

expo IQA 25

MADRID
May 20th,
21st & 22nd
2025

expoqa.eu

0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1
0 1 0 0 1 1 1 1 0 0 1
1 0 1 1 0 0 0 0 0 1
0 0 0 1 0 0 0
1 1 1 0 1 1 1 0
1 1 0 1 1 0 0

Cybersecurity in Action!

Testing for a Secure Digital Future

#Masterclass #expoQA25

Sara Martínez

```

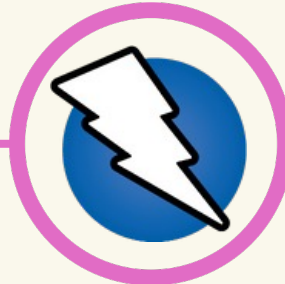
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 1 0 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 1 0 0 1 1 1 0 0
1 1 0 1 0 1 1 1 0 0
0 1 1 0 0 1 1 1 1
  
```

MAIN GOALS



SSDLC

Secure
Software
Development
Life
Cycle



PRESSURE

Be careful about new
vulnerabilities, risks
and attacks



AUTOMATION

Testing team goals:
Finding new security bugs



PROTECTION

Protect your
products from the
beginning

```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 1 1 1 1
  
```

AGENDA

- **CONTEXT: CODE, QUALITY, CYBERSECURITY, RISKS AND VULNERABILITIES**
- **QUALITY AND CYBERSECURITY CULTURE**
- **CODE ANALYSIS TOOLS**
- **HANDS ON! BREAK THE CODE**
- **SECURITY TEST AUTOMATION**
- **CI: GITHUB ACTIONS**

```
0 1 0 1 0 1 0 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 0 1
0 0 1 0 0 1 1 0 0 1
1 0 1 1 0 1 0 0
```

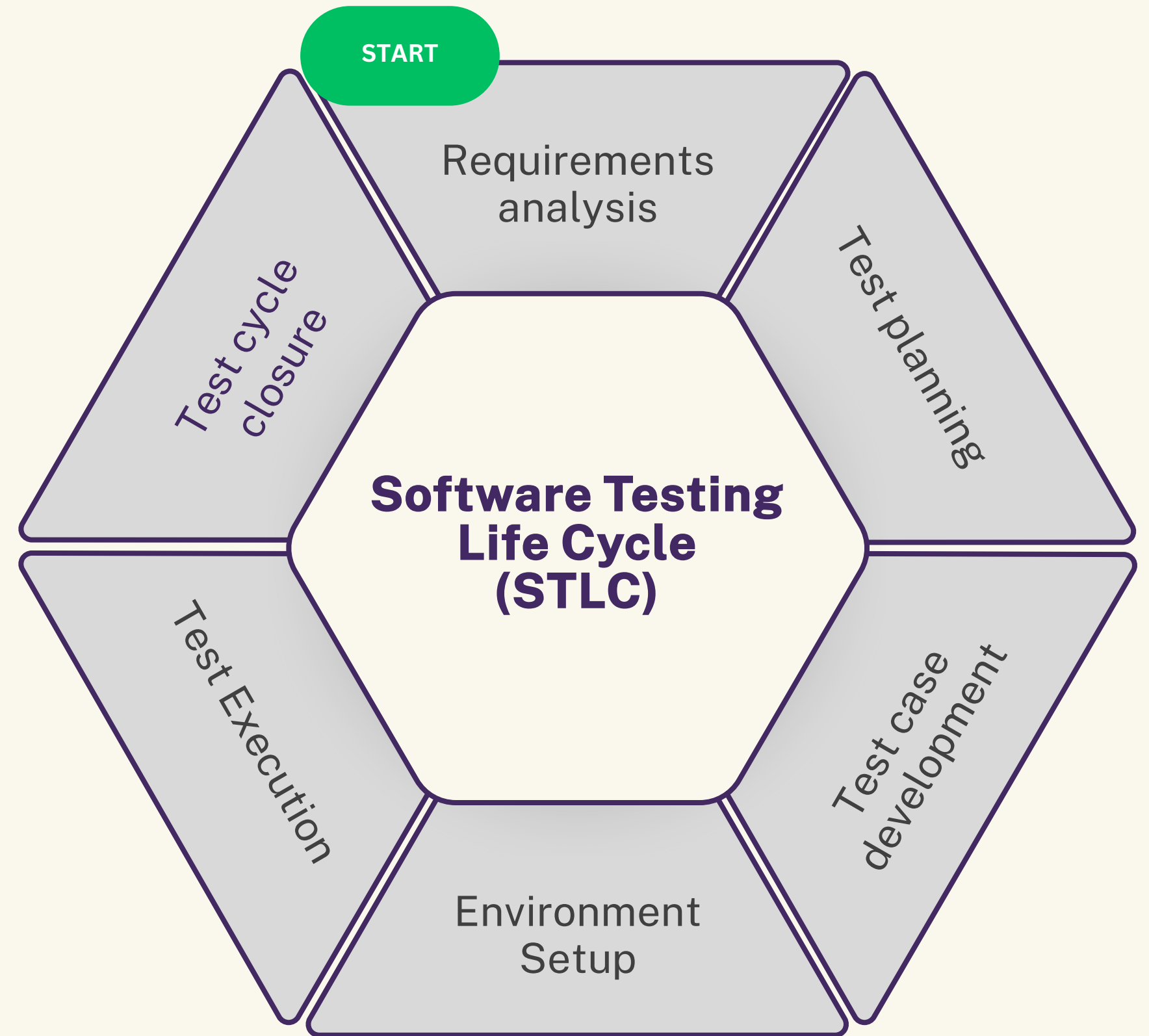
0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 1 0 0
1 1 1 0 1 1 0

A LITTLE BIT OF CONTEXT...

*....let's talk about code, quality,
cybersecurity, risks and vulnerabilities*

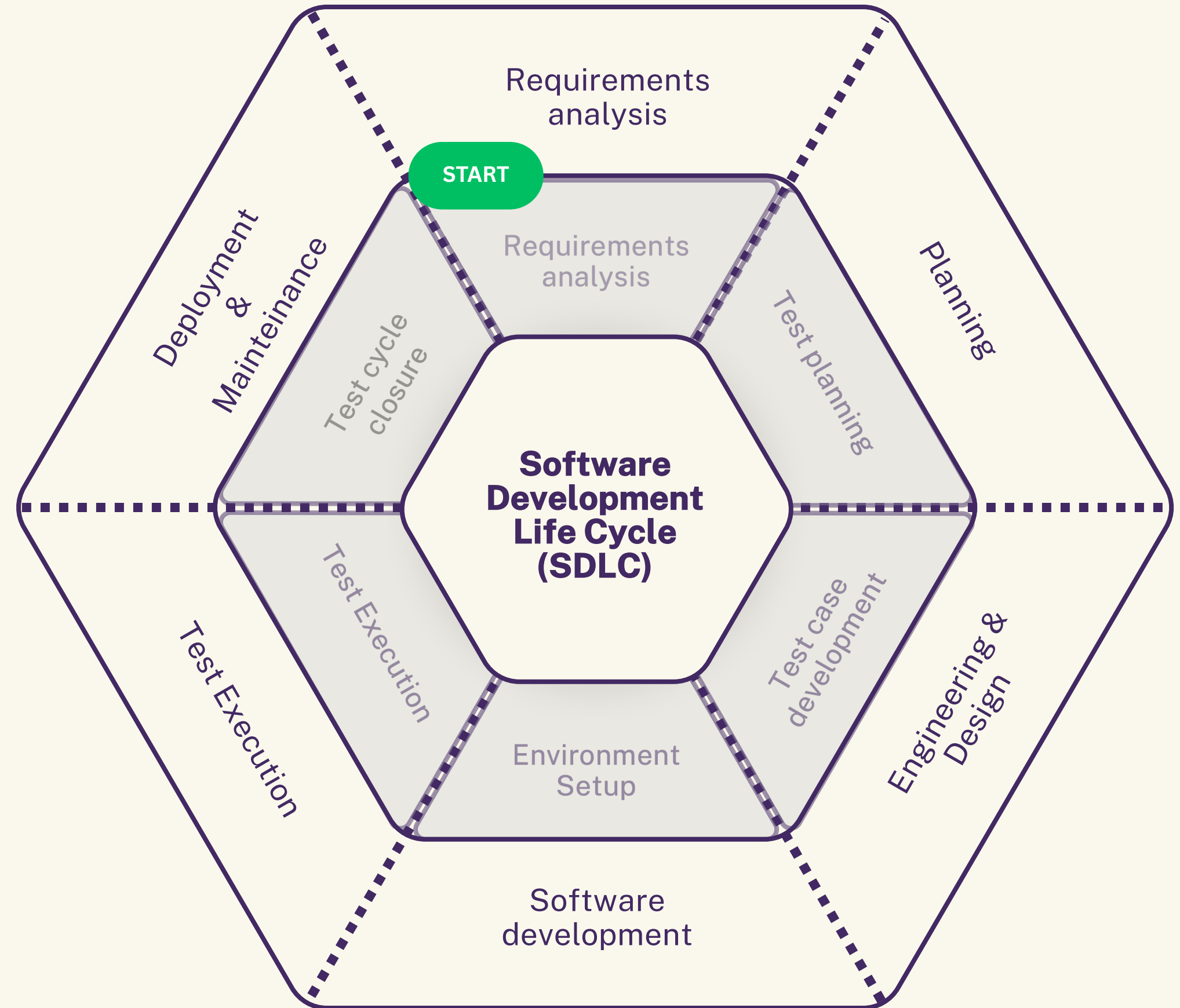
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
0 0 1 1 1 0 0 1 0 0 0

Software Testing Life Cycle



```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

Software Development Life Cycle



```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
  
```

CYBERSECURITY

«**Cybersecurity** is the **art** of **protecting** networks, devices, and data from unauthorized access or criminal use and the practice of **ensuring** confidentiality, integrity, and availability of information.»

Source: <https://www.cisa.gov/news-events/news/what-cybersecurity>

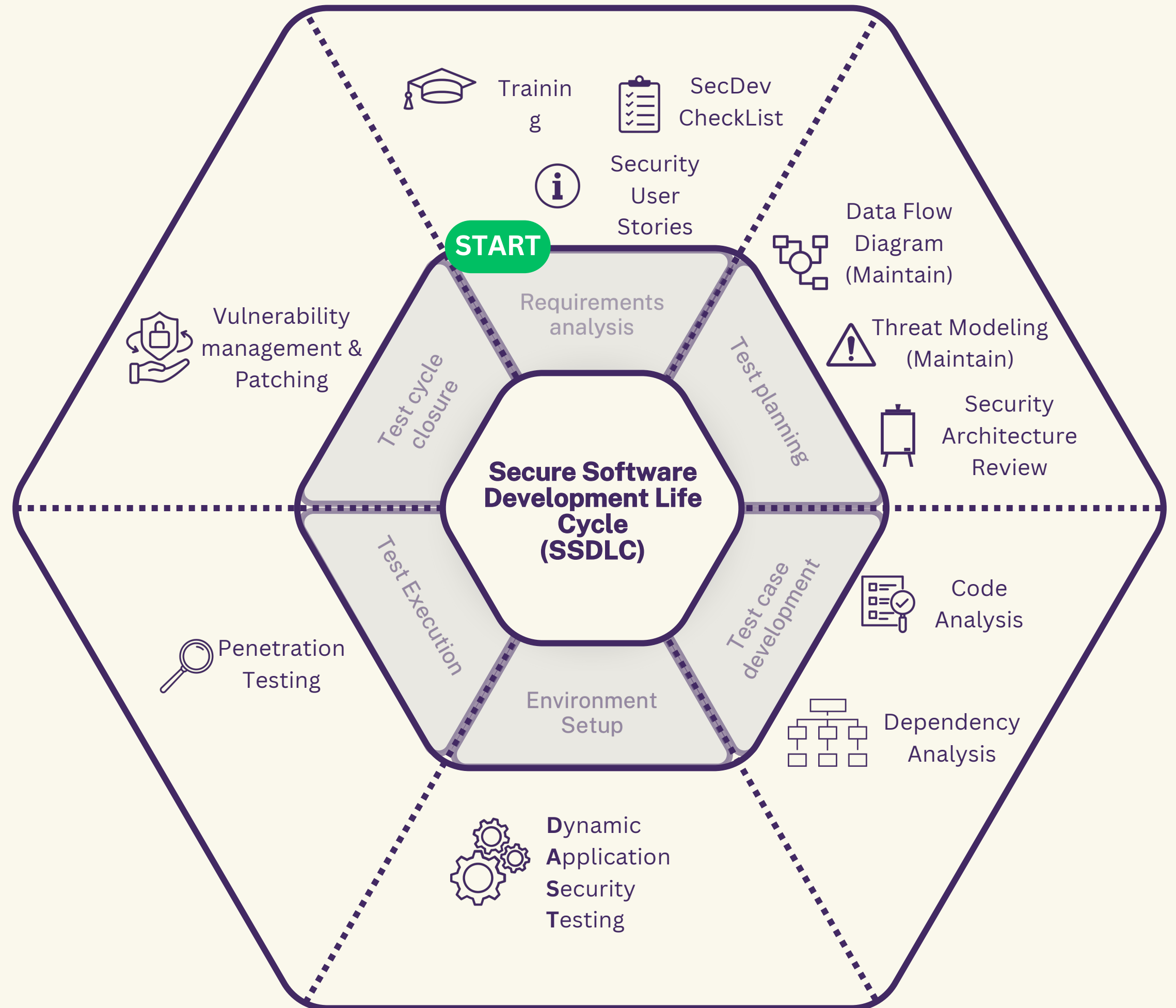


```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

SDLC Phase	STLC Phase	Security Activity
Requirements analysis	Requirements analysis	<ul style="list-style-type: none"> • Assess security risks and the threat landscape • Evaluate the potential impact of security incidents
Planning	Test planning	<ul style="list-style-type: none"> • Include security requirements • Incorporate regulatory compliance requirements
Engineering & Design	Test case development	<ul style="list-style-type: none"> • Develop threat models • Include security considerations in the architecture plan • Evaluate the impact of security on design decisions
Software development	Environment Setup	<ul style="list-style-type: none"> • Train developers on secure coding practices • Conduct static and dynamic application security testing • Evaluate software dependencies and reduce security risks
Test Execution	Test Execution	<ul style="list-style-type: none"> • Pentesting • Evaluate the security of the deployment environment • Review security configurations
Deployment	Test cycle closure	<ul style="list-style-type: none"> • Monitor the system for threats • Eliminate vulnerabilities and respond to intrusions

Source: <https://www.redhat.com/es/topics/security/software-development-lifecycle-security>

Secure Software Development Life Cycle



0 1 0 1 0 1 0 0 0 0 0 0 0
 0 0 1 0 1 0 1 1 0 1 0 1 0
 1 1 0 1 1 1 0 0 0 0 0 0 1
 1 0 0 0 0 0 1 0 1 0 1 1 1
 0 0 0 1 0 1 0 1 0 0 0 0 1
 0 1 1 0 1 0 1 0 1 1 1 0 0
 1 0 1 1 0 1 0 1 0 0 0 0 0
 0 1 1 0 0 0 1 0 1 0 1 1 1

If you are part of the cycle you will have **responsibility** for **quality** and **cybersecurity**.



```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1
```

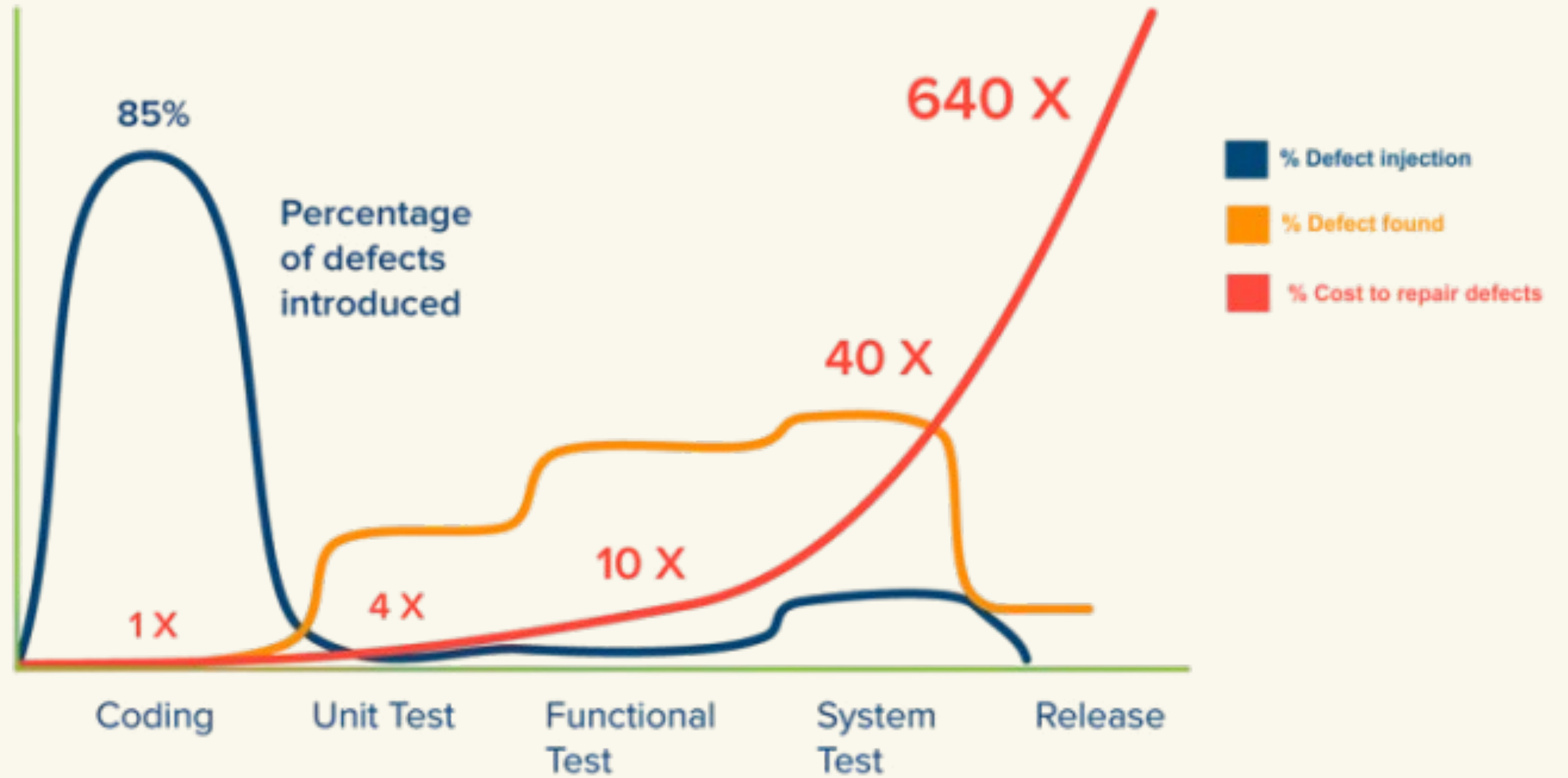
SOFTWARE DEVELOPER ENGINEER IN TEST



01. Development and testing skills.
02. It takes part in the entire product life cycle.
03. Testing pyramid
04. Knowledge focused on product stability, robustness and performance.
05. Focus on efficiency and automation.

```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1
```

SOFTWARE DEVELOPER ENGINEER IN TEST



```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
  
```

**WEAKNESSES
VULNERABILITIES
RISKS
THREATS**



```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 1 0 1
1 1 0 1 0 1
0 0 1 0 1
1 1 0 1 0

```

CWE

Common Weakness Enumeration (CWE) is a universal online dictionary of weaknesses that have been found in computer software.

TOP CWE OCCURRENCE

- Implementation 85.71%
- Architecture & Design 57.2%
- Operation 17.9%
- Installation 7.2%



Source: <https://waverleysoftware.com/blog/top-software-vulnerabilities/>

Source: <https://cwe.mitre.org>

```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

```

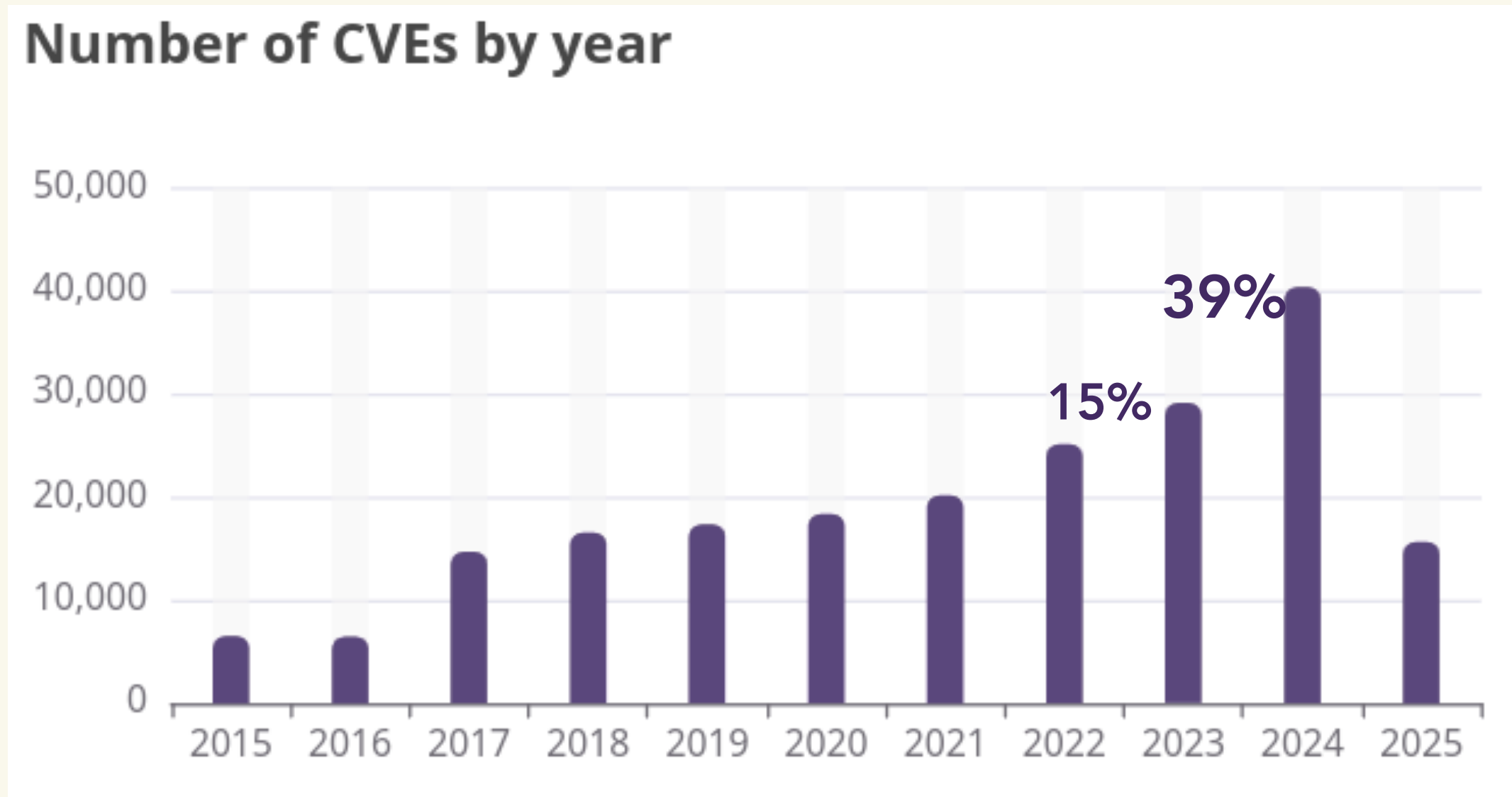
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 1 0 1 1

```

CVE

Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information-security vulnerabilities and exposures.

Source: <https://www.cvedetails.com>



```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0 0 1 1
1 1 0 1 0 1 1 1
0 0 1 0 1 1 0 0

```


CVE

Common Vulnerabilities and Exposures (CVE) system provides a reference method for publicly known information-security vulnerabilities and exposures.

Source: <https://www.theverge.com/news/649314/cve-mitre-funding-vulnerabilities-exposures-funding>

NEWS

The CVE program for tracking security flaws is about to lose federal funding



/ Financial support for the system that tracks publicly disclosed cybersecurity vulnerabilities expires on April 16th.

by Emma Roth
Apr 15, 2025, 10:41 PM GMT+2

17 Comments (17 New)

Illustration by Carlo Cadenas / The Verge

```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0 1
0 1 0 0 1 1 1 1 0 0 1 0 1 1
1 0 1 1 0 0 0 0 0 0 0 1
0 0 1 1 0 1 0 1 0 1 1 0
```



```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1

CrowdStrike Chaos Highlights Key Cyber Vulnerabilities with Software Updates

Veeam Software Vulnerabilities Let Attackers Trigger Remote Code Execution

Third-Party Cyber Risk Impacts the Health Care Sector the Most. Here's How to Prepare.

How to Secure Open Source Software: The Dilemma of the XZ Utils Backdoor

Microsoft Employees Data Exposed Via Third-Party Breach

CosmicBeetle Exploiting Old Vulnerabilities To Attack SMBs All Over The World

BREAKING NEWS!

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

RIPE Account Hacking Leads to Major Internet Outage

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
    1 1 0 1 1
        1 0 1 1
            0 0
    
```



The screenshot shows a 'Profile' page from a security tool. It includes the following information:

- Export Credentials** and **Export Cookies** buttons.
- Computer Name**: diego
- Operating System**: Windows 10 Pro [x64]
- Anti Virus**: Not Found
- Facebook** and **Virtualization** status.
- Initial Detection**: 2023-09-04 15:32:21 (Detected 1 time)
- Applications found**: auth, confluence, dnf, owa
- Installed software**: A list of application icons.
- Employee password reuse identified**: Warning icon.
- Machine ID**: ES_2023_09_04_11_14_sh2t5n
- Stealer Family**: Racoon
- IP Address**: 83.3...
- Malware Path**: Not Found
- Date Compromised**: 2023-09-04 00:00:00
- Latest Detection**: 2023-09-04 15:32:21
- Corporate Credentials Found: 78**
- URL**: https://access.ripe.net
- Login**: adminripe-ipnt@...
- Password**: ripeadmin

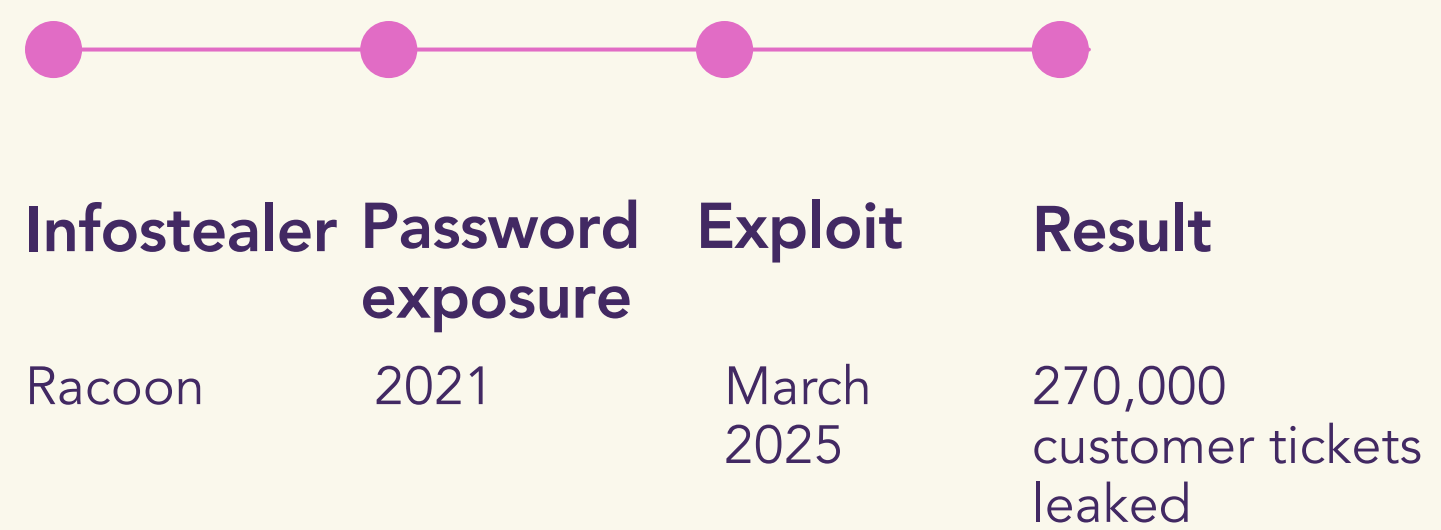
Source: <https://www.securityweek.com/ripe-account-hacking-leads-to-major-internet-outage-at-orange-spain/>

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1 1
0 1 1 0 1 1 0 1

```

Samsung Tickets Data Leak: Infostealers Strike Again in Massive Free Dump



Source: <https://www.infostealers.com/article/samsung-tickets-data-leak-infostealers-strike-again-in-massive-free-dump/>

VN 27. [REDACTED] 2021-07-13

VN 27. [REDACTED] 2021-07-1

Detected At: 2021-06-30 23:15

Compromised At: 2021-06-28 14:52

Corporate Credentials:

User Credentials:

PROFILE

IP Address: [REDACTED]

Computer Name: Not Found

Facebook: Not Found

Operating System: Not Found

Anti Virus: Not Found

EMPLOYEE DOMAINS:

- all-inkl.com
- alpto.com
- boltonhotel.co.nz
- boutiqueescorts.com.au
- climblife.us
- ivodigital.com

EMPLOYEE CREDENTIALS: EXPORT

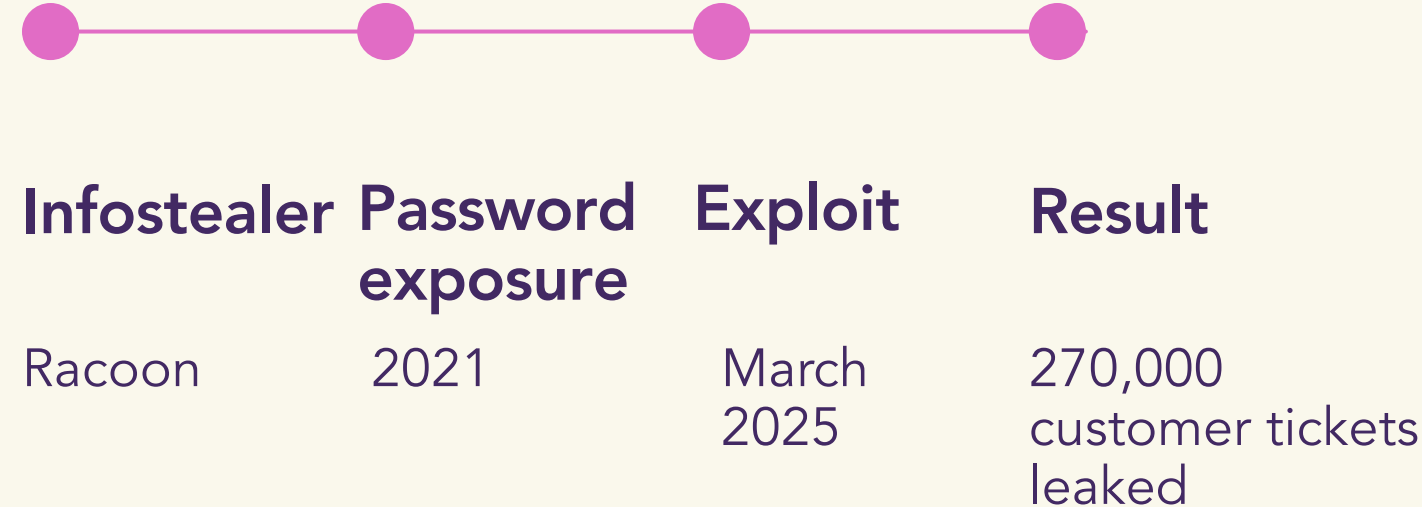
URL	LOGIN	PASSWORD
https://rpm-stage.spectos.com/en/users/login	deployer+samsung@spectos.com	[REDACTED] 7!
https://samsung-dev.spectos.com/FDB-10539/current/en/users/login	[REDACTED]+samsung@spectos.com	[REDACTED]
https://samsung-dev.spectos.com/dev/current/en/users/login	deployer+samsung@spectos.com	[REDACTED] 7!
https://rpm.spectos.com/en/users/login	deployer+samsung@spectos.com	[REDACTED] 7!

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
    1 1 0 1 1
        1 0
            0 0

```

Samsung Tickets Data Leak: Infostealers Strike Again in Massive Free Dump



Samsung Electronics (Germany) Customer Satisfaction Tickets - Leaked, Download!
 by GHNA - Sunday March 30, 2025 at 12:42 AM

10 hours ago (This post was last modified: 10 hours ago by GHNA.)

Hello BreachForums Community
 Today, I have uploaded Samsung Electronics (Germany)'s Customer Satisfaction Tickets for you to download, thanks for reading and enjoy!



Breached by @GHNA

In March 2025, Samsung Electronics (Germany), a global leader in technology and consumer electronics, suffered a data breach resulting in over 270,000 customer satisfaction tickets being leaked on a popular hacking forum. The tickets include PII of customers such as full names, addresses, e-mails, and more.

Sample (formatted):

```

{
  "_v": 6,
  "_id": "67d3e892f908e45d5e2af51e",
  "additionalDetails": [
  ],
  "additionalProperties": [
  ]
}

```

Profile: GHNA (Avatar: 地狱), GOD, Posts: 9, Threads: 8, Joined: Nov 2024, Reputation: 30

Source: <https://www.infostealers.com/article/samsung-tickets-data-leak-infostealers-strike-again-in-massive-free-dump/>

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0

```

CCOO



Web Inspection

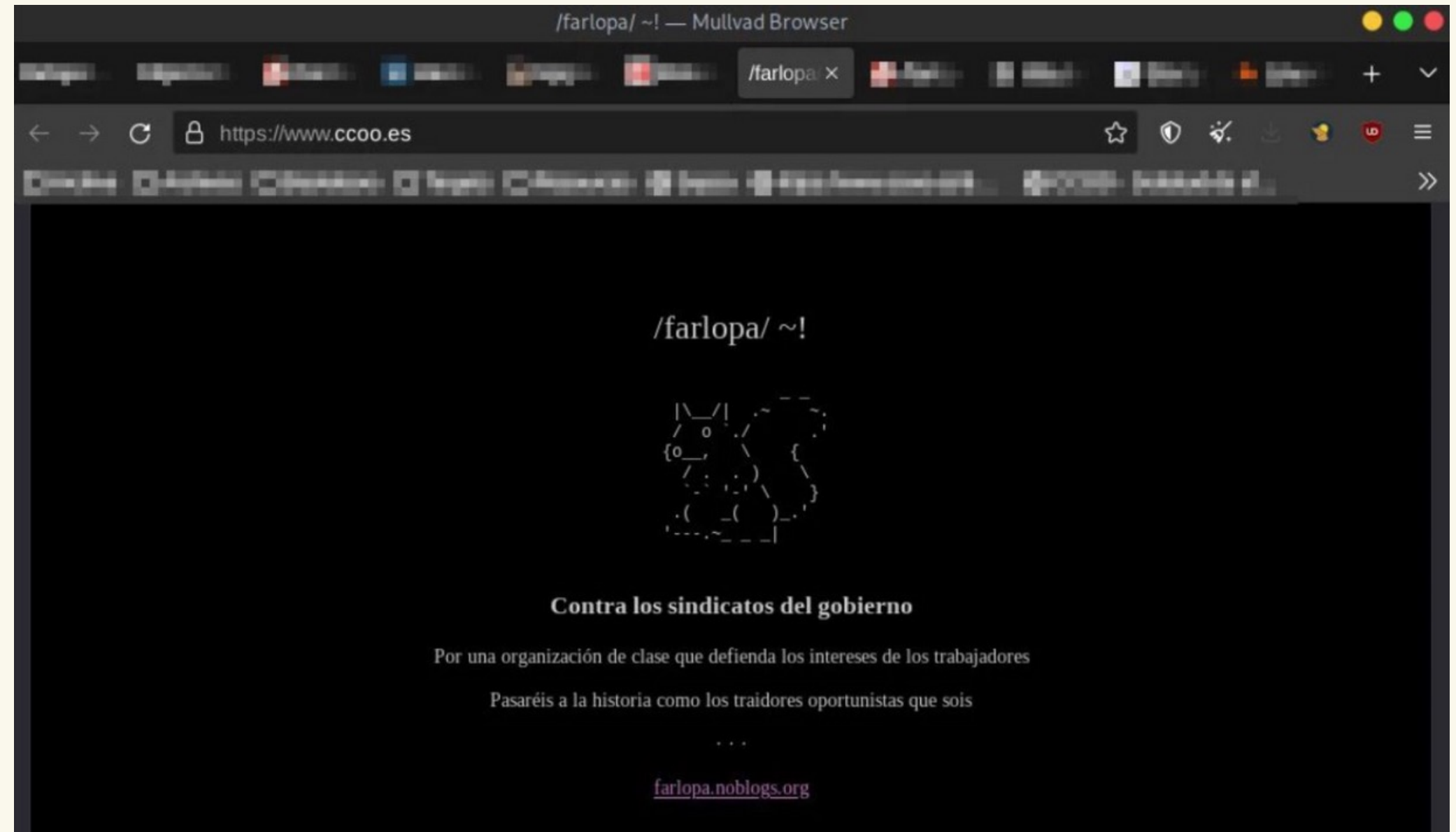
E: on url

Register form

File upload (.php)

File execution

Source: <https://blog.elhacker.net/2024/02/ccoo-comisiones-obreras-filtracion-22-mil-datos-personales.html>



0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 1 0 0 1 1 1 0 0 1
1 0 1 0 1 1 0 0 0 0 1
0 1 1 0 1 1 0 1 1 0 0

QUALITY AND CYBERSECURITY CULTURE

0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 1 1
1 0 1 0 1
0 1 1 0 1 1 0

CYBERSECURITY CULTURE

TOPICS

- Manual
- Functional
- Non Functional
- Mobile Testing
- Automation
- Management
- Accessibility
- AI
- Others...

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1 1
0 0 0 1 0 1 1
0 0 0 1 0 1 1 0 0
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 1 1

QUALITY CULTURE

RESOURCES

- Courses
- Certifications:
 - ISTQB, CSTE, CSQA, CAST...
- Conferences:
 - expoQA, EuroSTAR, AgileTestingDays...
- Communities

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 1 1 1

0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1 1
0 0 1 0 1 0

CYBERSECURITY CULTURE

TOPICS

- Network Security
- Cryptography
- **Application Security**
- Endpoint Security
- Identity and Access Management (IAM)
- Incident Response and Forensics
- Cloud Security
- Cyber Threat Intelligence (CTI)
- Governance, Risk, and Compliance (GRC)
- Penetration Testing and Ethical Hacking
- Others...

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

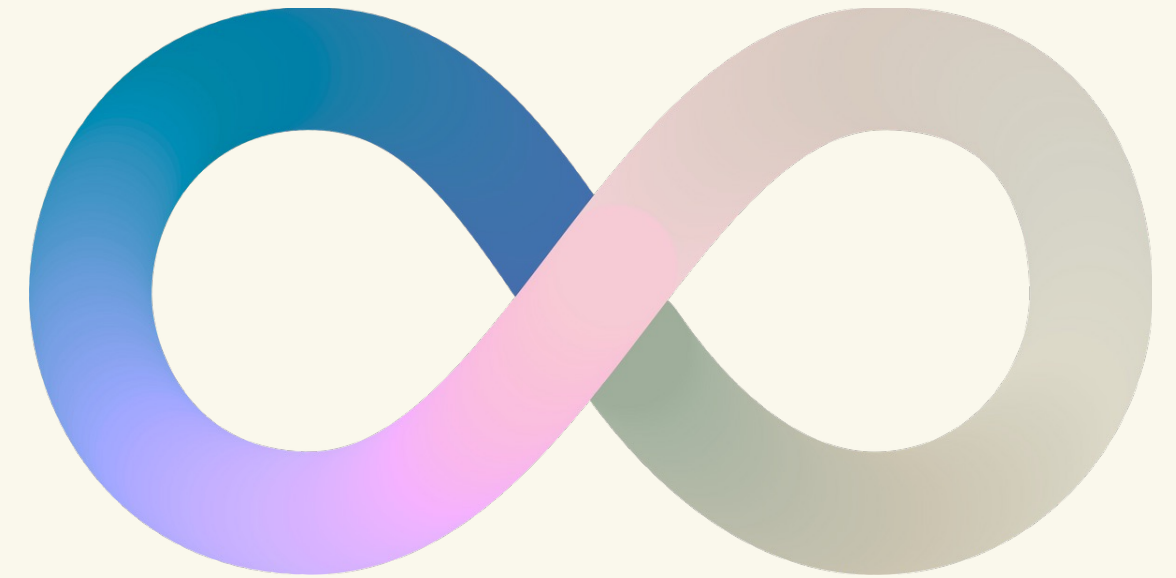
```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1 0 1
0 1 0 1 1 0

```

CYBERSECURITY CULTURE

RESOURCES



- Vocational training
- Career
- Masters Degree
- Certifications
 - CISSP, CEH, CISM, CISA, CCSP, CRISC, GSEC, CMA ...
- Training
- Conferences
- Workshops
- Bootcamps
- Open Resources and organisms:
 - OWASP
 - Portswigger
 - CISA

```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 1 1 1

```

"Secure" SOFTWARE DEVELOPER ENGINEER IN TEST

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

ISTQB (International Software Testing Qualifications Board)

- Security Tester (2016)

Self-taught

- Courses
- Conferences
- Communities

Cybersecurity

- SSDLC Rules
- XOPS
- AppSec
- Masters Degree
- OWASP, MITRE
- Portswigger, Hack The Box...



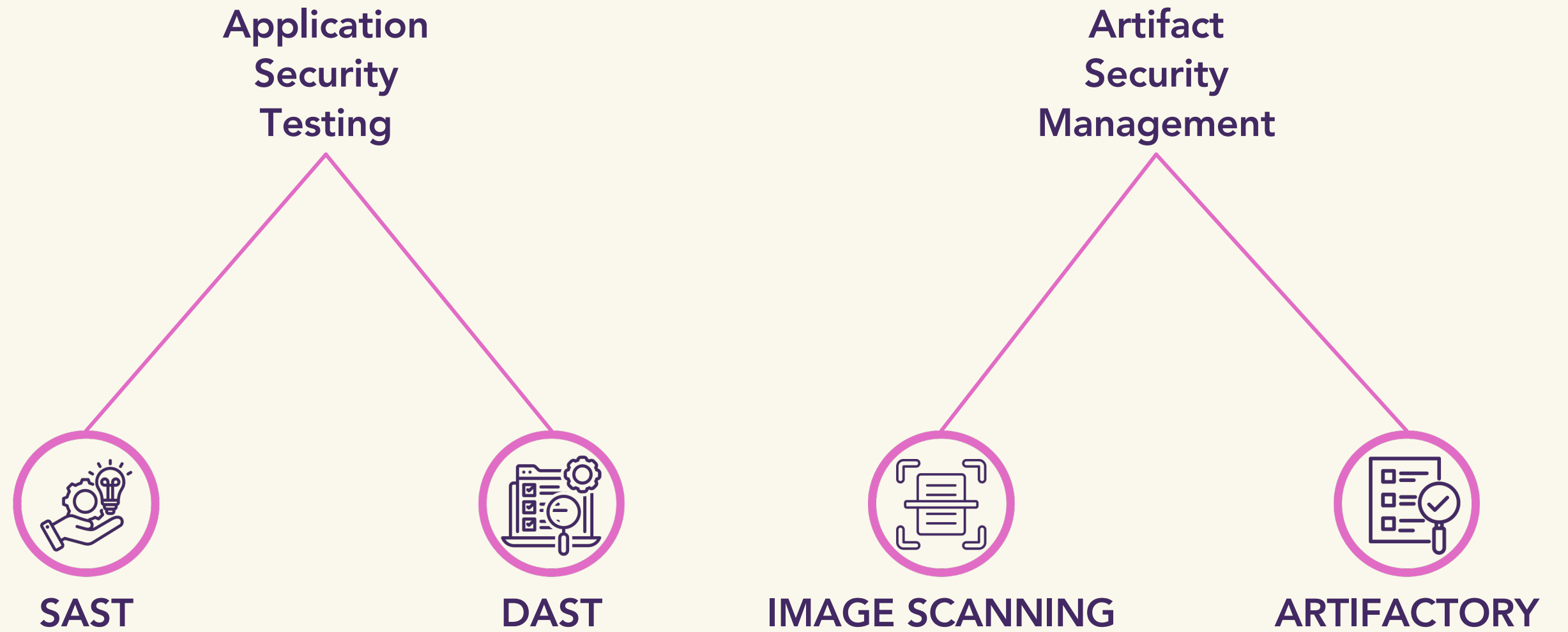
```
0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 0
1 1 0 1 1 1 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1
0 1 0 0 1 1 1 0 1 1 0 1
1 0 1 1 1 0 0 0 0 0 0 1
0 0 1 1 1 0 1 0 0 0 0 1
0 1 0 1 1 0 1 0 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 1
0 1 1 1 0 1 0 1 1 0 0 0
```

CODE ANALYSIS TOOLS

```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0 0
0 1 0 0 1 1 1 0 0 0 1
1 0 1 1 0 0 0 0 0 1
0 0 0 1 0 0 0 0 0 1
0 1 1 0 1 0 1 1 0 0

ANALYSIS TOOLS



```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
```

Application Security Testing

Static (SAST)

- Source code metrics
- **Aligned with security standards**
- Integration:
 - Code Editors
 - Version Control
- Tools:
 - https://owasp.org/www-community/Source_Code_Analysis_Tools #

Dynamic (DAST)

- Test Environment
- Isolated
- **Aligned with security standards**
- Tools:
 - <https://www.jit.io/blog/top-dast-tools-for-2024>

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0 0

```



Alerts

Name	Risk Level	Number of Instances
SQL Injection - Oracle - Time Based	High	1
SQL Injection - SQLite	High	2
Absence of Anti-CSRF Tokens	Medium	3
CSP: Wildcard Directive	Medium	14
CSP: script-src unsafe-inline	Medium	14
CSP: style-src unsafe-inline	Medium	14
Content Security Policy (CSP) Header Not Set	Medium	2
Cross-Domain Misconfiguration	Medium	10
Missing Anti-clickjacking Header	Medium	13
Cookie No HttpOnly Flag	Low	8
Cookie without SameSite Attribute	Low	13
Cross-Domain JavaScript Source File Inclusion	Low	33
Secure Pages Include Mixed Content	Low	12
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Low	18
Strict-Transport-Security Header Not Set	Low	111
Timestamp Disclosure - Unix	Low	12
X-Content-Type-Options Header Missing	Low	98
Authentication Request Identified	Informational	1
Cookie Poisoning	Informational	6
GET for POST	Informational	1
Information Disclosure - Suspicious Comments	Informational	50
Modern Web Application	Informational	15
Re-examine Cache-control Directives	Informational	10
Retrieved from Cache	Informational	4
Session Management Response Identified	Informational	8
User Agent Fuzzer	Informational	1335
User Controllable HTML Element Attribute (Potential XSS)	Informational	1

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1
```

Artifact Security Management

Artifact

- Access control
- Analysis of security and vulnerabilities
- **Aligned with security standards**
- Signature and validation
- Tools:
 - <https://bluelight.co/blog/how-to-choose-a-container-registry>

Image Scanner

- Artifact
- **Aligned with security standards**
- Third Party Libraries
- Signature and validation
- Tools:
 - <https://www.pingsafe.com/blog/container-scanning-tools/>

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

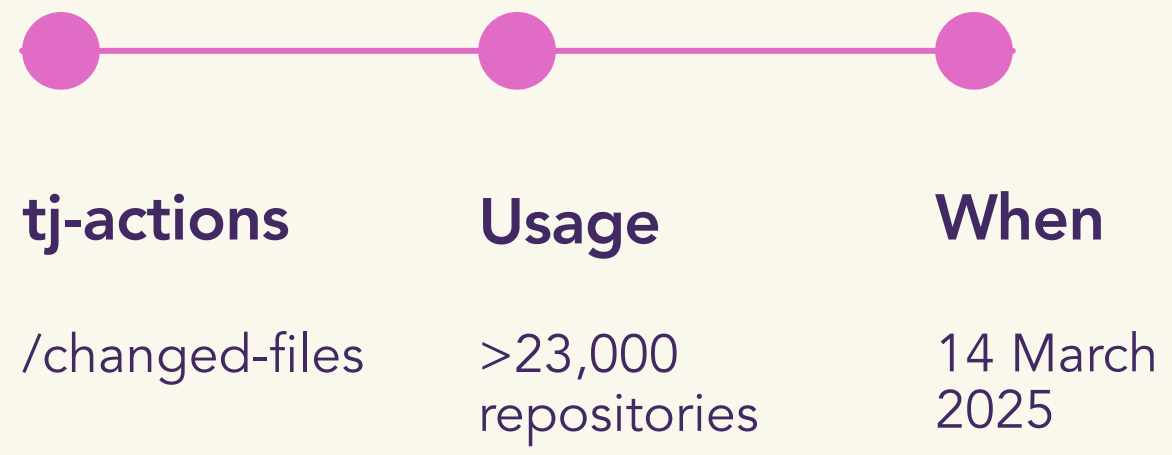
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1 1
0 0 1 0 0

Attack on DevOps: secrets leaked via malicious GitHub Action

How to respond to a compromised GitHub changed-files Action incident.

 Kaspersky Team

March 18, 2025



Source: <https://www.kaspersky.com/blog/malicious-github-action-changed-files/53179/>

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 0 1
0 0 1 1 0 1 1 0
1 1 1 0 1 0 0

HANDS ON! BREAK THE CODE

0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0

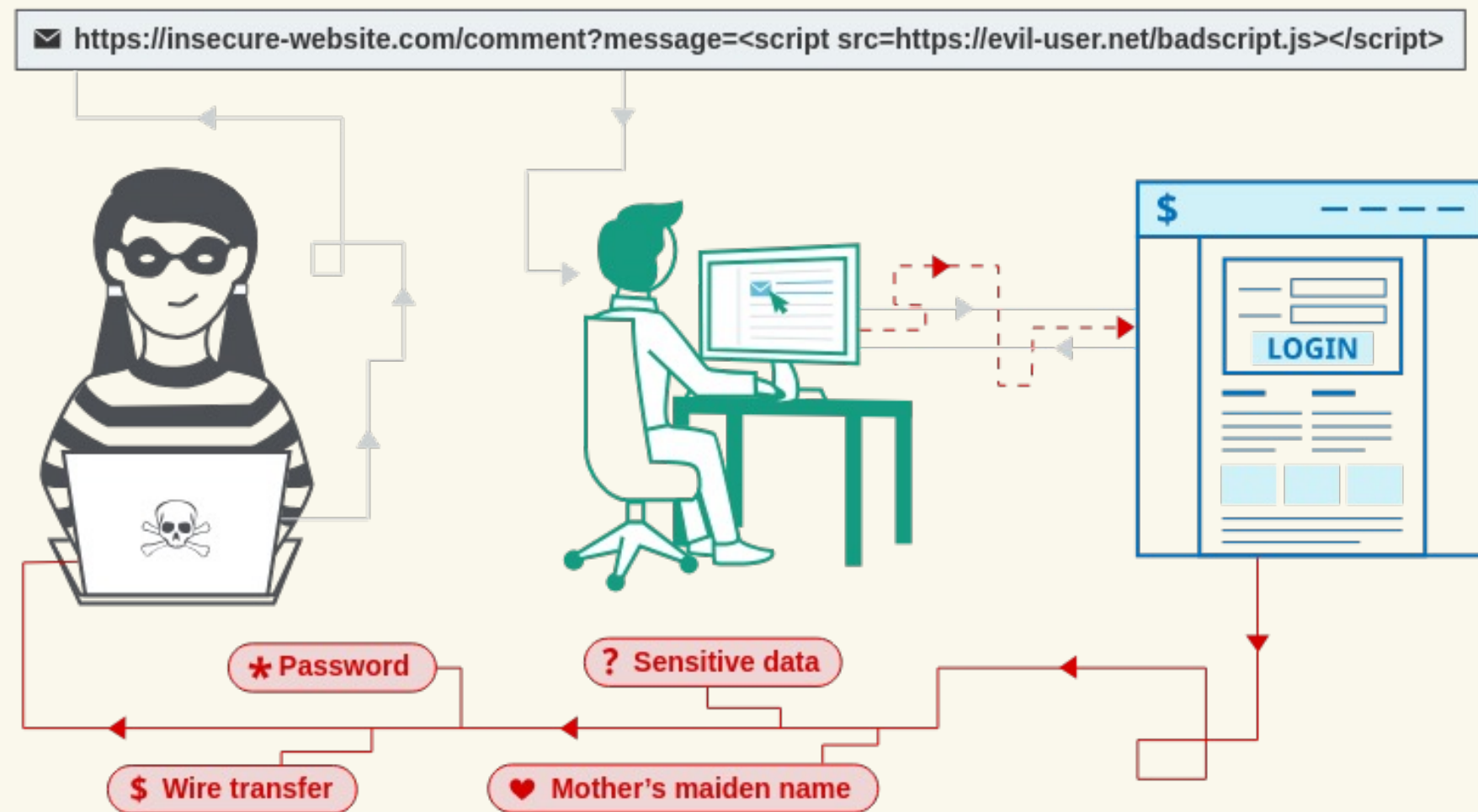
CVEs Most Exploited

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2015	1037	1104	221	776	152	6	249	50	8	46	379
2016	1180	1173	97	497	99	12	87	41	16	33	519
2017	2478	1542	505	1500	281	155	334	109	57	97	936
2018	2083	1731	503	2041	569	112	479	188	118	85	1248
2019	1205	2029	544	2387	488	126	560	137	103	121	907
2020	1217	1872	465	2201	436	110	415	119	131	100	812
2021	1663	2529	742	2724	548	91	520	126	192	133	677
2022	1863	3368	1788	3403	728	95	769	126	230	146	779
2023	1659	2236	2118	5129	764	112	1392	125	242	179	594
2024	1780	2522	2650	7456	944	257	1435	112	373	121	128
2025	369	534	644	2329	217	80	566	22	124	27	0
Total	16534	20640	10277	30443	5226	1156	6806	1155	1594	1088	6979

CVEs Most Exploited

Year	Overflow	Memory Corruption	Sql Injection	XSS	Directory Traversal	File Inclusion	CSRF	XXE	SSRF	Open Redirect	Input Validation
2015	1037	1104	221	776	152	6	249	50	8	46	379
2016	1180	1173	97	497	99	12	87	41	16	33	519
2017	2478	1542	505	1500	281	155	334	109	57	97	936
2018	2083	1731	503	2041	569	112	479	188	118	85	1248
2019	1205	2029	544	2387	488	126	560	137	103	121	907
2020	1217	1872	465	2201	436	110	415	119	131	100	812
2021	1663	2529	742	2724	548	91	520	126	192	133	677
2022	1863	3368	1788	3403	728	95	769	126	230	146	779
2023	1659	2236	2118	5129	764	112	1392	125	242	179	594
2024	1780	2522	2650	7456	944	257	1435	112	373	121	128
2025	369	534	644	2329	217	80	566	22	124	27	0
Total	16534	20640	10277	30443	5226	1156	6806	1155	1594	1088	6979

XSS - CROSS-SITE SCRIPTING



- **Reflected XSS:** malicious script comes from the current HTTP request.
- **Stored XSS:** malicious script comes from the website's database.
- **DOM-based XSS:** the vulnerability exists in client-side code rather than server-side code.

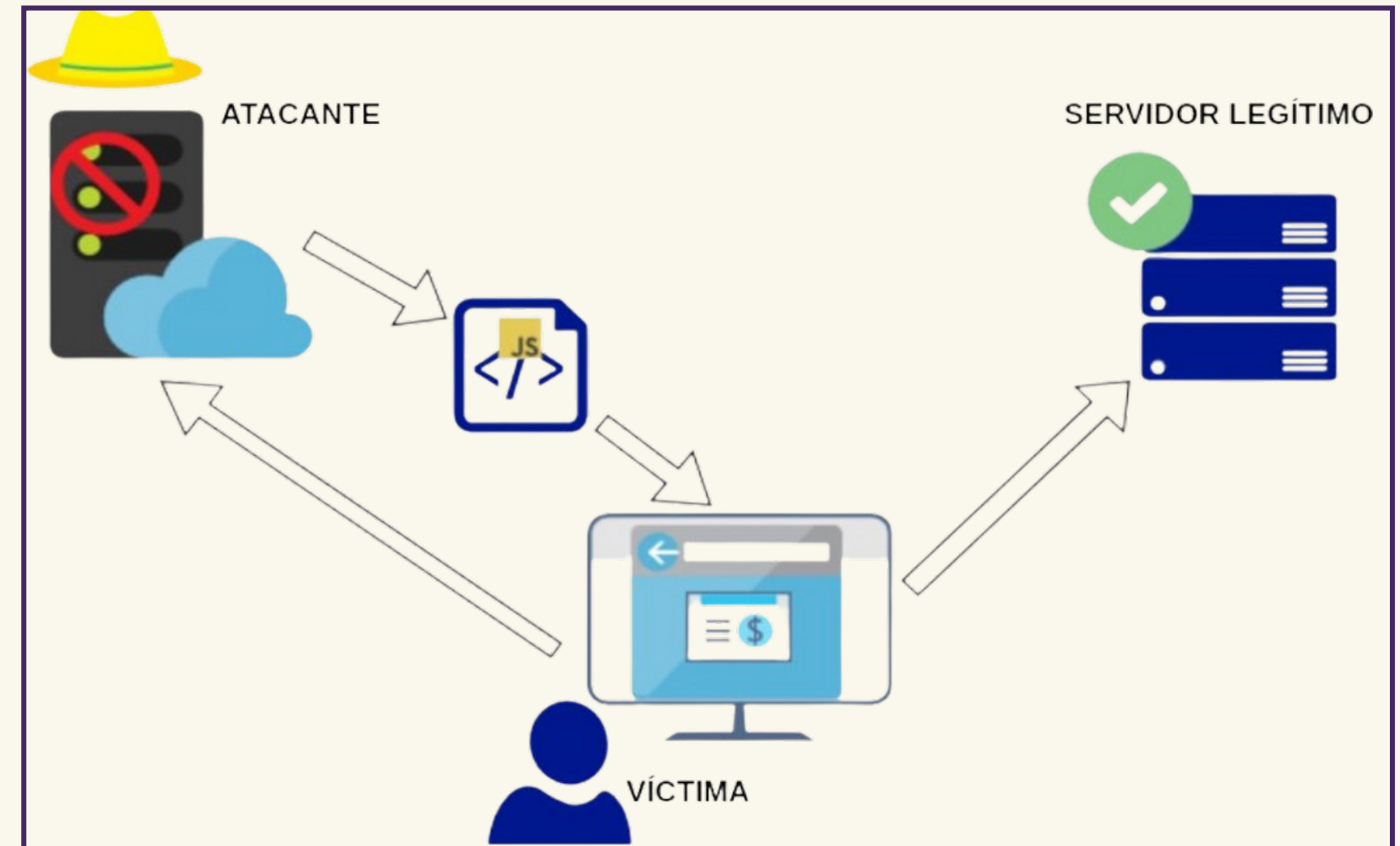
```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1 1
0 0 0
```

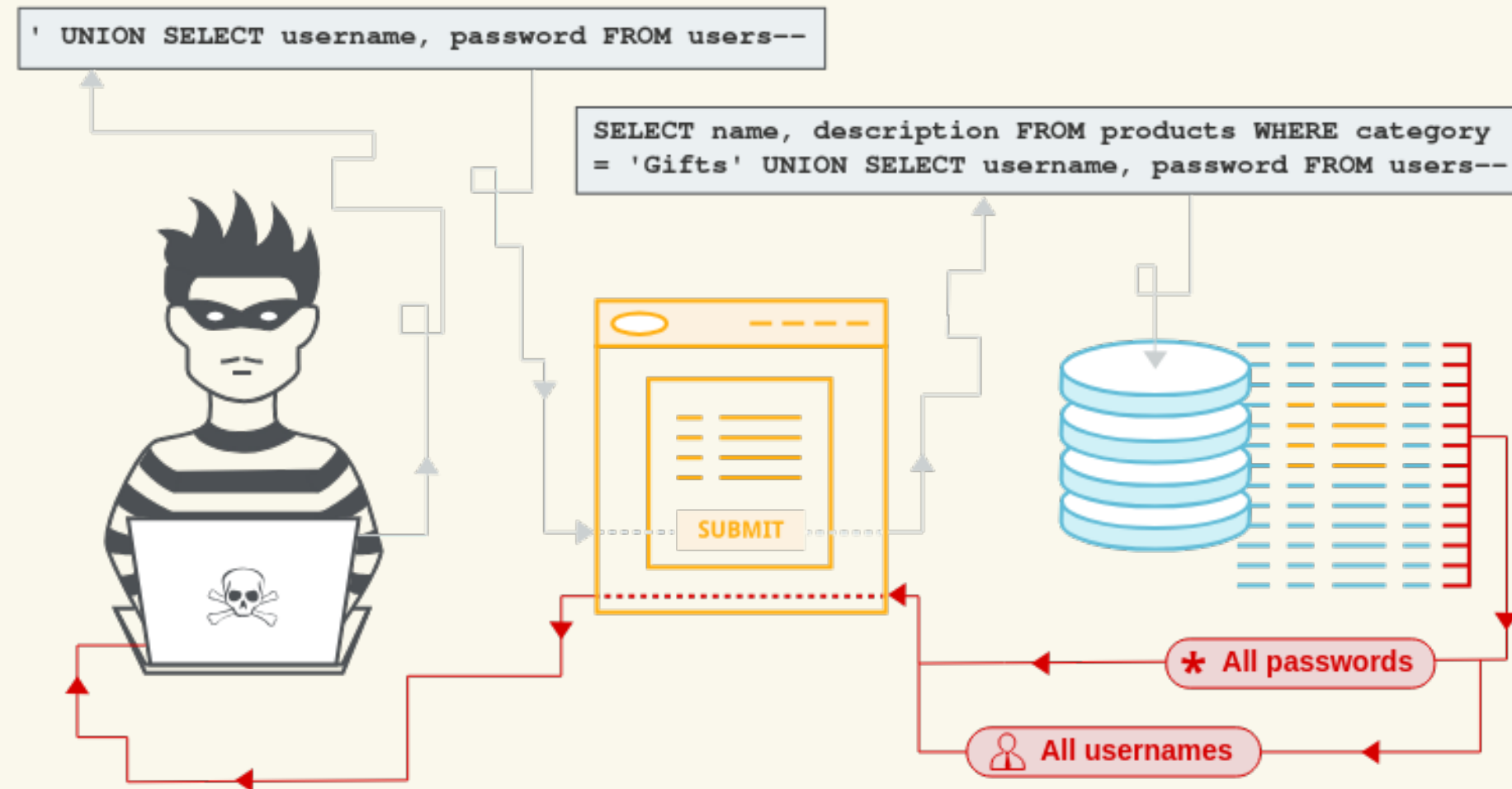
MAGECART - FORMJACKING

Irene Cotillas Torres
@IreCoT

Juan Carlos Fernández
@fernandez_jc



SQLi - SQL Injection



A successful SQL injection attack can result in unauthorized access to sensitive data, such as:

- Passwords.
- Credit card details.
- Personal user information.

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

Shopware Security Plugin Exposes Systems to SQL Injection Attacks

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 1 0 1 1
0 1 0 0
```

```
class PatchedAggregationParser extends AggregationParser
{
    public function buildAggregations(EntityDefinition
$definition, array $payload, Criteria $criteria,
SearchRequestException $searchRequestException): void
    {
        parent::buildAggregations($definition, $payload,
$criteria, $searchRequestException);
        foreach ($criteria->getAggregations() as $i =>
$aggregation) {
            if (str_contains($aggregation->getName(), '?') ||
str_contains($aggregation->getName(), ':')) {
                $searchRequestException->add(new
InvalidAggregationQueryException('Invalid character in
aggregation name'));
            }
        }
    }
}
```

Source: <https://cybersecuritynews.com/shopware-security-plugin-exposes-systems/>

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1 0 0
```

LET'S HACK!

LAB XSS:

- <https://portswigger.net/web-security/cross-site-scripting>

LAB SQLi:

- <https://portswigger.net/web-security/sql-injection>



```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

SECURITY TEST AUTOMATION

```
0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 0
1 1 0 1 1 1 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 1 0 0 0 1 0 1 0 1
0 1 0 0 1 1 1 0 0 1 0 1
1 0 1 1 1 0 0 0 0 0 0 1
0 0 1 1 1 0 1 0 0 0 0 1
0 1 0 1 1 0 1 0 1 1 0 0
```

```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1 1 0
0 0

- 01. Broken Access Control
- 02. Cryptographic Failures
- 03. Injection
- 04. Insecure Design
- 05. Security Misconfiguration
- 06. Vulnerable and Outdated Components
- 07. Identification and Authentication Failures
- 08. Software and Data Integrity Failures
- 09. Security Logging and Monitoring Failures
- 10. Server-Side Request Forgery (SSRF)



TOP 10:2024

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
```

SECURITY TEST SCENARIOS

Roles and permissions

- Privilege scalation

Files

- Allow only valid files

Login

- Strong Password
- 2FA
- Reestablishment procedures
- Cookies expiration

Forms


- Limit inputs (ie.XSS)

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1
0 0
```

EXAMPLE: Form with admin privileges

● ● ● <https://www.test.com/feature/edit>

 admin

EDITION FORM

param 1

param 2

param 3

param 4

param 5

param 6


SEND

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0 0 1
0 1 0 0 1 1 1 0 0 0 1
1 0 1 1 0 0 0 0 0 1
0 0 0 1 0 0
1 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
0 1 1 0 0 1 1 0 1 1 1
```

EXAMPLE: Form with restricted privileges

● ● ● <https://www.test.com/feature/edit>

 user_restricted

EDITION FORM

param 1

param 2

param 4

SEND

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0
```

WHY SO FUNCTIONAL?

```
@component
Feature: Edition form

@dev @security
Scenario: Check edition params for restricted user
    Given I login with user "user_with_restricted_permission"
    And I navigate to the "https://www.test.com/feature/" URL
    When I go to edition form
    Then the following params are visible and writeable
        | param 1 |
        | param 2 |
        | param 4 |
    And the following params are not visible in form
        | param 3 |
        | param 5 |
        | param 6 |
```

```
0 1 0 1 0 1 0 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1 1
0 0
```

WHY SO FUNCTIONAL?

```
@component
Feature: Feature data visibility

@dev @security
Scenario: Check feature params for restricted user - API
  Given I login to host "https://www.test.com/"
    | authentication |
    | user_with_restricted_permission |
  When I send a "GET" request to the "feature/" api endpoint
  Then the response status code is "200"
  And the response body contains the following params
    | param 1 |
    | param 2 |
    | param 4 |
  And the response body does not contain the following params
    | param 3 |
    | param 5 |
    | param 6 |
```

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 0 1 1
0 0

```

WHY SO FUNCTIONAL?

@dev @security

Scenario Outline: Check writable feature params for restricted user - API

Given I login to host "https://www.test.com/"

authentication	
user_with_restricted_permission	

And the body defined by file "restricted_edit_feature.json" with values

parameter	value
<parameter>	<value>

When I send a "PUT" request to the "feature/edit" api endpoint

Then the response status code is "200"

Examples:

parameter	value
param 1	val1
param 2	val2
param 4	val3

```

0 1 0 1 0 1 0 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0

```

WHY SO FUNCTIONAL?

```

@dev @security
Scenario Outline: Check non writable feature params for restricted user - API
  Given I login to host "https://www.test.com/"
    | authentication |
    | user_with_restricted_permission |
  And the body defined by file "restricted_edit_feature.json" adding values
    | parameter | value |
    | <parameter> | <value> |
  When I send a "PUT" request to the "feature/edit" api endpoint
  Then the response status code is "400"
  Examples:
    | parameter | value |
    | param 3 | val3 |
    | param 5 | val5 |
    | param 6 | val6 |

```

```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1 0 0

HANDS ON!

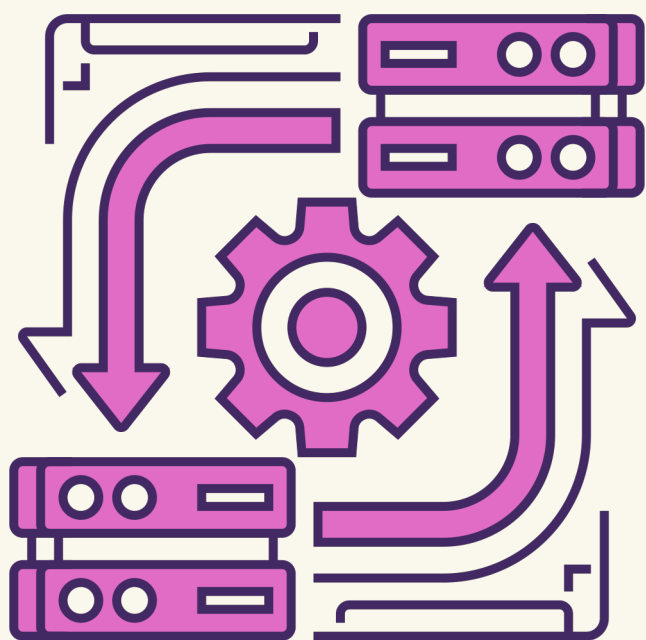
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0

```

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0

```

TEST AUTOMATION



Toolium

- Python
- Selenium
- API
- Behave

Code Security

- git-crypt / secrets
- Dependabot
- SAST
- PRs Management

 https://github.com/testingsoul/cyber_automate

```

Feature: XSS through inputs

Background:
  Given the user logs into PortSwigger

@security
Scenario Outline: Force XSS through input
  Given I open XSS laboratory
  When I search the value "<value>"
  Then there are no alerts in browser

Examples:
  | value |
  | <script>alert('Crash WEB')</script> |

```

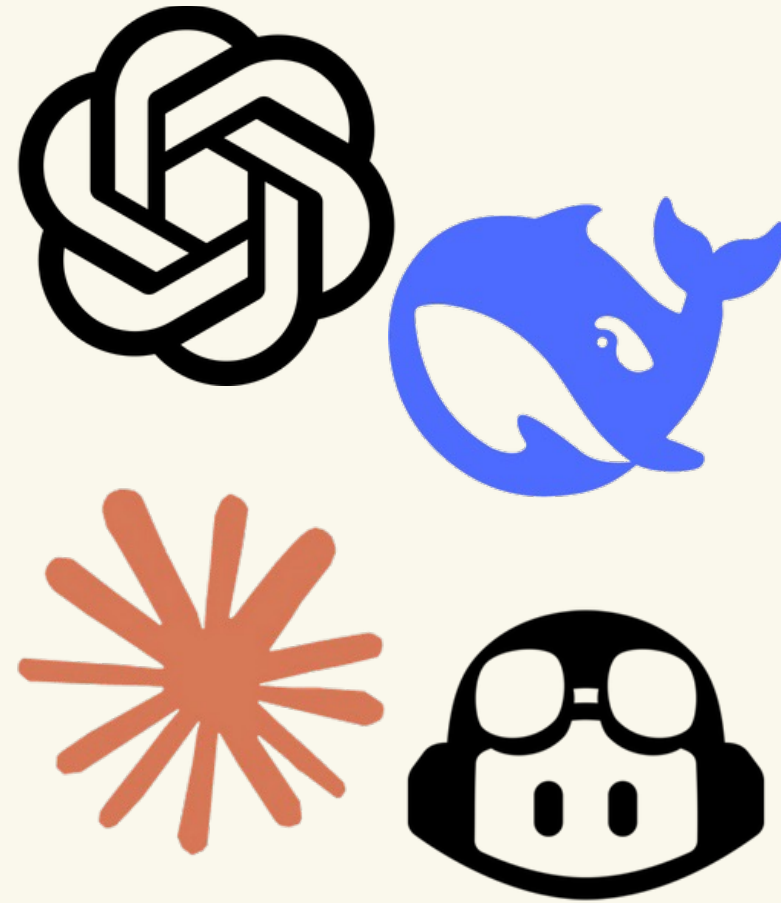
```

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

```

```
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 0 1 0 1 0 0
0 1 0 0 1 1 1 0 0 1 0 1
1 0 1 1 0 0 0 0 0 1
0 0 0 1 0 1 0 0
1 1 0 1 0 1 1 0
0 1 1 0 1 1 0
```

USE AI TO DEFINE SECURITY TEST SCENARIOS



Roles and permissions

- Privilege scalation

Files

- Alow only valid files

Login

- Strong Password
- 2FA
- Reestablishment procedures
- Cookies expiration

Forms

- Limit inputs (ie.XSS)

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

```
0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 0
1 1 0 1 1 1 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 0
0 1 0 0 1 1 1 0 0 1 0 1
1 0 1 1 0 0 0 0 0 0 1
0 0 1 1 0 0 1 0 0 0 1
0 1 0 0 1 0 0 0 0 0 1
0 1 1 1 0 1 0 1 1 0 0
```

CI: GITHUB ACTIONS

```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
```


GITHUB ACTIONS STRUCTURE

```
0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1
0 0 0 1 0 1 0 1 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1
```

```
name: Test Execution

> on: ...

jobs:
  build-and-test:
    name: Build and Test
    runs-on: ubuntu-latest
    > strategy: ...

    steps:
    - uses: actions/checkout@v2
    > - name: Set up Python...
    > - name: Unlock secrets...
    > - name: Install toolium...
    > - name: Start selenoid...
    > - name: Run acceptance tests...
    > - name: Upload Acceptance Test Results...
    > - name: Run security tests...
    > - name: Upload Security Test Results...

    > publish-acceptance-test-results: ...

    > publish-security-test-results: ...
```

GITHUB ACTIONS AS SIMPLE AS...

```
< - name: Run acceptance tests
<   run: |
<     behave test --junit --junit-directory test/output -t "~@security"
> - name: Upload Acceptance Test Results ...
< - name: Run security tests
<   run: |
<     behave test --junit --junit-directory test/output -t "@security"
> - name: Upload Security Test Results ...
```

Build and Test
Process completed with exit code 1.

Force XSS through input -- @1.1 (features.inputs.XSS through inputs) failed: features.inputs.XSS through inputs#L0
security-tests-results/TESTS-features.inputs.xml [took 9s]

Force SQLi through url (features.url.SQli through url) failed: features.url.SQli through url#L0
security-tests-results/TESTS-features.url.xml [took 8s]

```
0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1
```

Publish Acceptance Tests Results summary

Acceptance Tests Results

3 tests ±0 1 ✓ ±0 10s ⌚ -15s
2 suites ±0 2 🔄 ±0
2 files ±0 0 ✗ ±0

Results for commit 62647455. ± Comparison against earlier commit 56d2e37a.

[Job summary generated at run-time](#)

Publish Security Tests Results summary

Security Tests Results

3 tests ±0 0 ✓ -1 17s ⌚ -8s
2 suites ±0 1 🔄 -1
2 files ±0 2 ✗ +2

For more details on these failures, see [this check](#).

Results for commit 62647455. ± Comparison against earlier commit 56d2e37a.

[Job summary generated at run-time](#)

0 1 0 1 0 1 0 0 0 0 0 0 0 1
0 0 1 0 1 0 1 1 0 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1 1 0
1 0 0 0 0 0 1 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1 1 1
0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1 0 0

CONCLUSIONS

0 1 0 1 0 1 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1
1 1 0 1 1 1 0 0 0 0 0
1 0 0 0 0 0 1 0 1 0 1
0 0 0 1 0 1 0 1 0 0 0
0 1 1 0 1 0 1 0 1 1 1 0
1 0 1 1 0 1 0 1 0 0 0

0 1 1 0 1 0 1 0 1 1 1 0 0 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0 0 1
0 1 1 0 0 0 1 0 1 0 1 1 1 1 0
0 0 1 1 1 1 0 0 1 1 0 0 0 0 1
0 1 0 0 0 0 0 1 1 0 0 0 1 1 0
1 0 0 0 1 1 1 0 0 1 0 0 0 0 0
0 1 1 0 0 1 0 1 0 0 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 1 0 0 0
0 1 0 1 1 0 0 0 1 1 0 1 1 0 1
0 0 1 0 0 1 1 0 0 1 0 0 0 0 1
1 1 1 1 0 0 0 0 1 0 1 0 1 0
0 1 0 0 1 1 1 0 0 1
1 0 1 1 0 0 0 0 1
0 0 0 1 0 0
1 1 1 0 1 1
0 0 0 1 1 1 0 0

CONCLUSIONS

01. Security is a Continuous Process

02. Shift Left for Stronger Security

03. Raising of vulnerabilities and risks

04. Awareness and Culture Matter

05. Security testing is also functional

0 1 0 1 0 1 0 0 0 0 0 0 0
0 0 1 0 1 0 1 1 0 1 0 1 0
1 1 0 1 1 1 0 0 0 0 0 0 1
1 0 0 0 0 0 1 0 1 0 1 1 1
0 0 0 1 0 1 0 1 0 0 0 0 1
0 1 1 0 1 0 1 0 1 1 1 0 0
1 0 1 1 0 1 0 1 0 0 0 0 0
0 1 1 0 0 0 1 0 1 0 1 1 1

THANKS!

ANY QUESTION?

@swtestingsoul



saramartinezgine



r

sara@testingsoul.com



www.testingsoul.com



expo IQA 25

MADRID
May 20th,
21st & 22nd
2025

Thank you for attending

expoqqa.eu