

VERACODE
VERACODE
VERACODE

VOLUME 4

State of Software Security Report

The Intractable Problem of Insecure Software

December 7, 2011

VERACODE

Executive Summary

The following are some of the most significant findings in the Veracode State of Software Security Report, Volume 4, representing 9,910 application builds assessed in the last 18 months by Veracode on our cloud-based application security platform.

1. Application security performance declines steeply when current threat landscape is taken into account in the evaluation criteria
2. Vulnerabilities that can lead to remote code execution and backdoor functionality are found to be far more prevalent in commercial software
3. Cross-site Scripting and SQL Injection were found to affect higher percentage of Government applications than other industry sectors; SQL Injection trend is flat while declining in the overall dataset
4. A sizable proportion of Android applications were found to contain hard-coded cryptographic keys
5. Independent security verification of third-party software is being carried out by multiple industry segments
6. Greater knowledge of application security is associated with improved security quality scores
7. Development agility and application security are not mutually exclusive

Key Findings

1. Application security performance declines steeply when current threat landscape is taken into account in the evaluation criteria

Earlier this year Veracode transitioned from its risk-adjusted verification methodology used in prior volumes to a new policy intended to be more resilient against prevalent threats. This new policy makes the acceptability threshold more stringent, particularly for higher criticality applications. For example, it adopts a zero-tolerance policy towards frequently exploited vulnerabilities such as Cross-site Scripting (XSS) and SQL Injection (root cause of the Sony breach and countless others). Even a single instance of these types of vulnerabilities causes applications of higher criticalities to be deemed unacceptable. The result of this new policy on application performance was drastic. Over 8 in 10 applications across all supplier types failed to pass when first tested. This marks a steep decline from Volume 3 where the policy being used to measure acceptability was a bit more forgiving and in comparison only 58% of applications failed upon initial submission.

Recommendation: Vulnerabilities such as Cross-site Scripting (XSS) and SQL Injection continue to make headlines because they are frequently exploited by attackers. According to the Web Hacking Incident Database, 20% of reported incidents were caused by SQL Injection. This stands to reason as they are highly prevalent (as Figure 20 shows XSS is present in 68% of all web applications and SQL Injection is present in 32% of all web applications) so they present the “low hanging fruit.” Given this threat environment organizations should implement a program that allows for the discovery and timely remediation of such vulnerability types. Leveraging automated testing technologies allows organizations to scale such a program to their entire application inventory. Furthermore, organizations should adopt a policy where any occurrence of these types of frequently exploited vulnerabilities is not tolerated.

2. Vulnerabilities that can lead to remote code execution and backdoor functionality are found to be far more prevalent in commercial software

Internally developed, outsourced, and open source software development continues to shift to managed code languages and scripting languages which don't fall victim to many of the more serious vulnerability categories that can lead to remote code execution. These vulnerabilities, including buffer management errors, buffer overflows, and integer overflows are found much more often in commercial software due to the continued use of compiled C/C++ and Objective C languages.

An additional high risk vulnerability category, backdoor functionality, is also more prevalent in commercial software. Backdoor functionality, whether malicious or by design, is a category that all buyers of commercial software should be mindful of.

Recommendation: As we have demonstrated since Volume 1 of this report, there is meaningful reliance across industry sectors on commercial suppliers for business critical applications. This is causing organizations to seek independent security verification of commercial software as part of their procurement process (Refer to executive summary finding 5 for more details on third-party risk assessments). Understanding the unique nature of vulnerabilities afflicting commercial software can help inform the policy that should be applied to these applications for them to be considered acceptable. Remote code execution vulnerabilities and backdoor functionality should be explicitly scanned for when testing these applications. A clear requirement for C/C++ and Objective C language support also emerges when choosing security testing technologies for third-party software.

3. Cross-site Scripting and SQL Injection were found to affect higher percentage of Government applications than other industry sectors; SQL Injection trend is flat while declining in the overall dataset

We carried out a deeper analysis of U.S. Government applications (representing federal, state and local government) to investigate whether there are any key similarities or differences between those applications and the remainder of the dataset. Many of these applications process critical data such as PII, national security data and operate critical systems. What we found was that Government web applications had a much higher incidence of XSS issues as compared to Finance and Software—75% of Government web applications had XSS issues compared to 67% for Finance and 55% for Software (Table 7). A partial explanation of this maybe offered by the fact that Government applications utilized a higher percentage of ColdFusion than other industry segments. As Table 4 demonstrates ColdFusion has a higher incidence of XSS issues as compared to other platforms. ColdFusion also tends to be used by less experienced developers for creating web applications with greater ease. These developers are also less likely to be experienced in secure coding practices. Similarly, we found that 40% of government web applications had SQL Injection issues as compared to 29% for Finance and 30% for Software (Table 7). It was also interesting to observe that while SQL Injection was trending lower for the overall dataset (Figure 23) in Government applications it is remaining flat (Figure 30).

Recommendation: SQL injection and XSS are repeatedly in the headlines as the initial attack vector for high-profile, targeted breaches. It is crucial to reduce their occurrence in Government software applications if they are to be more resilient to cyber attacks. Government organizations are encouraged to double down on their efforts to train their development and security staff on how to avoid these errors or fix them quickly once they are found. In addition to training, diligent use should be made of automated testing techniques to expediently discover these vulnerabilities across all of their applications and to verify that the development team is following the guidance learned from their training.

4. A sizable proportion of Android applications were found to contain hard-coded cryptographic keys

This report included Android applications for the first time. Though the dataset was small—just under 1% of our dataset—we found that mobile developers tend to make the same mistakes as enterprise developers. One example of this was the practice of hard-coding cryptographic keys directly into the application. The problem is, once these keys are compromised, any security mechanisms that depend on the secrecy of the keys are then rendered ineffective. In our dataset, over 40% of Android applications contain at least one instance of this flaw (Table 5), a higher rate than we observe across all non-Android Java applications (only 17% of all non-Android Java applications had at least one instance of hard-coded cryptographic keys). More importantly, Android applications are easy to decompile, making it trivial for an attacker to extract and publicize hard-coded keys.

Recommendation: Hold mobile applications to the same security standards as other enterprise applications. Mobile applications are inherently more exposed than web applications because a motivated attacker can start reverse engineering simply by copying the executable off their phone. With that in mind, information embedded into the application—including cryptographic keys—should never be considered secret.

5. Independent security verification of third-party software is being carried out by multiple industry segments

Beginning in Volume 2, we started reporting on the third-party risk assessment market. We define this market as the independent security verification that is performed by software purchasers on applications they procure from their extended software supply chain. As the reliance on third-party software and components has grown, so has the awareness that security weaknesses embedded in those applications become a liability for the enterprise that is accepting that software. This recognition transcends the security community as you see calls for this level of due diligence from leaders in the sourcing and vendor management area as well. In this report we examined which industry segments are heeding this call to action and engaging in this process with their third-party software suppliers. We found enterprises representing at least eight different industry segments—Software, Finance, Aerospace & Defense, Government, Entertainment, Telecommunications, Insurance, and Oil & Gas. While Software and Finance account for the majority of the dataset, companies across the spectrum are starting to hold their software suppliers accountable.

Recommendation: The distribution of applications in an enterprise portfolio has remained relatively unchanged in the last few volumes of the report—approximately one-third of applications are characterized as third-party and two-thirds as internally developed. Oftentimes code developed by an outsourcing partner is labeled internally developed so the actual percentage of third-party software is believed to be even higher than reported. What has also remained true is that 30 to 70% of code components in internally developed applications are in fact third-party components and libraries. With such heavy reliance on code coming from outside an organization, a formal third-party risk assessment program becomes crucial to managing overall application risk. We recommend that all enterprises institute a policy that requires third-party vendors to demonstrate proof of independent security verification or to submit to that due diligence. We also recommend that sourcing and vendor management professionals include specific language in contracts that clarify the security threshold deemed acceptable by the enterprise and make that a requirement for any commercial transaction to proceed.

6. Greater knowledge of application security is associated with improved security quality scores

Beginning in Volume 3 we started our exploration of the state of application security knowledge amongst developers and security professionals. We reported some discouraging statistics that continue to hold true in this volume. More than half of all developers get a grade of C or lower on the application security fundamentals assessments. Top performers (those achieving grade of A) declined from 31% in Volume 3 to 27% in Volume 4 (Figure 34). However, there are some encouraging signs regarding the impact that greater proficiency in application security could have on the resultant security quality of applications being authored by those developers. In this volume we plotted the average security quality score against the average performance on the Veracode Application Security Fundamentals assessment (Figure 36). What we saw was a clustering of data points in the top right i.e. a high score on the assessment and a high score on the application scan. It would appear that those that had a high level of application security knowledge also delivered higher security quality applications. This is good news as it bodes well for the return on investment for security training. We will continue to explore this area further in future volumes to study this relationship further.

Recommendation: Application security training and education is not a formal part of most computer science curriculums and certainly not a consistent theme in the professional development opportunities made available to technology professionals in companies. Therefore the results obtained from the application security fundamentals assessment should come as no surprise. Organizations are strongly encouraged to institute developer training and education programs to ensure a high competency level on application security. As Figure 36 shows there is an association between developers with better knowledge of application security fundamentals and improved security quality of applications. Hopefully readers will be able to utilize this data to both create a sense of urgency around developer training and justify its return.

7. Development agility and application security are not mutually exclusive

It is imprudent to have a conversation about software security devoid of any consideration of software development lifecycles. Contemporary development lifecycles such as agile or extreme advocate frequent “releases” in short development cycles (timeboxing) which is intended to improve productivity and introduce checkpoints where new customer requirements can be adopted. The question we wanted to explore was, can software security be introduced into such modern development lifecycles without impact to the goal of speedy delivery of functionality. As Figure 11 shows the overwhelming majority (over 80%) of applications that failed to achieve acceptable security standards on initial submission and chose to remediate and resubmit, were able to pass within 1 week. This rapid turnaround time can easily be digested into any modern development lifecycle. This should serve to alleviate concerns around security slowing things down.

Recommendation: Conventional wisdom holds that integrating security into the software development lifecycle will result in more secure software. The data also suggests that this integration can be done with minimal impact on development lifecycles. Armed with this knowledge CIOs and CISOs can now more effectively counter concerns that might arise when initiating a formal application risk management program in their organization. We recommend that this formal approach be adopted for not just their internally developed applications but also those being procured from third-parties. Leveraging automation can be a cost-effective way to scale a formal program to all applications in an enterprise portfolio. Augmenting testing with appropriate training will also help tackle the issue more proactively by reducing the likelihood of introducing dangerous programming errors in the first place.



VERACODE

www.veracode.com
© 2011 Veracode, Inc.
All rights reserved.

ABOUT VERACODE

Veracode is the only independent provider of cloud-based application intelligence and security verification services. The Veracode platform provides the fastest, most comprehensive solution to improve the security of internally developed, purchased or outsourced software applications and third-party components. By combining patented static, dynamic and manual testing, extensive eLearning capabilities, and advanced application analytics, Veracode enables scalable, policy-driven application risk management programs that help identify and eradicate numerous vulnerabilities by leveraging best-in-class technologies from vulnerability scanning to penetration testing and static code analysis. Veracode delivers unbiased proof of application security to stakeholders across the software supply chain while supporting independent audit and compliance requirements for all applications no matter how they are deployed, via the web, mobile or in the cloud. Veracode works with customers in more than 80 countries worldwide including Global 2000 brands such as Barclays PLC and Computershare as well as the California Public Employees' Retirement System (CalPERS) and the Federal Aviation Administration (FAA). For more information, visit www.veracode.com, low on Twitter: @Veracode or read the ZeroDay Labs blog.