



Francis Potter

Solution Architect

fpotter@gitlab.com

<https://about.gitlab.com>

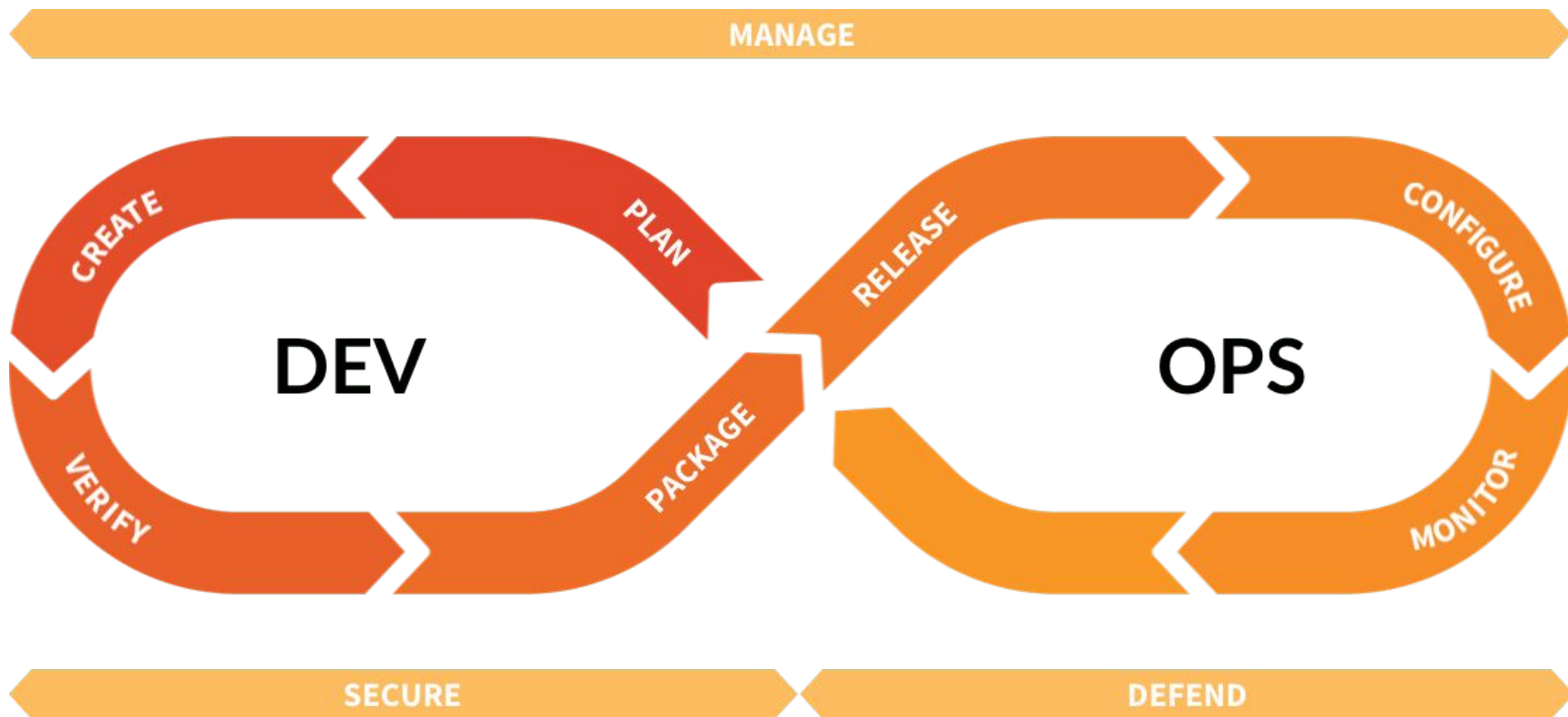
Application Security

at High Velocity

GOTO Conference
Apr 27-28, 2020









“We estimate that malicious cyber activity cost the U.S. economy between \$57 billion and \$109 billion in 2016.”

-- White House Council of Economic Advisors
Feb 2018

Vulnerabilities lead to Exploits



Adobe

153 mil user records
October 2013



eBay

145 mil users
May 2014



Equifax

148 mil consumers
July 2017



Marriott

500 mil customers
2014-2018



LinkedIn

165 mil user accounts
2012 and 2016



Zynga

218 mil user accounts
September 2019



**More than half of all small businesses
suffer a breach yearly but only 14% are
prepared to defend themselves**

-- CNBC
Oct 2019

Four tiers of threats



COMPUTE



STORAGE



DATABASE



NETWORK

Cloud Configuration

Responsibility:

Cloud Provider

Security Vendors

End-user

Four tiers of threats



NETWORKS

Network segmentation and Perimeter security

Network Threat Detection



FOUNDATION
SERVICES



COMPUTE



STORAGE



DATABASE



NETWORK

Cloud Configuration

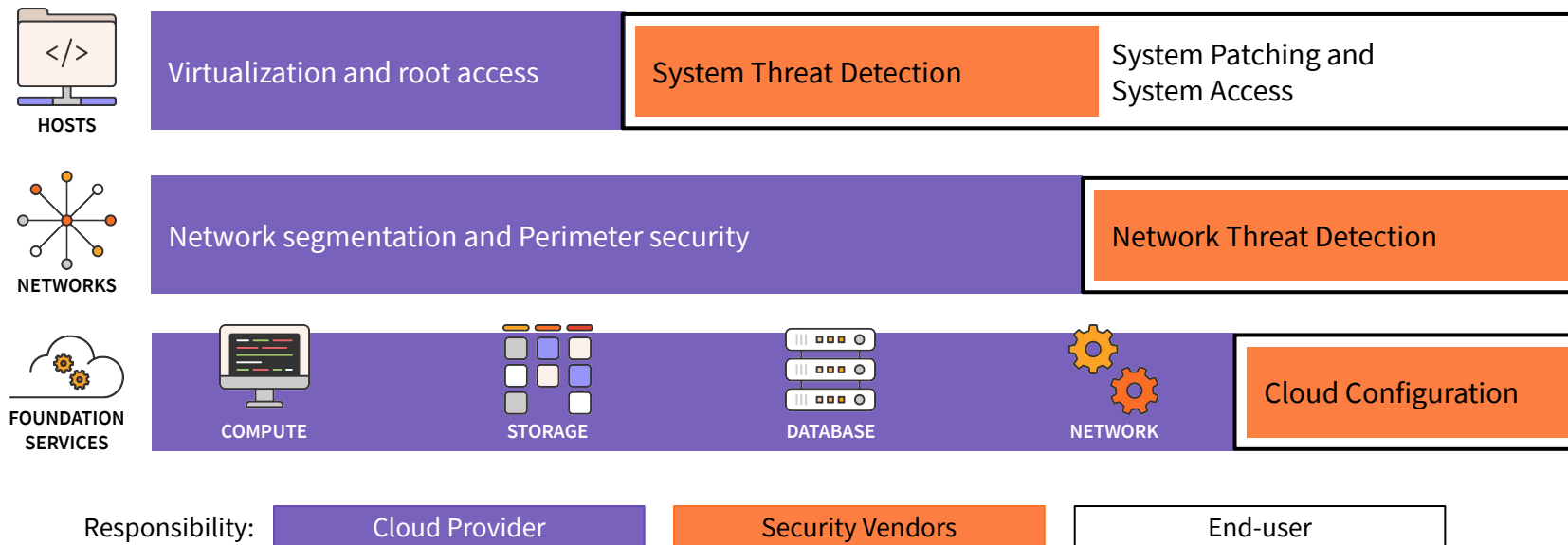
Responsibility:

Cloud Provider

Security Vendors

End-user

Four tiers of threats



Four tiers of threats



APPS

Application Threat Detection

Application Access

App Configuration and
App Patching



HOSTS

Virtualization and root access

System Threat Detection

System Patching and
System Access



NETWORKS

Network segmentation and Perimeter security

Network Threat Detection



FOUNDATION
SERVICES



COMPUTE



STORAGE



DATABASE



NETWORK

Cloud Configuration

Responsibility:

Cloud Provider

Security Vendors

End-user

Application Security vendors





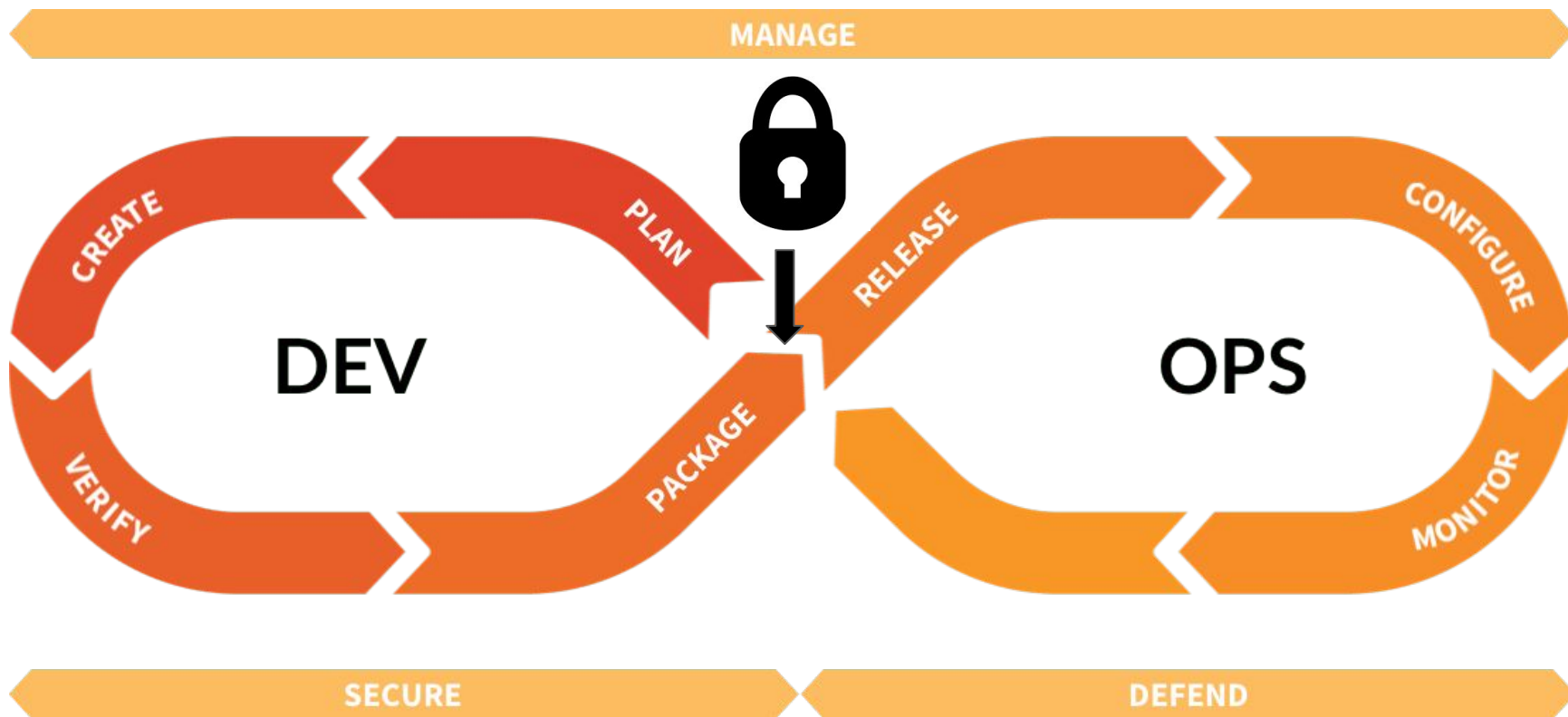
Is Security a Square Peg in a Round Hole of DevOps?

Established security tools were intended for a waterfall process at the end of SDLC and are incongruent with DevOps's

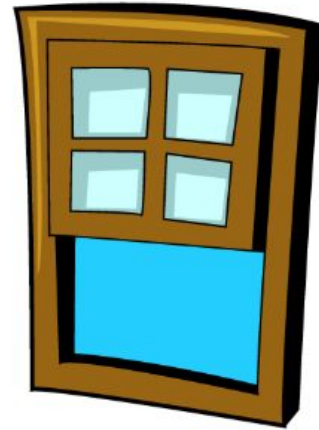
- **People**
- **Process**
- **Technology**



“Traditional” Application Security



Have to cover all your bases



Images from <http://clipart-library.com>



What if you could...

Scan all code, every time

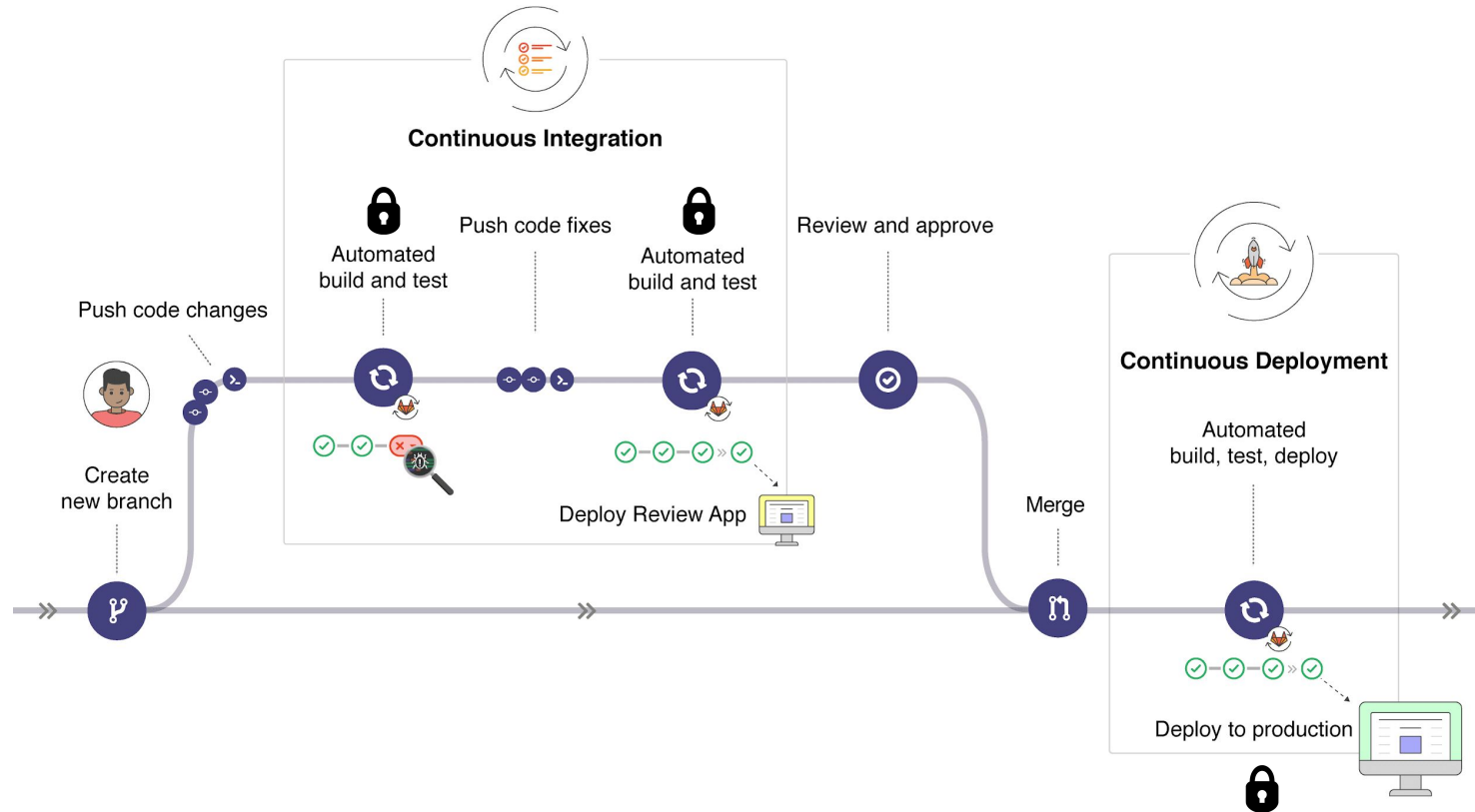
Seamlessly for dev

Using FEWER tools

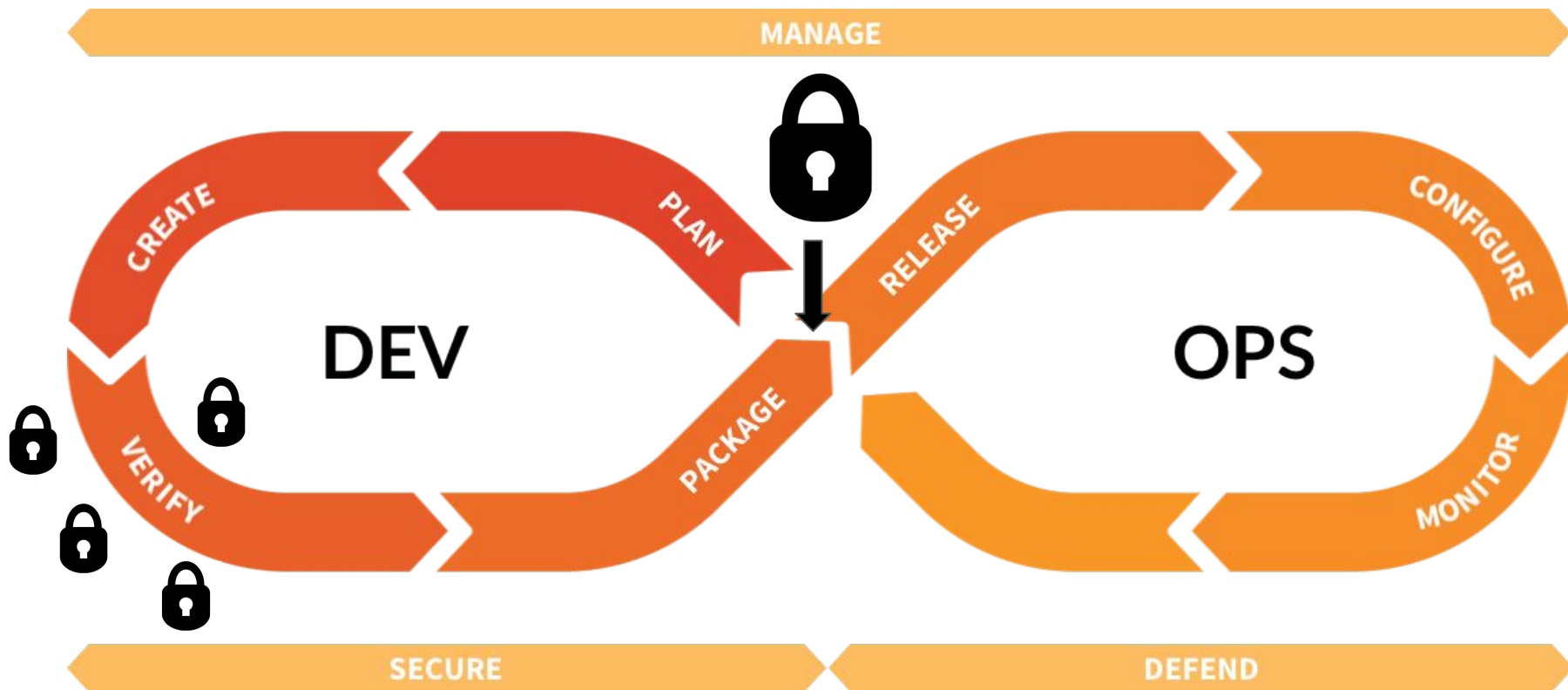
With Dev, Sec, and Ops on the same page

And happy compliance auditors

“Shifting Left” is Key



“Traditional” Application Security

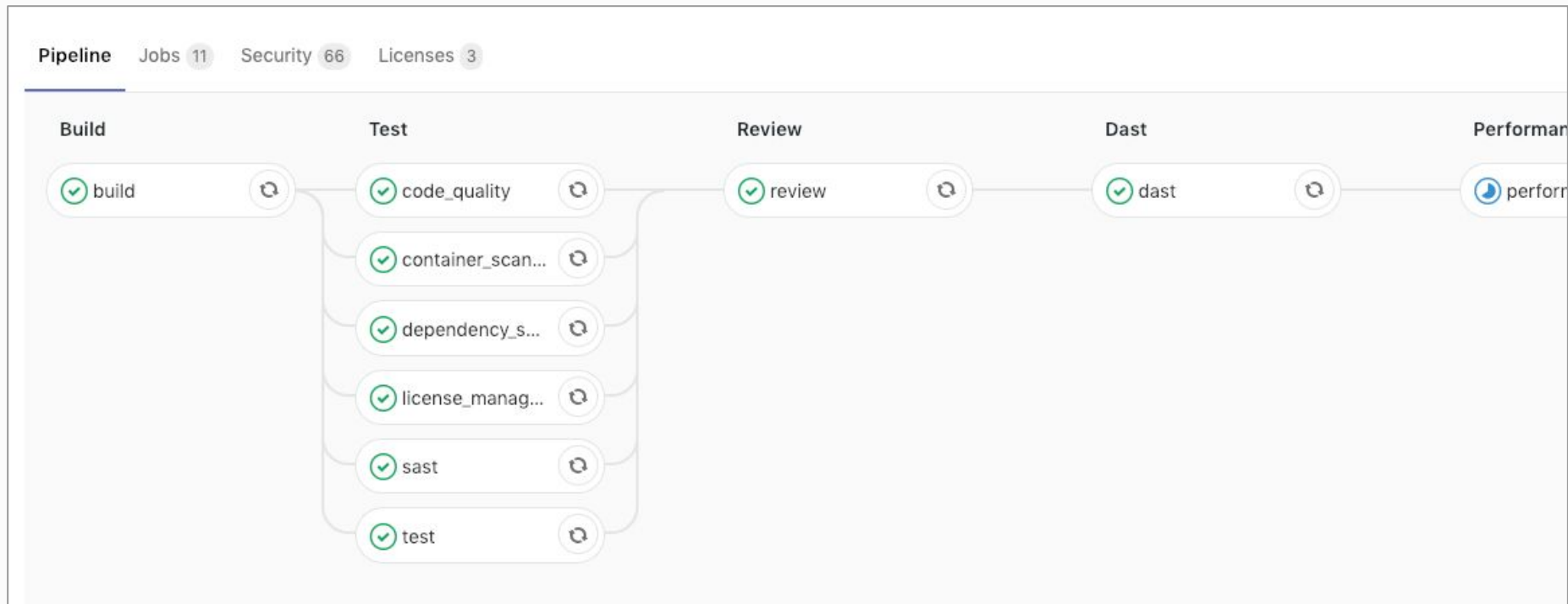


Types of Application Security



Type	Description
Static Application Security Testing (SAST)	Reads your code and finds vulnerabilities based on patterns
Dependency Scanning	Looks through your dependencies for those with known vulnerabilities
Container Scanning	Looks through your Docker (and other) containers for those with known vulnerabilities
Secret Detection	Finds passwords, API keys, and the like in your code
License Compliance	Assures that all dependency licenses comply with policy
Dynamic Application Security Testing	Scans a running application for potential attacks
Interactive Application Security Testing (IAST)	Sort of a combination of SAST and DAST
Fuzz Testing	Provides invalid, unexpected, or random data as inputs

Continuous Application Security = a United Workflow



Enabling Developer To Address Security Findings Quickly



WIP: Feature Branch (to demonstrate MR widgets)

Overview 1 Commits 5 Pipelines 16 Changes 6

0/1 thread resolved



Request to merge `feature-branch` into `master`

The source branch is 7 commits behind the target branch

Open in Web IDE

Check out branch



Pipeline #90341293 passed for d82a35f1 on `feature-branch`



Approve

Requires approval from Vulnerability-Check.



View eligible approvers



Security scanning detected 24 new, and 4 dismissed vulnerabilities for the source branch only

View full report

Expand



License Compliance detected 2 new licenses; approval required

Manage licenses

View full report

Expand

When vulnerabilities are present in an MR, you can easily **see** and **triage** them before the MR moves forward.

Vulnerability information at Developer's fingertips



! Security scanning detected 165 vulnerabilities for the source branch only

[View full report](#)

Collapse

! SAST detected 3 vulnerabilities for the source branch only ?

✗ Critical (Unknown): Password in URL in .autodevops.gitlab-ci.yml

✗ Medium (High): Spring CSRF unrestricted RequestMapping in src/main/java/hello/HelloController.java

✗ Low (Low): Found Spring endpoint in src/main/java/hello/HelloController.java

✓ Dependency scanning detected no vulnerabilities for the source branch only ?

! Container scanning detected 162 vulnerabilities for the source branch only ?

✗ Medium: CVE-2019-8321 in ruby2.5

✗ Medium: CVE-2018-18064 in cairo

✗ Medium: CVE-2017-9742 in binutils

✗ Medium: CVE-2017-9040 in binutils

✗ Medium: CVE-2019-7309 in glibc

! License management detected 9 licenses for the source branch only

[Manage licenses](#)

[View full report](#)

Collapse

● Apache 2.0 Used by accessors-smart, android-json, assertj-core, and 46 more

✓ BSD Used by asm

● CDDL + GPLv2 with classpath exception Used by javax.annotation-api

● Eclipse Public License - v 1.0, GNU Lesser General Public License Used by logback-classic, and logback-core

● Eclipse Public License 1.0 Used by junit

✓ MIT Used by jul-to-slf4j, mockito-core, and slf4j-api

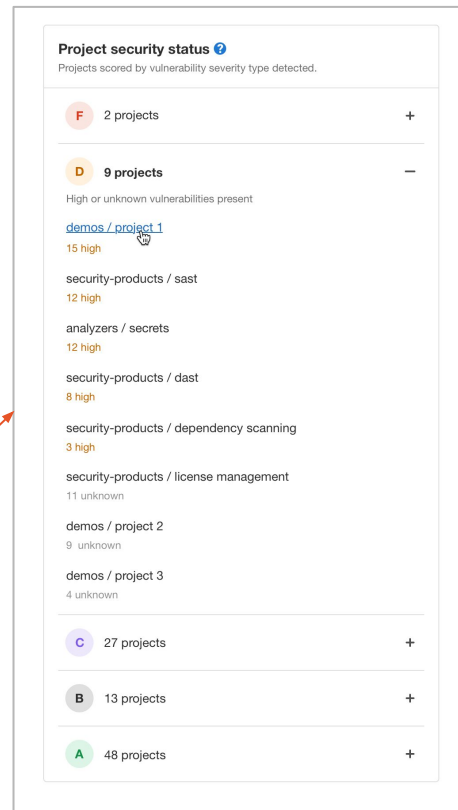
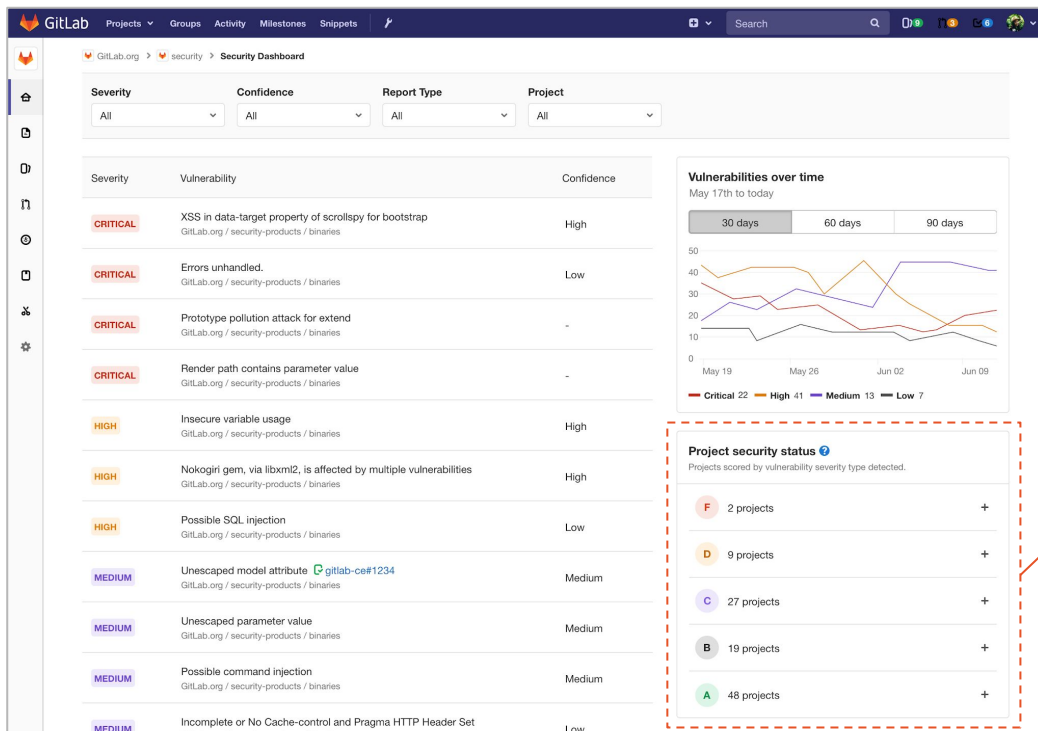
● New BSD License Used by hamcrest-core, and hamcrest-library

● Public Domain, per Creative Commons CC0 Used by LatencyUtils

● Public Domain, per Creative Commons CC0, Simplified BSD Used by HdrHistogram

! Merge This is a Work in Progress ? Resolve WIP status

Portfolio-level Executive Dashboards



And now we have...



DevSecOps

Contextual

Within existing
developer
workflows

Congruent

Supports rapid
iteration and
innovation

Integrated

File issues, auto
remediate, track
vulnerabilities

Efficient

Less tools and
context switching



If the DevOps practice is mature, teams are **3x** more likely to discover most security vulnerabilities before code is merged and in a test environment.

How do you automate application security testing within your software development pipeline?

34% Security testing results are included in the pipeline report used by developers

33% CI/CD automatically kicks off SAST scan

27% Developers use spell-check-like function for lite scan as they code

25% Don't know

20% CI/CD automatically kicks off DAST and/or IAST scan

Automation is critical to successful application security testing.



As part of their ramp up in GitLab, the dog-walking service recently furled automated security scanning and license management into their workflow, with Director of Engineering Dave Bullock noting how "great" it is to have those features baked into the pipeline so that immediate action can be taken when needed ... What previously took 40 minutes to an hour to accomplish, now takes just six minutes.



The security jobs in place are catching vulnerabilities from migrating to production through the product" explained Zaq Wiedmann, lead software engineer ... Wiedmann said the auditor also mentioned that Glympse had remediated security issues faster than any other company that he had worked with before in his 20-year career.



Francis Potter

Solution Architect

fpotter@gitlab.com

<https://about.gitlab.com>

Application Security

at High Velocity

GOTO Conference
Apr 27-28, 2020