fenrir.pro

Rapport d’Audit

[redacted]

Jeremie Amsellem (fenrir.pro)

Table des matières

Spécifications de l'Audit **3**

Versions du document **4**

Avertissement **5**

I - Synthèse **6**

II - Description de la méthodologie **6**

III - Détails du test d'intrusion **7**
Étendue du test . . . 7

IV - Prise d'empreintes **8**
IP et Domaines . . . 8
Informations techniques . . . 8
Code source disponible publiquement . . . 9
Technologies utilisées . . . 9
Documentation en ligne . . . 9

V - Scan **10**
Scan de ports . . . 10
Scan de configuration SSL/TLS . . . 10
Scan de configuration HTTP . . . 11
Scan de la configuration technique . . . 11
127.0.0.2 (▯▯▯▯▯▯▯▯▯_domain.org) . . . 11
127.0.0.3 (back-end.▯▯▯▯▯▯▯▯▯_domain.org) . . . 11

VI - Exploitation **13**
Services SSH . . . 13
ElasticSearch . . . 13
PostgreSQL . . . 14
Gestion des utilisateurs . . . 15
Inscription . . . 15
Authentification . . . 15
Gestion des sessions . . . 16
Mot de passe oublié . . . 18

Gestion des informations utilisateur 18
Edition des informations du compte 18
Modification de l'avatar 19
Edition du mot de passe 20
Mot de passe perdu 21
Déconnexion 21
Accueil 22

Spécifications de l'Audit

Date de début : 16/06/2021

Durée : 4.5 Jours

Référence du document : F-1215-INT

Commandé le : 15/06/2021

Par : [redacted]

Entreprise : [redacted]

Versions du document

Version	Date	Description
1.0	16/06/2021	Version initiale
1.1	20/06/2021	Ajout des audits de configuration
1.2	23/06/2021	Formatage et ajout des dernières informations

Avertissement

Un test d'intrusion est à considérer comme un instantané fait à un instant précis dans la vie de votre projet. Les découvertes et recommandations de ce rapport reflètent les informations découvertes au cours de la durée limitée du test et ne sauraient représenter les modification ou changements faits en dehors de cette période. Les analyses sur une durée limitée ne permettent pas de réaliser une évaluation complète des contrôles de sécurité. Fenrir a ciblé en priorité les mécanismes de sécurité les plus faibles, qui risquent d'être pris pour cible lors d'une attaque. Nous recommandons de réaliser ce type d'opérations à des intervalles réguliers (au minimum une fois par an) par une équipe offensive - interne ou externe pour assurer la pérennité de vos processus de sécurité

I - Synthèse

[redacted]

II - Description de la méthodologie

La méthodologie suivie pour la réalisation de ce test est inspirée de l'OSSTMM et de la méthodologie LPT d'EC-Council.

Elle est composée de 3 phases distinctes :

Une phase de **prise d'empreintes** au cours de laquelle des informations techniques et organisationnelles sont récupérées depuis des sources publiques.

Une phase de **scan** au cours de laquelle des hôtes, ports et services sont scannés sur les domaines et adresses précédemment découverts.

Une phase d'**exploitation** au cours de laquelle des vulnérabilités sont exploitées pour tenter de gagner des accès sur le système d'information cible.

III - Détails du test d'intrusion

Étendue du test

L'audit comprend deux hôtes :

- 127.0.0.2 (▒▒▒▒▒▒▒▒▒_domain.org)
  - Machine Ubuntu 20.04
  - Hostname ▒▒▒▒▒▒▒▒▒
  - Accès SSH sur le port 22 via l'utilisateur ubuntu

- 127.0.0.3 (back-end.▒▒▒▒▒▒▒▒▒_domain.org)
  - Machine Ubuntu 20.04
  - Hostname ▒▒▒▒▒▒▒▒▒
  - Accès SSH sur le port 22 via l'utilisateur ubuntu

IV - Prise d'empreintes

IP et Domaines

Une requête DNS vers le nom de domaine [redacted]_domain.org permet de noter que l'adresse IPv4 associée au domaine est *127.0.0.3*.

Les headers renvoyés en HTTPS permettent d'identifier un reverse proxy NGINX géré par OVH.

Une requête DNS vers le nom de domaine back-end.[redacted]_domain.org permet de noter que l'adresse IPv4 associée au domaine est *127.0.0.4*.

Les headers renvoyés en HTTPS permettent d'identifier un reverse proxy NGINX géré par OVH.

(Screenshot présentant les informations sur le site securitytrails.com)

> https ://securitytrails.com/domain/[redacted]_domain.org/history/a

Un historique des entrées DNS associées au domaine [redacted]_domain.org permet d'identifier l'adresse IP du serveur front-end à l'adresse **127.0.0.2** et l'adresse IP du serveur back-end à l'adresse **127.0.0.3**.

> **Note** : L'accessibilité de ces deux adresses IP limite l'avantage du reverse proxy qui permet de garder ces adresses confidentielles et limiter la surface d'attaque d'une personne ayant connaissance des domaines [redacted]_domain.org et back-end.[redacted]_domain.org.

Informations techniques

Le moteur de recherche Shodan indique que les ports **22 et 80** sont accessibles sur l'adresse IPv4 **127.0.0.2**.

Le port 22 correspond au service *OpenSSH 8.2p1 Ubuntu-4ubuntu0.2*

Le port 80 est un serveur *NGINX 1.18.0*

(Détails des informations disponibles sur Shodan)

Le moteur de recherche Shodan indique que les ports **22, 80 et 5432** sont accessibles sur l'adresse IPv4 **127.0.0.3**.

Le port 22 correspond au service *OpenSSH 8.2p1 Ubuntu-4ubuntu0.2*

Le port 80 est un serveur *NGINX 1.19.10*

Le port 5432 est un serveur *PostgreSQL*

(Détails des informations disponibles sur Shodan)

Code source disponible publiquement

Aucune référence au projet [redacted] n'a été trouvée dans du code source disponible publiquement.

Technologies utilisées

Les headers renvoyés et les fichiers envoyés par le serveur lors de l'accès à https ://[redacted]_domain.org indiquent la présence d'une application React ainsi que de la bibliothèque Lodash en version 4.17.10.

Les headers renvoyés par le serveur lors de l'accès à https ://back-end.[redacted]_domain.org ne permettent pas l'identification de technologies utilisées en back-end.

Documentation en ligne

Aucune documentation (technique ou organisationelle) en rapport avec le projet n'a été trouvée dans les sources disponibles publiquement.

V - Scan

Scan de ports

L'hôte à l'adresse **127.0.0.2** ([redacted]_domain.org) expose les ports suivants :

- 22/tcp ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
- 80/tcp http nginx 1.18.0 (Ubuntu)

L'hôte à l'adresse **127.0.0.3** (back-end.[redacted]_domain.org) expose les ports suivants :

- 22/tcp ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.2 (Ubuntu Linux; protocol 2.0)
- 80/tcp http nginx 1.19.10
- 5432/tcp postgresql PostgreSQL DB >= 9.6.0
- 9200/tcp elasticsearch

> **Note** : Les instances de PostgreSQL et ElasticSearch sont potentiellement exposées involontairement, si le back-end ou le front-end ne requièrent pas cette ouverture il serait amplement préférable de ne pas laisser ces ports accessibles sur le WAN

Scan de configuration SSL/TLS

La configuration HTTPS des deux domaines ([redacted]_domain.org et back-end.[redacted]_domain.org) est très globalement similaire : deux certificats Let's Encrypt valides (vérification de la signature par la Validation Authority et dates d'expiration OK).

Deux protocoles proposés par le serveur sont dépréciés :

TLS 1.0 et TLS 1.1.

Ceux-ci ne devraient plus être utilisés et remplacés par TLS 1.2.

Le protocole de chiffrement **SEED + 128+256 Bit CBC cipher** est lui aussi déprécié et ne devrait plus être offert par le serveur HTTPS.

Le reste de la configuration est robuste, HSTS est utilisé, le reste des protocoles est adéquat.

> **Note** : OVH a désactivé les protocoles concernés en interne sur leurs APIs (http ://travaux.ovh.net/?do=details&id=39357&edit=yep) et il n'y a malheureusement pas encore de date prévisionelle pour une désactivation globale sur leurs reverse proxies.

Scan de configuration HTTP

La configuration du serveur HTTP ne semble pas présenter de vulnérabilités, les headers concernant le HSTS, le X-XSS-Protection, et le ClickJacking sont correctement configurés et aucun répertoire exposant des informations sensibles n'a été découvert.

Les champs Etag ne permettent pas l'identification des serveurs derrière le reverse proxy.

Scan de la configuration technique

127.0.0.2 ([redacted]_domain.org)

La machine est une Ubuntu 20.04.2 dont les paquets sont à jour dans l'ensemble.

Rien de notable sur la configuration matérielle/réseau de cet hôte.

La version du kernel et la majorité des paquets installés sur l'hôte sont à jour.

La configuration du fichier /etc/sudoers est celle par défaut des VPS Ubuntu d'OVH, elle permet l'élévation des privilèges de l'utilisateur ubuntu sans mot de passe, il est également probable que les mots de passes de ce compte n'ait pas été changé depuis l'installation initiale. Pour garantir une confidentialité des informations de cette machine il serait préférable de ne pas autoriser l'execution de commandes sans mot de passe et changer le mot de passe de cet utilisateur.

Apparmor, ASLR et SELinux sont correctement activés.

> **Note** : Le détails de ce scan de configuration sont disponibles dans le dossier scans.

127.0.0.3 (back-end.[redacted]_domain.org)

La machine est une Ubuntu 20.04.2 dont les paquets sont à jour dans l'ensemble.

Rien de notable sur la configuration matérielle/réseau de cet hôte.

La version du kernel Linux utilisé comporte plusieurs vulnérabilités et il serait préférable de le mettre à jour (https ://ubuntu.com/security/notices/USN-4887-1) vers une version 5.8.0.

Apparmor, ASLR et SELinux sont correctement activés.

La configuration du serveur SSH autorise l'authentification via l'utilisateur root, à moins que la fonctionnalité soit indispensable, il serait préférable d'ajouter la configuration "PermitRootLogin no" dans le fichier */etc/ssh/sshd_config*.

La configuration du fichier /etc/sudoers est celle par défaut des VPS Ubuntu d'OVH, elle permet l'élévation des privilèges de l'utilisateur ubuntu sans mot de passe, il est également probable que les mots de passes de ce compte n'ait pas été changé depuis l'installation initiale. Pour garantir une confidentialité des informations de cette machine il serait préférable de ne pas autoriser l'execution de commandes sans mot de passe et changer le mot de passe de cet utilisateur.

On remarque encore une fois plusieurs ports ouverts sur cet hôte (22, 80, 5432, 9200), ceux-ci exposent des informations potentiellement sensibles et ne devraient pas être accessibles depuis le WAN.

Deux processus utilisent +30% de la RAM disponible et 200% (100% de 2 coeurs) de la bande passante du CPU.

```
# ps auxwww

systemd+ 1415365  324 30.1 3070028 2401280 ?     Ssl  Jun16 9682:58
 ↪  /tmp/kdevtmpfsi

systemd+ 1474002  195 30.2 2883560
 ↪  2405372 ?    Sl   13:30   6:14 /tmp/shellzsh -o pool.supportxmr.com:443 -u
 ↪  88B2cN4kbEJbCy3N9o5rsGSVgETMpQLRkF3zmnt3wt2DbTZedxBjG72j9wDHkyBzZGNp147gWWUKxVpeiZoq
 ↪  -k --tls
```

C'est au vu des URLs indiquées probablement un mineur de cryptomonnaies (XMR/Monero) qui a été installé par un acteur externe sur le serveur à la suite d'une intrusion.

> **Note** : Le détails de ce scan de configuration sont disponibles dans le fichier scans/18_06_2021_13h33_conf_audit_

VI - Exploitation

Services SSH

La version d'OpenSSH server utilisée est à jour et ne présente pas de vulnérabilité exploitable dans l'environnement actuel.

ElasticSearch

Un instance ElasticSearch est publiquement accessible sans authentification sur le port 9200 de l'hôte **127.0.0.3**.

Il est possible de lister les indices existants à l'aide de la requête : http ://127.0.0.3 :9200/_cat/indices ?v .

Trois indices existent : user_login, interface et admin_user.

Et d'enumérer les différentes entrées disponibles à l'aide des requêtes :

http ://127.0.0.3 :9200/admin_user/_search ?pretty=true

http ://127.0.0.3 :9200/user_login/_search ?pretty=true

http ://127.0.0.3 :9200/interface/_search ?pretty=true

Ou encore de récupérer l'ensemble des informations à l'aide de :

http ://127.0.0.3 :9200/_search ?pretty=true&size=200

Parmis les informations disponibles sous l'indice "admin_user", on remarque les adresses mail, numéros de téléphone, rôle, IDs des administrateurs.

Ces informations devraient au vu de leur caractère privé être stockées et accessible d'une manière sécurisée.

Les droits d'accès en écriture sont également accessibles sans authentification, ce qui permet l'ajout d'informations, par exemple à l'aide de la requête :

```
curl -X POST -H 'Content-Type: application/json'
 ↪  http://127.0.0.3:9200/admin_user/_doc -d
 ↪  '{"who":-1,"action":"intrusion",
"email":"jeremie.amsellem@protonmail.com","role":"super_admin"}'
```

On retrouve dans la réponse :

(Screenshot de la réponse du serveur)

Ou encore de supprimer des informations a l'aide d'une requête DELETE.

```
curl -X DELETE -H 'Content-Type: application/json'
 ↪  http://127.0.0.3:9200/admin_user -d '{"who":-1,"action":"intrusion",
"email":"jeremie.amsellem@protonmail.com","role":"super_admin"}'
```

Vulnérabilité	F-1215-V1
Identifiant CWE	CWE 200 - Exposition de données sensibles
Sévérité	Haute
Description	Plusieurs informations concernant les administrateurs de la plateforme ainsi que leurs actions sont accessibles sans authentification en lecture et en écriture via l'instance Elasticsearch sur le port 9200 de l'hôte **127.0.0.3**.
Exploitation	Des exemples d'exploitation sont disponibles dans les fichiers *./exploits/get_admin_emails.sh* et *./exploits/get_admin_phones.sh*.
Remédiation	Configurer le service pour écouter sur une adresse locale plutôt que sur 0.0.0.0.
Ressources	https ://cwe.mitre.org/data/definitions/200.html https ://www.bleepingcomputer.com/news/security/hackers-are-quick-to-notice-exposed-elasticsearch-servers/

PostgreSQL

Une instance PostgreSQL (version > 9.6.0) est exposée sur le port 5432 de l'hôte **127.0.0.3**.

Elle est accessible depuis le WAN sans mot de passe à l'aide du nom d'utilisateur **deploy** et du nom d'utilisateur **par défaut** de PostgreSQL : **postgres**.

Elle permet le consultation et la modification du contenu de la base de données de la plate-forme "buzzeo_production" ainsi que l'exécution de commandes via l'exploitation d'une fonctionnalité de Postgres (https ://medium.com/greenwolf-security/authenticated-arbitrary-command-execution-on-postgresql-9-3-latest-cd18945914d5).

Vulnérabilité	F-1215-V2
Sévérité	Critique
Description	Le moteur de base de données PostgreSQL est exposé sur le port 5432 de l'hôte **127.0.0.3** et accessbile sans mot de passe via le nom d'utilisateur par défaut : postgres. Il permet de parcourir et éditer le contenu des bases de données ainsi que le système de fichiers de l'hôte. Une execution de commande est également possible.
Exploitation	`$> psql -h 127.0.0.3 -U postgres`
Remédiation	Configurer le service pour écouter sur une adresse locale plutôt que sur 0.0.0.0.
Ressources	https ://cwe.mitre.org/data/definitions/200.html https ://medium.com/greenwolf-security/authenticated-arbitrary-command-execution-on-postgresql-9-3-latest-cd18945914d5

Gestion des utilisateurs

Inscription

L'inscription d'un utilisateur passe par l'envoi d'une invitation depuis l'interface administrateur.

Le lien d'invitation généré est de la forme :

https ://[redacted]_domain.org/sign-up ?email=jeremie.amsellem%40protonmail.com&invitation_token=Zy2Addy
xZVKigj7gGNBCAT-QZBzRFBkd8Es

La validation de l'e-mail ainsi que du token est correcte, le token ne semble pas être dérivé d'informations statiques accessibles.

L'expiration du token est correctement vérifiée et ne permet pas de validation multiple.

Authentification

L'authentification se fait par défaut via une requête POST sur la route **/api/v1/users/authenticate** avec un paramètre *user* contenant les champs *email*, *password* et *otp_attempt*.

{"user":{"email":"jeremie.amsellem@protonmail.com",
"password":"Test4242!","otp_attempt":"270284"}}

Les règles concernant les mots de passe qui sont validées par le front-end **et** le back-end sont :

- Au moins 8 caractères
- 1 chiffre
- 1 caractère majuscule
- 1 caractère minuscule
- 1 caractère spécial

En France, l'ANSSI recommande d'utiliser des mots de passe d'au moins 12 caractères au lieu des 8 caractères - "choisissez des mots de passe d'au moins 12 caractères de types différents (majuscules, minuscules, chiffres, caractères spéciaux)".

Les hashs de mots de passes ont pu être récupérés sans authentification depuis la base de données via la vulnérabilité *F-1215-V2*, ils sont chiffrés à l'aide de la fonction de hachage bcrypt (null-terminated, UTF-8), et salés 11 fois.

Les hashs de mot de passe sont stockés de manière sécurisée dans la base de données et se sont montrés résistants à des attaques de brute-force local au cours des tests.

Concernant les attaques **en ligne**, des limitations sont mises en place contre le brute-force et les attaques par dictionnaire, après 5 tentatives de connexion infructueuses, un compte est verrouillé pendant 30 minutes.

L'authentification multi-facteurs (MFA) est activée par défaut sur les comptes crées et ne peut pas être désactivée sans modification manuelle des informations de la base de données.

Gestion des sessions

Lors d'une requête de connexion fructueuse, un identifiant de session est renvoyé par le serveur et ajouté aux requêtes du client dans un header *X-User-Token*.

Un header *X-User-Email* contenant l'email de l'utilisateur courant est également communiqué. Sa validité est vérifiée et il ne permet pas l'authentification a l'aide de l'adresse d'un autre utilisateur de la plate-forme.

1000 identifiants de session ont été récupérés pour le compte de test fourni, entre le début des tests (16/06/2021) et le 22/06/2021 aucune variation n'a été remarquée sur ceux-ci.

Les identifiants de sessions, bien que possiblement dynamiques sont probablement générés à partir de données temporelles trop éloignées (à minima plusieurs jours) pour éviter la réutilisation d'un identifiant de session.

Ces identifiants ne sont pas générés à partir de données aléatoires, leur offrant une entropie très faible. En cas de compromission d'un User-Token, il pourraît être réutilisé pendant plusieurs jours sans qu'un utilisateur puisse invalider sa session.

Vulnérabilité	F-1215-V3
Sévérité	Moyenne
Description	Les identifiants de sessions *X-User-Token* renvoyés par le back-end Rails ne présentent pas de caractère aléatoire et d'expiration suffisantes pour garantir leur non réutilisation en cas de capture.
Remédiation	Ajouter des informations aléatoires lors de la génération des identifiants de session et forcer leur expiration après une durée plus courte.
Ressources	https ://www.ssi.gouv.fr/uploads/IMG/pdf/NP_Securite_Web_NoteTech.pdf https ://codeahoy.com/2016/04/13/generating-session-ids/ https ://cwe.mitre.org/data/definitions/613.html

Les informations de session sont stockées par le front-end dans des Cookies, ceux-ci n'utilisent pas l'option **httpOnly** et peuvent être récupérés en JavaScript dans le navigateur des clients.

L'option **Secure** n'est pas non plus utilisée, celle-ci empêche la transmission des informations du Cookie HTTP en clair et force l'utilisation du chiffrement.

(Screenshot de la récupération du cookie)

Vulnérabilité	F-1215-V4
Identifiant CWE	CWE 1004 - Cookies contenant des données sensibles sans paramètre HttpOnly
Sévérité	Haute
Description	Les Cookies renvoyés par le front-end n'utilisent pas les paramètres *httpOnly* et *secure*, ceux-ci permettent de garantir la confidentialité des informations stockées.
Remédiation	Lors de la mise en place des cookies, ajouter les paramètres secure et httpOnly.

Ressources	https ://medium.com/@ryanchenkie_40935/react-authentication-how-to-store-jwt-in-a-cookie-346519310e81 https ://owasp.org/www-community/HttpOnly https ://owasp.org/www-community/controls/SecureCookieAttribute

---

Mot de passe oublié

La fonctionnalité de mot de passe oublié utilise une requête POST vers la route **/api/v1/users/password** du back-end avec en body un objet JSON de la forme :

{"user":{"email":"[redacted]@lp1.eu"}}

Le back-end vérifie correctement la validité des e-mails, et la route ne semble pas vulnérable aux injections PostgreSQL.

Gestion des informations utilisateur

Edition des informations du compte

L'édition des informations utilisateur utilise une requête HTTP PUT sur la route **api/v1/users/122 du back-end** contenant dans le corps de la requête un objet de la forme :


```
{
   "invited_by_id" : 13,
   "email" : "jeremie.amsellem@protonmail.com",
   "first_name" : "Jérémie1",
   "role" : "super_admin",
   "title" : "M.",
   "otp_enabled" : true,
   "type" : "AdminUser",
   "_destroy" : false,
   "subsidiary_id" : "4",
   "full_name" : "Jérémie Amsellem",
   "phone" : "0607462492",
   "avatar" : {
```

```
      "url" : "https://back-
       ↪  end.████████_domain.org/uploads/admin_user/avatar/122/admin.html",
      "file_identifier" : "admin.html",
      "filesize" : 53
   },
   "qualification" : "technical_staff",
   "privacy_policy_accepted_at" : "2021-06-16T14:37:06.886+02:00",
   "updated_at" : "2021-06-19T14:38:29.368+02:00",
   "id" : 122,
   "last_name" : "Amsellem",
   "privacy_policy_accepted" : true,
   "created_at" : "2021-06-15T17:51:48.086+02:00"
}
```


La route ne semble pas vulnérable aux injections PostgreSQL, JSON et XSS.

Les informations suivantes ne peuvent pas être modifiées via cette route :

- id
- updated_at, created_at
- type
- avatar
- full_name
- privacy_policy_accepted
- privacy_policy_accepted_at
- otp_enabled
- invited_by
- subsidiary_id

L'élévation de privilège d'un compte Administrateur vers un compte Super Administrateur n'est pas possible, la modification de l'adresse e-mail ne fonctionne que partiellement et ne permet pas le re-connexion via l'interface.

Il ne semble pas qu'il soit possible de modifier de manière non autorisée les informations d'un autre utilisateur que l'utilisateur courant, l'authentification et les autorisations sont correctement contrôlées.

Modification de l'avatar

La modification de l'avatar utilise une requête PUT sur la route **/api/v1/admin_users/122** du back-end avec en paramètre *admin_user[avatar]* le fichier contenant l'avatar.

Le front-end effectue une vérification concernant le format des fichiers envoyés, le back-end permet l'envoi de fichiers sans limitation de format. Par exemple, des fichiers HTML peuvent être téléversés et servis sur le serveur HTTP à l'adresse back-end.[redacted]_domain.org, dans le but d'organiser des campagnes de phishing.

> Exemple : https ://back-end.[redacted]*domain.org/uploads/admin_user/avatar/122/*..

(Screenshot de l'upload sur le back-end)

Vulnérabilité	F-1215-V5
Identifiant CWE	CWE-434 - Téléversement non restreint de types de fichiers dangereux
Sévérité	Moyenne
Description	Les espaces d'upload de fichiers de la plate-forme permettent l'envoi de fichiers sans restriction sur leur types, ce qui permet l'hébergement de contenu de phishing ou autre contenu malveillant sur le serveur back-end.[redacted]_domain.org.
Remédiation	Restreindre les types de fichiers pouvant être envoyés sur les formulaires d'upload sur la back-end.
Ressources	https ://cwe.mitre.org/data/definitions/434.html

Les injections de fichiers (LFI, RFI) et de commandes, tout comme les injections PostgreSQL et XSS sur le nom des fichiers ne sont pas réalisables sur l'infrastructure actuelle, les entrées utilisateurs sont correctement validées par le back-end.

> **Note** : On remarque tout de même une réponse en erreur 500 sur un nom de fichier contenant le caractère de fin de chaîne de caractères : "%00". Par exemple : *https :admin3.php ;%00ls*

Edition du mot de passe

(Screenshot de l'espace de modification du mot de passe)

Lors de l'édition du mot de passe, le mot de passe actuel est correctelment validé, les restrictions liées aux mots de passe sont correctement vérifiées, la route ne semble pas vulnérable aux injections et les entrées utilisateurs sont correctement gérées par le back-end.

> **Note** : On note tout de même que lors du changement de mot de passe, la session (et l'identifiant de session associé) n'est pas renouvelée par le serveur. C'est une bonne pratique de sécurité de renouveler les sessions d'utilisateurs lors du changement des mots de passes et des adresses e-mail.

Mot de passe perdu

La fonctionnalité de mot de passe perdu utilise une requête POST sur la route **/api/v1/users/password** contenant un objet JSON de la forme :

```
{"user":{"email":"jeremie.amsellem@protonmail.com"}}
```

Une réponse *201 Created* est renvoyée par le serveur, en revanche aucun mail n'est reçu, il semble que la fonctionnalité ne soit pas encore complêtement utilisable.

Déconnexion

La déconnexion semble être gérée uniquement par le front-end de la plate-forme, aucune requête n'est faite vers le back-end et les identifiants de sessions peuvent être réutilisées même après une déconnexion.

En cas de compromission de l'identifiant de session d'un utilisateur (vol d'un appareil, vol de fichiers par un malware, interception de traffic, injection XSS […]), celui-ci ne possède donc pas de moyen pour l'invalider.

La durée étendue de validité des sessions et la non utilisation des paramètres *secure* et *httpOnly* rend dans le contexte de la plate-forme ▒▒▒▒▒▒▒▒ la vulnérabilité particulièrement notable.

Vulnérabilité	F-1215-V6
Identifiant CWE	CWE-613 - Expiration de sessions insuffisante
Sévérité	Moyenne
Description	Les sessions ne sont pas détruites par le back-end lors de la déconnexion. Celle-ci est gérée uniquement par le front-end.
Remédiation	Ajouter une route de déconnexion sur le back-end qui sera appellée par le front-end lors de la déconnexion.

Ressources	https ://cwe.mitre.org/data/definitions/613.html

Accueil

(Screenshot présentant l'écran d'accueil)

La page d'accueil du front-end affiche de nombreuses informations chargées sur le back-end via les requêtes GET suivantes :

- /api/v1/task_index_items ?per_page=10&page=1&q[s]=created_at%20desc&include[][task][][operation][]=cli
- /api/v1/sources ?per_page=5&page=1
- /api/v1/interface_templates ?include[]=interface_template_items&q[s]=undefined%20undefined&page=1&pe
- /api/v1/devices ?include[]=device_family&q[s]=undefined%20undefined&page=1&per_page=1&
- /api/v1/scenario_execution_logs ?per_page=10&page=1&q[s]=created_at%20desc&include[][scenario_execu
- /api/v1/places
- /api/v1/clients ?q[poc_admin_user_id_eq]=122&page=1&per_page=1
- /api/v1/leads ?per_page=1&page=1
- /api/v1/lead_transfer_reports ?per_page=10&page=1&q[s]=created_at%20desc&include[]=lead_transfer
- /api/v1/operations ?q[client_poc_admin_user_id_eq]=122&page=1&per_page=1
- /api/v1/clients ?q[s]=created_at%20desc&page=1&per_page=10
- /api/v1/operations ?include=client&page=1&per_page=1

Ces routes ne se sont pas vulnérable aux injections PostgreSQL et ne semblent pas permettre l'affichage de données en désaccord avec les règles concernant l'authentification et les autorisations de la plate-forme.
