Historique du projet

- Projet créé en 2013 (je n'ai pas le nom de l'auteur original).
- **Ricardo Iramar** (localisé au Brésil) à repris la gestion du projet en 2015.
- J'ai rejoint le projet en 2021 en tant que contributeur puis en tant que co-gestionnaire en 2022.

L'OWASP

- L'OWASP est l'acronyme de Open WorldWide Application Security Project est une fondation à but non lucratif, de loi américaine, créée en décembre 2001:
  - About the OWASP Foundation
- Elle a pour finalité de mettre à disposition des ressources (documentation, outils, guides,etc.) afin d'aider à améliorer le niveau de sécurité des logiciels. Initialement le type de logiciel était le web mais la fondation s'est diversifiée aux autres types comme le mobile, l'embarqué, etc.
- Ses moyens de financements sont principalement les suivants:
  - Sponsors.
  - Cotisations des membres (50$ par an / 500$ à vie): Membership
  - Bénéfices suite à l'organisation de conférences.
- Les dirigeants de la fondation sont élus par les membres tous les 5 ans (de mémoire):
  - Global Board
- Bien que initialement américaine, la fondation possède des "chapitres" (représentation locale de la fondation) dans de nombreux pays dont notamment le France, pour citer le pays hôte du podcast:
  - https://owasp.org/www-chapter-france/
  - https://www.meetup.com/fr-FR/owasp-france/

Objectif, philosophie et gestion du projet du projet

- Le projet à pour finalité de fournir des ressources au sujet des entêtes de sécurité HTTP.
  - https://owasp.org/www-project-secure-headers/
  - https://github.com/OWASP/www-project-secure-headers/
- Le projet fournit les type de ressources suivantes concernants les en têtes:
  - Documentation sur les entêtes avec des références.
  - Recommandations de configuration d'utilisations.
  - Outils et librairies par technologies permettant d'aider dans la mise en place.
  - Outils pour la validation des configurations.

    - Statistiques d'utilisation des entêtes HTTP de sécurité
- ➔ Le langue du projet est l'anglais afin d'être le plus accessible possible.
- ➔ Le projet est totalement libre et open source.
    - https://github.com/OWASP/www-project-secure-headers/?tab=readme-ov-file#licensing
- ➔ Le projet est totalement réalisé sur du temps personnel.
- ➔ Le projet est un projet officiel de l'OWASP et donc il est soumis à contrôles et des règles définies par le comité des projets de la fondation:
    - Projects
    - Project Committee
    - https://owasp.org/www-committee-project/#div-promotions
- ➔ Le projet est complètement hébergé sur GitHub.

Approche concernant le contenu du projet

- ➔ Tout le monde peut contribuer quelque soit son activité ou son expérience.
- ➔ Toutes les discussions sont ouvertes. On vise une transparence la plus totale possible:
    - https://github.com/oshp/oshp-tracking
- ➔ 📍 On ne prétend pas fournir des "**solutions parfaites car c'est un projet OWASP**", on propose des pistes/idées sur base de tests/POC réalisés ainsi que notre expérience au quotidien car Ricardo et moi travaillons tous deux dans le domaine de la sécurité des applications web. On va forcément se tromper et on prend en compte toute erreur/problème remontée endéans 2 jours maximum.
- ➔ A titre professionnel, j'utilise ce projet pour mes recommandations durant mes travaux offensifs ou défensifs sur des applications de type web. Cela me permet d'avoir une autre source de retour et de validation via les équipes de développements avec lesquelles je collabore.
- ➔ Toute information doit être sourcée.
- ➔ Chaque contribution est faite via une "Pull Request" qui est revue de façon ouverte par Ricardo et/ou moi. Par ailleurs, Ricardo valide aussi toute mes contributions car je passe par le même processus de validation/revue (on applique le principe de "*manger notre propre nourriture*").
    - https://en.wikipedia.org/wiki/Eating_your_own_dog_food

Mode de communication

- ➔ Pour informer à propos de la vie du projet, nous utiliser les réseaux sociaux X et LinkedIN avec un modéle de communication unique:
    - https://github.com/OWASP/www-project-secure-headers/?tab=readme-ov-file#social-media-communication
- ➔ Nous postons la communication avec nos comptes personnels pour des raisons de facilités.

Notes d’explications sur certains entête

Cross-Origin-Resource-Policy (CORP)

Il permet de définir une politique, qui sera aussi utilisée par *Cross-Origin-Embedder-Policy*, afin de spécifier si une requête (ex: script, image) venant d’une autre origine peut lire la ressource référencée par la requête.

Les valeurs possibles sont les suivantes:

- ➔ **same-site**: Seules les requêtes provenant du même site sont autorisées (cf le lien pour expliquer la notion de site).
- ➔ **same-origin**: Seules les requêtes provenant de la même origine sont autorisées (scheme + host + port).
- ➔ **cross-origin**: Pas de limitation.

Références utilisées:

- https://web.dev/articles/why-coop-coep#corp
- https://owasp.org/www-project-secure-headers/index.html#div-headers_cross-origin-resource-policy

Cross-Origin-Opener-Policy (COOP)

Il permet de spécifier si un document (une page web au sens du navigateur) partage ou pas son contexte de navigation avec un document ouvert venant d’une origine différente.

Les valeurs possibles sont les suivantes:

- ➔ **same-origin**: Le contexte de navigation est partagé uniquement si les deux documents sont issus de la même origine et que le document cible indique lui aussi cette valeur pour COOP.
- ➔ **same-origin-allow-popup**: Le contexte de navigation est partagé uniquement si les deux documents sont issus de la même origine et le document cible ne spécifie pas COOP ou bien avec la valeur **unsafe-none**.
- ➔ **unsafe-none**: Pas de limitation (valeur par défaut quand COOP est absent).

💡 Il est à noter que le document cible ouvert peut lui aussi définir cette entête afin d'empêcher le partage du contexte de navigation avec le document source.

Références utilisées:

- ➔ https://web.dev/articles/why-coop-coep#coop

➔ https://owasp.org/www-project-secure-headers/index.html#div-headers_cross-origin-opener-policy

Cross-Origin-Embedder-Policy (COEP)

Il permet de spécifier que le document courant (sur lequel COEP s'applique) ne peut charger des ressources d'une origine différente que si elle l'autorise via l'entête CORP (ou CORS).

Les valeurs possibles sont les suivantes:

➔ **require-corp**: Le document ne peut charger que des ressources issues de la même origine ou bien d'une origine différente si elle l'autorise via une entête CORP.
➔ **unsafe-none**: Pas de limitation (valeur par défaut quand COEP est absent).

💡 Pour bien comprendre CORP vs COEP:

- CORP s'applique du côté de la ressource chargée (propriétaire de la ressource).
- COEP s'applique du côté du "chargeur" de la ressource (consommateur de la ressource).

Références utilisées:

➔ https://web.dev/articles/why-coop-coep#coep
➔ https://owasp.org/www-project-secure-headers/index.html#div-headers_cross-origin-embedder-policy

Ressources additionnelles

➔ https://owasp.org/www-project-secure-headers/
➔ https://www.youtube.com/watch?v=dvx-Gr9DO_8
➔ https://caniuse.com/
➔ https://github.com/oshp/oshp-tracking/issues
➔ https://github.com/oshp/oshp-tracking/discussions
➔ https://github.com/oshp/oshp-stats/blob/main/stats.md
➔ https://github.com/oshp/oshp-validator
➔ https://github.com/oshp/oshp-validator?tab=readme-ov-file#tests-suite-mock-service