DevSecOps : de la sécurité dans mon DevOps

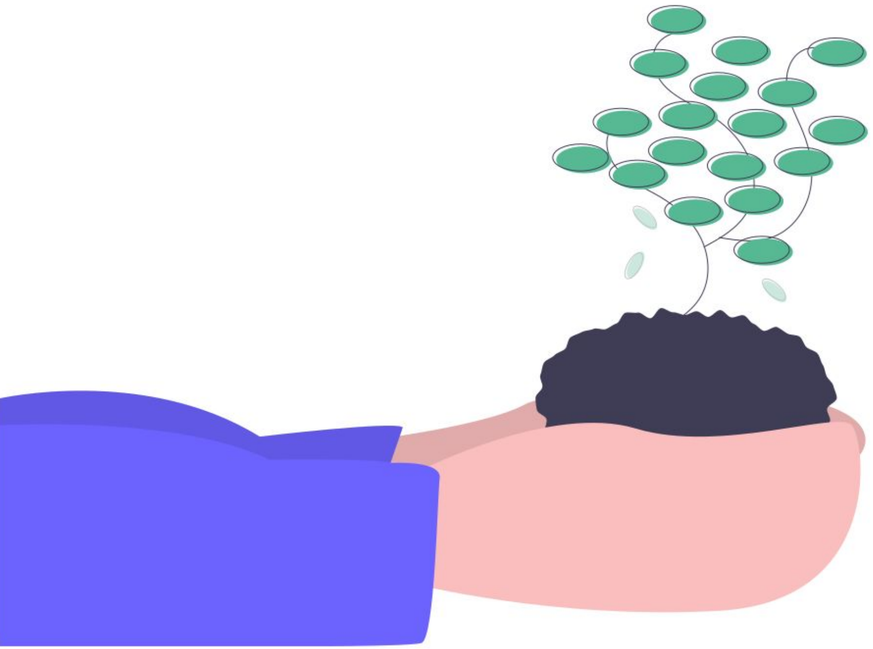
Introduction

@tgrall

@adrienpessu

Introduction

Introduction

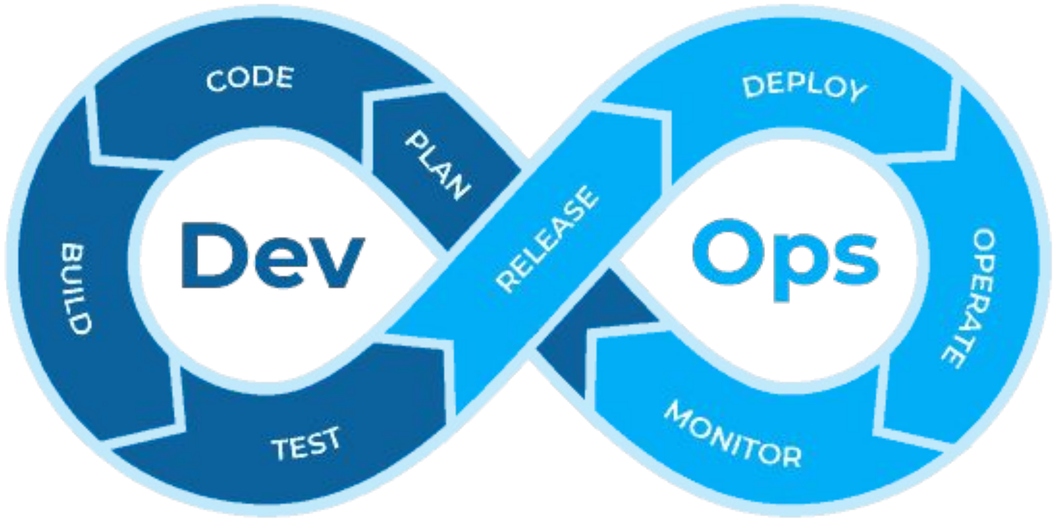


Introduction

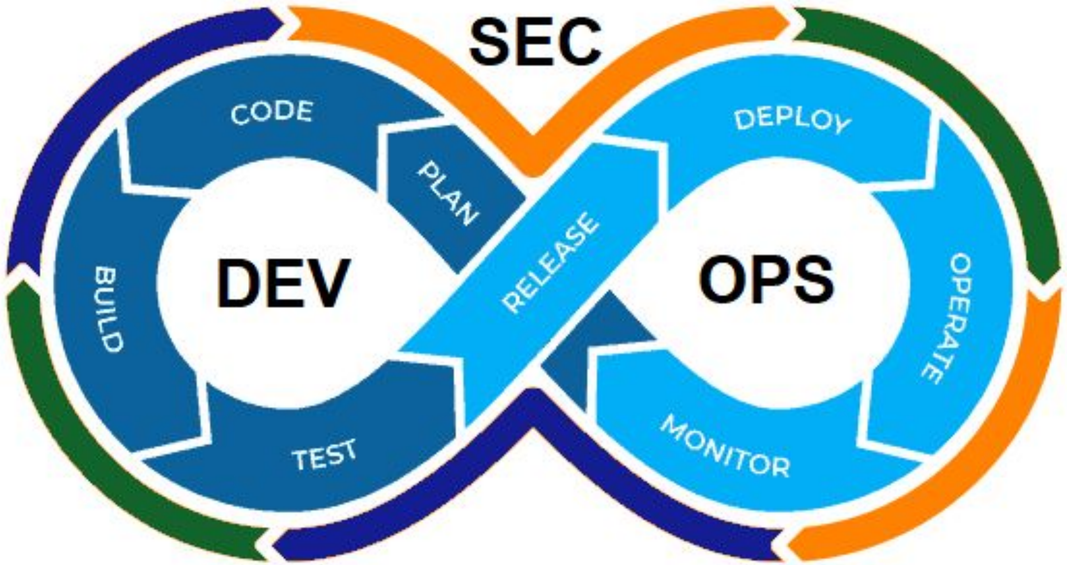
Oublier la cybersécurité, c'est "rouler à 200 km/h à moto sans casque”

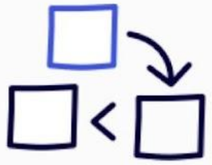
Guillaume Poupard, (futur ex-)patron de l'Anssi

Introduction



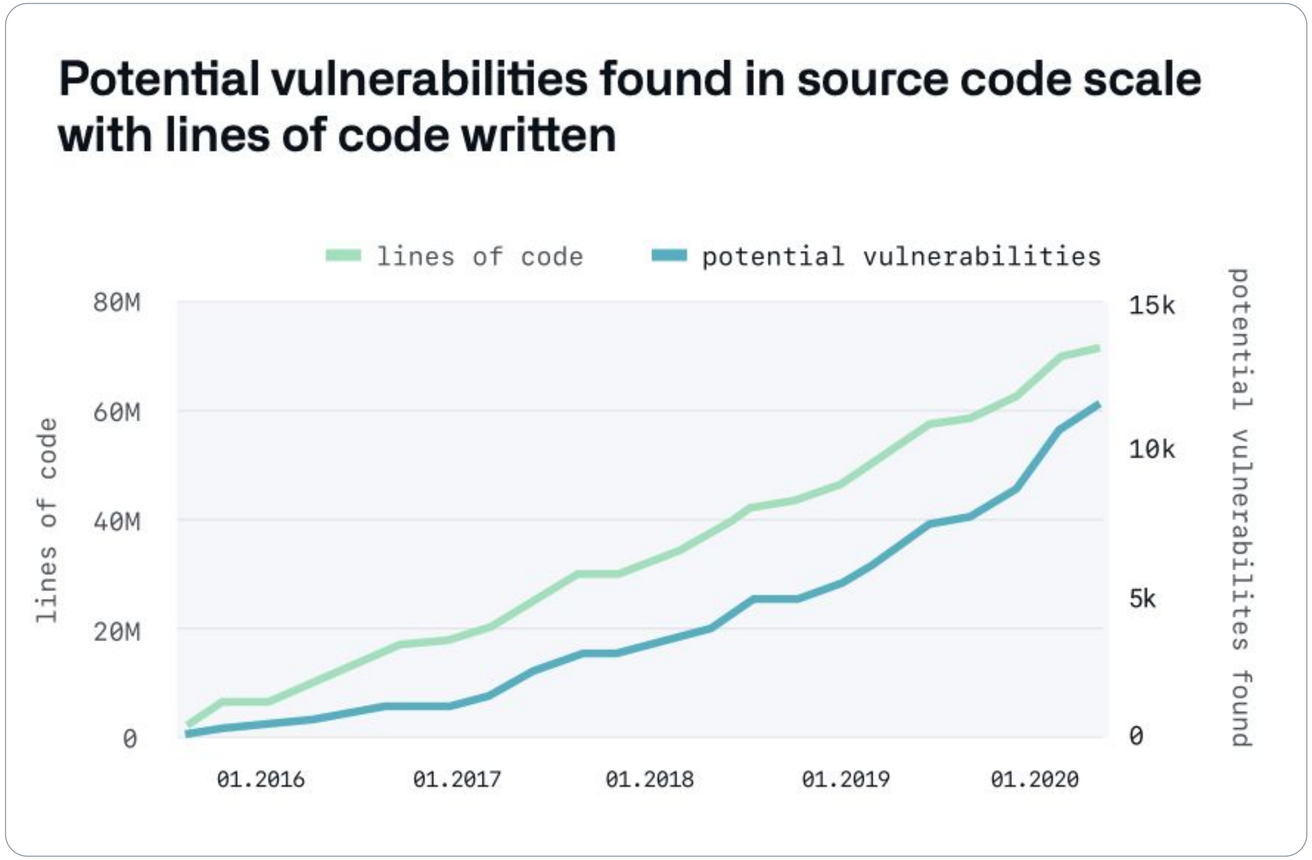
Introduction



				
Backlog	Code	Test	Deploy	Monitor

Backlog	Code	Test	Deploy	Monitor
Threat Modeling	SAST/SCA	DAST	IAST	Penetration test

Application Security : aujourd'hui



SCA (SOFTWARE COMPOSITION ANALYSIS)

GitHub Advisory Database / GitHub Reviewed / CVE-2021-44228

Remote code injection in Log4j

Critical severity | GitHub Reviewed | Published on 10 Dec 2021 • Updated 27 days ago

Vulnerability details | Dependabot alerts 0

Package	Affected versions	Patched versions
org.apache.logging.log4j:log4j-core (Maven)	>= 2.13.0, < 2.15.0 < 2.3.1 >= 2.4, < 2.12.2	2.15.0 2.3.1 2.12.2

Description

Summary

Log4j versions prior to 2.16.0 are subject to a remote code execution vulnerability via the ldap JNDI parser.
As per Apache's Log4j security guide: Apache Log4j2 <=2.14.1 JNDI features used in configuration, log messages, and parameters do not protect against attacker controlled LDAP and other JNDI related endpoints. An attacker who can control log messages or log message parameters can execute arbitrary code loaded from LDAP servers when message lookup substitution is enabled. From log4j 2.16.0, this behavior has been disabled by default.

Log4j version 2.15.0 contained an earlier fix for the vulnerability, but that patch did not disable attacker-controlled JNDI lookups in all situations. For more information, see the `Updated advice for version 2.16.0` section of this advisory.

Impact

Logging untrusted or user controlled data with a vulnerable version of Log4J may result in Remote Code Execution (RCE) against your application. This includes untrusted data included in logged errors such as exception traces, authentication failures, and other unexpected vectors of user controlled input.

Affected versions

Any Log4J version prior to v2.15.0 is affected to this specific issue.

The v1 branch of Log4J which is considered End Of Life (EOL) is vulnerable to other RCE vectors so the recommendation is to still update to 2.16.0 where possible.

Security releases

Additional backports of this fix have been made available in versions 2.3.1, 2.12.2, and 2.12.3

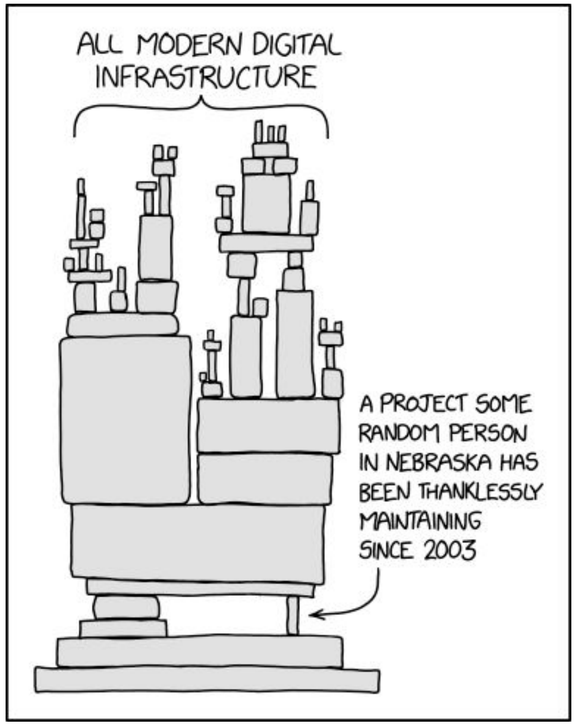
Affected packages

Only the `org.apache.logging.log4j:log4j-core` package is directly affected by this vulnerability. The `org.apache.logging.log4j:log4j-api` should be kept at the same version as the `org.apache.logging.log4j:log4j-core` package to ensure compatability if in use.

Remediation Advice

Updated advice for version 2.16.0

The Apache Logging Services team provided updated mitigation advice upon the release of version 2.16.0, which disables JNDI by



Copyright XKCD

SCA (SOFTWARE COMPOSITION ANALYSIS)

Checkmarx

DEPENDENCY-TRACK

sonatype

OWASP®

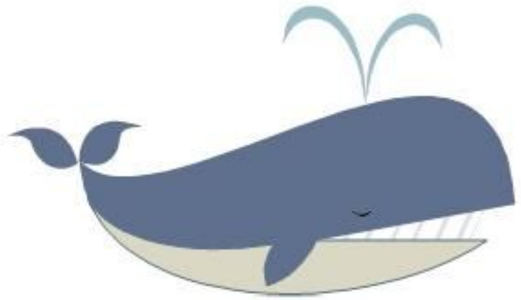
npm-audit

snyk

VERACODE

https://owasp.org/www-community/Component_Analysis

Docker



Falco

Docker

```
$ docker scan --file Dockerfile docker-scan:e2e
Testing docker-scan:e2e
...
✗ High severity vulnerability found in perl
  Description: Integer Overflow or Wraparound
  Info: https://snyk.io/vuln/SNYK-DEBIAN10-PERL-570802
  Introduced through: git@1:2.20.1-2+deb10u3, meta-common-packages@meta
  From: git@1:2.20.1-2+deb10u3 > perl@5.28.1-6
  From: git@1:2.20.1-2+deb10u3 > liberror-perl@0.17027-2 > perl@5.28.1-6
  From: git@1:2.20.1-2+deb10u3 > perl@5.28.1-6 > perl/perl-modules-5.28@5.28.1-6
  and 3 more...
  Introduced by your base image (golang:1.14.6)

Organization:      docker-desktop-test
Package manager:   deb
Target file:       Dockerfile
Project name:      docker-image|99138c65ebc7
Docker image:      99138c65ebc7
Base image:        golang:1.14.6
Licenses:          enabled

Tested 200 dependencies for known issues, found 157 issues.

According to our scan, you are currently using the most secure version of the selected base image
```

SAST (STATIC APPLICATION SECURITY TESTING)

Checkmarx

{} Find Security Bugs

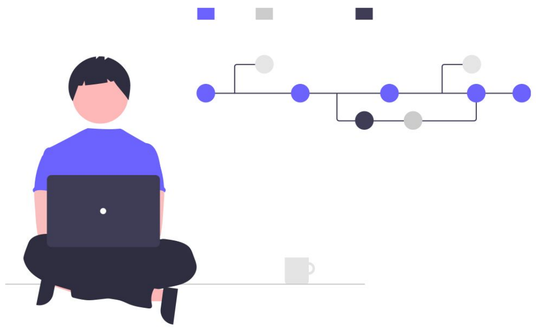
phpcs-security-audit v3

PMD
DON'T SHOOT THE MESSENGER

snyk

CODACY

GRAMMATECH



sonarqube

VERACODE

SAST (STATIC APPLICATION SECURITY TESTING)

<div text [innerHTML]="‘‘greeting.translate.key' | translate:{**name**: **name**}”>

Bonjour {{ name }}, bienvenue au DevFest

Bonjour Adrien, bienvenue au DevFest

SAST (STATIC APPLICATION SECURITY TESTING)

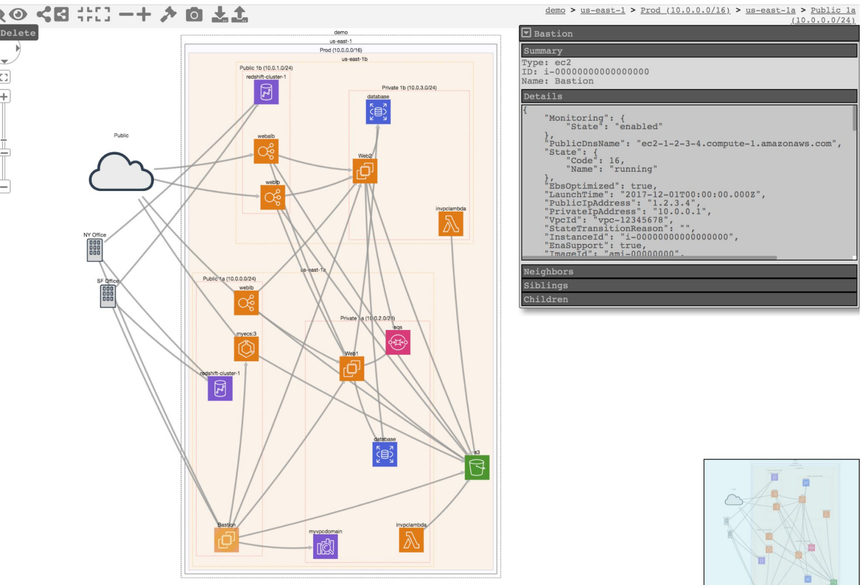
What about my **Secrets** ?

- Exposing your AWS access keys on Github can be extremely costly. (2017)

IaC

```
❯ tfsec /tmp/example

#1 CRITICAL Listener for application load balancer does not use HTTPS.

main.tf Line 4

   2  resource "aws_alb_listener" "my-alb-listener" {
   3    port     = "80"
   4    protocol = "HTTP"          "HTTP"
   5  }

          ID aws-elb-http-not-used
      Impact Your traffic is not protected
  Resolution Switch to HTTPS to benefit from TLS security features

  More Information
  - https://registry.terraform.io/providers/hashicorp/aws/latest/docs/resources/lb_listener
```

Cloud mapper



Commands

- `audit` : Check for potential misconfigurations.
- `collect` : Collect metadata about an account. More details here.
- `find_admins` : Look at IAM policies to identify admin users and roles, or principals with specific privileges. More details here.
- `find_unused` : Look for unused resources in the account. Finds unused Security Groups, Elastic IPs, network interfaces, volumes and elastic load balancers.
- `prepare` / `webserver` : See Network Visualizations
- `public` : Find public hosts and port ranges. More details here.
- `sg_ips` : Get geoip info on CIDRs trusted in Security Groups. More details here.
- `stats` : Show counts of resources for accounts. More details here.
- `weboftrust` : Show Web Of Trust. More details here.
- `report` : Generate HTML report. Includes summary of the accounts and audit findings. More details here.
- `iam_report` : Generate HTML report for the IAM information of an account. More details here.

If you want to add your own private commands, you can create a `private_commands` directory and add them there.

https://github.com/duo-labs/cloudmapper

Cloud tracker

```
cloudtracker --account demo --user alice
...
  cloudwatch:describealarmhistory
  cloudwatch:describealarms
- cloudwatch:describealarmsformetric
- cloudwatch:getdashboard
? cloudwatch:getmetricdata
...
+ s3:createbucket
...
```

https://github.com/duo-labs/cloudtracker

Conclusion


Backlog
Code
Test
Deploy
Monitor
Threat Modeling
SAST/SCA
DAST
IAST
Penetration test