

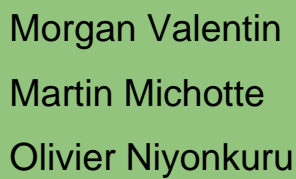


06/05/2020

Analyse Sécurité

Groupe 2TL2-5

Version 2



Morgan Valentin
Martin Michotte
Olivier Niyonkuru

Table des matières :

Risques encourus par nos VPS et notre infrastructure Docker :	2
Au niveau du VPS, on peut avoir :	2
Au niveau de Docker, on peut avoir :	2
Contre-mesure mises en place :	2
Au niveau du VPS :	2
Au niveau de docker :	2
Risques encourus par chacun des services déployés :	3
Services DNS :	3
Services WEB :	3
Services MAIL :	3
Services VOIP :	3
Contre-mesures proposées / mises en place pour chaque services :	4
1. Services DNS :	4
2. Services WEB :	4
3. Services MAIL :	4
4. Services VOIP :	5

Risques encourus par nos VPS et notre infrastructure Docker :

Les principaux risques portent sur la **confidentialité**, la **disponibilité** et l'**intégrité**. De plus, nos **services doivent être opérationnels** toutes la journée (24h/24h).

Au niveau du VPS, on peut avoir :

- Usurpation d'identité du serveur.
- Attaque sur le mot de passe ("Brute-force")
- Surcharge du serveur ("DOS/DDOS")
- Une attaque sur un ports spécifique qui n'a pas été bloquer par notre VPS.
- Etc...

Au niveau de Docker, on peut avoir :

- Une attaque via la machine hôte sur le conteneur.
- Une image docker corrompue ou infectée.
- Une attaque par le réseau pour avoir accès à la machine hôte.
- Etc...

Contre-mesure mises en place :

Au niveau du VPS :

- Désactivation de l'accès au compte **root** en ssh.
- Authentification par **clé ssh** :
 - Nos VPS, autorise uniquement l'accès en SSH aux machines dont il possède la **clé publique**.
- Désactivation de l'**authentification par mot de passe**.

Au niveau de docker :

- Stockage des images en local le temps d'implémenter les services sur les VPS, et vérifier qui a accès à ses services.
- Test sur GNS3 (en local), avant d'implémenter sur les VPS.

Risques encourus par chacun des services déployés :

Services DNS :

- Attaque de type “Man in the middle”
- Des réponses falsifiées
- Cache poisoning
- Surcharge du serveur l’empêchant de répondre aux requêtes.

Services WEB :

- Accès à la base de données et modification de celle-ci.

Services MAIL :

- Spam de mails
- Phishing
- Confidentialité des mails.
- Intégrité au niveau de celui qui envoie le mail.
- Surcharge du serveur l’empêchant de répondre aux requêtes.

Services VOIP :

- Confidentialité des appels et des messages vocaux.
- Usurpation d’identité.
- Surcharge du serveur l’empêchant de répondre aux requêtes.

Contre-mesures proposées / mises en place pour chaque services :

1. Services DNS :

- Nous avons décidé de séparer la répartition des serveurs en deux zones : Une zone interne à l'entreprise, "intranet" et une zone externe, "DMZ".
- On a choisi de mettre tous les services liés à la zone interne sur un seul VPS, et mettre tout ce qui doit être accessible depuis l'extérieur sur les deux autres VPS.
- De ce fait, il est beaucoup plus simple pour nous de sécuriser les données "critique" de l'entreprise "woodytoys".

2. Services WEB :

- Il y a 3 sites web à mettre en place, dont un (le site "erp") qui doit être accessible uniquement depuis l'intérieur de l'entreprise. Du coup, nous avons décidé de le placer sur le VPS avec tous les autres services qui doivent rester accessible à l'intérieur de l'entreprise.
- Nous avons également utilisé un certificat SSL afin d'obtenir TOUS nos sites en HTTPS.
 - ◆ Pour nos sites à accès externe (b2b et www), nous avons utilisé un certificat SSL par "letsencrypt" / "certbot"
 - ◆ Pour notre site à accès interne (erp), nous avons utilisé un "self signed made" certificat SSL.

3. Services MAIL :

- Tous les mots de passes des adresse mails (déjà présentes – ajouté via le script « addmailuser ») sont crypté en utilisant le chiffrement « SHA512 ».
 - ◆ Du coup, dans le fichier de configuration « dovecot-sql.conf.ext » (dovecot), il faut ajouter cette ligne « default_pass_scheme = SHA512-CRYPT » pour lui dire que les mot de passes encodés dans la DB utilise le chiffrement SHA512.

```
INSERT INTO `maildb`.`virtual_users`  
(`id`, `domain_id`, `password`, `email`, `maildir`)  
VALUES  
('2', '1', ENCRYPT('toto', CONCAT('$6$', SUBSTRING(SHA(RAND()), -16))), 'olivier@wt2-5.ephec-ti.be', 'wt2-5.ephec-ti.be/olivier/');
```

- Pour ce qui est du Spam, nous avons mis en place « spamassassin » et aussi un RR de type txt (SPF) dans notre zone DNS.

PS : Le mot de passe de l'adresse mail « olivier@wt2-5.ephec-ti.be » n'est PAS « toto », elle à été changé juste pour l'exemple.

4. Services VOIP :

- Nous n'avons pas réellement ajouté de la sécurité à proprement dites. En effet, Asterisk gère tout seul la sécurité au niveau de :
 - ◆ La connexion des utilisateurs
 - ◆ La création de sessions
 - ◆ ...
- Nous avons par contre mis en place le service fail2ban spécifiquement pour la VoIP car celui-ci est particulièrement touché par les "brute-force attacks" ! Nous détaillons l'implémentation de fail2ban ci-après.

5. Firewall : (Voir Wiki pour plus de détails)

UFW :

Nous avons mis en place un script Bash permettant d'appliquer les règles (autorisé que tel port soit ouvert ou non) en fonction du vps sur lequel ce script est exécuté.

On s'est basé sur le principe de fermé / bloqué TOUS les ports et de uniquement ouvrir les ports nécessaires pour nos services.

Petit exemple, sur le vps de Daniel qui héberge le service mail ainsi que le SOA externe, on aura donc les 4 ports relatifs au mails (SMTP - SMTPS et IMAP – IMAPS) + (DNS) et n'oublions pas SSH.

```
vps-olivier@vps797990:~$ sudo ufw status
Status: active
```

To	Action	From
--	----	----
22	ALLOW	Anywhere
25	ALLOW	Anywhere
53	ALLOW	Anywhere
143	ALLOW	Anywhere
465	ALLOW	Anywhere
587	ALLOW	Anywhere
993	ALLOW	Anywhere

PS : Faut savoir, que nous utilisons une règle dans le script qui commence à chaque fois par BLOQUER tous les ports.

Notre petit script se termine en lançant un scan nmap sur l'adresse ip publique du vps, pour ainsi vérifier si le script fait bien ce qui prétends faire.

Fail2ban :

Nous avons installé et configuré fail2ban uniquement pour le service de VoIP "Asterisk". Nous sommes bien conscients qu'en production il aurait fallu le configurer pour chacun de nos services mais dans le cadre de ce projet nous nous sommes limité à là où celui-ci était indispensable.

Voici une capture d'écran de l'état actuel de fail2ban appliqué à notre service de VoIP :

```
vps-martin@vps797989:~$ sudo fail2ban-client status asterisk
Status for the jail: asterisk
|- Filter
|   |- Currently failed: 2
|   |- Total failed:    1628043
|   `-- File list:      /home/vps-martin/VPS-Martin/VoIP/logs/security
`- Actions
    |- Currently banned: 19
    |- Total banned:    19
    `-- Banned IP list:  103.145.12.52 103.145.12.83 103.145.12.94 103.145.13.16 185.40.4.53 185.53.88.175 185.53.88.61
                        45.143.220.131 45.143.220.213 45.143.220.13 173.213.89.58 37.49.230.93 5.183.94.30 45.82.254.170 45.143.221.45 185.53
                        .88.171 46.105.117.221 103.145.12.113 103.145.12.54
vps-martin@vps797989:~$
```

On constate qu'en seulement quelques dizaines d'heure de fonctionnement un grand nombre d'adresses IPs ont déjà été bannie.

Pour plus d'informations quant à l'installation, la configuration et la résolution de problèmes rencontré lors de la mise en place de fail2ban, veuillez-vous référer à notre wiki sur Github!