An Empirical Study on the Effectiveness of Security Code Review

Anne Edmundson[1], Brian Holtkamp[2], Emanuel Rivera[3], Matthew Finifter[4], Adrian Mettler[4], David Wagner[4]

[1]Cornell University, [2]University of Houston-Downtown, [3]Polytechnic University of Puerto Rico, [4]University of California, Berkeley

Overview

We measure the effectiveness of developers at security review.

Research Questions

- *How effective are developers at security code review?*
- *If some developers are significantly more effective than others at the task, can we identify any demographic information that is useful in predicting effectiveness?*
- *Will multiple independent reviewers be significantly more effective than a single reviewer?*
- *How many reviewers are needed to find most or all of the bugs in a web application?*

Methodology

1) Hire web developers through oDesk.com

2) Developers conduct a manual security code review of a vulnerable web application

3) Gather detailed vulnerability reports and demographics from developers

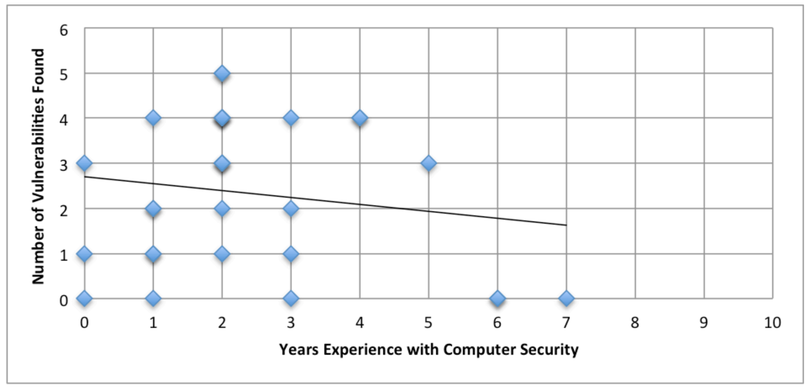
Vulnerability Type: Cross-Site Scripting (XSS)
Vulnerability Location: /foo/bar.php; Line 5
Vulnerability Description: Allows any external user regardless of privileges browsing the page to inject malicious JavaScript that can be reflected to any other users browsing the same page.
Impact: Could run malicious code on any visitor to the same webpage.
Steps to Exploit:
1) Load the page
2) Click on the text field
3) Insert malicious JavaScript code, such as <script>alert(1);</script>
4) Submit the form
5) Reload the page to receive the attack

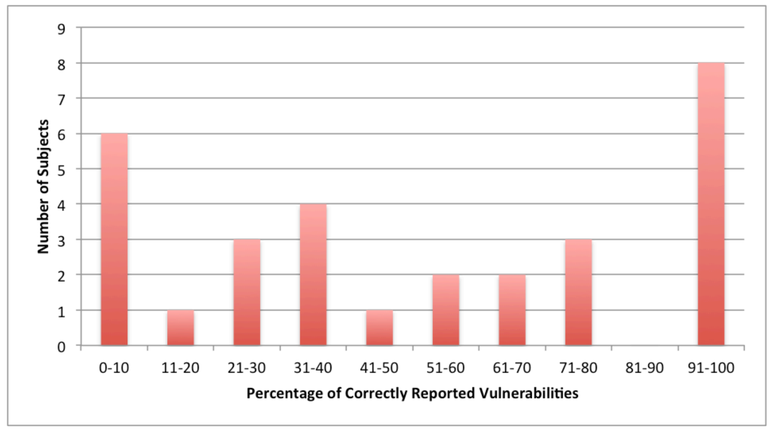
4) Analyze data to determine effectiveness

Vulnerable Web Application:
Contains 5 natural vulnerabilities and 2 artificial vulnerabilities

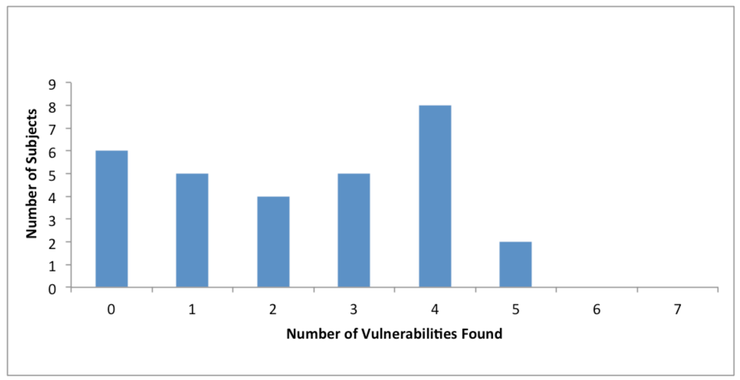
Results

There is an inverse relationship between the number of years of experience with computer security and the accuracy of the report.

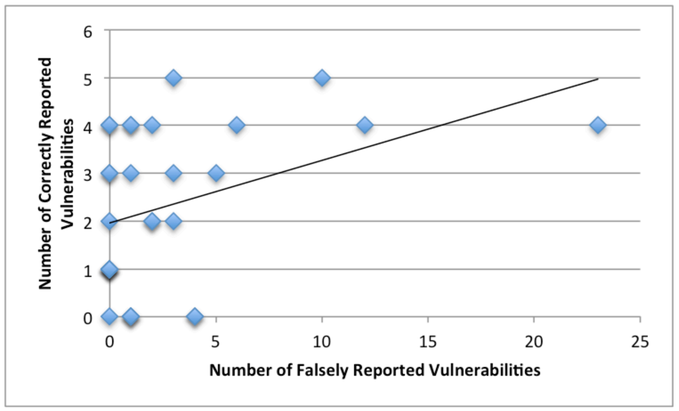




This bimodal distribution shows one group of developers with a very high false positive rate and another group with a very low false positive rate.



The average number of correct vulnerabilities found is 2.33.



The relationship between correctly reported vulnerabilities and falsely reported vulnerabilities.

Conclusion

- **20% of developers did not find any true vulnerabilities**
- **No developer found more than 5 of 7 vulnerabilities**
- **Effectiveness does not appear to be correlated to years of experience in development or self-reported understanding of the application**

Acknowledgements

This work was supported in part by TRUST (Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422). Special thanks to Aimee Tabor and the TRUST program staff.

This work was supported by the TRUST Center (NSF award number CCF-0424422)