Cisco Networking Academy
Mind Wide Open

Chapter 4: Implémentation des technologies Firewall

CCNA Security v2.0

Samir DIABI

Plan du chapitre

4.0 Introduction

4.1 Access Control Lists

4.2 Technologies des Firewall

4.3 Zone-Based Policy Firewalls

4.4 Résumé

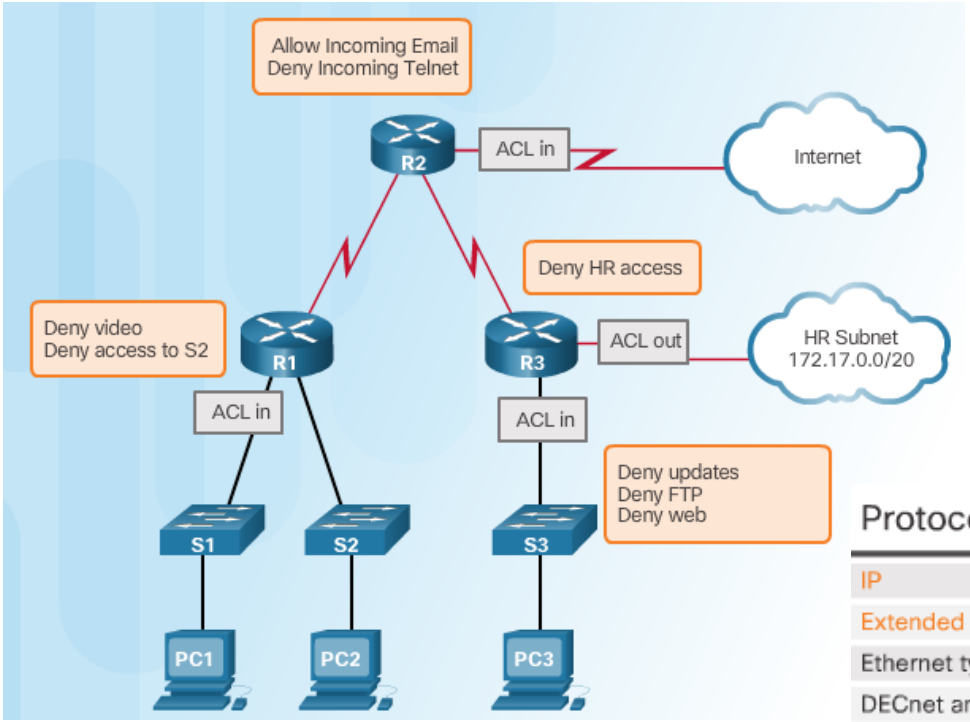
Section 4.1: Access Control List

A La fin de cette section, vous devez être capable de:

- Configurer les ACL IPv4 standard and extended avec CLI.
- Utiliser les ACLs pour faire face aux attaques réseaux
- Configurer ACLs IPv6 avec CLI.

4.1.1:
Configurer les ACL IPv4 standard and extended avec CLI.

Introduction aux Access Control Lists



Protocol	Range
IP	1-99, 1300-1999
Extended IP	100-199, 2000-2699
Ethernet type code	200-299
DECnet and Extended DECnet	300-399
XNS	400-499
Extended XNS	500-599
AppleTalk	600-699
Ethernet address	700-799
IPX	800-899
Extended IPX	900-999
IPX SAP	1000-1099
Extended transparent bridging	1100-1199

Configurer les ACL numérotées et nommées

Syntaxe d'une ACL Standard

```
access-list {acl-#} {permit | deny | remark} source-addr [source-wildcard][log]
```

Syntaxe d'une ACL Etendue Numérotée

```
access-list acl-# {permit | deny | remark} protocol source-addr [source-wildcard]
dest-addr [dest-wildcard][operator port][established]
```

Syntaxe d ACL nommée

```
Router(config)# ip access-list [standard | extended] name_of_ACL
```

Syntaxe d'une ACE Standard

```
Router(config-std-nacl)# {permit | deny | remark} {source [source-wildcard] | any}
```

Syntaxe d'une ACE étendue

```
Router(config-ext-nacl)# {permit | deny | remark} protocol source-addr [source-wildcard]
dest-address [dest-wildcard] [operator port]
```

Appliquer une ACL

Syntaxe pour appliquer une ACL à une interface

```
Router(config-if)# ip access-group {acl-#|name} {in|out}
```

Syntaxe pour appliquer une ACL aux VTY lines

```
Router(config-line)# access-class {acl-#|name} {in|out}
```

Example – ACL standard nommée

```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group NO_ACCESS out
```

Example – ACL étendue nommée

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
```

Appliquer ACL(Cont.)

Syntaxe d'application d'une ACL aux lines VTY

```
Router(config-line)# access-class {acl-#|name} {in|out}
```

Example – ACL Nommée pour VTY lines

```
R1(config)# ip access-list standard VTY_ACCESS
R1(config-std-nacl)# permit 192.168.10.10 log
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# access-class VTY_ACCESS in
R1(config-line)# end
R1#
R1#!The administrator accesses the vty lines from 192.168.10.10
R1#
*Feb 26 18:58:30.579: %SEC-6-IPACCESSLOGNP: list VTY_ACCESS permitted 0
192.168.10.10 -> 0.0.0.0, 5 packets
R1# show access-lists
Standard IP access list VTY_ACCESS
    10 permit 192.168.10.10 log (6 matches)
    20 deny any
```

ACL- Guide de configuration

- Créer une ACL globale et après appliquer la
- Assurer que la dernière ligne est une règle implicite deny any ou deny any any
- Rappelez vous que l'ordre des règles est très important parce que le ACL sont traitées du haut vers le bas. Dés que les conditions sont correct l'ACL s'arrête
- Rappelez vous que la règle spécifique doit être en haut de la liste
- Rappelez vous que seulement une seule ACL est autorisée par interface par protocole par direction
- Rappelez vous que une nouvelle règle est ajoutée à la fin de la liste par défaut
- Rappelez vous que les paquets générés par le routeur ne sont pas filtré par les ACL sortantes
- Placer les ACL standard au plus proche de la destination
- Placer les ACL étendues au plus proche de la source

Editer une Acl existante

Liste de contrôle d'accès avec 3 Entrées

```
Router# show access-lists
Extended IP access list 101
    10 permit tcp any any
    20 permit udp any any
    30 permit icmp any any
```

Access list éditée, en ajoutant un nouvel ACE et remplace la ligne ACE 20.

```
Router(config)# ip access-list extended 101
Router(config-ext-nacl)# no 20
Router(config-ext-nacl)# 5 deny tcp any any eq telnet
Router(config-ext-nacl)# 20 deny udp any any
```

access list actualisée avec quatre entrées

```
Router# show access-lists
Extended IP access list 101
     5 deny tcp any any eq telnet
    10 permit tcp any any
    20 deny udp any any
    30 permit icmp any any
```

Les numéros de séquence et les ACL Standard

ACL existante avec 4 entrées

```
router# show access-lists
Standard IP access list 19
    10 permit 192.168.100.1
    20 permit 10.10.10.0, wildcard bits 0.0.0.255
    30 permit 201.101.110.0, wildcard bits 0.0.0.255
    40 deny any
```

Liste d'accès a été modifié, avec une nouvelle ACE qui autorise une adresse IP spécifique.

```
router(config)# ip access-list standard 19
router(config-std-nacl)# 25 permit 172.22.1.1
```

Modifier une ACL en ajoutant une régle avant la ligne 20

```
router# show access-lists
Standard IP access list 19
    10 permit 192.168.100.1
    25 permit 172.22.1.1
    20 permit 10.10.10.0, wildcard bits 0.0.0.255
    30 permit 201.101.110.0, wildcard bits 0.0.0.255
    40 deny any
```

4.1.2:
Faire face aux attaques avec ACLs

Antispoofing avec ACLs

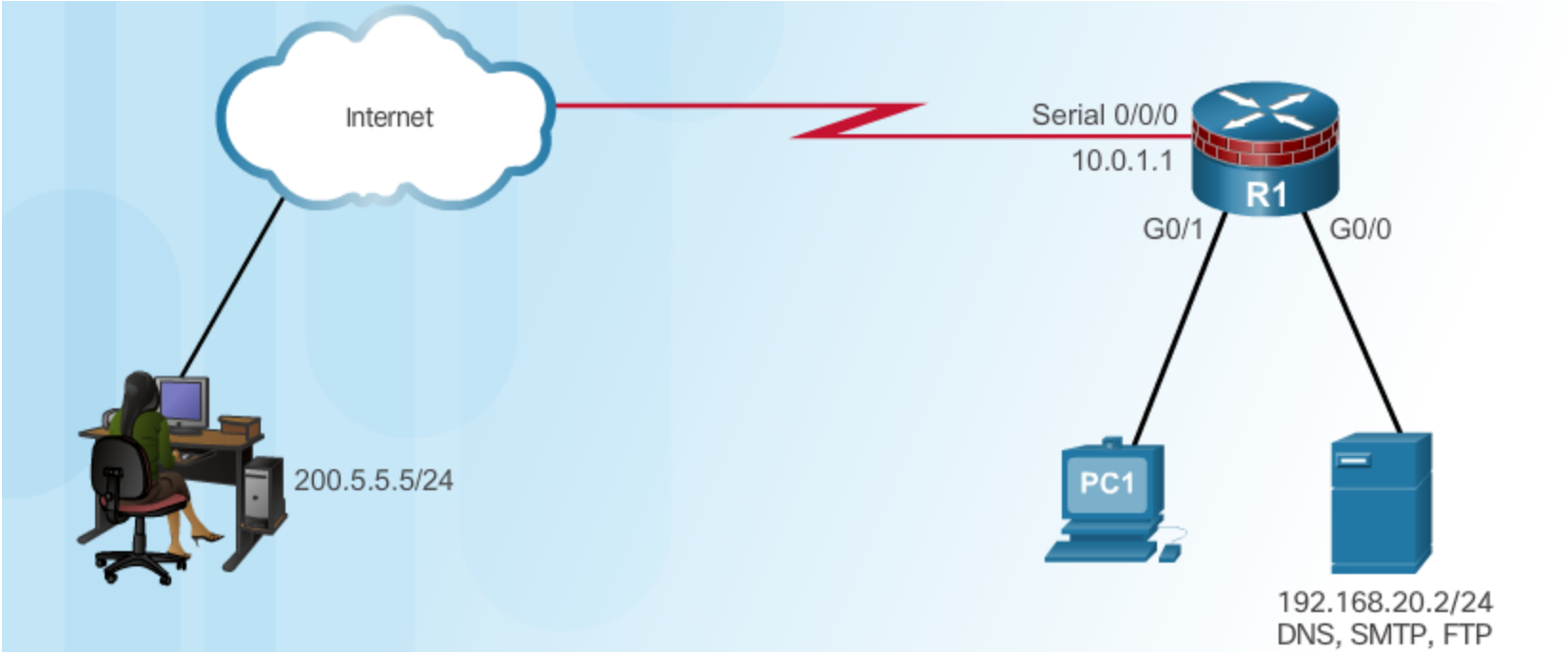
Inbound on S0/0/0

```
R1(config)# access-list 150 deny ip 0.0.0.0 255.255.255.255 any
R1(config)# access-list 150 deny ip 10.0.0.0 0.255.255.255 any
R1(config)# access-list 150 deny ip 127.0.0.0 0.255.255.255 any
R1(config)# access-list 150 deny ip 172.16.0.0 0.15.255.255 any
R1(config)# access-list 150 deny ip 192.168.0.0 0.0.255.255 any
R1(config)# access-list 150 deny ip 224.0.0.0 15.255.255.255 any
R1(config)# access-list 150 deny ip host 255.255.255.255 any
```

Inbound on G0/0

```
R1(config)# access-list 105 permit ip 192.168.1.0 0.0.0.255 any
```

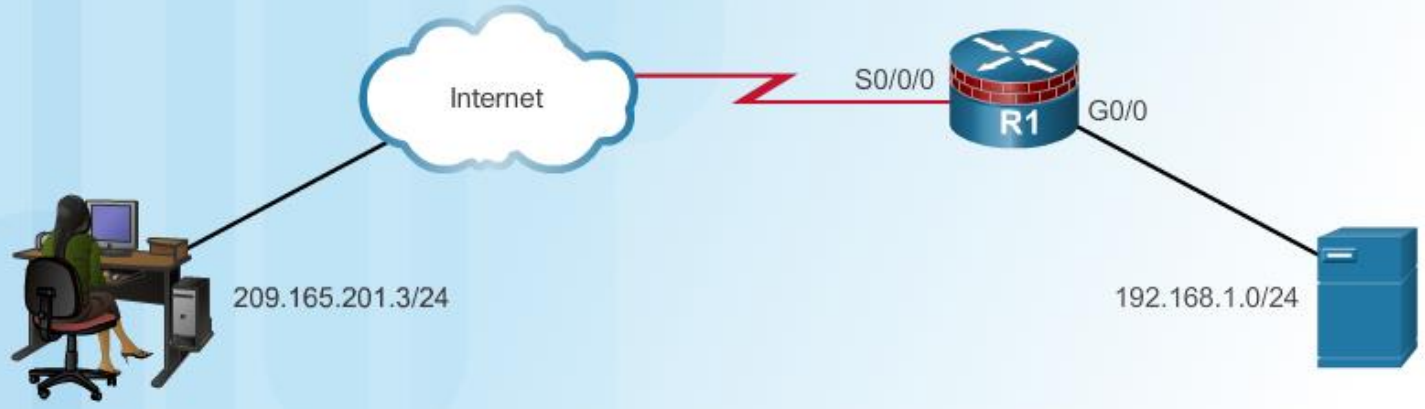
Autoriser le trafic nécessaire à travers un pare-feu



Inbound on Serial 0/0/0

```
R1(config)# access-list 180 permit udp any host 192.168.20.2 eq domain
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq smtp
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq ftp
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 10.0.1.1 eq 22
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq syslog
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq snmptrap
```

Faire face aux abus ICMP



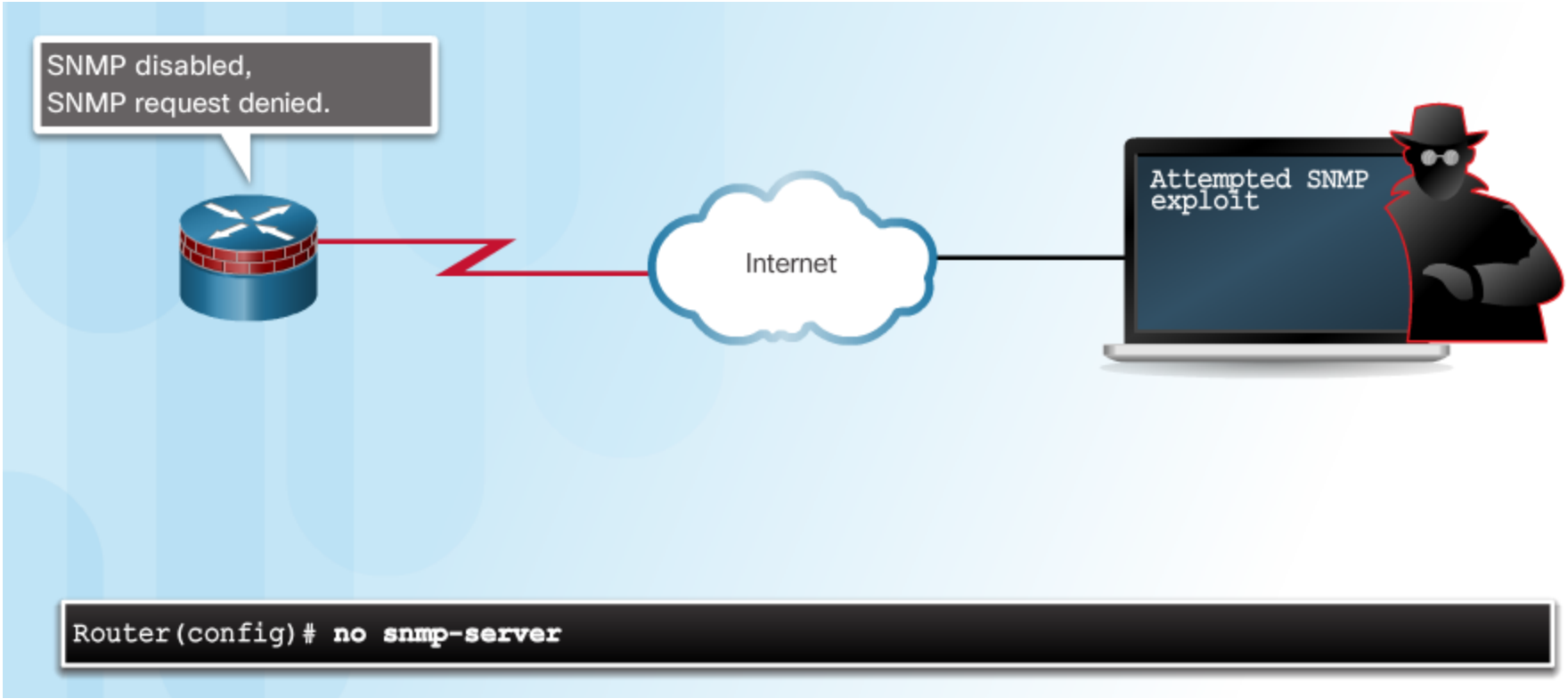
Inbound on S0/0/0

```
R1(config)# access-list 112 permit icmp any any echo-reply
R1(config)# access-list 112 permit icmp any any source-quench
R1(config)# access-list 112 permit icmp any any unreachable
R1(config)# access-list 112 deny icmp any any
R1(config)# access-list 112 permit ip any any
```

Inbound on G0/0

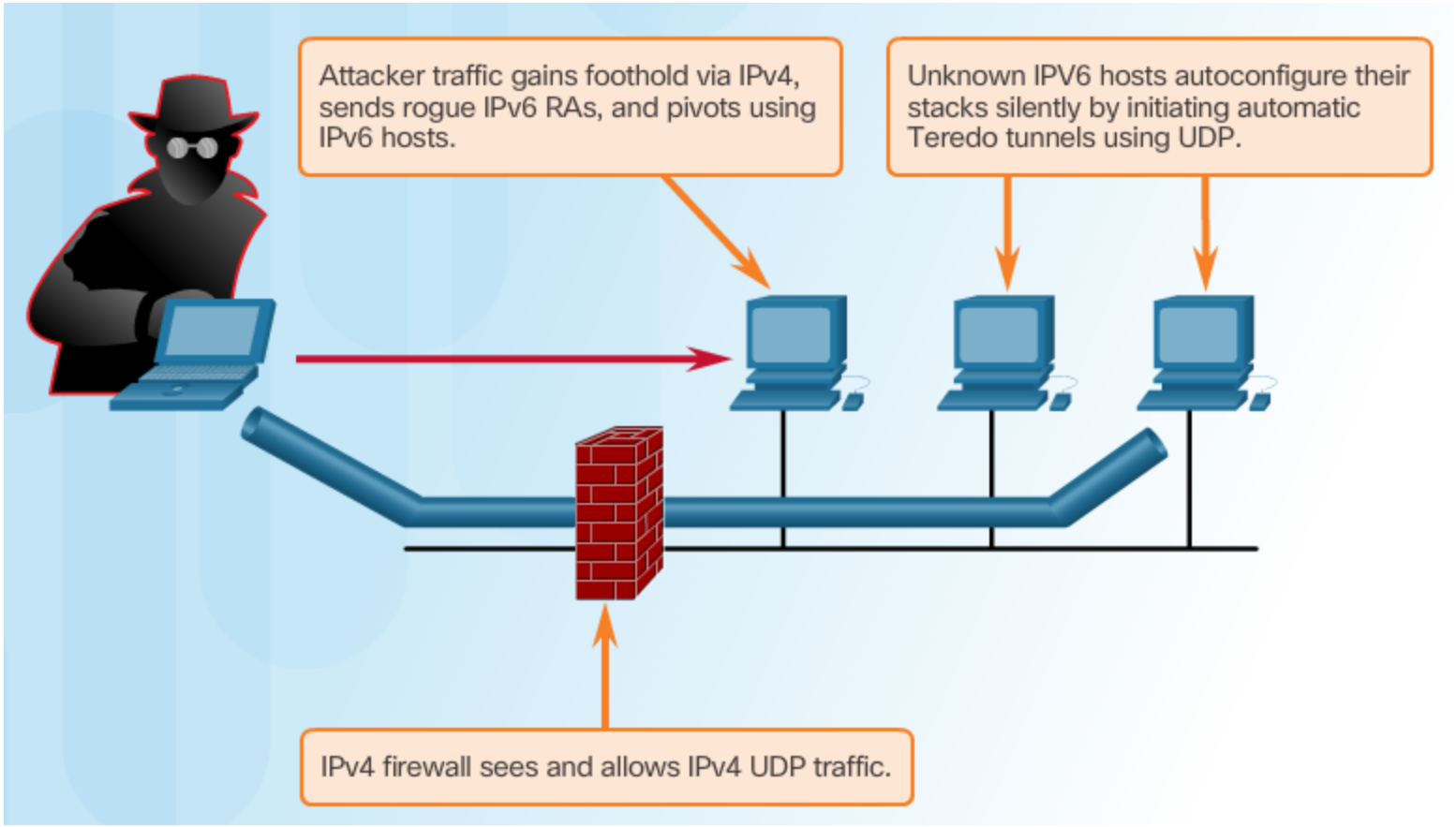
```
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any echo
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any parameter-problem
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any packet-too-big
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any source-quench
R1(config)# access-list 114 deny icmp any any
R1(config)# access-list 114 permit ip any any
```

Faire face aux SNMP Exploits



4.1.3:
IPv6 ACLs

Introduction aux ACLs IPv6



Syntaxe ACL IPv6

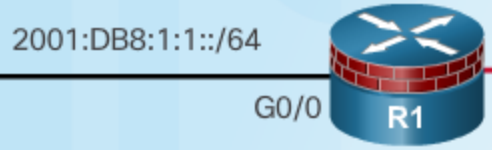
```
R1(config)# ipv6 access-list access-list-name
R1(config-ipv6-acl)# deny | permit protocol {source-ipv6-prefix/prefix-length | any | host
source-ipv6-address} [operator [port-number]] {destination-ipv6-prefix/prefix-length | any |
host destination-ipv6-address} [operator [port-number]]
```

Parameter	Description
`deny | permit`	Specifies whether to deny or permit the packet.
`protocol`	Enter the name or number of an Internet protocol, or an integer representing an IPv6 protocol number.

`source-ipv6-prefix/prefix-length`
 	
`destination-ipv6-address`
	The source or destination IPv6 network or class of networks for which to set deny or permit conditions.

`any`	Enter `any` as an abbreviation for the IPv6 prefix ::/0. This matches all addresses.
`host`	For `host source-ipv6-address` or `destination-ipv6-address`, enter the source or destination IPv6 host address for which to set deny or permit conditions.
`operator`	(Optional) An operand that compares the source or destination ports of the specified protocol. Operands are lt (less than), gt (greater than), eq (equal), neq (not equal), and range.
`port-number`	(Optional) A decimal number or the name of a TCP or UDP port for filtering TCP or UDP, respectively.

Configurer ACLs IPv6




```
R1(config)# ipv6 access-list LAN_ONLY
R1(config-ipv6-acl)# permit 2001:db8:1:1::/64 any
R1(config-ipv6-acl)# permit icmp any any nd-na
R1(config-ipv6-acl)# permit icmp any any nd-ns
R1(config-ipv6-acl)# deny ipv6 any any
R1(config-ipv6-acl)# end
R1# show ipv6 access-list
IPv6 access list LAN_ONLY
    permit ipv6 2001:DB8:1:1::/64 any sequence 10
    permit icmp any any nd-na sequence 20
    permit icmp any any nd-ns sequence 30
    deny ipv6 any any sequence 40
R1#
```

Section 4.2: Les technologies des Pare-feu

À la fin de cette section, vous devez être capable de:

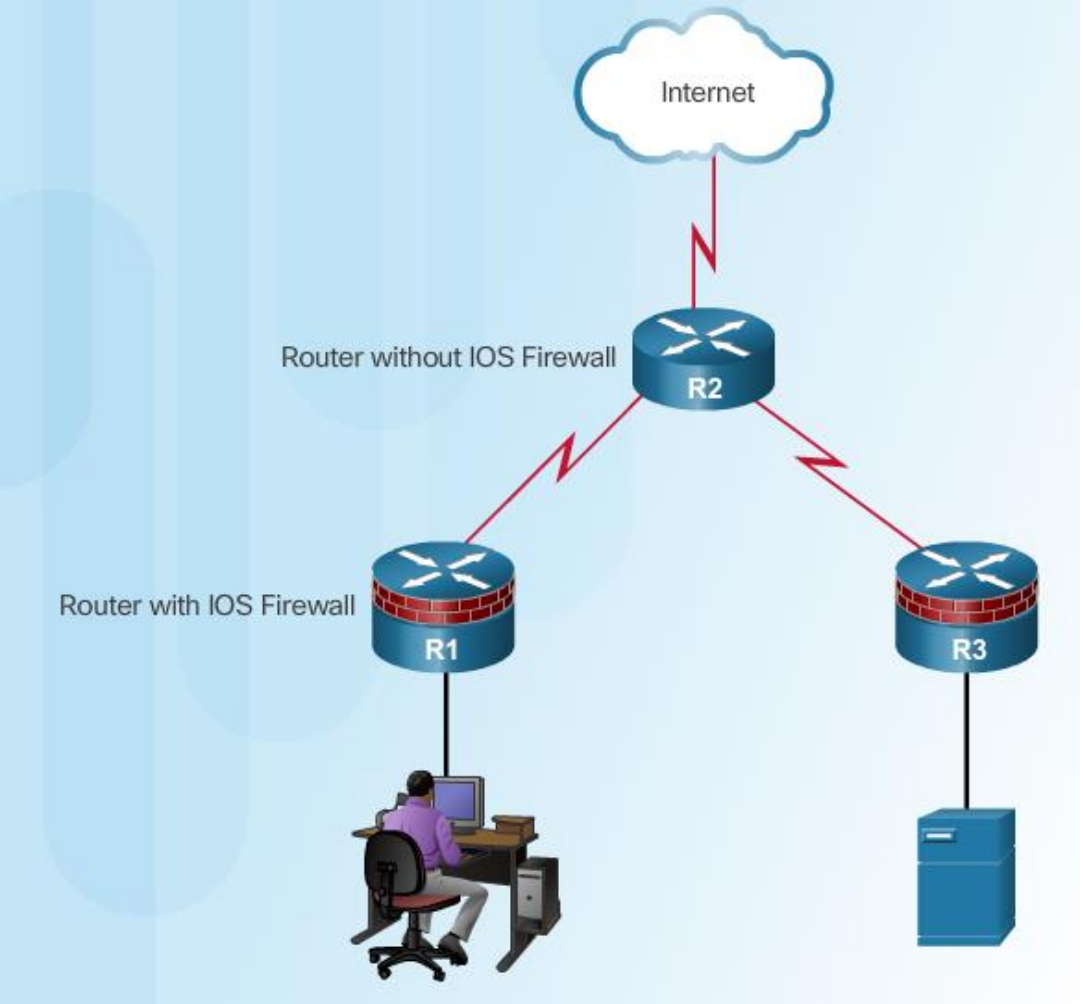
- Expliquer comment les pares-feu sont utilisés pour securiser les réseaux.
- Decrire les différents types des pare-feu
- Configurer un pare-feu classique
- Expliquer les régles de conception pour implementer les technologies des pare-feu

4.2.1:
Sécuriser un réseau avec des pare-feu

Définir les pare-feu

Tous les pare-feu:

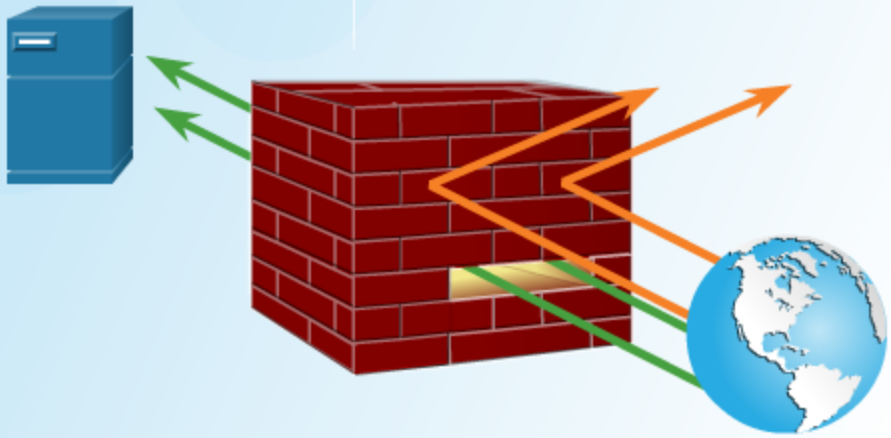
- Résistent aux attaques
- Sont le seul point de transit entre les réseaux parce que tous les flux de trafic passe à travers le pare-feu
- Faire respecter la politique de contrôle d'accès



Avantages et limites des Firewalls

- Allow traffic from any external address to the web server.
- Allow traffic to FTP server.
- Allow traffic to SMTP server.
- Allow traffic to internal IMAP server.

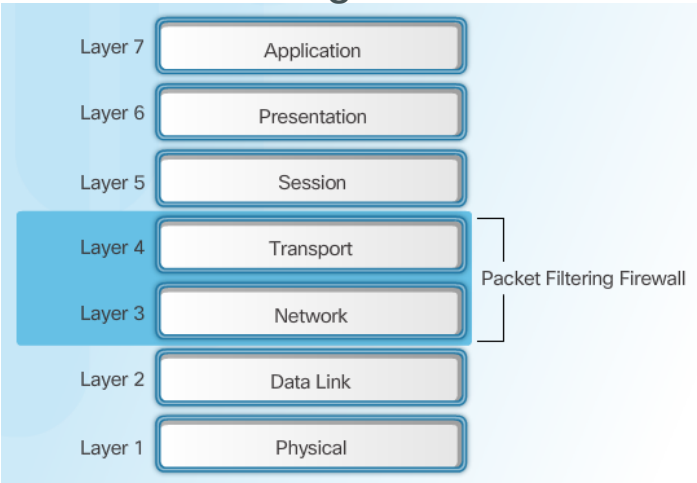
- Deny all inbound traffic with network addresses matching internal-registered IP addresses.
- Deny all inbound traffic to server from external addresses.
- Deny all inbound ICMP echo request traffic.
- Deny all inbound MS Active Directory.
- Deny all inbound MS SQL server ports.
- Deny all MS Domain Local Broadcasts.



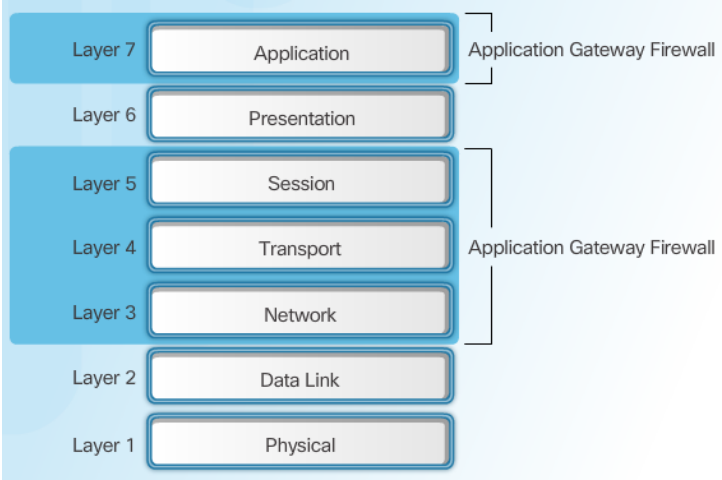
4.2.2:
Types des Firewalls

Type des Firewalls

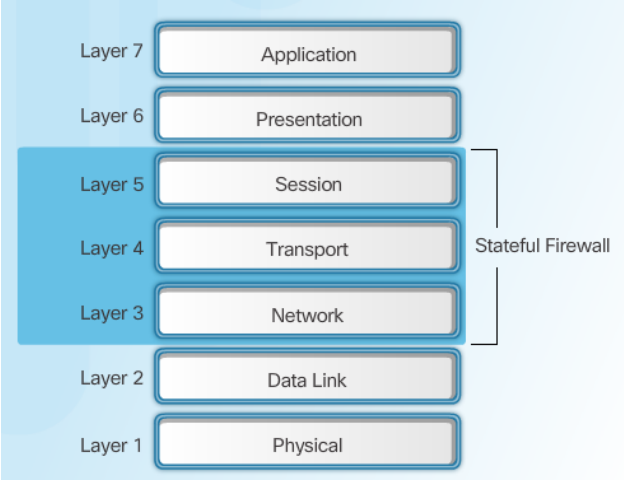
Packet Filtering Firewall



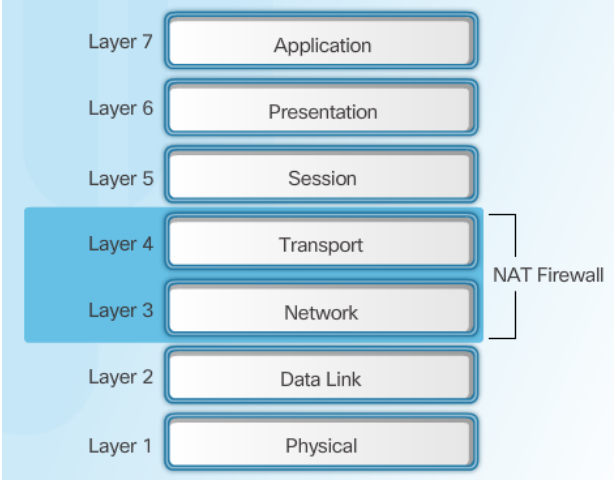
Application Gateway Firewall



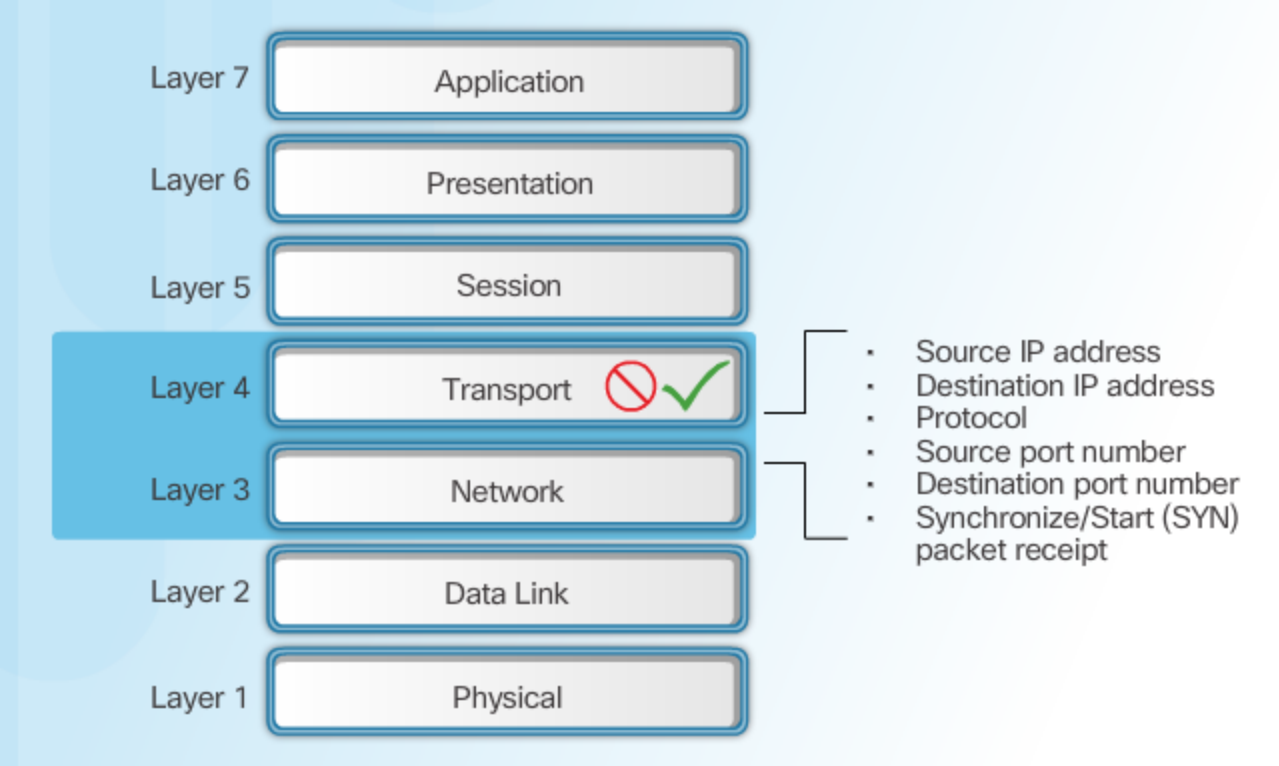
Stateful Firewall



NAT Firewall

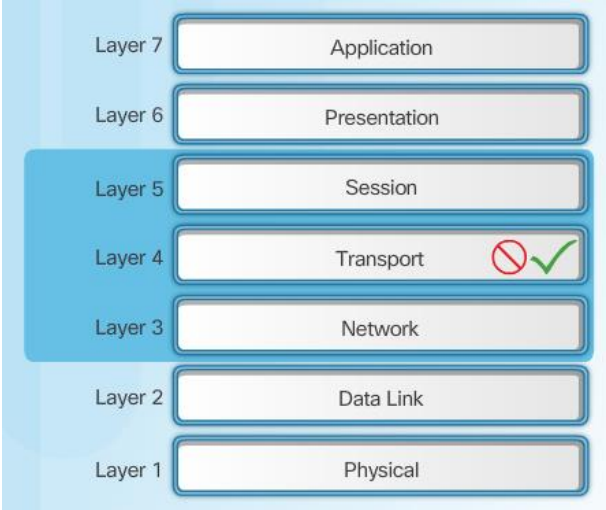


Filtrage des paquets par un pare-feu
Avantages et limites

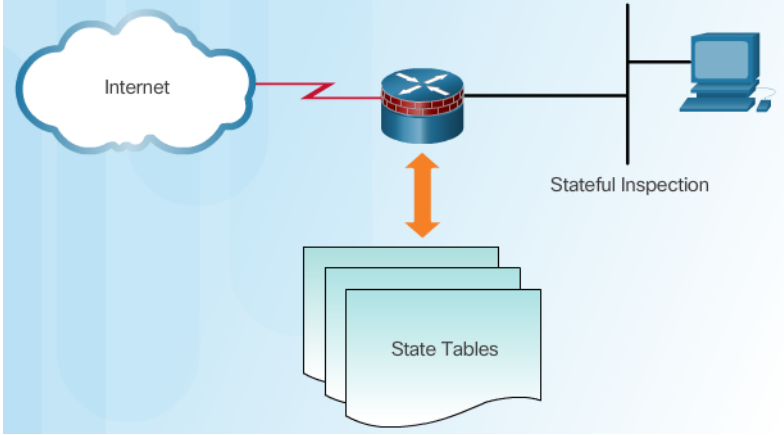


Stateful Firewalls

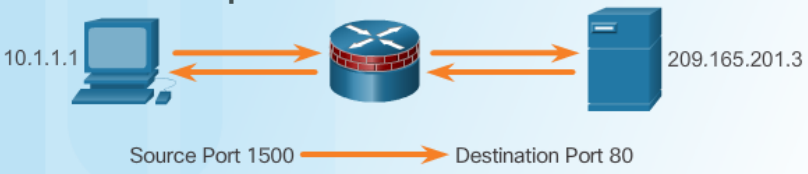
Stateful Firewalls



State Tables



Stateful Firewall Operation



Inside ACL (Outgoing Traffic)	Outside ACL (Incoming Traffic)
permit ip 10.0.0.0.0.0.0.255 any	Dynamic: permit tcp host 209.165.201.3 eq 80 host 10.1.1.1 eq 1500

Stateful Firewall avantages et limites

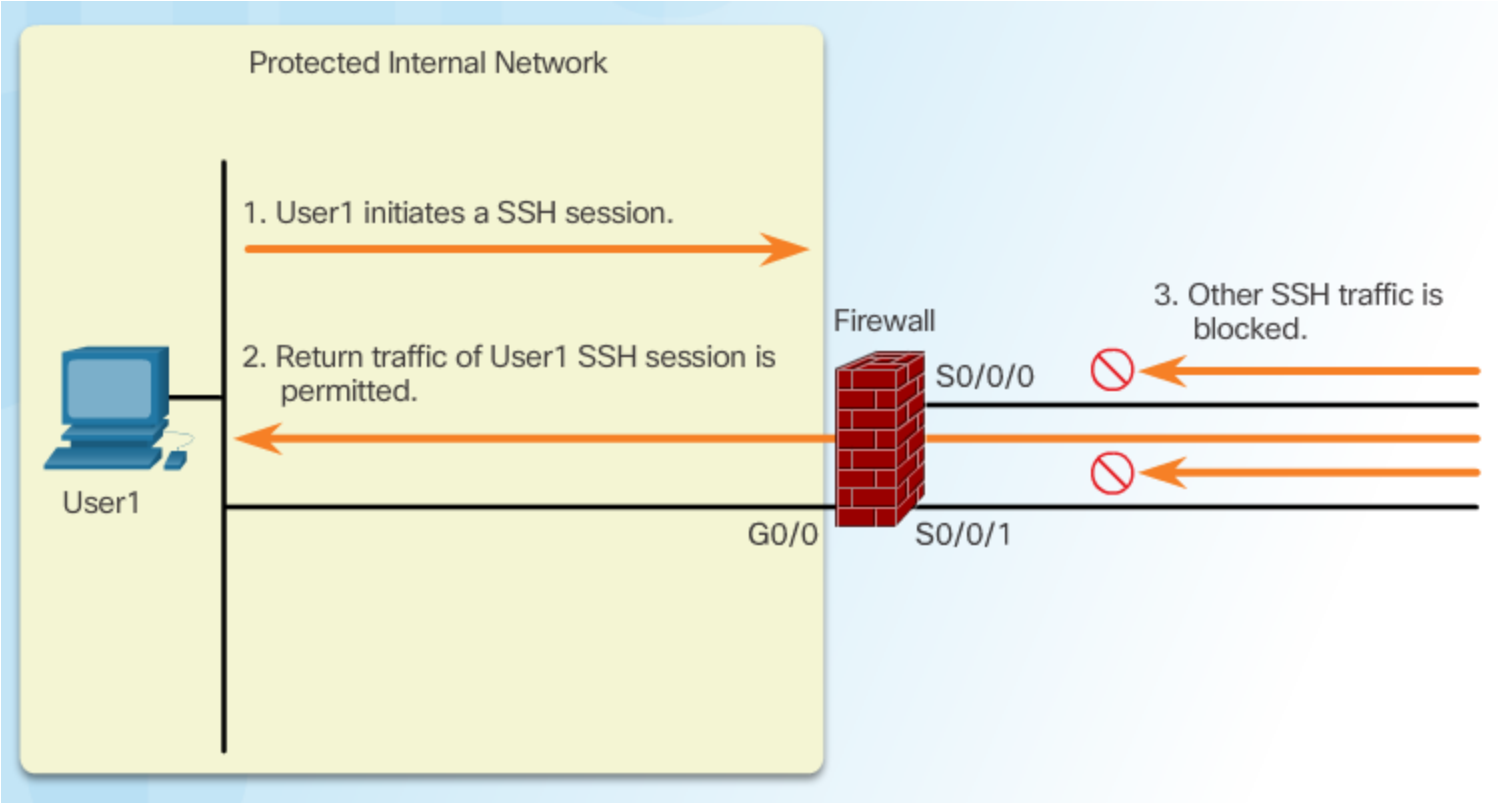
Benefits	Limitations
Primary means of defense	No Application Layer inspection
Strong packet filtering	Cannot filter stateless protocols
Improved performance over packet filters	Difficult to defend against dynamic port negotiation
Defends against spoofing and DoS attacks	No authentication support
Richer data log	

Next Generation Firewalls

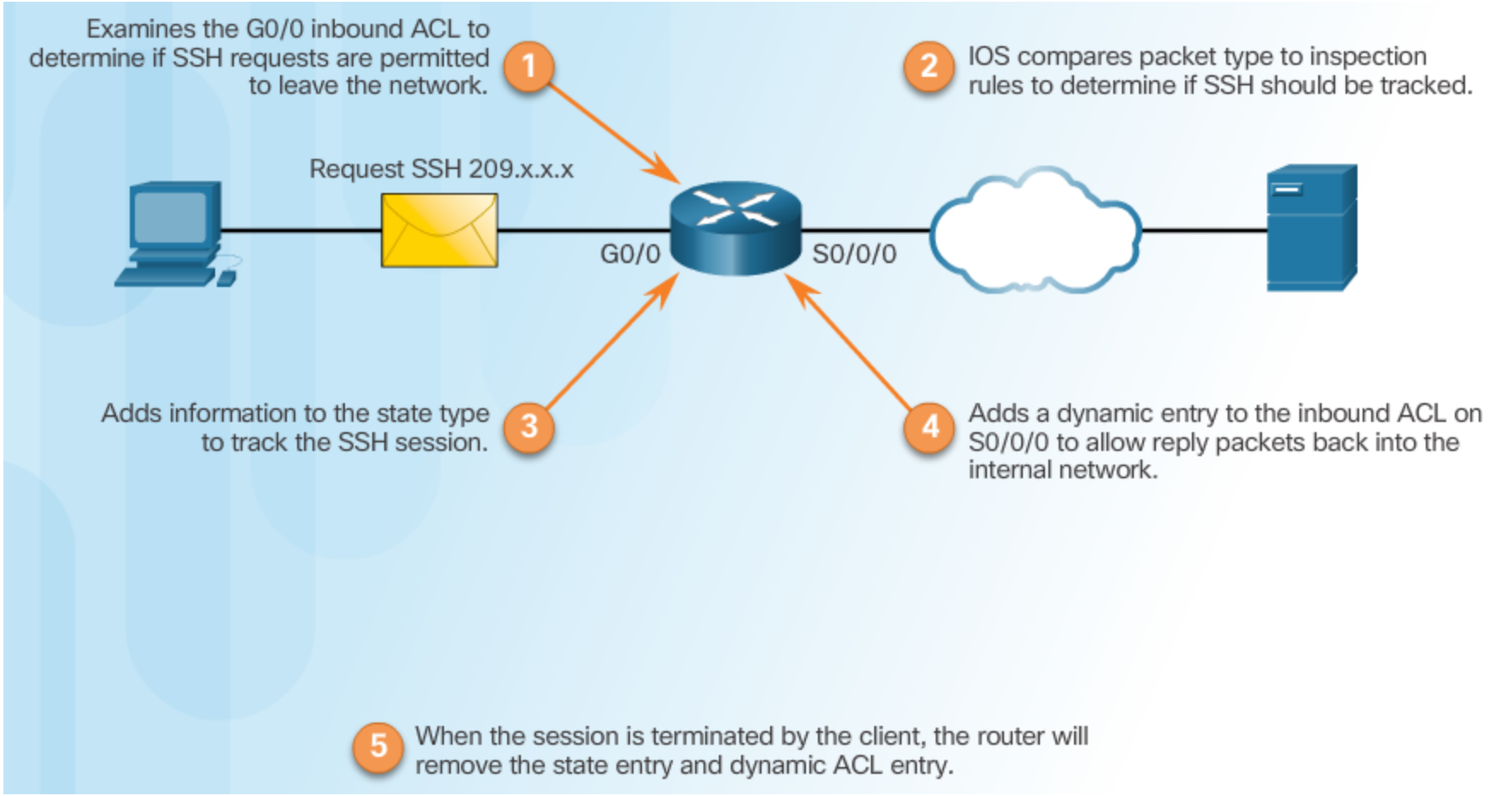
- Identification granuleuse, la visibilité et le contrôle des comportements au sein des applications.
- Restreindre l'utilisation du Web et de l'application web basée sur la réputation du site.
- protection proactive contre les menaces internet.
- L'application des politiques de sécurité basées sur l'utilisateur, matériel, role, type d'application , et les menaces
- NAT, VPN, et SPI performant
- Utilisation du IPS

Topic 4.2.3:
Pare-feu classique

Introduction aux pare-feu classique

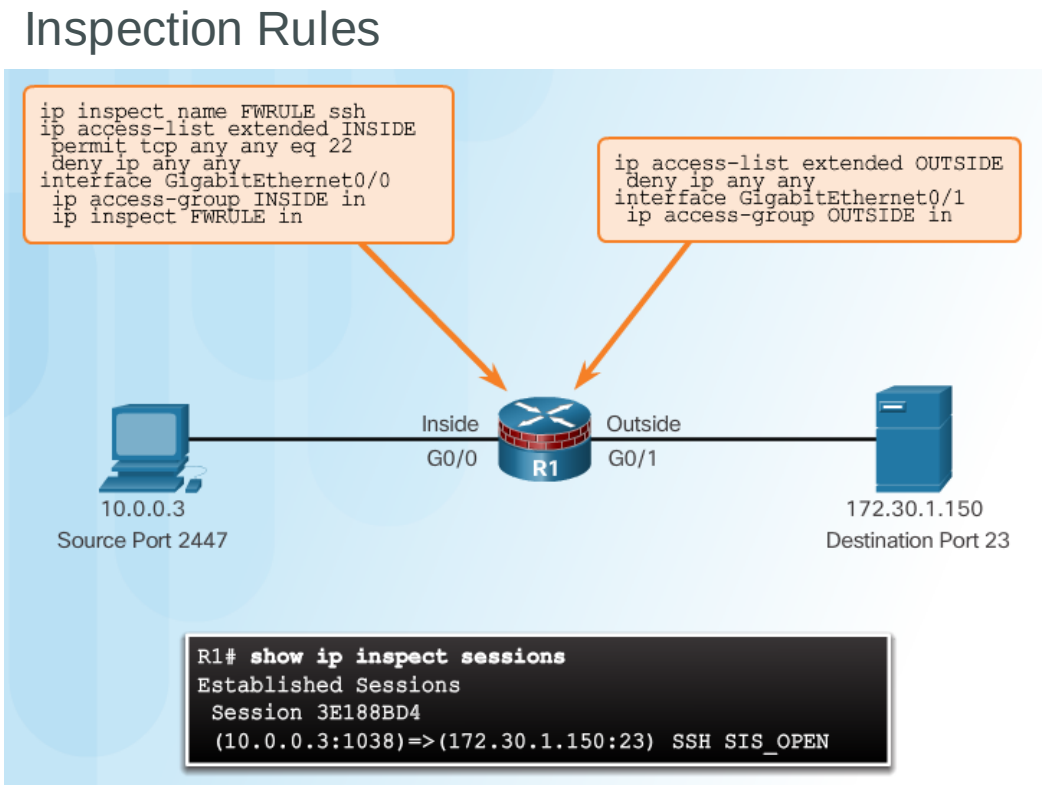


les operations des pare-feu classiques



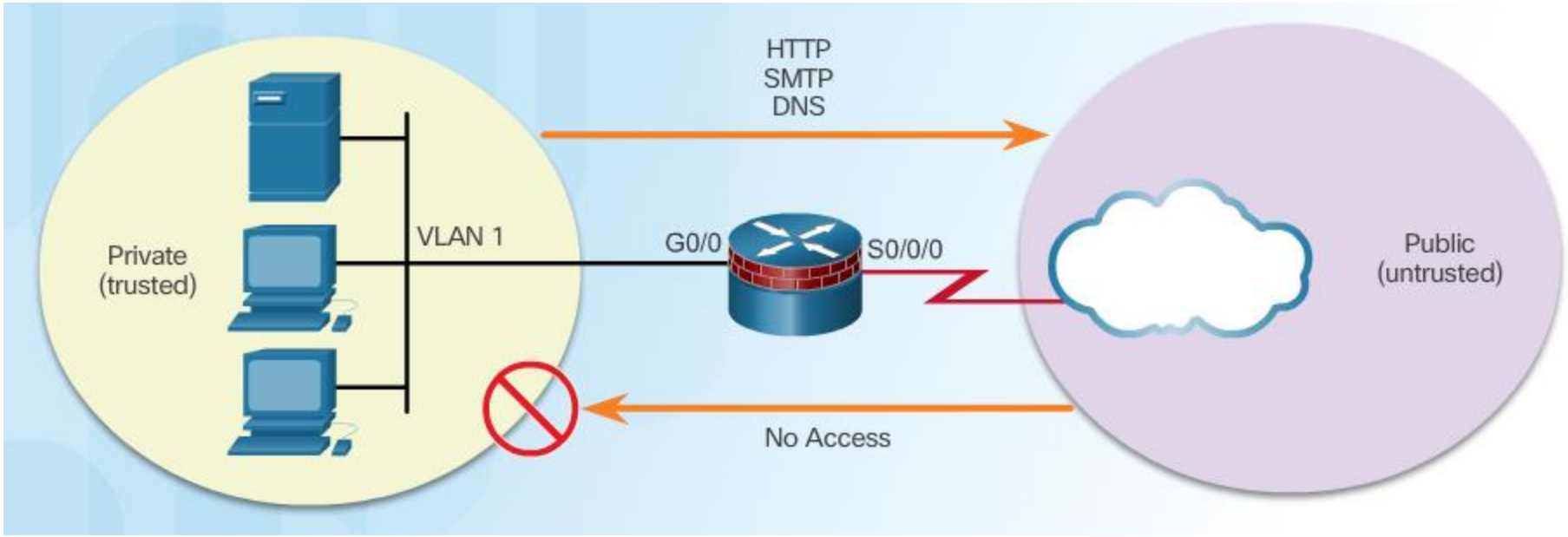
Configuration des pare-feu classiques

1. Choisir les interfaces internes et externes
2. Configurer les ACL pour chaque interface.
3. Définir inspection rules.
4. Appliquer une inspection rule à une interface.

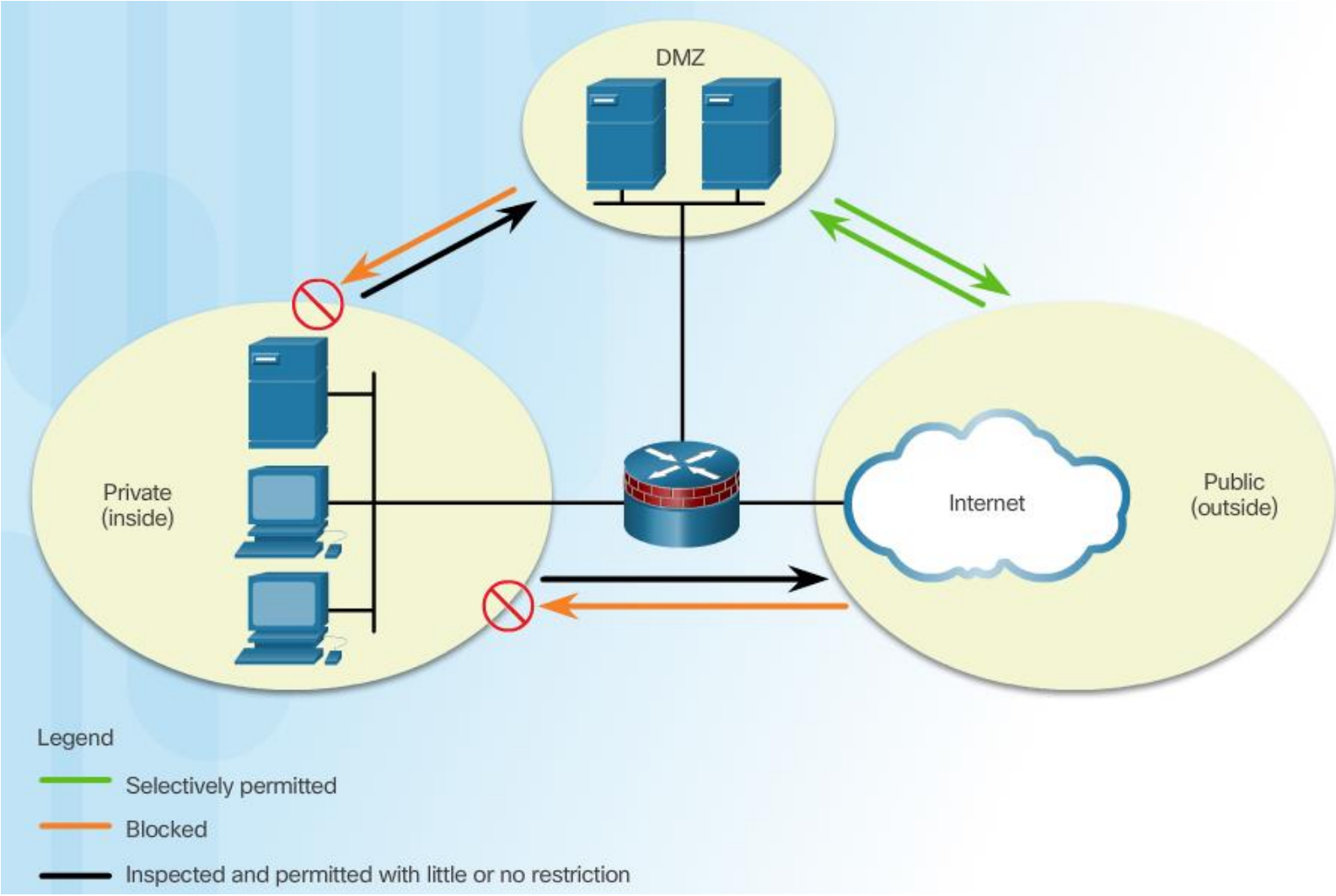


Topic 4.2.4:
les pare-feu dans la conception des réseaux

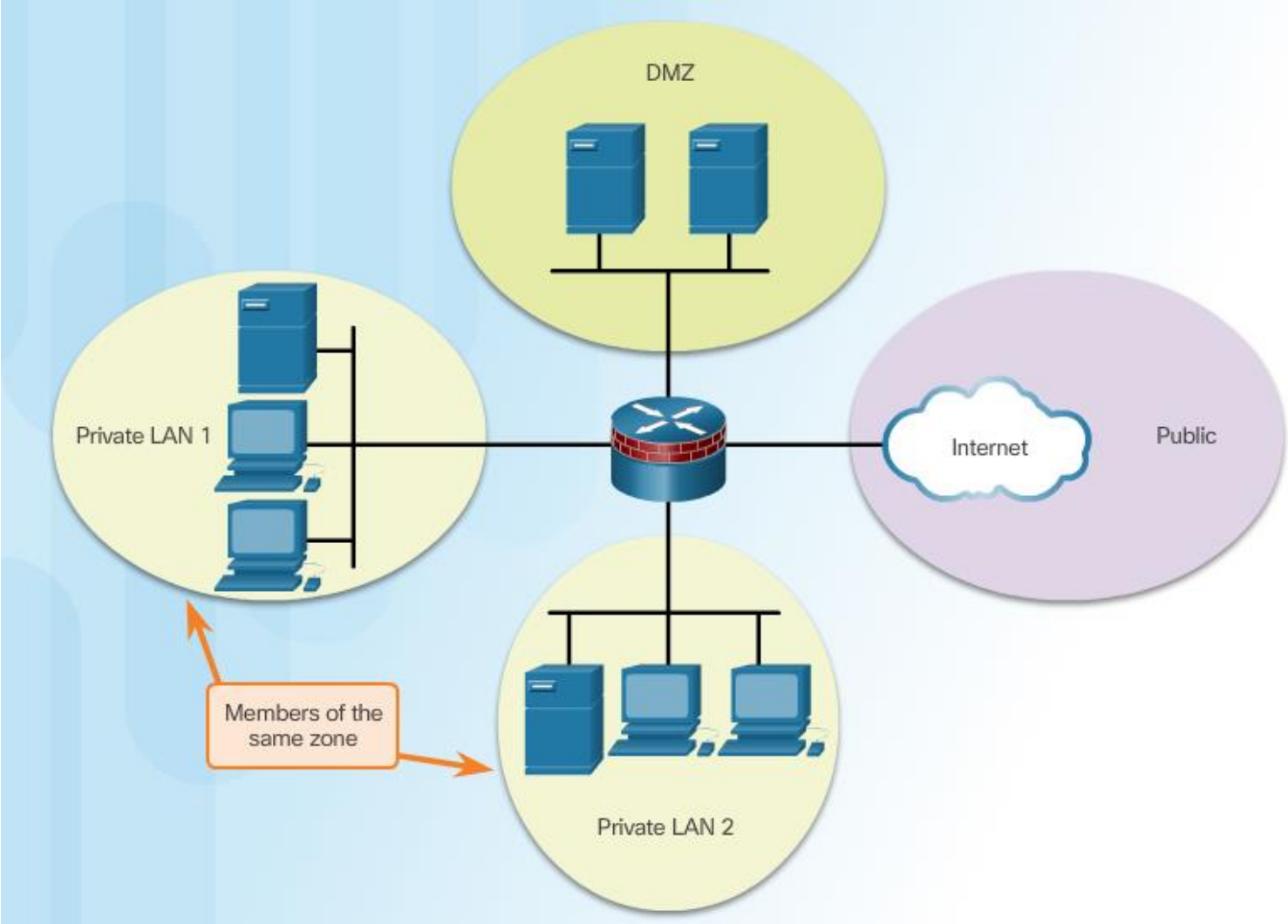
Les réseaux internes et externes



Demilitarized Zones



Zone-Based Policy Firewalls



Défense en couches

Considerations pour protéger le réseau:

- Network core security
- Perimeter security
- Endpoint security
- Communications security

Bonnes pratiques des pare-feu

- Positionner le pare-feu aux limites de sécurité.
- Utiliser exclusivement à un pare-feu pour la sécurité est insuffisant
- Refuser tout le traffic par défaut. Autoriser seulement les services nécessaires
- Assurer que l'accès physique au pare-feu est controlé
- Consulter les journaux (logs) firewall.
- Pratiquer la gestion des changements pour les changements de configuration du pare-feu.
- Rappelez-vous que les pare-feu protègent principalement des attaques techniques provenant de l'extérieur.

Section 4.3: Zone-Based Policy Firewalls

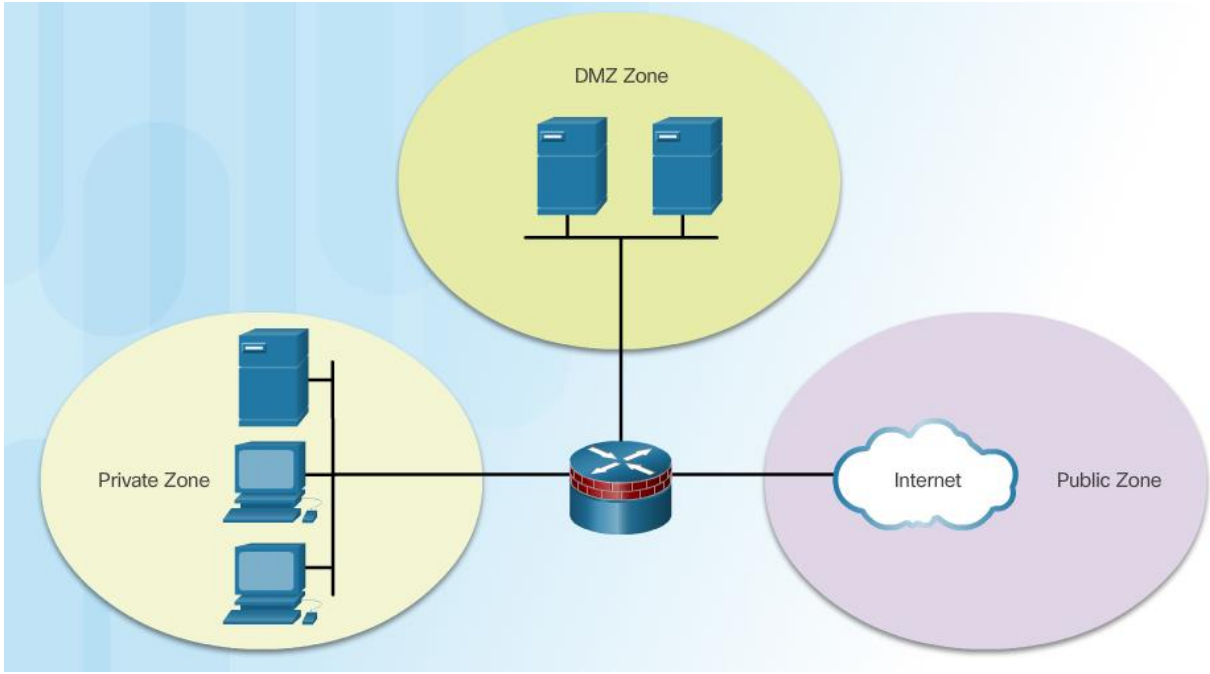
A la fin de cette section vous devez être capable de :

- Expliquer comment Zone-Based Policy Firewalls est utilisé pour aider à sécuriser le Réseau
- Expliquer l'exploitation des Zone-Based Policy Firewall.
- Configurer Zone-Based Policy Firewall avec CLI.

4.3.1:
Aperçu des Zone-Based Policy Firewall

Bénéfices des ZPF

- Indépendant des ACL
- La sécurité du routeur doit bloquer sauf autorisation explicite
- Les stratégies sont simple à lire et à dépanner avec C3PL
- Une stratégie affecte n'importe quel trafic, au lieu d'avoir besoin de plusieurs ACL et d'actions d'inspection



Conception ZPF

Les conceptions courantes comprennent: :

- LAN-to-Internet
- Firewalls entre serveurs publics
- Pare-feu redondants
- Firewalls complexes

Etapes de conception:

1. Déterminer les zones
2. Etablir des politiques entre les zones
3. Concevoir l'infrastructure physique
4. Identifier les interfaces dans les zones et fusionner les exigences de trafic

4.3.2:
Exploitation des ZPF

ZPF Actions

- **Inspect** - Configures Cisco IOS stateful packet inspections.
- **Drop** - Analogous to a deny statement in an ACL. A log option is available to log the rejected packets.
- **Pass** - Analogous to a permit statement in an ACL. The pass action does not track the state of connections or sessions within the traffic.

Rules for Transit Traffic

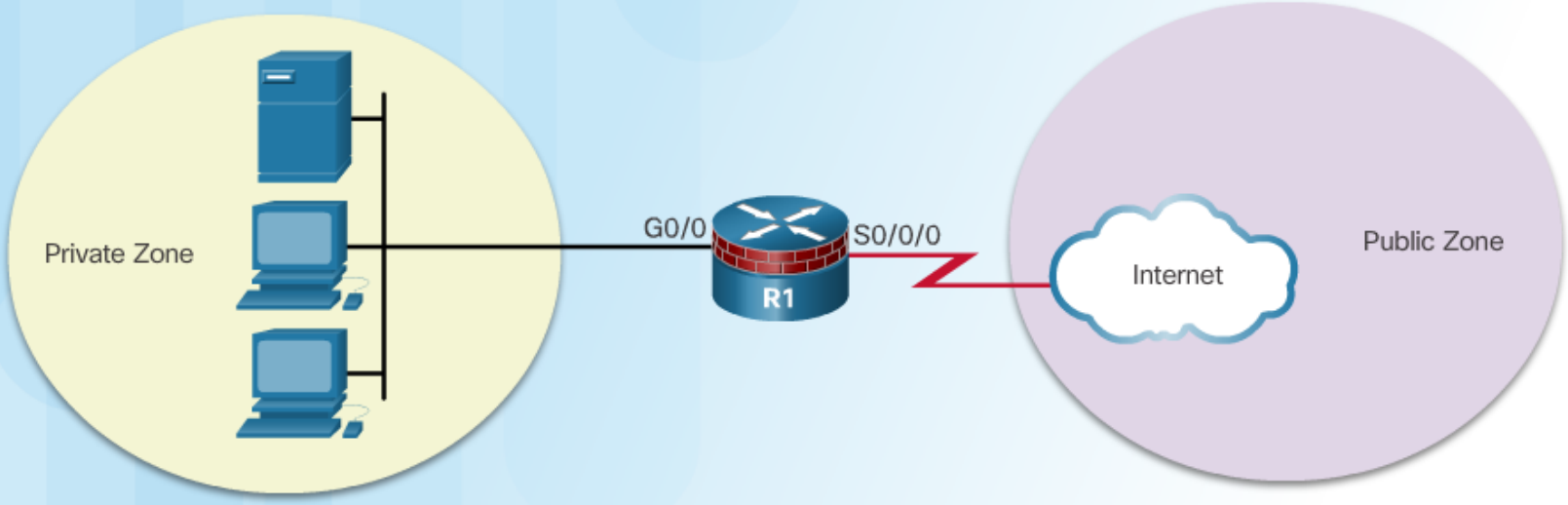
Source Interface Member of Zone?	Destination Interface Member of Zone?	Zone-Pair Exists?	Policy Exists?	Result
NO	NO	N/A	N/A	PASS
YES	NO	N/A	N/A	DROP
NO	YES	N/A	N/A	DROP
YES (private)	YES (private)	N/A	N/A	PASS
YES (private)	YES (public)	NO	N/A	DROP
YES (private)	YES (public)	YES	NO	PASS
YES (private)	YES (public)	YES	YES	INSPECT

Rules for Traffic to the Self Zone

Source Interface Member of Zone?	Destination Interface Member of Zone?	Zone-Pair Exists?	Policy Exists?	Result
YES (self-zone)	YES	NO	N/A	PASS
YES (self-zone)	YES	YES	NO	PASS
YES (self-zone)	YES	YES	YES	INSPECT
YES	YES (self-zone)	NO	N/A	PASS
YES	YES (self-zone)	YES	NO	PASS
YES	YES (self-zone)	YES	YES	INSPECT

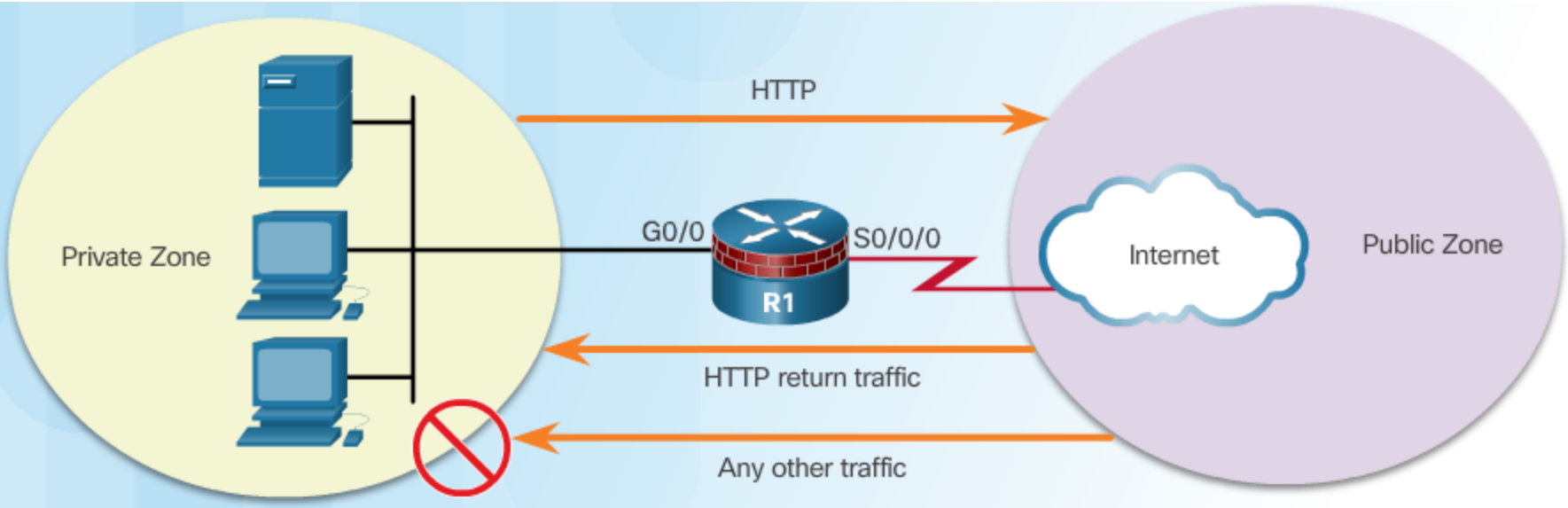
Topic 4.3.3: Configuring a ZPF

Configure ZPF



Step 1: Create the zones.
Step 2: Identify traffic with a class-map.
Step 3: Define an action with a policy-map.
Step 4: Identify a zone pair and match it to a policy-map.
Step 5: Assign zones to the appropriate interfaces.

Step 1: Create Zones



Syntax

```
Router(config)# zone security zone-name
```

Example

```
R1(config)# zone security PRIVATE
R1(config-sec-zone)# exit
R1(config)# zone security PUBLIC
```

Step 2: Identify Traffic

Command Syntax for `class-map`

```
Router(config)# class-map type inspect [match-any | match-all] class-map-name
```

Parameter	Description
match-any	Packets must meet one of the match criteria to be considered a member of the class.
match-all	Packets must meet all of the match criteria to be considered a member of the class.
class-map-name	Name of the class-map used to configure the policy for the class in the policy-map.

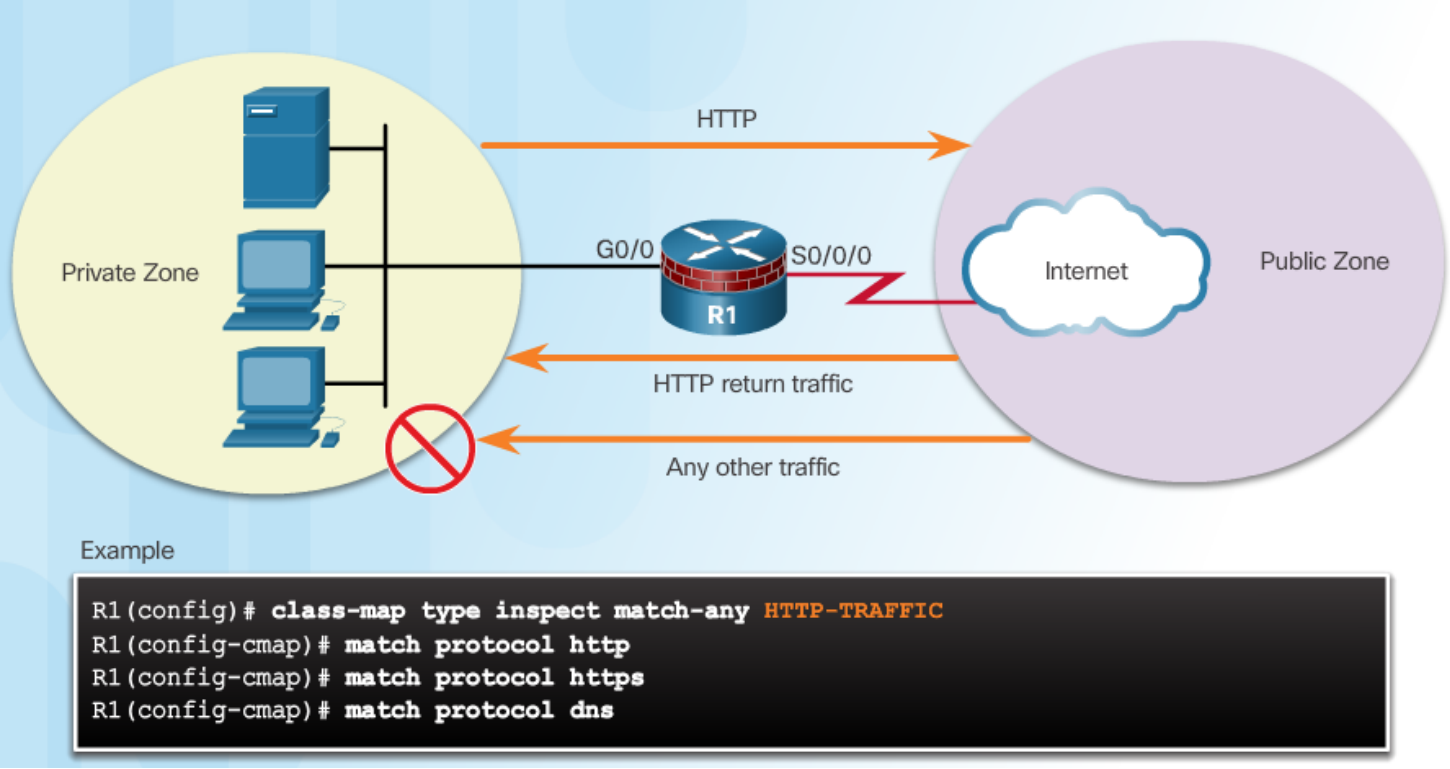
Sub-Configuration Command Syntax for `class-map`

```
Router(config-cmap)# match access-group {acl-# | acl-name }
Router(config-cmap)# match protocol protocol-name
Router(config-cmap)# match class-map class-map-name
```

Parameter	Description
match access-group	Configures the match criteria for a class-map based on the specified ACL number or name.
match protocol	Configures the match criteria for a class-map based on the specified protocol.
match class-map	Uses another class-map to identify traffic.

Step 2: Identify Traffic (Cont.)

Example `class-map` Configuration



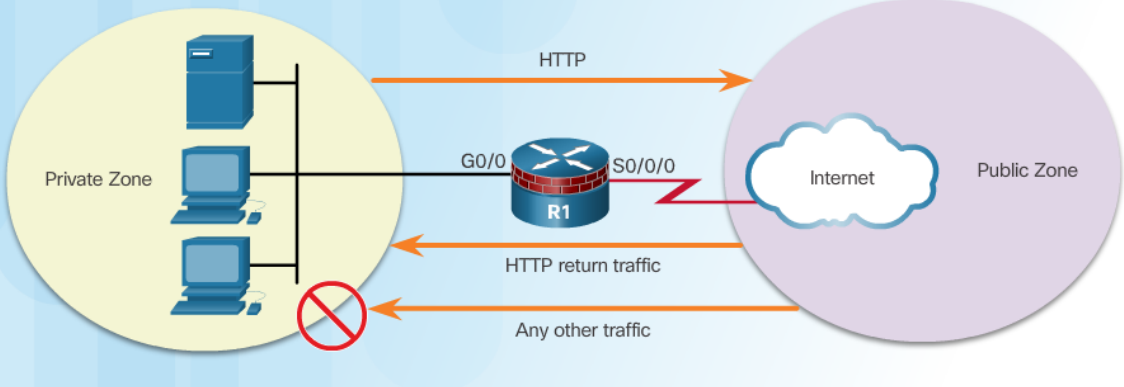
Step 3: Define an Action

Command Syntax for `policy-map`

```
Router(config)# policy-map type inspect policy-map-name
Router(config-pmap)# class type inspect class-map-name
Router(config-pmap-c)# { inspect | drop | pass }
```

Parameter	Description
inspect	An action that offers statebased traffic control. The router maintains session information for TCP and UDP and permits return traffic.
drop	Discards unwanted traffic
pass	A stateless action the allows the router to forward traffic from one zone to another

Example `policy-map` Configuration




```
R1(config)# policy-map type inspect PRIV-TO-PUB-POLICY
R1(config-pmap)# class type inspect HTTP-TRAFFIC
R1(config-pmap-c)# inspect
```

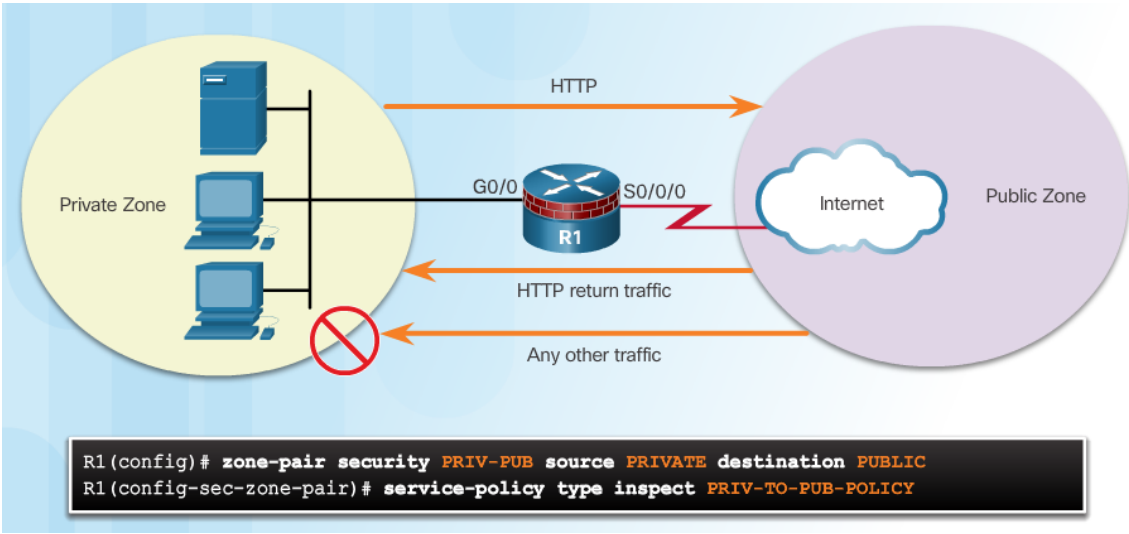
Step 4: Identify a Zone-Pair and Match to a Policy

Command Syntax for `zone-pair` and `service-policy`

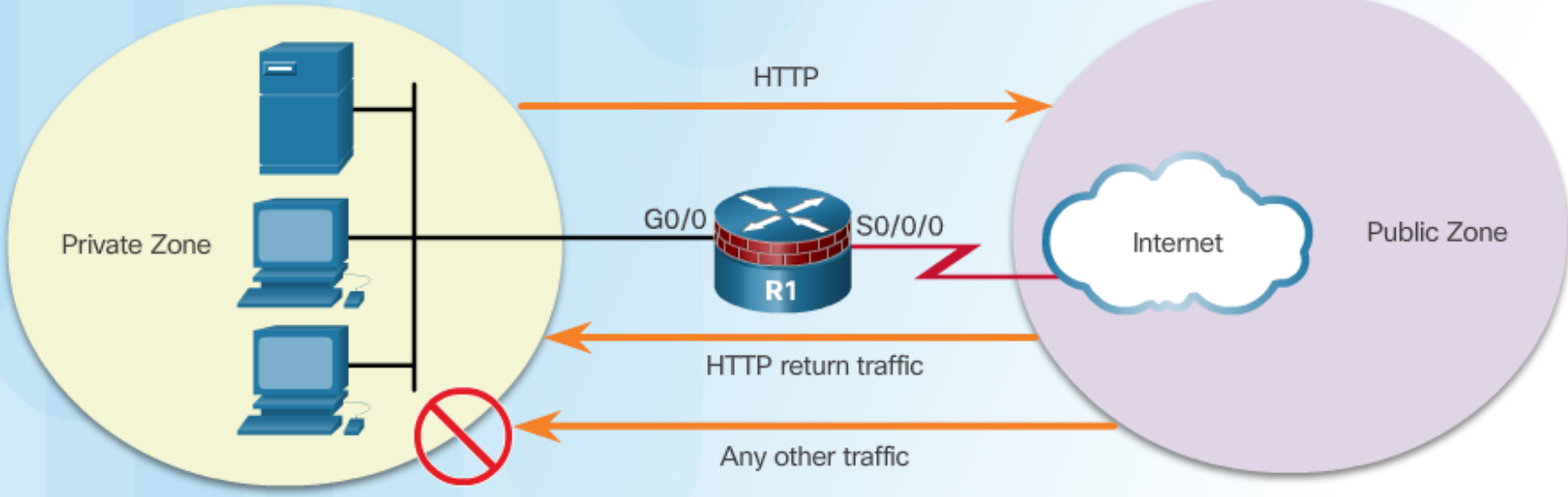
```
Router(config)# zone-pair security zone-pair-name source {source-zone-name | self
} destination {destination-zone-name | self }
Router(config-sec-zone-pair)# service-policy type inspect policy-map-name
```

Parameter	Description
`source source-zone-name`	Specifies the name of the zone from which traffic is originating.
`destination destination-zone-name`	Specifies the name of the zone to which traffic is destined.
`self`	Specifies the system-defined zone. Indicates whether traffic will be going to or from the router itself.

Example `service-policy` Configuration



Step 5: Assign Zones to Interfaces



Syntax

```
Router(config-if)# zone-member security zone-name
```

Example

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# zone-member security PRIVATE
R1(config-if)# interface Serial 0/0/0
R1(config-if)# zone-member security PUBLIC
```

Verify a ZPF Configuration

Verification commands:

- show run | begin class-map
- show policy-map type inspect zone-pair sessions
- show class-map type inspect
- show zone security
- show zone-pair security
- show policy-map type inspect

ZPF Configuration Considerations

- No filtering is applied for intra-zone traffic
- Only one zone is allowed per interface.
- No Classic Firewall and ZPF configuration on same interface.
- If only one zone member is assigned, all traffic is dropped.
- Only explicitly allowed traffic is forwarded between zones.
- Traffic to the self zone is not filtered.

Section 4.4: Summary

Chapter Objectives:

- Implement ACLs to filter traffic and mitigate network attacks on a network.
- Configure a classic firewall to mitigate network attacks.
- Implement ZPF using CLI.

Thank you.

Cisco Networking Academy

Mind Wide Open

Instructor Resources

- **Remember**, there are helpful tutorials and user guides available via your NetSpace home page. (https://www.netacad.com)
- These resources cover a variety of topics including navigation, assessments, and assignments.
- A screenshot has been provided here highlighting the tutorials related to activating exams, managing assessments, and creating quizzes.