Atelier 3. Comment mettre en place un pare-feu avec UFW sur Ubuntu 20.04

Introduction

UFW, ou Uncomplicated Firewall, est une interface de gestion de pare-feu simplifiée qui masque la complexité des technologies de filtrage de paquets de niveau inférieur telles que `iptables` et `nftables`. Si vous souhaitez commencer à sécuriser votre réseau, et vous n'êtes pas sûr de l'outil à utiliser, UFW peut être le bon choix pour vous.

Ce tutoriel vous montrera comment mettre en place un pare-feu avec UFW sur Ubuntu 20.04.

Conditions préalables

Pour suivre cet atelier, vous aurez besoin de :

- Un serveur Ubuntu 24.04 avec un non-root user sudo, que vous pouvez configurer en suivant notre tutoriel Configuration initiale du serveur avec Ubuntu 24.04.

UFW est installé par défaut sur Ubuntu. S'il a été désinstallé pour une raison quelconque, vous pouvez l'installer avec `sudo apt install ufw`.

Étape 1 — Utilisation d'IPv6 avec UFW (facultatif)

Ce document a été écrit en tenant compte d'IPv4, mais il fonctionnera aussi bien pour IPv6 pour autant que vous l'ayez activé. Si votre serveur Ubuntu a activé IPv6, assurez-vous qu'UFW est configuré pour prendre en charge IPv6 afin de gérer les règles de pare-feu pour IPv6 en plus d'IPv4. Pour ce faire, ouvrez la configuration UFW avec `nano` ou votre éditeur préféré.

```
1. sudo nano /etc/default/ufw
```

Ensuite, assurez-vous que la valeur d'`IPV6` est `yes`. Cela devrait ressembler à ceci :

```
2. /etc/default/ufw excerpt
```

```
3. IPV6=yes
```

Enregistrez et fermez le fichier. Maintenant, lorsque l'UFW est activé, il sera configuré pour écrire les règles de pare-feu IPv4 et IPv6. Cependant, avant d'activer UFW, nous voulons nous assurer que votre pare-feu est configuré pour vous permettre de vous connecter via SSH. Commençons par définir les politiques par défaut.

Étape 2 — Mise en place des politiques par défaut

Si vous commencez tout juste à utiliser votre pare-feu, les premières règles à définir sont vos politiques par défaut. Ces règles contrôlent la manière de traiter le trafic qui ne correspond pas explicitement à d'autres règles. Par défaut, UFW est configuré pour refuser toutes les connexions entrantes et autoriser toutes les connexions sortantes. Cela signifie que toute personne essayant d'atteindre votre serveur ne pourra pas se connecter, tandis que toute application à l'intérieur du serveur pourra atteindre le monde extérieur.

Remettons vos règles UFW à leur valeur par défaut afin que nous puissions être sûrs que vous pourrez suivre ce tutoriel. Pour définir les valeurs par défaut utilisées par UFW, utilisez ces commandes :

```
4. sudo ufw default deny incoming
5. sudo ufw default allow outgoing
```

Ces commandes définissent les valeurs par défaut pour refuser les connexions entrantes et autoriser les connexions sortantes. Ces paramètres par défaut du pare-feu peuvent suffire pour un ordinateur personnel, mais les serveurs doivent généralement répondre aux demandes entrantes d'utilisateurs extérieurs. Nous verrons cela plus loin.

Étape 3 — Autoriser les connexions SSH

Si nous activions notre pare-feu UFW maintenant, il refuserait toutes les connexions entrantes. Cela signifie que nous devrons créer des règles qui autorisent explicitement les connexions entrantes légitimes – connexions SSH ou HTTP, par exemple – si nous voulons que notre serveur réponde à ce type de demandes. Si vous utilisez un serveur cloud, vous voudrez

probablement autoriser les connexions SSH entrantes afin de pouvoir vous connecter à votre serveur et le gérer.

Pour configurer votre serveur afin d'autoriser les connexions SSH entrantes, vous pouvez utiliser cette commande :

```
6. sudo ufw allow ssh
```

Cela créera des règles de pare-feu qui autoriseront toutes les connexions sur le port `22`, qui est le port que le démon SSH écoute par défaut. UFW sait quel port `allow ssh` désigne parce qu'il est listé comme un service dans le fichier `/etc/services`.

Cependant, nous pouvons réellement écrire la règle équivalente en spécifiant le port au lieu du nom du service. Par exemple, cette commande fonctionne de la même manière que celle ci-dessus :

```
7. sudo ufw allow 22
```

Si vous avez configuré votre démon SSH pour utiliser un port différent, vous devrez spécifier le port approprié. Par exemple, si votre serveur SSH écoute sur le port `2222`, vous pouvez utiliser cette commande pour autoriser les connexions sur ce port :

```
8. sudo ufw allow 2222
```

Maintenant que votre pare-feu est configuré pour autoriser les connexions SSH entrantes, nous pouvons l'activer.

Étape 4 — Activation d'UFW

Pour activer UFW, utilisez cette commande :

```
9. sudo ufw enable
```

Vous recevrez un avertissement qui indique que la commande peut perturber les connexions SSH existantes. Nous avons déjà mis en place une règle de pare-feu qui autorise les connexions SSH, donc nous pouvons continuer. Répondez à l'invite avec `y` et appuyez sur `ENTER`.

Le pare-feu est maintenant actif. Exécutez la commande `sudo ufw status verbose` pour connaître les règles fixées. Le reste de ce tutoriel explique plus en détail comment utiliser UFW, par exemple en autorisant ou en refusant différents types de connexions.

Étape 5 — Autoriser d'autres connexions

À ce stade, vous devez autoriser toutes les autres connexions auxquelles votre serveur a besoin de répondre. Les connexions que vous devez autoriser dépendent de vos besoins spécifiques. Heureusement, vous savez déjà comment écrire des règles qui autorisent les connexions basées sur un nom de service ou un port ; nous l'avons déjà fait pour SSH sur le port `22`. Vous pouvez également le faire pour :

- HTTP sur le port 80, qui est ce qu'utilisent les serveurs web non cryptés, en utilisant `sudo ufw allow http` ou `sudo ufw allow 80`
- HTTPS sur le port 443, qui est ce qu'utilisent les serveurs web cryptés, en utilisant `sudo ufw allow https` ou `sudo ufw allow 443`

Il existe plusieurs autres moyens d'autoriser d'autres connexions, outre la spécification d'un port ou d'un service connu.

Plages de ports spécifiques

Vous pouvez spécifier des plages de ports avec UFW. Certaines applications utilisent plusieurs ports, au lieu d'un seul.

Par exemple, pour autoriser les connexions X11 qui utilisent les ports `6000-6007`, utilisez ces commandes :

```
10.  sudo ufw allow 6000:6007/tcp
11.  sudo ufw allow 6000:6007/udp
```

Lorsque vous spécifiez des plages de ports avec UFW, vous devez spécifier le protocole (`tcp` ou `udp`) auquel les règles doivent s'appliquer. Nous n'avons pas mentionné cela auparavant car le fait de ne pas spécifier le protocole autorise automatiquement les deux protocoles, ce qui est correct dans la plupart des cas.

Adresses IP spécifiques

Lorsque vous travaillez avec UFW, vous pouvez également spécifier des adresses IP. Par exemple, si vous souhaitez autoriser les connexions à partir d'une adresse IP spécifique, comme une adresse IP professionnelle ou personnelle de `203.0.113.4`, vous devez spécifier `from`, puis l'adresse IP :

```
12.  sudo ufw allow from 203.0.113.4
```

Vous pouvez également spécifier un port spécifique auquel l'adresse IP est autorisée à vous connecter en ajoutant `to any port` suivi du numéro de port. Par exemple, Si vous souhaitez autoriser `203.0.113.4` à se connecter au port `22` (SSH), utilisez cette commande :

```
13.  sudo ufw allow from 203.0.113.4 to any port 22
```

Sous réseaux

Si vous souhaitez autoriser un sous-réseau d'adresses IP, vous pouvez le faire en utilisant la notation CIDR pour spécifier un masque de réseau. Par exemple, si vous souhaitez autoriser toutes les adresses IP allant de `203.0.113.1` à `203.0.113.254` vous pourriez utiliser cette commande :

```
14.  sudo ufw allow from 203.0.113.0/24
```

De même, vous pouvez également spécifier le port de destination auquel le sous-réseau `203.0.113.0/24` est autorisé à se connecter. Une fois encore, nous utiliserons le port `22` (SSH) comme exemple :

```
15.  sudo ufw allow from 203.0.113.0/24 to any port 22
```

Connexion à une interface réseau spécifique

Si vous souhaitez créer une règle de pare-feu qui s'applique uniquement à une interface réseau spécifique, vous pouvez le faire en spécifiant « allow in on » suivi du nom de l'interface.

Vous pouvez consulter vos interfaces réseau avant de continuer. Pour ce faire, utilisez cette commande :

```
16.  ip addr
17.  Output Excerpt
18.  2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast
     state
19.  . . .
20.  3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group
     default
21.  . . .
```

Le résultat mis en évidence indique les noms d'interface réseau. Elles sont généralement nommées par quelque chose comme : `eth0` ou `enp3s2`.

Donc, si votre serveur a une interface réseau publique appelée `eth0`, vous pouvez l'autoriser à recevoir du trafic HTTP (port `80`) avec cette commande :

```
22.  sudo ufw allow in on eth0 to any port 80
```

Cela permettrait à votre serveur de recevoir des requêtes HTTP de l'Internet public.

Ou, si vous voulez que votre serveur de base de données MySQL (port `3306`) écoute les connexions sur l'interface de réseau privé `eth1`, par exemple, vous pourriez utiliser cette commande :

```
23.  sudo ufw allow in on eth1 to any port 3306
```

Cela permettrait à d'autres serveurs de votre réseau privé de se connecter à votre base de données MySQL.

Étape 6 — Refuser les connexions

Si vous n'avez pas modifié la politique par défaut des connexions entrantes, UFW est configuré pour refuser toutes les connexions entrantes. En général, cela simplifie le processus de création d'une politique de pare-feu sécurisée en vous obligeant à créer des règles qui autorisent explicitement le passage de ports et d'adresses IP spécifiques.

Cependant, il peut arriver que vous souhaitiez refuser des connexions spécifiques en fonction de l'adresse IP source ou du sous-réseau, peut-être parce que vous savez que votre serveur est

attaqué à partir de là. De plus, si vous souhaitez modifier votre politique d'entrée par défaut a **allow** (ce qui n'est pas recommandé), vous devrez créer des règles **deny** pour tous les services ou adresses IP pour lesquels vous ne souhaitez pas autoriser les connexions.

Pour écrire des règles **deny**, vous pouvez utiliser les commandes décrites ci-dessus, en remplaçant **allow** par **deny**.

Par exemple, pour refuser des connexions HTTP, vous pourriez utiliser cette commande :

```
sudo ufw deny http
```

Ou si vous souhaitez refuser toutes les connexions à partir de `203.0.113.4` vous pouvez utiliser cette commande :

```
sudo ufw deny from 203.0.113.4
```

Examinons maintenant comment supprimer des règles.

Étape 7 — Suppression de règles

Savoir comment supprimer des règles de pare-feu est tout aussi important que de savoir comment les créer. Il existe deux façons différentes de spécifier les règles à supprimer : par le numéro de la règle ou par la règle elle-même (de la même façon que les règles ont été spécifiées lors de leur création). Nous commencerons par la méthode **delete by rule number**, car elle est plus facile.

Par numéro de règle

Si vous utilisez le numéro de règle pour supprimer des règles de pare-feu, la première chose que vous voudrez faire est d'obtenir une liste de vos règles de pare-feu. La commande UFW status permet d'afficher des numéros à côté de chaque règle, comme illustré ici :

```
sudo ufw status numbered
Numbered Output:
Status: active

     To                         Action      From
```

```
31.       --                         ------      ----
32.  [ 1] 22                         ALLOW IN    15.15.15.0/24
33.  [ 2] 80                         ALLOW IN    Anywhere
```

Si nous décidons que nous voulons supprimer la règle 2, celle qui autorise les connexions sur le port 80 (HTTP), nous pouvons la spécifier dans une commande de suppression UFW comme celle-ci :

```
34.  sudo ufw delete 2
```

Cela affichera une demande de confirmation puis supprimera la règle 2, qui autorise les connexions HTTP. Notez que si vous avez activé IPv6, vous voudrez également supprimer la règle IPv6 correspondante.

Par règle réelle

L'alternative aux numéros de règle est de spécifier la règle réelle à supprimer. Par exemple, si vous voulez supprimer la règle `allow http`, vous pouvez l'écrire comme ceci :

```
35.  sudo ufw delete allow http
```

Vous pouvez également spécifier la règle par `allow 80`, au lieu de par nom de service :

```
36.  sudo ufw delete allow 80
```

Cette méthode supprimera les règles IPv4 et IPv6, si elles existent.

Étape 8 — Vérification de l'état et des règles d'UFW

À tout moment, vous pouvez vérifier le statut d'UFW avec cette commande :

```
37.  sudo ufw status verbose
```

Si UFW est désactivé, ce qui est le cas par défaut, vous verrez quelque chose comme ceci :

```
38.  Output
39.  Status: inactive
```

Si UFW est actif, ce qui devrait être le cas si vous avez suivi l'étape 3, la sortie indiquera qu'il est actif et énumérera toutes les règles qui sont définies. Par exemple, si le pare-feu est configuré pour autoriser les connexions SSH (port `22`) de n'importe où, la sortie pourrait ressembler à ceci :

```
Output
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
22/tcp                     ALLOW IN    Anywhere
```

Utilisez la commande `status` si vous souhaitez vérifier comment UFW a configuré le pare-feu.

Étape 9 - Désactivation ou réinitialisation d'UFW (facultatif)

Si vous décidez que vous ne voulez pas utiliser UFW, vous pouvez le désactiver avec cette commande :

```
40.  sudo ufw disable
```

Toutes les règles que vous avez créées avec UFW ne seront plus actives. Vous pourrez toujours exécuter la commande `sudo ufw enable` si vous devez l'activer plus tard.

Si vous avez déjà configuré des règles UFW mais que vous décidez de tout recommencer, vous pouvez utiliser la commande reset :

```
41.  sudo ufw reset
```

Cela désactivera l'UFW et supprimera toutes les règles qui ont été définies précédemment. Gardez à l'esprit que les règles par défaut ne retrouveront pas leurs paramètres d'origine, si vous les avez modifiées à un moment quelconque. Cela devrait vous permettre de repartir à zéro avec UFW.

Conclusion

Votre pare-feu est maintenant configuré pour autoriser (au moins) les connexions SSH. Veillez à autoriser toutes les autres connexions entrantes dont votre serveur a besoin, tout en limitant les connexions inutiles, afin que votre serveur soit fonctionnel et sécurisé.