WhiteSource

WHITESOURCE PRIORITIZE

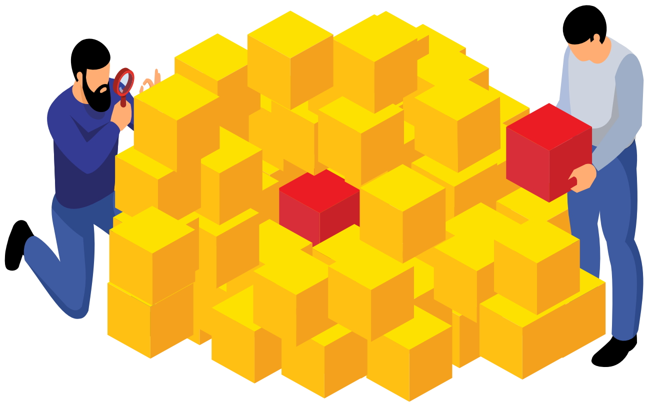
AN EXAMPLE FOR PRIORITIZING VULNERABILITIES

Software vulnerabilities have been on the rise in recent years, creating a challenge for developers to keep up with securing their applications. At the center of their struggle is the inability prioritize which of their open source vulnerabilities are the most pressing, due to the fact that they lack the visibility into which of their alerts are pointing to effective vulnerabilities.

In practice, we know that not all reported vulnerabilities are effective and demand your immediate attention since a vulnerability is effective only if your proprietary code is making calls to the vulnerable method. Based on our research, only 15% to 30% of the vulnerabilities are indeed effective.

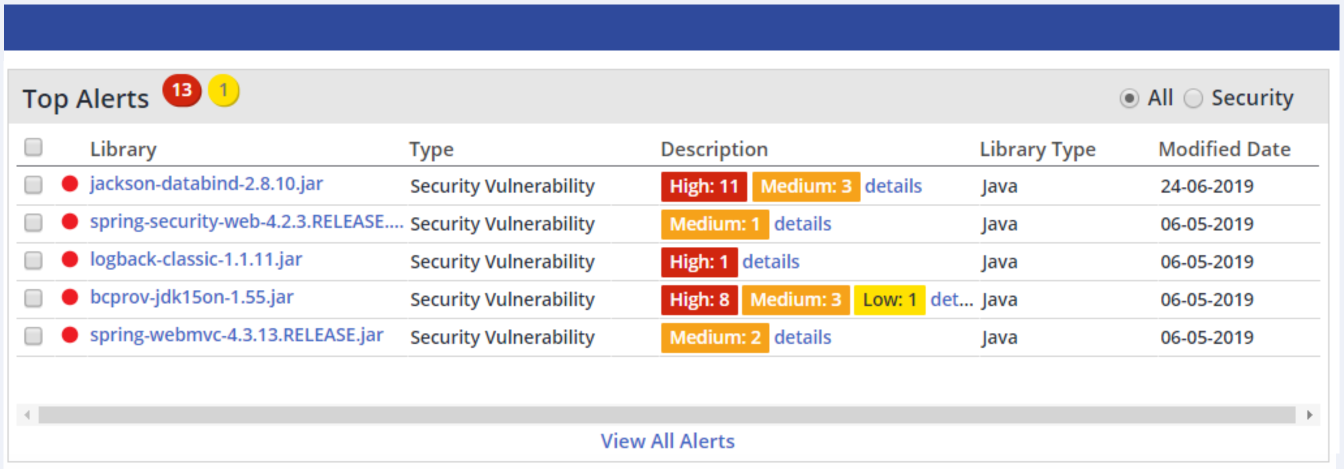
With WhiteSource Prioritize, you will be able to get a clear understanding of which of the reported vulnerabilities effectively impact your code, showing you which open source reported vulnerabilities are actually referenced from the proprietary code and directly impact your open source components.

Spending too much time researching vulnerabilities?
Focus on what matters with WhiteSource Prioritize!



WHITESOURCE PRIORITIZE IN ACTION

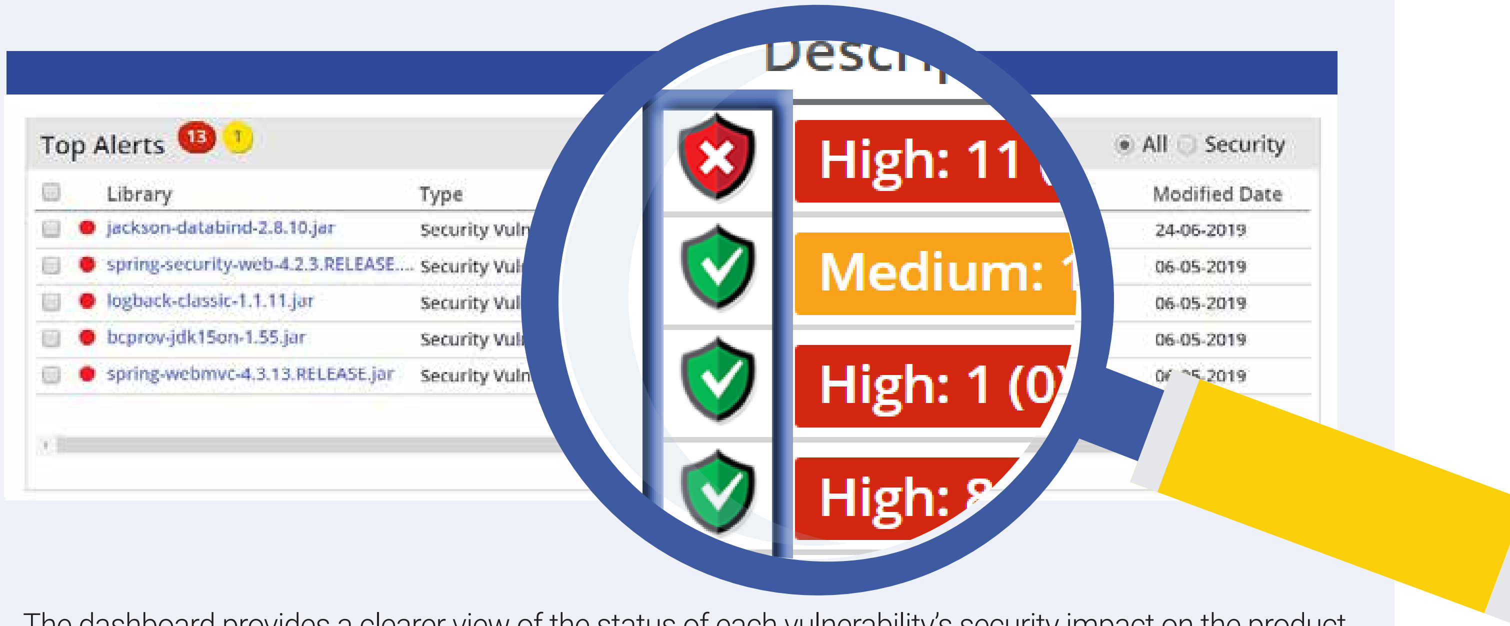
To demonstrate the power of WhiteSource Prioritize, we took a project in Java and analyzed it with WhiteSource to see if there were any reported vulnerabilities. The analysis turned up 13 security alerts but did not give us an indication of which of these vulnerabilities we should investigate first.


Top Alerts 13 1 [✓ All] Security

	Library	Type	Description	Library Type	Modified Date
☐	● jackson-databind-2.8.10.jar	Security Vulnerability	High: 11 Medium: 3 details	Java	24-06-2019
☐	● spring-security-web-4.2.3.RELEASE....	Security Vulnerability	Medium: 1 details	Java	06-05-2019
☐	● logback-classic-1.1.11.jar	Security Vulnerability	High: 1 details	Java	06-05-2019
☐	● bcprov-jdk15on-1.55.jar	Security Vulnerability	High: 8 Medium: 3 Low: 1 det...	Java	06-05-2019
☐	● spring-webmvc-4.3.13.RELEASE.jar	Security Vulnerability	Medium: 2 details	Java	06-05-2019

View All Alerts

We then ran the analysis again with WhiteSource Prioritize to see which of these reported vulnerabilities were effective. Effective vulnerabilities demand our immediate attention as the proprietary code is making calls to the vulnerable functionality and are critical to resolve quickly.



The dashboard provides a clearer view of the status of each vulnerability's security impact on the product (effectiveness). This is represented using shield icons, showing us whether or not our proprietary code is actually making calls to the vulnerable functionality.

In this project, only one vulnerability is effective and we should remediate it immediately (see red shield). All the other vulnerabilities are not effective and can be deprioritized.

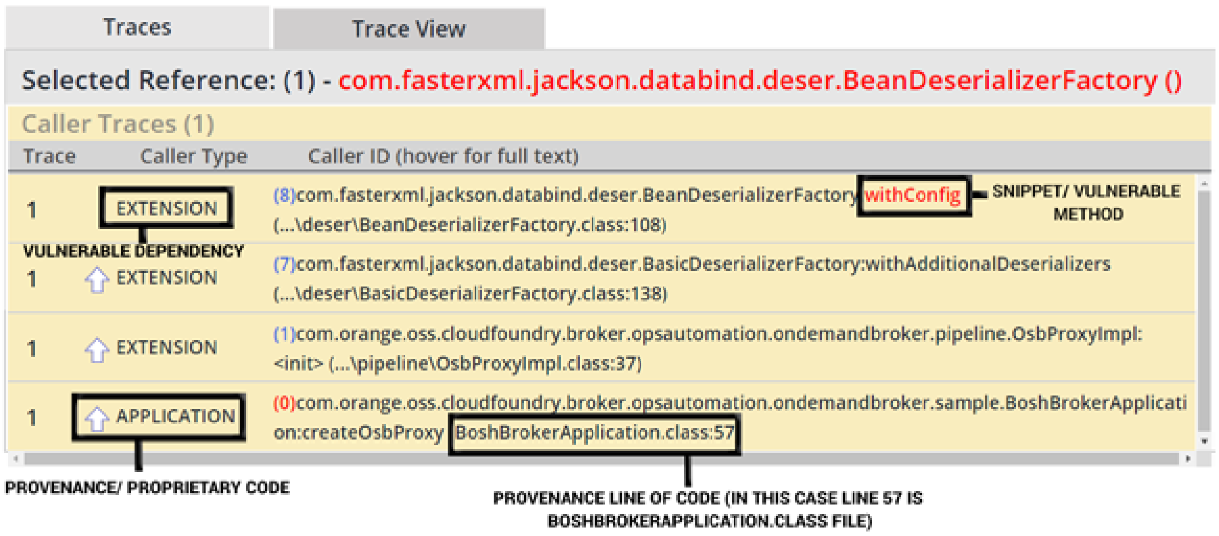
Each shield icon indicates the effectiveness level of each alert.

This is an effective vulnerability. Your proprietary code is making calls to the vulnerability.	The analyzed open source vulnerability could not be established.	This is not an effective vulnerability. Your proprietary code is NOT making calls to the vulnerability.	A new scan is recommended due to updated vulnerability information

TRACE ANALYSIS: ACTIONABLE INSIGHTS

Once WhiteSource Prioritize detects the effective vulnerabilities, it also provides a detailed trace analysis to help developers understand how they are using the vulnerable functionalities to optimize for quicker remediation processes.

The call graph shows the exact location of the vulnerability and where exactly the reference occurs in each library, i.e., filename, class name, and line in the code. It clearly shows if the vulnerable code is in a direct or indirect dependency, the exact location of the vulnerable snippet, and the provenance (the proprietary methods making the call). Notably, the system highlights both the snippet making the call (in the proprietary code) as well as the vulnerable snippet being called (in the open source code).



Traces
Trace View
SNIPPET/ VULNERABLE METHOD
VULNERABLE DEPENDENCY
PROVENANCE/ PROPRIETARY CODE
PROVENANCE LINE OF CODE (IN THIS CASE LINE 57 IS BOSHBROKERAPPLICATION.CLASS FILE)

Selected Reference: (1) - com.fasterxml.jackson.databind.deser.BeanDeserializerFactory ()

Caller Traces (1)

Trace	Caller Type	Caller ID (hover for full text)
1	EXTENSION	(8)com.fasterxml.jackson.databind.deser.BeanDeserializerFactory:withConfig (...\deser\BeanDeserializerFactory.class:108)
1	EXTENSION	(7)com.fasterxml.jackson.databind.deser.BasicDeserializerFactory:withAdditionalDeserializers (...\deser\BasicDeserializerFactory.class:138)
1	EXTENSION	(1)com.orange.oss.cloudfoundry.broker.opsautomation.ondemandbroker.pipeline.OsbProxyImpl:<init> (...\pipeline\OsbProxyImpl.class:37)
1	APPLICATION	(0)com.orange.oss.cloudfoundry.broker.opsautomation.ondemandbroker.sample.BoshBrokerApplication:createOsbProxy BoshBrokerApplication.class:57

The Trace Analysis saves your developers precious time researching the possible implications of patching or updating the vulnerable component, which will also significantly reduce the time required to resolve security vulnerabilities.

BOTTOM LINE:

Using the Effective Usage Summary Report we can see that only one of the 13 vulnerabilities that were detected in the beginning are effective and one requires urgent remediation.

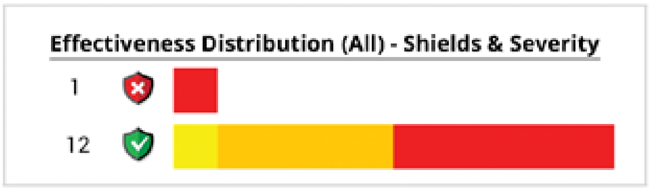
Of the six security alerts with high-severity vulnerabilities, only one was found to be effective.

All Alerts

Before Analysis:
13 alerts with reported security vulnerabilities

After Analysis:
1 alert (out of 13) found to be Effective or Suspected Effective (8%)

Savings: 92%

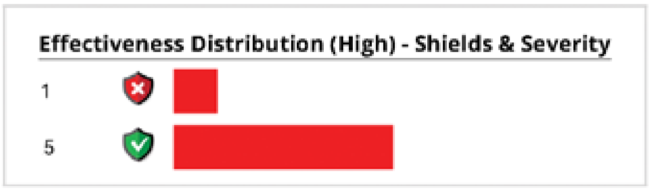


Alerts with high-severuty CVEs

Before Analysis:
13 alerts with reported security vulnerabilities

After Analysis:
1 alert (out of 13) found to be Effective or Suspected Effective (8%)

Savings: 83%



BENEFITS

1. Reduce security alerts by up to 85%, saving security and developers' time
2. Speed up the remediation process to reduce exposure to reported vulnerabilities
3. Gain visibility at a glance to focus remediation efforts
4. Streamlined collaboration between security and development

THINK WE'RE EXAGGERATING? **TRY US OUT!** Sign up for a free trial and be amazed by the ease and accuracy of the WhiteSource solution. www.whitesourcesoftware.com