

# {DevSecOps}

Application Security and DevSecOps  
- End to end



# Losses caused by cyber attacks reported to IC3



[https://www.ic3.gov/Media/PDF/AnnualReport/2023\\_IC3Report.pdf](https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf)

## Top 5 crime types:

- › Phishing
- › Non-Payment / Delivery
- › Data Breach
- › Identity Theft
- › Extortion

## Trends

- › Confidence fraud / Romance scams
- › Cryptocurrency
- › Ransomware
- › Tech support fraud

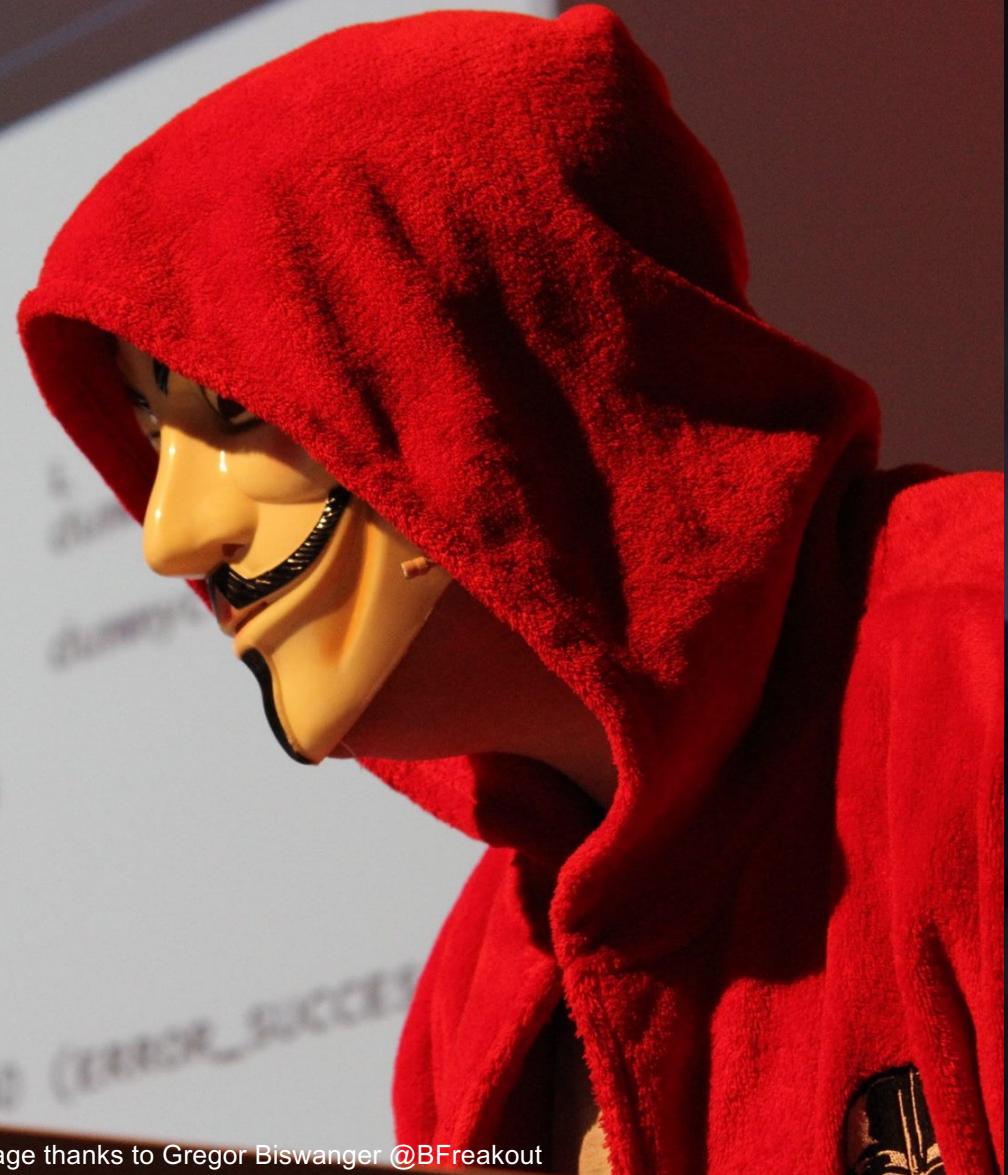
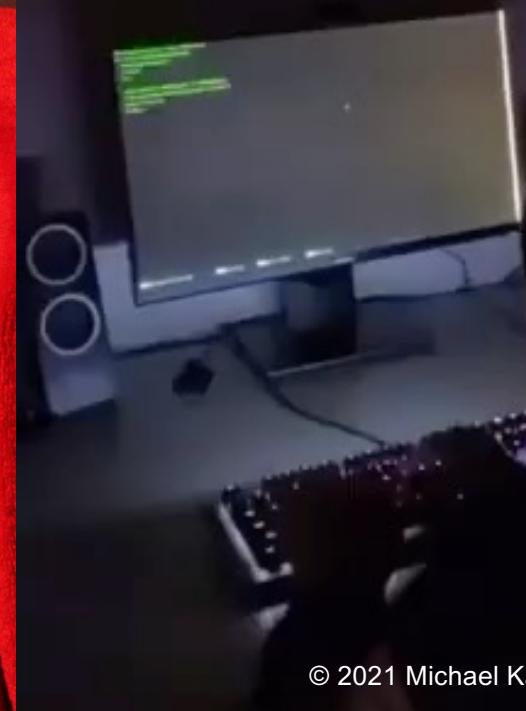


Image thanks to Gregor Biswanger @BFreakout

## Hackers in movies vs



© 2021 Michael Kaufmann @mike\_kaufmann



## How people think they get hacked



## How they really get hacked

<p>If you had to marry your spouse where you met them, where would your wedding have been?</p>	<p>Your porn name , is your middle name , and the first car you had.</p>	<p>How far away do you live from the place you were born?</p>
<p><b>The car you passed your drivers test in was a _____</b></p>	<p>A 'porn name' that exposes your middle name AND a street you grew up on are TWO pieces of info that people need.</p>	<p><b>NAME A SONG</b></p>
<p>STOP. THINK. DO NOT SHARE INFO.</p>	<p>In what city or town was your first job?</p>	<p><b>THAT TAKES YOU BACK TO HIGH SCHOOL</b></p>

# Michael Kaufmann

Founder & Managing Director, Xebia Germany



@mike\_kaufmann



@wulfland



<https://writeabout.net>

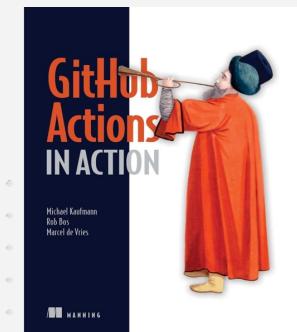
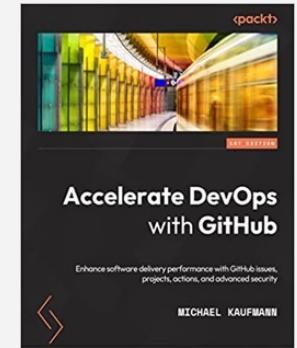


➤ 25 years software developer

➤ 15 years ALM & DevOps

➤ 10 years Git and GitHub

➤ Microsoft RD & MVP



# The event-stream incident



## Social engineering attack



Supply chain attack:  
event-stream@3.3.6 -> flatmap-stream@0.1.1



Code execution in build process  
targeting **copay**



Harvest the user's bitcoin and  
private keys

**Malicious Package in flatmap-stream**  
Critical severity | GitHub Reviewed | Published on 1 Sep 2020 · Updated on 1 Oct 2021

Follow

right9ctrl · 東京都 · Committed to this repository

GHSA ID: GHSA-9x64-5r7x-2q53

CWEs: CWE-506

CVSS Score: 9.8 Critical CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

This advisory has been edited. See History.

See something to contribute? Suggest improvements for this vulnerability

**Description**

Version 0.1.1 of `flatmap-stream` is considered malicious.

This module runs an encrypted payload targeting a very specific application, `copay`, and because they shared the same description it would have likely worked for `copay-dash`.

The injected code:

- Read in AES encrypted data from a file disguised as a test fixture
- Grabbed the npm package description of the module that imported it, using an automatically set environment variable
- Used the package description as a key to decrypt a chunk of data pulled in from the disguised file

The decrypted data was part of a module, which was then compiled in memory and executed.

This module performed the following actions:

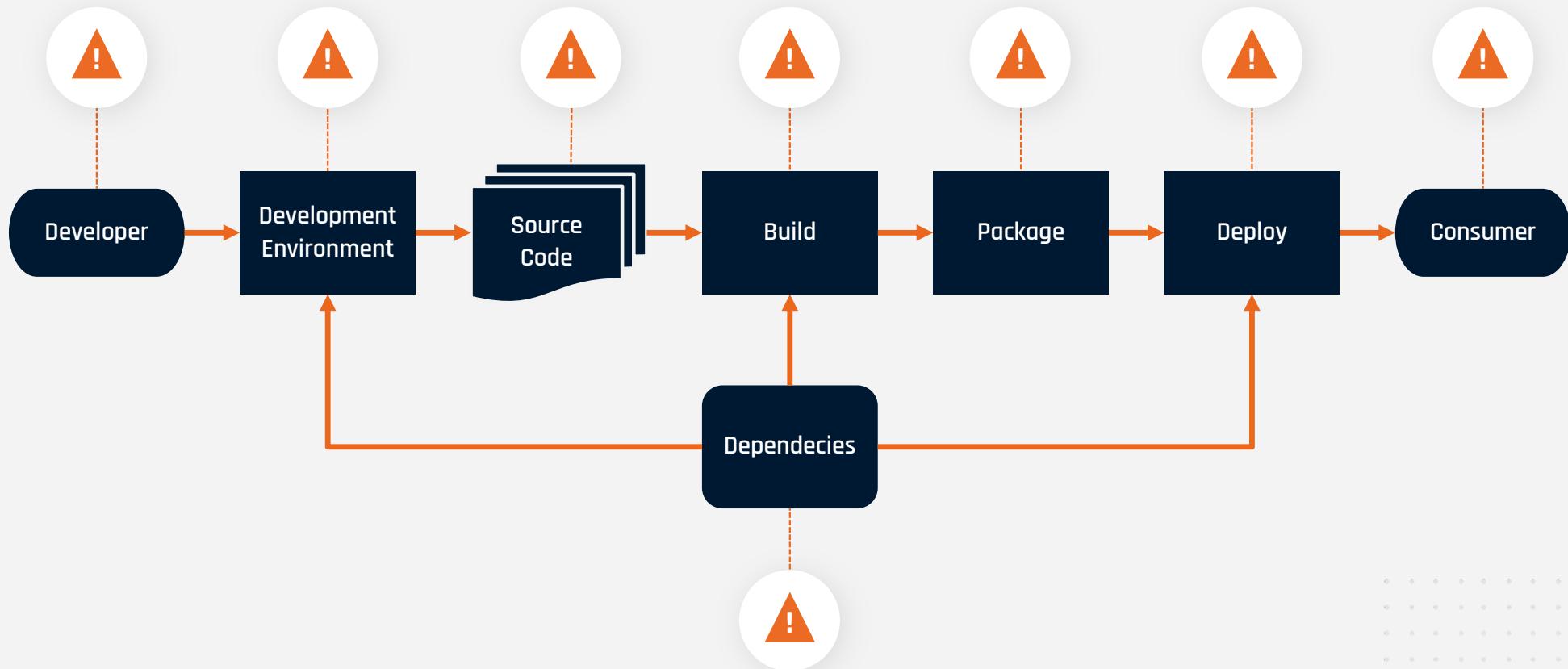
- Decrypted another chunk of data from the disguised file
- Concatenated a small, commented prefix from the first decrypted chunk to the end of the second decrypted chunk
- Performed minor decoding tasks to transform the concatenated block of code from invalid JS to valid JS (we believe this was done to evade detection by dynamic analysis tools)
- Wrote this processed block of JS out to a file stored in a dependency that would be packaged by the build scripts:

The chunk of code that was written out was the actual malicious code, intended to be run on devices owned by the end users of Copay.

This code would do the following:

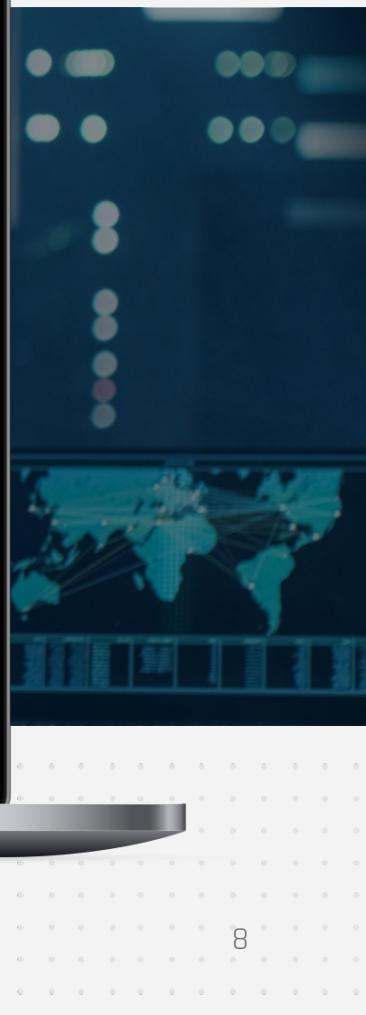
- Detect the current environment: Mobile/Cordova/Electron
- Check the Bitcoin and Bitcoin Cash balances on the victim's copay account
- If the current balance was greater than 100 Bitcoin, or 1000 Bitcoin Cash:
  - Harvest the victim's account data in full
  - Harvest the victim's copay private keys
  - Send the victim's account data/private keys off to a collection

# Attack vectors

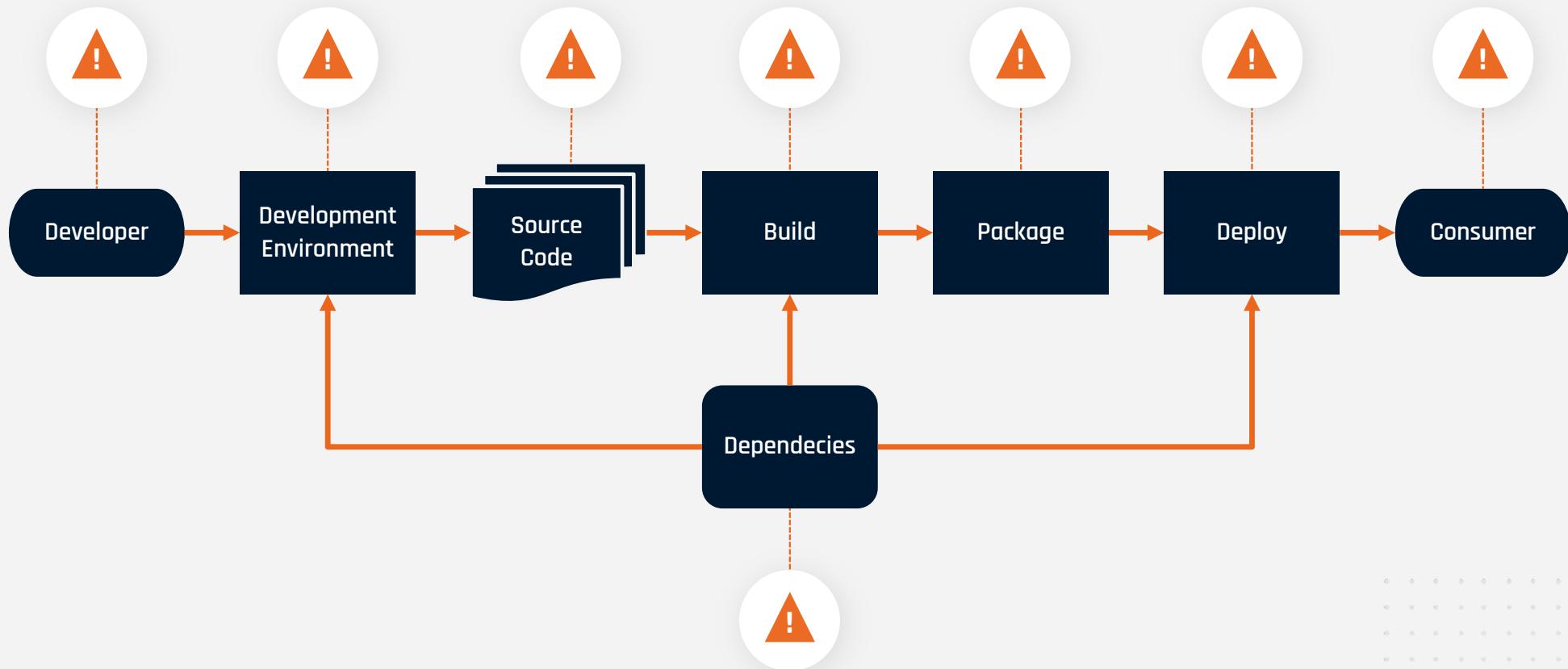




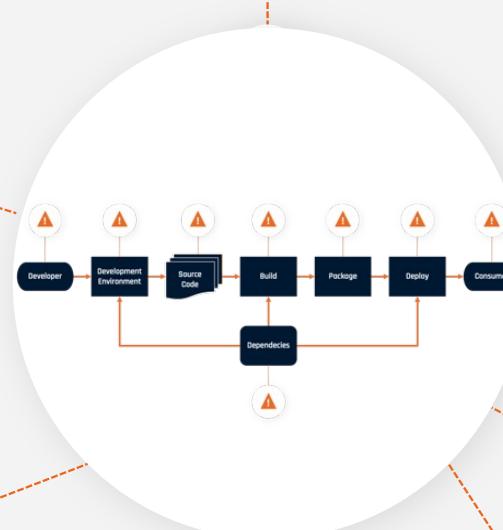
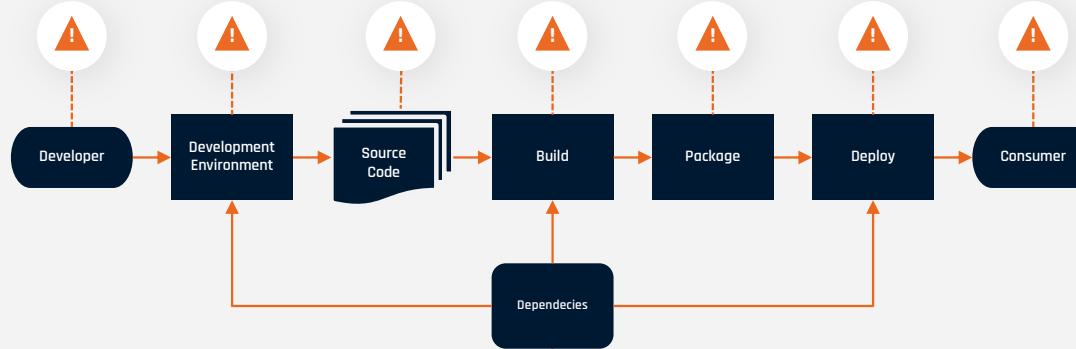
NoteBook



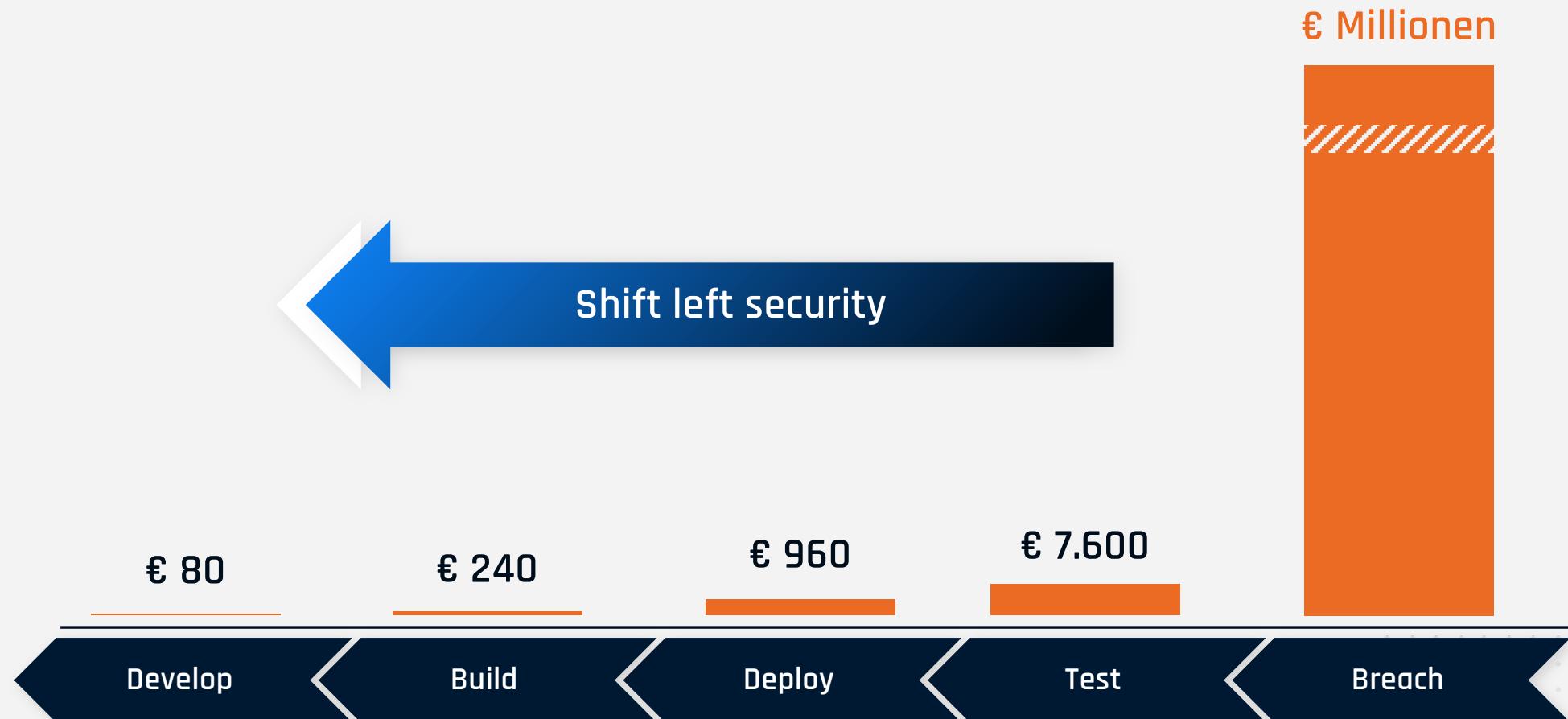
# Attack vectors



# Attack vectors



# Costs for fixing a security vulnerability



Source: Ponemon Institute Cost of a Data Breach 2020

# Supply Chain Security

# Software Composition Analysis (SCA)



GitHub (Dependency-Graph/Dependabot)



anchore (<https://anchore.com/>)



Dependency-Track  
(<https://dependencytrack.org/>)

Dependabot alerts

4 Open 0 Closed

Severity	Dependency	Last Updated	Manifest	Sort
Critical	marsdb	3 minutes ago by GitHub	package.json	
High	express-jwt	3 minutes ago by GitHub	package.json	
Moderate	sanitize-html	3 minutes ago by GitHub	package.json	
Critical	jsonwebtoken	3 minutes ago by GitHub	package.json	

Dependency graph

Dependencies Dependents Dependabot

We found potential security vulnerabilities in your dependencies.

Dependencies defined in these manifest files have known security vulnerabilities and should be updated:

package.json 7 vulnerabilities found

[View Dependabot alerts](#)

Only the owner of this repository can see this message.

These dependencies are defined in workshop-2021-learning-journey's manifest files, such as `package.json` and `frontend/package.json`.

Dependency	Vulnerability	Version
auth0 / express-jwt	Known security vulnerability	0.1.3
auth0 / node-jsonwebtoken	Known security vulnerability	0.4.0
c58 / marsdb	Known security vulnerability	0.6.11
apostrophecms / sanitize-html	Known security vulnerability	1.4.2
istanbuljs / istanbuljs		1.0.1
Seally / types-chai		4.2.14
DefinitelyTyped / DefinitelyTyped		7.1.3

# Dependency Management

## GitHub Dependency graph

► Dependabot **alerts**

► Dependabot  
**security updates**

► Dependabot  
**version updates**

The screenshot displays several GitHub Dependabot-related interfaces:

- Dependabot alerts:** A digest for the week of Jul 26 - Aug 2, showing a dependency update for `Newtonsoft.Json` from `< 13.0.1` to `~> 13.0.1`. It also lists a known vulnerability: `GHSA-5crp-9r3c-p9vr` (High severity).
- GitHub security alert digest:** A summary of security updates for `wulfland's` repository from Jul 26 - Aug 2, mentioning a pull request to bump `django` from `2.1.0` to `2.2.28`.
- SQL injection in Django #71:** A detailed view of a security alert for a Django injection vulnerability. It shows the alert was opened 2 months ago, has a critical severity (CVSS 9.8/10), and is associated with the package `django (pip)`. The affected version range is `>= 2.0.0, < 2.2.10`, and the patched version is `2.2.10`. The alert details the vulnerability: "Django 1.11 before 1.11.28, 2.2 before 2.2.10, and 3.0 before 3.0.3 allows SQL injection if untrusted data is used as a StringAgg delimiter (e.g., in Django applications that offer downloads of data as a series of rows with a user-specified column delimiter). By passing a suitably crafted delimiter to a contrib.postgres.aggregates.StringAgg instance, it was possible to break escaping and inject malicious SQL."
- Update requirements.txt #13:** A pull request to update `requirements.txt`. It shows a commit from `wulfland` adding `django-piston` (version 0.2.0) due to a high-severity vulnerability affecting `django-piston` and `django-tastypie`.

# Dependency Management

## GitHub Dependency graph

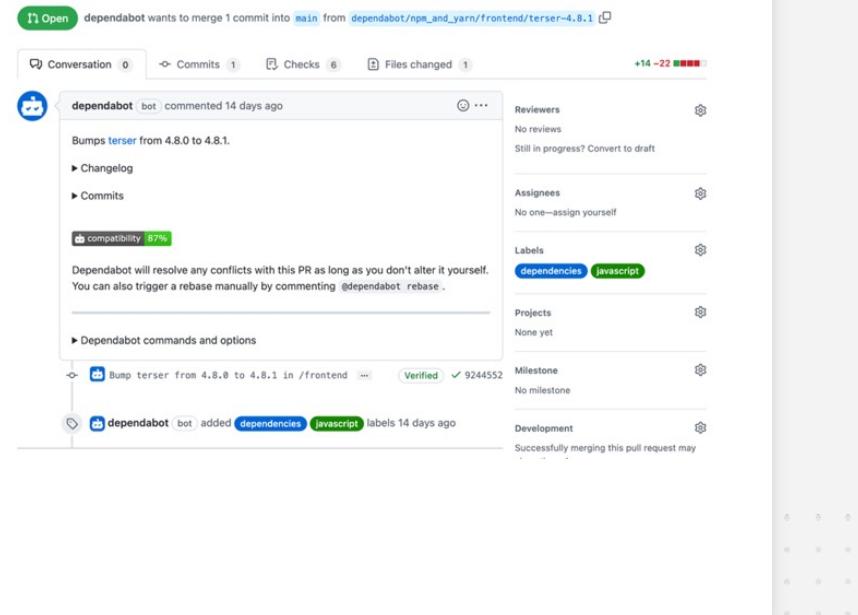
► Dependabot **alerts**

► Dependabot  
**security updates**

► Dependabot  
**version updates**

```
jobs:  
  dependabot:  
    runs-on: ubuntu-latest  
    if: ${{ github.actor == 'dependabot[bot]' }}  
    steps:  
      - name: Dependabot metadata  
        id: dependabot-metadata  
        uses: dependabot/fetch-metadata@v1.1.1  
        with:  
          github-token: "${{ secrets.GITHUB_TOKEN }}"  
  
      - name: Enable auto-merge for all patch versions  
        if: ${{steps.metadata.outputs.update-type == 'version-update:semver-patch'}}  
        run: gh pr merge --auto --merge "$PR_URL"  
        env:  
          PR_URL: ${{github.event.pull_request.html_url}}  
          GITHUB_TOKEN: ${{secrets.GITHUB_TOKEN}}
```

Bump terser from 4.8.0 to 4.8.1 in /frontend #50



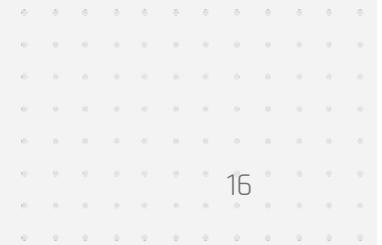
# Artifact attestations to establish provenance for builds

[actions/attest-sbom: Action for generating SBOM attestations for workflow artifacts \(github.com\)](#)

[actions/attest: Action for generating attestations for workflow artifacts \(github.com\)](#)

[Using artifact attestations to establish provenance for builds - GitHub Docs](#)

The SLSA framework is an industry standard used to evaluate supply chain security. It is organized into levels. Each level represents an increasing degree of security and trustworthiness for a software supply chain. Artifact attestations provides SLSA v1.0 Build Level 2.





## Demo: Demo: Dependabot

# Developer Security

# Attacking developers

Phishing / Spear Phishing

Social engineering

Unsecured connections to test systems

**"A developer is just a normal employee - that works as local admin, can push and execute code on various systems in minutes, and often runs unsecured web servers."**

# Phishing



Cgi [STAFF] shared "file" with you.

Reply all | Delete | Junk | ...

Cgi [STAFF] <dlawler@oakassociates.com>  
Mon 6/14, 1:33 PM  
Kaufmann, Michael

Deleted Items

This message was sent with high importance.

**EXTERNAL SENDER:** Do not click any links or open any attachments unless you trust the sender and know the content is safe.  
**EXPÉDITEUR EXTERNE:** Ne cliquez sur aucun lien et n'ouvrez aucune pièce jointe à moins qu'ils ne proviennent d'un expéditeur fiable, ou que vous ayez l'assurance que le contenu provient d'une source sûre.

Message from Cgi server.

Ho mes Limited.

**Cgi Share Document on SharePoint Groups**  
1 of your groups has new document for you

All Company

[redacted] @cgi.com at 2hrs +  
#contest

[Follow below to review this important document]  
[Preview Cgi Documents](#)

https://globalmail.ua.cgi.com/owa/projection.aspx

Reply all | Delete | Junk | ...

Urgent information about your April 2019 Deposit

Payroll Accounting <info@paymentreturn.com>  
Tue 5/25, 10:12 AM  
Kaufmann, Michael

**EXTERNAL SENDER:** Do not click any links or open any attachments unless you trust the sender and know the content is safe.  
**EXPÉDITEUR EXTERNE:** Ne cliquez sur aucun lien et n'ouvrez aucune pièce jointe à moins qu'ils ne proviennent d'un expéditeur fiable, ou que vous ayez l'assurance que le contenu provient d'une source sûre.

Dear Michael,

Unfortunately, we noticed during an internal review that there was an error in the last salary payment in April, this error was caused by an internal technical error which we have already corrected.

We apologize for the inconvenience caused.

Please check in the tool you know [Payroll Accounting](#), if your payment received matches the data on your statement. The adjusted payroll and the document for the following additional payment are available under the following link: [Payroll Accounting](#)

We apologize again for the error and wish them all the best and stay healthy.

Regards,  
Your payroll team

# Attacking developers

Typo squatting



Namespace shadowing



# Typo squatting

```
$ npm install crossenv
```



Steals all  
your  
environment  
variables

```
$ npm install cross-env
```



Normal  
package

# Namespace shadowing

```
$ npm install @azure/core-tracing
```

Normal  
package

```
$ npm install core-tracing
```

Upload data to  
a control server

# Credentials Developer

E-Mail	→ Spear phishing
Access machines	→ Log on, Mimikatz
Source	→ Inject code
Pipeline	→ Execute code / scripts
Access test environment	→ Test against prod?
Access prod?	

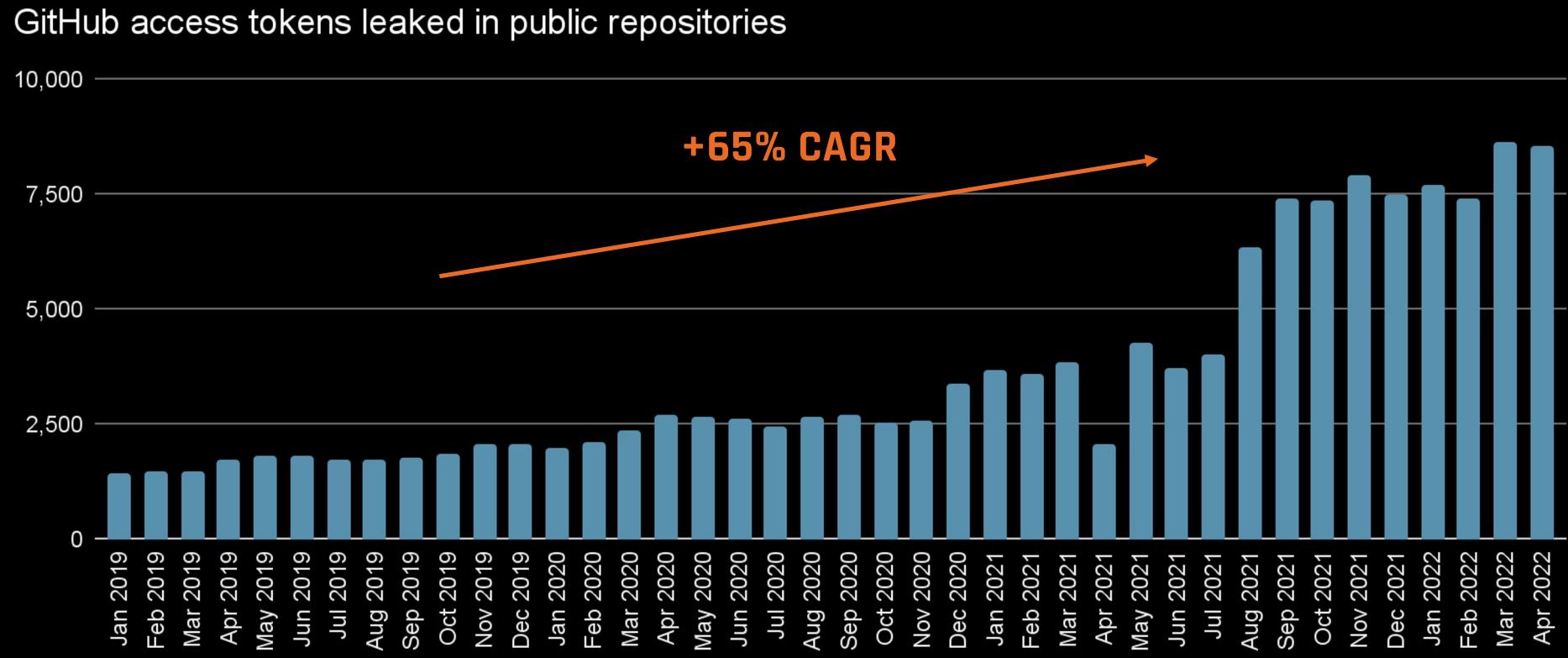


# Virtual DevEnv

- ▶ Virtual development environments
- ▶ Specific for project
- ▶ No local admin rights
- ▶ Codespaces / DevBox
- ▶ Secret scanning

# Secret Scanning

# We're seeing more credential leaks than ever



# Secret Scanning

## Code

- › GitHub Secret Scanning
- › gitLeaks
- › SpectralOps
- › Git-Secrets
- › Whispers
- › Gittyleaks
- › Git-all-secrets
- › ...

## Fileshare

- › Bash/PowerShell
- › Dumpster

Adafruit IO	Dropbox	Plivo
Adafruit IO Key	Dropbox Access Token	Plivo Auth Token
Adobe	Dropbox Short Lived Access Token	Postman
Adobe Device Token	Dynatrace	Postman API Key
Adobe JSON Web Token	Dynatrace Access Token	Proctorio
Adobe Service Token	Dynatrace Internal Token	Proctorio Consumer Key
Adobe Short-Lived Access Token	Finicity	Proctorio Linkage Key
Alibaba Cloud	Finicity App Key	Proctorio Registration Key
Alibaba Cloud Access Key ID and Access Key Secret pair	Frame.io	Proctorio Secret Key
Amazon Web Services (AWS)	Frame.io Developer Token	Pulumi
Amazon AWS Access Key ID and Secret Access Key pair	Frame.io JSON Web Token	Pulumi Access Token
Atlassian	GitHub	PyPI
Atlassian API Token	GitHub App Installation Access Token	PyPI API Token
Atlassian JSON Web Token	GitHub OAuth Access Token	RubyGems
Azure	GitHub Personal Access Token	RubyGems API Key
Azure Active Directory Application Secret	GitHub Refresh Token	Samsara
Azure DevOps Personal Access Token	GitHub SHS Private Key	Samsara API Token
Azure SAS Token	GoCardless	Samsara OAuth Access Token
Azure Service Management Certificate	GoCardless Live Access Token	SendGrid
Azure SQL Connection String	GoCardless Sandbox Access Token	SendGrid API Key
Azure Storage Account Key	Google Cloud	Shopify
Clojars	Google API Key	Shopify Access Token
Clojars Deploy Token	Google Cloud Private Key ID	Shopify App Shared Secret
CloudBees CodeShip	Hashicorp Terraform	Shopify Custom App Access Token
CloudBees CodeShip Credential	Terraform Cloud / Enterprise API Token	Shopify Private App Password
Databricks	Hubspot	Slack
Databricks Access Token	Hubspot API Key	Slack API Token
Datadog	Mailchimp	Slack Incoming Webhook URL
Datadog API Key	Mailchimp API Key	Slack Workflow Webhook URL
Discord	Mandrill API Key	SSLMate
Discord Bot Token	Mailgun	SSLMate API Key
Doppler	Mailgun API Key	SSLMate Cluster Secret
Doppler CLI Token	MessageBird	Stripe
Doppler Personal Token	MessageBird API Key	Stripe Live API Restricted Key
Doppler SCIM Token	npm	Stripe Live API Secret Key
Doppler Service Token	npm Access Token	Stripe Test API Restricted Key
	NuGet	Stripe Test API Secret Key
	OpenAI	Tencent Cloud
	Palantir	Tencent Cloud Secret ID
		Twilio
		Twilio Account String Identifier
		Twilio API Key
		Valour
		Valour Access Token



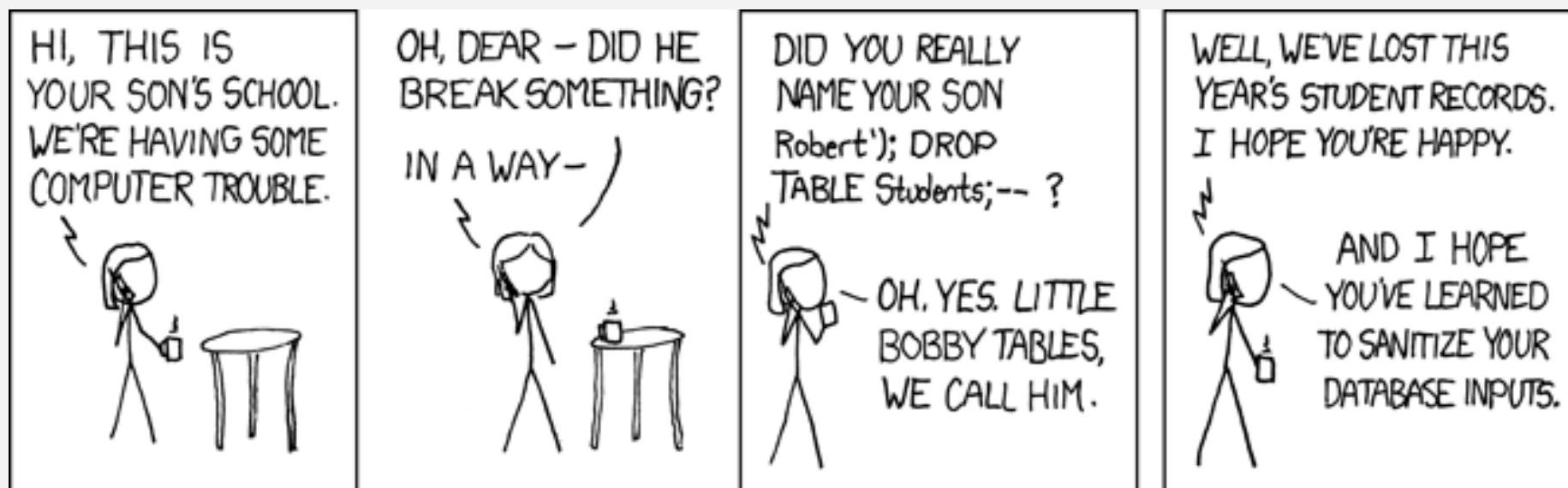
## Demo: Secret Scanning

# Code Security

# SQL Injection

```
txtUserId = getRequestId("UserId");
txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;
```

User ID: 105; DROP TABLE Suppliers



(Source: <https://xkcd.com/327/>)



4 years

On average, vulnerabilities go undetected for four years before being identified.  
Sometimes, even longer than that - Log4j was vulnerable for ~7 years

# Static Application Security Testing (SAST)

Analyze source code, bytecode, or binary code to identify security vulnerabilities in the application's codebase.

## ► Whitebox-Testing

- › GitHub Code Analysis
- › SonarQube, Fortify, Checkmarx, and Veracode
- › Semgrep (<https://semgrep.dev/>)
- › Mobile-Security-Framework (MobSF) (<https://github.com/MobSF/Mobile-Security-Framework-MobSF>)

The screenshot shows a GitHub Code Analysis interface. A code editor displays a file named `search-result.component.ts`. A specific line of code is highlighted with a red box, indicating a 'Cross-Site scripting vulnerability due to user-provided value'. The line of code is:

```
152     this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParam) // vuln-code-snippet vuln-line localXssChallenge
```

Below the code editor, a summary panel states: "Directly writing user input (for example, a URL query parameter) to a webpage without properly sanitizing the input first, allows for a cross-site scripting vulnerability." At the bottom, it says "First detected in commit 9474e68 3 days ago".

The screenshot shows a Semgrep analysis interface. A code editor displays the same `search-result.component.ts` file. A specific line of code is highlighted with a red box, indicating a 'Cross-Site scripting vulnerability due to user-provided value'. The line of code is identical to the one shown in the GitHub screenshot.

Below the code editor, a summary panel states: "Directly writing user input (for example, a URL query parameter) to a webpage without properly sanitizing the input first, allows for a cross-site scripting vulnerability. This kind of vulnerability is also called DOM-based cross-site scripting, to distinguish it from other types of cross-site scripting." It includes a 'Recommendation' section and an 'Example' section with sample code. At the bottom, it lists 'References' related to XSS and DOM-based XSS.

# Dynamic Application Security Testing (DAST)

Scans a running application to identify vulnerabilities by sending input requests and analyzing responses.



## Blackbox-Testing

- › OWASP ZAP ( Zed Attack Proxy, <https://owasp.org/www-project-zap> )
- › Burp Suite von PortSwigger ( <https://portswigger.net/burp> )

The screenshot shows the OWASP ZAP interface in Standard Mode. The top navigation bar includes 'Standard Mode', 'Sites', 'Contexts', 'HUB Contexts', and 'Sites'. Below this is a tree view of 'Sites' containing various URLs. The main pane displays a 'Text' dump of network traffic, showing several 'alert added' events. At the bottom, there's a table of captured messages with columns for 'Channel', 'Timestamp', 'Opcode', 'Bytes', and 'Payload'. The 'Payload' column shows JSON objects representing the alerts. A status bar at the bottom indicates 'Alerts: 0', 'Primary Proxy: localhost:8080', and 'Current Scans: 0'.

The screenshot illustrates the OWASP ZAP process. On the left, a browser window shows a 'Not Secure' warning for 'https://xyz-demo-shop.azurewebsites.net'. An annotation points to the ZAP icon in the toolbar with the text 'OWASP ZAP intercepts the traffic'. The main area shows a 'TAILWIND TRADERS' website with a shopping cart and promotional banners. A large annotation covers the interface with the text 'Head-Up-Display (HUB) to analyze and attack the site using the spider'. Another annotation on the right side of the interface says 'Items can be customized'. The bottom of the interface shows a 'History' tab with a count of 37, a 'WebSockets' tab, and a status bar indicating '0' for various metrics.

# Dynamic Application Security Testing (DAST)



## Blackbox-Testing

- › OWASP ZAP ( Zed Attack Proxy, <https://owasp.org/www-project-zap> )
- › Burp Suite von PortSwigger ( <https://portswigger.net/burp> )

Q OWASP ZAP Sort: Best Match

3 results for "OWASP ZAP"

Actions

- OWASP ZAP Full Scan By zaproxy 2022-01-09 11:45 115 stars
- OWASP ZAP API Scan By zaproxy 2022-01-09 11:45 2 stars
- OWASP ZAP Baseline Scan By zaproxy 2022-01-09 11:45 188 stars

jobs:

```
name: OWASP Full Scan
runs-on: ubuntu-latest
steps:
  - name: OWASP ZAP Full Scan
    uses: zaproxy/action-full-scan@v0.2.0
    with:
      # GitHub Token to create issues in the repository
      token: ${{ github.token }}
      target: https://target
```

ZAP Scanning Report

Sites: <http://xyz-demo-shop.azurewebsites.net> <https://xyz-demo-shop.azurewebsites.net>

Generated on Sun, 9 Jan 2022 19:16:45

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	3
Low	8
Informational	3
False Positives	0

Alerts

Name	Risk Level	Number of Instances
Content Security Policy (CSP) Header Not Set	Medium	3
Missing Anti-clickjacking Header	Medium	3
Proxy Disclosure	Medium	18
Cookie with SameSite Attribute None	Low	2
Cookie without SameSite Attribute	Low	2
HTTP Content Available via HTTP	Low	11
Incomplete or No Cache-control Header Set	Low	5
Private IP Disclosure	Low	1
Strict-Transport-Security Header Not Set	Low	11
Timestamp Disclosure - Unix	Low	20
X-Content-Type-Options Header Missing	Low	11
Cookie Stack Detector	Informational	18
Informational: Suspicious Comments	Informational	2
Modern Web Application	Informational	4

Alert Detail

Medium	Content Security Policy (CSP) Header Not Set
Description	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate including Cross Site Scripting (XSS) and data injection attacks. These attacks are used to to alter the user's displayed content. CSP provides a way for web application owners to declare approved sources of content that browsers should be allowed to load content types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as audio and video files.
URL	<a href="https://xyz-demo-shop.azurewebsites.net/">https://xyz-demo-shop.azurewebsites.net/</a>

A report is added as a build artifact in HTML, JSON, and Markdown

A GitHub Issue is created with a link to the workflow run

ZAP Full Scan Report #173

github-actions bot opened this issue 21 hours ago · 0 comments

Contributor

Site: <http://xyz-demo-shop.azurewebsites.net>

Site: <https://xyz-demo-shop.azurewebsites.net>

New Alerts

- Content Security Policy (CSP) Header Not Set [10038] total: 3:
  - <https://xyz-demo-shop.azurewebsites.net/>
  - <https://xyz-demo-shop.azurewebsites.net/robots.txt>
  - <https://xyz-demo-shop.azurewebsites.net/sitemap.xml>
- Missing Anti-clickjacking Header [10020] total: 3:
  - <https://xyz-demo-shop.azurewebsites.net/>
  - <https://xyz-demo-shop.azurewebsites.net/robots.txt>
  - <https://xyz-demo-shop.azurewebsites.net/sitemap.xml>
- Proxy Disclosure [40025] total: 18:
  - <https://xyz-demo-shop.azurewebsites.net>
  - <https://xyz-demo-shop.azurewebsites.net/>
  - <https://xyz-demo-shop.azurewebsites.net/apple-touch-icon.png>
  - <https://xyz-demo-shop.azurewebsites.net/favicon-16x16.png>
  - <https://xyz-demo-shop.azurewebsites.net/favicon-32x32.png>

# Runtime Application Self-Protection (RASP)

Monitors and protects applications in real-time, detecting and responding to security threats as they occur.



## Blackbox-Testing

- › Veracode Runtime Protection
- › F5 Advanced WAF with RASP
- › Datadog Application Security Management

The dashboard displays a chart of Attacks over time (04/24 to 05/24) with a peak around May 9th. A pie chart shows Attack Types: SQL Injection (356 total attacks), XSS - Cross-site Scripting, and others. Below is an Event Log table with columns: Status, ID, Date, Event Type, Server, Request Path, Origin IP, Rows, and User. It lists several protected events from May 19th, 2022.

Status	ID	Date	Event Type	Server	Request Path	Origin IP	Rows	User
Protected	76497784-6ab0-4c2	5/19/16 3:16 PM	Multiple Events	testapp-1 prot.verac...	/www/active-Reflected.XSS(RD)	10.0.1.100:443	0	
Protected	7a7550a0-4a07-42d	5/19/16 3:03 PM	Multiple Events	testapp-1 prot.verac...	/www/active-Reflected.XSS(RD)	10.0.1.100:443	0	
Protected	c9880050-7e10-430	5/19/16 2:52 PM	Multiple Events	testapp-1 prot.verac...	/www/active-Reflected.XSS(RD)	10.0.1.100:443	0	
Unprotected	798477fa-f719-44f	5/19/16 2:51 PM	Multiple Events	testapp-1 prot.verac...	/testapp-master.SNAPSHOT/test	10.0.1.100:443	0	

A screenshot of the Application Security interface. A critical signal for an SSRF vulnerability is shown, triggered by a malicious URL tampering attack on the auth-dotnet service. The signal details include the compromised URL (`http://localhost/`) and the reason (A user parameter controlled the domain part of the URL and directed it to a sensitive resource). A sample attack flow diagram shows the flow from a compromised URL through various services like net/http, web-store-mongo, product-recommendation, and auth-dotnet to the final target. Suggested next steps include blocking the attacker IP at the edge (WAF/CDN) and declaring an incident if escalation is needed.

# Infrastructure Scanning

## Container Vulnerability Analysis (CVA) / Container Security Analysis (CSA)

### Open source:

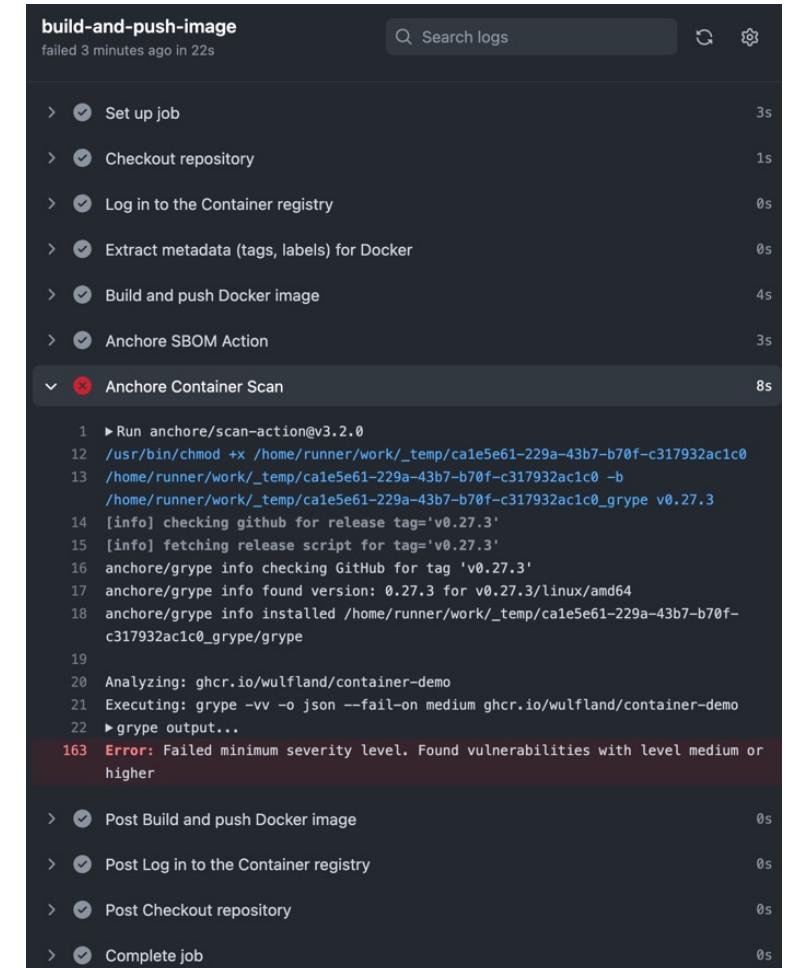
- › Anchore gryp  
<https://github.com/anchore/grype/>
- › Clair  
<https://quay.github.ioclair/>

### Commercial:

- › WhiteSource  
<https://www.whitesourcesoftware.com/solution-for-containers/>
- › Aqua  
<https://www.aquasec.com/products/container-security/>

```
- name: Anchore Container Scan
  uses: anchore/scan-action@v3.2.0
  with:
    image: ${{ env.REGISTRY }}/{{ env.IMAGE_NAME }}
    debug: true
```

<https://github.com/wulfland/container-demo/actions/runs/2179243137>



```
build-and-push-image
failed 3 minutes ago in 22s
Search logs

> ✓ Set up job 3s
> ✓ Checkout repository 1s
> ✓ Log in to the Container registry 0s
> ✓ Extract metadata (tags, labels) for Docker 0s
> ✓ Build and push Docker image 4s
> ✓ Anchore SBOM Action 3s
-> ✘ Anchore Container Scan 8s
  1 ► Run anchore/scan-action@v3.2.0
  12 /usr/bin/chmod +x /home/runner/work/_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0
  13 /home/runner/work/_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0 -b
  14 /home/runner/work/_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0_grype v0.27.3
  15 [info] checking github for release tag='v0.27.3'
  16 [info] fetching release script for tag='v0.27.3'
  17 anchore/grype info checking GitHub for tag 'v0.27.3'
  18 anchore/grype info found version: 0.27.3 for v0.27.3/linux/amd64
  19 anchore/grype info installed /home/runner/work/_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0_grype/grype
  20 Analyzing: ghcr.io/wulfland/container-demo
  21 Executing: grype -vv -o json --fail-on medium ghcr.io/wulfland/container-demo
  22 ► grype output...
  163 Error: Failed minimum severity level. Found vulnerabilities with level medium or
higher

> ✓ Post Build and push Docker image 0s
> ✓ Post Log in to the Container registry 0s
> ✓ Post Checkout repository 0s
> ✓ Complete job 0s
```

# Infrastructure Scanning

## Infrastructure policies

### Open source:

- › Checkov  
<https://www.aquasec.com/products/container-security/>
- › OpenVAS

### Commercial:

- › Defender for Cloud  
<https://azure.microsoft.com/en-us/services/defender-for-cloud>
- › Azure Policy  
<https://docs.microsoft.com/de-de/azure/governance/policy/>

```
- name: Checkov GitHub Action
  uses: bridgecrewio/checkov-action@master
  with:
    directory: ch15_sec/
    output_format: sarif

- name: Upload SARIF file
  uses: github/codeql-action/upload-sarif@v1
  with:
    sarif_file: results.sarif
    if: always()
```

## Code scanning

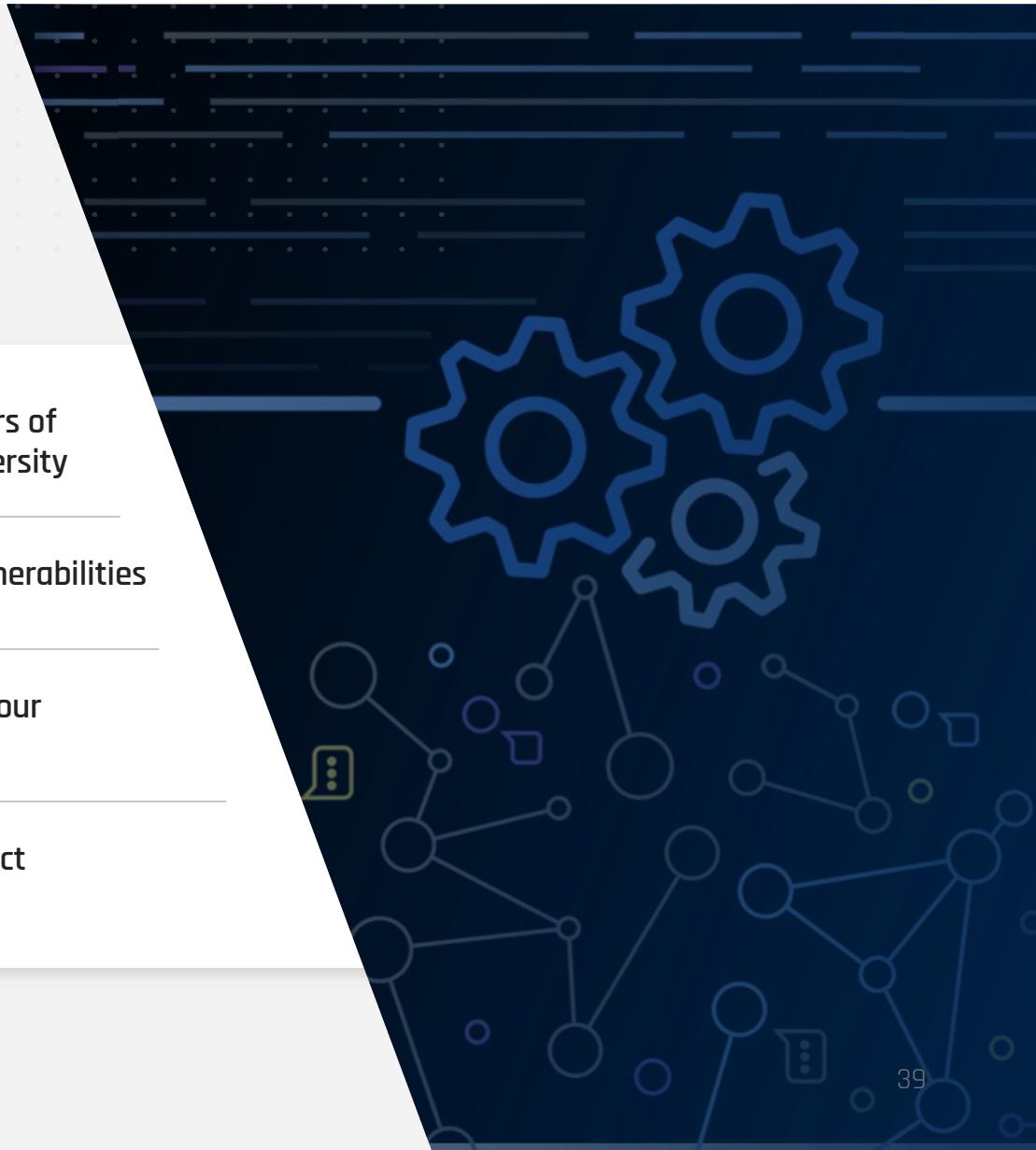
Latest scan	Branch	Workflow	Lines scanned	Duration	Result
10 minutes ago	main	CodeQL	1.45k / 1.39k ⓘ	5m 26s	21 alerts

Filters  tool:checkov is:open branch:main  
 Clear current search, filters and sorts

✓ 2 Open	✗ 0 Closed	Tool	Rule	Branch	Severity	Sort
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Ensure that S3 bucket has a Public Access block	Error	aws.tf:1	Detected 15 minutes ago by checkov	main
<input type="checkbox"/>	<input type="checkbox"/>	Ensure that S3 bucket has cross-region replication enabled	Error	aws.tf:1	Detected 15 minutes ago by checkov	main

# CodeQL: a revolutionary semantic code engine

- ▶ Advanced code analysis engine based on 13 years of research by a 30 person team from Oxford University
- ▶ Allows you to query your code's logic to find vulnerabilities
- ▶ Queries can be quickly customized to adapt to your specific threat topology
- ▶ Community-driven query set powers every project with a world-class security team





## Demo: Code analysis (Auto-Fix)

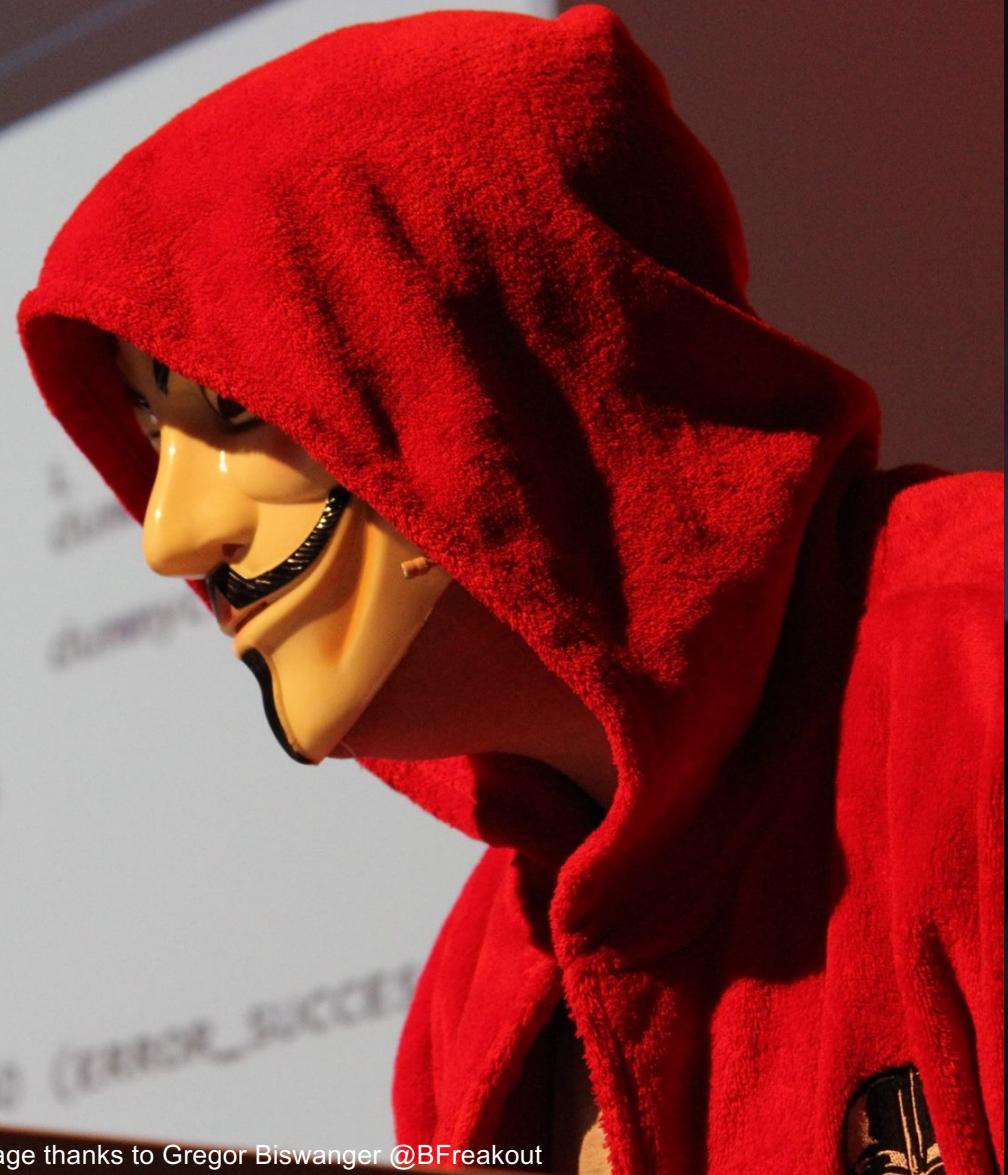
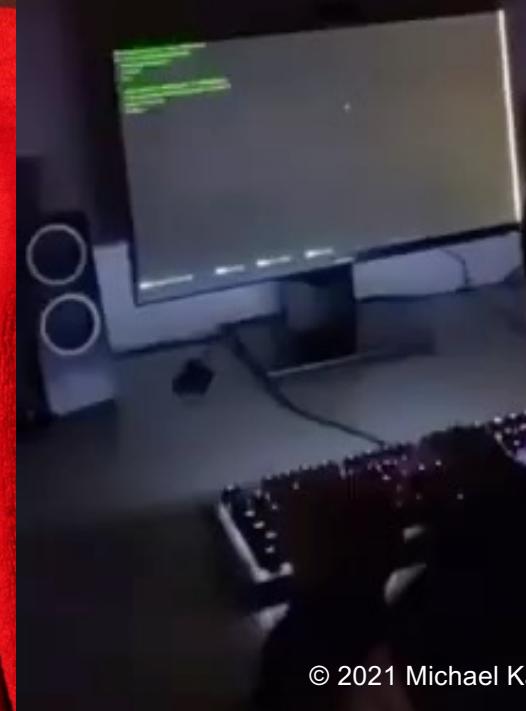


Image thanks to Gregor Biswanger @BFreakout

## Hackers in movies vs



© 2021 Michael Kaufmann @mike\_kaufmann

# Thank you



Blog : <https://writeabout.net>



Twitter : @mike\_kaufmann



GitHub : @wulfland



LinkedIn : <https://www.linkedin.com/in/mikaufmann/>