L3montree [Lemontree]
Cybersecurity

DevSecOps

So wird die CI-Pipeline zur Basis für sichere Softwareentwicklung

> whoami

- Tim Bastin – Software-Sicherheitsspezialist @ L3montree Cybersecurity
- Studium der Informatik an der Hochschule Bonn-Rhein-Sieg
- Knapp 10 Jahren Entwicklungserfahrung
- Leidenschaft für Software-Qualität
- Beispiel-Projekte:
  - Maintainer OWASP-Inkubationsprojekt DevGuard
  - Maintainer Open-Source Badge-Programm @ ZenDiS
  - Ehemaliger Maintainer IT-Sicherheits-Scanner @ BMI
  - Software Auditierung @ Finance

ANGRIFF VIA OPENSSH

Backdoor in XZ Utils gefährdet das Linux-Ökosystem

Zum Glück wurde die Backdoor entdeckt, bevor sie die breite Masse erreichen konnte. Angreifer hätten damit weltweit Millionen von Linux-Systemen infiltrieren können.

golem.de

«Das ist der verrückteste Angriff»: Ein Programmierer entdeckt per Zufall eine gefährliche Hintertüre im Code – wohl von einem Geheimdienst

nzz.ch

Aktenzeichen XZ ungelöst

In den vergangenen Tagen sind wir mit sehr viel Glück dem wohl größten Fiasko in der Geschichte des Internets gerade so entgangen. Wie konnte das passieren?

heise.de

ZDNet / Alerts

XZ Utils-Vorfall: Open Source als Software Supply Chain-Falle

Ende März wurde in der XZ Utils Bibliothek, ein Kernbestandteil vieler Linux Distributionen, Schadcode entdeckt.

zdnet.de

Es ging um eine halbe Sekunde

Dieser Deutsche (38) hat das Internet gerettet

Andres Freund entdeckt Hacker-Angriff, jetzt feiert ihn die ganze Welt

bild.de

XZ-Utils ist eine Bibliothek für die Komprimierung von Dateien

Sie wird in weitverbreiteten Linux-Distributionen verwendet

Was ist passiert?

Angreifer: Jia Tan
Zeitraum: Oktober 2021 bis März 2024
Angriffsmethode: Social Engineering. Aufbau von Vertrauen durch Beiträge über 2.5 Jahre

Angriffsziel: Implementieren einer Backdoor in der XZ-Utils Bibliothek

→ Langfristige Planung erforderlich
Der Angriff war mit großem Aufwand
Der Angriff zeugt von hoher Expertise

Angriffsziel war die Software-Lieferkette vieler Unternehmen

XZ-Utils ist kein Einzelfall

Hackers Can Abuse Visual Studio Marketplace to Target Developers with Malicious Extensions

Jan 09, 2023 Ravie Lakshmanan Supply Chain / CodeSec

Typosquatting campaign delivers r77 rootkit via npm

ReversingLabs discovered that one "s" was all that separated a legit npm package from a malicious twin that delivered the r77 rootkit — and was downloaded more than 700 times.

Researchers Uncover Obfuscated Malicious Code in PyPI Python Packages

Feb 10, 2023 Ravie Lakshmanan Supply Chain / Software Security

Developers beware: Imposter HTTP libraries lurk on PyPI

ReversingLabs researchers discovered dozens of malicious packages on Python Package Index that mimic popular libraries

CYBERSECURITY ADVISORY

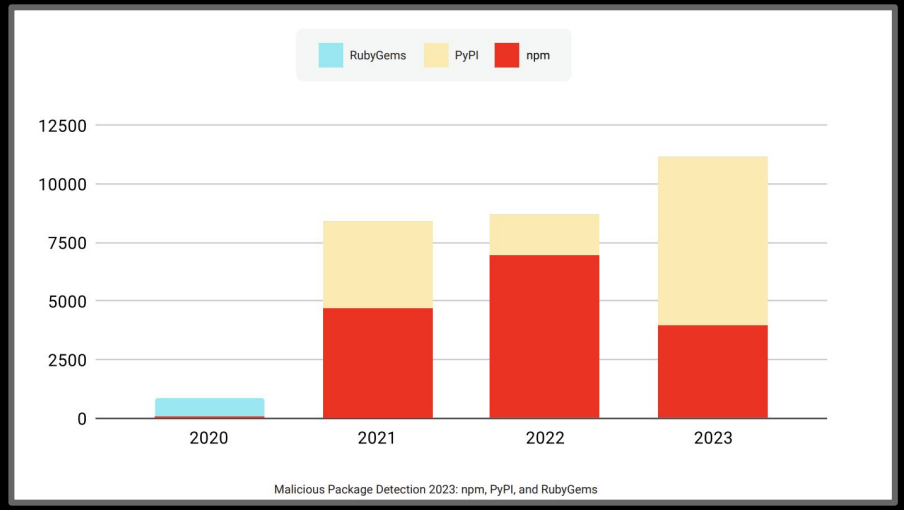
Russian Foreign Intelligence Service (SVR) Exploiting JetBrains TeamCity CVE Globally

Release Date: December 13, 2023 **Alert Code:** AA23-347A

RELATED TOPICS: NATION-STATE CYBER ACTORS, CYBER THREATS AND ADVISORIES, SECURING NETWORKS

Trendwende

Die Software-Supply-Chain rückt ins Visier von Angreifern



Anstieg um 1300 % an Schadcode behafteten Open-Source-Paketen innerhalb von drei Jahren

Der Gesetzgeber reagiert

(3) Die Mitgliedstaaten stellen sicher, dass die Einrichtungen bei der Erwägung geeigneter Maßnahmen nach Absatz 2 Buchstabe d des vorliegenden Artikels die spezifischen Schwachstellen der einzelnen unmittelbaren Anbieter und Diensteanbieter sowie die Gesamtqualität der Produkte und der Cybersicherheitspraxis ihrer Anbieter und Diensteanbieter, einschließlich der Sicherheit ihrer Entwicklungsprozesse, berücksichtigen. Die Mitgliedstaaten stellen ferner sicher, dass die Einrichtungen bei der Erwägung geeigneter Maßnah
die Ergebnisse der gemäß Artikel 22 A
dinierten Risikobewertungen in Bezug auf
ferketten berücksichtigen müssen.

NIS2-Richtlinie

ISO27001, A.1

8.25	Lebenszyklus einer sicheren Entwicklung	**Maßnahme** Regeln für die sichere Entwicklung von Software und Systemen müssen festgelegt und angewendet werden.
8.26	Anforderungen an die Anwendungssicherheit	**Maßnahme** Die Anforderungen an die Informationssicherheit müssen bei der Entwicklung oder Beschaffung von Anwendungen ermittelt, spezifiziert und genehmigt werden.
8.27	Sichere Systemarchitektur und Entwicklungsgrundsätze	**Maßnahme** Grundsätze für die Entwicklung sicherer Systeme müssen festgelegt, dokumentiert, aufrechterhalten und bei allen Aktivitäten der Informationssystemsentwicklung angewendet werden.
8.29	Sicherheitsprüfung bei Entwicklung und Abnahme	**Maßnahme** Sicherheitsprüfverfahren müssen definiert und in den Entwicklungslebenszyklus integriert werden.

Was ist die Software-Supply-Chain?

> Eine Software-Lieferkette besteht aus den Komponenten, Bibliotheken, Tools und Prozessen, die zur Entwicklung, Erstellung und Veröffentlichung eines Software-Artefakts verwendet werden.

[4] übersetzt

Die Softwarelieferkette nach “Supply Chain Levels for Software Artifacts (SLSA)”

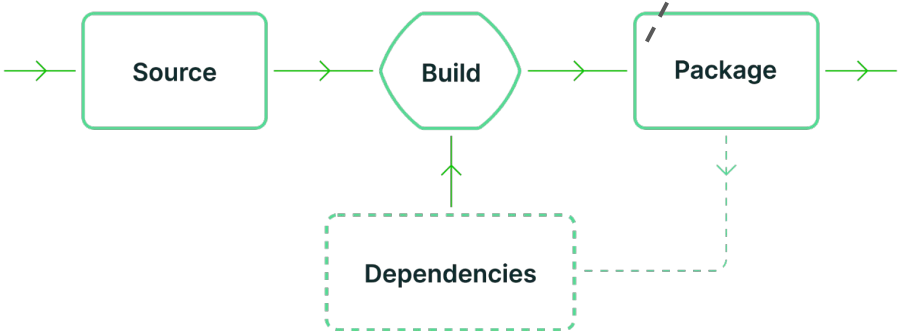
Lieferant

ein Unternehmen, das eine Software, Softwarekomponenten oder Softwarebibliotheken an ein anderes Unternehmen liefert.

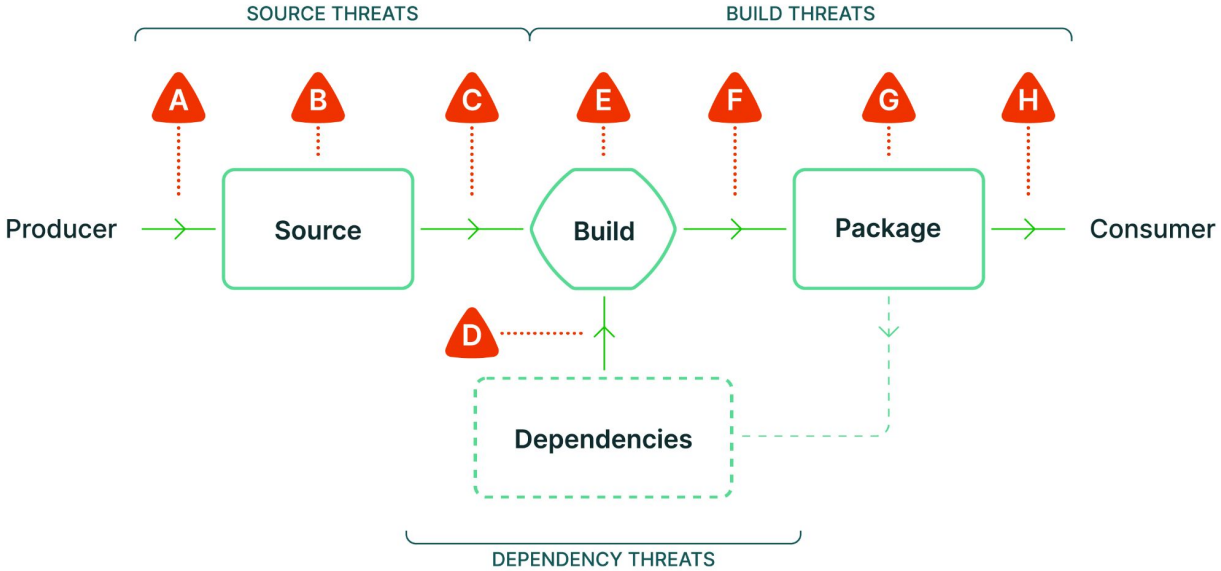
Kunde

das Unternehmen, das die vom Anbieter produzierte Software konsumiert.

Endkonsument oder selbst wieder Software-Lieferant



Threat-Modell der Software-Entwicklung selbst



SOURCE THREATS

A Submit unauthorized change

B Compromise source repo

C Build from modified source

DEPENDENCY THREATS

D Use compromised dependency

BUILD THREATS

E Compromise build process

F Upload modified package

G Compromise package registry

H Use compromised package

Software Supply Chain Attack Definition

„Ein Angriff auf die Lieferkette ist eine Art von Cyberangriff, bei dem nicht die Organisationen direkt angegriffen werden, sondern die **dazwischenliegenden Parteien**, wie z. B. Anbieter und deren Softwarecode.“

[1] übersetzt

„[...] Angreifer manipulieren das Endprodukt eines bestimmten Anbieters so, [...], dass es **von den Endnutzern über vertrauenswürdige Vertriebskanäle**, z. B. Download- oder Update-Sites, bezogen werden kann.“

[2] übersetzt

DevSecOps

Sicher sichere Software entwickeln

“

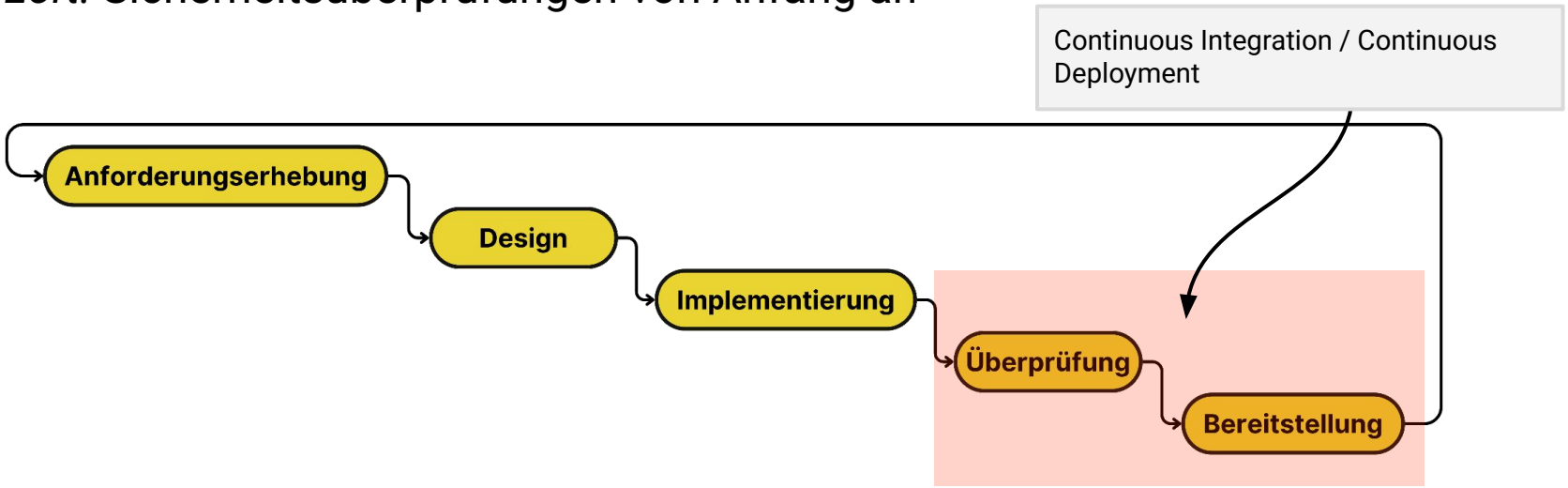
Everyone is responsible for security

Shannon Lietz (Founder DevSecOps Foundation)

DevOps → DevSecOps

Weiterentwicklung von DevOps – Sicherheit wird in den gesamten Softwareentwicklungs-Lebenszyklus integriert

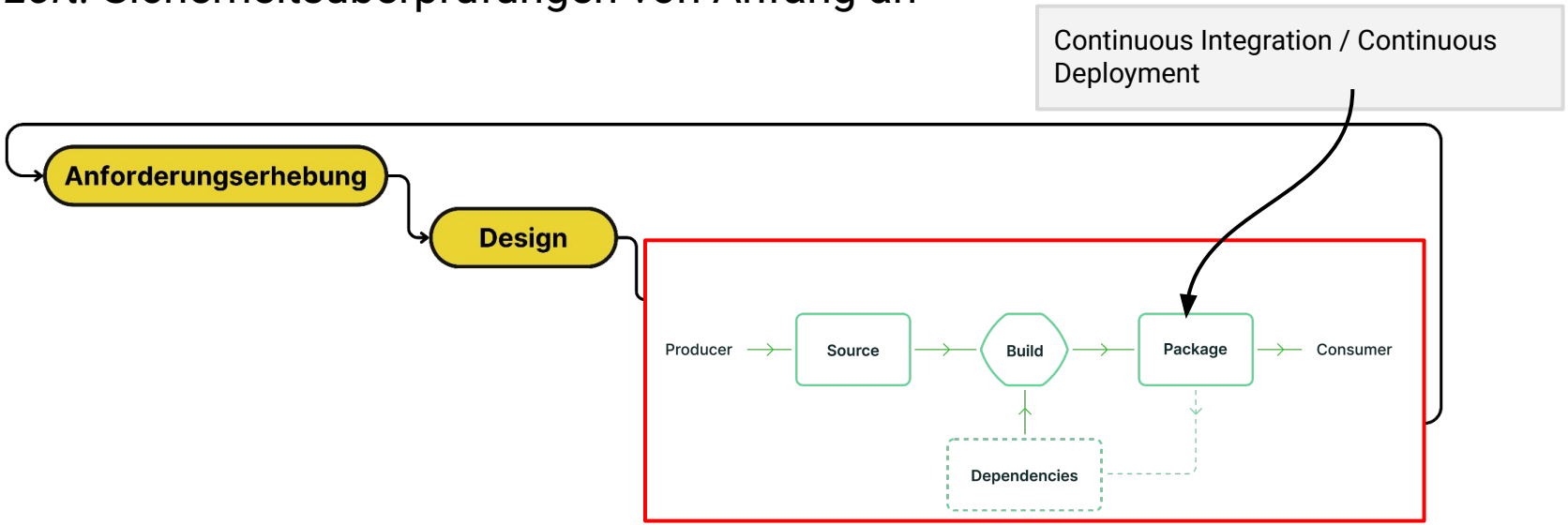
Shift-Left: Sicherheitsüberprüfungen von Anfang an



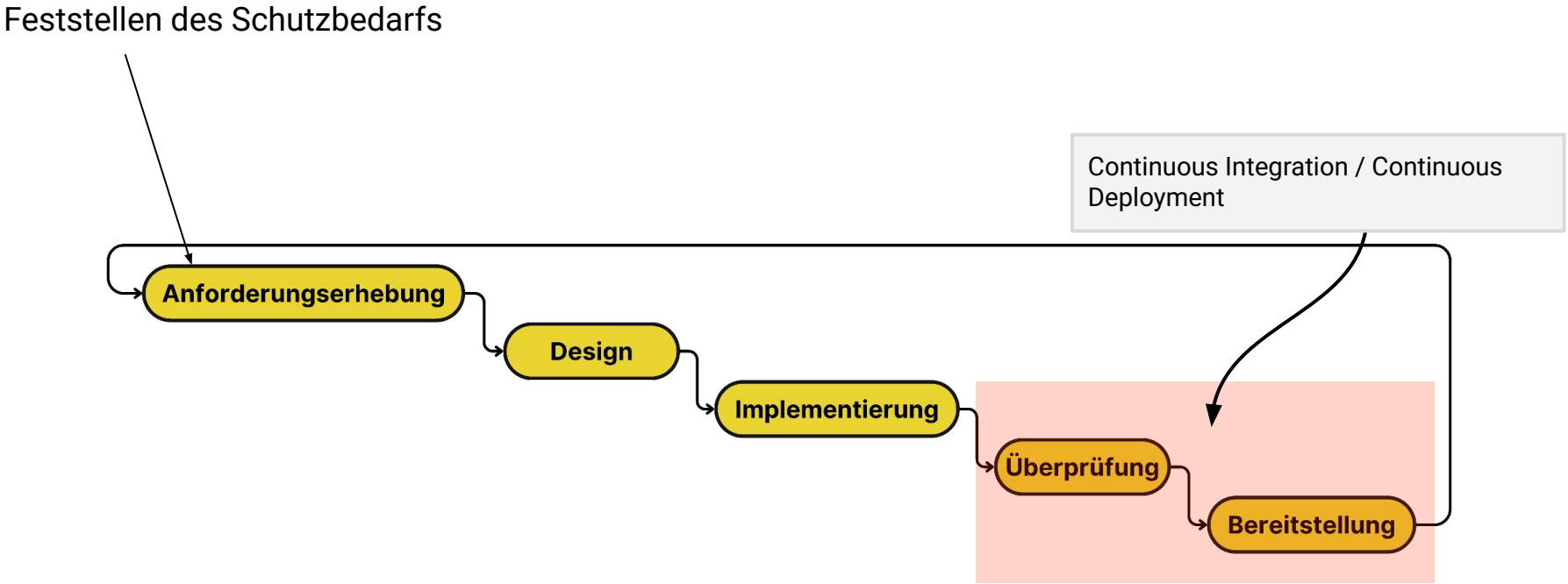
DevOps → DevSecOps

Weiterentwicklung von DevOps – Sicherheit wird in den gesamten Softwareentwicklungs-Lebenszyklus integriert

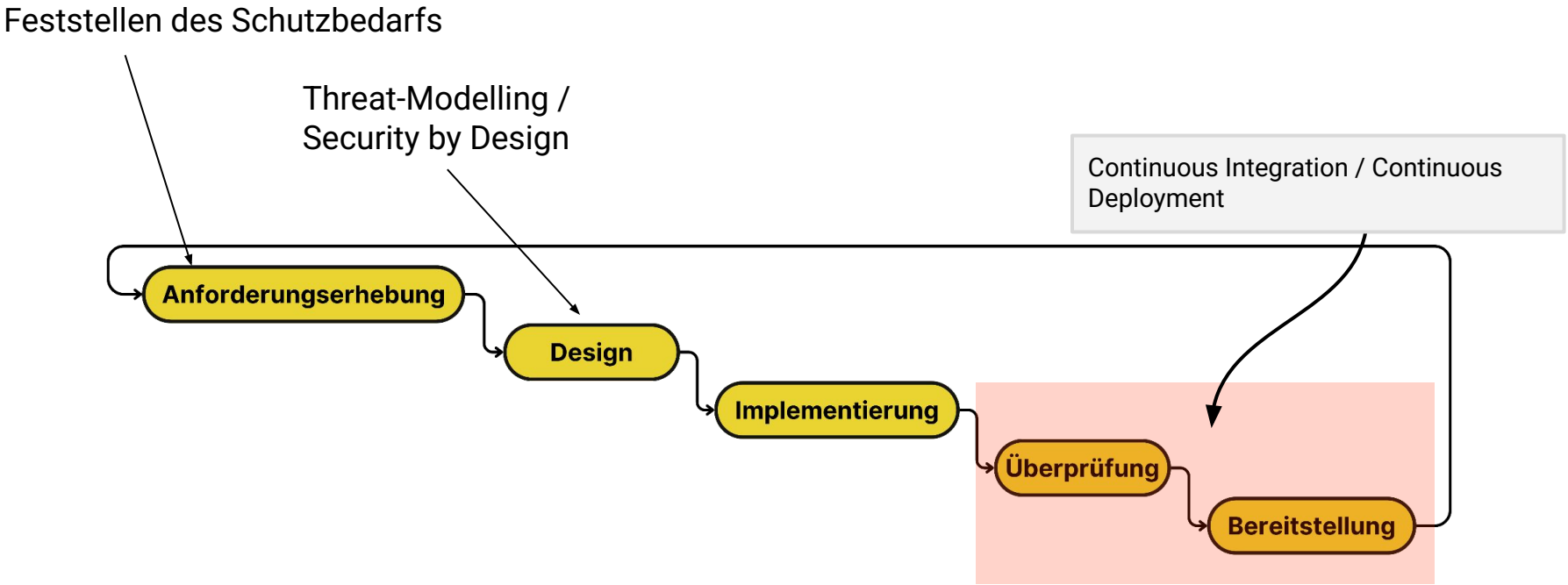
Shift-Left: Sicherheitsüberprüfungen von Anfang an



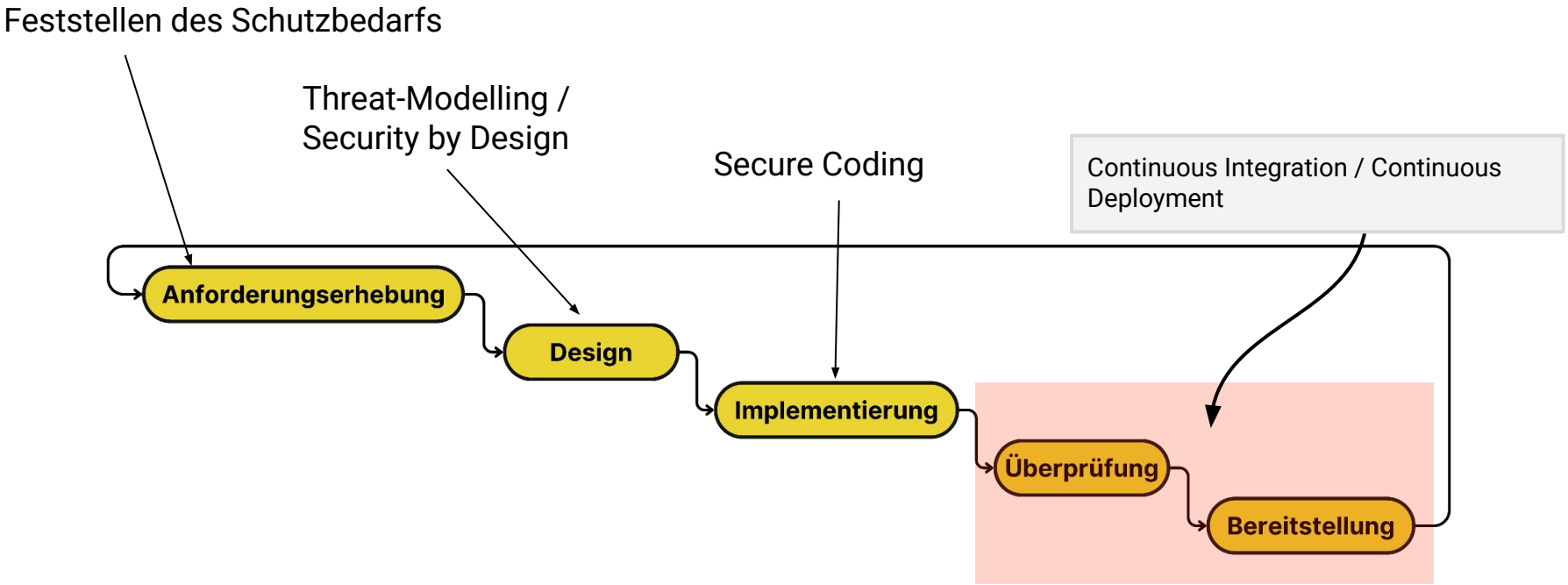
DevOps → DevSecOps



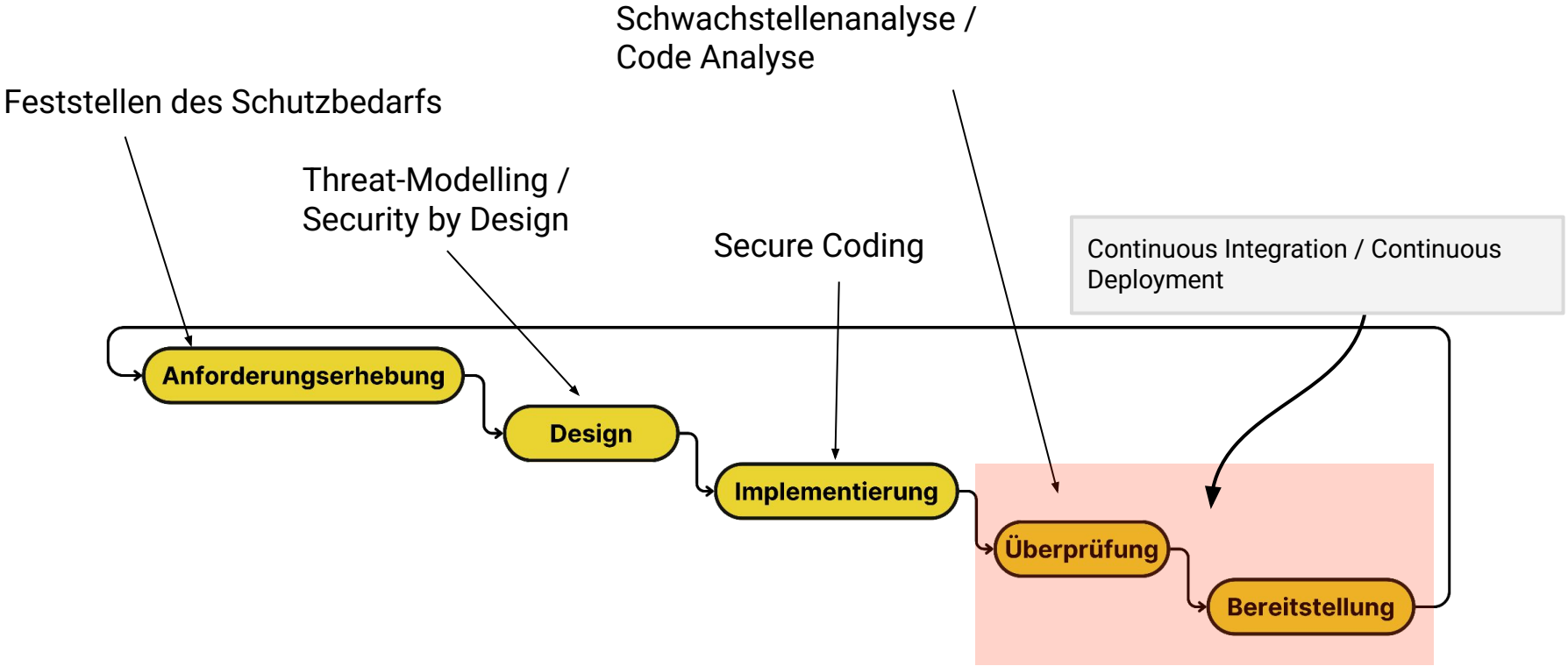
DevOps → DevSecOps



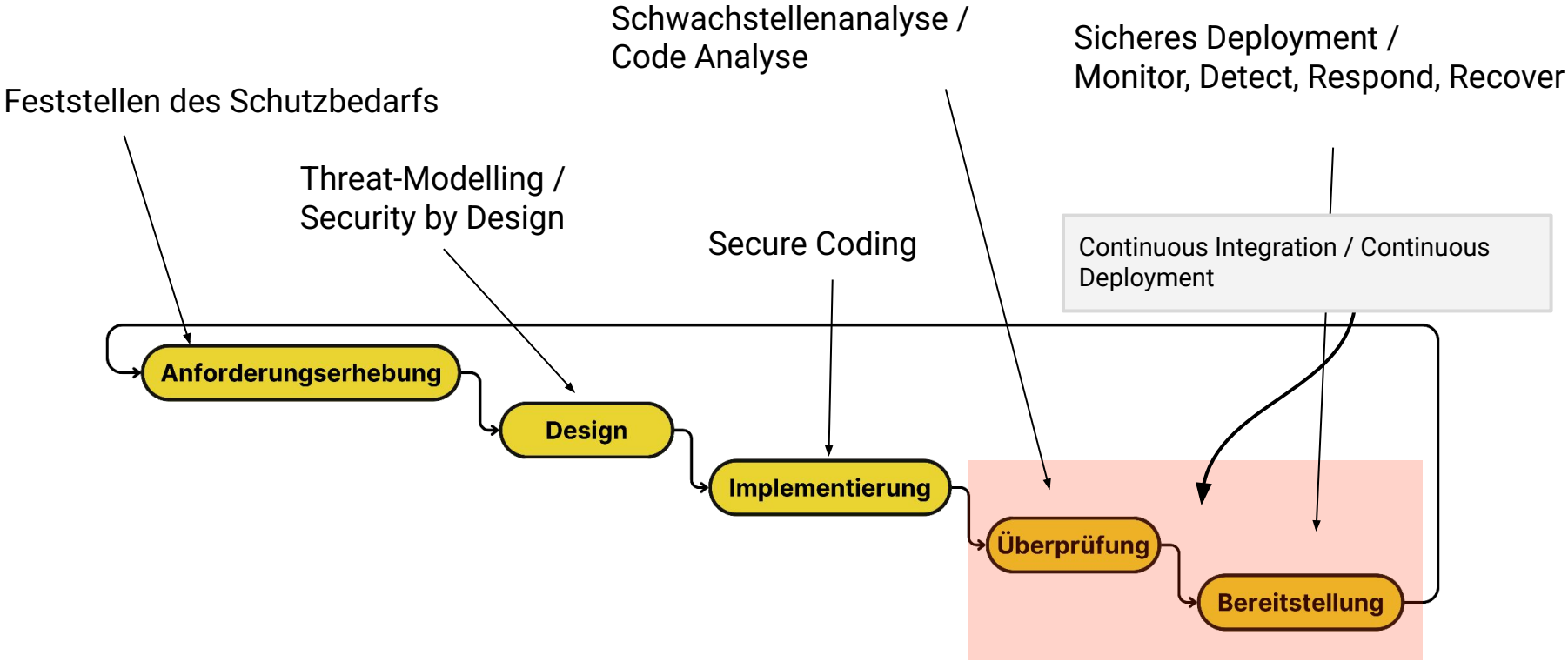
DevOps → DevSecOps



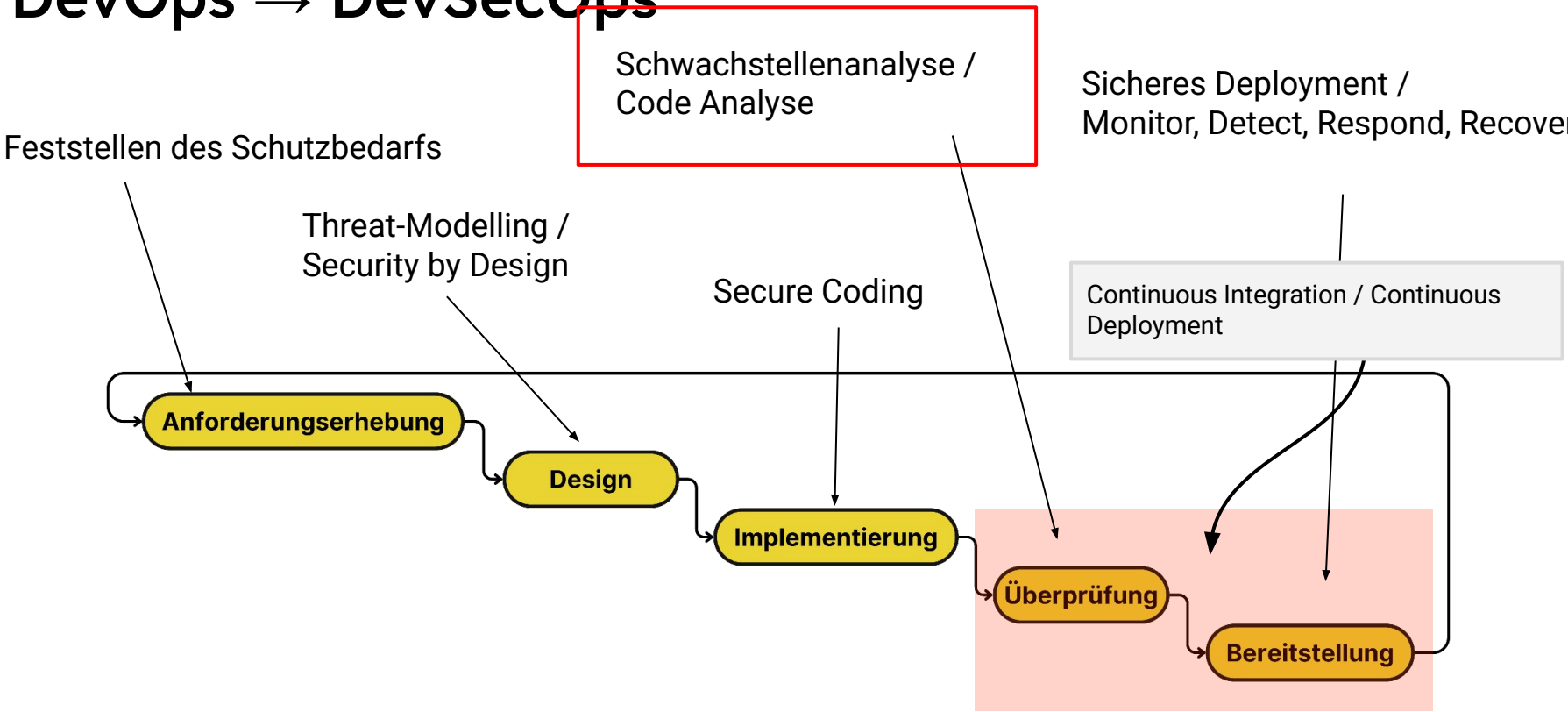
DevOps → DevSecOps



DevOps → DevSecOps

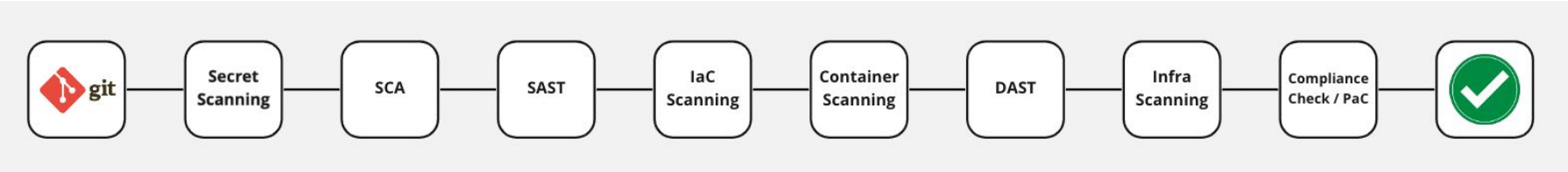


DevOps → DevSecOps



Schwachstellenanalyse in CI-Pipelines

OWASP-DevSecOps Pipeline



L3montree [Lemontree]
Cybersecurity

gitLeaks

Secret Scanning

Entdecken von unabsichtlich “hard-kodierten” Secrets im Code und der Git-Historie

```go
package main

import (
    "database/sql"
    "fmt"
    "net/http"

    _ "github.com/go-sql-driver/mysql"
)

func main() {
    password := "MYS3cureP4ssw0rd"
    db, err := sql.Open("mysql", fmt.Sprintf("user:%s@/dbname", password))
    if err != nil {
        panic(err)
    }
    defer db.Close()

    http.HandleFunc("/users", func(w http.ResponseWriter, r *http.Request) {
        // Vulnerable to SQL Injection
        username := r.URL.Query().Get("username")

        var userID string
        query := fmt.Sprintf("SELECT id FROM users WHERE username = '%s'", username)
        err := db.QueryRow(query).Scan(&userID)

        if err != nil {
            http.Error(w, "User not found", http.StatusNotFound)
            return
        }

        fmt.Fprintf(w, "User ID: %s", userID)
    })

    http.ListenAndServe(":8080", nil)
}
```

L3montree [Lemontree]
Cybersecurity

trivy

SCA Scanning

Erkennen von bekannten Sicherheitslücken in Softwareabhängigkeiten

```go
package main

import (
    "database/sql"
    "fmt"
    "net/http"

    _ "github.com/go-sql-driver/mysql"
)

func main() {
    password := "MYS3cureP4ssw0rd"
    db, err := sql.Open("mysql", fmt.Sprintf("user:%s@/dbname", password))
    if err != nil {
        panic(err)
    }
    defer db.Close()

    http.HandleFunc("/users", func(w http.ResponseWriter, r *http.Request) {
        // Vulnerable to SQL Injection
        username := r.URL.Query().Get("username")

        var userID string
        query := fmt.Sprintf("SELECT id FROM users WHERE username = '%s'", username)
        err := db.QueryRow(query).Scan(&userID)

        if err != nil {
            http.Error(w, "User not found", http.StatusNotFound)
            return
        }

        fmt.Fprintf(w, "User ID: %s", userID)
    })

    http.ListenAndServe(":8080", nil)
}
```

L3montree [Lemontree]
Cybersecurity

Semgrep

SAST Scanning

Erkennen von “Bad-Practices” in selbst erstelltem Code

```go
package main

import (
    "database/sql"
    "fmt"
    "net/http"

    _ "github.com/go-sql-driver/mysql"
)

func main() {
    password := "MYS3cureP4ssw0rd"
    db, err := sql.Open("mysql", fmt.Sprintf("user:%s@/dbname", password))
    if err != nil {
        panic(err)
    }
    defer db.Close()

    http.HandleFunc("/users", func(w http.ResponseWriter, r *http.Request) {
        // Vulnerable to SQL Injection
        username := r.URL.Query().Get("username")

        var userID string
        query := fmt.Sprintf("SELECT id FROM users WHERE username = '%s'", username)
        err := db.QueryRow(query).Scan(&userID)

        if err != nil {
            http.Error(w, "User not found", http.StatusNotFound)
            return
        }

        fmt.Fprintf(w, "User ID: %s", userID)
    })

    http.ListenAndServe(":8080", nil)
}
```

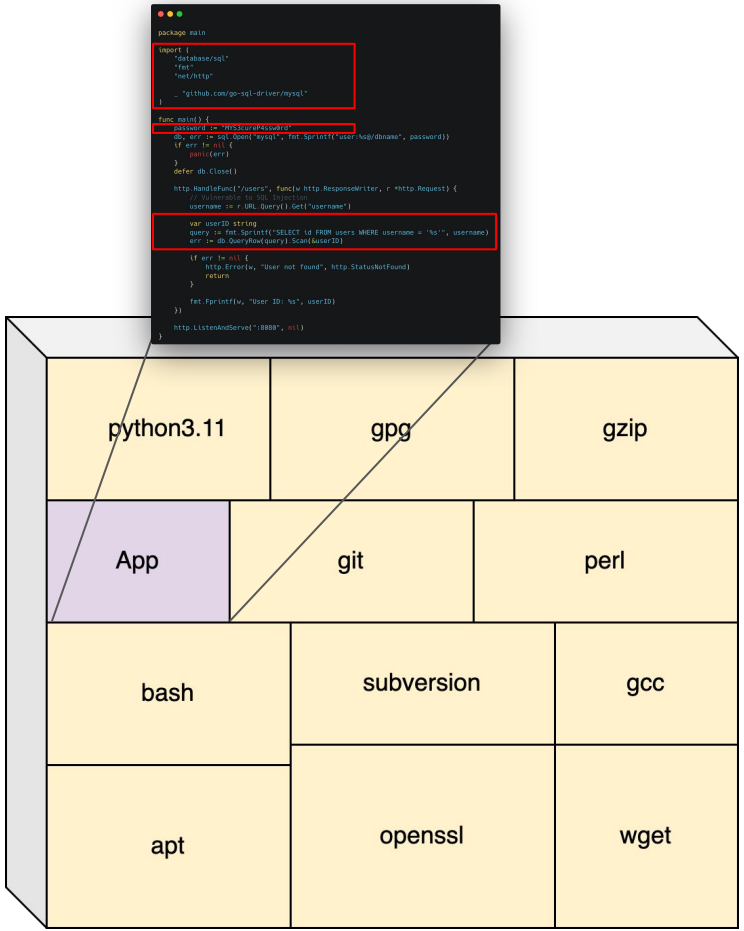
L3montree [Lemontree]
Cybersecurity

library/golang:latest enthält 206 installierte Pakete.

library/openjdk:latest enthält 111 installierte Pakete.

Container Scanning

Erkennen von bekannten Sicherheitslücken in Containern



trivy

L3montree [Lemontree]
Cybersecurity

checkov
by bridgecrew

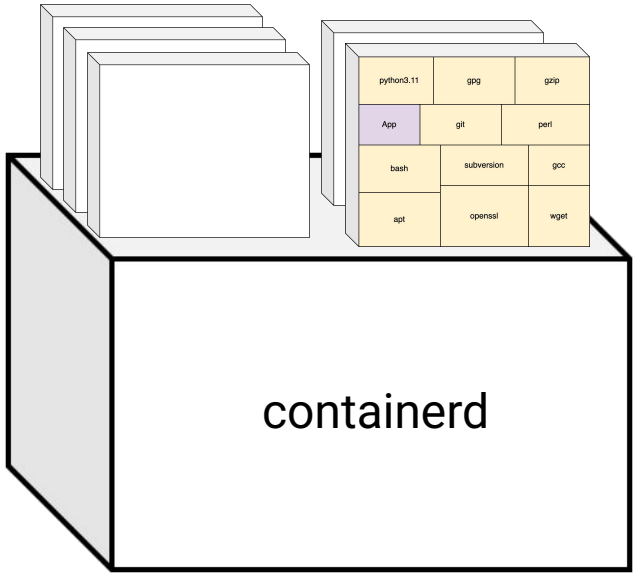
IaC Scanning

Erkennen von unsicheren Konfigurationen der Container-Runtime

eingeschränkte Rechte während der Laufzeit

keine Limitierung der maximalen Ressourcen-Nutzung

Schreib-Rechte für das Host-Filesystem



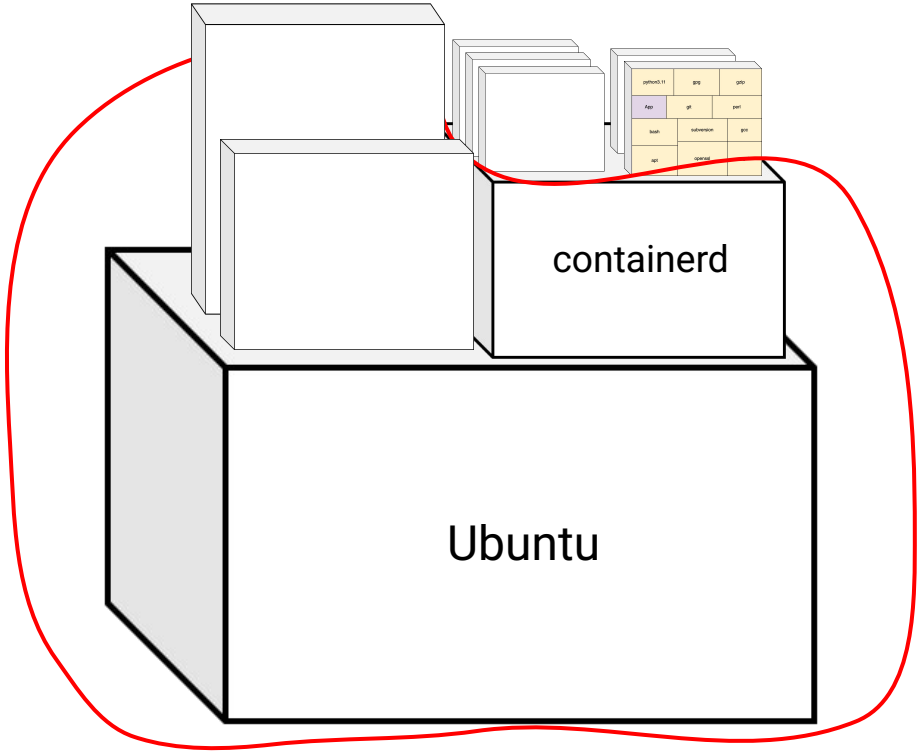
L3montree [Lemontree]
Cybersecurity

Infra Scanning

Scannen der Host-Systeme

★ Nur bedingt in CI-Pipelines umsetzbar

trivy



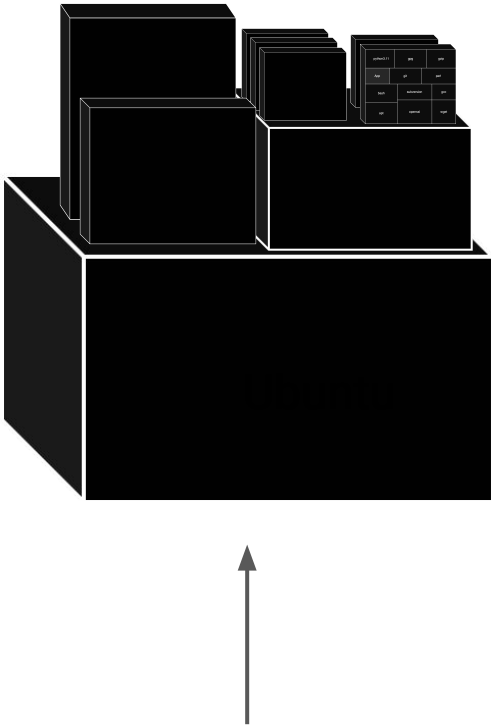
L3montree [Lemontree]
Cybersecurity

DAST Scanning

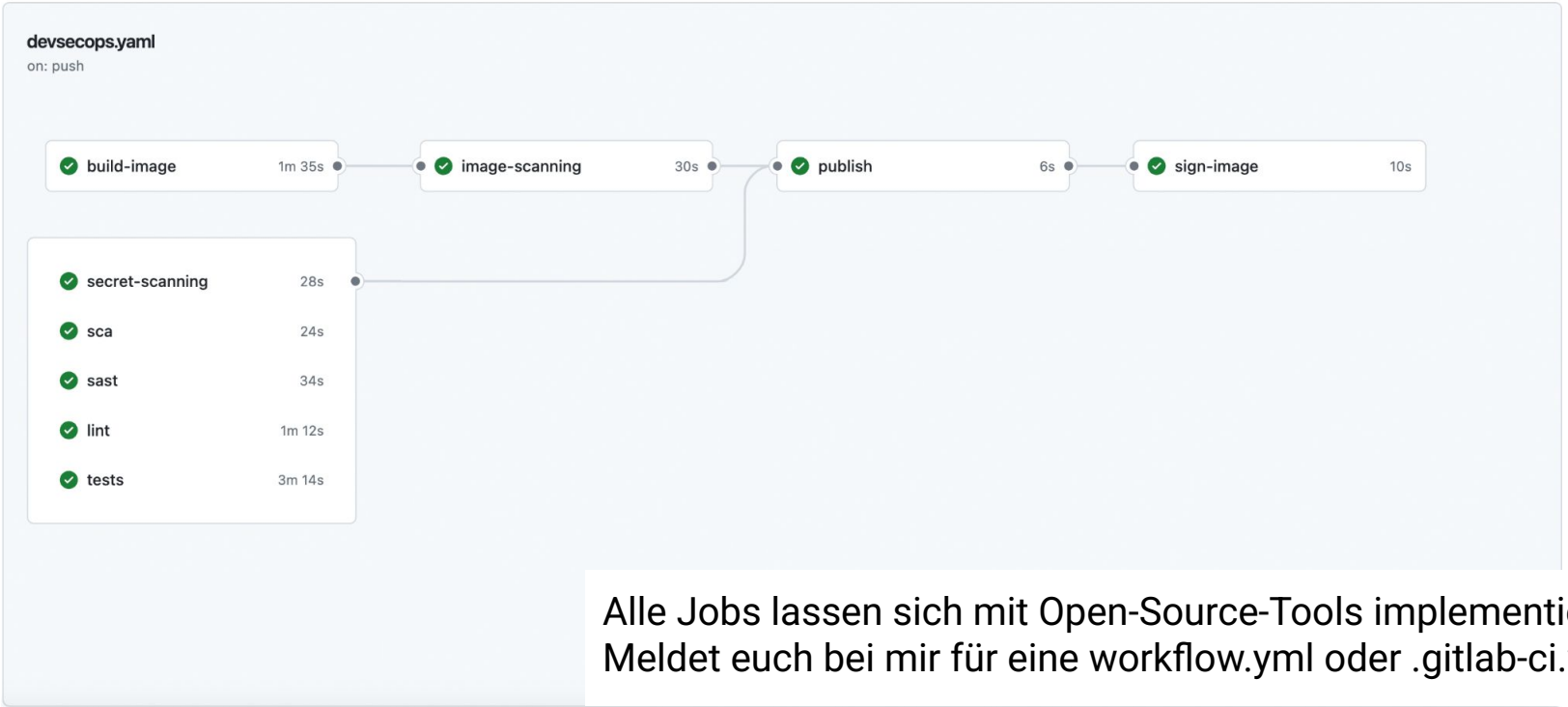
Black-Box Testing der Anwendung von Außen

★ Nur bedingt in CI-Pipelines umsetzbar

ZAP



GitHub Actions DevSecOps Pipeline



Alle Jobs lassen sich mit Open-Source-Tools implementieren.
Meldet euch bei mir für eine workflow.yml oder .gitlab-ci.yml.

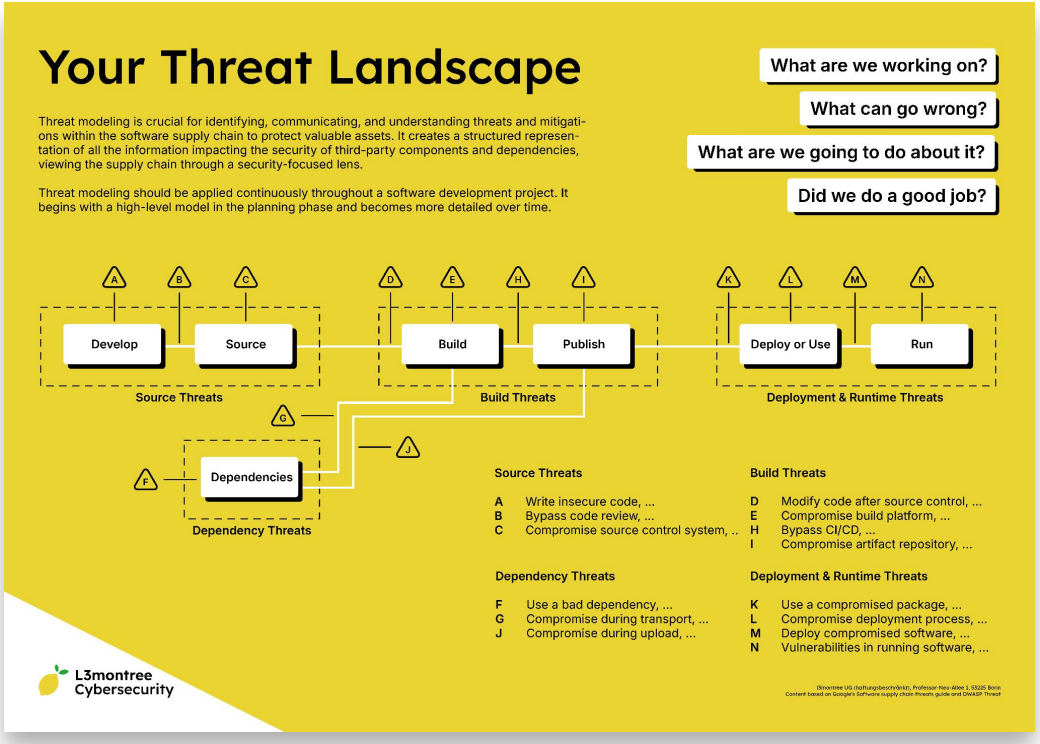
DevSecOps


Static Code Analysis
Threat Model
Code Signing Validation
code
plan
deploy
Dev
release
Ops
build
operate
Vulnerability Scan
Monitor
Detect
Respond
Recover
test
monitor
Penetration Test
Compliance Validation

Give-Away: Threat-Modell der Softwareentwicklung

Auch die Softwareentwicklung selbst muss abgesichert werden

Mehr zum Thema **Software-Supply-Chain-Security** in unseren Youtube-Videos



Zusammenfassung

- DevSecOps ermöglicht es sicher sichere Software zu programmieren
  - Der Software-Supply-Chain wird abgesichert
  - zusätzlich wird die produzierte Software signiert und nach Schwachstellen untersucht
- Insbesondere durch Open-Source-Tools lässt sich die Sicherheit der produzierten Software stark erhöhen
- Aufgrund von Software-Supply-Chain Angriffen ist es **nicht mehr ausreichend, nur** das Endprodukt einer Software zu verifizieren. Es müssen die Softwareentwicklungsmethoden verifiziert werden. Genau hier setzen die Regularien a.n

L3montree [Lemontree]
Cybersecurity

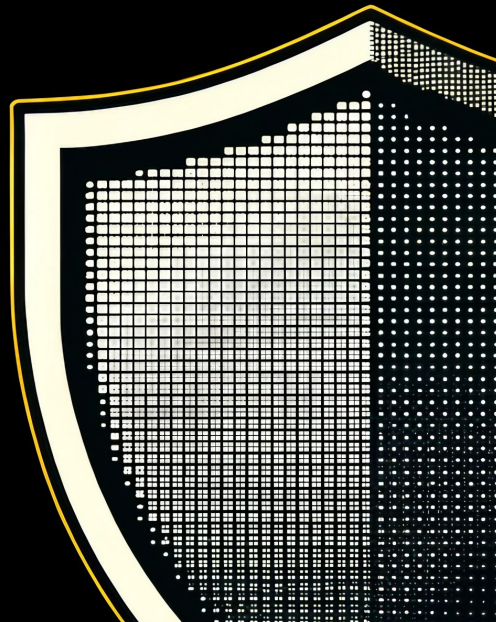
Securing your Software & Cloud-Native Environment

Email: info@l3montree.com

Telefon: +49 (0)228 92998568
+49 177 7438521

Adresse: Markt 3
53111 Bonn

LinkedIn L3montree



Quellen

[1] GeeksforGeeks. (2023). *Evolution of Software Development | History, Phases and Future Trends*. [online] Available at: https://www.geeksforgeeks.org/evolution-of-software-development-history-phases-and-future-trends/.

[2] Intelligence, G.T. (2020). Cybersecurity: Timeline. [online] Verdict. Available at: https://www.verdict.co.uk/cybersecurity-timeline

[3] Bundesamt für Sicherheit in der Informationstechnik. (n.d.). Die Lage der IT-Sicherheit in Deutschland. [online] Available at: https://www.bsi.bund.de/DE/Service-Navi/Publikationen/Lagebericht/lagebericht_node.html.

[4] "For Good Measure Counting Broken Links: A Quant's View of Software Supply Chain Security" (PDF). USENIX ;login. Retrieved 2022-07-04.

[5] Wintergerst, R. and Berlin, B.-P. (2023). Wirtschaftsschutz 2023. [online] Available at: https://www.bitkom.org/sites/main/files/2023-09/Bitkom-Charts-Wirtschaftsschutz-Cybercrime.pdf.

[6] owasp.org. (n.d.). OWASP DevSecOps Guideline | OWASP Foundation. [online] Available at: https://owasp.org/www-project-devsecops-guideline/.

[7] Supply-chain levels for software artifacts. (n.d.-b). SLSA. https://slsa.dev/

[8] Supply chain compromise, Technique T1195 - Enterprise | MITRE ATT&CK®. (n.d.). https://attack.mitre.org/techniques/T1195/

[9] Bedrohungen der Softwarelieferkette. (n.d.). Google Cloud. https://cloud.google.com/software-supply-chain-security/docs/attack-vectors

[10] Sonatype Inc. (n.d.). 2020 Software Supply Chain Report | Download. https://www.sonatype.com/resources/white-paper-state-of-the-software-supply-chain-2020

[11] www.threatmodelingmanifesto.org. (n.d.). Threat Modeling Manifesto. [online] Available at: https://www.threatmodelingmanifesto.org/.

[12] ReversingLabs: The state of Software Supply Chain Security 2024