|  |  |
| --- | --- |
| **FIEDORCZUlkkK Dominik**  **MIRICA Constantin**  **SHATSKIY Filipp** | **Coordinateur : VAN DEN SCHRIECK Virginie**  **E.P.H.E.C. Louvain-la-Neuve**  **Groupe 20 - 2TL2**  **16.05.2018** |

Rapport Sécurité

# **Sécurité VPS**

## **Failles envisagées**

Suite à notre expérience, nous avons envisagé différents points pouvant affecter le bon fonctionnement de nos services :

1. La découverte des failles
2. Trop de droits pour certains utilisateurs
3. L’accès au compte root via SSH
4. La connexion par un simple mot de passe aux différents comptes
5. La découverte de nos clés par « brute force »

*Serveur DNS :*

1. *Cache poisoning : lorsqu'un serveur DNS est obligé d'interroger un autre serveur DNS pour obtenir l'adresse IP d'un nom de domaine faisant l'objet d'une requête*
2. *Attaques DDoS : le but de l'attaque est de bloquer le fonctionnement de la ressource Web (déni de service total)*

*DB :*

*Injections SQL : Changer la requête SQL en cours en injectant une requête non prévue*

*Mail :*

1. *Phishing : Envoyer des mails avec le domaine de l’entreprise « @domaine.be » pour tromper les internautes et la plupart du temps en les dirigeants vers un formulaire où ils complètent leurs informations personnelles*
2. *Spam : envoyer énormément de mails peut bloquer relativement vite une boite mail*

## **Solutions implémentées**

Afin de résoudre ces problèmes, nous avons mis en place une série de systèmes destinés à remédier à ces failles :

1. Mises à jour systématiques de Ubuntu
2. Bien configurer les droits des différents utilisateurs
3. Couper l’accès au compte root via SSH
4. Nous avons activé une connexion par clé SSH et avons coupé l’accès par mot de passe. *Cette configuration est faite grâce à la commande « ssh-keygen » qui crée une clé publique et privée. Il faut copier la clé publique sur un serveur distant dans le fichier « authorized-keys ». Pour couper l’accès par mot de passe nous avons modifié le fichier « sshd\_config ».*
5. Nous avons utilisé la méthode Fail2Ban, qui empêche les attaques de types brute force et qui nous envoie des notifications par mail des attaques. *Celui-ci est un framework de prévention contre les intrusions, qui bloque les adresse IP qui se connectent trop de fois. Cette vérification est faite par une quantification du nombre d’échecs de connexion.*

# **Sécurité Web**

Pour une sécurité optimale mais aussi par nécessité, nous avons utilisé le certificat HTTPS et Reverse Proxy. Le HTTPS et un protocole qui sert à crypter les communications qui rentre et qui sorte de la page web.

***DB***

*Nous avons utilisé un mot de passe très complexe.*

# **Sécurité DNS**

Pour vérifier les connexions mais aussi les essais de connexion sur le serveur, nous envisageons l’utilisation d’une extension de Fail2Ban. Nous comptons aussi ajouter DNSsec pour une meilleure protection afin d’éviter des attaques comme : « phishing »… qui falsifient les réponses donnés par le DNS.

# **Sécurité Mail**

Un certificat SSL est mis en place, et il serait également judicieux d’utiliser un mot de passe crypté et non pas « plain text ».

# **Sécurité VOIP**

Pour Fail2Ban, sur les 3 VPS nous les avons réglés pour bloquer l’adresse IP après 3 essais non réussis de connexion. L’adresse IP est débloquée après 1h. Un mail est envoyé au propriétaire pour savoir l’adresse IP en question. Nous considérons que 3 fois c’est assez pour se rappeler un mot de passe.

***Mise en place d’un Pare-feu***

*Nous sommes en train de comprendre le fonctionnement du Firewall et nous voulons l’implémenter au plus vite.*