VULNERABILITY DECODER

Insecure Open Source Components

The Vulnerability

Open source and commercial, third-party components are the building blocks of applications, but developers frequently don't know which components are in their applications, making it difficult to update components when new vulnerabilities are discovered. That leaves applications vulnerable to attackers who can exploit an insecure component to take over the server or steal sensitive data.

97%

of the typical Java application is made up of open source libraries.

Source: SOSS v11

The Risks

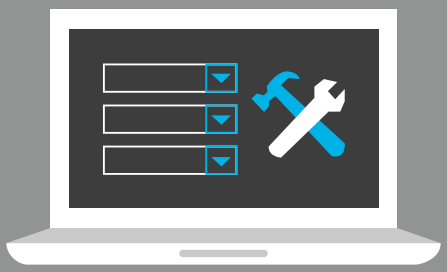
Open source code presents all the same risks as code developed in-house, but it can be much more difficult to maintain visibility into what components you're using and where. This can lead to vulnerabilities in components remaining hidden for a long time.

Example Breach

Attackers exploited a critical vulnerability in the Apache Struts 2 library to access data from the Canada Revenue Agency's web server, putting taxpayers at risk of identity theft. The same vulnerability has been linked to the Equifax breach that exposed data of 145 million consumers.

Prevention & Remediation

Preventing vulnerabilities in open source components requires vigilance in maintaining an inventory of all the components you're using, and updating those components when new vulnerabilities are discovered. The following recommendations from **OWASP** can help you reduce risk.



- ☑ Remove unused dependencies, unnecessary features, components, files, and documentation.
- ☑ Only obtain components from official sources over secure links.
- ☑ Continuously inventory the versions of both client-side and server-side components and their dependencies.
- ☑ Continuously monitor sources like CVE and NVD for vulnerabilities in the components.
- ☑ Subscribe to email alerts about vulnerabilities in the components you use.
- ☑ Avoid using libraries or components that are not maintained with regular updates.
- ☑ Use software composition analysis tools to automate the process of identifying components with known vulnerabilities.

Recommendations

- ☑ Get training in secure coding best practices, through on-demand eLearning courses, in-person security consultations, and professional development certifications and conferences.
- ☑ Use application security tools that integrate with your IDE, allow you to scan small batches of code instantaneously, and provide remediation guidance while you code.

Download the Secure Coding Best Practices Handbook

Join the Veracode Community
community.veracode.com

VERACODE