

Hello and welcome!

**#VismaAClabs @ligaac\_labs**  
will begin shortly....

Thursday, 22nd of April

# Introduction to **Application Security**

What will we learn

Attackers and targets

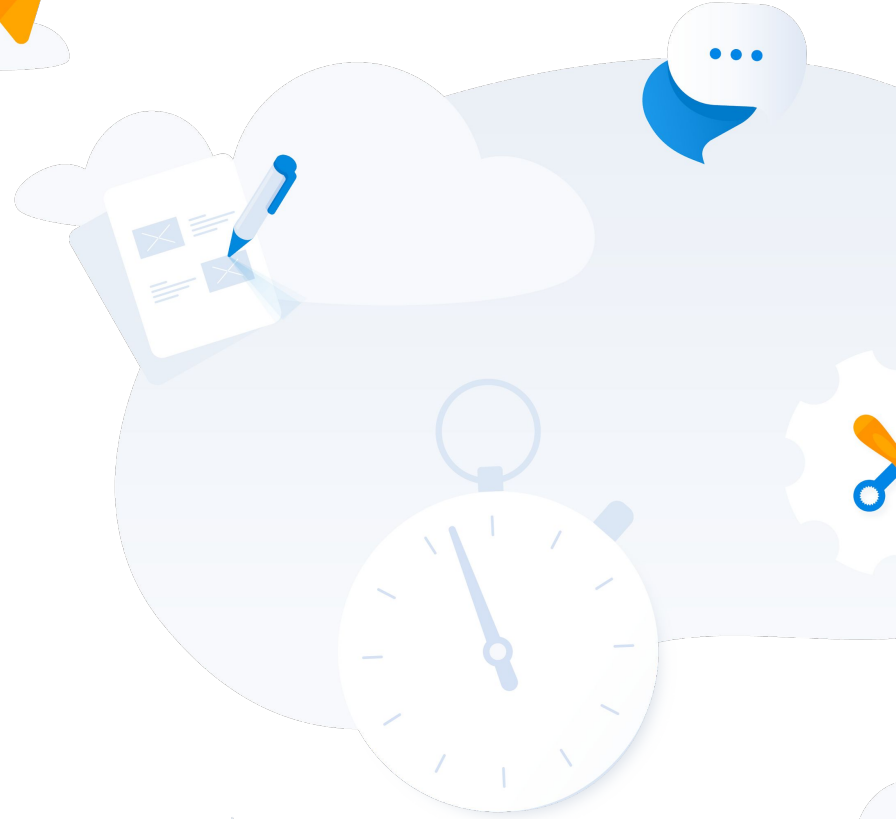
The software development life-cycle

Application security

Database security and data privacy

OWASP

CTF



## Sessions

Week 16 ( 22nd April )

Week 17

Week 18 ( 6th May )

Week 19 ( 13th May )

Week 20 ( 19th,20th May )

Week 21 ( 27th May )

Week 22 ( 3rd June )

Week 23 ( 10th June )

Session 1

Vacation

Session 2

Session 3

Session 4 ( 2h each day) 08 AM - 10 AM

CTF

Session 5 / CTI, Monitoring

Session 6 - final session

A typical day  
4 hour day

08:00 - 09:00

Work

09:00 - 09:10

Break

09:10 - 10:30

Work

10:30 - 11:00

Long break

11:00 - 12:00

Work

# Let's begin our journey

22nd April



# Agenda

Attackers and targets

The software development life-cycle

Introduction to OWASP

OWASP - Injection

# Targets and attackers

For many years security was understood as IT infrastructure security.

Cloud changed this. Today DevSecOps teams are responsible for everything.

Expectations are high, roles are combined , responsibility is shared.

Most breaches are because of poor software security.

Many breaches come from internal unhappy employees.



# Who has been hacked - data breaches



## Qatar National Bank

In July 2015, the Qatar National Bank suffered a data breach which exposed 15k documents totalling 1.4GB and detailing more than 100k accounts with passwords and PINs. The incident was made public some 9 months later in April 2016 when the documents appeared publicly on a file sharing site. Analysis of the breached data suggests the attack began by exploiting a SQL injection flaw in the bank's website.

**Breach date:** 1 July 2015

**Date added to HIBP:** 1 May 2016

**Compromised accounts:** 88,678

**Compromised data:** Bank account numbers, Customer feedback, Dates of birth, Financial transactions, Genders, Geographic locations, Government issued IDs, IP addresses, Marital statuses, Names, Passwords, Phone numbers, Physical addresses, PINs, Security questions and answers, Spoken languages

[Permalink](#)

*And many others...*



# Have you been pwned?

## Who is behind Have I Been Pwned (HIBP)

I'm Troy Hunt, a Microsoft Regional Director and Most Valuable Professional awardee for Developer Security, blogger at [troyhunt.com](https://troyhunt.com), international speaker on web security and the author of many top-rating security courses for web developers on [Pluralsight](https://www.pluralsight.com).

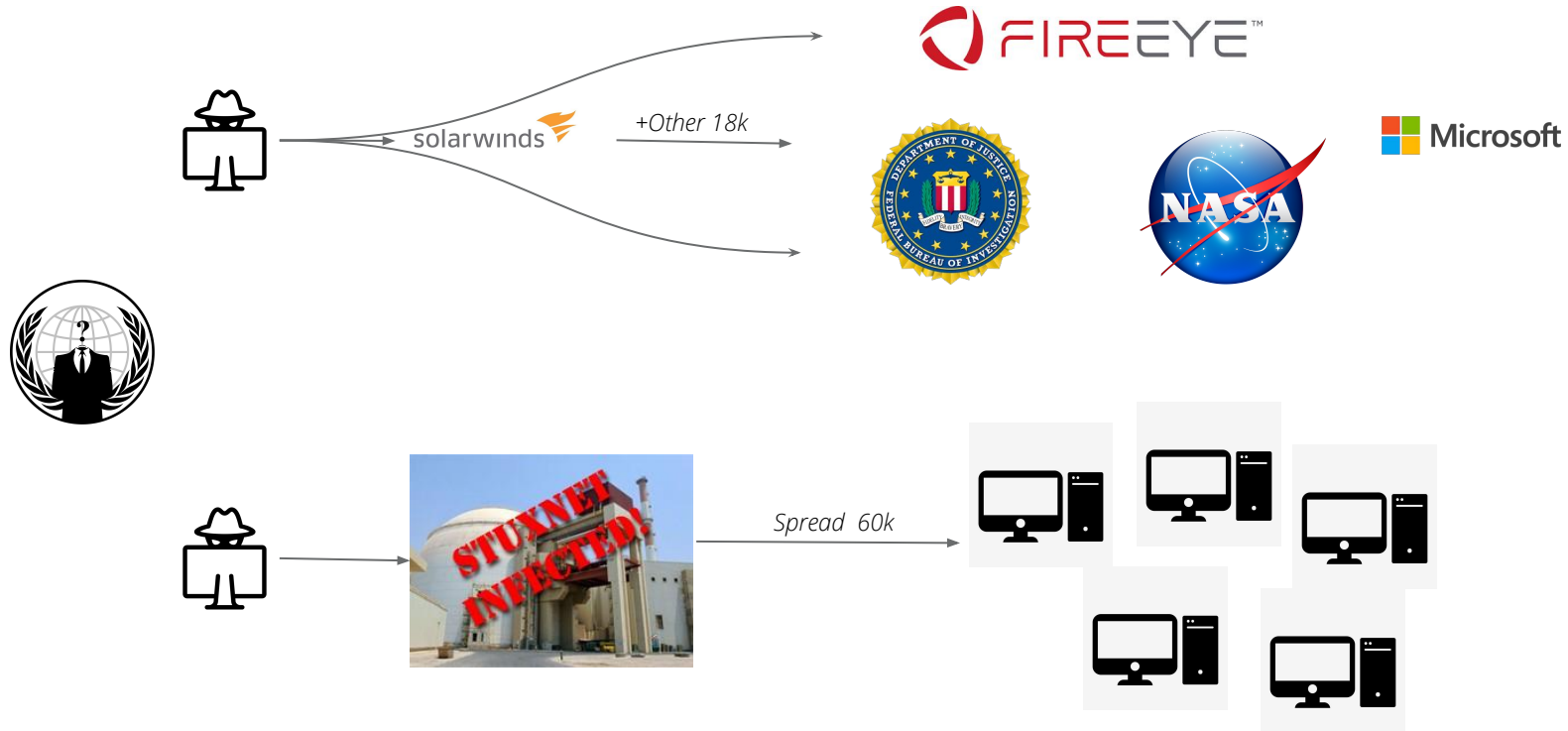
I created HIBP as a free resource for anyone to quickly assess if they may have been put at risk due to an online account of theirs having been compromised or "pwned" in a data breach. I wanted to keep it dead simple to use and entirely free so that it could be of maximum benefit to the community.

Short of the odd donation, all costs for building, running and keeping the service currently come directly out of my own pocket. Fortunately, today's modern cloud services like Microsoft Azure make it possible to do this without breaking the bank!



[HaveIBeenPwned](https://haveibeenpwned.com)

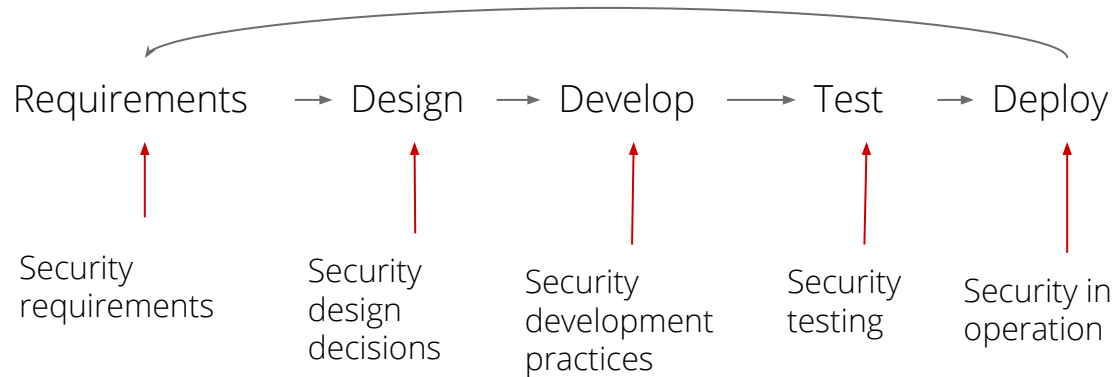
# Who has been hacked - cyber attacks



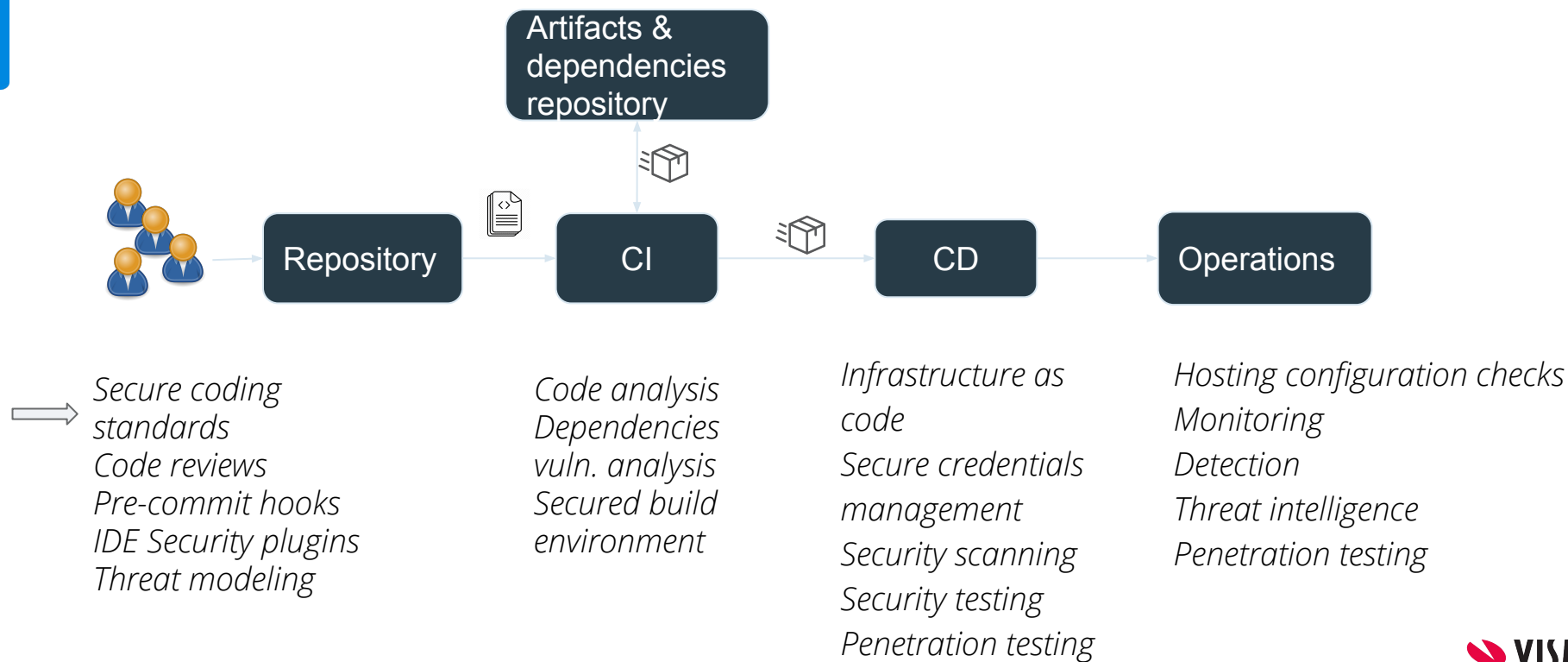
# Who are the attackers

- Hacktivists
  - motivated by civil disobedience to promote a political agenda or social change
  - poorly funded
- Online criminals
  - financial data
  - motivated by cash
- Nation states
  - cyber warfare
  - national or political interests
  - unlimited funded

# Security in the SDLC



# Security in the delivery pipeline



# OWASP

## Introduction

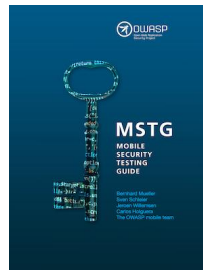
Our experience taught us that the current level of security of web-applications is not sufficient enough to ensure security. This is mainly because web-developers simply aren't aware of the risks and dangers that are lurking, waiting to be exploited by hackers.

Open Web Application Security Project

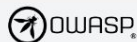
A nonprofit foundation that works to improve the security of software.

They run different projects helping the IT industry to learn how to strengthen application security.

Their projects expand to all the software delivery pipeline from security knowledge framework, to testing guidelines, training applications and testing tools.



# OWASP local meet-ups



PROJECTS CHAPTERS EVENTS ABOUT

Search OWASP.org

Donate

Join

## OWASP Timisoara

### Welcome

### Welcome to the OWASP Timisoara Chapter Homepage

Follow us on [Twitter](#). Follow us on [Meetup](#). Follow us on [LinkedIn](#). Subscribe on [YouTube](#).

Timisoara has an evolved software development community and one of the most important aspects that we aim to achieve is to continuously improve the application security world.

Everyone is welcome to join our chapter meetings, members and non-members. OWASP Timisoara Chapter meetings / events are free and open, so please join us!

The chapter leaders are [Catalin Curelaru](#) and [Daniel Ilies](#).

The Chapter Board Members are: Monica Iovan (Education), Ioana Piroksa ( PR/Marketing), Claudiu Ivan.

Anyone who wants to get involved and help the Chapter evolve is very welcome and please just contact us.

If you want to present at one of our meetings / events (please read the [speaker agreement](#)).

In case that you have any questions about the OWASP Timisoara Chapter, send an email to [Catalin Curelaru](#).

Next event: For details please check [Upcoming Events](#)!

- Past chapter leaders 2015 - 2019 Cornel Punga 2015 - 2019 Florina Rosiu

### Upcoming events

Please see our [Meetup page](#) for more details and to register as attendee

Watch 5

Star 1

The OWASP® Foundation works to improve the security of software through its community-led open source software projects, hundreds of chapters worldwide, tens of thousands of members, and by hosting local and global conferences.

### Leaders

[Catalin Curelaru](#)

[Daniel Ilies](#)

### Upcoming Global Events





# Overview OWASP Top 10

## ✓ **Injection**

Untrusted data is sent to an interpreter and tricks it to execute unintended commands.

## ✓ **Broken Authentication**

The attacker is able to assume other users identities.

## ✓ **Sensitive data exposure**

Sensitive data should be handled with care with exchanged with the browser.

## ✓ **XML External Entities**

XML processors evaluating external entities URI during XML document parsing.

## ✓ **Broken Access Control**

The attacker is able to access unauthorized functions, view sensitive data, change user data.

## ✓ **Security misconfiguration**

Insecure default configurations, unprotected storage and databases, verbose logging, misconfigured HTTP headers.

# OWASP Top 10

## ✓ **Cross-site scripting (XSS)**

Execute scripts in victim's browser to hijack user session, redirect to malicious sites, crypto mining

## ✓ **Insecure deserialization**

Untrusted data defines the data type of the object that the stream will be deserialized to.

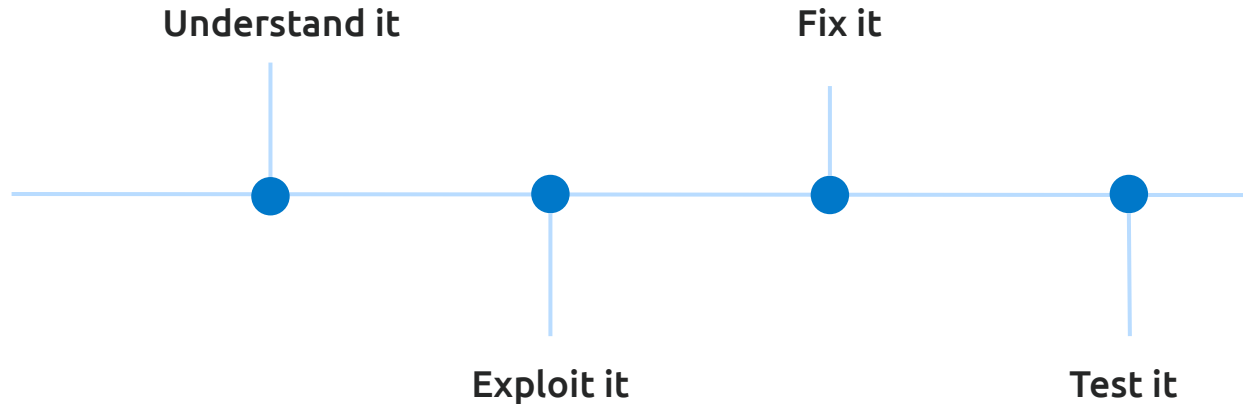
## ✓ **Using components with known vulnerabilities**

External libraries, packages, frameworks run with the same privileges as the application. Attackers exploit known vulnerabilities.

## ✓ **Insufficient logging and monitoring**

Helps attackers achieve their goals without being detected in time.

# Working model



# Injection

What ?

A user is allowed to send code to an application and that code gets executed by the application.

Multiple types of injection : Sql, NoSql, LDAP, OS

Code or command injection.

Why ?

Lack or poor user input or output validation. Lack of input encoding.

# Sql Injection

How ?

Username:

Password:

```
SELECT * FROM Users WHERE Name ="John Doe" AND Pass ="myPass"
```

---

# Injection

# Demo

# Injection

```
select name from sysobjects where id=(  
    select top 1 id from (  
        select top 3 id from sysobjects where xtype=char(85) order by id asc  
    ) sq order by id desc
```

What do you think the query does?

What represents *char(85)* ?

What can we do with this information that we found? Are we able to further exploit the application?

# Injection - let's fix it

- Database; least privilege
- Parameterized queries
- Stored procedures\*
  - when written correctly
- ORM Framework
- Data validation and whitelisting
  - Get away from strings
  - Use regular expressions to validate emails, phone numbers

What about blacklisting ?



Respect

Reliability

Innovation

Competence

Team spirit

