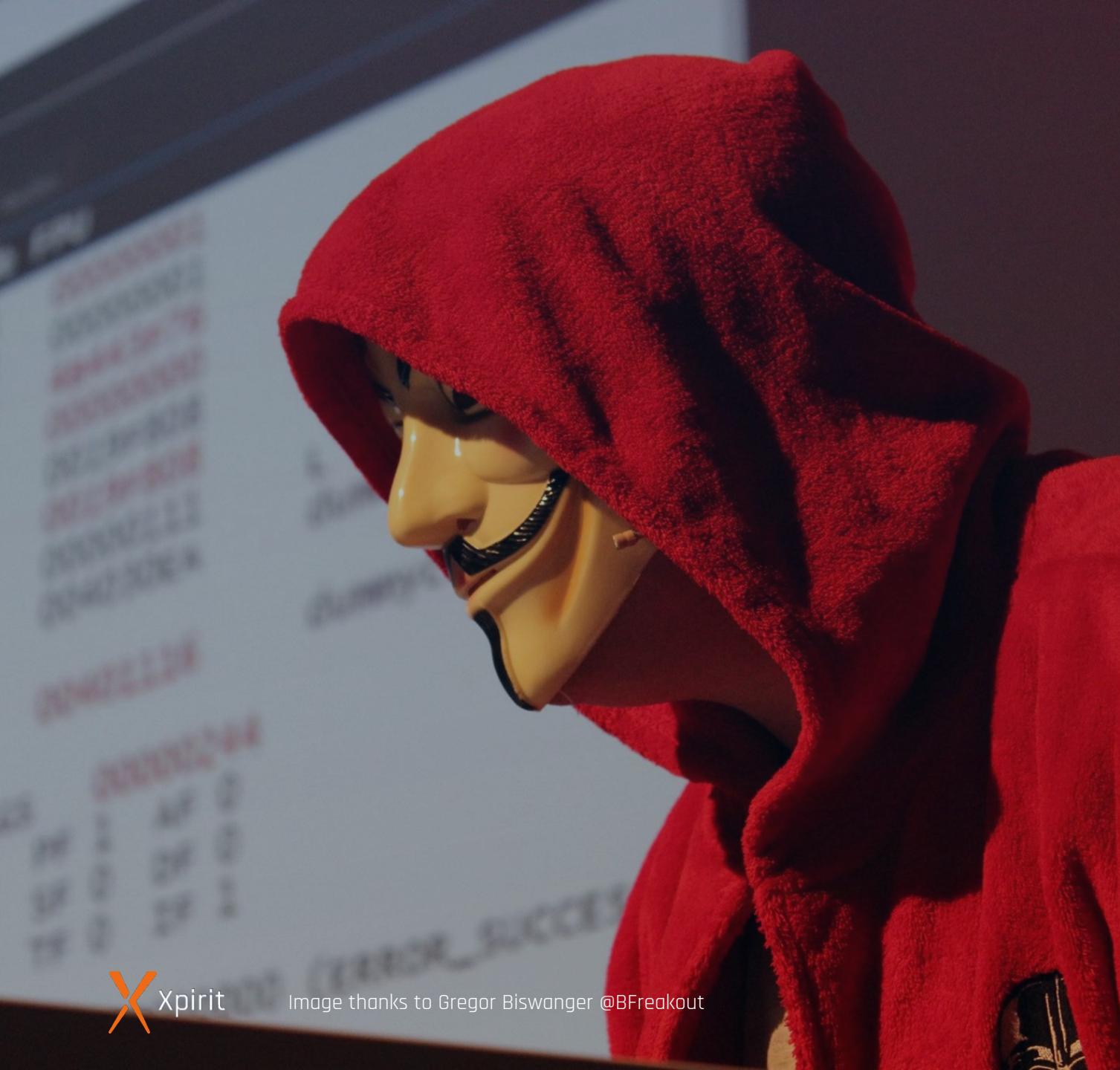




# { DevSecOps }

---

Application Security from  
start to finish



Hackers in movies  
vs



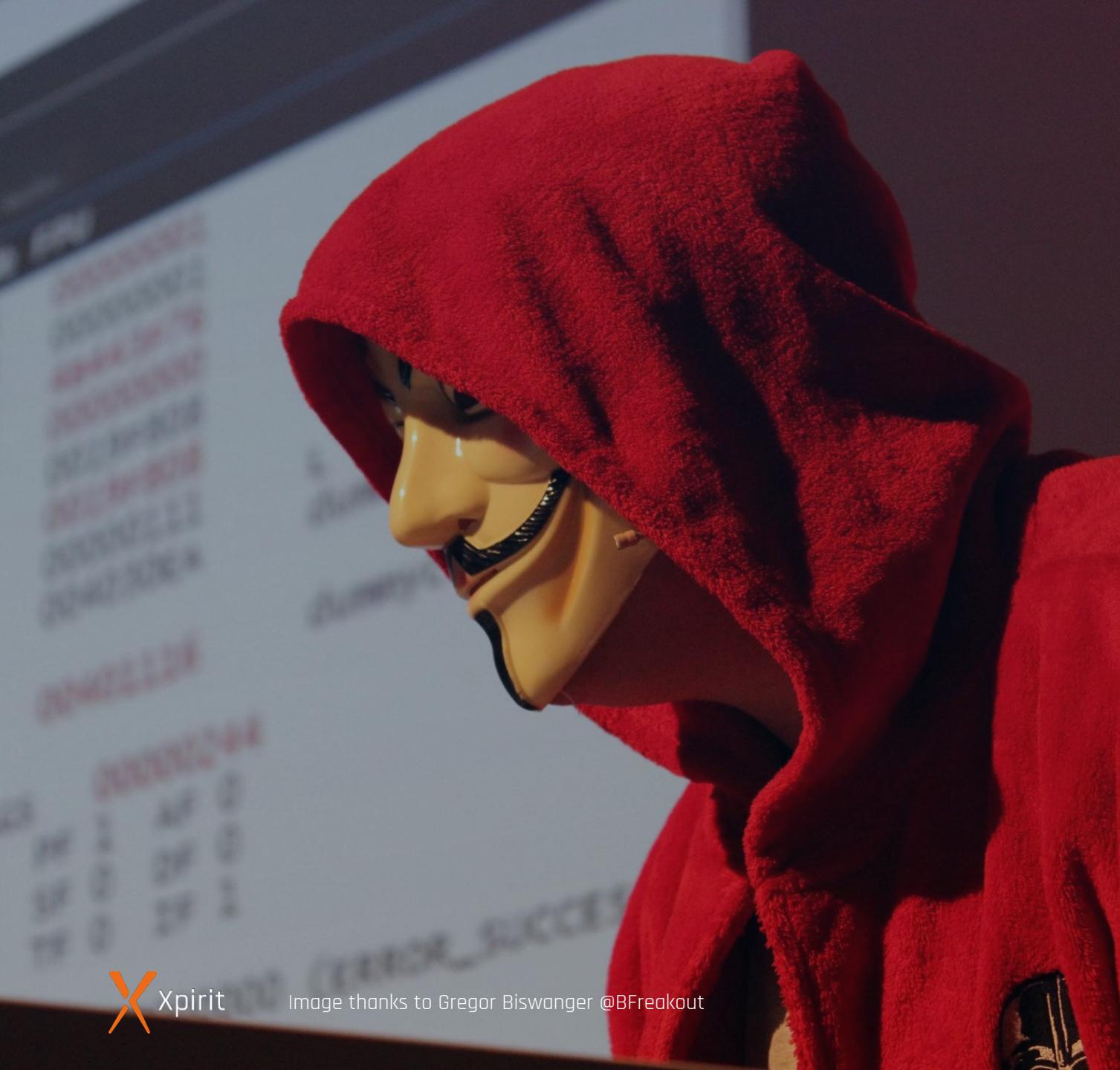


Image thanks to Gregor Biswanger @BFreakout



## How people think they get hacked



## How they really get hacked

The image is a collage of several social media posts and memes. It includes:

- A post from "94.1 KXQJ 9" asking, "If you had to marry your spouse where you met them, where would your wedding have been?" with 1004 comments and 234 shares.
- A post from "Up 99.3" asking, "You can call me Down Outlaw Supreme! Your name: Shannon" with 26.4K comments and 8.8K shares.
- A meme asking, "How far away do you live from the place you were born?" with 20.8K comments and 8.8K shares.
- A post from "NAME A SONG" asking, "The car you passed your drivers test in was a \_\_\_\_\_" with 14.8K comments and 6.8K shares. Below it, a note says: "Asking where you'd get married again is like asking where you met. A 'porn name' that exposes your middle name AND a street you grew up on are TWO pieces of info that people need. STOP. THINK. DO NOT SHARE INFO."
- A meme titled "THAT TAKES YOU BACK TO HIGH SCHOOL" showing a group of people from the 1980s.

# 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

Vulnerability details Dependabot alerts 0

Package flatmap-stream ( npm ) Affected versions = 0.1.1

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

Follow

right9ctrl 東京都 Committed to this repository

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

CWEs CWE-506

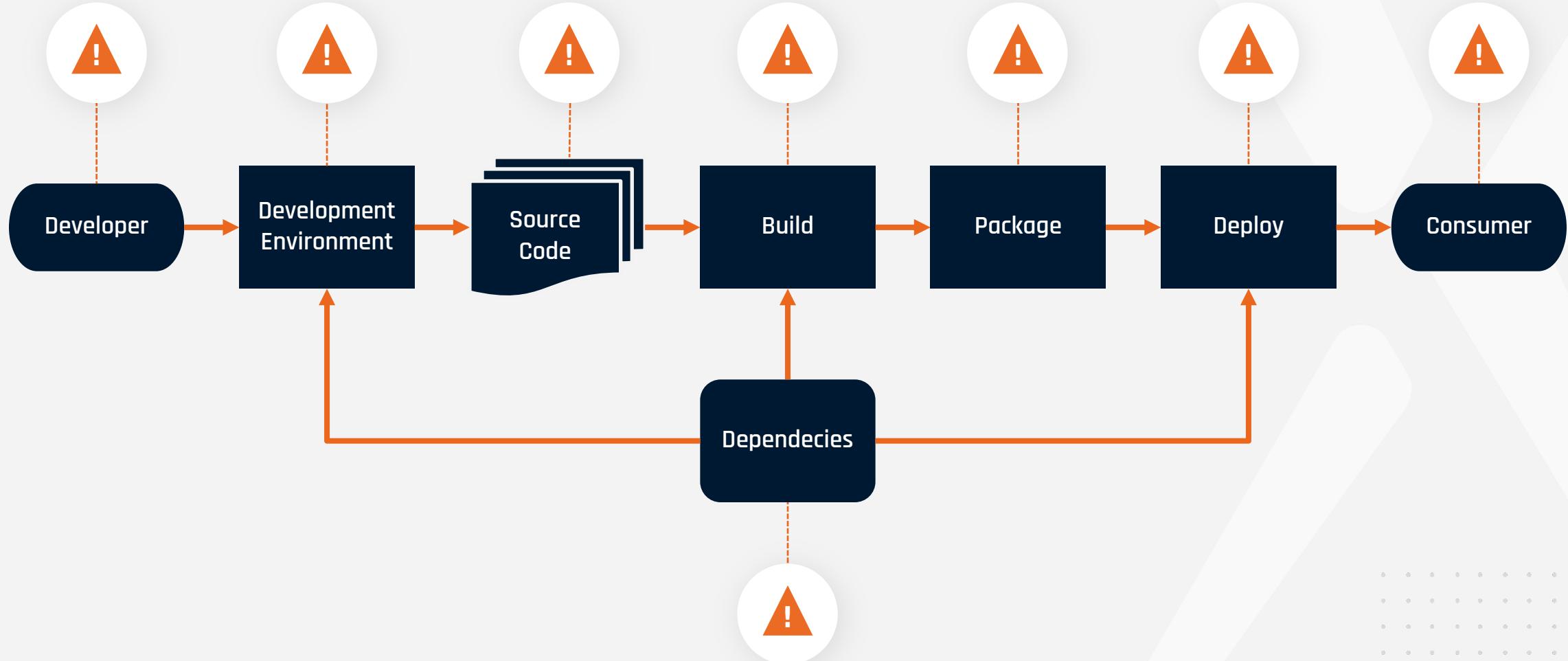
CVSS Score 9.8 Critical CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H

This advisory has been edited. See History.

See something to contribute? Suggest improvements for this vulnerability



# Attack vectors





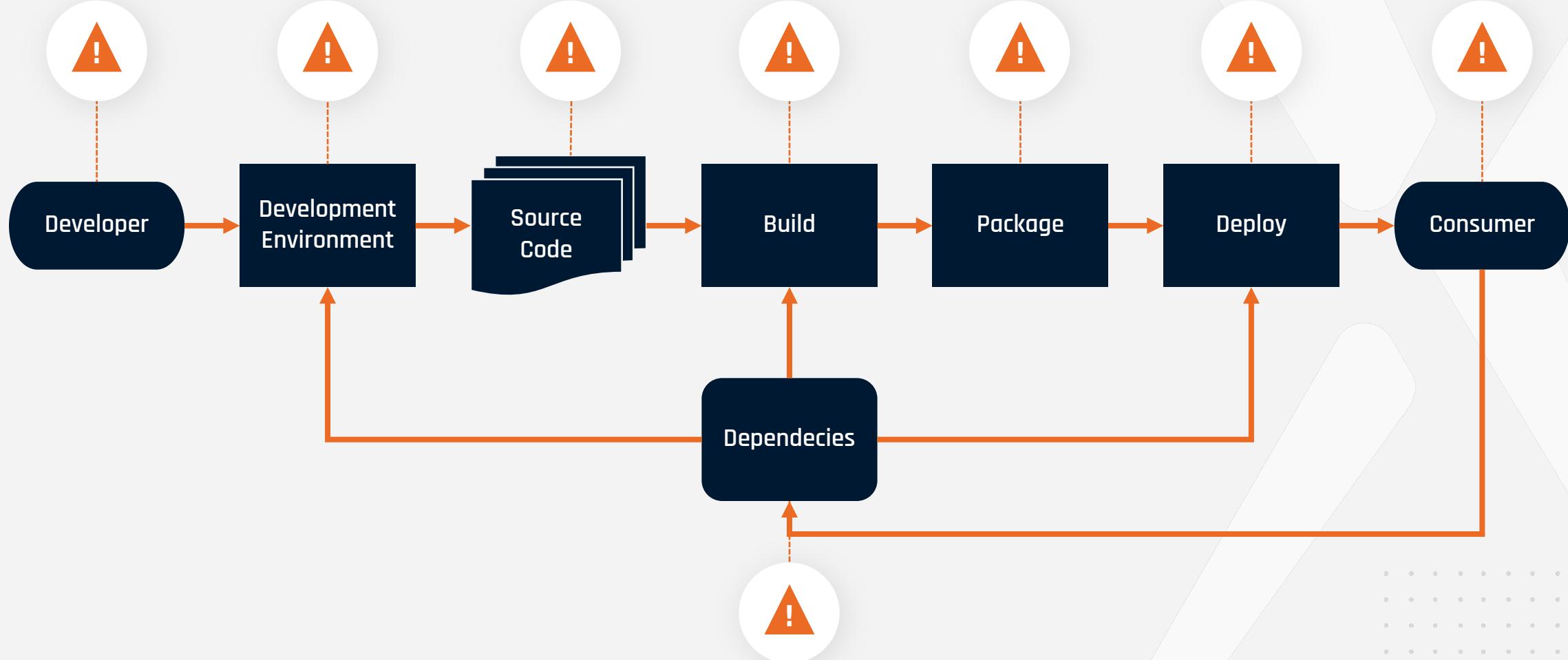
odule Graph ⓘ gatsby

Off Include devDependencies (→)  
Colorize by: Nothing (uncolored) ▾

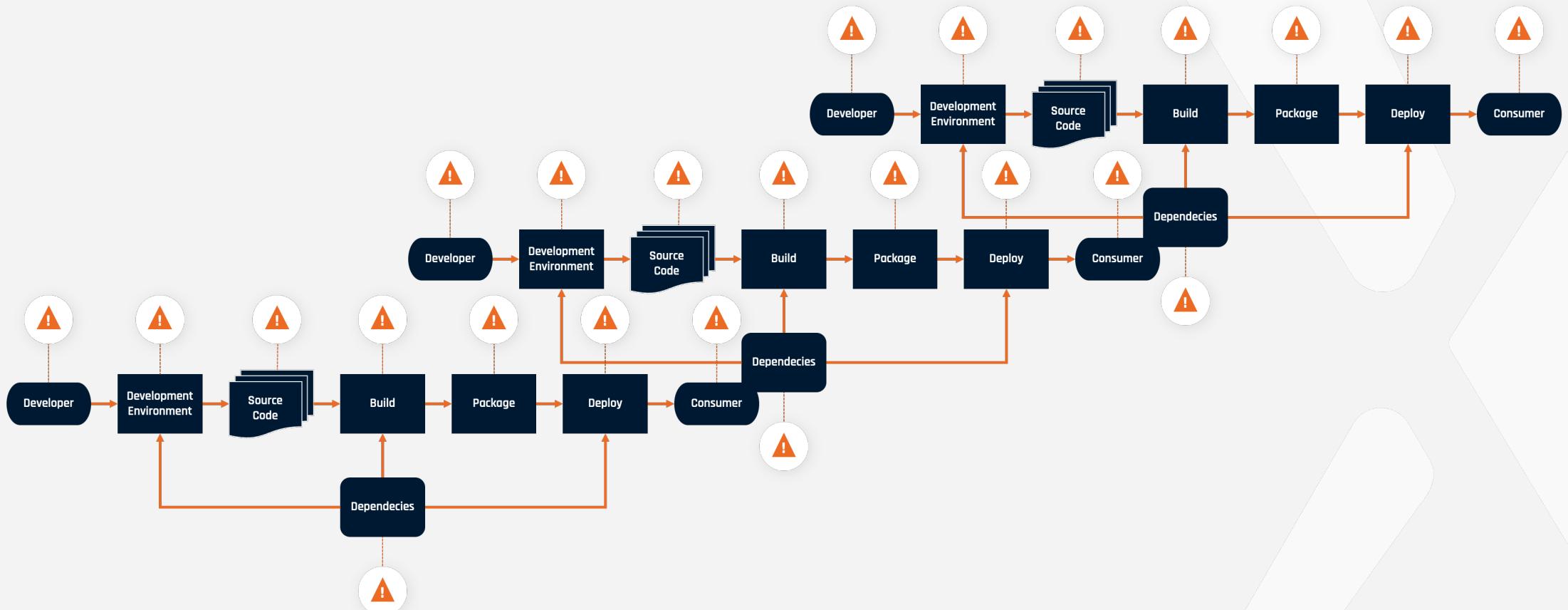
## 1189 Modules

- |                   |                   |                   |
|-------------------|-------------------|-------------------|
| @ampproject/r...  | @ardatan/aggr...  | @babel/code-f...  |
| @babel/compa...   | @babel/core       | @babel/eslint-... |
| @babel/genera...  | @babel/helper...  | @babel/helper...  |
| @babel/helper...  | @babel/helper...  | @babel/helper...  |
| @babel/helper-... | @babel/helper...  | @babel/helper...  |
| @babel/helper...  | @babel/helper...  | @babel/helper...  |
| @babel/helper...  | @babel/helper...  | @babel/helper...  |
| @babel/helper...  | @babel/helper...  | @babel/helper...  |
| @babel/helper-... | @babel/helper...  | @babel/helper...  |
| @babel/helpers    | @babel/highlight  | @babel/parser     |
| @babel/plugin-... | @babel/plugin-... | @babel/plugin-... |
| @babel/preset...  | @babel/preset...  | @babel/preset...  |
| @babel/runtime    | @babel/runtim...  | @babel/template   |

# Attack vectors



# Attack vectors



# Losses caused by cyber attacks reported to IC3



## Top 5 crime types:

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

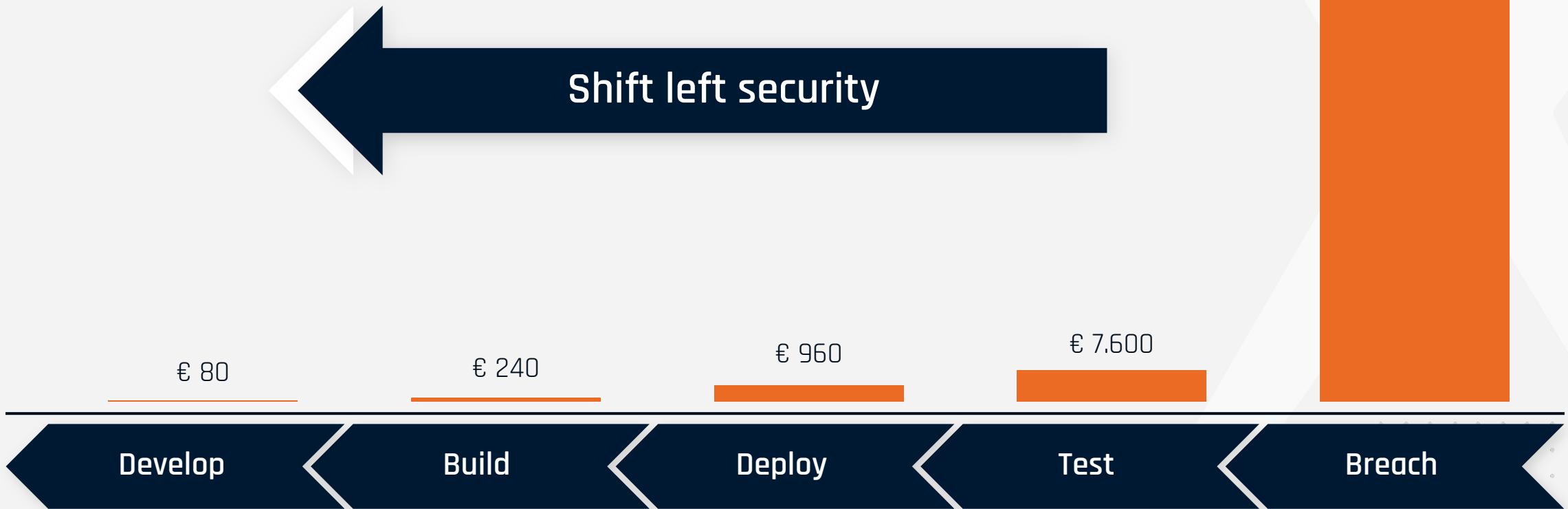


## Trends

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

# Costs for fixing a security vulnerability

€ Millionen



# Attack vector: developer

# 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

The image displays two side-by-side screenshots of phishing emails. Both emails are presented in a web-based email client interface.

**Email 1 (Left):** Subject: Cgi [STAFF] shared "file" with you. From: Cgi [STAFF] <dlawler@oakassociates.com>. Date: Mon 6/14, 1:33 PM. Body:

Cgi [STAFF] shared "file" with you.

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

@cgi.com at 2hrs +  
#contest

[Follow below to review this important document]

Preview Cgi Documents

**Email 2 (Right):** Subject: Urgent information about your April 2019 Deposit. From: Payroll Accounting <info@paymentreturn.com>. Date: Tue 5/25, 10:12 AM. Body:

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

# Credentials Developer



# Attacking developers

- ▶ Typo squatting

---

- ▶ Namespace shadowing

---



# Typo squatting

```
$ npm install crossenv
```



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



Steals all  
your  
environment  
variables

Normal  
package



# Namespace shadowing

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



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



Normal  
package

Upload data to  
a control server



# Typo squatting

- ▶ Attack supply chain at build time (npm install)
- ▶ Attack consumer at run time by shadowing a function
- ▶ Version ranges in transient dependencies can delay attack



# What to do?

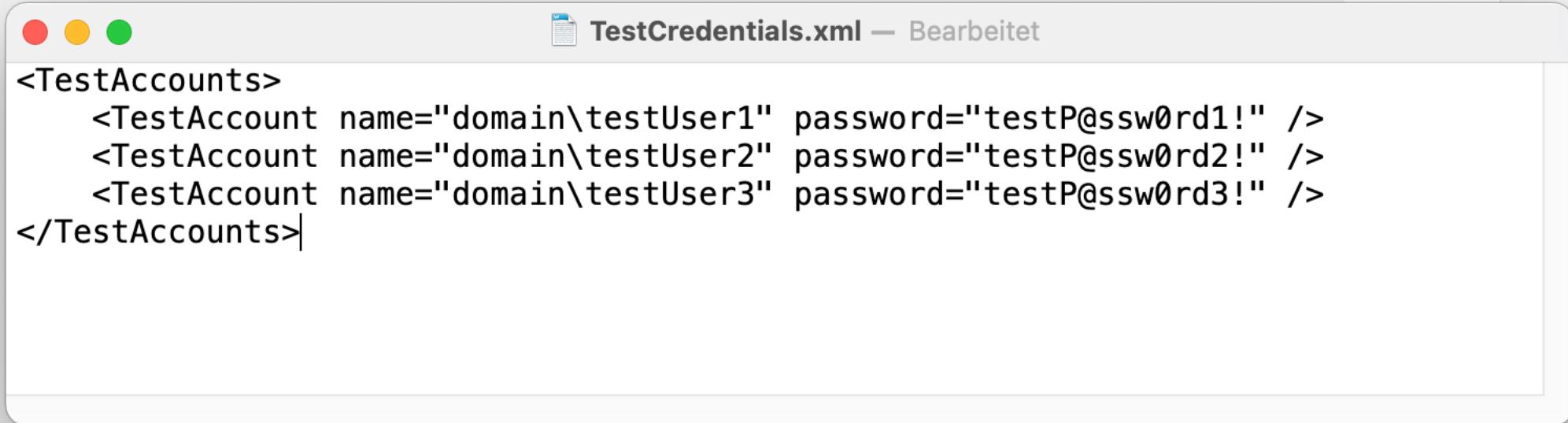
- ▶ Security Awareness Trainings
- ▶ Security Games
- ▶ Red team | blue team simulations
- ▶ Ephemeral, containerized environments

# Attack vector: dev environment

# Attack vector: dev environment

- ▶ Passwords in text files / memory
- ▶ Mimikatz
- ▶ Build tools
- ▶ Modify code
- ▶ Modify pipeline / execute code

# Credentials in text files



The screenshot shows a text editor window titled "TestCredentials.xml — Bearbeitet". The XML code inside the window defines three test accounts:

```
<TestAccounts>
    <TestAccount name="domain\testUser1" password="testP@ssw0rd1!" />
    <TestAccount name="domain\testUser2" password="testP@ssw0rd2!" />
    <TestAccount name="domain\testUser3" password="testP@ssw0rd3!" />
</TestAccounts>
```

# Credentials in text files

```
{  
  "/api/*": {  
    "changeOrigin": true,  
    "target": "https://api.project-demo.de",  
    "auth": "REDACTED"  
  }  
}
```

```
const id_token_string = 'REDACTED';  
  
app.use(  
  '/api/test',  
  createProxyMiddleware({  
    target: 'https://api.test.REDACTED.com/',  
    changeOrigin: true,  
    pathRewrite: {  
      '^/api/test': '/test',  
    },  
    headers: {  
      'Content-Type': 'application/json',  
      'Credentials': true,  
      'Cookie': `id_token=${id_token_string}`  
    },  
  });
```

Schema: <https://json.schemastore.org/appsettings.json>

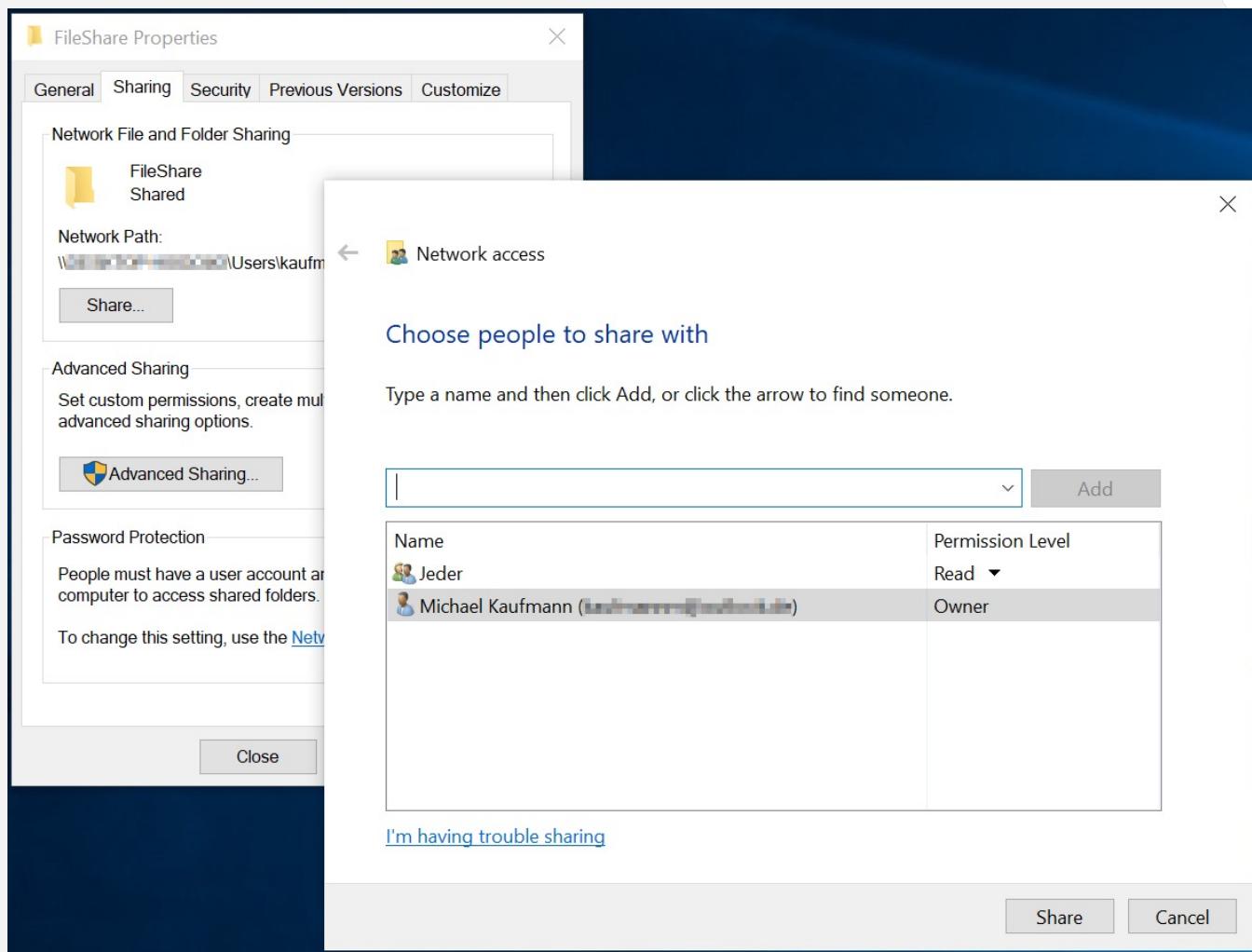
```
1  {  
2    "AppSettings": {  
3      "ScheduleExecutionIntervalInMinutes": 60  
4    },  
5    "MailSettings": {  
6      "Url": "Outlook.office365.com",  
7      "User": "REDACTED",  
8      "Password": "REDACTED",  
9      "ProcessedFolderName": "REDACTED",  
10     "ErrorFolderName": "REDACTED",  
11     "OutOfToleranceFolderName": "REDACTED"  
12   }  
13 }
```

# Credentials in text files

27 lines (27 sloc) | 1.45 KB

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3      <configSections>
4          <sectionGroup name="applicationSettings" type="System.Configuration.ApplicationSettingsG
5              <section name="SPOEmulators.Tests.Properties.Settings" type="System.Configuration.Cl
6          </sectionGroup>
7      </configSections>
8      <applicationSettings>
9          <SPOEmulators.Tests.Properties.Settings>
10             <setting name="OnPremUrl" serializeAs="String">
11                 <!-- Enter thr url to a on prem site for integration testing. -->
12                 <value>https://localhost/sites/dev</value>
13             </setting>
14             <!-- Enter thr url to a 0365 site for integration testing. -->
15             <setting name="0365Url" serializeAs="String">
16                 <value>http://xxxx.sharepoint.com</value>
17             </setting>
18             <!-- Enter credentials to connect to the site (0365 or on prem if neccessary) -->
19             <setting name="0365User" serializeAs="String">
20                 <value>user@tenant.onmicrosoft.com</value>
21             </setting>
22             <setting name="0365Password" serializeAs="String">
23                 <value>*****</value>
24             </setting>
25         </SPOEmulators.Tests.Properties.Settings>
26     </applicationSettings>
27 </configuration>
```

# Unsecured file shares / visible repositories



⚡ kaufm@DESKTOP-HI0G080 ➤ ~\OneDrive\Downloads\mimikatz\_trunk\x64  
› .\mimikatz.exe

```
.#####. mimikatz 2.2.0 (x64) #19041 May 31 2021 00:08:47
.## ^ ##. "A La Vie, A L'Amour" - (oe.eo)
## / \ ## /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##      > https://blog.gentilkiwi.com/mimikatz
'## v ##'      Vincent LE TOUX          ( vincent.letoux@gmail.com )
'#####'      > https://pingcastle.com / https://mysmartlogon.com ***/
```

```
mimikatz # privilege::debug
Privilege '20' OK
```

```
mimikatz # sekurlsa::logonpasswords
```

```
Authentication Id : 0 ; 12361794 (00000000:00bca042)
Session           : Service from 0
User Name         : 77046A79-3B72-43E3-B70D-E0E189B6DC86
Domain            : NT VIRTUAL MACHINE
Logon Server      : (null)
Logon Time        : 22.06.2021 08:26:38
SID               : S-1-5-83-1-3816489593-1138965362-3789618423-3260611593
```

```
msv :
tspkg :
wdigest :
* Username : DESKTOP-HI0G080$  
* Domain   : WORKGROUP
* Password  : (null)
kerberos :
ssp :
```

```
* Username : kaufmann@outlook.de
* Domain   : MicrosoftAccount
* Password : (null)

ssp :
credman :
[00000000]
* Username : nkwritesabout.net
* Domain   : outlook.office365.com
* Password : 00000000000000000000000000000000
[00000001]
* Username : (null)
* Domain   : MicrosoftOffice16 Data:SSPI:nkwritesabout.net
* Password : 00000000000000000000000000000000
[00000002]
* Username : github.auth
* Domain   : vscodevscode.github-authentication/github.auth
* Password : 00000000000000000000000000000000
```

```
00 4b 63 6c 43 46 37 4c 4f 72 4e 37 67 66 6a 44 55 36 4b 6a 51 36 65 6b 64 45 38 30 2f 6a 9a 35 61 78 57 42 4e 58 44 71
42 43 4a 52 44 74 42 39 47 63 34 45 4a 59 5a 69 43 36 4b 4a 2f 53 49 68 39 6e 55 67 6f 4e 6e 53 57 67 71 4a 47 6d 33 58
4e 30 34 34 75 4c 6e 4c 52 42 2f 38 32 43 51 8d 6a 45 41 43 61 2f 34 5a 4e 66 2b 47 54 6b 53 56 4d 63 75 52 79 49 74 4e
56 76 5a 5a 6f 35 42 68 4e 75 75 58 61 39 2f 4f 6a 58 5a 38 62 7a 7a 53 71 50 74 61 62 43 78 74 31 6b 6f 36 35 59 74 6d
55 6d 6a 6a 6a 54 95 6a 6a 66 67 6a 6a 65 6a 65 6a 6a
64 4b 63 6c 43 46 37 4c 4f 72 4e 37 67 66 6a 44 55 36 4b 6a 51 36 65 6b 64 45 38 30 2f 6a 9a 35 61 78 57 42 4e 58 44 71
42 43 4a 52 44 74 42 39 47 63 34 45 4a 59 5a 69 43 36 4b 4a 2f 53 49 68 39 6e 55 67 6f 4e 6e 53 57 67 71 4a 47 6d 33 58
4e 30 34 34 75 4c 6e 4c 52 42 2f 38 32 43 51 8d 6a 45 41 43 61 2f 34 5a 4e 66 2b 47 54 6b 53 56 4d 63 75 52 79 49 74 4e
56 76 5a 5a 6f 35 42 68 4e 75 75 58 61 39 2f 4f 6a 58 5a 38 62 7a 7a 53 71 50 74 61 62 43 78 74 31 6b 6f 36 35 59 74 6d
55 6d 6a 6a 6a 54 95 6a 6a 66 67 6a 6a 65 6a 65 6a 6a
64 4b 63 6c 43 46 37 4c 4f 72 4e 37 67 66 6a 44 55 36 4b 6a 51 36 65 6b 64 45 38 30 2f 6a 9a 35 61 78 57 42 4e 58 44 71
42 43 4a 52 44 74 42 39 47 63 34 45 4a 59 5a 69 43 36 4b 4a 2f 53 49 68 39 6e 55 67 6f 4e 6e 53 57 67 71 4a 47 6d 33 58
4e 30 34 34 75 4c 6e 4c 52 42 2f 38 32 43 51 8d 6a 45 41 43 61 2f 34 5a 4e 66 2b 47 54 6b 53 56 4d 63 75 52 79 49 74 4e
56 76 5a 5a 6f 35 42 68 4e 75 75 58 61 39 2f 4f 6a 58 5a 38 62 7a 7a 53 71 50 74 61 62 43 78 74 31 6b 6f 36 35 59 74 6d
55 6d 6a 6a 6a 54 95 6a 6a 66 67 6a 6a 65 6a 65 6a 6a
72 8d 8a 53 38 2b 72 53 6c 69 47 46 34 42 55 33 48 6a 7a 4f 5a 73 43 47 6a 6c 46 63 37 6d 73 46 6f 4c 78 64 78 6a 37 59
64 43 2b 47 48 5a 4a 37 68 79 2b 78 48 6f 46 48 7a 32 51 76 6d 2f 78 5a 6a 36 75 8d 8a 38 55 49 6c 47 38 52 2b 71 37 47
4d 55 34 44 6f 72 43 7a 69 4a 78 4a 47 31 59 67 49 51 34 51 45 79 61 7a 55 4c 49 67 4f 78 63 68 34 47 4f 53 44 41 6e 35
5a 53 6d 2b 41 4c 51 37 38 6b 37 53 6a 8d 8a 84 5a 40 34 74 6c 45 6a 73 31 77 68 63 35 59 60 2f 68 33 56 32 78 39 5a
```

69 57 37 74 2f 54 6e 2f 64 6d 74 55 73 2f 42 6e 41 3d 3d 0d 4a 00 00

[00000003]

\* Username : wulfland  
\* Domain : GitHub - https://api.github.com/wulfland

\* Password : 090d005700410070000

[00000004]

\* Username : wulfland@hotmail.com  
\* Domain : https://gitlab.com  
\* Password : 090d005700410070000

cloudap :

Cachedir : d470e18c8f14149c4504f151e1041892c5fe9c1a50cc2b0ea1fdaef444010495d

Key GUID : {00000000-5f1c-30a9-0000-000000000000}

PRT : []

DPAPI Key: 540039007500610039006a006d0057004100700034000b00300034005100070009005000310003006100790051006000

5a0001000c00003007B0000005800059005590078eff4292dd33423cd146db9f101ec09d1b69c4faaf (sha1: 1457570c0ea9fc74668051f3e1185ac  
c4411575)

Authentication Id : 0 ; 195058 (00000000:0002f9f2)

Session : Interactive from 1

User Name : DWM-1

Domain : Window Manager

Logon Server : (null)

Logon Time : 21.06.2021 22:24:46

SID : S-1-5-20-0-1

msv :

tspkg :

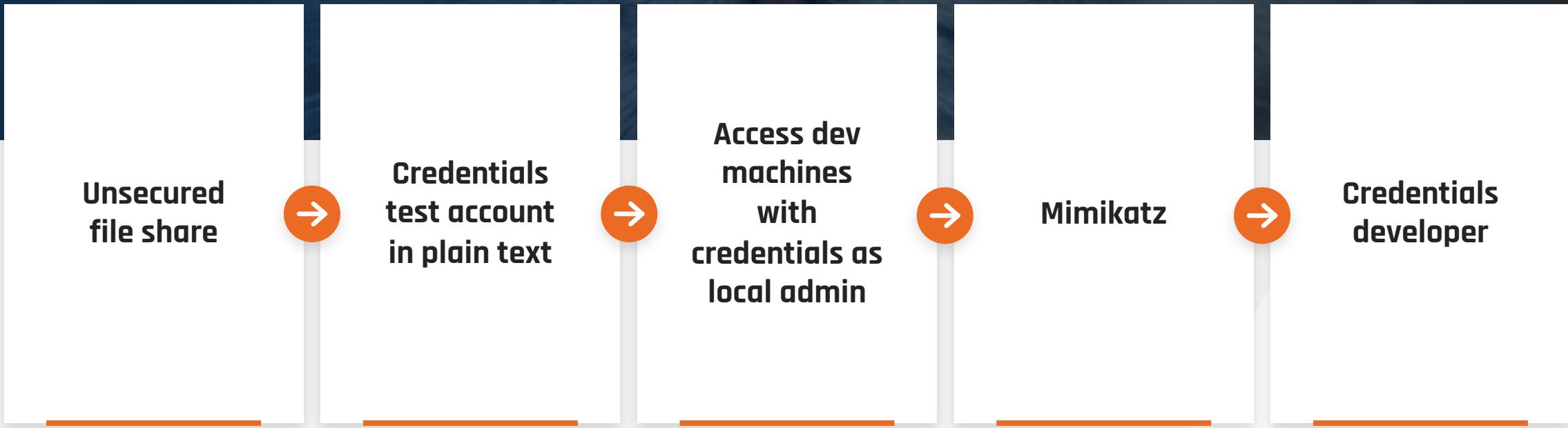
wdigest :

\* Username : DESKTOP-HJ8G000S

\* Domain : WORKGROUP

\* Password : (null)

# Example



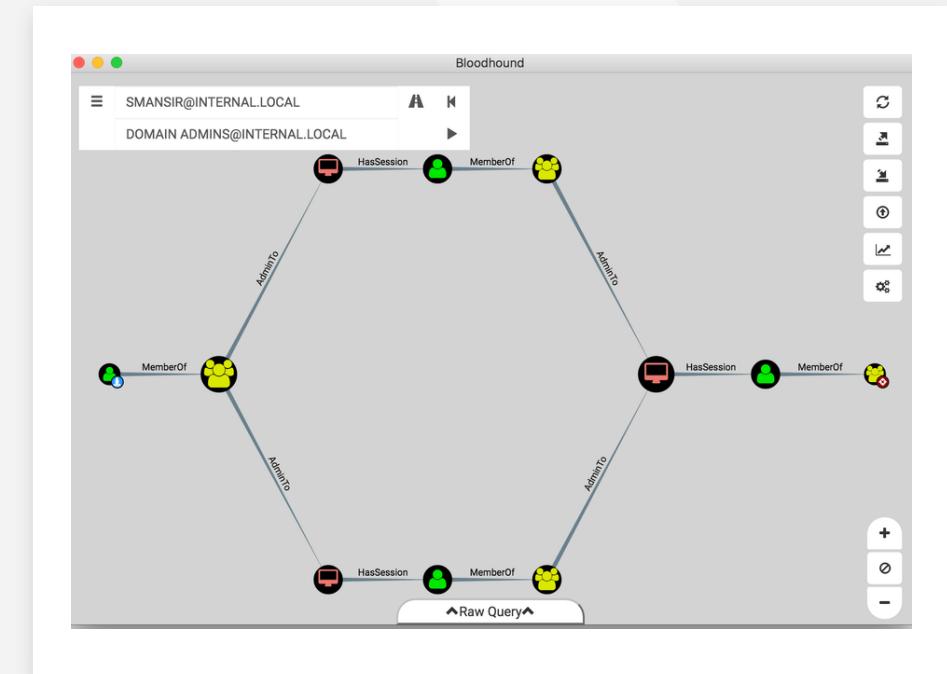
# From Dev to Prod

- ▶ **Bloodhound: <https://github.com/adaptivethreat/Bloodhound>**

- ▶ **grpmgr test01 Administrators /enum**

- ▶ **Other possible entry points:**

- Phishing
- ...
- Responder (<https://github.com/lqandx/Responder>)
- Weak passwords
- Pineapple



---

## Assume breach



“Fundamentally, if somebody wants to get in, they're getting in. Alright, good. Accept that.”

- Michael V. Hayden

former General of the US Air Force and  
former Director of NSA and CIA

# Zero-Trust-Policy



All systems are protected like if they were connected to the internet



Least privilege principle



MFA, SSL, always patched



Separate accounts

# What to do?

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

# Secret Scanning



## Code

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



## Fileshare

- › Bash/PowerShell
- › Dumpster

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

# Attack vector: supply chain

# Supply Chain Attacks

## Libraries / Packages



All libraries used in your applications:

- › Authentication
- › Encryption
- › Backend access
- › ...

## Software



Software and tooling used in the process of building your application:

- › npm ci
- › dotnet build / msbuild
- › Terraform
- › Splunk
- › ...

# Know your dependencies!

- ▶ Naming conflict of npm package with Kick in 2016 (<https://www.kick.com/>)
- 

- ▶ Npm sides with kick
- 

- ▶ Azer Koçulu retracted all packages
    - one of them **left-pad**
- 

- ▶ 11 lines of code - broke the internet
- 

```
1 module.exports = leftpad;
2 function leftpad (str, len, ch) {
3   str = String(str);
4   var i = -1;
5   if (!ch && ch !== 0) ch = ' ';
6   len = len - str.length;
7   while (++i < len) {
8     str = ch + str;
9   }
10  return str;
11 }
```

# Software Composition Analysis (SCA)

- › GitHub (Dependency-Graph/Dependabot)
- › anchore (<https://anchore.com/>)
- › Dependency-Track (<https://dependencytrack.org/>)

### Dependabot alerts

Dismiss all ▾

⚠ 4 Open ✓ 0 Closed

Manifest ▾ Sort ▾

Severity	Dependency	Author	Last updated	Action
critical severity	express-jwt	auth0	3 minutes ago by GitHub	View details
high severity	node-jsonwebtoken	auth0	3 minutes ago by GitHub	View details
moderate severity	sanitize-html	c58	3 minutes ago by GitHub	View details
critical severity	jsonwebtoken	DefinitelyTyped	3 minutes ago by GitHub	View details

### 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	Author	Vulnerability	Version
express-jwt	auth0	Known security vulnerability in 0.1.3	0.1.3
node-jsonwebtoken	auth0	Known security vulnerability in 0.4.0	0.4.0
sanitize-html	c58	Known security vulnerability in ^ 0.6.11	^ 0.6.11
sanitize-html	apostrophecms	Known security vulnerability in 1.4.2	1.4.2
nyc-config-typescript	istanbuljs	1.0.1	1.0.1
types-chai	Seally	4.2.14	4.2.14
chai-as-promised	DefinitelyTyped	7.1.3	7.1.3

# Frameworks



- ▶ **OWASP**  
Software Component Verification Standard
- 
- ▶ v1 since 2020: <https://xpir.it/SCVS>

- ▶ **S**upply chain **L**evels for **S**oftware  
**A**rtifacts
- 
- ▶ Currently in Alpha: <https://slsa.dev/>

# OWASP SCVS

	L1	L2	L3	L4
<b>V1</b> - Inventory				
<b>V2</b> - Software Bill of Materials (SBOM)				
<b>V3</b> - Build Environment				
<b>V4</b> - Package Management				
<b>V5</b> - Component Analysis				
<b>V6</b> - Pedigree and Provenance				



<https://owasp-scsvs.gitbook.io>

# Software Bill of Materials (V2 OWASP SCVS)

Multiple standards for SBoM formats:

## ► Software Package Data Exchange (SPDX)

- › Linux Foundation
- › Focuses on license information
- › ISO/IEC 5962:2021 - fulfills NTIA's minimum elements for a SBoM
- › Syft, Anchore  
(<https://github.com/marketplace/actions/anchore-sbom-action>)

```
- name: Anchore SBOM Action
  uses: anchore/sbom-action@v0.6.0
  with:
    path: .
    image: ${{ env.REGISTRY }}/{{ env.IMAGE_NAME }}
    registry-username: ${{ github.actor }}
    registry-password: ${{ secrets.GITHUB_TOKEN }}
```



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

## ► CycloneDX (CDX)

- › OWASP
- › Focuses on vulnerabilities and security
- › Used in OWASP Dependency Track
- › <https://cyclonedx.org/>



## ► Software Identification Tags (SWID)

- › SWID is an ISO/IEC industry standard (ISO/IEC 19770-2)
- › Focus on inventory in Software Asset Management
- › Snow, System Center, ServiceNow ITOM

# What to do?

- ▶ Know your dependencies
- ▶ Keep your dependencies up to date
- ▶ Ephemeral build environments

# Attack vector: vulnerabilities

# OWASP TOP 10

( <https://owasp.org/www-project-top-ten/> )

01	A01:2021-Broken Access Control	06	A06:2021-Vulnerable and Outdated Components
02	A02:2021-Cryptographic Failures	07	A07:2021-Identification and Authentication Failures
03	A03:2021-Injection	08	A08:2021-Software and Data Integrity Failures
04	A04:2021-Insecure Design	09	A09:2021-Security Logging and Monitoring Failures
05	A05:2021-Security Misconfiguration	10	A10:2021-Server-Side Request Forgery

# A03:2021 - Injection

- › 94% of the applications were tested for some form of injection
- › max incidence rate of 19%, an average incidence rate of 3%, and 274k occurrences.
- › 33 CWEs mapped. For example:
  - › CWE-79: Cross-site Scripting (XSS)
  - › CWE-89: SQL Injection
  - › CWE-73: External Control of File Name or Path



# SQL Injection

- › txtUserId = getRequestString("UserId");  
txtSQL = "SELECT \* FROM Users WHERE UserId = " + txtUserId;
- › 105; DROP TABLE Suppliers

```
11 module.exports = function searchProducts () {
12   return (req, res, next) => {
13     let criteria = req.query.q === 'undefined' ? '' : req.query.q || ''
14     criteria = (criteria.length <= 200) ? criteria : criteria.substring(0, 200)
15     models.sequelize.query(`SELECT * FROM Products WHERE ((name LIKE '%${criteria}%' OR description LIKE '%${criteria}%') AND deletedAt IS NULL) ORDER BY name`)
16     .then(([products]) => {
17       const dataString = JSON.stringify(products)
18       if (utils.notSolved(challenges.unionSqlInjectionChallenge)) { // vuln-code-snippet hide-start
19         let solved = true
20         models.User.findAll().then(data => {
21           const users = utils.queryResultToJson(data)
22           if (users.data?.length) {
23             for (let i = 0; i < users.data.length; i++) {
24               solved = solved && utils.containsOrEscaped(dataString, users.data[i].email) && utils.contains(dataString, users.data[i].password)
25               if (!solved) {
26                 break
27               }
28             }
29             if (solved) {
30               utils.solve(challenges.unionSqlInjectionChallenge)
31             }
32           }
33         })
34       }
35     })
36   }
37 }
```

# XSS (Cross-Site-Scripting)

```
143 // vuln-code-snippet start localXssChallenge xssBonusChallenge
144 filterTable () {
145   let queryParam: string = this.route.snapshot.queryParams.q
146   if (queryParam) {
147     queryParam = queryParam.trim()
148     this.ngZone.runOutsideAngular(() => { // vuln-code-snippet hide-start
149       this.io.socket().emit('verifyLocalXssChallenge', queryParam)
150     }) // vuln-code-snippet hide-end
151     this.dataSource.filter = queryParam.toLowerCase()
152     this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParam)
153     this.gridDataSource.subscribe((result: any) => {
154       if (result.length === 0) {
155         this.emptyState = true
156       } else {
157         this.emptyState = false
158       }
159     })
160   } else {
161     this.dataSource.filter = ''
162     this.searchValue = undefined
163     this.emptyState = false
164   }
165 }
```

```
var Affix = function (element, options) {
  this.options = $.extend({}, Affix.DEFAULTS, options)

  this.$target = $(this.options.target)
  .on('scroll.bs.affix.data-api', $.proxy(this.checkPosition, this))
  .on('click.bs.affix.data-api', $.proxy(this.checkPositionWithEventLoop, this))

  this.$element      = $(element)
  this.affixed      = null
  this.unpin        = null
  this.pinnedOffset = null

  this.checkPosition()
}
```

# Shift left security

# Static Application Security Testing (SAST)



## Whitebox-Testing

- › GitHub Code Analysis
- › SonarQube

- › Semgrep (<https://semgrep.dev/>)
- › Mobile-Security-Framework (MobSF) (<https://github.com/MobSF/Mobile-Security-Framework-MobSF>)

**Client-side cross-site scripting**  
Writing user input directly to the DOM allows for a cross-site scripting vulnerability.

**Open** **Error** **CWE-79** **CWE-116** **security**

Branch: main ▾

Dismiss ▾

frontend/src/app/search-result/search-result.component.ts

```
149     this.io.socket().emit('verifyLocalXssChallenge', queryParam)
150 } // vuln-code-snippet hide-end
151 this.dataSource.filter = queryParam.toLowerCase()
152 this.searchValue = this.sanitizer.bypassSecurityTrustHtml(queryParam) // vuln-code-snippet vuln-line localXssChallenge
```

Cross-site scripting vulnerability due to user-provided value.

CodeQL Show paths

```
153     this.gridDataSource.subscribe((result: any) => {
154       if (result.length === 0) {
155         this.emptyState = true
156       }
157     })
158   }
159 }
```

Tool Rule ID Query  
CodeQL js/xss View source

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.

Show more ▾

First detected in commit 9474e68 3 days ago

Create codeql-analysis.yml

frontend/src/app/search-result/search-result.component.ts#L152 on branch main

Verified 9474e68



Tool Rule ID Query  
CodeQL js/xss View source

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.

**Recommendation**

To guard against cross-site scripting, consider using contextual output encoding/escaping before writing user input to the page, or one of the other solutions that are mentioned in the references.

**Example**

The following example shows part of the page URL being written directly to the document, leaving the website vulnerable to cross-site scripting.

```
function setLanguageOptions() {
  var href = document.location.href,
    defit = href.substring(href.indexOf("default=")+8);
  document.write("<OPTION value=1>" + defit + "</OPTION>");
  document.write("<OPTION value=2>English</OPTION>");
}
```

**References**

- OWASP: DOM based XSS Prevention Cheat Sheet.
- OWASP: XSS (Cross Site Scripting) Prevention Cheat Sheet.
- OWASP DOM Based XSS.
- OWASP Types of Cross-Site Scripting.
- Wikipedia: Cross-site scripting.
- Common Weakness Enumeration: CWE-79.
- Common Weakness Enumeration: CWE-116.

Show less ^

# 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> )

The screenshot shows the OWASP ZAP interface. At the top, there's a navigation bar with tabs like Standard Mode, Sites, History, Search, Alerts, Output, and WebSockets. Below the navigation is a sidebar with sections for Contexts (Default Context, HUB Context) and Sites (listing various URLs). The main pane displays a captured message list with columns for Channel, Timestamp, Opcode, Bytes, and Payload. The payload column shows several alert messages, such as "event.type": "alert.added", indicating a potential security issue. At the bottom, there are buttons for Alerts, History, and Primary Proxy.

This screenshot illustrates the OWASP ZAP Head-Up Display (HUB) feature. It shows a browser window for 'Tailwind Traders' with a red warning icon indicating 'Not Secure'. The HUB overlay provides real-time analysis and control over the site's functionality. Annotations explain specific features: 'OWASP ZAP intercepts the traffic' points to the browser header; 'Head-Up-Display (HUB) to analyze and attack the site using the spider' points to the HUB interface; and 'Items can be customized' points to a section where site elements like 'start smart shopping By uploading a photo' and 'HIGHLY RECOMMENDED!' are shown.

# 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 Scans the web application with the OWASP ZAP Full Scan ⚡ 115 stars
- OWASP ZAP API Scan By zaproxy Scans the web application with the OWASP ZAP API Scan ⚡ 2 stars
- OWASP ZAP Baseline Scan By zaproxy Scans the web application with the OWASP ZAP Baseline Scan ⚡ 188 stars

jobs:

```
owasp:
  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

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

**ZAP Full Scan Report #173**

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

github-actions · bot · commented 21 hours ago · Contributor

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

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

# 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

Run anchore/scan-action@v3.2.0

/usr/bin/chmod +x /home/runner/work/\_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0

/home/runner/work/\_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0 -b

/home/runner/work/\_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0\_grype v0.27.3

[info] checking github for release tag='v0.27.3'

[info] fetching release script for tag='v0.27.3'

anchore/grype info checking GitHub for tag 'v0.27.3'

anchore/grype info found version: 0.27.3 for v0.27.3/linux/amd64

anchore/grype info installed /home/runner/work/\_temp/ca1e5e61-229a-43b7-b70f-c317932ac1c0\_grype/grype

Analyzing: ghcr.io/wulfland/container-demo

Executing: grype -vv -o json --fail-on medium ghcr.io/wulfland/container-demo

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

Add more scanning tools

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

Filters ▾ Q tool:checkov is:open branch:main

Clear current search, filters and sorts

✓ 2 Open  0 Closed

Tool ▾ Rule ▾ Branch ▾ Severity ▾ Sort ▾

Ensure that S3 bucket has a Public Access block ⓘ Error

aws.tf:1 • Detected 15 minutes ago by checkov

main

Ensure that S3 bucket has cross-region replication enabled ⓘ Error

aws.tf:1 • Detected 15 minutes ago by checkov

main

# What to do?

- ▶ SAST and DAST
- ▶ Infrastructure Scanning
- ▶ Shift left security
- ▶ Codespaces
- ▶ Secret scanning

# Security Information & Event Management (SIEM)

► Azure Sentinel

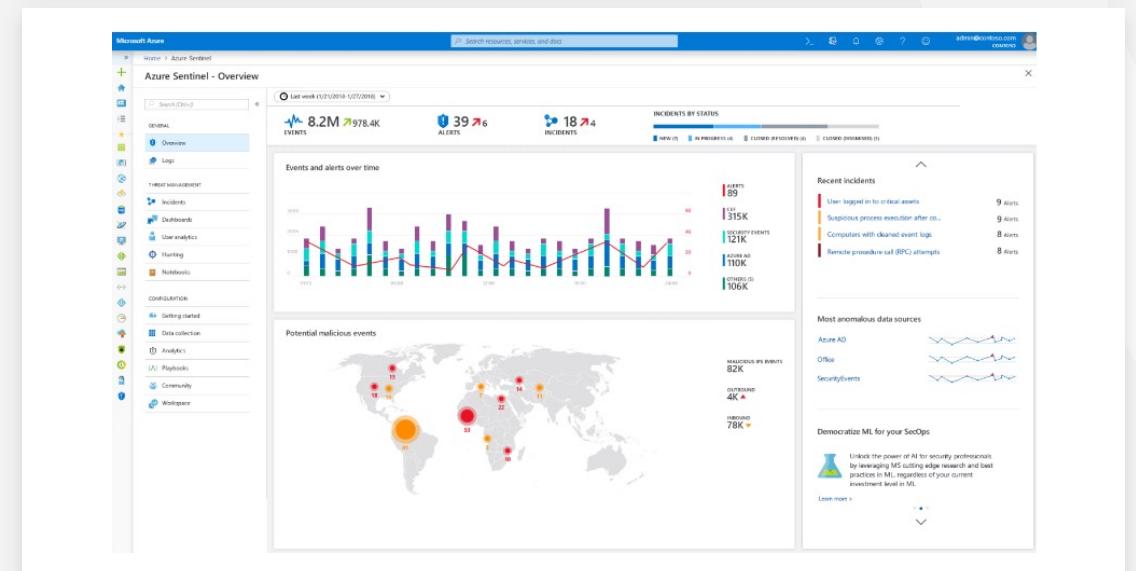
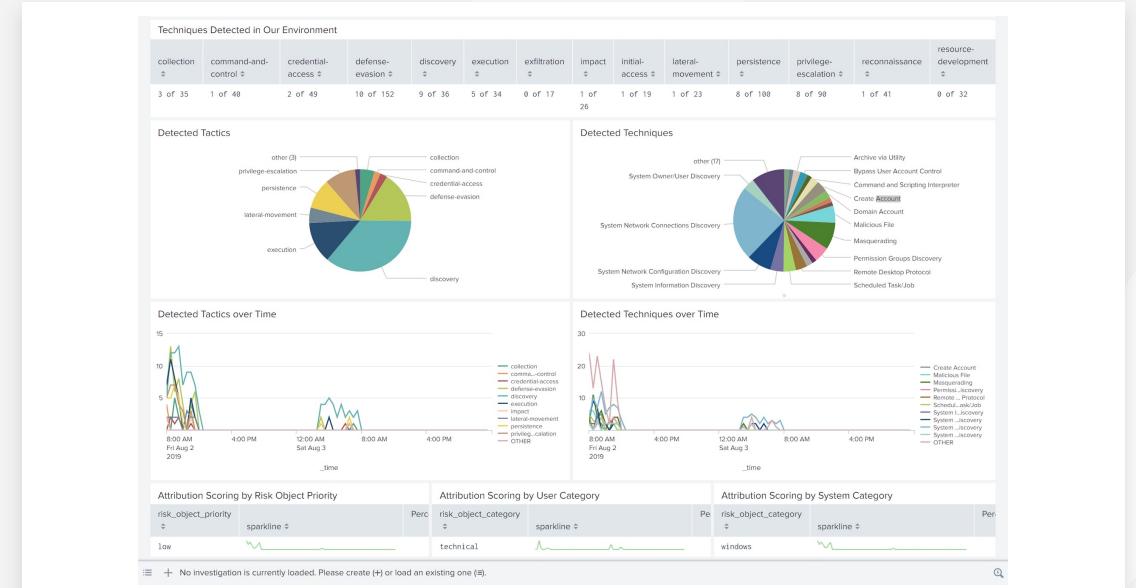
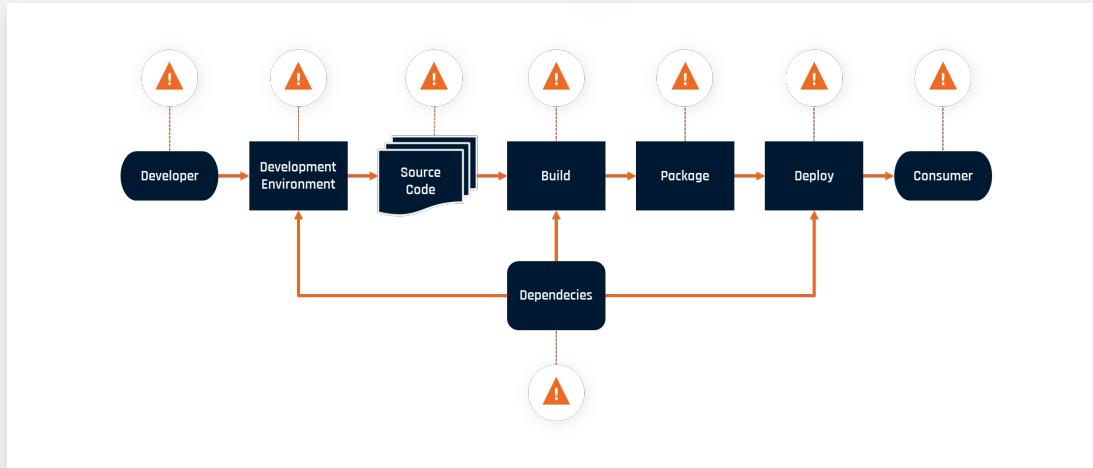
► Splunk

► Central logging

► Multi cloud/hybrid

► Detect anomalies (ML)

► Realtime warnings



# 6 tips to integrate security into your DevOps practices

1

**Build a security-first culture across the business**

2

**Integrate security in the early stages of the development lifecycle**

3

**Monitor and observe continuously with purpose**

4

**Embrace everything-as-code**

5

**Realize compliance with policy automation**

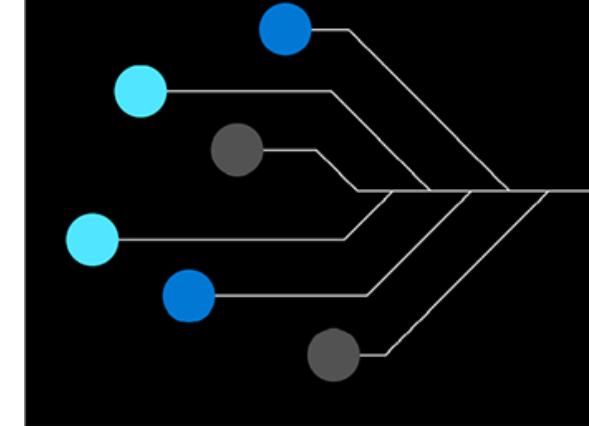
6

**Secure and visualize your software supply chain**

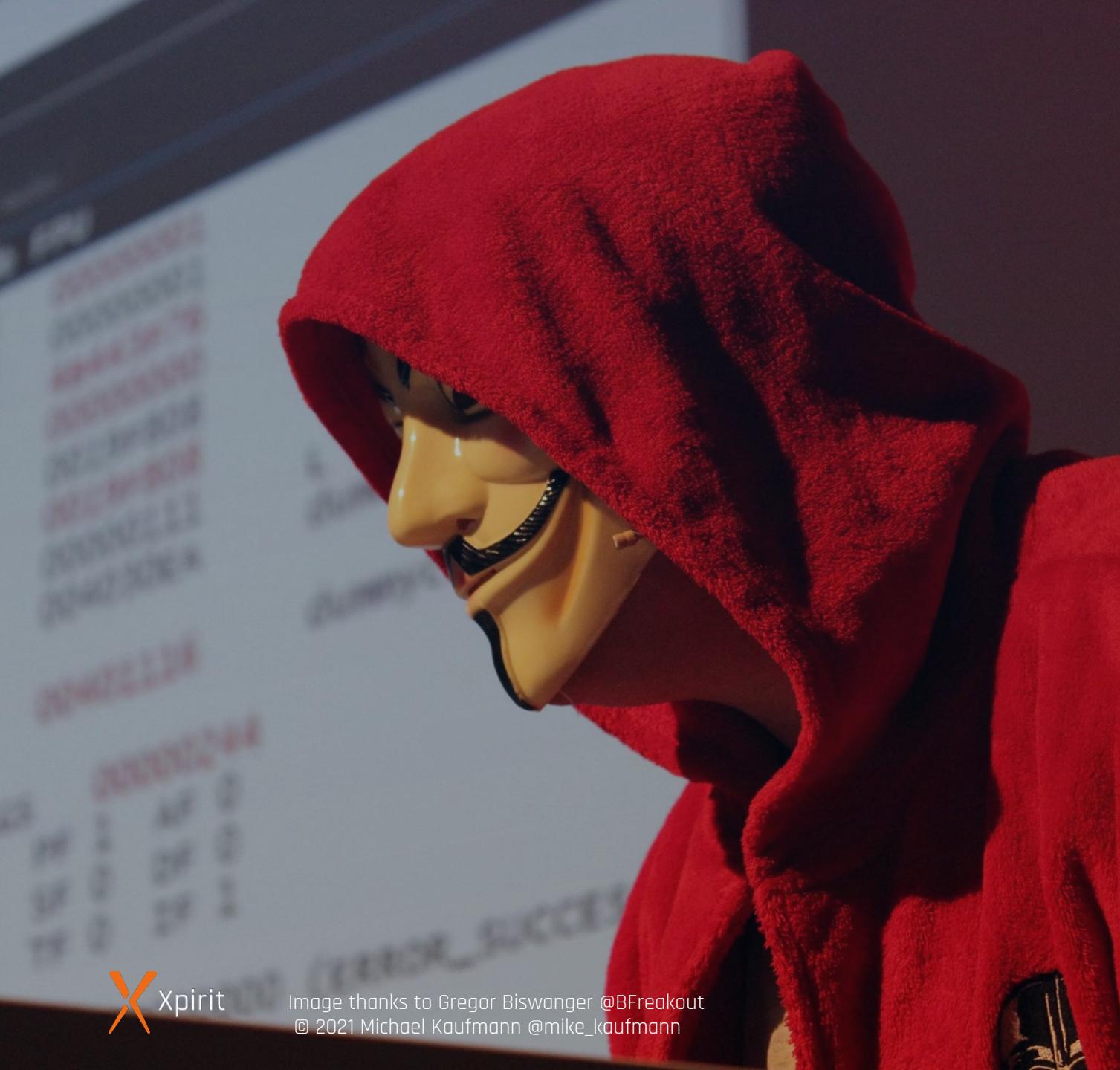
Modern App Development  
and Enterprise DevOps Series



## 6 Tips to Integrate Security into Your DevOps Practices



<https://azure.microsoft.com/en-us/resources/6-tips-to-integrate-security-into-your-devops-practices/>



## Hackers in movies vs



# Thank you

-  Blog : <https://writeabout.net>
-  Twitter : @mike\_kaufmann
-  GitHub : @wulfland
-  LinkedIn : <https://www.linkedin.com/in/mikaufmann/>