Talk Notes: The Influence of Programming Language and Framework on Application Security

Matthew Finifter
finifter@cs.berkeley.edu

February 14, 2011

Motivation

My personal experience developing software has led me to believe that I am more prone to introducing security vulnerabilities in some languages and in some frameworks than I am in others. Having said that, I realize that not everyone shares this experience or opinion. Since this is a workshop about measurement, I'm going to talk about how we can measure whether and to what extent this is the case.

Overview

I'll talk first about the problem I'm trying to address. Then, I'll discuss an experiment we performed to try to address the problem. I'll talk about the major shortcoming of this experiment, which is the size of the data set, and then I'll talk about ways we might address this shortcoming for a future experiment.

Problem

A web application developer has more choices than ever of programming language and web application development framework. Each has its advantages and disadvantages, and none clearly dominates.

Security is not the only concern, but it is one that is increasing. We see frameworks rapidly evolving to take on more responsibility for addressing security concerns. We would like to measure their successes and objectively compare them to one another from a security standpoint.

Experiment

We use data from a previous experiment consisting of 9 implementations of the same web application. 3 teams used PHP, 3 Java, and 3 Perl. Each team chose which frameworks to use, and there was little overlap in these choices. We performed both manual and black-box security analysis of each of the 9 implementations. I'll briefly present some of our results.

Results

This chart displays one set of our results: the number of vulnerabilities found in each implementation. We can see that the PHP implementations each contain more vulnerabilities than each of the Java implementations, and that 2 of the 3 Perl implementations have very few vulnerabilities, while 1 has the most of any implementation.

Results 2

I'm showing here a Venn diagram of the vulnerabilities found using black box testing versus those found using manual analysis. We can see that manual analysis found more vulnerabilities overall, but that the two techniques complement each other. I'd be interested to hear whether anyone else has any data points regarding the comparative effectiveness of these two techniques.

Results 3

This is a table that displays the level of framework support each team had access to for a few different classes of vulnerabilities, as well as whether or not these classes of vulnerabilities were present. For session management and CSRF, there is a strong association between framework support and whether or not the implementation is vulnerable.

Results 4

Overall, we found a few statistically significant results, but there are simply not enough data points to confirm many hypotheses. I'll shift my focus now to ideas for getting a larger data set that may allow us to confirm more hypotheses.

Larger data set

One idea, which is how these data were gathered, is a programming contest. Another is to use student programming projects. The third is to hire a lot of developers to write the same code. Sites like `guru.com` and `rentacoder.com` allow programmers to sell their services. Because many of these programmers are overseas in developing nations, the labor is relatively cheap.

Outsourced development

Our idea is to write a web application in multiple languages using multiple frameworks, but to leave out the implementation of a single security-relevant module. We would then hire developers to implement the missing module in different languages using different frameworks. There is a trade-off here between the sample size and the amount of data within each sample. That is, for a fixed cost, we can have a larger sample size with less implementation from each developer, or we can have a smaller sample size with more implementation from each developer.

Conclusion

We have found some evidence that choice of programming language and framework may play a role in application security. We would like to gather more data for a study of a larger scale, and we would love to hear feedback as to what this community thinks is the best way to do that.