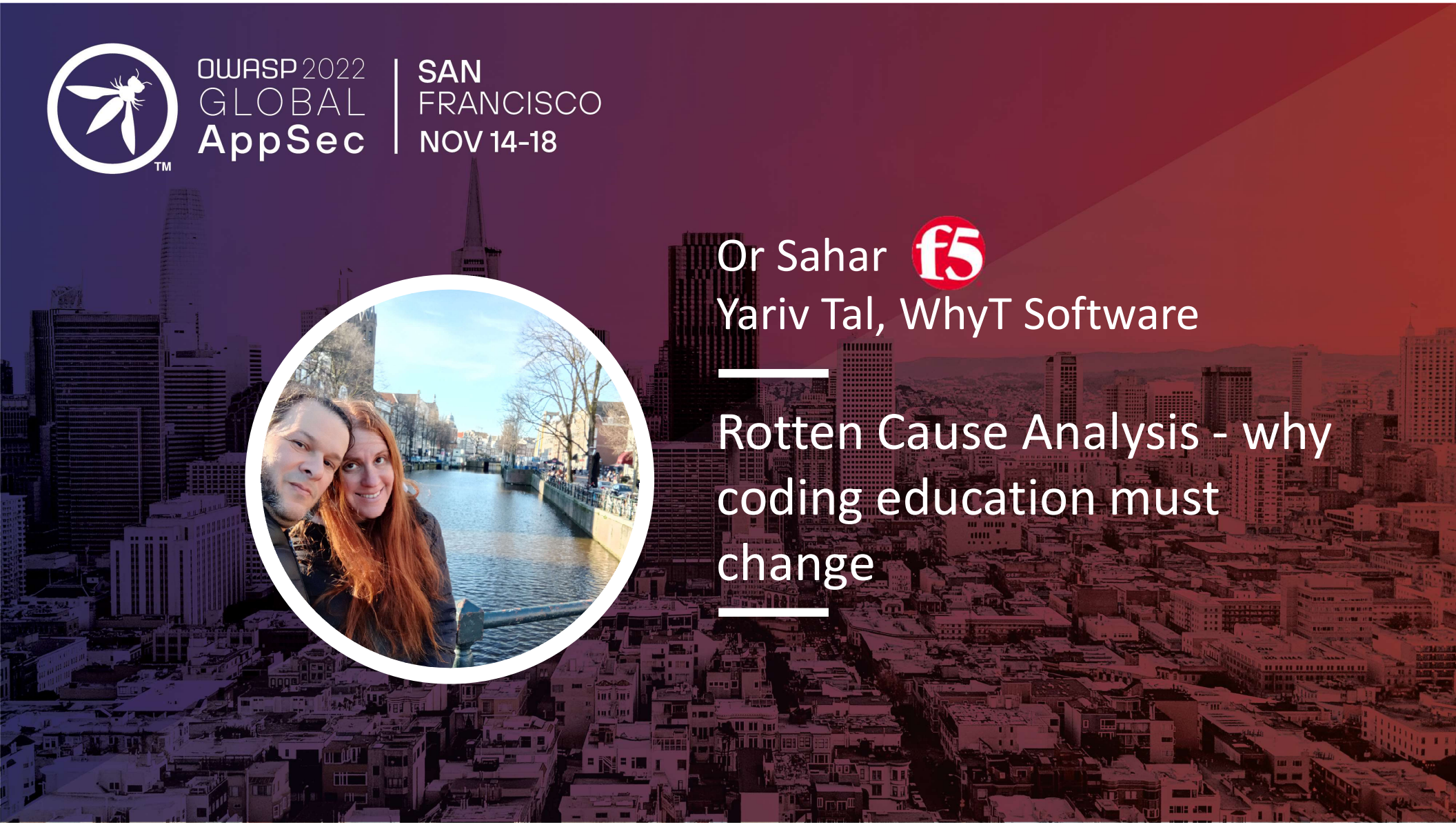
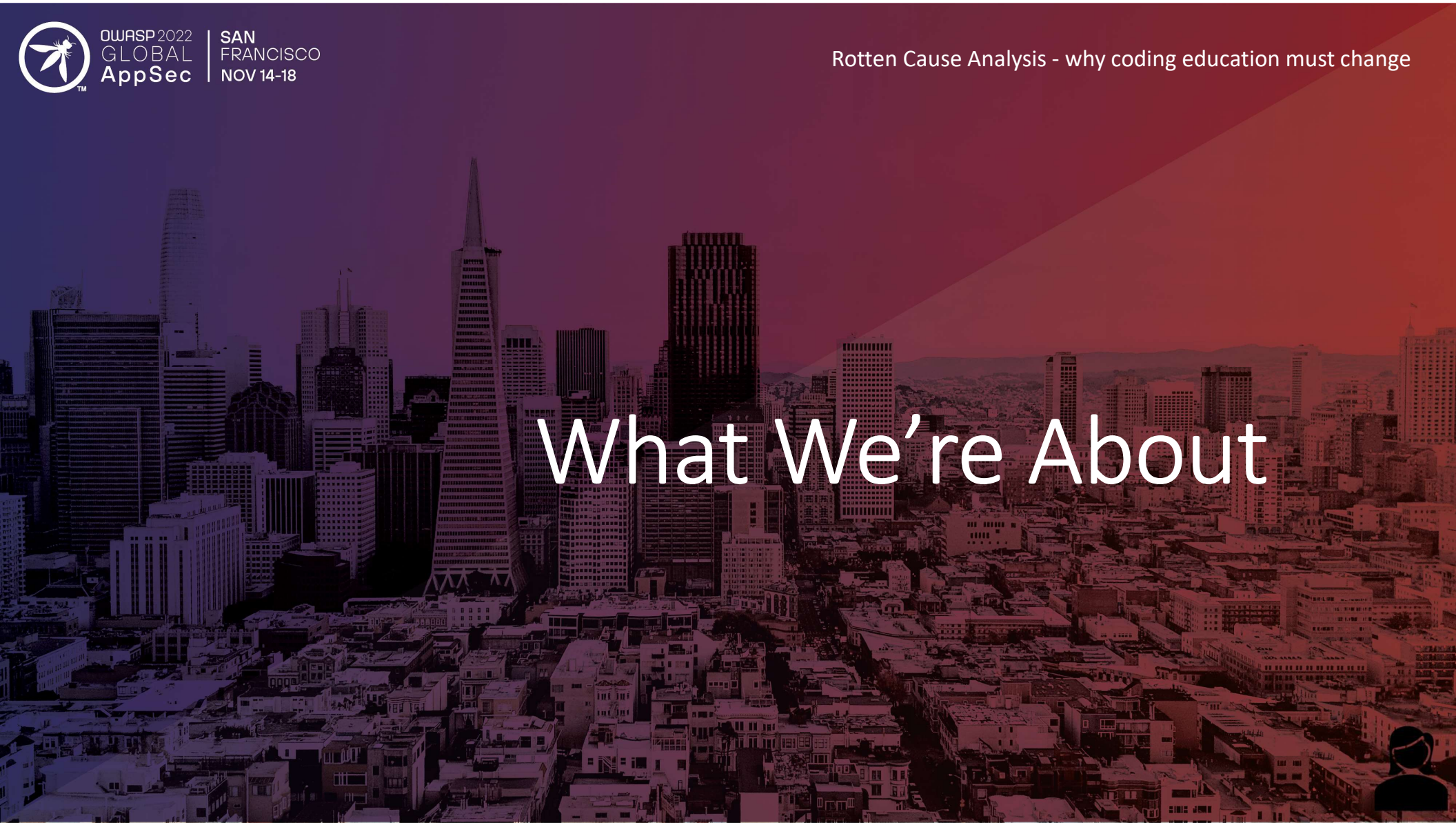
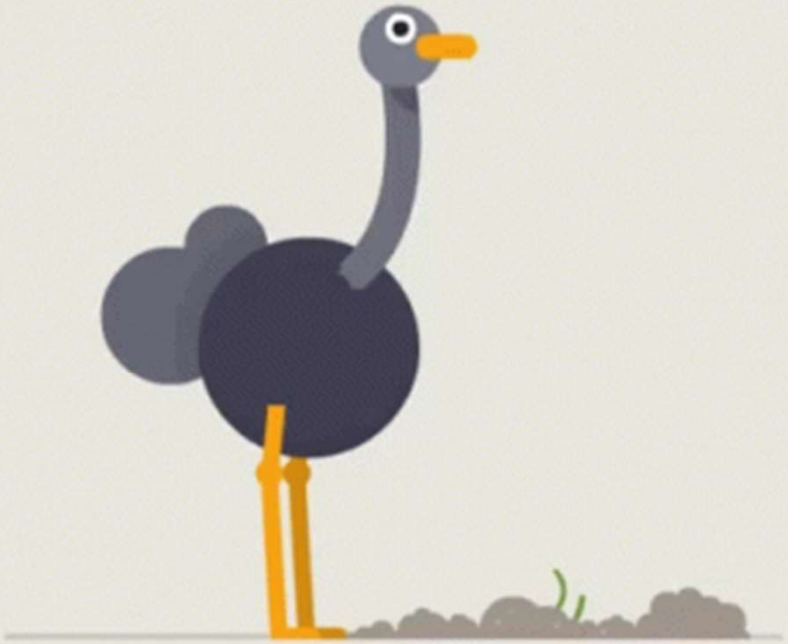


Or Sahar
Yariv Tal, WhyT Software

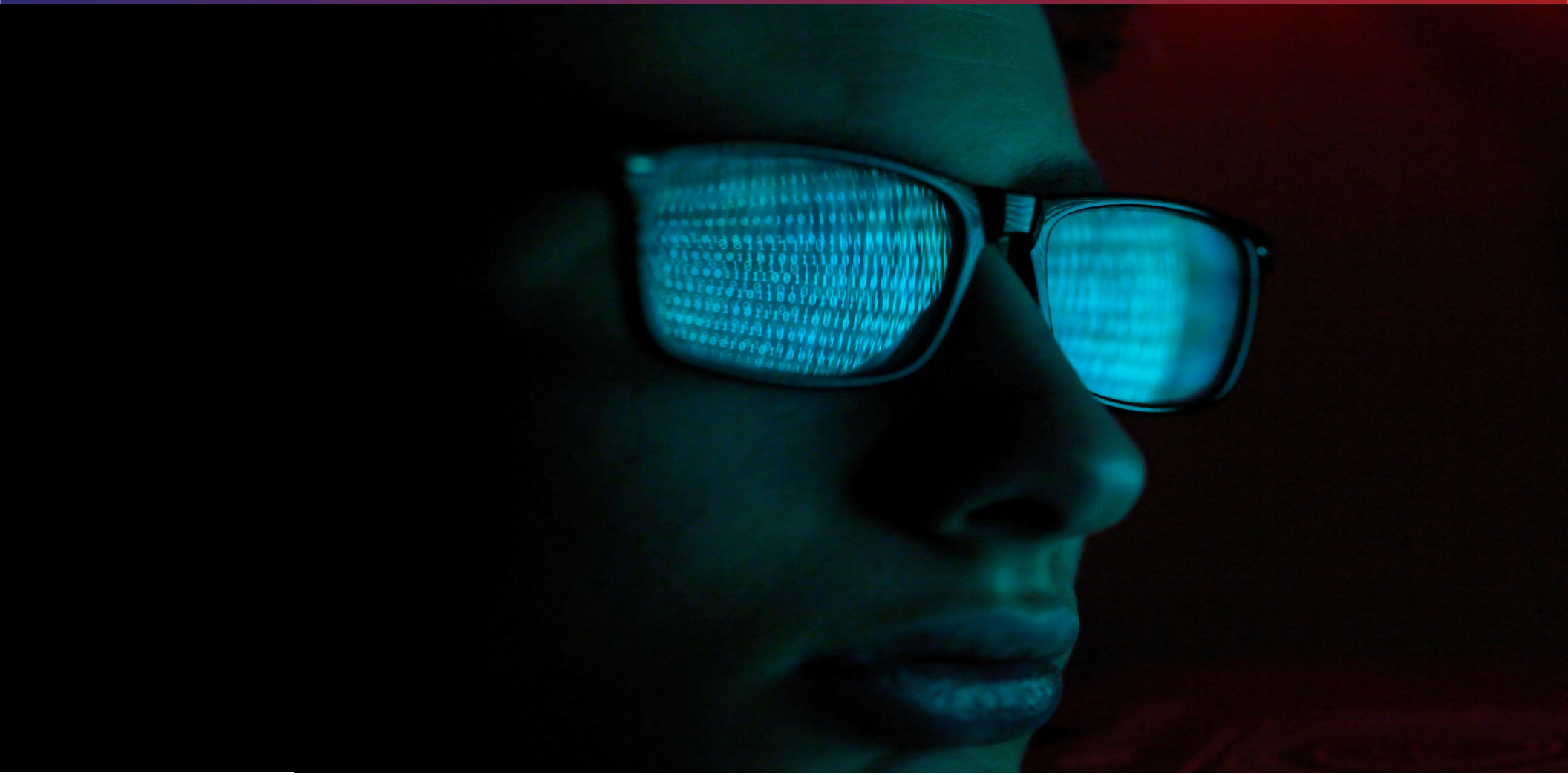
Rotten Cause Analysis - why coding education must change

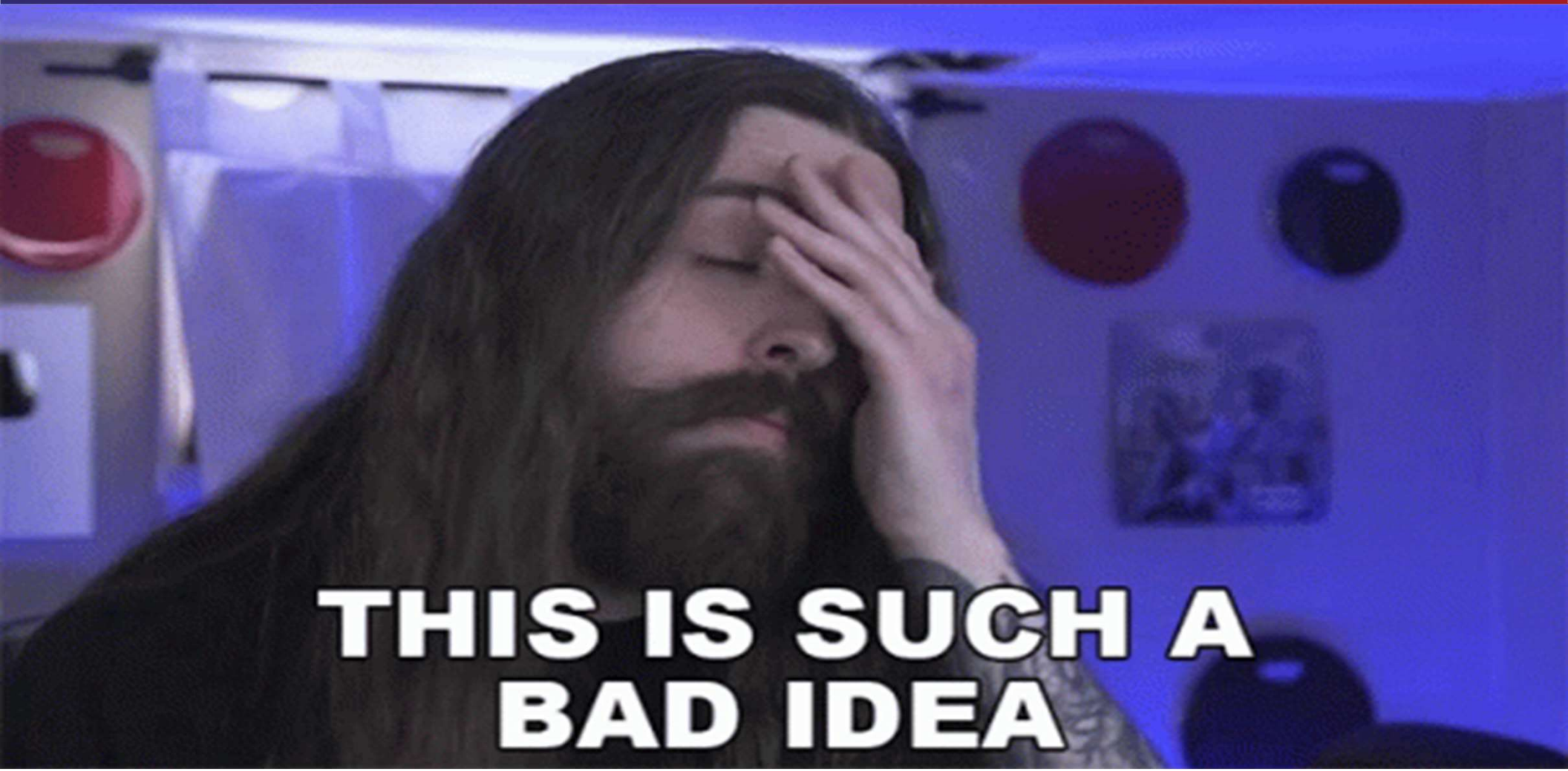


What We're About



Rotten Cause Analysis - why coding education must change

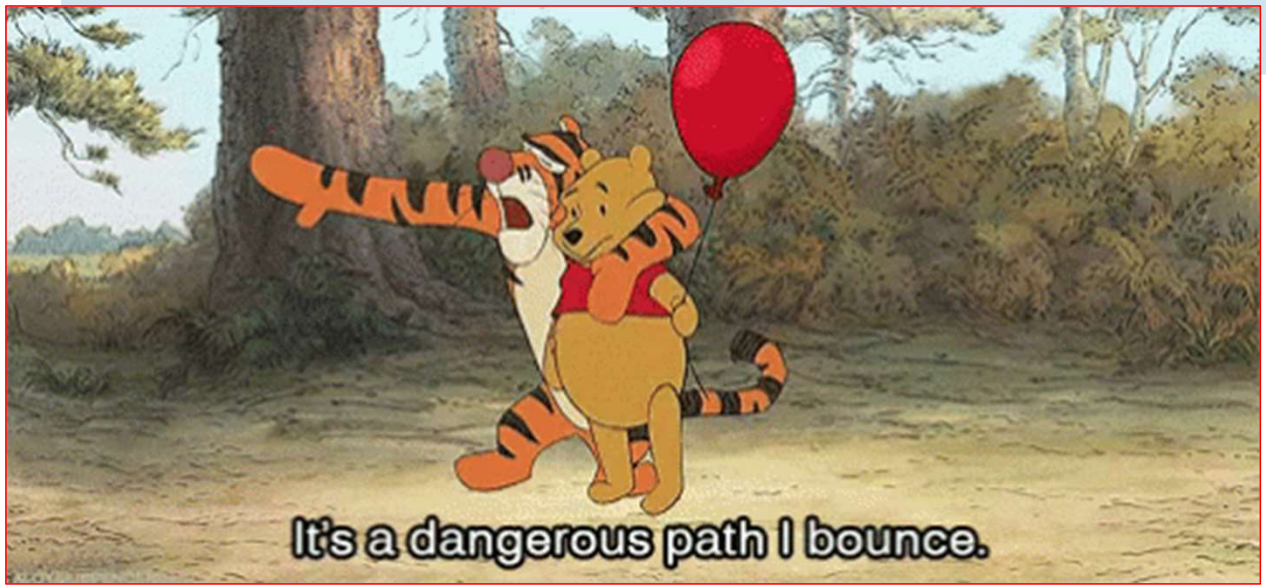



THIS IS SUCH A
BAD IDEA

Developer Career Path

Tabula Rasa → Developer → → → → Repeat

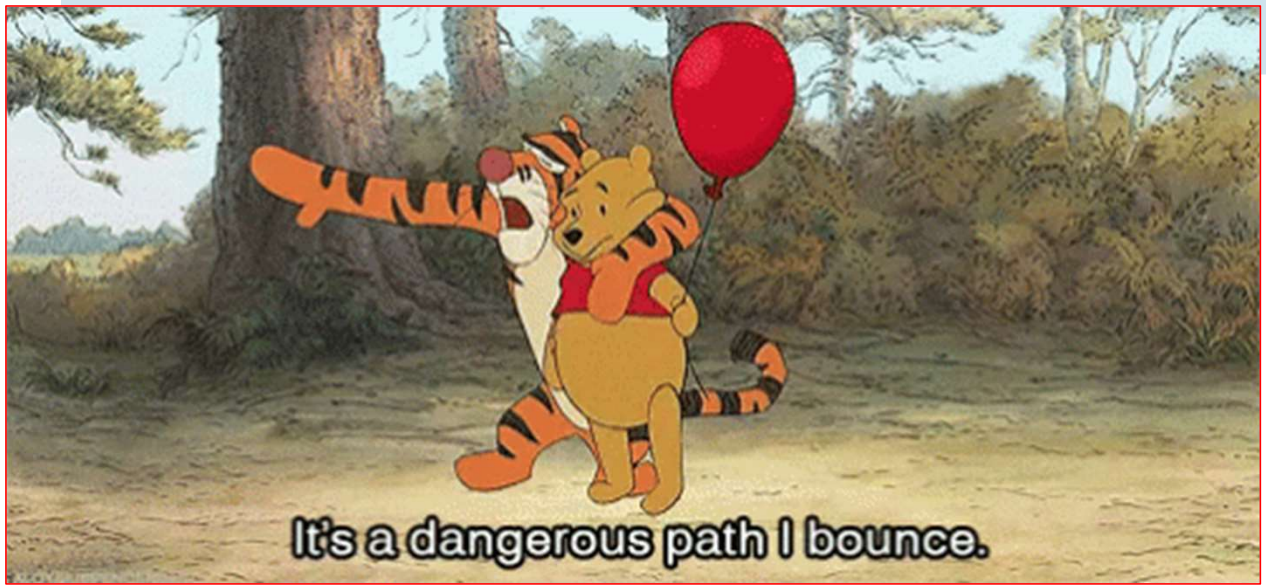
Learn Programming | **Job Training** | **Work** | **More Training** | **Work**



Secure (?) Developer Career Path

Tabula Rasa → Developer → → → → Repeat

Learn Programming | **Job Training** | **Work** | ***Security* Training** | **Work**



Really Secure (?) Developer Career Path

Tabula Rasa ← → Developer → → → Repeat ↻

Learn Programming | **Job Training + *Security* Training** | **Work** | **Work** | **Work**

SECURE Developer Career Path

Tabula Rasa ← → Developer → → → Repeat ↻

Learn *Secure* Programming | **Security Aware Job Training** | **Work** | **Security Refresher** | **Work**

Or Sahar

Senior security researcher with F5

Secure code Instructor

Application security consultation and PT

A veteran developer

Drug of choice : CVEs & Snowy mountains

Yariv Tal

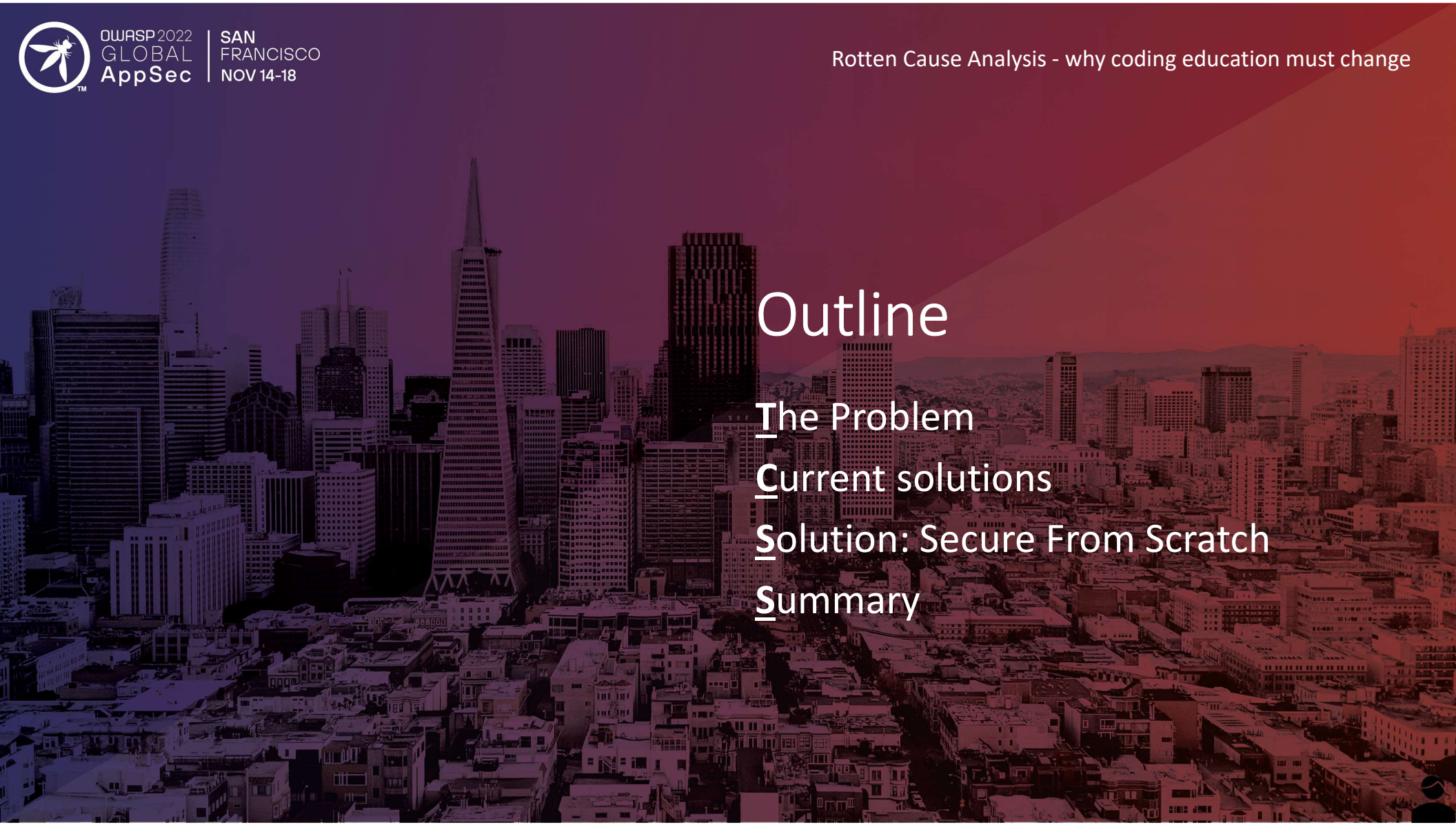
University lecturer

Bootcamps mentor

Seasoned developer (WhyT software)

Application security researcher

Drug of choice: Travelling & Roller coasters



Outline

The Problem

Current solutions

Solution: Secure From Scratch

Summary

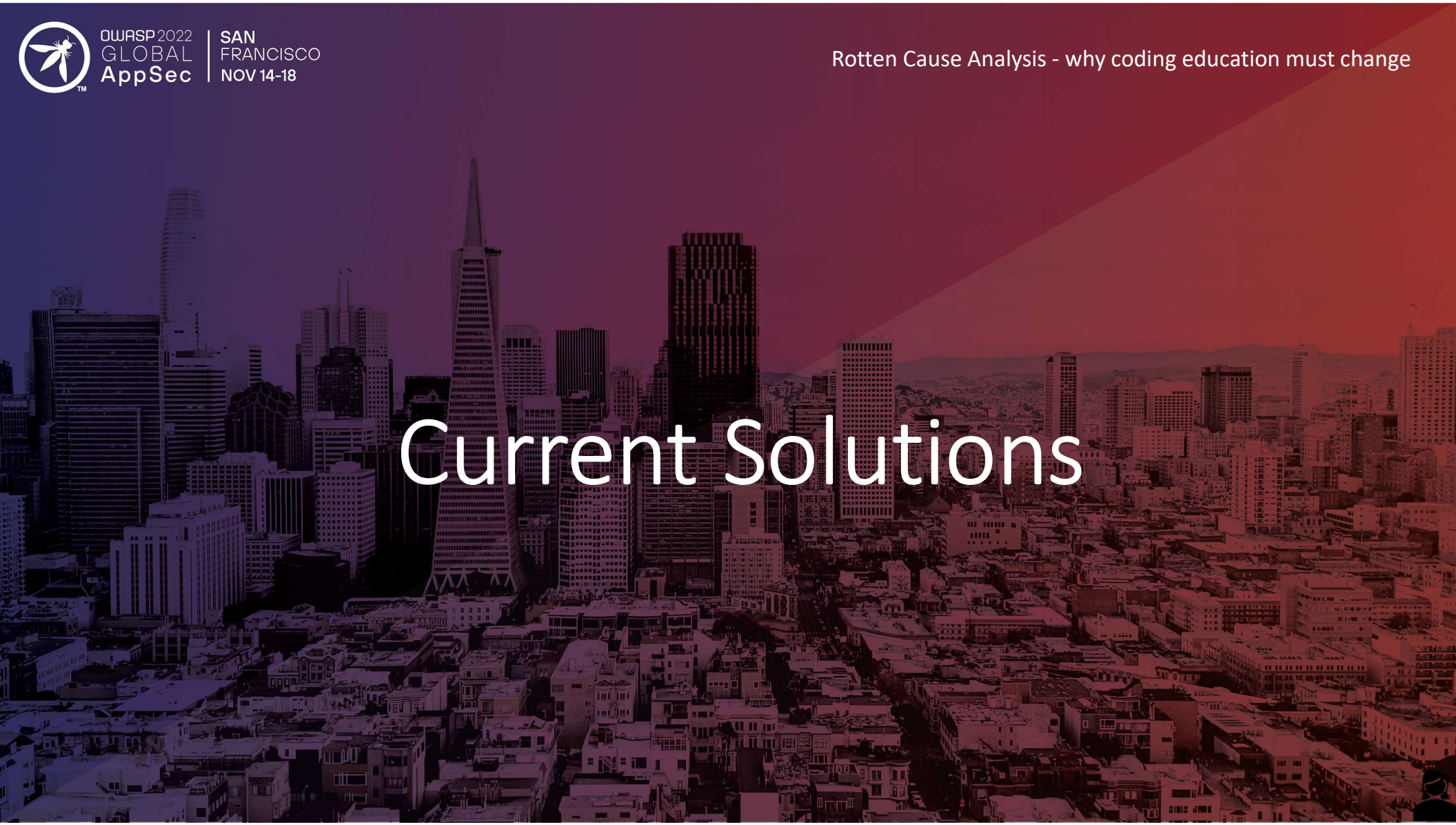
The Problem – Insecure Code

Problem

Code *is still* insecure!

- Enterprise level solutions
- Do they really solve?
- The same vulnerabilities again and again
- Non/Small software industry, schools

… The NVD database holds 8,051 vulnerabilities published in Q1 of 2022. This is about a 25 percent increase …



Current Solutions

Solutions

Current Solutions

Solutions > Cover It

Tools Post-Defect

SAST	DAST	SCA
WAF	API Gateway	Vulnerability Scanner
Bug Bounty Programs	PT	Supply Chains

Solutions → Cover It

AppSec Professional

Threat modeling	Code reviews	Teaches secure coding
Tools integration	Tools configuration	Tools interpretation
PT Coordinator	Secure CI/CD	Security Compass

The ratio … to Software Developers … approximately between 1:80 and 1:100.

Solutions > Do It Right

New Languages & Language Extensions

- Compiler Checked Memory Access
- Default immutability
- Default non-inheritable
- Sensitive Information Output Control

Solutions > Do It Right

Educate

Security

Secure C

OWASP Kn

Insecure D

The proportion of implementation vulnerabilities for 2008- 2016 is close to the 64% reported for 1998 - 2003

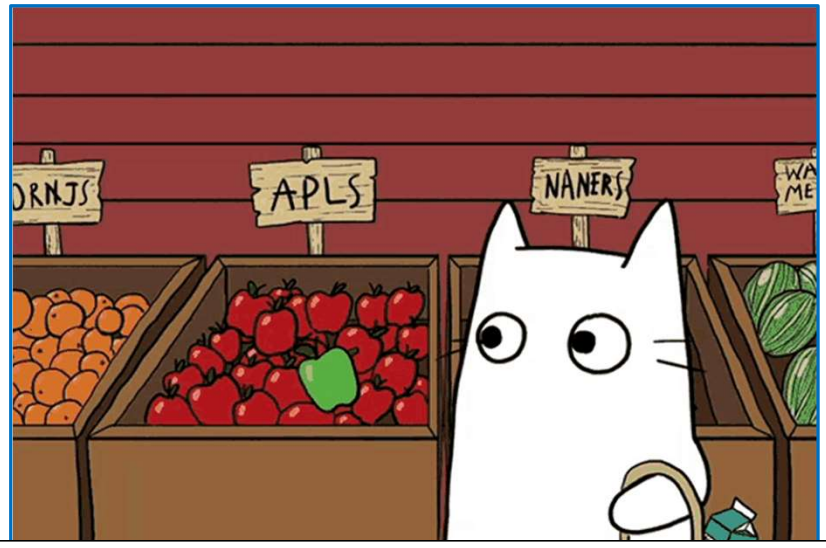
… The high proportion of implementation errors suggests that little progress has been made in reducing these vulnerabilities that result from simple mistakes …

We're Failing

Solutions > Do It Right

Why Education Fails

- Developers Aren't There
- Too Late
- Too Much I
- Separate Su

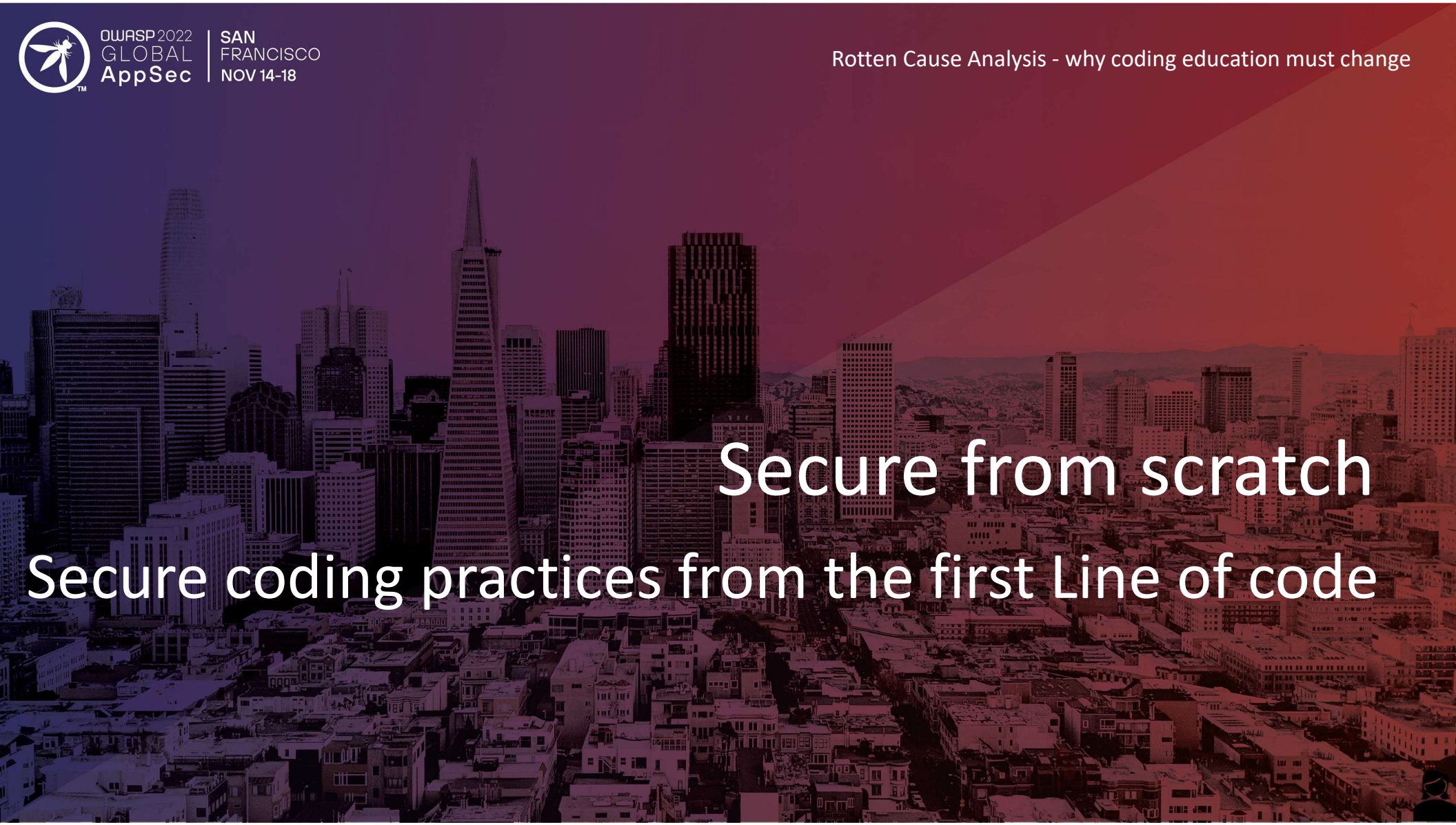


In the flask web server platform when calling the save() method to save an uploaded file to disk with a client supplied filename you must call the secure_filename method on the client supplied filename.

Solutions > Do It Right

Why Education Fails

- Developers Aren't There
- Too Late
- Too Much Information
- Separate Subject


OWASP 2022
GLOBAL
AppSec
SAN
FRANCISCO
NOV 14-18
Rotten Cause Analysis - why coding education must change
Secure from scratch
Secure coding practices from the first Line of code

SecureFromScratch

Secure from scratch into Practice

- When
- Where
- How

SecureFromScratch > When

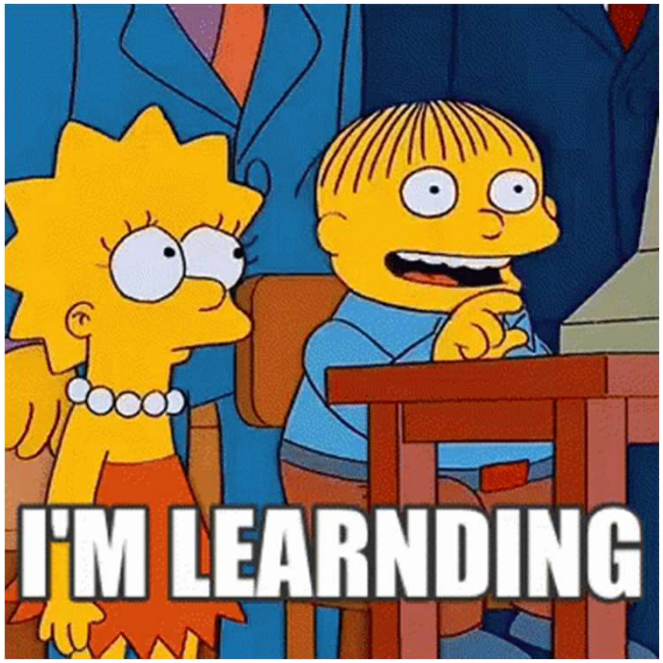
Always

- Secure From first line of code
- When there is time
- When there is no time
- In the beginning of one's career
- Before retirement

SecureFromScratch > Where

Any Learning Source

- Language/SDK docs
- 101 Fundamentals courses
- Tutorials, Coding sites
- Books
- Forums/Blogs/Communities
- Schools, University, Bootcamps



SecureFromScratch > How

Put Security First In Education Materials

Mind	Write with security mindset
Ask	Is It Safe?
Best	Mention secure code best Practices
Habits	Practice secure habits by default
Suggest	Suggest alternative to risky APIs

```
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
~# Hello Secure World
```

SecureFromScratch > How > Lang. Spec

Good Random

Python random– A warning when a functionality might put the application at risk

`random` — Generate pseudo-random numbers¶

Source code: Lib/random.py

This module implements pseudo-random number generators for various distributions.

For integers, there is uniform selection from a range. For sequences, there is uniform selection of a random element, a function to generate a random permutation of a list in-place, and a function for random sampling without replacement.

On the real line, there are functions to compute uniform, normal (Gaussian), lognormal, negative exponential, gamma, and beta distributions. For generating distributions of angles, the von Mises distribution is available.

Almost all module functions depend on the basic function `random()`, which generates a random float uniformly in the semi-open range [0.0, 1.0). Python uses the Mersenne Twister as the core generator. It produces 53-bit precision floats and has a period of 2**19937-1. The underlying implementation in C is both fast and threadsafe. The Mersenne Twister is one of the most extensively tested random number generators in existence. However, being completely deterministic, it is not suitable for all purposes, and is completely unsuitable for cryptographic purposes.

The functions supplied by this module are actually bound methods of a hidden instance of the `random.Random` class. You can instantiate your own instances of `Random` to get generators that don't share state.

Warning: The pseudo-random generators of this module should not be used for security purposes. For security or cryptographic uses, see the `secrets` module.

Warning: The pseudo-random generators of this module should not be used for security purposes. For security or cryptographic uses, see the `secrets` module.

SecureFromScratch > How > Lang. Spec

Bad Random

nodejs random() - There is no indication that this random method is not secure!

Math.random()

The `Math.random()` function returns a floating-point, pseudo-random number that's greater than or equal to 0 and less than 1, with approximately uniform distribution over that range — which you can then scale to your desired range. The implementation selects the initial seed to the random number generation algorithm; it cannot be chosen or reset by the user.

Try it

JavaScript Demo: Math.random()

```
1  function getRandomInt(max) {
2    return Math.floor(Math.random() * max);
3  }
4
5  console.log(getRandomInt(3));
6  // expected output: 0, 1 or 2
7
8  console.log(getRandomInt(1));
9  // expected output: 0
10
11 console.log(Math.random());
12 // expected output: a number from 0 to <1
13
```

SecureFromScratch > How > Course

Secquality Over Quantity

First Coding Course

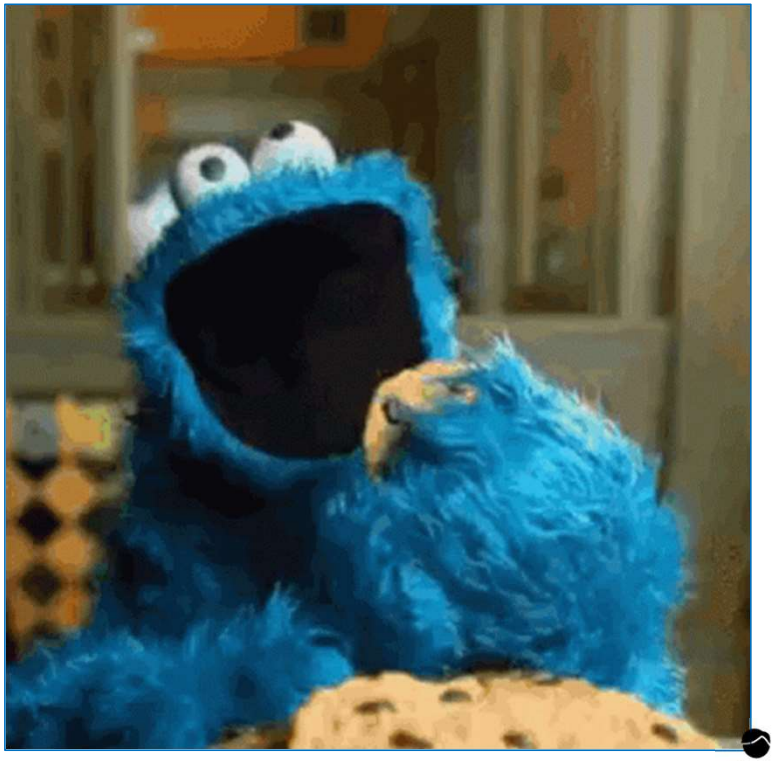
- Too early for security?
- Security is too hard?
- Instead of other subjects?
- Right time for habits
- Start with basic concepts
- Quality over Quantity

Create Good Habits

```
import secrets
dice1 = secrets.randbelow(6) + 1
dice2 = secrets.randbelow(6) + 1
print("Your throw is: ",dice1, " : ", dice2)
```

Form a Habit: use 'secrets' for random numbers

Prevent a bad habit:
Avoid string concatenation

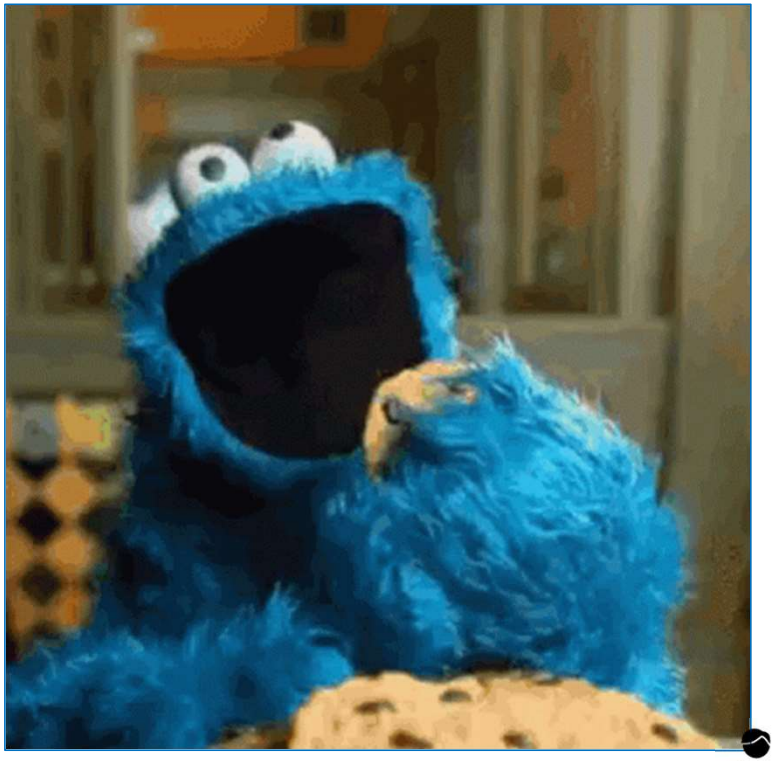


Create Good Habits

```
import secrets
dice1 = secrets.randbelow(6) + 1
dice2 = secrets.randbelow(6) + 1
print("Your throw is: ",dice1, " : ", dice2)
```

Form a Habit: use 'secrets' for random numbers

Prevent a bad habit:
Avoid string concatenation



SecureFromScratch > How > Course

NOT THIS!

```
import random
dice1 = random.randnit(1, 6)
dice2 = random.randint(1, 6)
print("Your throw is: " + dice1 + " : " + dice2)
```

!!! Possibly leads to CWE-330: Use of insufficiently random values

!!! String concatenation might lead to future injection

SecureFromScratch > How > Course

Teaching File Access

```
path = os.path.realpath(path)
if os.path.commonprefix((path, safe_dir)) != safe_dir:
    raise PermissionError()

with open(path, 'w') as f:
    f.write(str(result))
```

Simple Concept:
Canonicalization

Form a habit:
Validate path

Form a habit:
Automatic resource release

SecureFromScratch > How > Course

Teaching File Access

```
path = os.path.realpath(path)
if os.path.commonprefix((path, safe_dir)) != safe_dir:
    raise PermissionError()

with open(path, 'w') as f:
    f.write(str(result))
```

Simple Concept:
Canonicalization

Form a habit:
Validate path

Form a habit:
Automatic resource release

SecureFromScratch > How > Course

Teaching File Access

```
path = os.path.realpath(path)
if os.path.commonprefix((path, safe_dir)) != safe_dir:
    raise PermissionError()

with open(path, 'w') as f:
    f.write(str(result))
```

Simple Concept:
Canonicalization

Form a habit:
Validate path

Form a habit:
Automatic resource release

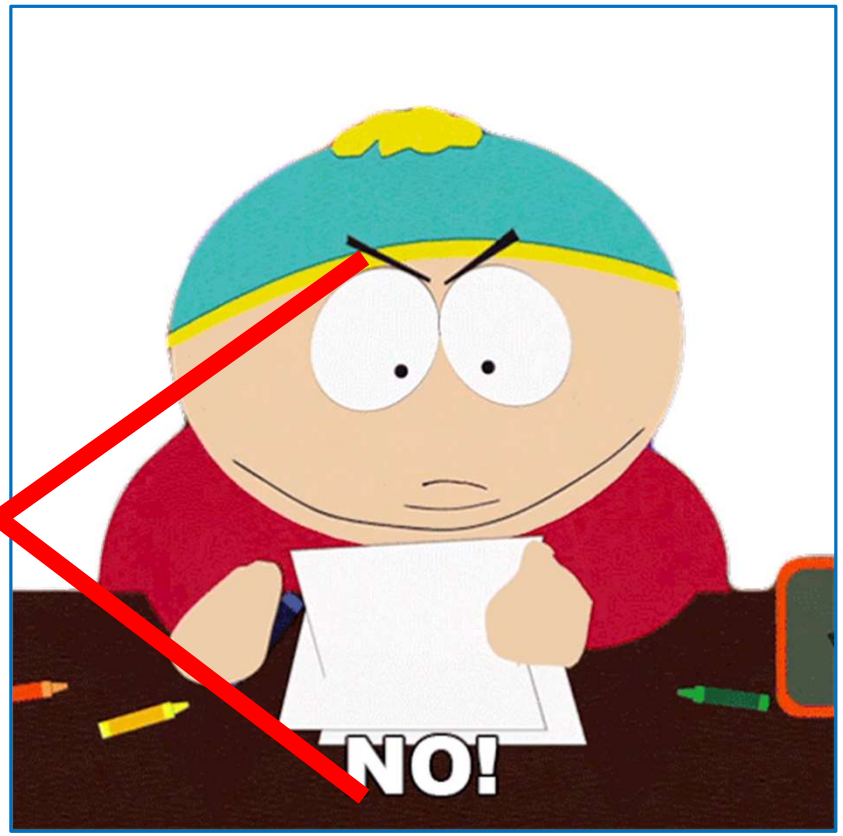
SecureFromScratch > How > Course

NOT THIS!

!!! Direct use of path might lead to path traversal

```
f = open(path, 'w')
f.write(str(result})
f.close()
```

!!! Manual release of resources. Might lead to CWE 772.



Be Weary of Ignoring Security Issues

```
password = input('Please enter password ')
If password == 'letmein':
    print('hello')
```

Highlight Insecurities: Hard coded secret within code, plain-text password

3.4K occurrences of secrets detected per AppSec engineer in 2021

Java – Same Same, But Different

Form a habit: Use final

```
final int num = SecureRandom
        .getInstanceStrong().nextInt();

final int sqrd = num * num;
```

Highlight Insecurities: Wrap-around

SecureFromScratch > How > Course

Java – Same Same, But Different

Form a habit: Use final

```
final int num = SecureRandom
        .getInstanceStrong().nextInt();

final int sqrd = num * num;
```

Highlight Insecurities: Wrap-around

SecureFromScratch > How > Course

Talk About 3rd Party Security

A06:2021-Vulnerable and Outdated Components was

Vulnerabilities and is #2 in the Top 10 community survey

data analysis. T gory m

Quickstart: Install a

NuGet package in

(Windows only)

```
<dependencies>
  <dependency>
    <groupId>jun
    <artifactId
    <version>4.1
    <scope>test
  </dependency>
</dependencies>
```

CWE-1104: Use of Unmaintained Third Par

Weakness ID: 1104
Abstraction: Base
Structure: Simple

Presentation Filter: Complete

Description

The product relies on third-party components that are not actively sup

SecureFromScratch > How > Tutorial

Secure Connection String

Tutorial: Get started with EF Core in an ASP.NET MVC web app

Trusted connection

No clear text

Password as secret

```
public void ConfigureServices(IServiceCollection services)
{
    services.AddDbContext<SchoolContext>(options =>
        options.UseSqlServer(Configuration.GetConnectionString("DefaultConnection")));

    services.AddControllersWithViews();
}
```

JSON Copy

```
{
  "ConnectionStrings": {
    "DefaultConnection": "Server=(localdb)\\mssqll...;Database=ContosoUniversity1;Trusted_Connection=True;
  },
```

.NET CLI

```
dotnet user-secrets set "DbPassword" "pass123"
```

SecureFromScratch > How > Tutorial

Clear text Creds!?

Uber hack linked to hardcoded se… spotted in PowerShell script

John Leyden 16 September 2022 at 15:26 UTC
Updated: 21 September 2022 at …

1,859 Android and iOS apps were containing hard-coded Amazon AWS credentials

September 1, 2022 By Pierluigi Paganini

The State of Secrets Sprawl 2022

In its 2022 report, GitGuardian extends its previous edition focused on public GitHub by depicting a realistic view of the state of secrets sprawl in corporate codebases.

Connect to your MongoDB Atlas cluster

Next, we need to connect to the MongoDB Atlas cluster we created earlier. Locate your connection string and add it to the `.env` file. Replace `<username>` and `<password>` with your credentials.

pymongo-fastapi-crud/.env

```
ATLAS_URI=mongodb+srv://<username>:<password>@sandbox.lqlql.mongodb.net/?retryWrites=true&w=m
DB_NAME=pymongo_tutorial
```

CWE-312: Cleartext Storage of Sensitive Information

Weakness ID: 312
Abstraction: Base
Structure: Simple

Presentation Filter: Complete

Description

The application stores sensitive information in cleartext within a resource that might be access…

Security Vulnera… …to CWE-312

CVSS Scores Greater Than: 0 1 2 3 4 5 6 7 8 9
Sort Results By : CVE Number Descending CVE Number Ascending CVSS Score Descending
Copy Results Download Results

#	CVE ID	CWE ID	# of Exploits	Vulnerability Type(s)	Publish Date	Update Date	Score
1	CVE-2021-37157	312			2021-11-10	2021-11-12	9.0
	An issue was discovered in OpenGamePanel OGP-Agent-Linux through 2021-08-14. $H						
2	CVE-2020-5805	312			2021-01-08	2021-01-14	9.0
	In Marvell QConvergeConsole GUI <= 5.5.0.74, credentials are stored in cleartext in t… use QCC may use the plaintext credentials to login to QCC.						
3	CVE-2019-11966	312			2019-06-05	2020-08-24	9.0
	A remote privilege escalation vulnerability was identified in HPE Intelligent Manageme…						
4	CVE-2018-19981	312		Bypass	2019-04-04	2021-05-10	9.0
	Amazon AWS SDK <=2.8.5 for Android uses Android SharedPreferences to store plain…						

SecureFromScratch > How > Tutorial

Basic Authentication!?

Flask
web development, one drop at a time

+

neo4j

Developer Guides / Getting Started with Neo4j and Flask

Python

```
driver = GraphDatabase.driver("bolt://[illegible]72.13.205:47929", auth=basic_auth("neo4j", "knock-cape-reserve"))
```

Then, in your text editor, open and create `flask-api/.env` and enter the appropriate information for `DATABASE_USERNAME`, `DATABASE_PASSWORD`, and `DATABASE_URL`. Then save the file.

```
DATABASE_USERNAME = 'your usernamer'
DATABASE_PASSWORD = 'your password'
DATABASE_URL = 'your URL'
```

To start the Flask API, run:

Bitdefender

CONSUMER INSIGHTS | LABS | BUSINESS INSIGHTS

INDUSTRY NEWS · 1 min read ·

Microsoft Announces Official Death of Basic Auth Officially on Oct. 1, 2022

SecureFromScratch > How > Tutorial

SQLite in a blog?

- No encryption for data at rest or in transit
- Size is limited
- No scaling
- No network access
- No concurrent users
- No user management

guides.rubyonrails.org/getting_started.html

3.1.2 Installing SQLite3

You will also need an installation of the SQLite3 database. acceptable version of SQLite3. Others can find installation

Verify that it is correctly installed and in your load `PATH`:

```
$ sqlite3 --version
```

Security guidelines in a different tutorial

The program should re

3.1.3 Installing Rails

To install Rails, use the `gem install` command provided

```
$ gem install rails
```

To verify that you have everything installed correctly, you s terminal:

```
$ rails --version
```

If it says something like "Rails 7.0.0", you are ready to con

SecureFromScratch > How

Put Security First In Communities

- Show the secure way first
- Avoid suggesting short cuts
- Reference security implications

Disable Security

"How do I disable Angular security?"

```
console.log(this.sanitized.bypassSecurityTrustHtml(value))
return this.sanitized.bypassSecurityTrustHtml(value);
```

There is no indication that bypassSecurityTrust* is dangerous!

DEMO : https://plnkr.co/edit/Qke2jktna55h40ubUI8o?p=preview

52

```
import { DomSanitizer } from '@angular/platform-browser'

@Pipe({ name: 'safeHtml'})
export class SafeHtmlPipe implements PipeTransform  {
  constructor(private sanitized: DomSanitizer) {}
  transform(value) {
    console.log(this.sanitized.bypassSecurityTrustHtml(value))
    return this.sanitized.bypassSecurityTrustHtml(value);
  }
}

@Component({
  ...
})
export class App {
  name:string
  html: safeHtml
  constructor() {
    this.name = 'Angular2'
    this.html = "<svg> ... </svg>";
  }
}
```

SecureFromScratch > How > Communities

Disable Security

"How do I disable Angular security?"

okta Developer

Bypassing Angular's security checks

🚨 Here be dragons! 🚨

Be careful about bypassing the built-in security mechanism; if you need to, read this section carefully:

What if you need to bind trusted values that Angular thinks are unsafe? You can mark values as trusted and bypass the security checks.

SecureFromScratch > How > Communities

Disable SSL?

The developer wants to ignore the certificate check.

Use the sample below from here

17

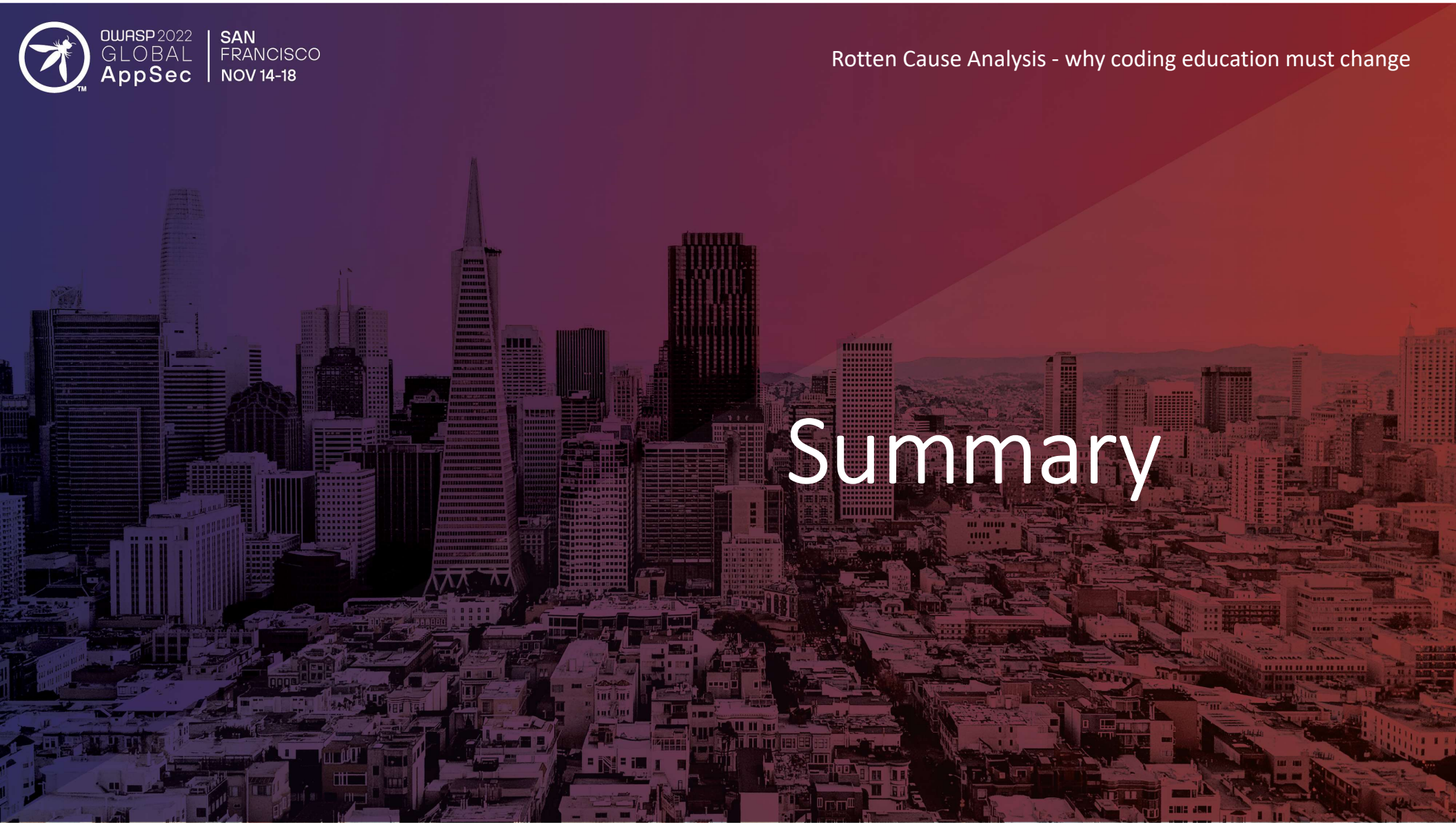
```
var httpClientHandler = new HttpClientHandler();
// Return `true` to allow certificates that are untrusted/invalid
httpClientHandler.ServerCertificateCustomValidationCallback =
    HttpClientHandler.DangerousAcceptAnyServerCertificateValidator;
var httpClient = new HttpClient(httpClientHandler);
```

73

Since there is only one global ServicePointManager, setting ServicePointManager.ServerCertificateValidationCallback will yield the result that all subsequent requests will inherit this policy. Since it is a global "setting" it would be prefered to set it in the Application_Start method in Global.asax.

Setting the callback overrides the default behaviour and you can yourself create a custom validation routine.

Security Warning?



Summary

Summary

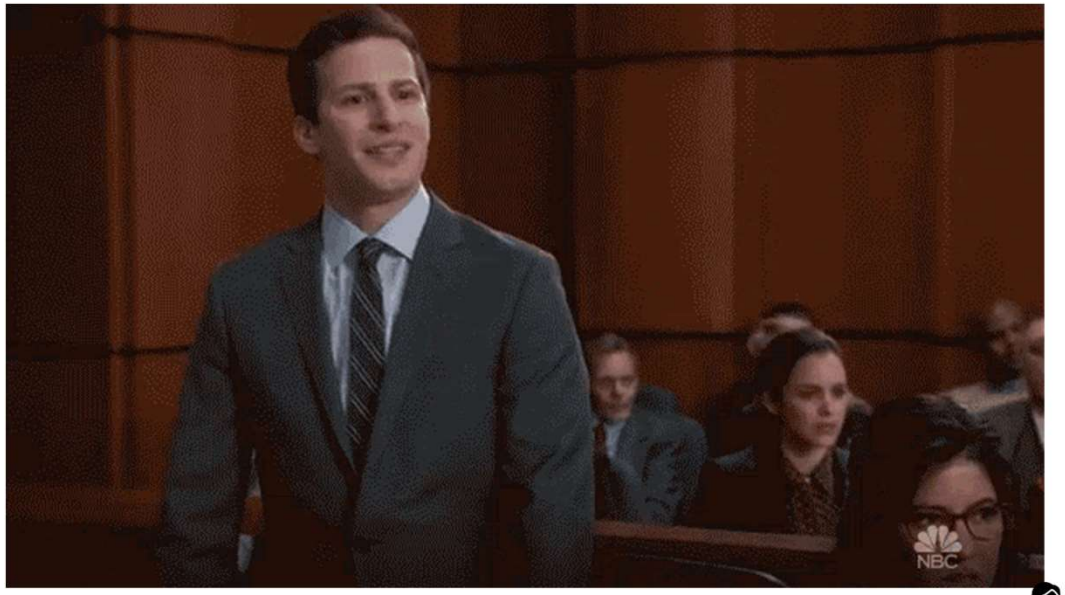
Summary

Problem	Insecure code
Solutions	Not enough, Post bad habits
Secure World	Shift education left
Examples	Course, Docs, Tutorials, Forum

Summary > OpenSource

github.com/SecureFromScratch/HelloSecureWorld

- Open source project
- Guidelines for educating
- Hello Secure World Course



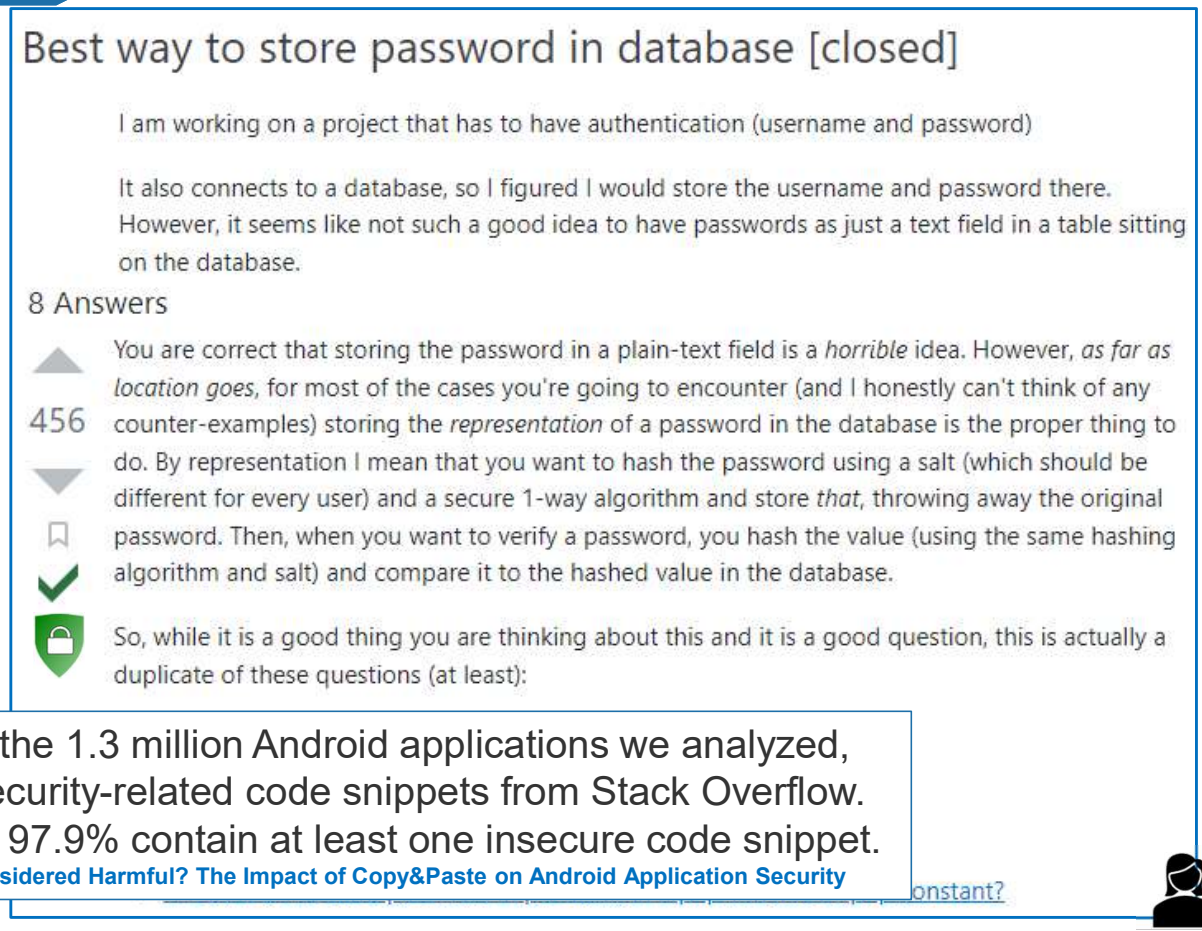
Summary > YOU!

What Can You Do?

Mng	**Secure coding as job requirement advantage**
Security Professional	**Be active in forums and coding sites**
Educator	**Change the way you teach**
Security Evangelist	**Contribute**

Summary | SecureV

The *Security Aware* Indicator



… 15.4% of the 1.3 million Android applications we analyzed, contained security-related code snippets from Stack Overflow. Out of these 97.9% contain at least one insecure code snippet.

Stack Overflow Considered Harmful? The Impact of Copy&Paste on Android Application Security

Looking to The Future

Change	Programming education will change
AppSec	Developers will understand you!
Join	Spread the word & Participate
Hello	Hello Secure World

Secure from Scratch

Yariv Tal **@YarivDevMentor**

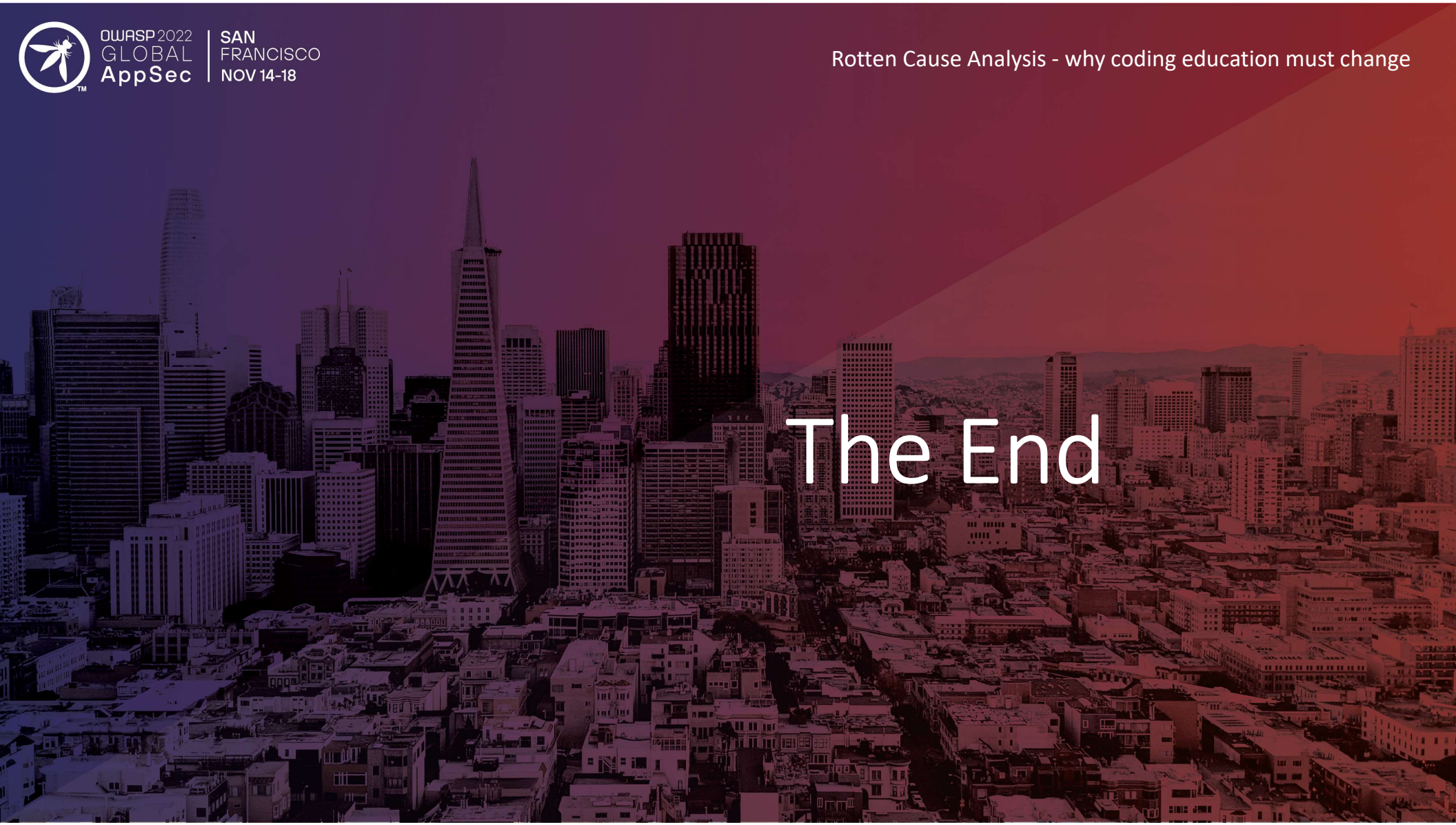
https://www.linkedin.com/in/yarivt/

Or Sahar **@securylight**

https://www.linkedin.com/in/securylight

Open Source GIT

https://github.com/SecureFromScratch/HelloSecureWorld



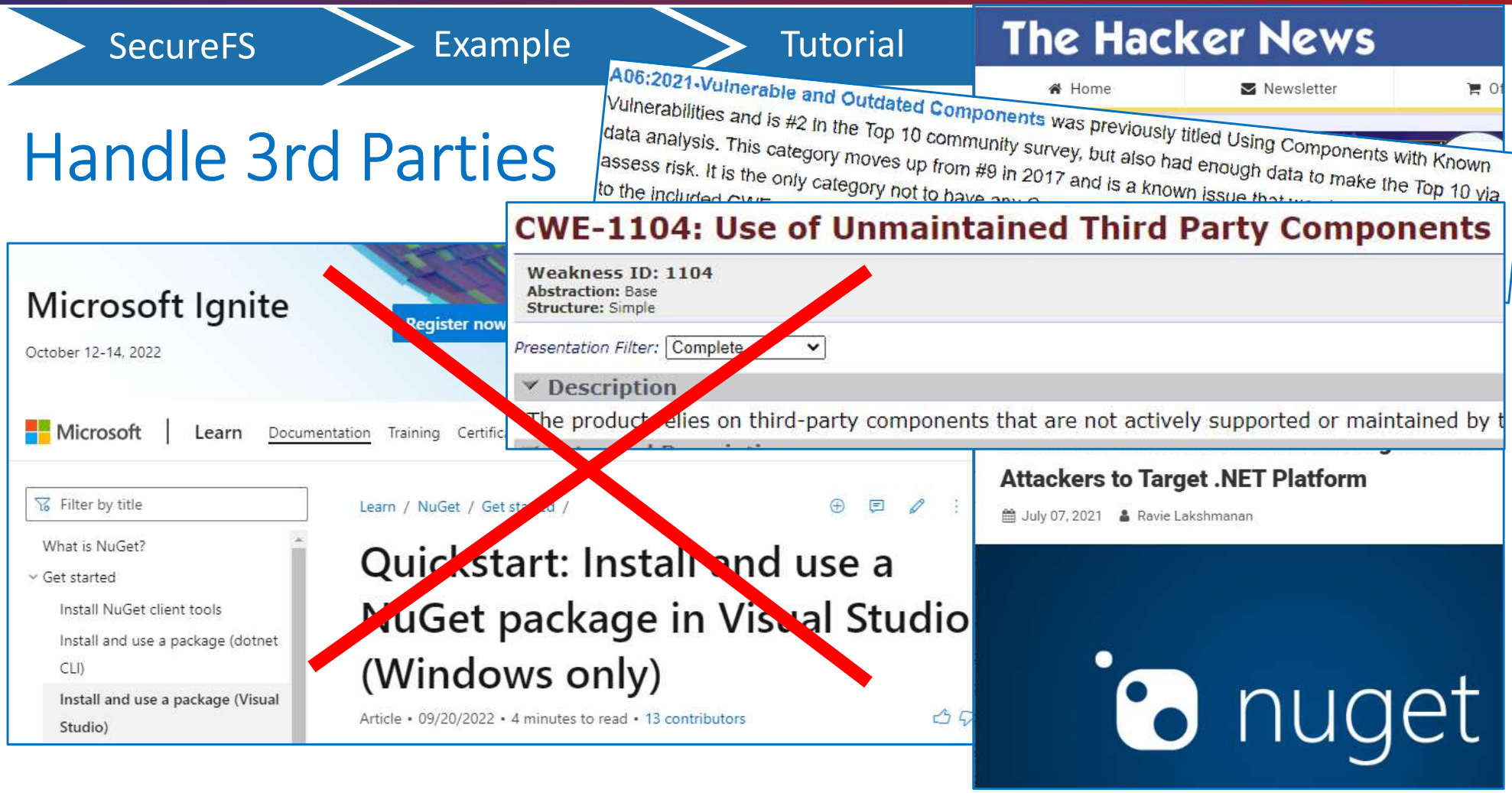
The End

Sample Guidelines for Educational Snippets

- Prefer showing the right way first
- Make secure code look good, insecure code look weird
- When faced with more than one option - default to the more secure one
- When using risky features – use them in a way that is either secure or demonstrates the insecurity immediately
- Always use language features that improve security
- Always follow secure coding guidelines and best practices
- Avoid suggesting short cuts
- Suggest alternative to risky APIs

SecureFS > Example > Tutorial

Handle 3rd Parties



Secure FS

Call For Action!

Join	Join Open-source project
Help	Help to promote the forum security indicator (the new V)
Come up	Come up with more ideas
Change	Change the way you teach
Change	Change the way you learn
HR	Secure programming as job requirement advantage