RAPPORT DE RECHERCHE DE VULNÉRABILITÉS ET PROPOSITION DE SOLUTIONS

Membre Du Groupe

- BAMBA ALLASSANE
- DJOMAN ANKRAN ANNE

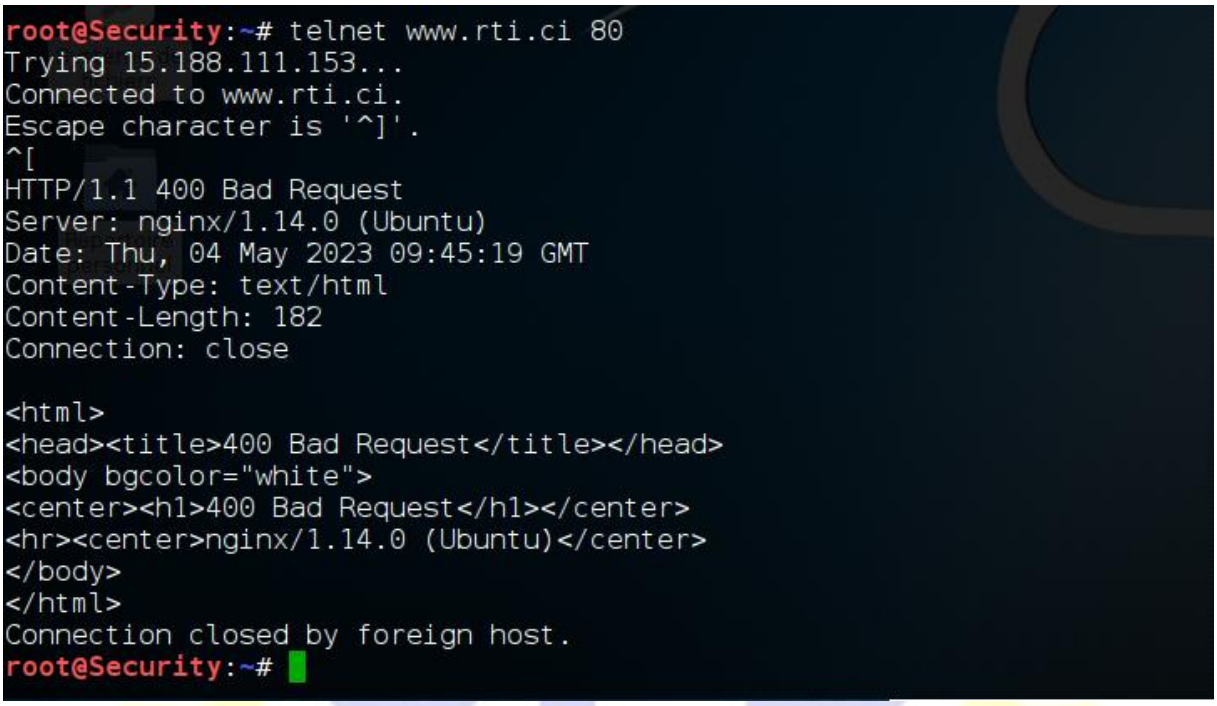
PROFESSEUR

Dr GOHOU BI

Dans le cadre de notre recherche nous travaillerons sur le site de www.rti.ci .

Ce projet consiste à utiliser Telnet sur le site de la RTI afin d'analyser les vulnérabilités de ngnix/1.14.0 sur (Ubuntu).

Apres avoir utiliser la commande Telnet sur le site de la RTI nous avons le résultats sont les suivants



Apres analyse du résultat :

- Recherchons les vulnérabilités qu'a connu ngnix/1.14.0 (Ubuntu)
- Proposer une solution des correctifs à la RTI
- Proposer une solution ainsi que les paramétrages à faire à la RTI pour ne plus qu'il réponde à Telnet

I- Recherchons les vulnérabilités qu'a connu ngnix/1.14.0 (Ubuntu)

Nginx (prononcé "engine-x") est un serveur web open source populaire utilisé pour servir des pages web statiques et dynamiques. Il peut également être utilisé comme un proxy inverse pour rediriger le trafic vers d'autres serveurs.

Nginx 1.14.0 est une version spécifique du logiciel Nginx, qui a été publiée le 17 avril 2018. Cette version est une version majeure qui a introduit plusieurs améliorations et fonctionnalités nouvelles par rapport à la version précédente.

Certaines des améliorations et des nouvelles fonctionnalités introduites dans Nginx 1.14.0 comprennent :
- Le support de TLS 1.3, la dernière version du protocole de chiffrement des communications en ligne
- L'amélioration des performances grâce à l'utilisation de l'API epoll sur les systèmes Linux
- L'ajout de nouvelles directives pour la configuration de Nginx, notamment pour la gestion des connexions TCP
- La correction de plusieurs bugs et vulnérabilités de sécurité.
Nginx 1.14.0 a été bien accueilli par la communauté des développeurs et des administrateurs système pour ses performances améliorées, ses nouvelles fonctionnalités et sa sécurité renforcée.

Les vulnérabilités découvertes durant nos recherches sont :
CVE CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516
CVE-2019-20372
CVE-2019-11043
CVE-2018-16843
CVE-2018-16844

1- Les vulnérabilités CVE CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516

- ***Les vulnérabilités liées à l'implémentation de la compression http/2***

Les CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516 sont toutes des vulnérabilités liées à l'implémentation de la compression HTTP/2 dans le serveur web Nginx version 1.14.0. Voici des exemples pratiques d'exploitation de ces vulnérabilités :

- CVE-2019-9511 : Un attaquant peut envoyer une requête HTTP/2 contenant des données de charge utile malveillantes, ce qui peut entraîner un déni de service par saturation de la mémoire vive du serveur Nginx. Ces vulnérabilités permettent à un attaquant distant de provoquer un déni de service (DoS) en envoyant des requêtes HTTP/2 malveillantes qui peuvent causer une saturation des ressources du serveur.
- CVE-2019-9513 : Un attaquant peut envoyer une requête HTTP/2 contenant un grand nombre de têtes de blocs malveillants, ce qui peut entraîner un déni de service par saturation de la mémoire vive du serveur Nginx.

- CVE-2019-9516 : Un attaquant peut envoyer une requête HTTP/2 contenant des données de charge utile malveillantes encodées de manière spécifique, ce qui peut entraîner un déni de service par saturation de la mémoire vive du serveur Nginx.

- ***Les vulnérabilités liées à la fragmentation des paquets dans le protocole TCP***

Les CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516 sont des vulnérabilités liées aussi à la fragmentation des paquets dans le protocole TCP. Voici quelques exemples de type de ces CVE :

- CVE-2019-9511 : Attaque de fragmentation par chevauchement de segments (OSA) dans le protocole TCP. Cette vulnérabilité permet à un attaquant de réduire les performances du réseau en envoyant des paquets TCP malformés qui contiennent des segments chevauchants. Cela peut conduire à une dégradation de la qualité de service (QoS) ou à une interruption du service (DoS).

- CVE-2019-9513 : Attaque de fragmentation à l'insu des connexions (Blind-Off-Path) dans le protocole TCP. Cette vulnérabilité permet à un attaquant de fragmenter les paquets TCP à l'insu des connexions existantes, ce qui peut conduire à une réduction de la bande passante, à une perte de paquets ou à une interruption du service.

- CVE-2019-9516 : Attaque de fragmentation par débordement de tampon (BOA) dans le protocole TCP. Cette vulnérabilité permet à un attaquant de provoquer un débordement de tampon en envoyant des paquets TCP malformés qui contiennent des segments de taille anormalement grande. Cela peut conduire à une augmentation de la consommation de ressources système, ce qui peut ralentir le système ou le rendre instable.

2- Les vulnérabilités CVE-2019-20372

La CVE-2019-20372 est une vulnérabilité de type injection de code dans le logiciel de gestion de contenus PluXml. Voici un exemple académique pratique de cette CVE :

- Un attaquant peut exploiter cette vulnérabilité en envoyant une requête HTTP contenant du code malveillant dans l'URL d'un formulaire ou d'un commentaire du site web PluXml. Le code malveillant peut alors être exécuté sur le serveur web, permettant à l'attaquant de prendre le contrôle du site ou de voler des informations sensibles.

- La vulnérabilité CVE-2019-20372 affecte les versions de NGINX antérieures à la version 1.17.3 et peut permettre à un attaquant distant d'exécuter du code arbitraire à distance ou de provoquer un déni de service en envoyant des requêtes HTTP spécialement conçues.

- L'attaque par injection de commandes L'attaquant pourrait envoyer une requête HTTP contenant des commandes malveillantes dans l'en-tête de la requête, qui seraient exécutées sur le serveur NGINX si elles ne sont pas correctement filtrées ou validées.

3- Les vulnérabilités CVE-2019-11043

La vulnérabilité CVE-2019-11043 affecte les versions de NGINX antérieures à la version 1.17.2 et peut permettre à un attaquant distant de provoquer un déni de service en envoyant des requêtes spécialement conçues.

- Cette vulnérabilité consiste à envoyer des requêtes HTTP contenant des paramètres d'URL spécialement conçus pour exploiter une vulnérabilité de débordement de

tampon dans le code de traitement des requêtes FastCGI de NGINX. L'attaquant pourrait ainsi provoquer un déni de service en faisant planter le serveur NGINX.

4- La vulnérabilité CVE-2018-16843

La vulnérabilité CVE-2018-16843 est une vulnérabilité de déni de service (DoS) dans le serveur web Nginx version 1.14.0 et antérieure. Elle est causée par une validation insuffisante des paquets TCP envoyés à un serveur Nginx, ce qui peut permettre à un attaquant distant d'envoyer des paquets spécialement conçus pour provoquer un crash du serveur.

Voici quelques exemples de scénarios d'exploitation de cette vulnérabilité :

- Attaque DoS : Un attaquant peut exploiter cette vulnérabilité en envoyant des paquets TCP spécialement conçus pour provoquer un crash du serveur Nginx. Cela peut entraîner une interruption de service pour les utilisateurs légitimes et une perte de données.
- Injection de code malveillant : Un attaquant peut utiliser cette vulnérabilité pour injecter du code malveillant dans le serveur Nginx, ce qui lui permettrait de prendre le contrôle du serveur et d'accéder à des données sensibles.
- Escalade de privilèges : Un attaquant peut exploiter cette vulnérabilité pour exécuter du code malveillant avec les privilèges du compte d'utilisateur Nginx, ce qui pourrait lui permettre d'obtenir un accès plus étendu au système.

5- Les vulnérabilité CVE-2018-16844

Les vulnérabilités CVE-2018-16844 est une autre faille de sécurité dans la version 1.14.0 de Nginx qui permet à un attaquant distant de provoquer un déni de service (DoS) en envoyant des requêtes HTTP spécialement conçues au serveur Nginx.

- L'exploitation de cette vulnérabilité consiste à envoyer une requête HTTP contenant un en-tête "Range" spécialement conçue. Cet en-tête peut provoquer une boucle infinie dans le traitement de la requête HTTP, entraînant ainsi une surcharge de la mémoire du serveur Nginx et un déni de service.

Ces exemples décrivent les mécanismes de ces vulnérabilités et les conséquences potentielles de leur exploitation. Ils peuvent être utilisés pour sensibiliser les utilisateurs et les administrateurs système aux risques associés à ces vulnérabilités et les aider à prendre des mesures pour les prévenir ou les corriger.

II- Proposer une solution des correctifs à la RTI

1- Les vulnérabilités CVE CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516

Pour corriger ces vulnérabilités, les utilisateurs de Nginx doivent mettre à jour leur serveur vers une version ultérieure qui contient les correctifs de sécurité pour ces vulnérabilités. Les administrateurs système peuvent également configurer Nginx pour désactiver la compression

HTTP/2 et utiliser une version antérieure du protocole HTTP pour minimiser les risques d'attaques. Il est également recommandé de surveiller les journaux d'accès pour détecter les tentatives d'exploitation de ces vulnérabilités.

Voici quelques solutions pratiques pour se protéger contre les attaques CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516 :

- Mettre à jour vers la dernière version de nginx : la dernière version stable de nginx inclut des correctifs pour ces vulnérabilités. Il est donc important de mettre à jour votre version de nginx pour vous protéger contre les attaques.

Pour ce faire, vous pouvez exécuter la commande suivante dans votre terminal :

- Sudo apt-get update
- Sudo apt-get upgrade nginx

Une fois la mise à jour terminée, redémarrez le service nginx avec la commande suivante :

- Sudo systemctl restart nginx

- Assurez-vous également que votre système d'exploitation dispose des correctifs nécessaires pour traiter ces vulnérabilités. Il est également recommandé de suivre les meilleures pratiques de sécurité telles que la configuration appropriée de votre pare-feu pour bloquer les requêtes malveillantes.
- Configurer des limites de bande passante et de mémoire : en configurant des limites de bande passante et de mémoire sur votre serveur nginx, vous pouvez empêcher les attaquants d'utiliser des requêtes HTTP/2 malveillantes pour provoquer une saturation des ressources du serveur.
- Utiliser un pare-feu pour filtrer les requêtes HTTP/2 malveillantes : en utilisant un pare-feu avec des règles de filtrage des requêtes HTTP/2 malveillantes, vous pouvez bloquer les attaques avant qu'elles n'atteignent votre serveur nginx.
- Limiter le nombre de sessions HTTP/2 : en limitant le nombre de sessions HTTP/2 sur votre serveur nginx, vous pouvez empêcher les attaquants de créer trop de sessions HTTP/2 et de provoquer une saturation des ressources.
- Utiliser un service de détection des attaques DDoS : les services de détection des attaques DDoS, tels que Cloudflare ou Akamai, peuvent détecter et bloquer les attaques avant qu'elles n'atteignent votre serveur nginx.
- Suivre les meilleures pratiques de sécurité : en suivant les meilleures pratiques de sécurité pour votre serveur nginx, telles que la configuration appropriée des certificats SSL, la gestion des journaux d'accès et l'utilisation de mots de passe forts, vous pouvez réduire le risque d'attaques malveillantes.

En mettant en place ces solutions pratiques, vous pouvez protéger votre serveur nginx contre les attaques CVE-2019-9511, CVE-2019-9513 et CVE-2019-9516.

2- Les vulnérabilités CVE-2019-20372

Pour corriger cette vulnérabilité :

- Les utilisateurs de PluXml doivent mettre à jour leur logiciel vers la version 5.6.1 ou ultérieure, qui contient un correctif pour cette vulnérabilité. Les administrateurs système peuvent également prendre d'autres mesures pour renforcer la sécurité de leur site web, telles que la mise en place de pare-feu et la surveillance des journaux d'accès.

- Il est recommandé de mettre à jour NGINX vers la dernière version disponible (1.17.3 ou supérieure). Si la mise à jour n'est pas possible, une solution de contournement consiste à désactiver l'utilisation de variables dans les directives de configuration de NGINX en utilisant la directive "uninitialized_variable_warn off;" dans le fichier de configuration. Cela désactive l'interprétation des variables non initialisées dans les directives de configuration, réduisant ainsi le risque d'exploitation de la vulnérabilité.

3- Les vulnérabilités CVE-2019-11043

Pour corriger ces vulnérabilités :

- Il est recommandé de mettre à jour NGINX vers la dernière version disponible (1.17.2 ou supérieure). Si la mise à jour n'est pas possible, une solution de contournement consiste à désactiver le traitement FastCGI en utilisant la directive "fastcgi_pass off ;" dans le fichier de configuration de NGINX. Cela empêche les requêtes FastCGI d'être traitées par NGINX et réduit ainsi le risque d'exploitation de la vulnérabilité.

4- Les vulnérabilités CVE-2018-16843

Pour corriger ces vulnérabilités :

- Il est recommandé de mettre à jour le serveur Nginx vers la version 1.15.6 ou supérieure. Si cela n'est pas possible, il est possible de désactiver le support TCP_DEFER_ACCEPT dans la configuration de Nginx pour atténuer le risque d'exploitation de la vulnérabilité.
  Pour ce faire, il faut ajouter ou modifier la ligne suivante dans le fichier de configuration Nginx (/etc/nginx/nginx.conf) :
  tcp_nopush on;
- Il est également recommandé de surveiller les logs du serveur Nginx pour détecter toute activité suspecte et de mettre en place des mesures de sécurité supplémentaires pour réduire le risque d'attaques.

5- Les vulnérabilité CVE-2018-16844

Pour corriger ces vulnérabilités :

- Mettre à jour la version de Nginx vers la dernière version stable

- Mettre en place des règles de pare-feu pour bloquer les requêtes HTTP contenant l'en-tête "Range" spécialement conçue
- Surveiller les journaux du serveur Nginx pour détecter toute tentative d'exploitation de cette vulnérabilité
- Sensibiliser les utilisateurs et les administrateurs système à la sécurité informatique et mettre en place des politiques de sécurité appropriées.

III- Proposer une solution ainsi que les paramétrages à faire à la RTI pour ne plus qu'il réponde à Telnet

La commande "Telnet www.siteweb.com 80" est utilisée pour tester la connectivité réseau et la communication avec un serveur web en établissant une connexion Telnet avec le port 80 du serveur web.

Pour empêcher un site web de répondre à cette commande, vous pouvez bloquer les connexions Telnet sortantes vers le port 80 du site web en utilisant un pare-feu ou un logiciel de sécurité. Cela peut être fait en configurant les règles de pare-feu pour bloquer les connexions Telnet sortantes vers le site web ou en utilisant un logiciel de sécurité pour bloquer les connexions Telnet sortantes vers le port 80.

Alternativement, le site web peut également être configuré pour ne pas accepter les connexions Telnet entrantes en bloquant l'accès au port 23, qui est utilisé pour les connexions Telnet.

Sous Linux, vous pouvez utiliser le pare-feu iptables pour bloquer les connexions Telnet entrantes sur le port 23. Voici les étapes à suivre :

- Ouvrez une session en tant que super utilisateur sur votre système Linux.

Tapez la commande suivante pour lister les règles de pare-feu actuelles :

```
iptables -L
```

- Si vous n'avez pas encore de règle pour bloquer les connexions entrantes sur le port 23, vous pouvez ajouter une règle avec la commande suivante :

```
iptables -A INPUT -p tcp --dport 23 -j DROP
```

- Si vous avez déjà une règle pour bloquer les connexions entrantes sur le port 23, vous pouvez la modifier en utilisant la commande suivante :

```
iptables -R INPUT -p tcp --dport 23 -j DROP
```

- Pour vérifier que la règle a été appliquée, tapez la commande suivante :

```
iptables -L
```

La règle que vous venez d’ajouter devrait apparaître dans la liste.

Ces commandes bloqueront les connexions entrantes sur le port 23, ce qui empêchera les connexions Telnet entrantes. Notez que cela bloquera également d’autres types de connexions utilisant le port 23, donc assurez-vous que cela ne gênera pas d’autres applications nécessitant ce port. De plus, ces règles ne sont que temporaires et ne seront pas persistantes après le redémarrage du système. Pour rendre les règles persistantes, vous devez les enregistrer dans un script de démarrage iptables ou utiliser une interface graphique pour configurer le pare-feu.

Il est important de noter que bloquer les connexions Telnet peut également affecter d’autres applications qui utilisent Telnet pour communiquer avec des serveurs distants. Il est donc important de configurer les règles de pare-feu ou le logiciel de sécurité avec prudence pour éviter les conséquences involontaires.

La configuration pour empêcher un site web d'accepter des connexions Telnet entrantes en bloquant l'accès au port 23 dépend du serveur web utilisé. Voici quelques étapes générales que vous pouvez suivre :

- Ouvrez le fichier de configuration du serveur web. Selon le serveur web que vous utilisez, le fichier peut se trouver dans différents emplacements. Par exemple, pour Apache, vous pouvez trouver le fichier de configuration dans `/etc/apache2/apache2.conf` ou `/etc/httpd/conf/httpd.conf`. Pour Nginx, vous pouvez trouver le fichier de configuration dans `/etc/nginx/nginx.conf`.

- Cherchez la section du fichier de configuration qui définit les directives de sécurité du serveur. Pour Apache, cette section peut s'appeler `<Directory>`, `<Location>` ou `<Files>`. Pour Nginx, il s'agit de la section `location`.

- Ajoutez la directive suivante pour bloquer les connexions entrantes sur le port 23 :

Deny from all

Cette directive empêchera toutes les connexions entrantes sur le port 23, y compris les connexions Telnet.

- Redémarrez le serveur web pour appliquer les changements de configuration.

Ces étapes empêcheront votre site web d'accepter les connexions Telnet entrantes en bloquant l'accès au port 23. Notez que cela peut également empêcher l'accès à d'autres services qui utilisent le port 23, tels que le client FTP traditionnel. Si vous devez utiliser le port 23 pour d'autres services, vous devrez trouver une solution alternative pour les protéger contre les connexions non autorisées.