Sonatype

CLOSE UP

NEXUS FIREWALL

QUALITY AT VELOCITY

By Mike Hansen, Sonatype SVP of Products

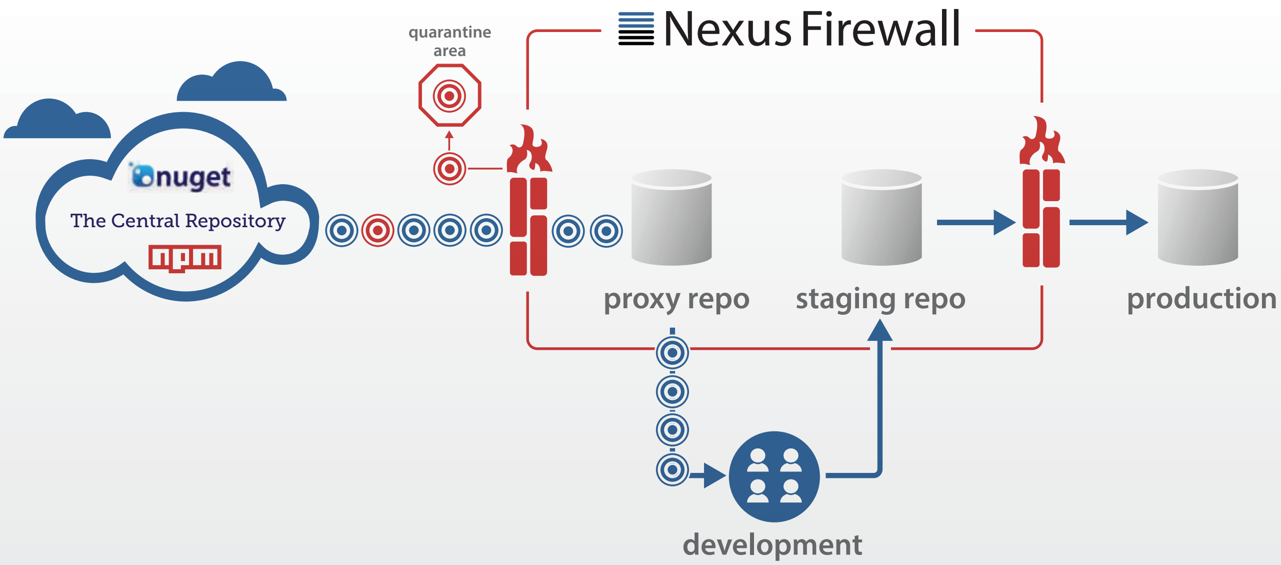
The quantitative research summarized below, covering over 7,000 repositories across nearly 100 countries, highlights some of the challenges with quality at modern development velocities. Respond by leveraging automation in your repository manager to improve application quality and reduce unplanned work while lowering exposure to risk.

Repository managers like Nexus and Artifactory have been serving software components for development teams and their tooling for years now. Now, we are introducing an innovative way to improve speed and reduce risk through the quarantine of components with known vulnerabilities. With the integration of Nexus Firewall, you can shield your application development from waste and risk by automatically blocking unacceptable software components inbound and preventing release of applications containing such components outbound.

The Firewall also goes beyond blocking, providing organizations with the visibility and data needed to make ideal decisions for open source component selection early, significantly reducing waste related to rework and eliminating avoidable risk.

What is Nexus Firewall?

Nexus Firewall offers perimeter quality control for software development. Similar to a network firewall, it leverages rules you define that automatically shield you from unacceptable software components entering and another set for stopping them from exiting your application development. The basic concept looks like this:



Why the Repository?

The repository manager has become a critical piece of the DevOps toolchain and is commonly used across the entire application life cycle. It is effectively a binary parts warehouse for your applications. Given how integral this warehouse has become, automation can be used to significantly reduce unplanned work and eliminate avoidable risk.

To understand this opportunity, let's take a quantitative look at the world's software development ecosystem.

Global Visibility

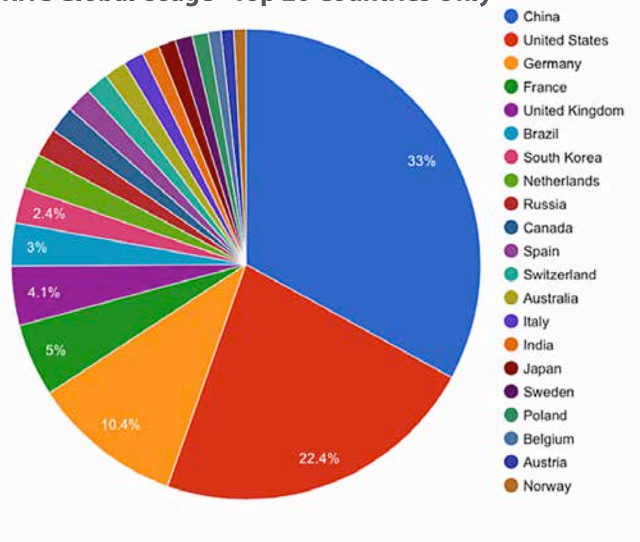
Not long ago Sonatype introduced Nexus RHC (Repository Health Check) – a simple way to get basic visibility into what OSS is flowing into development via your repository manager. Usage of this service is now significant, with over 30 million components across over 15,000 Nexus instances being analyzed every day across the globe.

There are also around a million developers using those repositories. With the law of big numbers working for us, we can gain a statistically valid view into the behaviors of the modern software development ecosystem. The findings are quite telling.

The Data

We analyzed over 7,000 repositories containing 500 or more software components using the RHC service during the past 90 days. These repositories are representative of those used by development teams in medium to large organizations. The question was straightforward: To what extent are bad things flowing in, causing downstream rework and creating avoidable risk scenarios?

RHC Global Usage - Top 20 Countries Only



In three months time, the average number of new vulnerabilities that flowed into the repositories analyzed was 69. That is an average of 23 vulnerabilities per month, which is a little more than one every weekday. If you only include higher risk vulnerabilities – those with a CVSS score of 5 or greater – the average number was 48, or about 16 per month. That is a little less than one every weekday. For a sense of the kind of vulnerabilities in this higher risk category, the Heartbleed bug had a score of 5, the lowest risk in this grouping.

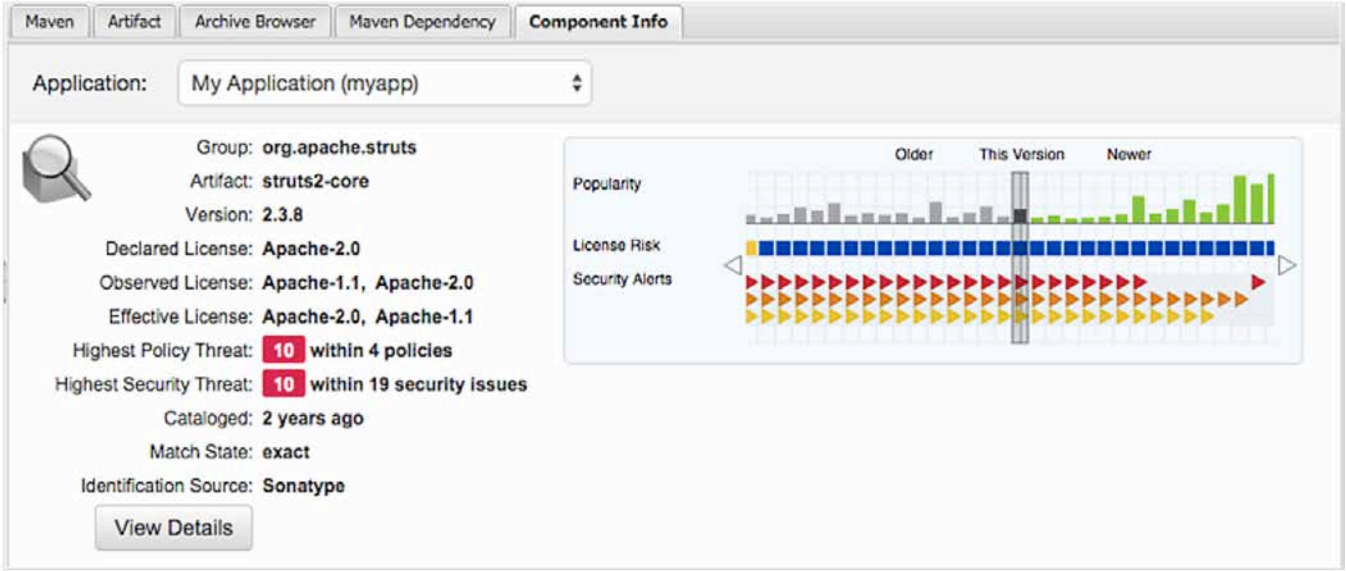
These are not isolated instances or outliers. In the set of repositories analyzed, 98.3% consumed at least 1 vulnerable component and 97.7% consumed at least 1 higher risk (CVSS >= 5) component. In a period of just three months, nearly every repository consumed something the organization running it would rather not have, and that is only with respect to known vulnerabilities. There are plenty of other attributes that are worthy of consideration given availability of the corresponding data.

Whether or not security is of particular concern, the real point is that a lack of visibility and control combined with an abundance of supply has led to inferior quality parts routinely being used by the world's development organizations, adding avoidable risk and slowing us down.

Enter Nexus Firewall

The consumption of these unacceptable or, at a minimum, inferior components is now unnecessary. Avoiding this is one of the primary benefits provided by Nexus Firewall. For the vast majority of these vulnerable components, there are non-vulnerable alternatives ready to be used in their place. The challenge is that it is difficult for developers to understand these risks and even harder to avoid them.

Avoiding risk can now be done through automation, and when combined with the remediation capabilities provided by the Firewall, drive the right choices early in the development process. An example of the kind of information made available is shown below.



Application: My Application (myapp)

Group: org.apache.struts
Artifact: struts2-core
Version: 2.3.8
Declared License: Apache-2.0
Observed License: Apache-1.1, Apache-2.0
Effective License: Apache-2.0, Apache-1.1
Highest Policy Threat: 10 within 4 policies
Highest Security Threat: 10 within 19 security issues
Cataloged: 2 years ago
Match State: exact
Identification Source: Sonatype

The information is divided into two areas. On the left side is component data, which includes details related to the component itself. To the right, there's a graphical display of any security or license issues, as well as popularity data for each version of the component displayed. Selecting different versions updates this information accordingly, including whether or not a particular version passes the established criteria.

Going (Way) Beyond Security

The consumption of vulnerable components is a quantifiable problem that we used to illustrate the current challenges that every organization has with open source component selection. However, security vulnerabilities are only one of the many dimensions supported by Nexus Firewall.

Others include architectural aspects such as rules covering component age and popularity, intellectual property coverage with rules related to open source licensing and technical debt controls associated with technology stack selection. A comprehensive ruleset will yield significant overall quality and efficiency benefits for an organization, allowing them to operate with a lower risk profile.

Quality at Velocity – Just Flip the Switch

Simply by enabling Nexus Firewall, organizations can immediately improve quality and reduce waste and exposure to risk. None of this comes at the expense of development speed.

Firewall automatically quarantines components that do not pass your rules preventing quality issues from entering the software being developed, immediately reducing risk and avoiding wasteful rework at some later point. Quality is improved, efficiencies are gained and risk is reduced, all through automation and all at the speed of modern development.