The Influence of Programming Language and Framework on Application Security

Matthew Finifter and David Wagner {finifter, daw}@cs.berkeley.edu

UC Berkeley

February 14, 2011

Motivation

- Some languages and frameworks more prone to vulnerabilities?
- ▶ How can we find out empirically?

Overview

- ▶ The problem
- Experiment
- ► Not enough data
- ▶ How can we gather better data?

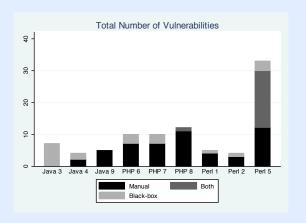
Problem

- Many language and framework choices
 - ► None clearly superior
- Security increasingly important
 - Languages and frameworks evolving to meet this need
- ▶ We need to measure how successful they are

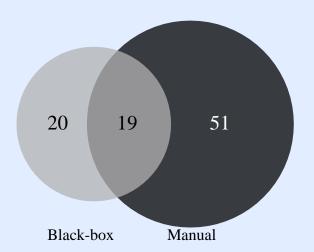
Experiment

- ► Data gathered from a previous study (Prechelt 2007)
- ▶ 9 implementations of same web app: 3 PHP, 3 Java, 3 Perl
- ► Teams chose which framework(s) to use
 - ► Little overlap in framework choice
- Manual and black-box security analysis of each implementation

Results



Results (2)



Results (3)

		CSRF		Session Management		Password Storage	
Team	Language	Vulnerable?	Framework	Vulnerable?	Framework	Vulnerable?	Framework
Number			Support		Support		Support
1	Perl	Х	none		opt-in	Х	opt-in
2	Perl	Х	none	Х	none	Х	none
5	Perl	Х	none	Х	none		opt-out
3	Java		manual		opt-out		none
4	Java		always on		opt-in	Х	opt-in
9	Java	Х	none		opt-in		none
6	PHP	Х	none		opt-out	Х	opt-in
7	PHP	Х	none		opt-out	Х	none
8	PHP	Х	none		opt-out	Х	opt-in

Results (4)

- ▶ A few interesting, significant results
- ▶ But not as many as we would like

Larger data set

- Programming contest
- Student programming projects
- Outsourced development
 - guru.com, rentacoder.com, etc.

Outsourced development

- We write web application in multiple languages using multiple frameworks
- ► Hire programmers for single security-relevant module
- ► Sample size vs. module size

Conclusion

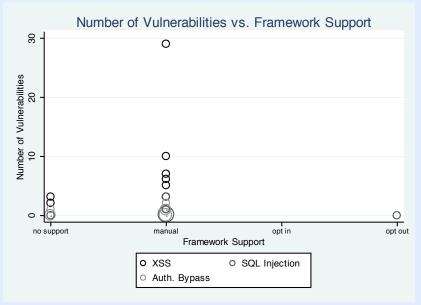
- Have performed small-scale experiment
- Some evidence that language and framework choice influence security
- ▶ Need better data for study of larger scale

Thank you!

Matthew Finifter, finifter@cs.berkeley.edu

14

Results (5)



Results (6)

