↪ Can the developed methods/tools be successfully adapted to large scale industrial scenarios?

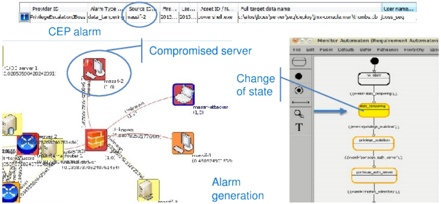
Hydroelectric power plant



PSA integration $\Sigma_P := \Sigma_1$

Source: MASSIF CIPC demo

Olympic Games security



CEP alarm

Compromised server

Change of state

Alarm generation

Monitoring attack path event $\in \Sigma_i$?

Source: MASSIF OOGG demo

"The PSA requires building a model which corresponds to the business process. . . . From the moment the model was defined, the configuration and use of the PSA is easy."

— Comment from PSA evaluation in the MMT scenario (MASSIF D2.3.3)

PSA Performance – Fraud Chain Detection (FCD)

↪ How does PSA behave, compared to state-of-the-art machine learning?

- Classical ML algorithms can perform such a task
  - Need a learning database,
  - Require human time to analyse/detect the chain

↪ Can be difficult in an operational case

	PART		C4.5		Random forest		FCD	
	Normal	Fraud	Normal	Fraud	Normal	Fraud	Normal	Fraud
Actual Normal	465,721	27	465,741	7	465,740	8	465,747	1
Actual Fraud	397	214	381	230	385	226	60	551
Precision	88.79%		97.04%		96.58%		99.81%	
Recall	35.02%		37.64%		36.98%		90.18%	

True positive (TP), False positive (FP), False negative (FN), Precision $\frac{TP}{TP+FP}$, Recall $\frac{TP}{TP+FN}$

- FCD makes the detection easier
  - No needs for learning database
  - Detect the whole chain instead of a faulty transaction

↪ Easy to use in an operational case

M. Zhdanova, J. Repp, R. Rieke, C. Gaber, and B. Hemery,
No smurfs: Revealing fraud chains in mobile money transfers.
In *Availability, Reliability and Security (ARES), 2014*. IEEE.

Reasoning tends to correct itself, and the more so, the more wisely its plan is laid. Nay, it not only corrects its conclusions, it even corrects its premises.

— Charles Sanders Peirce,
The First Rule of Logic (1898)

Conclusions

Conclusions

1. 'Security by Design' needs to be complemented by security compliance tracking @ runtime.
2. Hybrid simbiotic simulation and behavior conformance tracking enrich this approch.
3. Fault forecasting by simbiotic simulation (e.g. analysis of attack graphs) and the optimisation of the network security policy based on this analysis can improve the fault tolerance of a CPS.
4. Model-based analysis supporting fault detection and fault prediction is applicable at runtime for security analysis of real-world applications.
5. Model-based observing systems can be extended by security models to judgemental systems. Anticipated behaviour helps to predict possible failures.
6. CPS need to be designed for security assessment at runtime.
7. Goals, policies, measurement information, and decision rules used in security management need a meta model that consolidates the necessary security strategy information.

> *"It's hard to make predictions, especially about the future."*
>
> Niels Bohr

> *"Correct attribution is hard, especially for the past."*
>
> Doug Arnold, 2010

Спасибо Thanks

Future work

Connect asset and context conditions (*:for + :if*)

- Utilize external (on-the-fly) information, e.g., attack graph, IF-MAP,
- Threat knowledgebase (mapping Structured Threat Information eXpression)

Enrich event stream reasoning (*:on*)

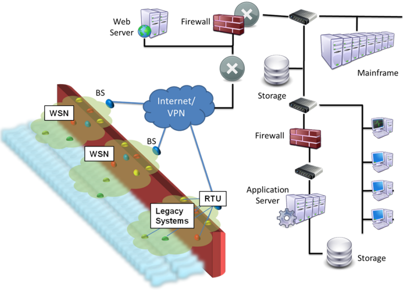
- Availability of models – utilize process discovery
- Improve cross-instance reasoning
- Make use of event history e.g. use neural, immune, neuro-fuzzy classifiers and deep learning techniques for security analysis
- Use of distributed knowledge across network infrastructures to enable co-operative reasoning and response

Formalize security pertinence (*: why*)

- Derive measurement requirements from security goals
- Integrated security meta model, e.g., cross-tool ontology
- Domain knowledge, e.g. vignettes
- Use upcoming standardised formats, e.g. Cyber Observable eXpression, OpenIOC, TAXII protocol

Security-by-Design: Safety by Construction - Well-behaved scalable systems

Elicit security requirements systematically

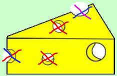


Sec. req. engineering

- Agree on definitions
- Identify vulnerable &/or critical assets
- Identify sec. objectives & dependencies
- Identify threats & develop artifacts
- Risk assessment
- Elicit security requirements
- Categorize & prioritize requirements
- Requirement inspection
- Repository improvement

Mellado et al., 2007

Completeness



e.g., misuse case analysis

Sec. measure independence

- protocols SSL/TLS/VPN/IPv6
- infrastructure PKI, PDP/PEP, **TPM**
- end-to-end/hop-by-hop

e.g., solution patterns

Security requirements elicitation

Security goal

Whenever a critical action happens, the input actions that presumably led to it must actually have happened.

Formally, requirements are defined by specific constraints regarding sequences of actions than can or can not occur in a system's behaviour. *Authenticity* can be seen as the assurance that a particular action has occurred in the past.

$auth(a, b, P)$: Whenever an action b happens, it must be authentic for an Agent P that in any course of events that seem possible to him, a certain action a has happened.

Analyse information flow

- Derive dependencies for a functional model, in which atomic actions are set into relation by defining the functional flow among them.
- Actions of interest are specifically the boundary actions (mapping(sensors) – control(actuators)).

Andreas Fuchs, Roland Rieke.
Identification of Security Requirements in Systems of Systems by Functional Security Analysis.
In *Architecting Dependable Systems VII, 2010*, Springer LNCS 6420.

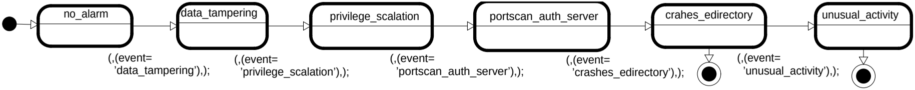
Security Strategy in MASSIF Olympic Games Scenario

Asset (: for) Application for processing accreditation data.

Event Stream (: on) Security events from testbed reproducing the OG infrastructure processed by CEP forwarding malicious activities to security analysis.

1. Data tampering on the the sports entries web server;
2. Privilege escalation on the sports entries web server;
3. Portscan auth server network recognition (port scan) on authentication server;
4. Crashes edirectory remote privilege escalation on the authentication server;
5. Unusual activity multiple login attempts on the accreditation web server.

Condition (: if) Aggregate alarms w.r.t. specified “low-and-slow” attack path.


no_alarm
data_tampering
privilege_scalation
portscan_auth_server
crahes_edirectory
unusual_activity
(,(event= 'data_tampering'),);
(,(event= 'privilege_scalation'),);
(,(event= 'portscan_auth_server'),);
(,(event= 'crashes_edirectory'),);
(,(event= 'unusual_activity'),);

Action (: do) If a critical state is reached, generate a security alert.

Security Pertinence (: why) The goal “prevent unauthorized access to the accreditation data” is linked to this security directive.

Elsa Prieto, Rodrigo Diaz, Luigi Romano, Roland Rieke, Mohammed Achemlal.
MASSIF: A Promising Solution to Enhance Olympic Games IT Security.
In *Global Security, Safety and Sustainability & e-Democracy*, LNICST 99, 2012. Springer.