

Chapter 1

Threat Modeling in Agile Software Development

Martin Gilje Jaatun
SINTEF Digital, Norway

Karin Bernsmed
SINTEF Digital, Norway

Daniela Soares Cruzes
SINTEF Digital, Norway

Inger Anne Tøndel
SINTEF Digital, Norway

ABSTRACT

Threat modeling is a way to get an overview of possible attacks against your systems. The advantages of threat modeling include tackling security problems early, improved risk assessments, and more effective security testing. There will always be limited resources available for security, and threat modeling will allow you to focus on the most important areas first. There is no one single “correct” way of doing threat modeling, and “agile” is no excuse for not doing it. This chapter describes the authors’ experiences with doing threat modeling with agile development organizations, outlining challenges to be faced and pitfalls to be avoided.

DOI: 10.4018/978-1-5225-6313-6.ch001

1. INTRODUCTION

Threat modeling has been identified as one of the most important activities in the Security Development Lifecycle (SDL) (Howard & Lipner, 2006). According to Jeffries (Jeffries, 2012), Microsoft SDL author Michael Howard states: “If you’re only going to do one activity from the SDL, it should be threat modeling”. The main idea behind threat modeling is to *think like an attacker*. A well-defined threat model helps to identify threats to the different assets of a system by utilizing well-grounded assumptions on the capabilities of any attacker interested in attacking such a system. It enables the teams to identify critical areas of design, which need to be protected. Over time, various threat modeling approaches and methodologies have been developed, and are being used in the process of designing secure applications (Cruzes, Jaatun, Bernsmed, & Tøndel, 2018). The approaches vary from conceptual frameworks to practical methodologies. To speed up software delivery, many organizations have adopted an agile software development approach, in which development teams produce code in shorter iterations with frequent feedback loops. In agile software development, however, threat modeling is not widespread, and the practitioners have few sources of recommendations on how to proceed to adopt the practice in their process. In addition, in agile software development, it is often challenging in itself to adopt security practices, either because security practices are not prioritized, or because the practitioners are not able to see the relevance and importance of the activities to the improvement of the security in the project (Cruzes et al., 2018). Studies in software security usually focus on software security activities in general, and there are few empirical studies focusing on specific practices in agile software development. The threat modeling activity is particularly important in software security, since many security vulnerabilities are caused due to architectural design flaws (McGraw, 2004). Furthermore, fixing such vulnerabilities after implementation may be very costly, requiring workarounds which sometimes increase the attack surface. A well-defined threat model helps to identify threats to different assets of a system by utilizing well-grounded assumptions on the capabilities of any attacker interested in exploiting such a system. It also enables the development teams to identify critical areas of the design which need to be protected, as well as mitigation strategies. However, threat modeling can also be challenging to perform for developers, and even more so in agile software development.

This chapter is based on results from the ongoing *SoS-Agile - Science of Security for Agile Software Development* research project (<https://www.sintef.no/en/digital/sos-agile/>) which investigates how to meaningfully integrate software security into agile software development activities. The project started in October 2015 and will end in October 2020, and involves many software development companies in Norway. The method of choice for the project is Action Research, which is an appropriate

12 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/threat-modeling-in-agile-software-development/221710

Related Content

Queuing Theory Contributions and Applications in Health Service: A Study in the Field of Management Problems

Alexandre Beraldi Santos, Ana Carolina Sanches Zeferino, Ilma Rodrigues de Souza Fausto, Sandra Maria do Amaral Chavesand Saulo Cabral Bourguignon (2023). *Cases on Lean Thinking Applications in Unconventional Systems* (pp. 163-183). www.irma-international.org/chapter/queuing-theory-contributions-and-applications-in-health-service/313654

Experience with Automatic Product Derivation of Mobile Applications Using Model-Driven Techniques

Elder Cirilo, Uirá Kulesza, Mário Torresand Carlos Lucena (2012). *Handbook of Research on Mobile Software Engineering: Design, Implementation, and Emergent Applications* (pp. 113-123). www.irma-international.org/chapter/experience-automatic-product-derivation-mobile/66463

Domain-Specific Language for Describing Grid Applications

Enis Afgan, Purushotham Bangaloreand Jeff Gray (2009). *Software Applications: Concepts, Methodologies, Tools, and Applications* (pp. 328-365). www.irma-international.org/chapter/domain-specific-language-describing-grid/29396

Service Discovery Architecture and Protocol Design for Pervasive Computing

Feng Zhu, Wei Zhu, Matt Mutkaand Lionel M. Ni (2012). *Advanced Design Approaches to Emerging Software Systems: Principles, Methodologies and Tools* (pp. 83-101). www.irma-international.org/chapter/service-discovery-architecture-protocol-design/55437

A Comparative Analysis of Access Control Policy Modeling Approaches

K. Shantha Kumariand T.Chithraleka (2012). *International Journal of Secure Software Engineering* (pp. 65-83). www.irma-international.org/article/comparative-analysis-access-control-policy/74845