Firewall - Proxy

Hainaut Patrick 2023

But

- Vous donner quelques notions de sécurité réseau
- Le sujet est vaste mais il faut bien commencer par quelque chose d'autant qu'on ne peut plus s'en passer

Introduction

- La sécurité informatique dans son ensemble est une branche importante de l'informatique
- Le but est de protéger l'intégrité des données et leur confidentialité
- Nous allons ici nous intéresser à quelques éléments de sécurité des réseaux
- Tout ce qui est sécurité au niveau d'un PC est abordé au cours d'OS

LE FIREWALL

But

- Un firewall (pare-feu en français) est un outil (physique ou logiciel) qui permet de sécuriser un réseau par rapport à un autre réseau
- Il va filtrer les entrées dans le système informatique, à partir du réseau, en se basant sur un ensemble de règles de sécurité
- Pour qu'il soit efficace, il faut le placer entre deux réseaux et s'assurer que toutes les communications réseaux passent par lui

Politique de sécurité

- Le firewall doit être configuré pour coller aux besoins et spécificités de l'entreprise
- Il y a lieu de définir une politique de sécurité en fonction de l'entreprise, que l'on répercutera dans la configuration du firewall
- On peut soit tout autoriser par défaut et bloquer les services dangereux, soit tout bloquer et autoriser les services nécessaires
- La deuxième option est plus sécuritaire

Politique de sécurité

- Par exemple; soit une société de commerce en ligne avec une vitrine sur Internet

- Les employés ont besoin:
  - d'accéder à Internet -> HTTP et HTTPS
  - d'accéder aux mails -> POP3 et IMAP
  - d'administrer le serveur via une console -> SSH
  - de mettre à jour les fichiers du serveur -> FTP
  - que le serveur soit accessible depuis Internet -> HTTP et HTTPS

- On bloquera tout trafic réseau sauf les protocoles cités ci-dessus

Politique de sécurité

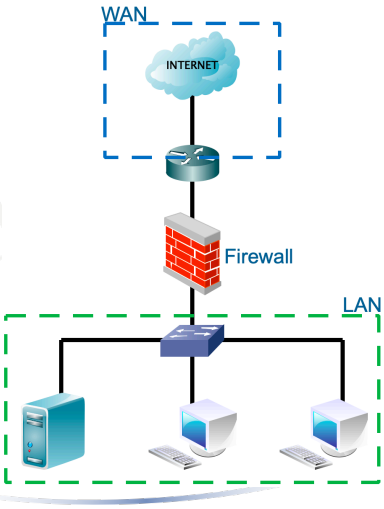
- On peut aller plus loin en créant des groupes d'utilisateurs avec des droits différents et/ou en limitant l'accès Internet à certains sites

- Exemple: seuls les employés s'occupant du site Web ont l'autorisation d'utiliser FTP

- Avant de configurer le firewall, il est essentiel de réunir les différents collaborateurs et de mettre au point une politique de sécurité solide et réfléchie

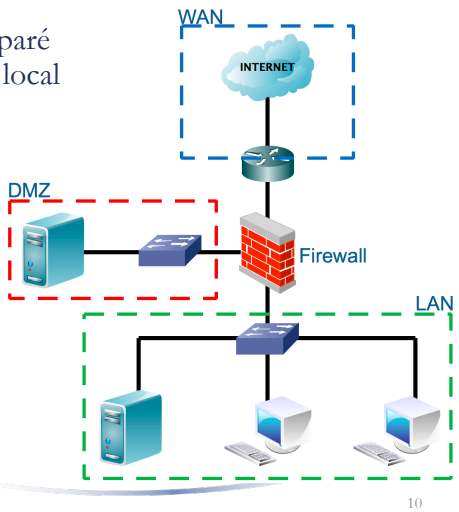
Protection de base

- Ici, le firewall est placé à l'entrée du réseau d'entreprise (ou domestique) et toutes les communications en provenance ou à destination de l'extérieur passent par lui



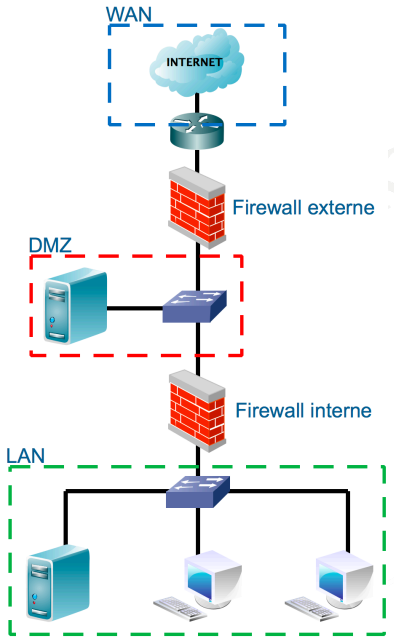
DMZ (DeMilitarized Zone)

- La DMZ est zone démilitarisée qui constitue un sous-réseau séparé à la fois d'Internet et du réseau local
- On va y loger les serveurs qui doivent être accessibles depuis Internet comme les serveurs Web, par exemple
- Si un pirate à accès au serveur de cette zone, il n'aura pas automatiquement accès aux machines du LAN



DMZ renforcée

- On peut aussi mettre la DMZ entre le réseau Internet et le réseau local et utiliser deux firewalls, ce qui renforce encore la sécurité



Fonctionnement du firewall

- Le firewall va permettre:
  - de bloquer le trafic entrant non désiré
  - de réguler et de filtrer tout autre trafic entrant, bien que le filtrage soit beaucoup moins utilisé depuis l'avènement de protocoles sécurisés et cryptés, basés sur TLS (successeur de SSL), comme HTTPS
  - de réguler le trafic sortant
- Le firewall ne permet pas:
  - d'éliminer les virus
  - de se protéger d'une personne malveillante à l'intérieur du réseau

Fonctionnement du firewall

- Le firewall va travailler sur les couches:
  - 3 -> paquets IP
    -> bloque certaines IP, ICMP, …
  - 4 -> messages TCP et UDP et les ports d'entrée/sortie
    -> bloque les protocoles travaillant sur ces couches comme telnet, ssh, …
  - 7 -> applications
    -> bloque certaines applications réseaux

Principales catégories

- Pare-feu sans états (stateless firewall)
  - les plus anciens et les plus basiques: traite chaque paquet en fonction d'une liste de règles établies (ACL pour Access Control List par exemple)
  - On les trouve essentiellement dans les routeurs
- Pare-feu à états (stateful firewall)
  - Permettent de tenir compte des protocoles connectés comme TCP et de vérifier que chaque paquet d'une connexion est bien en lien avec le précédent
  - Pour les protocoles sans connexion comme UDP, ils permettent de n'autoriser que le trafic entrant en réponse à une requête sortante

Principales catégories

- Pare-feu applicatif (application firewall)
  - dernière génération de firewall mais la plus gourmande en ressources
  - permet de vérifier la conformité du trafic à un protocole attendu (association protocole/port, exemple: trafic HTTPS sur le port 443)
- Portail captif (captive portal)
  - permet d'intercepter les utilisateurs qui veulent accéder au réseau et plus généralement à Internet pour leur présenter une page de connexion
  - l'accès peut être libre moyennant une inscription ou régit par un serveur contenant une base de données des utilisateurs autorisés (serveur RADIUS)
  - utilisé dans des espaces publics, des hôtels, l'horeca, des administrations, ….

Quelques implémentations actuelles

- Pare-feux sous forme de composants d'OS:
  - Linux Netfilter/iptables (à partir du noyau 2.4), libre
  - Packet Filter d'OpenBSD, généralisé sur les autres BSD, libre
  - Windows Defender Firewall
- Pare-feux sous forme d'OS:
  - pfSense
  - OPNsense
  - IPFire

Quelques implémentations actuelles

- Des sociétés commercialisent des pare-feux sous forme de boitiers:
  - Cisco
  - Check Point
  - Juniper
  - Nortel
  - Fortinet
  - …

LE PROXY

Notions élémentaires

- Proxy
  - Un proxy ou serveur mandataire (mandatory server) est un intermédiaire, un serveur « mandaté » par une application pour faire la requête sur Internet à sa place
  - Au départ le proxy se justifiait parce que le réseau mandataire n'était pas forcément TCP/IP alors qu'Internet oui
  - Maintenant, la plupart des réseaux que l'on rencontre sont des réseaux TCP/IP

Notions élémentaires sur les proxys

- Cache-proxy
  - Et comme les débits du début étaient faibles (modem 56k), c'est la fonction « cache » du proxy qui a d'abord été utilisée
    Une page consultée par un internaute était stockée « en cache » sur un disque local et disponible immédiatement si quelqu'un redemandait cette page
  - Comme la plupart des sites étaient statiques, il y avait peu de chances que le contenu ait changé et cela améliorait assez bien la vitesse de consultation
  - Les pages étaient néanmoins rafraichies régulièrement

Notions élémentaires sur les proxys

- Proxy-filtre
  - C'est la fonction communément employée du proxy actuellement
  - Comme pour le cache-proxy, il sert toujours d'intermédiaire mais filtrant
  - On peut filtrer des sites particuliers ou des catégories de sites (via recours à des blacklists)
  - C'est très utile dans le milieu du travail pour éviter que les employés n'utilisent l'accès au net pour leur loisir et que la productivité de la boite soit en baisse
  - C'est très utile aussi pour éviter l'accès à des sites illicites car en cas de représailles des autorités, c'est le propriétaire de la connexion qui sera reconnu responsable

Notions élémentaires sur les proxys

- Proxy explicite
  - Pour avoir accès à Internet, les utilisateurs doivent renseigner une adresse de proxy dans leur navigateur
  - Ils sont donc au courant de l'existence du proxy et en mesurer les conséquences comme le fait que leur historique de navigation est connu du proxy et donc de l'employeur … qui peut procéder à des analyses de fréquentation de sites comme facebook …

Notions élémentaires sur les proxys

- Proxy transparent
  - Le proxy transparent ne nécessite pas de configuration au niveau du poste utilisateur
  - Il est par conséquent « transparent » pour l'internaute
  - Mais légalement, l'utilisateur doit être prévenu qu'il est soumis à un proxy et donc probablement à un controle (un peu comme pour les caméras de sécurité)

Notions élémentaires sur les proxys

- HTTP et HTTPS
  - Pour l'HTTP, pas de soucis vu que tout passe en clair, on sait donc arrêter ce qu'on veut …
  - On peut faire du filtrage d'url et du filtrage de contenu
  - Mais en HTTPS, tout est crypté …

Notions élémentaires sur les proxys

- HTTP, HTTPS et proxy explicite
  - Alors, soit on utilise un proxy explicite, où on renseigne explicitement les sites qu'on veut visiter au proxy, pas de soucis, l'url est en clair et on peut faire du filtrage d'url mais pas de contenu (qui est crypté)
  - Si on veut filtrer le contenu, il faut décrypter l'HTTPS, et on doit donc "capturer" la requête et/ou la réponse et pour cela faire une attaque de type "man in the middle" (on crée un certificat reconnu comme autorité sur le réseau local, un CARoot maison, qui est valide pour tous les sites visités et qui intercepte et décrypte donc le flux pour l'examiner avant de reconstituer un flux crypté qui continue vers la destination prévue)
  - C'est une technique appelée Squid-in-the-middle ou plus généralement SSL Bump

Notions élémentaires sur les proxys

- HTTP, HTTPS et proxy transparent
  - Si on utilise un proxy transparent, l'HTTPS ne passe par le proxy par défaut et on doit donc "capturer" la requête et pour cela utiliser le SSL Bump pour pouvoir faire un filtrage d'url (et de contenu par la même occasion)

Notions élémentaires sur les proxys

- SSL Bump et légalité
  - Vous êtes tenus d'avertir vos utilisateurs que vous effectuer une interception SSL sur le réseau pour bloquer certains sites
  - Décrypter momantanément un contenu crypté qui n'est pas sensé être décrypté est un fameux trou de sécurité
  - Si l'employé amène son portable au sein de l'entreprise, du contenu privé peut être analysé et là, on enfreint la loi sur la vie privée …
  - Un lien vers un très bon article sur le sujet: https://www.silicon.fr/5-questions-comprendre-dechiffrement-ssl-100250.html?inf_by=59556d97681db8f17f8b45aa

Notions élémentaires sur les proxys

- Proxy explicite sans configuration sur le client
  - C'est possible avec WAD (Web Proxy Auto Discover protocol) qui est un protocole de découverte de proxy
  - Et donc, là pas besoin d'utiliser SSL Bump pour faire du filtrage d'url

Notions élémentaires sur les proxys

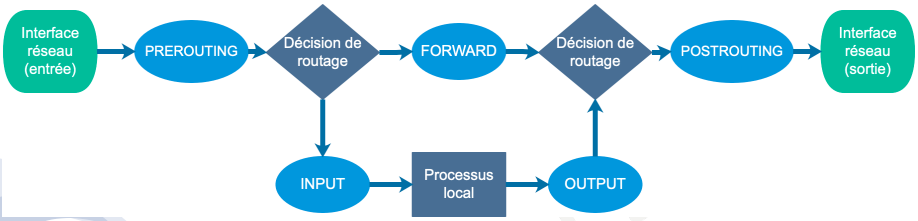
- Reverse-proxy
  - Un reverse-proxy est un cache proxy et/ou filter proxy monté à l'envers
  - Il sert de relais lors de l'accès par les internautes aux serveurs Web internes
  - Les serveurs Web internes sont donc protégés des attaques directes
  - S'ils existent plusieurs serveurs Web au contenu identique, une répartition de charge peut être effectuée

MISE EN PRATIQUE: NETFILTER

Netfilter

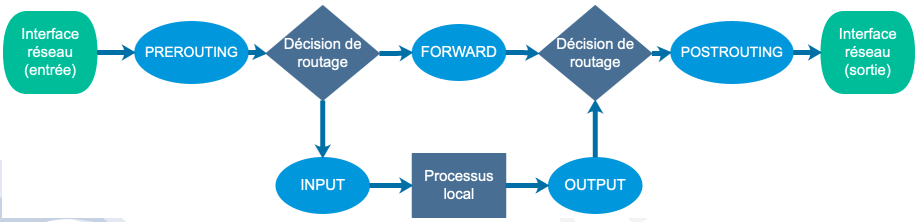
- Netfilter est un module du noyau Linux (depuis les noyaux 2.4 et 2.6) qui offre la possibilité de contrôler, modifier et filtrer les paquets IP, et de suivre les connexions
- C'est un pare-feu à états basé sur les couches réseau et transport
- Pour agir sur les paquets de données, Netfilter va donc se baser:
  - Sur l'entête IP (pour les adresses IP source et destination et les protocoles de cette couche)
  - Sur l'entête TCP ou UDP (pour les ports source et destination et les protocoles de cette couche)

Les cinq chaines



- Netfilter se présente comme une série de cinq chaines qui seront des points d'accrochage, sur lesquels des modules de traitement des paquets vont se greffer
- Ces chaines sont: PREROUTING (NF_IP_PRE_ROUTING)
  INPUT (NF_IP_LOCAL_IN)
  FORWARD (NF_IP_FORWARD)
  OUTPUT (NF_IP_LOCAL_OUT)
  POSTROUTING (NF_IP_POSTROUTING)

Les cinq chaines



- La partie inférieure représente le trajet des paquets qui entrent et qui sortent vers et depuis un processus local (SMB, FTP, HTTP etc.)
- La partie supérieure représente le trajet des paquets qui traversent notre passerelle dans sa fonction de routeur

Les tables et les chaines associées

- Avec Netfilter, on peut agir sur 3 tables (FILTER, NAT et MANGLE) qui agissent à leur tour sur un certain nombres des cinq chaines précitées

IPtables

- IPtables est l'interface "ligne de commande" permettant de configurer Netfilter
- Cette commande va permettre, entre autres, d'écrire des règles au niveau des trois tables
- Ces règles vont agir au niveau des cinq chaines
- On a déjà eu l'occasion d'utiliser iptables dans la manipulation Linux sur le partage de connexion, pour faire du NAT

La table « Filter »

- Cette table va contenir toutes les règles qui permettront de filtrer les paquets (bloquer ou autoriser). Cette table contient trois chaines :
  - **la chaine INPUT:**
    Cette chaine décidera du sort des paquets entrant **localement** sur l'hôte, c'est-à-dire les paquets destinés à cet ordinateur;
  - **la chaine OUTPUT:**
    Ici, ce ne sont que les paquets émis par **l'hôte local** qui seront filtrés, c'est-à-dire les paquets créés par cet ordinateur;
  - **la chaine FORWARD:**
    Enfin, les paquets qui traversent l'hôte, suivant les routes implantées, seront filtrés ici.

La table « Nat »

- Cette table permet d'effectuer toutes les translations d'adresses nécessaires. Elle contient comme chaines:
  - **La chaine PREROUTING:**
    Permet de faire de la translation d'adresse de destination. Cette méthode est intéressante si l'on veut faire croire au monde extérieur, par exemple, qu'il y a un serveur WEB sur le port 80 de la passerelle, alors que celui-ci est hébergé par un hôte du réseau privé, sur le port 8080
  - **La chaine POSTROUTING:**
    Elle permet de faire de la translation d'adresse de la source, comme du masquage d'adresse, la méthode classique pour connecter un réseau privé comme client de l'Internet, avec une seule adresse IP publique (NAT)
  - **La chaine OUTPUT:**
    Celle-ci va permettre de modifier la destination de paquets générés localement (par la passerelle elle-même)

La table « Mangle »

- Cette table permet le marquage ou la transformation des paquets. Elle contient toutes les chaines par défaut:
  - **La chaine PREROUTING:** tous les paquets entrant sur l'hôte
  - **La chaine INPUT:** tous les paquets destinés à l'hôte
  - **La chaine OUTPUT:** tous les paquets créés par l'hôte
  - **La chaine FORWARD:** tous les paquets traversant l'hôte
  - **La chaine POSTROUTING:** tous les paquets quittant l'hôte

Suivi de connexion

- Le suivi de connexion est un concept essentiel dans Netfilter
- C'est une sorte d'intelligence artificielle qui permet d'établir des liens de cause à effet entre les paquets qui passent dans la pile
- C'est ce qui fait que Netfilter est un pare feu à états

Suivi de connexion

- Si l'on met en place un système capable de mémoriser ce qu'il se passe sur la couche TCP, alors il va devenir possible de savoir si une connexion est dans l'un de ces états :
  - NEW
    nouvelle connexion (elle contient le flag SYN),
  - ESTABLISHED
    connexion déjà établie, elle ne devrait pas contenir de SYN ni de FIN,
  - RELATED
    la connexion présente une relation directe avec une connexion déjà établie, qui la crée (exemple: échange de données FTP après établissement de la connexion FTP)
  - INVALID
    la connexion n'est pas conforme, contient un jeu de flags anormal, n'est pas classable dans l'une des trois catégories précédentes.

Suivi de connexion en UDP

- Là, c'est plus délicat puisqu'il n'y a justement pas de connexion. Il sera donc impossible de définir de façon précise l'état d'un échange UDP. Ce que l'on pourra faire, c'est mettre en place un « timer » pour décider de l'état d'un paquet UDP
- Nous pouvons prendre l'exemple simple d'une requête DNS depuis notre réseau privé:
  - Le premier paquet UDP sort de notre réseau, sur un port connu et identifié (53) vers un serveur DNS. Nous pouvons décider de le laisser passer et nous le qualifions de NEW. Il déclenche un timer
  - si avant expiration du « timer », nous recevons un paquet UDP du dit serveur DNS, nous considèrerons que c'est un paquet ESTABLISHED

Ecriture et lecture des règles

- Les règles écrites sont lues séquentiellement par Netfilter
- Dès qu'une règle correspond, elle est appliquée et la lecture de la liste s'arrête
- On va donc placer les règles les plus précises en début de liste et les règles les plus globales en fin de liste

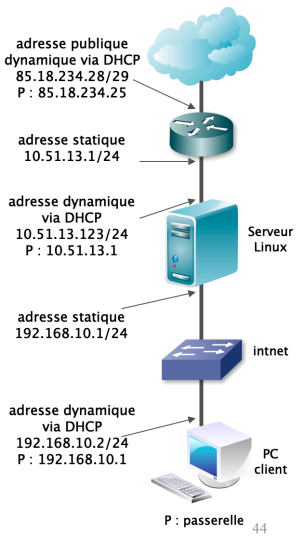
Exemple de politique de sécurité

- Supposons une PME où l'on interdit tout trafic réseau sauf l'accès au surf Internet, aux mails et à SSH pour administrer le serveur à distance
- Le ping à partir du LAN est autorisé
- On pourrait définir cette liste de règles:

Décision	IP source	IP destination	Port source	Port destination	Protocole
autorise	*	*	*	443 (HTTPS)	TCP
autorise	*	*	*	25 (SMTP)	TCP
autorise	*	*	*	143 (IMAP)	TCP
autorise	*	IP serveur	*	22 (SSH)	TCP
autorise	LAN	*	-	-	ICMP
interdit	*	*	*	*	*

Application de ces règles avec iptables

- On reprend notre schéma réseau déjà utilisé pour Manip7, 8 et 10 et on considère que ces manip sont effectuées
  - serveur DHCP actif avec 8.8.8.8 comme DNS
  - firewalld et selinux désactivés
  - iptables installé et la règle de NAT établie
  - Installation par défaut d'open-ssh
- Si ce n'est pas le cas, veuillez vous référer aux Manip 7, 8 et 10



Traduction en règles iptables

- Pour appliquer les règles de la dia 43, comme on est dans du filtrage, on va donc agir sur la table filter
- Au niveau de cette table, on va d'abord définir la politique générale
  - Soit on laisse tout passer par défaut
  - Soit on interdit tout par défaut

- Ici, on va tout interdire et autoriser juste ce qu'il faut
- Or, si on regarde l'état actuel de la table filter, tout est autorisé

```
[root@srvlinux ~]# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
[root@srvlinux ~]#
```


Politique par défaut

- On va donc modifier la politique par défaut, qui peut prendre trois options: ACCEPT, DROP, REJECT

- Dans notre cas, on tape les commandes:
  iptables --policy INPUT DROP
  iptables --policy FORWARD DROP
  iptables --policy OUTPUT DROP

```
[root@srvlinux ~]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
[root@srvlinux ~]#
```

- DROP et REJECT auront le même effet (interdire le passage des paquets de données) mais DROP n'envoie pas de message d'erreur à l'expéditeur ce qui est plus sécuritaire

Politique par défaut

- A partir de maintenant, les PC clients du LAN n'ont plus accès à Internet (chaine FORWARD), on ne sait plus faire un ping ou se connecter en SSH sur le serveur (chaine INPUT) et le serveur lui-même n'a plus accès à Internet (chaine OUTPUT)
- Remarque: INPUT et OUTPUT étant sur la même ligne (voir dia 32), bloquer un des deux, bloque les communications en entrée ET en sortie (mais pas en FORWARD si la chaine est laissé en ACCEPT)

Autoriser le trafic vers Internet

- Pour les PC clients, on va agir sur la chaine FORWARD
- La commande suivante ouvre le trafic vers Internet:

iptables -A FORWARD -i enp0s8 -o enp0s3 -p tcp --dport 443
-m conntrack --ctstate NEW, ESTABLISHED, RELATED -j ACCEPT

-A pour Append (ajouter)
-i pour spécifier la carte réseau d'entrée: ici la carte du LAN; enp0s8
-o pour spécifier la carte réseau de sortie: ici la carte du WAN; enp0s3
-p pour spécifier le protocole visé par la règle
--dport pour spécifier le port de destination: ici le port serveur HTTPS
-m conntrack pour activer le suivi de connexion
--ctstate pour définir les états du paquet de données (les PC clients peuvent établir une nouvelle connexion et répondre à une connexion établie ou en relation)
-j pour spécifier la décision (ACCEPT, DROP ou REJECT)

Autoriser le trafic vers Internet

- Avec la commande précédente, on va dans le sens LAN -> WAN
- Pour que la requête vers Internet aboutisse et revienne, il faut autoriser le retour
- On pourrait écrire (on verra dans quelques dias une règle plus simple):

iptables -A FORWARD -i enp0s3 -o enp0s8 -p tcp --sport 443
-m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

La carte d'entrée est maintenant la carte WAN et la carte de sortie, la carte LAN
Le port 443 devient maintenant un port source
Le port de destination sur le PC client est un port aléatoire dont le numéro est compris entre 1024 et 65535
On n'accepte pas ici de trafic initié par Internet (NEW), on se prémunit ainsi des attaques frontales

Autoriser le trafic vers Internet

- Pour pouvoir surfer sur Internet avec le PC client, il faut pouvoir effectuer des requêtes DNS
- Il faut donc introduire une règle spécifique:

iptables -A FORWARD -i enp0s8 -o enp0s3 -p udp --dport 53
-m conntrack --ctstate NEW, ESTABLISHED, RELATED -j ACCEPT

DNS utilise UDP comme protocole de transport
le port par défaut du serveur DNS est 53

- Pour la règle de retour, on pourrait écrire:

iptables -A FORWARD -i enp0s3 -o enp0s8 -p udp --sport 53
-m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

Autoriser le ping vers Internet

- Pour pouvoir faire un ping d'un serveur sur Internet, il faut autoriser le trafic ICMP:

iptables -A FORWARD -i enp0s8 -o enp0s3 -p icmp
-m conntrack --ctstate NEW, ESTABLISHED, RELATED -j ACCEPT

- Pour la règle de retour, on pourrait écrire:

iptables -A FORWARD -i enp0s3 -o enp0s8 -p icmp
-m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

- On ne précise pas de port vu qu'on se situe sur la couche réseau et pas sur la couche transport

Règle de retour

- On a autorisé 3 protocoles et on a donc écrit 3 règles en ce sens + les règles de retour indispensables

- Comme, de toute façon, on n'accepte de nouveau trafic en provenance d'Internet, on pourrait simplifier ces règles de retour et en écrire une seule, valable pour tous les protocoles:

iptables -A FORWARD -i enp0s3 -o enp0s8
-m conntrack --ctstate ESTABLISHED, RELATED -j ACCEPT

Règle de retour

- Et si, comme c'est le cas ici, on n'utilise pas de protocole créant des paquets en relation, on peut enlever l'état RELATED des commandes précédentes, ce qui donne au final:

iptables -A FORWARD -i enp0s8 -o enp0s3 -p tcp --dport 443
-m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

iptables -A FORWARD -i enp0s8 -o enp0s3 -p udp --dport 53
-m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

iptables -A FORWARD -i enp0s8 -o enp0s3 -p icmp
-m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

iptables -A FORWARD -i enp0s3 -o enp0s8
-m conntrack --ctstate ESTABLISHED -j ACCEPT

Exercice

- Il faut que le PC client puisse utiliser un client mail (pour le webmail, ça fonctionne déjà)

- Ecrivez les règles lui permettant d'envoyer et de recevoir de mails à partir d'un client mail (Thunderbird, Outlook, …)

Autoriser l'accès SSH sur le serveur

- Pour autoriser cet accès, nous allons travailler sur la chaine INPUT
- Pour les PC clients, on peut écrire la règle suivante:

iptables -A INPUT -p tcp -s 192.168.10.0/24 --dport 22
-m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

-s précise l'adresse source: ici, on indique l'entièreté du réseau LAN, on pourrait préciser uniquement une adresse IP, la seule à pouvoir se connecter en SSH sur le serveur

Le port par défaut de SSH est le 22

L'adresse du serveur, à préciser dans le client SSH est ici 192.168.10.1

Autoriser l'accès SSH sur le serveur

- Si on veut pouvoir se connecter en SSH sur le serveur, à partir du réseau externe (dans notre exemple: 10.51.13.0/24), on peut ajouter la règle suivante:

iptables -A INPUT -p tcp -s 10.51.13.0/24 --dport 22
-m conntrack --ctstate NEW, ESTABLISHED -j ACCEPT

L'adresse du serveur, à préciser dans le client SSH est ici l'adresse reçue par le DHCP du réseau externe

Autoriser l'accès SSH sur le serveur

- Si on veut pouvoir se connecter en SSH sur le serveur, à travers Internet, depuis le domicile par exemple:
  - Si le serveur a une adresse publique, pas de soucis, on écrit la même règle que précédemment avec le réseau de son domicile
  - Si le serveur a une adresse privée, comme c'est le cas dans notre exemple (10.51.13.0/24), là ça se complique, il faut écrire une règle de redirection de port sur la box/routeur qui a d'un coté, une adresse publique et de l'autre, une adresse privée dans le réseau du serveur

Autoriser l'accès SSH sur le serveur

- Au niveau de la règle de retour, dans tous les cas, on peut écrire:

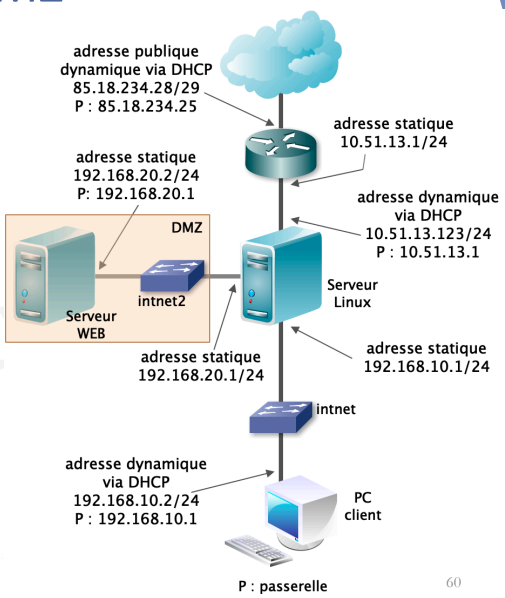
iptables -A OUTPUT -p tcp --sport 22
-m conntrack --ctstate ESTABLISHED -j ACCEPT

Exercice

- On aimerait que le serveur puisse faire un ping des serveurs externes (IP et nom de domaine) mais qu'on ne puisse pas le pinguer de l'extérieur
- Ecrivez les règles lui permettant de le faire
- On aimerait aussi que les PC clients puissent faire un ping du serveur et que le serveur puisse faire un ping des PC clients
- Ecrivez les règles permettant cela

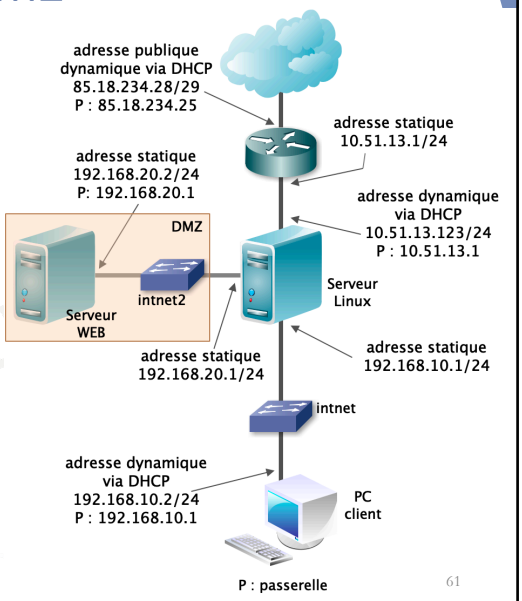
Application à la DMZ

- Sur le schéma précédent, on ajoute un réseau qui constituera la DMZ
- Le serveur Web qu'elle contient doit pouvoir être accessible d'Internet
- Ecrivez les règles iptables pour permettre cela



Application à la DMZ

- Vous pourrez tester du PC hôte si vous avez accès au serveur Web (mais pas au serveur Linux)
- Vous pourrez tester du PC client si vous avez également accès au serveur Web



MISE EN PRATIQUE: PFSENSE

Introduction

- **pfSense** est un routeur/pare-feu (router/firewall) open source basé sur le système d'exploitation FreeBSD
- Il existe une version commerciale; pfSense plus
- Il existe aussi OPNSense, un fork de pfSense, open source, qui contient plus ou moins les mêmes fonctionnalités mais est organisé différemment

- BSD est un OS dérivé de Unix et mis au point à l'université de Berkeley en Californie
- FreeBSD est conçu spécialement pour être utilisé comme serveur
- OpenBSD est spécialisé dans la sécurité informatique

Introduction

- pfSense utilise le pare-feu à états Packet Filter (pare-feu officiel d'OpenBSD), des fonctions de routage et de NAT lui permettant de connecter plusieurs réseaux informatiques

- Outre les fonctions de routage (avec gestion des VLANs) et de firewall, il peut aussi jouer (entre autre) les rôles de:
  - Serveur Proxy
  - Serveur DHCP
  - Serveur DNS
  - NAT
  - VPN

Introduction

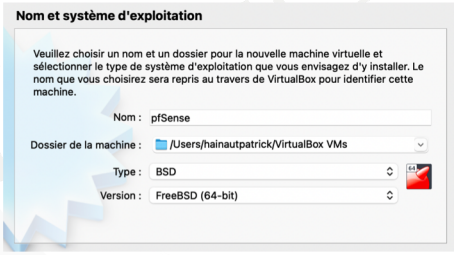
- pfSense convient pour la sécurisation d'un réseau domestique ou de petite entreprise
- Après une brève installation manuelle pour assigner les interfaces réseaux, il s'administre ensuite à distance depuis l'interface web et gère nativement les VLAN (802.1q)
- Comme sur les distributions Linux, pfSense intègre aussi un gestionnaire de paquets pour installer des fonctionnalités supplémentaires

Installation

- Vous pouvez télécharger pfSense sur le site officiel: www.pfsense.org/download
- pfSense combine à la fois l'outil et l'OS, donc pas besoin d'OS hôte
- Les versions actuelles sont uniquement en 64 bits
- Au niveau du matériel recommandé, on est sur un processeur 64 bits d'1GHz, avec 1Gb de RAM, et au moins 4 Gb de HDD, ce qui permet de recycler des machines un peu plus anciennes

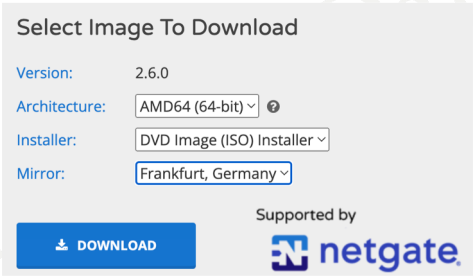
Installation

- Dans le cadre de notre labo, nous l'installerons sous VirtualBox
- Nous créerons donc une VM de type FreeBSD 64 bits avec 1Gb de RAM, 16 Gb de HDD et 3 cartes réseaux (1 en accès par pont (bridge), 2 en réseau interne (internal): intnet et intnet2)



Installation

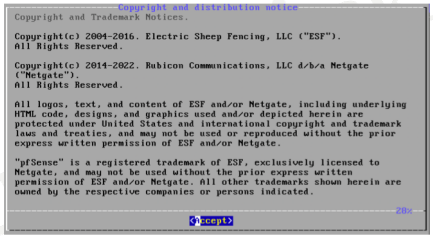
- Pour l'installation sous VirtualBox, nous choisirons une image ISO
- Prenez la dernière version disponible (qui n'est pas forcément celle illustrée ici)



- Pour une machine dédiée, on choisira avantageusement l'image USB

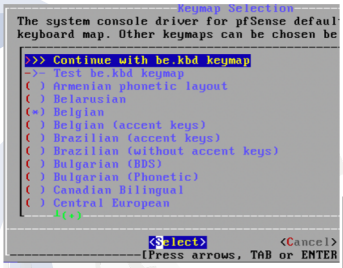
Installation

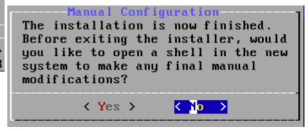
- Après l'écran de bienvenue, le système démarre automatiquement l'installation

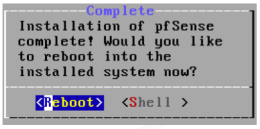


Installation

- On peut modifier le type de clavier même si cela ne change rien au niveau de la console, on reste en QUERTY …
- On choisit une partition BIOS sans apporter d'autres modifications







Première configuration

- La configuration en ligne de commande se résume à assigner les interfaces
- On constate, dans notre cas, que l'assignation est déjà correcte
- L'adresse du WAN dépend du serveur DHCP présent sur votre réseau

```
FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 76751b254d4dc7d52b34

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.0.171/24
 LAN (lan)       -> em1        -> v4: 192.168.1.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```


Première configuration

- Dans le cas où on veut la modifier, on choisit l'option 1 dans le menu
- Le système demande ensuite si on veut configurer les VLANs
- On peut le faire après, donc on répond non (N)

```
0) Logout (SSH only)
1) Assign Interfaces
2) Set interface(s) IP address
```

```
Enter an option: 1

Valid interfaces are:

em0     08:00:27:4f:8a:a9   (up) Intel
.
em1     08:00:27:93:67:2c   (up) Intel
.
em2     08:00:27:7b:0a:fa (down) Intel
.

Do VLANs need to be set up first?
If VLANs will not be used, or only for
say no here and use the webConfigurato

Should VLANs be set up now [y|n]?
```

Première configuration

- Ensuite, il faut indiquer quelle carte est l'interface WAN, quelle carte est l'interface LAN, et indiquer éventuellement d'autres cartes optionnelles

```
Should VLANs be set up now [y|n]? n

If the names of the interfaces are not known, auto-detection can
be used instead. To use auto-detection, please disconnect all
interfaces before pressing 'a' to begin the process.

Enter the WAN interface name or 'a' for auto-detection
(em0 em1 em2 or a): em0

Enter the LAN interface name or 'a' for auto-detection
NOTE: this enables full Firewalling/NAT mode.
(em1 em2 a or nothing if finished): em1

Enter the Optional 1 interface name or 'a' for auto-detection
(em2 a or nothing if finished):

The interfaces will be assigned as follows:

WAN  -> em0
LAN  -> em1

Do you want to proceed [y|n]?
```


Menu système

- Le reste de la configuration se fait à distance via l'interface Web
- Il faut revenir sur la machine seulement pour des manipulations système comme la réinitialisation du mot de passe

```
FreeBSD/amd64 (pfSense.home.arpa) (ttyv0)

VirtualBox Virtual Machine - Netgate Device ID: 76751b254d4dc7d52b34

*** Welcome to pfSense 2.6.0-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.0.171/24
 LAN (lan)       -> em1        -> v4: 192.168.1.1/24

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

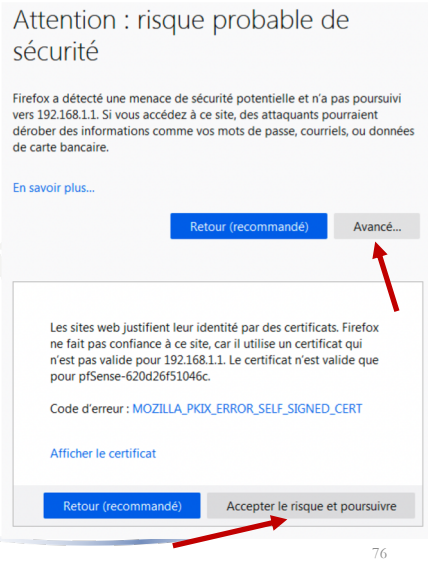
Connexion à pfsense

- On peut accéder à l'interface de configuration de pfsense en se connectant à partir d'un PC client du LAN
- Pfsense active par défaut un serveur DHCP qui nous donne les paramètres IP nécessaires

```
C:\Users\admin>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

   Connection-specific DNS Suffix  . : home.arpa
   Link-local IPv6 Address . . . . . : fe80::d9d:3a6a:ccde:d1fa%10
   IPv4 Address. . . . . . . . . . . : 192.168.1.100
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::a00:27ff:fe13:ce62%10
                                       192.168.1.1

Tunnel adapter isatap.home.arpa:

   Media State . . . . . . . . . . . : Media disconnected
   Connection-specific DNS Suffix  . : home.arpa

C:\Users\admin>
```

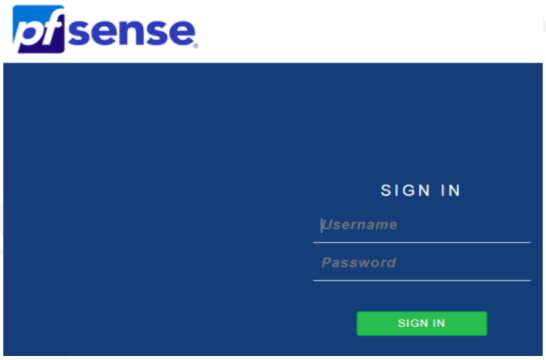

Connexion à pfsense

- Si on tape dans un navigateur l'adresse LAN de pfsense (192.168.1.1 par défaut), on arrive d'abord sur une page d'avertissement car l'interface web de pfsense est en HTTP par défaut
- Il faut accepter de poursuivre



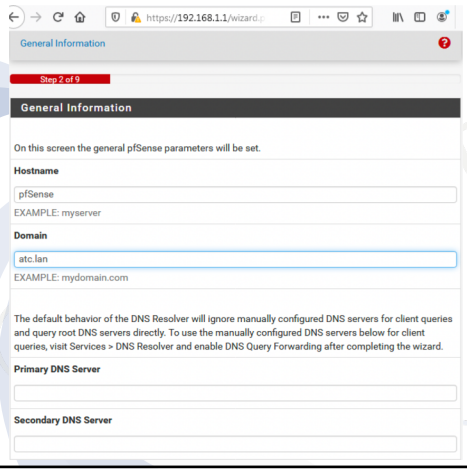
Connexion à pfsense

- On arrive alors sur la page de connexion
- Le login par défaut est **admin** et le mode passe **pfsense**



Assistant de configuration web

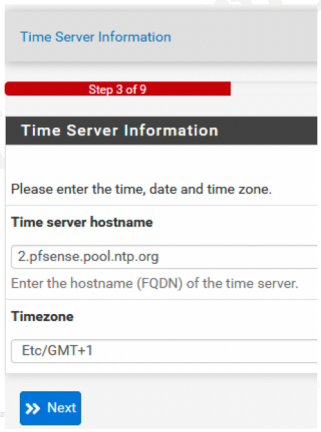
- A la première connexion, un assistant se lance pour la configuration initiale



- Il faut indiquer le nom de la machine, le domaine éventuel et les DNS éventuels
- Si on n'indique pas d'adresse de serveur DNS, le système utilise l'adresse du ou des DNS fourni par le DHCP de la carte WAN

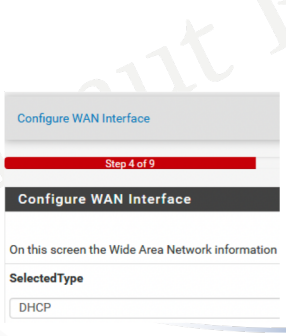
Assistant de configuration web

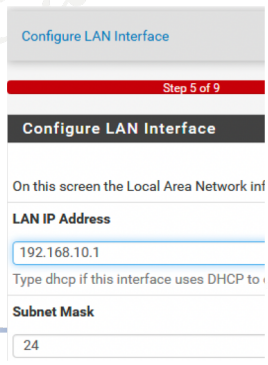
- Pfsense est un serveur NTP (Network Time Protocol), c'est à dire un serveur de temps réseau qui permettra aux PC clients de se synchroniser sur une horloge internet fiable



Assistant de configuration web

- On laisse l'interface WAN en DHCP
- On peut modifier l'adresse par défaut de l'interface LAN ainsi que le masque de sous-réseaux
- On prendra 192.168.10.1/24 pour que ça corresponde à notre schéma réseau
- On ne configure pas la deuxième carte interne pour l'instant





Assistant de configuration web

- Il est temps de changer le mot de passe par défaut

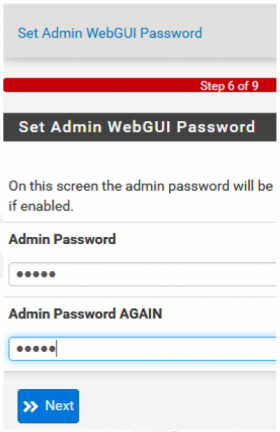
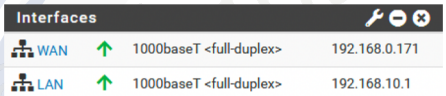


Tableau de bord

- Une fois cette configuration initiale terminée, on accède au tableau de bord (dashboard)



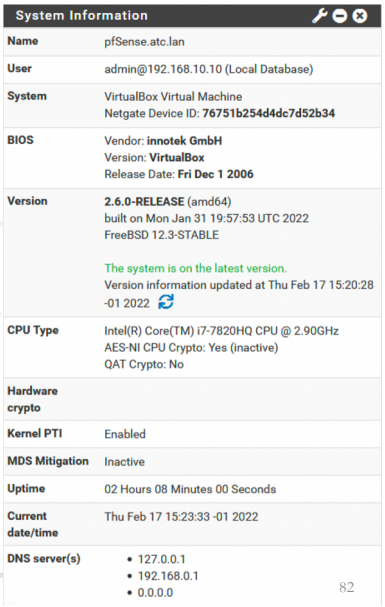
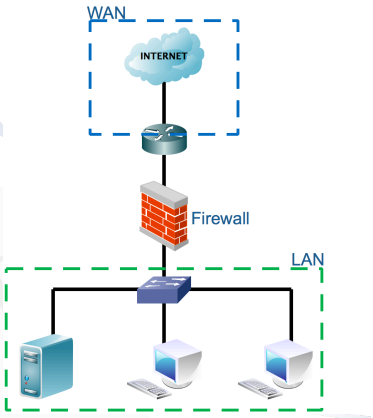
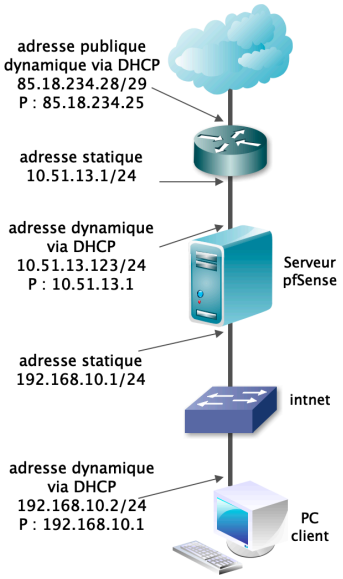


Schéma réseau 1

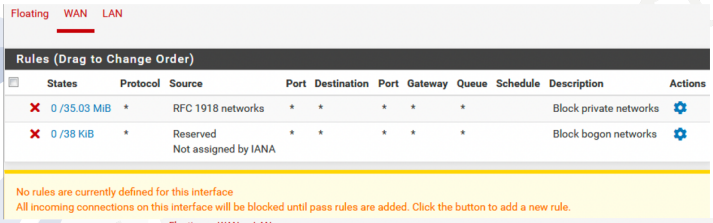
- Un premier schéma réseau assez simple qu'on connait déjà:

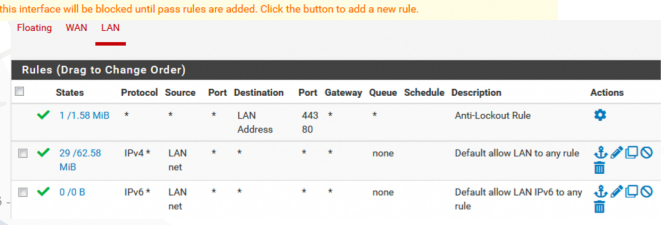




Trafic par défaut

- Par défaut, pfSense bloque tout trafic initié du WAN vers le LAN et autorise tout trafic du LAN vers le WAN, le NAT est activé et le PC client a donc accès à Internet





Scénario 1: mise en place d'un proxy

- Dans une entreprise, un établissement éducatif ou pour un hotspot wifi on placera souvent un proxy filtrant interdisant l'accès à certaines catégories de sites et/ou à des sites en particulier
- Pour l'exemple, ici, on interdira tous les sites ayant trait au sexe et on interdira l'accès à Facebook sauf de 12h à 13h

Scénario 1: mise en place d'un proxy

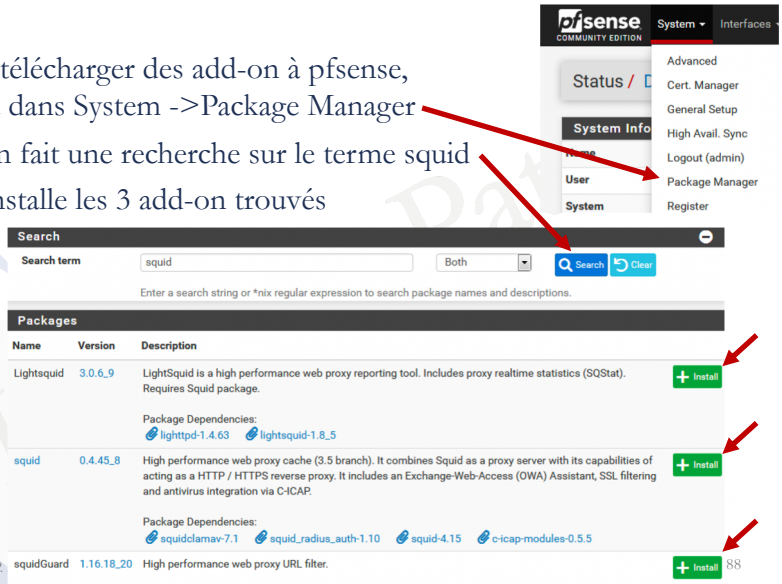
- Nous allons installer ici un proxy dédié et transparent
- Il faudra ensuite remplir la blacklist du proxy avec les sites interdits
- Vous pouvez utiliser une des listes suivantes suivant le contexte: https://dsi.ut-capitole.fr/blacklists/download/
- Comme proxy, nous utiliserons Squid
- Pour gérer la blacklist, nous utiliserons SquidGuard
- Pour avoir un rapport détaillé, nous utiliserons Lightsquid

Téléchargement des paquets

- Pour télécharger des add-on à pfsense, on va dans System -> Package Manager
- Là, on peut faire une recherche sur le terme squid
- On installe les 3 add-on trouvés

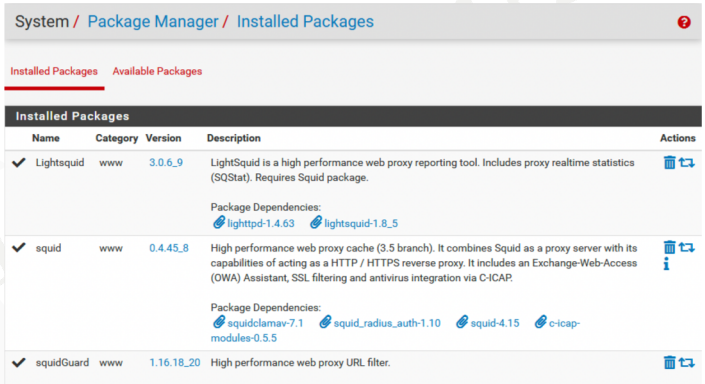
Téléchargement des paquets

- Pour télécharger des add-on à pfsense, on va dans System ->Package Manager
- Là, on fait une recherche sur le terme squid
- On installe les 3 add-on trouvés



Téléchargement de paquets

- Une fois les paquets installés, on les retrouve dans Installed Packages et on peut voir la version, les dépendances, les mettre à jour et les supprimer



Certificats

- A l'heure actuelle, on ne peut pas se contenter de gérer le trafic HTTP, il faut surtout gérer le trafic HTTPS
- En effet, HTTPS est devenu le protocole par défaut des navigateurs et HTTP est déprécié voir obsolète
- Avec HTTP, les données passent en clair et sont sujettes à une interception par un pirate
- Avec une généralisation de formulaires, de documents officiels et du commerce en ligne, il y avait lieu de sécuriser les échanges, ce qui est le cas avec HTTPS

Certificats

- Pour sécuriser HTTP et passer à HTTPS, on doit utiliser des certificats TLS/SSL
- TLS (Transport Layer Security) est le successeur de SSL (Secure Sockets Layer) et est un protocole qui agit au niveau de la couche transport (couche 4 OSI) suivant un modèle client-serveur
- TLS permet:
  - d'authentifier le serveur
  - de chiffrer l'échange de données
  - d'assurer l'intégrité des données
  - plus rarement d'authentifier le client car c'est géré par la couche application

Certificats

- Il y a trois types de certificats SSL:
  - Les certificats à validation de domaine (DV)
  - Les certificats à validation d'organisation (OV)
  - Les certificats à validation étendues (EV)
- Les différences ne se situent pas au niveau du chiffrement mais au niveau de la vérification de la société qui demande le certificat

Certificats DV

- Les certificats DV sont les plus faciles et rapides à obtenir et aussi les moins chers (entre 0 et 250€/an): l'autorité de certification vérifie juste que le nom de domaine appartient bien au demandeur, le nom du demandeur ou de sa société n'apparait pas dans le certificat
- On sait que les données échangées sont chiffrées mais on n'est pas sûr de leur destinataire
- C'est suffisant pour un site web d'information
- L'autorité de certification let's encrypt permet d'obtenir un certificat SSL DV gratuitement

Certificats OV

- Là, en plus des exigences du certificat DV, l'autorité de certification vérifie l'identité de la personne demanderesse et qu'elle est bien propriétaire de la société demanderesse (dans le cas d'une société)
- Le nom du demandeur ou de sa société apparait dans le certificat
- Cela revient entre 25 et 350€/an et il est généralement délivré dans les deux jours ouvrables
- Convient pour un site de e-commerce

Certificats EV

- Là, en plus des exigences du certificat OV, la société demanderesse est soumise à un audit et on vérifie sa fiabilité, sa raison sociale, son pays d'origine, …
- Ces informations apparaissent dans le certificat
- N'est pas délivré aux particuliers
- Cela revient entre 50 et 450€/an
- Convient pour un grand site de e-commerce, les administrations, les banques, …

Dans notre cas

- On veut filtrer le trafic HTTPS, on se contentera donc d'un certificat DV
- On peut soit prendre le certificat interne à pfsense ou un certificat let's encrypt
- Notre domaine étant un .lan non reconnu dans l'espace DNS, on choisira le certificat interne
- La procédure pour installer un certificat let's encrypt est cependant expliqué juste après

Algorithmes de chiffrement

- Pour mettre en œuvre TLS/SSL, on utilise un algorithme de chiffrement;
  - soit symétrique: une clé secrète, utilisée pour le chiffrement et le déchiffrement est utilisée (AES, blowfish, DES, …)
  - soit asymétrique: une clé publique est utilisée pour le chiffrement et une clé privée (différente) est utilisée pour le déchiffrement
- L'inconvénient de l'algorithme symétrique est que la clé de chiffrement doit être transmise au destinataire de manière sure
- Pour sécuriser le trafic HTTP, on utilisera généralement un algorithme asymétrique

Algorithmes asymétriques

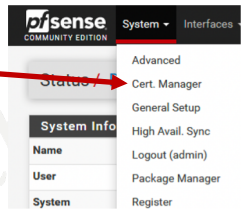
- On trouve comme algorithmes asymétriques actuels:
  - RSA (initiales de ses 3 inventeurs, du MIT)
  - ECDSA (Elliptic Curve Digital Signature Algorithm)
  - EdDSA (Edwards-curve DSA)
- Actuellement, on recommande encore RSA mais avec une longueur de clé de 4096 bits

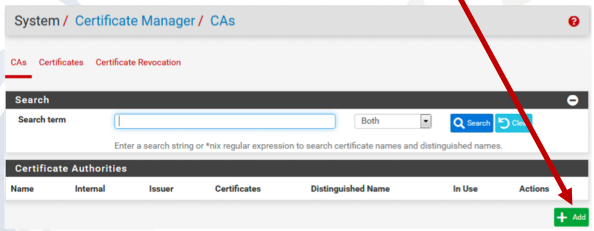
Fonctions de hachage

- Une fonction de hachage appliquée à un message transmis va permettre d'obtenir une chaîne binaire de taille fixe et de valeur unique
- On pourra vérifier l'intégrité des données/validité d'une signature, en appliquant la même fonction de hachage sur un message reçu et en comparant avec la valeur de hachage transmise
- On utilise habituellement SHA (Secure Hash Algorithm), version 2 ou 3

Installation du certificat interne

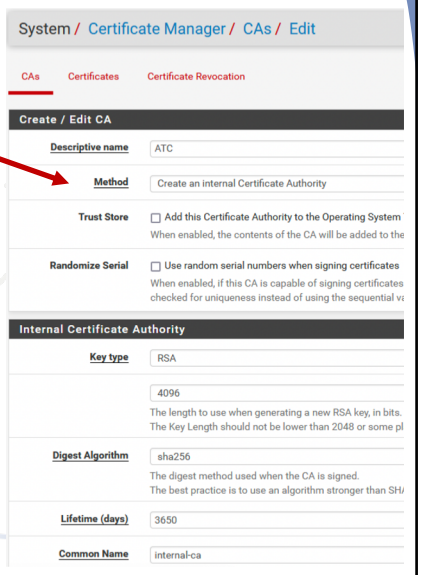
- Pour créer une autorité de certification interne, on va dans **System** / **Cert. Manager**
- Au niveau de **Cas**, on clique sur **Add**





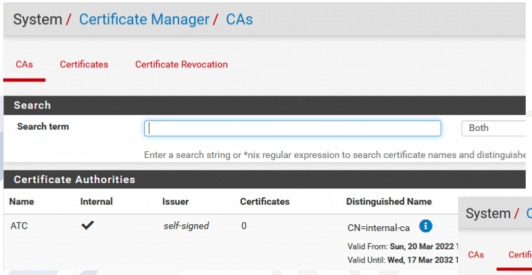
Installation du certificat interne

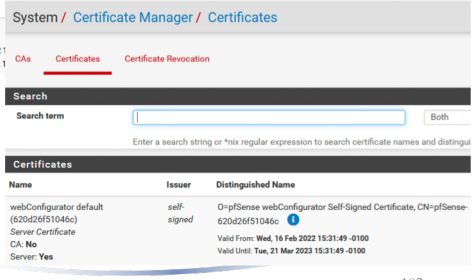
- On donne un nom (peu importe lequel) et on choisit de créer une autorité de certificat interne
- Au niveau du type de clé, on choisit RSA avec une longueur de 4096 bits
- Le reste peut être laissé par défaut
- Validez en cliquant sur Save dans le fond de la page



Installation du certificat interne

- L'autorité de certificat est créée ainsi que le certificat

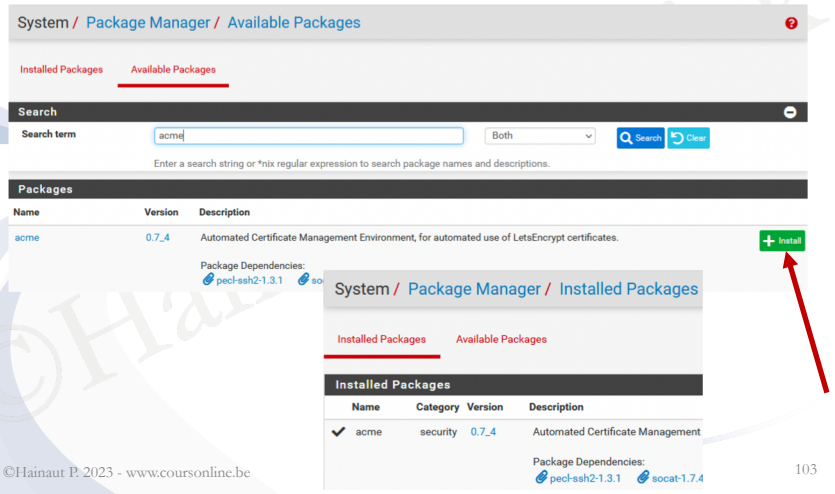




- On voit que l'autorité de certification créée est valable 10 ans alors que le certificat devra être renouvelé dans un peu plus d'un an

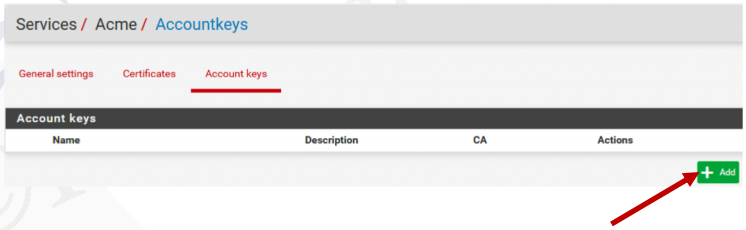
Installation du certificat let's encrypt

- On installe d'abord le paquetage "acme"



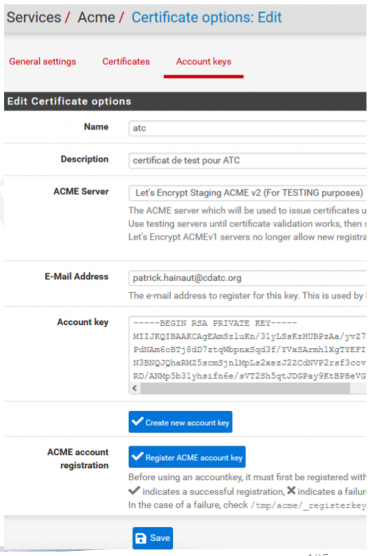
Installation du certificat let's encrypt

- Ensuite, on peut aller dans **Services/Acme Certificates**
- On crée d'abord un compte via **Account keys** -> **Add**



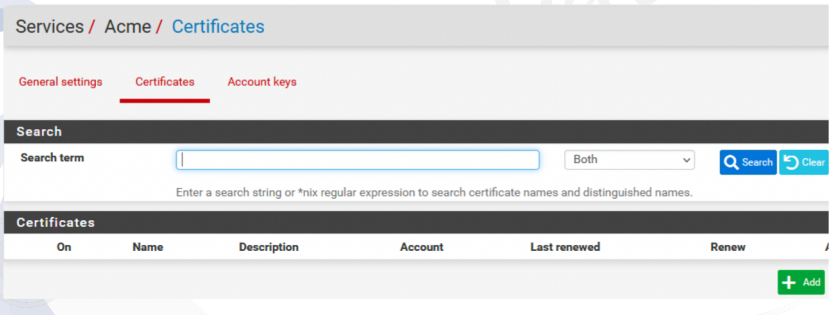
Installation du certificat let's encrypt

- On indique un nom, une description
- On choisit le serveur Let's Encrypt (testing ou production) suivant ce qu'on veut en faire
- On clique sur **create new account key**
- On clique sur **Register ACME account key**
- On clique sur **Save**



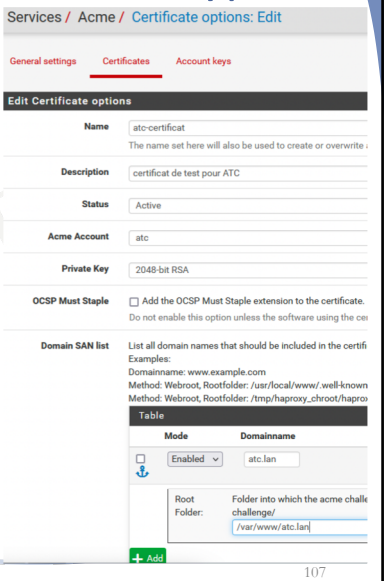
Installation du certificat let's encrypt

- On crée ensuite le certificat en allant sur **Certificates** -> **Add**



Installation du certificat let's encrypt

- On indique un nom, une description
- On choisit le status **Active** et le compte Acme qu'on vient de créer
- On prend une clé de 2048 bits
- On indique le nom de domaine et l'endroit où on peut trouver les fichiers du site



Scénario 2: mise en place d'un firewall

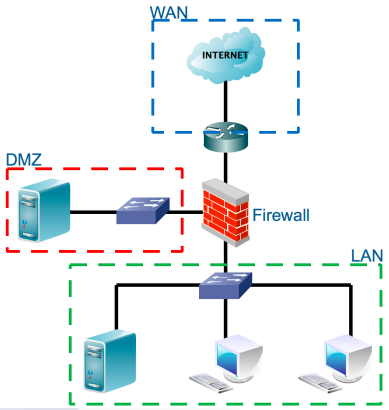
- Soit à interdire tout accès initié de WAN vers le LAN
- Soit à rendre obligatoire le passage par le proxy pour le trafic Web du LAN vers le WAN
- Soit à interdire tout accès du LAN vers le WAN sauf pour une série de services bien déterminés:
  - HTTP (via proxy)
  - HTTPS (via proxy)
  - SSH
  - DNS
  - PING

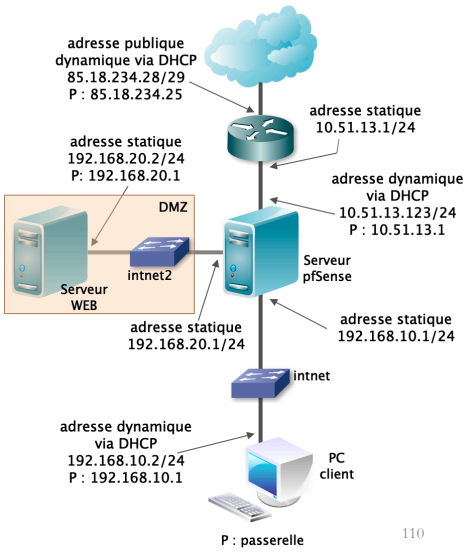
Scénario 2: mise en place d'un firewall

- Pour la mise en place de ce firewall, pas besoin d'installer de service supplémentaire, ça se fait avec les éléments internes de pfSense

Schéma réseau 2

- Un deuxième schéma réseau incluant une DMZ:





Conclusion

- Cette introduction à pfSense permet de préssentir les différentes possibilités de cet outil
- Ce cours va bien sur s'étoffer au cours du temps