**Zoey - Documentation - Hébergement**
**Groupe 10**


**Marin Bouanchaud**

**Clémentine Gilama**

**Matthis Rousselle**

**Lucien Boisseau-Sable**

**Amélie Rubiales**

# Zoey - documentation : hébergement

1. Guide de migration

    a. Principes préliminaires

Ce guide est un document interne à Zoey. Il est indispensable de s'y référer et de suivre ses recommandations lors de l'installation du service : migration sur un nouveau serveur, mise à jour, installation du service sur un serveur de test ou en version bêta, réinstallation suite à une panne du serveur.

La rédaction de ce guide a été finalisée et validée le **17 mars 2022**. Il est impératif de le mettre à jour aussi souvent que nécessaire, dès qu'une information n'est plus valide, sans jamais y partager d'informations confidentielles (données personnelles, identifiants de connexion, clés d'API ou de chiffrement). La documentation complète de la plateforme Zoey est un facteur majeur de son bon fonctionnement.

La plateforme Zoey doit être hébergée avec
- un serveur web Apache
    - avec support PHP (version 8.1)
- un serveur MySQL (version 5.7)

Ce guide présentera des recommandations pour la migration de Zoey en utilisant le serveur SFTP et l'accès SSH fournis par notre fournisseur d'hébergement actuel (Ionos). L'accès à la base de données peut se faire via l'interface PhpMyAdmin fournie par l'hébergeur, ou par un accès en lignes de commande.

Pour garantir un bon fonctionnement de la migration, il est nécessaire de respecter l'ordre des étapes suivantes.

    b. Définitions

Racine du serveur web : dossier généralement nommé "public_html", "htdocs", ou "/var/www/html" utilisé par le serveur Apache pour servir les pages webs et les fichiers du site. Son emplacement peut être défini dans le fichier de configuration de Apache.

Code source : ensemble des fichiers permettant de faire fonctionner le site (notamment des fichiers PHP, JS, HTML, CSS, JSON, et images bitmap et vectorielles non dépendantes des utilisateurs).

Données utilisateurs et base de données : ensemble des données de la base de données, stockées sur le serveur MySQL, variables en fonction notamment des utilisateurs et de leurs actions sur la plateforme.

Images uploadées : images téléversées sur la plateforme (via les formulaires du site côté back-office ou directement en SFTP pour les administrateurs, ainsi que via les formulaires du site par les utilisateurs connectés), sauvegardées sur le serveur web.

c. Récupération du code source

Le code source de Zoey (en version de développement et version de production) est hébergé sur GitHub et accessible uniquement par l'équipe de développeurs internes.
Il existe également des sauvegardes de la dernière version de production (voir partie "2. Plan de reprise d'activité").

Dans le cas d'une installation sur le serveur public de Zoey, il faut installer le code source en **version de production**. Une version de production de Zoey doit avoir été validée par une période de tests dans un environnement privé mais similaire à celui de la production (serveur de tests et de bêta).

Note : il est nécessaire de mettre à jour le fichier **model/PDO.php** avec les informations correspondantes à la nouvelle base de données, voir partie h. Installation de la base de données.

d. Interruption du service pendant la migration

Avant de réaliser les étapes suivantes, il est nécessaire de restreindre l'accès des utilisateurs au service dans son état actuel. En effet, toute donnée utilisateur enregistrée après les deux étapes suivantes ne pourrait pas être retrouvée dans la nouvelle version de Zoey, après réinstallation.

Il convient donc de placer dans le répertoire racine du serveur une nouvelle page index.php "page en travaux" qui indique aux utilisateurs tentant d'accéder au service que Zoey est momentanément indisponible.
Pour favoriser une meilleure expérience utilisateur, il faut s'assurer que cette période d'indisponibilité du service est réduite au maximum, et qu'elle est effectuée dans la mesure du possible en heures creuses : les pics de connexion se situent en fin de journée (entre 16h et 21h), mais si la migration est réalisée sur certaines plages horaires et dans la nuit en heure française, le nombre d'utilisateurs impacté peut être minimisé.

Une fois cette page installée, il convient d'interdire l'accès au reste du serveur via une règle de configuration Apache écrite dans un fichier ".htaccess" dans le répertoire "public_html" du serveur.

e. Récupération des données utilisateur sur le serveur de base de données

Les données utilisateur sont stockées dans la base de données (sauf les images uploadées sur la plateforme, voir partie suivante).

Il faut télécharger l'ensemble de ces données à l'aide de la commande "mysqldump".

Dans le cas d'une installation d'une nouvelle version de Zoey avec un changement de structure de base de données, il est nécessaire de ne télécharger que les données de la table.

Cette opération peut être réalisée à l'aide de la commande suivante (en remplaçant les champs entre-crochets par les informations appropriées) :

mysqldump -u [user] -p[pass] --no-create-info mydb > zoey_save_[date].sql

Si seules les données ont été téléchargées, il sera nécessaire d'installer séparément la structure de la base de données. Celle-ci est livrée dans un fichier SQL avec chaque sauvegarde de la version de production de Zoey (voir partie "2. Plan de reprise d'activité" pour réaliser ces sauvegardes ou télécharger une nouvelle structure de la base de données).

f. Récupération des images uploadés sur le serveur de fichiers

Les images uploadées par les utilisateurs et les administrateurs de Zoey sont stockées dans le répertoire "public/images/upload", pour toutes les images ajoutées via des formulaires. Les images utilisées pour le blog sont stockées dans le répertoire "public/images/blog".

Il est nécessaire de récupérer le contenu de ces deux dossiers pour pouvoir les migrer sur un nouveau serveur.

Dans le cas où les images ne doivent pas être migrées (par exemple : installation d'une version de tests), il est tout de même nécessaire de conserver le fichier "DefaultProfile.png" (sauvegardé par défaut dans le code source) dans le répertoire "public/images/upload".

g. Installation des codes sources

Note : avant une installation en version de production, les codes sources migrés depuis la version de production doivent être minifiés.

Les codes sources doivent être installés dans le répertoire "public_html".

L'architecture doit correspondre au schéma suivant :

**/public_html**
|
|___/controller
|___/model
|___/private_crypt
|___/public
    |___/css
    |___/js
    |___/template
    |___/images
        |___/badges
        |___/blog
        |___/icons
        |___/presentation
        |___/upload

|___/services
|___/vendor
        |___… (répertoires et fichiers nécessaires à jQuery, google API, tinify API)
|___/view
|
|       (index, fichier renvoyé par défault)
|___index.php
|
|       (fichiers nécessaires à la Progressive Web App et aux navigateurs)
|___offline.html
|___script.js
|___service-worker.js
|___manifest.json
|___logo192.png
|___logo512.png
|___favicon.ico
|
|       (fichiers de configuration du serveur et autres)
|___.htaccess
|___php.ini
|___robots.txt

Note : il convient de vérifier que le répertoire “public/images/upload” est accessible en droits d’écriture par le serveur PHP.

h. Installation de la base de données

Sur le serveur MySQL, il faut créer une nouvelle base de données et y importer le fichier sql qui contient le dump.

Si la structure a été exportée séparément des données, il faut d’abord importer la structure, puis importer les données. Pour régler une éventuelle erreur liée à la correspondance des clés étrangères, il est possible soit de “désactiver la vérification des clés étrangères” (dans PhpMyAdmin), soit d’importer table par table les données, dans l’ordre des dépendances des relations. Cette erreur est généralement liée à l’ordre des tables exportées dans le fichier SQL du dump.

i. Tests de fonctionnement et réouverture de la plateforme

Avant de réouvrir la plateforme aux utilisateurs, il faut réaliser des tests sur le site en production, pour confirmer que la migration s’est bien déroulée. Il faut notamment tester les points sensibles suivants :

- connexion sur le site avec Google
- connexion sur le site traditionnelle (sans Google)
- upload d’une photo (publication d’un post, envoi d’un message avec une photo, changement de photo de profil)
- tout autre fonctionnalité nouvelle qui vient d’être installée ou qui a récemment présenté des dysfonctionnements.

En cas de comportement anormal détecté pendant cette phase de tests, il faut impérativement stopper la procédure de migration et ne pas réouvrir le site au public tant que la source du problème n'est pas identifiée et réglée. Pour éviter des interruptions trop longues du service, si le problème persiste et demande des investigations plus poussées pour effectuer la levée de doute, il est possible de réinstaller l'ancienne version du site (code source et base de données) en attendant un correctif. Ensuite, se référer aux parties "2. Plan de reprise d'activité, a. Problèmes les plus fréquents et b. Détection de problèmes et débugage".

Pour réouvrir la plateforme aux utilisateurs, il faut enlever le index.php "page en travaux" provisoire, et enlever la règle Apache provisoire dans le fichier "htaccess".

2. Plan de reprise d'activité :
    a. Problèmes les plus fréquents

- erreur SQL
    - vérifier que le fichier **model/PDO.php** contient les bonnes informations

- problème d'affichage
    - vider le cache du navigateur et recharger la page

- image non trouvée par le navigateur
    - vérifier que le site est bien connecté à la bonne base de données (**model/PDO.php**) et qu'elle a été mise à jour (les noms d'images enregistrés en base de données doivent correspondre aux noms des images stockés dans le répertoire "public/images/upload" (ou "public/images/blog" pour le blog)

- erreur lors de l'upload d'une image
    - vérifier que le serveur a les droits d'écriture sur le dossier "public/images/upload"

    b. Sauvegardes bihebdomadaires

Il faut réaliser une sauvegarde des données utilisateurs de Zoey 2 fois par semaine : dump de la base de données et archivage des photos du dossier upload (voir partie 1. Guide de migration).

Ces sauvegardes doivent être conservées au moins en triple, sur un serveur cloud (différent du serveur de production), et sur deux disques durs stockés à des endroits différents. Les archives doivent être conservées pendant au moins deux semaines afin de pouvoir facilement revenir à une version antérieure en cas de panne du serveur principal de production.

    c. Surveillance et procédures d'urgence

Une surveillance du site doit se faire quotidiennement par l'équipe, et plus profondément par l'équipe de développement à des moments stratégiques : lancement d'un événement qui génère des pics de connexion, mise à jour, reprise des opérations suite à une maintenance planifiée ou non.

Ces surveillances ponctuelles plus poussées doivent intervenir 1 fois par semaine minimum et vérifier les indicateurs fournis par nos outils de monitoring (voir partie d. Processus de surveillance manuels ou automatisés).

De plus, cette surveillance est accompagnée d'une veille des réseaux sociaux et des canaux de contacts (adresse mail : contact@zoey-app.fr), qui seront privilégiés par les utilisateurs pour nous contacter en cas de problème.

La page d'erreur de Zoey (qui apparaît lorsqu'une exception PHP est déclenchée), invite les utilisateurs à décrire leur problème par mail.

### d. Outils de monitorings manuels ou automatisés

La surveillance de Zoey doit s'appuyer sur les outils suivants :

#### i. Alertes Ionos

Surveiller les notifications de maintenances planifiées et de maintenances d'urgence par notre fournisseur d'hébergement, notamment si elles impliquent une mise hors ligne du serveur.

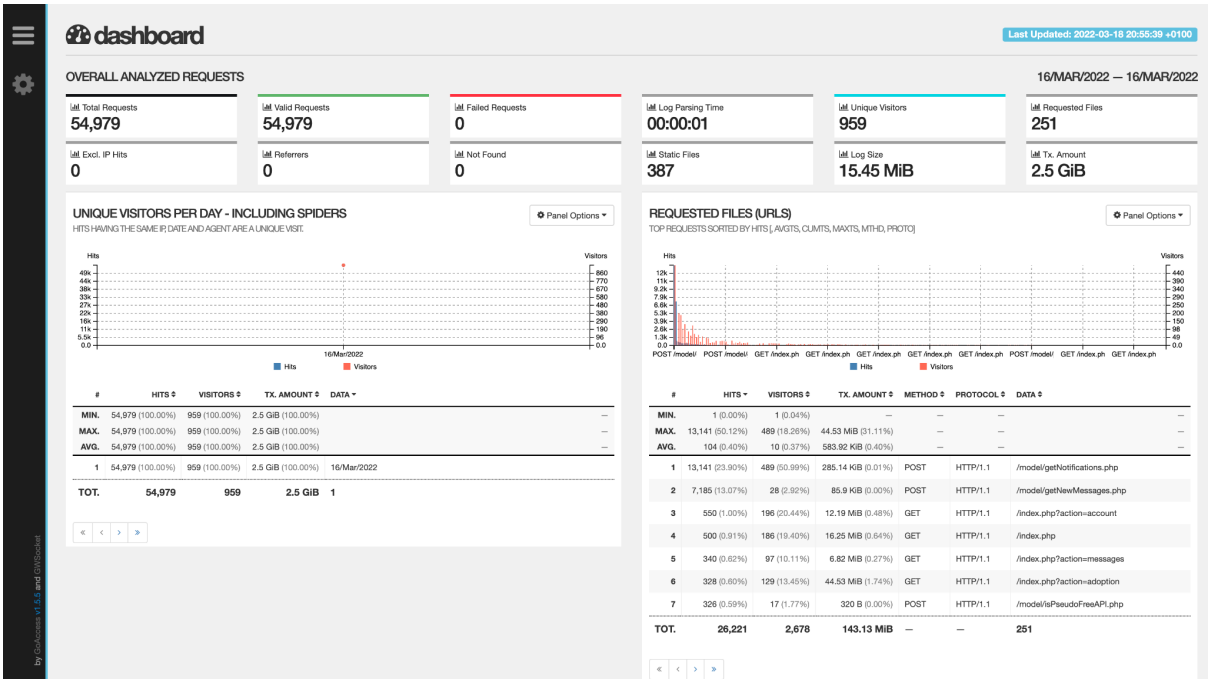
#### ii. Analyses Ionos

Analyser les statistiques fournies par notre hébergeur : nombre de requêtes ayant échoué, vitesse moyenne de réponse aux requêtes…


iii. Analyse des logs

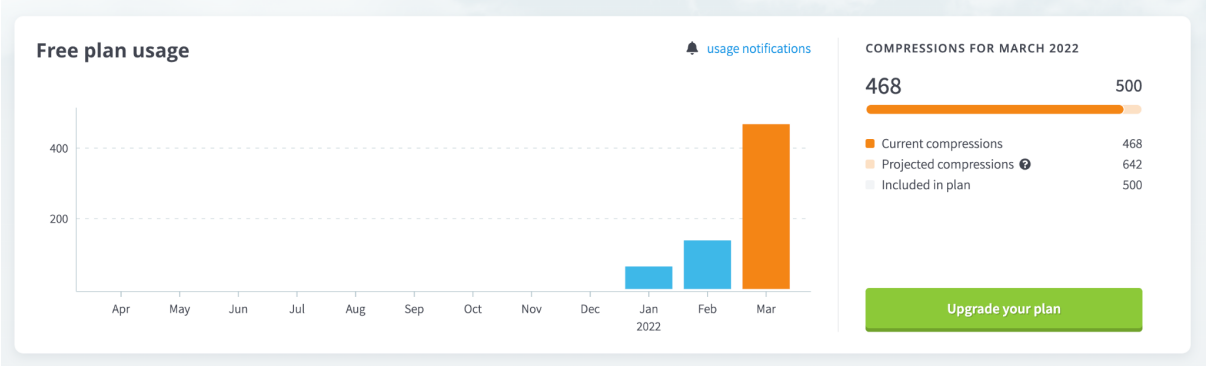
Pour analyser ces informations de manière plus précise, il est possible de télécharger les logs du serveur et de les analyser.

Nous utilisons l'outil goaccess en ligne de commande, qui génère à partir des fichiers de logs une page web "tableau de bord" pour analyser visuellement les informations.




iv. Alertes Tinify

L'API Tinify que nous utilisons pour automatiquement redimensionner et compresser les photos stockées sur notre serveur nous fournit un outil de visualisation du nombre de compressions, des projections (estimation) des compressions à venir sur le mois, et des alertes paramétrables lorsque certains paliers sont atteints.
Nous disposons de plusieurs clé d'API qu'il est possible d'intervertir en cas de désactivation ou de problème technique sur l'une d'entre elles.



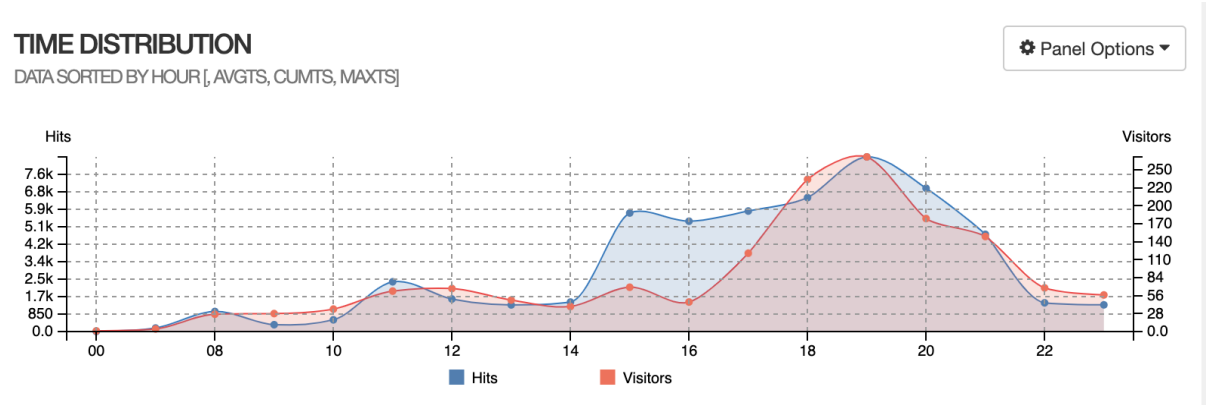

3. Performances
    a. Analyse des logs

L'analyse des logs (voir partie 2. Plan de reprise d'activité d. Outils de monitorings manuels ou automatisés iii. Analyse des logs) permet d'analyser les performances du site et de détecter des liens cassés ou des pages mal optimisées (voir c. Mises à jour préventives planifiées).

    b. Pics de connexion

Nous analysons régulièrement les pics de connexion pour pouvoir les prévoir et anticiper les besoins de montée en charge.

L'analyse des logs ci-dessous nous montre par exemple que plus de 270 visiteurs uniques ont effectués des requêtes vers le serveur sur le créneau de 19h le mercredi 16 mars 2022 :




Le pic de connexion correspond au lancement d'un jeu-concours qui a réuni de nombreux utilisateurs sur le site, suite à des publications sur les réseaux sociaux.

    c. Prévention

Pour prévenir des baisses de performances (côté serveur comme côté client), nous mettons en place différentes mesures :

- minification des codes envoyés en production
- compression de toutes les images stockées sur le serveur
- chargement progressif des pages qui contiennent de nombreuses images (lazy loading en javascript et algorithme optimisé pour le chargement à l'infini du fil d'actualité)

d. Mises à jour préventives planifiées

Cette analyse des logs a révélé que deux algorithmes javascript mal optimisés ont généré à eux deux sur cette journée du 16 mars près de 21 000 requêtes d'appels AJAX vers deux scripts PHP. Cette grande quantité de requêtes, liée à l'utilisation mal optimisée d'intervalles en javascript, pénalise les performances de temps de processeur côté serveur, de l'utilisation du processeur côté client, et du volume réseau pour les clients et le serveur.

Un correctif est actuellement en cours de développement pour optimiser ces algorithmes.

Cette expérience montre que le système actuel est déjà très résilient et capable de résister à des dizaines de milliers de requêtes par jour sans aucune baisse de performance.

Pour limiter l'utilisation du temps de processeur, la compression des images est réalisée par une API externe, et nous ne gérons pas à ce jour pas de vidéos.

e. Projections

Projections sur l'utilisation de l'ensemble des services de Zoey en 2022 (blog, adoption, réseau social) :

- Plus de 1500 compressions de photos via l'API par mois (à ce jour, mi-mars, nous en sommes déjà à la moitié)
- 100 nouveaux comptes utilisateurs par mois en moyenne
- 6 posts par jour en moyenne
- 100 visiteurs par jour sur le site en moyenne, avec des pics à 500 visiteurs simultanés.

Pour un poids moyen de 40ko par photo après compression, nous avons selon ces projections de quoi stocker toutes les photos uploadées jusqu'à la fin de l'année avant de devoir envisager l'investissement dans de nouvelles infrastructures pour supporter la quantité de stockage et la montée en charge.

4. Mesures de sécurité
    a. Conditions générales d'utilisation

L'utilisation de la plateforme (particulièrement pour les utilisateurs connectés, qui peuvent poster des informations publiques et privées textes et images sur le serveur) est régie par les conditions générales d'utilisation de Zoey et par les textes légaux en vigueur au niveau français et européen.

L'acceptation de ces CGU par les utilisateurs est obligatoire pour s'inscrire. Toutes les mentions légales sont consultables sur https://zoey-app.fr/index.php?action=legal

Ces CGU définissent les conditions et les limites des autorisations d'accès des utilisateurs à la plateforme : il est interdit de tenter de modifier le site ou son contenu s'il a été posté par

un autre utilisateur, d'usurper l'identité d'un tiers, de se connecter sur le compte d'un autre utilisateur, de s'introduire dans le back-office d'administration du site (réservé aux administrateurs), de poster du contenu qui appartient à un tiers, de violer une ou plusieurs lois en vigueur dans le droit français et européen, ou encore de tenter de déchiffrer les échanges chiffrés entre des utilisateurs et la plateforme.

b. Protection des données

i. DPO et gestion interne des autorisations d'accès

Marin Bouanchaud est le Data Protection Officer de Zoey. Il supervise les droits d'accès au back-office d'administration et au back-office de Ionos (voir partie c. Droits d'accès aux systèmes privés).
L'accès aux données privées des utilisateurs est particulièrement surveillé et restreint. Il est strictement interdit à toute personne étrangère à l'équipe Zoey et nous respectons une charte interne pour gérer les accès à ces données.

ii. RGPD

Aucune donnée personnelle n'est collectée en dehors du cadre réglementaire du RGPD et du droit applicable.
Les données personnelles de nos utilisateurs sont stockées dans des serveurs en France et dans l'Union Européenne.
Nous ne les communiquons en aucun cas à des tiers et nous n'utilisons aucun système ou cookie de tracking publicitaire.

iii. messagerie chiffrée

Les messages échangés entre les utilisateurs sur la messagerie privée de Zoey sont chiffrés par un algorithme AES 128 bits avec le module OpenSSL de PHP. La clé de chiffrement est stockée et protégée en dehors de la base de données, dans le répertoire "/private_crypt", protégé par une règle "deny from all" dans la configuration Apache.

c. Droits d'accès aux systèmes privés et connection

Tous les systèmes privés de Zoey (le back-office sur zoey-app.fr et le back-office sur l'interface web de notre hébergeur) sont protégés par des mots de passe accessibles uniquement aux administrateurs habilités internes à l'équipe Zoey.
Ces mots de passe sont régulièrement changés par sécurité, et sont générés avec une forte complexité.

Le système de connexion de Zoey est basé sur une preuve soit par mot de passe soit par jeton d'identification fourni par une API de Google au format standard JSON Web Token (signés par le serveur de Google, échangé dans des requêtes chiffrées, et protégé par des dates d'expiration pour éviter les attaques par rejeu).

Le back-office de Zoey est protégé par ce système de connexion de Zoey, mais également par l'attribution de droits administrateurs possibles uniquement manuellement en base de données, afin d'empêcher tout utilisateur non autorisé à devenir administrateur.

d. Prévention des failles

i. injections SQL

Toutes les requêtes vers la base de données sont effectuées via l'API PDO dans PHP, dans des requêtes préparées pour empêcher les injections SQL.

ii. failles XSS

Toutes les informations affichées provenant directement ou indirectement de la base de données sont traitées par l'algorithme "htmlspecialchars" de PHP pour prévenir des risques de faille XSS (injection de code HTML et de scripts JavaScript dans la page).

iii. Gestion des exceptions et des erreurs

Toutes les pages PHP sont construites sur un modèle "try / catch" qui permet de gérer les exceptions PHP et d'afficher des messages d'erreurs dans une page dédiée, avec des boutons de redirection vers l'accueil et un bouton de contact pour signaler l'erreur aux administrateurs.

Les erreurs PHP sont mises en mode "production" et désactivées dans le fichier de configuration php.ini.

De plus, les requêtes aboutissant à un chemin non valide sont également redirigées par les règles htaccess du "routeur" qui fait fonctionner l'architecture MVC, vers index.php qui analyse les paramètres de la requêtes et renvoie soit la page par défaut soit la page demandée si elle existe.

Pour les API répondant aux appels AJAX, si l'utilisateur doit être connecté pour avoir le droit d'y accéder, une vérification est effectuée grâce aux variables de session et si l'utilisateur n'est pas connecté, l'API de renvoie rien et rejette la demande.

e. Veille et prévention

i. analyse des logs : détection des tentatives d'intrusion et des attaques

L'analyse des logs permet de surveiller les activités suspectes sur la plateforme : scan des chemins existants, appels à des pages d'administration privées, tentative d'ouvrir des fichiers protégés, attaques DoS.

ii. veille sur les vulnérabilités

Nous utilisons des versions à jour de tous nos systèmes informatiques, notamment des serveurs, et nous effectuons une veille permanente des menaces en cours en France et des découvertes de failles sur https://www.cert.ssi.gouv.fr/. Nous suivons les recommandations de l'ANSSI sur https://www.ssi.gouv.fr/.