Paper Review CS846, Mar. 20
Security cannot be measured and *Do Bugs Foreshadow Vulnerabilities? A study of the Chromium Project*

Wenhan Zhu (Cosmos)
w65zhu@uwaterloo.ca
University of Waterloo
Waterloo, Canada

ABSTRACT

This week's chapter from the book being reviewed here is **Security cannot be measured** by *Andrew Meneely* and **Do Bugs Foreshadow Vulnerabilities? A study of the Chromium Project** by *Felivel Camilo, Andrew Meneely and Meiyappan Nagappan.*

Meneely's chapter gave a overview of what is security, why it's a emergent results, some historical evolution of our understanding of the problem and why security cannot not be measured.

Camilo et al.'s paper is a study on *Chrominum* which focuses on pre and post release relationships of bugs and vulnerabilities of files trying to understand the relationship between. Although some correlations were found but they were all weak, and the major take away is that bugs and vulnerabilities are highly likely to be dissimilar by this study and in the future there should be more research targeting specifically on the two of them especially vulnerability given it's a fairly new focus.

KEYWORDS

paper review

1 SUMMARY

The chapter talks about the problem of security in 5 unescapable facts. The frist is that security in a system is defined by negativity. The common way to see the problem is that we have things that should not happen but we do not have definitive things about doing what can increase security in general. The second is that people come to the realization of vulnerability is unavoidable just like bugs compared to the previous denial of the problem in the early days of software development. The third is vulnerability does not have a linear correlation with security. The general idea here is that vulenrability have different effects and the effect to security are different. The forth is that there is a lack of tracking on design decisions and often flawed design lead to vulnerabilities that's never aforementioned or named differently. Last but not least, the attackers of systems which exploit vulnerabilities are also evolving have inovations of methods happen everyday. We would never being able to answer the question of whether it can stand against an unknown attack.

The paper is an evaluation on *Chromium*, the open source browser by *Google*. They first evaluted the research question that are files with bugs fixes also being fixed for vulnerabilities. Using *Mann-Whitney-Wilcoxon* text, by looking at the number of pre-release bugs on both vulnerable and neutral files from future reports, they found out that vulnerable files on average have a larger number of pre-release bugs. Applying logistic regression analysis on comparing categorios of bug to predict vulnerabilities, they only find a weak association. This suggests that bugs and vulnerabilities are dissimilar groups empirically. By examine the file with most bugs, they found that they have vulnerabilities but not many. And reversely, none of the top 20 files with most vulnerabilities per line of source code is in the top 20 buggiest files.

2 THOUGHTS

The emergent property of security makes it a very hard thing to maintaine in software. And in design, the vulnerabilities could happen anywhere which makes it very hard to catch before release. I've previously went to a talk the speaker (Dan Berry) mentioned that when designing a system, we need to plan everything otherwise if we plan to add things after it'll never be perfect. His example was the *http* protocal, although people has been trying to add security by modifying it, the newer *https* replaced it currently for better security.

I like the paper's idea its trying to show, however, I'm having a hard time understanding the evaluations. The main problem I have understanding the paper is the for the second research question understanding are some types of bugs more closely related to vulnerabilities than others. The method of evaluation they used is to create models then evaluate the results of the model based on 2 criteria. I can't quite understand logically why this method of evaluation would yield the results they are claiming.

3 RATINGS

I would rate *Meneely*'s chapter 4/5. I think it's a good chapter that tries to convey the idea of what security problems are and their state in software engineering and how we should approach them.

I would rate *Camilo et al.*'s work a 3/5. I understand what they are trying to do and the results they claim. However, I have a hard time understanding how most of their evaluation work and the discussion about analysis is very confusing to follow.