For Security and IT Leaders
October 2020
(YSCA)

Securing your Enterprise Applications from Open Source Vulnerabilities with MergeBase

Why you need to take action today and how you can effectively protect yourself from adversaries

MERGE BASE

Overview

For Enterprises that have an SCA

Key Challenges

- Many Traditional SCA tools only focus on identifying security issues but lack actionable guidance to prioritize, triage, and resolve. As a result, a large percentage (57%) of enterprises reported that they do not know which vulnerabilities post the highest risk to the business.
- Without guidance, enterprises are overwhelmed with a growing backlog of unresolved security issues that has increased their exposure to data breaches. 42% of enterprises say a data breach occurred because a patch was available but was not applied.
- Traditional SCA tools also produce excessive levels of false positives. Most enterprises underestimate the true cost of false positives and which bears a significant burden. A 2000 employee enterprise, with 300 applications scanning 8 times a year with false positive rate at 20%, spent $1.35 million[1] annually triaging false positives.

Recommendations

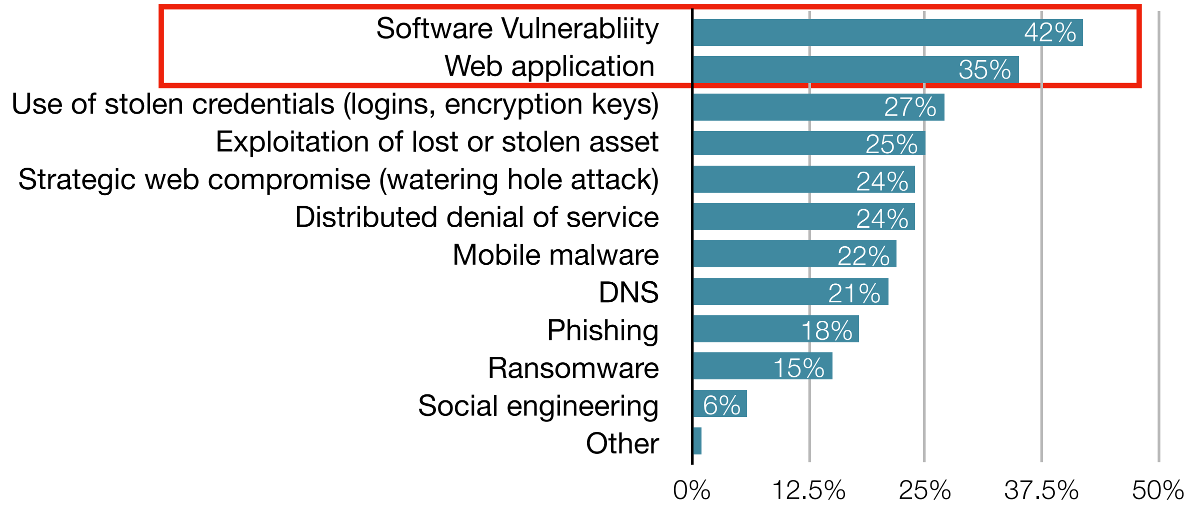
- Don’t accept the status quo. Consider the benefits of a 3rd-gen SCA solution to your cybersecurity defence that include guidance and prioritization so that you can target the vulnerabilities that post the highest risk. Mature security enterprises are accelerating their implementation of a diversified cybersecurity strategy because they realize it is the best defence to today’s adversaries.

- Do not underestimate the true cost of false positives. Take inventory of the burden false positives have had on your budget, productivity, and team morale. 3rd-gen SCA solutions minimize false positives so that it frees your resources to focus on delivering customer value.

[1] Fully loaded hourly wage = $110. Hours spent annually spent triaging false positives = 12,288 hours. Spending an average time of 18 mins to triage and manage each vulnerability result.

Open Source Vulnerabilities are still easy for Adversaries to Exploit. Approximately 30% of all vulnerabilities in open source also include publicly available exploits in the form of proof-of-concept code and Nessus/Metasploit rules. Adversaries no longer have to target the custom applications of a target. Instead, they can simply point the exploit at a range of IP addresses and escalate their actions to those that are exploitable. Adversaries recognize that many organizations track the open source they use poorly and therefore are not aware of the risk in their applications and systems. It's no surprise that application and software vulnerabilities are on the rise and remain the most common attack vector.

How Was this external attack carried out?



Source: Forrester: The State Of Application Security,

Most Traditional SCA solutions are not useful in Determining which Vulnerabilities pose the Greatest risk

Even with an SCA, most enterprises have limited visibility into the open source they use, the dependencies across their applications, and corresponding known vulnerabilities. A recent report by IBM stated that "57% say their enterprise do not know which vulnerabilities pose the highest risk to their business". They know they have hundreds or thousands of known vulnerabilities in their applications but don't know the real risk because a) they don't know if those vulnerabilities are reachable by an attacker and b) don't know which of the exploitable vulnerabilities pose the highest risk.

Traditional SCAs lack Guidance and Prioritization Features

Traditional SCA only focuses on identifying security issues but lack actionable guidance to prioritize, triage, and resolve. Under ideal circumstances, you would patch and resolve, except your business stakeholders don't have confidence that addressing vulnerabilities won't risk breaking the application. Without guidance, enterprises are overwhelmed with a growing backlog of unresolved security issues that have increased their exposure to data breaches. IBM reported that 42% of enterprises say a data breach occurred because a patch was available but was not applied. Managing known vulnerabilities is difficult, but if you settle with the status quo, you will be putting your enterprise at risk.

90%

OF ALL ENTERPRISES USE OPEN SOURCE

— GARTNER

57%

SAY THEIR ORGANIZATION DO NOT KNOW WHICH VULNERABILITIE S POSE THE HIGHEST RISK TO THEIR BUSINESS

— IBM

40%

INCREASE IN ENQUIRIES ON SCA FROM 2019 TO 2020

— GARTNER

42%

SAY A DATA BREACH OCCURRED BECAUSE A PATCH WAS AVAILABLE FOR A KNOWN VULNERABILITY BUT NOT APPLIED

— IBM

The Real Cost of False Positives

After committing your team to triage the results from your traditional SCA tools, they often find an overwhelming number of false positives. Your team is further fatigued from expending significant effort and producing little value to the business or the customer. At a crossroads, you now have a choice to continue this path of inefficiency or hire professional services from your vendor to help you manage false positives. Unfortunately, your resources are still utilized to support the professional services team. To illustrate, a 2000 employee enterprise, with 300 applications scanning 8 times a year with false positive rate at 20%, spent $1.35 million[1] annually triaging false positives. Do not underestimate the true cost of false positives. Take inventory of the burden false positives have had on your budget, productivity, and team's morale. 3rd-gen SCA solutions minimize false positives so that it frees your resources to focus on delivering customer value.

Consider a 3rd-gen SCA for your Enterprise Today

With adversaries working 24/7, open source vulnerabilities being easy to exploit, and your competitors accelerating their adoption of Software Composition Analysis (SCA); this will only encourage adversaries to concentrate on enterprises with weak defences. Don't accept the status quo and consider a 3rd-gen SCA solution to your portfolio of cybersecurity defences.

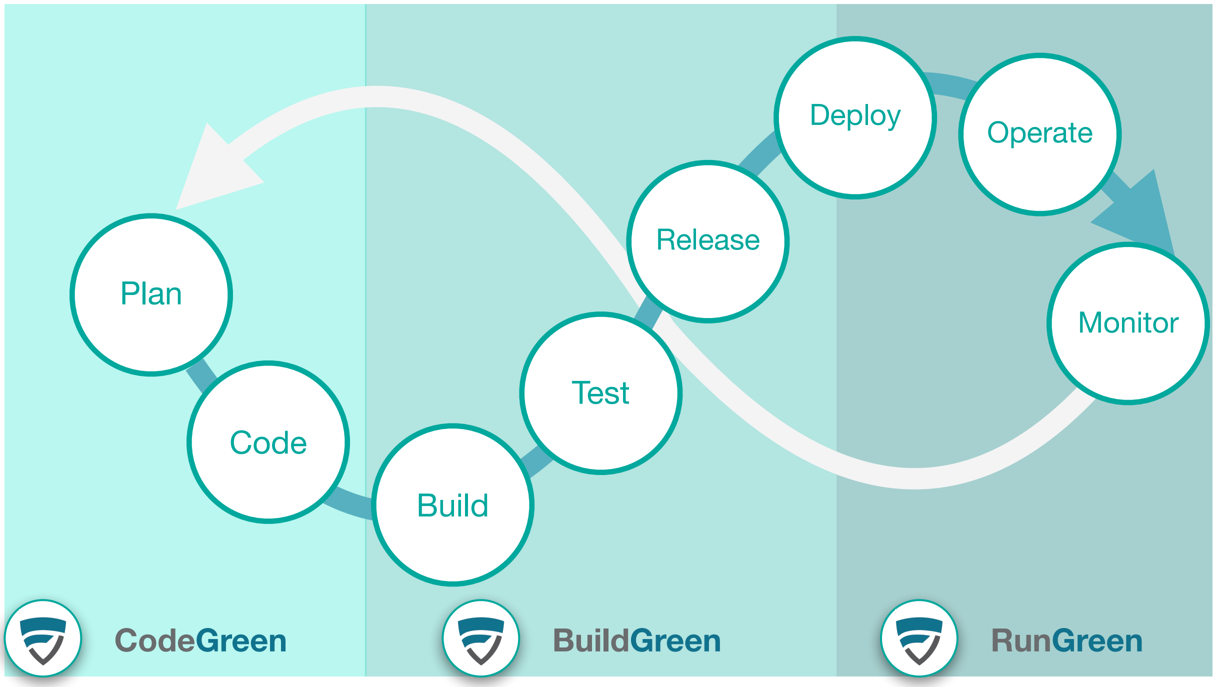
[1]. Fully loaded hourly wage = $110. Hours spent annually spent triaging false positives = 12,288 hours. Spending an average time of 18 mins to triage and manage each vulnerability result.

Introducing MergeBase

Trusted by security and development teams at top enterprises, MergeBase provides security and development teams with visibility to the real risk in their applications from vulnerable open source components at every stage of the software development lifecycle with **CodeGreen**, **BuildGreen**, and **RunGreen**.

MergeBase accelerates triage by minimizing false positives and deemphasizing vulnerabilities in unused code. It automates remediation during development and can block attacks on vulnerable components in production.

MergeBase throughout your SDLC



CodeGreen

Awareness: CodeGreen alerts Developers to known vulnerabilities early in the development process, enabling overall cost savings and quick resolution.

Enterprise Controls: CodeGreen prevents vulnerabilities from even entering the Enterprise code base.

BuildGreen

Identification: BuildGreen accurately identifies and report vulnerabilities during the build and deployment process, with very low false positive rates.

Control: BuildGreen can stop builds that contain vulnerabilities outside of enterprise policy levels.

RunGreen

Visibility: RunGreen tracks your applications instances to all data centre including cloud and gives you a complete and up to date overview of risk and actual usage.

Protection: RunGreen can instantly reduce risk in production for vulnerabilities that have not been remediated yet.

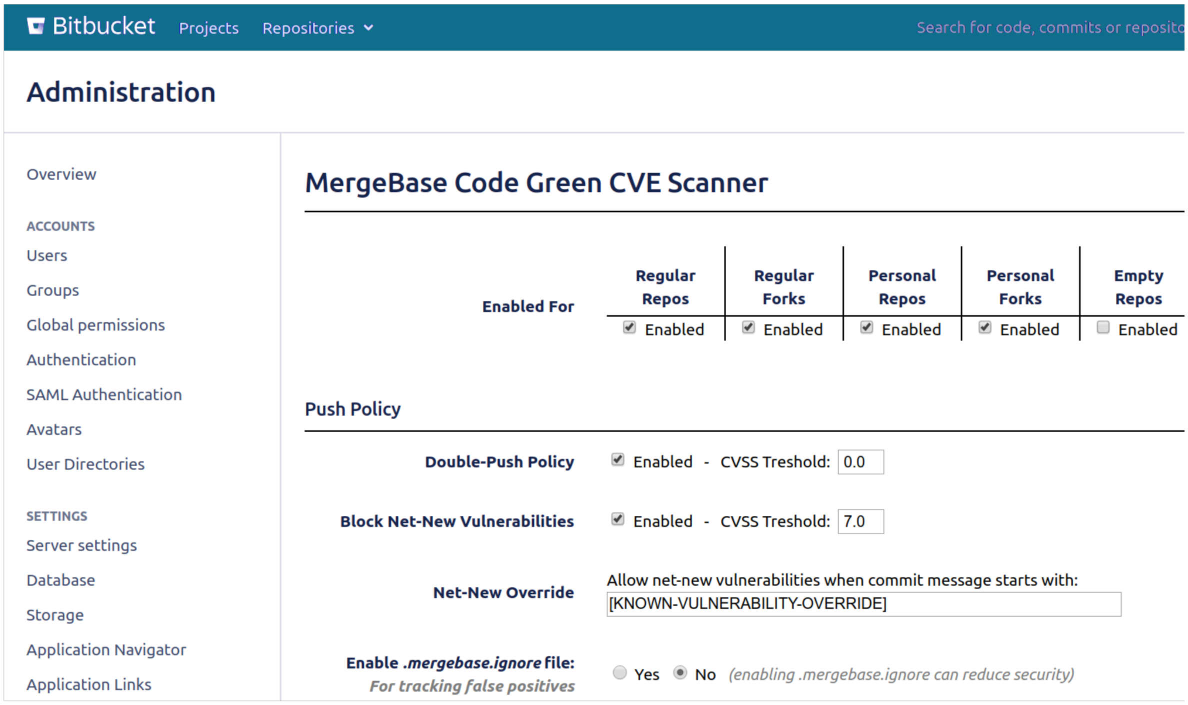
MergeBase empowers your security and development teams to effectively find and reduce the real risks in open source software more rapidly than ever before.

Unlike traditional tools, MergeBase goes above and beyond CVE's from the NVD because your enterprise needs every advantage against adversaries.

Shift Left with CodeGreen

Instead of resolving blockages at the end of a project when it is most expensive and impactful to the enterprise, with CodeGreen you can empower developers to resolve vulnerabilities as early as possible. CodeGreen is an early-warning defence for your in-house software development and integrates directly into code repositories such as Bitbucket. CodeGreen is **developer friendly** and adds three robust controls to your code repository to help you manage and reduce known vulnerabilities within internal software projects.

> “When it came to information security and compliance, we found that blockages at the end of the project were much more expensive than at the beginning and InfoSec blockages were among the worst"
>
> **Justin Arbuckle,** former chief architect at **GE**



Administration

Overview

ACCOUNTS
Users
Groups
Global permissions
Authentication
SAML Authentication
Avatars
User Directories

SETTINGS
Server settings
Database
Storage
Application Navigator
Application Links

MergeBase Code Green CVE Scanner

	Regular Repos	Regular Forks	Personal Repos	Personal Forks	Empty Repos
Enabled For	☑ Enabled	☑ Enabled	☑ Enabled	☑ Enabled	☐ Enabled

Push Policy

Double-Push Policy ☑ Enabled - CVSS Treshold: 0.0

Block Net-New Vulnerabilities ☑ Enabled - CVSS Treshold: 7.0

Net-New Override Allow net-new vulnerabilities when commit message starts with: [KNOWN-VULNERABILITY-OVERRIDE]

Enable *.mergebase.ignore* file: *For tracking false positives* ○ Yes ● No *(enabling .mergebase.ignore can reduce security)*

1. High Severity Vulnerability Control: Sign-off Policy

Since new vulnerabilities are discovered every day, development and security teams need a way to coordinate their response, especially when such vulnerabilities are severe. CodeGreen provides a "Sign-off Control" to leverage Github/Gitlab/Bitbucket style pull-request workflows for exactly this purpose (a "pull-request" is a lightweight review built on top of Git). Administrators can configure CodeGreen to require approval/sign-off when critical vulnerabilities are discovered in the project code. This mechanism has the following benefits:

- **Creates awareness**: Developers naturally become aware of new critical known vulnerabilities in their software system's libraries.
- **Lowers cost**: By shifting left, problems are addressed earlier in the development lifecycle, organizations will lower costs and impact. This approach encourages developers to address the vulnerability directly in the code and avoid requiring the sign-off. Problems are addressed before they even go to test or review.

- Promotes **risk-based decision-making**: If the developer cannot easily remove the vulnerability, the sign-off provides transparency and accountability to let vulnerable new code enter the main development branches.

CodeGreen controls generate awareness and collaboration between the development and security teams about the problem and provide strong incentives and alignment to see the vulnerability addressed directly in code or to move forward understanding the enterprise's risk implications.

2. Medium Severity Vulnerability Control: Double-Push

What happens when a software project contains known vulnerabilities above a severity threshold? CodeGreen will initially reject all new code submissions and provide clear information on the vulnerabilities and associated risk. The developer is now able to efficiently conduct a risk assessment on this vulnerability. If the developer concludes that the risk is acceptable, they can re-submit their code with known vulnerabilities. Developer actions are logged to establish clear accountabilities and traceability.

3. Low Severity Vulnerability Control: Block Net-New Vulnerabilities:

A developer will often discover a more suitable open source library to help them complete their work faster rather than implement the entire solution from scratch. Unfortunately, without the right SCA tools, there is a strong chance that by introducing a new open source library into the code base, the developer could be introducing "net-new" vulnerabilities (that is, vulnerabilities that are not already a part of the system). To address this scenario, CodeGreen can be configured to block libraries that bring in new vulnerabilities. The developer can seek an alternative library or a more recent version of the library that do not introduce new vulnerabilities.

Prioritization with CodeGreen

Many traditional SCA solutions, overwhelm the users with volumes of vulnerabilities with no means to prioritize what to work on first. With CodeGreen, security teams can initially set the "double-push" threshold to 9.5, disable the "sign-off" threshold, and encourage the engineering staff to clear out all vulnerabilities above that 9.5 threshold. Once vulnerabilities are cleared out the threshold can be gradually reduced, e.g., to 9.0, then 8.5, and so on, until your desired policy level is reached. A gradual reduction of the threshold avoids overwhelming security and development teams with too many vulnerabilities without a path to manage them.

CodeGreen White list for Low Risk Vulnerabilities to Reduce the Noise

CodeGreen provides a whitelist with vulnerabilities, in case you want to ignore low risk vulnerabilities (and are deemed low because of the way your software configures or uses the library). These are omitted from reports and bypass all of the CodeGreen controls.

Establishing the whitelist requires approval.

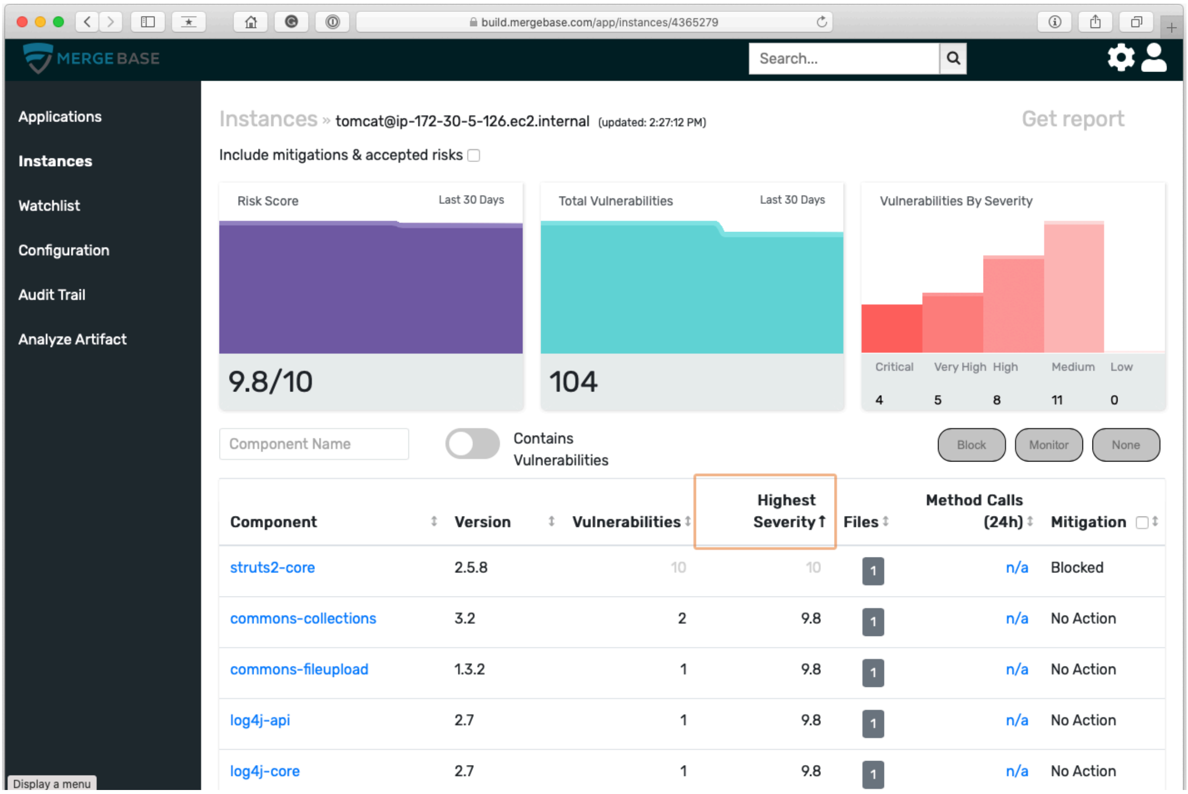
Product Spotlight: BuildGreen

BuildGreen detects vulnerabilities in Java, Python, Ruby, Scala, Python, Ruby, JavaScript, and .NET.

Build a Secure Future Today with BuildGreen

BuildGreen is a powerful solution for identifying the real risk of open source and commercial software libraries at build time or in existing applications. BuildGreen can even stop builds that contain severe vulnerabilities.

BuildGreen allows analysts and developers to use it directly, and it can be operated autonomously by bots, scripts, and continuous integration (CI) systems such as Jenkins and TeamCity.



Instances » tomcat@ip-172-30-5-126.ec2.internal (updated: 2:27:12 PM)

Include mitigations & accepted risks ☐

Risk Score — Last 30 Days: 9.8/10

Total Vulnerabilities — Last 30 Days: 104

Vulnerabilities By Severity

Critical	Very High	High	Medium	Low
4	5	8	11	0

Component Name — Contains Vulnerabilities

Component	Version	Vulnerabilities	Highest Severity	Files	Method Calls (24h)	Mitigation
struts2-core	2.5.8	10	10	1	n/a	Blocked
commons-collections	3.2	2	9.8	1	n/a	No Action
commons-fileupload	1.3.2	1	9.8	1	n/a	No Action
log4j-api	2.7	1	9.8	1	n/a	No Action
log4j-core	2.7	1	9.8	1	n/a	No Action

Visibility into Dependencies

Developers often work in environments where some dependencies might not be coming from the regular repository channels or artifact repositories (or local artifact caches) that can be held to acceptable level of trust.

BuildGreen extracts dependency data from Ruby, Python, Java, .NET and JavaScript build systems and understandS the dependency tree output from Maven and Grunt. It reconstructs precise version information (including copy/paste provenance) from built Java artifacts (e.g., *.jar, *.war, *.ear). BuildGreen can also reconstruct the "moment-of-copying" in copy/paste scenarios where portions of code were imported from open source libraries without proper attribution or record keeping.

BuildGreen's ability to analyze both build metadata and the raw built (Java) artifacts is an excellent way to verify that deployed builds truly match their specified dependency metadata. These special capabilities ensure that open source component supply chains (including local and internal caches) have not been compromised.

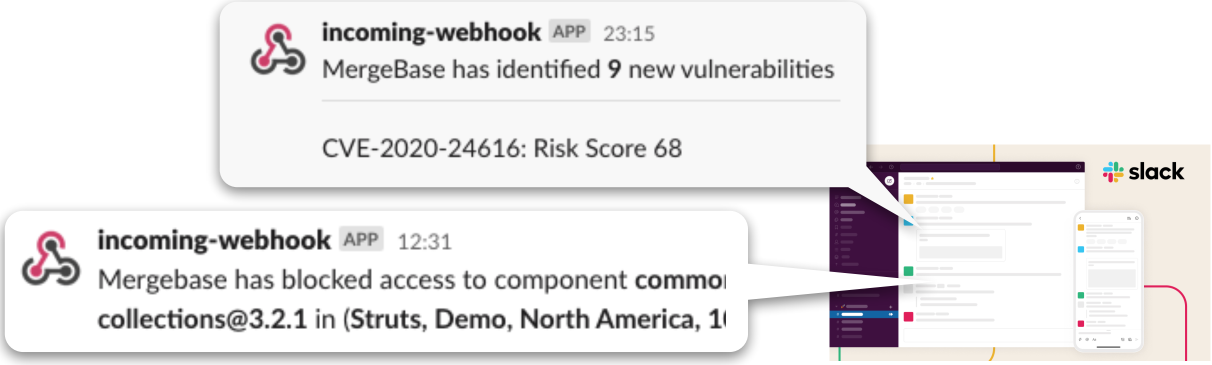
RunGreen for Runtime Protection

RunGreen detects and automatically defends against known-vulnerabilities at runtime. RunGreen empowers security analysts more effectively than traditional SCA tools because RunGreen does not require any expertise with software code or build systems.

Visibility and Monitoring of Real Risk

Whether your application is in a data centre or the cloud, you can set Rungreen to monitor a vulnerable library you specify. Once suspicious access of that library occurs, a notification is broadcast.

Notifications for detection, and blocking can be broadcast to all stakeholders



incoming-webhook APP 23:15
MergeBase has identified **9** new vulnerabilities

CVE-2020-24616: Risk Score 68

incoming-webhook APP 12:31
Mergebase has blocked access to component **commo**
collections@3.2.1 in (**Struts, Demo, North America, 1**

RunGreen provides security analysts with an instant component inventory and "live" vulnerability reports for a given application. This component inventory is far superior to the "bills of material" (BOM) produced by traditional SCA vendors because RunGreen references actual software running in production. Unlike vulnerability reports generated by traditional SCA's at earlier stages of the software development lifecycle, the live vulnerability report is based on automatic tracking of deployed libraries within every application instance in every data centre or cloud it is deployed to. RunGreen gives security analysts a complete picture of the system's overall risk profile that incorporates known vulnerabilities.

Prioritize with Data-Driven Guidance

RunGreen collects high-level usage data for all libraries in a system, including the last invocation time and library usage frequency over the previous 24 hours. Knowing usage frequency helps inform and prioritize triage and patching work for development and engineering teams. For example, developers might instruct build scripts to omit the library entirely if a library is never used. If a highly utilized library contains a severe vulnerability, security analysts can use this data to help encourage development groups to patch the library sooner rather than later.

Effective options for When you are Unable to Eliminate Known Vulnerabilities

Enterprises often face scenarios that prevent them from eliminating known vulnerabilities. Sometimes, resolving a vulnerability involves a large scale upgrade that is not feasible for your enterprise to implement. Rungreen offers effective options in these scenarios. You can disable the library or specifically the suspicious method in the library to prevent any execution or invocation of its functionality ("blocking") and be marked for closer monitoring. Once blocked, any attempt by an adversary to invoke the library a notification is broadcast to operations and security teams.

Security Self Check:
1. Are you able to track minute by minute what vulnerabilities all your applications have as deployed to data centres or the cloud?

2. What effective options does your enterprise have when you are not able to eliminate all known application vulnerabilities?

RunGreen tracks your applications instances to all data centre including cloud and gives you a complete and up to date overview of risk and actual usage.

RunGreen can instantly reduce risk in production for vulnerabilities that have not been remediated yet.

Engage with a **MergeBase** Security Expert

Over 90% of all enterprises have embraced open source as a means to accelerate development to deliver customer value. Unfortunately, open source introduces exploitable vulnerabilities that put your enterprise at risk. Learn more about how you can protect your enterprise from the #1 cause of breaches and accelerate development by working with a MergeBase Security Expert.

Security Expert Inquiry
To help understand how MergeBase can help protect your Enterprise form open source risk, connect with us for remote consultation so we can answer all your questions.

Security Expert Advisory
Work with a Security expert on a specific engagement in the form of a workshop with your leaders to help navigate open source risks with MergeBase

Learn more: www.mergebase.com
Contact us: info@mergebase.com

About MergeBase

MergeBase provides security and development teams with visibility to the real risk in their applications from vulnerable open source components at every stage of the development lifecycle.
MergeBase accelerates triage by minimizing false positives and deemphasizing vulnerabilities in unused code. It automates remediation during development and can block attacks on vulnerable components in production.

Protect your Enterprise's Future Today