Rapport de proposition de création d' une IHM pour l' outil TSUNAMI

HACKTOBERFEST 2022

# Proposition Création D'IHM

Pour l'outil de scan de vulnérabilités **TSUNAMI** de Google.

**Conception et implémentation**

D'une interface web pour l'outil TSUNAMI développé par google.

cyna

SUP DE VINCI
INGÉNIERIE DES SYSTÈMES D'INFORMATION

Présenté par :

**KHADY GUEYE**

HACKTOBERFEST 2022

# Proposition de création D'IHM

## Pour l'outil de scan de vulnérabilités TSUNAMI de Google.

Conception et implémentation d'une interface web pour l'outil TSUNAMI développé par google.

Présenté par :

Khady Gueye

# Résumé du Document

Google a développé en interne son propre scanner de vulnérabilité pour les réseaux comme le sien, à savoir un réseau composé de milliers, pour ne pas dire de millions, d'équipements connectés à Internet. Nommé Tsunami, il s'agit de l'outil utilisé en interne par Google et celui-ci devient open source : il est disponible sur GitHub. Cependant, cet outil n'est disponible qu'en ligne de commande.

C'est dans ce contexte que ce rapport expose le processus mis en place pour concevoir et implémenter une interface graphique web pour rendre plus accessible l'utilisation de cet outil.

Le projet s'inscrit dans le cadre du HackToBerFest 2022 organisé par la cyna en collaboration avec SUP DE VINCI paris.

Le rapport développe quatre axes principaux :

- Description et présentation du projet
- Conception et design de l'application
- Implémentation et technologies
- Perspectives

La phase de clarification des spécifications et de définition du périmètre du projet a conduit au choix de technologies comme Vue.js/Sass pour le front end, node.js et Docker pour l'API, git pour le contrôle de version, Netlify pour l'hébergement front end/déploiement automatique et Heroku pour l'hébergement backend et API.

Une version d'essai de l'application web a été réalisée dans le but de fournir un prototype d'utilisation minimale et donc d'ouvrir les perspectives à des améliorations futures.

# Sommaire

# Description du projet

Le but de ce projet est de créer une interface visant à faciliter l'utilisation de l'outil en ligne de commande tsunami, développé par Google.

## Spécifications fonctionnelles et contraintes

L'utilisateur doit pouvoir scanner un site avec l'aide de son url, ainsi que voir/modifier la liste des plugins proposés par google.

L'outil doit faire abstraction de toute la partie architecture docker et ligne de commande et faire un focus sur l'utilisation.

## Cadre légal du projet

```
Copyright 2019 Google Inc.

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.
```

Tsunami est un projet open source de google, distribué sous licence. Chaque fichier source doit contenir la licence, comme ils l'ont précisé dans la repository Github du projet tsunami-scanner-plugins.

"Every file containing source code must include copyright and license information. This includes any JS/CSS files that you might be serving out to browsers. (This is to help well-intentioned people avoid accidental copying that doesn't comply with the license.)"

# Conception et design

## Architecture de Tsunami

L'outil tsunami est actuellement hébergé dans un conteneur docker. C'est cependant une version 'pré-alpha' et nous pouvons avoir droit à des changements majeurs comme Google nous le précise.

### Current Status

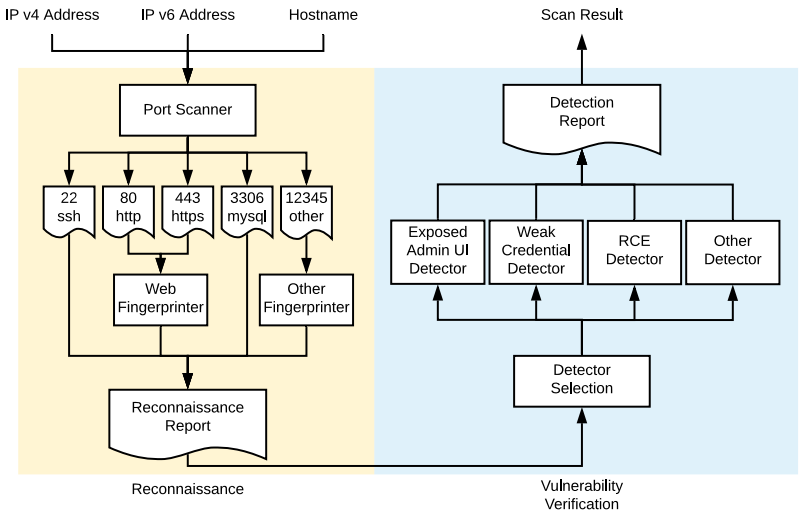
- Currently Tsunami is in 'pre-alpha' release for developer preview.
- Tsunami project is currently under active development. Do expect major API changes in the future.

À ce jour, Tsunami suit un processus en 2 étapes codé en dur lors de l'analyse d'un réseau :

- Reconnaissance : dans un premier temps, Tsunami identifie les ports ouverts, puis enregistre les protocoles, services et autres logiciels exécutés sur l'hôte cible via un ensemble de plug-ins d'empreintes digitales. Tsunami s'appuie sur des outils existants tels que nmap pour certaines de ces tâches.

- Vérification des vulnérabilités : sur la base des informations recueillies à l'étape 1, Tsunami sélectionne tous les plug-ins de vérification des vulnérabilités correspondant aux services identifiés et les exécute afin de vérifier les vulnérabilités sans faux positifs.

Plus d'informations dans la documentation : [https://github.com/google/tsunami-security-scanner/blob/master/docs/orchestration.md](https://github.com/google/tsunami-security-scanner/blob/master/docs/orchestration.md)

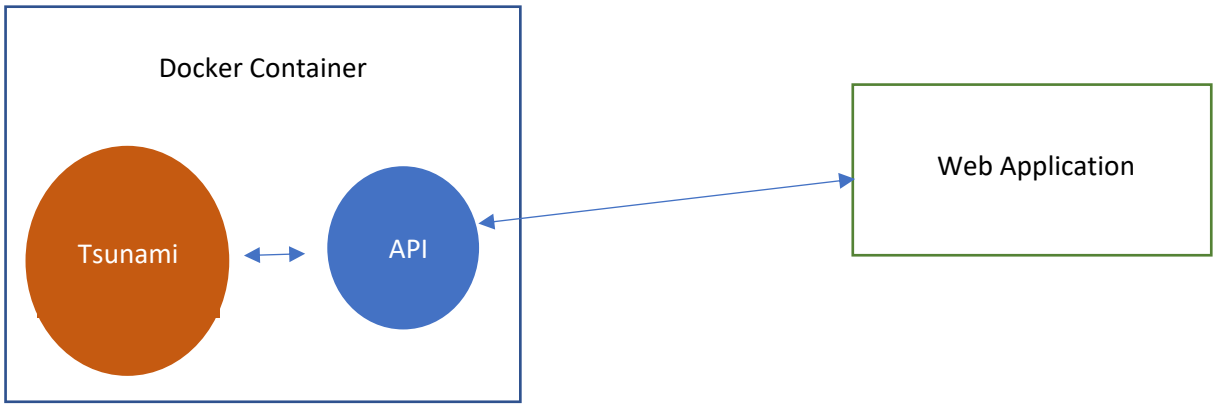


# Notre architecture

Nous conçu une API, 'dockerisée' à son tour se basant sur le container docker de Tsunami mis à disposition. Ce container est chargé de vérifier l'intégrité de toutes les dépendances nécessaires à l'exécution de tsunami et lancer le serveur d'écoute de requêtes.

Le fait de mettre notre API dans un container docker nous permet d'intégrer facilement toutes les dépendances d'environnement nécessaires à l'exécution du projet et cela, peu importe l'hébergeur et ses restrictions.

Nous avons donc ensuite créé une interface web qui va récupérer les données renvoyées par l'API et les afficher à l'écran.

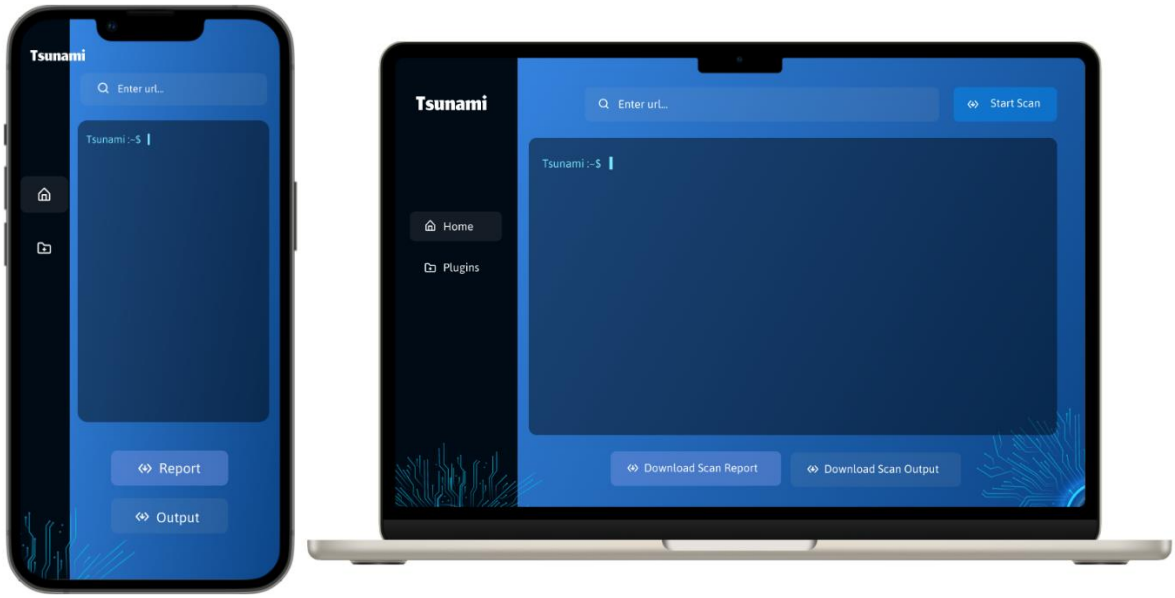


## Maquettes Graphiques

L'idée qu'on se fait d'un projet n'est pas toujours claire, raison pour laquelle il est important de réaliser des maquettes, cela permet aussi de faciliter la modélisation.

Nous avons donc, à l'aide de l'outil en ligne **Figma**, réalisé des maquettes réalistes pour notre plateforme.

Voici donc la maquette responsive réalisée pour la page de scan de la plateforme.




# Implémentation et tests

## Présentation des Technologies Utilisées

### Figma :

Figma est un éditeur de graphiques vectoriels et un outil de prototypage collaboratif. Il est principalement basé sur le web, Son utilisation est gratuite et intuitive, ce qui nous a permis de réaliser toutes les maquettes de notre application.

## Vue.Js :


Vue.JS est un Framework JavaScript open-source utilisé pour construire des interfaces utilisateur et des applications web monopages. Il utilise un système de Routing et d'affichage dynamique qui garantit une rapidité d'exécution mais aussi de développement.

Le choix s'est porté sur cette technologie à cause de sa facilité de prise en main mais aussi de son efficacité.

## NodeJS :


Node.js est un environnement d'exécution JavaScript asynchrone piloté par les événements. Il semblait être le choix idéal pour mettre en place un serveur rapidement avec express et écouter les requêtes. Son avantage est qu'il est largement scalable et peut répondre aux besoins de ce projet.

## Git, Netlify, Heroku :




Pour la sauvegarde du projet ainsi que le versionning, nous avons privilégié le célèbre outil **Git**, via l'application Github.

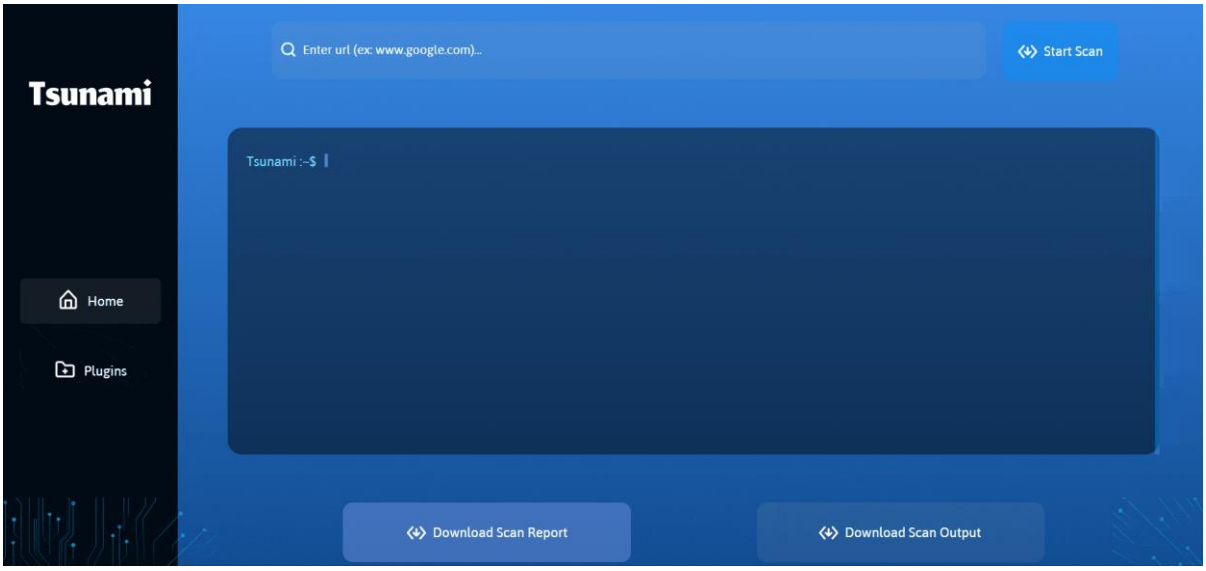
**Netlify** propose des solutions d'hebergement cloud sans serveur et gratuit pour des sites statiques. Il s'intègre facilement avec github pour un déploiement continu et automatisé.

Nous avons dans notre cas paramétré netlify pour qu'il réalise un deploiement à chaque push sur la branche master de notre projet sur gitHub.

Pour l'hebergement de notre conteneur back, nous avons choisi le service en ligne **Heroku** qui propose un large choix de services gratuits et un déploiement continu paramétrable avec git. Il permet tres facilement d'heberger son site via sa ligne de commande intégrée que ce soit avec des fichiers classiques ou un container docker et d'avoir un domaine gratuit et fonctionnel.

# Résultats et perspectives

Cf : https://tsunami-front.netlify.app/




Ce prototype a été réalisé dans le but de démontrer le potentiel de l'application web. Il est bien sûr prêt pour plusieurs améliorations notamment :

- **L'utilisation de Firebase pour l'affichage de données en temps** réel : Pour certaines analyses de grande envergure, le chargement peut être long, le transfert des données entre le back et le front peut s'avérer lent ou incomplet et peux même saturer les buffers de la connexion entre les deux, même avec l'utilisation du Stream de Node.js. Il serait donc plus judicieux de prévoir un moyen d'envoyer les données en live, au fur et à mesure de l'analyse, sans saturer le serveur.

- **L'optimisation du mode de réponse de l'API et sécurisation :** il serait intéressant de réfléchir à la sécurisation de l'API en ajoutant des autorisations spécifiques afin que ce logiciel ne soit pas utilisé dans un but malveillant.

- **Stockage et historique d'analyses :** Un système de monitoring des analyses effectuées par l'utilisateur connecté pourrait permettre d'accéder plus facilement à ses rapports de scan en fonction de la date, l'année… Il serait même envisageable de pouvoir comparer les résultats entre deux dates et de mener des enquêtes plus approfondies.

- **Amélioration du design et des animations d'affichage et de chargement**

- **Compléter le système de management de plugins Tsunami**

- **Mise en page et design du rapport de scan ainsi que le stdout.**

- **Et encore beaucoup d'autres fonctionnalités pour aller toujours plus loin !**

# Conclusion

Tout au long de la conception, nous avons essayé de tenir compte des besoins utilisateur et de faire abstraction de toute la partie installation, mise en place, etc. qui pourraient ralentir l'utilisation de Tsunami, un outil à fort potentiel

En passant par toutes les étapes essentielles de développement ; de la conception au déploiement du premier prototype et en mettant en œuvre des techniques modernes de développement, nous avons réalisé cette 1ere version de l'application web, qui, pourrait être une première pierre à cet édifice.

Des recherches ont été effectués sur l'outil tsunami pour en cerner le potentiel. Ce projet ouvre des perspectives d'amélioration future et nous espérons qu'il pourra servir le monde de la cybersécurité.