When Are OSS Developers More Likely to Introduce Vulnerable Code Changes? A Case Study

Amiangshu Bosu[1], Jeffrey C. Carver[1], Munawar Hafiz[2],
Patrick Hilley[3], and Derek Janni[4]

[1] University of Alabama
[2] Auburn University
[3] Providence College
[4] Lewis & Clark College
asbosu@ua.edu,carver@cs.ua.edu,munawar@auburn.edu,philley@friars.providence.edu,derekjanni@lclark.edu

Abstract. We analyzed peer code review data of the Android Open Source Project (AOSP) to understand whether code changes that introduce security vulnerabilities, referred to as vulnerable code changes (VCC), occur at certain intervals. Using a systematic manual analysis process, we identified 60 VCCs. Our results suggest that AOSP developers were more likely to write VCCs prior to AOSP releases, while during the post-release period they wrote fewer VCCs.

Keywords: Open Source, OSS, FOSS, security, vulnerability.

1 Motivation

The presence of a security vulnerability in a widely used software, like Android, can be critical. Therefore, project management will be able to make informed decisions on allocating scarce security experts, if they can predict 'where', and 'when' vulnerable code changes (VCC) are more likely to be introduced. Most of the prior studies [1,2] on vulnerability prediction considered the 'where' aspects. To investigate the 'when' aspects, this study analyzes 60 VCCs from the Android Open Source Project (AOSP) to determine when AOSP developers are more likely to introduce VCCs.

2 Study Design

The AOSP project uses the Gerrit code review system for changes submitted to the main project branch. Qe mined 18,110 changes from Aug. 09 - May 13. Using an empirically built and validated set of keywords, we identified 741 potential VCCs. Using a two step manual analysis process, we determined that 60 changes were VCCs. Figure 1 overviews our approach.

We hypothesized that: *"developers introduce more vulnerabilities early in a project when it is less mature and developers are less focused on security"*.

L. Corral et al. (Eds.): OSS 2014, IFIP AICT 427, pp. 234–236, 2014.

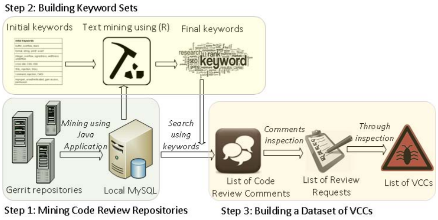

Fig. 1. Research Method

3 Results and Conclusion

The red line in Figure 2 shows that for AOSP, project age does not seem to be the cause of VCCs. Similarly, the gray line in Figure 2 shows that the ratio of VCCs/total changes also does not seem to describe a pattern for introduction of VCCs. As neither of these factors explained the pattern, we conducted exploratory analysis on another factor, release cycle (vertical green lines in Figure 2). It seems that the release cycle may be able to partially explain the introduction of VCCs. The number of VCCs was generally increasing prior to a release and decreasing after a release (although the pattern does not hold in all cases). These results suggest that further study of VCC introduction patterns is needed across more projects. There appear to be factors other than age and total number of changes that may predict when VCCs are introduced. As further research identifies these factors, we can provide more concrete advice to project managers regarding when to emphasize security testing.

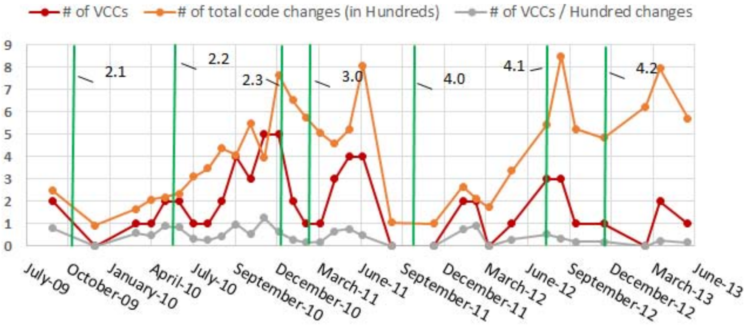

Fig. 2. Number of VCCs identified during code review over 45 months

Acknowledgment. This research is supported by NC State Science of Security lablet.

References

1. Meneely, A., Williams, L.: Secure open source collaboration: an empirical study of linus' law. In: Proc. 16th ACM Conf. on Comp. and Comm. Security, pp. 453–462 (2009)
2. Neuhaus, S., Zimmermann, T., Holler, C., Zeller, A.: Predicting vulnerable software components. In: Proc. 14th ACM Conf. Comp. and Comm. Security, pp. 529–540 (2007)