KHOUTH
Elouane
BTS SIO spécialité SISR

Rapport de stage de BTS SIO SISR
au CROUS de Nantes Pays-de-la-Loire

Présentation des CROUS

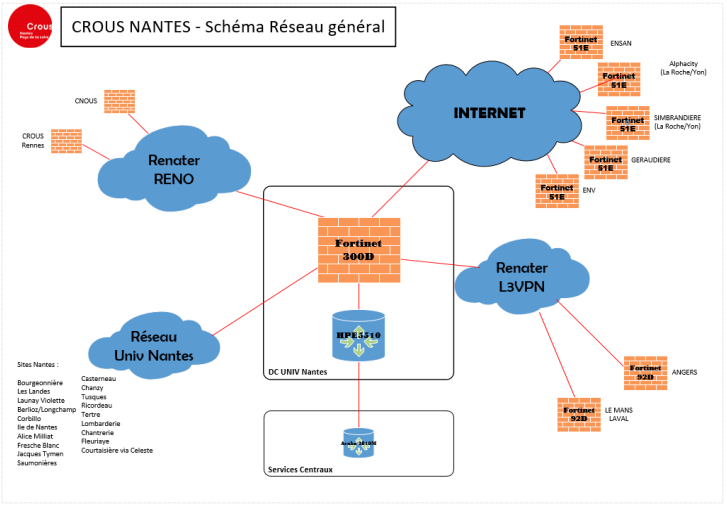
Les Centre Régionaux des Oeuvres Universitaires et Scolaires sont des établissements publics gérant les services de la vie étudiante dans l'enseignement supérieur. Bien que ce nom soit familier à un très grand nombre de personnes, beaucoup de gens ignorent encore la liste complète de leurs missions. Ils attribuent et gèrent les résidences universitaires destinées aux étudiants et permettent ainsi aux étudiants de se loger à moindre coûts. Pour aller avec ça, il y a également des plateformes de restauration diverses à disposition des étudiants comme les Restaurants Universitaires, les cafétérias, les Food-Trucks, le tout fonctionnant de la même manière et aux mêmes tarifs partout en France. Les CROUS octroient également des bourses et autres aides financières aux étudiants en difficultés pour garantir le bon déroulement de leurs études. Pour finir, les CROUS organisent des activités culturelles, sportives et sociales pour les étudiants afin qu'ils puissent s'épanouir et s'intégrer dans la vie étudiante.
Comme dit dans l'intitulé de l'organisme, les CROUS sont régionaux, il y a donc plusieurs centres en France, notamment à Amiens, Bordeaux, Reims, Poitiers, Nantes etc…

Présentation du CROUS de Nantes

Le CROUS de Nantes, au sein duquel j'ai eu la chance de pouvoir faire mon stage de première année, est très important en France car il gère un grand nombre de secteurs en plus de la région Pays de la Loire. Les villes de Nantes (Loire-Atlantique), Angers (Maine-Et-Loire), Le Mans (Sarthe), Laval (Mayenne) et La Roche-sur-Yon (Vendée) y sont directement rattachées. De plus, cet organisme précis sert de plateforme d'appel pour d'autres CROUS de l'Ouest comme Poitiers, Rennes ou Créteil.
Étant donné que le parc informatique du CROUS de Nantes s'étend sur toute la région Pays de la Loire, il est assez logique que l'infrastructure lui conviennent. Pour expliquer brièvement le fonctionnement et l'organisation de la structure, on a un grand Data Center situé sur le campus du Tertre à l'université de Nantes, qui fait le lien entre toutes les villes de la région à l'aide de liaisons fibres partout. Les locaux des services centraux sont partagés entre les téléconseiller.e.s, gestionnaires, secrétaires, directeurs des différents départements et aussi par l'équipe de Direction des Systèmes d'Informations, dont j'ai fait partie durant mon stage.
La DSI a pour but de déployer tous les services mis en place par le CNOUS (Centre National des Oeuvres Universitaires et Scolaires) et de veiller au bon fonctionnement de ces derniers en optimisant leur utilisation pour les rendre ergonomiques et économiques. Avec le grand nombre de services proposés, il y a énormément d'éléments à prendre en compte constamment, des problèmes de tous types à régler. La gestion des tickets d'incidents se partage sur trois niveaux, le premier représentant les petits incidents plus faciles à résoudre, et le troisième pris en charge par la DSI selon la gravité du ticket.

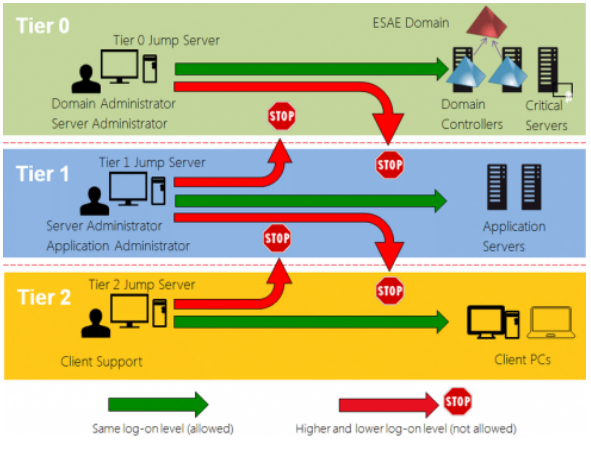
Explication de l'infrastructure

La structure informatique de l'organisme est assez complexe. Comme dit précédemment, elle est composée premièrement du Datacenter, mais aussi de deux baies de stockage situées dans les services centraux. Il y a plusieurs VLAN représentant chaque département du CROUS comme la DSI, le Management, le CESIR (niveau 1) etc…
Voici le schéma du réseau général de l'organisme :



Toute la structure est placée derrière un pare-feu Fortinet redondé et situé dans le Datacenter. Les différentes liaisons internet sont gérées par RENATER (Réseau National de Télécommunications pour la Technologie, l'Enseignement et la Recherche). Les services centraux ont une liaison directe avec une partie spécifique du Datacenter qui leur est dédiée. On y retrouve deux pare-feux, des switch redondés, un cluster de serveurs composé de 4 ESXi fonctionnant avec 30 To de stockage chacun.

La DSI intègre un prestataire en Cybersécurité ayant pour objectif la bonne mise en pratique des règles dictées par l'ANSSI et le RGPD. La gestion des habilitations est donc très réglementée et basée sur le principe de Tiering. Cette notion implique de créer plusieurs types de comptes ayant des habilitations différentes. Dans le cas du CROUS de Nantes, il y a les comptes Admin 0 ayant des accès sur les contrôleurs de domaine, les Admin 1 ayant accès aux serveurs et applications et les Admin 2 ayant accès aux postes seulement. Ayant eu du mal comprendre le fonctionnement au début, ce schéma disponible sur le Wiki interne de la structure explique parfaitement la notion.
Le Tier 0 est l'admin 0, le tier 1 l'admin 1 etc…



La spécificité de ce système est marquée par le fait que le Tier 0 ne puisse pas accéder au niveau du dessous etc… Il y a donc une obligation de créer un compte admin particulier pour chaque utilisateur habilité. Par exemple, à mon arrivée, on m'a attribué un compte utilisateur classique "ekhouth" mais aussi un compte admin 1 "adm1.ekhouth" avec lequel j'ai pu créer des serveurs et les configurer.
Cette notion est très importante et doit être respectée dans toutes les circonstances.

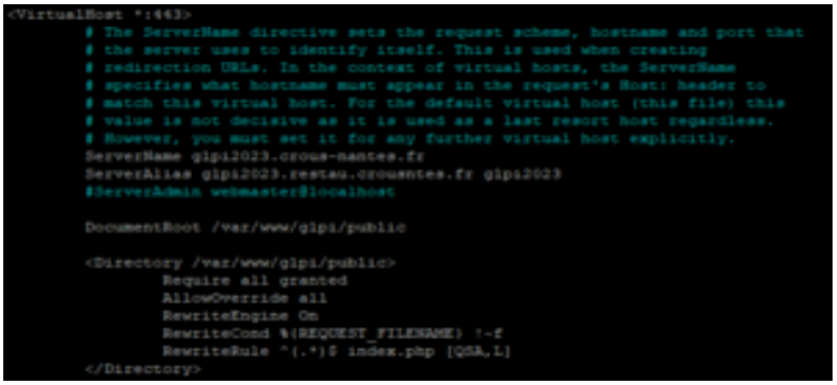
Présentation de la première mission : Installation de GLPI 10

Durant mon stage, j'ai participé à plusieurs (petits) projets, tous très intéressants. La première mission qui m'a été confiée a été l'installation de GLPI 10 de A à Z. Bien que la structure ait déjà GLPI en version 9 en service, les récentes mises à jour du logiciel ont en quelques sortes rendu dysfonctionnel FusionInventory, installé sur tous les équipements du parc informatique. FusionInventory est un logiciel distinct ayant pour mission d'envoyer toutes les informations des équipements sur lesquels il est installé au serveur pour centraliser toutes les données du parc. Depuis cette mise à jour, GLPI a développé son extension : l'Agent GLPI, qui a le même but mais un fonctionnement différent. L'agent GLPI communique directement avec l'inventaire natif de GLPI 10 alors que FusionInventory nécessitait un plugin supplémentaire.
L'agent GLPI a donc été déployé sur l'ensemble du réseau, mais la version de GLPI que la structure possédait ne permettait pas encore son utilisation. Ma mission a donc été d'installer GLPI 10.

La première chose à faire était de créer une machine virtuelle avec VMWare sur leur infrastructure. Cette machine virtuelle a été initialisée à l'aide d'un template de Debian 11 prévue pour les serveurs web, il y a donc un environnement LAMP installé dès la création de la machine, ce qui facilite la chose. L'accès à la machine s'effectue en SSH avec PuTTY avec, premièrement, l'adresse IP par défaut de la template. J'ai dû ensuite réserver une adresse IP libre sur IPAM (gestionnaire d'adresses IP) et configurer le serveur Apache. Bien que ce serveur web soit accessible uniquement depuis le réseau interne, il m'a tout de même été demandé de générer un certificat SSL pour accéder au site en HTTPS. Le Crous faisant partie de la catégorie "enseignement supérieur", l'organisme a la possibilité de créer ces certificats sans limite et gratuitement avec le site Sectigo. J'ai donc généré une clé à l'aide de OpenSSL et ai poursuivi les démarches pour valider le certificat. Pour aller avec, il a fallu créer une entrée DNS dans l'Active Directory et choisir un nom de domaine pour ce serveur. Étant donné que les trois AD de la structure sont redondés, il s'est passé un certain temps avant qu'ils soient tous à jour. Après ça, j'ai reconfiguré le serveur Apache en activant certains modules comme *rewrite* pour réécrire l'URL si besoin et *status* pour la supervision du service, ainsi que la page *ssl-default.conf.*

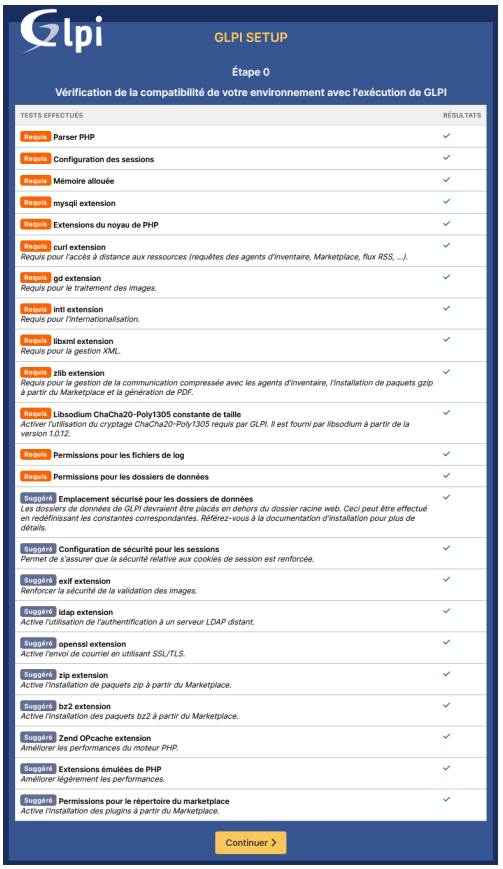
```
<VirtualHost *:80>
        ServerName glpi2023.crous-nantes.fr
        ServerAlias glpi2023.restau.crousntes.fr glpi2023

        DocumentRoot /var/www/glpi/
        # Redirection des requêtes HTTP vers HTTPS
        Redirect "/" "https://glpi2023.crous-nantes.fr/"
        Redirect "/glpi2023/" "https://glpi2023.crous-nantes.fr/"

        <Directory /var/www/glpi/>
                AllowOverride all
        </Directory>

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log "apache-access-%v %h %l %u %t \"%r\" %>s %b"
</VirtualHost>
```

Capture d'écran du Virtual Host HTTP du serveur Apache



Capture d'écran du Virtual Host HTTPS du serveur Apache

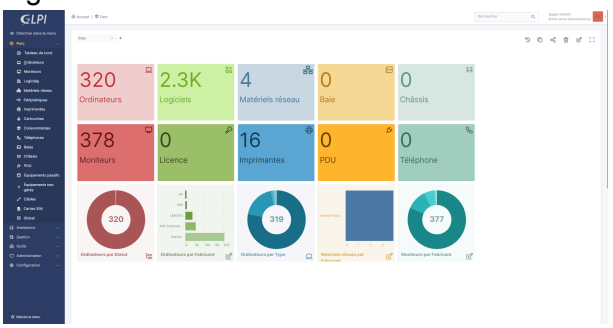
Cette page contient les informations de localisation de la clé privée et des clés publiques du certificat SSL. Après ça, le site était accessible en HTTPS sous deux noms de domaines différents pour rediriger les requêtes de l'ancien GLPI vers celui-ci, et j'ai pu commencer l'installation de GLPI 10.



L'installation de GLPI n'est pas très compliquée, il suffit juste de suivre les instructions et de corriger les éventuels problèmes survenant, comme la suppression du fichier *install.php* dans les fichiers de GLPI ou la création de liens symboliques pour les dossiers *files* et *config* pour sécuriser l'installation.

Une fois dans l'interface, j'ai pu ajouter les extensions GLPI Inventory et Data Injection (importation massive de données depuis l'ancienne version de GLPI) afin de mettre en service l'application. La configuration du plugin GLPI Inventory s'est faite sur la base de

FusionInventory de l'ancienne version, j'ai simplement reproduit ce qui avait déjà été fait pour faire en sortes que tout fonctionne correctement et que les différents appareils du parc informatique envoient leurs informations au serveur. Cela prend en compte les plages d'adresses IP des différents lieux (Angers, Nantes, Laval etc…) ainsi que les Active Directory par lesquels passent les requêtes SNMP. Une fois cette configuration faite, le serveur recevait bien toutes les informations de tous les appareils du parc.
Pour faciliter la transition de l'ancienne version à celle-ci, j'ai aussi dû importer tous les documents administratifs, les cartes sim, les téléphones, les contrats (et les trier) à l'aide de Data Injection. De cette manière, les personnes responsables de la gestion du parc informatique ont pu changer de serveur facilement et continuer leurs activités.



Capture d'écran de l'interface de GLPI après sa configuration

Quelques jours après avoir terminé cette installation, je me suis rendu compte que la version que j'avais installée n'était pas la dernière en date, j'ai donc dû tester le processus de mise à jour d'un service en production en créant une snapshot du serveur web, et en effectuant la mise à jour de GLPI, en corrigeant les nouveaux problèmes apparus etc…

Ce projet a tenu sur l'intégralité de ma période de stage, car certaines choses que je n'avais pas en tête n'allaient pas, et au fil des semaines, les gestionnaires de parc informatique me faisaient remonter quelques problèmes survenant dans certains cas particuliers afin que je les résolve.

Présentation de la deuxième mission : installation de Vaultwarden

Ma seconde grosse mission a été de mettre en place un nouveau gestionnaire de mots de passe pour la structure. Le CROUS de Nantes possède déjà une solution de ce type, mais l'inconvénient de celle-ci réside dans son interface peu ergonomique et trop compliquée pour les utilisateurs peu à l'aise avec les outils informatiques. Le but ici était de trouver un gestionnaire de mots de passe ayant les mêmes fonctionnalités que la solution déjà en place (KeePass 2.0 et KeePass XC) avec une interface plus parlante pour les utilisateurs. L'intérêt d'utiliser un gestionnaire de mot de passe est de pouvoir avoir un mot de passe différent pour chaque site/service utilisé. Cela incite les utilisateurs à se sensibiliser sur la sécurité de leurs postes et comptes pour diminuer les risques de fuites de données ou de piratage. Le RSSI de la structure avait déjà fait ses recherches avant mon arrivée, il m'a donc proposé de me renseigner sur Vaultwarden.

Premièrement, il faut d'abord préciser que Vaultwarden est un fork de Bitwarden. Bitwarden est un gestionnaire de mots de passe avec interface, extension de navigateur, application Android et IOS, logiciel Windows, MacOS, Linux qui possède toutes les caractéristiques de KeePass, à savoir :

→ Génération de mots de passe forts et aléatoires selon certains critères (longueur, nombre de caractères spéciaux/chiffres/majuscules) + passphrases

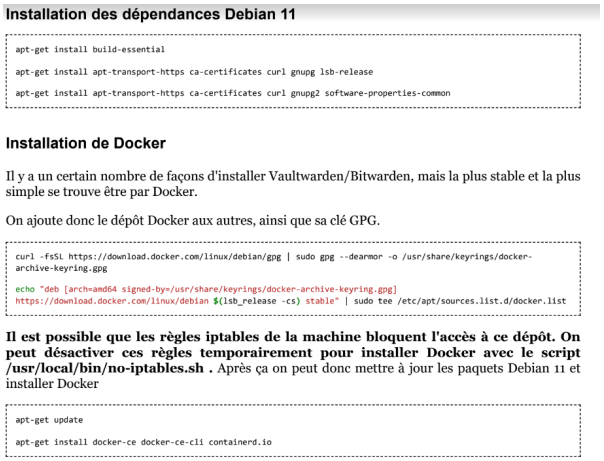
→ Partage de certains mots de passe avec d'autres utilisateurs

→ Tri des mots de passe en fonction des services

Même si Vaultwarden présente exactement les mêmes fonctionnalités que Bitwarden, celui-ci est plus sécurisé car il est programmé en Rust, un langage de programmation développé par Mozilla et connu pour être très sûr.

Il existe un certain nombre de manières d'installer Vaultwarden, j'ai cependant choisi une façon plus "simple" en utilisant Docker. Docker permet de lancer des applications dans des conteneurs et de les gérer plus facilement. Il utilise en quelque sorte des machines virtuelles indépendantes avec leurs dépendances pour pouvoir se servir de services plus facilement. Dans ce cas là, Vaultwarden tourne dans un conteneur qui possède toutes les caractéristiques d'un serveur web linux avec donc Apache, SQLite etc…

Pour utiliser Docker correctement, il a fallu installer toutes les dépendances nécessaires.



Installation des dépendances Debian 11

```
apt-get install build-essential

apt-get install apt-transport-https ca-certificates curl gnupg lsb-release

apt-get install apt-transport-https ca-certificates curl gnupg2 software-properties-common
```

Installation de Docker

Il y a un certain nombre de façons d'installer Vaultwarden/Bitwarden, mais la plus stable et la plus simple se trouve être par Docker.

On ajoute donc le dépôt Docker aux autres, ainsi que sa clé GPG.

```
curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg]
https://download.docker.com/linux/debian $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list
```

Il est possible que les règles iptables de la machine bloquent l'accès à ce dépôt. On peut désactiver ces règles temporairement pour installer Docker avec le script /usr/local/bin/no-iptables.sh . Après ça on peut donc mettre à jour les paquets Debian 11 et installer Docker

```
apt-get update

apt-get install docker-ce docker-ce-cli containerd.io
```

Extrait de ma page de guide d'installation de Vaultwarden sur le wiki interne

La dernière chose à faire après l'installation correcte de Docker était la configuration d'Apache pour que le service puisse être accessible en HTTPS car Vaultwarden n'autorise pas une utilisation non sécurisée. De plus, le service doit être placé derrière un reverse-proxy, et Apache possède cette fonctionnalité.

La machine virtuelle hébergeant le service final possède donc tout ce qu'il faut pour permettre l'utilisation de Vaultwarden.

Les commandes d'utilisation de Docker étant complètes, j'ai pu adapter le conteneur de Vaultwarden en fonction des besoins du CROUS. Le service doit être accessible sur le port 80 qui doit rediriger vers le port 443 de la même manière que GLPI pour s'assurer que le site sera utilisé de manière sécurisée pour tous les utilisateurs. La commande de lancement du conteneur ressemble à ça :

```
docker run -d --name vaultwarden \
           -e ADMIN_TOKEN=Le_Token_Admin \
           -v /vw-data/:/data/ \
           -e WEBSOCKET_ENABLED=true \
           -p 127.0.0.1:8080:80 \
           -p 127.0.0.1:3012:3012 \
           --restart always vaultwarden/server:latest
```

La mention "ADMIN_TOKEN" définit un jeton administrateur utilisé pour accéder au panel d'administration du serveur Vaultwarden depuis l'adresse *https://vaultwarden.crous-nantes.fr/**admin*** (l'adresse est fictive). Ce jeton peut être généré grâce aux outils de OpenSSL :

→ *openssl rand -base64 48*

Pour expliquer brièvement les options utilisées :

-d : pour lancer le container en arrière plan, en Daemon

--name : le nom du container

-v : le volume monté pour les fichiers de configuration du serveur

-e WEBSOCKET_ENABLE : pour utiliser un Websocket et activer les notifications

-p : pour rediriger les flux du port 8080 serveur vers le port 80 du container. (Le port 3012 est utilisé pour le websocket)

--restart always : force le conteneur à se redémarrer en cas de problème

Pour faciliter l'utilisation et la configuration du serveur, j'ai créé un script permettant de lancer, réinitialiser ou redémarrer le service.

Réinitialisation :

```
docker stop /vaultwarden
docker rm /vaultwarden
systemctl restart docker
docker run -d --name vaultwarden -e ADMIN_TOKEN=Le_Token_Admin -v /vw-data/:/data/ -e WEBSOCKET_ENABLED=true -p
127.0.0.1:8080:80 -p 127.0.0.1:3012:3012 --restart always vaultwarden/server:latest
```

Redémarrage:

```
CONTAINER_ID=$(echo $(docker ps | grep vaultwarden | awk 'NR==1 {print $1}'))
docker stop $CONTAINER_ID
systemctl restart docker
docker start $CONTAINER_ID
```

La variable CONTAINER_ID récupère l'ID du conteneur nommé Vaultwarden déjà en lancement.

Il est important de redémarrer le service Docker entre chaque arrêt et lancement du conteneur car, pour des raisons qui me sont encore obscures jusqu'ici, une erreur survient très fréquemment lorsqu'il n'est pas redémarré, c'est pour ça que j'ai rajouté *systemctl restart docker* dans les deux scripts. Ceci-dit, il est tout à fait possible que ce problème ne survienne que sur l'infrastructure du CROUS.

Une fois tout ça mis en place, j'ai pu démarrer la réelle configuration du serveur Vaultwarden via le panel administrateur et notamment mettre en place le service SMTP pour que les utilisateurs puissent se connecter avec leur adresse mail de la structure (...@crous-nantes.fr) et qu'ils puissent confirmer leur identité, être notifiés des actions effectuées sur leur compte etc…

Le CROUS de Nantes utilise Microsoft Exchange pour l'envoi de mails mais aussi un relais SMTP pour automatiser les envois depuis les imprimantes ou services nécessitant une telle fonctionnalité. La configuration du service SMTP pour Vaultwarden a donc été assez simple, il a suffit de rentrer l'adresse du serveur relais SMTP, et comme ces envois ne sont malheureusement pas sécurisés pour le moment, aucun identifiant de connexion au relais n'a été nécessaire. Après ça, les utilisateurs de Vaultwarden ont pu recevoir des mails du service. Il y a toutefois eu un problème qui est survenu. Après un certain temps, le service SMTP de l'application ne fonctionnait plus et arrêtait donc d'envoyer les mails aux utilisateurs, mais se remettait en fonctionnement après un redémarrage de docker. Je me suis donc familiarisé avec Cron, un planificateur de tâches Linux pour pouvoir redémarrer Docker automatiquement toutes les 6 heures. Après ça, l'envoi de mails a pu s'effectuer en continu sans problème.

Vaultwarden n'étant pas un service français, tous les mails envoyés automatiquement étaient donc rédigés en anglais par défaut et n'ayant, à ce moment là, pas trouvé de traductions officielles pour personnaliser ces templates, j'ai été contraint de traduire les 30

templates de mails par moi-même pour ensuite les importer dans les fichiers de Vaultwarden. Trouver le moyen de faire ceci m'a pris quelques heures de recherche sur internet pour, au final, trouver la solution qui est très simple. Il suffisait de créer un dossier templates/email dans la partition réservée aux fichiers du conteneur et d'y ajouter les fichiers traduits.

En tant qu'autre élément de personnalisation, il m'a été demandé de modifier l'image d'accueil du site pour la remplacer par le logo du CROUS de Nantes afin que les utilisateurs soient sûrs d'être arrivés sur le bon site internet. Cette étape m'a pris énormément de temps puisqu'il fallait d'abord que je comprenne comment fonctionnaient réellement Docker et ses conteneurs, comment accéder aux fichiers des conteneurs et comment les modifier. J'ai trouvé grâce à internet la commande *docker cp* qui permet de transférer des fichiers depuis la machine hôte vers le conteneur et vice versa en utilisant l'ID du conteneur. J'ai donc copié l'image voulue vers le conteneur dans le dossier web-vault/images, et remplacé son nom par le nom de l'image déjà en place pour ne pas avoir à modifier au fichier CSS le plus possible. Cependant, la résolution de cette image étant définie dans le fichier CSS principale, j'ai quand même dû rechercher la définition de sa résolution dans les centaines de lignes que composent ce fichier pour les adapter aux dimensions de l'image finale. Après ça, un simple redémarrage du conteneur suivi de la suppression du cache du navigateur et j'ai pu observer que l'image d'accueil de Vaultwarden avait bien changé. Étant donné que c'est le contenu du conteneur qui a été modifié, ce changement sera effacé à chaque fois que le conteneur sera redémarré malheureusement, mais il n'y avait aucun autre moyen de faire ce changement.

Une fois le service opérationnel en phase de test, j'ai dû présenter à tous les membres de la DSI cette solution avec un Powerpoint et une démonstration de son utilisation. Une fois la solution acceptée, j'ai pu démarrer son installation en production pour que le site soit accessible depuis toute la structure.

Après deux semaines de fonctionnement, la supervision du service a remonté un problème de surcharge du stockage de la machine virtuelle hébergeant Vaultwarden. Après quelques temps de recherche, je me suis aperçu que le dossier contenant les logs de tout le système était saturé grâce à la commande *df*. En fouillant ce dossier et en regardant la taille des fichiers présents avec la commande *du -sh*, j'ai compris que les fichiers *syslog* et *user.log* ainsi que leurs archives étaient la source de ce problème de surcharge car à eux seuls, ils occupaient 95% du stockage. En regardant le contenu de ces fichiers en temps réel, j'ai pu constater qu'à chaque minute, ils recevaient 15 lignes d'erreurs venant d'Audit, un service mis en place sur toutes les machines virtuelles linux pour scanner toutes les activités de logs du système pour les renvoyer vers *syslog.* Après plusieurs jours de recherche et de test, j'ai compris que ces logs venaient de Docker, plus particulièrement du conteneur de Vaultwarden. Malheureusement, je n'ai pas trouvé de "vraie solution", j'ai donc premièrement créé un script permettant de supprimer les logs et ses archives lorsque l'espace disque était occupé à plus de 85% :

```
#!/bin/bash

limite=85
usage=$(df -h --output=pcent / | awk 'NR==2 {print $1}' | tr -d '%')

if [ $usage -gt $limite ]; then
        echo -n > /var/log/syslog
        echo -n > /var/log/user.log
        if [ -f "/var/log/syslog.1" ]; then
                rm -f /var/log/syslog.1
        fi
        if [ -f "/var/log/user.log.1" ]; then
                rm -f /var/log/user.log.1
        fi
fi
```

Il a ensuite suffit de planifier l'exécution de ce script une fois par jour grâce à Cron.

Cette solution étant temporaire, j'ai donc ensuite empêché Audit d'écrire dans *syslog,* et augmenté la fréquence d'archivage des logs, ce qui a réglé le problème d'une meilleure manière.
Le service est donc opérationnel et optimisé pour durer dans le temps.
Pour faciliter son utilisation et ses modifications, j'ai rédigé un guide d'utilisation pour les utilisateurs avec deux annexes pour les fonctionnalités spécifiques destinées aux administrateurs, mais aussi une page sur le wiki interne expliquant son installation, la procédure de mise à jour, de personnalisation etc… De cette manière, même après mon départ, la DSI pourra continuer de surveiller Vaultwarden et le faire perdurer.

Présentation des autres petites missions :

Déplacement d'un NAS de sauvegarde :

Durant ma première semaine, il m'a été demandé de déplacer le NAS de sauvegarde de la DSI dans une des baies de stockage de la structure. Ce NAS était jusqu'ici installé dans un bureau mais le but était donc de le placer avec le reste de l'infrastructure de stockage. Il possède 4 interfaces réseau, ce qui permet de lui attribuer plusieurs adresses IP. Il fallait premièrement lui définir une autre adresse IP sur une deuxième interface pour pouvoir tout de même accéder à son interface pendant la procédure. J'ai encore une fois choisi une adresse libre via IPAM. Après ça, j'ai donc vissé le support du NAS sur la baie de stockage, et connecté le port de la première interface et de la deuxième sur deux ports du cluster de switch en prenant soin de bien retenir le numéro de ces ports pour pouvoir ensuite les configurer dans l'interface du switch. Ces switches n'étant pas de chez Cisco, j'ai dû me familiariser avec le langage utilisé par les appareils Aruba, faits par HP. Il fonctionne de la même manière que celui de Cisco, cependant la syntaxe et les applets ne sont pas les mêmes. J'ai donc déplacé les ports correspondant au NAS du VLAN 'Jail' (sert à bloquer tous les nouveaux appareils par défaut) vers un VLAN particulier de cette manière :

```
#conf t
#interface <numéro_du_port>
#untagged VLAN_01  (les numéros de VLAN sont fictifs)
#untagged VLAN_07
#write memory
```

Cette manipulation peut s'effectuer sur le CLI du switch mais aussi sur l'interface graphique disponible sur l'intranet de la structure si l'on possède les droits nécessaires.
Tout semblait bien configuré, mais l'accès à l'interface web du NAS était toujours impossible. Après plusieurs heures de tests et de vérifications, nous avons eu l'idée de vérifier que le pare-feu ne bloquait pas les connexions entrantes dans cette situation là, et il s'est avéré que c'était le cas. Une fois ce problème réglé, nous avons pu accéder correctement aux données de l'appareil pour continuer à sauvegarder tous les documents de la DSI.

Résolution d'un problème de fuite mémoire sur H3 :

H3 est le service fonctionnant avec Tomcat et permettant aux étudiants de réserver leur logement étudiant, il est donc important qu'il fonctionne correctement pour leur permettre de payer sereinement et de s'assurer qu'ils disposeront d'un logement pour leurs études. Lors de ma troisième semaine, un problème d'ordre national est apparu sur ce service. D'après les différents appels des utilisateurs, le site semblait cesser de fonctionner très rapidement, même après des redémarrages manuels. Après recherches, les membres de la DSI sont

arrivés à la conclusion que cela venait des fuites de mémoire créées par le code source Java du logiciel. Bien que je ne sois pas réellement qualifié pour résoudre ce genre de problèmes, j'ai tout de même essayé d'apporter une solution temporaire pour permettre aux étudiants de faire leurs démarches. J'ai simplement automatisé le redémarrage de Tomcat en fonction du taux d'utilisation de la mémoire du service à l'aide d'un script Bash. Il a fallu que je trouve un moyen d'accéder à ces informations et que je réussisse à les utiliser dans un script. En cherchant un peu, je me suis aperçu que je pouvais exporter ces informations en XML, ce qui pouvait s'avérer utile. J'ai donc utilisé des librairies de parsing XML pour arriver à mes fins.
Voici le script final :

```
curl -s -u "[illegible]" http://h3pp.crous-nantes.fr:8080/manager/status/manager/status?XML=true >
file.xml

usageUsed=$(echo $(xmllint --xpath "string(//memorypool[@name='G1 Old Gen']/@usageUsed)" file.xml))
usageMax=$(echo $(xmllint --xpath "string(//memorypool[@name='G1 Old Gen']/@usageMax)" file.xml))

if ((usageUsed < (usageMax * 60 / 100))); then
        exit 0
else
        exit 99
fi
```

Pour expliquer rapidement la structure de ce script :

La commande *curl* permet de télécharger un document, son option *-u* envoie des identifiants de connexion si besoin (les informations du script final sont masquées sur la capture d'écran). On redirige donc ces informations vers un fichier puis on crée des variables qui vont prendre la valeur maximum de la mémoire et la valeur de l'utilisation actuelle à l'aide de *xmllint* qui sert à isoler des informations relatives à la chaîne de caractère passée en paramètre. La suite du script est simplement un test pour vérifier l'utilisation, il renvoie le code 0 si l'utilisation ne dépasse pas 60%, et le code 99 dans le cas échéant.
La suite du processus s'effectue sur Monit : on doit lui demander de redémarrer le service Tomcat si le code de retour du script est égal à 99. Pour ce faire, il faut simplement aller dans le fichier de configuration de Monit, et écrire ces lignes :

```
check program testUsageMemoire with path "/bin/bash -c '/etc/monit/testUsageMemoire.sh'"
with timeout 300 seconds
if status = 99 then restart
```

Après ça, H3 était de nouveau fonctionnel, bien que cette solution ne convienne évidemment pas sur le long terme. J'ai simplement voulu essayer d'aider la DSI pour qu'ils puissent se concentrer sur le cœur du problème pour arriver à une solution définitive.

Gestion du processus de sauvegardes des bases de données

Le CROUS de Nantes utilise le logiciel Veeam Backup pour sauvegarder leur infrastructure régulièrement. Cette solution a été mise en place par un prestataire et fonctionne correctement. Ils ont mis en place trois types de sauvegarde : bronze, argent et or. La sauvegarde bronze s'effectue quotidiennement, l'argent plus régulièrement et l'or encore plus régulièrement. Ces tag sont attribués aux services en fonction de leur importance et de la criticité en cas de perte de données.
Bien que tout fonctionne correctement, il y avait un point que la DSI voulait améliorer. Ils voulaient pouvoir sauvegarder les dump de bases de données contenues dans les machines virtuelles sur le serveur de sauvegarde pour pouvoir les restaurer en cas de problème. J'ai donc fait des recherches pour être capable de créer un script bash de dump. Je me suis

servi de la commande *mysqldump* avec les options -u et -p qui permettent de rentrer les informations de connexion à la base de données. J'ai ensuite redirigé la sortie de la commande vers un fichier qui sera sauvegardé pendant le backup.
Pour mieux comprendre le déroulement de la sauvegarde, il faut d'abord expliquer les notions de pre-freeze et post-thaw. Un script pre-freeze va s'exécuter avant que la machine virtuelle soit gelée dans son état actuel dans le but d'être sauvegardée, et un script post-thaw va s'exécuter une fois que la machine a été sauvegardée et remise dans un état fonctionnel. Dans notre cas, le script pre-freeze va contenir la commande de dump pour produire un fichier contenant le dump de la base de données, et le script post-thaw va effacer ce dump de la machine. De cette façon, le dump sera présent dans la sauvegarde de la machine mais pas dans la machine en fonctionnement.
Les seuls problèmes auxquels j'ai été confronté concernent l'utilisation de Veeam Backup, qui n'est pas un logiciel très simple à prendre en main. De plus, n'étant que stagiaire dans la structure, je ne possédais aucun droit d'accès au logiciel car les sauvegardes sont très sensibles.

En conclusion de ce rapport de stage au sein du CROUS de Nantes Pays-de-la-Loire, il est indéniable que cette expérience va jouer un rôle majeur dans le développement de mes compétences informatiques et dans la consolidation de mes valeurs professionnelles. En effet, au cours de ces dernières semaines, j'ai eu l'opportunité de plonger au cœur d'un environnement informatique complexe et exigeant dans lequel le travail d'équipe est indispensable.
D'un point de vue technique, ce stage m'a permis d'appliquer des connaissances acquises au cours de ma formation en BTS SIO, mais aussi mes connaissances personnelles en grande partie et ce de manière concrète. J'ai pu mettre en œuvre des solutions de maintenance et de gestion au sein d'un réseau très complexe, tout en travaillant aux côtés de professionnels aguerris. La résolution de problèmes en temps réel m'a poussé à aiguiser mes capacités d'analyse ainsi que ma réactivité. Ces compétences étant essentielles dans ce domaine, je suis heureux d'avoir pu évoluer sur ce point.
Au-delà des compétences techniques, ce stage m'a également sensibilisé aux différentes notions de cybersécurité vues pendant la formation et à la manière dont elles sont appliquées au quotidien dans les entreprises. Le fait que la structure manipule une quantité faramineuse de données d'étudiants pousse à être vigilant en permanence et tout le personnel du service informatique du CROUS en est conscient et fait son maximum pour prioriser la sécurité, le respect des règles de l'ANSSI et du RGPD.
En somme, ce stage au CROUS de Nantes a été une expérience formatrice et enrichissante. Il a contribué à l'élargissement de mon champ de compétences techniques et au renforcement de mes valeurs professionnelles, et ce malgré sa courte durée. Je tiens à exprimer ma gratitude envers l'ensemble de l'équipe qui m'a accueilli et encadré avec beaucoup de bienveillance et de patience. Grâce à ce stage et à eux, je suis désormais encore plus sûr de vouloir faire de l'informatique mon métier.