



1 Hello,  
2 I'm Roeland from Aikido,  
3 talking about



4  
5 SecOps, AppSec, Software Security, ...  
6 Shift Left & OWASP top 10  
7  
8 ↳ aikido.dev



**Cybersecurity**  
strategisch onderzoek en industrie-impact

# Hacking is a growing problem

**64%**

**Of companies have experienced  
a cyberattack in 2022**

**\$4.350.000**

**Average cost of a data breach in 2022**

\* Accenture Cybersecurity Index <https://www.accenture.com/us-en/insights/cyber-security-index>

\* IBM Cost of a Data Breach Report <https://www.ibm.com/reports/data-breach>

# Pushing software security to go mainstream

## The Old World

Enterprise only



We'll take care about AppSec when we're bigger and have time.

## The New World

Tech companies

SW agencies

Enterprise



We need to care about AppSec from day 1.

Shift-left

Frameworks

Cloud

Incidents

Open-source

Dev Tooling

SOC2 & ISO27001

 As a ... CTO, CISO, Security Engineer, ...

**ISO 27001:2022**

**SOC 2 Type 2**

**GRC systems (Vanta, Drata, ...)**

**Pentest**

...

**OWASP top 10**

**Security Questionnaires**

**Bug Bounty Program**

**EU CRA - NIS2**

🔖 As a ... CTO, VP Engineering, DevOps Engineer, ...

**Leaked Secrets**

**Cloud Misconfigurations**

**Open-Source Dependencies**

**Malware**

...

**License Risks**

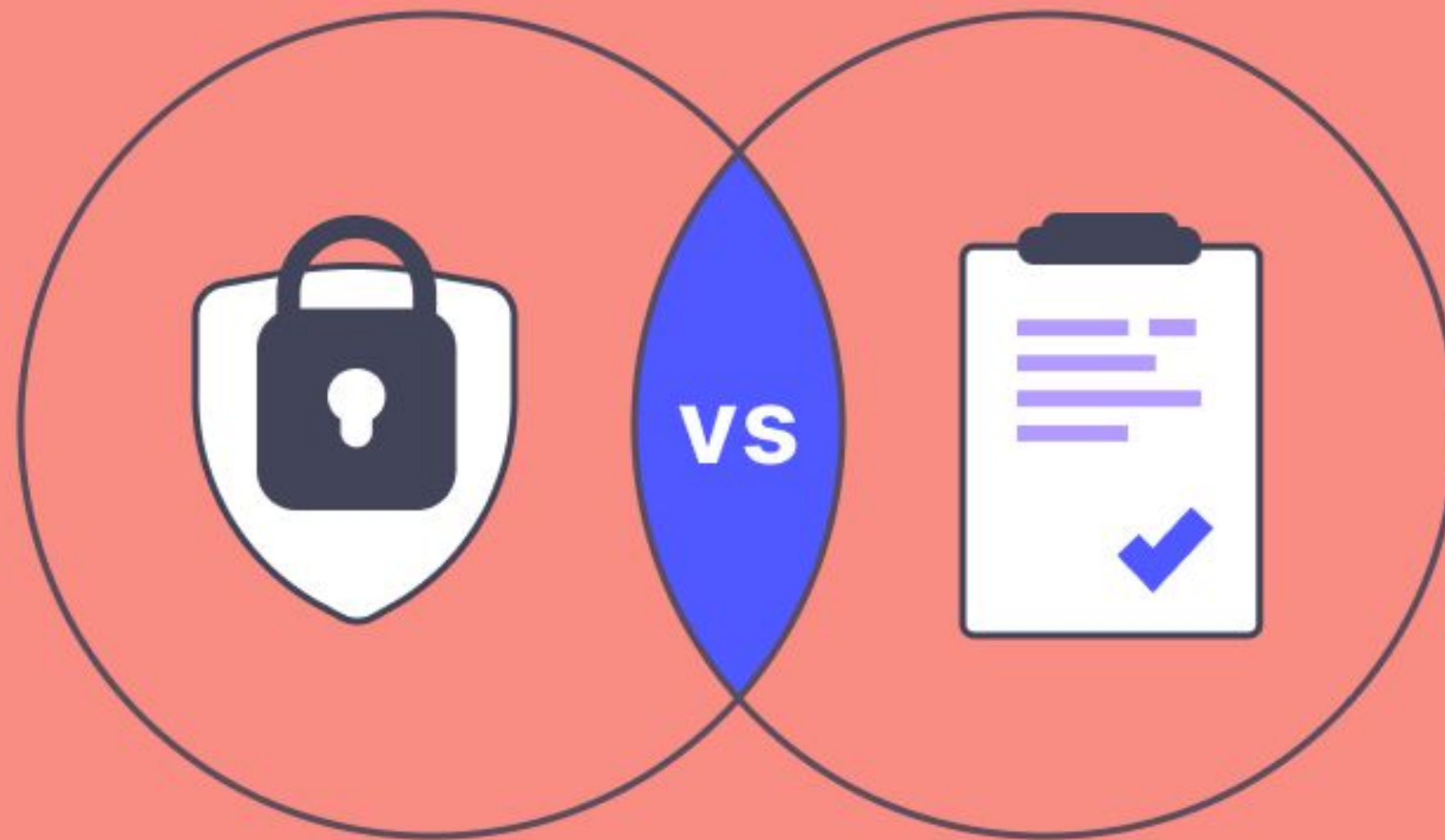
**Outdated Packages**

**Dynamic Testing - DAST**

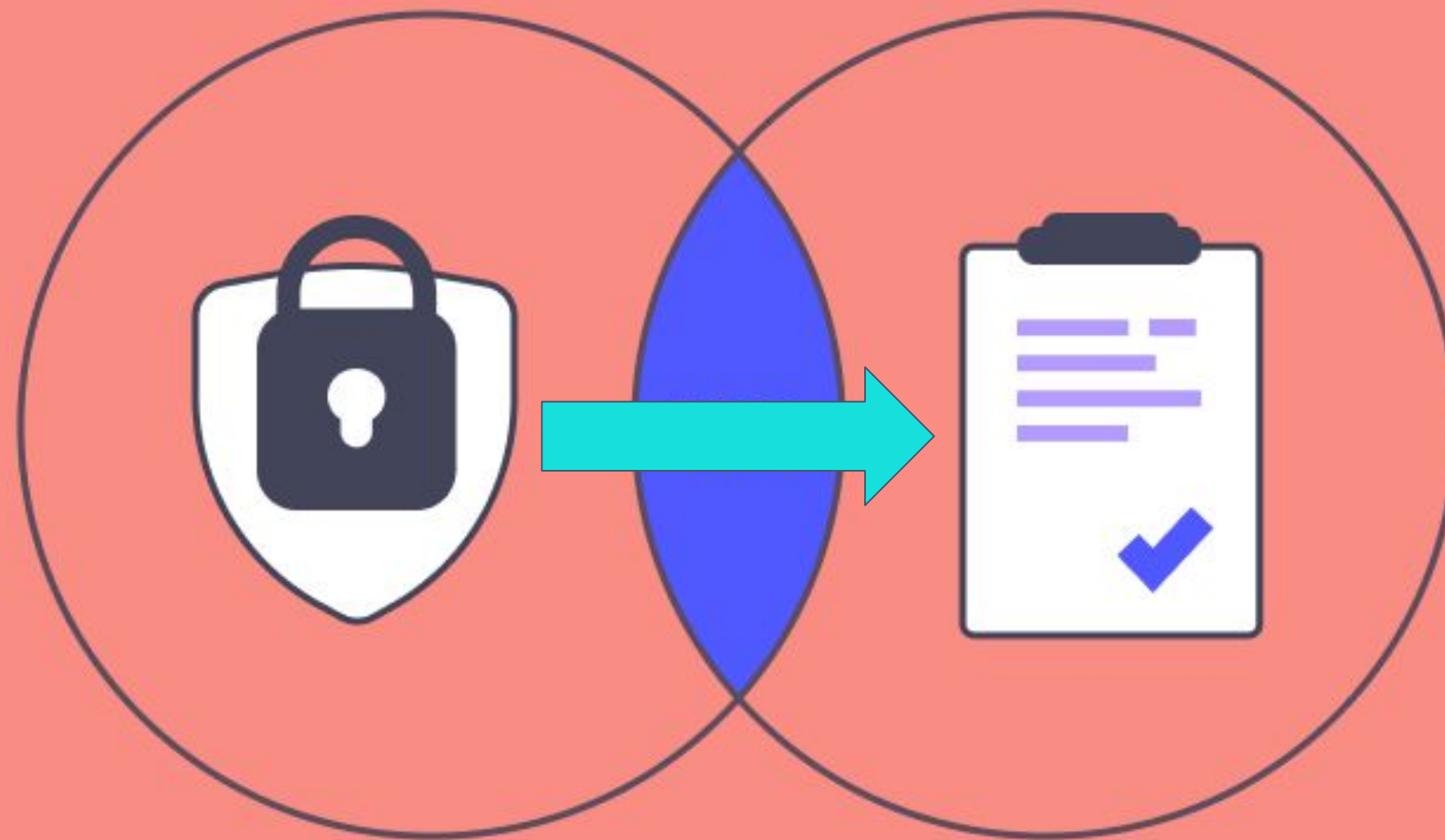
**Static Testing - SAST**



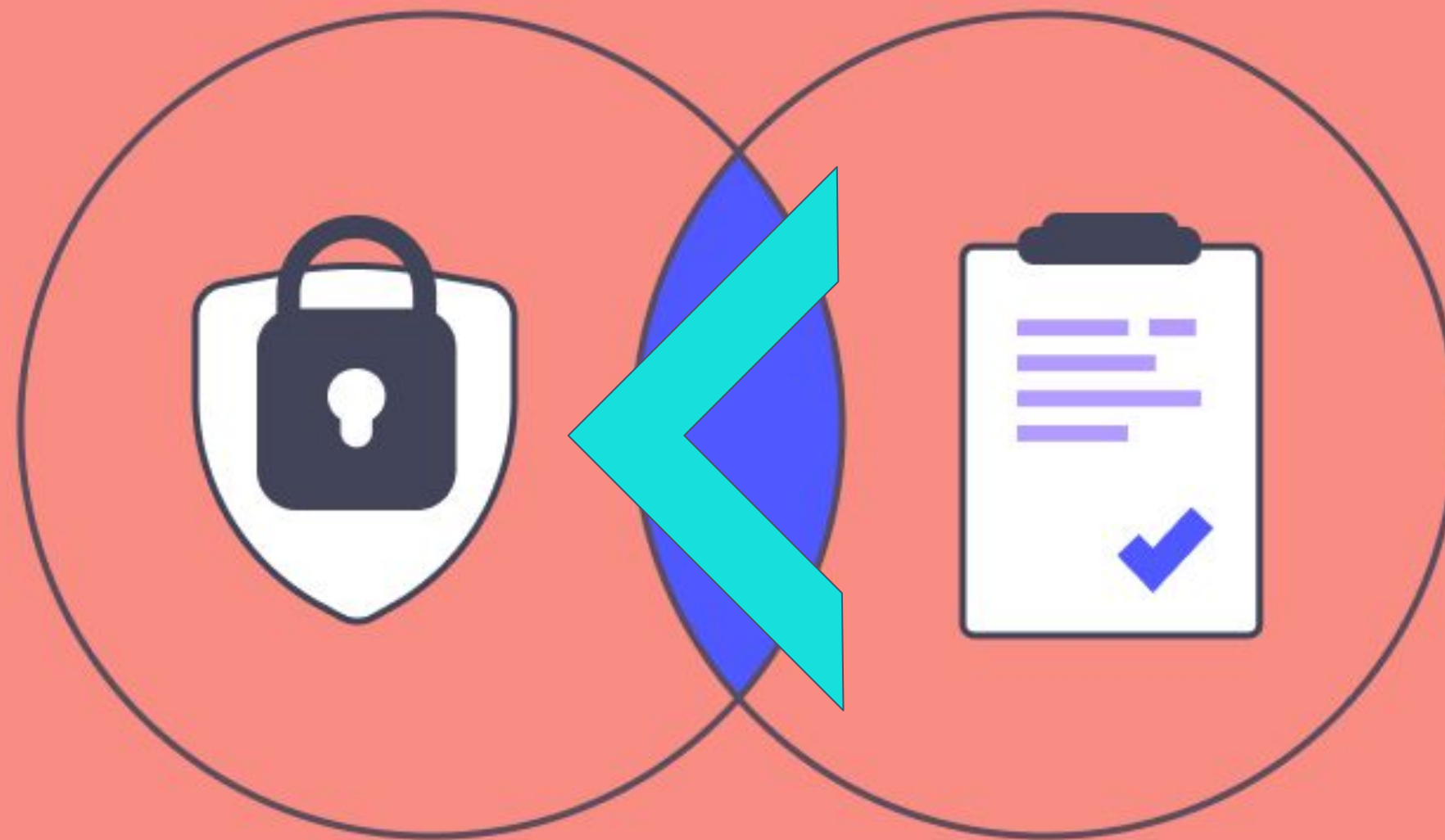
# Security vs. Compliance



# Security vs. Compliance



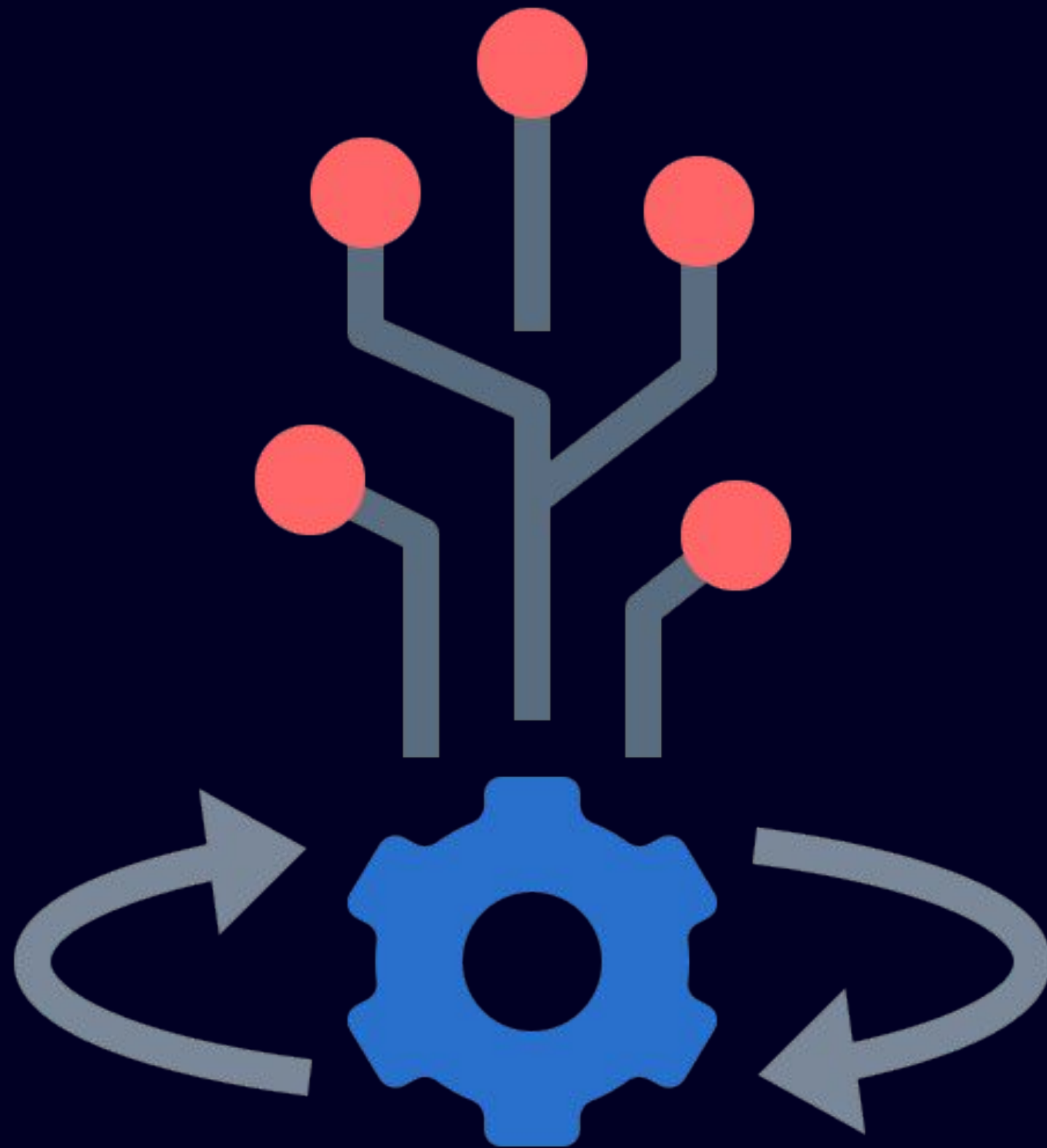
# Security vs. Compliance







## Integrate & Automate



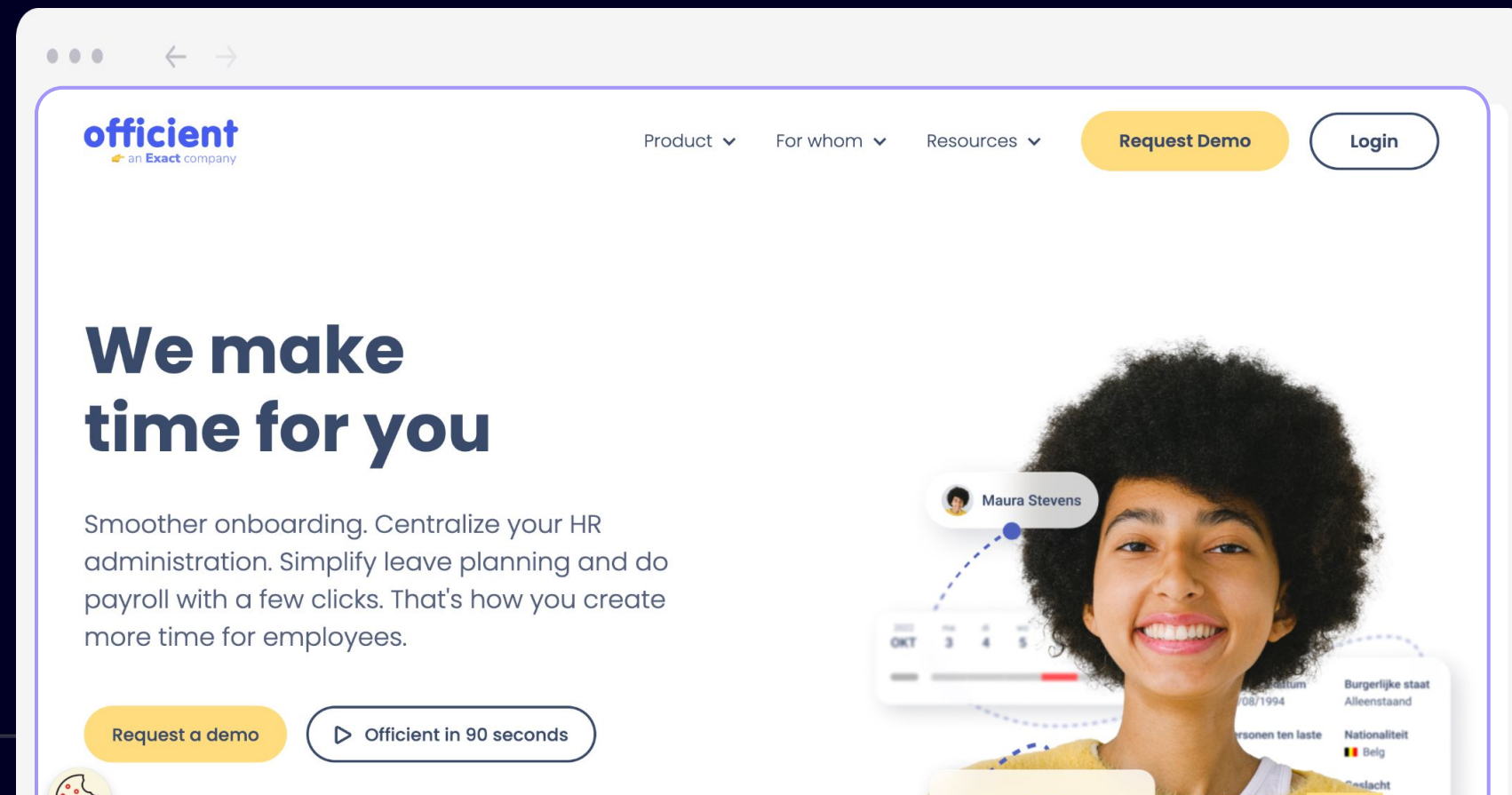


Where to **start**?

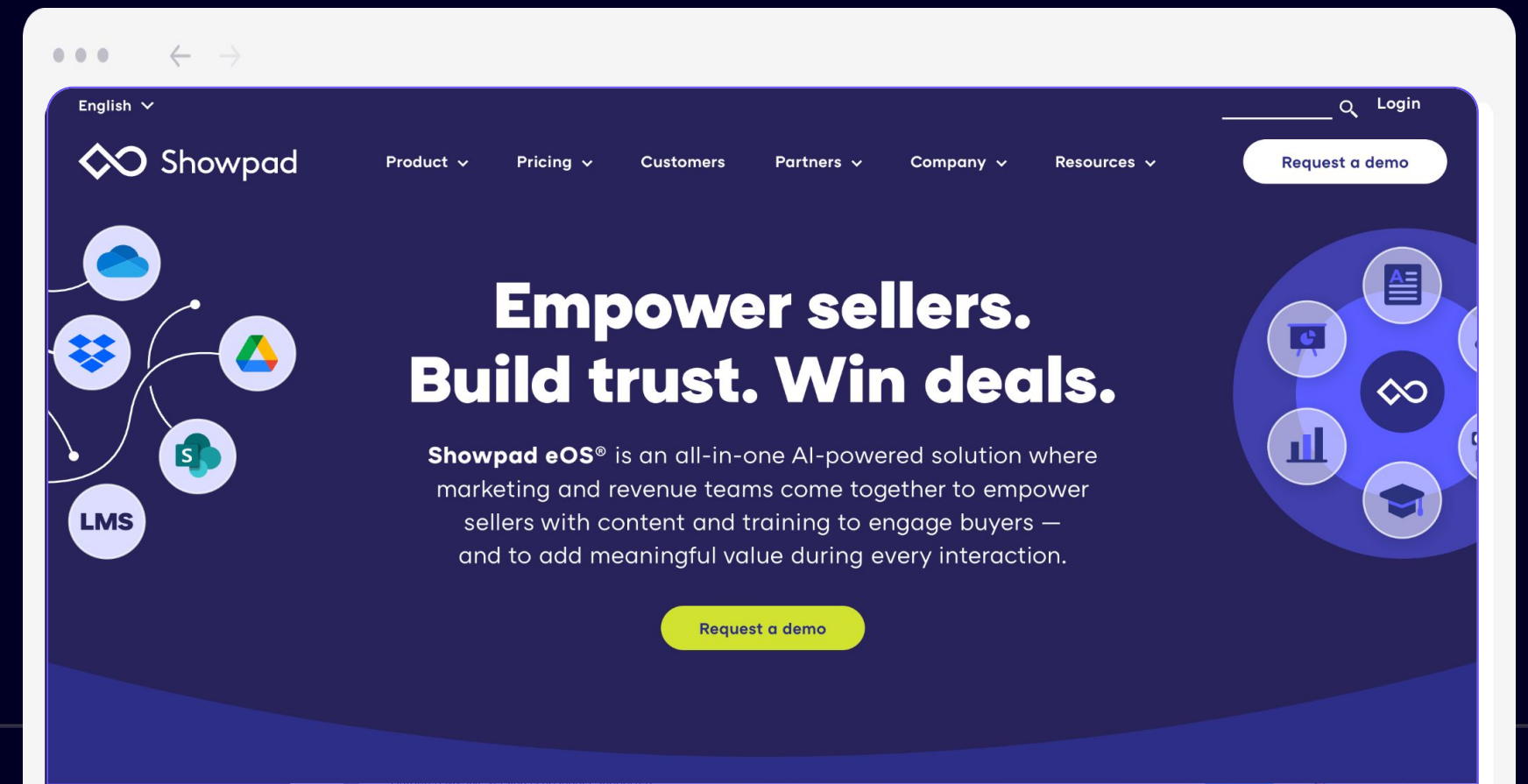
What to **focus** on?

# Past Experiences

## Officient HR Tech



## Showpad Sales Tech





# OWASP TOP 10





# Open Web Application Security Project

https://owasp.org/www-project-top-ten/

Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



- **A01:2021-Broken Access Control** moves up from the fifth position; 94% of applications were tested for some form of broken access control. The 34 Common Weakness Enumerations (CWEs) mapped to Broken Access Control had more occurrences in applications than any other category.
- **A02:2021-Cryptographic Failures** shifts up one position to #2, previously known as Sensitive Data Exposure, which was broad symptom rather than a root cause. The renewed focus here is on failures related to cryptography which often leads to sensitive data exposure or system compromise.
- **A03:2021-Injection** slides down to the third position. 94% of the applications were tested for some form of injection, and the 33 CWEs mapped into this category have the second most occurrences in applications. Cross-site Scripting is now part of this category in this edition.
- **A04:2021-Insecure Design** is a new category for 2021, with a focus on risks related to design flaws. If we genuinely want to “move left” as an industry, it calls for more use of threat modeling, secure design patterns and principles, and reference architectures.
- **A05:2021-Security Misconfiguration** moves up from #6 in the previous edition; 90% of applications were tested for some form of misconfiguration. With more shifts into highly configurable software, it’s not surprising to see this category move up. The former category for XML External Entities (XXE) is now part of this category.
- **A06:2021-Vulnerable and Outdated Components** was previously titled Using Components with Known Vulnerabilities and is #2 in the Top 10 community survey, but also had enough data to make the Top 10 via data analysis. This category moves up from #9 in 2017 and is a known issue that we struggle to test and assess risk. It is the only category not to have any Common Vulnerability and Exposures (CVEs) mapped to the included CWEs, so a default exploit and impact weights of 5.0 are factored into their scores.
- **A07:2021-Identification and Authentication Failures** was previously Broken Authentication and is sliding down from the second position, and now includes CWEs that are more related to identification failures. This category is still an integral part of the Top 10, but the increased availability of standardized frameworks seems to be helping.
- **A08:2021-Software and Data Integrity Failures** is a new category for 2021, focusing on making assumptions related to software updates, critical data, and CI/CD pipelines without verifying integrity. One of the highest weighted impacts from Common Vulnerability and Exposures/Common Vulnerability Scoring System (CVE/CVSS) data mapped to the 10 CWEs in this category. Insecure Deserialization from 2017 is now a part of this larger category.
- **A09:2021-Security Logging and Monitoring Failures** was previously Insufficient Logging & Monitoring and is added from the industry survey (#3), moving up from #10 previously. This category is expanded to include more types of failures, is challenging to test for, and isn’t well represented in the CVE/CVSS data. However, failures in this category can directly impact visibility, incident alerting, and forensics.
- **A10:2021-Server-Side Request Forgery** is added from the Top 10 community survey (#1). The data shows a relatively low incidence rate with above average testing coverage, along with above-average ratings for Exploit and Impact potential. This category represents the scenario where the security community members are telling us this is important, even though it’s not illustrated in the data at this time.

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**



# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**

**‘Code to cloud’ → SAST, IaC, CSPM**

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**

**‘Code to cloud’ → SAST, IaC, CSPM**

**‘Code to Runtime’ → SAST, DAST, RASP, WAF**

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**

**‘Code to cloud’ → SAST, IaC, CSPM**

**‘Code to Runtime’ → SAST, DAST, RASP, WAF**

**Threat Intelligence, Architecture → Intel & Training**

# OWASP top 10

- **A01:2021-Broken Access Control**
- **A02:2021-Cryptographic Failures**
- **A03:2021-Injection**
- **A04:2021-Insecure Design**
- **A05:2021-Security Misconfiguration**
- **A06:2021-Vulnerable and Outdated Components**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**

**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**

**‘Code to cloud’ → SAST, IaC, CSPM**

**‘Code to Runtime’ → SAST, DAST, RASP, WAF**

**Threat Intelligence, Architecture → Intel & Training**

**Cloud → CSPM, ASPM, ...**

# OWASP top 10

- **A01:2021-Broken Access Control**  
**Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...**
- **A02:2021-Cryptographic Failures**  
**‘Code to cloud’ → SAST, IaC, CSPM**
- **A03:2021-Injection**  
**‘Code to Runtime’ → SAST, DAST, RASP, WAF**
- **A04:2021-Insecure Design**  
**Threat Intelligence, Architecture → Intel & Training**
- **A05:2021-Security Misconfiguration**  
**Cloud → CSPM, ASPM, ...**
- **A06:2021-Vulnerable and Outdated Components**  
**Code, Containers, VMs → SCA, Malware, SupplyChain, ...**
- **A07:2021-Identification and Authentication Failures**
- **A08:2021-Software and Data Integrity Failures**
- **A09:2021-Security Logging and Monitoring Failures**
- **A10:2021-Server-Side Request Forgery (SSRF)**



# OWASP top 10

- |  |   |
|--|---|
| • <b>A01:2021-Broken Access Control</b>                      | <b>Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...</b>        |
| • <b>A02:2021-Cryptographic Failures</b>                     | <b>‘Code to cloud’ → SAST, IaC, CSPM</b>                        |
| • <b>A03:2021-Injection</b>                                  | <b>‘Code to Runtime’ → SAST, DAST, RASP, WAF</b>                |
| • <b>A04:2021-Insecure Design</b>                            | <b>Threat Intelligence, Architecture → Intel &amp; Training</b> |
| • <b>A05:2021-Security Misconfiguration</b>                  | <b>Cloud → CSPM, ASPM, ...</b>                                  |
| • <b>A06:2021-Vulnerable and Outdated Components</b>         | <b>Code, Containers, VMs → SCA, Malware, SupplyChain, ...</b>   |
| • <b>A07:2021-Identification and Authentication Failures</b> | <b>Cloud, Databases, ... → IAM, SSO, CSPM, ...</b>              |
| • <b>A08:2021-Software and Data Integrity Failures</b>       |   |
| • <b>A09:2021-Security Logging and Monitoring Failures</b>   |   |
| • <b>A10:2021-Server-Side Request Forgery (SSRF)</b>         |   |

# OWASP top 10

- |  |   |
|--|---|
| • <b>A01:2021-Broken Access Control</b>                      | <b>Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...</b>        |
| • <b>A02:2021-Cryptographic Failures</b>                     | <b>‘Code to cloud’ → SAST, IaC, CSPM</b>                        |
| • <b>A03:2021-Injection</b>                                  | <b>‘Code to Runtime’ → SAST, DAST, RASP, WAF</b>                |
| • <b>A04:2021-Insecure Design</b>                            | <b>Threat Intelligence, Architecture → Intel &amp; Training</b> |
| • <b>A05:2021-Security Misconfiguration</b>                  | <b>Cloud → CSPM, ASPM, ...</b>                                  |
| • <b>A06:2021-Vulnerable and Outdated Components</b>         | <b>Code, Containers, VMs → SCA, Malware, Supply Chain, ...</b>  |
| • <b>A07:2021-Identification and Authentication Failures</b> | <b>Cloud, Databases, ... → IAM, SSO, CSPM, ...</b>              |
| • <b>A08:2021-Software and Data Integrity Failures</b>       | <b>CI, Git, Databases → SAST</b>                                |
| • <b>A09:2021-Security Logging and Monitoring Failures</b>   |   |
| • <b>A10:2021-Server-Side Request Forgery (SSRF)</b>         |   |

# OWASP top 10

- |  |   |
|--|---|
| • <b>A01:2021-Broken Access Control</b>                      | <b>Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...</b>        |
| • <b>A02:2021-Cryptographic Failures</b>                     | <b>‘Code to cloud’ → SAST, IaC, CSPM</b>                        |
| • <b>A03:2021-Injection</b>                                  | <b>‘Code to Runtime’ → SAST, DAST, RASP, WAF</b>                |
| • <b>A04:2021-Insecure Design</b>                            | <b>Threat Intelligence, Architecture → Intel &amp; Training</b> |
| • <b>A05:2021-Security Misconfiguration</b>                  | <b>Cloud → CSPM, ASPM, ...</b>                                  |
| • <b>A06:2021-Vulnerable and Outdated Components</b>         | <b>Code, Containers, VMs → SCA, Malware, Supply Chain, ...</b>  |
| • <b>A07:2021-Identification and Authentication Failures</b> | <b>Cloud, Databases, ... → IAM, SSO, CSPM, ...</b>              |
| • <b>A08:2021-Software and Data Integrity Failures</b>       | <b>CI, Git, Databases → SAST</b>                                |
| • <b>A09:2021-Security Logging and Monitoring Failures</b>   | <b>‘Code to Runtime’ → SIEM, Monitoring</b>                     |
| • <b>A10:2021-Server-Side Request Forgery (SSRF)</b>         |   |

# OWASP top 10

- |  |   |
|--|---|
| • <b>A01:2021-Broken Access Control</b>                      | <b>Crown Jewels (Cloud, Code, ...) → CSPM, ASPM, ...</b>        |
| • <b>A02:2021-Cryptographic Failures</b>                     | <b>‘Code to cloud’ → SAST, IaC, CSPM</b>                        |
| • <b>A03:2021-Injection</b>                                  | <b>‘Code to Runtime’ → SAST, DAST, RASP, WAF</b>                |
| • <b>A04:2021-Insecure Design</b>                            | <b>Threat Intelligence, Architecture → Intel &amp; Training</b> |
| • <b>A05:2021-Security Misconfiguration</b>                  | <b>Cloud → CSPM, ASPM, ...</b>                                  |
| • <b>A06:2021-Vulnerable and Outdated Components</b>         | <b>Code, Containers, VMs → SCA, Malware, Supply Chain, ...</b>  |
| • <b>A07:2021-Identification and Authentication Failures</b> | <b>Cloud, Databases, ... → IAM, SSO, CSPM, ...</b>              |
| • <b>A08:2021-Software and Data Integrity Failures</b>       | <b>CI, Git, Databases → SAST</b>                                |
| • <b>A09:2021-Security Logging and Monitoring Failures</b>   | <b>‘Code to Runtime’ → SIEM, Monitoring</b>                     |
| • <b>A10:2021-Server-Side Request Forgery (SSRF)</b>         | <b>‘Code to Runtime’ → SAST, DAST, RASP</b>                     |

🚩 As a ... CTO, Head of DevOps, VP Engineering, ...

**Leaked Secrets**

**Cloud Misconfigurations**

**Open-Source Dependencies**

**Malware**

...

**License Risks**

**Runtime protection**

**Dynamic Testing - DAST**

**Static Testing - SAST**



As a ... CTO, Head of DevOps, VP Engineering, ...



GitGuardian

Cloud Misconfigurations

Open-Source Dependencies

Malware

...

License Risks

Runtime protection

Dynamic Testing - DAST

Static Testing - SAST

As a ... CTO, Head of DevOps, VP Engineering, ...



Open-Source Dependencies

Malware

...

License Risks

Runtime protection

Dynamic Testing - DAST

Static Testing - SAST

As a ... CTO, Head of DevOps, VP Engineering, ...



...

Malware

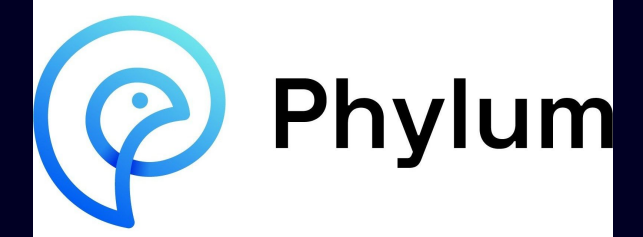
License Risks

Runtime protection

Dynamic Testing - DAST

Static Testing - SAST

As a ... CTO, Head of DevOps, VP Engineering, ...



...

License Risks

Runtime protection

Dynamic Testing - DAST

Static Testing - SAST

As a ... CTO, Head of DevOps, VP Engineering, ...



...



Runtime protection

Dynamic Testing - DAST

Static Testing - SAST



As a ... CTO, Head of DevOps, VP Engineering, ...



...



Dynamic Testing - DAST

Static Testing - SAST



As a ... CTO, Head of DevOps, VP Engineering, ...



...



Static Testing - SAST

As a ... CTO, Head of DevOps, VP Engineering, ...



...



But... current security software s\*cks

### Scattered tooling

Mix of closed & open-source. Alerts across solutions. Lacks overview.



### Noisy

Loads of false positives. Notification overload & alert fatigue.



### Confusing

UX Nightmare like a fighter-jet cockpit. Makes you feel dumb. No Self-serve.



### Pricy

Ridiculous prices. Can't try before you buy. Intransparent & pushy sales.





# All-in-One Software Security

## Code (ASPM)



### Static Code Analysis (SAST)

Semgrep Gosec Custom



### Secrets Detection

Gitleaks



### Malware

Phylum



### Infrastructure as Code

Checkov

## Containers



### Open Source Dependencies (SCA)

Trivy Syft Grype



### Open Source License Risks

Grype Custom



### Outdated Software

endoflife.date

## Cloud (CSPM)



### Cloud Posture Management (CSPM)

CloudSploit Custom



### Agentless Virtual Machine Scanning

Aikido Custom Scanner

## Domains



### Dynamic Testing (DAST)

ZAP Nuclei Custom



### Authenticated DAST

Aikido Custom Scanner



### End-point API Scanning

Akto

## Runtime



### Firewall for Node.js

Zen by Aikido



### Firewall for Python

Zen by Aikido

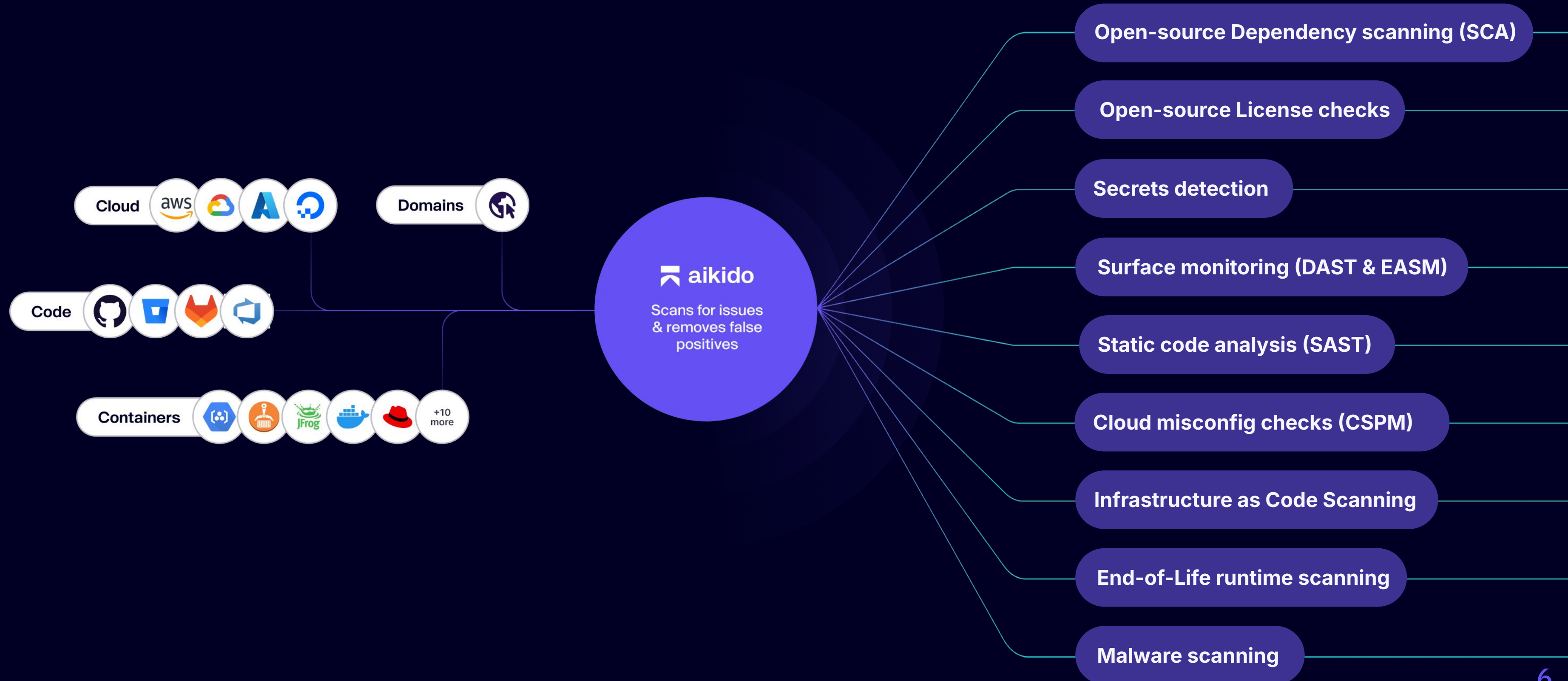


### Firewall for Ruby, PHP, .NET and Java

Zen by Aikido

Coming soon

# How it works



That's why they built

Try with your git or a demo account or Talk to an expert



11-tools-in-1

Noise massively reduced

UI & UX stellar

Tech & Vendor agnostic

Price not ridiculous

Starter plan

Mindmeld AI

Feed

Hot Links

Critical 2

Autofix 22

Ignored

Solved

Repos

admin-email-digest 2

company-service-surface 2

testing-design-library-supe... 2

Hello, Amber!

999

- 99 Critical
- 23 Medium
- 99 High
- 12 Low

16 issues solved up from 12 last month

18 new issues in the last month

45 auto triaged issues in the last month

Search

All types

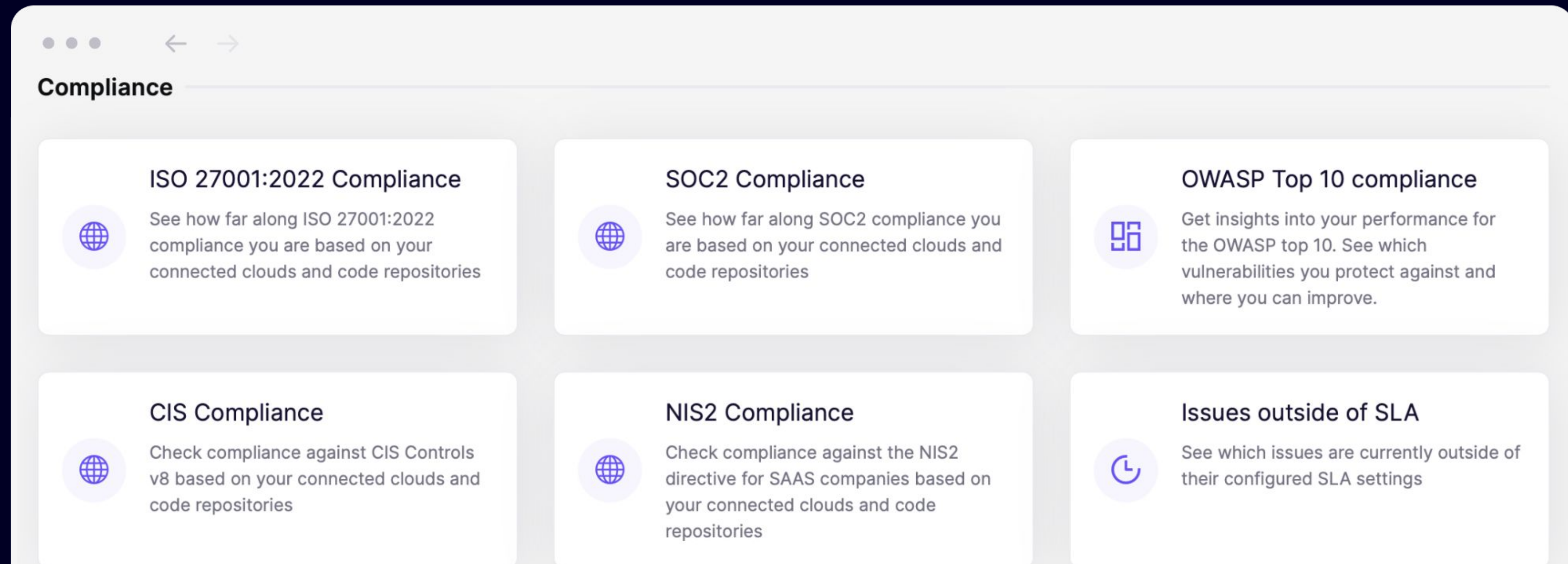
Filtered to critical

Refresh

Type	Name	Severity	Location	Age	Status	Assignee
OS Dependencies	Django - Old version allo	Critical	poetry.lock	20 h	New	
Cloud posture	Load balancer allows inv	High	Production	5 h	To Do	
Exposed secrets	GCP api key secret expo	High	marketing.py	2 m	PR Open	Bert
SAST	Using Pickle can lead to remote code execution					
Surface monitoring						
Infrastructure as code						



# Killer for compliance automation



# Killer for compliance automation

## OWASP top 10 breakdown

75% 

[Export To PDF](#)

A01:2021 Broken access control

80%  

A02:2021 Cryptographic failures

89%  

A03:2021 Injection

68%  

A04:2021 Insecure design

0%  

A05:2021 Security misconfiguration

78%  

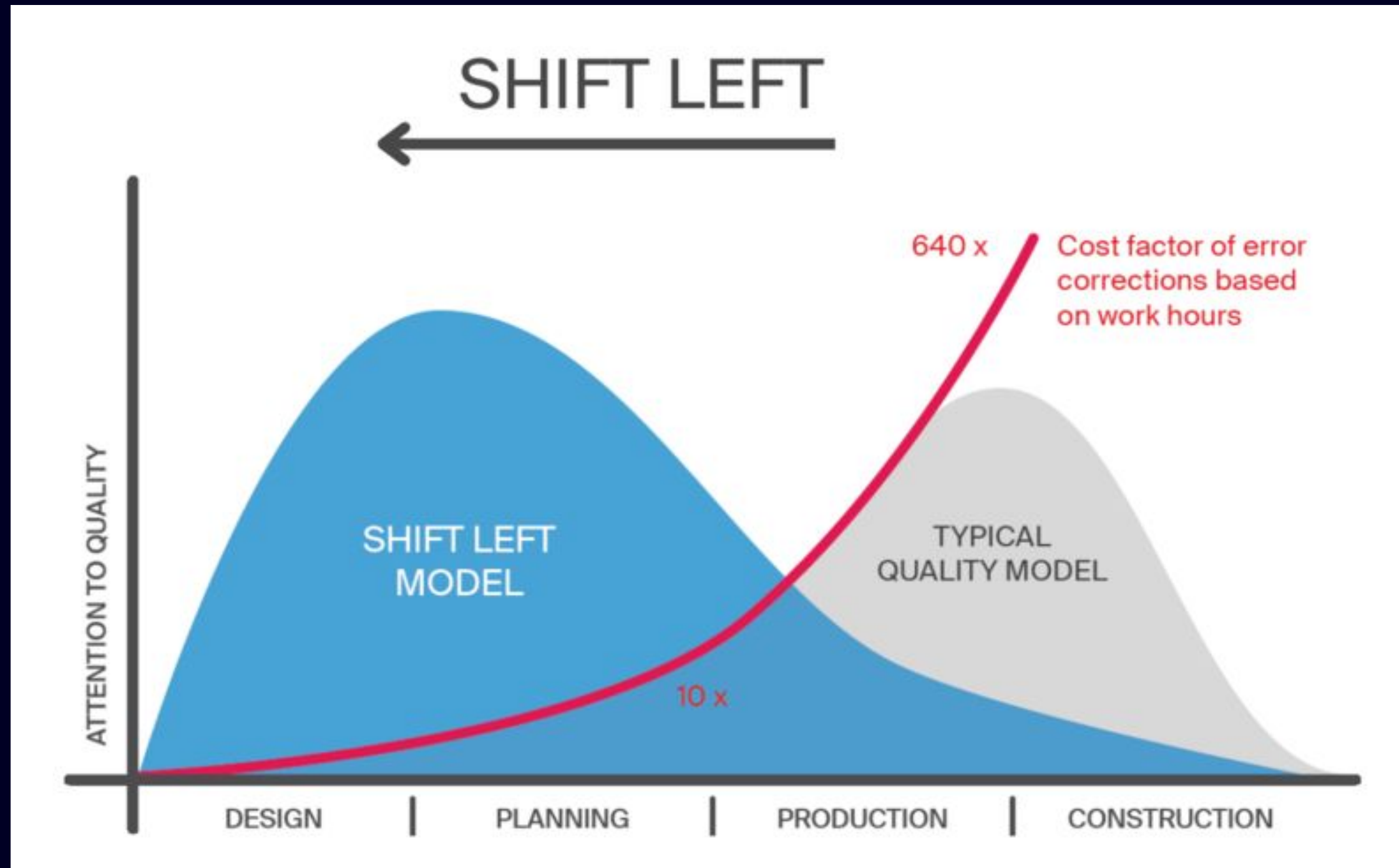
A06:2021 Vulnerable and Outdated Components

73%  



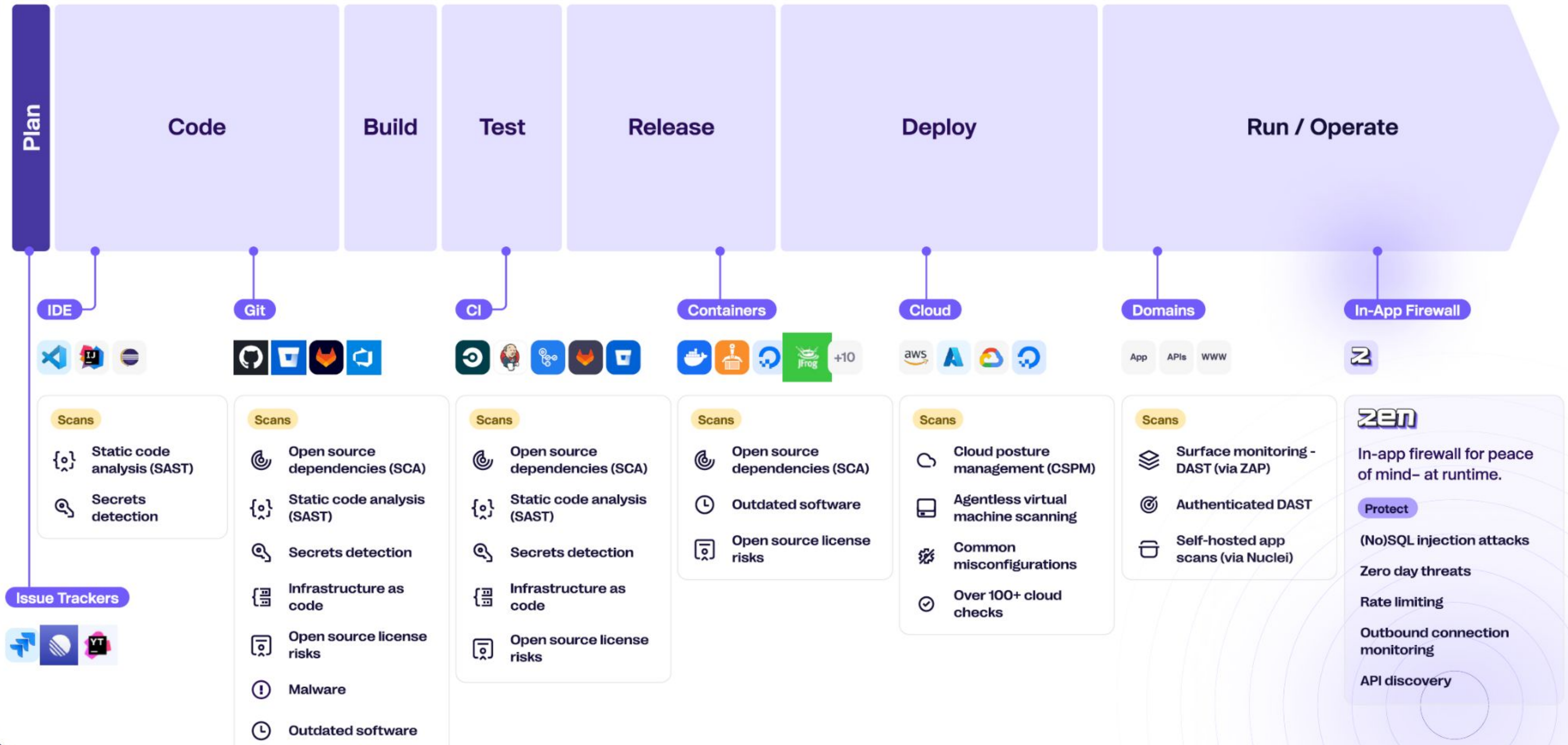
Shift Left

# Shift Left





# Aikido Shift-Left



## Aikido Shift-Left

×

### Manage configuration for **ACCESS-NYC**

This configuration will be applied to selected repos.

Scan Failure For Minimum Severity

Critical ▾

Dependency Scan

☒

SAST Scan

☒

IaC Scan

☐

Secrets Scan

☐

Cancel

Save Configuration



# Aikido Shift-Left

## Manage configuration for **ACCESS-NYC**

This configuration will be applied to selected repos.

Scan Failure For Minimum Severity

Critical

Dependency Scan



SAST Scan



IaC Scan




Secrets Scan



Cancel

Save Configuration



```
125   res.status(500).send('Error loading config');
126   }
127   });
128
129   app.list
130   cons
131   });
132   const ap
133
134   app.get(
135     const PW = "sk_live_fakestripeapikeyleaked12"
136     res.status(200).send(STRIPE_API_KEY)
137   );
138
139   app.registerMethod(
140     'fetch',
141     Acl.ensure(function* (encryptedToken) {
142       try {
143         const decrypted = decryptJSON(encryptedToken, ENCRYPTION_ALGORITHM, ENCRYPTION_KEY, OLD_ENCRY
144         yield management.users.findOne({ _id: decrypted._user });
145       } catch (e) {
146         console.error(e);
147       }
148     })
149   )
```

Found a Stripe Access Token, posing a risk to payment processing services and sensitive financial data. (No matching Aikido repository found - noise suppression not possible) Aikido Secrets(stripe-access-token)

Aikido: Report false positive

View Problem (\F8) No quick fixes available



# Democratize Software Security

## Strong traction within regulated industries, ICP

Healthtech

Securitytech

Legaltech

Fintech

HRtech

Devtech

Climatech

Enterprise SaaS

Agencies

Non-software core

Group-companies

...

## Driven by:

**SOC2 & ISO27001 ↗**

**Bundling ↗**

**Cost Effectiveness ↗**

**Ease of use ↗**

**Automated ↗**

**Integrated ↗**

# Winning 100% inbound, our Product Led Growth engine

## Self-Service

Free

€0

Up to 2 users

- ✓ 10 repos & 1 cloud
- ✓ Nightly rescans
- ✓ Slack alerts

## Sales-Assisted

Standard

€299/month

Up to 10 users

- ✓ 100 repos & 3 clouds
- ✓ CI gating
- ✓ Task-tracker integrations
- ✓ Reporting

Pro

€599/month

Up to 10 users

- ✓ 250 repos & 10 clouds
- ✓ On-prem scanning
- ✓ IDE plugins
- ✓ Malware detection

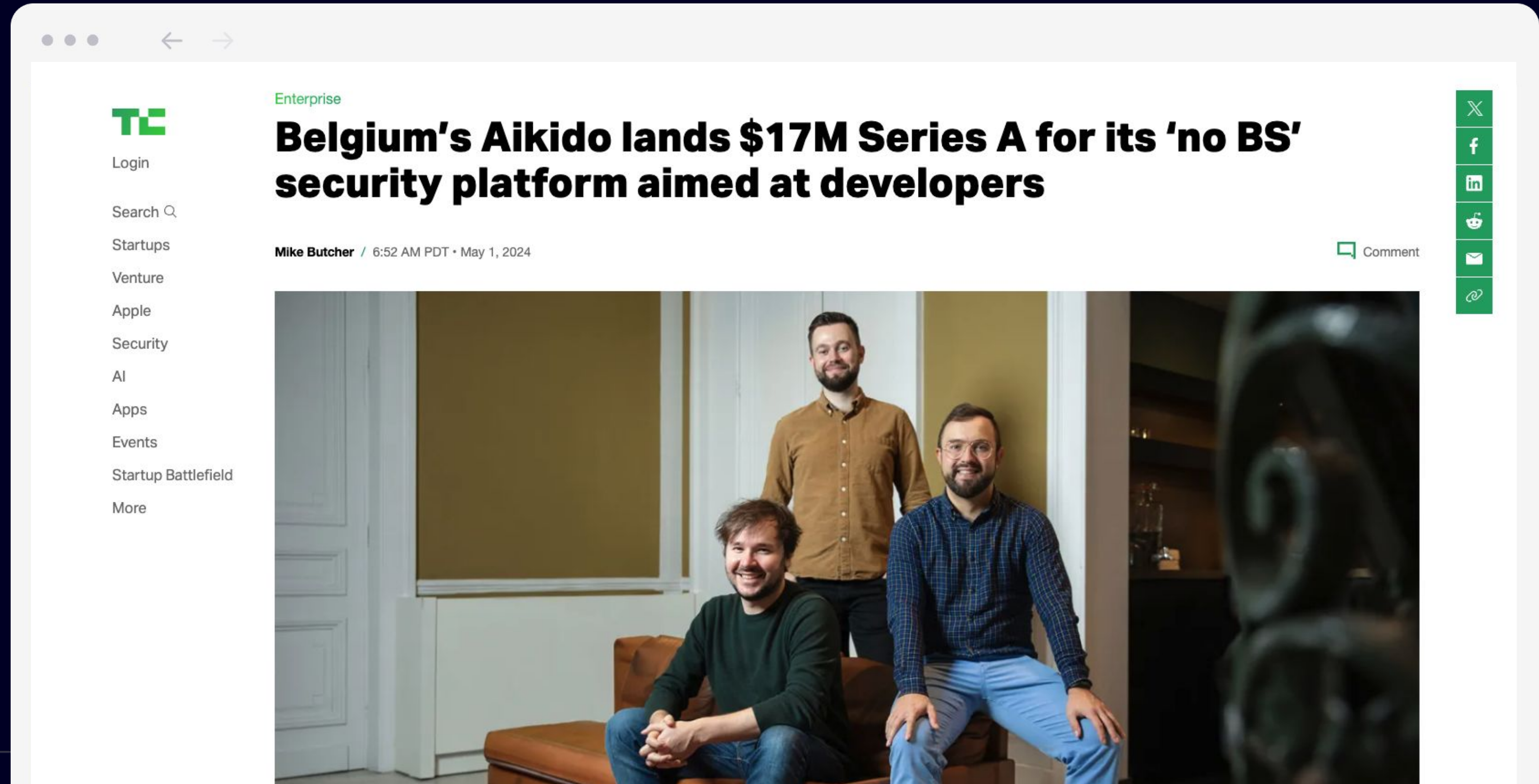
## Sales-Led

Enterprise

Custom

- ✓ ∞ repos & clouds
- ✓ Admin Portal
- ✓ Training & onboarding
- ✓ Enterprise support

# We're gaining steam because we refuse to compromise





- 1 Thank
- 2 you!
- 3
- 4
- 5
- 6
- 7
- 8 ↳ aikido.dev





1

2

3

↳ aikido.dev

# Annex

& back-up slides

# The competition is

 covering only few areas,

  hard to use,  expensive,

   creating tons of false positives,

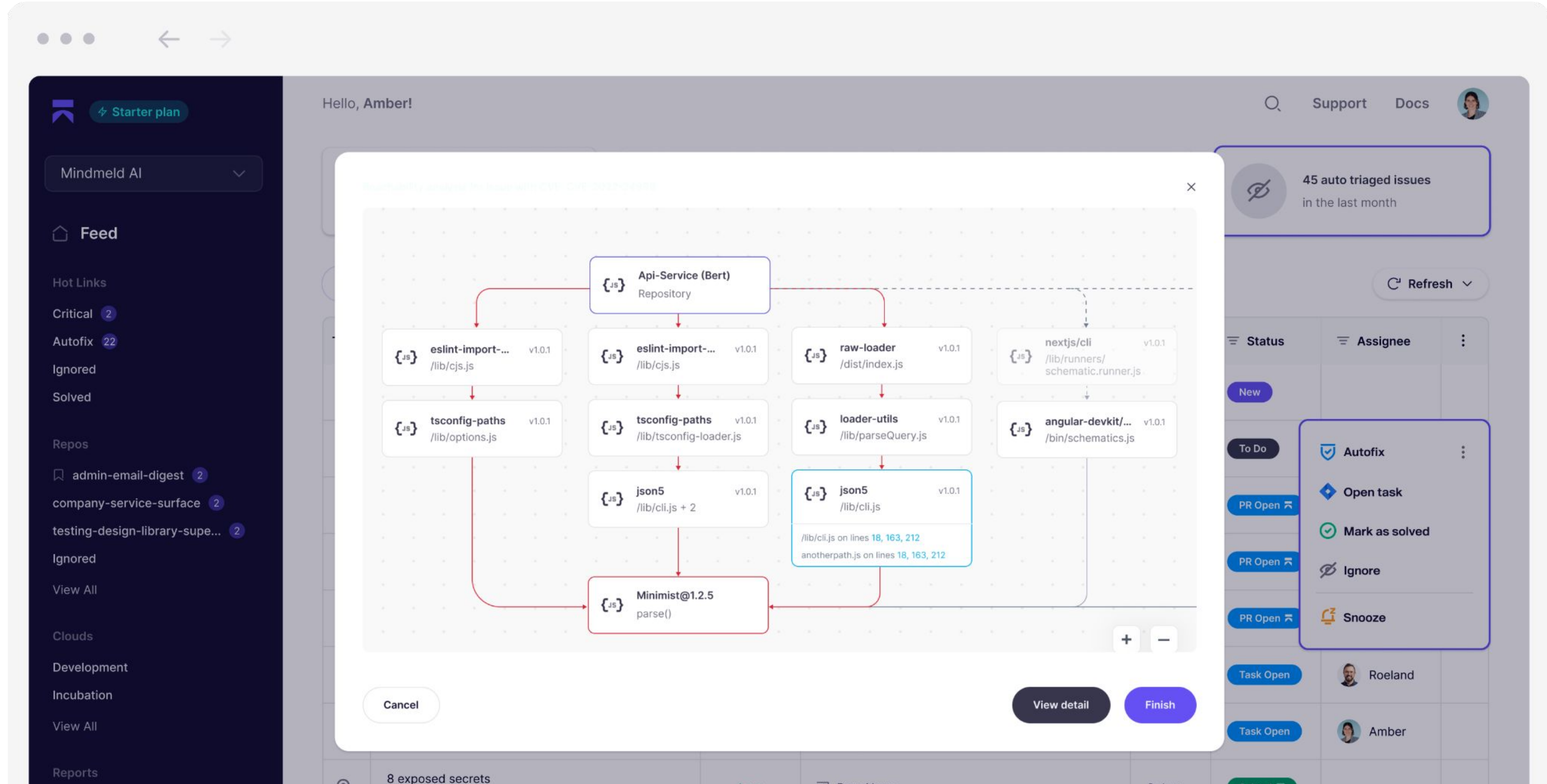
 built on legacy tech,  sales-heavy

.

# Automatic triaging and fixing false positives

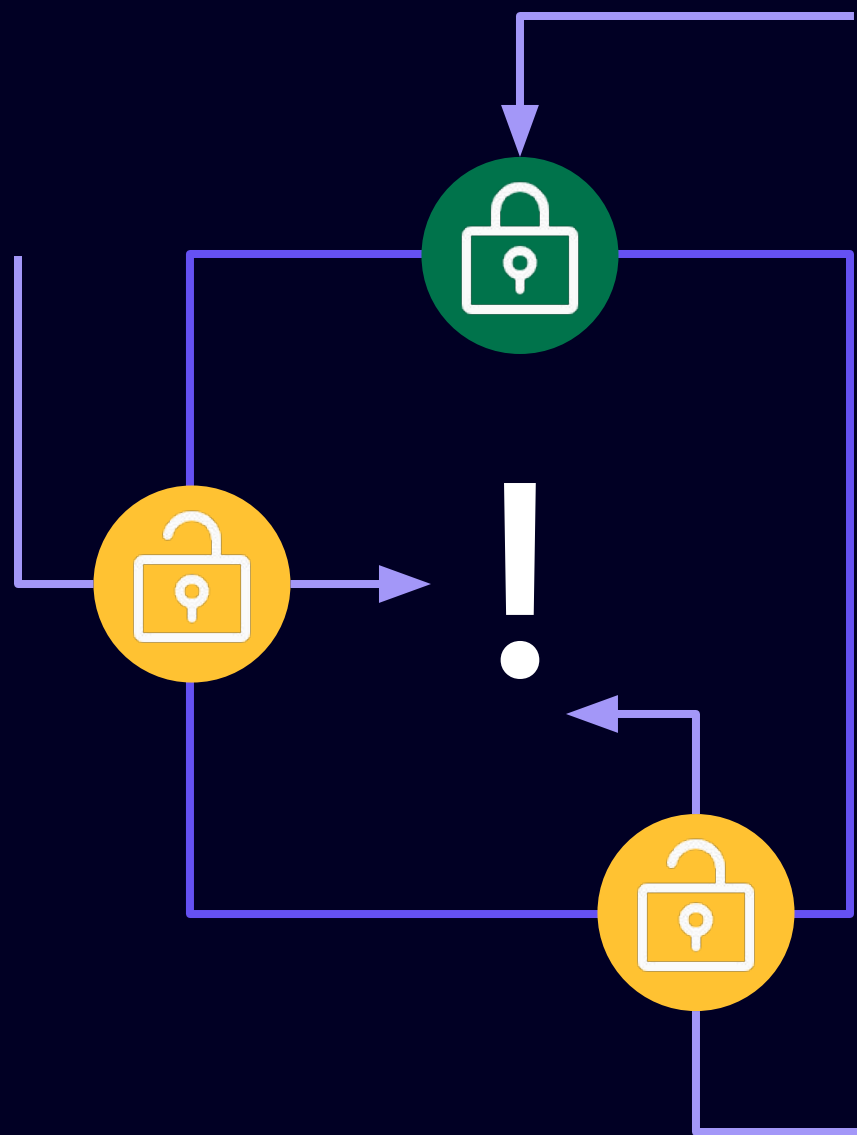
#	Reason for ignoring	# issues	Description
1	● Affected function not in use	↑ 84	Reachability analysis has determined that this vulnerability can not affect your environment.
2	● Dependency not used in production	↑ 67	This package is a part of the toolchain you are using to build & test your application. Since it's not being shipped into the production, solving the issue is not urgent.
3	● Exploits effects only end-user browser speed	↑ 44	This exploit can only cause the application to become slower in the end-user's browser. Since there is no security risk and can only affect uptime for 1 user at a time, Aikido has downgraded its severity.
4	● Frontend path traversal attacks	↑ 34	This exploit can cause path traversal attacks, but Aikido assumed there is no direct filesystem access from the browser, so those attacks are of no importance.
5	● Library not typically used in packaged frontend ...	↑ 21	This library is usually not used in any packaged front-end application because it mimics browser behavior in nodejs. Thus, Aikido assumes it is unused, and thus not important to fix.
6	● CVE not valid	↑ 18	This CVE number has either been reserved for a future publication or has been retracted.

# Reachability Analysis



[Optional – if you're still looking]

# White-box level **pentesting**, without giving access to your code



- + Built for SaaS [we collaborate with top-notch bounty hunters]
- + White-box level insights [bespoke testing based on your setup]
- + Fair price for actual, manual work [no automated tests]
- + Continuously keeping you secure [we monitor your environment every 24h]

↳ For more information, click [here](#)