
EVERYTHING YOU NEED TO KNOW ABOUT
OPEN SOURCE RISK
VERACODE

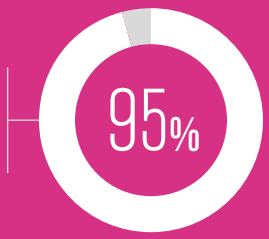
OPEN SOURCE REPRESENTS OPPORTUNITY AND RISK

Open source frameworks have changed the business world in profound ways. They've ushered in a level of speed, innovation, and convenience that significantly alters the IT equation. With large numbers of developers and others contributing to a project, it's possible to advance and evolve software in ways that wouldn't have been imaginable in the past. What's more, this form of open collaboration benefits everyone by making software available at a lower cost point — and sometimes even at no cost.

Make no mistake, open source software libraries are here to stay, and are now being used across industries and throughout governmental and educational organizations. One study found that 96 percent of all scanned applications contain some open source components and that the use of open source code increased from 36 percent to 57 percent over a one-year period ending in 2017.[1] What's more, a typical software application now contains an average of 257 open source components. These may span both development languages and toolkits. In the end, this equates to applications being composed of up to 90 percent open source code.

FACT

95% of IT organizations now rely on open source software.[2]



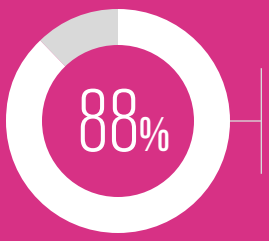
But there's a dark side to open source frameworks:

They can introduce new and sometimes dangerous risks to an enterprise. The use of open source code increases the number of users affected as well as the number of exposure points. The impact of the 2014 **Heartbleed security bug** proved that out all too painfully, attacking upwards of 800,000 websites.

1 HelpNetSecurity. "**The percentage of open source code in proprietary apps is rising**." May 22, 2018.
2 Veracode. ***Everything You Need to Know about Maturing Your Application Security Program***.

At this point, the question isn't *Will my organization and software developers use open source? It's What libraries are we using? How and where are we using them? Do they contain vulnerabilities? Do the vulnerabilities impact my application?* and *How does the use of open source code change our risks and responsibilities?*

To answer these questions, it's vital to have a strategy and framework in place to manage open source libraries and components. Otherwise, the road to digital transformation will likely be paved with frustrations, problems, and even failures. Making matters worse, these vulnerabilities don't only affect your organization, they potentially undermine your customers and clients.



FACT

Nearly 88% of Java applications have at least one vulnerability in a component.[3]

Find out more about how open source components build risk into apps by checking out these resources:

GUIDE
Components: Increasing Speed and Risk

VIDEO
"How Components Build Risk into Apps"

VIDEO
"The Good and the Bad of Code Reuse"

3 Veracode. *State of Software Security, Volume 9.*

5 THINGS TO KNOW ABOUT **OPEN SOURCE**

1. Open source is now a reality for all development teams.

2. Agile and DevOps have changed the equation. It's nearly impossible to build software entirely from scratch and still meet delivery deadlines.

3. Building software entirely from scratch also leads to higher development costs.

4. Open source offers essential and sometimes leading-edge software capabilities that can't be otherwise achieved.

5. Open source also carries potential risks, making a strong security policy the center of effective development today.

ANNUAL REPORT

For more information about trends in software risks and vulnerabilities, review the latest version of our annual report, *State of Software Security, Volume 9*.

OPEN SOURCE RISK EVOLVES

Business and technology are undergoing unprecedented changes. The cloud, mobile technology, the Internet of Things (IoT), and fundamental advances in processing power have introduced opportunities to take your organization to greater heights. But, increasingly, gains are predicated on moving faster and faster. Open source code and software, while delivering speed and flexibility that align nicely with development models like DevOps, also shift who's responsible for coding quality and vulnerabilities that may lead to security gaps and breaches.

But the challenges don't stop there. Numerous questions revolve around which specific libraries, components, and code an enterprise uses. Do these libraries contain vulnerabilities, and if so, what threat do they represent? Not all vulnerabilities are created equal, so it's important to understand whether the vulnerability is actually dangerous and, if so, what level of damage it can cause. These challenges are exacerbated by the use of both direct and indirect open source libraries. In some cases, these indirect dependencies may extend five or 10 levels deep into a project or application.

Simply using open source libraries isn't a security threat to the business.

The real problem is not knowing that what you're using contains vulnerabilities and that they're exploitable in your application.

For development teams, the idea of managing open source code and protecting against security risks effectively can seem overwhelming. However, it's simply not feasible to generate every piece of code for every application from scratch. In fact, open source use leads to significant competitive advantages. **Leveraging open source allows developers to spend 90 percent of their time on the 10 percent of the application that differentiates you from your competitors** — the more time developers can spend there, instead of building table stakes functionality that can be provided by OSS libraries, the better.

It's also not possible to thoroughly vet every source and code library prior to downloading components — there are just too many open source components being used. For instance, developers downloaded more than 52 billion Java components and more than 12 billion Docker Hub components in 2017.[4] Over the past five years, the use of open source software increased by a factor of 5x.[5] Yet it's possible to construct a framework that minimizes risks and maximizes protections. It just requires a clear strategy as well as a practical framework for addressing open source risk.

GUIDE

Use our guide, *Understanding Your Open Source Risk*, to build a better open source security framework.

4 BrightTALK. "The Open Source Library Conundrum: Managing Your Risk."
5 Ibid.

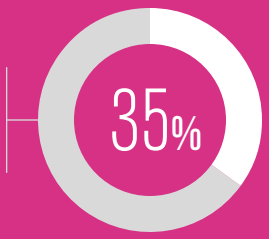
DEVELOPING AN OPEN SOURCE SECURITY STRATEGY IS VITAL

Open source has emerged as one of the most essential tools for enterprise software development. Today, online libraries, modular components, and pre-built code serve as the foundation for DevOps and other enterprise initiatives. The challenges related to using open source code effectively and safely revolve around identifying new and different types of threats, risks, and problems quickly and then taking action to address the vulnerabilities promptly. Organizations that wait for external fixes in the open source library or believe that simply having more "eyes on the code" to spot problems may be lulled into a false sense of security.

Open source software risks revolve around three key areas: visibility, security, and governance. Understanding these factors and how they ripple through open source security helps an enterprise formulate a stronger cybersecurity strategy.

FACT

Mature AppSec programs have a 35% higher policy pass rate than new programs.[7]



Being able to answer the following questions about your organization's coding practices in each of the three key areas is vital to creating a cybersecurity strategy that effectively protects your software:

VISIBILITY

SECURITY

GOVERNANCE

7 Veracode. *Everything You Need to Know about Application Security Policies*.

VISIBILITY

Where are open source software components in use?

It's crucial to have complete visibility into your organization's use of open source software and code. In the open source world, however, the problem is particularly tricky because organizations often rely on libraries that, in turn, rely on other libraries that rely on other libraries (and so on and so on). So, even though a developer could be pulling in only a few open source libraries directly, those libraries could easily pull in hundreds of other open source libraries with them — including all of their vulnerabilities. In addition, while one segment of a library may not be vulnerable, other subsets may be, and it's critical to know that in order to understand how to proceed once you find out a library has a vulnerability.

What open source versions do we have in use?

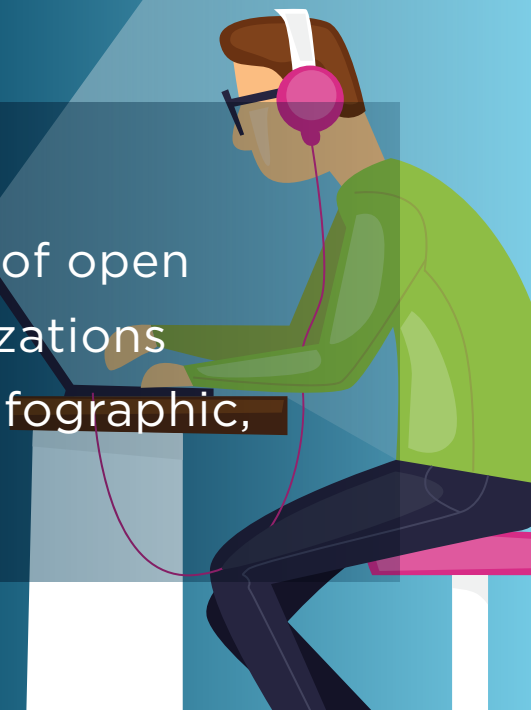
It's also necessary to understand the versions of the components in use. Are they the most recent? Are they older? It's a mistake to assume that the groups within your organization automatically update to the most recent and secure versions of open source software. It's also a mistake to assume they're using the best tools to detect those vulnerabilities.

When do we check on the status of open source?

Still another concern is how often development teams and other groups are checking on patches and updates. The period between when a coding flaw or vulnerability is detected and when it's fixed is crucial. Adding to the problem: The National Vulnerability Database is overrun with submissions, and it can take several months to sort through submissions and officially disclose a vulnerability.

INFOGRAPHIC

Get a quick look at the use and risks of open source components and what organizations can do to protect their code in our infographic, *Reining in Software Component Risk*.



SECURITY

What patching policies do we have in place?

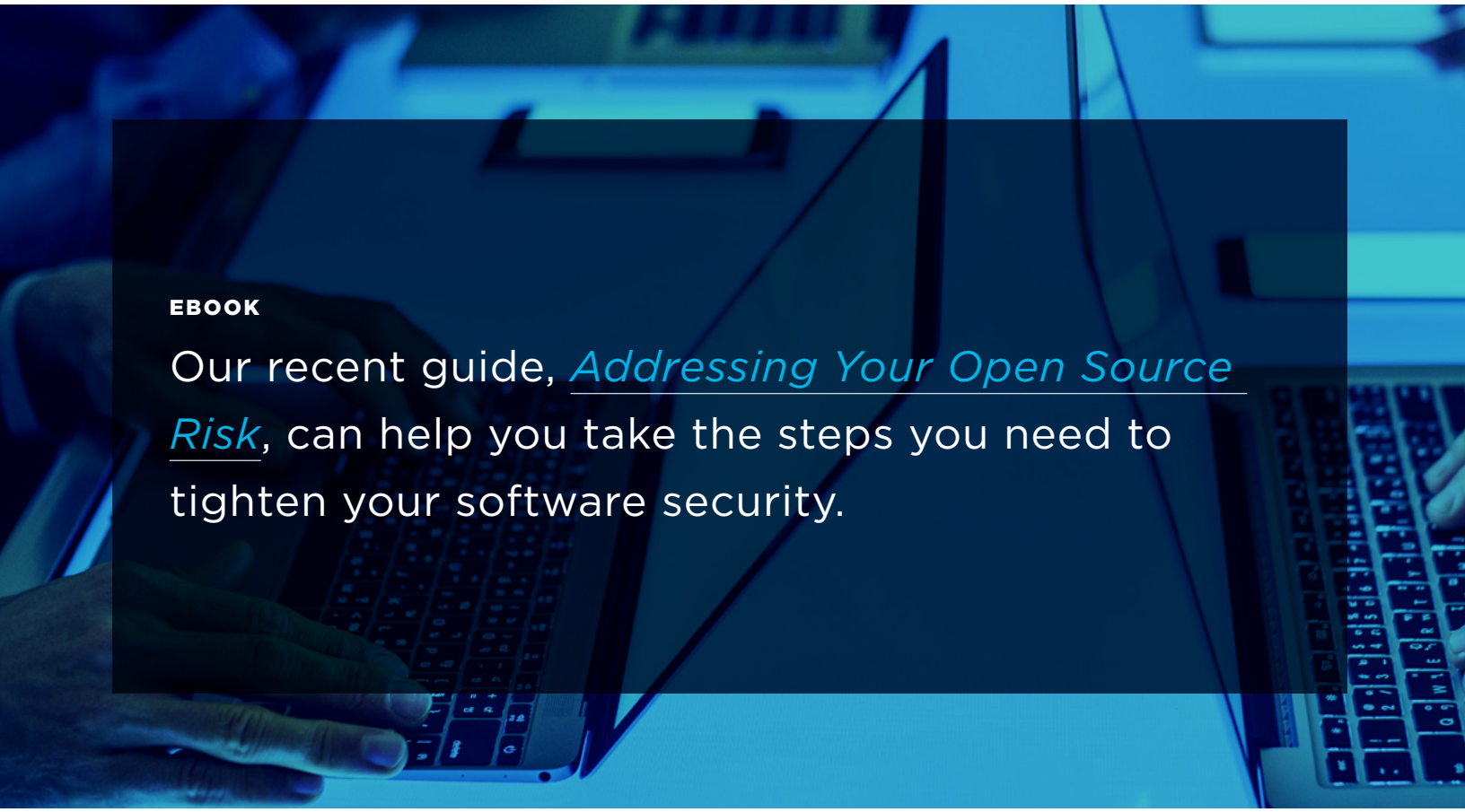
It's one thing to check on the status of open source updates, it's another to ensure that patches and updates take place in a timely manner. A clear policy and established procedures must be in place to oversee code patches and updates.

What is our vulnerability management approach?

There's also a need to understand what happens and how to react in the event of an issue with open source code (or any other software). This framework can help your enterprise navigate the inevitable coding vulnerabilities that occur and address them in a prompt and effective manner.

Do we have testing and validation systems in place?

This is one of the most crucial aspects of application security. It does no good to simply know there's a problem with open source code. It's more important to understand how it impacts your organization.



EBOOK

Our recent guide, *Addressing Your Open Source Risk*, can help you take the steps you need to tighten your software security.

GOVERNANCE

Do we fully understand the implications of copyrights and licensing?

Licensing terms are a critical issue. It's important to understand how the software can be used, how it can be modified, and how so-called "copyleft" licensing affects usage and modification.

Does the open source software we're using match our organization's compliance policies?

A critical element of open source usage is to ensure that it's used in accordance with your organization's policies and procedures — and that necessary controls are in place to ensure compliance.

How do we control whether insecure libraries make it into production?

With the ability to integrate security into development tool chains, you can stop development from proceeding when critically vulnerable libraries are causing your application to be exploitable.

GUIDE

To learn more about effective governance and building a culture that focuses on application security, check out our guide, *Everything You Need to Know About Getting Application Security Buy-In*.

VERACODE'S APPROACH TO MANAGING OPEN SOURCE RISK

A fundamental problem for organizations is balancing the need for developers to move fast and generate code and for security teams to lock down protections and avoid breaches. These two goals don't have to conflict, however. By rethinking and rewiring processes — and putting the right framework in place — it's possible to detect problems and address them promptly. Here are seven essential strategies for securing open source:

Establish deep visibility into open source and other code.

Understanding your application inventory is critical. Veracode research shows that only half of all organizations maintain inventories of components and subcomponents. An equal percentage recognize the need to update components even when vulnerabilities are made public.[8] The upshot? A complete inventory of open source code is a good starting point for securing the enterprise.

Share accountability between security and development groups.

These days, the management of open source code extends beyond your development teams. In fact, putting development teams in charge of security is a surefire way to undermine protection of your code. A best practice methodology approaches protection from a shared perspective — communication and collaboration are essential in order to prioritize risks and fixes, choose tools, and orchestrate education and training.

Create a concise and focused open source security policy.

It's a lot easier to address security risks when everyone is marching in the same direction. An organization benefits when it introduces benchmarks and metrics, establishes a set of priorities for remediation, and puts processes and workflows in place to support this framework. The right tools, such as software composition analysis, can help establish blacklists for high risk versions of open source code.

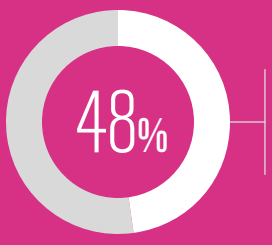
8 Veracode. "The Open Source Library Conundrum: Managing Your Risk."

Manage technical debt. It's important to have policies and procedures in place so that teams stay ahead of attackers. Although open source libraries and code are constantly updated, patched, and fixed, attackers can also see what's posted in the National Vulnerability Database. This means that it's vital to consistently monitor and make changes as new open source updates take place.

Establish security champions. The ability to understand the business and development needs of your organization, along with the security required to protect its assets, can go a long way toward strengthening protection. What's more, establishing a security champion on your development team can aid in translating and disseminating technical data and ensuring that your organization is effectively addressing vulnerabilities in open source code and libraries.

Test code early and often. More than anything, it's critical to test open source code at all stages of development and deployment. Continuous monitoring through static and dynamic testing can find vulnerabilities and errors that may otherwise go undetected. In some cases, they may not be apparent unless an organization uses specific business logic.

Use a solution that gives you actionable results. Veracode SCA takes security to the next level by using a proprietary vulnerability database based on data mining, proprietary machine learning, and our security research process. With this database, you get advanced notice of vulnerabilities, far earlier than if you were to only rely on the NVD. In addition, our solution helps you prioritize which vulnerabilities to address through our vulnerable methods capability; you'll know if you are actually using a vulnerable part of a library and won't spend time fixing issues that don't matter.



FACT

Developers scanning code early and often fix 48% more flaws than those who don't.[9]

9 Veracode. *Everything You Need to Know about Application Security Policies*.

RESOURCES

Discover best practices related to managing open source code and maximizing security in our report:

→ *A Best Practice Guide to Managing Your Open Source Risk*

Learn practical steps for reducing open source component risk without slowing down delivery in our webinars:

→ "How to Manage Open Source Risk Within Your Application Security Program"

→ "Building Blocks of Secure Development"

CODING FOR SUCCESS

Although there are many issues and aspects related to open source security, in the end, it all comes down to a few basic goals:

1. Identifying and cataloguing all open source and commercial code
2. Putting tools and processes in place to identify vulnerabilities
3. Using specialized tools to address risks and problems

Static and dynamic scanning, along with tools such as Veracode SCA, can accomplish this task. They establish a robust framework for managing code and using automation to identify vulnerabilities. An organization that adopts this approach is far better equipped to use open source at maximum advantage and minimum risk.

In today's Agile, Dev-Ops world, this level of protection isn't simply a good idea, it's paramount.

RESOURCE ROUNDUP

Find out more about how open source components build risk into apps:

GUIDE

Components: Increasing Speed and Risk

VIDEOS

- "How Components Build Risk into Apps"
- "The Good and the Bad of Code Reuse"

Learn more about software risks and vulnerabilities:

State of Software Security Volume 9

Use our guide to build a better open source security framework:

Understanding Your Open Source Risk

Find out more about what your organization can do to protect its open source code:

Reining in Software Component Risk

Our recent guide can help you take the steps you need to tighten your software security:

Addressing Your Open Source Risk

Build a culture that focuses on application security with smart tips from our guide:

Everything You Need to Know About Getting Application Security Buy-In

---

Discover best practices related to managing open source code and maximizing security:

A Best Practice Guide to Managing Your Open Source Risk

Learn how to reduce open source component risk without slowing delivery:

WEBINARS

- "How to Manage Open Source Risk Within Your Application Security Program"
- "Building Blocks of Secure Development"

Take your AppSec initiative to a higher level by reading our guide:

Everything You Need to Know about Maturing Your Application Security Program

---

Find out how Veracode can help protect your company from open source risk:

Solving Your Open Source Risk with SourceClear

See what our experts have to say about getting your software to market promptly and safely:

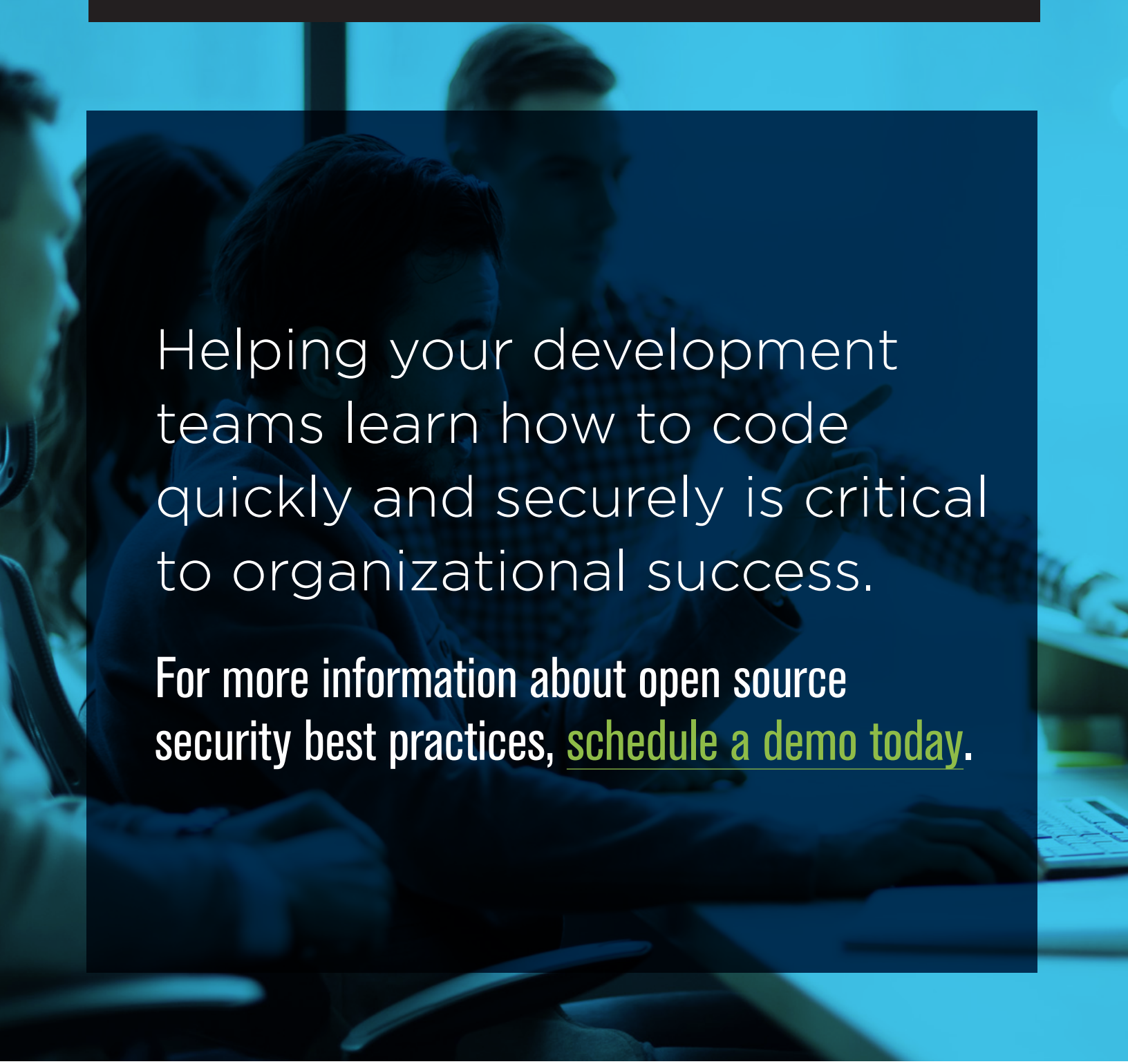
"Open Source Code — A Blessing or a Curse?"

Learn what your development teams need to do to code quickly and securely:

"The Open Source Library Conundrum: Managing Your Risk"

See how one vulnerable component can grow into generations of risky software:

Family Tree of Vulnerabilities



Helping your development teams learn how to code quickly and securely is critical to organizational success.

For more information about open source security best practices, schedule a demo today.

VERACODE

Veracode gives companies a comprehensive and accurate view of software security defects so they can create secure software and ensure the software they are buying or downloading is free of vulnerabilities. As a result, companies using Veracode are free to boldly innovate, explore, discover, and change the world.

With its combination of automation, integrations, process, and speed, Veracode helps companies make security a seamless part of the development process. This allows them to both find and fix security defects so that they can use software to achieve their missions.

Veracode serves more than 2,000 customers worldwide across a wide range of industries. The Veracode Platform has assessed more than 8 trillion lines of code and helped companies fix more than 36 million security flaws.

Learn more at **www.veracode.com**, on the **Veracode blog**, and on **Twitter**.