# Protecting your web application/API with CrowdSec

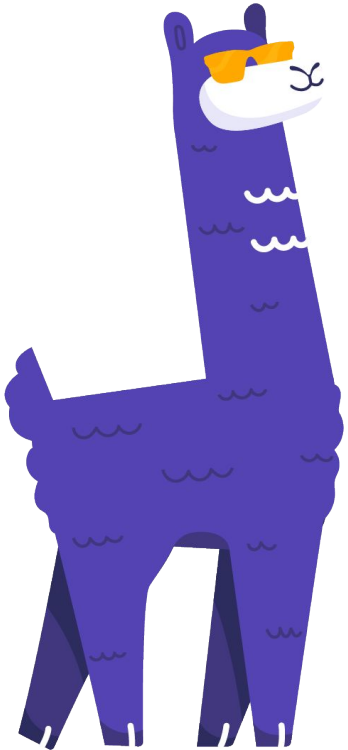
*(and common sense)*

CrowdSec

# Agenda

- Why care about AppSec?
- OWASP top 10
- SDLC and AppSec
- AppSec suggestions
- CrowdSec and AppSec
- Wrapping up
- Questions

CrowdSec



# But before we start...

CrowdSec

# What 20 years of infosec experience has taught me...

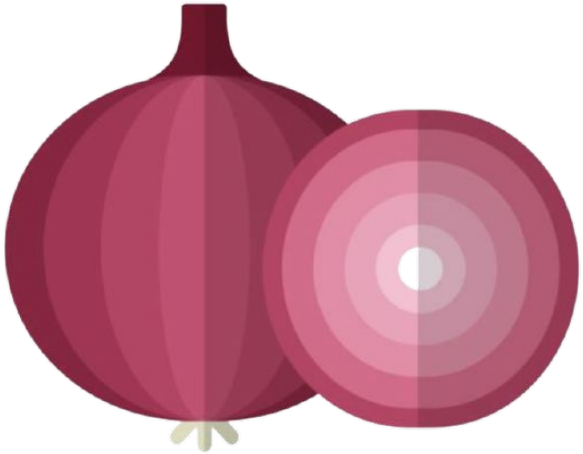
CrowdSec

CrowdSec

# Don’t start with a pentest

CrowdSec

# Good security is layered

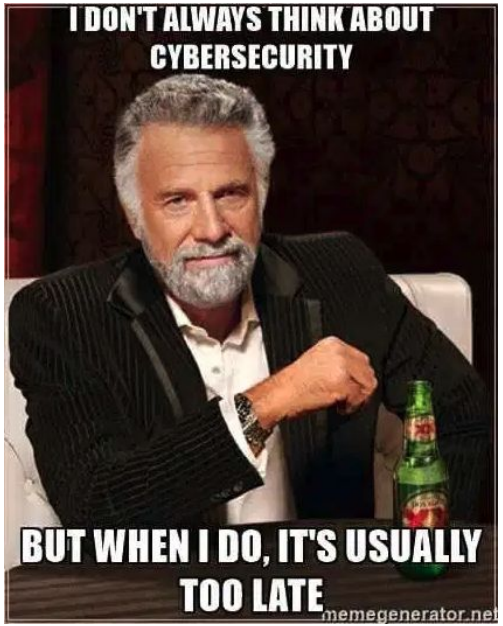


CrowdSec

# Communication and culture is more important than you'd think

CrowdSec

# You’re working with humans. They’re irrational by definition



CrowdSec

# Don’t start with AppSec

https://www.cisecurity.org/controls

# Why application security is important

With **cloud computing** this is literally the only place left to screw up.

Where in the stack is a vulnerability most likely?

# Don’t forget API security

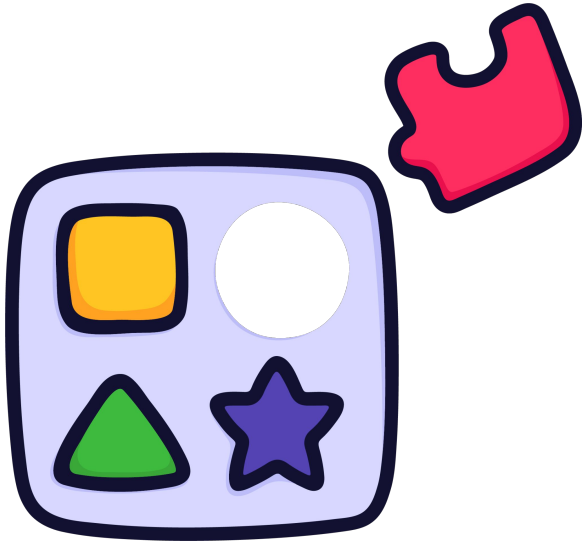
Easy to overlook

# Why focus on application security


OWASP Top 10:2021
A01 Broken Access Control
A02 Cryptographic Failures
A03 Injection
A04 Insecure Design
A05 Security Misconfiguration
A06 Vulnerable and Outdated Components
A07 Identification and Authentication Failures
A08 Software and Data Integrity Failures
A09 Security Logging and Monitoring Failures
A10 Server Side Request Forgery (SSRF)

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

# A05 Security Misconfiguration



Not really coding related
It's important *how you implement* any application

Insufficient hardening and missing patches:

- Permissions on cloud services
- Unnecessary features
- Default accounts/passwords
- After upgrade, security features not enabled
- Security settings in application frameworks not enabled
- Security patches not installed (this can be really critical)

https://owasp.org/Top10/A05_2021-Security_Misconfiguration/

# A04 Insecure Design

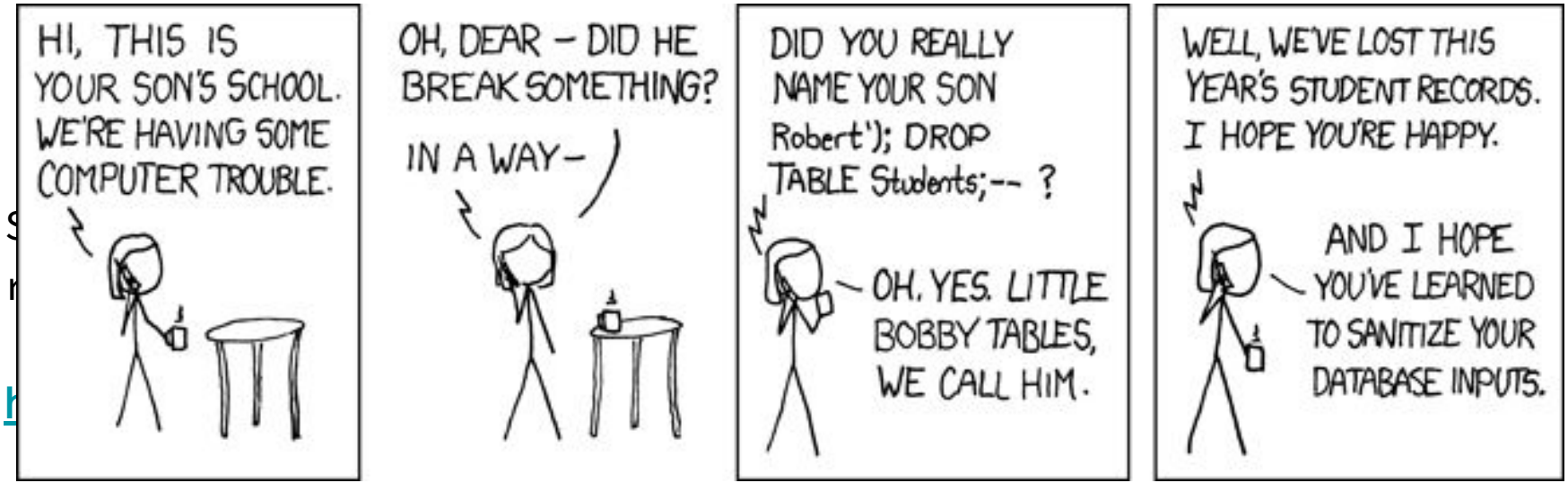
Broad category
“Missing or ineffective control design”

Typically the result of a lack of business risk profiling

This is where AppSec comes in. More later!


https://owasp.org/Top10/A04_2021-Insecure_Design/

# A03 Injection




https://owasp.org/Top10/A03_2021-Injection/

# A02 Cryptographic Failures

Insufficient protection of data in transit and/or rest

- Clear text data transmissions
- Legacy cryptographic in use
- Default/reused/insecure/leaked keys in use (or leaked in repo)
- Use of self signed certs
- Deprecated hashing functions (SHA1/MD5)


**Pro tip:** Don't implement any cryptographic functions yourself (unless you REALLY know what you're doing)

https://owasp.org/Top10/A02_2021-Cryptographic_Failures/

# A01 Broken Access Control

When access control is faulty somehow

- No least privileges or deny by default
- Bypassing access controls by modifying URL in browser (classic!) or API request
- API with missing ACL for POST/PUT/DELETE

Leads to

- Unauthorized information disclosure (Hello, GDPR fine)
- Modification or deletion of data


https://owasp.org/Top10/A01_2021-Broken_Access_Control/

# I’m worried! Now what?

Think strategically
Turning into a communication task:

- Documentation
- Speak the language of the intended audience (management)
- Talk about financial risk
- CIA triad is essential

# Free book tip!

Great overview to get started
Very practical approach

👍


https://shehackspurple.ca/books/

# SDLC vs AppSec


SDLC is a well defined process and a ***common language***
AppSec - not so much

https://www.cisecurity.org/controls

# How to get started?

SAMM

Start with determining your baseline

Best way to grasp **SAMM** is by using it
Please give back to the community by sharing your results

https://owaspsamm.org/

# Common pitfall

Using OWASP top 10 as a checklist

Instead use:

- OWASP ASVS
- OWASP Proactive Controls

https://owasp.org/www-project-proactive-controls/
https://owasp.org/www-project-application-security-verification-standard/

# AppSec program: Why and what?

Goal: All software we create and maintain is secure.

AppSec program to improve security posture

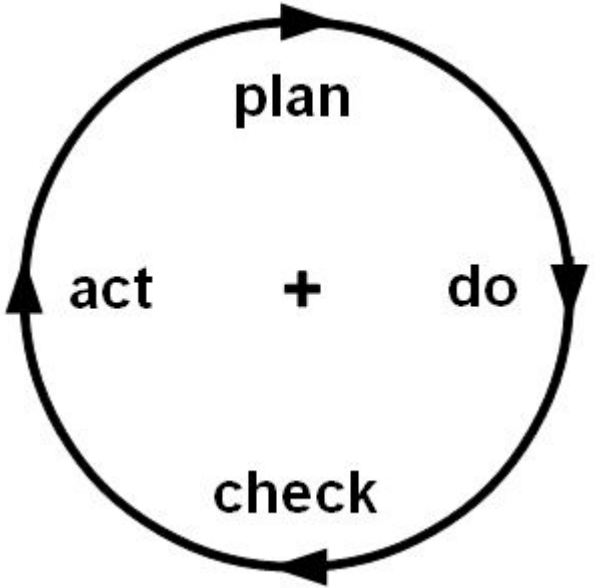
A program formalizes all activities:

- Threat modelling
- Code review of all PRs
- Adding security checks to pipeline

# Remember: Have fun and experiment

Continuous improvement, based on

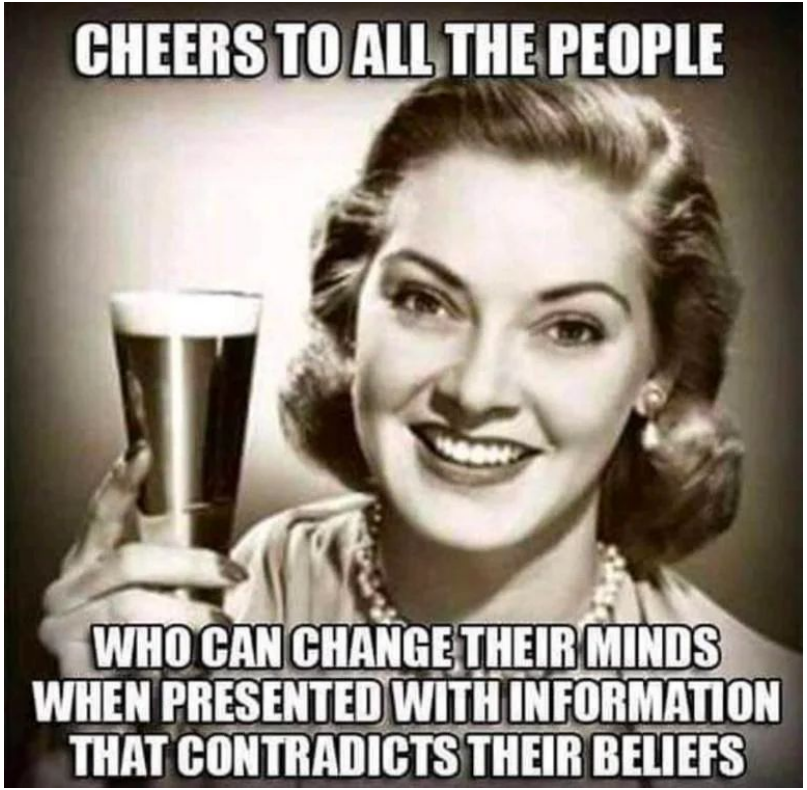
- Metrics
- Experimentation
- Feedback




# People change management...

People not in security tend to think security is a pain
So you always start in minus
So follow a few guidelines:

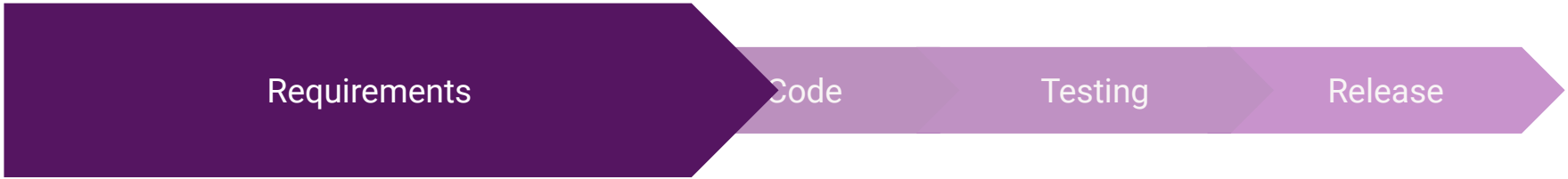
- Has to make sense
- Explain why
- Make it as easy as possible
- Follow existing processes as much as possible



# Mapping AppSec to existing processes

Phases of SDLC

# Requirements




Security Requirements
Security User Stories

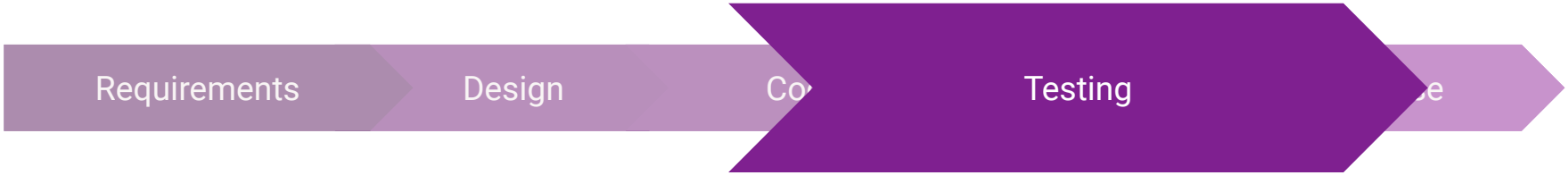
# Design


Threat Modelling
Design Review
Secure Design Principles Applied

# Code


Education
Policies/guidelines
Code Review

# Testing




SAST
Linting
Secrets scanning

DAST
SCA
Security Unit Tests
Penetration Test

# Release

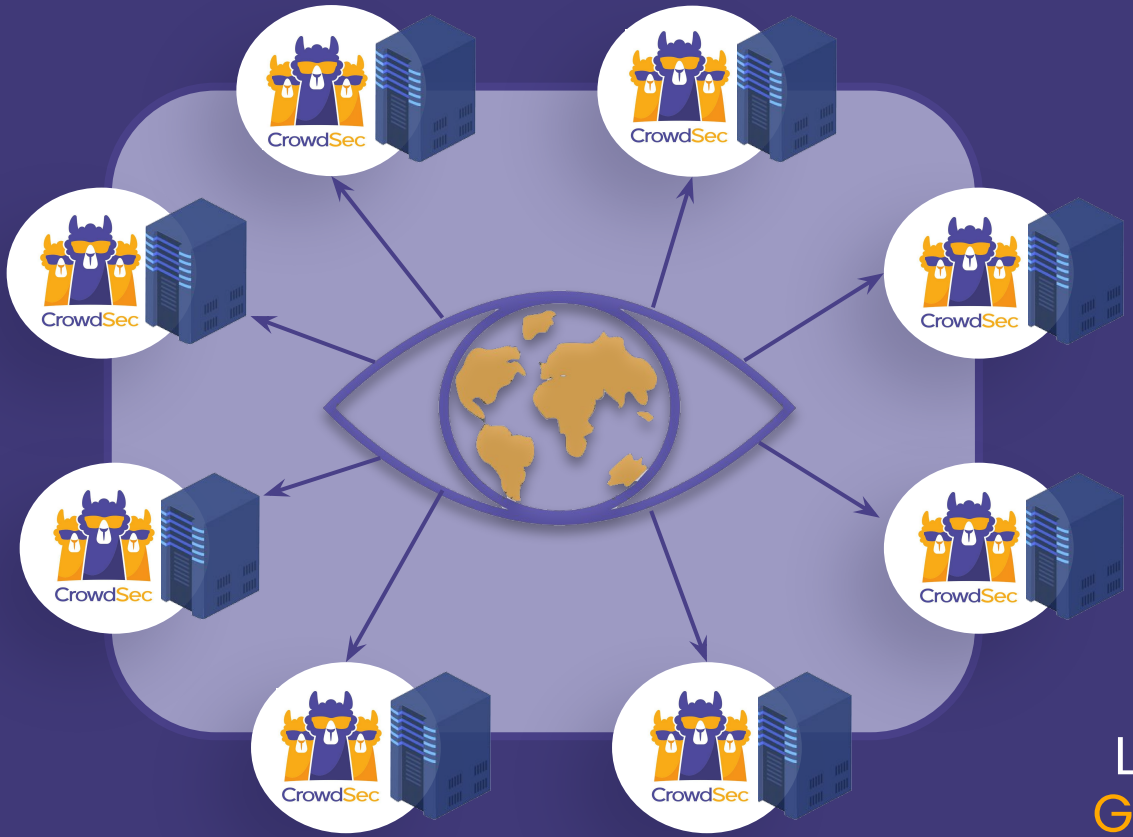

Logging
Monitoring
Alerting
Incident Response

WAF (often also reverse proxy)
RASP

# A few words on CrowdSec

CrowdSec

# Building the *Waze of Cyber Security*




Local IPS
Global CTI

CrowdSec

# Free, forever.

OPEN SOURCE (MIT)

FREE (to use, copy, modify)

MIT license
. Free forever!

Transparent, auditable
and trustable.

We monetize access to
CTI for those not sharing

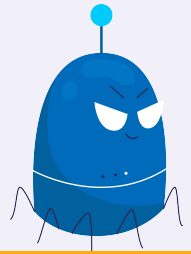
Open to contribution

A fair model: Software against signals..

CrowdSec

# Slowly conquering the world - together

*Already collecting signals from 172 countries.*



CrowdSec

# The massively collaborative IPS

**Behavior**
*(aggressing you)*

**Reputation**

*(aggressing many)*

## 1 > Acquire


Syslog, Journald, AWS Cloudtrail, AWS Kinesis, Docker log, Windows event log, file

## 2 > Detect





## 3 > Remedy



## 4 > Share


*(This process is fully automated)*

CrowdSec

# CrowdSec already deals with


**50+ attacks** & unwanted behaviors...

# CrowdSec and AppSec

**Philosophy:** improve what's already out there







Same could be in theory be done with other WAFs (openappsec.io, OWASP Coraza)

Or literally anything else that generates a log

# Wrap up

- Good security (like orcs and onions) has layers
- Plenty of good reasons to focus on AppSec
- Communicating with stakeholders is hard but important
- APIs shouldn’t be forgotten (the attacker won’t)
- SAMM gives a great overview
- Don’t forget the human side
- Do what you can with automated tools
- FOSS is great!

# Thanks for your attention!

Join our friendly Discord community at https://discord.gg/crowdsec
(we also have workshops)

Follow us on Twitter: @crowd_security

Send me a mail:
klaus@crowdsec.net

Or simply DM me!