Impact of Plugins on Web Application Security

Cincinnati Chapter Meeting
June 29th, 2010

James Walden and Maureen Doyle
Northern Kentucky University

OWASP

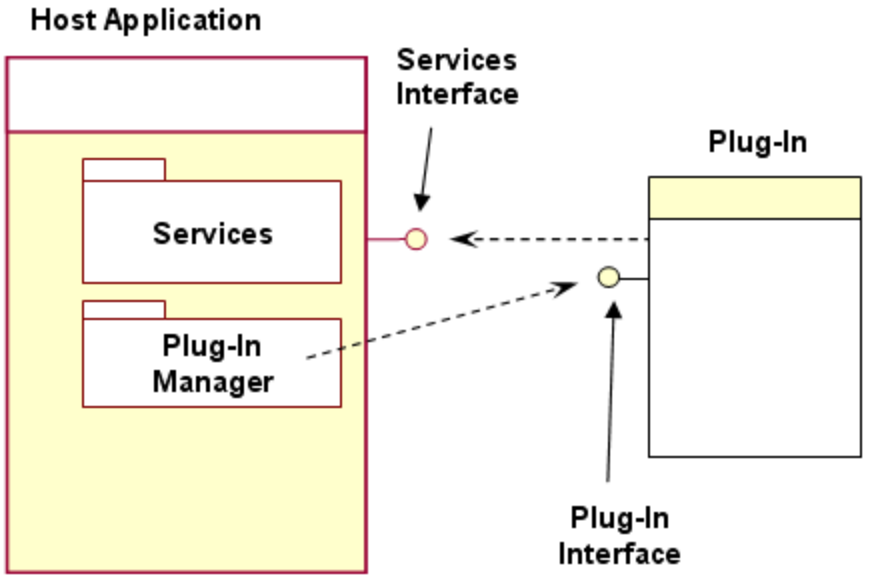
Topics

1. Plugins
2. Measuring Vulnerabilities
3. Plugin Vulnerabilities
4. Comparing Core and Plugin Security
5. OWASP Top 10 Vulnerabilities
6. Conclusions

Plugins

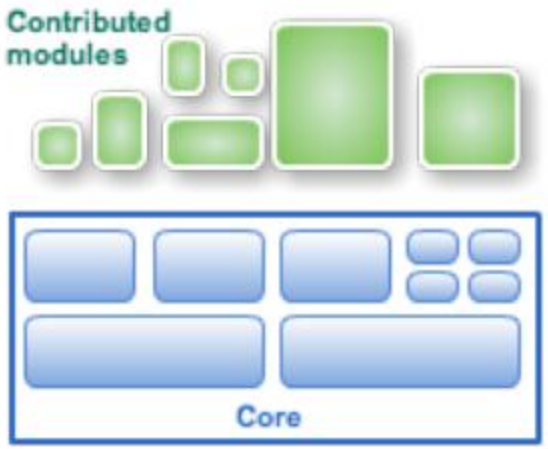
Plugins add features to web applications:

- Advertising
- E-commerce
- Media
- Security
- Site Navigation
- Statistics
- Themes
- User Management



What makes up a web application?

- Is it the core code or code code + plugins?
  - Some apps are almost always deployed with plugins.
  - Plugins are written by non-core developers.
  - Core site may or may not track plugin security.
- Some apps are packaged in distributions with plugins such as Drupal which has:
  - OpenAtrium (Development Seed)
  - Acquia Drupal
  - OpenPublish
  - Pressflow (Four Kitchens)



Research Objective

Goal: Identify differences between security of core code and plugins for web applications.

Research questions:

1. Are plugins less secure than core code?
2. How are vulnerabilities distributed across plugins?
3. How do different applications compare in terms of plugin security?

Measuring Vulnerabilities

Reported Vulnerabilities in NVD or OSVDB

- Coarse-grained time evolution.
- Difficult to correlate with revision.
- Undercounts actual vulnerabilities.

Dynamic Analysis

- Expensive.
- False positives and negatives.
- Requires installation of application.

Static Analysis

- False positives and negatives.
- Static Analysis Vulnerability Density = vulns/kloc.

Measuring Web Application Vulnerabilities

- NVD doesn't offer a web application category.
- Even if they did
  - Commercial web sites don't require users to patch, so vulnerabilities are rarely sent to public vuln DBs.
  - We have to report on open source vulnerabilities.
- Advantages of open source
  - Publicly reported vulnerabilities.
  - Source code available to measure vulnerabilities.
  - Source code available for software metrics.
  - Multiple versions of source code available, making it possible to do time comparisons.

Open Source Web Applications

Selection process

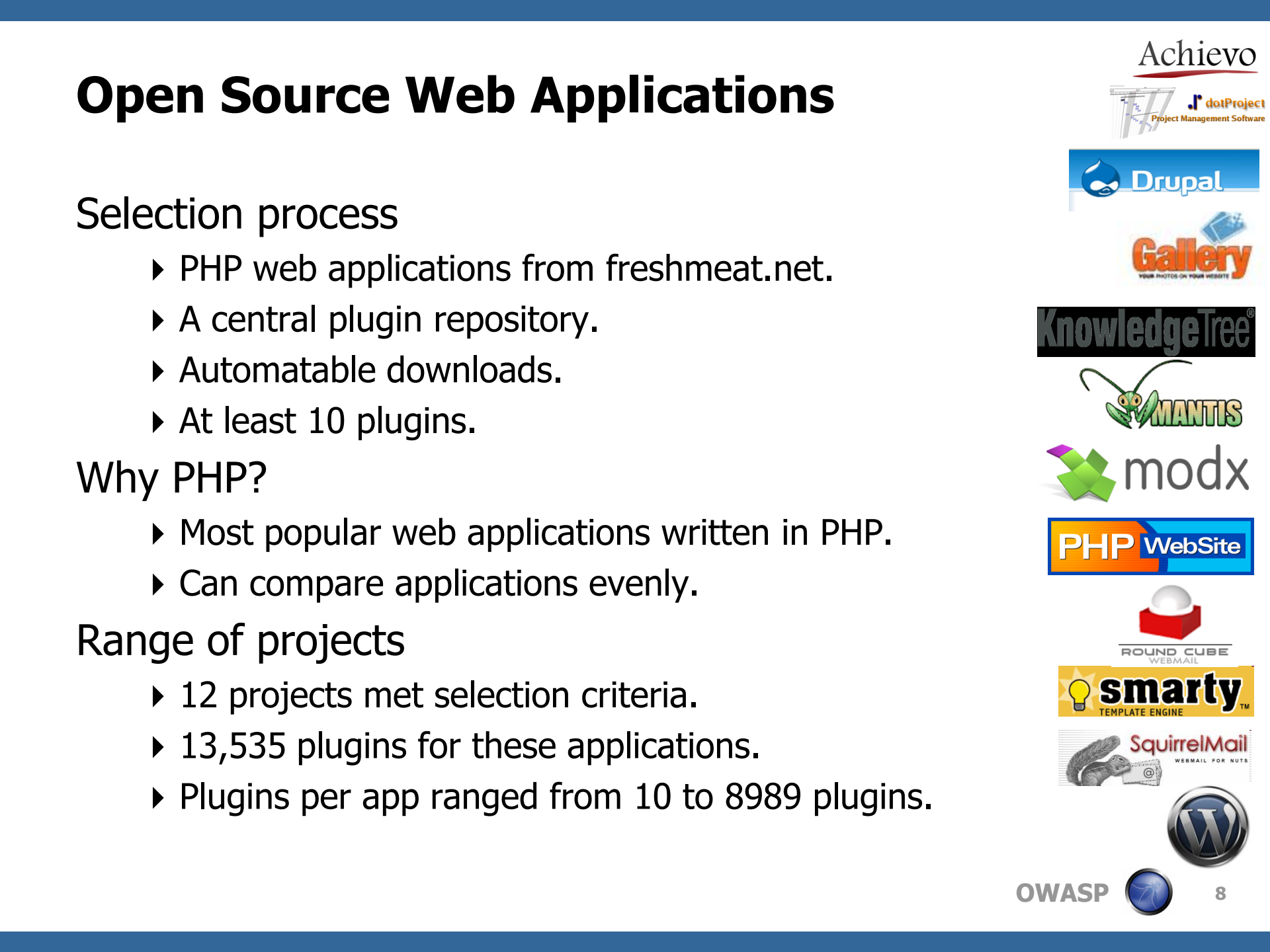
- PHP web applications from freshmeat.net.
- A central plugin repository.
- Automatable downloads.
- At least 10 plugins.

Why PHP?

- Most popular web applications written in PHP.
- Can compare applications evenly.

Range of projects

- 12 projects met selection criteria.
- 13,535 plugins for these applications.
- Plugins per app ranged from 10 to 8989 plugins.



Open Source Applications are Targets

Results

Plugins slightly less secure than core.

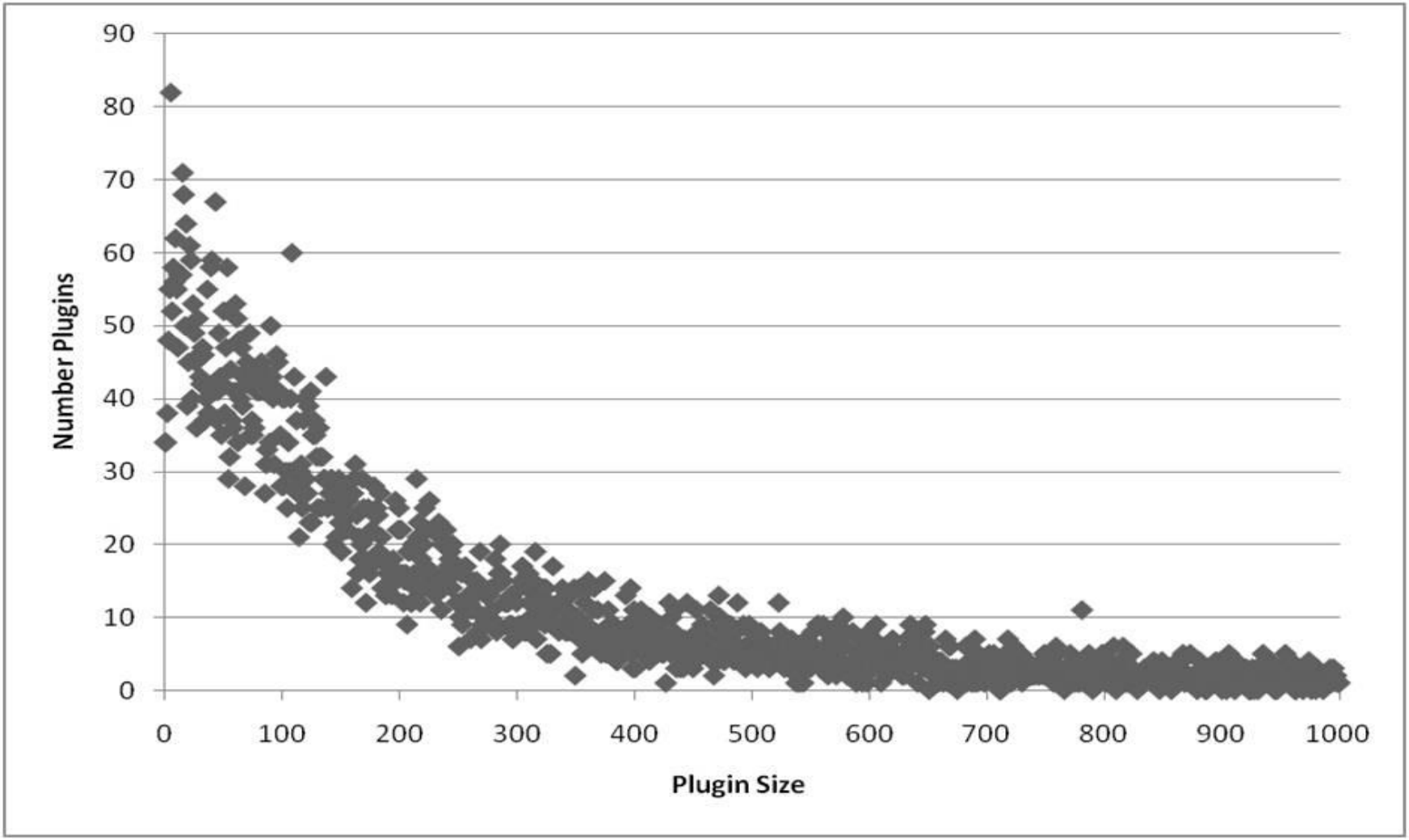
- Plugins made up 91% of 11.7 MLOC.
- Contained 92% of 135,907 vulnerabilities.

Plugin SAVD correlates strongly with code size.

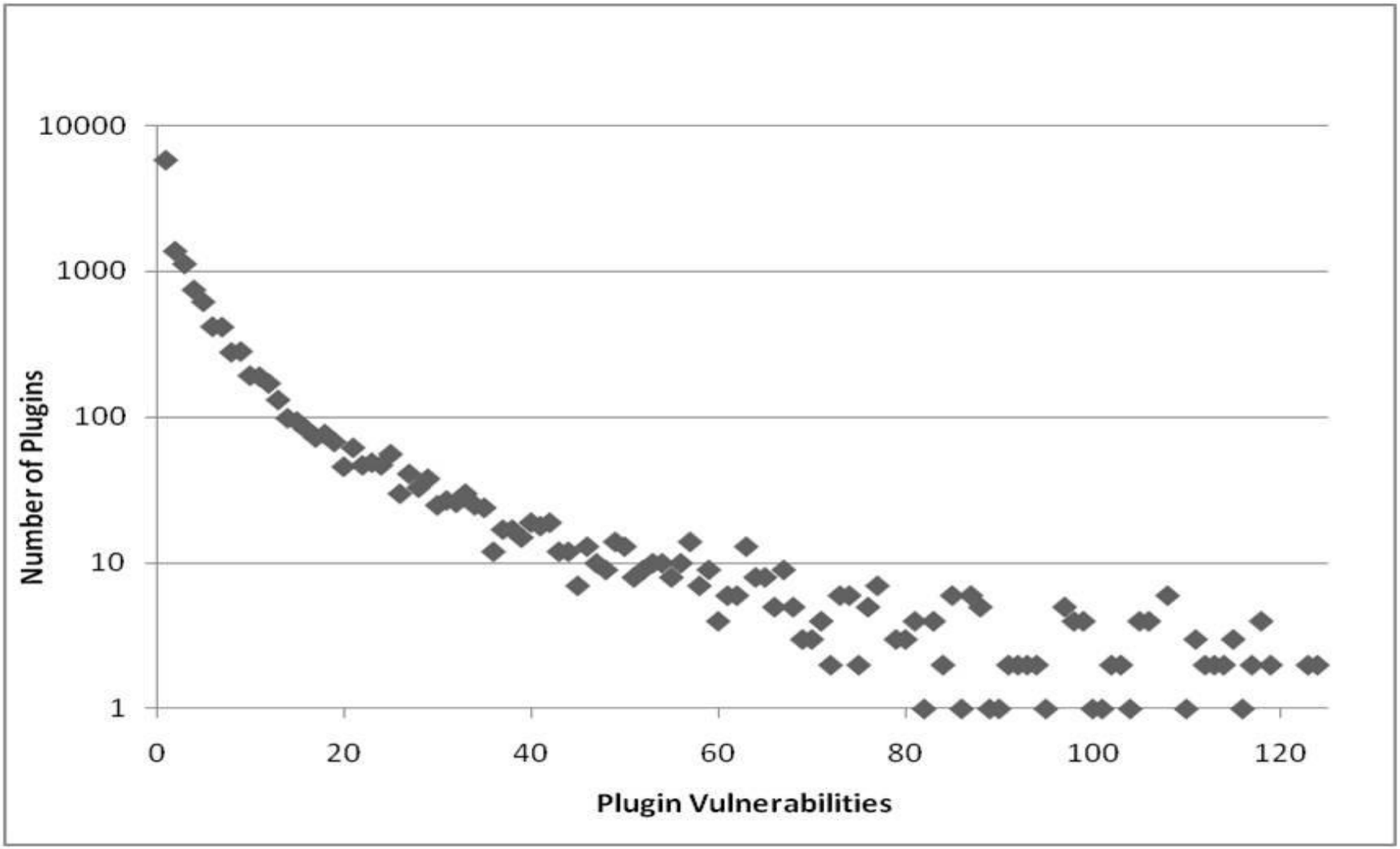
- $\rho = 0.91$.
- Larger plugins are more likely to have vulnerabities.

Core SAVD does not correlate with code size.

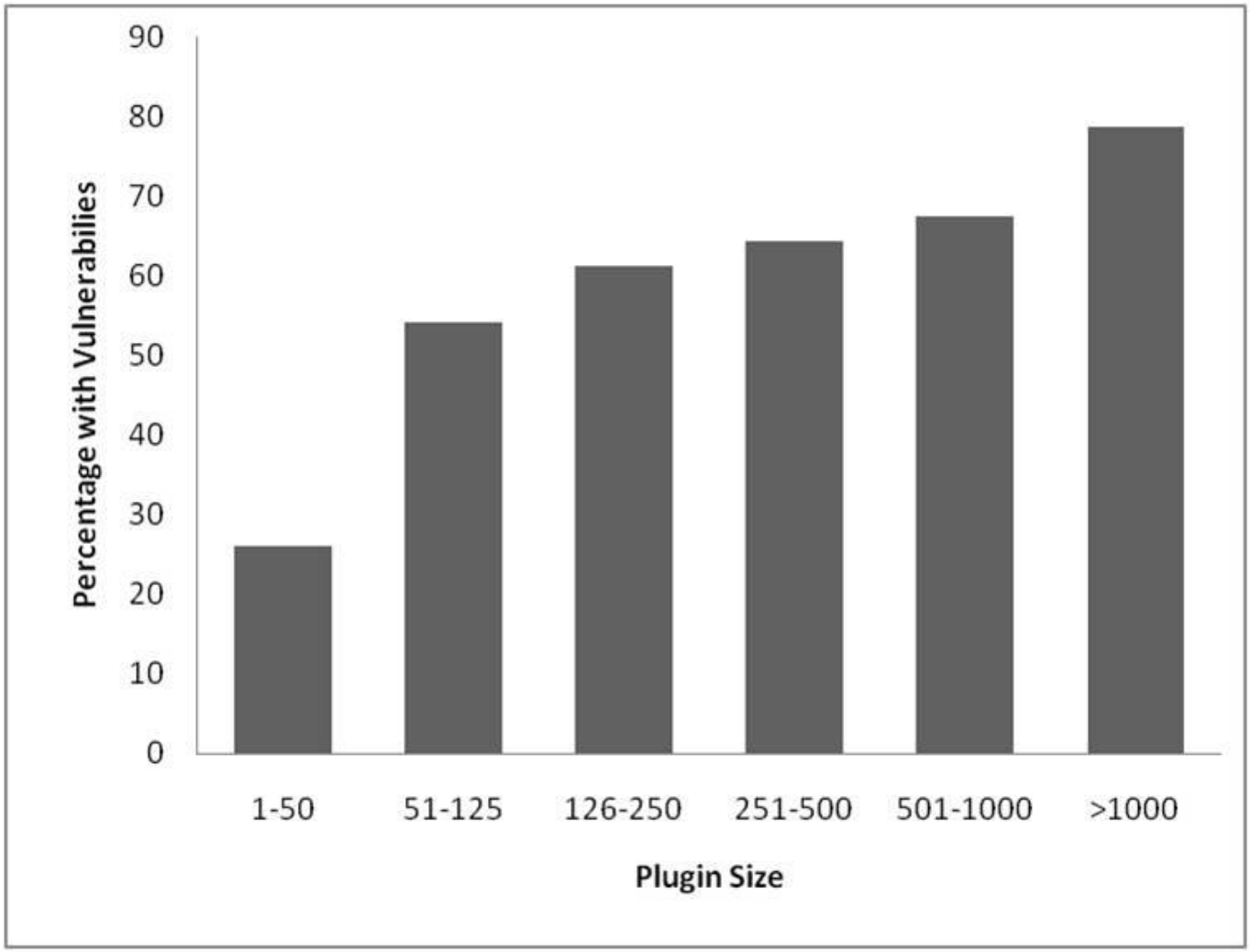
Plugin Size Distribution



Plugin Vulnerability Distribution



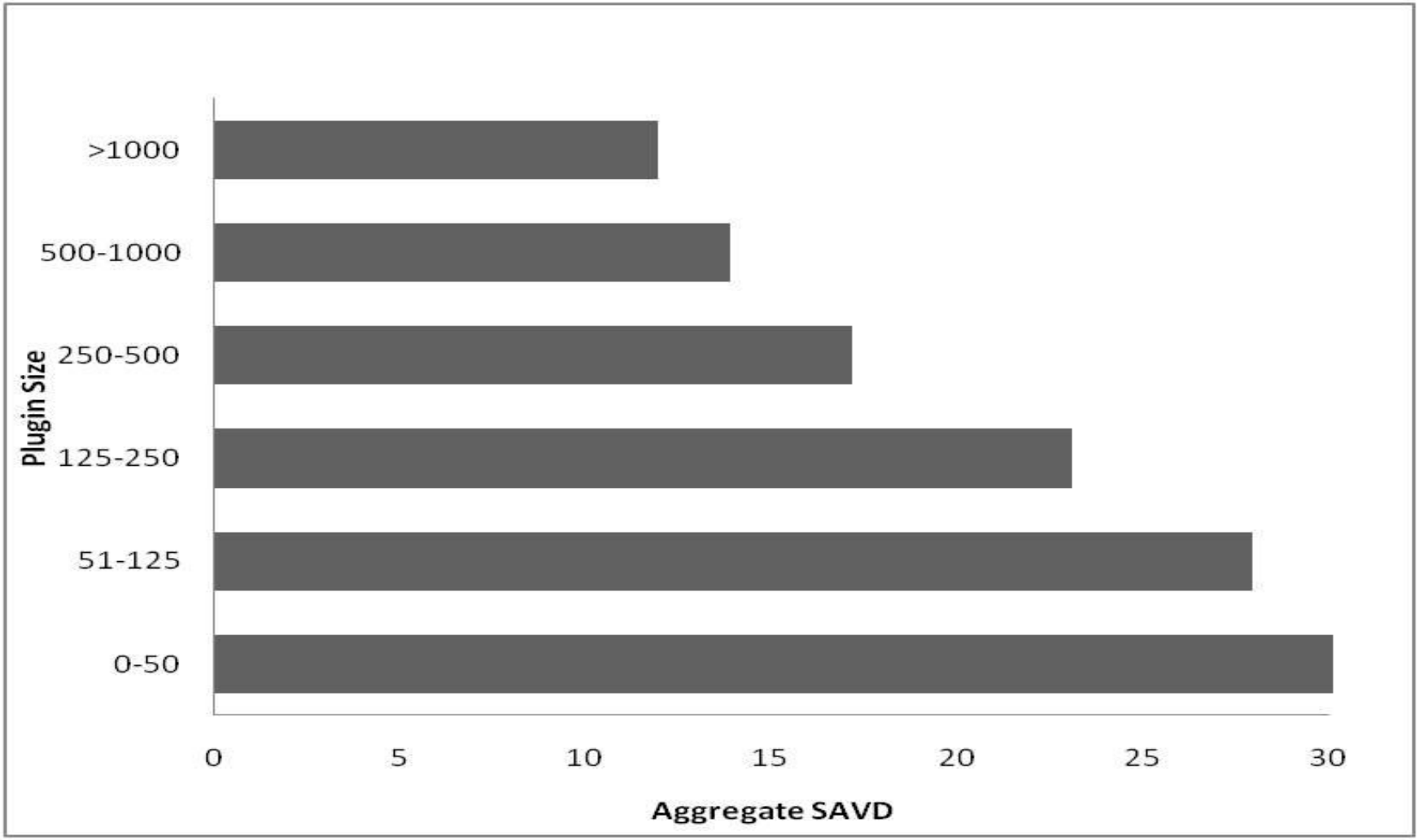
Percentage of Vulnerable Plugins by Size



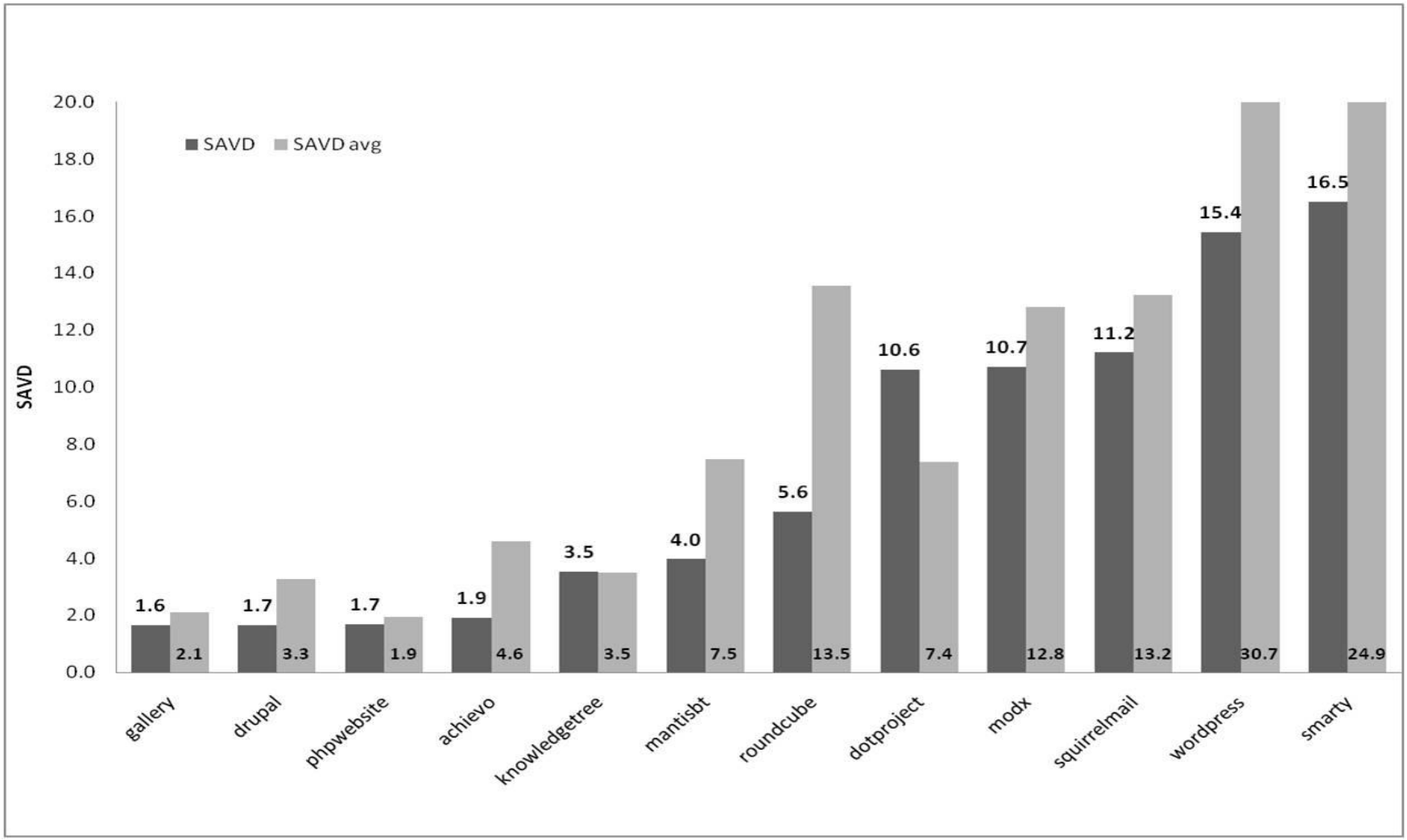
Static Analysis Vulnerability Density (SAVD)

- Number of vulnerabilities found by a static analysis tool per 1000 lines of source code.
  - Fortify SourceAnalyzer 5.8.0
- Aggregate SAVD
  - Use aggregate of source code for all plugins.
  - Total vulnerabilities / Total KSLOC
- Average SAVD
  - Compute SAVD for each plugin individually.
  - Average individual plugin SAVD values.

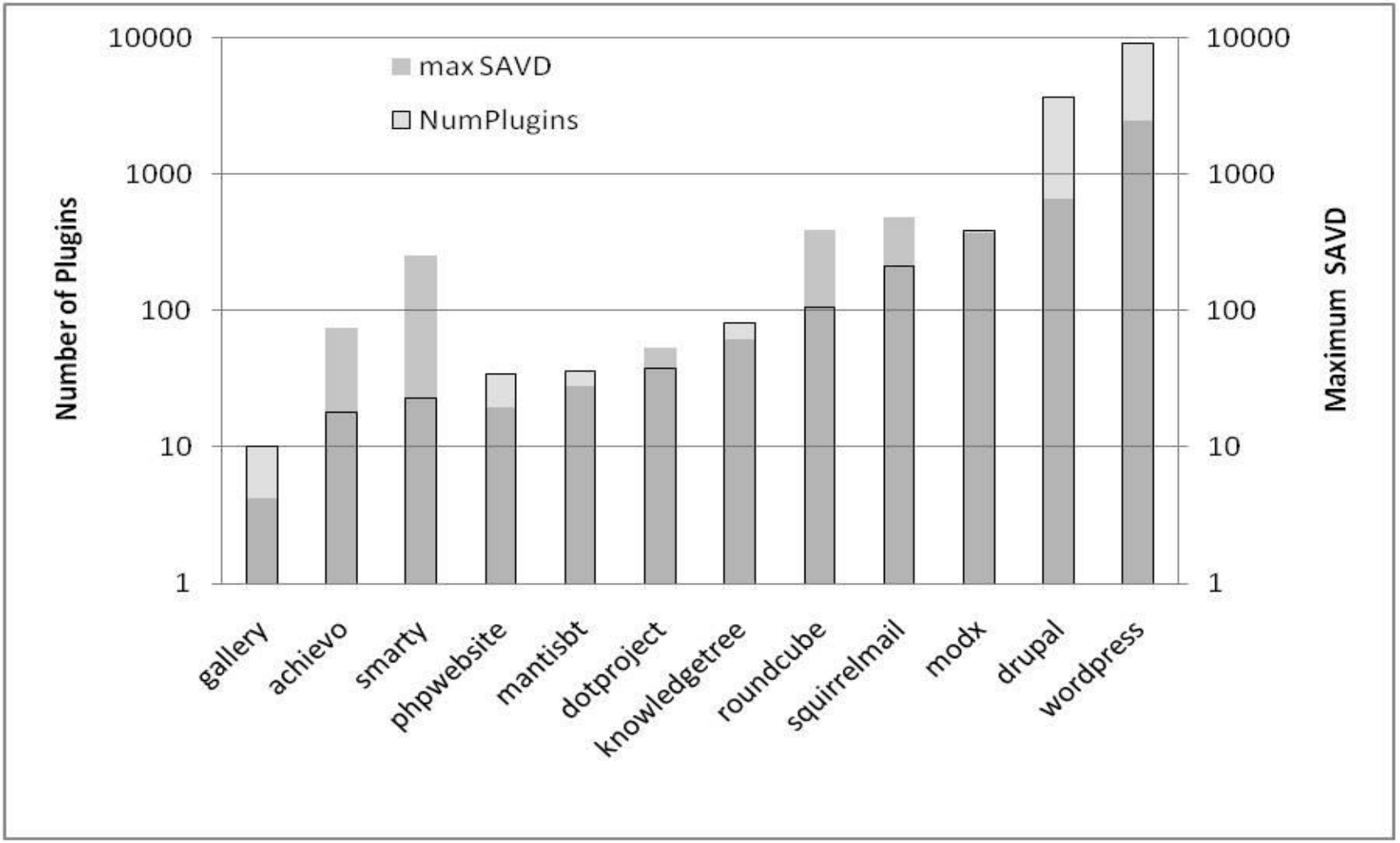
SAVD by Plugin Size



Average vs. Aggregate SAVD of Plugins



Plugin Counts and Maximum Plugin SAVD



Do plugins make your site less secure?

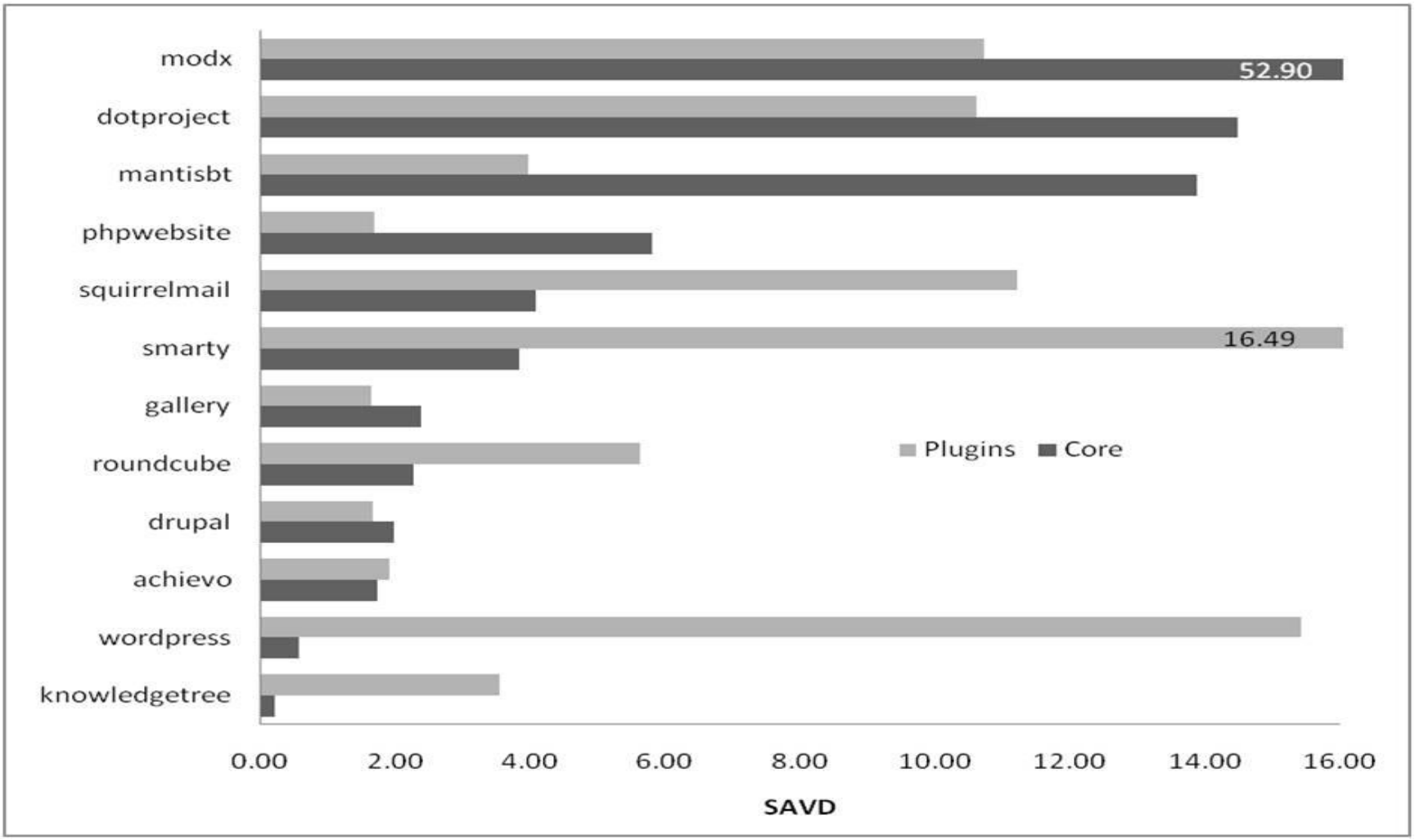
Core code developed by small core team.

- Team experienced with core code over years.
- May or may not be paid full-time developers.
- Most sites have some form of security information.

Plugins developed by many people.

- Wide variety of programming experience.
- Few develop more than one plugin and so have little experience with application compared to core team.
- Few plugins mention security unless a vulnerability has been previously reported.

Core vs. Plugin SAVD



Drupal Core vs. Plugins

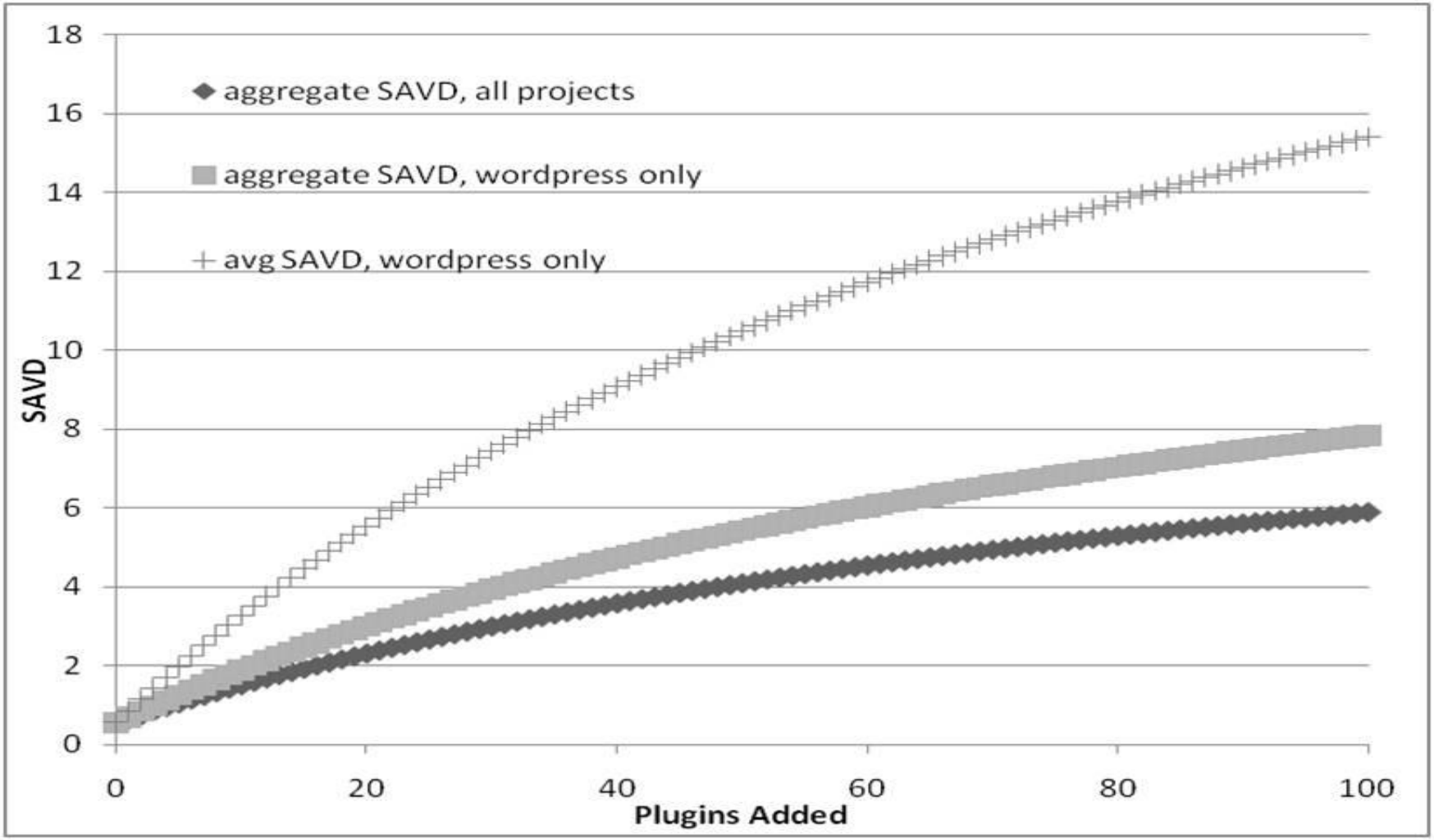
- Drupal tracked both core and plugin vulns since 2006.
- Most popular CMS with 1.58% of web sites including whitehouse.gov

Year	Core	Contrib
2009	8	115
2008	11	64
2007	10	22
2006	12	21

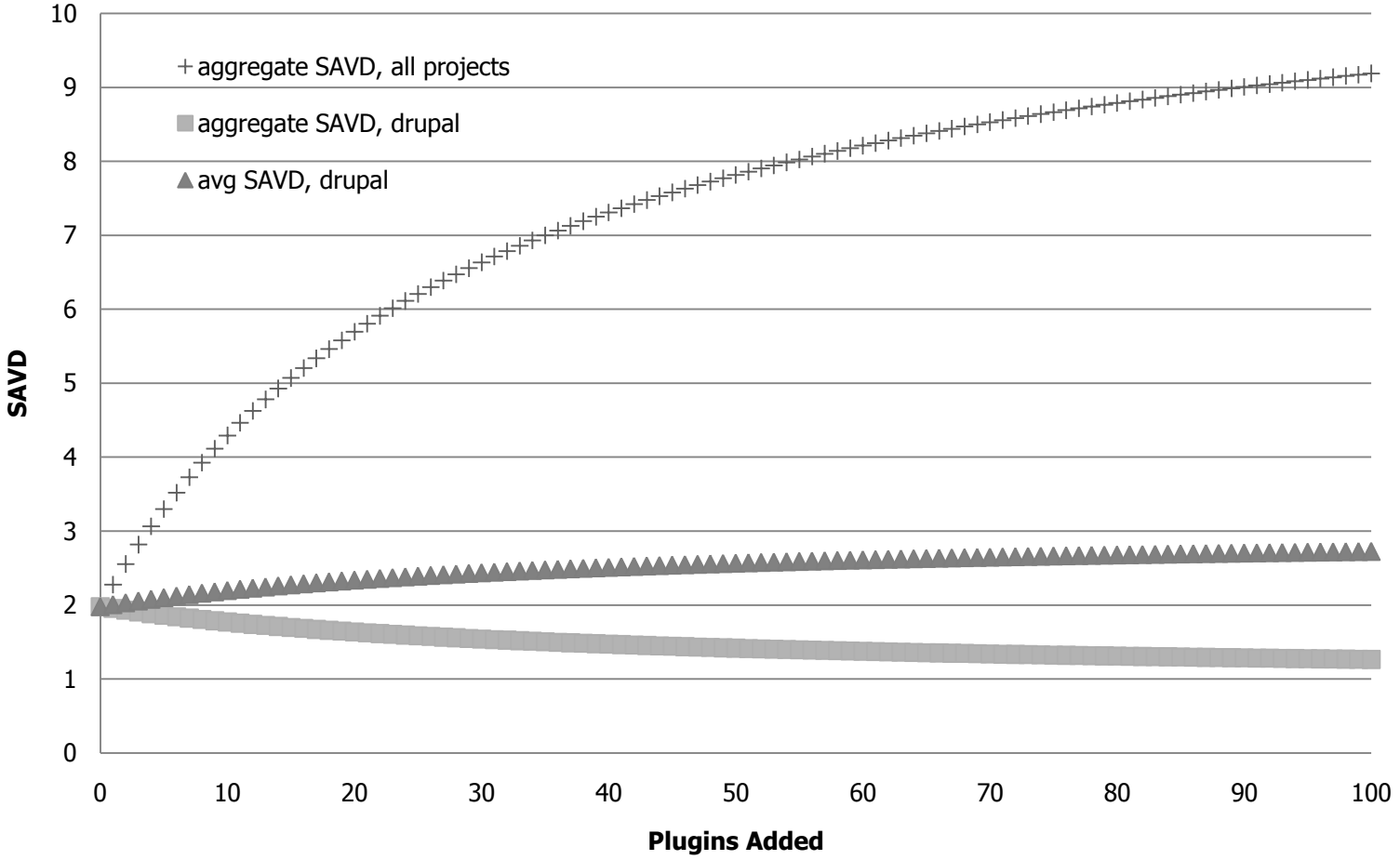
www.drupalsecurityreport.org

- Secure coding documentation.
- XSS Filter API.
- DB API to handle SQLi attacks.
- Input validation API.

WordPress: Effect of Adding Plugins on SAVD



Drupal: Effect of Adding Plugins on SAVD

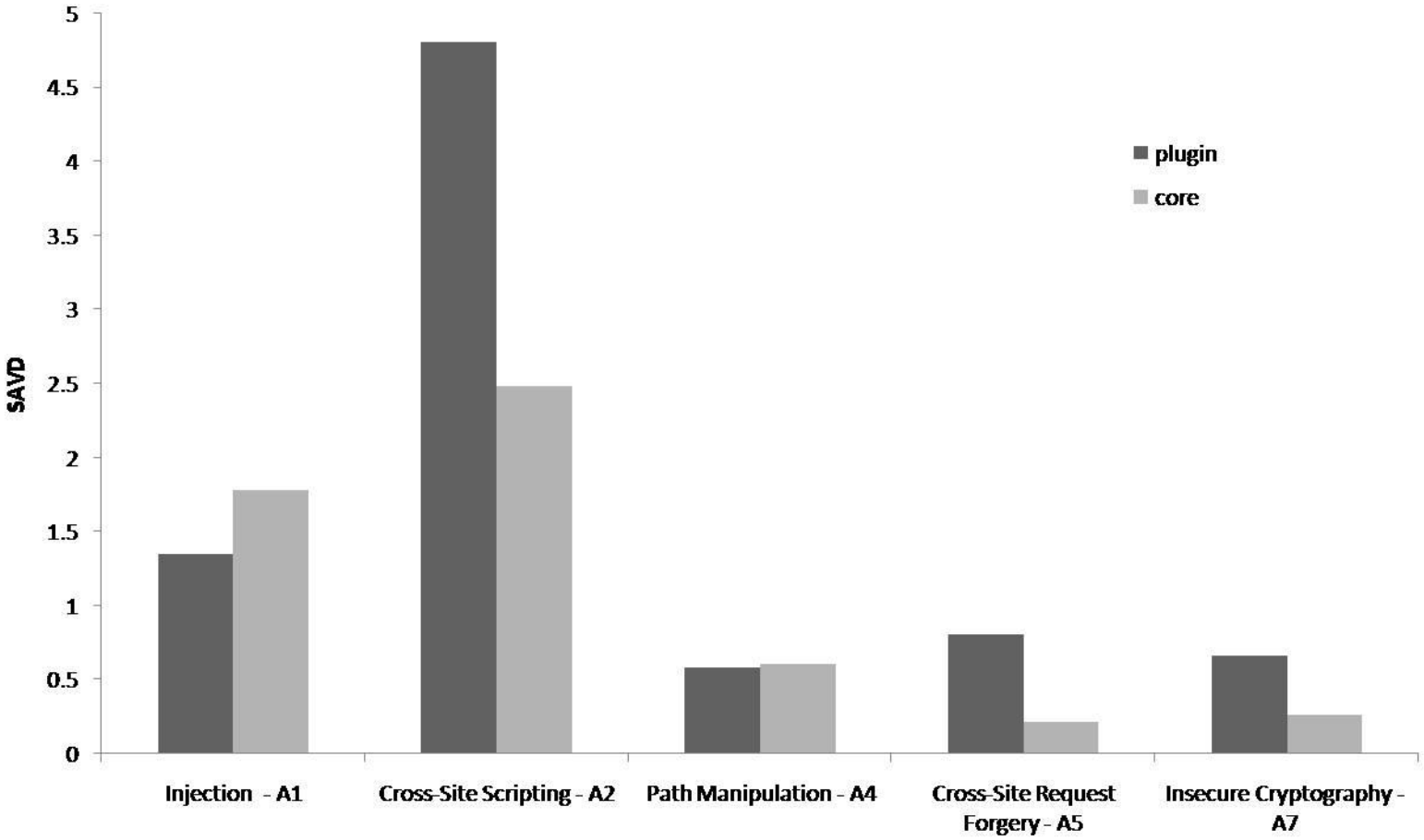


Vulnerability Categories

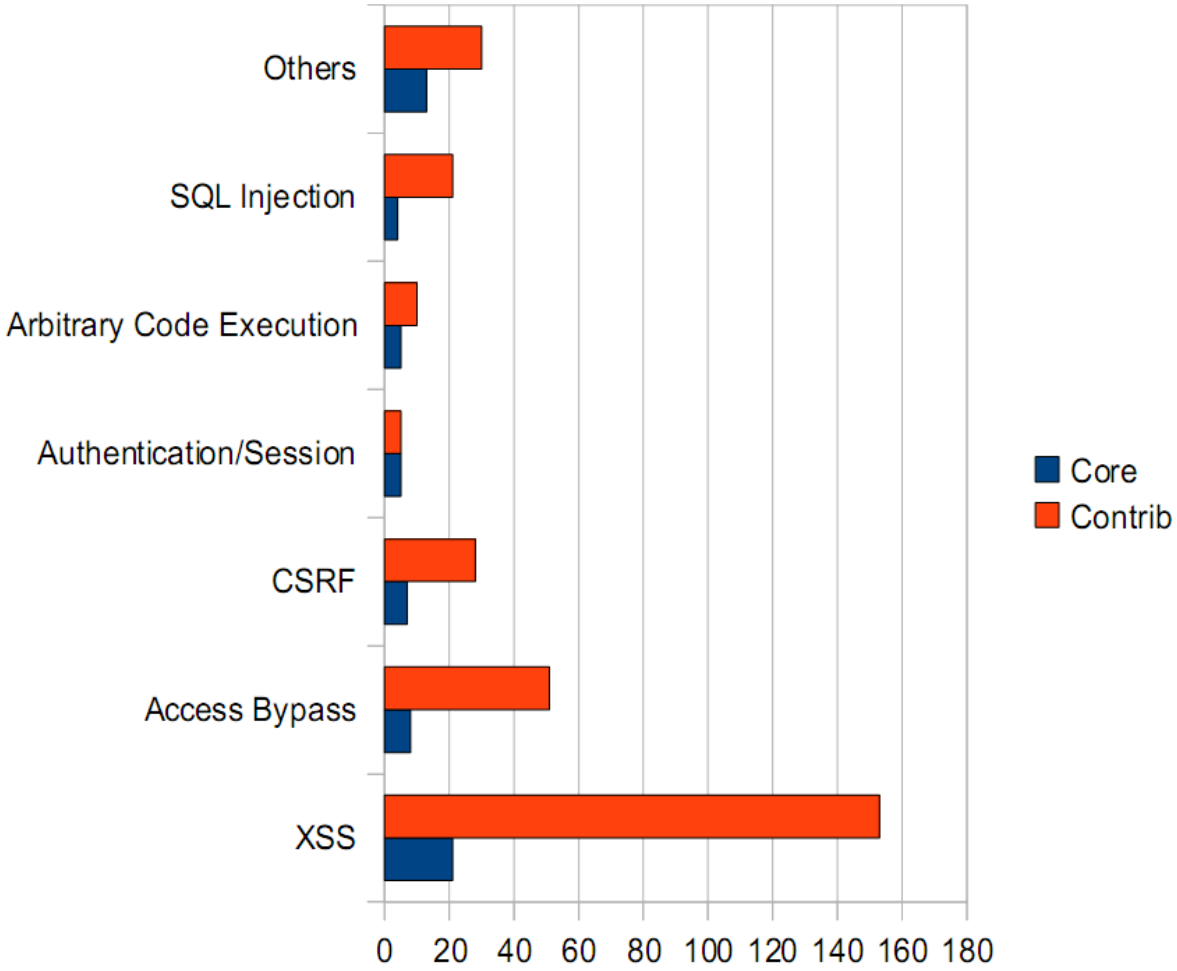
Mapped Fortify categories to OWASP Top 10 2010.

- SCA 5.8 reports 73 categories, only 25 in this code.
- 18 of 25 categories mapped to 5 of OWASP Top 10.
- 7 remaining categories did not map to Top 10.

OWASP Top 10: Core vs. Plugin SAVD

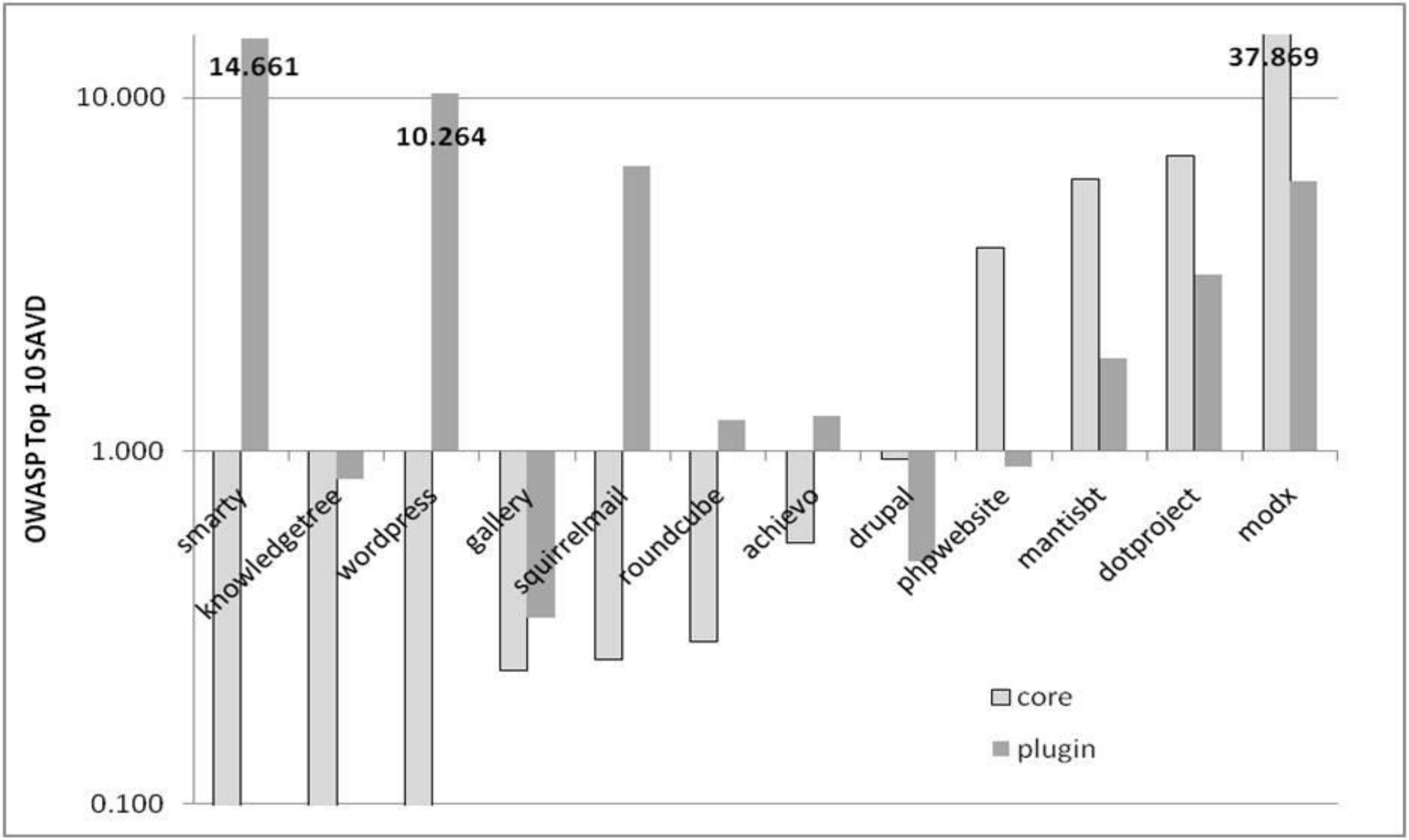


Drupal: Core vs. Plugins by Category



www.drupalsecurityreport.org

OWASP Vulnerabilities: Core vs. Plugin by App



Conclusions

- Plugin code is not always worse than core code.
  - Older apps with more plugins tend to have more secure core code.
  - Security documentation tends to indicate apps with more secure core code.
  - Large number of NVD vulnerabilities does not necessarily indicate poor security.
- Plugin size is important for security
  - 30% of plugins <50 lines have vulnerabities
  - Over 50% of plugins >50 lines have vulnerabilities