FoMSESS Jahrestagung 2020

Extended Abstracts

Die Fachgruppe FoMSESS[1] im GI-Fachbereich Sicherheit beschäftigt sich mit der Anwendung von Formalen Methoden und Software Engineering auf die Entwicklung sicherer Systeme.

In ihren Jahrestreffen bietet die Fachgruppe die Möglichkeit, über aktuelle Forschungsarbeiten zu berichten und zu diskutieren und sich mit Gleichgesinnten zu vernetzen.

Das im Jahr 2020 durchgeführte Jahrestreffen wurde pandemiebedingt online gestaltet. Und obwohl ein physisches Treffen von den Teilnehmenden sicherlich immer vorgezogen würde, gelang es zwei Nachmittage mit interessanten Vorträgen und lebhaften Diskussionen zu füllen. Dabei bekamen die Vortragenden die Möglichkeit Extended Abstracts ihrer Beiträge zu verfassen, um diese auf der FoMSESS-Seite zu veröffentlichen. Das Ergebnis sehen Sie gerade vor sich.

Viel Spaß beim Lesen!

Andreas Nonnengart, Dezember 2020

[1] https://fg-fomsess.gi.de/

FogProtect: Protecting sensitive data in the computing continuum
— Extended abstract —

Zoltán Ádám Mann[*]

University of Duisburg-Essen, Essen, Germany

Abstract

Recent approaches like fog computing or edge computing extend the concept of cloud computing to the edge of the network. The resulting *computing continuum* comprises cloud data centers, devices offering cloud-like services near the network edge, called fog nodes, and end devices. Sensitive data stored or processed in the computing continuum is exposed to a variety of threats, potentially compromising security and privacy.

FogProtect is a Horizon 2020 research and innovation project, running from 2020 until 2022. FogProtect aims at ensuring end-to-end data protection across the computing continuum. To achieve this goal, FogProtect develops and combines increased management capabilities to address key aspects of data protection in the computing continuum.

1 Introduction

The computing continuum (shown schematically in Figure 1) is the result of a recent trend in which computing power is not only available in centralized cloud data centers, but also in decentralized compute nodes called fog nodes. This way, end devices can offload their demanding computing tasks to either a fog node or a cloud data center [3, 7]. Both of these options have their advantages and disadvantages. Communication with a nearby fog node incurs low latency, but the computing power of fog nodes is limited. In contrast, the cloud offers virtually unlimited computing power, but with a considerable network latency. Depending on the application and the specific situation, the available fog and cloud resources can be leveraged to find their optimal usage [2].

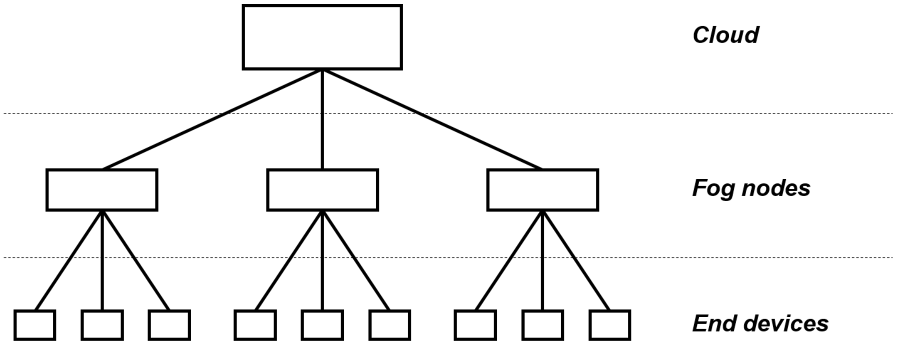

Figure 1: The computing continuum

Processing or storing sensitive data in the computing continuum is associated with significant risks. In particular, the handling of personal data must be compliant with the European General

[*] Joined work with the FogProtect consortium, see https://fogprotect.eu/

Data Protection Regulation [4]. Data protection is already challenging in the cloud [6]. The additional use of fog nodes and end devices makes data protection even harder. On the one hand, the characteristics of fog nodes and end devices, such as their physical vulnerability and mobility, lead to new challenges. On the other hand, increasingly dynamic changes at the edge, such as frequent connectivity changes, result in data protection risks that are hard to foresee and address by design [5].

2 The FogProtect project

The EU Horizon 2020 project FogProtect aims at addressing these challenges and ensuring end-to-end data protection across the computing continuum [1]. For this purpose, FogProtect devises new approaches in multiple areas, including data-protection-aware adaptive service and resource management, end-to-end security management, and run-time risk management. A central element in the FogProtect approach is the use of automatic run-time adaptations to react to changes and ensure the continued satisfaction of data protection requirements.

FogProtect uses formal methods and software engineering techniques in multiple areas. On the one hand, formal models are used to capture the configuration of the computing continuum, problematic configuration patterns, adaptation rules, and security policies. On the other hand, several algorithms have to be elaborated for risk analysis, the analysis of the implications of adaptations, or the search for the best adaptation in a given situation.

Acknowledgments

Research leading to these results received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no. 871525 (FogProtect).

References

[1] Dhouha Ayed, Eva Jaho, Clemens Lachner, Zoltán Ádám Mann, Robert Seidl, and Mike Surridge. FogProtect: Protecting sensitive data in the computing continuum. In *8th European Conference on Service-Oriented and Cloud Computing (ESOCC), accepted*, 2020.

[2] Julian Bellendorf and Zoltán Ádám Mann. Classification of optimization problems in fog computing. *Future Generation Computer Systems*, 107:158–176, 2020.

[3] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. Fog computing and its role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, pages 13–16, 2012.

[4] General Data Protection Regulation. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC. *Official Journal of the European Union*, page L119, 2016.

[5] Zoltán Ádám Mann, Andreas Metzger, and Klaus Pohl. Situativer Datenschutz im Fog-Computing. *Informatik Spektrum*, 42(4):236–243, 2019.

[6] Nazila Gol Mohammadi, Zoltán Adám Mann, Andreas Metzger, Maritta Heisel, and James Greig. Towards an end-to-end architecture for run-time data protection in the cloud. In *44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pages 514–518. IEEE, 2018.

[7] Luis M Vaquero and Luis Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM Computer Communication Review*, 44(5):27–32, 2014.

Formalization of the Layered Privacy Language LPL

Jens Leicht
paluno - The Ruhr Institute for Software Technology
University of Duisburg-Essen
Germany
E-mail: jens.leicht@uni-due.de

November 13, 2020

Data protection legislation, like the General Data Protection Regulation (GDPR) [1] of the European Union, makes the creation of privacy policies for online services and websites difficult. Many aspects have to be considered when compiling a privacy policy and some important requirements may be missed by the policy author. We work on assistance for policy authors during the creation of privacy policies, by providing them feedback on the privacy policy they are creating.

For this feedback we formalize the privacy policies, as well as the requirements stipulated by regulations. We use these formalizations to find internal conflicts in the policies as well as conflicts with regulations, which then can be used to provide feedback to the author, before the policy is published. This allows policy authors to make adjustments, resolving internal conflicts and improving compliance with regulations.

In previous work [2] we performed a survey on 18 privacy policy languages, considering the expressiveness of theses languages concerning data protection regulations. The results of the survey showed four languages that were suitable for expressing policies that comply with data protection regulations. For the formalization we take a look at a language that was part of the survey, but was not considered suitable at the time of creation of the survey, the Layered Policy Language (LPL) by Gerl et al. [3]. LPL is under constant development and thus now fulfils all requirements that were examined in the survey.

The Layered Policy Language aids the negotiation of privacy policies between the end-user and the service provider that collects the data from the user. Policies written in this language also support the negotiation of policies further down the data usage chain. Service providers can provide the policy together with the data to data processors, so that these can follow the privacy policy agreed up on with the user. This principle is known as *Sticky Policies* [4]. LPL provides many elements that are catered to the GDPR and thus is well suitable for the usage in our formalization of policies and legislation.

We perform the formalization of LPL using prolog[1]. Our formalization on the one hand performs correctness checks according to the language definition,

[1] `https://www.swi-prolog.org/`

as well as checks for consistency between the elements of a policy. On the other hand, the formalization of requirements stated in regulations allows the implementation of compliance checks, which can inform the policy author about any issues contained in the policy.

An example for the consistency checks are internal references between policy elements. Some policy element may reference other elements inside the policy. Here we can make sure that the referenced elements are of the correct type and actually exist. Furthermore, we can check whether a combination of attributes, provided for a policy element, is logically combinable or whether a conflict exists between those attributes. The removal of inconsistencies in the policies also improves the readability and, thus, makes it easier for end-users to understand the policies.

As an example for compliance checks, we can take a look at data transfers to countries outside the European Union. These transfers must fulfil certain requirements to be compliant with the GDPR. Our formalization can check whether all requirements are met. In case some information is missing, or the transfer is not allowed at all, we can inform the policy author. The policy can then be updated, by either providing the necessary information of removing the third country transfer overall.

References

[1] European Parliament and Council of the European Union, "Regulation 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)," *Official Journal of the European Union*, vol. L119, pp. 1–88, 2016. [Online]. Available: http://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L:2016:119:TOC

[2] J. Leicht and M. Heisel, "A survey on privacy policy languages: Expressiveness concerning data protection regulations," in *2019 12th CMI Conference on Cybersecurity and Privacy (CMI)*. IEEE, 2019, Conference Proceedings, pp. 1–6.

[3] A. Gerl, N. Bennani, H. Kosch, and L. Brunie, *LPL, Towards a GDPR-Compliant Privacy Language: Formal Definition and Usage*. Springer, 2018, pp. 41–80. [Online]. Available: https://doi.org/10.1007/978-3-662-57932-9_2

[4] S. Pearson and M. Casassa-Mont, "Sticky policies: An approach for managing privacy across multiple parties," *Computer*, vol. 44, no. 9, pp. 60–68, 2011. [Online]. Available: https://doi.org/10.1109/Mc.2011.225

Pattern-based data protection of dynamic systems at run time – Extended Abstract

Florian Kunz[*]

Universität Duisburg-Essen, Essen, Germany

November 13, 2020

Keywords: self-adaptation, data protection, security, privacy, cloud computing, fog computing, edge computing

Abstract

The protection of sensitive data is becoming increasingly more important. There are already several options to protect data. However, these options differ greatly in their costs. These costs can be monetary, or they can be a limitation of the system's functionality. Much of this sensitive data is located in distributed systems, such as cloud systems. It is particularly difficult to protect the sensitive data there because the configuration of these systems is continuously changing.

To solve this problem, we present the RADAR (**R**un-time **A**daptations for **DA**ta p**R**otection) approach. This approach starts with the development time and protects the system during runtime. By applying adaptations during runtime, RADAR can ensure that the data in a distributed system is protected in the most efficient way at all times. In addition, RADAR ensures that the adaptations that have the least possible negative impact on the costs and functionality of the system are performed.

1 Introduction

Many software systems process, store, or transfer data that must be protected from unauthorized access. For example, personal data must be protected in compliance with applicable laws, such as the General Data Protection Regulation (GDPR) of the European Union (EU) [1]. Also non-personal data may require protection, e.g., in the case of trade secrets.

Data processing software is often deployed in the cloud [2]. Cloud platforms are associated with special data protection risks, stemming from virtualization, multi-tenancy etc. [3]. The complexity of cloud systems, including many different hardware and software components, services, and different types of stakeholders, offers a large attack surface, thus making it especially challenging to protect data stored and/or processed in the cloud [4]. Recent trends to distribute cloud-like services to the network edge, often referred to as fog computing or edge computing, increase the difficulties of data protection even further [5].

2 Overview

In order to challenge this problem, we present **RADAR** (**R**un-time **A**daptations for **DA**ta p**R**otection), which builds on our previous research[6]. RADAR is a model-based approach to automatically enforce data protection at run time in dynamic environments using adaptations.

As shown in Figure 1, RADAR is divided into two phases, the design time (upper part of the figure) and the deployment / run time (lower part of the figure). At design time, three types of

[*] joined work with Zoltán Ádám Mann, Jan Laufer, Julian Bellendorf, Andreas Metzger, Klaus Pohl

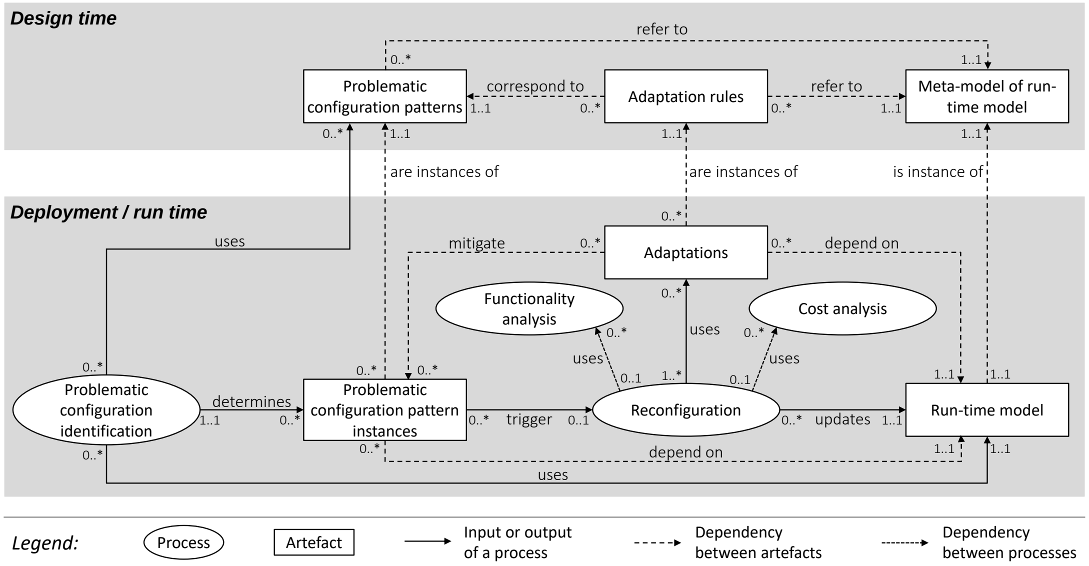

Figure 1: Overview of the RADAR approach

artefacts are created. The Meta Model, the Problematic Configuration Patterns (PCP) and the Adaptation Rules. At deployment time a run-time model is instantiated, which is permanently updated during run time. This run-time model is constantly checked for instances of PCPs. If at least one instance of a PCP is found, a reconfiguration of the run-time model is initiated. During reconfiguration, the system searches for adaptation sequences that allow the instances of PCPs found to be mitigated. The found adaptation sequences are then examined for their influence on functionalities and costs. The best adaptation sequence is the one that mitigates most of the found PCP instances and has the least negative impact on functionality and costs. This adaptation sequence is then applied to the run-time model.

Acknowledgment. This work was partially supported by the European Union's Horizon 2020 research and innovation programme under grant 871525 (FogProtect).

References

[1] General Data Protection Regulation, Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC, Official Journal of the European Union (2016) L119.

[2] I. A. T. Hashem, I. Yaqoob, N. B. Anuar, S. Mokhtar, A. Gani, S. U. Khan, The rise of "big data" on cloud computing: Review and open research issues, Information Systems 47 (2015) 98–115.

[3] V. Chang, M. Ramachandran, Towards achieving data security with the cloud computing adoption framework, IEEE Transactions on Services Computing 9 (1) (2015) 138–151.

[4] M. Ali, S. U. Khan, A. V. Vasilakos, Security in cloud computing: Opportunities and challenges, Information Sciences 305 (2015) 357–383.

[5] R. Roman, J. Lopez, M. Mambo, Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges, Future Generation Computer Systems 78 (2018) 680–698.

[6] F. Kunz, Z. A. Mann, Finding risk patterns in cloud system models, in: 2019 IEEE 12th International Conference on Cloud Computing (CLOUD), IEEE, 2019, pp. 251–255.