SSH-FTP-WebServer-DNS-Wireshark

Ce guide montre des exemples de comment configurer différents services pour capturer des protocoles réseau entre deux machines en utilisant Wireshark.

Prérequis

- Deux machines Debian connectées au même réseau local.

- Accès root ou sudo sur les deux machines.

- Connexion réseau fonctionnelle entre les machines.

Étape 1: Installation de Wireshark

Sur chaque machine Debian, installer Wireshark :

sudo apt update

sudo apt install wireshark -y

Lors de l'installation, si demandé, choisis d'autoriser les utilisateurs non-root à capturer des paquets (groupe `wireshark`).

Ajouter utilisateur au groupe `wireshark`(remplace `ton_utilisateur` par ton nom d'utilisateur) :

sudo usermod -aG wireshark ton_utilisateur

newgrp wireshark

Étape 2: Configuration de SSH pour Accès Sécurisé (Facultatif)

Si on veut accéder aux machines via SSH, installer et configurer SSH :

sudo apt install openssh-server -y

Assure-toi que le service SSH est activé :

sudo systemctl enable ssh

sudo systemctl start ssh

Étape 3: Configuration des Services à Analyser

Pour capturer différents protocoles, on peut configurer divers services.

Voici quelques exemples :

FTP

Installer un serveur FTP sur une machine :

sudo apt install vsftpd -y

Activer et démarrer le service :

sudo systemctl enable vsftpd

sudo systemctl start vsftpd

Connecter l'autre machine avec un client FTP :

ftp adresse_ip_machine1

HTTP

Installer Apache sur une machine pour avoir un serveur web :

sudo apt install apache2 -y

On peut ensuite accéder à ce serveur depuis l'autre machine via un navigateur ou avec `curl`

curl http://adresse_ip_machine1

Configuration de Base d'Apache

1. Vérifier que le service Apache est actif :

sudo systemctl status apache2

- Le service doit être affiché comme `active (running)`.

2. Configuration des Hôtes Virtuels (Virtual Hosts) :

Apache utilise des hôtes virtuels pour servir plusieurs sites à partir d'un seul serveur.

- Créer un répertoire pour le site (par exemple `/var/www/example.com`).

sudo mkdir -p /var/www/example.com

- Attribuer les permissions appropriées :

sudo chown -R $USER:$USER /var/www/example.com

sudo chmod -R 755 /var/www

- Créer un fichier de configuration pour le site :

sudo nano /etc/apache2/sites-available/example.com.conf

Ajouter le contenu suivant :

```
<VirtualHost *:80>
ServerAdmin webmaster@example.com
ServerName example.com
ServerAlias www.example.com
DocumentRoot /var/www/example.com
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- Activer le site :

sudo a2ensite example.com.conf

- Désactiver le site par défaut (facultatif) :

sudo a2dissite 000-default.conf

3. Redémarre Apache pour appliquer les changements :

sudo systemctl restart apache2

Test de la Configuration

1. Créer une page de test :

echo "<html><body><h1>It works\!</h1></body></html>" > /var/www/example.com/index.html

2. Accèder au site via un navigateur :

Navigue vers `**http://adresse_ip_de_ta_machine**` ou `**http://example.com**` (si `example.com` est configuré dans ton `/etc/hosts`).

DNS

Installer `bind9` sur une machine pour un serveur DNS simple :

sudo apt install bind9 -y

Configurer le fichier `/etc/bind/named.conf.options` pour autoriser les requêtes :

sudo nano /etc/bind/named.conf.options

Ajouter ou modifier :

allow-query { any; };

Démarrer le service :

sudo systemctl enable bind9

sudo systemctl start bind9

Configuration de Base Bind9

1. Configuration de `named.conf.local` :

Créer les fichiers de zone pour ton domaine.

sudo nano /etc/bind/named.conf.local

Ajouter la configuration suivante : En vert avec l'IP de la VM utilisé pour l'exercice

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
};
zone "0.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168.0";
}
```

```
zone "example.com" {
        type master;
        file "/etc/bind/db.example.com";
};
zone "135.233.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168.233.135";
};
```

- Remplacer `example.com` par ton nom de domaine.

- Remplacer `192.168.0` par ton sous-réseau.

Note : `135.233.168.192.in-addr.arpa` correspond à l'adresse IP inversée pour `192.168.233.135`.

2. Création du fichier de zone pour `example.com` :

sudo nano /etc/bind/db.example.com

Utiliser ce modèle de fichier :

```
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.example.com.
@       IN      A       192.168.0.1
ns1     IN      A       192.168.0.1
www     IN      A       192.168.0.1
```

En vert avec l'IP de la VM utilisé pour l'exercice

```
;
; BIND data file for example.com
;
$TTL    604800
@       IN      SOA     ns1.example.com. root.example.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1.example.com.
@       IN      A       192.168.233.135
```

Explication des entrées :

- `@ IN SOA` *définit les paramètres du serveur principal de la zone.*
- `@ IN NS` *indique le serveur de noms pour la zone.*
- `@ IN A` *définit l'adresse IP pour* `example.com`*.*
- `ns1 IN A` *et* `www IN A` *définissent les adresses IP pour les sous-domaines* `ns1.example.com` *et* `www.example.com`*.*

3. Création du fichier de zone inversée :

sudo nano /etc/bind/db.192.168.0

Utiliser ce modèle de fichier :

;

; BIND reverse data file for local loopback interface

;

$TTL604800

@ IN SOA ns1.example.com. root.example.com. (

2 ; Serial

604800 ; Refresh

86400 ; Retry

2419200 ; Expire

604800) ; Negative Cache TTL

;

@ IN NS ns1.example.com.

1 IN PTR example.com.

```
;
; BIND reverse data file for 192.168.233.135
;
$TTL   604800
@      IN      SOA    ns1.example.com. root.example.com. (
                      2         ; Serial
                 604800         ; Refresh
                  86400         ; Retry
                2419200         ; Expire
                 604800 )       ; Negative Cache TTL
;
```

En vert avec l’IP de la VM utilisé pour l'exercice

Explication des entrées :

- `@ IN SOA` définit les paramètres du serveur principal de la zone.
- `@ IN NS` indique le serveur de noms pour la zone.
- `135 IN PTR` lie l'adresse IP `192.168.233.135` au nom de domaine `example.com`.

4. Vérifier les fichiers de configuration :

sudo named-checkconf

sudo named-checkzone example.com /etc/bind/db.example.com

sudo named-checkzone 0.168.192.in-addr.arpa /etc/bind/db.192.168.0

sudo named-checkconf

sudo named-checkzone example.com /etc/bind/db.example.com

sudo named-checkzone 135.233.168.192.in-addr.arpa /etc/bind/db.192.168.233.135

5. Redémarrer Bind9 pour appliquer les changements :

sudo systemctl restart bind9

6. Tester de la Configuration

Tester la résolution DNS :

Utiliser `nslookup` ou `dig` depuis une autre machine ou la même machine :

nslookup example.com 192.168.0.1

nslookup example.com 192.168.233.135

ou pour vérifier le nom de domaine :

dig @192.168.0.1 example.com

dig @192.168.233.135 example.com

Pour vérifier l'adresse IP :

nslookup 192.168.233.135 192.168.233.135

ou

dig -x 192.168.233.135 @192.168.233.135

7. Configuration de /etc/resolv.conf pour utiliser Bind9

Pour que le système utilise Bind9 pour la résolution DNS, ajouter Bind9 comme serveur DNS dans `/etc/resolv.conf

sudo nano /etc/resolv.conf

Ajouter la ligne suivante :

nameserver 192.168.0.1

nameserver 192.168.233.135

Remplacer `192.168.0.1` par l'adresse IP de ton serveur Bind9.

Test DNS depuis l'autre machine :

nslookup example.com adresse_ip_machine1

Notes Supplémentaires

- *Sérialisation : Augmente le numéro de série dans le champ SOA chaque fois que tu modifies le fichier de zone pour indiquer des changements.*
- *Sécurité : Assure-toi que les pare-feu et autres dispositifs de sécurité autorisent le trafic sur le port 53 (DNS).*

Conclusion

Avec ces étapes, tu as configuré **Apache** pour servir des pages web et **Bind9** pour résoudre les noms de domaine sur Debian. Pour des configurations plus complexes, tu peux

explorer les fonctionnalités avancées d'Apache et Bind9, comme la configuration SSL/TLS pour Apache ou la mise en place de DNS secondaire avec Bind9.

Étape 4: Capturer le Trafic avec Wireshark

Sur l'une des machines (ou les deux), utiliser Wireshark pour capturer le trafic :

1. Lancer Wireshark :

   wireshark

2. Sélectionner l'interface réseau appropriée (par exemple `eth0`, `wlan0`,'ens33').

3. Cliquer sur le bouton Start pour commencer la capture.

4. Effectue des actions pour générer du trafic réseau (accède au serveur web, FTP, DNS, etc.).

5. Observe et analyse les paquets capturés.

Pour plus de détails aller voir la documentation Wireshark