TP Implémentation d’une architecture d’accès à Internet

Préparation et recommandations

Une entreprise utilise le plus souvent un plan d’adressage privé pour l’ensemble des équipements internes à son réseau. Cela pose un problème lors de l’accès à Internet, notamment lorsque les utilisateurs utilisent leur browser, puisque ces adresses sont non routables. Pour corriger cela il existe plusieurs solutions comme nous allons le voir dans le reste du TP.
Pré-requis :

Connaitre les définitions des termes Firewall, DMZ.
Connaitre les commandes UNIX suivantes : ip addr, ip route, ping, traceroute, tcpdump.

Travail à effectuer :

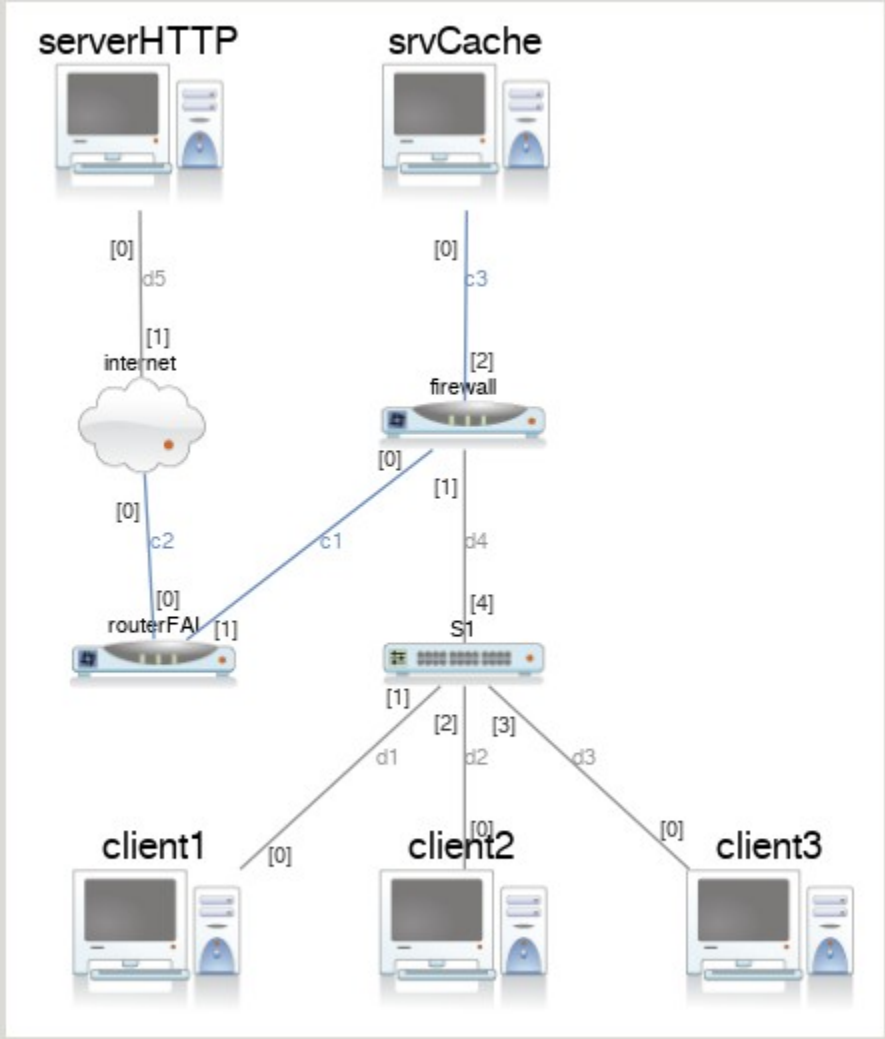
Lire l’introduction de la documentation sur le firewall linux :
https://wiki.nftables.org/

Introduction

Vous êtes en charge de réaliser l’accès à Internet d’une petite entreprise, pour simuler cette architecture, vous utiliserez un projet comme représenté en figure 1.

Les usagers internes sont appelés « clients » Pour cela vous pouvez administrer les équipements client1, client2, client3, switch, firewall, servCache représentant les équipements locaux, utilisant des adresses IP privées. Les « clients » dans le réseau interne (192.168.100.0/24) en .1 .2 et .3 respectivement pour client1 client2 et client3. Le serveur de cache DNS est en DMZ (192.168.110.1/2).

Toutes les interfaces du firewall ont l’adresse IP 254 pour la partie station (192.168.100.254, 192.168.110.254, 193.23.23.254). Vous avez un contrat avec un fournisseur d’accès Internet représenté par le routeur R1, et Internet représenté par N1 et serveurs_web. Bien sûr vous n’avez pas d’accès de configuration à ces équipements. Votre fournisseur d’accès Internet vous fourni le routeur R1 configuré ainsi qu’une plage d’adresses IP publiques 193.23.23.0/24 librement utilisable pour vous (sauf l’adresse .1 utilisée par le routeur).



Mise en œuvre de la solution

Avec cette solution les clients accèdent directement aux serveurs Internet sans passer par un proxy HTTP, les clients ayant des adresses privées, une translation d'adresse doit être effectuée pour que ces clients puissent accèder à Internet. Cette communication doit fonctionner pour tout type de flux. (ici nous nous limiterons à autoriser les seuls flux sortants HTTP et ICMP).

Pour créer ce genre d'architecture, il faut une démarche structurée, définir des étapes et valider chaque étape avant de passer à la suivante. Pour cette architecture simple, les trois étapes sont le routage, le NAT puis le filtrage.

1 – Routage.

- Activer le routage sur le firewall, et faites en sorte que cette modification soit effectuée à chaque reboot en modifiant le fichier /etc/sysctl.conf
- Indiquer la route par défaut pour les clients, pour le firewall et le proxy et installez les sur les clients (soit via l'interface marionnet soit sur le terminal avec la commande ip route dont vous regarderez le manuel).

Question : Sans translation d'adresse, quelles sont les différentes stations qui peuvent communiquer entrent-elles ? Faites des tests de ping entre ces stations afin de vérifier que le routage fonctionne à travers le firewall. Est-ce conforme à l'attendu ?

2 – Translation d'adresse.

- Le client 1 va être translaté vers internet en utilisant l'adresse IP publique du firewall (on appelle cela du masquerading), effectuez cette translation d'adresse en appliquant la règle appropriée sur le firewall :
- Le client2 doit être différencié vis-à-vis des serveurs web et sa translation d'adresse utilisera donc l'adresse source 193.23.23.2, effectuez cette translation d'adresse, que faut-il faire de plus pour que cela fonctionne ?

Question : vérifiez en effectuant un ping depuis client1, client2, client3, vers www.monsite.fr et www.mysite.com, et observez sur le firewall la translation d'adresse en utilisant la commande tcpdump -i eth0 -n puis tcpdump -i eth1 -n

Question : Avec quelle adresse source les paquets icmp arrivent-ils sur le serveur

web ? Vérifiez avec tcpdump.

2b – Résolveur DNS.

Configurez la machine srvCache pour qu'elle serve de résolveur pour tous les clients (cf TP 1). Il doit être configuré pour transférer les requêtes DNS sur le serverHTTP. Pour cela il devra être translaté vers internet en utilisant l'IP 193.23.23.3.

3 – Filtrage.

Maintenant que le routage et la translation d'adresse fonctionne, vous allez filtrer les accès des clients vers Internet pour n'autoriser que les flux conformément à la table ci-dessous.

Source	Destination	Protocole	Action
Tous les clients	Extérieur	HTTP, HTTPS	Autoriser
Tous	Tous	ICMP	Autoriser
srvCache	Extérieur	DNS	Autoriser
Tous les clients	srvCache	DNS	Autoriser

Quand on met en place un firewall, il faut la aussi une démarche structurée :

On commence par créer une table de filtrage

nft add table ip filter

Puis par créer une « chain » de politique par défaut « tout interdire » :

nft 'add chain ip filter forward { type filter hook forward priority 0 ; policy drop ; }';

Puis par autoriser dans cette « chain » les connexions établies :

nft insert rule ip filter goout ct state established counter accept

Ensuite on ajoute les règles de filtrage...

nft add rule ip filter forward ....

Question : Essayez d'accéder au port 80 de www.monfakesite.fr, que constatez-vous ? Que pouvez-vous dire sur le niveau OSI auquel se place le filtrage effectué ainsi ?

Quel est le statut HTTP de la page quand vous accédez à www.monfakesite.fr ?