OWASP
Open Web Application Security Project

How To Hack Your Own Apps

A Workshop

Tanya Janca

Tanya.Janca@owasp.org
@shehackspurple
OWASP Ottawa Chapter Leader
OWASP DevSlop Project Leader

Nicole Becher

Nicole.Becher@owasp.org
@thedeadrobots
OWASP Brooklyn Chapter Leader
OWASP DevSlop Project Leader

OWASP
Open Web Application
Security Project

OWASP
Open Web Application
Security Project

How To Hack Your Own Apps

A Workshop

Tanya Janca
Tanya.Janca@owasp.org
@shehackspurple
OWASP Ottawa Chapter Leader
OWASP DevSlop Project Leader

Nicole Becher
Nicole.Becher@owasp.org
@thedeadrobots
OWASP Brooklyn Chapter Leader
OWASP DevSlop Project Leader

OWASP
Open Web Application Security Project

outline

Introductions:
Who are we

OWASP
Open Web Application Security Project

outline

Introductions:
Who are we

Introductions:
Who are we

Goal: Applic
Securit

OWASP
Open Web Application Security Project

OWASP
Open Web Application Security Project

outline

Introductions: Who are we

Goal: Application Security

Introductions: Who are we

Goal: Applic
Securit

OWASP
Open Web Application Security Project

Introductions: Who are we

Goal: Applic Securit

ecure

OWASP
Open Web Application Security Project

outline

Introductions: Who are we

Goal: Application Security

Why Secure Coding?

OWASP
Open Web Application Security Project

outline

Introductions: Who are we

Goal: Applic Securit

ecure

OWASP
Open Web Application Security Project

outline

Introductions: Who are we
Goal: Application Security
Why Secure Coding?
How Will We Secure Code?

6 / 57

Our Toolbag

OWASP
Open Web Application Security Project

outline

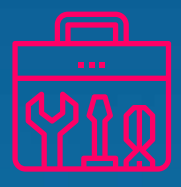
Introductions: Who are we

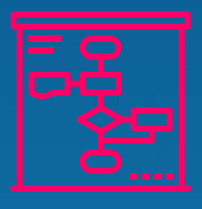
Goal: Applic Securit

ecure

Our Toolbag

Creating Tes





OWASP
Open Web Application Security Project

outline

Introductions: Who are we
Goal: Application Security
Why Secure Coding?
How Will We Secure Code?
Our Toolbag

OWASP
Open Web Application Security Project

outline

Introductions: Who are we

Goal: Applic Securit

ecure

Our Toolbag

Creating Tes

OWASP
Open Web Application Security Project

outline

Introductions: Who are we
Goal: Application Security
Why Secure Coding?
How Will We Secure Code?
Our Toolbag
Creating Test Plans

OWASP
Open Web Application Security Project

outline

Introductions: Who are we

Goal: Applic Securit

ecure

Our Toolbag

Creating Tes

P

OWASP
Open Web Application Security Project

outline

Introductions: Who are we
Goal: Application Security
Why Secure Coding?
How Will We Secure Code?
Our Toolbag
Creating Test Plans
Disclaimer

Who are we

- OWASP

Nicole Becher

pentester/forensics/infosec/c
faculty @nyuniversity, fellow
@newamcyber, political junk
marathoner, martial artist,
conservationist & animal lov

Tanya Janca

DevSlop Project Team

- Two AppSec/Developer/Pen
- Collaborating to make open

)WASP
/a
:r.
ne my
vorld.

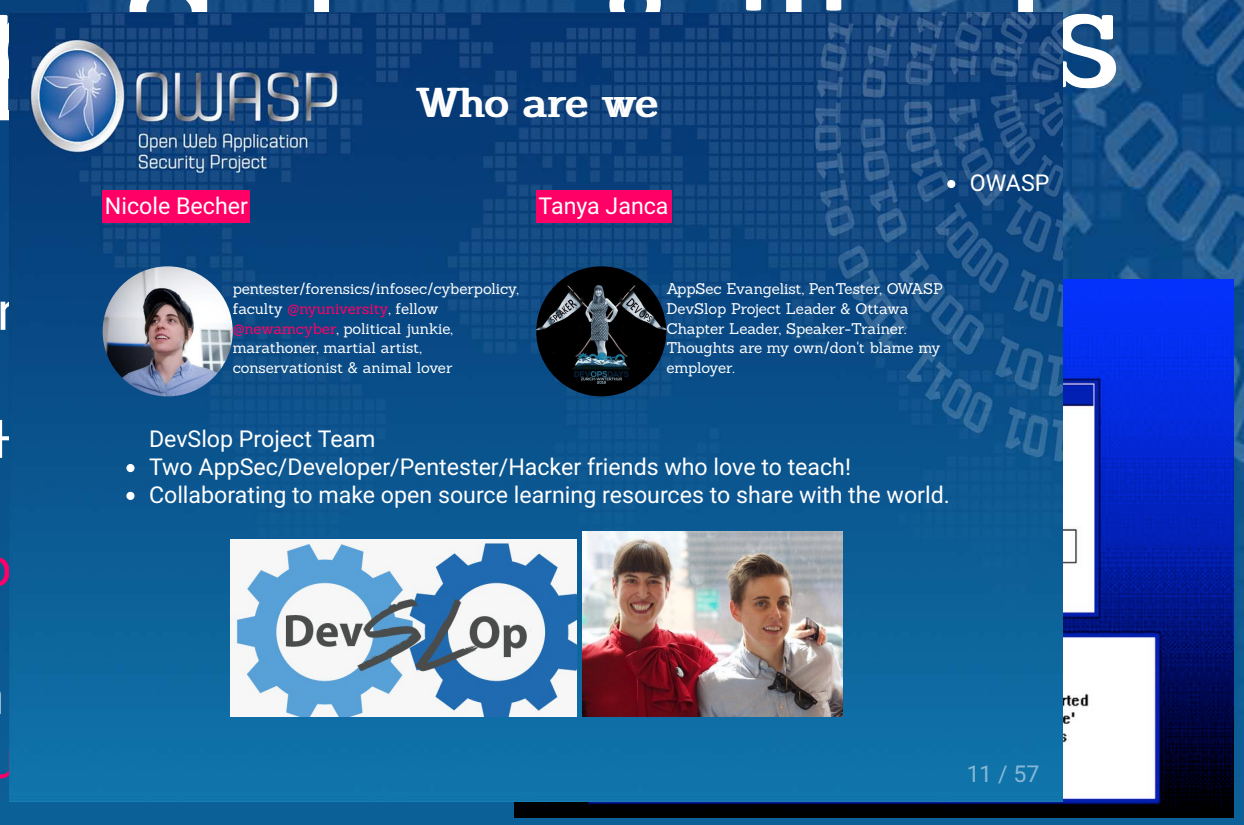
OWASP
Open Web Application Security Project

outline

Introductions: Who are we | Goal: Application Security | Why Secure Coding? | How Will We Secure Code? | Our Toolbag | Creating Test Plans | Disclaimer | WORKSHOP

Workshop Setup & Tools

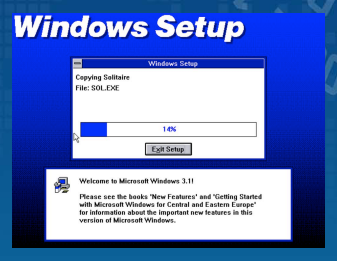
- Laptop with administrator pr
  ability to install software
- Preferably not Windows 10 H
  Hyper-V, needed for Docker)
- Modern Web Browser, Firefo
  maybe even Edge!
- Wifi Connection (Hotels can
- Ability to run a VM with Virtu
- Docker installed
- If want you can pair up, make friends



Goal: Teach Beginner Application Security

Workshop Setup & Tools

- Laptop with administrator privileges and ability to install software
- Preferably not Windows 10 Home (has no Hyper-V, needed for Docker)
- Modern Web Browser, Firefox, Chrome, Safari, maybe even Edge!
- Wifi Connection (Hotels can be weird.)
- Ability to run a VM with Virtual Box/VMware
- Docker installed
- If want you can pair up, make friends



Write

To give developers the power to write the highest quality code possible

Te

To save de time by allowing them to test for and fix vulnerabilities early in the SDLC

vulnerabilities fixed before the QA or security team even sees the application

at

y apps a chance for security attention.

Why Write Secure Code

Catch Early

Everyone knows that the earlier you fix a bug the cheaper and easier it is to do

Ensu CI

To maint confider integrity and availability of the information in your apps

ble t t

someone break in via your app would be embarrassing to you and your employer.

Because a poorly written application is a window directly into your data for a malicious actor.

OWASP
Open Web Application Security Project

Goal: Teach Beginner Application Security

Write	Test	Fix	Repeat
To give developers the power to write the highest quality code possible	To save developers time by allowing them to test for and fix vulnerabilities early in the SDLC	To have all common vulnerabilities fixed before the QA or security team even sees the application	To give legacy apps a chance for security attention.

13 / 57

OWASP
Open Web Application Security Project

Where To Start?

Proxy & Test

Scanning Your

ement

Today we will learn the basics of scanning a web app, so you can find your security own bugs.

Hopefully, decide that to test your own code with what you learn today, before you send it to QA.

CI/CD pipeline, if you have one.

e to be reative, you will get better at both coding and testing.

OWASP
Open Web Application Security Project

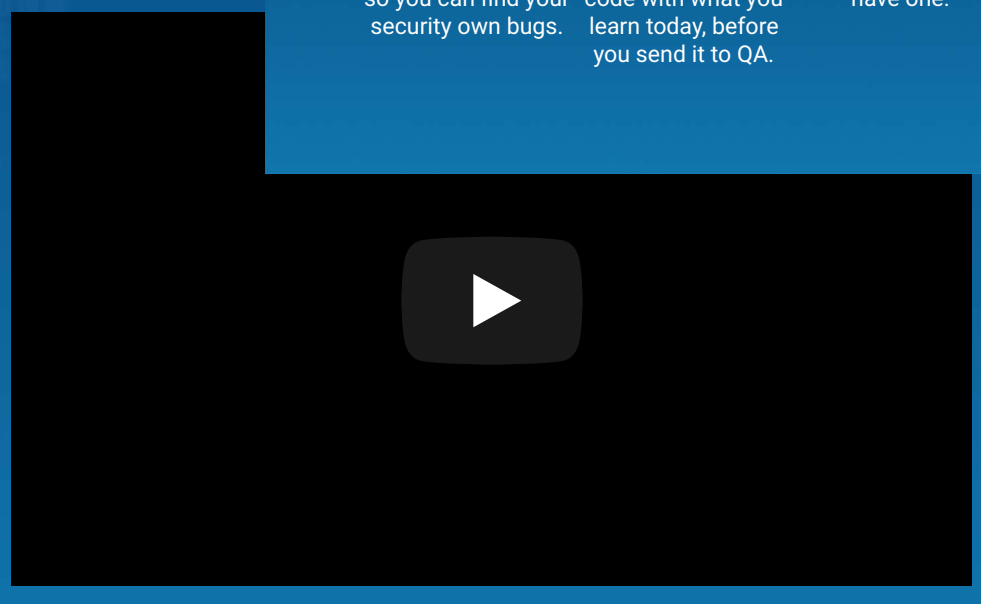
Why Write Secure Code

Catch Early	Ensure CIA	Avoid a Situation	Possible Pivot Point
Everyone knows that the earlier you fix a bug the cheaper and easier it is to do	To maintain the confidentiality, integrity and availability of the information in your apps	Security issues are a big deal, having someone break in via your app would be embarrassing to you and your employer.	Because a poorly written application is a window directly into your data for a malicious actor.

OWASP
Open Web Application Security Project

Learning Web Proxies: OWASP ZAP

- OWASP Zed Attack Proxy (Z
- Easy to use, built for beginne
- Built by OWASP, an internatic leaders in security
- Zap can become an automa your build server
- This session covers beginne
- Other great proxies Burp (no



Disclaimer

- OWASP Zed Attack Proxy (Z
and can cause serious dama
- Never use Zap to attack web
- This tool and this lesson are
not to help you misuse this t
- You always need permissio
- Using Zap and any other hac
application can have very se

OWASP
Open Web Application Security Project

Testing Plan

1. Interception

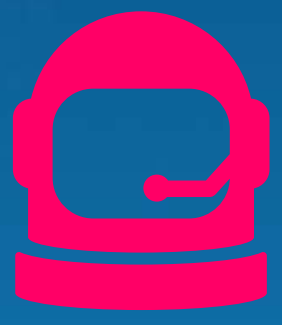
Set up your Proxy

2. Target

ate

RL

4. Manual

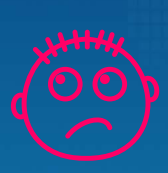


3. Active

OWASP
Open Web Application Security Project

Disclaimer

- OWASP Zed Attack Proxy (Zap) is a testing/hacking tool that can be misused and can cause serious damage.
- Never use Zap to attack websites unless you have consent to test them.
- This tool and this lesson are to help you create better code that is more secure, not to help you misuse this tool.
- You always need permission.
- Using Zap and any other hacking tools on anything besides your own application can have very severe consequences, both legally and professionally.



Demo 1
Learning Outcomes

1. Tools

Set up of Proxy/work space

OWASP
Open Web Application Security Project
Testing Plan
1. Interception
2. Target
3. Enumerate
Set up your Proxy
Set the target URL
Spider the URL
4. Manual
5. Active
18 / 57

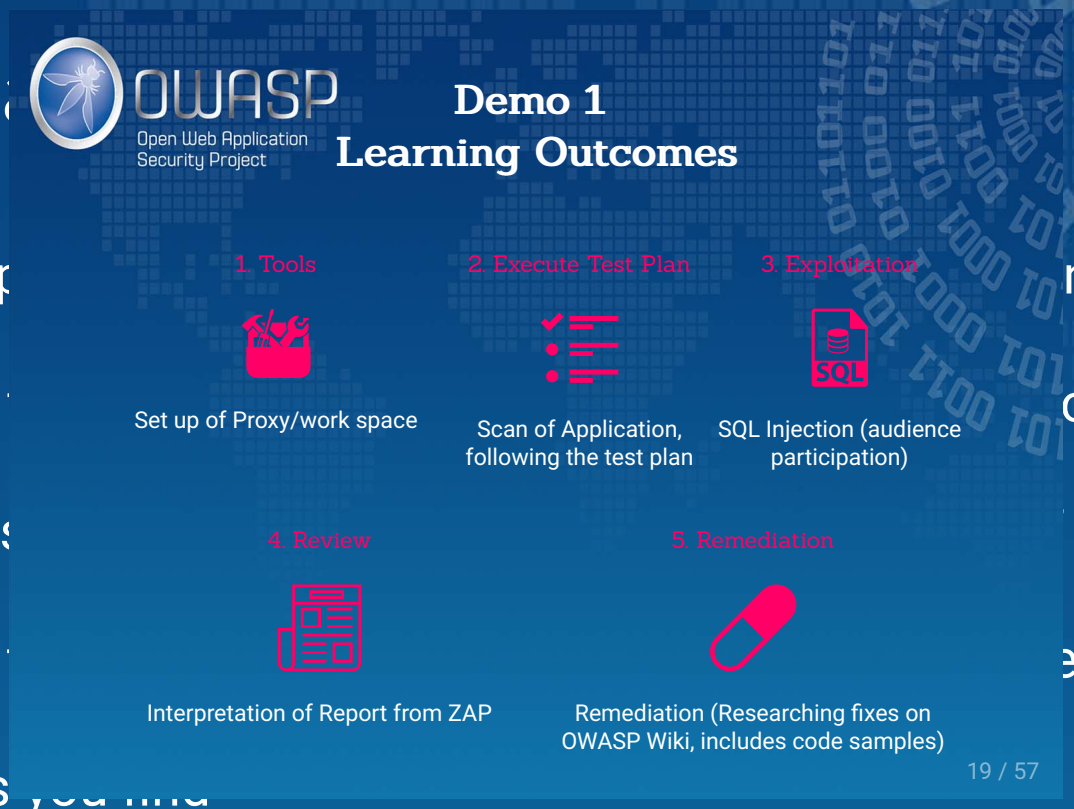
4. Review

Interpretation of Report from ZAP

Remediation (Researching fixes on OWASP Wiki, includes code samples)

You are not yet

- Zap is not full-proof, like all p
  perfect.
- The point of this exercise is
  your code/coding
- Professional Penetration Tes
  assurance
- Use the resources available
  better coder
- Please **actually** fix the bugs you find



OWASP
Open Web Application Security Project

Workshop

- Install Zap
- Load up VMs/Containers
- Set up of Proxy/work space
- Scan of Application, followir
- Manual explore of the web a
- High Fives for XSS!

OWASP
Open Web Application Security Project

You are not yet a hacker

- Zap is not full-proof, like all point-and-click software, it misses things, it's not perfect.
- The point of this exercise is to catch all the low-hanging fruit, and to improve your code/coding
- Professional Penetration Testers will still be required, for a higher level of assurance
- Use the resources available to you (Zap + OWASP Wiki) to become an even better coder
- Please actually fix the bugs you find

OWASP
Open Web Application Security Project

Break

OWASP
Open Web Application Security Project

Workshop

- Install Zap
- Load up VMs/Containers
- Set up of Proxy/work space
- Scan of Application, following the test plan
- Manual explore of the web application
- High Fives for XSS!

Part 2: API and Web App Hacking with

Break


OWASP
Open Web Application
Security Project


OWASP
Open Web Application
Security Project
Part 2: API and Web App Hacking with
pixi
pixi
share the world
DevSlOp
23 / 57

OWASP
Open Web Application Security Project

OWASP
Open Web Application Security Project

New Technology

motivation

The web is changing

OWASP
Open Web Application
Security Project

mo

OWASP
Open Web Application
Security Project

motivation

New Technology

New Technology

The web is changing

The web is changing

New everything!

OWASP
Open Web Application
Security Project

mo

OWASP
Open Web Application
Security Project

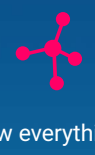
motivation

New Technology

Dev

New Technology

Dev is Changing



The web is changing

New everything!

The web is changing

New everything!

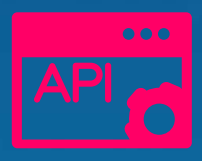
Can create new risk

motivation

New Technology

The web is changing

Problem



API's are often unprotected

Dev
OWASP
Open Web Application
Security Project

motivation

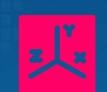
New Technology

The web is changing

Dev is Changing

New everything!

Complexity



Can create new risk

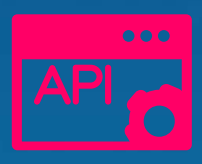
27 / 57

motivation

New Technology

The web is changing

Problem



API's are often unprotected

Dev

Ne

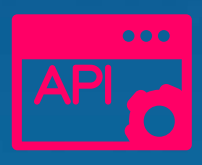
Tech change == Testing change

motivation

New Technology

The web is changing

Problem



API's are often unprotected

Dev

Ne

Tech change == Testing change

New Approaches, Tools & Techniques

OWASP
Open Web Application Security Project

The technical ecosystem is changing. Right now.

There is a paradigm shift, and ng of the end of monolithic applicat

No more excessively long relea
No more Waterfall.

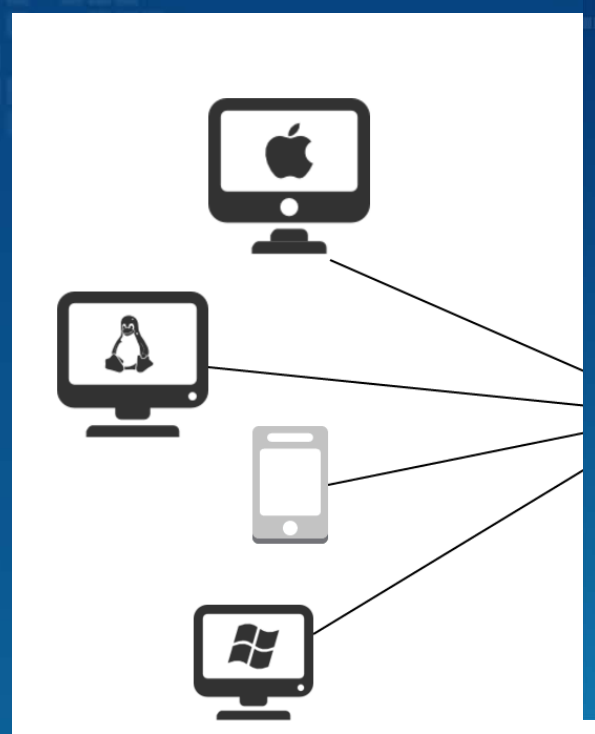
Security must keep pace.

OWASP
Open Web Application Security Project

motivation

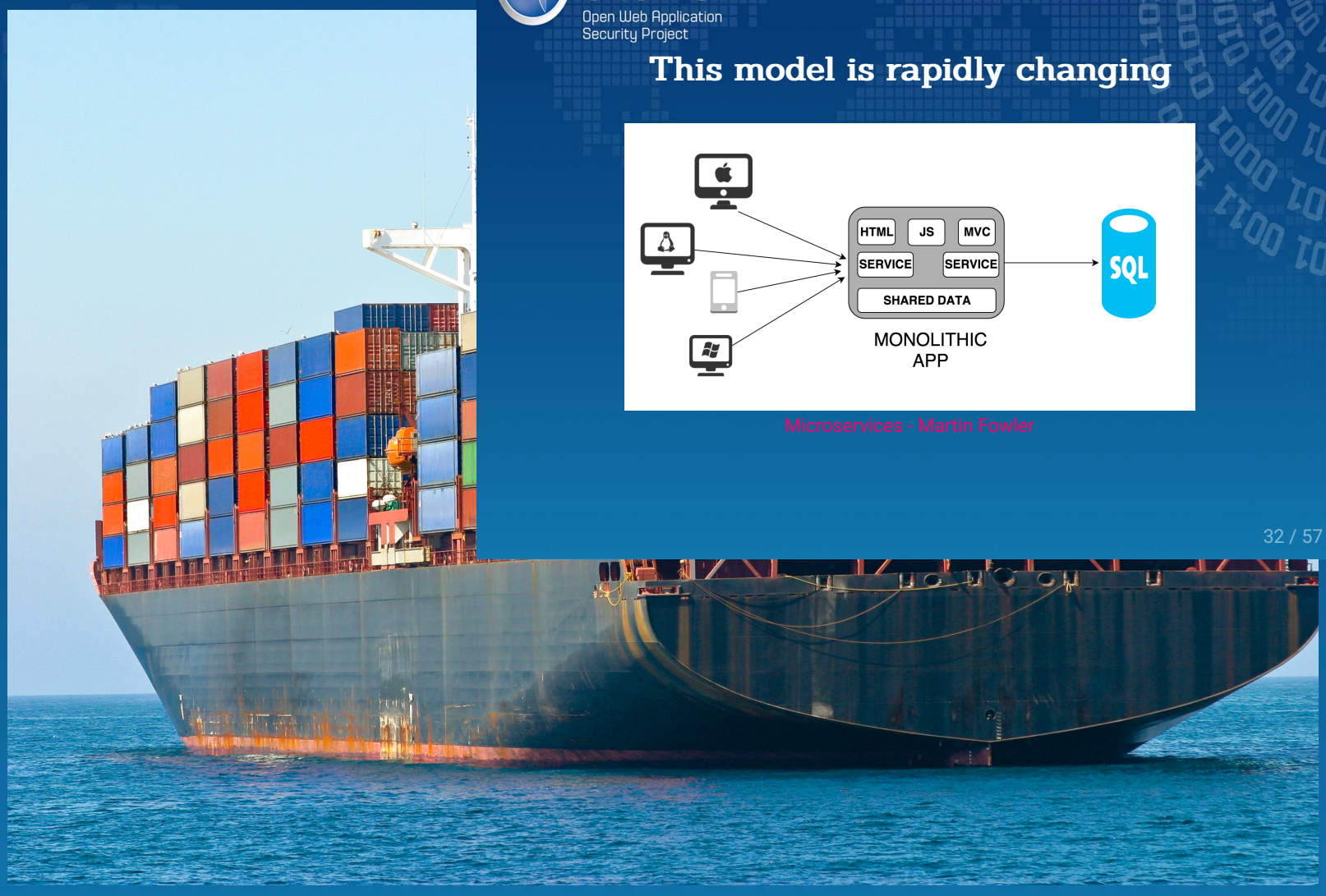
New Technology	Dev is Changing	Complexity
The web is changing	New everything!	Can create new risk
Problem	Evolution	New Tools
API's are often unprotected	Tech change == Testing change	New Approaches, Tools & Techniques

This model is rapidly changing



Microservices - Martin Fowler

The web is getting more and more complex.



OWASP
Open Web Application Security Project

This model is rapidly changing

Microservices - Martin Fowler

OWASP
Open Web Application Security Project

OWASP
Open Web Application Security Project

The web is getting more and more complex.

OWASP
Open Web Application Security Project

What are microservices?

In short, the microservice archite
style is an approach to developin
single application as a suite of s
services, each running in its own
and communicating with lightwe
mechanisms, often an HTTP res
API. These services are built aro
business capabilities and indepe
deployable by fully automated
deployment machinery. There is
minimum of centralized manage
these services, which may be wri
different programming languages and use
different data storage technologies.

-- James Lewis and Martin Fowler

OWASP
Open Web Application Security Project

Microservice Examples

Netflix uses microservices arc calls
everyday from more than 800 di o API.
Each API call then prompts ai ces.

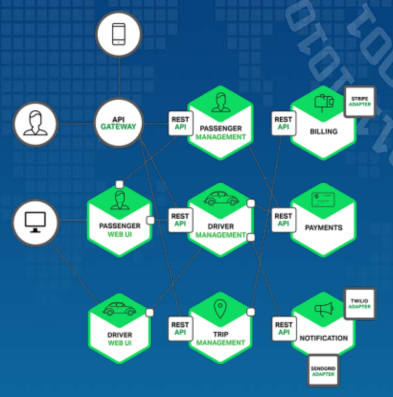
[redacted]

OWASP
Open Web Application Security Project

What are microservices?

In short, the microservice architectural style is an approach to developing a single application as a suite of small services, each running in its own process and communicating with lightweight mechanisms, often an HTTP resource API. These services are built around business capabilities and independently deployable by fully automated deployment machinery. There is a bare minimum of centralized management of these services, which may be written in different programming languages and use different data storage technologies.

-- James Lewis and Martin Fowler



35 / 57

OWASP
Open Web Application Security Project

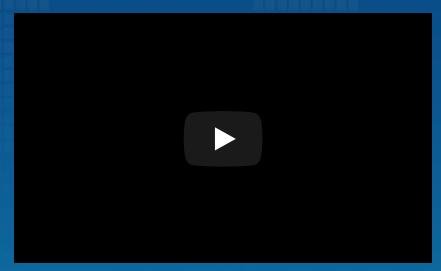
PROS

- Code organized around busi
- Web containers start fast == deployment.
- Code changes require only r service(s) to be modified & redeployed, not entire applic
- Better fault isolation: if one microservice fails, others wi
- Easy to scale and integrate v party services
- No long-term commitment to technology stack

OWASP
Open Web Application Security Project

Microservice Examples

Netflix uses microservices architecture. It receives more than one BILLION calls everyday from more than 800 different types of devices to its streaming-video API. Each API call then prompts around 5 additional calls to the back end services.



on and

in

as

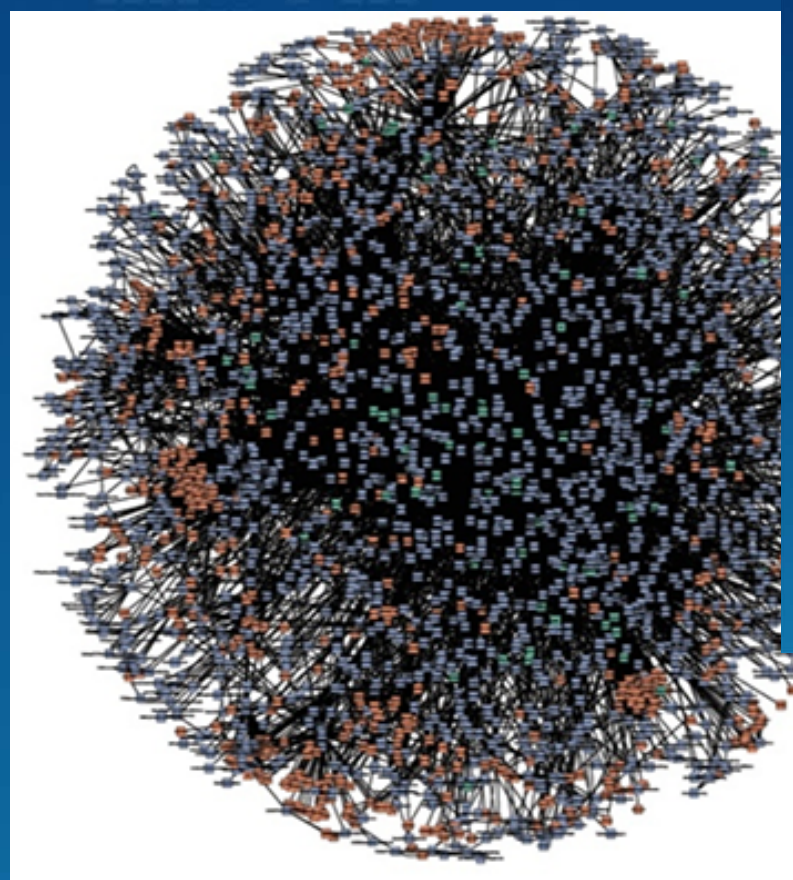
erance,

y, &

deal with message formats

- Implementing the mechanism of communication between services
- Situations that spans several services requires communication and cooperation between different teams

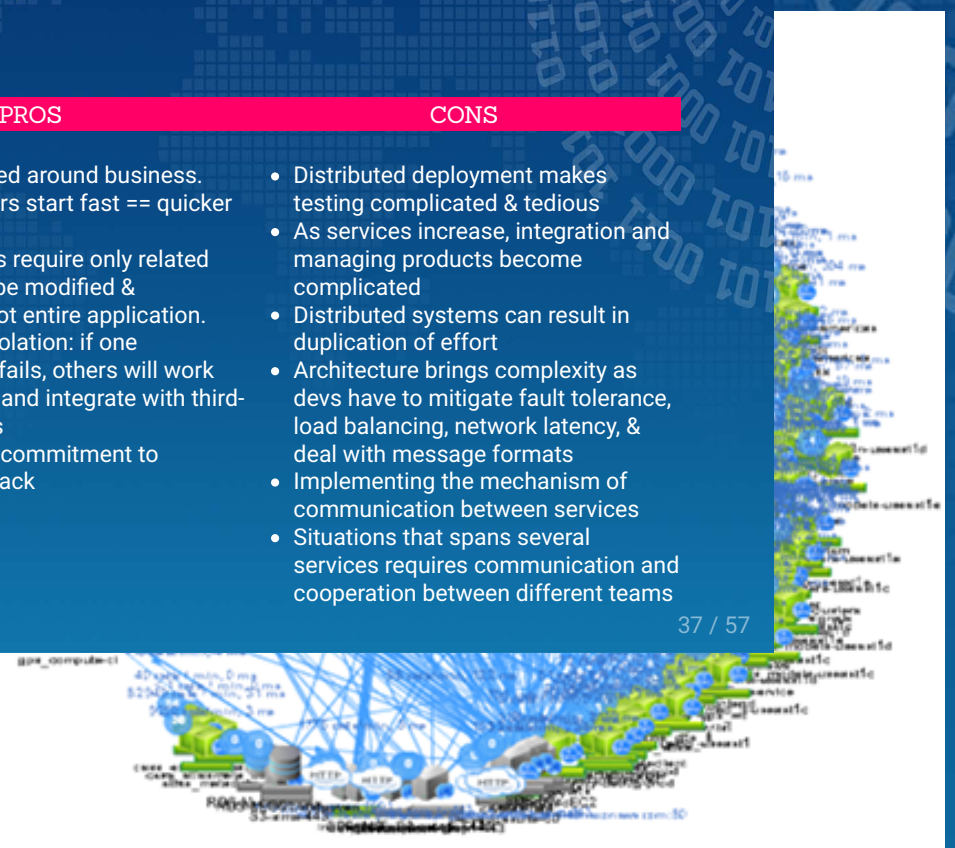
OWASP
Open Web Application Security Project



OWASP
Open Web Application Security Project

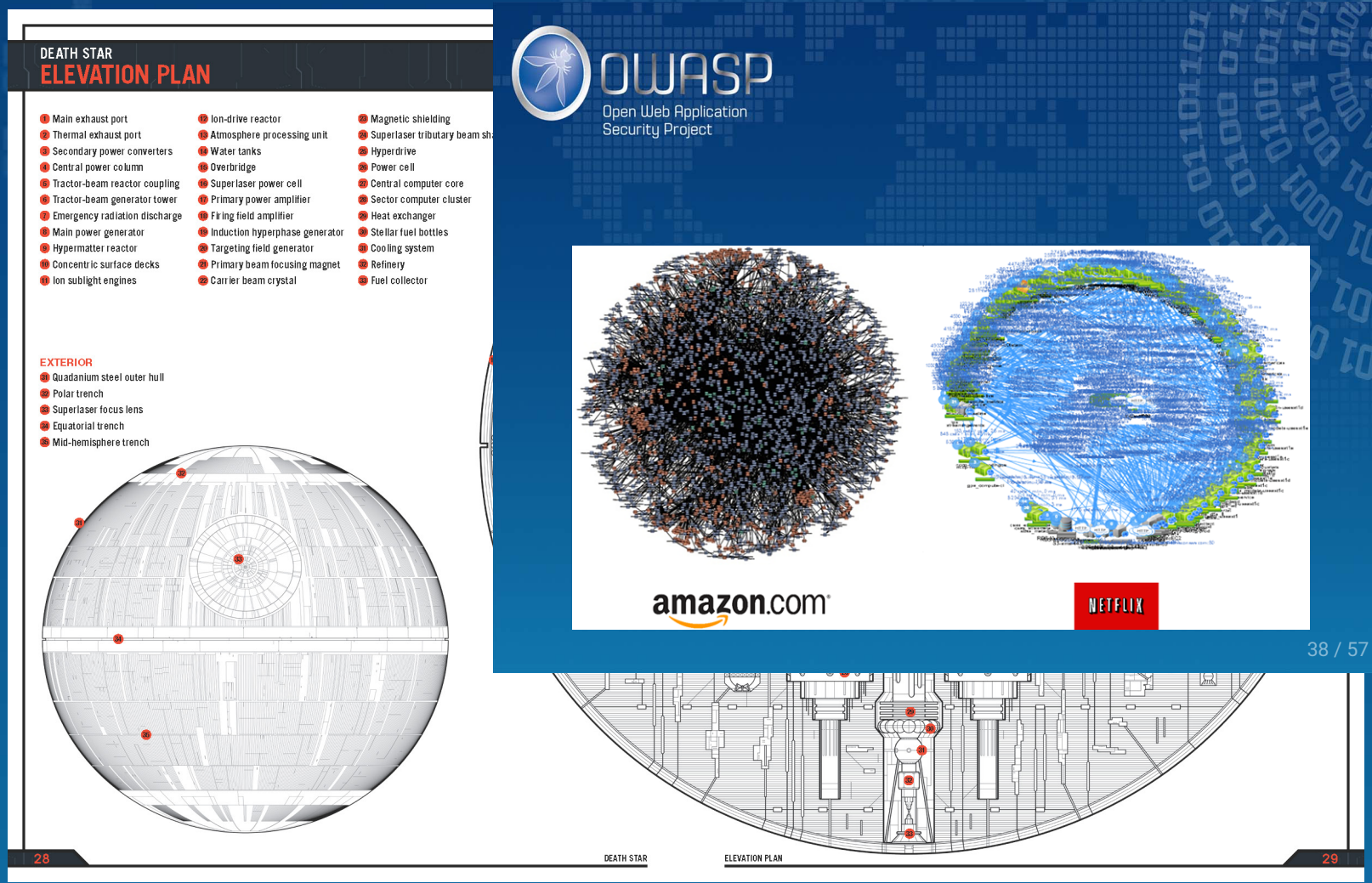
PROS	CONS
• Code organized around business. • Web containers start fast == quicker deployment. • Code changes require only related service(s) to be modified & redeployed, not entire application. • Better fault isolation: if one microservice fails, others will work • Easy to scale and integrate with third-party services • No long-term commitment to technology stack	• Distributed deployment makes testing complicated & tedious • As services increase, integration and managing products become complicated • Distributed systems can result in duplication of effort • Architecture brings complexity as devs have to mitigate fault tolerance, load balancing, network latency, & deal with message formats • Implementing the mechanism of communication between services • Situations that spans several services requires communication and cooperation between different teams

37 / 57



amazon.com

NETFLIX



One blast to the reactor module **and** the whole system goes down

OWASP
Open Web Application Security Project

Explosion of API Economy/webservices

Microservices

Containers

Frontend

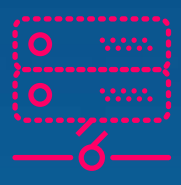
3rd Party APIs

Open Data

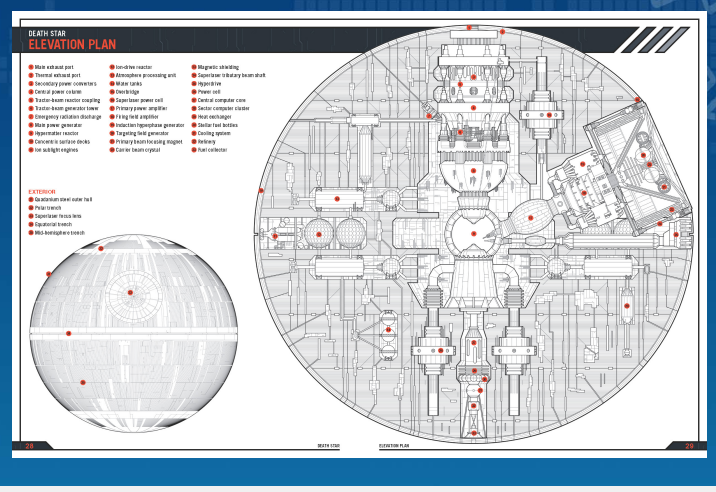
Serverless

Cloud

ation



OWASP
Open Web Application Security Project



One blast to the reactor module **and** the whole system goes down

39 / 57

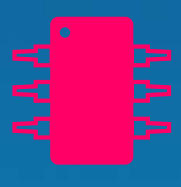
Continuous Integration

Continuous Delivery

Automated Pipelines

SaaS

Federated Identity



OWASP
Open Web Application Security Project

Security Implications of API Economy

Infosec need to

- Move faster - automating an and everything
- Reproducible and accurate r
- Verify/examine/test in short more often
- More training and more tool ever
- Reduce bottlenecks and sto gate" to developers

OWASP
Open Web Application Security Project

API Technology

API's generally travel over HTTP, are served on Web Servers, generally follow RESTful design patterns, but dont respond w

SOAP - Simple Object Access Pr sfer

- An actual protoco
- More like an enve
- More overhead
- XML only

Response Formats

XML eXtensible markup language

JSON Javascript Object NOtaton

JSON Swagger used to define API functionality

Don't ignore web services and APIs: just because they don't have pretty GUIs doesn't mean they can't be hacked!

OWASP
Open Web Application Security Project

OWASP Top 10 2017

OWASP Top 10 – 2017 (New)
A1 – Injection
A2 – Broken Authentication and Session Management
A3 – Cross-Site Scripting (XSS)
A4 – Broken Access Control (Original category in 2003/
A5 – Security Misconfiguration
A6 – Sensitive Data Exposure
A7 – Insufficient Attack Protection (NEW)
A8 – Cross-Site Request Forgery (CSRF)
A9 – Using Components with Known Vulnerabilities
A10 – Underprotected APIs (NEW)

OWASP
Open Web Application Security Project

API Technology

API's generally travel over HTTP, are served on Web Servers, generally follow RESTful design patterns, but dont respond with HTML

SOAP - Simple Object Access Protocol

- An actual protocol
- More like an envelope
- More overhead
- XML only

REST - Representational State Transfer

- Lightweight
- More like a postcard
- Uses HTTP CRUD
- JSON notation

Response Formats

XML eXtensible markup language

JSON Javascript Object NOtaton

Swagger used to define API functionality

Don't ignore web services and APIs: just because they don't have pretty GUIs doesn't mean they can't be hacked!

42 / 57

st, but
sion
fined
pe by
ernal
s. APIs

d
era.
f
nd
now things with the expansion of IoT.

API insecurity IRL

The unsecured APIs allow anyone who knows the VIN of a car to access non-critical features such as climate control and battery charge management from anywhere across the Internet. Additionally, someone exploiting the unauthenticated APIs can see the car's estimated driving range.

OWASP
Open Web Application Security Project

API insecurity IRL

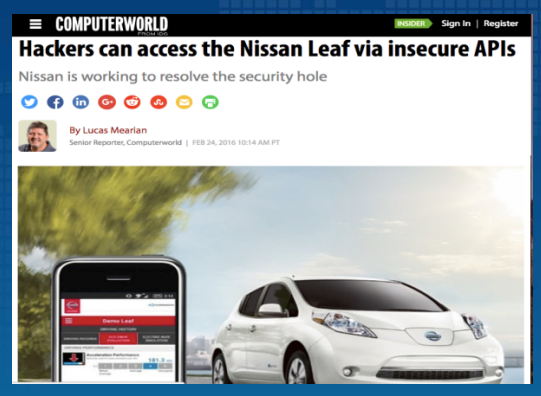
I added a breakpoint and edite
allow me to play for ever, no m
existent server checks! This w
much as I want, but I don't war
make the game much easier sc

Looking some more, I found th
details specify the number of c
is played by getting candy of o
means a considerably easier ga
game won't end at all! I discov

```
http://candycrush.king.co

{
  "levelData":{
    "numberOfColours":4,
    "gameModeName":"Light
    "pepperCandyMax":1,
    "pepperCandySpawn":3,
    "chameleonCandyMax":0
  },
}
```

OWASP
Open Web Application Security Project
API insecurity IRL



The unsecured APIs allow anyone who knows the VIN of a car to access non-critical features such as climate control and battery charge management from anywhere across the Internet. Additionally, someone exploiting the unauthenticated APIs can see the car's estimated driving range.

How to cheat at CandyCrush and get unlimited lives!

OWASP
Open Web Application
Security Project

What is Pixi

Pixi is a vulnerable web app & A
vulnerabi

pi
share the world

OWASP
Open Web Application
Security Project

API insecurity IRL

mmon

I added a breakpoint and edited `lives` to always be 5, which did allow me to play for ever, no matter how much I lost. Yay for non-existent server checks! This wasn't great, though. Sure, I could play as much as I want, but I don't want to play at all! I wanted to see if I could make the game much easier somehow.


Productivity Sapper Saga

Looking some more, I found the call that loads the level, and the level details specify the number of colors to use on the level. Since the game is played by getting candy of one color to line up in a row, fewer colors means a considerably easier game. However, too few colors and the game won't end at all! I discovered that four colors is the sweet spot, and edited the level accordingly:

```
https://candycrush.king.com/api/gameStart

{
  "levelData":{
    "numberOfColours":4,
    "gameModeName":"Light up",
    "pepperCandyMax":1,
    "pepperCandySpawn":3,
    "chameleonCandyMax":0
  },
}
```

How to cheat at CandyCrush and get unlimited lives!

What is Pixi

Pixi is pai

A photo sharing app!

Create &

Runs on the MEAN stack (MongoDB, Angular.js, Node, Express.js)

Services running in docker containers

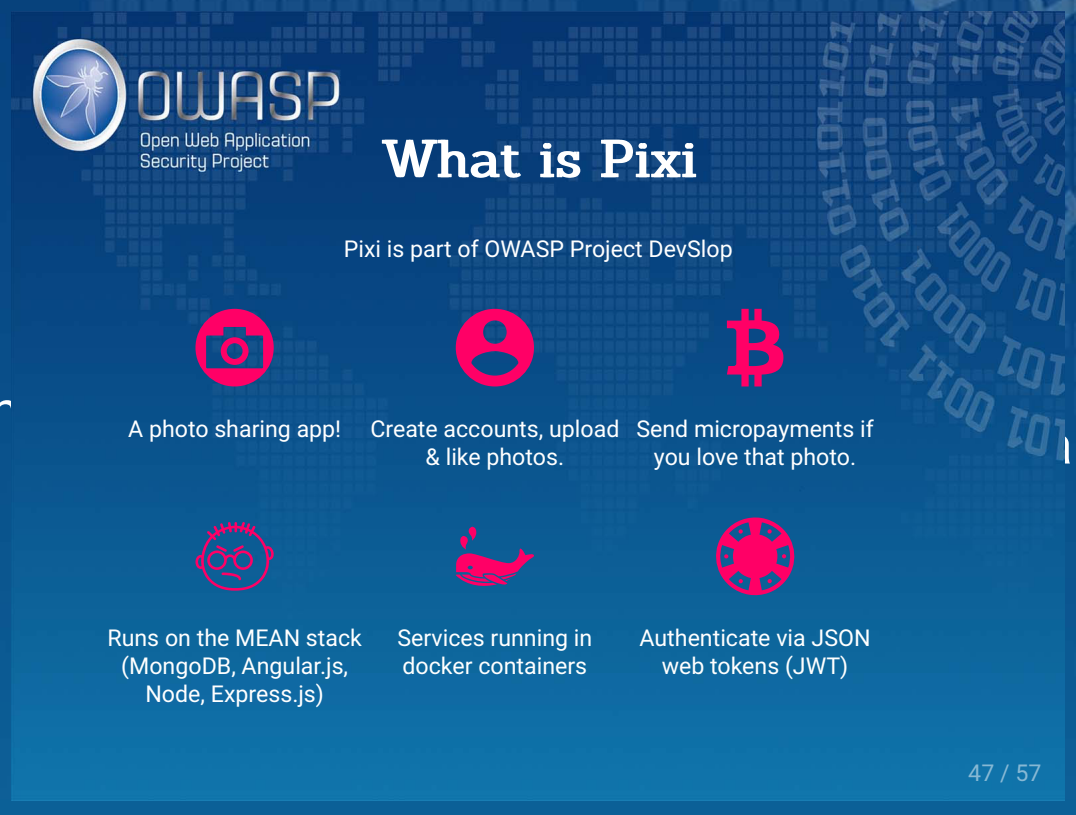
Authenticate via JSON web tokens (JWT)

OWASP
Open Web Application Security Project

What is Pixi

Pixi is a vulnerable web app & API, which (hopefully) helps to understand common vulnerabilities in web apps & services.

The MEAN Stack



mongoDB.

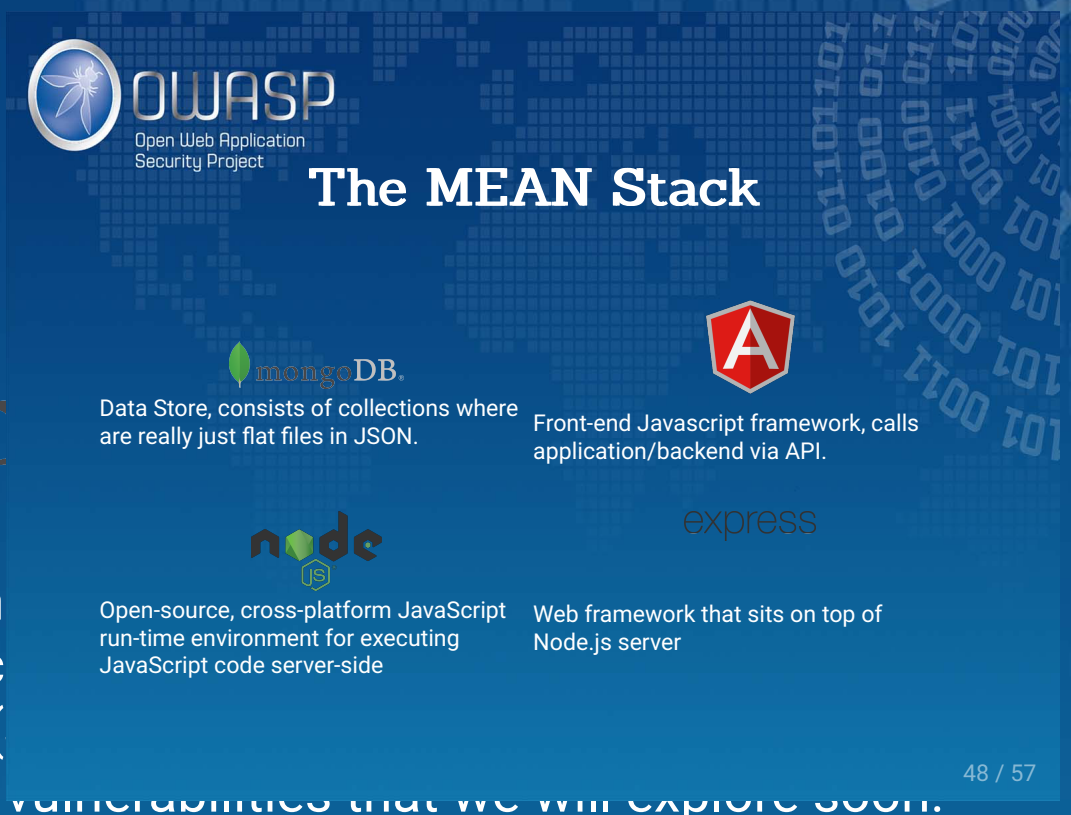
Data Store, consists of collectior are really just flat files in JSON.

Open-source, cross-platform JavaScript run-time environment for executing JavaScript code server-side

Web framework that sits on top of Node.js server

OWASP
Open Web Application Security Project

MongoDB



- There are many non-relation
- Mongo is a data store that c files in Javascript Object Notation (
- Mongo has its own security vulnerabilities that we will explore soon.

OWASP
Open Web Application Security Project

Angular.js

OWASP
Open Web Application Security Project

MongoDB

mongoDB.

- There are many non-relational databases these days.
- Mongo is a data store that contains “collections” which are really just flat files in Javascript Object Notation (JSON).
- Mongo has its own security vulnerabilities that we will explore soon.

Developed by Google, but open-s PI calls to the middle/back end. Very useful for single-page applications Generally pretty good about input sanitization, but we will see examples of bypassing validation.

OWASP
Open Web Application
Security Project

Node.js

Historically, JavaScript was used

ripts
written in JavaScript are embedd e by a
JavaScript engine in the user's w e used
for server-side scripting, and runs page
content before the page is sent t

Server-side javascript framework. Event-Driven & Asynchronous

Why use Node? Fast! Easy to write modules/packages.

OWASP
Open Web Application
Security Project

Express.js

Web framework that sits on top c

Why use Express? Lets you quick

node functions.

OWASP
Open Web Application
Security Project
Node.js

dled by

Historically, JavaScript was used primarily for client-side scripting, in which scripts written in JavaScript are embedded in a webpage's HTML, to be run client-side by a JavaScript engine in the user's web browser. Node.js enables JavaScript to be used for server-side scripting, and runs scripts server-side to produce dynamic web page content before the page is sent to the user's web browser. C

Server-side javascript framework. Event-Driven & Asynchronous

Why use Node? Fast! Easy to write modules/packages.

OWASP
Open Web Application Security Project

Docker

OWASP
Open Web Application Security Project

Express.js

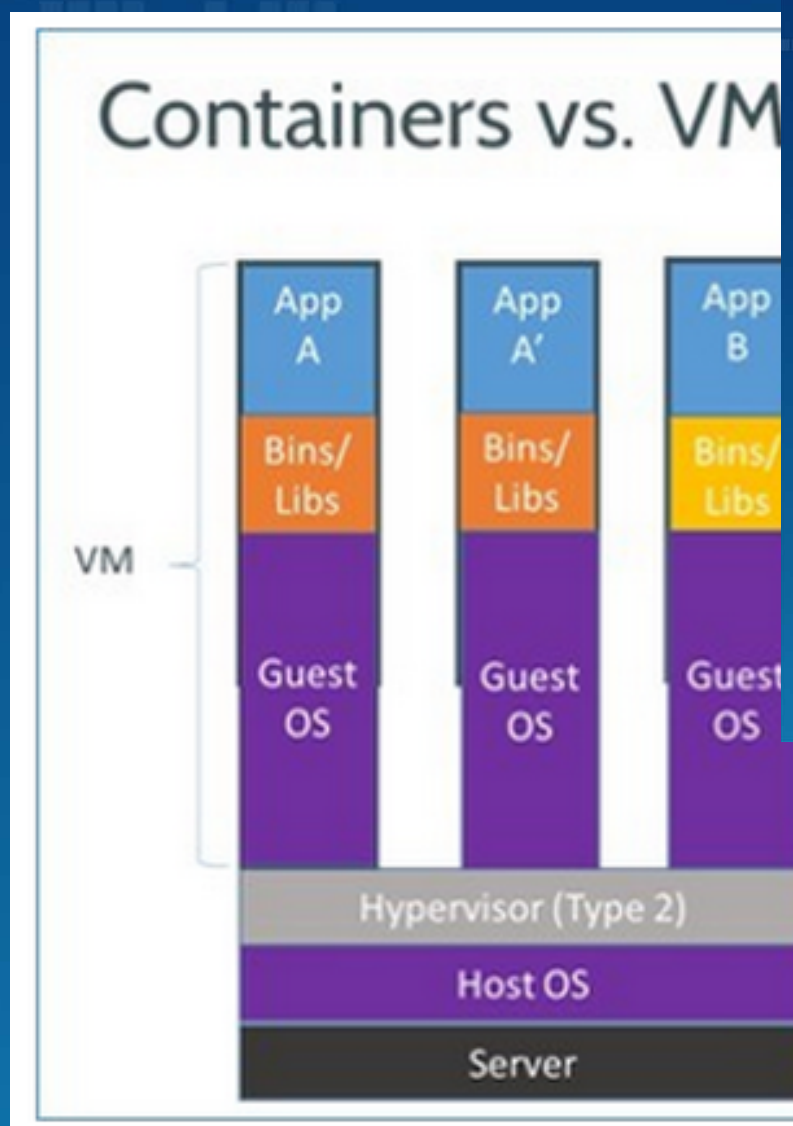
express

Web framework that sits on top of Node.js to handle web requests.

Why use Express? Lets you quickly create web routes/API routes that are handled by node functions.

What is Docker? Linux containers : isolated CPU, memory, block I/O, of the host operating system. service c rs.

- Can scale dynamically, if there is an increase in traffic/load
- Fits the paradigm: One monolith vs Many Containers
- Unique Security Challenges



OWASP
Open Web Application
Security Project

Docker

What is Docker? Linux containers with self-contained execution environments: isolated CPU, memory, block I/O, and network resources that share the kernel of the host operating system. service can be contained in separate docker containers.

- Can scale dynamically, if there is an increase in traffic/load
- Fits the paradigm: One monolith vs Many Containers
- Unique Security Challenges

Docker Security Challenges

- Start Docker containers with
  instead of root.
- Remove SUID flags from you
  attacks even harder.
- Configure Docker control gr
  resources each container ca
  based DoS attacks.
- Use namespaces in Docker
  Namespaces help assure th
  can't affect those in other c
- Don't use images from repos you don't trust. Avoid public repos if they're not from an official source and you don't know the maintainer.
- Consider using a tool to validate containers from your registries.

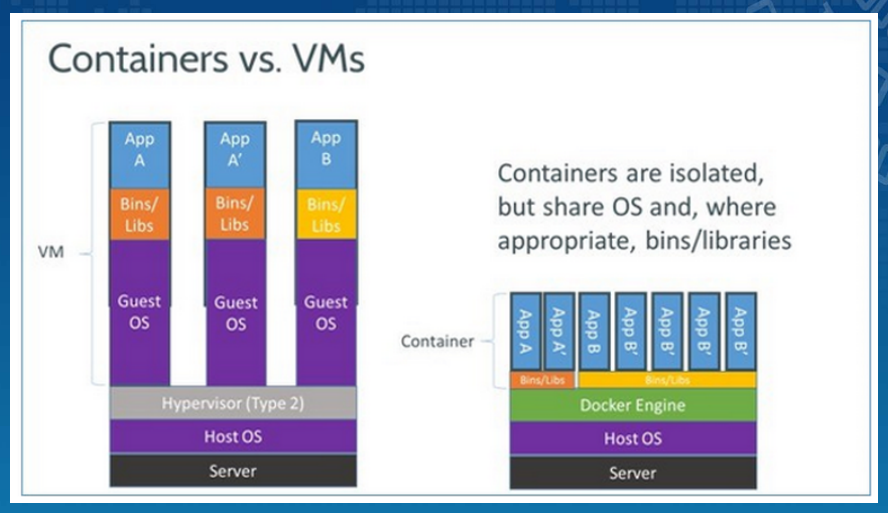
er

calation

ainer-

iner

OWASP
Open Web Application
Security Project



OWASP
Open Web Application
Security Project

OpenAPI/S

OWASP
Open Web Application
Security Project

Docker Security Challenges

- Start Docker containers with the -u flag so that they run as an ordinary user instead of root.
- Remove SUID flags from your container images. This makes privilege escalation attacks even harder.
- Configure Docker control groups, which let you set limits on how many resources each container can use. This can help prevent preventing container-based DoS attacks.
- Use namespaces in Docker to isolate containers from one another. Namespaces help assure that a user or process running inside one container can't affect those in other containers.
- Don't use images from repos you don't trust. Avoid public repos if they're not from an official source and you don't know the maintainer.
- Consider using a tool to validate containers from your registries.

OpenAPI/Swagger is just an API l the routes and methods of an API, ar . consumption. Meant to allow automated consumption by external services/applications.

JSON Web Tokens (JWT)

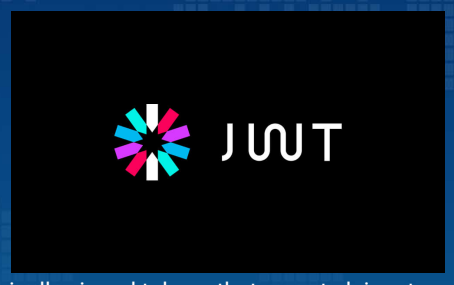
JWT are cryptographically signed ng
service. Very similar to session c e can
authenticate and decode the tok
symmetrically or asymmetrically.

- Symmetric key, is when each service has the private key that is used to verify the tokens
- Asymmetric is when each service has the public key to verify a token that was signed once by the private key.

Security risks related to key storage and overloaded data in JWT.

OWASP
Open Web Application Security Project

JSON Web Tokens (JWT)



JWT are cryptographically signed tokens that assert claims to an authenticating service. Very similar to session cookies, but much more universal. Any service can authenticate and decode the token as long as it has the key. Can be signed symmetrically or asymmetrically.

- Symmetric key, is when each service has the private key that is used to verify the tokens
- Asymmetric is when each service has the public key to verify a token that was signed once by the private key.

Security risks related to key storage and overloaded data in JWT.