

# ITÉRATION 1

## Faibles de sécurité

### Modalités

- Travail individuel en autonomie
- 1 jour en présentiel

### Objectifs

L'objectif de ce module est de pouvoir utiliser des bases de données de sécurité pour connaître les problèmes des logiciels existants.

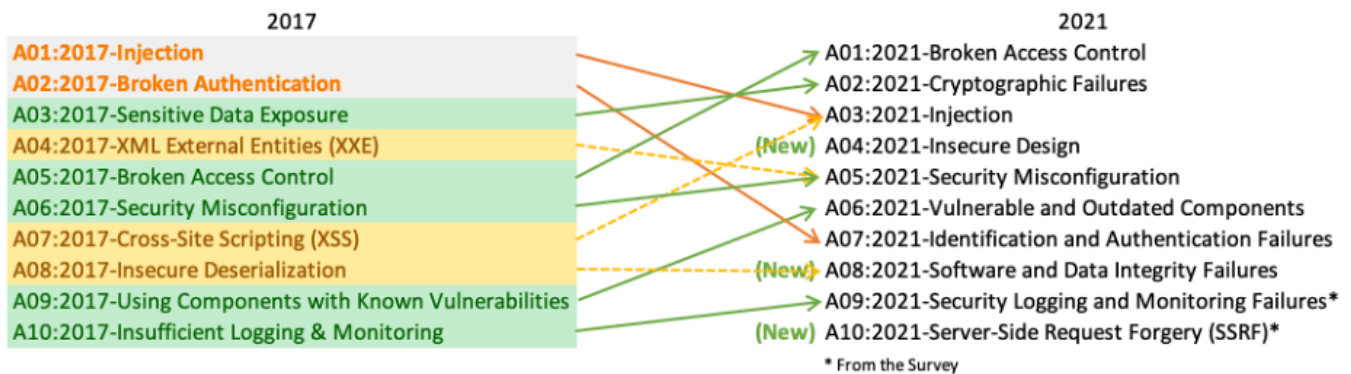
### Compétences

- Effectuer une veille technologique
- Identifier les problèmes et déterminer les équipes concernées
- Comprendre les normes communes de dénomination des problèmes de sécurité (CVE, GHSE etc)

## 1.1 – Retour à la liste OWASP

1h – Présentiel

Vous souvenez-vous de la liste OWASP des problèmes de sécurité les plus fréquents dans les applications Web ? Étudions à nouveau la liste (<https://owasp.org/www-project-top-ten/>)



La liste évolue dans le temps. Pouvez-vous dire pourquoi?

Vous pouvez consulter les descriptions des différentes catégories de problèmes. Quels types de problèmes avez-vous rencontrés au cours des mois précédents ? Pensez à un problème qui pourrait avoir un impact sur la sécurité du projet pendant le cours, votre travail ou une autre situation. Décrivez-le dans une note, trouvez une ou plusieurs catégories correspondantes dans la liste. La note pourrait avoir une forme comme :

Problème:

xxxxxx

Catégorie(s) OWASP et explication:

xxxxxx

### RESSOURCES

- La liste "top 10" de OWASP <https://owasp.org/www-project-top-ten/>

### COMPÉTENCES ASSOCIÉES

- Identifier les problèmes et déterminer les équipes concernées

## 1.2 – Un exemple de problème de sécurité

2h – Présentiel

Nous pensons souvent qu'un problème de sécurité peut provenir d'un site de connexion bancaire, d'une application Web ou d'un appareil. Cela peut être n'importe où... y compris votre éditeur de texte.

L'article que nous lisons ici est une véritable analyse de sécurité:

<https://blog.aquasec.com/can-you-trust-your-vscode-extensions>

Lisez l'article (également disponible en PDF à côté du kit) et essayez d'identifier tous les problèmes de sécurité avec leurs catégories OWASP. Pourriez-vous identifier d'où vient chaque problème (conception, l'implémentation ...) ?

Rédigez une note énumérant tous les problèmes que vous remarquez. Indiquez de quel module ils proviennent (l'application elle-même, le marché des extensions, la conception générale, autre...)

### RESSOURCES

- <https://blog.aquasec.com/can-you-trust-your-vscode-extensions>

### COMPÉTENCES ASSOCIÉES

- Identifier les problèmes et déterminer les équipes concernées
- 

## 1.3 – CVE - c'est quoi?

1h – Présentiel

Il est compliqué de parler de bugs spécifiques (dans ce cas des bugs de sécurité) en utilisant des descriptions seulement. C'est pourquoi les chercheurs en sécurité utilisent des noms CVE (Common Vulnerability Enumeration) au lieu de dire *"un problème dans sqlite découvert en 2021 qui provoque..."*


Un numéro CVE est un moyen unique d'identifier un problème de sécurité spécifique. Chaque CVE spécifique peut appartenir à une classe décrite par l'OWASP.

Il existe des bases de données stockant des informations sur les CVE : quel package est affecté, dans quelle version, quelle version contient un correctif, existe-t-elle une solution de contournement, etc.

Les deux plus importants sont CVE et son extension NVD (National Vulnerability Database). Regardez la vidéo pour savoir comment lire les entrées NVD.

D'autres bases de données existent également. Par exemple, GitHub permet de créer directement un avis de sécurité (*security advisory*, une description d'un problème et les fix associés). Une CVE peut être assignée, mais ce n'est pas obligatoire.

## RESSOURCES

-  CVE checking an entire distribution, Marta Rybczynska (2:06 - 9:18)
- <https://www.cve.org/>
- Un exemple de page NVD pour un problème en PHP:  
<https://nvd.nist.gov/vuln/detail/CVE-2022-31628>
- Un exemple de GitHub Security Advisory pour un module de Jenkins:  
<https://github.com/advisories/GHSA-p2fr-mq9m-6w6p>

## COMPÉTENCES ASSOCIÉES

- Comprendre les normes communes de dénomination des problèmes de sécurité (CVE, GHSA etc)

## 1.4 — Accéder à la base de données des vulnérabilités

3h — Présentiel

Nous pouvons accéder automatiquement aux bases de données de vulnérabilités pour savoir quels problèmes existent dans quelle version. Votre version php est vulnérable ? Vous pouvez le découvrir avec un petit script...

Votre tâche est d'écrire un script permettant de rechercher des problèmes connus dans une version donnée d'un package donné.

Si vous utilisez du code trouvé en ligne, marquez bien quelle partie est votre travail et laquelle vient de quelqu'un d'autre.

## RESSOURCES

- [https://en.wikipedia.org/wiki/Common\\_Platform\\_Enumeration](https://en.wikipedia.org/wiki/Common_Platform_Enumeration)
- <https://nvd.nist.gov/developers/vulnerabilities>

## COMPÉTENCES ASSOCIÉES

- Effectuer une veille technologique

- Comprendre les normes communes de dénomination des problèmes de sécurité (CVE, GHSA etc)
- 

## 1.5 (optionnel) – Conduit.. le retour

1h+ – Présentiel

Revenons au projet que nous connaissons... Installez "Conduit" à nouveau.

Êtes-vous en mesure de savoir lesquelles de ses dépendances ont des problèmes de sécurité ?

### RESSOURCES

- <https://github.com/LivingStoneHgS/conduit-backend-tests>

### COMPÉTENCES ASSOCIÉES

- Effectuer une veille technologique
  - Identifier les problèmes et déterminer les équipes concernées
  - Comprendre les normes communes de dénomination des problèmes de sécurité (CVE, GHSA etc)
- 

## Livrables

Dans ce projet, nous devrions avoir :

- Une note sur une expérience d'un problème de sécurité
- Une note sur les problèmes de sécurité pour le VSCode mentionné dans le blog
- Un script pour accéder aux CVE d'un programme donné à partir de la base de données NVD