**Secur DevOps** Formation 25 mai 2023 - CRF

OWASP

A quel endroit Ci/cd ? build (ex : peut-on acheter nombre négatif de produits)

ShiftLeft (Security By Design): prendre en compte secu AVANT la decouverte dans les news

2 .Contexte

ENISA European Union Agency for Cybersecurity : *(ex WannaCry ransomware attack)*

<https://en.wikipedia.org/wiki/Server_Message_Block>

*Amazon pousse un projet en prod chaque 11 secondes*

EP005 : Project Zero, Zero-day attack, Ms Aurora, loPHT

SCA - SAST - DAST - IaC

**SAST (Static Application Security Testing**) is the automated analysis of written code (compiled or uncompiled) for security vulnerabilities. SAST products parse your code into different pieces that it can further analyze, in order to find vulnerabilities that are many layers deep in regard to functions and subroutines.

Fortify (Opentext) – Semgrep.dev – Checkmarx.com

**DAST (Dynamic Application Security Testing)** is interaction with your running application with the purpose of finding and managing vulnerabilities it may have. DAST testing is partially an attempt to find business logic problems, but also to find implementation problems (security bugs/code problems).

BurpSuite (Portswigger) – Qualys.com – Tenable.com (.io)

**Software Composition Analysis (SCA)** is verification of the third-party libraries, frameworks and components used within your application; all of the code that you and your team did not write is considered by SCA tools. The CVE (Common Vulnerability Enumeration) database, exploit DB, independent security researchers, research that the vendor has performed themselves, and information released by the frameworks or third-party component creators.

Dependency-check (owasp.org) - Sonatype.com

**Fuzzing** is submitting malformed input to an application to verify that it fails gracefully

<https://www.clouddefense.ai/blog/the-differences-between-sca-sast-and-dast>

<https://docs.gitlab.com/ee/user/application_security/sast/>

**IaC - Infrastructure As Code**

Ansible.com – Terraform.io (Hashicorp)

SECURE BY DESIGN

Penser la sécurité comme une fonctionnalité :

* SecDevOps
* Critère d’acceptance du BDD (As an .., I want to secure…)
* Modéliser les menaces (Abuser Story, Evil-user Story..)
* Rendre la sécurité accessible (supports, conseils..)

<https://www.vaadata.com/blog/fr/pentest-application-saas-quels-sont-les-principaux-enjeux-de-securite/>

* Respect des deadlines
* Qualité et sécurité
* Automatisation
* Responsabilisation

#### **Cloisonnement vertical des données**

Les applications SaaS fonctionnent généralement en proposant différents rôles pour les utilisateurs, par exemple : utilisateur, manager, administrateur.

#### **Cloisonnement horizontal des données**

La spécificité d’une plateforme SaaS, c’est qu’elle héberge tous ses clients sur une même infrastructure, des mêmes serveurs, une même base de données. La séparation des données selon les comptes se fait au niveau de la couche applicative, par des contrôles d’authentification et de droits.

*Chaîne de bug minime -> gros bug vulnérable*

Environnement de dev

* Poste de dev doit être sécurisé (antivirus, vpn..) Plugins des IDE, navigateurs
* Gestionnaire de code source sécurisé
  + Cloisement utilisateur / équipe
  + Tokens et clés avec Principe de Moindre Privilège

Gitlab.com – Bitbucket.org – Vaultproject.io - keeper.io

<https://docs.docker.com/engine/swarm/secrets/>

**Principes de Sécurité**

* Never Trust User Input (valider entrées utilisateur)
* Minimize Surface Attack Area (chaque élément ajoute un risque, use only necessary code)
* Whitelist Over Blacklist (auth spécifique)
* Principe of last privilege (que les droits nécessaires pour fonctionner)

4. OWASP

*is an online community that produces freely-available articles, methodologies, documentation, tools, and technologies in the field of web application security.*

+ A01:2021 Broken Access Control (Directory Path Traversal)

<https://portswigger.net/web-security>

+ A02:2021 Cryptographic Failure

<https://en.wikipedia.org/wiki/Rainbow_table>

Insecure TLS

<https://en.wikipedia.org/wiki/Advanced_Encryption_Standard>

<https://words.filippo.io/the-ecb-penguin/>

*Chiffrement : protège la donnée / Hash: vérifier l’intégrité*

+ A03:2021 Injections

<https://portswigger.net/web-security/xxe>

Os-command injection

+ A04:2021 Insecure design

Ex: redirection sans verification du GET

+ A05:2021 Security Misconfguration

* Comptes par default, pile d’erreur affichée, paramètre sécurité manquant, obsolète

+ A06:2021 Vulnerable outdated component

+ A07:2021 Identification and Auth Failure

* Auth autorise la brutforce, password défault ou connus, récupération mot de passe faible ou non hashé, id\_session dans l’url, réutilise l’id de session

+ A08:2021 Software and Data Intégrity Failure

+ A09:2021 Securty Logging and Monitoring Failure

+ A10:2021 Server Side Request Forgery

Kontra (training security): <https://application.security/>