VOLUME I

State of Software Security Report

The Intractable Problem of Insecure Software

Executive Summary

March 1, 2010

VERACODE

Executive Summary

As the only cloud-based application risk management services platform in the world to perform and aggregate results from multiple security testing techniques and application types across all participants in the global software supply chain, Veracode is in the unique position of being able to offer the broadest and deepest repository of code level application security intelligence in the world. In this first-ever report of its kind Veracode found strong code-level evidence to support the following observations:

1. Most software is indeed very insecure.
2. Third-party software is a significant percentage of the enterprise software infrastructure, and third-party components are a significant percentage of most applications.
3. Open source projects have comparable security, faster remediation times, and fewer Potential Backdoors than Commercial or Outsourced software.
4. A significant amount of Commercial and Open Source software is written in C/C++ making it disproportionately susceptible to vulnerabilities that allow attackers to gain control of systems.
5. The pervasiveness of easily remedied vulnerabilities indicates a lack of developer education on secure coding.
6. Software of all types from Finance and Government sectors was relatively more secure on first submission to Veracode for testing.
7. Outsourced software is assessed the least, suggesting the absence of contractual security acceptance criteria.

Key Findings

1. Most software is indeed very insecure.

Regardless of software origin, 58% of all applications submitted for verification did not achieve an acceptable security score for its assurance level upon first submission to Veracode for testing when assessed using Veracode's risk adjusted verification methodology.[1] When evaluating against OWASP Top 10 (2007) and CWE/SANS Top 25 Most Dangerous Programming Errors (2009) standards, neither of which adjust for risk, Internally Developed applications fared the poorest, with failure rates as high as 88%. Extrapolating from the application sample set, more than half of the software deployed in enterprises today is potentially susceptible to an application layer attack similar to that used in the recent Heartland[2] or Google[3] security breaches.

Recommendation(s): Implement a comprehensive, risk-based application security program. Design your secure software initiative for breath and depth. Going deep on a handful of applications and ignoring the rest will not lower your organization's overall application risk. Each development team should be implementing a minimum process for application security as part of the development lifecycle. Establishing a security verification step for about-to-be deployed applications is the best place to start.

[1] Refer to Methodology section in Addendum for a description of Veracode's risk adjusted verification methodology.

[2] www.computerworld.com/s/article/9126379/Heartland_data_breach_could_be_bigger_than_TJX_s, www.businessweek.com/technology/content/jul2009/tc2009076_891369_page_2.htm

[3] www.forbes.com/2010/01/14/google-china-mcafee-technology-cio-network-hackers_print.html, www.wired.com/threatlevel/2010/01/operation-aurora

2. Third-party software is a significant percentage of the enterprise software infrastructure, and third-party components are a significant percentage of most applications.

Of applications in the sample set, 60% were designated as Internally Developed, 30% were designated as Commercial, and 10% were designated as Open Source or Outsourced. Regardless of the designation, Veracode observed that between 30% and 70% of all code comprising Internally Developed applications was identifiably from third-parties. Furthermore, there was a "nesting effect" as third-party components themselves often contained other third-party components.

> Recommendation(s): Implement security acceptance criteria and policies for an approved list of third-party suppliers, and conduct security testing on third-party components prior to integrating into final application. Do not develop a false sense of security and control when developing applications with internal teams given the abundance of third-party code integrated into all software.

3. Open Source projects have comparable security, faster remediation times, and fewer Potential Backdoors than Commercial or Outsourced software.

As noted above, no software supplier excelled at delivering secure software upon first submission. Only 39% of submitted Open Source applications and 38% of Commercial applications were acceptable on first submission when evaluated against the CWE/SANS Top 25 Most Dangerous Programming Errors. Open Source applications fared somewhat better than Internally Developed applications, which had an acceptable rate of only 31% against the same industry benchmark.

Open Source project teams remediated security vulnerabilities faster than all other users of Veracode's application risk management services platform. Open Source applications took only 36 days from first submission to reach an acceptable security score, compared to 48 days for Internally Developed applications and 82 days for Commercial applications. This is not surprising given the numerous political and organizational complexities of enterprise development efforts and the formal, customer-centric release plans of Commercial software vendors.

Finally, Open Source contained the fewest Potential Backdoors of any software supplier; substantially less than 1% of vulnerabilities detected across all Open Source applications fell into this category. The relative absence of Potential Backdoors is apparent testimony to the positive effect of transparency in the Open Source community.

> Recommendation(s): Do not fall victim to the fear, uncertainty, and doubt surrounding the use of Open Source software in critical business infrastructure. However, given the risks associated with using code of unknown security, test Outsourced, Commercial, and Open Source suppliers as rigorously as you would test Internally Developed code for security quality and backdoors.

4. A significant amount of Commercial and Open Source software is written in C/C++ making it disproportionately susceptible to vulnerabilities that allow attackers to gain control of systems.

More than 30% of applications were identified as Commercial and supplied to the enterprise by Independent Software Vendors. Commercial suppliers are more likely to use C/C++ than any other language, which can increase risk. Of the C/C++ applications Veracode analyzed, 42% contained vulnerabilities that, if exploited, could result in remote code execution. The vulnerability in Internet Explorer 6 that enabled the Aurora attacks is an example of a remote code execution vulnerability. These classes of defects, including buffer overflows, integer overflows, use after free, and others, are well-known coding errors that have been difficult to eradicate from C/C++ based programs. To exacerbate the problem for enterprises, much of the software written in C/C++ is purchased from software vendors, not built internally. As Gartner[4] and others are recommending, organizations should have purchased software reviewed for security to mitigate this risk.

Recommendation(s): Critical business systems often comprise multiple tiers and development languages. Many contain a hybrid of managed and native code originating from a heterogeneous software supply chain. C/C++, with its idiosyncratic vulnerabilities, is pervasive in the supply chain and no verification process that ignores it will be successful.

Despite the higher likelihood of remote execution vulnerabilities in C/C++, do not be complacent about the risks presented by software written in other languages. Our data reinforces the fundamental notion that serious coding vulnerabilities exist across all languages.

5. The pervasiveness of easily remedied vulnerabilities indicates a lack of developer education on secure coding.

Cross-site Scripting (XSS), the most prevalent vulnerability category by overall frequency and the third most prevalent by number of affected applications, is a stark illustration of the challenges of writing secure code. Despite nearly a decade of focus on cross-site scripting as a serious security threat, its continued prevalence reflects both the pervasive nature of the problem and the evolving threat landscape (i.e. increasing use of dynamic web content). Cross-site Scripting remains as rampant as ever, undeterred by the wide availability of libraries intended to eliminate the risk via proper output encoding. Better education of web developers on this vulnerability and others such as SQL Injection is essential.

Recommendation(s): Implement specific developer training initiatives as part of your overall security program. Follow the lead of corporations such as Microsoft in addressing as many coding mistakes as possible during the education phase of the Secure Development Lifecycle.[5] Educating developers is a cost effective way of preventing security vulnerabilities from being introduced into critical applications. Remember that developer education only helps significantly on new code. Use static analysis on legacy code to eliminate the XSS vulnerabilities already in your code base.

[4] blogs.gartner.com/neil_macdonald/2010/01/14/more-application-security-goodness-from-owasp

[5] blogs.msdn.com/sdl/archive/2009/01/27/sdl-and-the-cwe-sans-top-25.aspx

6. Software of all types from Finance and Government sectors was relatively more secure on first submission to Veracode for testing.

More than half of applications in the Financial Related Industries and Government sectors were deemed acceptable at first submission, using Veracode's risk adjusted verification methodology. This placed them at the top of the more than 15 industries represented in the data set. To a certain extent, this revelation is unsurprising, given that Financial Related Industries have historically been among the first to invest in comprehensive application security programs. Additionally, both sectors have suffered some of the most prominent public breaches in the past, which may have encouraged them to bolster their software security initiatives. The performance of these sectors should be encouraging for all of us who rely on the services they provide; however, room for improvement certainly exists.

> Recommendation(s): Look to organizations with high risk profiles and learn what they have done to implement operating controls in complex environments. It is instructional for lower performing sectors to realize that improvement is possible.

7. Outsourced software is assessed the least, suggesting the absence of contractual security acceptance criteria.

Software identified as Outsourced by submitters accounted for only 2% of the applications in the data set, which was surprising when considering that many enterprises are increasingly relying on offshore development shops as a cost saving measure. With the primary motivation being cost reduction, it is likely that these Outsourcing contracts neglect to define specific security acceptance requirements. This could be one reason why Outsourced software was underrepresented in our data. However, as noted earlier, most applications labeled as Internally Developed actually contained a significant percentage of third-party code, including Outsourced components that were not identified separately.

> Recommendation(s): Do not overlook security requirements when contracting for Outsourced development. When drafting procurement contracts, insist upon the authority to perform independent security testing and set minimum acceptance criteria. This will ensure that you are not charged for rework due to security defects. See the *OWASP Secure Software Contract Annex*[6] and *SANS Application Security Procurement Language*[7] for sample contract language.

[6] www.owasp.org/index.php/OWASP_Secure_Software_Contract_Annex

[7] www.sans.org/appseccontract

VERACODE

SSSR/0310

ABOUT VERACODE

Veracode is the world's leader in cloud-based application risk management. Veracode SecurityReview is the industry's first solution to use patented binary code analysis, dynamic web assessments, and partner or Veracode delivered manual penetration testing, combined with developer e-learning and access to open source security ratings to independently assess and manage application risk across internally developed applications and third-party software without exposing a company's source code. Delivered as a cloud-based service, Veracode provides the simplest, most complete, and most accurate way to implement security best practices, reduce operational cost and comply with internal security policies or external standards such as OWASP Top 10, CWE/SANS Top 25 and PCI.