Generation of Applicative Attacks Scenarios Against Industrial Systems

Maxime Puys Marie-Laure Potet Abdelaziz Khaled

VERIMAG, University of Grenoble Alpes / Grenoble-INP, France
Firstname.Name@univ-grenoble-alpes.fr

Oct. 24, 2017

FPS 2017

This work was partially funded by the SACADE (ANR-16-ASTR-0023) project.

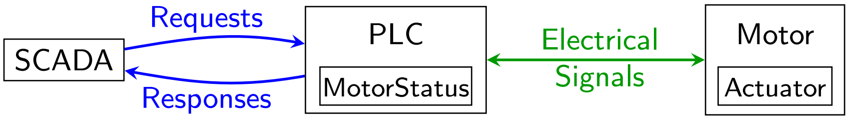
Industrial Systems 1/2

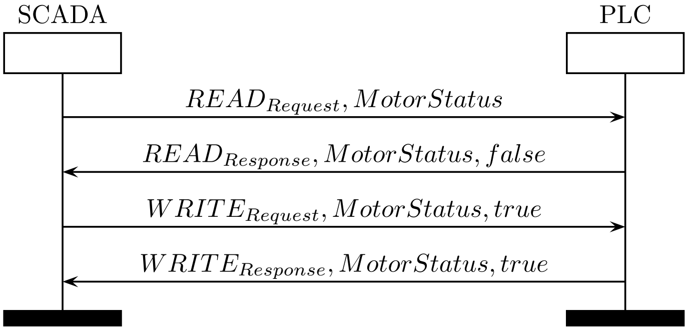
Hot topic

- Since Stuxnet (2009):
  - Complex attack ending up in increasing speed of Iranian centrifuges to damage them.
  - Also attacked the process monitoring to trick operators.
- Protection becoming a priority for government agencies.

Industrial Systems 2/2

- A SCADA controls a PLC which controls a motor.
- Variable *MotorStatus* on the PLC.





Industrial Communication Protocols

MODBUS (1979)

- No security at all.
- Some academic works to secure it (not used in practice):
  - Cryptographic asymmetric signatures [FCMT09]
  - Message Authentication Codes [HEK13]

OPC-UA (2006)

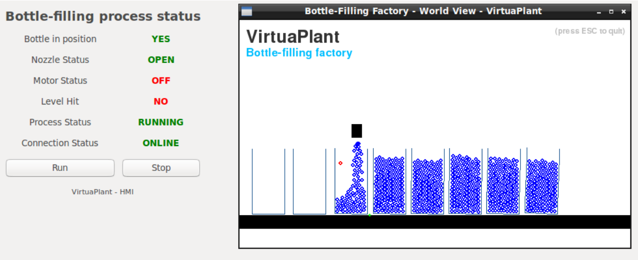
- Security layer: OPC-UA SecureConversation (similar to TLS).
- Three security modes:
  - None, Sign, SignAndEncrypt.

Prior Works on formal verification of security properties

- OPC-UA Handshake, SAFECOMP'16 [PPL16]
- OPC-UA and MODBUS Transport, SECRYPT'17 [DPP^{+}17]

Case Study: Bottle-filling Factory

- Process simulator: `https://github.com/jseidl/virtuaplant`



Variables:

- Conveyor belt
- Nozzle
- Position captor
- Level captor
- On/Off Switch

Properties:

- Nozzle only opens when a bottle is detected.
- Conveyor belt only starts when the bottle is full.
- Nozzle only opens when conveyor belt is stopped.

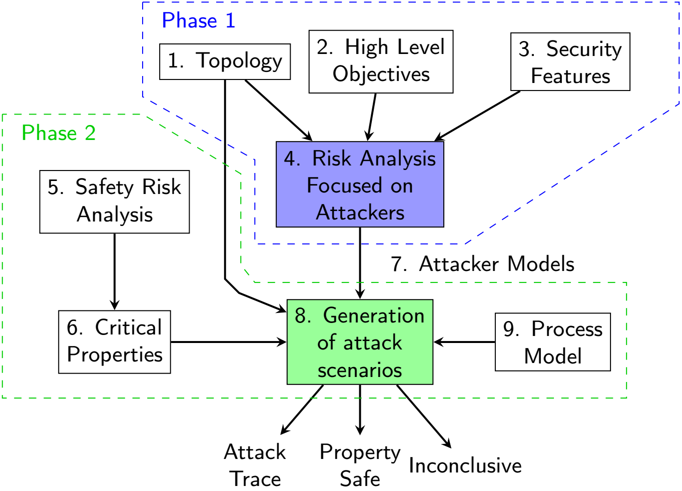
Contributions

- A^2SPICS: Find applicative attacks on industrial systems:
  - Considering an attacker already in the system;
  - What possible actions on the industrial process.
  - E.g.: Nozzle opens with no bottles under it.
- Implementation using the UPPAAL model-checker;
- Proof-of-concept on a case study.

Table of Contents

Table of Contents

The A^2SPICS Approach

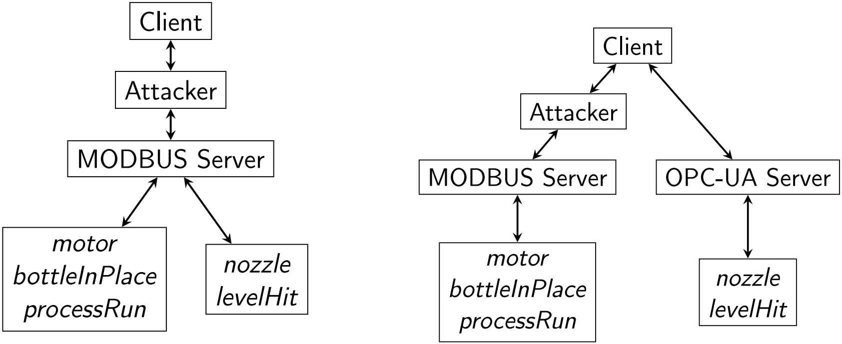


- Phase 1 presented at AFADL 2016, Besançon.

Two examples of topologies

Network topology of the system:

- Communication channels between components;
- Position of attackers.

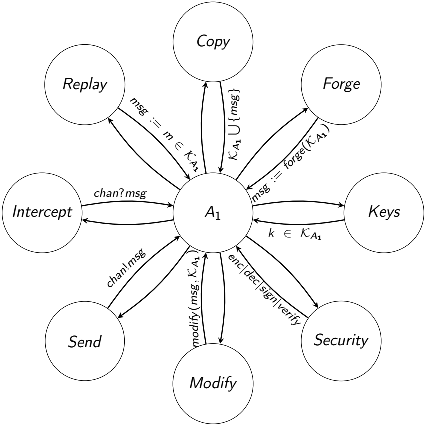


Attackers 1/2

Characterized by:

- Position in the topology:
  - On a channel (Man-In-The-Middle);
  - On a corrupted component (virus, malicious operator, etc).
- Capacities:
  - Possible actions on messages (intercept, modify, replay, etc);
  - Deduction system (deduce new information from knowledge, e.g.: encrypt/decrypt).
- Initial knowledge:
  - Other components;
  - Process behavior;
  - Cryptographic keys, etc.

Attackers 2/2

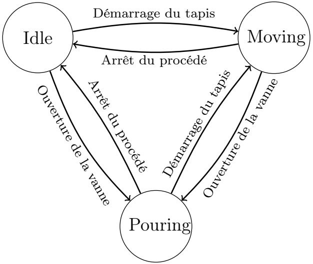


Four attackers:

- A_1 = close to Dolev-Yao;
- Other are subsets of A_1.

Attacker	Modify	Forge	Replay
A_1	✓	✓	✓
A_2	✓	✗	✗
A_3	✗	✓	✗
A_4	✗	✗	✓

Behaviors and Safety Properties



(a) Automaton of the behavior of the process

Current State	Next State	Guard	Actions
Idle	Moving	$processRun = true \wedge bottleInPlace = false$	$motor := true$
Idle	Pouring	$processRun = true \wedge bottleInPlace = true$	$nozzle := true$
Moving	Pouring	$bottleInPlace = true$	$motor := false \wedge nozzle := true$
Pouring	Moving	$levelHit = true$	$motor := true \wedge nozzle := false$
Moving	Idle	$processRun = false$	$motor := false \wedge nozzle := false$
Pouring	Idle	$processRun = false$	$motor := false \wedge nozzle := false$

(b) Transitions Details

Properties: CTL formula:

- Φ_1: At all time and on each path, *nozzle* is never *true* if *bottleInPlace* is *false*).
  $A\Box\neg(nozzle = true \text{ and } bottleInPlace = false)$
- Φ_2: $A\Box\neg(motor = true \text{ and } levelHit = false)$
- Φ_3: $A\Box\neg(nozzle = true \text{ and } motor = true)$

Table of Contents

Analysis tools

Generic verification tools vs. Protocol verification tools

- Generic tools: model-checkers, smt-solvers, etc.
- Protocol verification tools: embed attacker logic.
- Trade-off: tool optimized for verification with attackers vs. granularity.

UPPAAL

- Model-checker created in 1995 at Aalborg and Uppsala Universities.
- Models specified as automata communicating over channels.
- Outputs an attack trace when falsified properties.

Results on the case study

All attackers on all properties (Intel i5-4590 CPU@3.30GHz, 16GB RAM):

- ✓ = attack found;
- ✗ = no attack found;
- $\mathcal{O}$ = inconclusive (here, out of memory).

Topologies	Properties	A_1	A_2	A_3	A_4
T_1	Φ_1	✓	✓	✓	✗
	Φ_2	✓	✓	✓	✗
	Φ_3	✓	✓	✓	✗
T_2	Φ_1	$\mathcal{O}$	$\mathcal{O}$	✗	✗
	Φ_2	✓	✓	✓	✗
	Φ_3	✓	✓	✓	✗

Table of Contents

Timings

Topologies	Properties	A_1	A_2	A_3	A_4
T_1	Φ_1	0.43 s	0.07 s	1.05 s	0.84 s
	Φ_2	0.52 s	0.10 s	0.69 s	0.35 s
	Φ_3	0.47 s	0.04 s	0.37 s	0.42 s
T_2	Φ_1	Out of memory		601 s	31.55 s
	Φ_2	0.66 s	0.23 s	2.17 s	35.20 s
	Φ_3	0.78 s	0.21 s	2.35 s	34.85 s

Observations on results on the POC:

- A_2 obtains same results as A_1 faster (not all capacities of Dolev-Yao are needed to find attacks in this case);
- A_3 globally needs more time but is able to conclude on Φ_1 (less state-space needed);
- A_4 is globally the slowest: as it does not find any attacks, UPPAAL explores all paths.

Conclusion

- A^2SPICS: Find applicative attacks on industrial systems:
  - Considering an attacker already in the system;
  - What possible actions on the industrial process.
  - E.g.: Nozzle opens with no bottles under it.

- Implementation using the UPPAAL model-checker;

- Proof-of-concept on a case study.

Related Works

- Survey on assessment of security in industrial system ([CBB+15, PCB13, KPCBH15]).
- Comparison criteria from [KPCBH15, CBB+15]:

Ref.	Type	Focus	Process model	Probabilistic	Automated
[BFM04]	Model	A	No	No	No
[MBFB06]	Model	A	No	Yes (E)	No
[PGR08]	Model	A	No	Yes (E,H)	No
[TML10]	Model	A	No	Yes (H)	Yes
[CAL+11]	Formula	N/A	Yes	Yes (N/C)	Yes
[KBL15]	Model	A	No	Yes (E)	Yes
[RT17]	Model	A,G	Yes	No	Yes
A^{2}SPICS	Model	A,G	Yes	No	Yes

- Rely on Cl-Atse (protocol verification tool)
  - Dolev-Yao intruder ⇒ less precise control on attacker capacities
- A^{2}SPICS aims at modeling attackers resulting on risk analysis

Limitations

- Time and state of the process are discretized (e.g.: the bottle is either empty or full).
- Number of actions per attack is bounded (configurable, classical limitation of model-checking).
- Model only considers logical state of variables:
  - real state (i.e. if a bottle is physically present or not);
  - logical state (i.e. if the variable *bottleInPlace* is set to *true*);
  - properties are verified on logical state;
  - if a captor is written, a decorrelation is introduced.

    $\Rightarrow$ Can lead to missed attacks (e.g.: Φ_1).

Perspectives

- Study how to address model limitation (real state of process).
- Assess example from [RT17] for a better comparison.
- Tentative of automation with ProVerif and Tamarin.
  - Apply formalisms of [RT17].
- Allow collusions between intruders.

Conclusion

Thanks for your attention!

Maxime Puys

Maxime.Puys@univ-grenoble-alpes.fr

Differences between Industrial and Business IT

- Really long-term installations, hard to patch, lot of legacy hosts.
- Security objectives are different from traditional systems:
  - Availability, integrity, authentication and non-repudiation.
- Messages are READ/WRITE commands to PLCs.
  - Sometimes SUBSCRIPTIONS, RPCs or grouped commands.
  - Industrial protocols: MODBUS, OPC-UA.
- Attack examples: change the value of a WRITE request to change a temperature, change a READ response to mislead operators.

Disambiguation

Security concepts

- Safety = Protection against identified/natural difficulties.
  - Historic industrial concern.
- Cybersecurity = Protection against malicious adversaries.
  - Often called Security.

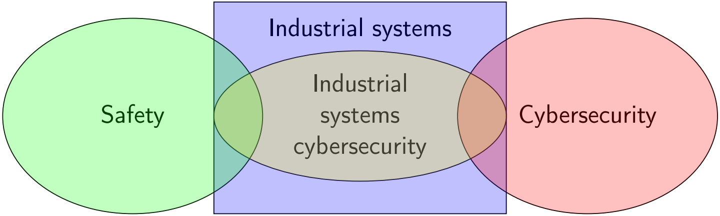


Figure : Relations among security concepts

- Ludovic Pietre-Cambacedes' thesis: On the relationships between safety and security, Telecom ParisTech and EDF, 2010.

Safety and Security

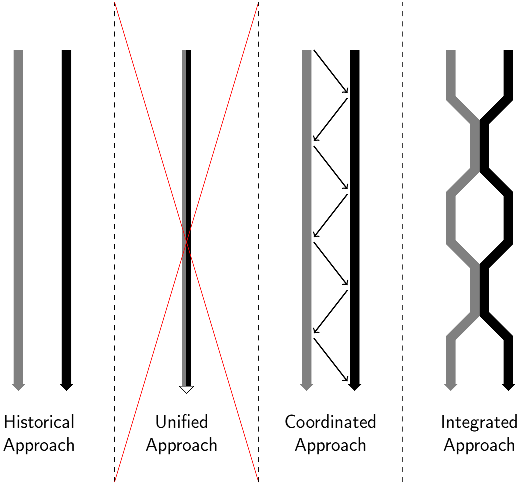


Figure : How to link safety and security [PC10]

Purdue Model

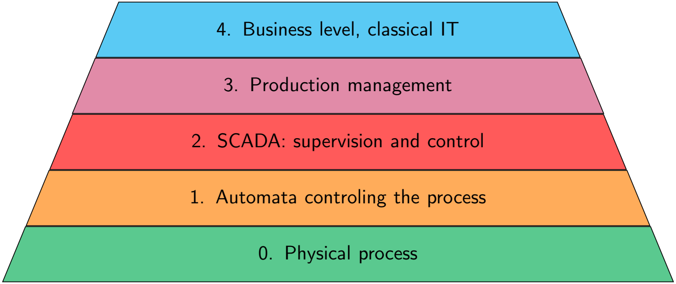


Figure : Purdue model [Wil91]

Motivations on Studying OPC-UA Security

Official specifications: 978 pages.

Several terms redefined afterward:

For this reason, the OpenSecureChannel Service **is not the same as the one specified in the Part 4.** – Part 6, Release 1.02, Page 41.

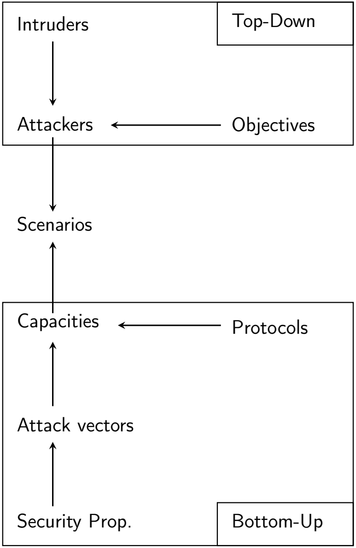
Highly context dependent:

Some SecurityProtocols do not encrypt the entire Message with an asymmetric key. **Instead, they use the AsymmetricKeyWrapAlgorithm to encrypt a symmetric key** [...]. – Part 6, Release 1.02, Page 27.

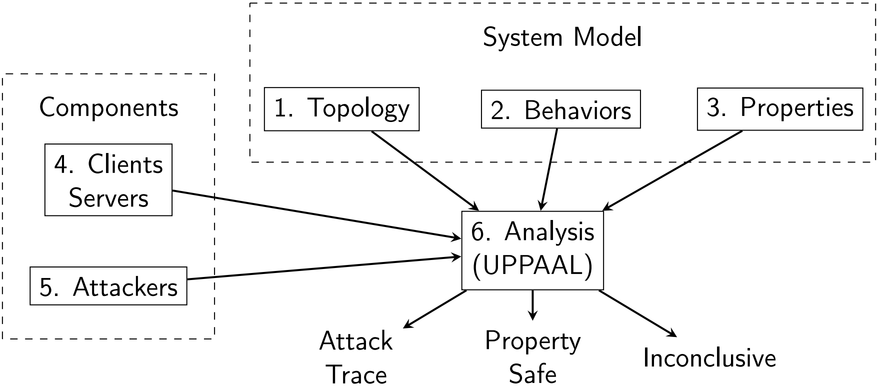
The AsymmetricKeyWrapAlgorithm element of the SecurityPolicy structure defined in Table 22 **is not used by UASC implementations.** – Part 6, Release 1.02, Page 37.

Phase 1: Attacker Models

- Presented at AFADL 2016, Besançon.
- Risk analysis focused on attackers.
- Based on:
  - Topology of the system;
  - Attacker objectives;
  - Security features of protocols.
- Objectives are security vuln., e.g.:
  - Modify a message;
  - Circumvent authentication.
- Yields attacker models in terms of:
  - Position in the topology;
  - Capacities (actions and deduction).



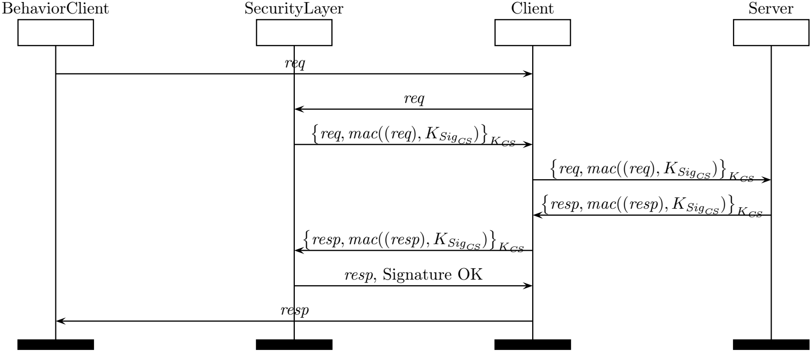
Phase 2: Generation of Attack Scenarios



Clients and Servers

For a transport protocol:

- Encapsulate and decapsulate applicative message into packets.
- Reusable for a model to another.
- BehaviorClient generates applicative messages.
- SecurityLayer performs cryptographic operations.



References I

- Eric J Byres, Matthew Franz, and Darrin Miller, *The use of attack trees in assessing vulnerabilities in scada systems*, Proceedings of the international infrastructure survivability workshop, 2004.
- Alvaro A Cárdenas, Saurabh Amin, Zong-Syun Lin, Yu-Lun Huang, Chi-Yen Huang, and Shankar Sastry, *Attacks against process control systems: risk assessment, detection, and response*, Proceedings of the 6th ACM symposium on information, computer and communications security, ACM, 2011, pp. 355–366.
- Yulia Cherdantseva, Pete Burnap, Andrew Blyth, Peter Eden, Kevin Jones, Hugh Soulsby, and Kristan Stoddart, *A review of cyber security risk assessment methods for SCADA systems*, Computers & Security **56** (2015), 1 – 27.

References II

Jannik Dreier, Maxime Puys, Marie-Laure Potet, Pascal Lafourcade, and Jean-Louis Roch, *Formally verifying flow integrity properties in industrial systems*, SECRYPT 2017 - 14th International Conference on Security and Cryptography (Madrid, Spain), July 2017, p. 12.

IgorNai Fovino, Andrea Carcano, Marcelo Masera, and Alberto Trombetta, *Design and implementation of a secure MODBUS protocol*, Critical Infrastructure Protection III (Charles Palmer and Sujeet Shenoi, eds.), IFIP Advances in Information and Communication Technology, vol. 311, Springer Berlin Heidelberg, 2009, pp. 83–96 (English).

G. Hayes and K. El-Khatib, *Securing MODBUS transactions using hash-based message authentication codes and stream transmission control protocol*, Communications and Information Technology (ICCIT), 2013 Third International Conference on, June 2013, pp. 179–184.

References III

S Kriaa, M Bouissou, and Y Laarouchi, *A model based approach for SCADA safety and security joint modelling: S-Cube*, IET System Safety and Cyber Security, IET Digital Library, 2015.

Siwar Kriaa, Ludovic Pietre-Cambacedes, Marc Bouissou, and Yoran Halgand, *A survey of approaches combining safety and security for industrial control systems*, Reliability Engineering & System Safety **139** (2015), 156–178.

Miles A McQueen, Wayne F Boyer, Mark A Flynn, and George A Beitel, *Quantitative cyber risk reduction estimation methodology for a small scada control system*, System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on, vol. 9, IEEE, 2006, pp. 226–226.

Ludovic Piètre-Cambacédès, *The relationships between safety and security*, Theses, Télécom ParisTech, November 2010.

References IV

Ludovic Piètre-Cambacédès and Marc Bouissou, *Cross-fertilization between safety and security engineering*, Reliability Engineering & System Safety **110** (2013), 110–126.

Sandip C Patel, James H Graham, and Patricia AS Ralston, *Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements*, International Journal of Information Management **28** (2008), no. 6, 483–491.

Maxime Puys, Marie-Laure Potet, and Pascal Lafourcade, *Formal analysis of security properties on the OPC-UA SCADA protocol*, Computer Safety, Reliability, and Security - 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016, Proceedings, 2016, pp. 67–75.

References V

Marco Rocchetto and Nils Ole Tippenhauer, *Towards formal security analysis of industrial control systems*, Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, ACM, 2017, pp. 114–126.

Chee-Wooi Ten, Govindarasu Manimaran, and Chen-Ching Liu, *Cybersecurity for critical infrastructures: Attack and defense modeling*, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans **40** (2010), no. 4, 853–865.

Theodore J Williams, *A reference model for computer integrated manufacturing (cim): A description from the viewpoint of industrial automation: Prepared by cim reference model committee international purdue workshop on industrial computer systems*, Instrument Society of America, 1991.