All you ever wanted to know about Frontend Security!

catawiki

2024

Agenda

1. About Us
2. About Catawiki
3. Introduction to evolution of frontend security
4. Problems we're trying to solve
5. Selecting our security tooling
6. Pragmatic Implementation
7. Fostering adoption
8. Live Demo
9. TL;DR

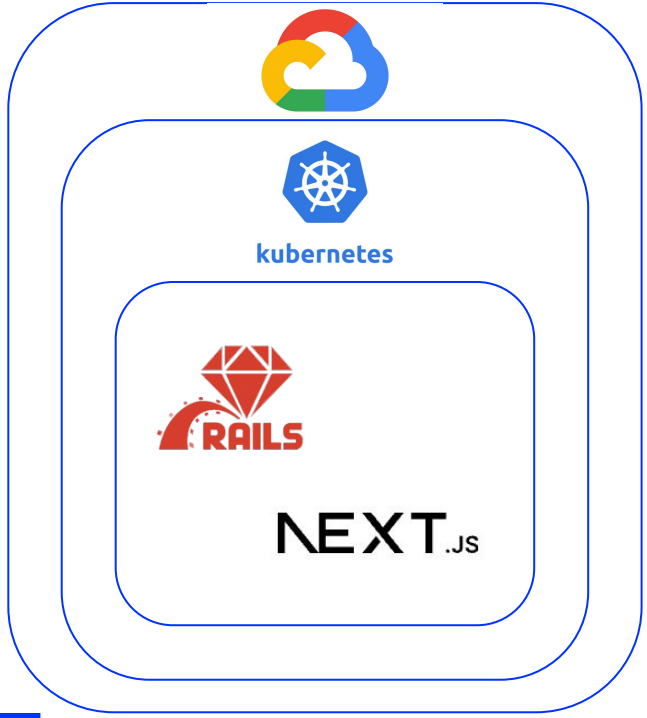
About Speakers

- 🔐 Lead Product Security @Catawiki
- 💡 Fmly Cloud & Cybersecurity consulting @KPMG & Devoteam
- 🤓 SME on AWS Cloud and DevSecOps
- ⛳ Casual Golfer, 🖼️ Art Collector, 🍱 Korean Cuisine

About Speakers

- 💻 Principal Software Engineer @Catawiki
- 💡 Frontend Web focused, ~15 years of experience
- 🧗 Bouldering, 🏍️Motorcycles, 🥗Greek Cuisine

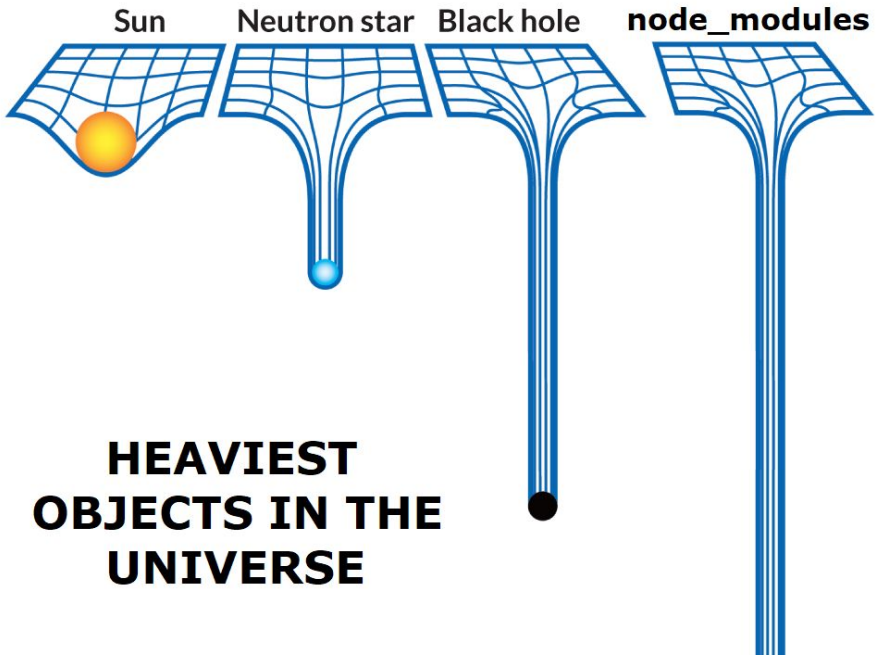
About Catawiki




```
if (!empty($_POST['email']) && !empty($_POST['password']))
{
    $query    = 'SELECT * FROM `user` WHERE `email` = \''.$_POST['email'].'\' AND `password` = \''.md5($
```

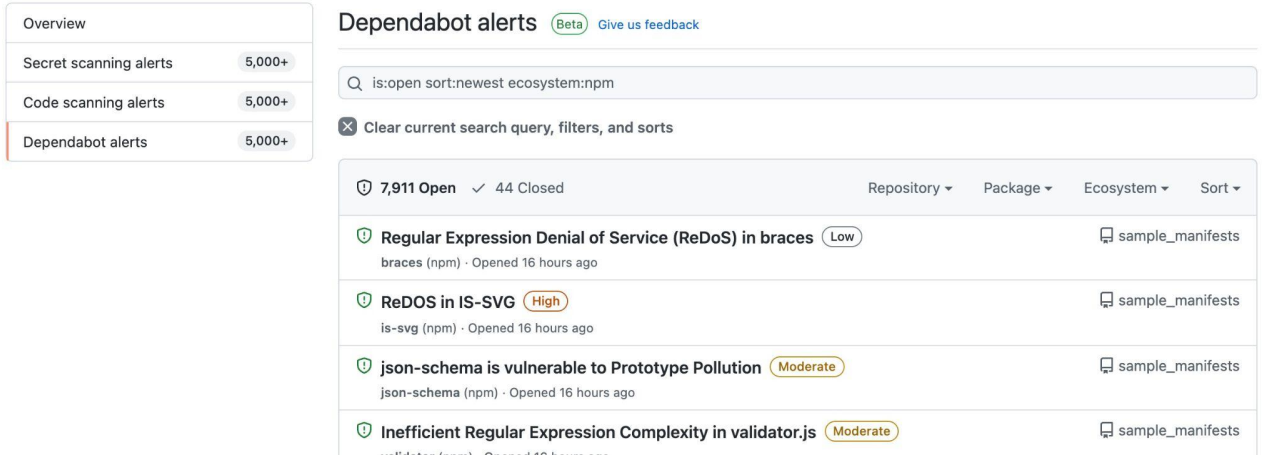
NEXT.JS

What are the most common Frontend security pitfalls?


Sun
Neutron star
Black hole
node_modules
HEAVIEST OBJECTS IN THE UNIVERSE

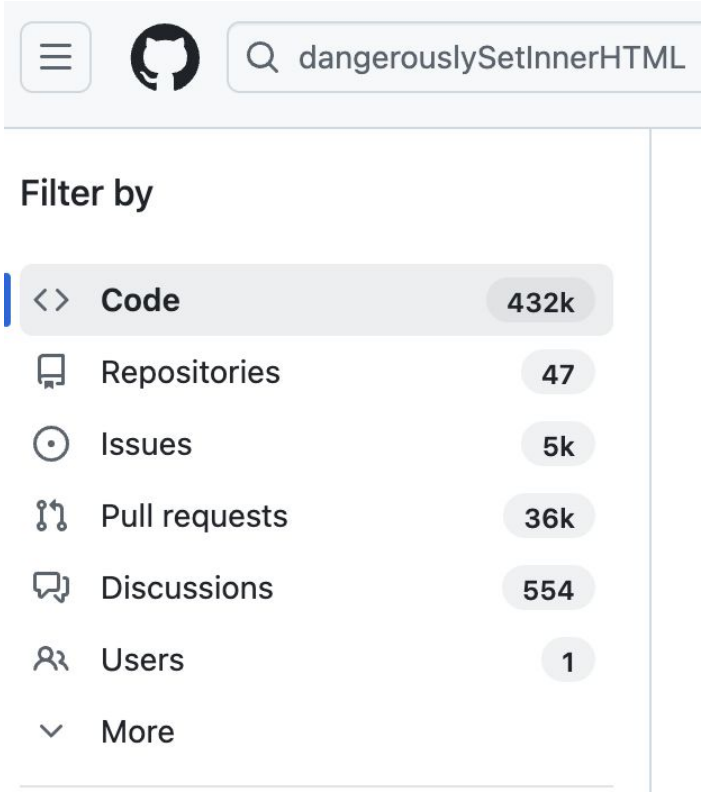
Tooling takes care for most of the dependency vulnerabilities if utilised

```
up to date, audited 1138 packages in 6s

59 packages are looking for funding
  run `npm fund` for details

44 vulnerabilities (43 moderate, 1 high)

To address issues that do not require attention, run:
  npm audit fix

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.
```



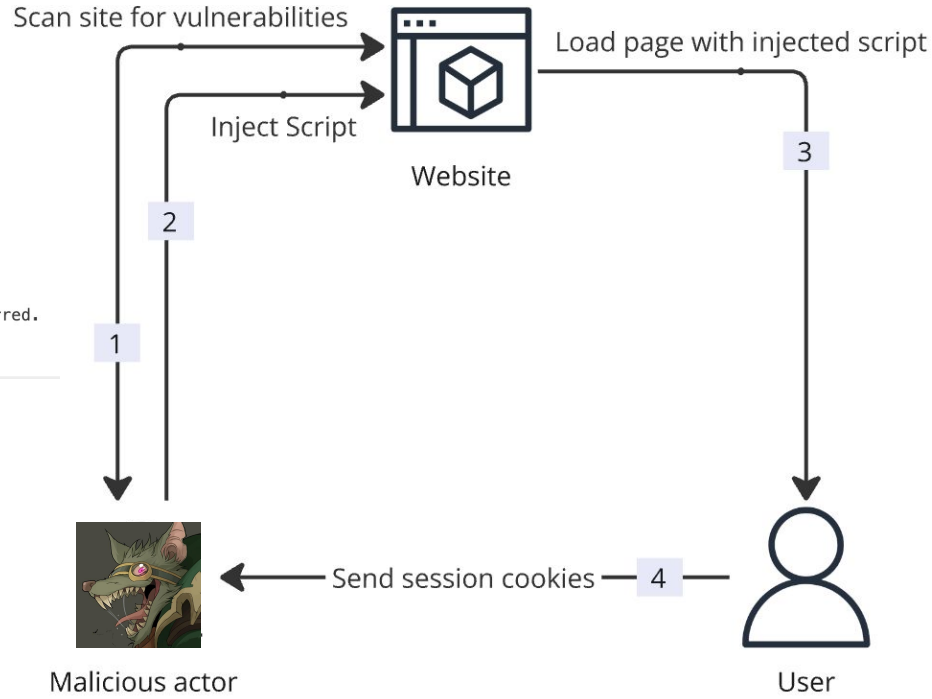
```
port React from 'react';
nst HTMLComponent = () => {
  const htmlSnippet : string = '<p>Hello, <strong>world</strong>!</p>';

  return (
      <div dangerouslySetInnerHTML={{ __html: htmlSnippet }} />
```

A lot of adrenaline enthusiasts...

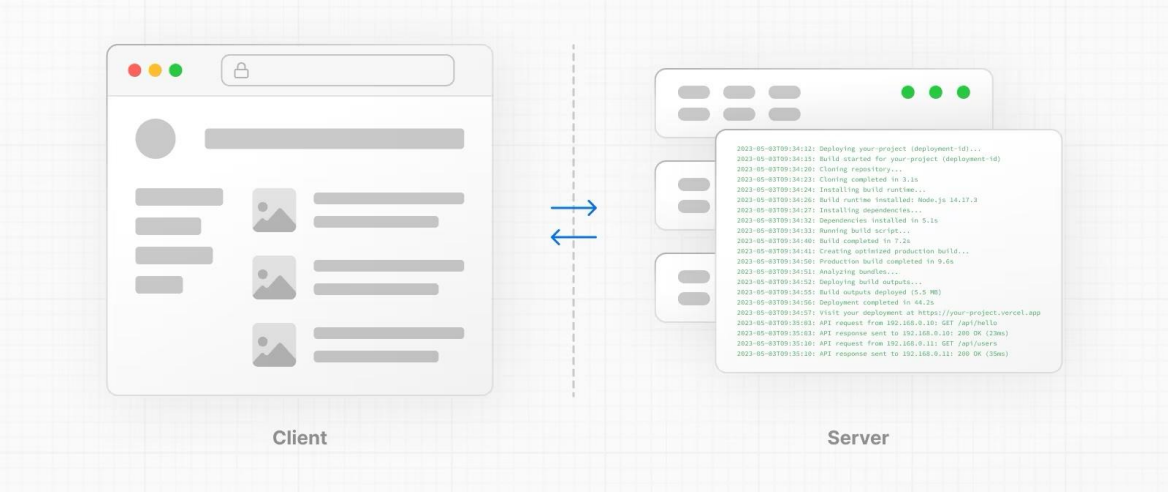


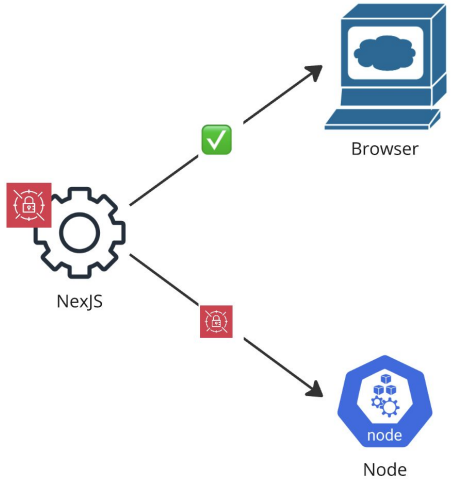
Security Vulnerabilities from Developer Code

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cache State Watcher</title>
</head>
<body>
  <div class="cache-state fresh">
    Error message: Error<script src="http://remote.com/malicious.js"></script> occurred.
  </div>
</body>
```



Isomorphic apps are still not well understood





A

Problems we're trying to solve

Discovery

1. Are we running obsolete systems in production?
2. Are we running obsolete dependencies in production?
3. Is our infrastructure configuration secure?
4. Does our code itself contains vulnerabilities?

Prioritisation

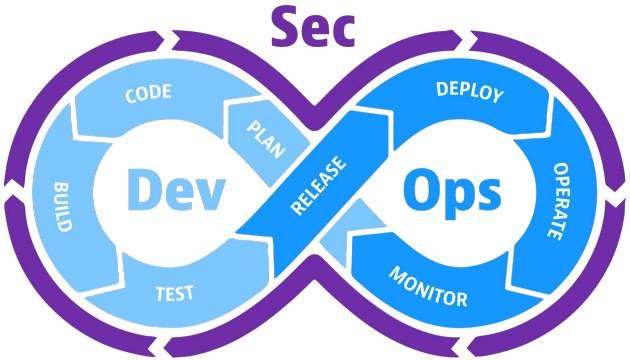
1. What are our crown jewels? 👑
2. What sensitive data are we storing or processing?
3. Are vulnerabilities actually exploitable?

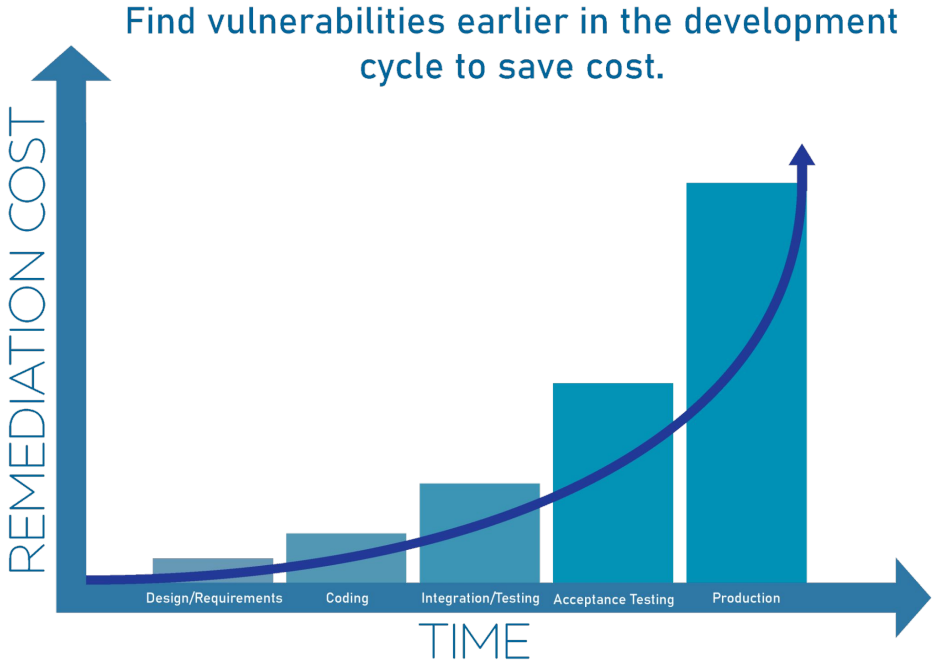
Remediation

1. Who should address vulnerabilities?
2. How promptly should vulnerabilities be addressed?
3. How to ensure they get addressed in time?

Enhancing Security during Development

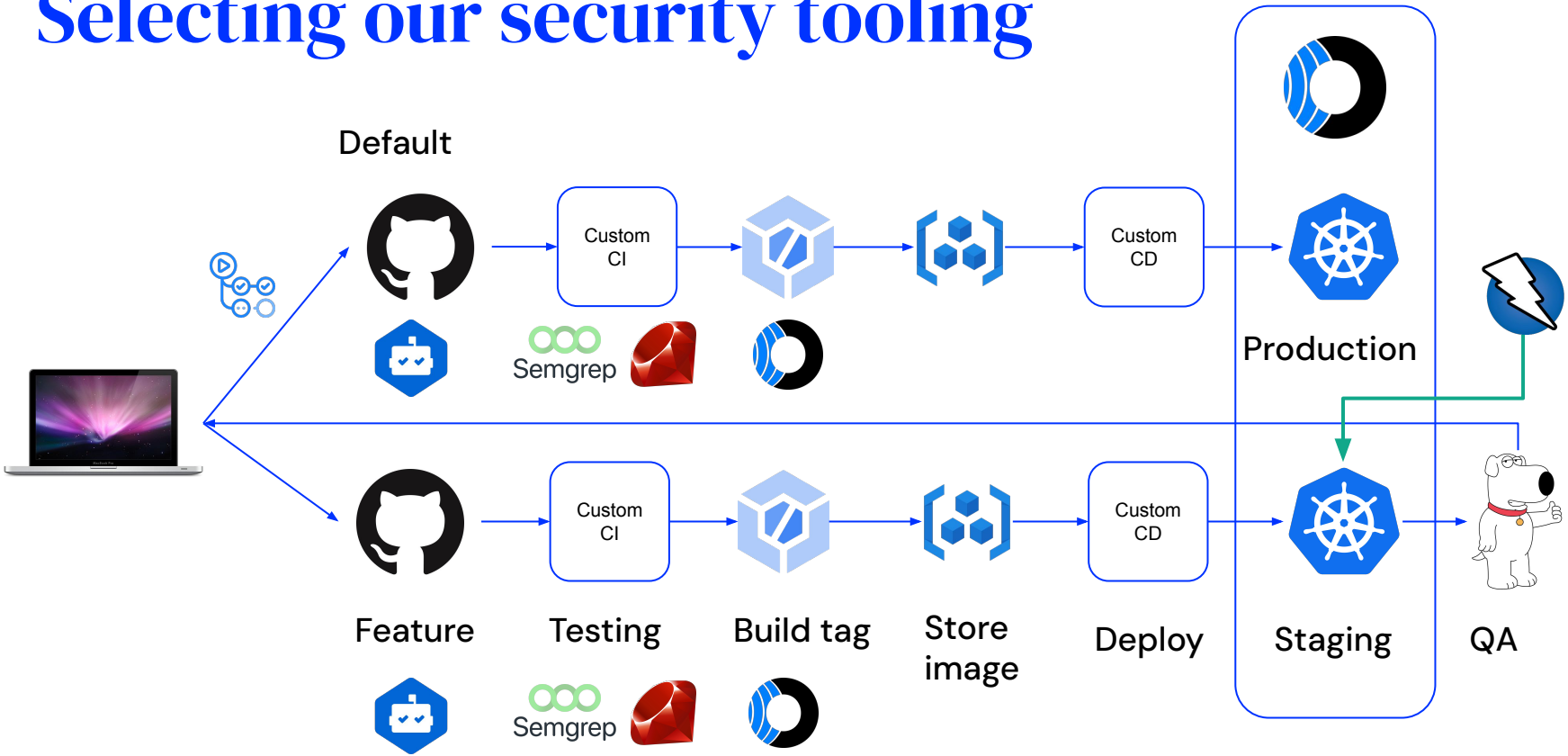
Integrating security early in development reduces the impact 💰💰 of vulnerabilities



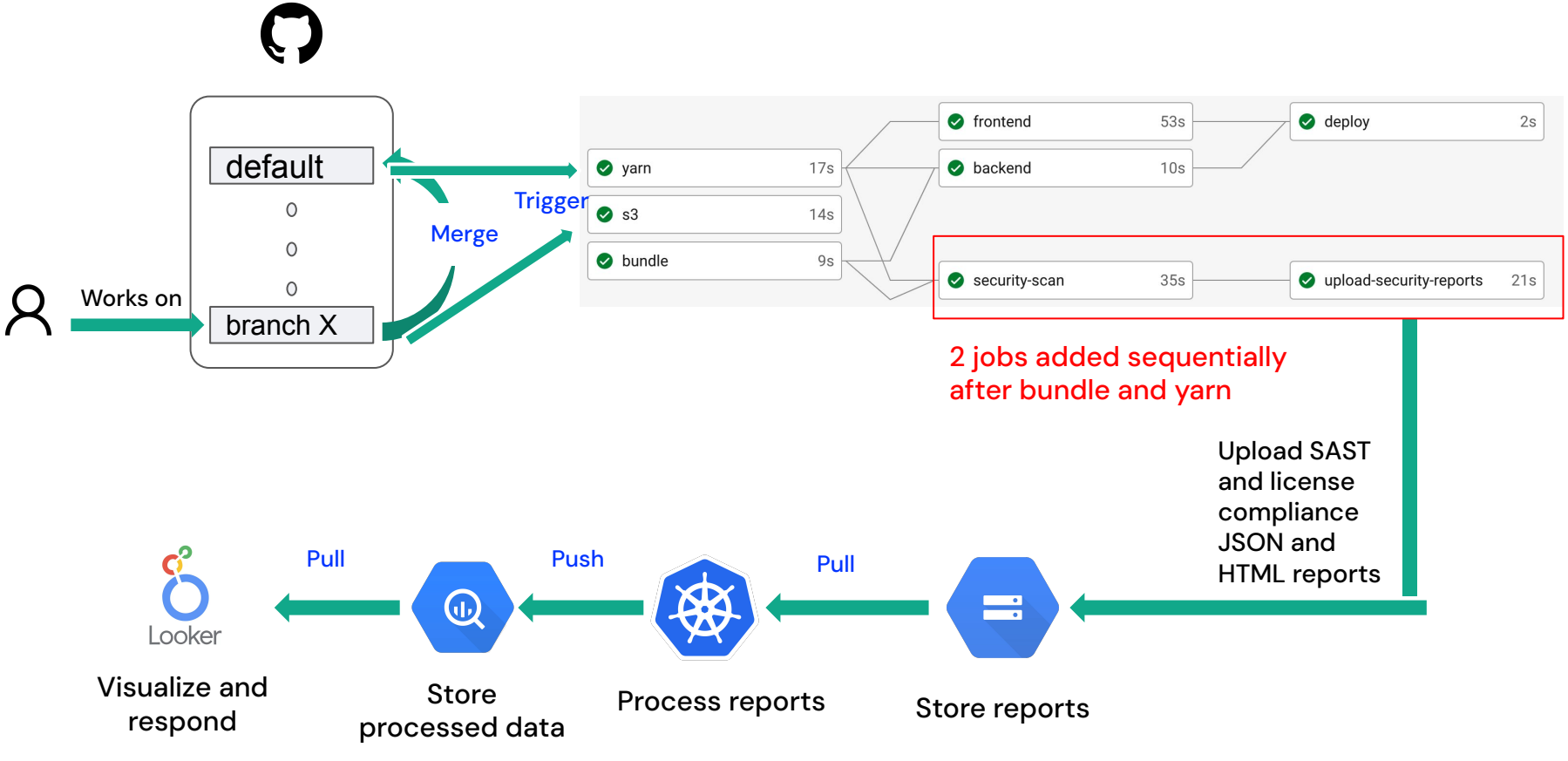


Source: https://www.linkedin.com/pulse/understanding-threat-modeling-trolleyesecurity/

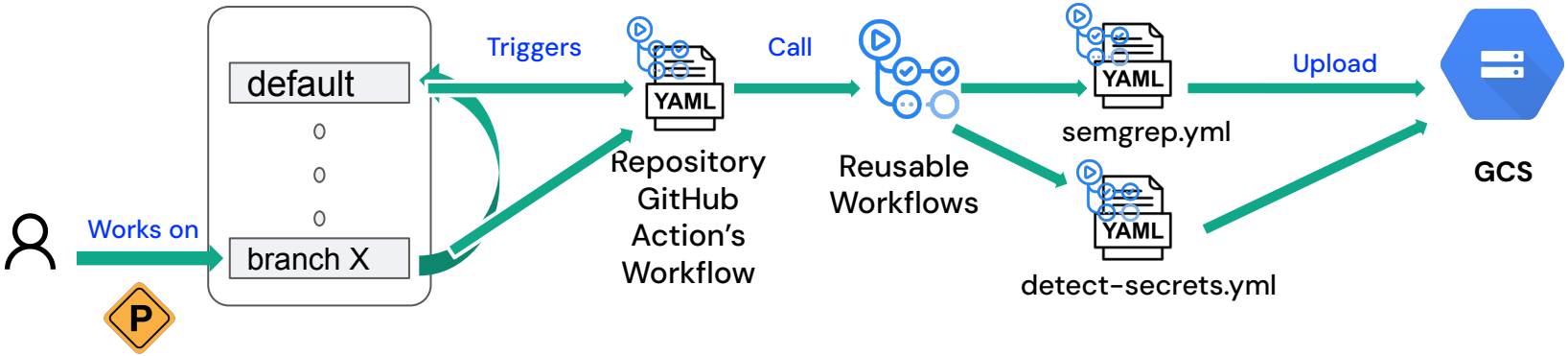
Selecting our security tooling



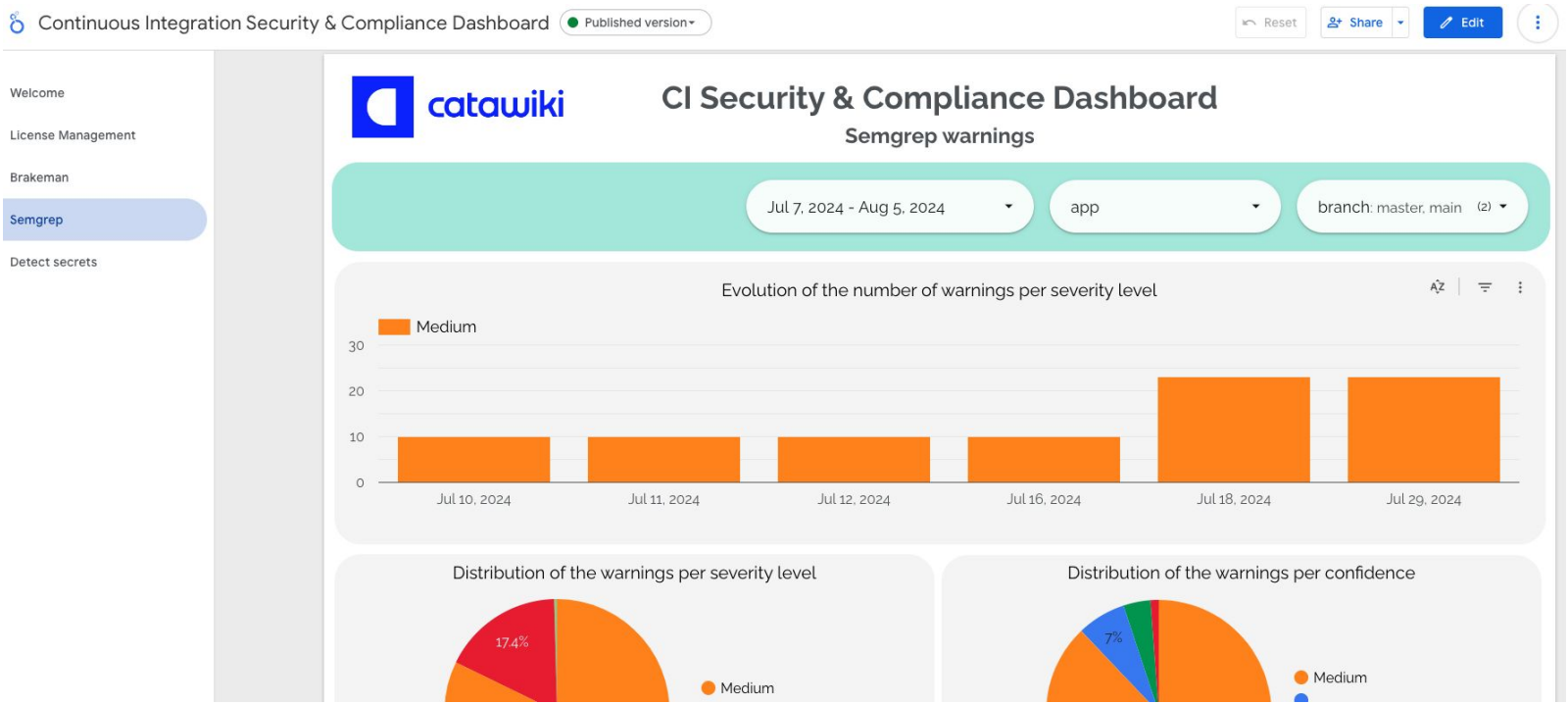
Pragmatic Implementation: Iteration I



Pragmatic Implementation: Iteration II



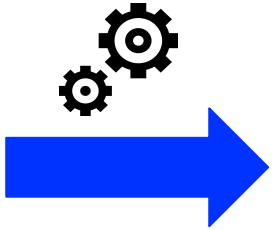
Pragmatic Implementation: Iteration II



Fostering Adoption: Streamlining

Jira Software

Semi-automated



Automated daily exports to Looker Studio

Looker

Twilio

Fostering Adoption: Gamifying

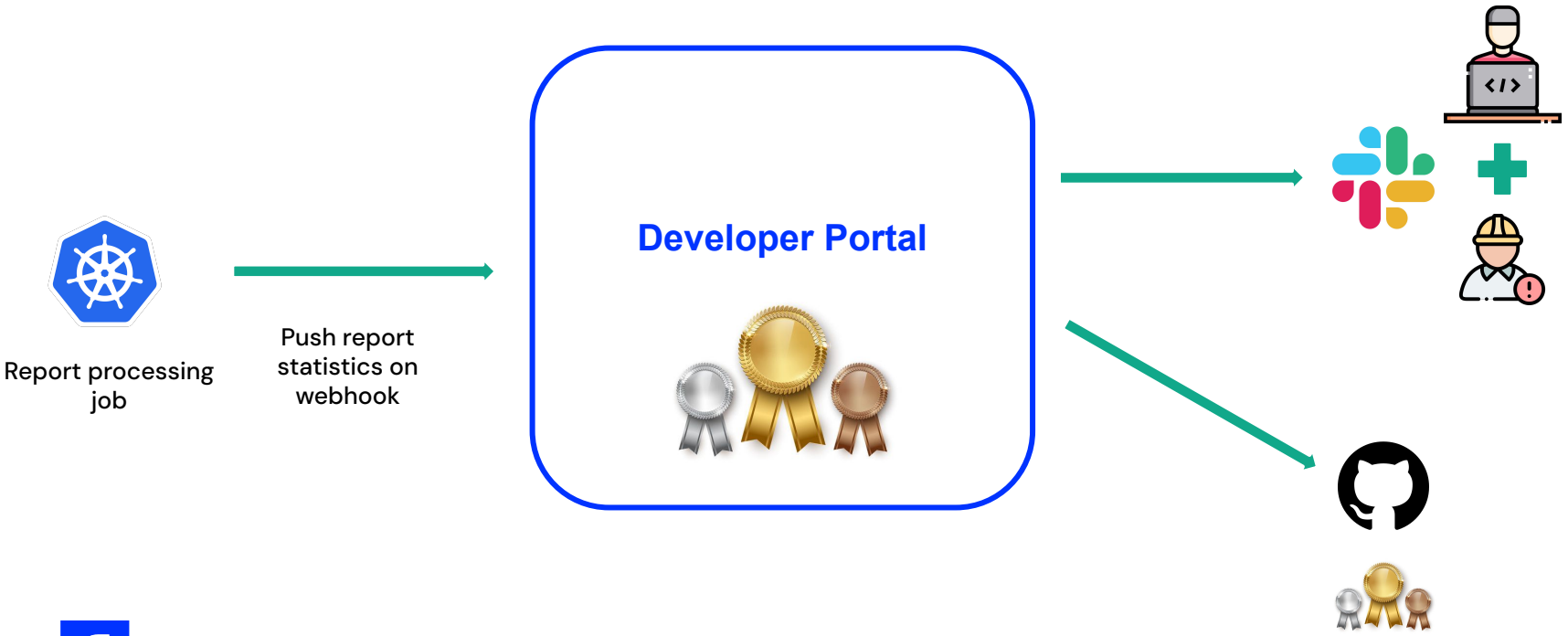
Security and Vulnerability Newsletters

Team and service security leadership board and perks

Developer L&D trainings

Advanced workshops for security champions

Fostering Adoption: Service badges



Live demo

TL;DR

1. Diligence and discipline
2. Repetition breeds retention
3. Testing redundancy is your safety net
4. Gamify learning to make it fun and effective
5. Don’t let hackers cash in!

Thank You!

Aristide Bouix
Product Security Engineer

aristide.bouix@gmail.com
linkedin.com/in/aristide-bouix/
@ArisvdZ
@ArisBee
@arisbcollection

Srgjan Dimitrijevikj
Principal Frontend Engineer

srgjan3@gmail.com
linkedin.com/in/srgjan-dimitrijevikj-2a196130
https://www.srgjan-d.com/