开源社
kaiyuanshe

云原生应用程序安全交付指南

小马哥 极狐GitLab DevSecOps 技术布道师/LFAPAC 开源布道师

2023 第八届中国开源年会 开源：川流不息、山海相映

目录

CONTENTS

173

218K

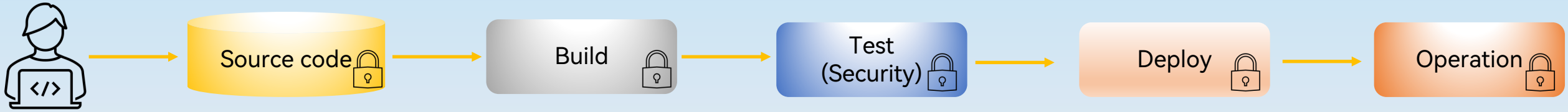
14.7M

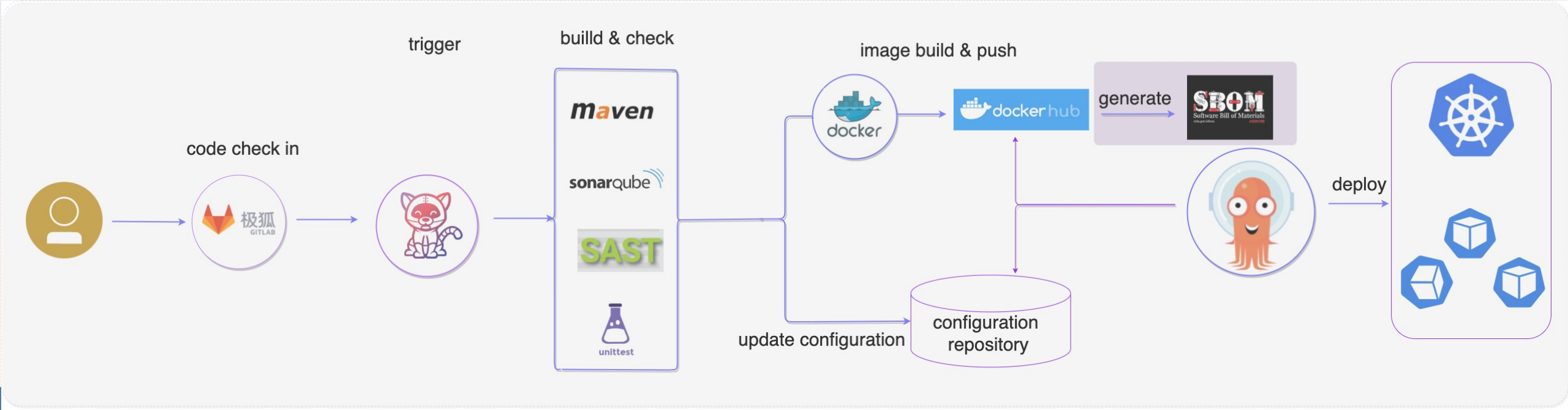
190

1260

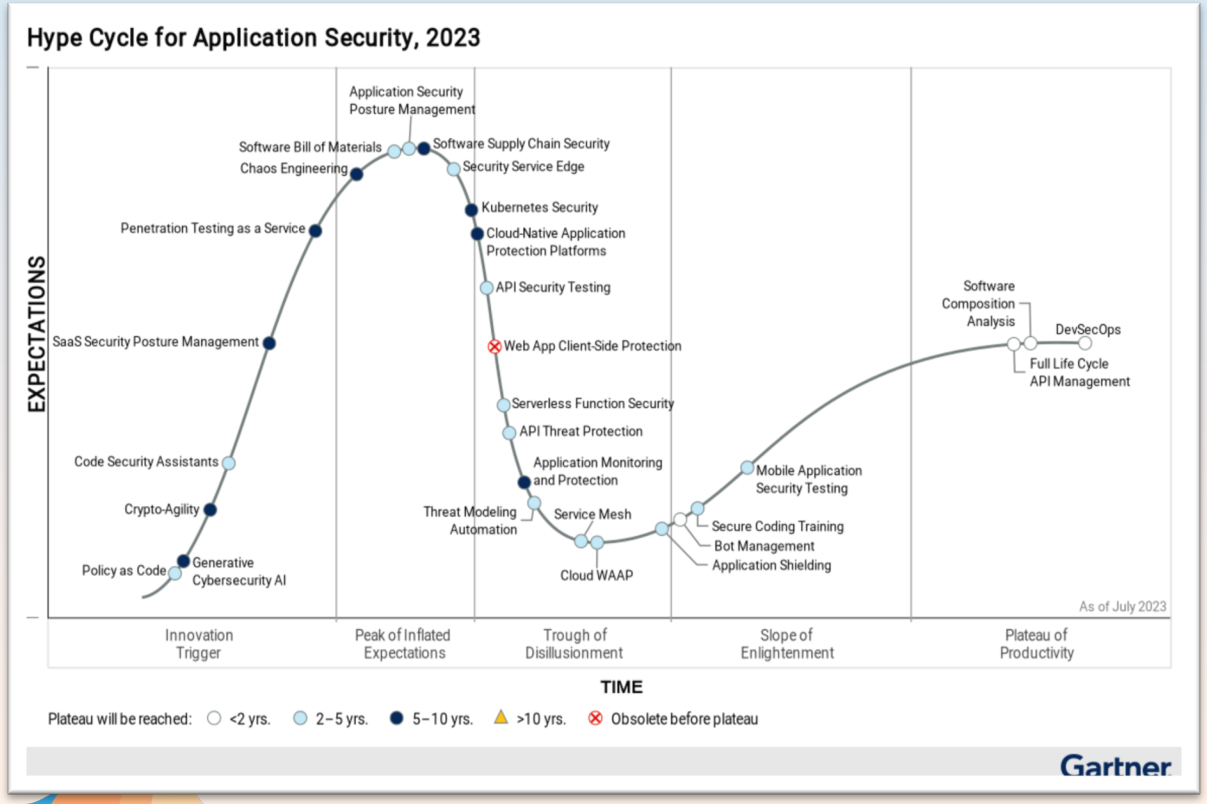
4.07M

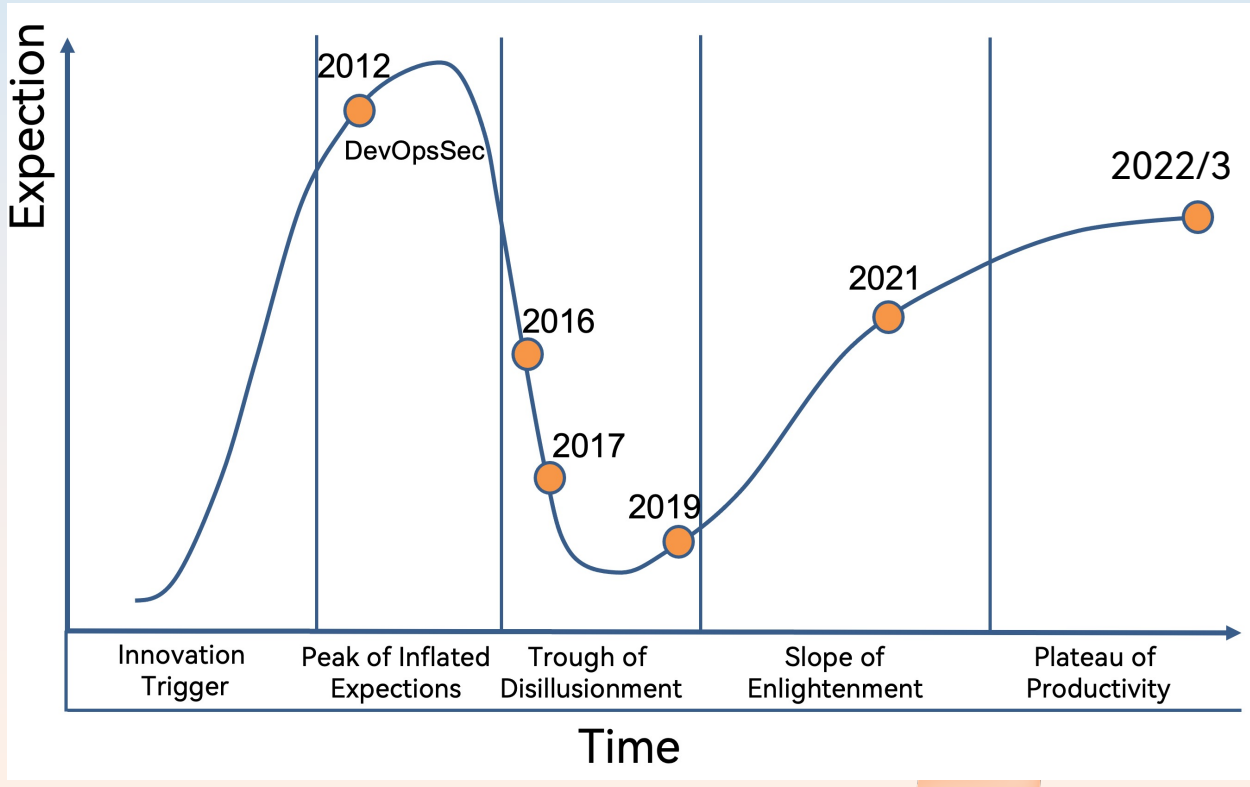
开源社 kaiyuanshe





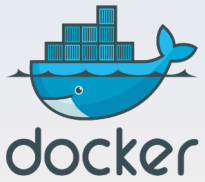
DevSecOps：应用程序安全交付新范式

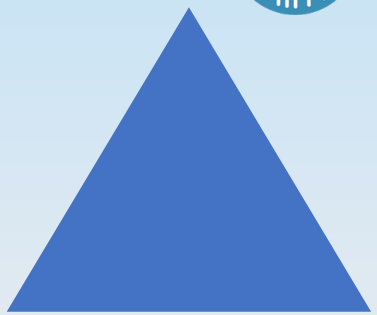




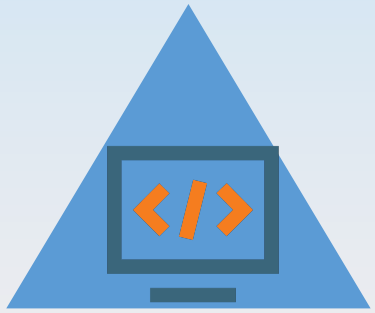
deployment.yml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: coscon-deployment
  labels:
    app: curl
spec:
  replicas: 3
  selector:
    matchLabels:
      app: curl
  template:
    metadata:
      labels:
        app: curl
    spec:
      containers:
      - name: curl
        image: dllhb/devsecops-curl:4.0.0
        ports:
        - containerPort: 80
        env:
        - name: USERNAME
          value: "xiaomage"
        - name: PASSWORD
          vaule: "HelloCosCon@2023"
```



深度 + 广度：纵深防御



Single source truth of source code

Code Review

Audit

Secret Detection

SCA

XAST

Fuzzing testing

Pen testing

License compliance

Dockerfile best practice

Audit

Image scanning

Image signature

RBAC

Network Policy

Audit

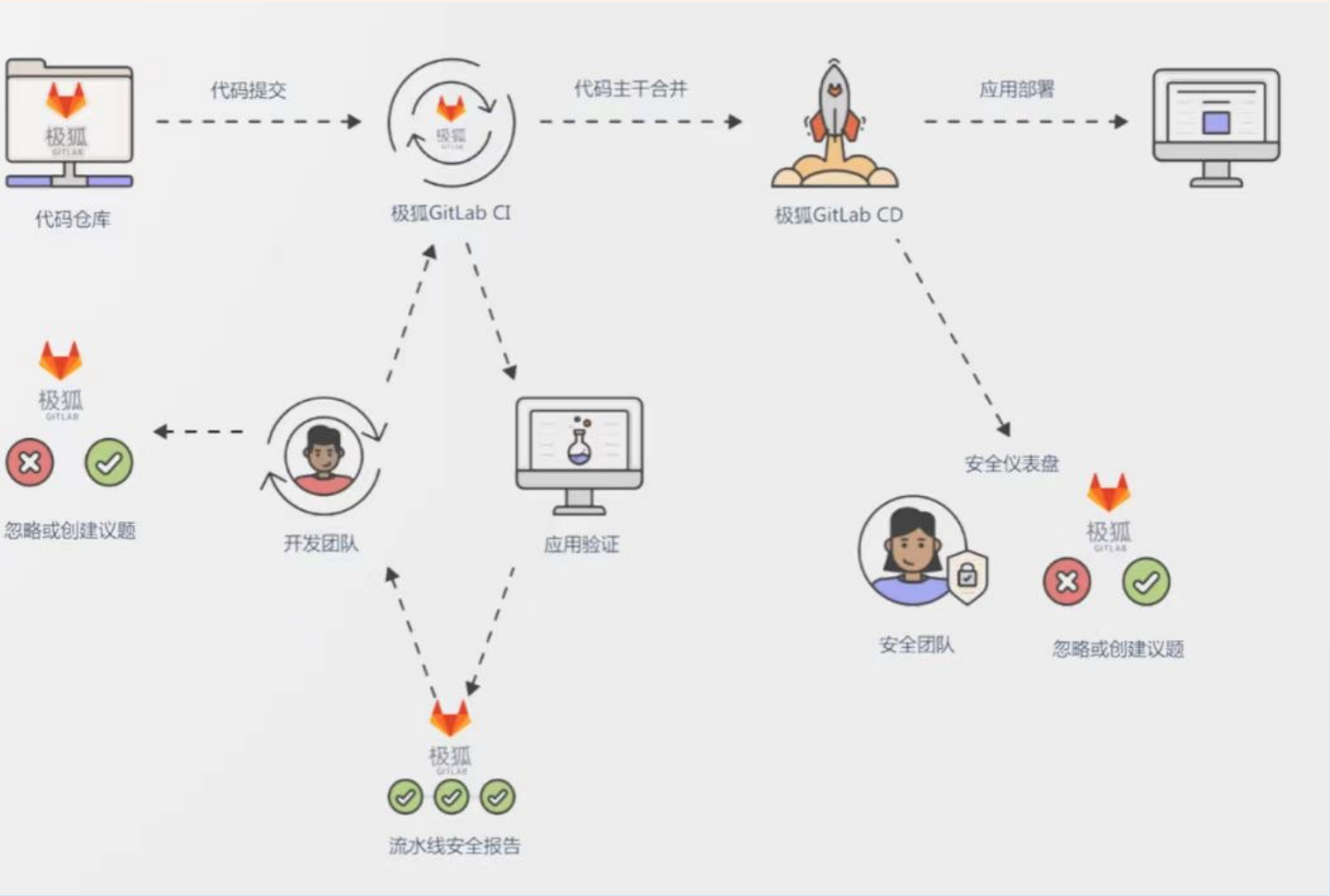
Resource limit

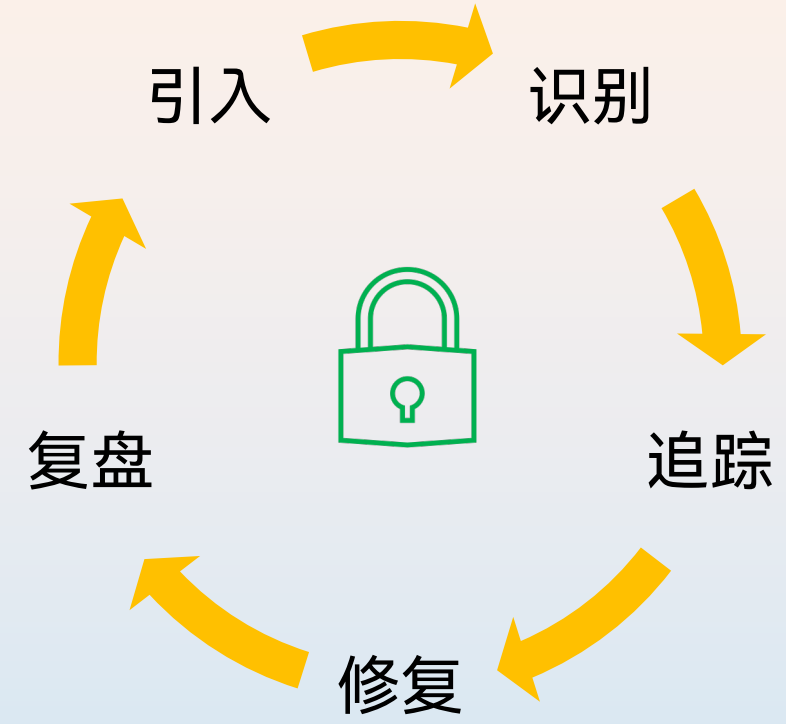
etcd enforce

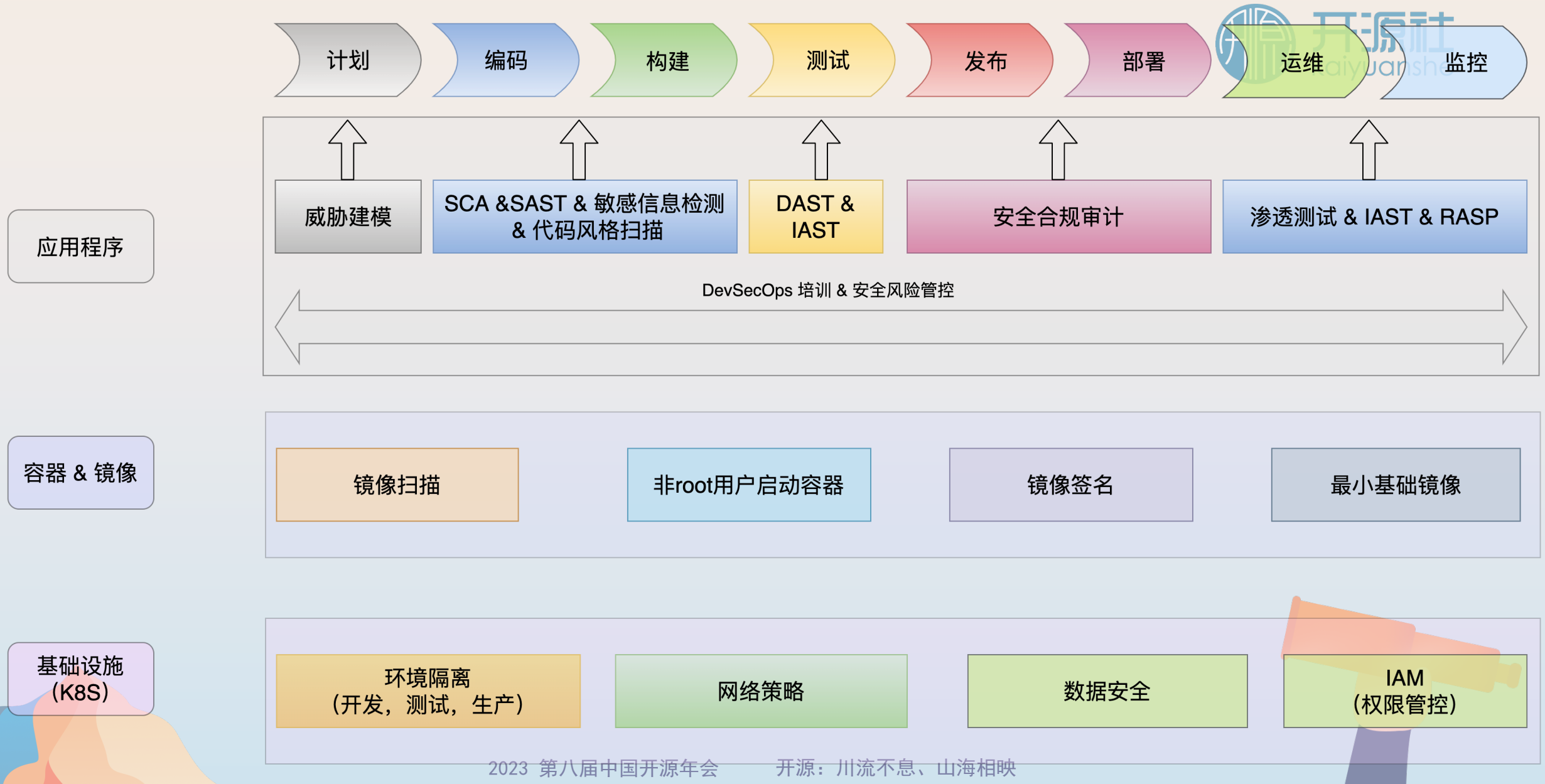
安全持续左移，早发现，早修复

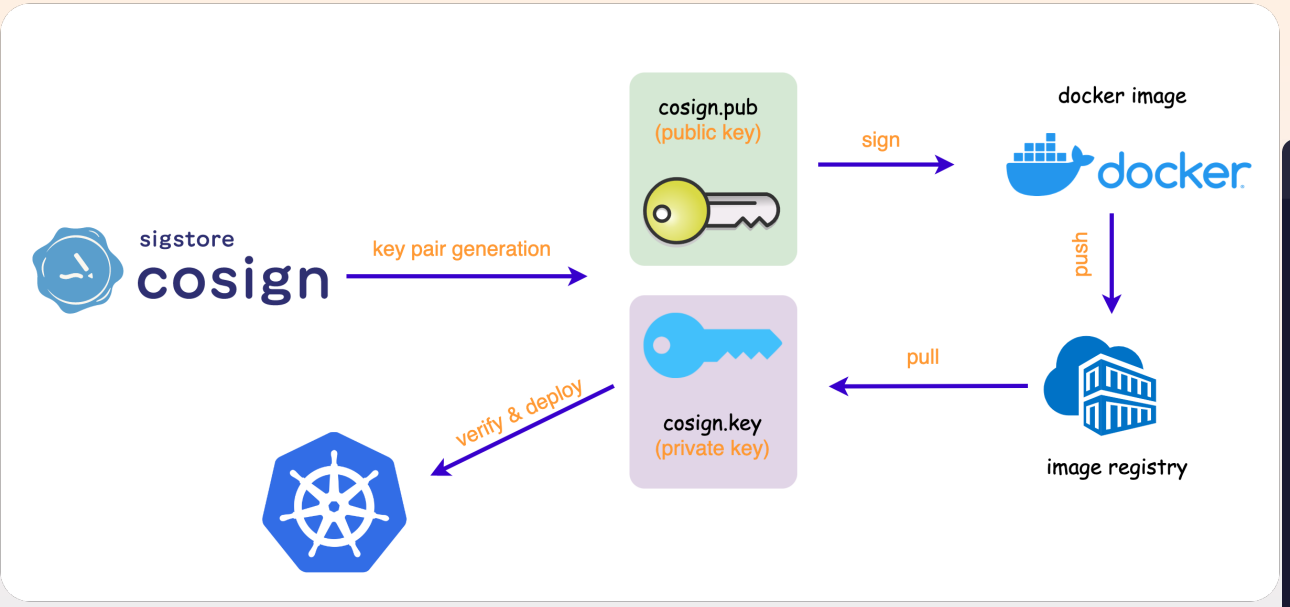
从静态到动态，从简单到复杂

持续投入应对动态变化的安全


代码提交
代码主干合并
应用部署
代码仓库
极狐GitLab CI
极狐GitLab CD
忽略或创建议题
开发团队
应用验证
安全仪表盘
安全团队
忽略或创建议题
流水线安全报告


引入
识别
追踪
修复
复盘


计划
编码
构建
测试
发布
部署
运维
监控
应用程序
威胁建模
SCA &SAST & 敏感信息检测 & 代码风格扫描
DAST & IAST
安全合规审计
渗透测试 & IAST & RASP
DevSecOps 培训 & 安全风险管控
容器 & 镜像
镜像扫描
非root用户启动容器
镜像签名
最小基础镜像
基础设施 (K8S)
环境隔离 (开发，测试，生产)
网络策略
数据安全
IAM (权限管控)



cosign

```
$ cosign verify --key cosign.pub dllhb/go-demo:v1.0.0 | jq .
Verification for index.docker.io/dllhb/go-demo:v1.0.0 --
The following checks were performed on each of these signatures:
  - The cosign claims were validated
  - The signatures were verified against the specified public key
  - Any certificates were verified against the Fulcio roots.
[
  {
    "critical": {
      "identity": {
        "docker-reference": "index.docker.io/dllhb/go-demo"
      },
      "image": {
        "docker-manifest-digest": "sha256:888988496480bb0be8984b43a84970589e41110a2d25440f145031fd396dd2db"
      },
      "type": "cosign container image signature"
    },
    "optional": null
  }
]
```

snappify.com

cosign

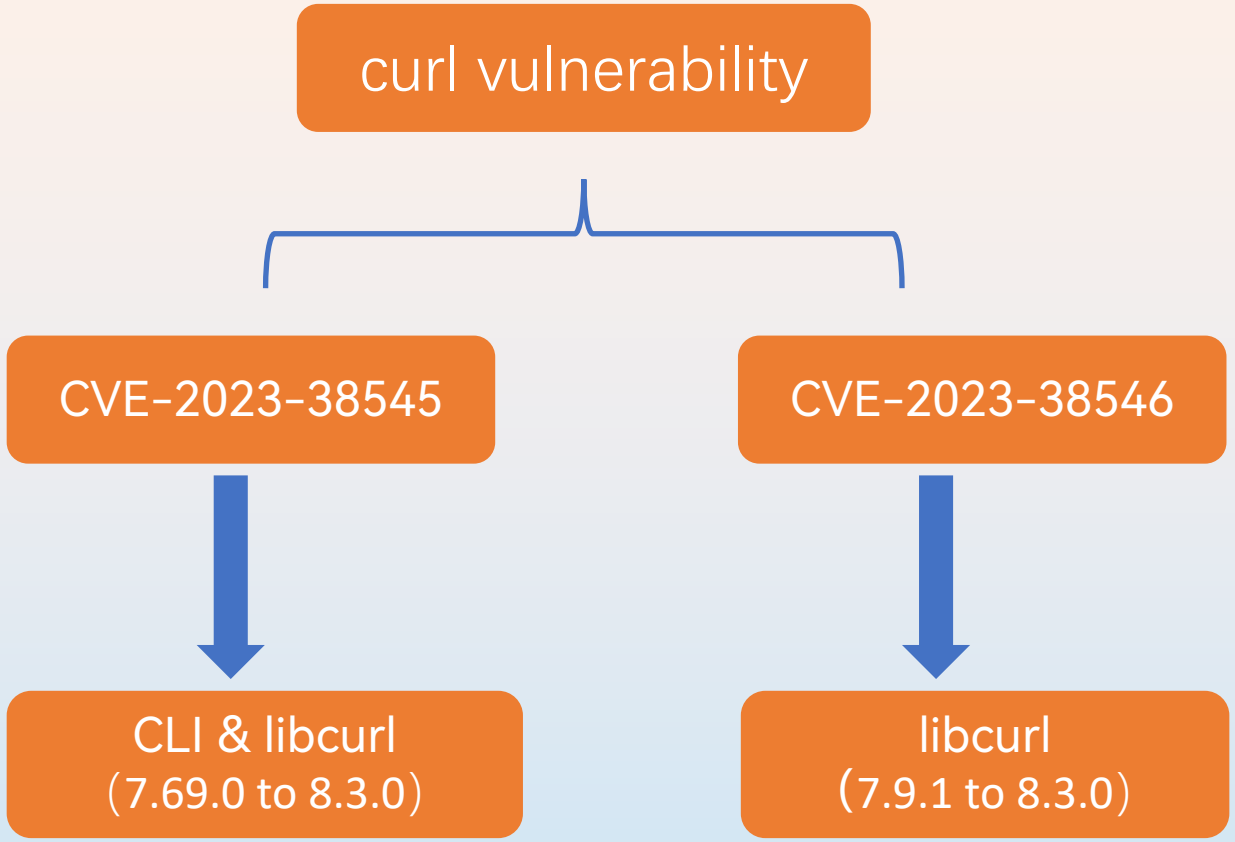
```
$ cosign sign --key cosign.key dllhb/go-demo:v1.0.0
Enter password for private key:
Pushing signature to: index.docker.io/dllhb/go-demo
```

snappify.com

cosign

```
$ cosign verify --key cosign.pub dllhb/go-demo:v1.0.0 | jq .
Error: no matching signatures:

main.go:46: error during command execution: no matching signatures:
```

snappify.com



syft-sbom

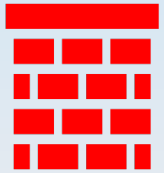
```
syft packages dllhb/curl-devsecops:4.0.0
New version of syft is available: 0.94.0
 ✔ Parsed image
 ✔ Cataloged packages      [19 packages]

NAME                    VERSION       TYPE
alpine-baselayout       3.2.0-r16     apk
alpine-keys             2.4-r0        apk
apk-tools               2.12.7-r0     apk
brotli-libs             1.0.9-r5      apk
busybox                 1.33.1-r8     apk
ca-certificates         20230506-r0   apk
ca-certificates-bundle  20220614-r0   apk
curl                    8.0.1-r0      apk
libc-utils              0.7.2-r3      apk
libcrypto1.1            1.1.1t-r2     apk
libcurl                 8.0.1-r0      apk
libretls                3.3.3p1-r3    apk
libssl1.1               1.1.1t-r2     apk
musl                    1.2.2-r4      apk
musl-utils              1.2.2-r4      apk
nghttp2-libs            1.43.0-r0     apk
scanelf                 1.3.2-r0      apk
ssl_client              1.33.1-r8     apk
zlib                    1.2.12-r3     apk
```

snappify.com

AIGC 浪潮下的应用安全思考

开源社 kaiyuanshe

AIGC 降低了漏洞修复，安全落地的门槛



test.js

```
const express = require('express');
const app = express();
const port = 3000;

// 不安全：直接调用 eval 解析 json，有安全漏洞，触发静态代码扫描告警
const sUserInput = getURLParam("json_val");
const jsonstr1 = `{"name":"a","company":"b","value":"${sUserInput}"}`;
const json1 = eval(`(${jsonstr1})`);

// Static Files
app.use(express.static('public'));
app.use('/css', express.static(__dirname + 'public/css'));
app.use('/js', express.static(__dirname + 'public/js'));
app.use('/img', express.static(__dirname + 'public/img'));


//  Listen on port 3000
app.listen(port, () => console.info(`Listening on port ${port}`));

const email_validator = require('./email_validator.js');
email_validator('workshop@jihulab.com')

```


CVE List CNAs WGs Board About News & Blog

NVD Go to for: CVSS Scores CPE Info

Search CVE List | Downloads | Data Feeds | Update a CVE Record | Request CVE IDs

TOTAL CVE Records: 200731

NOTICE: Transition to the all-new CVE website at WWW.CVE.ORG and CVE Record Format JSON are underway.

NOTICE: Changes are coming to CVE List Content Downloads in 2023.

HOME > CVE > CVE-2021-44832

Printer-Friendly View

CVE-ID

CVE-2021-44832 Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Apache Log4j2 versions 2.0-beta7 through 2.17.0 (excluding security fix releases 2.3.2 and 2.12.4) are vulnerable to a remote code execution (RCE) attack when a configuration uses a JDBC Appender with a JNDI LDAP data source URI when an attacker has control of the target LDAP server. This issue is fixed by limiting JNDI data source names to the java protocol in Log4j2 versions 2.17.1, 2.12.4, and 2.3.2.

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CISCO:20211210 Vulnerabilities in Apache Log4j Library Affecting Cisco Products: December 2021
- URL:https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-apache-log4j-qRuKNEbd
- CONFIRM:https://cert-portal.siemens.com/productcert/pdf/ssa-784507.pdf
- URL:https://cert-portal.siemens.com/productcert/pdf/ssa-784507.pdf
- CONFIRM:https://security.netapp.com/advisory/ntap-20220104-0001/
- URL:https://security.netapp.com/advisory/ntap-20220104-0001/
- FEDORA:FEDORA-2021-1bd9151bab
- URL:https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/T57MPJUW3MA6QGWZRTMCHHMMPQNVKGFC/
- FEDORA:FEDORA-2021-c6f471ce0f
- URL:https://lists.fedoraproject.org/archives/list/package-announce@lists.fedoraproject.org/message/EVV25FXL4FU5X6X5BSL7RLQ7T6F65MRA/
- MISC:https://issues.apache.org/jira/browse/LOG4J2-3293
- URL:https://issues.apache.org/jira/browse/LOG4J2-3293
- MISC:https://lists.apache.org/thread/s1o5vlo78ypqxnzn6p8zf6t9shtq5143
- URL:https://lists.apache.org/thread/s1o5vlo78ypqxnzn6p8zf6t9shtq5143
- MISC:https://www.oracle.com/security-alerts/cpuapr2022.html
- URL:https://www.oracle.com/security-alerts/cpuapr2022.html
- MISC:https://www.oracle.com/security-alerts/cpujan2022.html
- URL:https://www.oracle.com/security-alerts/cpujan2022.html
- MISC:https://www.oracle.com/security-alerts/cpujul2022.html
- URL:https://www.oracle.com/security-alerts/cpujul2022.html
- MLIST:[debian-lts-announce] 20211229 [SECURITY] [DLA 2870-1] apache-log4j2 security update

test.js

```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4
5 // 不安全：直接调用 eval 解析 json，有安全漏洞，触发静态代码扫描告警
6 const sUserInput = getURLParam("json_val");
7 const jsonstr1 = `{"name":"a","company":"b","value":"${sUserInput}"}`;
8 const json1 = eval(`(${jsonstr1})`);
9
10 // Static Files
11 app.use(express.static('public'));
12 app.use('/css', express.static(__dirname + 'public/css'));
13 app.use('/js', express.static(__dirname + 'public/js'));
14 app.use('/img', express.static(__dirname + 'public/img'));
15
16
17 //  Listen on port 3000
18 app.listen(port, () => console.info(`Listening on port ${port}`));
19
20 const email_validator = require('./email_validator.js');
21 email_validator('workshop@jihulab.com')
22
```

J

please find the vulnerability in below code


```
const express = require('express');
const app = express();
const port = 3000;

// 不安全: 直接调用 eval 解析 json, 有安全漏洞，触发静态代码扫描告警
const sUserInput = getURLParam("json_val");
const jsonstr1 = `{"name":"a","company":"b","value":"${sUserInput}"}`;
const json1 = eval(`(${jsonstr1})`);

// Static Files
app.use(express.static('public'));
app.use('/css', express.static(__dirname + 'public/css'));
app.use('/js', express.static(__dirname + 'public/js'));
app.use('/img', express.static(__dirname + 'public/img'));


//  Listen on port 3000
app.listen(port, () => console.info(`Listening on port ${port}`));

const email_validator = require('./email_validator.js');
email_validator('workshop@jihulab.com')
```


在提供的代码中，我们可以看到以下漏洞：

1. eval函数调用：代码中使用eval函数解析JSON字符串，这种做法是不安全的。因为使用eval函数可以执行任意的JavaScript代码，攻击者可以利用这个函数注入恶意代码，从而造成安全漏洞。
2. 未对输入进行验证：代码中的sUserInput变量从getURLParam函数中获取，但是并未对其进行任何的输入验证或过滤，这可能导致诸如跨站脚本攻击（XSS）等问题。
3. 未使用HTTPS：代码中未使用HTTPS协议进行通信，这可能会导致数据在传输过程中被窃听或篡改，造成安全风险。

test.js

```
1 const express = require('express');
2 const app = express();
3 const port = 3000;
4
5 // 不安全：直接调用 eval 解析 json，有安全漏洞，触发静态代码扫描告警
6 const sUserInput = getURLParam("json_val");
7 const jsonstr1 = `{"name":"a","company":"b","value":"${sUserInput}"}`;
8 const json1 = eval(`(${jsonstr1})`);
9
10 // Static Files
11 app.use(express.static('public'));
12 app.use('/css', express.static(__dirname + 'public/css'));
13 app.use('/js', express.static(__dirname + 'public/js'));
14 app.use('/img', express.static(__dirname + 'public/img'));
15
16
17 //  Listen on port 3000
18 app.listen(port, () => console.info(`Listening on port ${port}`));
19
20 const email_validator = require('./email_validator.js');
21 email_validator('workshop@jihulab.com')
22
```

.gitlab-ci.yml

```
include:
  - template: Security/SAST.gitlab-ci.yml

stages:
  - test
```


Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

状态: Detected

说明: Detected eval(variable), which could allow a malicious actor to run arbitrary code.

项目: demo-show / dev-sec-ops-devsecops

文件: app.js:8

标识: eslint.detect-eval-with-expression , CWE-95 , A1 - Injection , ESLint rule ID security/detect-eval-with-expression

严重级别: 中

工具: SAST

扫描工具提供者: Semgrep

取消 | 忽略漏洞 | 创建议题

流水线 #1127050 已通过 使用提交 30efb250 于 sast-demo 刚刚

需要 2 个来自 Security Approve 的批准。

核准人	核准	已评论	已核准人
所有符合条件的用户	可选		
Security Approve	0 / 2		

其余1项

安全扫描检测到1个潜在的漏洞 0个严重， 0个高危 以及 其他1项 完整报告

SAST检测到1个潜在的漏洞 0个严重， 0个高危 以及 其他1项

新增

中 Improper Neutralization of Directives in Dynamically Evaluated Code ('Eval Injection')

下面是改进后的代码示例：

```php
const express = require('express');
const app = express();
const port = 3000;

// 安全：使用JSON.parse()代替eval函数
const sUserInput = getURLParam("json_val");
const jsonstr1 = `{"name":"a","company":"b","value":"${sUserInput}"}`;
const json1 = JSON.parse(jsonstr1);

// 对输入进行验证和过滤
function getURLParam(param) {
    const urlParams = new URLSearchParams(window.location.search);
    const value = urlParams.get(param);
    return value ? value.replace(/<[^>]*>/g, '') : '';
}

// Static Files
app.use(express.static('public'));
app.use('/css', express.static(__dirname + 'public/css'));
app.use('/js', express.static(__dirname + 'public/js'));
app.use('/img', express.static(__dirname + 'public/img'));


//  使用HTTPS协议
const fs = require('fs');
const https = require('https');
const options = {
    key: fs.readFileSync('server.key'),
    cert: fs.readFileSync('server.crt')
};

https.createServer(options, app).listen(port, () => console.info(`Listening

const email_validator = require('./email_validator.js');
email_validator('workshop@jihulab.com');
```

Update 2 files

编辑 代码

开放中 JingHe Ma 请求将 sast-demo 合并到 main 4分钟前

概览 0 提交 2 流水线 2 变更 2

- /.gitlab-ci.yml
- /app.js

0 0

流水线 #1127053 已通过 使用提交 3959c13f 于 sast-demo 1分钟前

Collapse eligible ap

需要 2 个来自 Security Approve 的批准。

核准人	核准	已评论	已核准人
所有符合条件的用户	可选		
Security Approve	0 / 2		

其余1项

安全扫描 未检测到新漏洞。

完整报告

SAST 未检测到新漏洞。

合并受阻：必须获得所有必需的批准。

合并详情

- 2 提交 和 1 个合并提交 将被添加到 main。
- 源分支将被删除。

开源社 kaiyuanshe

Code Review

Thread Model

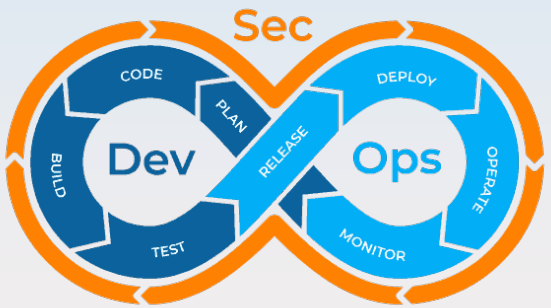
Vulnerability Identify

Application security testing

Security Training

……

Tools Suggestion



https://levelup.gitconnected.com/complete-chatgpt-guide-for-devsecops-top-20-most-essential-prompts-ef21e0aa4830

开源社
kaiyuanshe

THANK YOU

QUESTIONS?

欢迎扫码打卡
积分可兑换对应礼品哟！

微信公众号：开源社KAIYUANSHE
视频号：开源社KAIYUANSHE
新浪微博：开源社
B站：开源社KAIYUANSHE

简书：开源社
头条：开源社
Facebook：KaiyuansheChina
Twitter：开源社KAIYUANSHE

扫码关注开源社公众号

扫码添加讲师联系方式