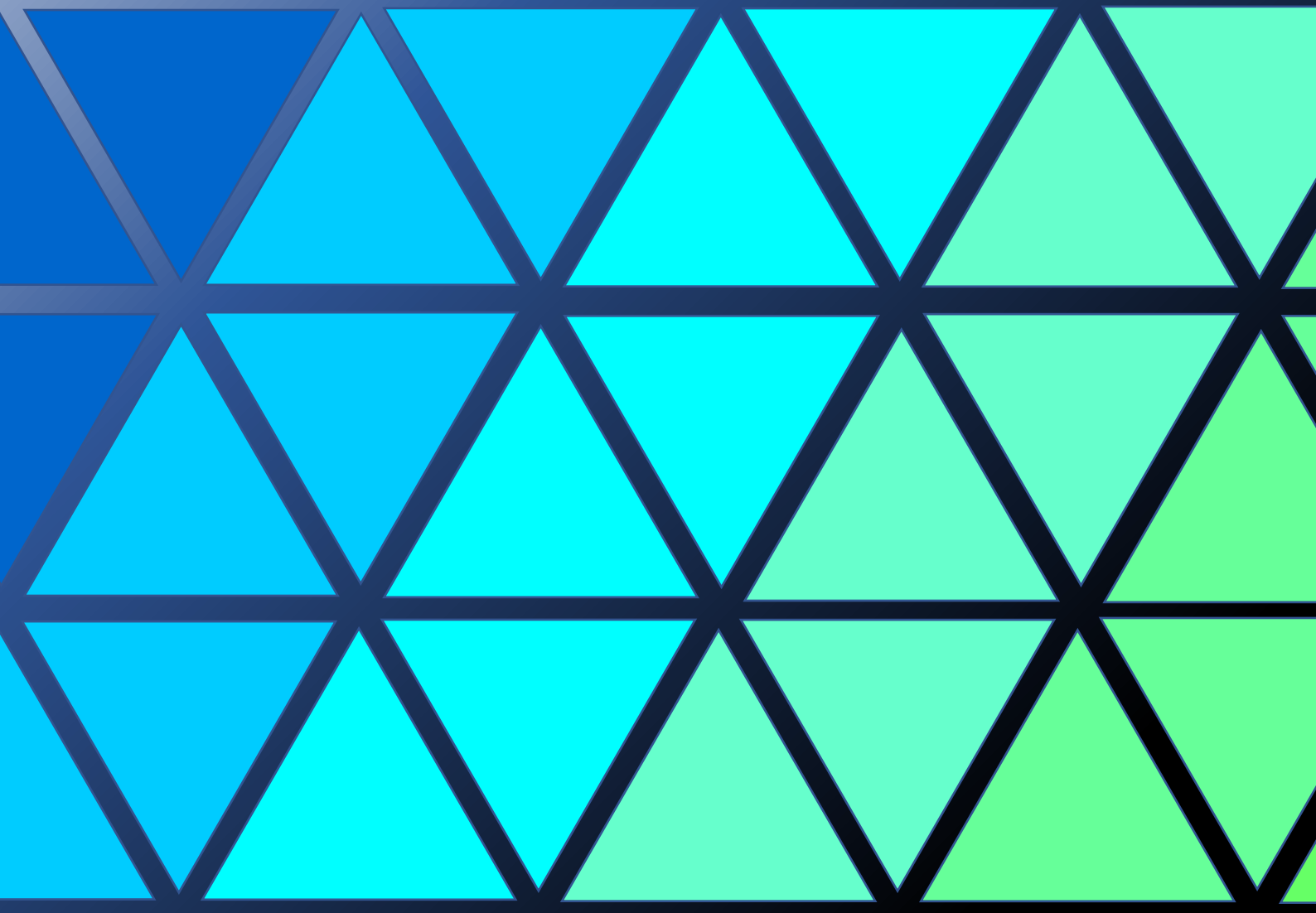
Hardening Serveur Web

Document Technique

Avril 2021



Rédacteur : Soulard Morgan

Sommaire

Introduction

Problématique :

Mon projet a pour cible les entreprises qui ne serais pas orienté IT mais qui souhaiterai quand même la mise en place d'un site web pour gagner en visibilité.

En tenant compte des moyens financier des différentes entreprises j'ai élaborer deux scénarios, un pour les entreprises disposant de peu de moyen et un autre destiné à l'entreprise disposant de plus de moyen financier dans le but d'acquérir un service web plus sécurisé avec plus de disponibilité.

Scénario 1 :

Le serveur web repose sur une unique machine disposant de docker, l'infrastructure du réseau docker et composé de 4 machines, 1 reverse proxy et 3 serveurs Nginx load-balancé.

Scénario 2 :

Le serveur web repose sur 4 machines différentes. 1 reverse proxy et 3 serveurs web.

Configuration Logiciel :

Au niveau logiciel les deux scénarios reposent sur les mêmes bases :

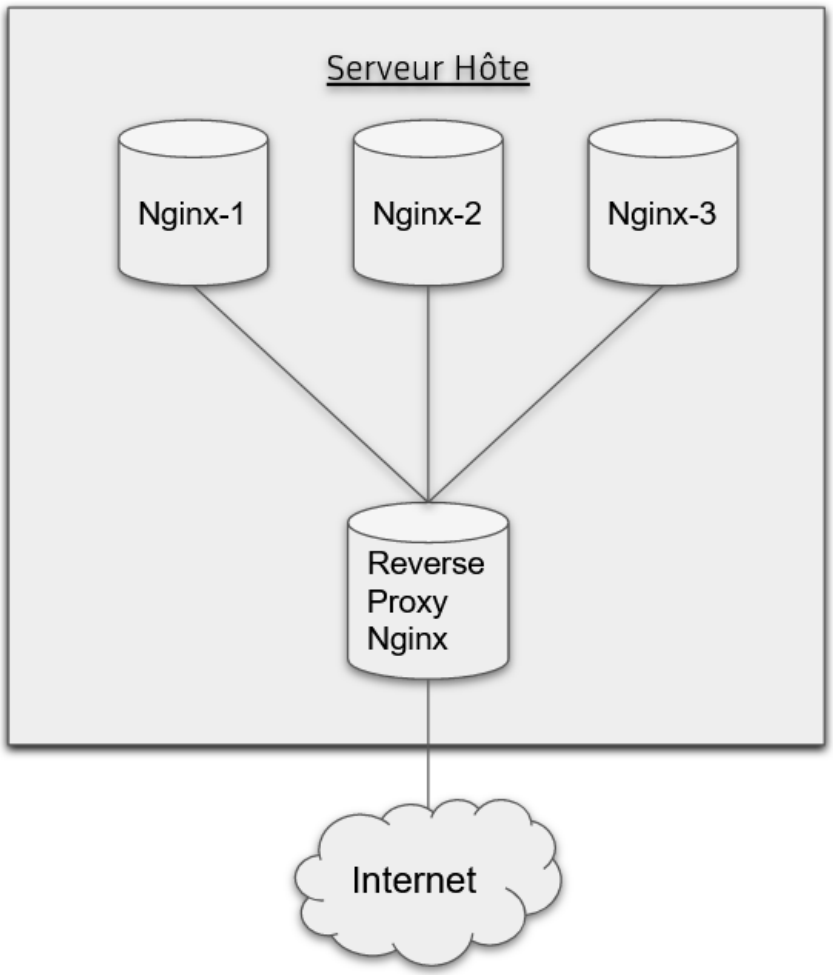
- Configuration solide du différent pare-feu de chaque serveur
- Utilisation d'un utilisateur avec des droit limité
- Mise en place d'un contrôle d'intégrité des fichier grâce à l'HIDS Aide
- Configuration hardené du/des serveur Nginx

Infrastructure

Pour une question de moyen j'ai opté pour le scénario N°1 qui permet tout de même la mise en place d'un serveur web hardené.

L'infrastructure repose donc sur le principe de containérisation que j'ai mis en place grâce à Docker & Docker compose. Un des avantages de cette solution et le déploiement ultra rapide et la possibilité de mettre en marche/arrêt toute l'infrastructure avec une simple commande.

De plus grâce au volumes docker il est possible de conserver les fichiers souhaiter même une fois l'infrastructure éteinte. Ce qui présente un clair intérêt pour les fichiers de configuration et les données du serveur web.



Solutions

Haute Disponibilité :

La mise en place de load-balancing permet d'assurer la disponibilité du serveur web. Le reverse proxy permet la répartition de charge, dans le cas où l'un des trois serveur Nginx viendrai à ne plus répondre pour quelque raison que ce soit le serveur web continuerai à fonctionner grâce au deux autre serveur Nginx.

Certificat TLS :

Dans le but d'éviter qu'une personne mal intentionnée ne puisse capturer le trafic entre le serveur et c'est différent client le trafic et chiffrer à l'aide de certificat TLS.

Personnalisation :

Par défaut les serveurs Nginx communiquent des informations sur les différentes pages d'erreur pour éviter cela les pages d'erreurs on était personnalisé.

Cacher la version :

Au cours des échange entre les clients et le serveur, le serveur communique par défaut dans l'entête http son numéro de version. Le fait de communiquer le numéro de version peut permettre à une personne malintentionnée de cibler une attaque dans le cas où la dites version serai vulnérable. La configuration des serveurs a donc été mise en place dans le but que le numéro de version ne soit pas communiqué.

Utilisateur Dédié :

Les différents services utilisés sont exécutés depuis un utilisateur dont les droits sont restreints, dans le cas où les services serait compromis l'accès aux autres services de l'hôte deviendrai difficilement accessible.

Pare-feu :

Le pare feu de la machine hôte permet de bloquer toute tentative de connexion qui ne serai pas sur l'un des ports nécessaires au bon fonctionnement du serveur ce qui permet de réduire la surface d'attaque du serveur web.

HIDS Aide :

L'HIDS Aide a était mise en place pour surveiller les données de la machine hôte et tout particulièrement dans les volumes de partage docker. Un contrôle des volumes et effectuer toutes les 10 minutes puis stocker dans des fichiers de logs pour permettre de les consulter au besoin.

Idées

Au cours de mon projet toute les mesures de sécurité non pas pu être mise en place en voici quelqu'une qui pourrait permettre de renforce la sécurité du serveur.

FailOver IP :

La mise en place d'un FailOver IP permettra de bannir une adresse ip depuis laquelle le serveur reçoit un nombre bizarrement élever de requête dans le but d'éviter les attaques par brute force ou le déni de service du serveur.

Keep Alived :

Pour éviter un SPOF (Single Point Of Failure) la mise en place d'un second reverse proxy est possible. Il serait capable de prendre le relai si le reverse proxy principal venais à défaillir.

Shell Restreint :

L'utilisateur dédiée permet de restreindre l'accès au différent processus dans le cas où l'utilisateur viendrai à être compromis néanmoins il est tout à fait possible de limiter encore plus l'accès de cet utilisateur en configurant un Shell restreint qui permettrai à cet utilisateur de pouvoir uniquement démarrer ou arrêter le serveur web.

Notifications :

Il pourrait être intéressant d'être alerter pas un canal sécuriser comme Télégramme par le serveur dans le cas d'un arrêt du serveur, d'une détection d'intrusion ou tout autre évènement qui aurait lieu sur le serveur.