VERACODE

White Paper

Five Steps to Managing Third-party Application Security Risk

Contents

Executive Summary 3
Software Applications: Gateway to Critical Data 4
Mixed Application Portfolio: A Challenge to Secure 5
Five Key Steps to Managing Third-party Application Risk 8
Step 1 – Identify Your Application Porfolio 8
Step 2 – Assess Security and Risk Posed 9
Step 3 – Fix What Matters Most 10
Step 4 – Learn and Improve 11
Step 5 – Manage Third-party Application Risk 11
How Veracode Can Help 12
About Veracode 13
Annex A – Sample Outsourced Application Contract Language 14

Executive Summary

Application Security is rising to the top of the agenda for Security and Engineering executives. According the Computer Emergency Response Team (CERT), 75% of new attacks target the application layer[1]. The 2009 Verizon Data Breach report states that "Financial Services firms were singled out and fell victim to some very determined, very sophisticated and - unfortunately – very successful attacks in 2008. This industry accounted for 93% of the over 285 million records compromised". One thing is clear - Application vulnerabilities are real and hackers are targeting industries that offer the best avenues for illicit monetary gains.

At the same time, economic, competitive and time-to-market pressures are driving enterprises to use third-party commercial off-the-shelf (COTS), open source and outsourced code as part of their application development process. While this mixed code base of unknown security quality may be an acceptable artifact of modern application development and acquisition, it pushes liability onto the enterprise resulting in an unacceptable level of unbounded corporate risk.

In response to this emerging trend, analyst firm Gartner has advised their enterprise clients that ***"…Application security testing should be mandatory for outsourced development and maintenance."*** Joseph Feiman, Gartner VP and Fellow, went on to recommend that "Enterprises should also consider long-term arrangements with service providers that will be conducting deployed applications' dynamic security testing on a continuous basis (because hackers will be inventing new types of attacks against deployed applications)."[2]

However, until now, enterprises have lacked an efficient manner to analyze the security of their mixed code base. Security testing has been limited to manual analysis by consultants, using internal teams with source code tools or trusting the ISV, outsourcer or open source project to test their own code. These approaches fail to deliver an independent verification of application security, don't scale to cover an enterprise's entire third-party application portfolio and can add significant time and costs to projects.

This whitepaper outlines a five step process that enterprises can apply to their third-party application portfolio to gain visibility into their security state and make informed purchase, integration, deployment and maintenance decisions. From software risk assessments to embedding specific contract language into procurement contracts, these key steps provide guidance that enterprises can swiftly implement to simply and cost-effectively meet regulatory requirements, establish a third-party governance framework and protect their critical assets.

[1] Microsoft Security Intelligence Report 2008 – Based on data from the DHS NVD & CERT

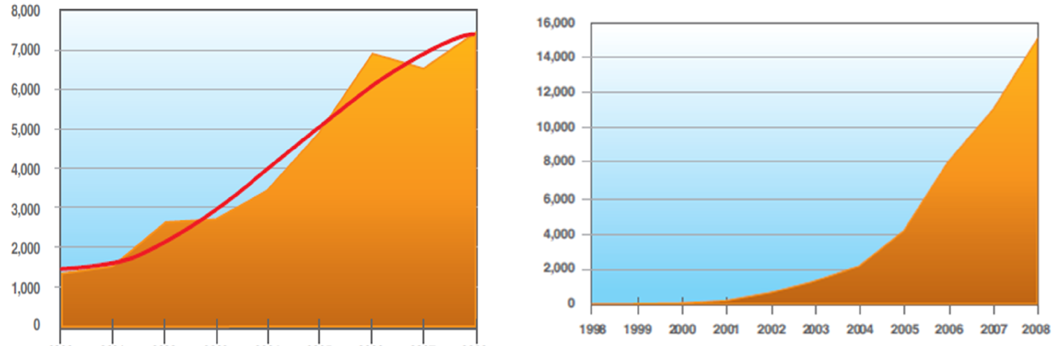
[2] Joseph Feiman, "Application Security Testing Should Be Mandatory", 2007, Gartner ID Number: G00146313

Software Applications: A Hacker's Preferred Gateway to Critical Data

Today's software applications have become the enterprise's ''new perimeter''. With better network-level security technology hardening the network perimeter, malicious attackers are now focusing their efforts to strike at the least defended and arguably the hardest to defend gateway --- - the application. Applications-- particularly consumer facing web applications-- are designed to allow access to a myriad of critical data including Personally Identifiable Information (PII) and financial transaction information. While hackers were once satisfied with defacing Websites, unleashing denial-of-service attacks and trading illicit files through targeted networks, modern attackers are profit-driven. Financial and customer data have become valuable commodities and exploiting applications that control access to this critical data is a hacker's easiest and most profitable path to illicit monetary gains.

Recent industry statistics confirm this trend:

- **Application vulnerabilities are on the rise:** Data from IBM X-Force's 2008 Trend and Risk Report (*Figure 1*) reveals that the number of vulnerabilities has risen dramatically and has exceeded 7,000 new vulnerability disclosures in the past year – an all time high[3]. It further states that the most prevalent type of vulnerabilities affecting servers today is unquestionably vulnerabilities related to Web applications (accounting for 54% of all vulnerability disclosures).



IBM X-Force: 2008 Vulnerability Disclosure Count

IBM X-Force: 2008 Cumulative Count of Web Application Vulnerabilities

Figure 1: IBM X-Force Trend and Risk Report

- **Enterprises are the target:** Meanwhile, Gartner and NIST report that 78% of threats target business information[4].
- **Enterprises are falling victim:** A recent survey conducted by Forrester Research had 62% of respondents claiming that they have experienced breaches due to software insecurity in the last 12 months[5].

The message is clear – overwhelming majority of vulnerabilities are in software applications and hackers are turning to these applications as the medium of choice when targeting critical business information.

[3] IBM X-Force 2008 Trend and Risk Report

[4] Theresa Lanowitz, "Now Is the Time for Security at the Application Level" 2005, Gartner

[5] Application Risk Management in Business Survey, Forrester Research, 2009

What's in your SOUP? : The Mixed Application Portfolio Challenge

In order to protect your enterprise applications and develop an effective application risk management strategy, it is important to understand the underlying diversity and pedigree of the applications in your portfolio.

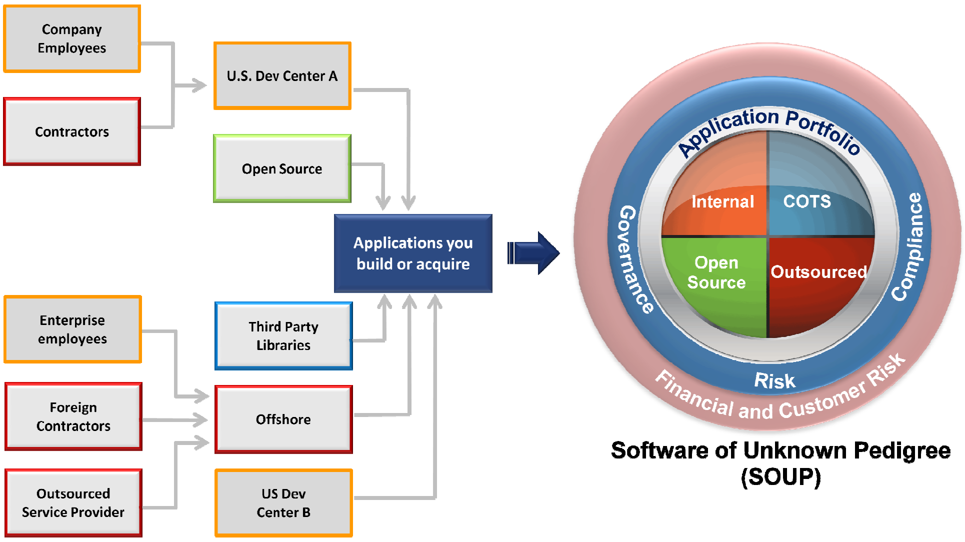
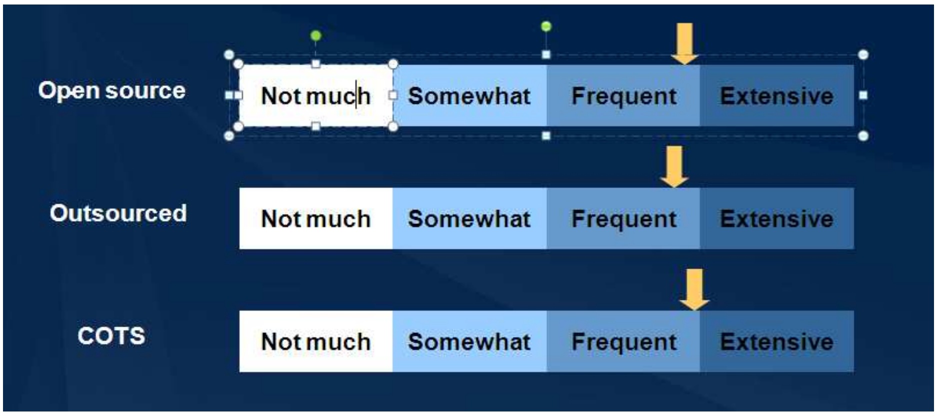


Figure 2: Mixed Application Portfolio of Unknown Security Quality

As depicted in *Figure 2*, virtually all modern applications can be characterized as Software of Unknown Pedigree (SOUP). A typical enterprise application portfolio consists of a mixed bag of internally developed, outsourced, commercial-off-the-shelf (COTS) and open source applications. Furthermore, application development is being carried out by a disparate set of internal and external teams that may have widely varying application security skill-sets, development best practices and security verification standards in place. Protecting your organization from the threat posed by insecure applications means securing this SOUP, including applications acquired from third-parties.

The aforementioned Forrester survey also found the respondents confirming that third-party code is being used pervasively. *Figure 3* depicts the responses when asked how extensively they are using these application categories for applications that they deemed business critical.



Open source	Not much	Somewhat	Frequent	Extensive
Outsourced	Not much	Somewhat	Frequent	Extensive
COTS	Not much	Somewhat	Frequent	Extensive

Base: 204 IT Professionals

Source: Online Survey of 204 Application and Risk Management Professionals.

Figure 3: Survey responses regarding Usage of Third Party Application for Business Critical Applications

According to analyst firm Gartner, offshore software development is expected to rise from $50 billion today to over $88 billion within four years[6] and InformationWeek has reported that over two-thirds of the companies in the InformationWeek 500 use at least some offshore software development to build and maintain their applications[7].

As seen by these industry reports use of third-party code seems pervasive. However, knowledge of the security quality of third-party applications and the resultant risk remains low. Also from the Forrester Survey, only a third required rigorous security testing before accepting and implementing code from outsourcers. That leaves a gaping hole in their efforts to protect their enterprise from the risk posed by insecure applications.

Regulators and enterprises are recognizing the significant contribution from third-party code to the overall state of application security in an enterprise. The regulatory climate on the topic is moving in the direction of stronger mandates and will influence a change in behavior both on the buyer and seller side. Some noteworthy regulations and organizational mandates currently in place or being formulated include:

Us Dept. of the Treasury's Office of the Comptroller of the Currency (OCC) Bulleting 2008-16 – Requires all national banks to implement application risk management programs for all applications, whether internally developed, vendor-acquired, or contracted for. Vulnerabilities in applications increase operational and reputation risk as unplanned or unknown weaknesses may compromise the confidentiality, availability, and integrity of data.
http://occ.treas.gov/ftp/bulletin/2008-16.html

Cybersecurity Act of 2009 (Senate Bill 773) - Will establish standards and a certification program for measuring software security using a prioritized list of software weaknesses known to lead to exploited

[6] "Applications Services Scenario: 2008 to 2012 — Trends and Directions, 2008", 2008, Gartner

[7] Mary Hayes Weier, "The Second Decade Of Offshore Outsourcing: Where We're Headed", Nov. 2007, InformationWeek

and exploitable vulnerabilities. Software development organizations must show documented testing results demonstrating they comply with the standard during the software development process. http://www.opencongress.org/bill/111-s773/text

European Commission Consumer Protection – Proposal to expand existing consumer and business protection laws to provide legal recourse against software vendors and developers. Under the proposal software companies could be held responsible for the security and efficacy of their products.

State of NY & Depository Trust & Clearing Corp (DTCC) Require Software Security – The State of NY and DTCC authored the SANS Application Security Procurement Language which provides organizations a standard method to require application security testing and training as part of the software they purchase. These organizations now require software providers to meet the requirements of the contract as a prerequisite for doing business with their organizations. http://www.sans.org/appseccontract/

In summary, enterprises may be able to outsource application development but they cannot outsource regulatory compliance and the risk they inherit from the software supply chain. Therefore addressing third-party code has to form an integral part of any organization’s application risk management strategy if an enterprise is to be truly protected and in compliance.

Five Key Steps to Managing Third-party Application Risk

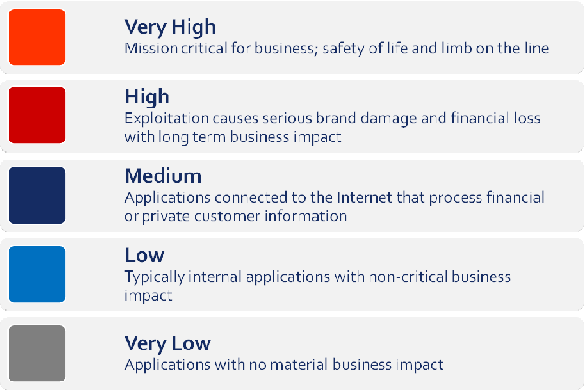
Enterprises face an uphill battle in managing security risks across their extended software supply chain. Nonetheless, despite the difficulties, identifying, controlling and reducing this unbounded risk is critical. Based on its experience working with enterprises that rely on COTS applications, open-source and outsourced code, Veracode has compiled a list of five key steps which will help enterprises manage risk from third-parties in a simple and cost-effective manner.

Step 1 – Identify Your Application Portfolio

You cannot secure what you don't know exists. While it may seem obvious that organizations need to inventory their application portfolio as part of an application risk management program, in practice it can be a challenging exercise. It is common to see application "sprawl" as individual groups or business units may have contracted work, purchased COTS applications, integrated in open source or third-party libraries without appropriate cataloging. When conducting an application inventory, involve business units, procurement and vendor management to ensure you identify all software that *has entered* or *is entering* the organization through third-parties. Leverage data such as results of network scans (where they identify web servers) or lists of purchased SSL certificates as clues to discover additional applications in your portfolio. This may not identify all applications (for example, apps with self-signed certificates or multiple apps running on the same web server) but will provide you with a starting point for your inventory.

Assign Business Criticality (Assurance Level)

It is also important to recognize that not all applications are created equal. As important as it is to know *what* exists in your application portfolio it is equally as important to recognize *how much* you need to care about it. This requires an organization to understand the risk that the application poses to the business. This can be achieved by assigning an assurance level for each application based on business risk factors such as reputation damage, regulatory impact, financial loss, operational risk, sensitive information disclosure, personal safety, and legal violations. For example, an enterprise engaged in online credit card transactions will need to know which applications in their portfolio are subject to the Payment Card Industry Data Security Standard (PCI-DSS) regulation and assign those applications a higher assurance level. The following chart (*Figure 4*) from NIST provides guidance on selecting an assurance level based on business risk:



Very High
Mission critical for business; safety of life and limb on the line

High
Exploitation causes serious brand damage and financial loss with long term business impact

Medium
Applications connected to the Internet that process financial or private customer information

Low
Typically internal applications with non-critical business impact

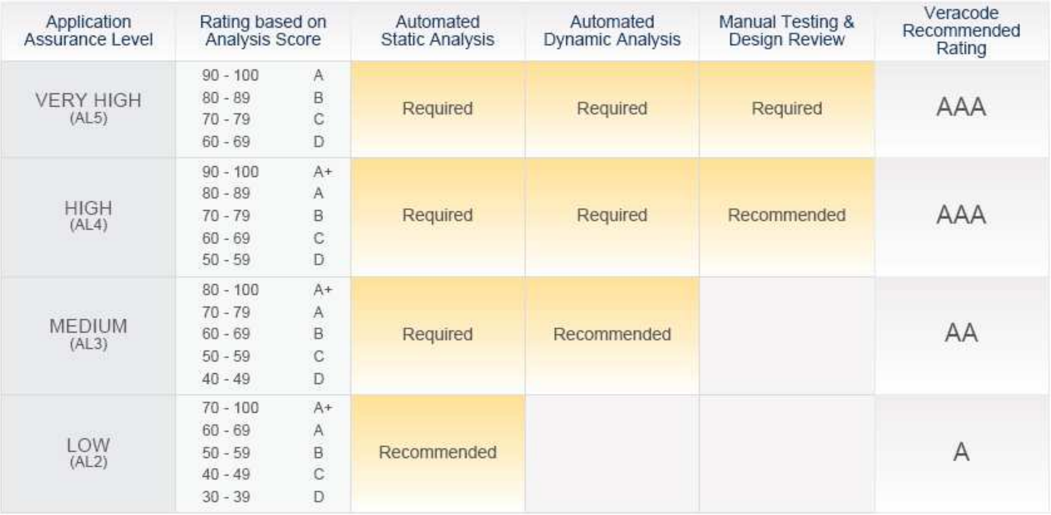
Very Low
Applications with no material business impact

Figure 4: NIST Application Assurance Level Chart

Designate Security Policy

The next step after assigning an assurance level is to designate a security policy commensurate with the business value from the applications. To demonstrate how security ratings/policies can be applied, we will use Veracode's SecurityReview service as an example. Various application testing techniques are combined with a scoring system based on the Common Vulnerability Scoring System (CVSS) and the Common Weakness Enumeration (CWE) standards to produce a Security Quality Score (SQS) for each application. The SQS combined with the assurance level the enterprise selected is used to assign an easy to understand letter grade (A, B, C, D or F) that designates the security rating for the application.

Thus, enterprises can set an acceptable rating/policy – "A" for example and outsourcing providers know they must achieve that for the application to be accepted. Setting thresholds and using standard-based scoring removes the subjectivity and "gray-area" on what constitutes acceptance and clarifies the process for both the enterprise and provider. Below is a chart that demonstrates how organizations can use assurance levels, quality scores and testing methods to achieve an overall rating:



Application Assurance Level	Rating based on Analysis Score		Automated Static Analysis	Automated Dynamic Analysis	Manual Testing & Design Review	Veracode Recommended Rating
VERY HIGH (AL5)	90 - 100 80 - 89 70 - 79 60 - 69	A B C D	Required	Required	Required	AAA
HIGH (AL4)	90 - 100 80 - 89 70 - 79 60 - 69 50 - 59	A+ A B C D	Required	Required	Recommended	AAA
MEDIUM (AL3)	80 - 100 70 - 79 60 - 69 50 - 59 40 - 49	A+ A B C D	Required	Recommended		AA
LOW (AL2)	70 - 100 60 - 69 50 - 59 40 - 49 30 - 39	A+ A B C D	Recommended			A

Figure 5: Independent rating system to aggregate assurance levels, quality scores and testing methods

In addition to the assurance level and associated security policy, enterprises should also use the Identify step to capture meaningful meta-data about the application such as origin, development team owner, deployment state etc. Documenting these important defining characteristics of your application portfolio will provide a better understanding of the biggest sources of risk, where accountability lies and the most effective security verification and remediation path.

Step 2 – Assess Security Risks of Applications

Once an organization understands *what* applications they have in their portfolio and *how much* they need to care about each, they need to verify whether the security state of the application is in compliance with the security rating/policy deemed most appropriate for it. A combination of testing techniques may have to be adopted dependent on application type and code availability.

Distinguishing Application Type:

The meta-data gathered during the Identify step can be used to determine the most effective security verification technique that can and should be applied to the application. There are generally three forms of security testing that can be performed:

- **Static Application Security Testing (SAST)**: SAST is a set of technologies designed to analyze application source code, byte code and binaries for coding and design conditions that are indicative of security vulnerabilities. SAST solutions analyze an application from the "inside out" in a non-running state. SAST particularly when performed against the final integrated application (binary) is a good option for testing third-party code where source may not be readily available.
- **Dynamic Application Security Testing (DAST)**: DAST technologies are designed to detect conditions indicative of security vulnerability in an application in its running state. For web-facing applications, a combination of static and dynamic analysis may offer the best automated coverage.
- **Manual Analysis:** This involves an expert human being performing either a manual code review or a manual penetration test where they attempt to discover design flaws or more complicated vulnerabilities that may not be possible to detect via automated means alone. For high assurance applications in a portfolio, manual analysis should be used to augment the testing performed by automated means.

Determining Code Availability:

The final integrated application typically consists of third-party libraries and components that get integrated at build-time for which you don’t have or don’t own the source. In such circumstances (and it is frequent when dealing with third-party code), binary analysis may be the only simple and cost-effective method to analyze the code statically. Not only does this allow the enterprise to look at the final integrated application without needing source but it offers some inherent advantages for testing for backdoors and malicious code. In all cases regardless of source code availability, assessment should be performed with due consideration of IP rights. As is the case with COTS software, IP rights rest with the vendor. Any assessment the enterprise performs should be done in collaboration with the vendor and there should be responsible handling of any resultant findings. Often it is easiest for an enterprise to avoid any IP violation risks by engaging a trusted independent third-party to perform the assessment and mediate between the vendor and enterprise organization.

Step 3 – Fix What Matters Most

All the intelligence gathered during the Identify and Assess steps feeds into the Fix step by enabling organizations and their developers to prioritize which applications and vulnerabilities pose the highest risk and should be given remediation priority. It is important to note that not everything needs to be fixed – just as all applications are not created equal, neither are all vulnerabilities. When reviewing the findings from the Assess phase, it is important to consider aspects such as placement and exposure (e.g. external web sites), regulatory impact (e.g. any OWASP Top 10 vulnerability needs to be remediated for PCI compliance) and exploitability (e.g. are there compensating controls that render the vulnerability less exploitable).

For third-party code that is under consideration for purchase, the Fix phase offers an opportunity for the buyer and seller to dialog based on the security standards that should be built into the contract. For example, the enterprise may choose to mandate that all High priority vulnerabilities be fixed prior to code acceptance. This will help ensure that only third-party code that conforms to the designated security policy enters the organization's ecosystem. In fact Gartner recommends that application security testing should be mandatory for all outsourced development and maintenance. Indeed the American Law Institute (ALI) has issued a new set of legal principles for software contracts which states that parties who receive payment for software "warrants to any party in the normal chain of distribution that the software contains no material hidden defects of which the transferor was aware at the time of the transfer." Veracode has created a "Recommended Outsourced Software Development Security Contract Language" which organizations can use as part of their contracts (See Annex A).

For third-party code that is already deployed, the Fix phase may offer an opportunity to discuss upcoming product releases where the vulnerabilities have already been addressed or can be addressed. It may also require a discussion around maintenance or renewal contracts where security standards may now be introduced into the contract if they were not previously.

Depending on the severity of the vulnerabilities and the time-to-fix communicated by the third-party, the enterprise can also make decisions around any intervening compensating controls that may need to be applied to prevent risk of exploits while the underlying code weakness is remediated.

Step 4 – Learn and Improve

Education is another critical step in the process. Without clear knowledge of what constitutes good coding practices, many developers will repeat mistakes without knowing it. This is not surprising given that this type of training isn't readily available in universities and is typically cost prohibitive. Enterprises should look to leverage online computer-based training services to provide proactive education to the developer and security communities in their organization. This will not only benefit code they are developing internally, but will allow their team to exercise better judgment when assessing the true risk posed by vulnerabilities discovered in third-party code. Enterprises should also investigate the application security education practices being adopted by their vendor community.

Step 5 – Manage Third-party Application Risk

It's been said that "knowledge is power" and in the application risk space this means understanding which applications matter most to the business and which applications pose the greatest risk. As we have seen, why they matter may vary from company to company. One organization may be concerned with the assurance level of an application because it is critical to revenue generation (for example an airline with an on-line ticketing site). Another company may be concerned with an application because it manages valuable assets (for example a bank's online checking solution). What does remain constant, though, is that all companies need to *manage* their risk level.

This is exactly what they will be well-positioned to do by correlating the intelligence gathered from the steps described thus far. The Identify step will help enterprises understand the sources of risk, the proportion coming from third-parties and consider the commensurate security policies that need to be applied to those code bases. The Assess step will help them understand the vulnerability state and resultant risk exposure from the acquisition of third-party code or from an already deployed application. The Fix phase offers the opportunity to work collaboratively with the third-party to achieve the desired security rating and institute the appropriate contractual protections for future engagement. The Learn phase allows the Enterprise and the vendor to improve application security skill-set and knowledge.

By maintaining the data derived from these steps in a centralized repository, benchmarking and trending information across internal teams, vendors and outsourcing partners can be generated. The enterprise now has the opportunity to get consistent performance metrics across their application portfolio. For example, they can get a consolidated view of regulatory compliance across internal and third-party applications. They can identify third-parties that are contributing most significantly to their application portfolio and their application risk. They can prioritize remediation efforts with vendors, upgrades to new product releases and optimize contract negotiations by being better informed.

Together the Identify, Assess, Fix and Learn steps help the enterprise *Manage* application risk across their entire mixed application portfolio.

How Veracode Can Help

Veracode's SecurityReview® is delivered as a turnkey service that can be customized to fit your organization's unique needs and accommodate the changing threat and security requirements' landscape. It empowers organizations with a robust solution to quickly kick-start and operationalize a security program that covers your entire mixed application portoflio. It provides unlimited access to a centralized application risk management platform, multiple testing techniques, security intelligence information on open source projects and eLearning. SecurityReview Highlights are described below:

Application Risk ManagementPlatform
A centralized view of risk and security information to manage, track and report on your application portfolio across your entire enterprise available 24x7 through Veracode's secure, web-based SaaS platform hosted in a secure Software Assurance Center.

Static Binary Analysis
Veracode's patented automated static binary analysis reviews code in its "final" compiled version, including libraries and 3rd party components, without requiring organizations to expose their intellectual property in the form of source code. This approach results in the most accurate and complete security testing available in the industry.

Dynamic Analysis
Veracode's automated web application vulnerability scanning, also known as dynamic analysis or black-box testing, empowers companies to identify and remediate security issues in their web applications before hackers can exploit them.

eLearning
SecurityReview integrates web-based secure programming training modules for developers and security personnel to meet formal training and competency testing requirements.

Open Source Ratings Database
Veracode's database of security ratings for enterprise-class open source projects enables organizational understanding of the risk/benefit trade-off of integrating open source versus commercially software.

Custom Policies and Compliance Management
Veracode's assessment and risk-management platform allows organizations to meet the application security requirements of PCI, OCC Bulletin 2008-16, FISMA, HIPAA, SOX, GLBA and industry standards such as OWASP Top 10 and SANS Top 25 by setting security policies for assessments.

Security Advisor Services
Leading enterprises leverage Veracode's full range of application lifecycle services including application inventory support, remediation advice, build and upload support and program management services.

As an expert in application security, Veracode is uniquely suited to provide independent verification and validation (IV&V) of third-party applications without the need for costly on-site consultants. Veracode's Ratings System produces a software security rating based on respected industry standards including MITRE's Common Weakness Enumeration (CWE) for classification of software weaknesses and FIRST's Common Vulnerability Scoring System (CVSS) for severity and ease of exploitability and NIST's application assurance levels. These universally accepted vulnerability scoring methods provide a clear audit trail enabling enterprises to automate the security acceptance testing of outsourced applications and meet both internal and external security and compliance requirements and reduce their exposure to risk.

About Veracode

Veracode provides the world's leading Application Risk Management Platform. Veracode SecurityReview 's patented and proven cloud-based capabilities allow customers to govern and mitigate software security risk across a single application or an entire enterprise application portfolio with unmatched simplicity. Customers include the world's largest and most security-aware organizations across every industry. Recognized as a Gartner "Cool Vendor" and with The Wall Street Journal's "Technology Innovation Award," The Banker's "Information Security Project of the Year" with SC Magazine's "Best Security Solution for Financial Services, 2009 SD Times 100 list," Information Security "Readers' Choice Award," and AlwaysOn Northeast's "Top 100 Private Company ," Veracode is Software Security Simplified™.

Based in Burlington, Mass., Veracode is backed by .406 Ventures, Atlas Venture and Polaris Venture Partners.

Contact Information

Veracode, Inc.
4 Van De Graaff drive
Burlington, MA 01803
+1 781 425-6040.
For more information, visit www.veracode.com.

Appendix A – Sample Outsourced Application Contract Language

This sample contract Annex is intended to help enterprises negotiate the purchase of outsourced software development. Most software development contracts focus on features, functionality and delivery timeframes. Additionally, they may require the developer to show a certain level of application security competency or attempt to include liability clauses as part of the contract process. Frequently, the parties have very different views on what defines application security and what has actually been agreed to in the contract. The following languages lays out a simple process, utilizing independent security reviews and industry standard benchmarks, which allows both outsourced developers and enterprises to ensure that application security is embedded in the deliverable.

Portions of this document incorporate details from the OWASP Secure Software Contract Annex and the SwA Working Group's "Software Assurance (SwA) in Acquisition: Mitigating Risks to the Enterprise" paper. Organizations are free to use the following sample language, however, as with any legal agreement, we recommend you contact a qualified attorney prior to entering into any contract.

Sample Contract Annex

1. INTRODUCTION

This Annex is made to ______________________ ("Agreement") between Client and Developer. Client and Developer agree to maximize the security of the software according to the following terms.

2. ORIGIN, LIBRARIES, FRAMEWORKS, AND PRODUCTS

(a) Disclosure

Developer shall disclose all binary executables (i.e. compiled or byte code; source code is not required) of the software, including all libraries or components.

(b) Origin

Developer shall disclose the origin of all software components used in the product including any open source or 3rd party licensed components.

3. SECURITY REVIEWS

(a) Independent Review

Developer shall have their software reviewed for security flaws, in binary format (i.e. compiled or byte code; source code is not required), by an independent organization that specializes in application security, at their expense, prior to delivery to the Client.

(b) Review Coverage

Security reviews shall cover all aspects of the software delivered, including third party components, and libraries.

(c) Scope of Review

At a minimum, the review shall cover common software vulnerabilities. The review may include a combination of static analysis of the binary code, dynamic web application vulnerability scanning, and manual penetration testing.

(d) Issues Discovered

Overall application security ratings with aggregate number of flaws found by the independent organization shall be reported to both Client and Developer. Detailed reports of specific vulnerability instances within the application will only be provided to the Developer. All issues will be tracked and remediated as specified in the Security Issue Management section of this Annex.

(e) Standard Benchmarks

To ensure that all parties have a common understanding of any security issues uncovered, the independent organization that specializes in application security shall provide a rating based on industry standards as defined by First's Common Vulnerability Scoring System (CVSS) and Mitre's Common Weakness Enumeration (CWE).

(f) Review Frequency

Reviews shall be conducted to revalidate the software prior to delivery of any new major or minor release prior to delivery to Client.

4. SECURITY ISSUE MANAGEMENT

(a) Identification

Developer will track all security issues uncovered during the security review and the entire life cycle, whether a requirements, design, implementation, testing, deployment, or operational issue. The risk associated with each security issue will be evaluated, documented, and reported to Client as soon as possible after discovery.

(b) Protection

Developer will appropriately protect information regarding security issues and associated documentation to help limit the likelihood that vulnerabilities in operational Client software are exposed.

(c) Remediation

Client and Developer shall create a mutually agreed upon remediation roadmap to resolve security issues that are identified. Developer shall make all commercially feasible efforts to fix all high level issues prior to delivery to Client.

5. SECURITY ACCEPTANCE AND MAINTENANCE

(a) Acceptance

The software shall not be considered accepted until the independent review is complete and all security issues have been assigned to a mutually agreed upon remediation roadmap.

(b) Investigating Security Issues

After acceptance, if security issues are discovered or reasonably suspected, Developer shall assist Client in performing an investigation to determine the nature of the issue.

(c) Other Security Issues

Developer shall use all commercially reasonable efforts consistent with sound software development practices, taking into account the severity of the risk, to resolve all security issues as quickly as possible.