Eliciting security requirements with misuse cases

Paper by Guttorm Sindre and Andreas L. Opdahl
Reviewed by Ryan Darras

I. Suggestion for acceptance

Mildly accept. Easy read, builds a good story and relatable, but figures and tables are garbage.

II. Summary

The authors consider misuse cases for eliciting security requirements as opposed to use cases because use cases only really focus on the main goal of the average user when using the application. Security requirements often times go right over the heads of the team developing the application because of this, but the authors believe that misuse cases will provide insight towards these security requirements.

III. Positive points

The introduction did a fantastic job of getting the point across. I know exactly what I am going to be reading at this point in time (Abstract should have done this job, but I'll still consider it a positive even though I'm not a huge fan of the abstract).

Overall, this paper was incredibly easy to read and did a great job at defining some concepts that the average reader might not know.

Table 3 was much better than the other figures and tables in the paper. It could be considered a standalone reference to understand misuse cases.

IV. Negative points

Fig 1. Looks like a developer made it. A little bit more time and effort could have generated a more attractive image.

More neutral than negative, but I'm also not a fan of Table 1. It sort of does a good job being a reference for the surrounding text but I think better examples could have been used (table 4, for example was much better).

Table 2 is terrible. Do I read left to right like English? Up-down? How do I use this table to solicit a story.

V. Potential future work

They claim that, "the proposed template is more comprehensive than comparable approaches" but I would love to see a survey paper that actually compares these approaches in real world scenarios. A writer can easily say that their template is more comprehensive, but you don't know when something is comprehensive because anything you are missing is something that you don't know you are missing (I think there is some theorem based on this idea).

At my last software development job, every time we added a new feature to the app we would have to create a unit test to test our newly developed feature. I think it would be interesting to see something similar where every use case had a misuse case (or more) associated with it so that you could objectively search every feature for security vulnerabilities. A paper that focused on the benefits and difficulties of achieving this would be an interesting read in my opinion.

The authors say it as, "use cases are popular tools for eliciting functional requirements but less suited for extra-functional requirements". Future work could look at using misuse cases with other types of extra-functional requirements that aren't just security focused.