State of Software Security Report Volume 2

Jeff Ennis, CEH
Solutions Architect
Veracode

VERACODE

Agenda

- Background – Metrics, Distribution of Applications
- Security of Applications
- Third Party Risk
- Summary

Background – Basis for insights

- For over three years, Veracode has been providing automated security analysis of software to large and small enterprises across various industry segments.
- One of the residual effects is the wealth of security metrics derived from the anonymized data across varied industries and types of applications.
- These metrics offer valuable insights on the quality of application security and issues related to the current state-of-practice and maturity of security in software.
- Veracode was founded in 2006 by application security experts from @stake, Guardent, Symantec, and VeriSign.
- Veracode provides automated security assessment capabilities in the cloud. Automated techniques include static binary analysis and dynamic analysis. Manual test data (if performed) is included in the analysis

The Data Set + Metrics

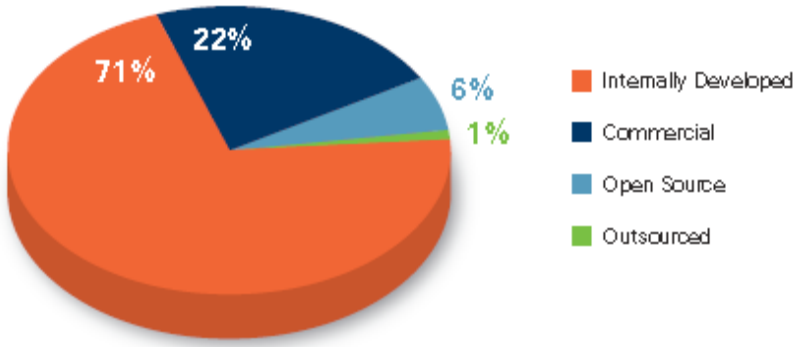
- **Enterprise**
  - Industry vertical (enumerated)
- **Application**
  - Application Supplier Type (internal, purchased, outsourced, open source)
  - Application Type (Web facing / Non-web)
  - Assurance Level (1 to 5)
  - Language (enumerated)
  - Platform (enumerated)
- **Scan**
  - Scan Number
  - Scan Date
  - Lines of Code

- **Metrics**
  - Flaw Count
  - FlawPercent
  - ApplicationCount
  - First Scan Acceptance Rate
  - Veracode Risk Adjusted Score
  - MeanTimeBetweenScans
  - Days to Remediation
  - Scans to Remediation
  - PCI pass/fail
  - SANS Top25 pass/fail
  - OWASP pass/fail
  - Two flavors: ’04 and ’07

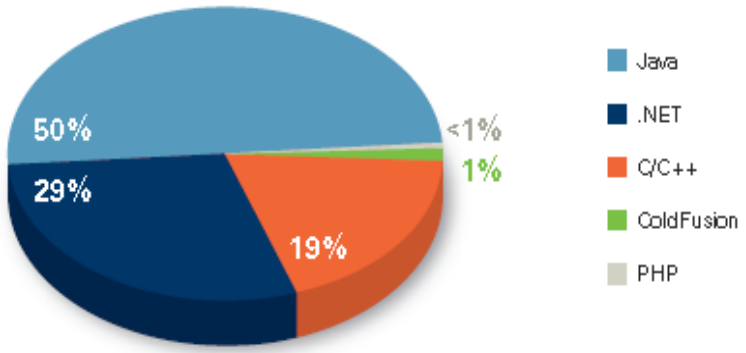
2922 Applications and billions of lines of code

SOSS Volume 2 Data Distribution

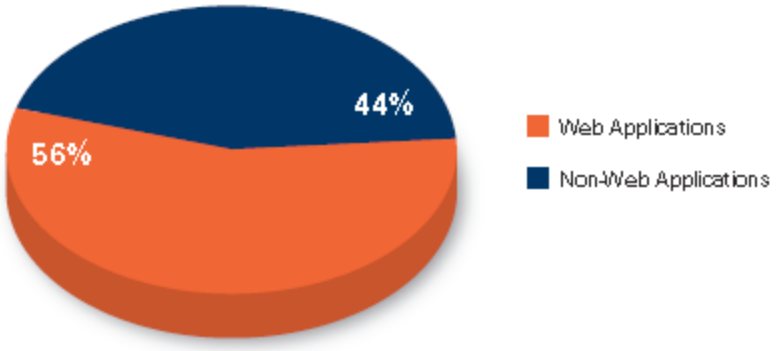
Applications by Supplier



Applications by Language Family



Web versus Non-Web Applications



Business Criticality (and Application Source)

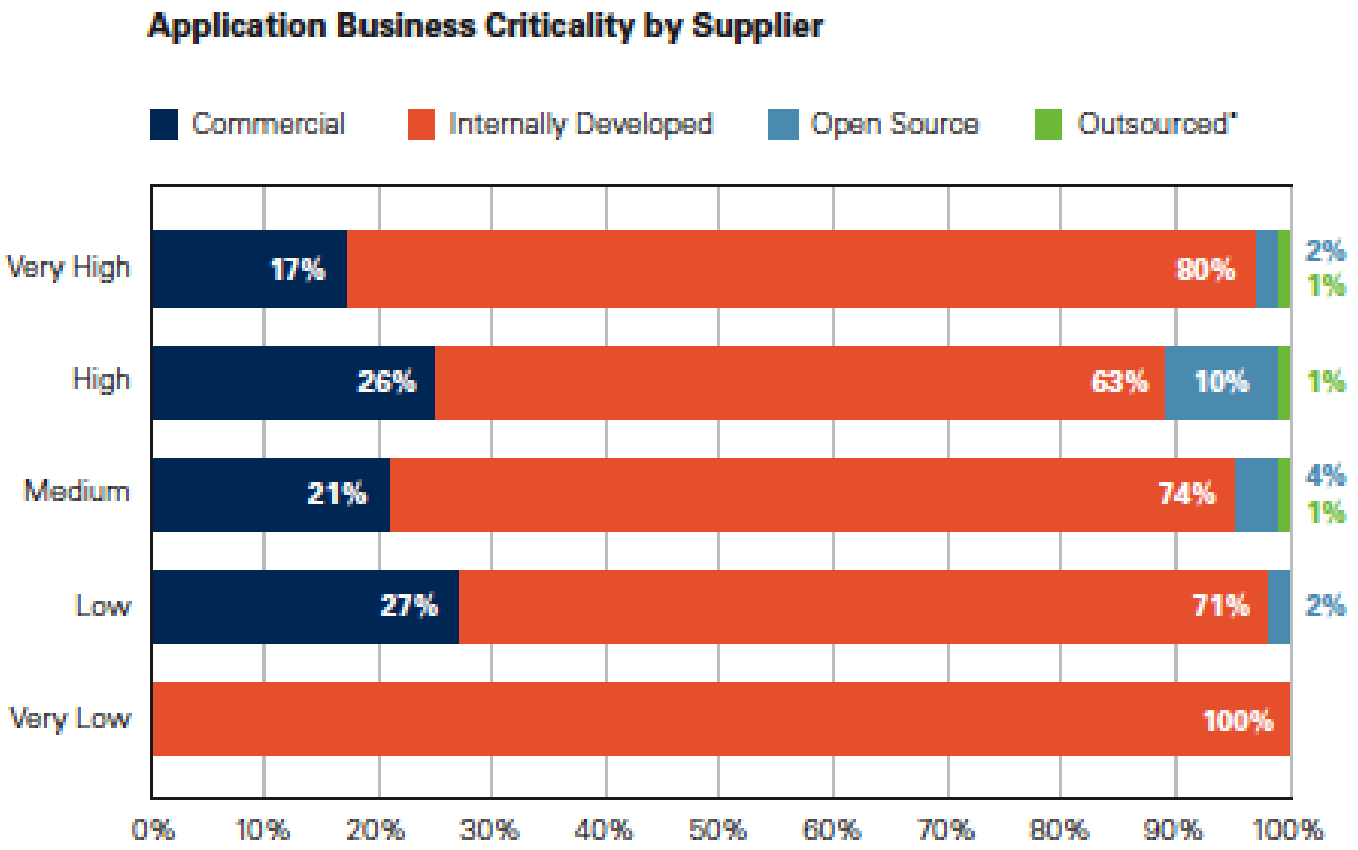
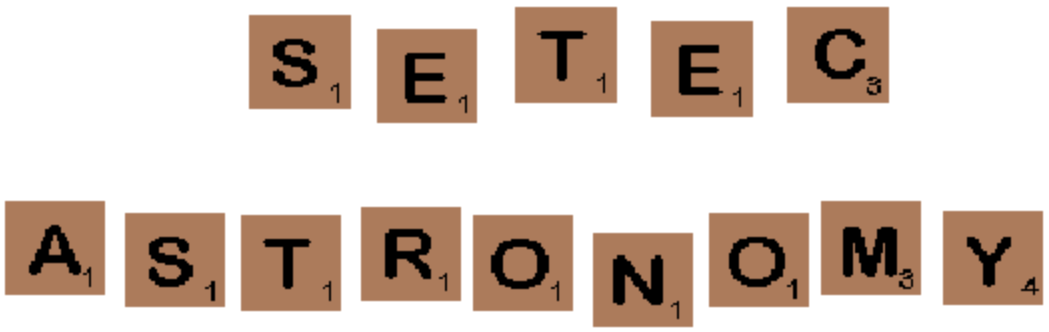


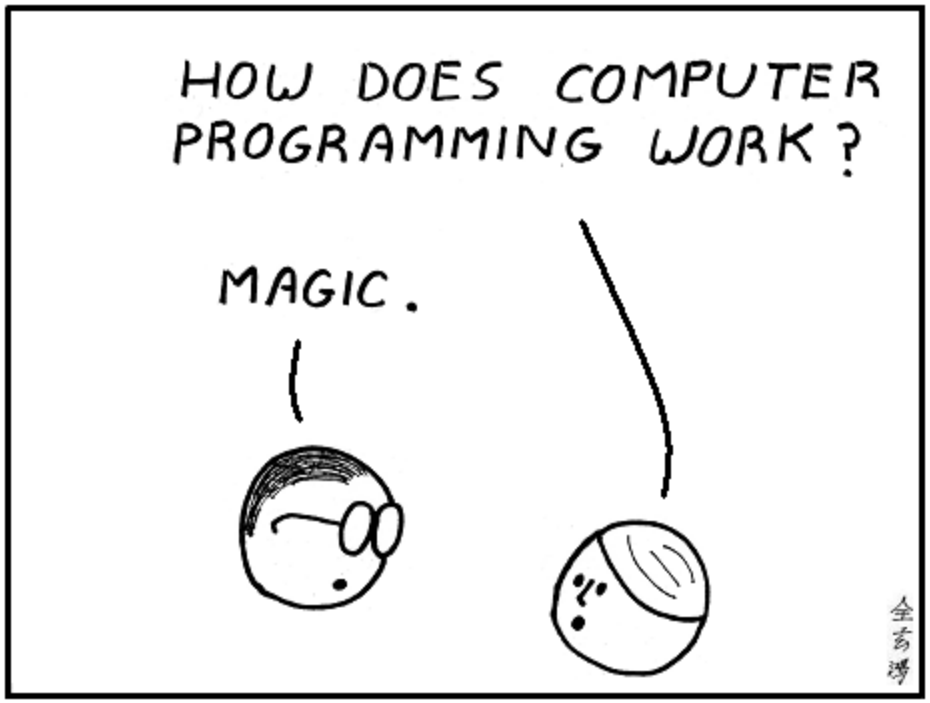
Figure 2: Application Business Criticality by Supplier

(* small sample size)

Security of Applications



Internally Developed – Not So Much



76% of the code components of applications that were labeled as internally developed were third-party components (e.g. open source libraries, commercial third-party libraries etc.)

Application Security – Scanning Results (first submission)

The majority of software (provided by customers for scanning)

_____ **Secure (Pass)**

_____ **Insecure (Fail)**

More than Half of Software Failed

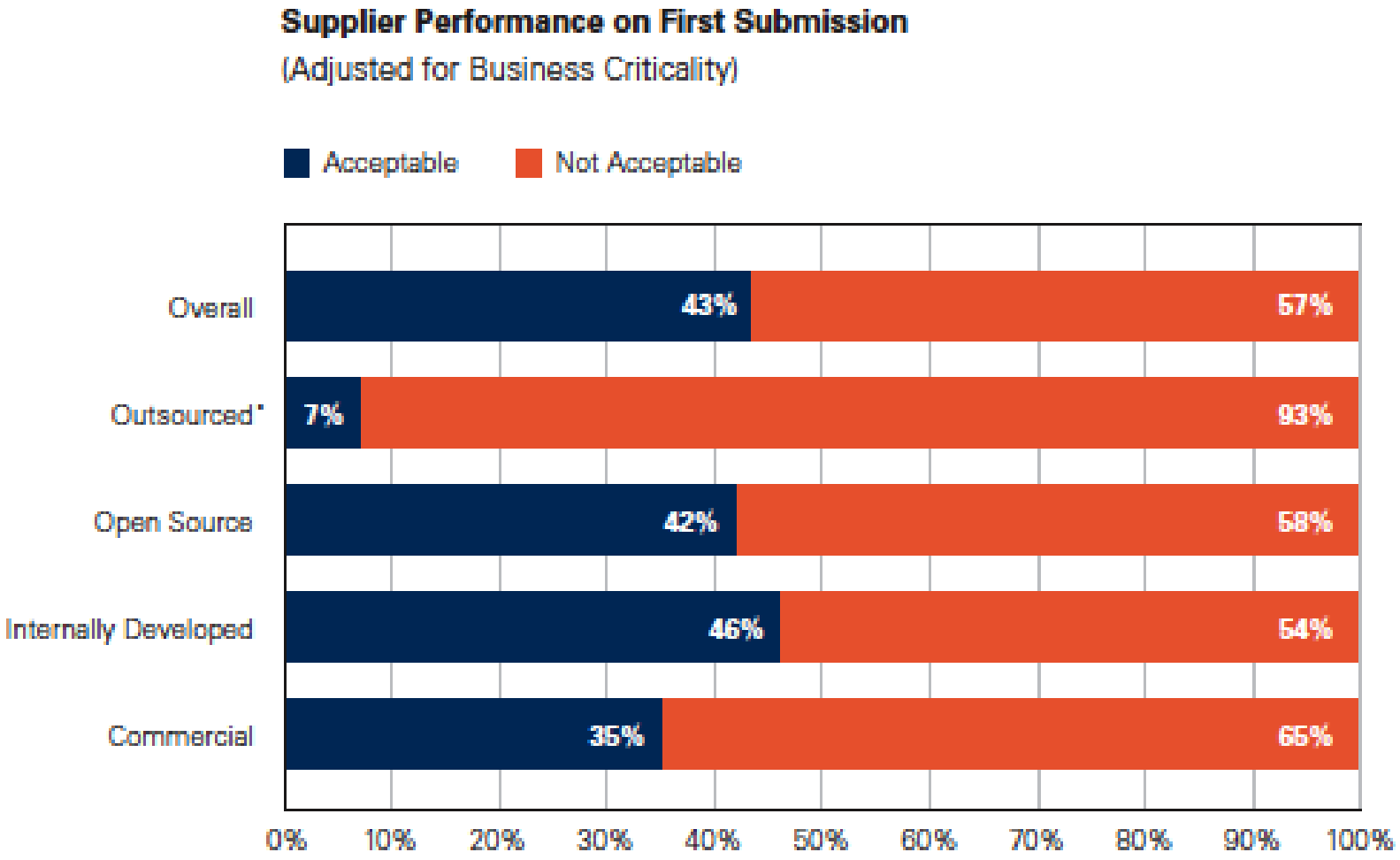


Figure 3: Supplier Performance on First Submission (Adjusted for Business Criticality)

Majority compliant with OWASP Top 10?

8 out of 10 Web Apps Do Not Comply with OWASP Top 10

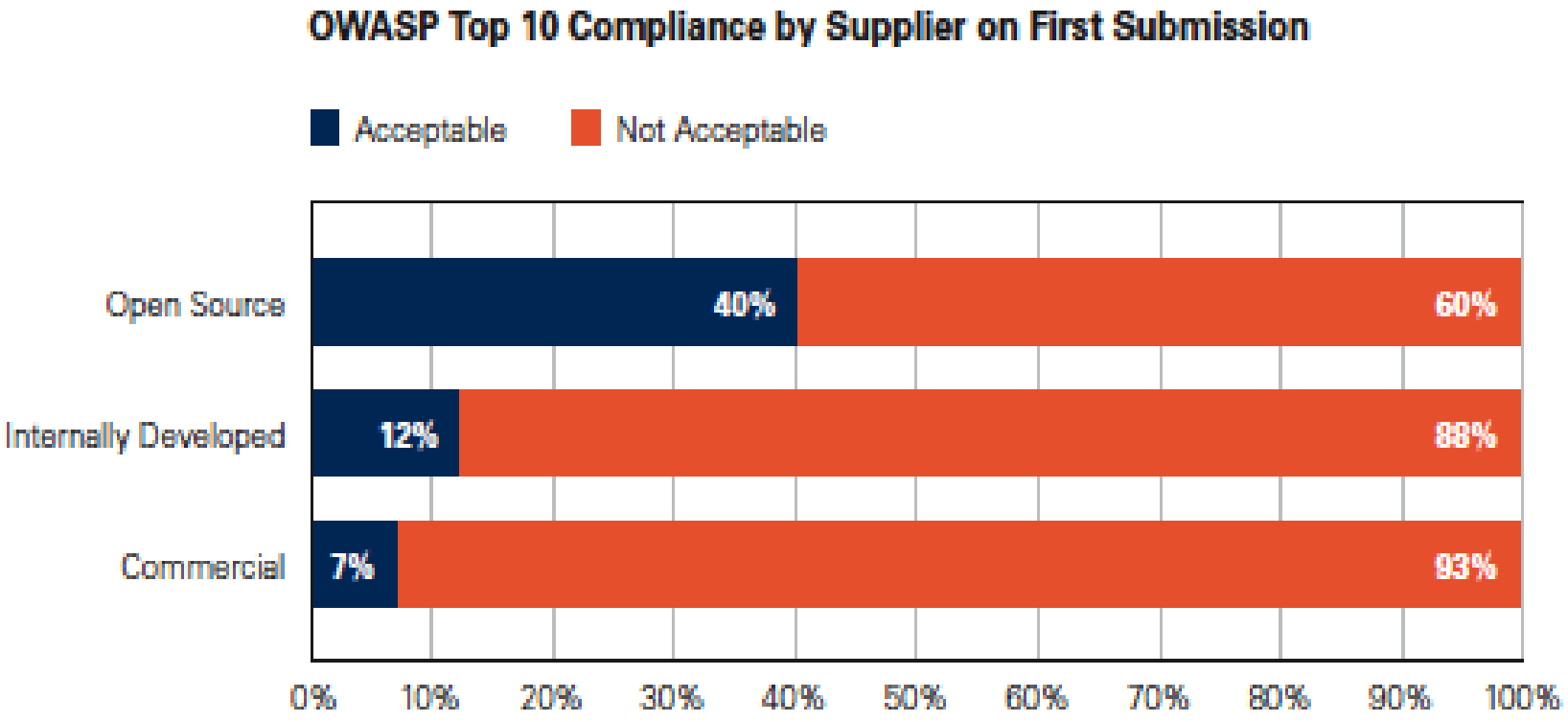


Figure 5: OWASP Top 10 Compliance by Supplier on First Submission

Most Prevalent Vulnerability?

Flaw Percent = Flaw Count / Total

- **SQL Injection**
- **Cross-Site Scripting (XSS)**
- **Cryptographic Issues**
- **CRLF Injection**
- **Buffer Overflow**

Cross-site Scripting Remains the Most Prevalent

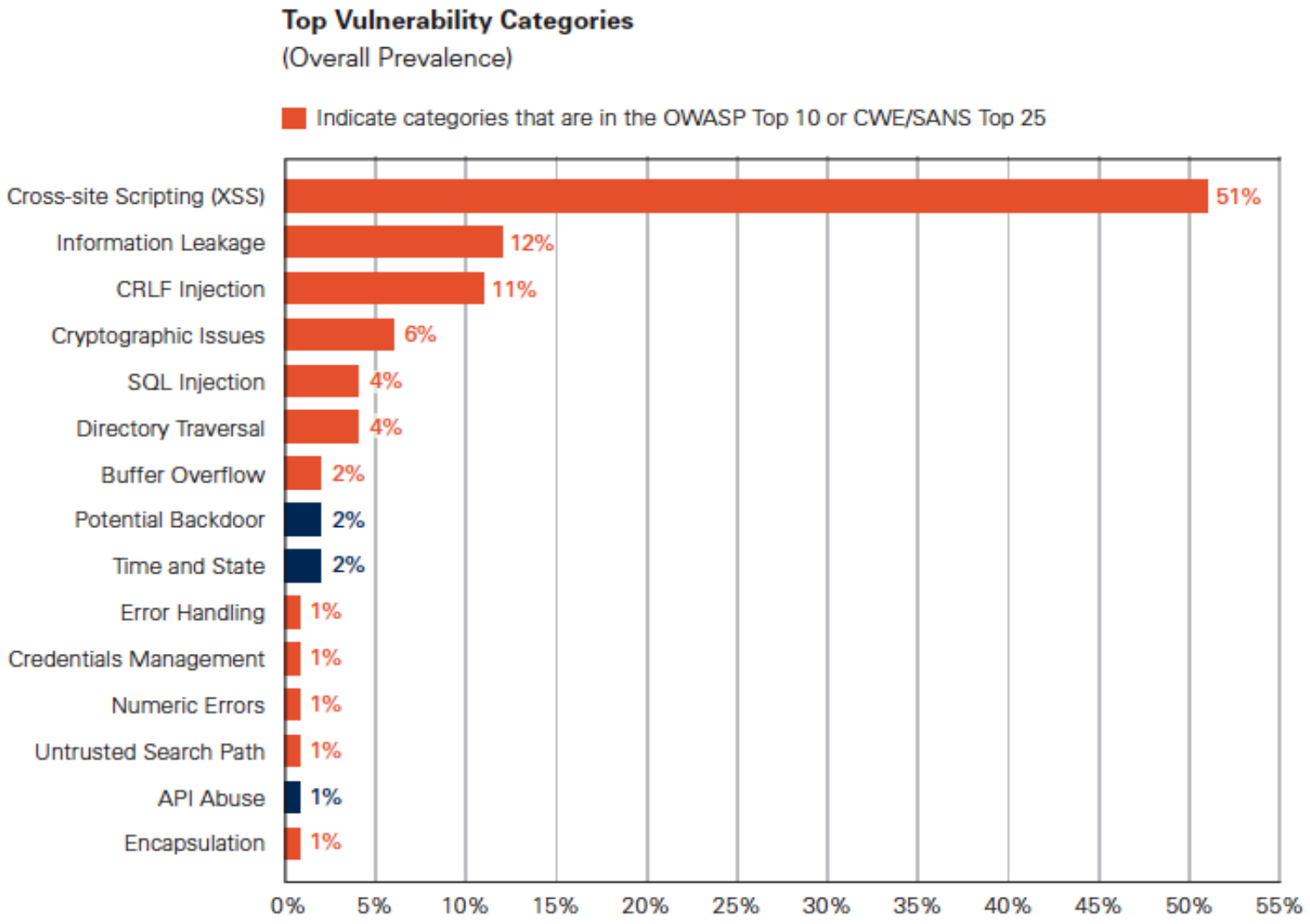


Figure 13: Top Vulnerability Categories (Overall Prevalence)

Which Language Led in Exposure to XSS?

- Java
- .NET

What is the leading issue regarding C/C++ ?

- Crypto Issues
- Error Handling
- Buffer Overflow

Cross-site Scripting Remains the Most Prevalent

Vulnerability Distribution by Language

Java		C/C++		.NET	
Cross-site Scripting (XSS)	46%	Buffer Overflow	32%	Cross-site Scripting (XSS)	66%
CRLF Injection	17%	Potential Backdoor	21%	Cryptographic Issues	13%
Information Leakage	16%	Error Handling	18%	Directory Traversal	8%
Cryptographic Issues	7%	Numeric Errors	13%	CRLF Injection	4%
Directory Traversal	4%	Buffer Mgmt Errors	7%	Information Leakage	4%
SQL Injection	3%	Cryptographic Issues	3%	Insufficient Input Validation	2%
Time and State	2%	Directory Traversal	2%	SQL Injection	1%
Untrusted Search Path	2%	Dangerous Functions	1%	Credentials Mgmt	1%
Credentials Mgmt	1%	Time and State	<1%	Potential Backdoor	<1%
Encapsulation	1%	Race Conditions	<1%	Time and State	<1%
API Abuse	1%	API Abuse	<1%	Error Handling	<1%
Insufficient Input Validation	<1%	Format String	<1%	OS Command Injection	<1%
Race Conditions	<1%	OS Command Injections	<1%	Buffer Overflow	<1%
OS Command Injection	<1%	Credentials Mgmt	<1%	Untrusted Search Path	<1%
Dangerous Functions	<1%	Untrusted Search Path	<1%	Dangerous Functions	<1%

Table 4: Vulnerability Distribution by Language

VERACODE

No single method of application security testing is adequate by itself

Vulnerability Distribution by Analysis Type

Static		Dynamic		Manual	
Cross-site Scripting (XSS)	52%	Information Leakage	44%	Cross-site Scripting (XSS)	26%
CRLF Injection	11%	SQL Injection	27%	Information Leakage	21%
Information Leakage	11%	Cross-site Scripting (XSS)	26%	Other	12%
Cryptographic Issues	6%	Server Configuration	2%	Cryptographic Issues	11%
Directory Traversal	4%	OS Command Injection	<1%	SQL Injection	11%
SQL Injection	3%	Other	<1%	Authorization Issues	7%
Buffer Overflow	3%	Session Fixation	<1%	Authentication Issues	5%
Potential Backdoor	2%	Cryptographic Issues	0%	Insufficient Input Validation	2%
Time and State	2%	Insufficient Input Validation	0%	Credentials Mgmt	2%
Error Handling	1%	Authentication Issues	0%	Directory Traversal	1%

Table 5: Vulnerability Distribution by Analysis Type

Applications with the Best First-Scan Acceptance Rate?

- Outsourced
- Open Source
- Internally Developed
- Commercial

VERACODE

Internal Apps have Best First Scan Acceptance Rate

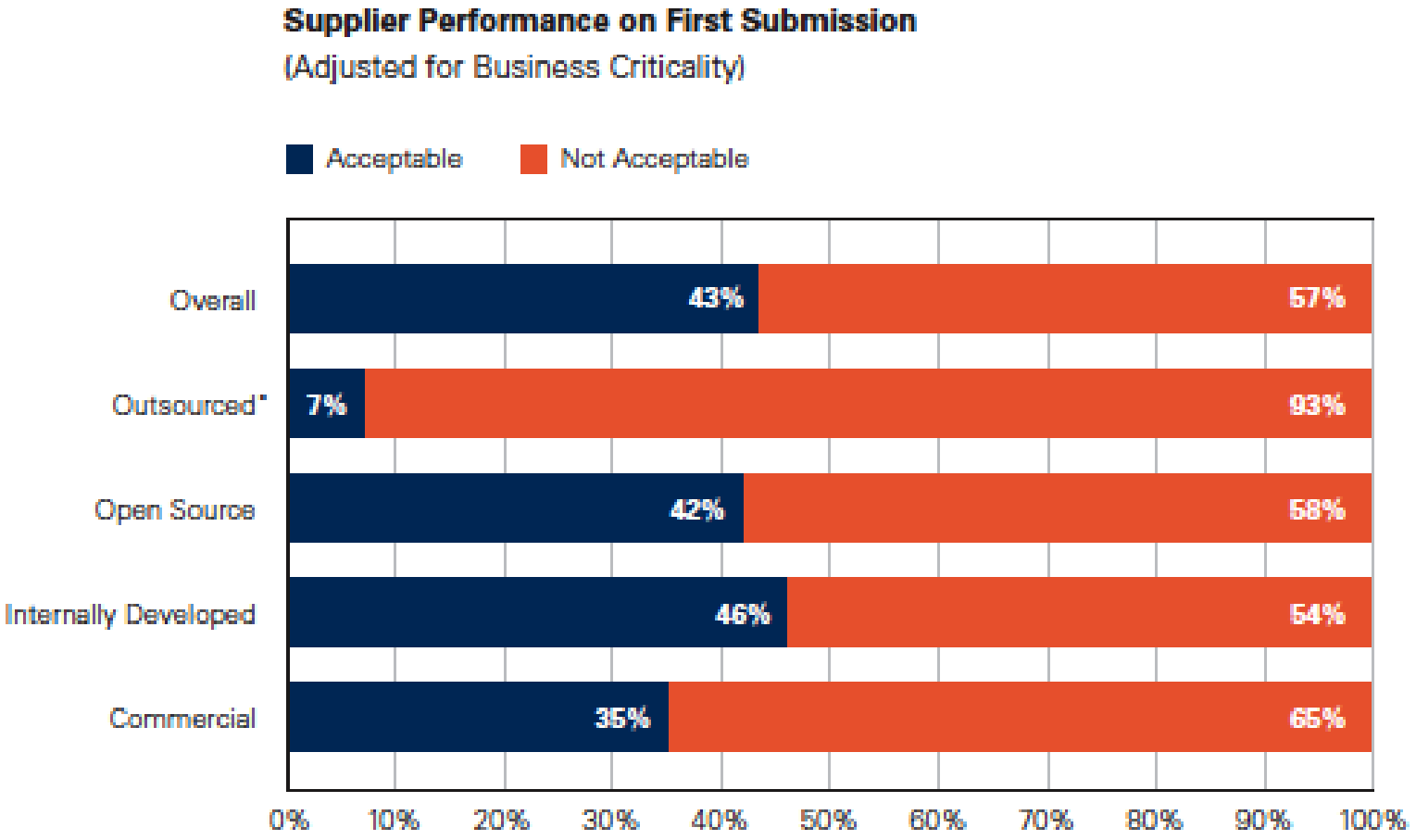


Figure 3: Supplier Performance on First Submission (Adjusted for Business Criticality)

Shortest Remediation Cycle?

- **Outsourced**
- **Open Source**
- **Internally Developed**
- **Commercial**

Developers Repaired Security Vulnerabilities Quickly

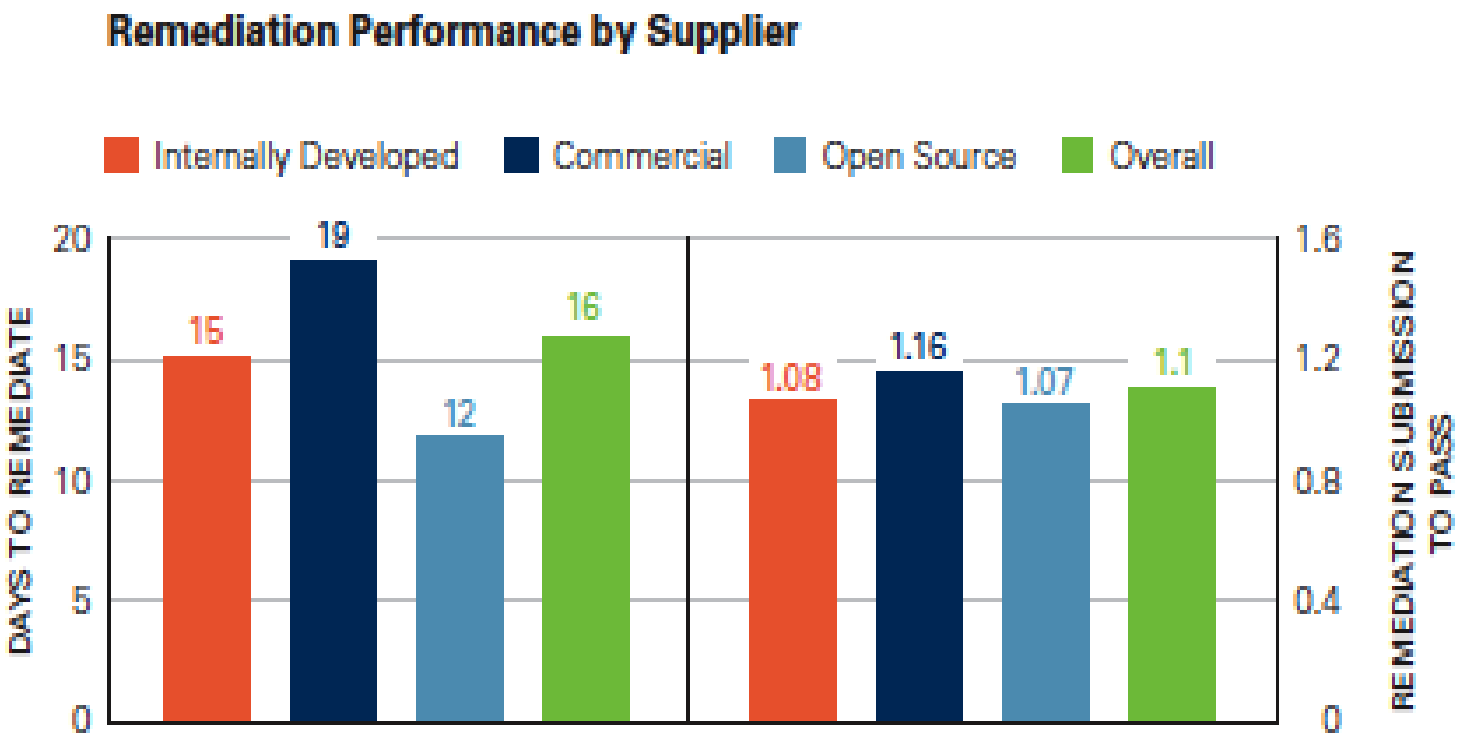


Figure 4: Remediation Performance by Supplier

Financial Sector Spotlight

Security quality is not commensurate with Business Criticality for Financial Industry applications

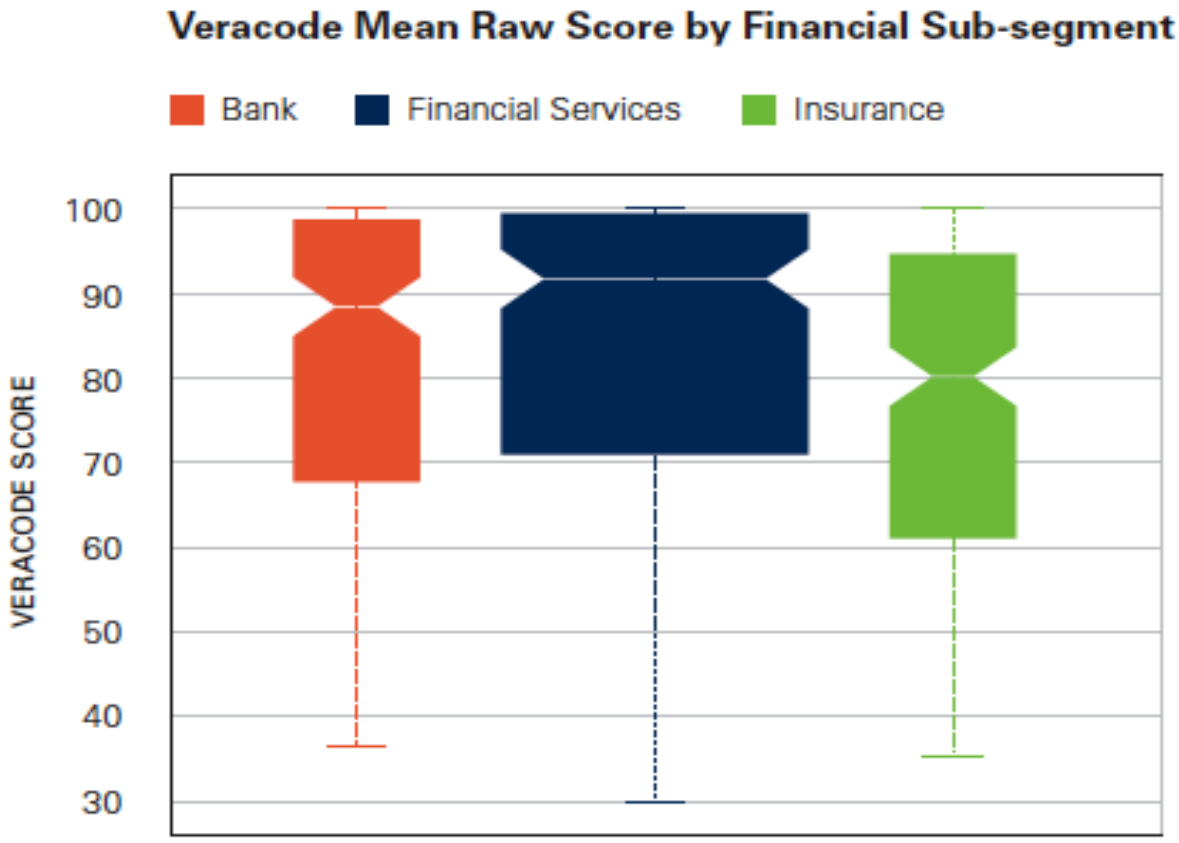
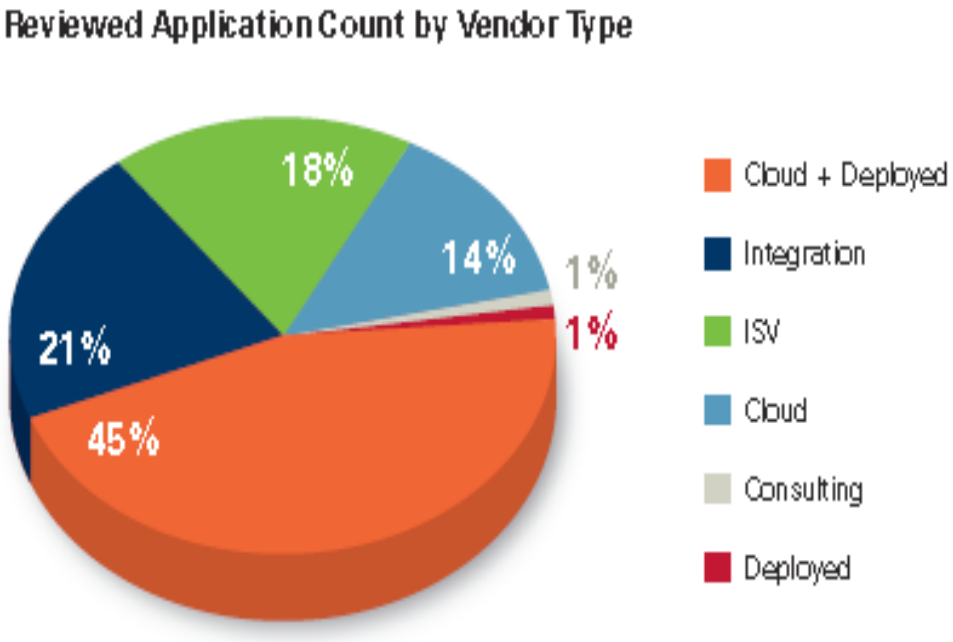


Figure 15: Veracode Mean Raw Score by Financial Sub-segment

Banks, insurance, and financial services companies have among the best raw security quality scores.

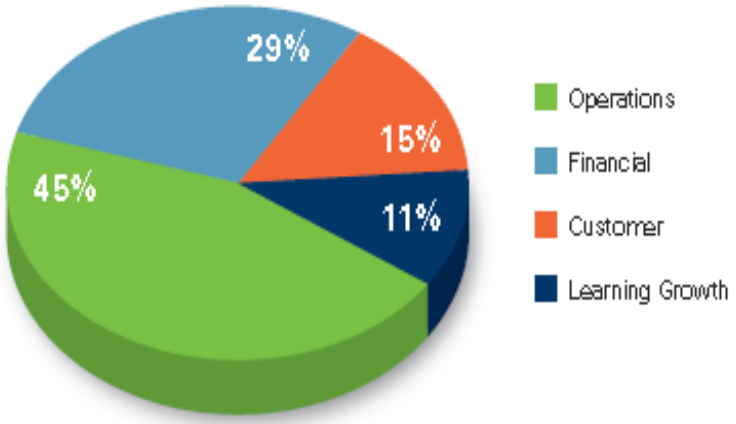
Third-Party Assessments

Suppliers of Cloud/Web Apps Most Frequently Subjected to Third-party Risk Assessments

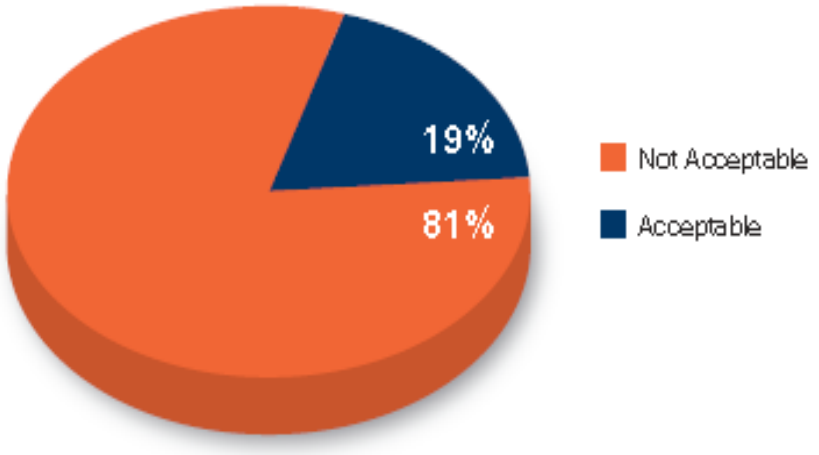


Third-party Risk Assessments (more)

Requested Third-party Assessments by Application Purpose



Third-party Assessments: Performance Upon Initial Submission



Three-quarters of all third-party assessments required less than 11 days to achieve acceptable levels of security quality.

Trends and Conclusions

- Lower than average SQL Injection and XSS prevalence in an app is an indicator that the development team understands secure coding.
- Static analysis is being performed in addition to dynamic analysis on web applications.
- First mobile app risks appearing in the wild. Both vulnerabilities such as the PDF iOS 4 vulnerability used by jailbreakme.com and mobile apps with trojan functionality.
- Backdoor (likely intentional) in critical software such as Seimens SCADA product discovered and exploited
- Uptick in cloud based software being tested
- Overall, older platforms getting more mature SDLC as developers take to mobile and cloud

Thank You

www.veracode.com

VERACODE