

UE SEC-105

Travaux pratiques

Mise en œuvre d'un outil ou
d'un concept de sécurité

Installation et utilisation de Vaultwarden

Suivi du document

Date	Description	Auteur
28/10/2024	Création de l'exemple	Julien Le Coz

Autorisation de partage

☒ J'autorise le partage du présent document aux autres élèves « Sec-105 » (la présente page sera supprimée pour garantir l'anonymisation du document)

Composition de :

- premier auditeur CNAM : Julien Le Coz

Table des matières

Description de l'objectif de sécurité visé.....	5
Schéma de principe.....	6
Implémentation.....	7
Conclusion.....	7
Sources.....	8
Sources WEB.....	8
Sources papier.....	8
Annexes (facultatives).....	9

Description de l'objectif de sécurité visé

A l'heure actuelle ma façon de gérer mes mots de passe est très problématique: je me contente de les stocker sur mon disque dur dans un fichier chiffré grâce à l'éditeur vim (algorithme Blowfish2)

Bien que cet algorithme soit à priori actuellement sûr, consulter ce fichier n'est pas très pratique et le copié-collé n'est pas possible. Cela se traduit par l'utilisation de mots de passe qui ne sont que moyennement complexes, et la réutilisation de certains de ces mots de passe. Bien sûr cela ne m'incite pas non plus à leur modification régulière. Qui plus est, mes mots de passe sont disponibles uniquement sur mon PC.

Je suis marié et j'ai 4 enfants. La gestion que fait ma femme de ses mots de passe est tout simplement catastrophique et les enfants perdent régulièrement leur accès à certains services où notent des mots de passe dans des carnets. Evidemment les mots de passe Netflix et autres nous sont régulièrement demandés vu que l'information ne leur est pas facilement accessible.

Je possède un PC sous Linux et ma femme un PC sous Windows. Nous avons tous un téléphone mobile, dont 5 appareils Android et un appareil Apple. Je recherche donc une solution qui soit utilisable quelque soit la plateforme pour stocker et diffuser ces informations de façon sécurisée et facile d'utilisation pour inciter les moins sensibles à ces enjeux à l'utiliser.

Dans mon entreprise nous utilisons Keypass pour stocker tous les mots de passe commun. La gestion des mots de passe personnels est à la charge de chacun, ce qui sous-entend l'utilisation d'au moins deux solutions différentes. Une solution permettant de gérer plusieurs comptes permettrait de rendre cette gestion plus claire et inciter tout le monde à utiliser systématiquement un gestionnaire de mot de passe.

Je vais donc chercher à configurer et déployer une instance de Vaultwarden sur un serveur virtuel d'Ionos de façon à ce que chaque membre de mon foyer puisse l'utiliser et tester ainsi l'intérêt et l'utilisation d'un gestionnaire de mot de passe de la façon la plus concrète qui soit. **Il serait bon de sauvegarder toutes ces données, comme les options de sauvegarde automatique sont payantes, je vais plutôt envisager de la gérer moi-même en m'aidant d'un serveur que je loue déjà chez un autre hébergeur.**

J'ai aperçu qu'un serveur Passbolt mettait à disposition une API. Par curiosité je vais tester cette API en tentant de mettre en place un CRUD par le biais d'une application Nodejs en ligne de commande. Ce n'est pas la technologie la plus adéquate pour ça, mais Javascript est ma zone de confort et comme le développement de l'application en elle-même n'est pas l'objet de ce dossier, Nodejs fera très bien l'affaire.

Ce que je vais mettre en place en précisant les objectifs

- Comment je vais m'y prendre

Comment je vais tester que j'ai atteint ou non mes objectifs

Schéma de principe

Par le biais d'un « dessin » et éventuellement d'une explication complémentaire je décris mon lab. Je peux notamment faire apparaître :

- Le ou les serveurs
- Le ou les applications
- Les IP ou nom
- Les flux et protocoles
- Les fichiers et/ou dossiers notables

Implémentation

Après avoir créé un compte utilisateur avec des droits sudo sur mon serveur virtuel Debian, je me suis assuré de pouvoir m'y connecter via SSH afin de pouvoir confortablement faire toutes les manipulations nécessaires à l'installation de Vaultwarden. Je me suis aussi créé un sous-domaine et une adresse mail dédiée.

Installation :

Petite mise à jour du système et on commence par installer les paquets requis pour l'installation de Docker. (nécessaire pour l'installation de Vaultwarden)

```
julien@ionos-server:~$ sudo apt-get install docker-compose apt-transport-https ca-certificates gnupg2 software-properties-common
```

On télécharge ensuite la clef GPG depuis le référentiel Docker. Cette clef permet de s'assurer que les paquets proviennent d'une source fiable

```
julien@ionos-server:~$ curl -fsSL https://download.docker.com/linux/debian/gpg | sudo apt-key add -
```

On ajoute le référentiel Docker aux sources du gestionnaire de paquets...

```
julien@ionos-server:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/debian $(lsb_release -cs) stable"
```

... et on fait une mise à jour pour s'assurer d'avoir les dernières informations concernant les paquets en question.

```
julien@ionos-server:~$ sudo apt-get update
```

On accède à la stratégie de cache pour vérifier que les paquets d'installation proviennent bien de Docker.

```
julien@ionos-server:~$ sudo apt-cache policy docker-ce
```

Et on peut enfin finir par installer Docker en lui-même !

```
julien@ionos-server:~$ sudo apt-get install docker-ce
```

Docker est installé, mais une petite vérification à posteriori n'est jamais inutile

```
julien@ionos-server:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Sun 2024-11-17 19:58:37 UTC; 20s ago
 TriggeredBy: ● docker.socket
    Docs: https://docs.docker.com
   Main PID: 12306 (dockerd)
     Tasks: 7
    Memory: 29.6M
       CPU: 286ms
    CGroup: /system.slice/docker.service
            └─12306 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Nov 17 19:58:37 ionos-server systemd[1]: Starting docker.service - Docker Application Container Engine.
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.261687165Z" level=info msg="Starting daemon"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.340264988Z" level=info msg="Graph driver"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.340938611Z" level=info msg="Loading"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.617779211Z" level=info msg="Default"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.674300445Z" level=info msg="Loading"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.688068183Z" level=info msg="Documentation"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.688298688Z" level=info msg="Daemon"
Nov 17 19:58:37 ionos-server dockerd[12306]: time="2024-11-17T19:58:37.719756640Z" level=info msg="API"
Nov 17 19:58:37 ionos-server systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-22/22 (END)
```

Nous avons donc bien une instance fonctionnelle de docker qui tourne sur le serveur.

Docker installé, nous avons déjà fait le plus gros du boulot : il ne nous reste plus qu'à lancer deux conteneurs.

- Le conteneur dans lequel tournera notre instance de Vaultwarden
- Le conteneur dans lequel nous allons faire tourner un reverse-proxy afin d'établir une

connexion via https pour protéger nos interactions avec l'instance de Vaultwarden des attaques type « man in the middle »

On commence par créer un dossier qui contiendra notre configuration puis à l'intérieur de ce dossier un fichier de configuration pour nos deux conteneurs (détails du docker-compose.yml en annexe). Nous allons utiliser des volumes afin de conserver hors du conteneur certaines des données consommées par notre instance de Vaultwarden. **D'une part on s'assure ainsi de ne pas supprimer toutes ces données à chaque redémarrage du conteneur, d'autre part, comme nous le verrons plus tard cela sera pratique pour gérer tout ce qui a trait à la sauvegarde des données en question.**

```
julien@ionos-server:~$ mkdir vaultwarden && cd vaultwarden && touch docker-compose.yml
```

Pour le reverse-proxy, nous allons utiliser Caddy qui a le gros avantage d'être facile à déployer et de gérer automatiquement le protocole HTTPS. Il nous faut un fichier de configuration, fichier qui sera utilisé par l'instance de Caddy (détails de Caddyfile en annexe)

```
julien@ionos-server:~/vaultwarden$ touch Caddyfile
```


La configuration des conteneurs est terminée. C'est maintenant l'heure de vérité. On monte les conteneurs...

```
julien@ionos-server:~/vaultwarden$ sudo docker-compose up -d
```

Et tout semble bien se passer. Ici aussi une petite vérification ne serait pas de trop...

```
julien@ionos-server:~/vaultwarden$ sudo docker ps
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS
NAMES
eef7e65bc713   vaultwarden/server:latest           "/start.sh"             15 seconds ago Up 14 seconds (hea
lth: starting) 80/tcp
vaultwarden
897a4f90e924   caddy:2                             "caddy run --config ..." 15 seconds ago Up 14 seconds
0.0.0.0:80->80/tcp, :::80->80/tcp, 0.0.0.0:443->443/tcp, :::443->443/tcp, 443/udp, 201
9/tcp caddy
julien@ionos-server:~/vaultwarden$
```

Tout semble correct ! Rendons nous sur l'URL dédiée à notre VPS pour voir ce que ça raconte :

Et cette installation est un succès tout ce qu'il y a de plus manifeste !

Configuration :

Comme la page web nous invite à créer un compte, nous allons donc nous y coller.

Créez un compte

Adresse électronique (requis)

Vous utiliserez votre adresse électronique pour vous connecter.

Nom

Comment doit-on vous appeler ?

Mot de passe principal (requis)

Important : Votre mot de passe principal ne peut pas être récupéré si vous l'oubliez ! 12 caractères minimum

Ressaisir le mot de passe principal (requis)

Indice du mot de passe principal

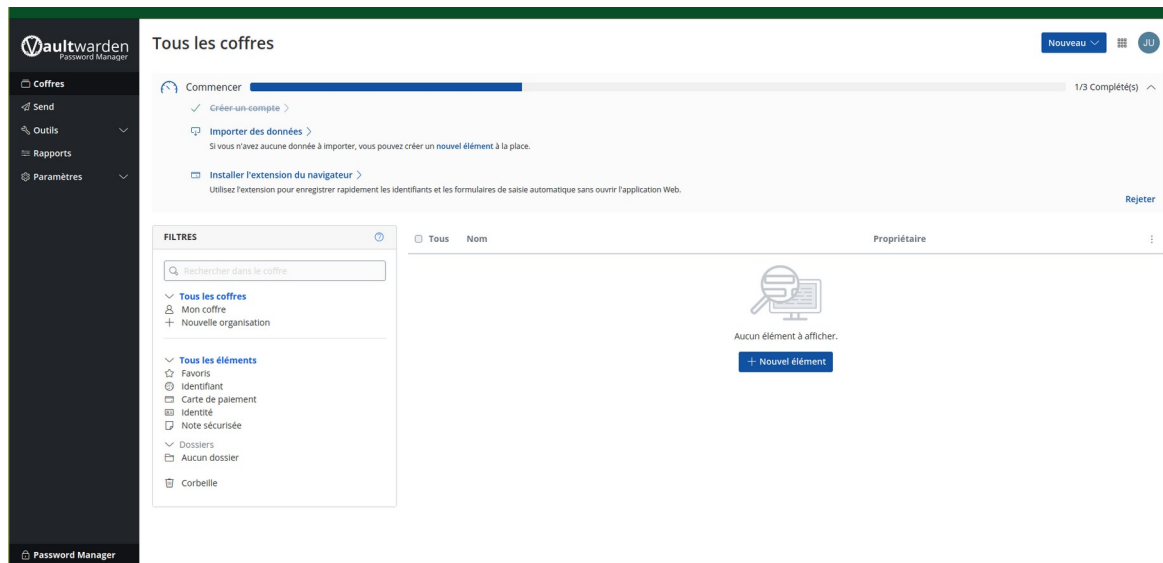
Un indice de mot de passe principal peut vous aider à vous souvenir de votre mot de passe si vous l'oubliez.

☒ Vérifier les brèches de données connues pour ce mot de passe

Créez un compte

Vous avez déjà un compte ? [Se connecter](#)

Un formulaire tout ce qu'il y a de plus classique, bien qu'une intention toute particulière soit bien évidemment donnée au master password, qui se doit d'être particulièrement robuste (longueur, complexité...). Une fois ce compte créé nous allons pouvoir nous connecter pour accéder au « coffre » et tester les fonctionnalités de Vaultwarden.



Nous voici enfin dans le vif du sujet.

- Utilisation
- Utilisation avec navigateur
- Config
- Sécurisation
- Sauvegarde (tester dump)
- Utilisation avec appli
- Consommation API ?

URL et SMTP !!!!!

Sécu !!!

Port SSH

SIGNUPS_ALLOWED : pour empêcher la création de nouveaux comptes, sinon n'importe quelle personne qui accède à l'interface de votre Vaultwarden pourrait s'inscrire dessus et l'utiliser. Autrement dit, on désactive les inscriptions.

- **DOMAIN** : pour déclarer le nom de domaine, ce qui permet à Vaultwarden d'avoir connaissance de son propre nom de domaine

- SIGNUPS_ALLOWED=false
- INVITATIONS_ALLOWED=false
- SHOW_PASSWORD_HINT=false

```
sudo docker-compose down
sudo docker-compose up -d
```


Conclusion

Sources

Sources WEB

- <https://zatoufly.fr/creer-son-serveur-vaultwarden-avec-docker/>
- <https://belginux.com/vaultwarden/>
- <https://rdr-it.com/deployer-vaultwarden-avec-docker/>
- <https://wiki-tech.io/SelfHosted/Bitwarden>
- <https://www.ionos.fr/digitalguide/serveur/configuration/installer-docker-sur-debian-11/>
- <https://docs.vultr.com/how-to-install-vaultwarden-on-ubuntu-20-04>

Sources papier

-

Sources humaines !

- William, lead dev, qui m'a confirmé que ma façon d'envisager la gestion des sauvegardes était viable
- Steven, admin réseau, qui m'a parlé des forces, faiblesses et enjeux autour des gestionnaires de mots de passe

Annexes (facultatives)

Lister ici les fichiers de configuration, les version d'OS et applications utilisés ou tout autre éléments que vous jugez utiles.

docker-compose.yml

```
version: '3'

services:
  vaultwarden:
    image: vaultwarden/server:latest
    container_name: vaultwarden
    restart: always
    environment:
      - WEBSOCKET_ENABLED=true
    volumes:
      - ./vw-data:/data

  caddy:
    image: caddy:2
    container_name: caddy
    restart: always
    ports:
      - 80:80
      - 443:443
    volumes:
      - ./Caddyfile:/etc/caddy/Caddyfile:ro
      - ./caddy-config:/config
      - ./caddy-data:/data
    environment:
      - DOMAIN={ici mon domaine}
      - EMAIL={ici mon adresse email pour le certificat SSL}
      - LOG_FILE=/data/access.log
```


Caddyfile

```
{ici mon domaine}:443 {  
    log {  
        level INFO  
        output file server.logs {  
            roll_size 10MB  
            roll_keep 10  
        }  
    }  
  
    # Get a cert by using the ACME HTTP-01 challenge.  
    tls {ici mon adresse email pour le certificat SSL}  
  
    encode gzip  
  
    # Headers to improve security.  
  
    header {  
        # Enable HSTS  
        Strict-Transport-Security "max-age=31536000;"  
  
        # Enable cross-site filter (XSS)  
        X-XSS-Protection "1; mode=block"  
  
        # Disallow the site to be rendered within a frame (clickjacking protection)  
        X-Frame-Options "DENY"  
  
        # Prevent search engines from indexing  
        X-Robots-Tag "none"  
  
        # Remove Caddy branding  
        -Server  
    }  
  
    # Redirect notifications to the WebSocket.  
    reverse_proxy /notifications/hub vaultwarden:3012  
  
    reverse_proxy vaultwarden:80 {  
        header_up X-Real-IP {ici mon domaine}  
    }  
}
```