GitLab

Integrating security into DevOps
Aka DevSecOps

Cindy Blake, CISSP

Product Marketing,
Secure & Defend, GitLab

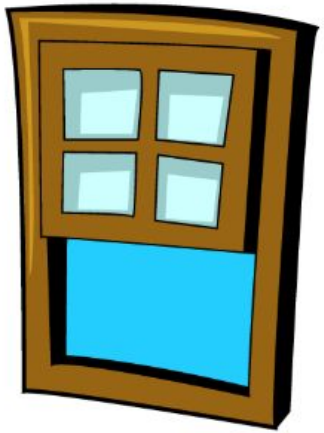
linkedin.com/in/cblake2000

@cblake2000

gitlab.com/cblake

- IT Director, Developer, Strategic Marketing, Product Marketing
- Security evangelist
- Based in Bryan, Texas

Do you have traditional application security?



Images from http://clipart-library.com

Are you struggling to fit Sec into DevSecOps?

Key trends that affect security programs:

- “Shift Left” to find/fix vulns earlier
- The need to scale the security program
- DevOps/Agile collaboration
- MVC and Iterative deployments
- Policy-driven automation
- Serverless, cloud-native

Challenges traditional app sec:

- Security teams have separate tools/processes and lack SDLC visibility
- Tests occur after individual code changes are merged with others’
- Scans test the entire application or the repo (out of context)
- Focus is on the app, what about IaC?

Executive Order on Improving the Nation's Cybersecurity

Biden Order To Require New Cybersecurity Standards In Response To SolarWinds Attack

- Notification required
- Threat intel sharing
- Software Bill of Materials
- Modernization
  - Zero Trust architectures
  - Cloud services
  - Data analytics

GitLab

Desired outcomes

- Greater efficiencies for both security and dev via
  - Collaboration
  - streamlined processes for both
  - Single source of truth
- Consistent compliance to policy
  - Cleaner and easier audits
  - Automated and consist use
  - Able to assess drift
- Reduced security exposure
  - Greater visibility to application risks, earlier
- Predictable costs
  - Security scans
  - Project time/budgets

Rethink Security as an Outcome, not a Department

“Your most important security product won’t be a security product .”

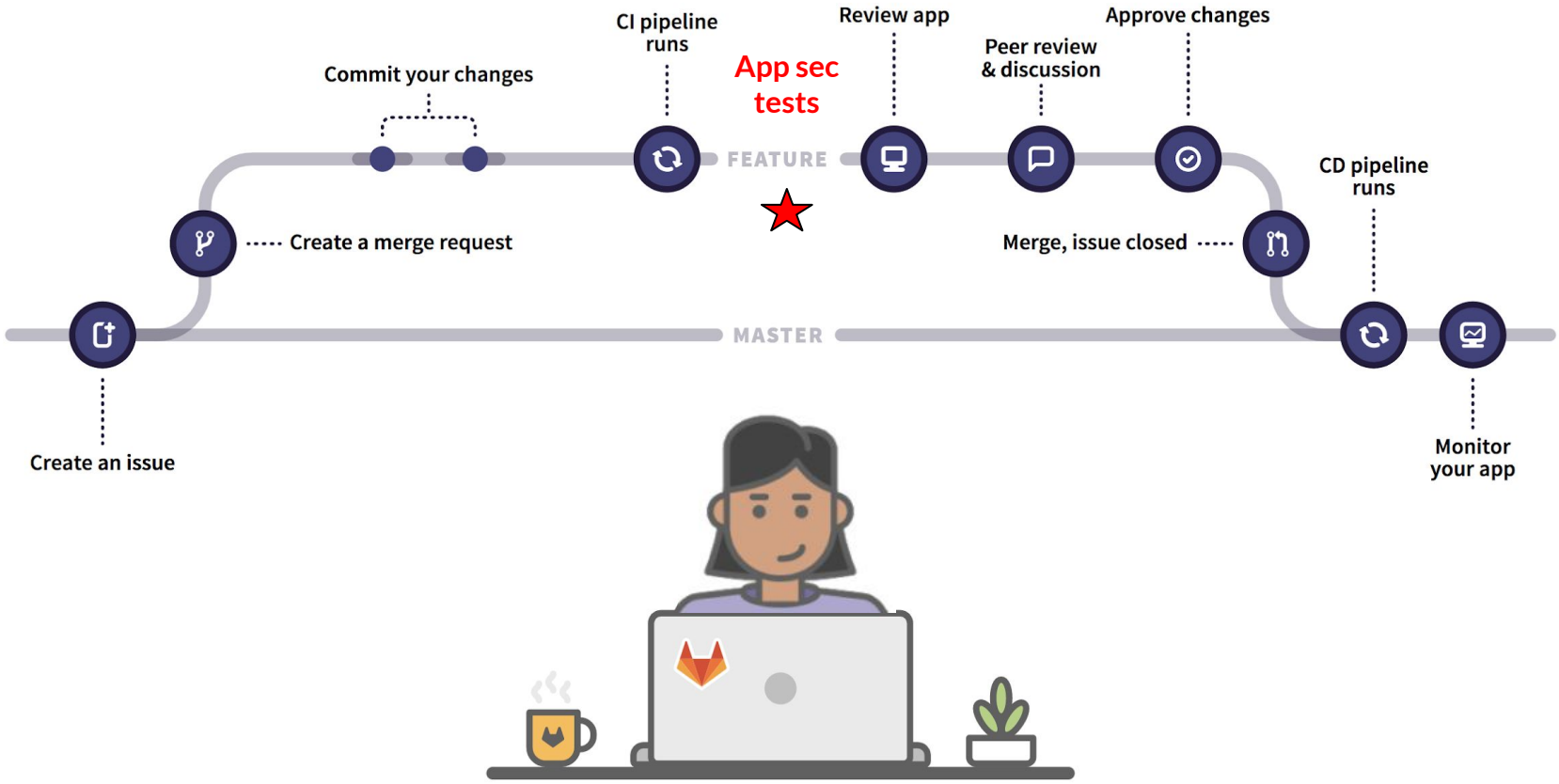
CISO of VMWare

New problems need new solutions

The CI pipeline is your software assembly line. It must be:

- Standardized
- Protected
- Measured
- Inspected

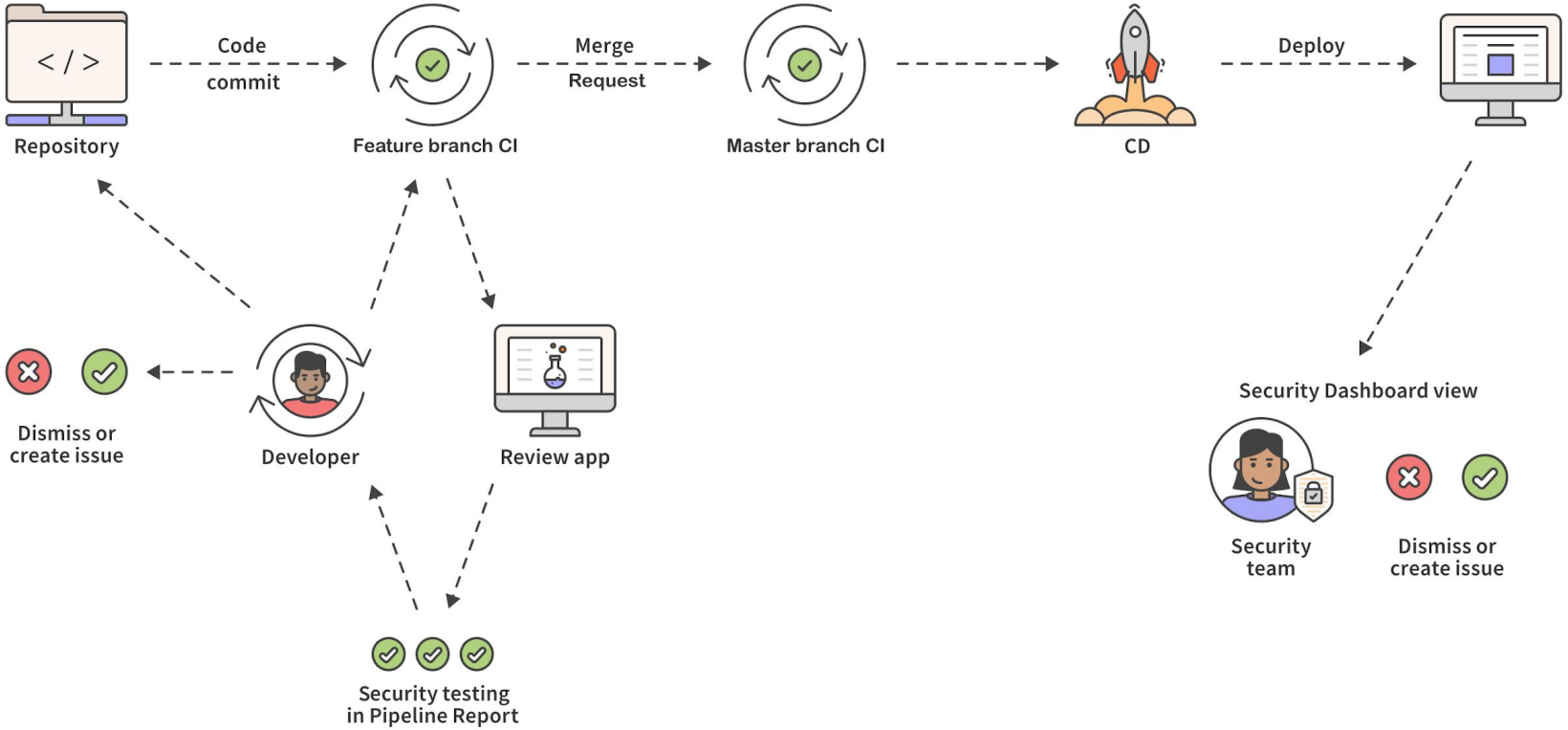
GitLab seamlessly tests for vulnerabilities within the developer workflow



Secure scanners today

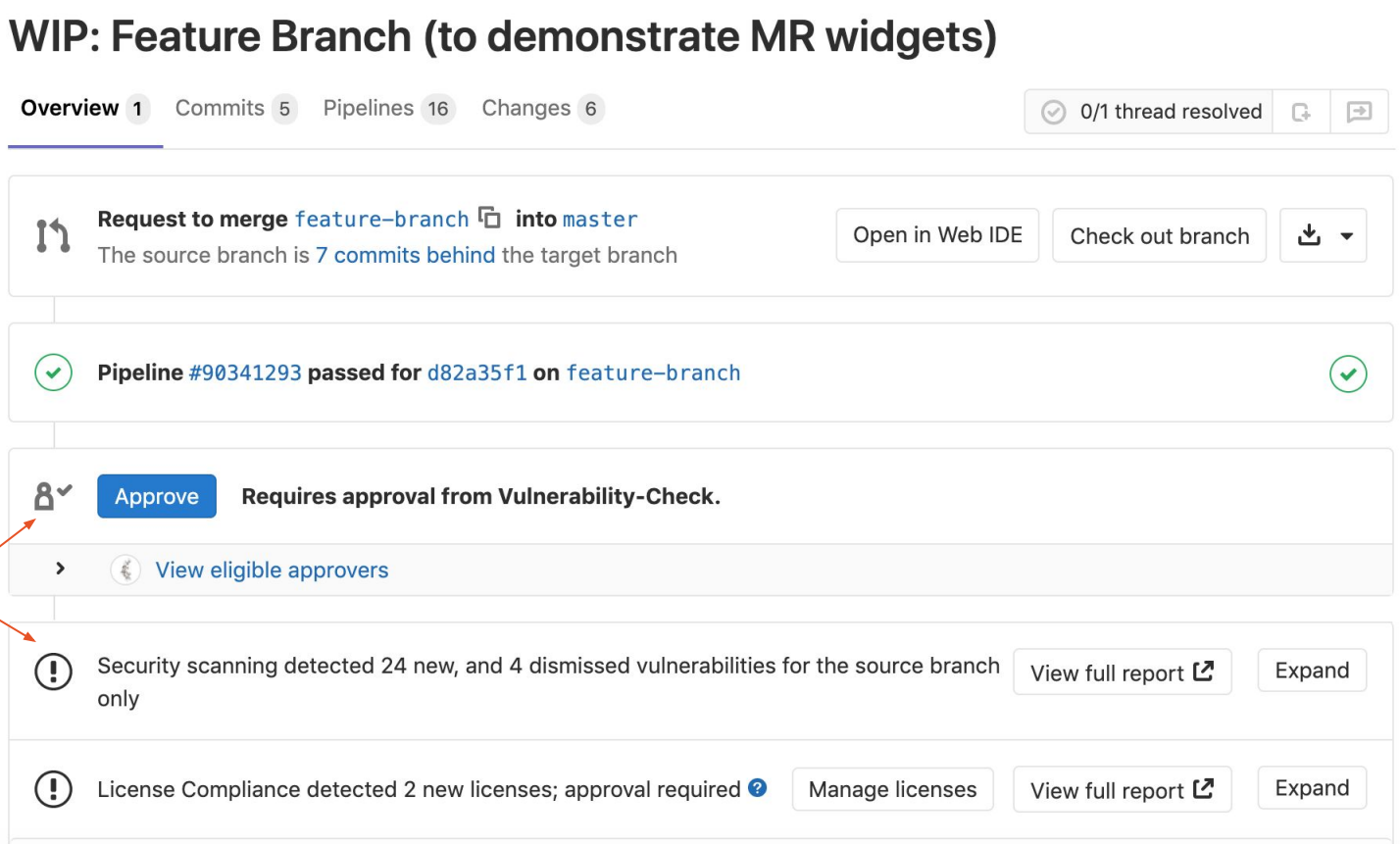
SAST	Static Application Security Testing scans the application source code and binaries to check for weaknesses and vulnerabilities.
DAST	Dynamic Application Security Testing analyzes your running web application for known runtime vulnerabilities leveraging the Review App.
Dependency Scanning	Analyze external dependencies (e.g. libraries like Ruby gems) for known vulnerabilities on each code commit with GitLab CI/CD.
Vulnerability Management	View, triage, trend, track, and resolve vulnerabilities detected in applications giving you full visibility to your organization's risk.
Secret Detection	Check for credentials and secrets in code commits and project history for allowing you to proactively resolve prior improper disclosure.
Container Scanning	Analyze your containers for known security vulnerabilities in the application environment leveraging public vulnerability databases.
License Compliance	Upon commit, project dependencies are searched for approved and blacklisted licenses defined by custom policies per project.
Fuzz Testing	Input unexpected and malformed data into your application to measure response and stability with the goal of finding unknown vulnerabilities.

Automate, allowing security to focus on exceptions

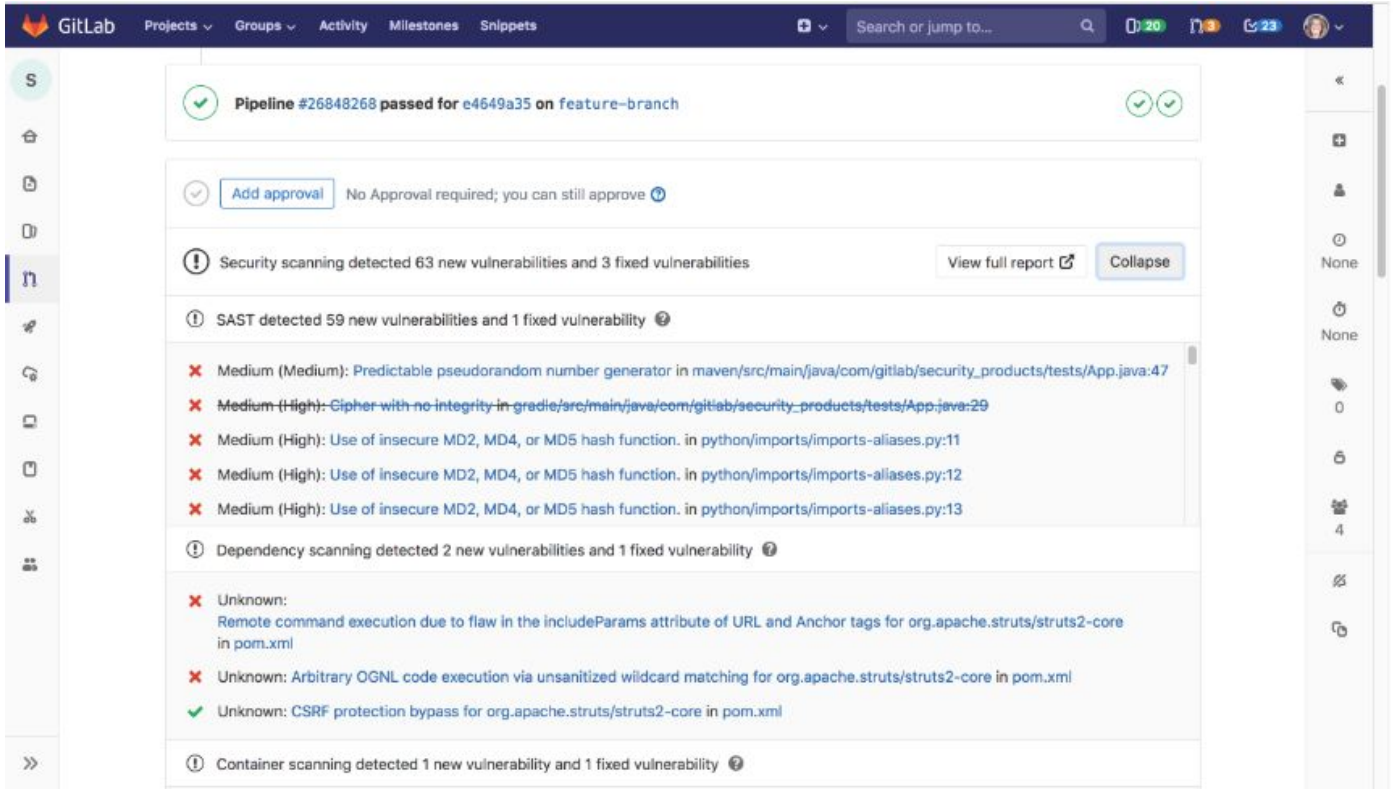


Enabling Developer To Address Security Findings

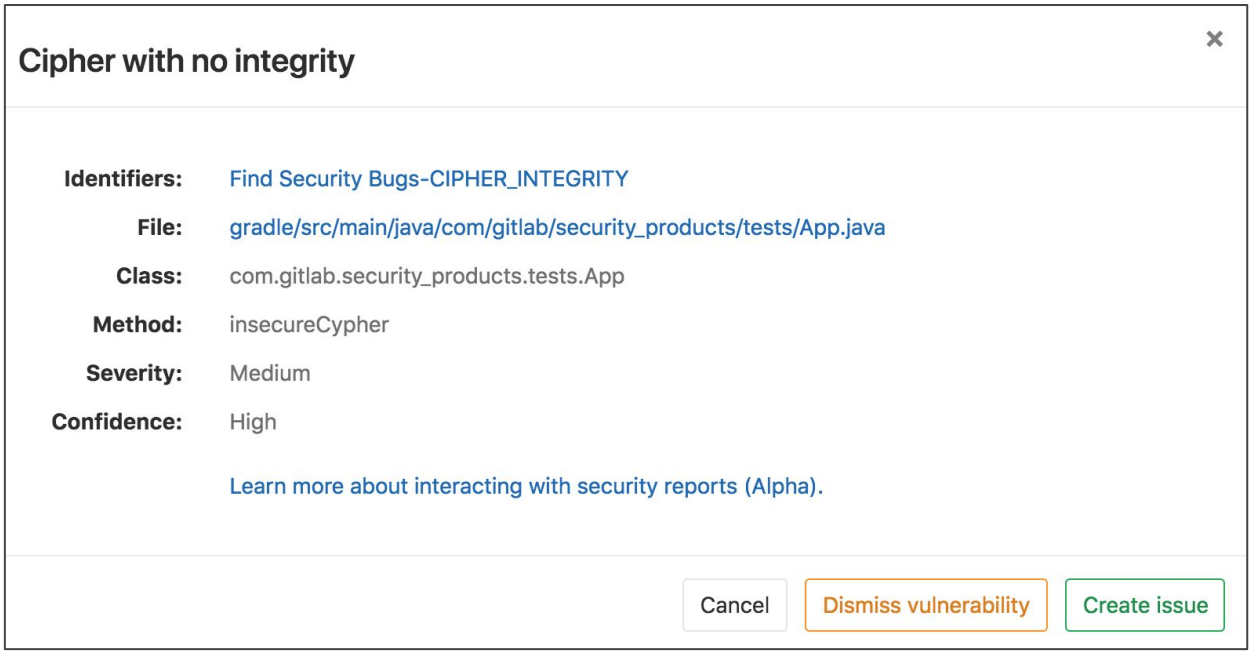
When vulnerabilities are present in an MR, you can easily **see** and **triage** them before the MR moves forward.



..in a developer-friendly way...with drill-down capabilities



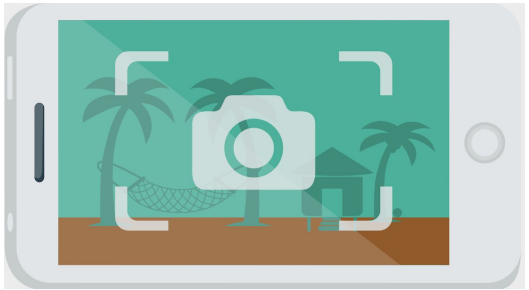
Developers can take action



Advantage of this approach

- Contextual
  - Within CI/CD dev workflow - accountable person
  - MR pipeline for dev
  - Security dashboard for Security
- Congruent with DevOps processes
  - Iterative within dev, tests every code change
  - Immediate cause/effect of code changes
- Integrated with DevOps tools
  - Create issues
  - Auto remediation
  - Production feedback
- Efficient and automated
  - Eliminate work wherever possible
  - No context-switching
  - Less tracking/triaging and more value-added security

Simplicity and Integration Wins!



VS

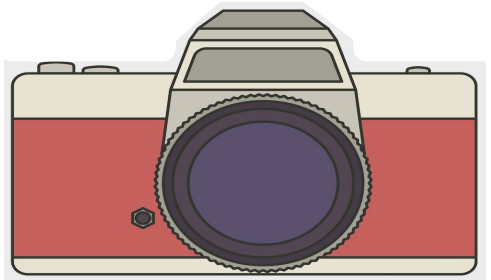
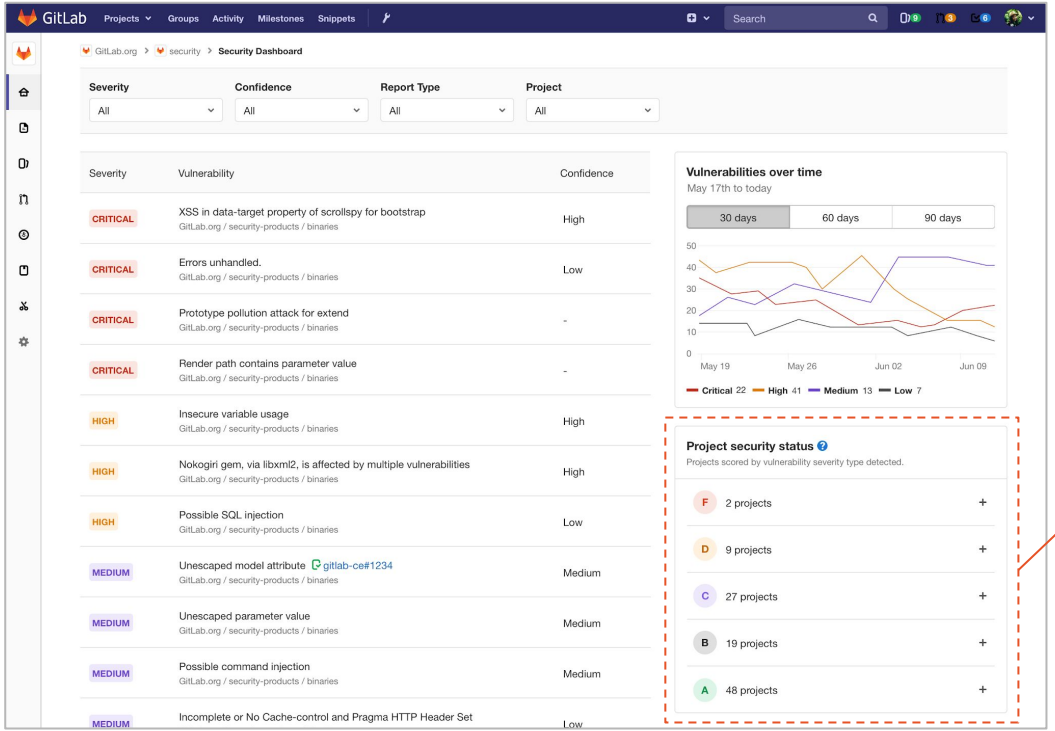
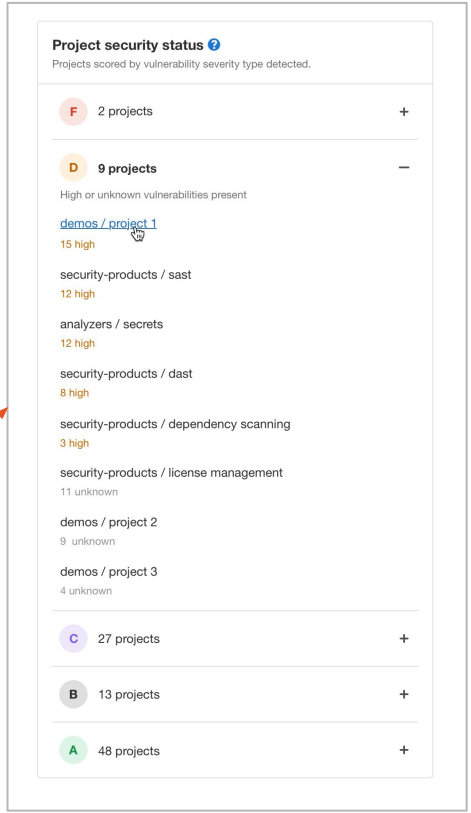


Image source: www.pngtree.com

Providing Visibility Into Security Risk

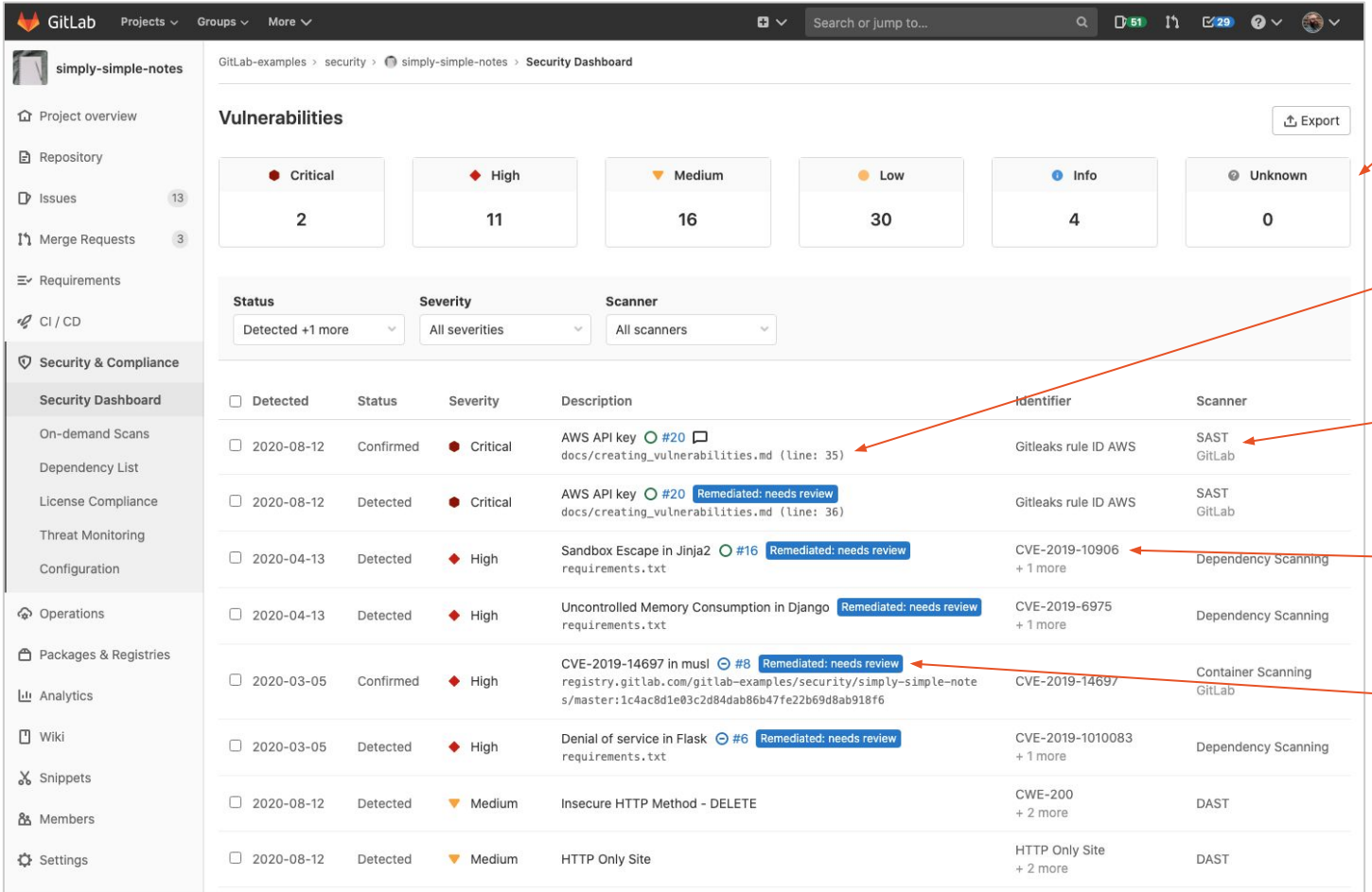
Quickly understand your at risk projects with Project Security Grades





Manage Security Risk Globally

Providing visibility into security risk at the Project level



Project's current vulnerability state

Line of code where vulnerability resides

Scanner that identified the vulnerability

Industry or public reference identifier

Remediated vulnerability awaiting review

Advantages for the security pro

- Automated, consistent CI pipelines
- Earlier visibility into risks and their remediation
- Predictable costs, even when testing more code more ways
- Reduced friction with dev
- Comprehensive feedback to dev

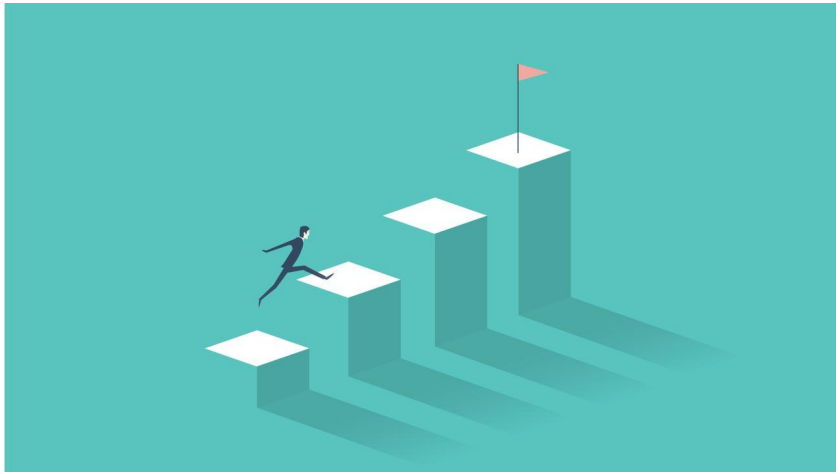


Image source: www.pngtree.com

We can work with incumbents, or replace them

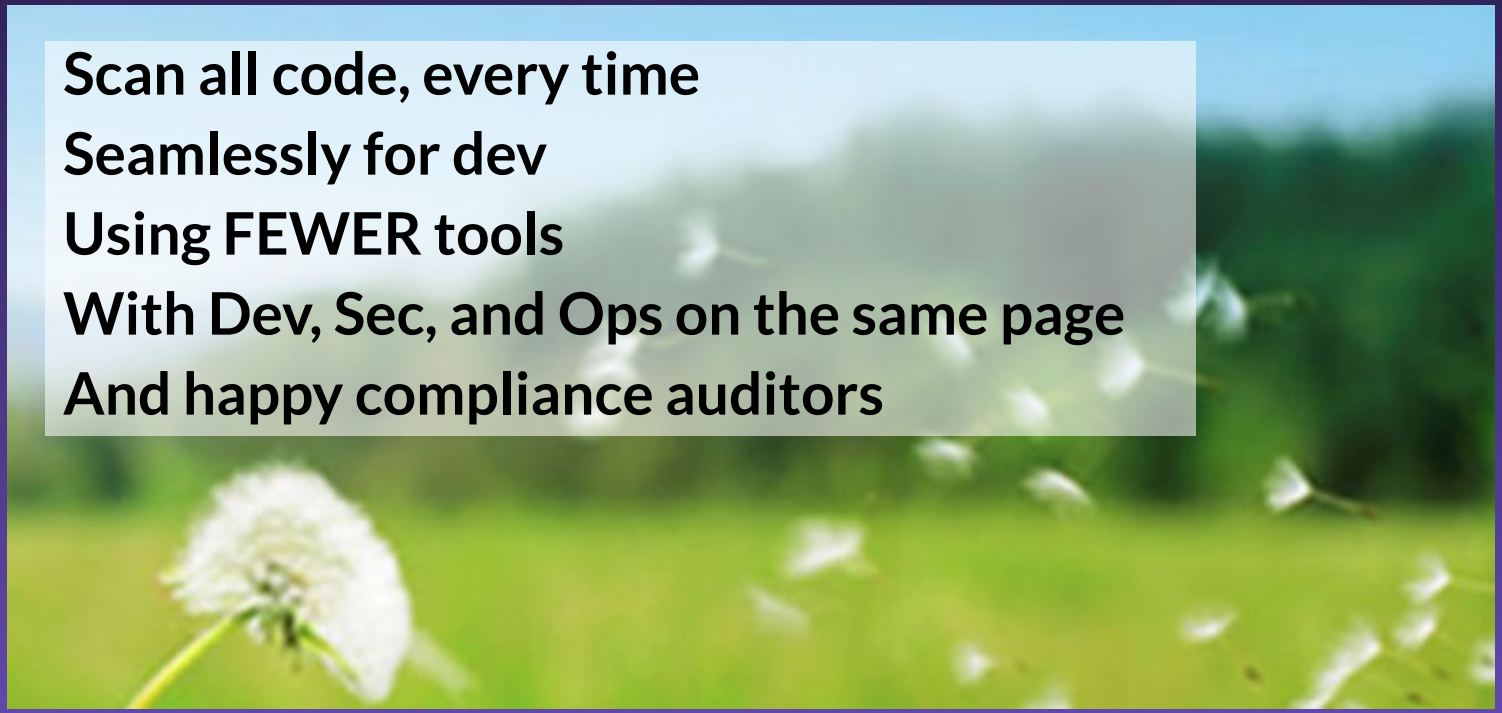
"GitLab Secure replaced Veracode, Checkmarx, and Fortify in my DevOps toolchain. GitLab scans faster, is more accurate, and doesn't require my developers to learn new tools" *- Financial services organization*

"GitLab Secure enables us to ship faster. Our other scanner tools could take up to a day to finish scanning whereas GitLab scans finish as little a few minutes"
- Healthcare services organization

"GitLab Secure gives us unlimited scanning capability and across our entire GitLab repo. This is obviously a very "shift-left" move as issues will be identified directly in the repo for review and triage. We will be able to get the most coverage this way and it limits the onboarding." *- Grocery retailer*

Customers are actively migrating to GitLab

The secret to successfully integrating Security into DevOps:



GitLab

Everyone Can Contribute!