

B1.2 Répondre aux incidents et aux demandes d'assistance et d'évolution

SOMMAIRE :

<i>Création d'un fichier permettant de gérer chaque paramètre d'un PDF généré automatiquement via code.....</i>	<i>2</i>
<i>Création d'un Bot Discord nommé RIMU SCAN.....</i>	<i>3</i>
<i>Mise en place d'un Site permettant de générer des certificats SSL.....</i>	<i>5</i>
<i>Création du site « Mines ».....</i>	<i>7</i>
<i>Test d'intrusion sur borne Wi-Fi.....</i>	<i>9</i>
<i>Mise en place d'un GLPI.....</i>	<i>11</i>

Création d'un fichier permettant de gérer chaque paramètre d'un PDF généré automatiquement via code

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : Ce projet m'a permis de collecter les besoins spécifiques des utilisateurs en termes de paramètres à gérer pour le PDF, de suivre les modifications demandées et de les orienter vers les sections appropriées du code.
- Traiter des demandes concernant les applications : La gestion des paramètres du PDF a nécessité de répondre aux demandes d'évolution et de corrections concernant l'application utilisée pour générer les PDFs.

Résumé du Projet :

Le but de ce projet était de faire en sorte que tous les paramètres soient modifiables facilement dans le code sans avoir à chercher un peu partout, permettant un gain de temps significatif lors des modifications. Cela a été réalisé en PHP avec le CMS Dolibarr. Chaque paramètre modifiable était centralisé en haut du fichier de code, ce qui a permis de simplifier les modifications et de réduire les erreurs.

Pour organiser le code et les fichiers de manière efficace, des sauvegardes régulières du code ont été effectuées, permettant de réaliser de nombreux tests sans risque de perte de données en cas de fautes de frappe.

Illustrations :

```
//-----//
//-----Géner Chaque Paramètres-----//
//-----//

// FAC = Facture / BL = Bon de livraison OU fiche expédition /
$typeFichier = "FAC";

$this->header($pdf, $object, $outputlangs, $typeFichier);

$noteHeight = $this->_Note($pdf, $object, $outputlangs, $CoordonneeHautNote = 80);

$this->body($pdf, $object, $outputlangs, $typeFichier, $CoordonneeHautDuTableau = 80, $TailleTableau = 255, $DistanceEntreNote = 4, $noteHeight, $hauteurTableauAutrePage = 35);

$this->_tableau_tot($pdf, $object, 0, $bottomlasttab, $outputlangs, $tableautotHauteur = 258.5);
```

Fig. 1 - Interface de configuration des paramètres PDF

- *Description* : Une capture d'écran montrant l'interface de configuration des paramètres PDF dans le code.

Création d'un Bot Discord nommé RIMU SCAN

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : Le bot a été développé pour répondre aux besoins des traducteurs de manga, en collectant les demandes de mises à jour et en les orientant vers les fonctionnalités appropriées du bot.
- Traiter des demandes concernant les applications : Le développement du bot a inclus la gestion des demandes de nouvelles fonctionnalités et la correction de bugs signalés par les utilisateurs.

Résumé du Projet :

Le projet "RIMU BOT" vise principalement à simplifier la tâche des traducteurs de manga en ligne en leur fournissant les dernières sorties de manière automatisée sur l'application Discord, évitant ainsi la nécessité de consulter plusieurs sites. Le bot effectue cette tâche en retournant les dernières sorties avec un lien de redirection, indiquant également depuis quand elles sont disponibles. De plus, le bot se met à jour toutes les heures pour garantir des informations toujours à jour.

Développé en utilisant Node JS, JS et JSON, ce bot interagit avec les utilisateurs de plusieurs manières. Les utilisateurs peuvent invoquer le bot en utilisant des commandes spécifiques telles que `"/suivi [nom du manga]"` pour savoir s'il y a une nouvelle sortie pour ce manga, ou `"/WatchList"` pour recevoir des mises à jour automatiques dans le salon où la commande a été effectuée. Le bot est hébergé sur Lordhosting, et utilise Axios pour rechercher les dernières sorties de manga sur les sites internet.

Illustrations :

`/bloodhounds`



/WatchList :



Une partie des commandes disponible :



Fig. 3 - Interface de commande du bot Discord RIMU SCAN

- *Description : Une capture d'écran montrant les commandes du bot sur Discord.*

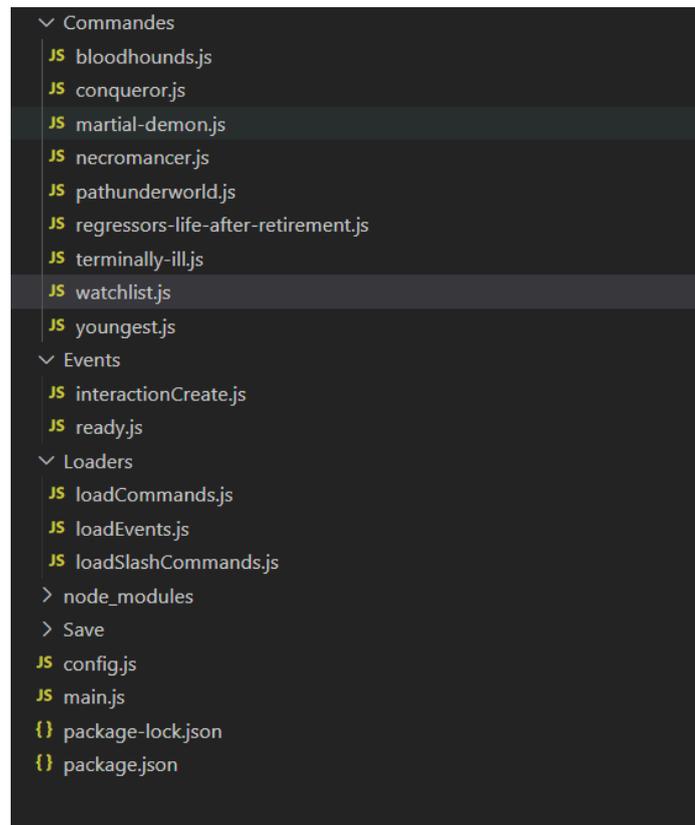


Fig. 4 - Schéma de l'architecture du bot RIMU SCAN

- *Description : Un schéma illustrant l'architecture technique du bot.*

Mise en place d'un Site permettant de générer des certificats SSL

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : Le site permet de recueillir les demandes de génération de certificats SSL et de suivre leur traitement.
- Traiter des demandes concernant les applications : Le développement du site a impliqué la gestion des demandes d'accès, la génération des certificats SSL et la gestion des sauvegardes de manière sécurisée.

Résumé du Projet :

L'objectif principal du site était de créer un site permettant aux personnes autorisées de créer un certificat SSL facilement en répondant simplement à un questionnaire. Il fallait un administrateur pouvant accorder l'accès au site aux personnes de l'entreprise ayant les autorisations nécessaires. De plus, les sauvegardes des certificats SSL ne devaient pas être stockées sur la même machine que le site afin d'assurer la sécurité des données.

Le site a été développé en utilisant principalement PHP, JavaScript, HTML, et des commandes Linux. La sécurité des données et des communications a été assurée en transférant les données via un réseau fermé et sécurisé, avec des clés aléatoires générées pour chaque téléchargement de fichier, rendant presque impossible le téléchargement de fichiers non autorisés.

Illustrations :

The image shows a screenshot of an administrative interface for an SSL certificate site. The interface is divided into three main sections, each with a title and a list of user entries. The background is a solid orange color.

- Utilisateur Valider:** This section contains two entries. Each entry consists of a redacted name field and a 'Refuser' button.
- Utilisateur En attente:** This section contains six entries. Each entry consists of a redacted name field and two buttons: 'Valider' and 'Refuser'.
- Utilisateur Refusé:** This section is currently empty.

Fig. 5 - Interface admin du site certificat SSL

- *Description : Une capture d'écran montrant l'interface admin.*



Fig. 6 - Interface d'oublie de mot de passe

- *Description : Un Interface qui permet à l'utilisateur de modifier son mot de passe oublié en recevant un mail*

Création du site « Mines »

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : Le site a été conçu pour répondre aux besoins de gestion et de suivi des équipements des mines, collectant les données et orientant les demandes vers les sections appropriées du site.
- Traiter des demandes concernant les applications : Le développement du site a inclus la gestion des fonctionnalités et des améliorations demandées par les utilisateurs.

Résumé du Projet :

Le Site de Gestion Mines est une plateforme complète permettant la gestion détaillée de plusieurs mines et de leurs équipements associés. Les fonctionnalités incluent la visualisation des mines avec leurs équipements et leur localisation, la

possibilité d'échanger des équipements et d'ajouter directement de nouvelles données à la base de données depuis le site. La page d'accueil propose plusieurs écrans présentant des statistiques essentielles telles que les dernières modifications et les nouvelles mines ajoutées.

Les données sont collectées et traitées à l'aide d'une base de données SQL. Le site utilise PHP pour son développement et son fonctionnement, avec des routes API définies pour gérer les différentes fonctionnalités. L'utilisation de Composer a simplifié la gestion des dépendances et des packages PHP.

Documents produits :

- Schéma de l'architecture du site

Illustrations :

```
1 <?php
2
3 use App\Http\Controllers\Api\ApiAuthController;
4 use App\Http\Controllers\Api\ConcessionController;
5 use App\Http\Controllers\Api\MineController;
6 use Illuminate\Http\Request;
7 use Illuminate\Support\Facades\Route;
8 use App\Http\Controllers\ConcessionController2;
9
10 /*
11 |-----
12 | API Routes
13 |-----
14 |
15 | Here is where you can register API routes for your application. These
16 | routes are loaded by the RouteServiceProvider and all of them will
17 | be assigned to the "api" middleware group. Make something great!
18 |
19 |*/
20
21 Route::middleware('auth:sanctum')->get('/user', function (Request $request) {
22     return $request->user();
23 });
24
25 Route::get('/mines/listing', [MineController::class, 'getAllMines']);
26 //Route::post('/mines/add', [mineController::class, 'store']);
27
28
29
30 Route::get('/concessions', [ConcessionController::class, 'getAllConcessions']);
31 Route::get('/concession/{id}/mines', [ConcessionController::class, 'getMinesInConcession']);
32
33 Route::get('/concession/{id}', [ConcessionController2::class, 'afficheConcession']);
34
35
```

Fig. 7 - Schéma de l'architecture du site "Mines"

- Description : Un schéma représentant la structure et les différentes sections du site "Mines".



Fig. 8 - Capture d'écran du tableau de bord avec les statistiques des mines

- *Description* : Une capture d'écran montrant les statistiques essentielles telles que les dernières modifications et les nouvelles mines ajoutées.

Test d'intrusion sur borne Wi-Fi

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : Le test d'intrusion a permis de collecter les vulnérabilités identifiées et de suivre leur correction.
- Traiter des demandes concernant les services réseau et système, applicatifs : Les vulnérabilités identifiées ont été traitées pour améliorer la sécurité du réseau et des systèmes.

Résumé du Projet :

Le principal objectif de notre test d'intrusion sur la borne Wi-Fi était d'acquérir une compréhension approfondie des pratiques de hacking, telles que les attaques par déni de service (DDoS) et d'autres techniques de piratage. Cela nous a permis de mieux comprendre comment se défendre efficacement contre ces attaques et de renforcer la sécurité de notre infrastructure réseau.

Pour réaliser ce test d'intrusion, nous avons utilisé une combinaison d'outils et de méthodologies, y compris des machines virtuelles (VM), Wireshark pour l'analyse de trafic réseau, ainsi que d'autres outils spécialisés en sécurité informatique. Un environnement de test contrôlé a été préparé, comprenant une borne Wi-Fi dédiée et l'installation d'un site web fictif conçu pour simuler des attaques réalistes.

Les vulnérabilités principales identifiées lors de ce test comprenaient des failles d'injection SQL, des attaques de type "man in the middle", ainsi que des vulnérabilités dues à l'absence de mises à jour critiques. Pour remédier à ces vulnérabilités, un ensemble de recommandations a été élaboré, centré sur l'importance de maintenir les systèmes à jour, l'utilisation de protocoles de chiffrement forts, la formation continue des équipes IT en matière de sécurité, et l'adoption de meilleures pratiques de sécurité réseau.

Documents produits :

- Rapport d'analyse des vulnérabilités
- Guide de mise en œuvre des recommandations de sécurité

Illustrations :

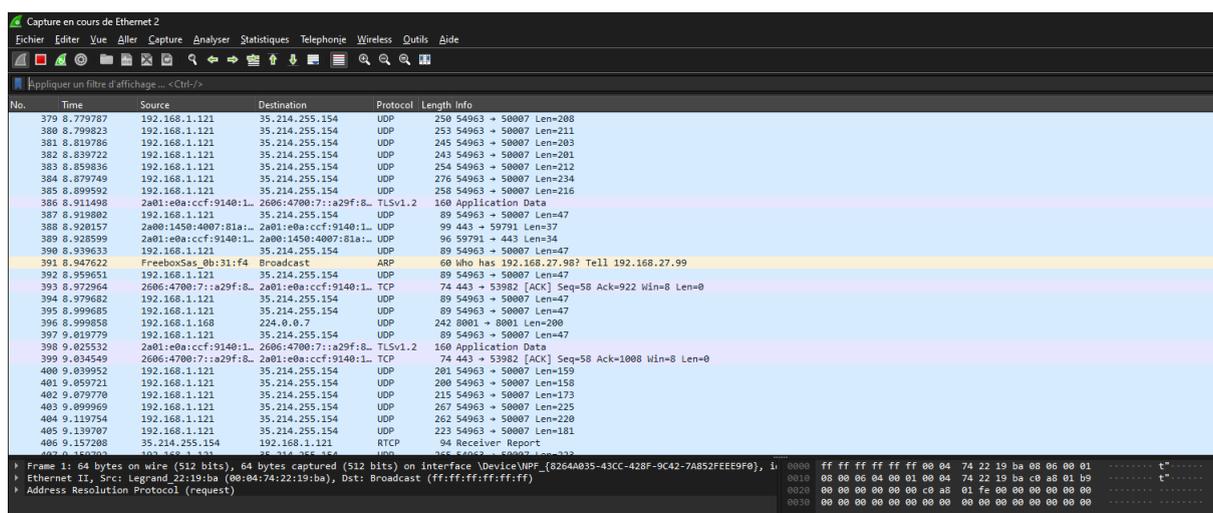


Fig. 9 - Analyse de trafic réseau avec Wireshark

- *Description* : Une capture d'écran de Wireshark montrant l'analyse de trafic réseau pendant le test d'intrusion.

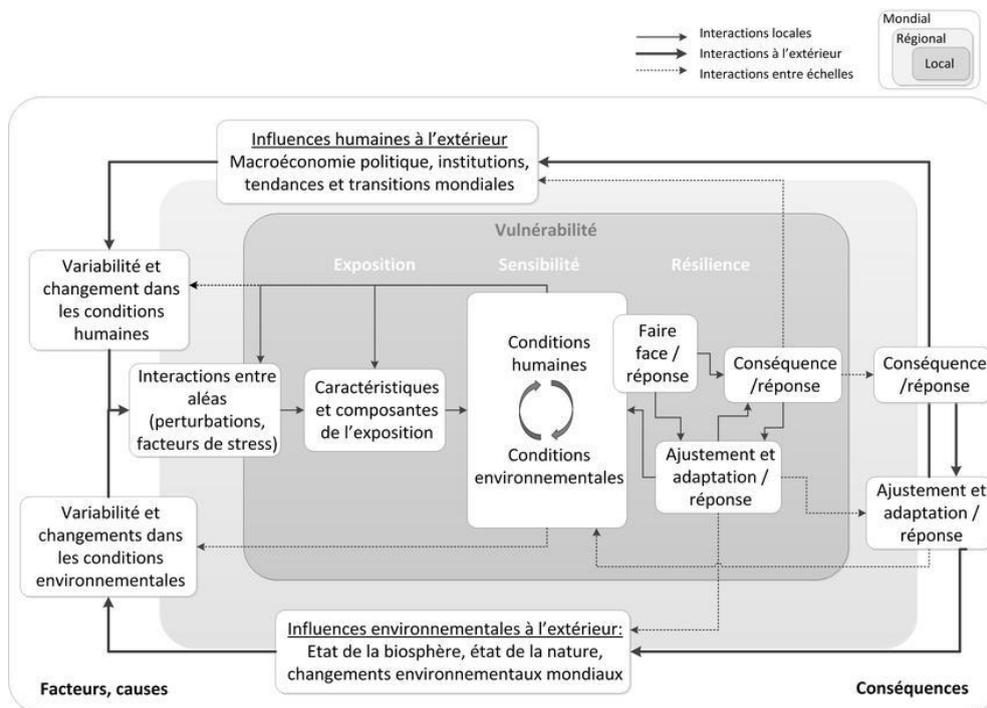


Fig. 10 - Diagramme des vulnérabilités identifiées

- *Description* : Un diagramme illustrant les vulnérabilités.

Mise en place d'un GLPI

Sous-compétences concernées :

- Collecter, suivre et orienter des demandes : GLPI a permis de centraliser la collecte et le suivi des demandes d'assistance, facilitant leur orientation vers les services appropriés.
- Traiter des demandes concernant les services réseau et système, applicatifs : Le système de ticketing de GLPI a amélioré le traitement des demandes concernant les services réseau, système et applicatifs.

Résumé du Projet :

Le projet de mise en œuvre de GLPI a été initié avec l'objectif de transformer nos méthodes de gestion de parc informatique et de support utilisateur. Par son déploiement, nous avons atteint plusieurs buts significatifs :

- Efficacité accrue dans le recensement des ressources : L'utilisation de GLPI a révolutionné notre capacité à inventorier et à suivre l'état de nos équipements informatiques. La distinction claire entre les équipements en stock et ceux en service nous a permis d'optimiser l'utilisation de nos ressources.
- Amélioration de la gestion du parc informatique : Avec une vue d'ensemble plus précise sur l'état de notre parc, la prise de décisions concernant le remplacement ou la mise à jour des équipements est devenue plus rapide et plus éclairée, contribuant ainsi à une meilleure allocation des ressources.

Documents produits :

- Guide d'installation de GLPI sur une machine virtuelle
- Guide d'utilisation basique de GLPI

Illustrations :



Fig. 11 - Interface de gestion des équipements dans GLPI

- *Description* : Une capture d'écran montrant la gestion des équipements dans GLPI, avec une vue d'ensemble des ressources disponibles et en cours d'utilisation.

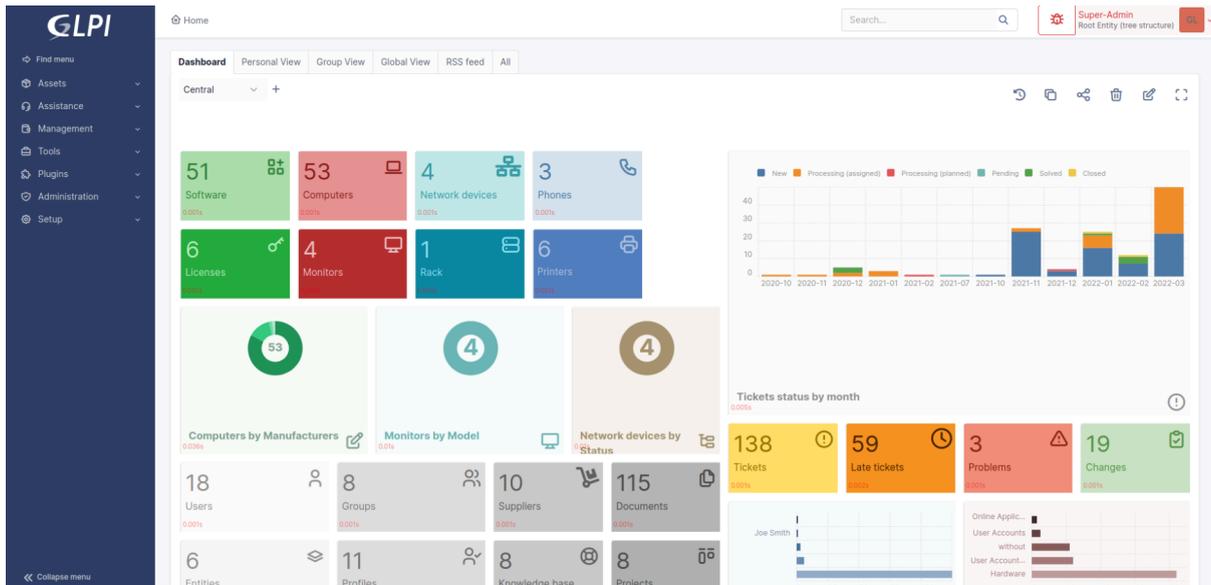


Fig. 12 - Tableau de bord de GLPI

- *Description : Une capture d'écran du tableau de bord de GLPI, affichant les tickets ouverts, les incidents récents et les statistiques de gestion du parc informatique.*