Software security, by the numbers
Why are we here?
Chris Wysopal, CTO & Co-Founder

• 15+ years focused solely on application security
• One of the original security researchers from mid 90’s. Was part of hacker think tank, The L0pht.
• Founded Veracode to build automated application security testing
• My son now keeps me busy installing Minecraft mods.
What is the SoSS report?

SoSS is the “BEFORE”

Breach Reports are the “AFTER”
What’s so special about this?

• Anonymized data from our customers to illustrate AppSec trends
  - More than 200,000 application assessments performed over the past 18 months
What questions can we answer?

• Which industries are doing the best job of reducing application-layer risk?
• Do I have more serious vulnerabilities than my peers?
• What percentage of vulnerabilities do my peers remediate?
• How many of our applications should pass the OWASP Top 10 when initially assessed?
• What are the Top 10 most common vulnerabilities in our vertical?
• Which programming languages are my peers using?
What are the headlines?
WHO DOES BEST AGAINST INDUSTRY STANDARDS?

What is the percentage of applications that fail an OWASP Top 10 policy upon initial submission?
WHO FIXES THE MOST VULNERABILITIES?

What is the percentage of known vulnerabilities remediated by each industry vertical, in order to reduce application layer risk?

- Manufacturing: 81%
- Financial Services: 65%
- Retail + Hospitality: 60%
- Technology: 50%
- Healthcare: 43%
- Government: 27%
Industry comparison: flaw profiles
Security quality on first scan

<table>
<thead>
<tr>
<th>Industry</th>
<th>Compliant</th>
<th>Out of Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial Services</td>
<td>42%</td>
<td>58%</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>35%</td>
<td>65%</td>
</tr>
<tr>
<td>Technology</td>
<td>32%</td>
<td>68%</td>
</tr>
<tr>
<td>Healthcare</td>
<td>31%</td>
<td>69%</td>
</tr>
<tr>
<td>Retail and Hospitality</td>
<td>30%</td>
<td>70%</td>
</tr>
<tr>
<td>Other</td>
<td>30%</td>
<td>71%</td>
</tr>
<tr>
<td>Government</td>
<td>24%</td>
<td>76%</td>
</tr>
</tbody>
</table>
Even the best performing industry sector fails for 3 out of 5 applications.
Programming languages used

Financial Services
Government
Healthcare
Manufacturing
Other
Retail and Hospitality
Technology
All Industries

.NET
C++
Java
Android
iOS
Classic ASP
ColdFusion
PHP
JavaScript
Programming language is not a clear indicator of security quality.
Flaw density

- Manufacturing
- Technology
- Financial Services
- Healthcare
- Other
- Government
- Retail & Hospitality
- All Industries

Flaw Density
- All Flaws (Flaws/MB)
- Very High and High Severity Flaws (Flaws/MB)

Bar chart showing flaw density across different industries.
Flaw density is only useful for benchmarking or tracking a particular application over time.
Top vulnerabilities by industry
Industry comparison: remediation profiles
Flaw remediation rates

• Wide spread in % of flaws found vs. fixed
• May be due to external factors (industry regulations, competitive landscape) and/or security policies
Remediation by flaw density

- Small improvement in flaw density overall, greater improvement in very high and high severity flaws
Impact of remediation coaching

- For applications that used a readout, the development team fixed more than 2.5x the average # of flaws per megabyte
How safe is the software I buy?
Sources of software (IDG survey)

- Internally developed: 39%
- Sourced from commercial software vendor: 33%
- Outsourced (i.e. developed by third party): 29%
Commercial quality on first scan

- Code you purchase is just as likely (maybe more likely) to contain vulnerabilities as code you build.
- Most organizations don’t hold their suppliers accountable for security quality in the same way they do for functionality or performance.
Three characteristics of a successful program
#1: Metrics driven

“What gets measured gets managed.”

—Peter Drucker
#2: Remediation focused

2.5x!
#3: Habit forming
#3: Habit forming

MONSTERS PREVENT CODE ROT!
THANK YOU
Appendix slides
Financial Services summary

Includes: Banking, Finance, Insurance

• **Initial quality:**
  - 42% of apps passed OWASP Top 10 on first scan—but almost 3 in 5 still fail

• **Flaw density:**
  - 36 flaws/MB overall
  - 5 very high or high severity flaws/MB

• **Remediation:**
  - Fixed 65% of flaws found during 18 month period

• **SQL Injection prevalence:**
  - 29% of applications had at least one SQL Injection vulnerability, the second lowest

---

**Language Mix**

- .NET 42%
- Java 48%
- Android 1%
- ColdFusion 0%
- JavaScript 0%
- Other 1%

**App Vuln Prevalence**

- Code Quality: 65%
- Crypto Issues: 60%
- Info Leakage: 58%
- CRLF Injection: 52%
- Cross-Site Scripting: 49%
- Directory Traversal: 48%
- Insufficient Input Validation: 41%
- SQL Injection: 29%
- Credentials Mgmt: 25%
- Time & State: 23%
Manufacturing summary

Includes: Manufacturing, Aerospace

- **Initial quality:** 35% of apps passed OWASP Top 10 on first scan – but almost 2/3 still fail

- **Flaw density:** 352 flaws/MB overall
  - 54 very high or high severity flaws/MB

- **Remediation:** Fixed 81% of flaws found during 18 month period

- **SQL Injection prevalence:** 31% of applications had at least one SQL Injection vulnerability

---

**Language Mix**

- Java 37%
- .NET 36%
- C++ 10%
- ColdFusion 9%
- Classic ASP 4%
- iOS 2%
- Other 2%
- PHP 1%
- JavaScript 1%
- Android 0%

**App Vuln Prevalence**

- Code Quality 56%
- Crypto Issues 51%
- Info Leakage 49%
- CRLF Injection 45%
- Cross-Site Scripting 45%
- Directory Traversal 40%
- Insufficient Input Validation 33%
- SQL Injection 31%
- Time & State 24%
- Credentials Mgmt 17%
Technology summary

Includes: Technology, Telecommunications, Electronics, Software, Security Products & Services, Consulting

• **Initial quality:** 3
  - 32% of apps passed OWASP Top 10 on first scan

• **Flaw density:** 5
  - 83 flaws/MB overall
  - 29 very high or high severity flaws/MB

• **Remediation:** 5
  - Fixed 50% of flaws found during 18 month period

• **SQL Injection prevalence:** 3
  - 28% of applications had at least one SQL Injection vulnerability

---

Language Mix

<table>
<thead>
<tr>
<th>Language</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Java</td>
<td>45%</td>
</tr>
<tr>
<td>.NET</td>
<td>30%</td>
</tr>
<tr>
<td>PHP</td>
<td>6%</td>
</tr>
<tr>
<td>C++</td>
<td>4%</td>
</tr>
<tr>
<td>iOS</td>
<td>6%</td>
</tr>
<tr>
<td>Android</td>
<td>6%</td>
</tr>
<tr>
<td>Classic ASP</td>
<td>1%</td>
</tr>
<tr>
<td>ColdFusion</td>
<td>1%</td>
</tr>
<tr>
<td>JavaScript</td>
<td>1%</td>
</tr>
<tr>
<td>Other</td>
<td>3%</td>
</tr>
</tbody>
</table>

---

App Vuln Prevalence

<table>
<thead>
<tr>
<th>Vulnerability</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code Quality</td>
<td>70%</td>
</tr>
<tr>
<td>Crypto Issues</td>
<td>62%</td>
</tr>
<tr>
<td>Info Leakage</td>
<td>62%</td>
</tr>
<tr>
<td>CRLF Injection</td>
<td>54%</td>
</tr>
<tr>
<td>Directory Traversal</td>
<td>49%</td>
</tr>
<tr>
<td>Cross-Site Scripting</td>
<td>48%</td>
</tr>
<tr>
<td>Insufficient Input Validation</td>
<td>37%</td>
</tr>
<tr>
<td>Credentials Mgmt</td>
<td>30%</td>
</tr>
<tr>
<td>SQL Injection</td>
<td>28%</td>
</tr>
<tr>
<td>Time &amp; State</td>
<td>26%</td>
</tr>
</tbody>
</table>
Healthcare summary

• **Initial quality:**
  - 31% of apps passed OWASP Top 10 on first scan

• **Flaw density:**
  - 15 flaws/MB overall
  - 2 very high or high severity flaws/MB

• **Remediation:**
  - Fixed 43% of flaws found during 18 month period

• **Crypto Issues prevalence:**
  - 80% of applications had at least one cryptography-related vulnerability

---

Language Mix

- Java 30%
- .NET 52%
- iOS 7%
- Android 4%
- Classic ASP 2%
- Other 2%
- PHP 3%
- ColdFusion 2%
- JavaScript 0%

App Vuln Prevalence

- Crypto Issues 80%
- Info Leakage 61%
- Code Quality 60%
- Insufficient Input Validation 48%
- CRLF Injection 46%
- Cross-Site Scripting 45%
- Directory Traversal 43%
- SQL Injection 32%
- Time & State 26%
- Credentials Mgmt 23%
Retail and Hospitality summary

• **Initial quality:** 5
  - 30% of apps passed OWASP Top 10 on first scan

• **Flaw density:** 2
  - 28 flaws/MB overall
  - 5 very high or high severity flaws/MB

• **Remediation:** 6
  - Fixed 43% of flaws found during 18 month period

• **SQL Injection prevalence:** 1
  - 25% of applications had at least one SQL Injection vulnerability, the lowest prevalence

Language Mix

- .NET 45%
- Java 37%
- C++ 5%
- iOS 4%
- Android 4%
- Classic ASP 2%
- Other 3%
- PHP 2%
- ColdFusion 1%
- JavaScript 0%

App Vuln Prevalence

- Code Quality 68%
- Crypto Issues 63%
- Info Leakage 55%
- CRLF Injection 54%
- Directory Traversal 52%
- Cross-Site Scripting 44%
- Insufficient Input Validation 44%
- SQL Injection 25%
- Credentials Mgmt 24%
- Time & State 21%
Government summary

- **Initial quality:** 24% of apps passed OWASP Top 10 on first scan

- **Flaw density:** 62 flaws/MB overall
  - 7 very high or high severity flaws/MB

- **Remediation:** Fixed 27% of flaws found during 18 month period

- **SQL Injection prevalence:** 40% of applications had at least one SQL Injection vulnerability

### Language Mix

- Java: 36%
- .NET: 53%
- PHP: 4%
- Classic ASP: 3%
- C++: 1%
- Other: 0%
- ColdFusion: 3%

### App Vuln Prevalence

- Cross-Site Scripting: 70%
- Code Quality: 66%
- Info Leakage: 62%
- Insufficient Input Validation: 52%
- Crypto Issues: 51%
- CRLF Injection: 48%
- Directory Traversal: 45%
- SQL Injection: 40%
- Credentials Mgmt: 20%
- Time & State: 19%
All other industries summary

- **Initial quality:** 6
  - 30% of apps passed OWASP Top 10 on first scan

- **Flaw density:** 6
  - 124 flaws/MB overall
  - 26 very high or high severity flaws/MB

- **Remediation:** 4
  - Fixed 52% of flaws found during 18 month period

- **SQL Injection prevalence:** 6
  - 37% of applications had at least one SQL Injection vulnerability

Language Mix

App Vuln Prevalence

- Crypto Issues: 65%
- Code Quality: 59%
- Info Leakage: 53%
- CRLF Injection: 48%
- Cross-Site Scripting: 46%
- Credentials Mgmt: 37%
- Insufficient Input Validation: 34%
- Directory Traversal: 34%
- SQL Injection: 32%
- Time & State: 23%