PRE DEVOPS
Code is being pushed manually

3.500.000 Developers Globally

2-4 Releases/year
THE AGE of DEVOPS
Code pushed automatically

30.000.000
Developers globally

4.000
Releases / year

70%
Of applications are vulnerable

6-9 hours
Time it takes to fix a vulnerability
THE LAST BOTTLENECK
SECURITY OPERATIONS
EVERY VULNERABILITY IS A PROCESS

Multiple Scanners X Multiple Layers X Multiple Pipelines
EVERY VULNERABILITY IS TIME

Slow, Manual & Un-Scalable Workflow that’s Extremely Hard to Manage at Scale

- Developers’ Time

- Time spent on validating and implementing fix
- Time spent on configuring compensating controls to issues that can’t be fixed
- Time spent on validating actual risk

IDENTIFY >>> REMEDIATE >>> MITIGATE >>> ACCEPT RISK
EVERY VULNERABILITY IS RISK

Slow, Manual & Un-Scalable Workflow that’s Extremely Hard to Manage at Scale

Unmitigated Risk

Risk from vulnerabilities unpatched over long time-windows

Risk from unmitigated vulnerabilities in production

Risk from uncontrolled vulnerable code

IDENTIFY

REMEDIATE

MITIGATE

ACCEPT RISK
STAY SAFE
Move slow

MOVE FAST
Be exposed
STAY SAFE
MOVE FAST
STAY SAFE
REZILION PRODUCTS

**prioritize**
Validate Actual Attack Surface,
Patch 70% less

- 15 min to deploy
- 60 min to value

**mitigate**
SAFELY release code with known vulnerabilities

- Deterministic
- Automated

70%  

85%
REZILION
PRIORITIZE

Validate Actual Attack Surface,
Patch 70% less

70%
Missed release deadlines

Tensions between Security and AppDev

Failed audits - Years long Patching backlog

Release Faster

Eliminate Wasted Time

Erase Tech Debt

prioritize CI

prioritize PROD
REZILION IN CI

1. CI Plugin - add Prioritize plug-in to CI build test
2. Security test - Run Prioritize security test with SCA scan
3. Validation - augment scan feed to validate vulnerabilities
4. Display - show results in CI or managed dashboard

- Single tool with coverage across CI/CD
- Validates vulnerabilities risk in open source and custom code
- ~70% reduction in manual effort fixing unexploitable issues

Vulnerabilities
- Dismiss
- Create issue
New CISO – Deliver Value in the First 90 Days …
Board & Execs asking how Secure you are?
What Progress did we make?
What BU is lagging?

Understand true Attack Surface

Too much time spent on Unplanned Work in Production

Reduce patching by 70%
1. **Instrument** – run Prioritize script or Container on Stage hosts/images

2. **Security feed** – Prioritize consumes SCA/vulnerability scanner feed

3. **Validation** – augment scan feed to validate vulnerabilities

4. **Display** – show results in scanner or managed Dashboard

---

**Single tool with coverage across CI/CD**

**Validates vulnerabilities risk in open source, compiled apps and OS**

**~70% reduction in manual effort fixing unexploitable issues**
REZILION
MITIGATE

85%

mitigate
SAFELY release code with
known vulnerabilities
Pressure from Business to Accept Risk to allow releases and avoid rollbacks

Mitigate Unpatched Vulnerabilities
REZILION MITIGATE

1. **Instrument** - run Mitigate as a script or Container on prod hosts/containers
2. **Security feed** consumes and validates vulnerability scanner feed
3. **Mitigation** – a compensating control for unpatched vulnerabilities

Uses Prioritize for validation
Maps vulnerabilities against Rezilion Mitigation identifiers
~85% reduction in manual effort fixing unexploitable issues

---

**Vulnerabilities**
- Dismiss
- Create issue

---

**Repositories**
- External
- Internal

**Code Commit** → **Build/Test** → **Merge request** → **Review/Stage** → **Deploy** → **Host/App** → **Scan**
REVERSE ENGINEERING
Across multiple platforms, workload types and application runtimes
AUTOMATED PRODUCT SECURITY

Reduces and Mitigates Attack Surface from Dev to Prod & from OS to App

Application Layer
Container Layer
Os Layer

Full Stack

Comprehensive
Dev-to-Prod

Scan
Validate
Mitigate
Remediate
Enforce
Automate Manual Product-Security Work
Release Faster and Improve Security Posture
Make Agile Product Development Possible