Design and Analysis of Self-protection: Adaptive Security for Software-Intensive Systems

Charilaos Skandylas, Linnaeus University, Sweden [1]

Abstract

Today's software landscape features a high degree of complexity, frequent change and many uncertainties. Cyber-attacks are a common occurrence and their consequences are often severe, thus, systems that are able to dynamically defend themselves from attacks are highly needed. Such systems are called self-protecting systems and aim to autonomously identify, analyse and mitigate threats by adapting their structure and behavior at runtime. This research project will contribute towards providing software-intensive systems with self-protection or adaptive-security capabilities. We enhance the security of architecture-based self-adaptive systems by equipping them with (proactive and reactive) self-protection capabilities as well as attack monitoring. Moreover, we provide a framework that allows an open, decentralized system system to mitigate security threats and self-organize to maximize the average trust between the system entities while maintaining their security policies.

Keywords

Self-Protection, Adaptive Security, Architecture-based Adaptation

1. Introduction

Cyber-attacks grow in sophistication and scale as adversaries develop novel strategies, tactics, and techniques which leads to a frequently changing threat landscape that poses a challenge for society to promptly adapt strategies and employ new tactics and techniques to defend or mitigate such attacks. A system's security challenges during its life-cycle are numerous and change often due to either the system itself evolving or due to the development of new attack strategies, tactics, and techniques. This frequent change generates uncertainty concerning how to strategize a system's cyber-defense, including which tactics to employ and techniques to implement. Since multiple security characteristics are volatile, uncertainty becomes a significant factor when studying a system's security. These characteristics include the system's attack surface, the known vulnerabilities of its components, and the attackers' capabilities. Moreover, the attackers' toolset and capabilities develop, which warrants that a system that claims to protect itself from attacks must include *self-adaptation capabilities* to provide adequate defenses.

Self-adaptive systems, i.e., systems that dynamically change their structure or behavior at run-time to cope with uncertainties in their environment and their dynamic nature, have been often employed to counteract complexity, uncertainties in the environment and changing goals. *Self-protecting systems*, is the class of self-adaptive systems that dynamically alter their capabilities at run-time to either provide defenses that can stop or lessen the consequences of attacks or alter their structure and/or behaviour to defend by making the attacks impossible or less likely. Self-adaptation can aid in mitigating security relevant uncertainties and provide timely countermeasures to complex and frequent attacks. Self-adaptive systems however are not immune to attacks themselves. In particular, given the complexity and evolving nature of these systems, adversaries that target them can exploit their adapting nature to perform attacks that are often not possible on contemporary non-adaptive systems. These attacks are often based on the changing attack surface and further information exposed during a self-adaptive

15th European Conference on Software Architecture: Doctoral Symposium, September 13–17, 2021, Växjö, Sweden

✉ charilaos.skandylas@lnu.se (C. S. L. U. S.)
🌐 https://lnu.se/en/staff/charilaos.skandylas (C. S. L. U. S.)

system's runtime. Therefore, it is necessary to analyze the security of a self-adaptive system as a whole, including the adaptation mechanisms and possible future states in the analysis.

Adaptation can be employed in combination with security relevant techniques to enable dynamic protection in systems that feature uncertainty. In that context, security enforcement approaches are enhanced with adaptation capabilities and can therefore, dynamically alter their analysis capabilities and security enforcement to deal with uncertainties in the security policies, active entities, the maliciousness of entities and arbitration between different components. Such approaches can mitigate threats and the inherent uncertainty efficiently, enabling systems to dynamically change their security functionality and add defenses or reduce their attack surface.

This project aims to address the following research questions:

- **Q-I:** How can we utilize threat modeling and quantitative risk analysis to formally specify and analyze the security of a self-adaptive system under uncertainty at runtime?
- **Q-II:** How can the results of runtime security analysis be combined with adaptation-based techniques to provide self-protecting capabilities to software-intensive systems?
- **Q-III:** How can adaptive security be achieved in an open decentralized system by combining trust-based adaptations, decentralized information flow and decentralized control?

2. Related Work

Security in self-adaptive systems can be viewed either as the analysis the security status of the system, which we refer to as security analysis, or as the provision of solutions to identify, analyze and cope with threats autonomously that we refer to as self-protection.

Security analysis of a self-adaptive system can concerns the security analysis of its basic components and their interactions. Security analysis of the managed system is domain specific. Research has been carried out in cloud based systems [1, 2], service-oriented systems [3, 4], mobile ad-hoc networks [5, 6] and component based systems among other domains. To the best of our knowledge, there has been no research that analyzes formally or otherwise the security of the managing system or of the adaptation mechanisms associated with the adaptation process. Security analysis of the adaptation process is critical since self-adaptive systems are additionally to vulnerable attacks that exploit the adapting or evolving nature of the system.

Self-protection can be accomplished in a variety of ways via a wide berth of techniques. Yuan et al. [7] provide a taxonomy of self-protection that features the following interesting dimensions: self-protection levels, meta-level separation, theoretical foundation, control topology and response timing. Self-protection levels refers to the extend to which the system is protected from attacks. The dominant self-protection level is monitoring and detecting threats (95%) followed by responding and protecting (86%) and planning and prevention (18%). Most approaches (83%) feature some degree of meta-level separation among the managing and managed layer. Multiple theoretical foundations for self-protection have been studied, with over 90% of them being based on heuristics [8, 9, 10]. Other theoretical foundations include optimization and learning [11, 12] and formal methods [13, 14]. Enforcement normally happens at the system boundary level, while response timing is mostly reactive and the control topology is often centralized.

When it comes to architectural self-protection which is the focus of this research project, Yuan et.al.,[15] provide a set of architectural security patterns to facilitate self-protection. These patterns include a protective wrapper pattern to defend against DoS attacks, a software rejuvenation pattern used to mitigate the effects of attacks that cannot be defended against and an agreement based redundancy pattern that is capable of defending against XSS and injection attacks. Each pattern details architectural adaptation, threat detection and mitigation. The authors finally demonstrate how to incorporate their approach into Rainbow[16], a well known and adopted architecture-based self-adaptation framework. Schmerl et.al., [17] build upon the previous work and provide a methodology to compose security-related tactics into higher level strategies to respond to DoS attacks. They base their solution in utility theory and combine security strategies with business qualities to select the strategy that better fits the business context. Probabilistic model checking is employed to select the strategies to be applied to the system by reasoning about the effect of a strategy both to security and other quality objectives.

3. Research Project

This research project is centered around the problem of providing self-protecting capabilities and adaptive security to software intensive systems. Thus, we take a problem-centered approach *design science research method* (DSRM) [18] for the research project design.

Problems We have identified the four sub-problems that follow. **P-I** refers to the security analysis of self-adaptive systems, which requires reasoning about the security of the system in all stages of adaptation. **P-II** refers to proactively protecting a self-adaptive system by selecting the most secure adaptation available when information about its components and vulnerabilities is known and control over its adaptation capabilities is assertable. **P-III** refers to protecting a self-adaptive system at runtime, utilizing efficient countermeasure selection or secure strategy synthesis. **P-IV** refers to the design and implementation of adaptive security in open systems where little about the maliciousness and intentions of the entities is known.

Objectives To address the above sub-problems, we identify the objectives that follow. **O-I** entails the development of a security analysis approach for self-adaptive systems. The analysis must be able to model and reason about the system security under uncertainty and aid in developing suitable protection techniques. **O-II** aims to design a secure adaptation approach that selects the most secure of the available system adaptations, thus, enhancing the security of the system by reducing its attack surface. **O-III** aims to design of two approaches, one to select the best available countermeasure for an identified attack and another to provide secure strategy synthesis i.e., altering the system's strategies to mitigate the effects of or to add effective countermeasures for the identified attack. **O-IV** entails the design and implementation of an adaptive security approach for open, decentralized systems where there is uncertainty in the entities residing in the system, their established trust and their maliciousness.

P-I is addressed by augmenting the architectural models with security-related information, such as vulnerabilities and then utilizing threat modeling at runtime to perform runtime security analysis. We build a threat model that shows the attacker and system interaction during and after adaptation. Using that threat model, we perform quantitative security risk assessment or probabilistic model checking to analyze the security of adaptations. **P-II** is addressed by employing security analysis at runtime whenever an adaptation is triggered and harnessing the analysis results to proactively protect the system. The system is able to keep an updated view of its security status before, during and after adaptation, and select the enabled adaptation that leads to the most secure system state available. We address **P-III** using two different approaches, the first approach entails selecting the most effective countermeasures available to the system to enable it to protect itself in response to security attacks. We consider countermeasure effectiveness, where in the architecture the countermeasure should be applied, and at what time. Countermeasure selection is achieved by a combination of model finding and model checking, we identify the possible countermeasure placements and model check a set of security properties on each of them to identify the most secure one. The second approach entails secure strategy synthesis, i.e., modifying a strategy at run-time to both meet the system's stakeholder goals and mitigate identified attacks. The solution to **P-IV** is based on a combination of decentralized information flow control, trust establishment and update, and decentralized control architectures. Entities in the system are grouped based on their trust to form trust contexts, i.e., hierarchies of trusted elements or other trust contexts. We achieve adaptive security via the combination of runtime trust-aware DIFC enforcement and decentralized control.

Demonstration To demonstrate the applicability of our solutions, we implement prototypes. **D-I** implements security analysis via metric evaluation. We add a security dimension to the utility evaluation, which enables us to consider security in adaptation selection. **D-II** performs security analysis via probabilistic model checking. The verification results of each enabled strategy are aggregated and used to rank the strategies, the most secure of which is selected to adapt the system. **D-III** consists of two prototypes, one implementing runtime countermeasure selection where the best available countermeasure is selected to defend against identified attacks and another prototype implementing secure strategy synthesis, i.e., the strategy selected for execution is modified to add mitigations against identified attacks. **D-IV** implements a set of decentralized control architectures that employ adaptive trust-aware DIFC enforcement to maximize average trust among system entities and enforce security in a decentralized system.

Evaluation We evaluate our solutions by performing experiments using the implemented prototypes. The case studies for **D-I** and **D-II** include ZNN, a well known self-adaptation exemplar news site and InsecureDocumentStore, a document storage and retrieval application of our own design. The case studies for **D-IV** include a microservice-based item shop and a randomly-generated decentralized system. In **E-I**, we evaluate the security level of the system by comparing the values of security metrics after each adaptation. In **E-II** and **E-III**, we perform two types of experiments: (i) experiments to evaluate the effectiveness of the approach in terms of providing sufficient security to the system, and (ii) experiments to evaluate the performance of our solution. In **E-IV**, we measure the average level of trust established and the accompanying running time. The experiments feature multiple scenarios with different system configurations, varying in system scale, number of malicious entities and degrees of system openness.

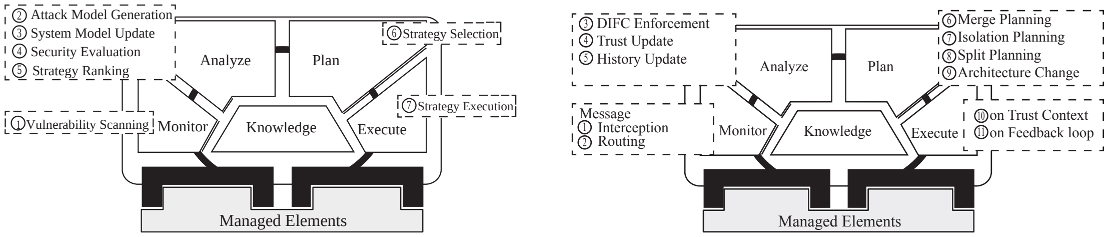

Figure 1: MAPE-K architectures enabling self-protection for **P-I** and **P-II** (left) and **P-IV** (right)

4. State of Research

P-I and **P-II** have been partially addressed in [19] and [20]. In [19] we provide runtime security analysis by utilizing threat modeling and security metrics, while in [20] we provide a formal approach centered around formal architectural modeling of the system and the attacker and probabilistic security verification of each available strategy. The strategies are ranked and the best ranked strategy is applied. **P-IV** has been partially addressed in [21], where we provide an approach combining trust-based techniques, decentralized information flow control and decentralized architectures to provide adaptive security in an open decentralized system. Fig. 1 shows our solutions for **P-I**, **P-II** and **P-IV** as reference MAPE-K loop architectures.

Our current research is split between addressing the shortcomings of our previous research and addressing **P-III**. We have been working on a modular, scalable analysis approach, that improves upon [20], combining modular threat model generation and incremental security analysis. Our extensions to [21] include defenses that shore up the model's shortcomings as well as further security hardening via encryption and additional adaptation options. In terms of addressing **P-III** we are actively working on countermeasure selection using a combination of model finding and verification. We automatically generate all available countermeasure options able to defend against an identified attack and use probabilistic model checking to identify the best option in terms of guaranteeing security.

In our future work, to further address **P-III**, we will work on secure strategy synthesis, i.e., on dynamically modifying enabled strategies to provide mitigations against identified attacks in addition to meeting stakeholder goals. We finally aim to provide a self-protection framework for software-intensive systems utilizing both proactive and reactive methods.

References

[1] J. Du, X. Gu, N. Shah, Adaptive data-driven service integrity attestation for multi-tenant cloud systems, in: Nineteenth IEEE International Workshop on Quality of Service, 2011.

[2] S. Ma, Y. Wang, Self-adaptive access control model based on feedback loop, in: International Conference on Cloud Computing and Big Data, 2013.

[3] J. Li, B. Li, T. Wo, C. Hu, J. Huai, L. Liu, K. Lam, Cyberguarder: A virtualization security assurance architecture for green cloud computing, in: Future Generation Computer Systems, 2012.

[4] C. G. Yee, W. H. Shin, G. S. V. R. K. Rao, An adaptive intrusion detection and prevention (id/ip) framework for web services, in: International Conference on Convergence Information Technology (ICCIT 2007), 2007.

[5] D. Farid, M. Zahidur Rahman, Anomaly network intrusion detection based on improved self adaptive bayesian algorithm, in: Journal of Computers, 2010.

[6] H. Abie, Adaptive security and trust management for autonomic message-oriented middleware, in: 6th International Conference on Mobile Adhoc and Sensor Systems, 2009.

[7] E. Yuan, N. Esfahani, S. Malek, A systematic survey of self-protecting software systems, in: ACM Transactions Autonomic Adaptive Systems, 2014.

[8] H. Reiser, Fault and intrusion tolerance on the basis of virtual machines, 2008.

[9] F. M. Sibai, D. A. Menascé, Defeating the insider threat via autonomic network capabilities, in: Third International Conference on Communication Systems and Networks, 2011.

[10] F. M. Sibai, D. A. Menascé, Countering network-centric insider threats through self-protective autonomic rule generation, in: 6th International Conference on Software Security and Reliability, 2012.

[11] G. . Jabbour, D. A. Menasce, The insider threat security architecture: A framework for an integrated, inseparable, and uninterrupted self-protection mechanism, in: International Conference on Computational Science and Engineering, 2009.

[12] G. Jabbour, D. A. Menascé, Policy-based enforcement of database security configuration through autonomic capabilities, in: 4th International Conference on Autonomic and Autonomous Systems (ICAS'08), 2008.

[13] J. Guttman, A. Herzog, Rigorous automated network security management, in: International Journal Information Security, 2005.

[14] A. V. Taddeo, A. Ferrante, Run-time selection of security algorithms for networked devices, in: 5th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2009.

[15] E. Yuan, S. Malek, B. Schmerl, D. Garlan, J. Gennari, Architecture-based self-protecting software systems, in: 9th International Conference on Quality of Software Architectures, 2013.

[16] D. Garlan, S. . Cheng, A. . Huang, B. Schmerl, P. Steenkiste, Rainbow: architecture-based self-adaptation with reusable infrastructure, in: Computer, 2004.

[17] B. Schmerl, J. Cámara, J. Gennari, D. Garlan, P. Casanova, G. A. Moreno, T. J. Glazier, J. M. Barnes, Architecture-based self-protection: Composing and reasoning about denial-of-service mitigations, in: Symposium and Bootcamp on the Science of Security, 2014.

[18] K. Peffers, T. Tuunanen, C. E. Gengler, M. Rossi, W. Hui, V. Virtanen, J. Bragge, Design science research process: A model for producing and presenting information systems research, 2020.

[19] N. Khakpour, C. Skandylas, G. S. Nariman, D. Weyns, Towards secure architecture-based adaptations, in: 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, 2019.

[20] C. Skandylas, N. Khakpour, Design and implementation of self-protecting systems: A formal approach, in: Future Generation Computer Systems, 2021.

[21] C. Skandylas, N. Khakpour, J. Andersson, Adaptive trust-aware decentralized information flow control, in: IEEE International Conference on Autonomic Computing and Self-Organizing Systems, 2020.