VERACODE SECURITY GUIDE

Best Practices of

THIRD-PARTY SOFTWARE SECURITY

VERACODE

The average enterprise's software ecosystem has become big, complex... and insecure. Consider:

The typical enterprise today has

372 MISSION-CRITICAL APPLICATIONS

with at least some code from external sources (IDG).

The applications Veracode scans have an average of

46 UNIQUE COMPONENTS

75% OF THIRD-PARTY APPLICATIONS

Veracode scanned in a recent 18-month period were not compliant with the OWASP Top 10 policy for security vulnerabilities (Veracode 2016 State of Software Security report).

97% OF ALL JAVA APPLICATIONS

Veracode recently scanned in an 18-month period had at least one component with a known vulnerability.

Regulations from organizations including FS-ISAC, the New York Department of Financial Services, NIST, PCI, MAS and more

NOW INCLUDE MANDATES SURROUNDING THE USE OF THIRD-PARTY APPLICATIONS AND/OR OPEN SOURCE CODE.

The result of these stats?

Third-party software (think Target breach) and open source components (think Heartbleed and the more recent "Struts-Shock") are now a top target for cyberattackers. In response, every organization needs to understand and better manage the risks inherent in its reliance on vendor-supplied software, outsourced code and open source components. Yet most IT departments don't have visibility into what open source components developers are using or have used, and also don't have the time, budget or internal resources to run a meaningful vendor software testing program.

Challenges with Securing Vendor Software, Outsourced Code and Open Source Components

For third-party applications and code, traditional test methods can be laborious and may cover only a fraction of externally sourced software in use.

Security testing of vendor packages has been limited to manual penetration testing by consultants, internal teams using source code analysis tools, or trusting the software vendor, outsourcer or open source project to secure its own code. These approaches fail to deliver an independent verification of application security, or scale to cover an enterprise's entire vendor application portfolio. Plus, these approaches can add significant time and costs to projects. Another complication is vendor reluctance to expose their source code for security testing in the first place.

In addition, manually locating every open source component in use at an organization and then finding out if it contains a vulnerability is a nearly impossibly cumbersome and time-consuming exercise.

For example, to understand the risk associated with open source components, you would have to carry out the following five steps:

1. **ASK DEVELOPERS**
   to track all open source projects your organization is using.
2. **REQUIRE VENDORS**
   to disclose bill of materials for commercial software.
3. **FOLLOW PUBLIC VULNERABILITY DISCLOSURES**
   for each component.
4. **ALERT ENGINEERS**
   about insecure libraries.
5. **CONTACT ALL DEVELOPMENT TEAMS**
   to figure out if they're using a particular component (in a rapid vulnerability response).

In this guide, we offer our best practices for overcoming these challenges and improving software security across your entire application landscape.

Best Practices for Securing Third-Party Applications

Define a Vendor Application Security Policy

When undertaking any vendor security compliance effort, it's important to get the right people in the organization involved early in the process. We recommend a cross-functional steering committee that could involve IT security professionals, vendor managers, risk auditors, business unit representatives, sourcing or procurement managers, as well as legal. It is also important to confirm who will actually be performing the software testing — internal development, penetration testers, code reviewers, compliance auditors or a third-party solution provider.

Once assembled, the committee should define the enterprise's vendor software security compliance policy, identifying business goals and completing the following:

DETERMINE WHAT TYPE OF SECURITY TESTING *is required (i.e., static, dynamic, manual).*

DETERMINE THE TESTING PRODUCTS OR SERVICES TO BE USED, *and how they will safeguard vendors' intellectual property.*

DOCUMENT THE ANALYSIS *timeline and frequency of testing.*

DEFINE POST-ANALYSIS *next steps for every possible outcome, with potential impacts on the vendor relationship.*

DOCUMENT VULNERABILITY REMEDIATION EXPECTATIONS *and acceptance criteria (for example: OWASP TOP 10 must be passed, and anything that fails must be fixed).*

DOCUMENT A MITIGATION PROCESS *for false-positive results, or design mechanisms that override these flaws.*

DEFINE AN EXCEPTION AND ESCALATION PROCESS *for uncooperative vendors, which may include non-compliance penalties.*

Communicate Vendor Application Security Requirements

Once the enterprise policy has been defined, it's time to introduce the software security analysis mandate to all vendors and suppliers.

Some tips on this communication:

IT SHOULD COME FROM THE ENTERPRISE ITSELF — *from the highest level possible, such as the CIO or CISO — and go to the highest-value business stakeholder possible — not technical support or development personnel.*

IT MUST CLEARLY STATE THE REASONS BEHIND THE NEW REQUIREMENTS *and the business goals to be achieved, without a lot of technical details.*

WHENEVER POSSIBLE, SET UP AN IN-PERSON MEETING *or live conversation to broker the introduction to the testing team. This will engender trust with the people who will actually be working with the vendor on its compliance efforts.*

GIVE THE VENDOR A CLEAR UNDERSTANDING OF THE ANCILLARY BENEFITS *of its demonstrated commitment to producing more secure software. These could include a promise to speed future renewal contracts or grant a more formal recognition of "preferred vendor status."*

UPDATE CONTRACTS *and legal relationships with software vendors in alignment with this requirement.*

Ensure Vendor Commitment and Education

Once a vendor has committed to comply with the application security mandate, you can start a deeper education on the technical aspects of the effort. Provide written guidance that instructs the vendor on all aspects of the analysis process, testing methodologies, expectations and timelines. The goal is to obtain a firm agreement from the vendor that it will follow established compliance procedures.

Intellectual property protection is one of the most common objections from software vendors when confronting third-party testing regimens. Most are reluctant to allow outside parties to have direct access to their source code. It is best to acknowledge this early in the education process and detail how the program plans to safeguard their code.

Manage Third-Party Test Execution and Compliance

Actually administering the enterprise's software security compliance effort often proves the most challenging undertaking.

Tasks include:

- ANALYZING VENDOR SOFTWARE
- ISSUING AND INTERPRETING TEST REPORTS
- PRIORITIZING VULNERABILITIES FOR REMEDIATION
- DEVELOPMENT TEAMS MAKING FIXES
- RETESTING THE SOFTWARE TO CONFIRM COMPLIANCE WITH POLICY

The enterprise's testing team must provide consistent project management and status reporting to the steering committee to drive the program and minimize delays.

This process might offer an opportunity to discuss the vendor's upcoming product releases where the vulnerabilities may 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.

Understand and Improve Vendor Security Results

By maintaining test results derived from these efforts in a centralized repository, you can generate benchmarking and trending information across the entire vendor application portfolio. You can then use this information to prioritize upgrades to new product releases and optimize contract negotiations.

Best Practices for Secure Use of Open Source Components

Most organizations today not only depend on outside sources for applications, but also for components of their own internally developed code base.

However, open source components are increasing risk and leading to breaches. But open source component use is not the problem; visibility is. The reason so many organizations struggle to keep their components free of vulnerabilities is because they simply don't know they are using a vulnerable component.

For example:

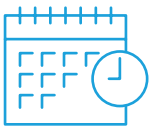
OUR MOST RECENT STATE OF SOFTWARE SECURITY REPORT (BASED ON OUR PLATFORM DATA) FOUND THAT:

The second most prevalent component vulnerability was released in January

2008

And yet it was patched in

2010



It's the second most common vulnerability, and it was patched seven years ago.

It's clear to see now why the chaos that ensued after Heartbleed could happen — a significant number of organizations don't even realize that they are using a very risky component. Unless your company has a dynamic inventory of components used by developers, you will be unable to remediate or mitigate the risk when a vulnerability in an open source component is disclosed.

Best Practices to Ensure the Security of Outsourced Code

Outsourcing application development allows organizations to realize cost savings and provides the flexibility necessary to scale.

However, it also introduces significant risk in the form of security vulnerabilities and malicious backdoors. Veracode recommends five key steps to help enterprises implement security into their outsourced application development.

1 Take a Risk-Based Approach to Application Security (Threat Modeling)

You have selected an application as a good candidate for outsourcing, but it is also necessary to understand its impact to the business. Determining business criticality, or assurance level, is an important step in obtaining a clear understanding of the security risk in your outsourced application portfolio.

The business criticality should be based on six core potential impact dimensions:

- INCONVENIENCE, DISTRESS OR DAMAGE TO STANDING OR REPUTATION
- FINANCIAL LOSS OR AGENCY LIABILITY
- HARM TO ORGANIZATION PROGRAMS OR STAKEHOLDERS
- UNAUTHORIZED RELEASE OF SENSITIVE INFORMATION
- PERSONAL SAFETY
- CIVIL OR CRIMINAL VIOLATIONS

2 Establish Security Metrics and SLAs with Outsourcing Providers

Outsourced software development contracts typically emphasize features, quality, time and costs. Thus, the burden and risks of application security have fallen solely on the enterprise. Establish clear metrics and SLAs surrounding application security with your outsourcing partners as part of the procurement and contract processes. This should be in alignment with your security policy for internally developed applications of a similar risk rank.

3 Conduct Independent Application Security Testing

Independent testing requires that it's conducted by a third party that has no vested interest in anything beyond providing accurate results and supporting evidence. In contrast, the buyer or the seller conducting this testing might have a vested interest in ensuring suppression of relevant results. Ideally, conduct this testing with an automated service that doesn't require access to source code.

4 Set Remediation Timeframe for Outsourced Applications

You can leverage software security ratings to decide which applications are secure enough to be accepted or deployed and which applications need remediation by the outsourcing provider before software acceptance (and payment!).

Perhaps equally as important is to specifically establish a timeline for addressing those security findings that are unacceptable according to your security policy. Remediation timeframes can be as simple as "all Very High severity findings must be addressed within 14 days" to as a granular as prescribing timeframes for specific CWEs, such as "CWE 89 must be remediated within five business days." Establish remediation plans in conjunction with the development team responsible for the application, rather than in a vacuum, so that you can take the ability and knowledge of the development team into account along with any development milestones.

5 Outsource Applications to Providers That Have Obtained Security Verification

Application security expertise should become a key element in the evaluation of outsourced application partners. Ensure that you work only with partners that have been formally validated by an independent quality seal of approval and use secure development tools in their development lifecycle. (For example, see details on the VerAfied security mark.)

How Veracode Can Help

Veracode's Vendor Application Security Testing and Software Composition Analysis help you get visibility into the third-party applications and code in use at your organization and assess the security of this external software.

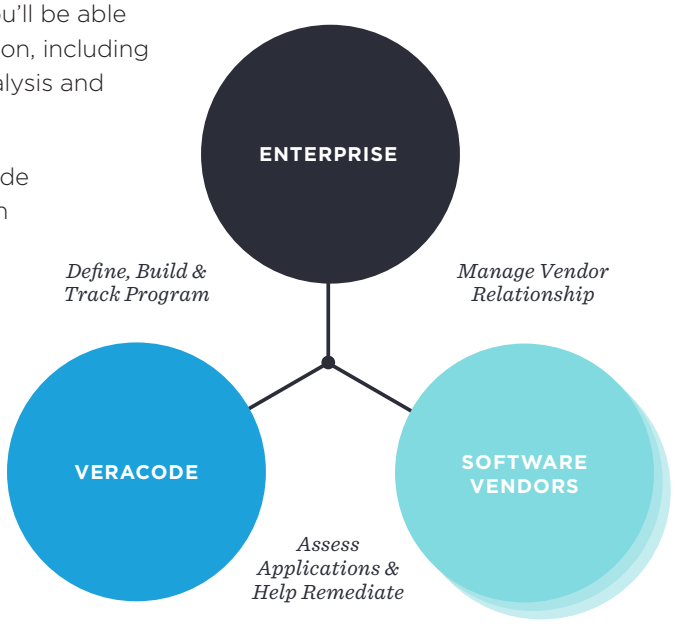
Veracode Vendor Application Security Testing

Veracode Vendor Application Security Testing (VAST) provides a scalable program for managing third-party software risk. Because Veracode scans binaries rather than source code, vendors will be more comfortable with the assessments because they don't have to disclose their intellectual property.

We work with you to formulate a strategy for contacting your independent software vendors, defining policies for compliance that can include a mix of automated and manual testing methods, and getting them into compliance. Once you have reached out to your software vendors based on our proven process, we'll handle the rest of the program management, including follow-ups with vendors, assessments, and removing any roadblocks to compliance. If you already have a vendor assessment program, we can help you to improve and scale it. In addition, our application security consultants are available to developers who need coaching on how to address vulnerabilities. Veracode can even review software vendors' mitigation proposals to provide you a qualified third-party opinion that will stand up to auditing scrutiny.

No matter how complex your corporate policy is, you'll be able to see a simple pass or fail for each vendor application, including static and dynamic scans, software composition analysis and manual penetration tests.

Your entire program is managed through the Veracode Application Security Platform, which provides you an overview of all of your vendors' compliance status. The platform helps foster collaboration between Veracode, the software vendors, and you to track progress and results. In addition to seeing a simple pass/fail, you'll be able to access detailed reports on each application.

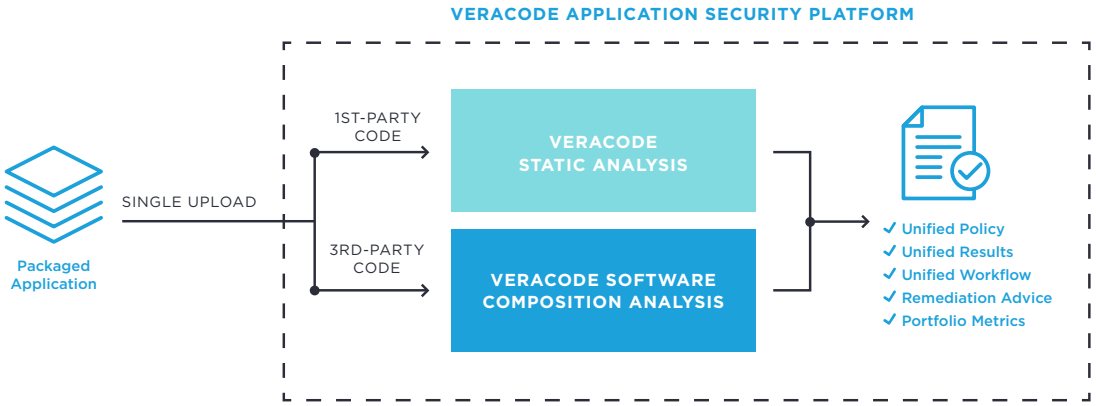


Veracode Software Composition Analysis

Veracode Software Composition Analysis (SCA) helps you build an inventory of your open source components to identify vulnerabilities.

When a big vulnerability hits the news, Veracode helps you quickly identify which applications in your organization are vulnerable. Once you find a vulnerability in an open source component, you can immediately see whether the latest version of the component addresses it. This saves precious time as you're formulating your action plan. You can also manually blacklist certain components, leading to an automatic policy audit fail for any application that uses it.

And SCA's results are not isolated from the rest of your scan results; the Veracode Application Security Platform analyzes both proprietary and open source code in a single scan, providing you visibility across your entire application landscape.



CONCLUSION

The pressure to produce more software faster is only going to increase, and with it the dependence on third-party software. But quality software is secure software, and customers and regulators will increasingly demand quality, secure applications — regardless of how an organization puts them together. Organizations need to ensure, and prove, that they are doing everything possible to protect every piece of code they use from cyberattack.

Find out how we can help; sign up for our weekly Platform Demo, or Contact Us.