

# CCNA Security 2.0 Study Material – Chapter 4: Implementing Firewall Technologies

---

 [itexamanswers.net/ccna-security-2-0-study-material-chapter-4-implementing-firewall-technologies.html](http://itexamanswers.net/ccna-security-2-0-study-material-chapter-4-implementing-firewall-technologies.html)

October 6, 2017

## Chapter Outline:

---

### 4.0 Introduction

### 4.1 Access Control Lists

### 4.2 Firewall Technologies

### 4.3 Zone-Based Policy Firewalls

### 4.4 Summary

---

## Section 4.1: Access Control List

---

Upon completion of this section, you should be able to:

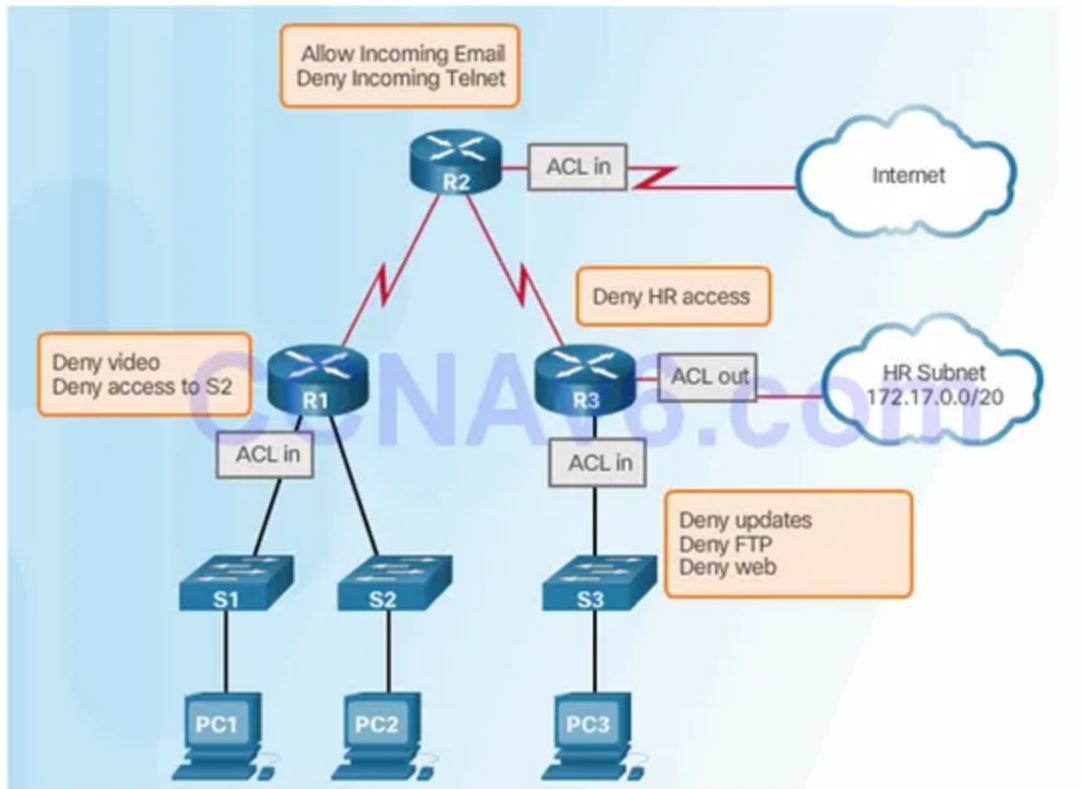
- Configure standard and extended IPv4 ACLs using CLI.
- Use ACLs to mitigate common network attacks.
- Configure IPv6 ACLs using CLI.

### Topic 4.1.1: Configuring Standard and Extended IPv4 ACLs with CLI

---

#### Introduction to 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

## Configuring Numbered and Named ACLs

### Standard Numbered ACL Syntax

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

### Extended Numbered ACL Syntax

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

## Named ACL Syntax

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

## Standard ACE Syntax

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

## Extended ACE Syntax

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

## Applying an ACL

---

### Syntax – Apply an ACL to an interface

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

### Syntax – Apply an ACL to the VTY lines

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

### Example – Named Standard ACL

### Example – Named Extended ACL

```
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
```

```

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

```

Syntax – Apply an ACL to the VTY lines

```

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

```

Example – Named ACL on VTY lines with logging

```

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 Configuration Guidelines

---

- Create an ACL globally and then apply it.
- Ensure the last statement is an implicit `deny any` or `deny any any`.
- Remember that statement order is important because ACLs are processed top-down. As soon as a statement is matched the ACL is exited.
- Ensure that the most specific statements are at the top of the list.
- Remember that only one ACL is allowed per interface, per protocol, per direction.
- Remember that new statements for an existing ACL are added to the bottom of the ACL by default.
- Remember that router generated packets are not filtered by outbound ACLs.
- Place standard ACLs as close to the destination as possible.
- Place extended ACLs as close to the source as possible.

## Editing Existing ACLs

---

Existing access list has three entries

```
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 has been edited, which adds a new ACE and replaces ACE line 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
```

Updated access list has four entries

```
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
```

## Sequence Numbers and Standard ACLs

---

Existing access list has four entries

```
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
```

Access list has been edited, which adds a new ACE that permits a specific IP address.

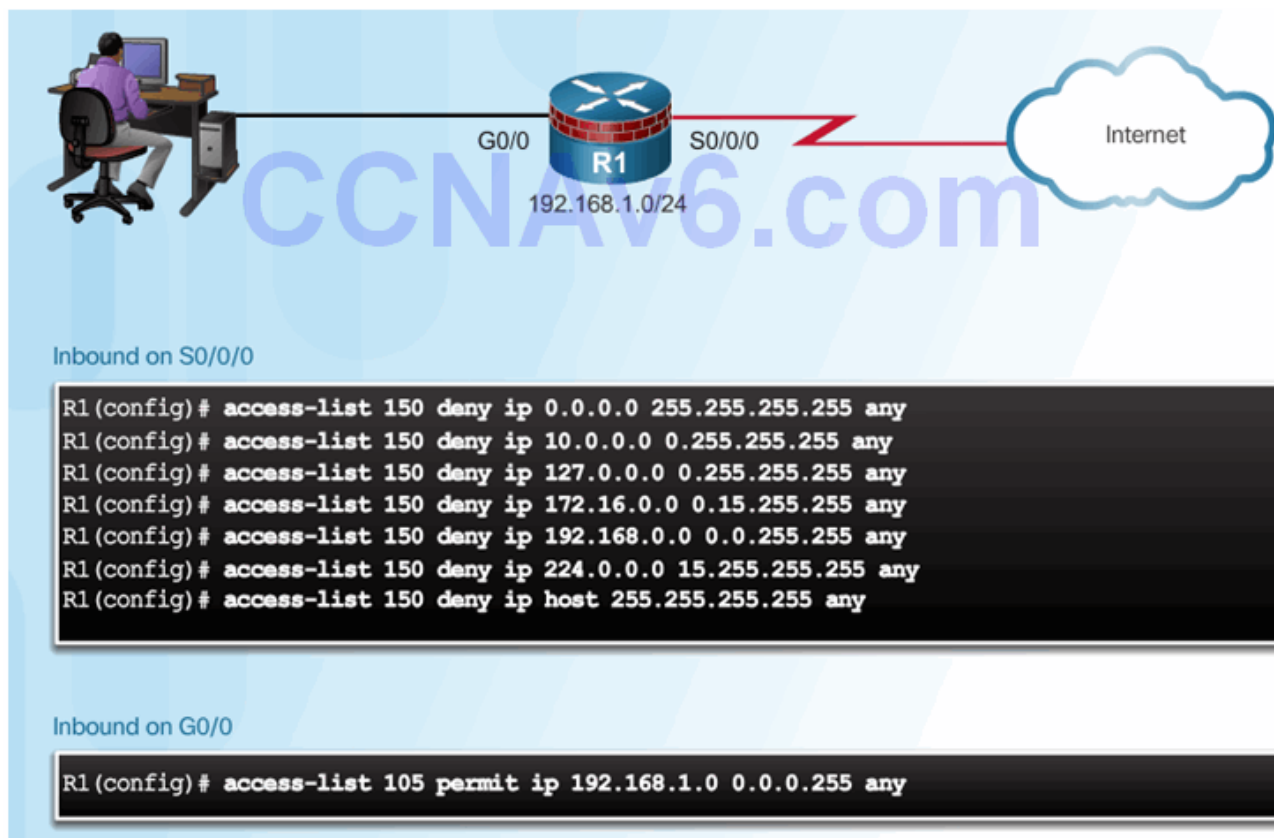
```
router(config)# ip access-list standard 19
router(config-std-nacl)# 25 permit 172.22.1.1
```

Updated access list places the new ACE before line 20

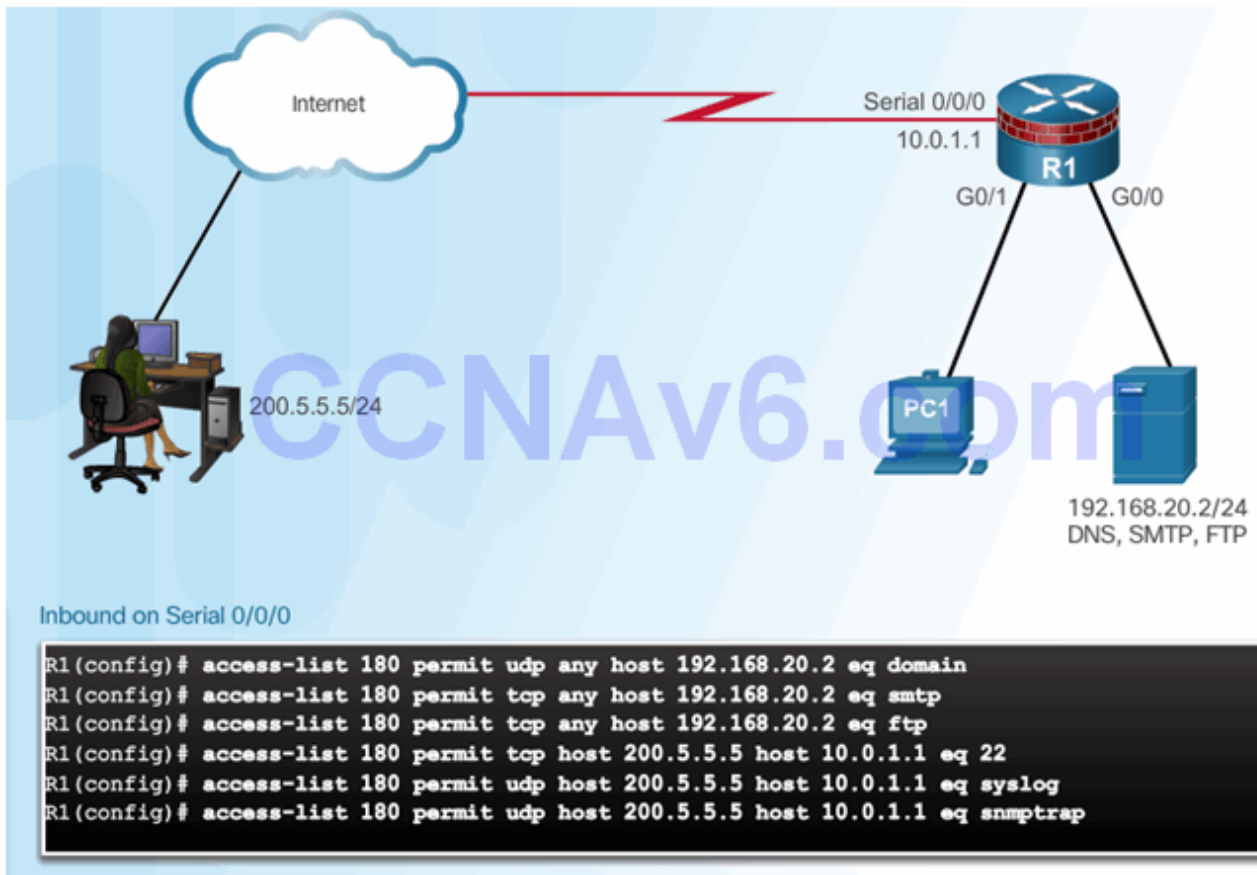
```
router# show access-lists
Standard IP access list 19
 10 permit 192.168.100.1
 25 permit 172.22.1.1
 20 permit 10.10.10.0, wildcard bits 0.0.0.255
 30 permit 201.101.110.0, wildcard bits 0.0.0.255
 40 deny any
```

## Topic 4.1.2: Mitigating Attacks with ACLs

### Antispoofing with ACLs



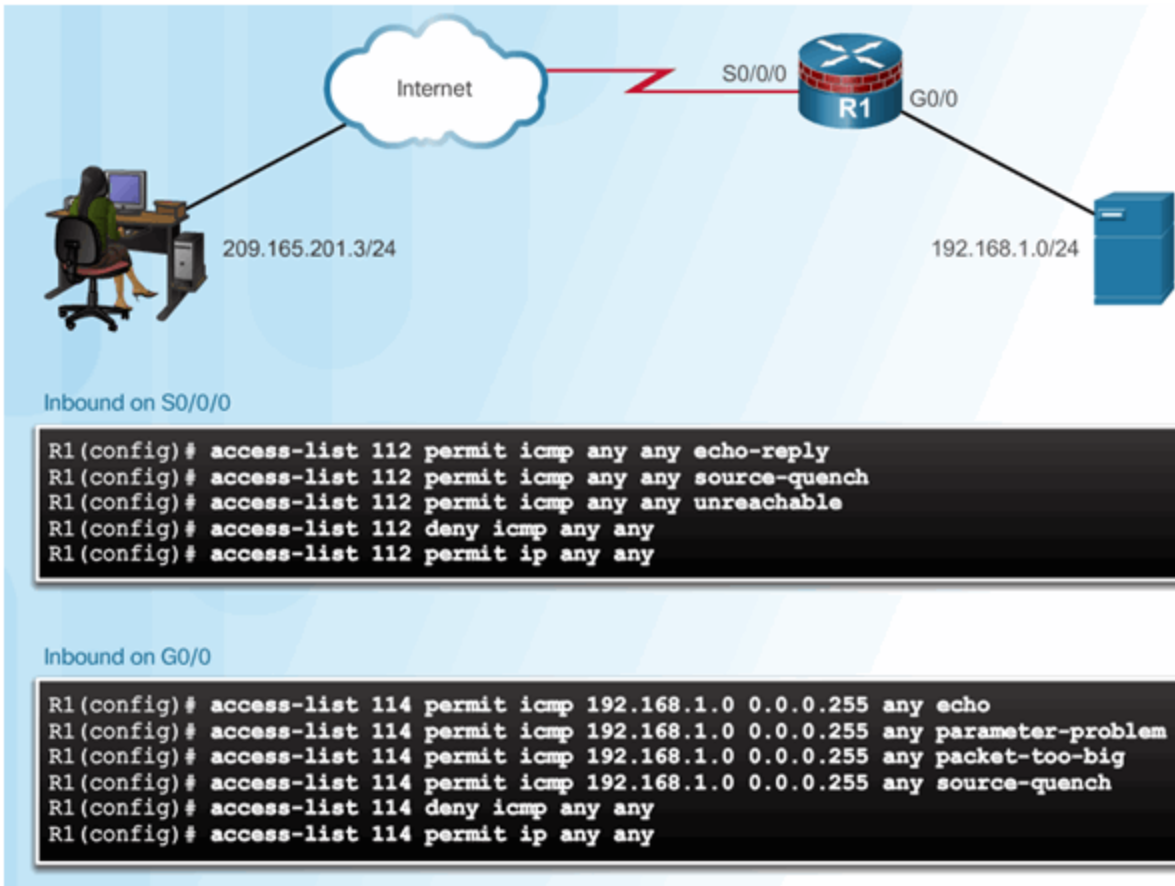
### Permitting Necessary Traffic through a Firewall



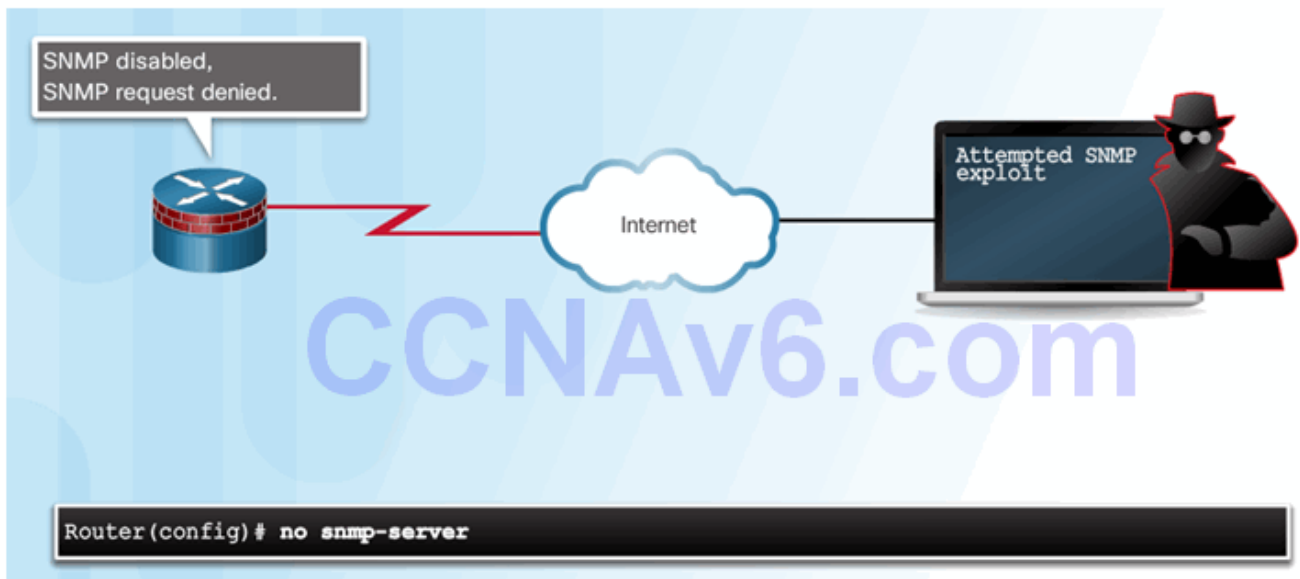
## Mitigating ICMP Abuse

---





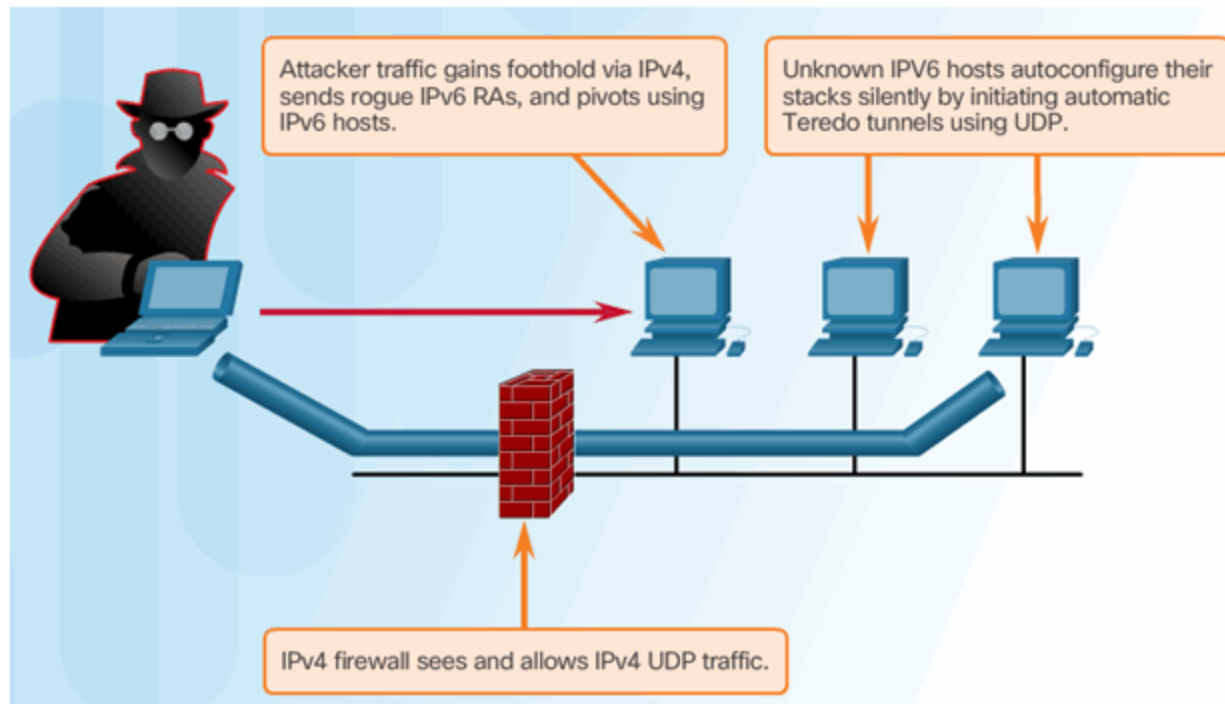
## Mitigating SNMP Exploits



## Topic 4.1.3: IPv6 ACLs

### Introducing IPv6 ACLs



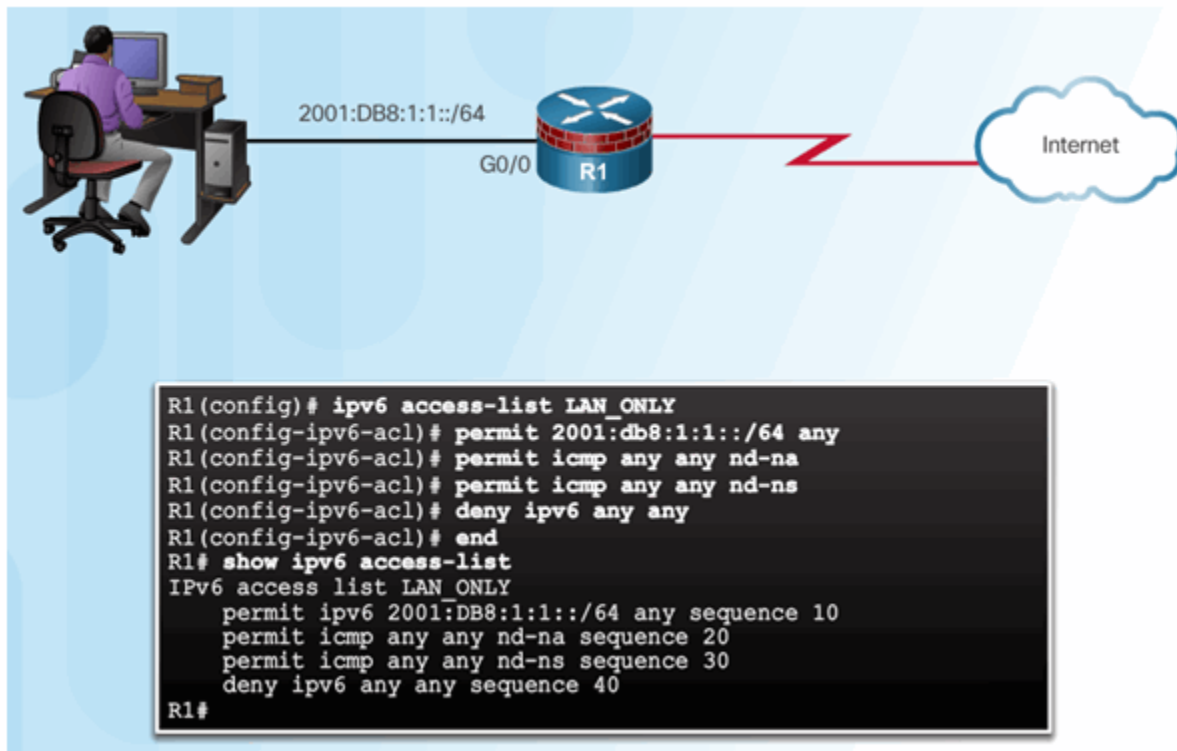


## IPv6 ACL Syntax

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

## Configure IPv6 ACLs



## Section 4.2: Firewall Technologies

---

Upon completion of this section, you should be able to:

- Explain how firewalls are used to help secure networks.
- Describe the various types of firewalls.
- Configure a classic firewall.
- Explain design considerations for implementing firewall technologies.

### Topic 4.2.1: Securing Networks with Firewalls

---

#### Defining Firewalls

---


All firewalls:

- Are resistant to attack
- Are the only transit point between networks because all traffic flows through the firewall
- Enforce the access control policy

#### Benefits and Limitations of Firewalls

---

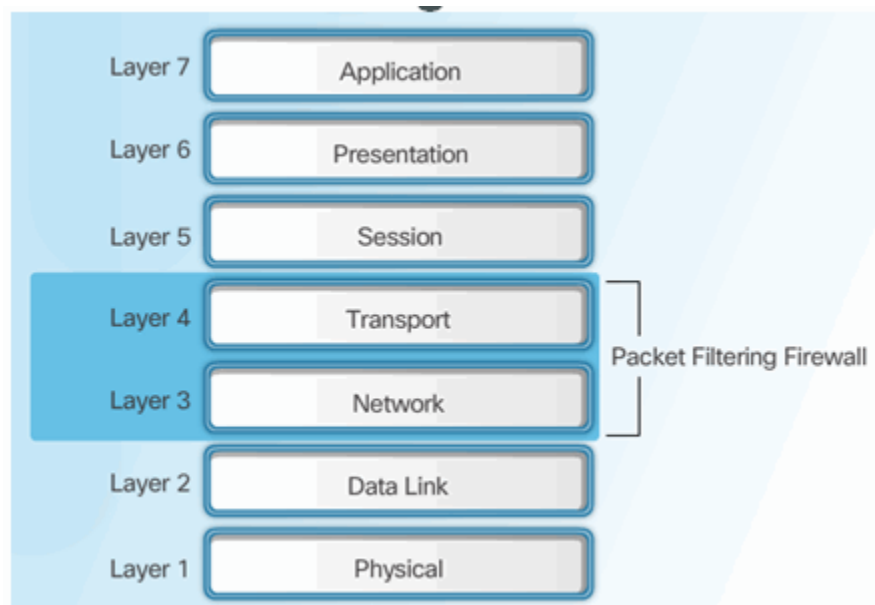
- **Allow** traffic from any external address to the web server.
- **Allow** traffic to FTP server.
- **Allow** traffic to SMTP server.
- **Allow** traffic to internal IMAP server.
- **Deny** all inbound traffic with network addresses matching internal-registered IP addresses.
- **Deny** all inbound 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.



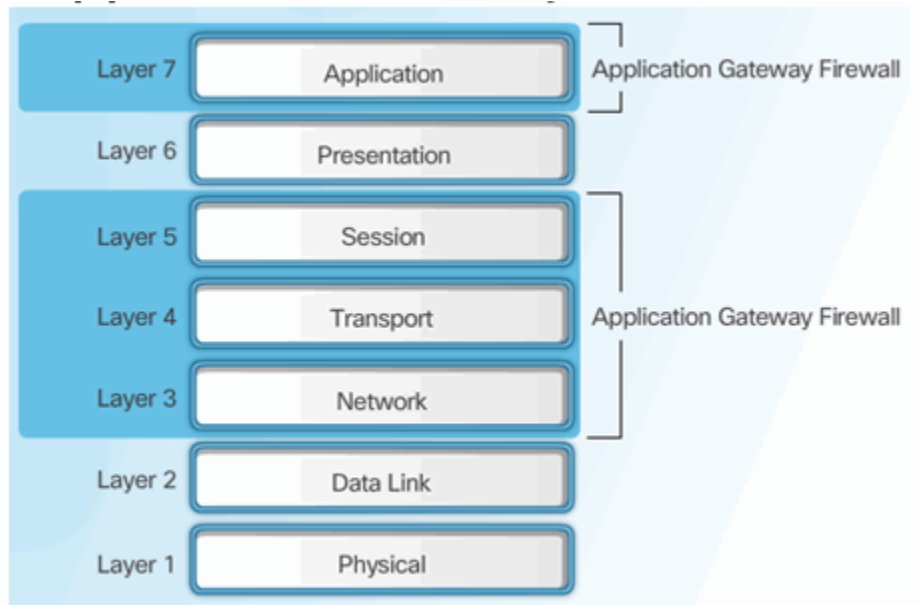
## Topic 4.2.2: Types of Firewalls

### Firewall Type Descriptions

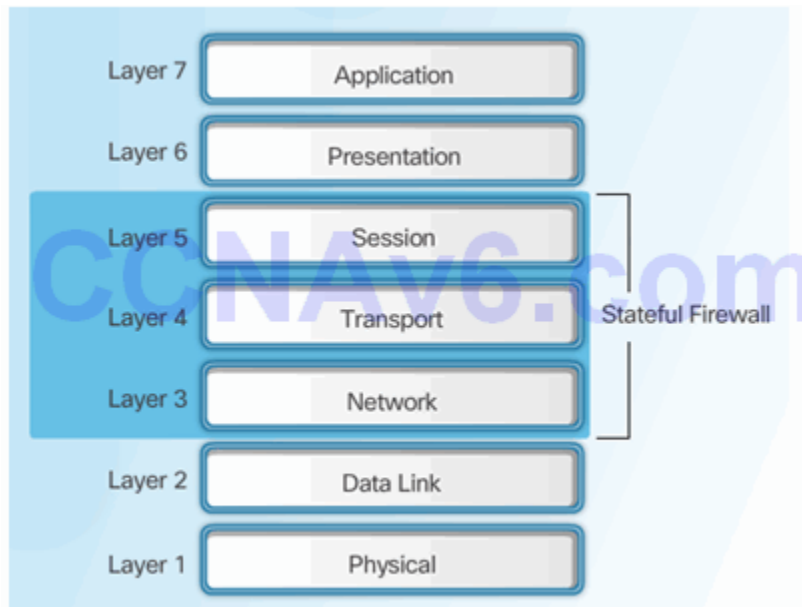
#### Packet Filtering Firewall



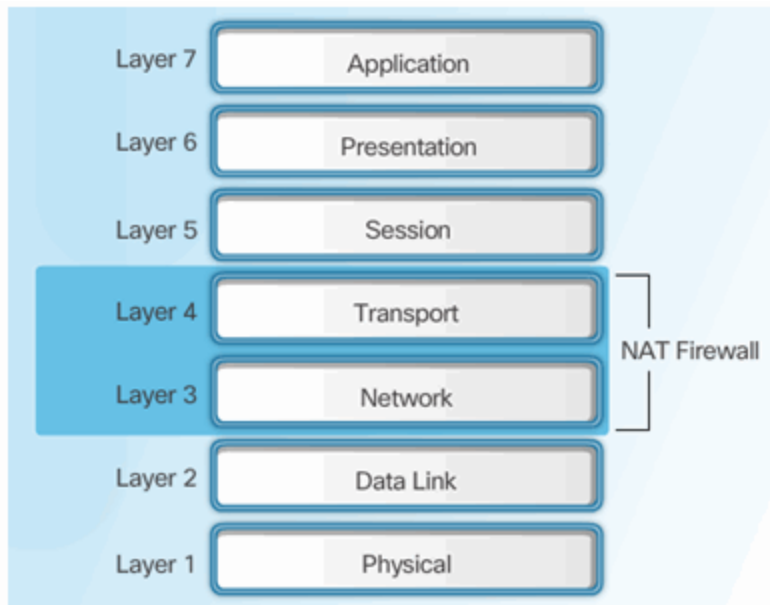
#### Packet Filtering Firewall



## Stateful Firewall

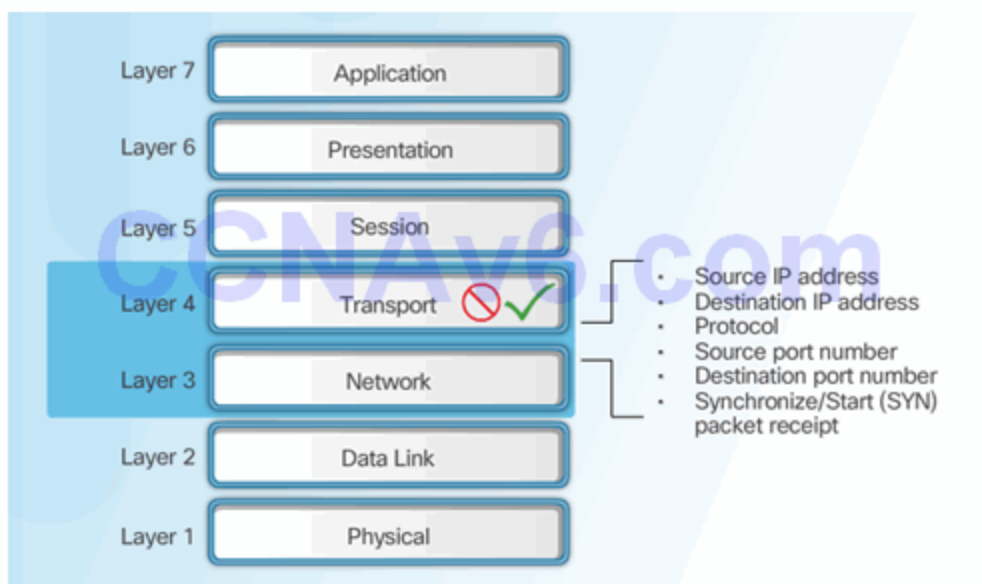


## NAT Firewall



## Packet Filtering Firewall Benefits & Limitations

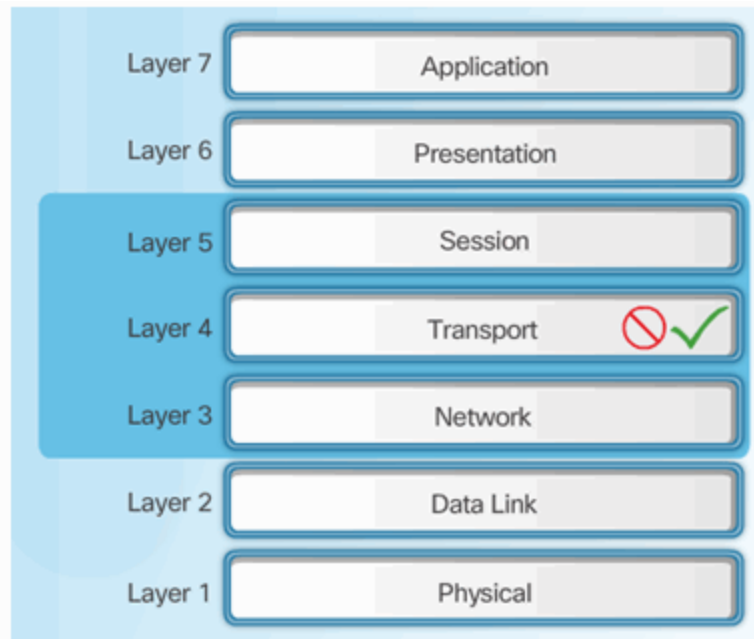
---



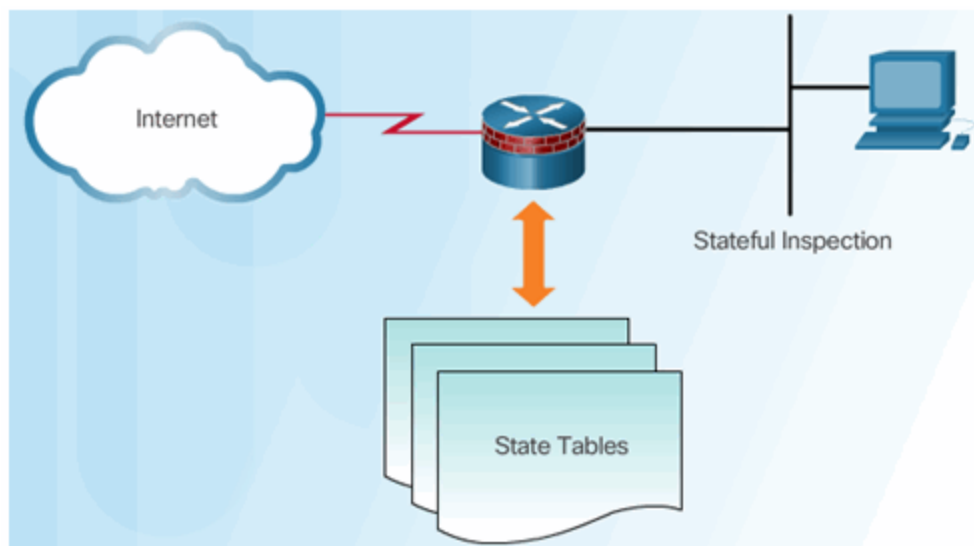
## Stateful Firewalls

---

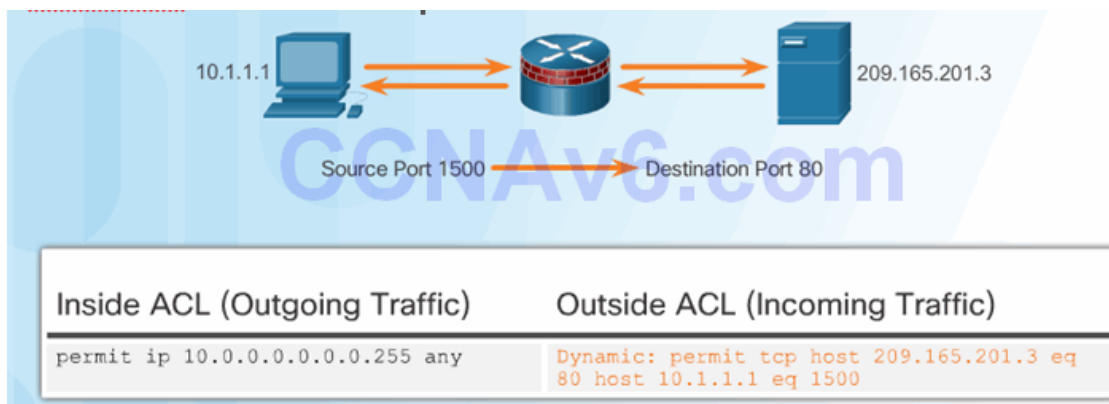
### Stateful Firewalls



## State Tables



## Stateful Firewall Operation



## Stateful Firewall Benefits and Limitations

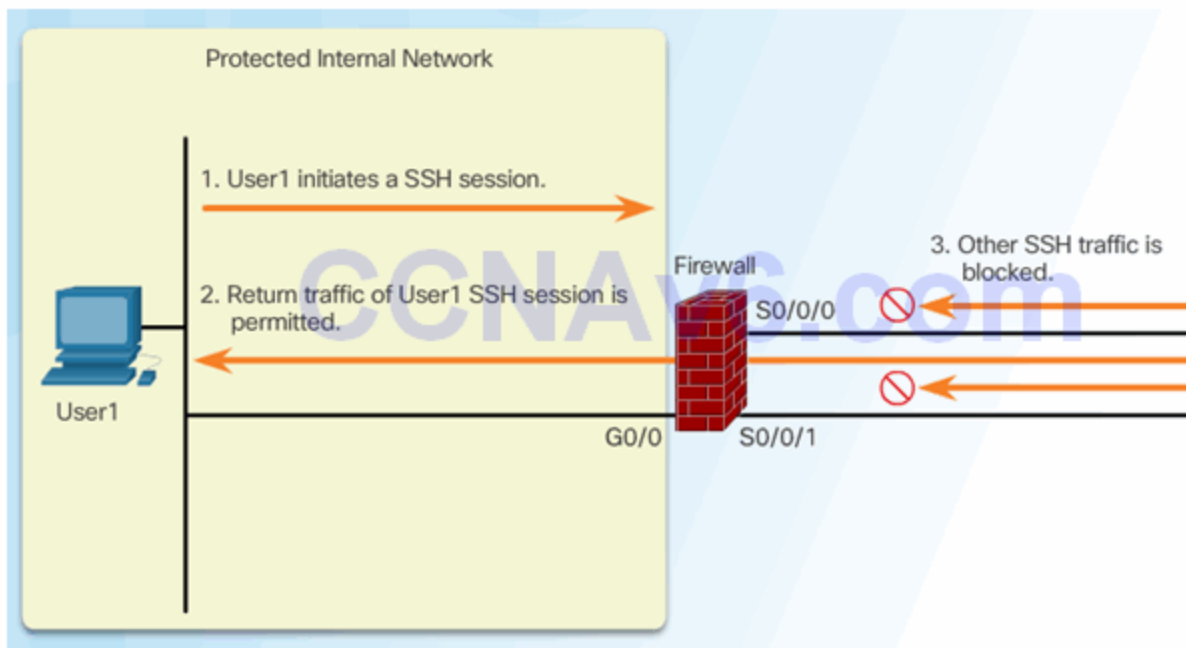
Benefits	Limitations
Primary means of defense	No Application Layer inspection
Strong packet filtering	Cannot filter stateless protocols
Improved performance over packet filters	Difficult to defend against dynamic port negotiation
Defends against spoofing and DoS attacks	No authentication support
Richer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

## Next Generation Firewalls

- Granular identification, visibility, and control of behaviors within applications
- Restricting web and web application use based on the reputation of the site
- Proactive protection against Internet threats
- Enforcement of policies based on the user, device, role, application type, and threat profile
- Performance of NAT, VPN, and SPI
- Use of an IPS

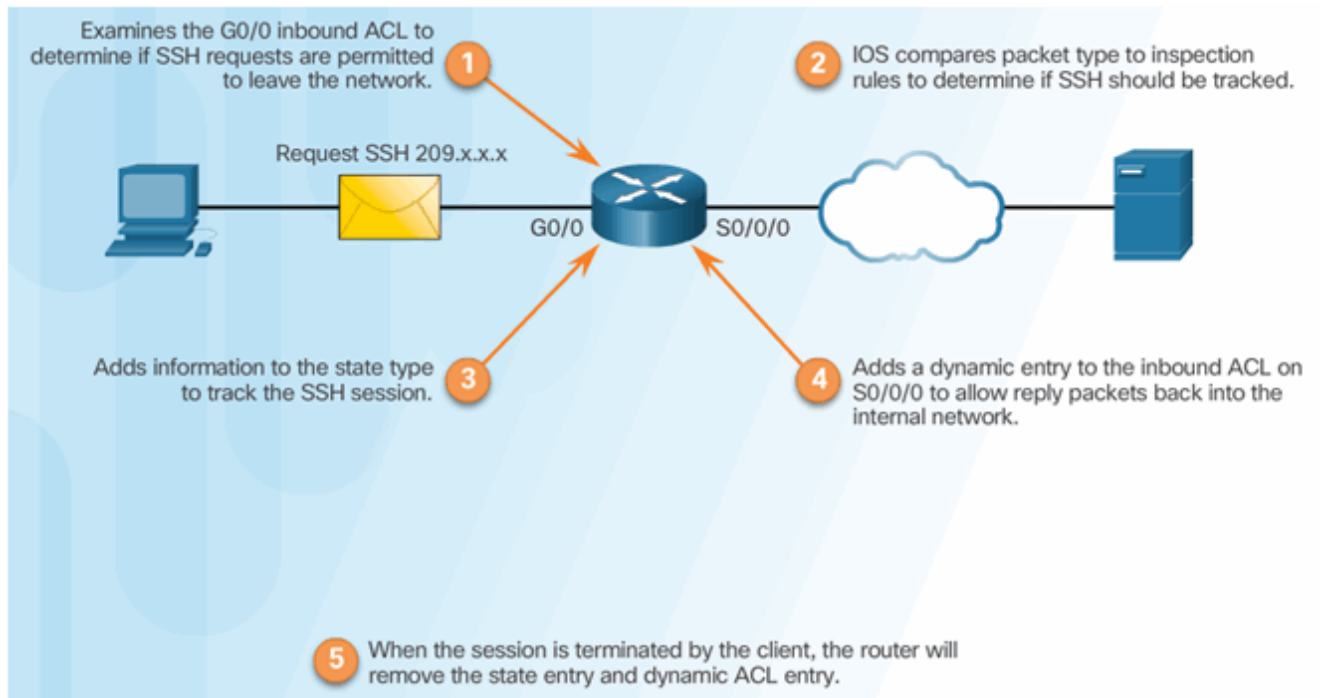
## Topic 4.2.3: Classic Firewall

### Introducing Classic Firewall



### Classic Firewall Operation



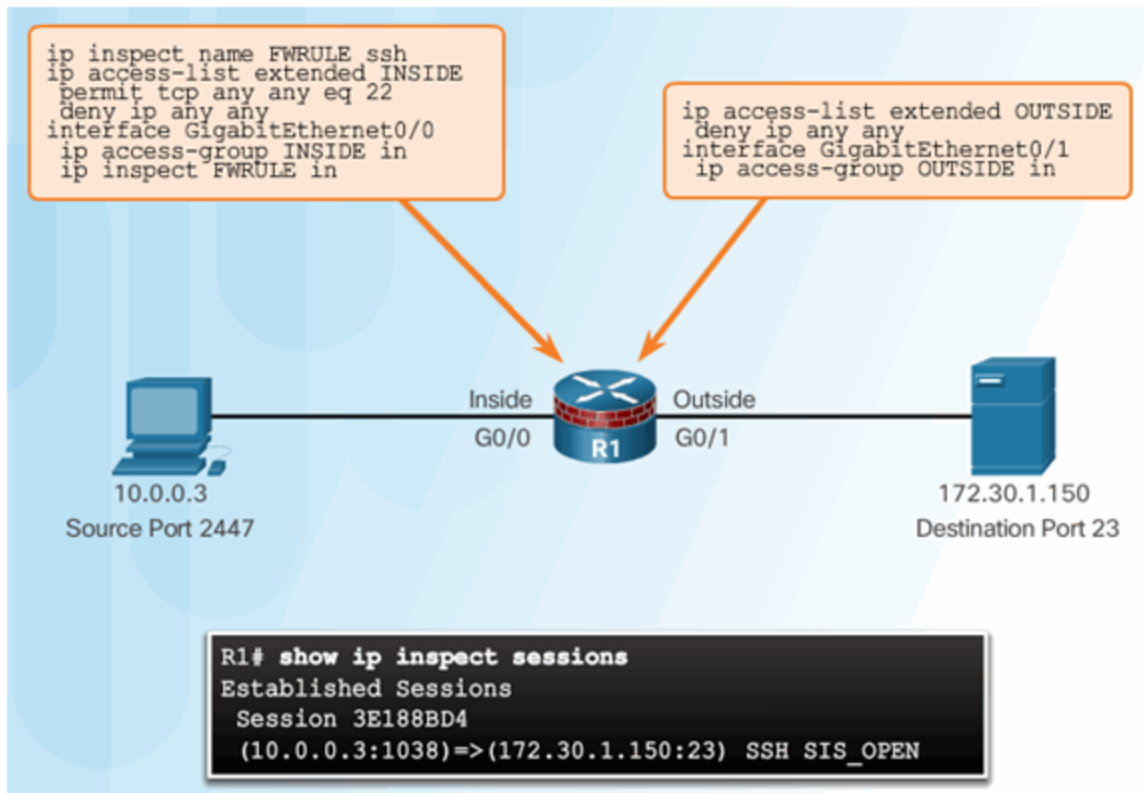


## Classic Firewall Configuration

---

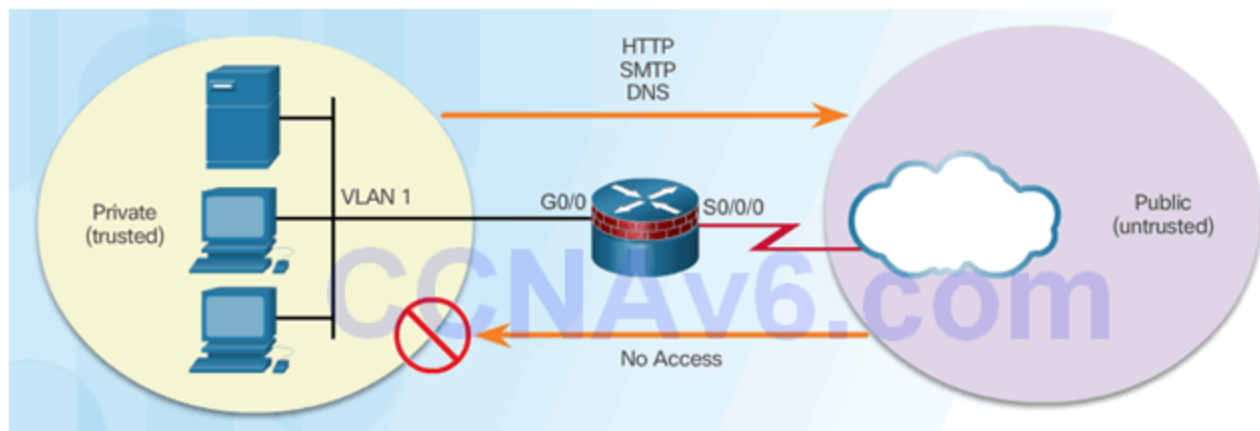
1. Choose the internal and external interfaces.
2. Configure ACLs for each interface.
3. Define inspection rules.
4. Apply an inspection rule to an interface.

### Inspection Rules

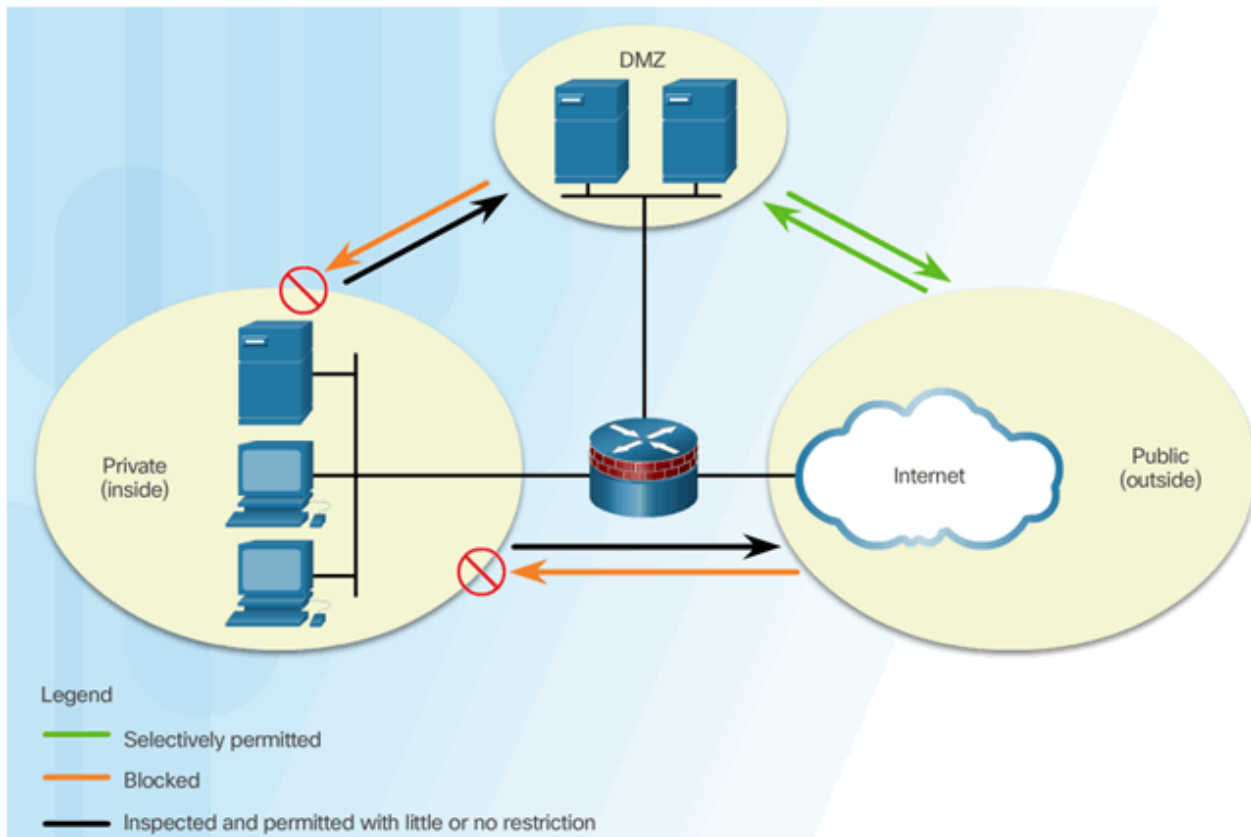


## Topic 4.2.4: Firewalls in Network Design

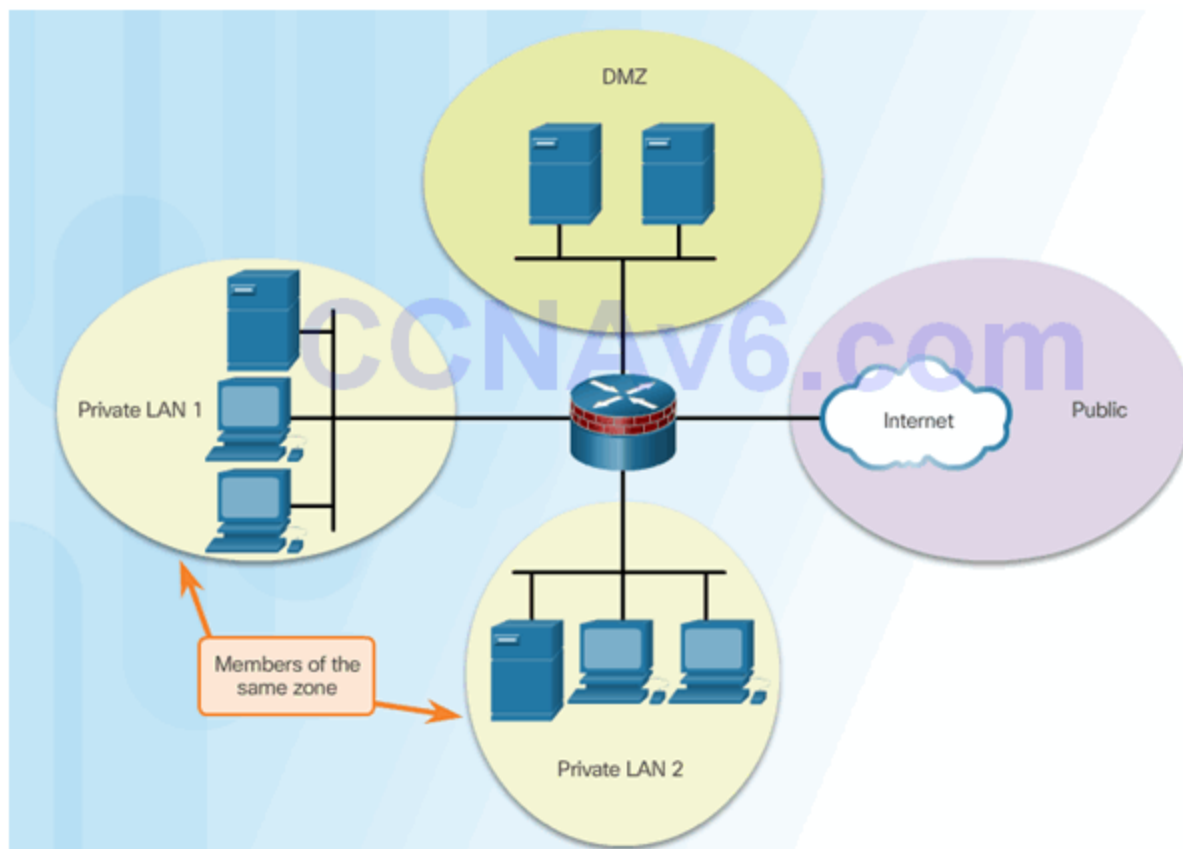
### Inside and Outside Networks



### Demilitarized Zones



## Zone-Based Policy Firewalls



## Layered Defense

---

Considerations for network defense:

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

Firewall best practices include:

- Position firewalls at security boundaries.
- It is unwise to rely exclusively on a firewall for security.
- Deny all traffic by default. Permit only services that are needed.
- Ensure that physical access to the firewall is controlled.
- Monitor firewall logs.
- Practice change management for firewall configuration changes.
- Remember that firewalls primarily protect from technical attacks originating from the outside.

## Section 4.3: Zone-Based Policy Firewalls

---

Upon completion of this section, you should be able to:

- Explain how Zone-Based Policy Firewalls are used to help secure a network.
- Explain the operation of a Zone-Based Policy Firewall.
- Configure a Zone-Based Policy Firewall with CLI.

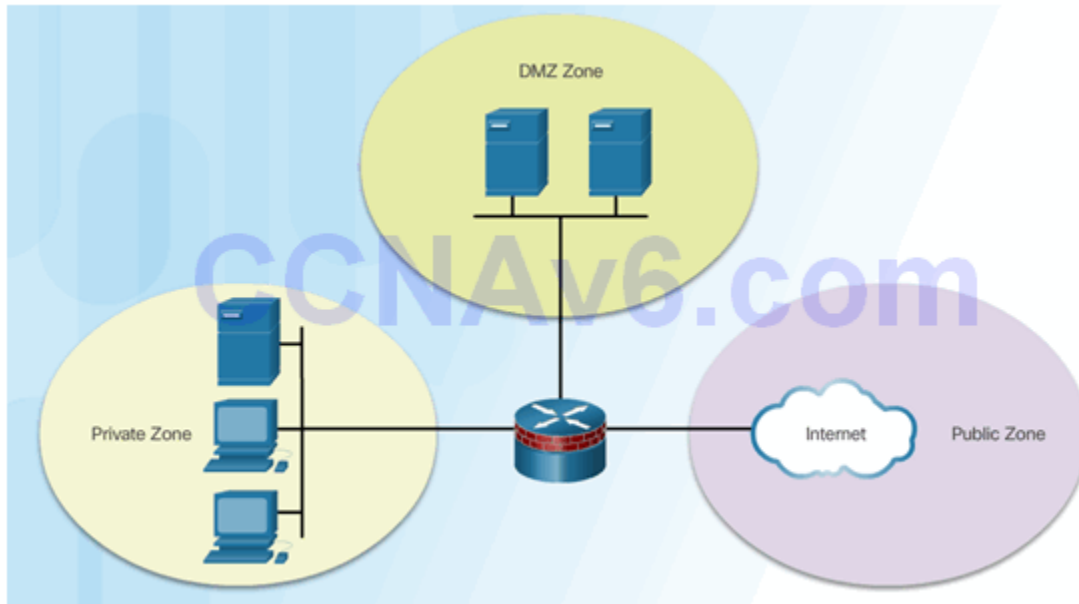
### Topic 4.3.1: Zone-Based Policy Firewall Overview

---

#### Benefits of ZPF

---

- Not dependent on ACLs
- Router security posture is to block unless explicitly allowed
- Policies are easy to read and troubleshoot with C3PL
- One policy affects any given traffic, instead of needing multiple ACLs and inspection actions



## ZPF Design

---

Common designs include:

- LAN-to-Internet
- Firewalls between public servers
- Redundant firewalls
- Complex firewalls

Design steps:

1. Determine the zones
2. Establish policies between zones
3. Design the physical infrastructure
4. Identify subsets within zones and merge traffic requirements

## Topic 4.3.2: ZPF Operation

---

### 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

---

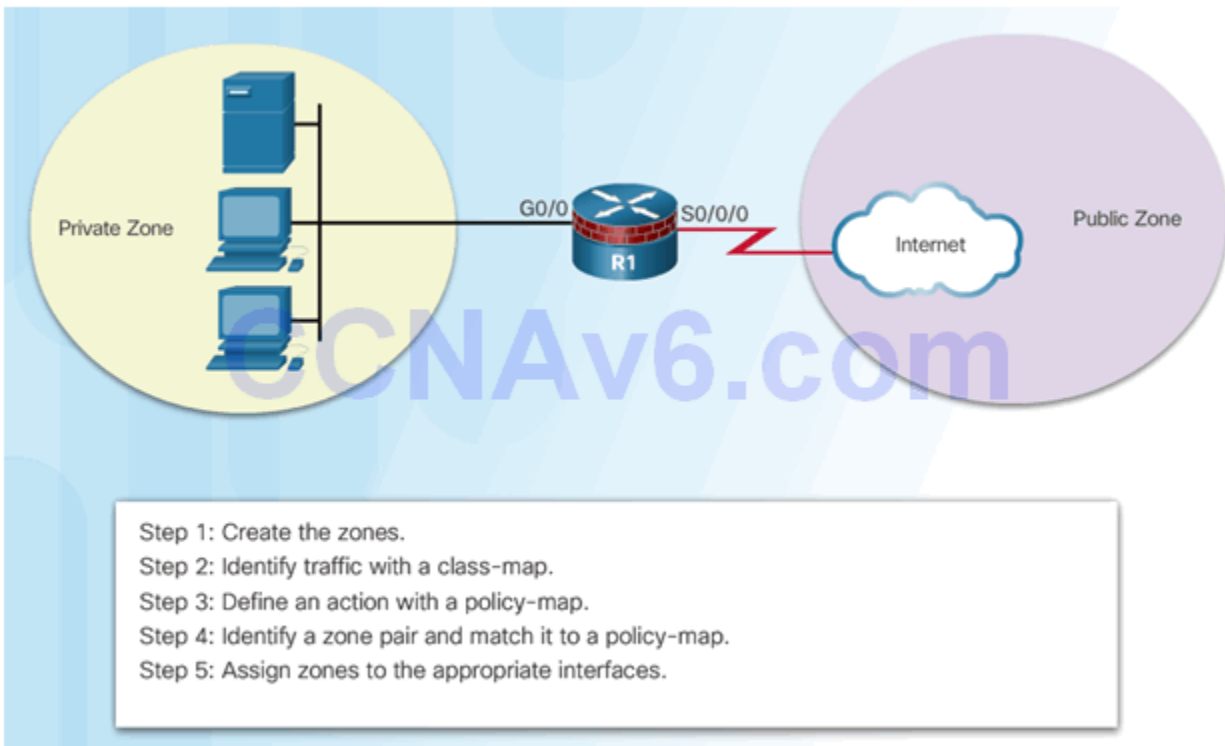
Source Interface Member of Zone?	Destination Interface Member of Zone?	Zone-Pair Exists?	Policy Exists?	Result
NO	NO	N/A	N/A	PASS
YES	NO	N/A	N/A	DROP
NO	YES	N/A	N/A	DROP
YES (private)	YES (private)	N/A	N/A	PASS
YES (private)	YES (public)	NO	N/A	DROP
YES (private)	YES (public)	YES	NO	PASS
YES (private)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

### Rules for Traffic to the Self Zone

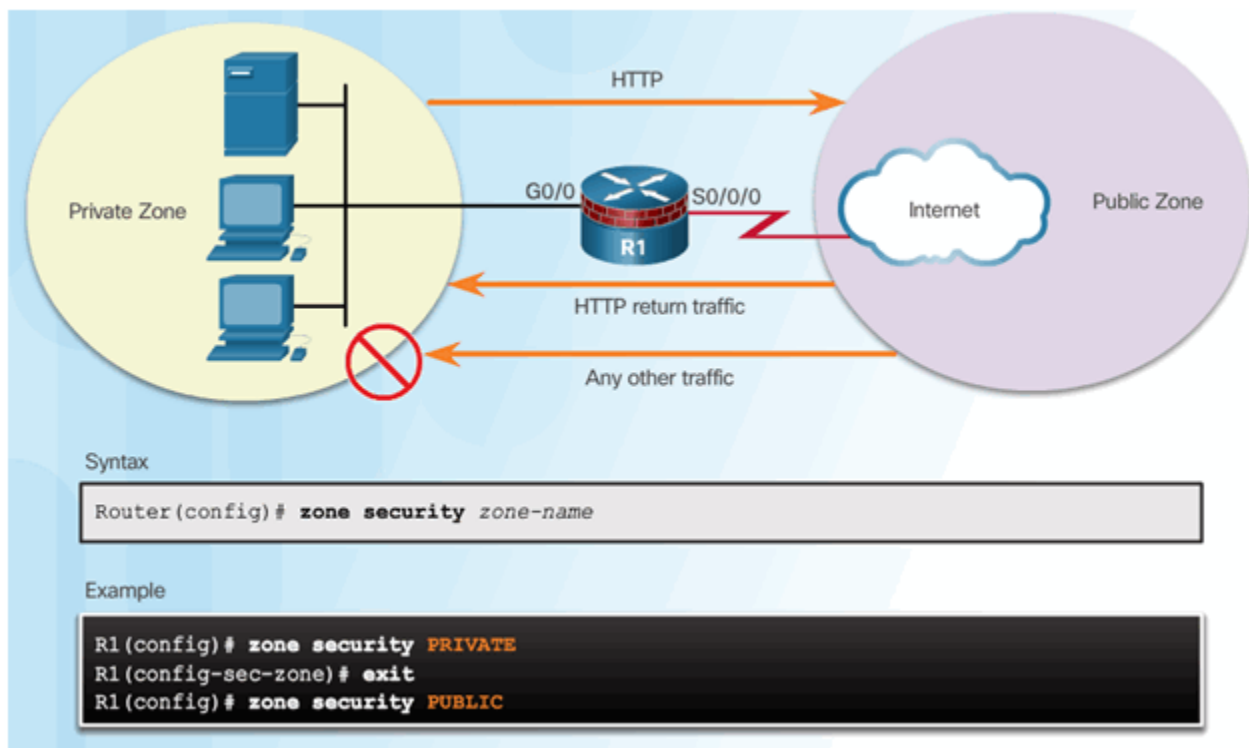
Source Interface Member of Zone?	Destination Interface Member of Zone?	Zone-Pair Exists?	Policy Exists?	Result
YES (self-zone)	YES	NO	N/A	PASS
YES (self-zone)	YES	YES	NO	PASS
YES (self-zone)	YES	YES	YES	INSPECT
YES	YES (self-zone)	NO	N/A	PASS
YES	YES (self-zone)	YES	NO	PASS
YES	YES (self-zone)	YES	YES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

### Topic 4.3.3: Configuring a ZPF

#### Configure ZPF



#### Step 1: Create Zones



## Step 2: Identify Traffic

### Command Syntax for **class-map**

Router(config)# <b>class-map</b> 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.

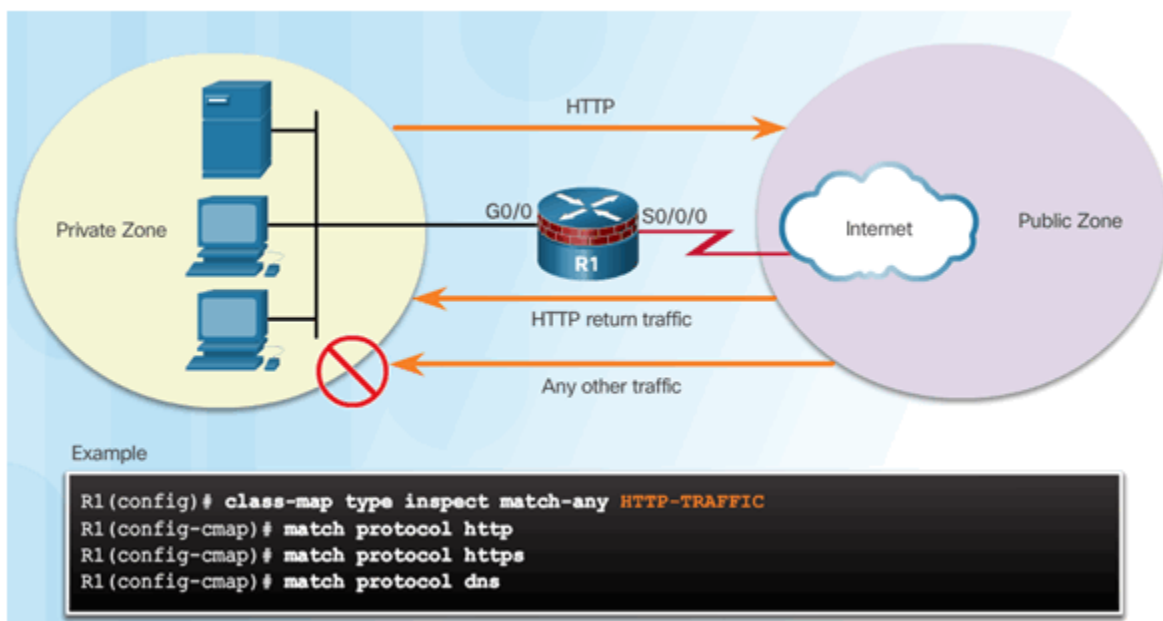
### Sub-Configuration Command Syntax for **class-map**



```
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
<code>match access-group</code>	Configures the match criteria for a class-map based on the specified ACL number or name.
<code>match protocol</code>	Configures the match criteria for a class-map based on the specified protocol.
<code>match class-map</code>	Uses another class-map to identify traffic.

## Example **class-map** Configuration



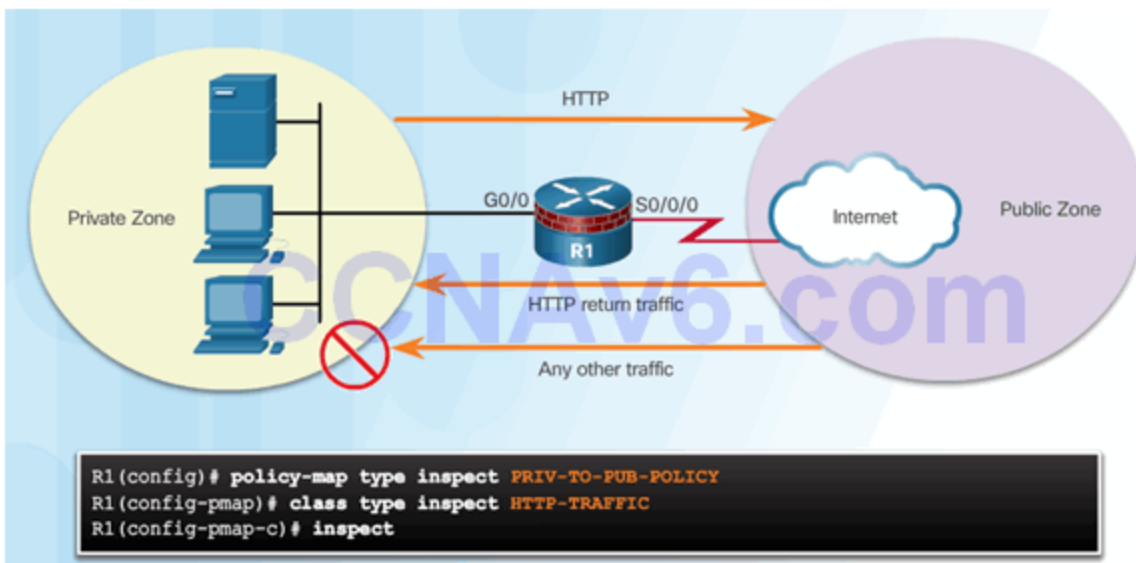
## Step 3: Define an Action

### 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



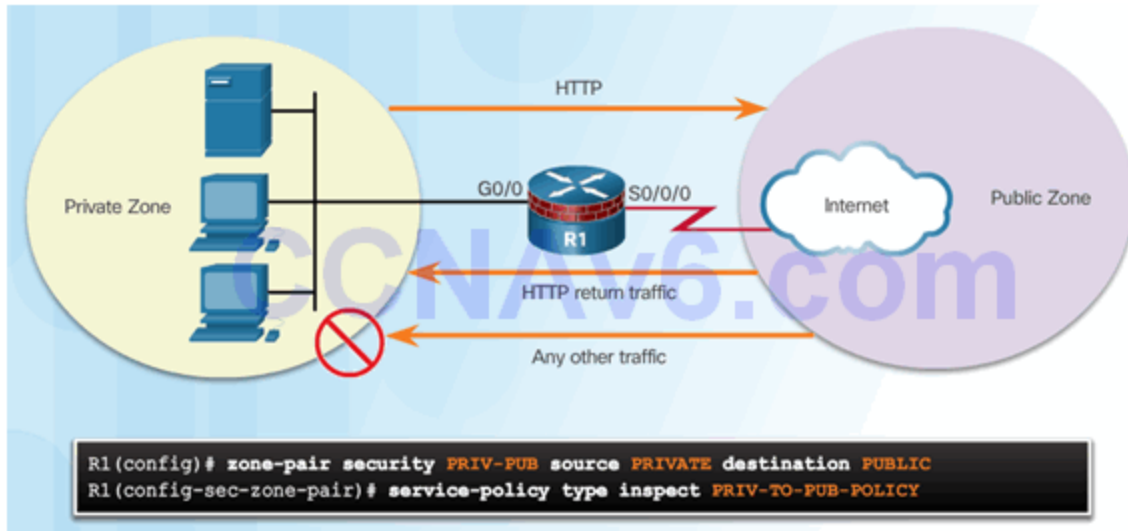
### Step 4: Identify a Zone-Pair and Match to a Policy

#### Command Syntax for **zone-pair** and **service-policy**

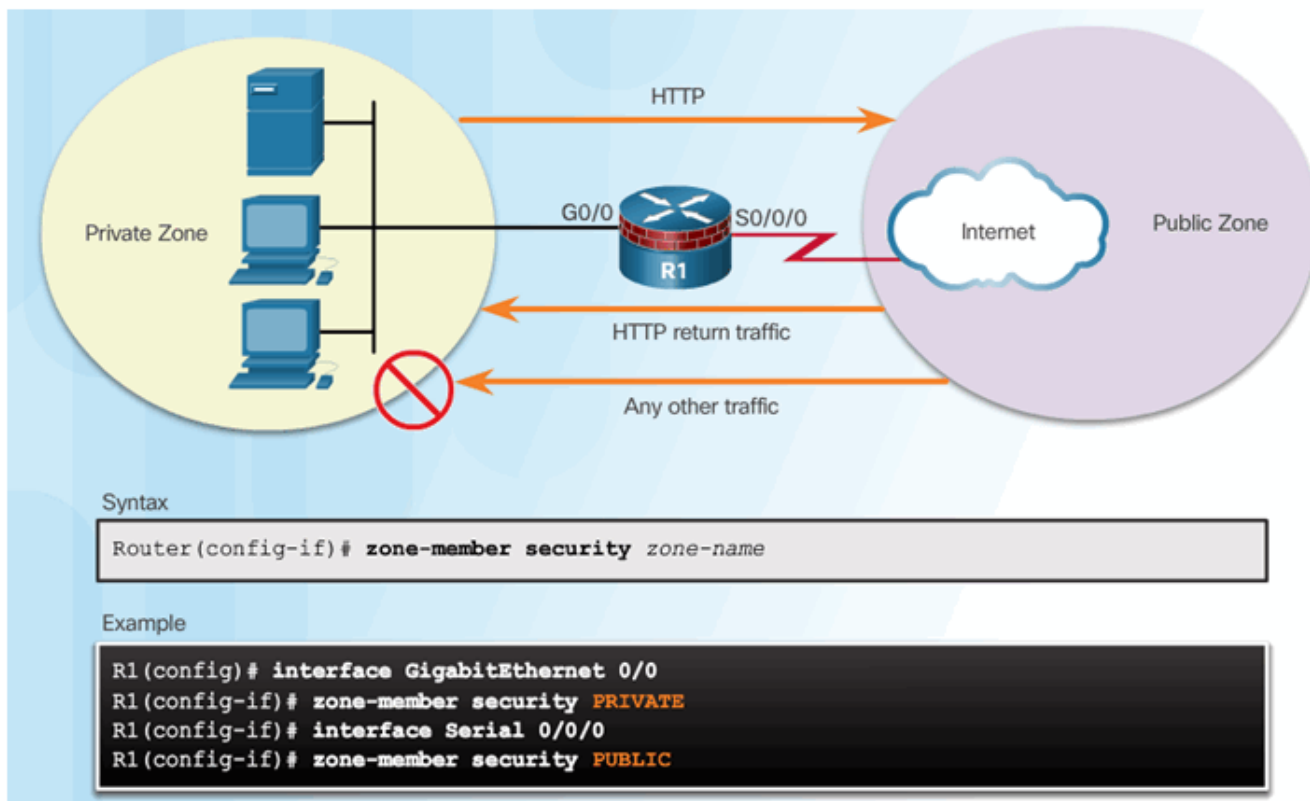
```
Router(config)# zone-pair security zone-pair-name source {source-zone-name | self } destination {destination-zone-name | self }
Router(config-sec-zone-pair)# service-policy type inspect policy-map-name
```

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



## 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.

## Download Slide PowerPoint (pptx):

---

[sociallocker id="54558"]



**CCNASv2\_InstructorPPT\_CH4.pptx**

**3.72 MB**

**2210 downloads**

---

...

[Download](#)

[/sociallocker]