A Taxonomy of Dynamic Changes Affecting Confidentiality [*]

Maximilian Walter, Stephan Seifermann, Robert Heinrich
{maximilian.walter, stephan.seifermann, robert.heinrich}@kit.edu
Karlsruhe Institute of Technology (KIT)

Abstract

Industry 4.0 facilitates dynamic production processes for highly tailored individual products that require intense cooperation between different organisations. The enabler of such cooperation are cyber-physical systems (CPSs). A set of policies also considering dynamic changes of a request context during runtime has to protect the confidentiality of involved systems. Analysing policy effectiveness already during design time can avoid costly confidentiality flaws. However, the changes that can be evaluated during design time are not clear. Therefore, we identified typical dynamic changes from use cases we gathered with two industrial partners and categorized them accordingly.

1 Introduction

As part of Industry 4.0, the manufacturing process undergoes a digital transformation: machines and sensors communicate with each other or human operators. Communication is no longer restricted to a single organization but includes all participants throughout the whole production process, including suppliers and customers [5]. Therefore, it spreads multiple different organization. This leads to further automation of the manufacturing process that requires new complex processes and complex systems. Both have to adapt to dynamically changing production environments to maintain automation despite of changing requirements or incidents. For instance, the production process has to shift the workload from a broken machine to other ones to minimize production loss. This also requires adjusting the access control policies to allow workers to enter new floors or let maintenance staff access more detailed information of the broken machine. Future generation cyber-physical systems (CPSs) might provide the functionality for this transformation [5].

The downside of intense cross-organisation communication or Industry 4.0 in general are potential confidentiality issues [10]. For instance, logging data of a machine might leak details of the production process or names of operating workers. Therefore CPSs must only share them with relevant participants.

Design time analyses are beneficial for early detection and cost-efficient correction of confidentiality issues in systems. However, design time analyses cannot predict the impact of every change. Therefore, a classification of dynamic changes is needed.

In this paper, we analyse use cases and confidentiality requirements for next-generation CPSs that we created together with two industrial partners, which are active in the Industry 4.0 environment. We focused on confidentiality and tested our results for confidentiality examples. However, the results might fit for other quality aspects as well. First, we identified dynamic changes, which we define in Section 2. Second, we derive a categorization for confidentiality affecting changes in Section 3 that consists of two dimensions: change type and type level. Section 4 concludes the paper.

2 Definition of Dynamic Changes

We focus on dynamic changes that affect confidentiality. We define dynamic changes, which design-time analyses can handle, as follows:

A dynamic change can be every context change during runtime, which is detectable during runtime and foreseeable during design-time.

We derived the focus on context changes in the first part of the definition from a context-aware role-based access control approach [9] for pervasive computing systems. This fits to confidentiality because access control systems are a commonly applied mechanism to establish confidentiality by means of policies and policy enforcement. As described in the approach, such context changes are highly dynamic, i.e. they can occur often and in an unpredictable time during runtime. The context of a software system consists of ambient conditions of the system and participants. The physical location of entities is a good example of these ambient conditions that Zhang and Parashar [2] consider to be relevant as well. Besides the ambient conditions, Kulkarni and Tripathi [9] see dynamic integration of services or resources into a software system as a dynamic change of the context. This is consistent with Dougherty et al. [3] who include changes of the software system into the dynamic system context. For instance, a change of a component during runtime can lead to different results of the applica-

[*] This work is supported by the DFG (German Research Foundation), grant number HE8596/1-1 (FluidTrust). We thank our project partners from CAS Software AG (Germany), IMA sro (Czech Republic), and Charles University in Prague (Czech Republic) for fruitful discussions.

tion that now require different measures to preserve confidentiality.

The second part of the definition is the detection during runtime. Without the ability to detect a change, the system cannot react to it and adapt policies to maintain confidentiality.

While dynamic changes happen in an unpredictable time during runtime, the range of possible changes still has to be known during design time of the policies. Otherwise, incorporating the changes in the policies is not possible. For instance, a redeployment of a component can happen at an unpredictable time during runtime based on decisions of a cloud hosting provider. Anyway, the software designer knows possible deployment targets because of a service-level agreement with the cloud provider. These types of changes match the definition of programmed changes of Endler [1]. Such changes are known and foreseen during design time but happen during runtime. Therefore changes must be foreseeable during design-time. An analysis using this definition shows first promising results [8].

The definition does not guarantee predictability of change impacts during desing time. One reason for this could be missing input data or expressiveness of the used underlying models.

3 Categories of Dynamic Changes

We analyzed different use cases and requirements [6, 7] based on our definition in Section 2 to identify dynamic change categories. We created categories for dynamic changes based on our findings.

3.1 Categories

Based on the different use cases and requirements, we identified two dimensions: the changed entity and the type level. We found five categories in the first dimension and two in the second. The five categories of the first dimension are as follows.

Actors are similar to subjects in access control [4] that interact with the system, such as humans, organizations, or machines.

Input is the information or data object we insert into the system.

Ambient conditions are attributes directly derived from the environment such as the physical location or the current time.

Results of operations describe dependencies to previous operations. The order of operation executions might change the confidentiality of information.

State of the System describes changes based on the current state of the system or business process.

In the second dimension, we distinguish between type-level information and identity information. In case of a worker, the role would be the type level and the particular worker would be the identity information. While design time analyses can easily handle type-level information, the analyses on the identity level might be more complicated. For the identity level, the specific information might be missing since the system usually does not run yet.

4 Conclusion

In this paper, we provided a definition of dynamic changes and created a categorization of dynamic changes with an impact on confidentiality of the system afterwards. The categorization is based on use cases defined with industrial partners.

The categorization provides security analysts with a starting point for systematically investigating potential confidentiality flaws of CPSs and their environment when moving to Industry 4.0.

In the future, we want to extend our categorization for not planned dynamic changes, which will add a certain amount of uncertainty.

References

[1] M. Endler. “A language for implementing generic dynamic reconfigurations of distributed programs”. In: *Proceedings of the 12th Brazilian Symposium on Computer Networks*. 1994.

[2] G. Zhang and M. Parashar. “Context-aware Dynamic Access Control for Pervasive Applications”. In: *CNDS’04*. 2004.

[3] D. Dougherty, K. Fisler, and S. Krishnamurthi. “Specifying and Reasoning About Dynamic Access-Control Policies”. en. In: *Automated Reasoning*. Vol. 4130. 2006, pp. 632–646.

[4] G. Brose. “Access Control”. In: *Encyclopedia of Cryptography and Security*. 2011, pp. 2–7.

[5] M. Hermann, T. Pentek, and B. Otto. “Design Principles for Industrie 4.0 Scenarios”. In: *HICSS’16*. 2016, pp. 3928–3937.

[6] R. Al-Ali et al. *Use Cases in Dataflow-Based Privacy and Trust Modeling and Analysis in Industry 4.0 Systems*. Tech. rep. 9. Karlsruhe, 2018.

[7] S. Seifermann and M. Walter. “Evolving a Use Case for Industry 4.0 Environments Towards Integration of Physical Access Control”. In: *EMLS’19*. Softwaretechnik Trends. 2019.

[8] N. Boltz, M. Walter, and R. Heinrich. “Context-Based Confidentiality Analysis for Industrial IoT”. In: *SEAA 2020*. accepted, to appear.

[9] D. Kulkarni and A. Tripathi. “Context-aware role-based access control in pervasive computing systems”. en. In: *SACMAT ’08*.

[10] K. Zhou, T. Liu, and L. Zhou. “Industry 4.0: Towards future industrial opportunities and challenges”. In: *FSKD’15*. IEEE, pp. 2147–2152.