KONDUKTO

How to Create A Security Program From Scratch?

Case Study

How Ödeal uses Kondukto to jump-start their AppSec program

A case study about creating an AppSec program with Ödeal.

Chapters

The Challenges

- Security was not baked into the software **development lifecycle** in a structured way. Security tools were randomly used by the software development teams at will.
- **Penetration tests** performed on a quarterly basis were **putting pressure on development teams** as the remediation of vulnerabilities was interfering with their development activities.
- There was **no visibility into the vulnerabilities of applications** and there was **no supervision of security metrics** in the absence of a security engineer.

KONDUKTO | ödeal

How to Create A Security Program From Scratch?

Case Study

About Ödeal

Founded in 2014, Ödeal is the **provider of a payment technology** that allows subscribed businesses to receive payments by debit or credit cards using their mobile phones. With **more than 68.000 subscribed businesses** and a solid business model, the company has received funding in 2021 and has been growing its technology teams rapidly since then.

The Situation

As a rapidly growing fintech start-up with a growing tech team, Ödeal was in the midst of **restructuring its CI/CD processes.**

Due to the sensitive nature of business, security was a concern among the leadership team but with so much on developers' plates, time could hardly be allocated to **creating a CI/CD pipeline** integrated with security tools.

There was a need for a platform where **security tests** could **easily be integrated with CI/CD pipelines** which could also provide visibility on the overall security posture of applications.

KONDUKTO | ödeal

How to Create A Security Program From Scratch?

Case Study

Our Approach

1. As a first step, **applications** existing on the ALM tool were **automatically fetched to Kondukto** after integrating Kondukto with the ALM tool in minutes.

2. Based on the tech stack used in each application, the relevant **open source security tools** on Kondukto **were configured and associated** with relevant applications.

3. As Ödeal already had a git-flow branching mechanism in place, **security tests** were positioned in **between feature and development branches** so that in each pull request, security tests could be triggered on open-source security tools by Kondukto.

4. A company-wide script was written in such a way that if the repo was recently created on the ALM tool and not yet on Kondukto, it was **automatically created through the CLI based on the hierarchy** used by software development teams.

5. Leveraging the label-based automation capabilities of Kondukto, **different automation policies** were **applied** to applications with different labels (i.e. internal, external, GDPR).

6. **PR decoration** functionality of Kondukto CLI also allowed developers to **gain access to the vulnerabilities** discovered in each PR on their ALM tool so that they were aware of the vulnerabilities before they advanced to further stages in the pipeline.

Results

Without spending a fortune on commercial security tools and even without having a security team at the company, **the technical aspect of DevSecOps transition was easily achieved** by writing a company-wide pipeline script that is used in all applications.

Thanks to this script, now any team that transitions to the new CI/CD pipeline **automatically has its projects onboarded to Kondukto and starts running pre-defined security tests** within the pipelines **in a self-serve manner.**

In the absence of a security team, label-based automation rules encouraged basic **threat modeling** activities in the development teams which led to **increased awareness about the risk perceptions** of applications. This approach also **helped to keep the spotlight on the vulnerabilities** that are most likely to pose a real threat without creating noise for the development teams.