VERACODE GUIDE

A BEST PRACTICE GUIDE TO

Managing Your Open Source Risk

VERACODE

The use of open source software (OSS) is commonplace among software developers across all industries.

With the benefits of expedited development timelines and speed to market, it's not surprising. But many of these components contain vulnerabilities that leave organizations exposed to risk. In fact, our most recent *State of Software Security report* found that 88 percent of all Java applications contain at least one vulnerable component. This high number of risky components combined with the fact that most organizations don't even know which components they are using creates a sobering reality.

There are nearly 1,100 new open source projects being brought to market every day — how do you keep up with that? Veracode has compiled a few key practices that every organization should consider regarding open source use.

KEY PRACTICE #1

Share accountability between security and development.

Don't make the mistake of assuming that open source security solely rests on the shoulders of your security team. Conversely, a quick way to disenfranchise your development team is to start pointing fingers about any risky open source libraries they may be using. Managing open source risk is a shared responsibility in that security needs to empower developers to take ownership of managing the open source code they embed in their applications. This means collaboration to ensure that security goals are understood across both teams; prioritizing security education for developers; and empowering dev teams with the tools they need to test their code early in the development process.

KEY PRACTICE #2

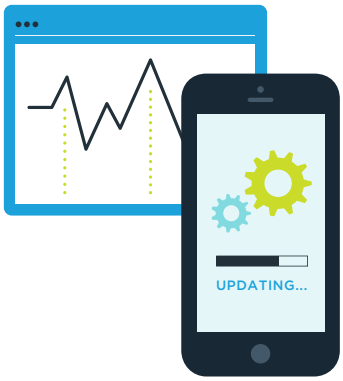
Understand your application inventory.

A good starting point when building an open source security program is taking inventory of your applications. Code repositories, build servers, and web discovery tools are great ways to collect metadata to track what applications you have deployed, what OSS components they use, which ones are under active development, and which teams manage them. These bits of information are critical to understand where you should focus your efforts when a new vulnerability is disclosed.

KEY PRACTICE #3

Create a concise OSS security policy.

A concise open source security policy is a surefire way to ensure that your entire team is on the same page when it comes to understanding acceptable risk. Establishing benchmarks on the severity of vulnerabilities found in your open source libraries (usually indicated by a CVSS score) will provide a more efficient means to prioritize remediation and mitigation workflows. Some security solutions, such as Veracode's Software Composition Analysis, will allow you to establish component blacklists to ensure that risky versions of open source libraries are barred from being implemented into your application portfolio.



KEY PRACTICE #4

Manage your technical debt.

It is typical for development teams to neglect updates to their open source libraries within their production apps. This is understandable considering that updates can require code modifications, which require retesting. However, as subsequent versions of open source projects are released, your team amasses technical debt. Why is this important? Vulnerabilities for legacy OSS versions are regularly disclosed on public forums like the National Vulnerability Database (NVD), and because they're public record, the attackers are fully aware of them as well. To get ahead of this, make sure your team is consistently making the necessary updates as new OSS versions are released. This will save you from having to pay off that technical debt all at once and save your team a lot of time and investment. At a minimum, you should understand what technical debt you are accumulating.

This becomes even more important for software vendors who have a decision to make on how their legacy applications will be supported. Should a new vulnerability be disclosed that affects an older version of your product, your customers are going to want to know that since switching over to the latest version may require some work on their part. Build credibility by being transparent about how you handle your open source as well as your patch and communication plan in the event of a disclosure.

KEY PRACTICE #5

Assign security champions and incident response teams.

It's best practice to have someone with a deep understanding of both security and development to act as an intermediary between the two organizations. These "security champions" can be a tremendous asset to help curb open source risk as well. When you are maintaining a database of your open source libraries to stay ahead of risk, your security champions can be your lifeline to help maintain this critical information. They can help you understand what OSS libraries are being used by their team, the version, if there are vulnerabilities currently susceptible to exploitation, and they can captain your incident response team should a new vulnerability be released. Ultimately, this helps establish timelines to address vulnerabilities and also makes your application teams far more self-reliant.

KEY PRACTICE #6

Test your code early in the SDLC…and after production.

As more rapid development methodologies like DevOps become mainstream, many security professionals are wondering how to incorporate security into the development lifecycle. Just as with testing your own proprietary code, testing for open source vulnerabilities early can help minimize risk once it's time to ship to production. It doesn't stop there, however. Continuous monitoring post-deployment via dynamic analysis can help identify vulnerabilities that can be exploited through the open web as well. It's also important to keep in mind that static and dynamic testing may not uncover open source vulnerabilities that are only exploitable through business logic. Incorporating human testers into your ongoing security testing routine can help you gain an understanding of your OSS pedigree and determine an action plan should you be susceptible to an attack.

Open source code is a valuable asset that has completely changed the way modern applications are built.

However, adopting open source also comes with a degree of responsibility. Take full advantage of the benefits while also being wary of the risk; there's no such thing as a free lunch.

For more information on how Veracode can help you get a handle on open source risk, visit our website.

VERACODE

Veracode is a leader in helping organizations secure the software that powers their world. Veracode's SaaS platform and integrated solutions help security teams and software developers find and fix security-related defects at all points in the software development lifecycle, before they can be exploited by hackers. Our complete set of offerings help customers reduce the risk of data breaches, increase the speed of secure software delivery, meet compliance requirements, and cost effectively secure their software assets — whether that's software they make, buy or sell. Veracode serves over a thousand customers across a wide range of industries, including nearly one-third of the Fortune 100, three of the top four U.S. commercial banks and more than 20 of the Forbes 100 Most Valuable Brands.

Learn more at veracode.com, on the Veracode Blog, and on Twitter.