ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

September 25 - 27, 2018
Amsterdam, The Netherlands

Security Approaches for Microservice Architectures

-Kameshwara Rao Marthy

What are Microservices

Small ***Autonomous*** services that ***work together***, modelled around a ***business domain***

Microservices, are ***fine-grained***, ***single-function component*** services that can be ***scaled*** and ***deployed independently***, enabling organizations to update or add new features to an application without necessarily affecting the rest of the application's functionality.

Advantages of Microservices

Micro service architectures are becoming extremely important for organizations as they give agility, scalability, flexibility with engineering approach and architectural style of computing in building software.

Smart architectures to Auto-scale the individual components to meet increased demand. When the event is over, sense the drop in traffic, and scale back accordingly. The app is available the entire time, leaving no gap in user experience.

Huge break through as customers expect uninterrupted, seamless digital experiences.

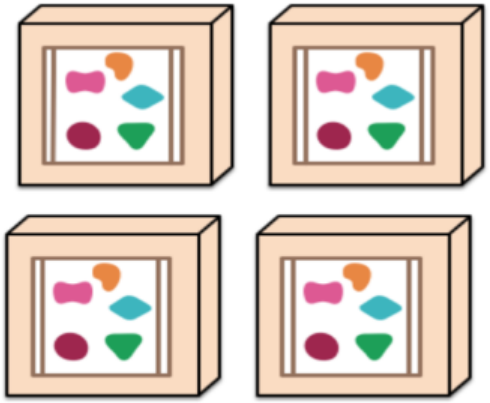
ons

EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate

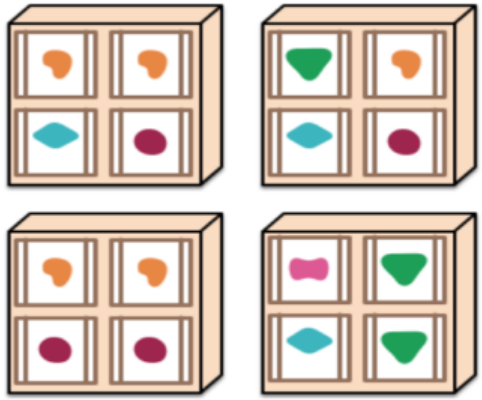
A monolithic application puts all its functionality into a single process...

... and scales by replicating the monolith on multiple servers



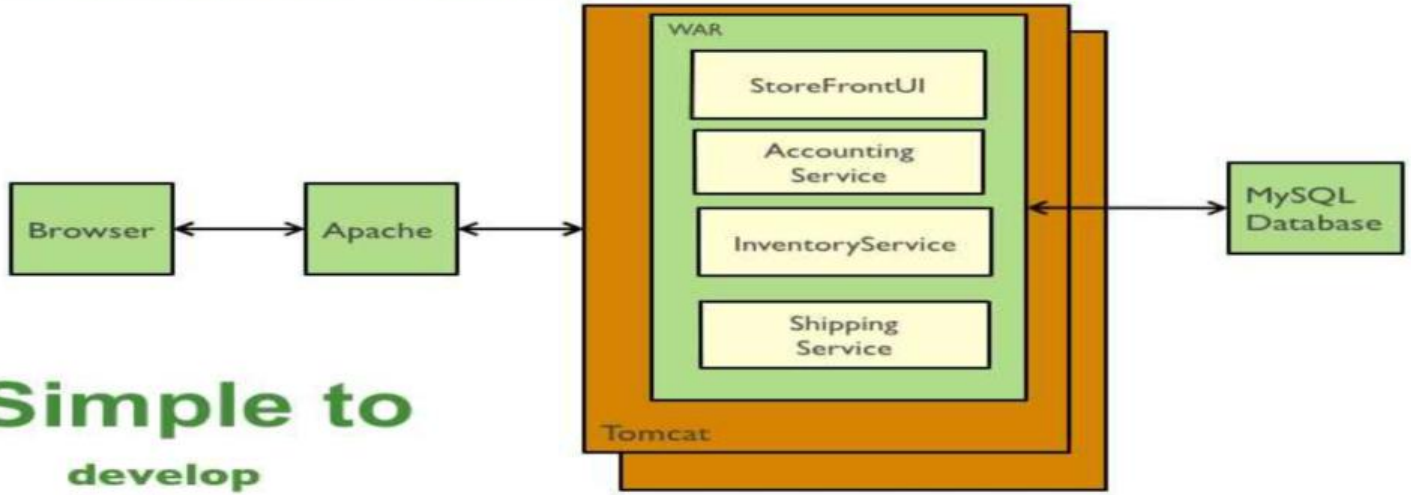
A microservices architecture puts each element of functionality into a separate service...

... and scales by distributing these services across servers, replicating as needed.



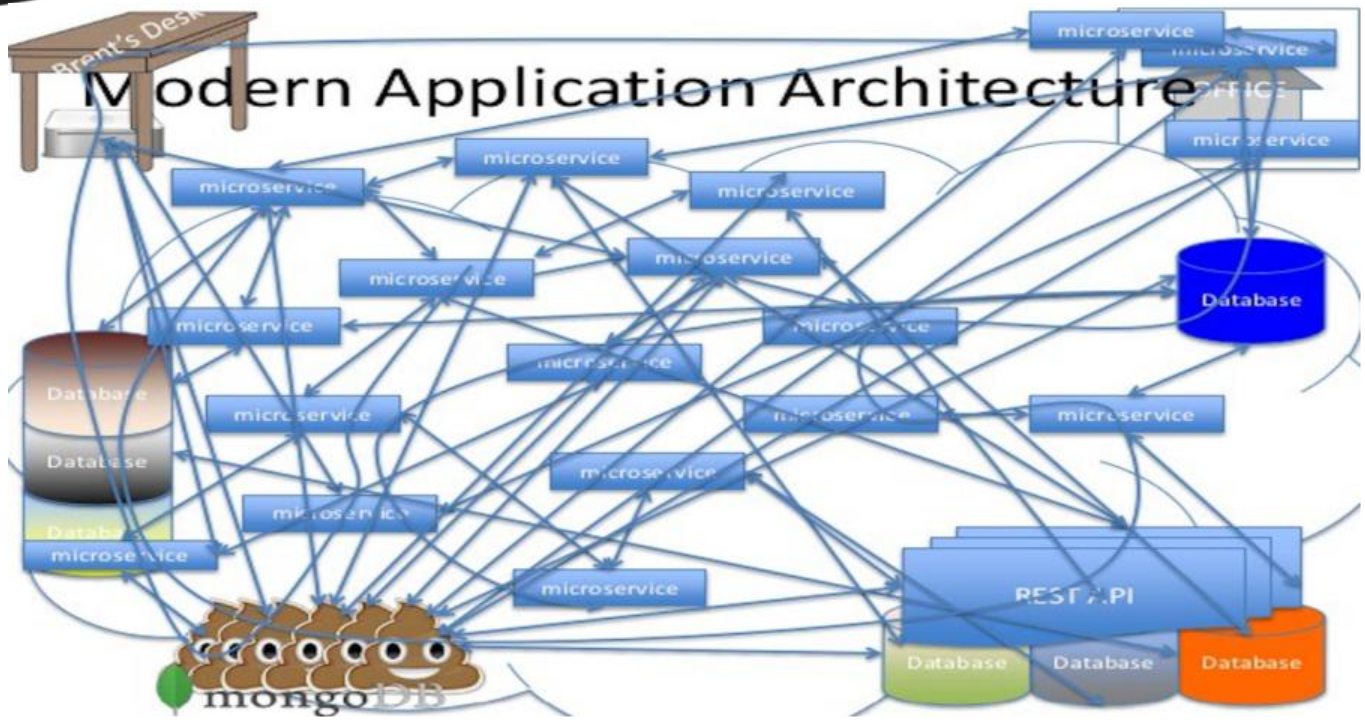
- Wait!! it has so many advantages ? But what are the Hardships??

Traditional web application architecture



Simple to

- develop
- test
- deploy
- scale

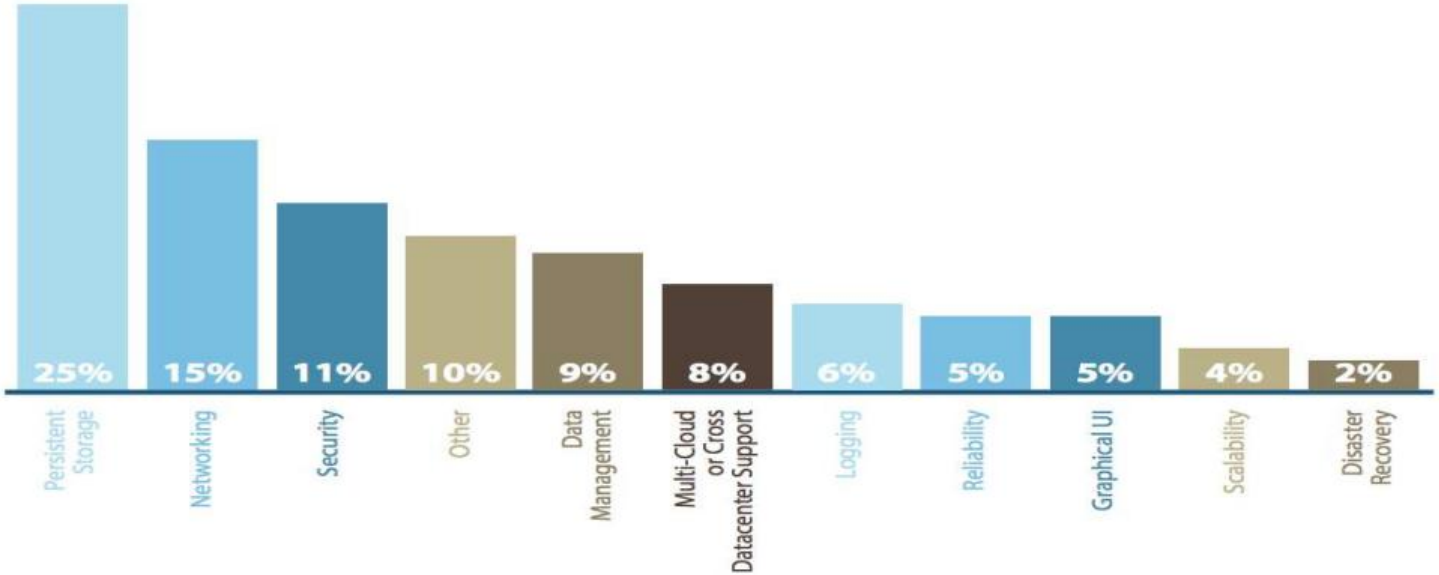

Modern Application Architecture
microservice
Database
REST API
mongoDB

Design for visibility to make inspection and debugging easier.

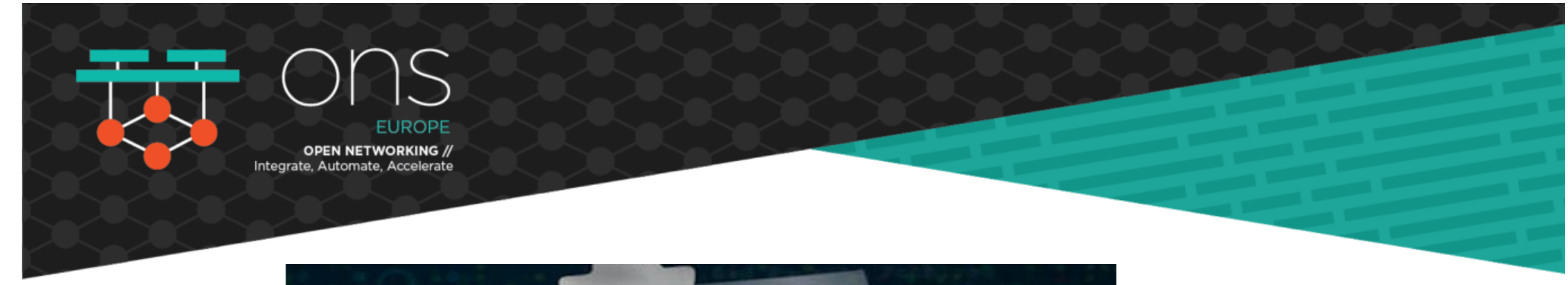
— Basics of the Unix Philosophy

Why security matters?

Microservice Adoption Challenges – Security stands THIRD




ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
YOU HAVE BEEN
HACKED !


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

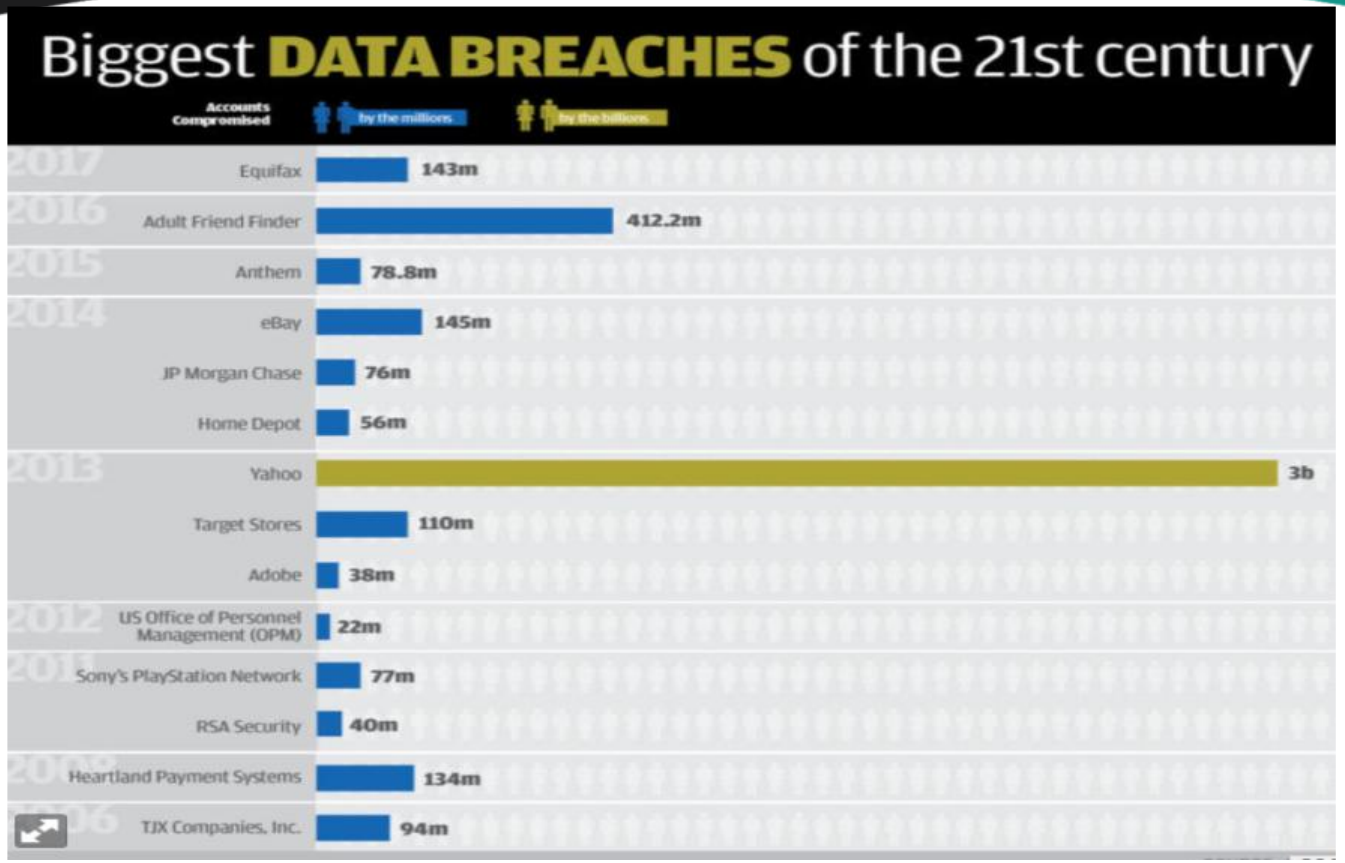

CYBERATTACK
PERSONAL INFORMATION EXPOSED
EQUIFAX
AS MANY AS
143
MILLION
CUSTOMERS

ons

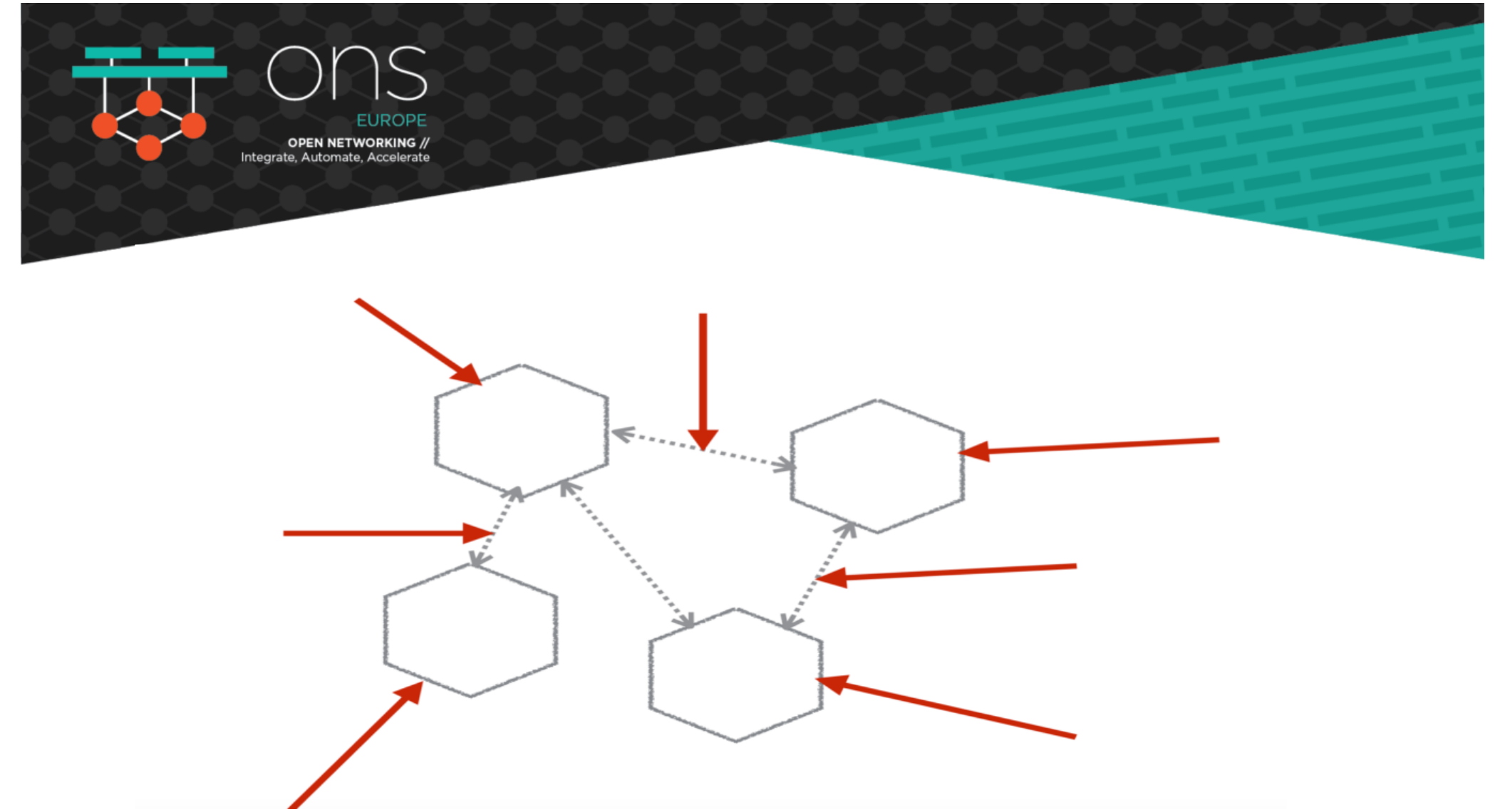
EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate

Biggest DATA BREACHES of the 21st century



- Security is frequently mentioned as the top concern for moving to Microservice architectures.
- Enterprises need to be confident that their data is secure in these architectures.
- Surveys conducted by Forrester Research, the Cloud Native Computing Foundation, and 451 Research revealed that 35-45% of participants reported security as a primary concern regarding running Microservices architectures in production environments


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

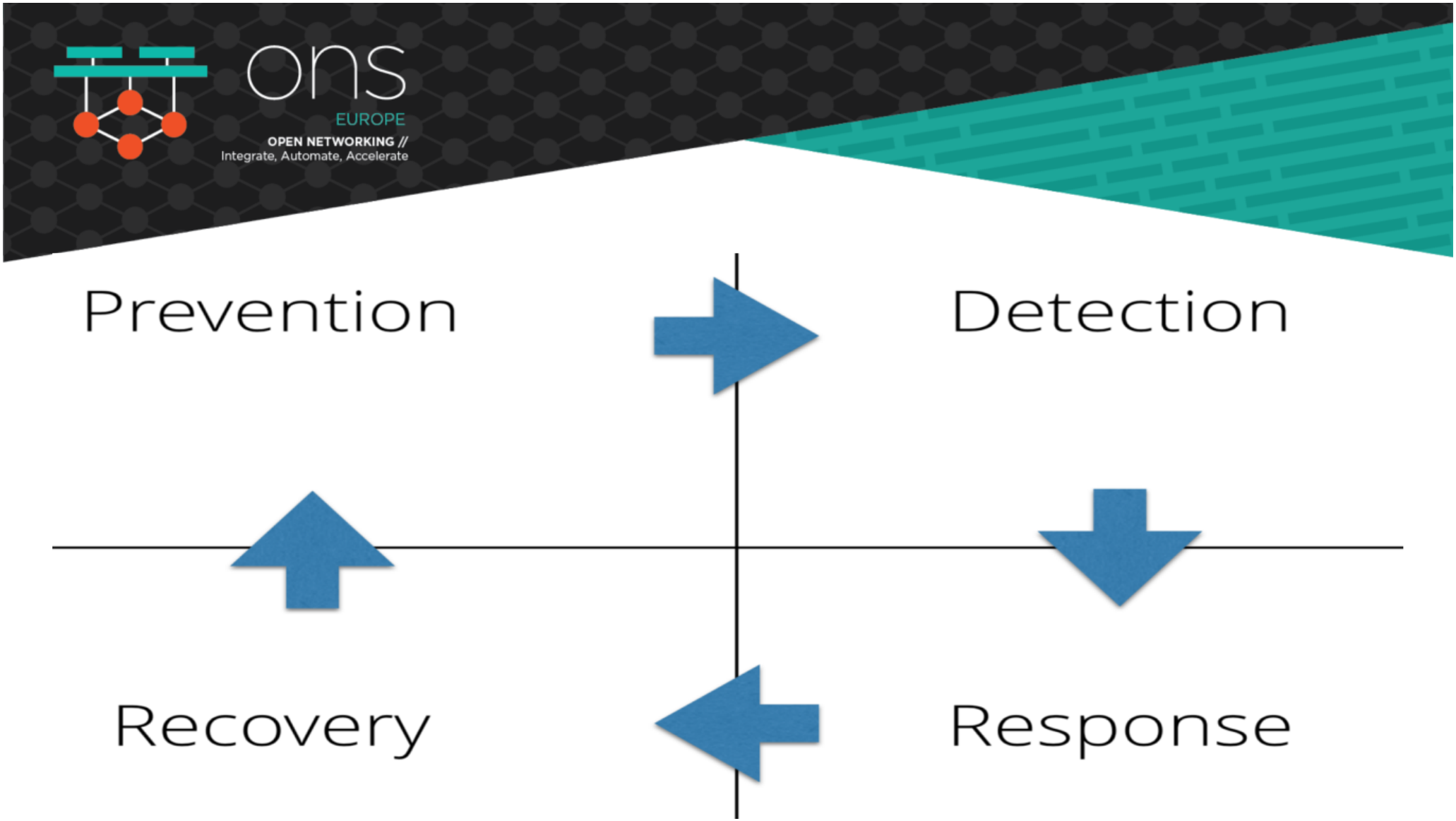
Difference in Security requirements for Monoliths and Micro services ?

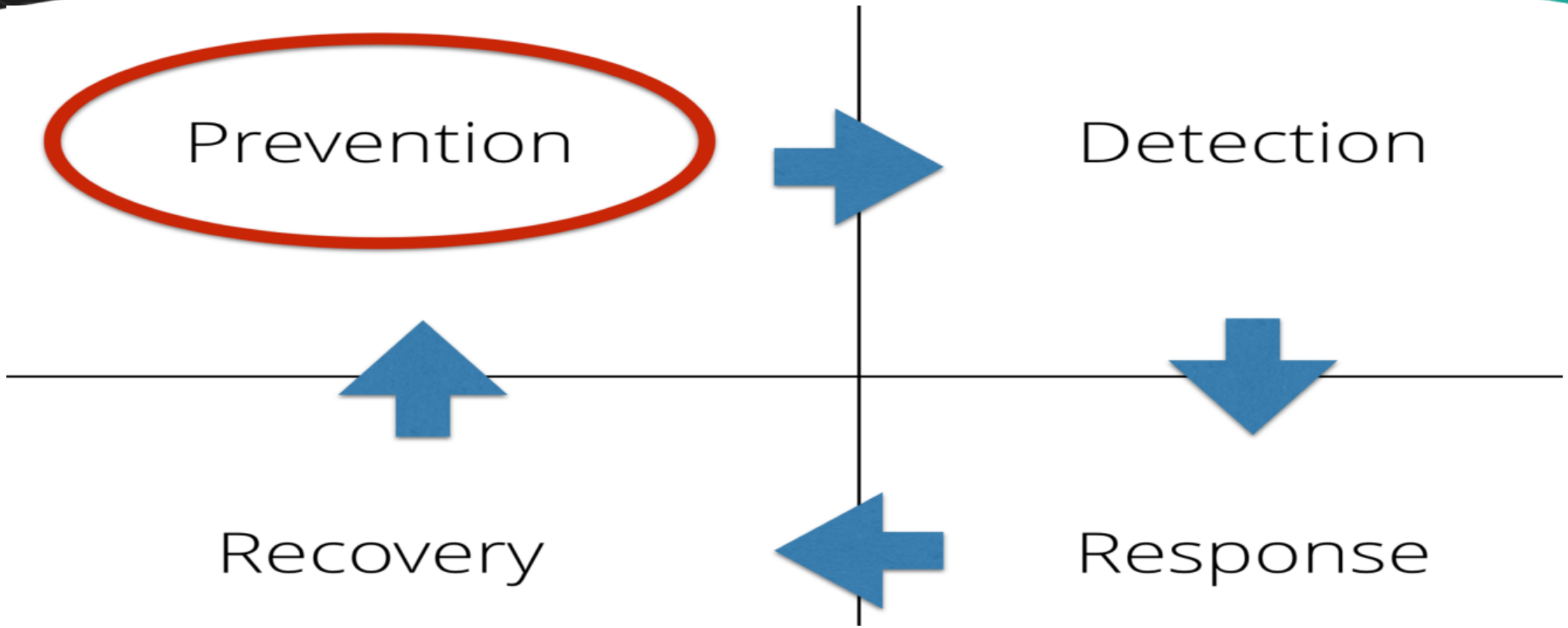
Monoliths have defined boundaries around which we can build our security perimeters

But with Microservices the

- attack surface is Broader
- more processes
- More intercommunication calls
- More Networking requirements.

So how can we secure our Microservices??


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
Prevention
Detection
Recovery
Response


Prevention
Detection
Recovery
Response

Prevention

- Prevention is better than cure. Applies every where!!
- Some times take a step back and think rationally
- Focus more on Securing stuff should be higher priority than investing on monitoring.
- Security aspect should be discussed right from day 1 of the project and not at the end.
- We can’t prevent the attacks but we can significantly reduce the number of attacks if we can Increase the cost of invoking one to hack or break the safe.

Threat Modelling

Process by which potential threats, such as vulnerabilities can be identified, enumerated, and prioritized – all from a hypothetical attacker's point of view.

Threat Modelling - STRIDE

STRIDE is a model of threats developed by ***Praerit Garg*** and ***Loren Kohnfelder*** at Microsoft for identifying security threats. It provides a mnemonic for security threats in six categories. They are:

- **S**poofing of user identity
- **T**ampering
- **R**epudiation
- **I**nformation disclosure (privacy breach or data leak)
- **D**enial of service (D.o.S)
- **E**levation of privilege

https://en.wikipedia.org/wiki/STRIDE_(security)

HTTPS:

- Always use for Data in Transit
- Server guarantees!
- Payload not manipulated...

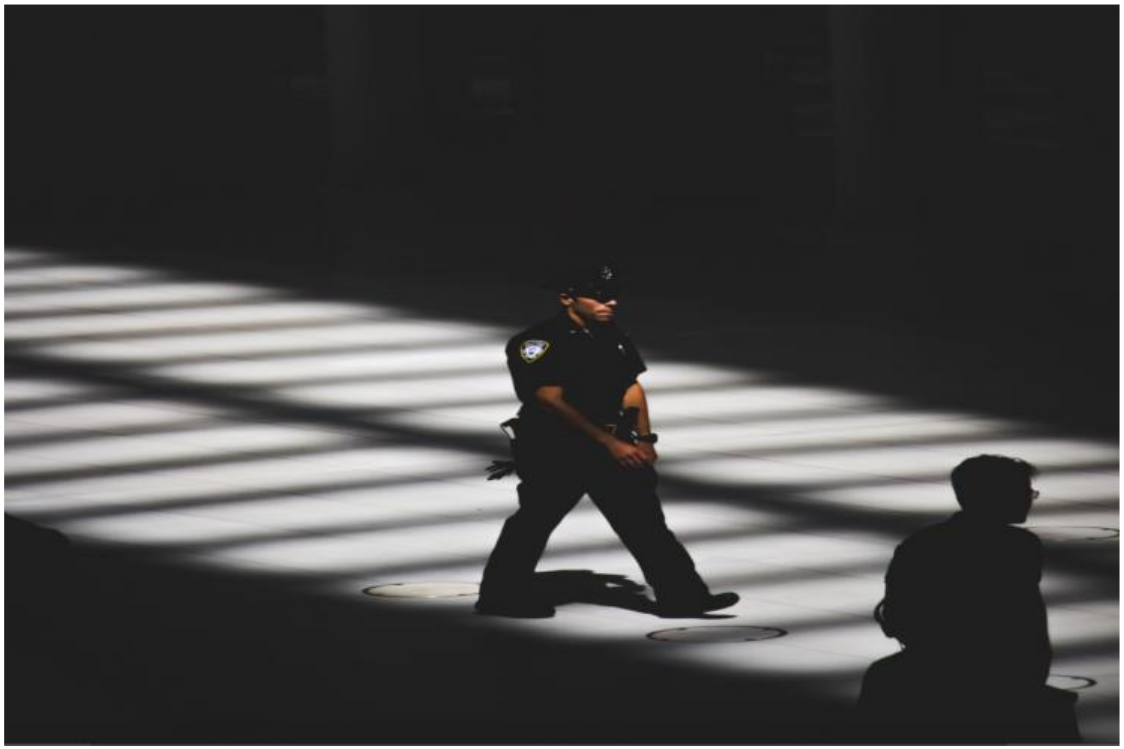
Authentication & Authorization

In microservices, since we don’t generally have centralized user management at every level, its better to adopt to industry standards such as

- Oauth2
- Open Id Connect

ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Implicit trust: Confused Deputy



Confused Deputy problem

A user who has access to the system can begin forging different requests with different identifiers compared to their original assigned identities and roles used when initially signing into the system. This confuses the service into thinking they are someone else or have a different list of roles than what was originally granted during the sign on.

In another scenario, the user originally has access and roles but it is later restricted or revoked from the system. Different data cache mechanisms or leaked keys (such as automated backups) are still able to obtain access to the server side resources.


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Data at Rest?

Database encryption

In Monoliths, if we want to safe guard the data at rest, we used to encrypt the DB tables.

But With microservices, the data is decomposed into different parts and stored at different places.

Eg: user service, payment service and catalog service for a single web application..

Database encryption

we can choose which ones are important to be encrypted and which ones can be left in plain text.. Cost factor is associated..

Also the decryption key has to stored some where. Most of times we end up having the decryption key on the same server. Not a good practice.

Managing Credentials and Secrets:

Consul, Mysql, etcd —> all store things in plain text

Auth with DB/KV store is still an issue..

Auditing and revoking is not present in all the tools.

Using Configuration management systems to store credentials

Chef, Ansible and puppet -> Offers vault solution to store passwords and other secrets.

Problems:
Centralized storage
No API's
Convergence time when there are updates and changes
Auditing and revoking not up to the mark

Probably better than having nothing.

Never keep sensitive data in Code Repositories

Not a good practice to store credentials in GIT.

Can use a tool like ***Git Rob.***

Gitrob is a tool to help find potentially sensitive files pushed to public repositories on Github. Gitrob will clone repositories belonging to a user or organization down to a configurable depth and iterate through the commit history and flag files that match signatures for potentially sensitive files. The findings will be presented through a web interface for easy browsing and analysis.

Using AWS KMS

Full lifecycle management of keys available.

If in AWS, better to use AWS KMS to manage & monitor all the keys getting used in the infrastructure. We can have policies around the key mgmt solution.

Hashicorp Vault

Advantages:

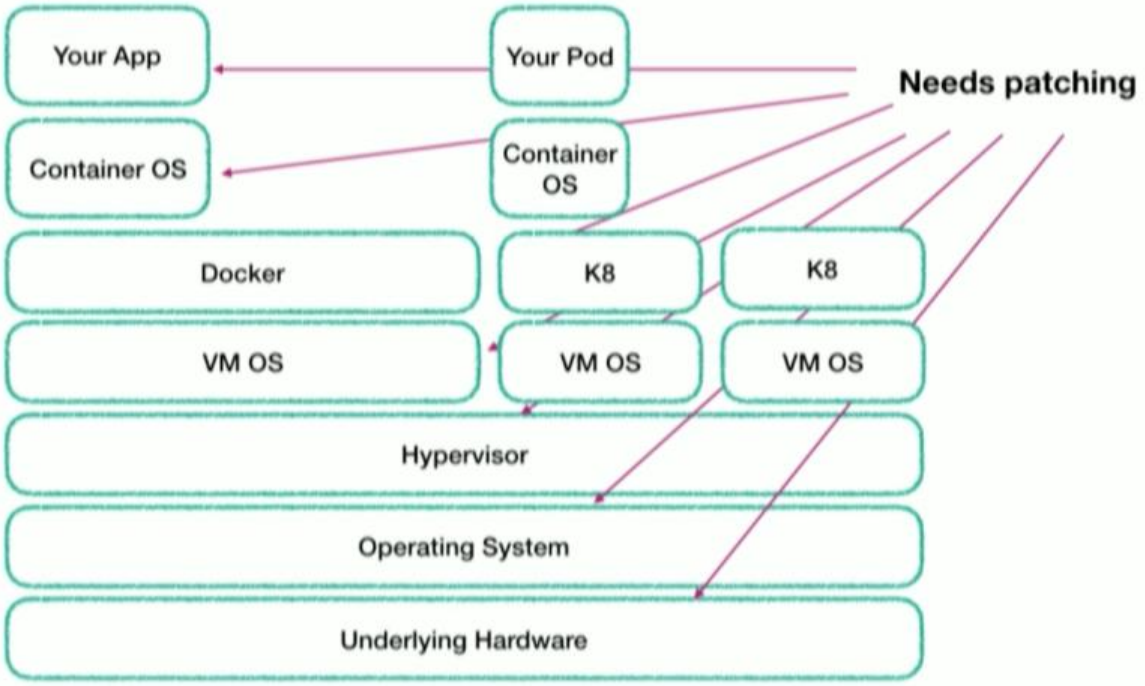
Transit backed. -- Encryption
Time limited tokens
sealed / unsealed state
HTTP API — programmatic access
Dynamic key generation – Generate keys on the fly

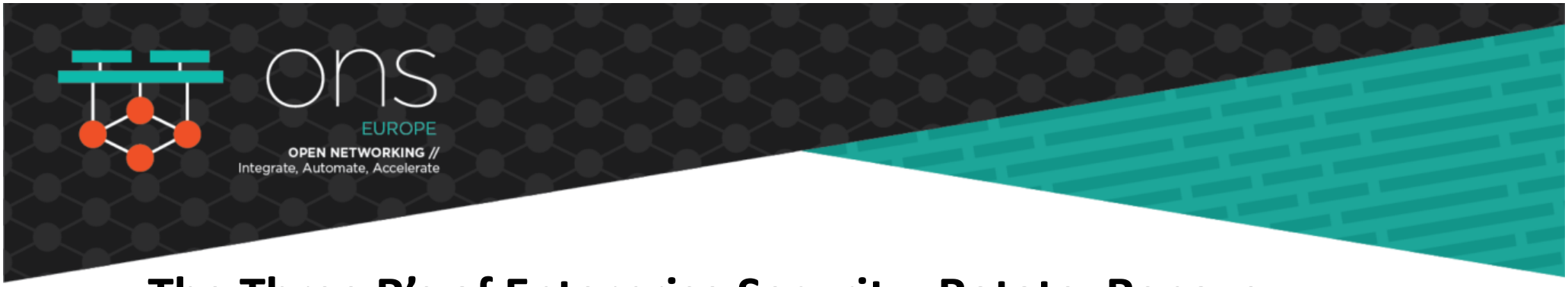
Patching

"44 percent of security breaches occur after vulnerabilities and solutions have been identified. In other words, the problems could have been avoided if found vulnerabilities had been addressed sooner."

- Forbes/BMC, 2016

https://betanews.com/2016/01/12/data-breaches-and-cyber-attacks-are-often-caused-by-failing-to-patch-known-vulnerabilities/

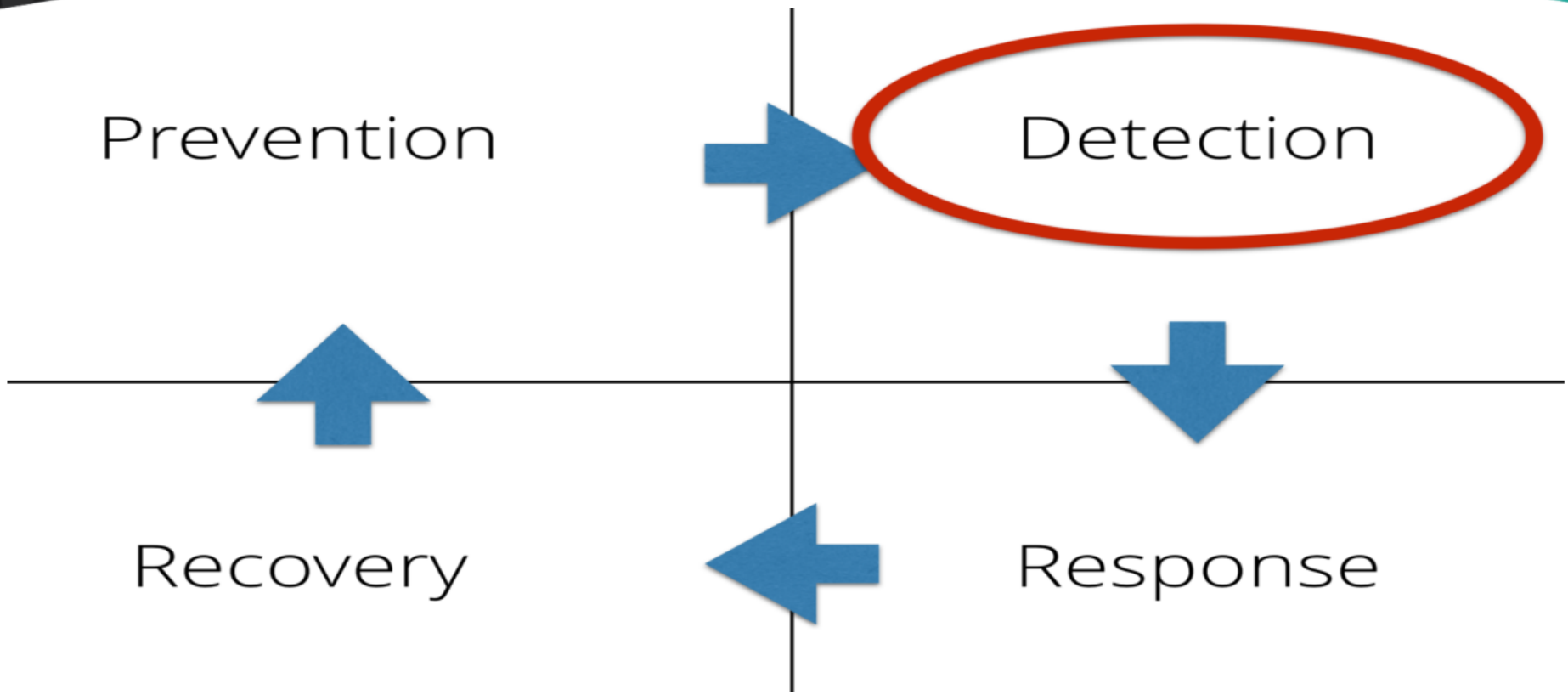

Your App
Your Pod
Needs patching
Container OS
Container OS
Docker
K8
K8
VM OS
VM OS
VM OS
Hypervisor
Operating System
Underlying Hardware



The Three R's of Enterprise Security: Rotate, Repave, and Repair

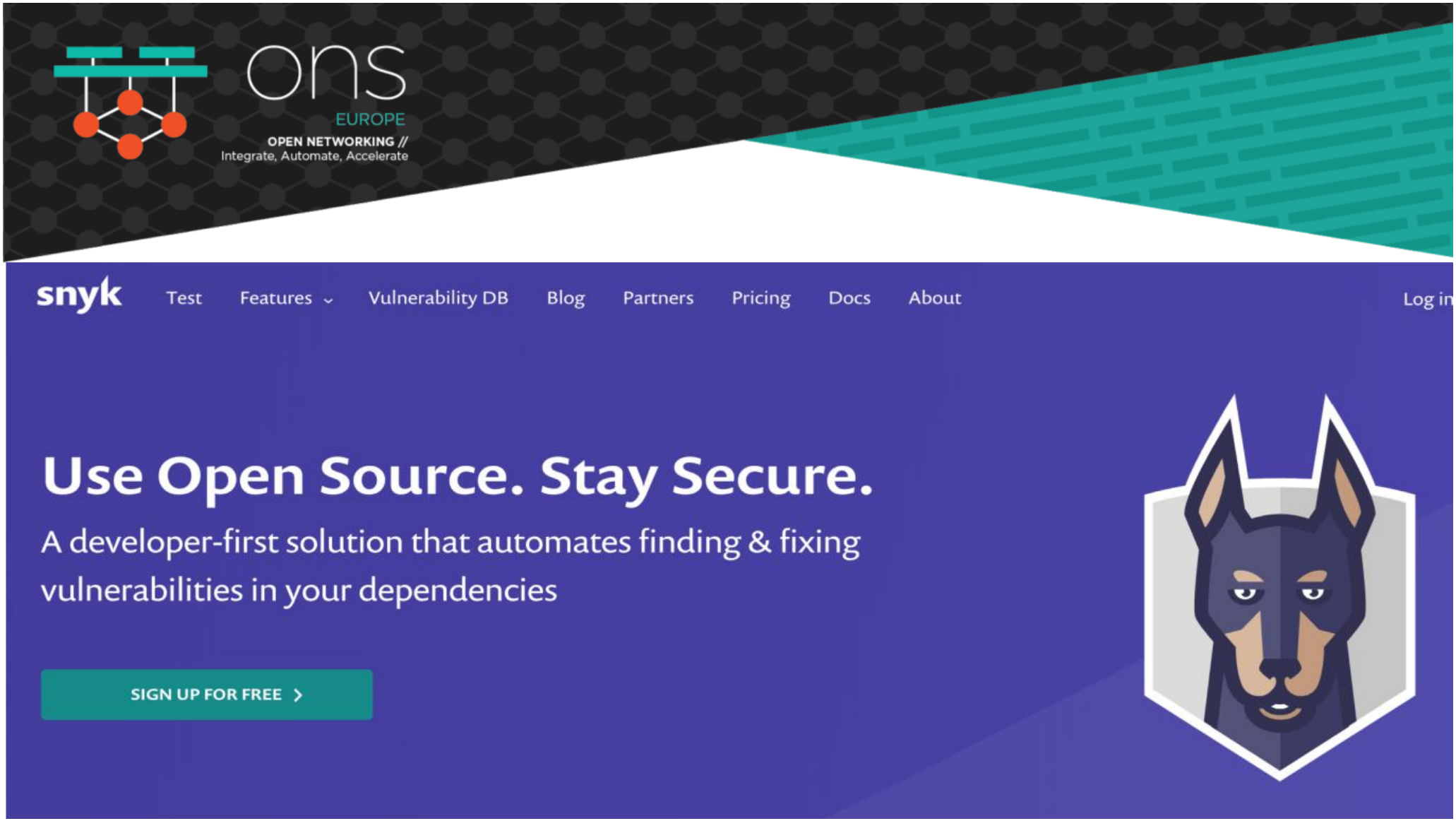
Rotate - Short lived credentials!
Repair - patch your stuff regularly!
Repave - burn the stuff down!

https://builttoadapt.io/the-three-r-s-of-enterprise-security-rotate-repave-and-repair-f64f6d6ba29d

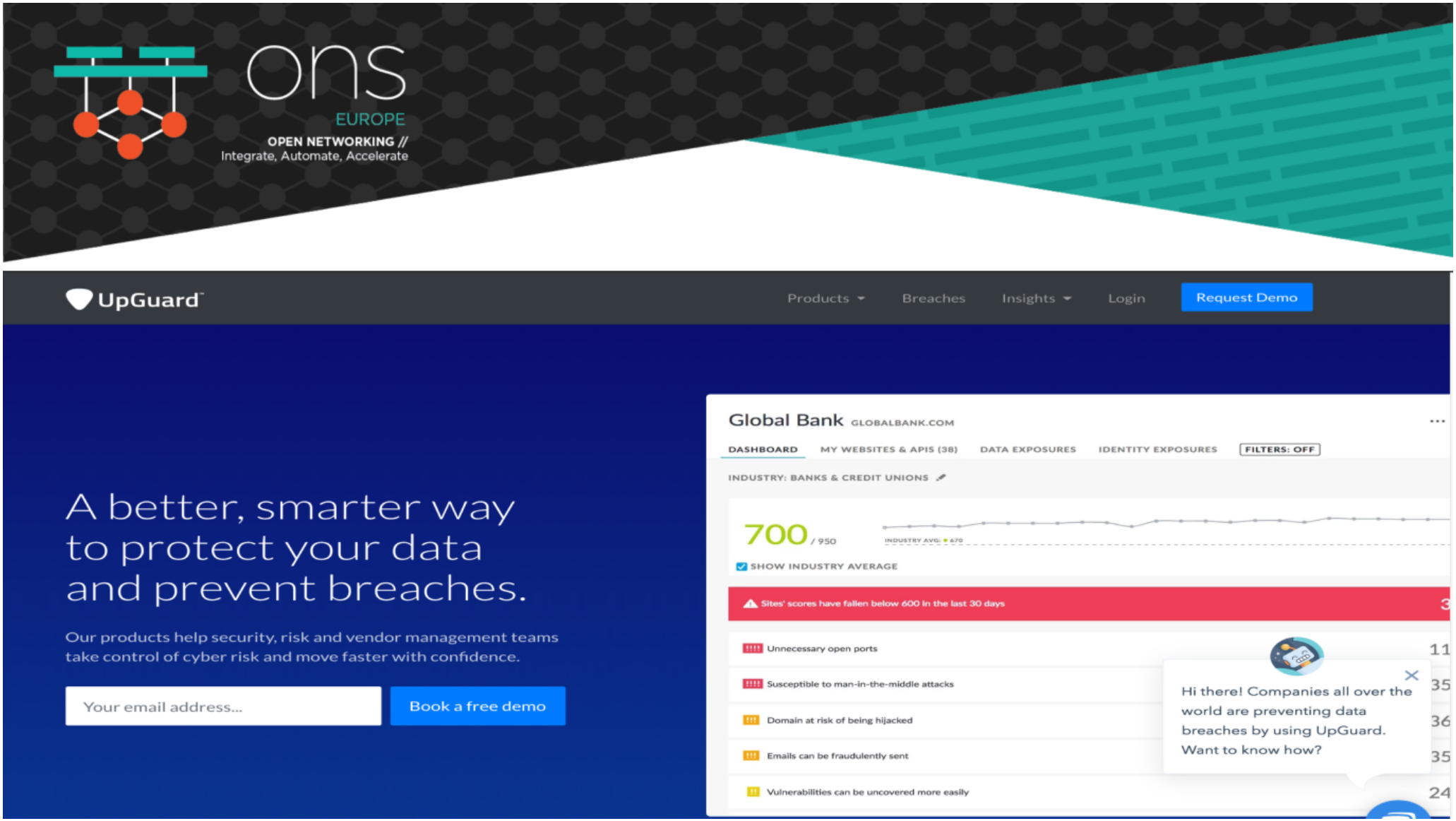

Prevention
Detection
Recovery
Response

Detection

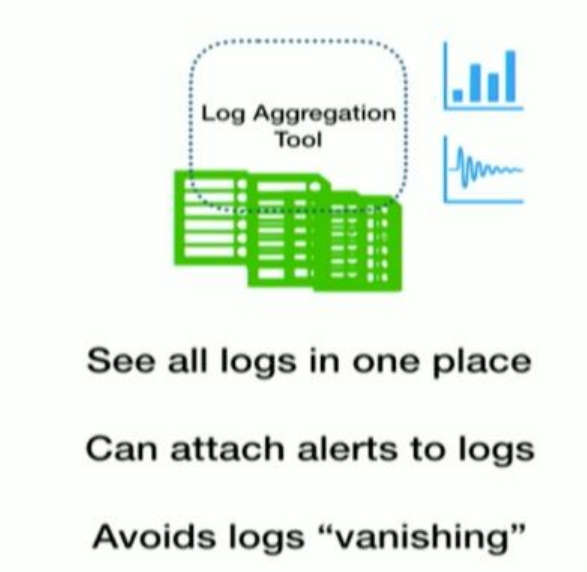
Continuously detect and protect against attacks, anytime, anywhere.


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
snyk
Test
Features
Vulnerability DB
Blog
Partners
Pricing
Docs
About
Log in
Use Open Source. Stay Secure.
A developer-first solution that automates finding & fixing vulnerabilities in your dependencies
SIGN UP FOR FREE >


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
ModSecurity
Open Source Web Application Firewall
Trustwave SpiderLabs
About
Code
Documentation
Demos
Developers
Help
Rules
Status
ModSecurity 3.0
NOW AVAILABLE
Get Code
Source / Binaries
Get Rules
Free / Commercial
Get Help
Support


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
UpGuard
Products
Breaches
Insights
Login
Request Demo
A better, smarter way to protect your data and prevent breaches.
Our products help security, risk and vendor management teams take control of cyber risk and move faster with confidence.
Your email address...
Book a free demo
Global Bank GLOBALBANK.COM
DASHBOARD
MY WEBSITES & APIS (38)
DATA EXPOSURES
IDENTITY EXPOSURES
FILTERS: OFF
INDUSTRY: BANKS & CREDIT UNIONS
700 / 950
SHOW INDUSTRY AVERAGE
Sites' scores have fallen below 600 in the last 30 days
Unnecessary open ports
Susceptible to man-in-the-middle attacks
Domain at risk of being hijacked
Emails can be fraudulently sent
Vulnerabilities can be uncovered more easily
Hi there! Companies all over the world are preventing data breaches by using UpGuard. Want to know how?

Log Aggregation

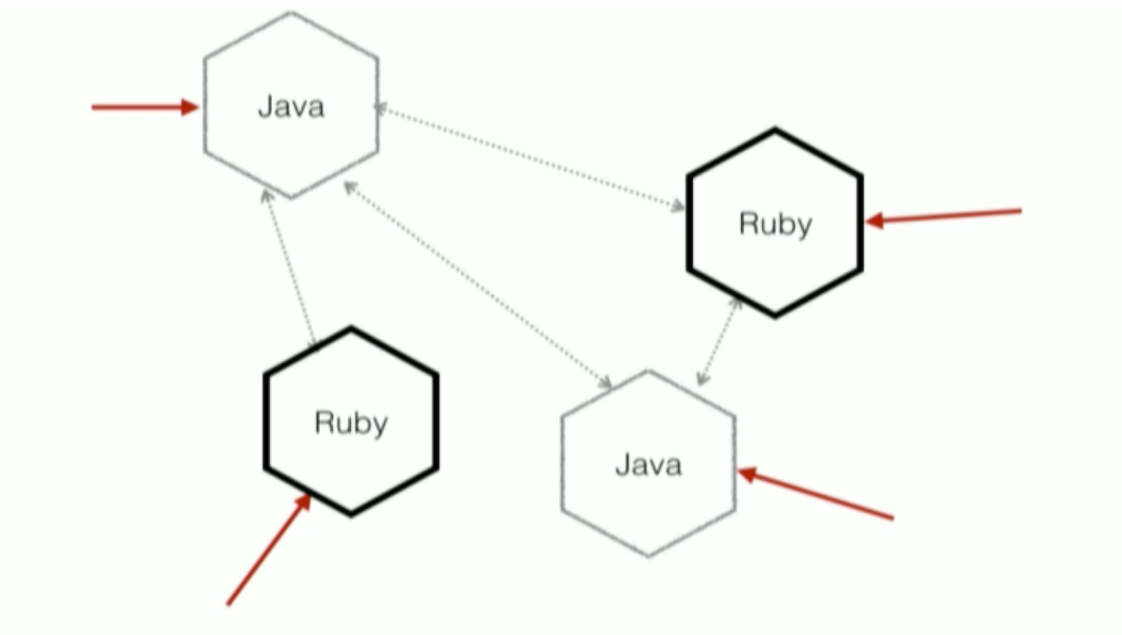


For Enterprises – May be DataDog; For Open Source -May be some thing like ELK stack.

ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Polyglot architecture

More stuff to track

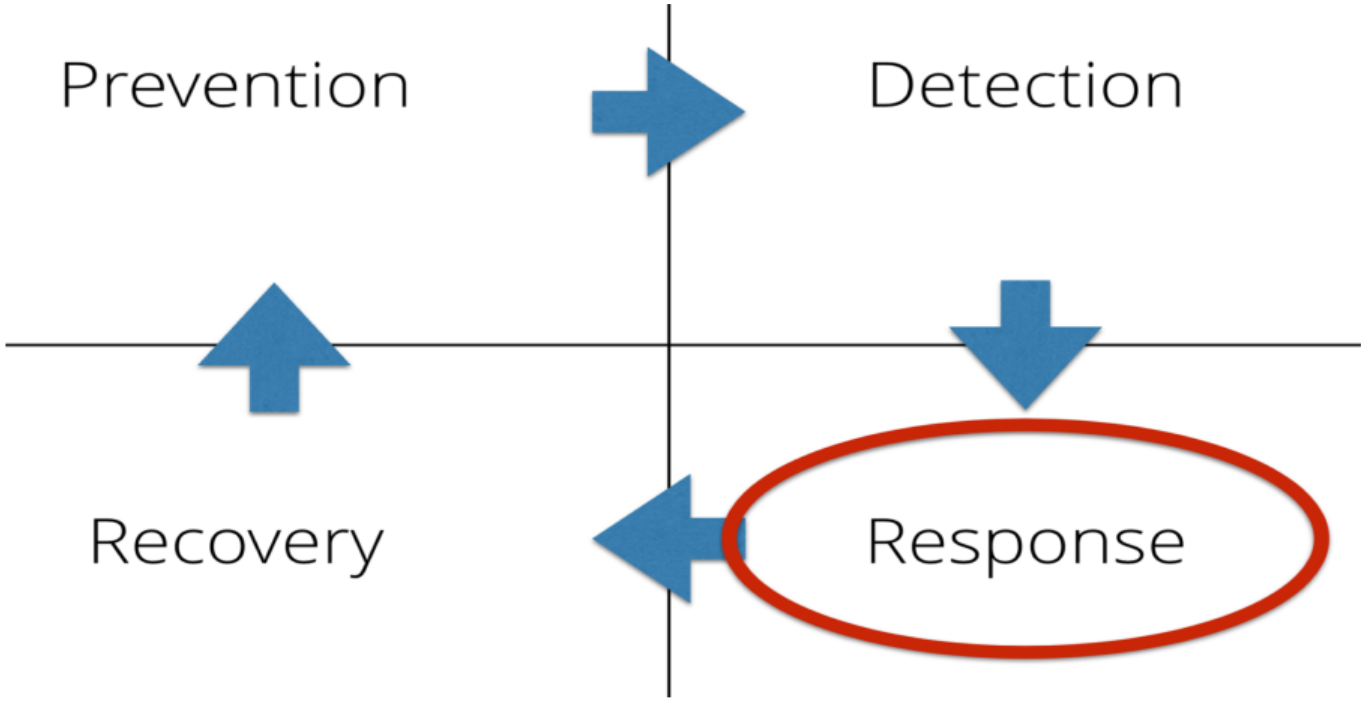


Polyglot architecture

Advantage - one vulnerability cannot break the entire system.

Different languages used to write different services in micro service platforms.
More things to control and more things to be possibly broken

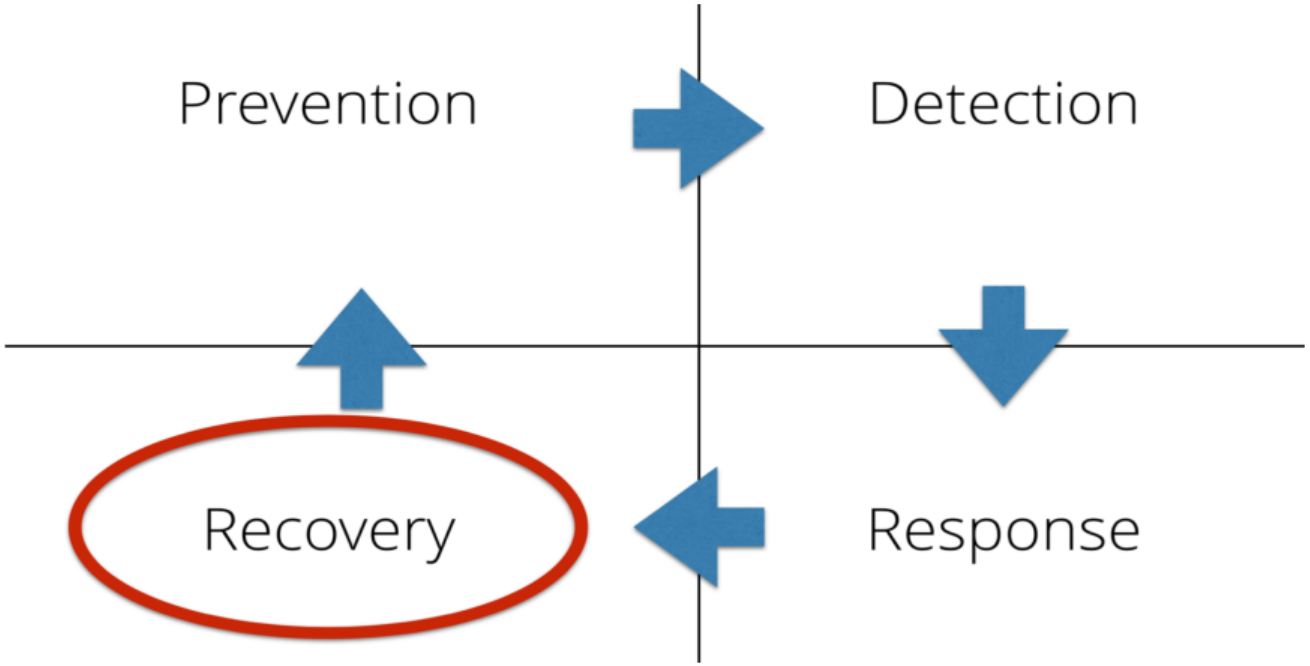
Use tools Snyk.io or like npm check to check for outdated, incorrect, and unused dependencies.


Prevention
Detection
Recovery
Response

DON'T WAIT FOR A DISASTER TO DRAFT AN ACTION PLAN!!!!

Pwned - https://haveibeenpwned.com

- Takes email address and tell if your email address is part of any data breaches.
- Very useful as most of us tend to have same passwords for the email address we use for different accounts.


Prevention
Detection
Recovery
Response

- When you are trying to recover post an attack, its always better to Repave (Burn every thing down!!) .

- Chances of trails of virus/trojans, affected libraries and leftovers in your systems. So its better to start building from scratch.

- *Cost of rebuild is very high!! Particularly if things are not automated completely.*

Automate your infrastructure completely.

Use tools like Chef, Puppet, Ansible, Cloud formation, Terraform etc etc.. You can easily repeat the build process and have audits.

Backups

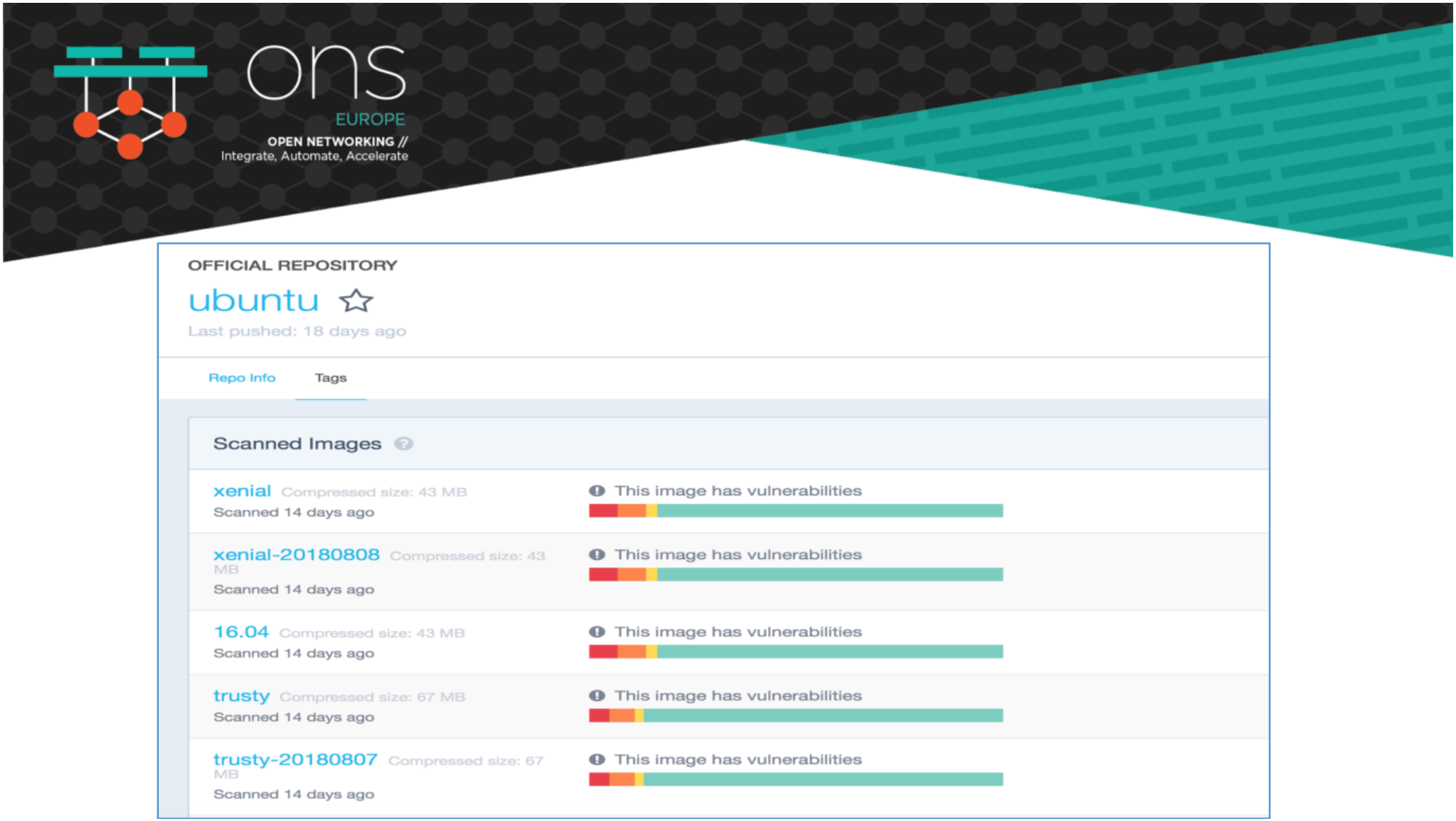
The condition of any backup is unknown until a restore has been attempted.

Docker Security

Over 30% of Official Images in Docker Hub Contain High Priority Security Vulnerabilities

Docker Hub is a central repository for Docker developers to pull and push container images. We performed a detailed study on Docker Hub images to understand how vulnerable they are to security threats. Surprisingly, we found that more than 30% of images in official repositories are highly susceptible to a variety of security attacks (e.g., Shellshock, Heartbleed, Poodle, etc.). For general images – images pushed by docker users, but not explicitly verified by any authority – this number jumps up to ~40% with a sampling error bound of 3%.

https://banyanops.com/blog/analyzing-docker-hub/


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate
OFFICIAL REPOSITORY
ubuntu
Last pushed: 18 days ago
Repo Info
Tags
Scanned Images
xenial Compressed size: 43 MB
Scanned 14 days ago
This image has vulnerabilities
xenial-20180808 Compressed size: 43 MB
Scanned 14 days ago
This image has vulnerabilities
16.04 Compressed size: 43 MB
Scanned 14 days ago
This image has vulnerabilities
trusty Compressed size: 67 MB
Scanned 14 days ago
This image has vulnerabilities
trusty-20180807 Compressed size: 67 MB
Scanned 14 days ago
This image has vulnerabilities

Scan results for **ubuntu:trusty**

19 of 145 components are vulnerable

Scanned 14 days ago

Provide Feedback

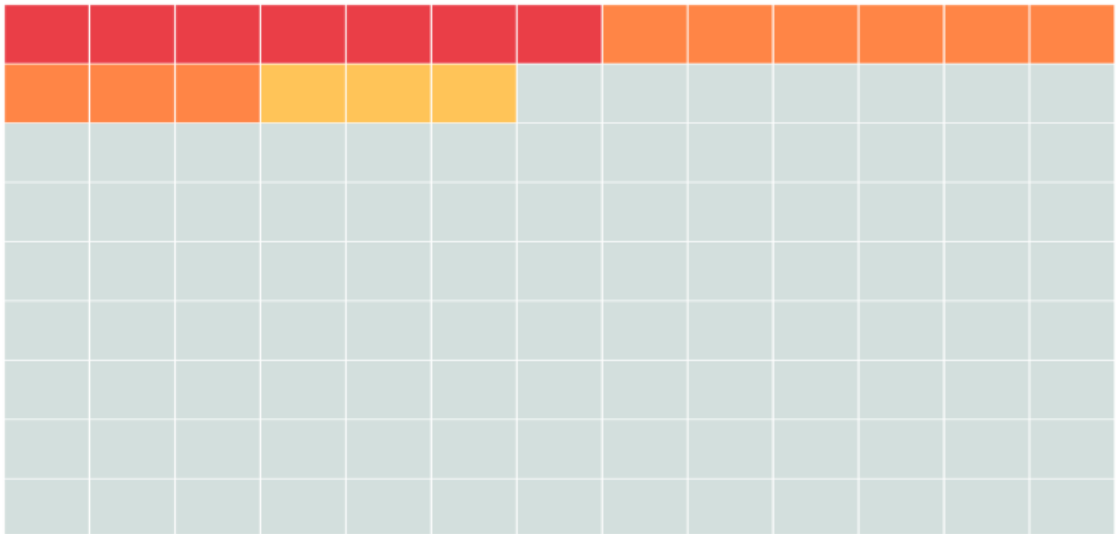
Layers

1 ADD file:b52dc89539ef...bbf609440626583 in /

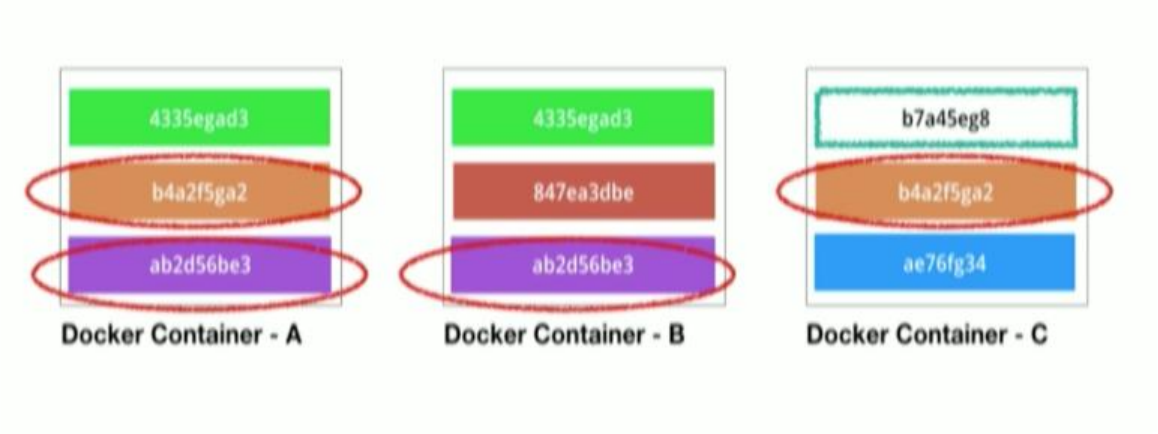
Compressed size: 64.0MB

COMPONENT	VULNERABILITY	SEVERITY
glibc 2.19 LGPL:Lgpl License	CVE-2014-9984	Critical
	CVE-2018-6485	Critical
	CVE-2018-11236	Major
	CVE-2018-11237	Major
	CVE-2017-12132	Major
	CVE-2017-12133	Major
	CVE-2017-15671	Major
	CVE-2016-10228	Major
ncurses 5.9+20140118-1ubuntu1 MIT-like:Permissive License	CVE-2017-10684	Critical
	CVE-2017-10685	Critical
	CVE-2017-16879	Major
	CVE-2017-11113	Major

Components



Layered base scanning



clair

Clair is an open source project for the static analysis of vulnerabilities in application containers (currently including appc and docker).

1. In regular intervals, Clair ingests vulnerability metadata from a configured set of sources and stores it in the database.
2. Clients use the Clair API to index their container images; this creates a list of *features* present in the image and stores them in the database.
3. Clients use the Clair API to query the database for vulnerabilities of a particular image; correlating vulnerabilities and features is done for each request, avoiding the need to rescan images.
4. When updates to vulnerability metadata occur, a notification can be sent to alert systems that a change has occured.

Our goal is to enable a more transparent view of the security of container-based infrastructure. Thus, the project was named `Clair` after the French term which translates to *clear*, *bright*, *transparent*.

ons

EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate

Center for
Internet Security®

CIS Docker 1.13.0 Benchmark

v1.0.0 - 01-19-2017

Docker Bench for Security

- The Docker Bench for Security is a script that checks for dozens of common best-practices around deploying Docker containers in production.

- The tests are all automated, and are inspired by the CIS Docker Community Edition Benchmark v1.1.0. We are releasing this as a follow-up to our Understanding Docker Security and Best Practices blog post.

- We are making this available as an open-source utility so the Docker community can have an easy way to self-assess their hosts and docker containers against this benchmark.

https://github.com/docker/docker-bench-security

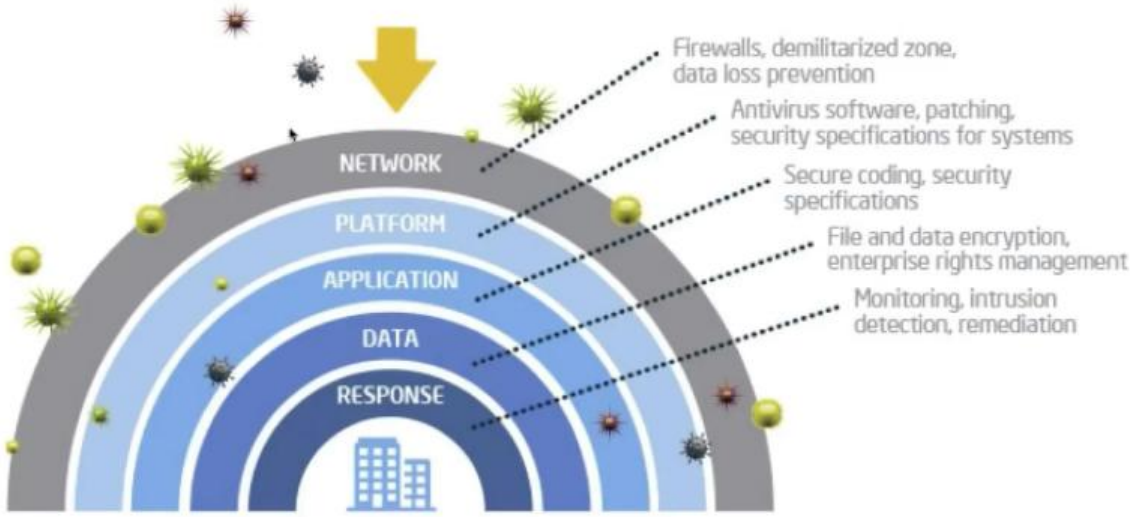
OWASP Zed Attack Proxy (ZAP)

Open Web Application Security Project (OWASP)

The OWASP Zed Attack Proxy (ZAP) is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Security Paradigms

- ***Defense in Depth***


NETWORK
PLATFORM
APPLICATION
DATA
RESPONSE
Firewalls, demilitarized zone, data loss prevention
Antivirus software, patching, security specifications for systems
Secure coding, security specifications
File and data encryption, enterprise rights management
Monitoring, intrusion detection, remediation

Security Paradigms

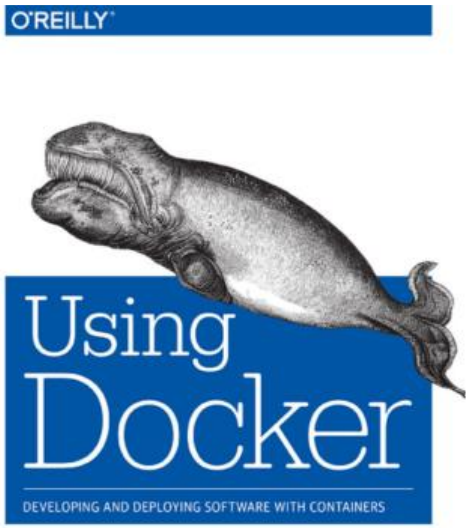
- ***Least Privilege:***

The generic goal of administrators is to hand out the least amount of privileges. The goal of attackers is to gain as much privileges needed to gain access to sensitive information.

ons

EUROPE

OPEN NETWORKING //
Integrate, Automate, Accelerate



Adrian Mouat

Sam Newman


ons
EUROPE
OPEN NETWORKING //
Integrate, Automate, Accelerate

Thank you!