Editorial Preface

A Stitch in Time Saves Nine

Martin Gilje Jaatun, SINTEF Digital, Trondheim, Norway & University of Stavanger, Stavanger, Norway

Penetrate-and-patch used to be the prevailing approach to software security. We've all heard the old chestnut: "We just need to make it work first; we'll worry about security later!" These have ended up as the famous last words of many a software manager.

Another frequent excuse has been that "we cannot do secure software engineering because it doesn't fit our agile development methodology – but I beg to differ. Fair enough, many of the traditional Secure Software Development Lifecycles such as Microsoft SDL, CLASP and the Cigital Touchpoints have a distinct waterfall-ish tinge, but there are plenty of distinct software security activities that can be aligned with any agile methodology, as evidenced by BSIMM and OpenSAMM. In the SoS-Agile project (http://sintef.no/sos-agile) we are working with software development organizations to extend their software security arsenals with activities that work for them – in many cases just being told what to do, and being showed how, is all it takes.

Continuing our campaign of outreach to ordinary developers, we are this year organizing the International Workshop on Secure Software Engineering in DevOps and Agile Development (http://secse.org) as part of the XP conference (19th International Conference on Agile Software Development) in Porto, Portugal, May 21-25 – if you are lucky, the submission deadline is still not passed by the time you read this, so hope to see you there!

This issue contains three articles. First, Reis and Abreu present an extended version of a paper presented at SecSE 2017. "A Database of Existing Vulnerabilities to Enable Controlled Testing Studies" shows how a repository of real software vulnerabilities can be used for better testing of tools that are intended to discover such vulnerabilities, avoiding potential bias of artificial test cases.

Then, Behera and Bhaskari present a novel approach to assembly-language code obfuscation in "Self-modifying code – A provable technique for enhancing program obfuscation".

Finally, Yu et al. provide an overview of security patterns from the last two decades in "Goal Modelling for Security Problem Matching and Pattern Enforcement", where they also demonstrate how their own pattern-based transformation tool can be used to detect security patterns in stakeholder requirements.

Martin Gilje Jaatun
Editor-in-Chief
IJSSE