

# ACLs and Firewall Technologies

November 4, 2014

# What this lecture is about:

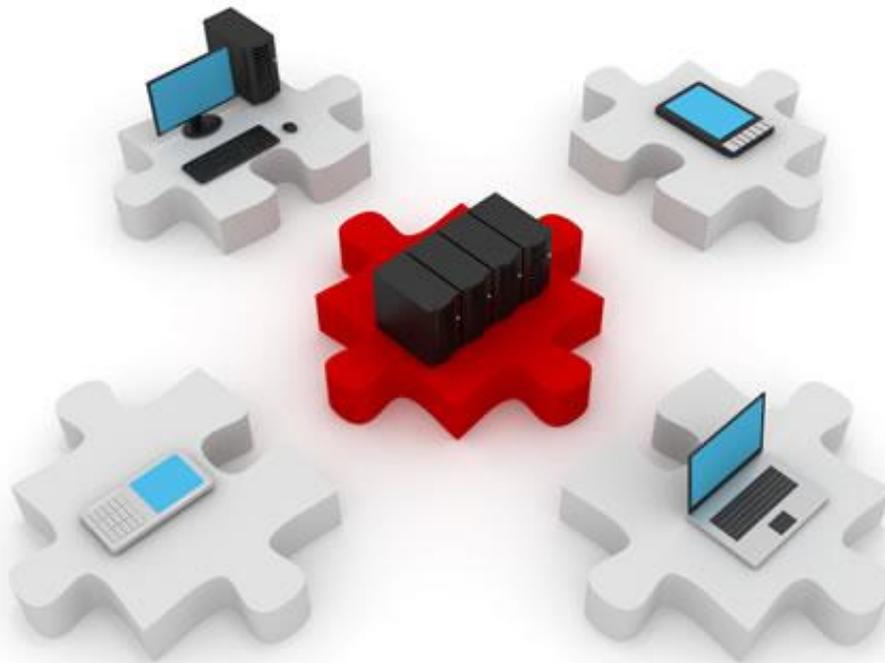
---

- ▶ Time-based ACLs
- ▶ Troubleshooting ACLs
- ▶ Attack mitigation using ACLs
- ▶ Cisco IOS Firewall - CBAC

# ACLs, season one – The beginnings

---

- ▶ Quick recap!
- ▶ What you've learned so far:
  - ▶ By the process of decision, ACLs are:
    - ▶ Standard (filter only the source IP address)
    - ▶ Extended (filter source, destination, protocols...)
  - ▶ By the way they are created:
    - ▶ Numbered
    - ▶ Named: use a meaningful name
  - ▶ You know you can filter certain TCP packets (“established”, “ack”, “fin”...)



## Time-based ACLs

Not even services work 24/7...

# Time-based ACLs

---

- ▶ Have the same functionality as extended ACLs
- ▶ Can also control access based on time
  - ▶ “no access to company servers outside the working hours”
  - ▶ “employee access to messenger service only during lunch”
  - ▶ “eu.logon.worldofwarcraft.com” only on weekends 😊
- ▶ Can provide a more secure access to resources
  - ▶ Allowing a certain type of traffic indefinitely might not be desirable
  - ▶ Network overview: time-based ACLs can log traffic only at certain times of the day

# Configuring time-based ACLs

---

- ▶ First, a “time range” object must be created globally

- ▶ Example – absolute time range:

```
R1(config)#time-range MYTIME
```

```
R1(config-time-range)#absolute start 08:00 27 November 2009  
end 20:00 28 November 2009
```

- ▶ Example – periodic time range:

```
R1(config)#time-range PERIODIC
```

```
R1(config-time-range)#periodic daily 09:30 to 12:00
```

# Absolute and periodic events

---

- ▶ Absolute events take place only once
  - ▶ They last for the entire duration on the timespan
- ▶ Extended ranges for absolute entries:
  - ▶ Omitting the start time will default to the current time
  - ▶ Omitting the end time will default to 23:59 on 31<sup>st</sup> of Dec, 2035
- ▶ Periodic events can repeat every (keyword list):
  - ▶ Monday, Tuesday, Wednesday, ...
  - ▶ Daily
  - ▶ Weekdays
  - ▶ Weekend

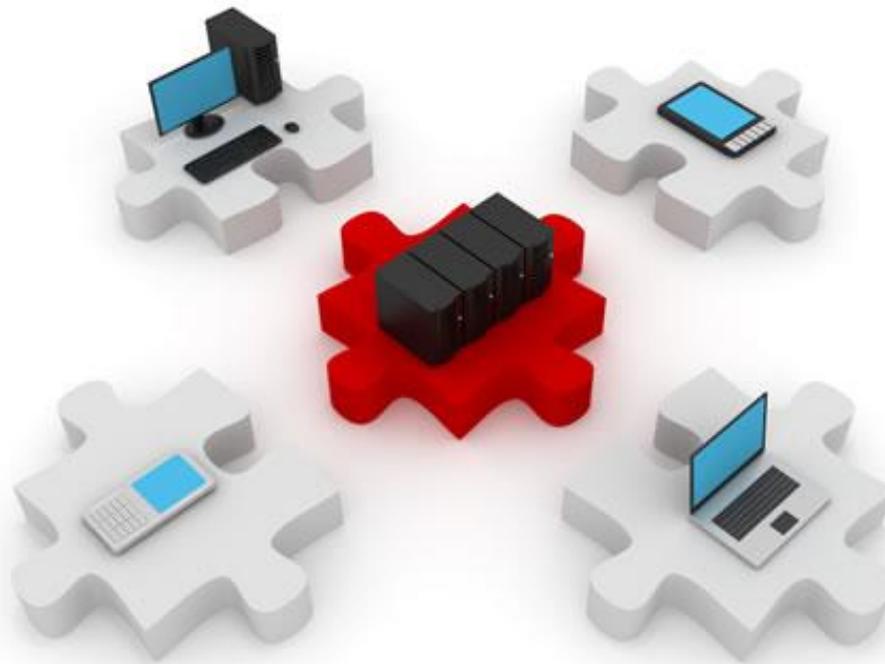
# Using time ranges with ACLs

---

- ▶ Using the “time-range” keyword”

```
R1(config)#access-list 101 permit tcp any host 192.168.1.1  
eq www time-range MYTIME
```

- ▶ The “time-range” entries are only checked during their respective time intervals
- ▶ Don’t forget to set the clock right!
  - ▶ Remember NTP!



# Troubleshooting ACLs

ACL misconfiguration

# Debugging access lists

---

- ▶ Use the “show access-lists” command to view all configured ACLs on the router or a specific ACL:
  - ▶ Non-applied ACLs are also displayed

```
Router# show access-list MyACL
```

```
Extended IP access list MyACL
```

```
10 permit tcp host 21.35.80.22 eq telnet host 21.23.77.101
20 permit tcp host 21.35.80.25 eq 16100 host 21.23.77.101 (149407 matches)
30 permit tcp host 21.35.80.25 eq 17600 host 21.23.77.101 (80592 matches)
40 permit tcp host 21.35.80.27 eq 10701 host 21.23.77.101 (26008 matches)
```

- ▶ The number of matched packets is also displayed for each entry
  - ▶ Matches include permitted and denied packets

# Number of matches

---

- ▶ Analyze the number of matches per entry in order to:
  - ▶ Determine whether the traffic flows as expected
    - ▶ Test load balancing
  - ▶ Determine whether the ACL entries are in the correct order
    - ▶ Ex: denying TCP after permitting IP has no effect
    - ▶ Ex: permitting a certain service to a subnet after denying all traffic to the subnet has no effect
  - ▶ Optimize the ACL
    - ▶ Higher numbers should be at the beginning of the ACL
    - ▶ Stepping through all the entries of an ACL for each packet uses CPU cycles
  - ▶ Security
    - ▶ Matches on explicitly denied services or types of packets can indicate attack attempts

# Real-time traffic

---

- ▶ Use “debug” commands to view allowed and dropped packets:

```
R1#debug ip packet
```

```
IP packet debugging is on
```

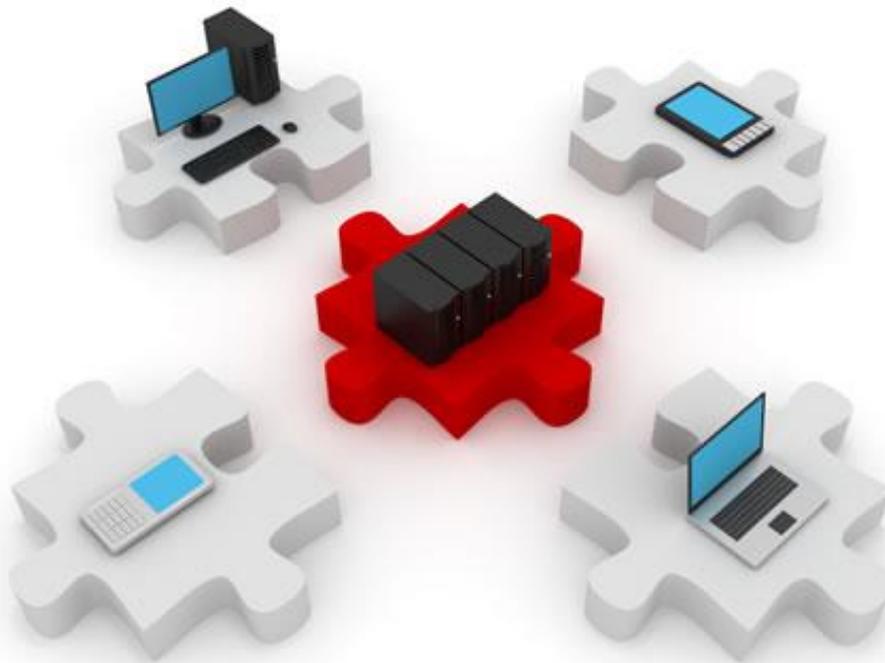
```
R1#
```

```
Oct 30 09:31:47.668: IP: s=10.2.2.2 (FastEthernet0/0), d=10.1.1.1, len 100, access denied
Oct 30 09:31:47.668: IP: s=10.1.1.1 (local), d=10.2.2.2 (FastEthernet0/0), len 56, sending
Oct 30 09:31:49.668: IP: s=10.2.2.2 (FastEthernet0/0), d=10.1.1.1, len 100, access denied
Oct 30 09:31:49.668: IP: tableid=0, s=10.1.1.1 (local), d=10.2.2.2 (FastEthernet0/0), routed via FIB
Oct 30 09:31:49.668: IP: s=10.1.1.1 (local), d=10.2.2.2 (FastEthernet0/0), len 56, sending
Oct 30 09:31:51.668: IP: s=10.2.2.2 (FastEthernet0/0), d=10.1.1.1, len 100, access denied
Oct 30 09:31:51.668: IP: s=10.1.1.1 (local), d=10.2.2.2 (FastEthernet0/0), len 56, sending
Oct 30 09:31:57.996: IP: s=10.2.2.2 (FastEthernet0/0), d=10.1.1.1 (FastEthernet0/0), len 44, rcvd 3
```

- ▶ Displayed packets can also be filtered using an access list:

```
R1#debug ip packet ?
```

```
<1-199>      Access list
<1300-2699>  Access list (expanded range)
detail       Print more debugging detail
```

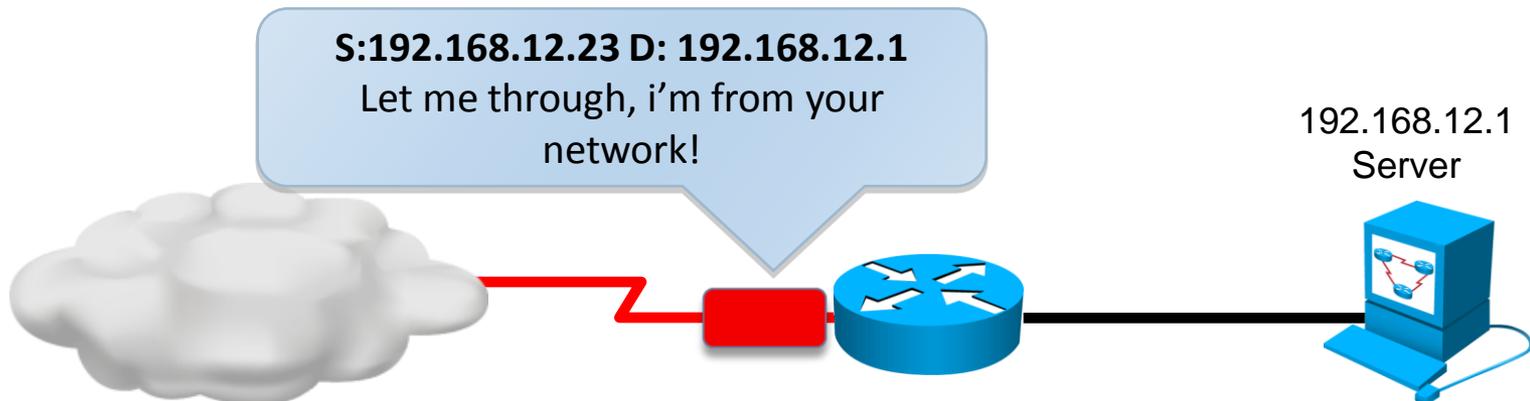


## Mitigating Attacks With ACLs

Just when you thought the ACLs were over...

# Prevent address spoofing

- ▶ You should not allow inbound packets having a source IP address from inside your private network
- ▶ Generally, the following addresses should not be allowed:
  - ▶ Any local host address (127.0.0.0/8)
  - ▶ Any private addresses (RFC 1918 – specific)
  - ▶ Any addresses from the IP multicast range (224.0.0.0/4)



# Example ACL

---

- ▶ Example general anti-spoofing configuration:

```
R2(config-std-nacl)#do sh access-list NO_SPOOF
Standard IP access list NO_SPOOF
 10 deny    127.0.0.0, wildcard bits 0.255.255.255
 20 deny    10.0.0.0, wildcard bits 0.255.255.255
 30 deny    172.16.0.0, wildcard bits 0.15.255.255
 40 deny    192.168.0.0, wildcard bits 0.0.255.255
 50 deny    224.0.0.0, wildcard bits 15.255.255.255
```

- ▶ Also, packets coming from inside the network with a source address other than one from your own subnets should not be allowed

# Mitigating DoS Smurf attacks

---

- ▶ Do you remember what a smurf attack was?
- ▶ Smurf attack: sending a spoofed ping request to the broadcast address of a subnet
  - ▶ All hosts reply with an echo-reply
  - ▶ The network can become saturated
- ▶ The “evil” way:
  - ▶ The attacker also spoofs the source IP address of the ping packets
  - ▶ A victim’s real IP address is used
  - ▶ All replies go back to one single victim
  - ▶ The network acts as an amplifier for the attack

# Mitigating DoS Smurf Attacks

---

- ▶ Disable directed broadcasts on a per-interface basis:

```
R2(config)#int fastEthernet 0/0
```

```
R2(config-if)#no ip directed-broadcast
```

- ▶ Starting with IOS version 12.0, this is the default setting

# Preventing TCP attacks

---

- ▶ A device maintains the state of every active TCP connection
- ▶ A TCP SYN flood overwhelms the device's operating system by opening a large number of TCP sessions without closing them
- ▶ You've learned about a way to prevent this. How?
  - ▶ Answer: Use an extended ACL to block SYN TCP packets from the outside

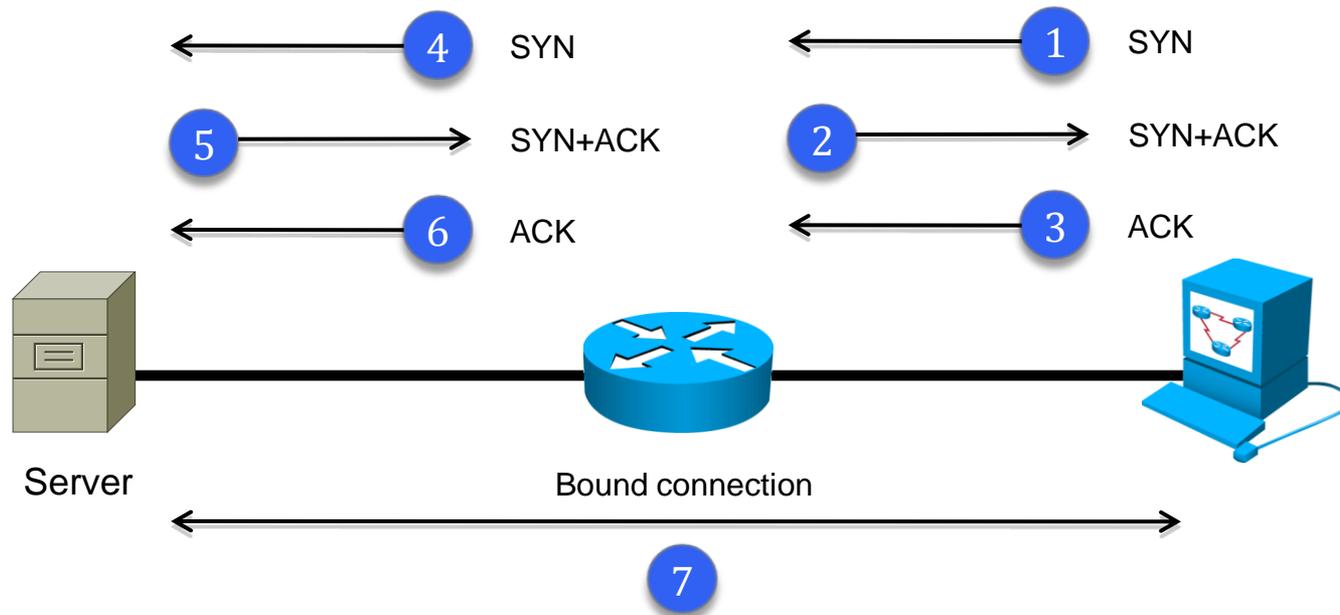
```
R2 (config-ext-nacl)#deny tcp any 192.168.2.0 0.0.0.255 syn
```
- ▶ Another way: using **TCP Intercept**
  - ▶ ...AND access lists 😊

# TCP Intercept

---

- ▶ Helps prevent SYN-flooding attacks by intercepting and validating TCP connection requests
- ▶ Intercepts TCP SYN connections from clients to servers on according to an extended ACL
  - ▶ Establishes a connection to the client, on behalf of the server
  - ▶ Establishes a connection to the server, on behalf of the client
  - ▶ Connection attempts from unreachable clients will not reach the servers – it would fail the first attempt
- ▶ If **illegitimate** requests are detected, actions are taken:
  - ▶ Half-open connections are closed after a threshold
  - ▶ A timeout timer is started for all sessions

# TCP Intercept behaviour



- 1 – 3: Session establishment between client and router
- 4 – 6: Session establishment between router and server

# Example TCP Intercept Configuration

---

- ▶ Define an access list to identify connections to be intercepted:

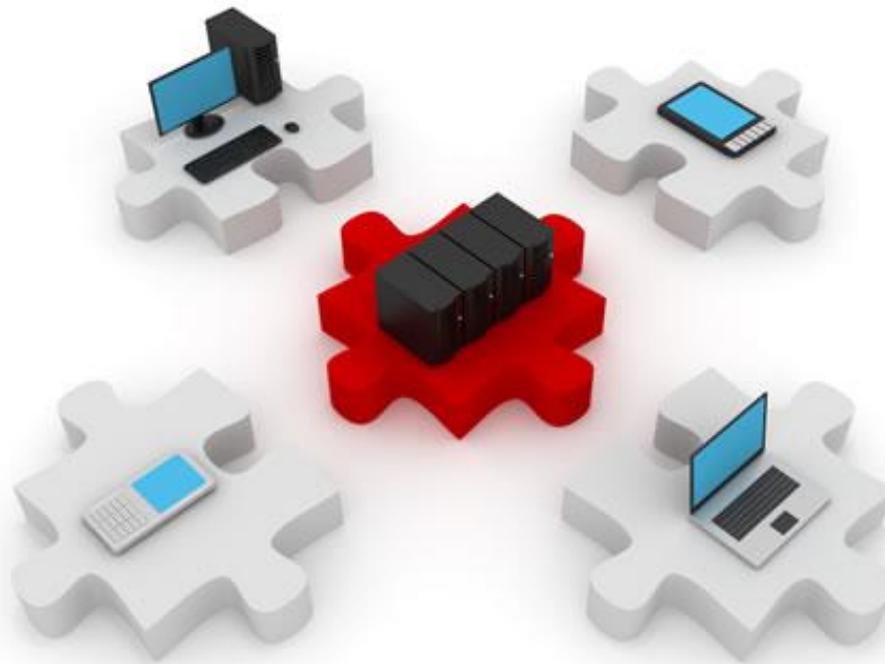
```
ip access-list extended TCP_INTER
  permit tcp any host 192.168.1.1 eq www
  permit tcp any host 192.168.1.1 eq smtp
  permit tcp any host 192.168.1.1 eq pop3
```

- ▶ Configure TCP Intercept parameters:

```
ip tcp intercept list TCP_INTER
ip tcp intercept connection-timeout 1800
ip tcp intercept max-incomplete low 1000
ip tcp intercept max-incomplete high 1500
ip tcp intercept drop-mode oldest
```



Drop oldest or  
random connections



# Firewall Technologies

Who let the dogs... in?

# Firewalls

---

- ▶ Original term..
- ▶ Firewall: A system that enforces an access control policy between networks
- ▶ A firewall must be:
  - ▶ Resistant to attacks (why?)
  - ▶ The only transit point (why?)
  - ▶ Responsible for enforcing access control policy
    - ▶ Network access policies are implemented on firewalls that manage all inbound connections

# Benefits of firewalls

---

- ▶ Can hide sensitive data
- ▶ Prevent malicious data from entering a network
- ▶ Can prevent exploits
- ▶ Central point for implementing security policies

# Drawbacks of firewalls

---

- ▶ A single firewall is also a single point of failure
  - ▶ Misconfiguration can make the entire network vulnerable
- ▶ Many applications cannot be passed over a firewall securely
- ▶ Network performance slows down
- ▶ Unauthorized traffic can be tunneled or hidden
  
- ▶ Your users will constantly try to find new ways of bypassing your firewall

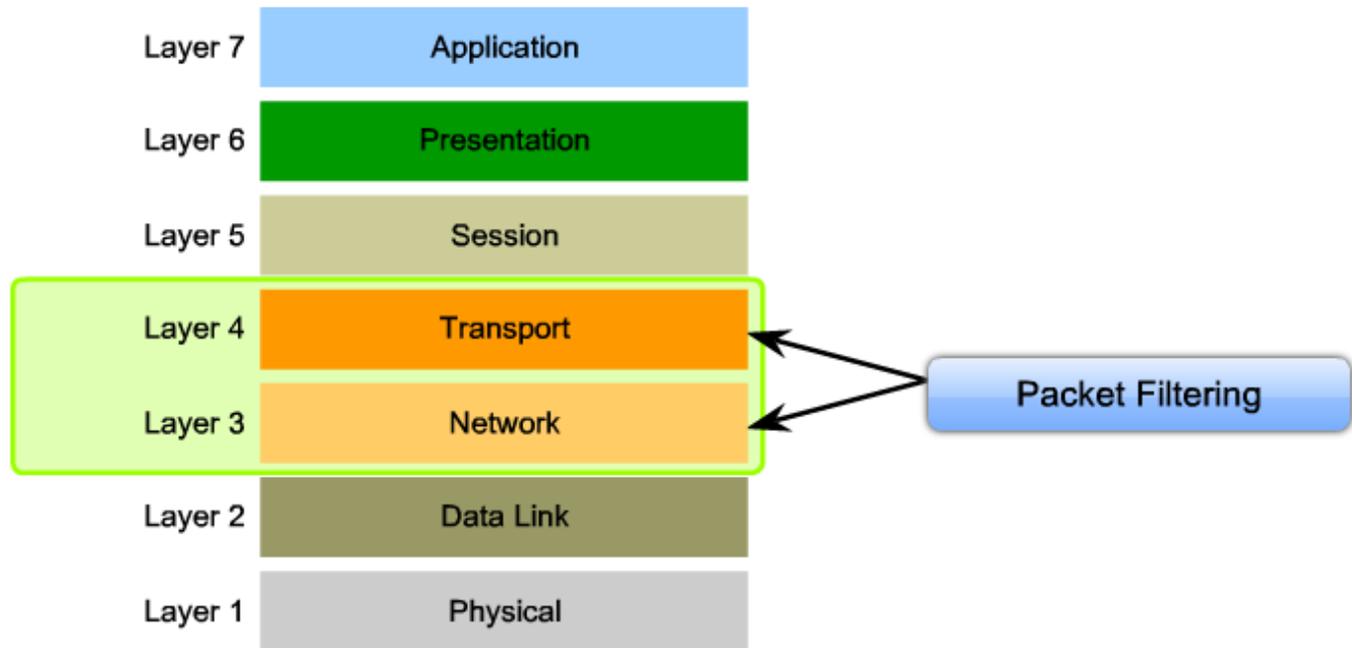
# Firewall flavors

---

- ▶ **Software-based**
  - ▶ Simple applications that make software decisions about traffic entering or leaving the machine they are running on
  - ▶ Suited for single computers or very small networks
  
- ▶ **Dedicated devices – hardware processing**
  - ▶ Cisco PIX
  - ▶ Cisco ASA

# Types of firewalls: Packet filtering

---



- ▶ Packet-filtering firewall (or stateless firewall)
  - ▶ Has a limited ability to filter packets based on layer 3 and layer 4 information.
  - ▶ That's right! Access lists!

# Packet filtering pros and cons

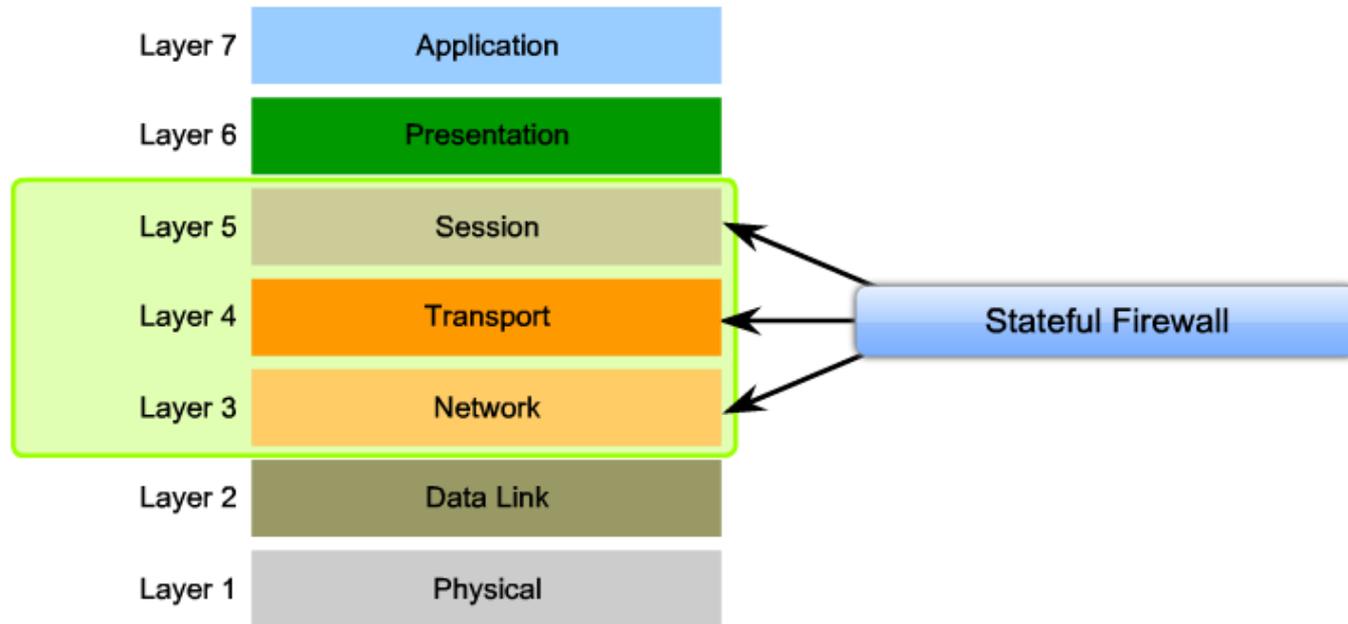
---

- ▶ Simple rules, easy to implement and update
- ▶ Low CPU requirements
- ▶ Supported by all routers
- ▶ Low cost
- ▶ Susceptible to IP spoofing
- ▶ Complex ACLs are difficult to maintain and update
- ▶ Limited functionality
- ▶ Stateless



# Types of firewalls – Stateful firewall

---



- ▶ Also monitors the state of connections
  - ▶ Initiation, data transfer, termination
- ▶ Can detect abnormal connection behaviour that might indicate attacks or exploits

# Stateful firewalls pros and cons

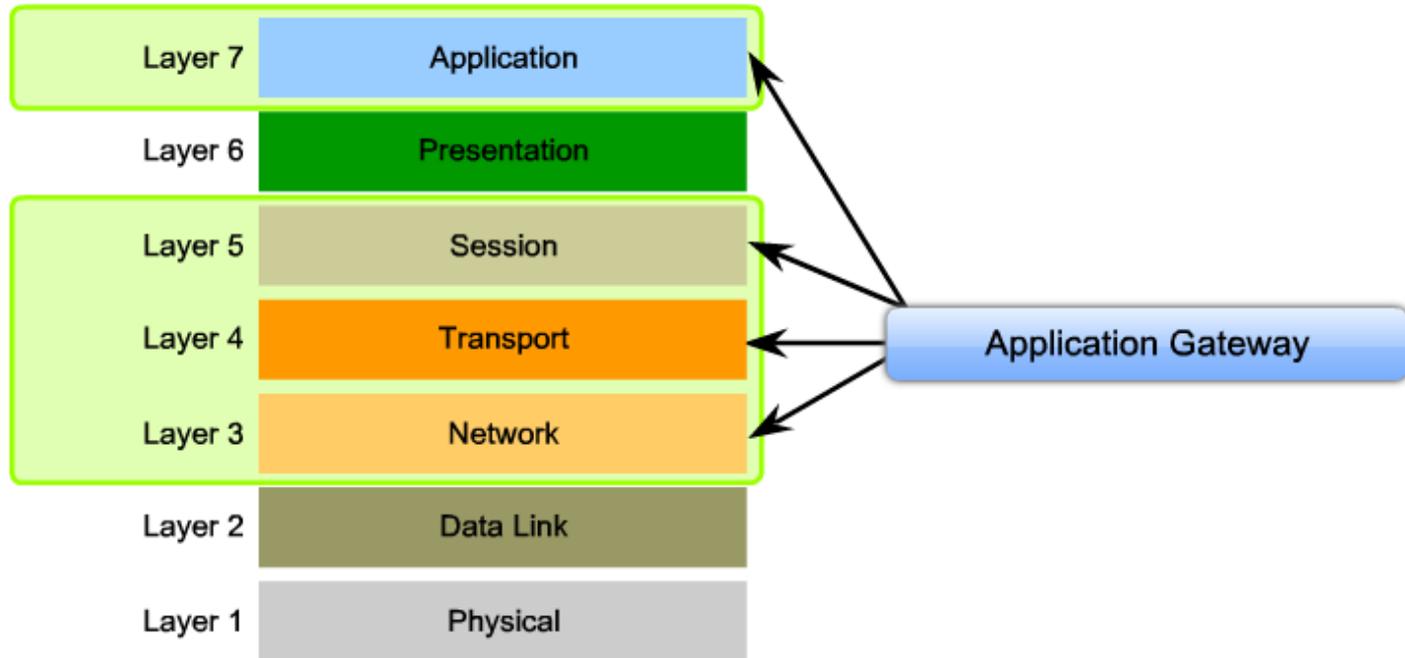
---

- ▶ More granular control for packet filtering
- ▶ Can defend against DoS attacks
- ▶ Offer more precise statistic data
- ▶ They do not examine the application-layer content
- ▶ Not all protocols are stateful (UDP, ICMP)



# Types of firewalls – Application Layer Gateway

---

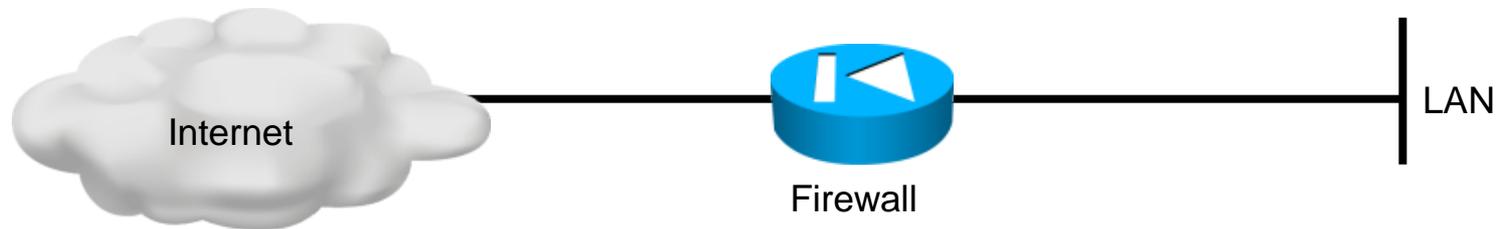


- ▶ Also known as a “proxy firewall”
- ▶ Also analyzes the application layer in its decisions
- ▶ Control and filtering is mostly done in software

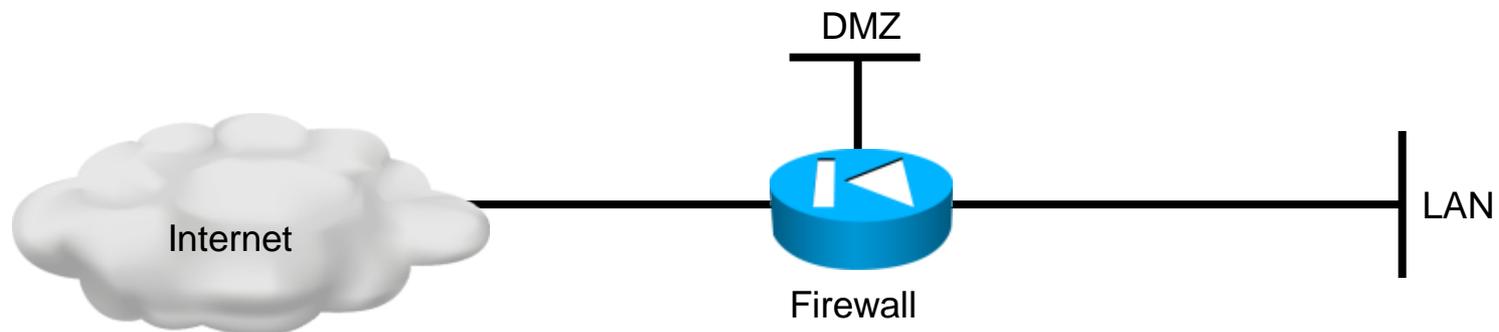
# Firewalls in network design

---

- ▶ The simplest design:



- ▶ The DMZ design:



# Trust

---

- ▶ Considering the second design, a firewall would have three interfaces:
  - ▶ A “trusted” interface facing the local network (inside)
  - ▶ An “untrusted” interface connecting to the Internet (outside)
  - ▶ A “DMZ” interface
- ▶ Usual policies:
  - ▶ No connections can be made from outside to the inside
  - ▶ The inside network can access the DMZ and the Internet
  - ▶ The DMZ can be accessed from the internet and the LAN
    - ▶ The place for public services (DNS, HTTP, SMTP, etc)

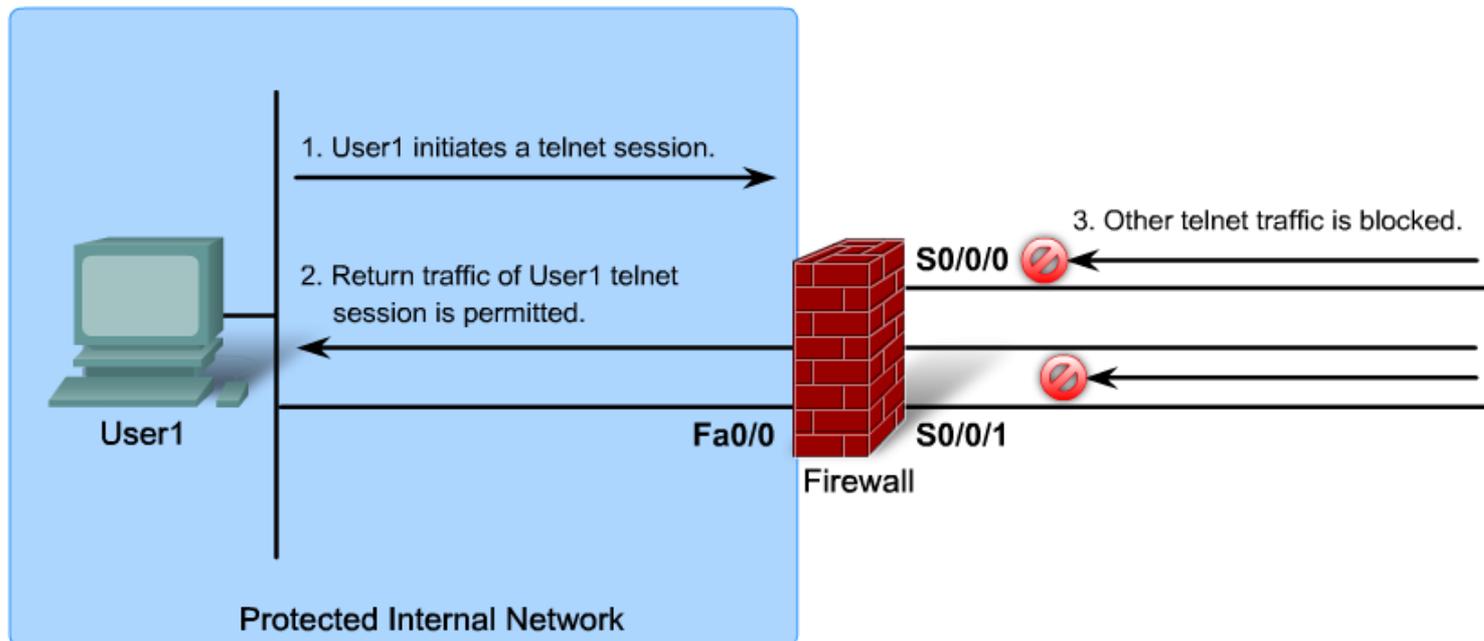
# The DMZ

---

- ▶ The DMZ is intentionally exposed
  - ▶ Public services must be ... publicly accessible
- ▶ For security reasons, the DMZ must not be allowed to connect to the LAN.
  - ▶ In case the DMZ is compromised, the internal network must still be secured.
- ▶ The firewall allows traffic to the DMZ, with restrictions
  - ▶ Only permit necessary traffic, block everything else
  - ▶ Must detect abnormal usage of DMZ services (attacks, exploits)

# CBAC = Context-Based Access Control

- ▶ Solution available within the Cisco IOS Firewall
- ▶ Intelligent TCP and UDP filter, inspects application layer protocol and session information
- ▶ Stateful session tracking



# CBAC

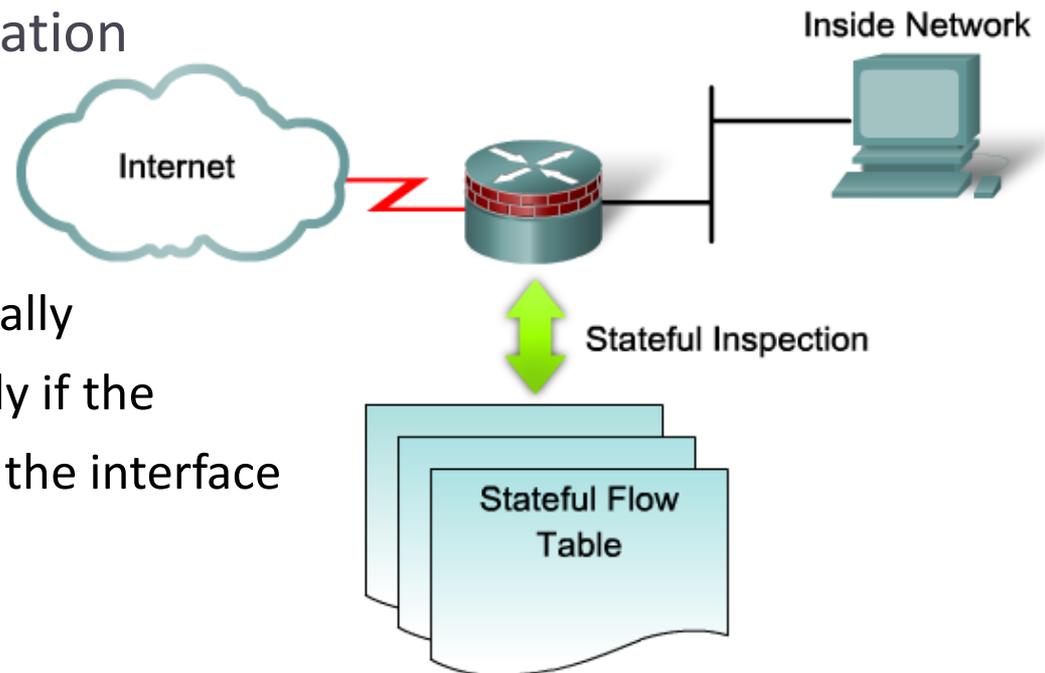
---

## ▶ Functionality

- ▶ Monitors TCP setup (three-way-handshake)
  - ▶ Tracks TCP sequence numbers
  - ▶ Inspects DNS queries and replies
  - ▶ Inspects ICMP message types
  - ▶ Supports applications that rely on multiple connections
  - ▶ Inspects embedded addresses (for NAT/PAT)
  - ▶ Inspects application-layer information
- 
- ▶ Based on timeouts for stateless protocols, to prevent spoofing
  - ▶ CBAC is not intended to protect against internal threats

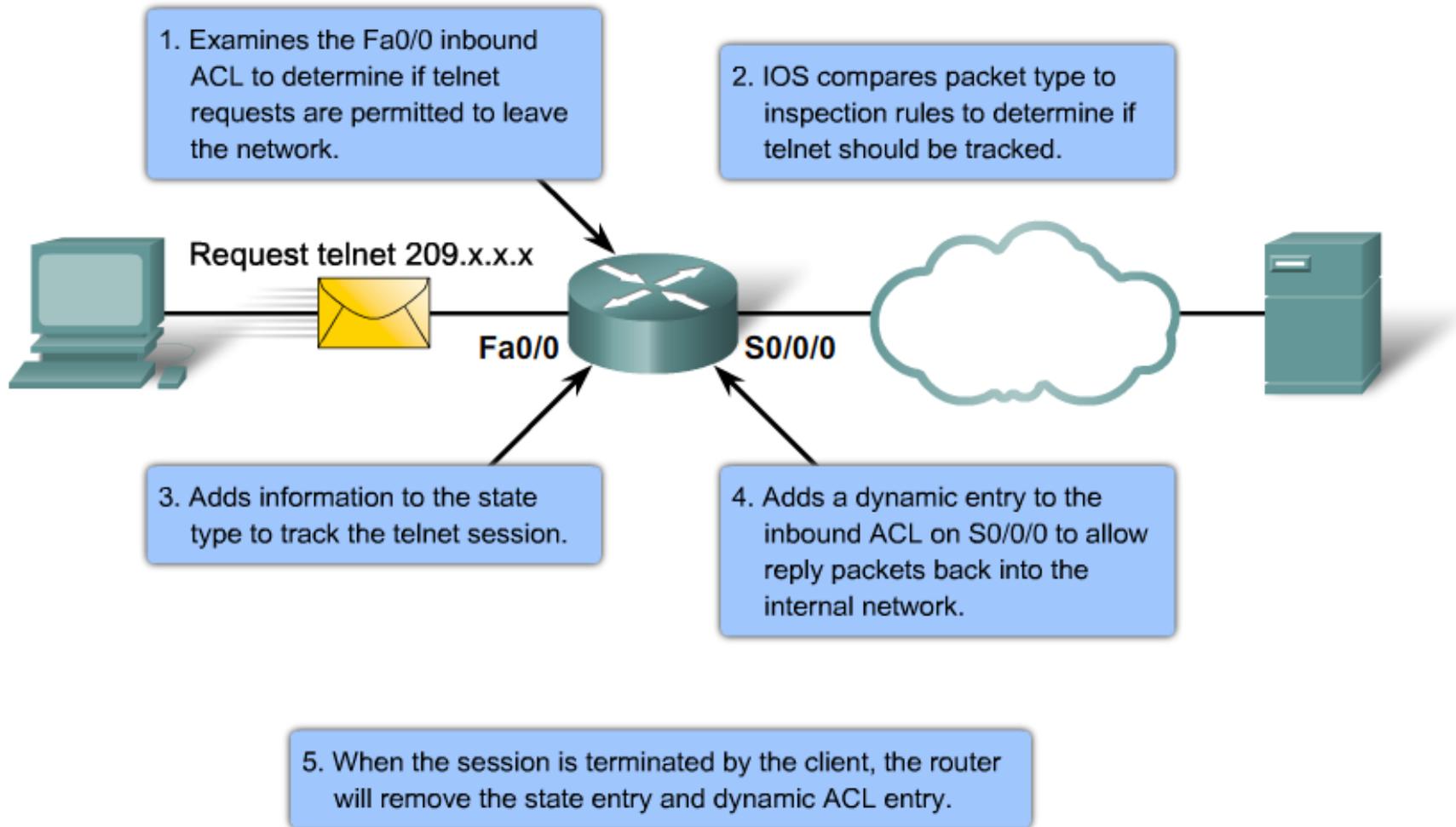
# How does CBAC work?

- ▶ Creates temporary openings in ACLs to allow valid traffic replies back inside the network
- ▶ CBAC tracks:
  - ▶ Application-layer protocol information
  - ▶ Connection state information



- ▶ The state table adapts dynamically
- ▶ Inspection rules are applied only if the packet passes the inbound ACL of the interface

# CBAC operation example



# CBAC protocol operation

---

## ▶ TCP handling

- ▶ Each session is described and tracked by:
  - ▶ Endpoint addresses and port numbers
  - ▶ Sequence numbers
  - ▶ Flags
- ▶ Packets are checked against the current state and discarded if they do not comply

## ▶ UDP handling

- ▶ There is no trackable connection state
- ▶ Traffic is allowed back in for a preconfigured time interval

## ▶ Other IP protocols

- ▶ CBAC inspects protocols that negotiate protocol numbers (FTP)
- ▶ Other protocols, like GRE and IPsec are treated in a connectionless manner

# CBAC inspection rules

---

- ▶ CBAC firewall rules are called “inspection rules”
- ▶ An inspection rule is applied to an interface with regards to the direction of traffic (in/out), just like an ACL
- ▶ The rule must be configured to inspect all the required protocols
  
- ▶ The Cisco Firewall engine recognizes illegal application-specific commands and can take several actions:
  - ▶ Generate alert messages
  - ▶ Protect certain system resources
  - ▶ Block packets from possible attackers

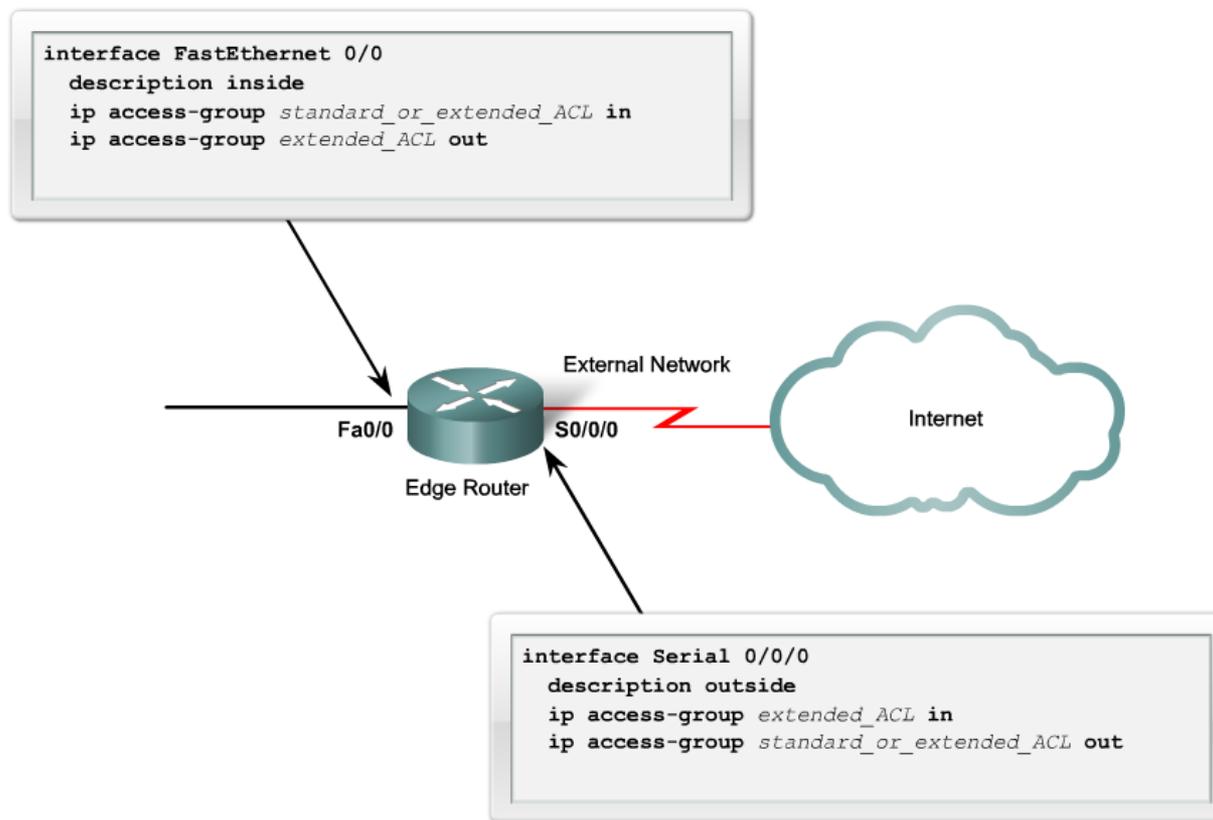
# CBAC and TCP-based DoS attacks

---

- ▶ The connection state database is also used to determine abnormal connections
- ▶ Three thresholds are provided by the Cisco IOS Firewall:
  - ▶ Total number of half-opened TCP sessions
  - ▶ Number of half-opened TCP sessions in a time interval
  - ▶ Number of half-opened TCP sessions for a certain host
- ▶ If a threshold is exceeded, the firewall acts in one of two ways:
  - ▶ Starts sending reset messages to the oldest connections in the table in order to free up resources
  - ▶ Temporarily blocks all SYN packets, to preserve resources

# Before configuring CBAC: ACLs

- ▶ Remember that traffic must be permitted through the interface ACL before it gets inspected by the CBAC rules



# Configuring CBAC – inspection rules

---

- ▶ An inspect rule can specify:
  - ▶ Generic protocols, like TCP, UDP, ICMP
  - ▶ Specific application-layer protocols
- ▶ An inspection rule consists of a series of statements
  - ▶ Each statement of a rule has the same rule name and a specific protocol to inspect

```
R2 (config)#ip inspect name FWRULE bittorrent
```

```
R2 (config)#ip inspect name FWRULE edonkey
```

```
R2 (config)#ip inspect name FWRULE pop3
```

```
R2 (config)#ip inspect name FWRULE smtp
```

```
R2 (config)#ip inspect name FWRULE http
```

```
R2 (config)#ip inspect name FWRULE https
```

- ▶ The firewall will inspect all TCP and UDP connections, but protocols in the inspect rules will be enhanced – application-level analysis

# Configuring CBAC – inspection rules options

---

- ▶ Other options available for each entry:

```
R2(config)#ip inspect name FWRULE ssh alert on audit-trail on  
timeout 3600
```

```
R2(config)#ip inspect name FWRULE irc alert off audit-trail off
```

- ▶ The “alert” keyword controls syslog messages
- ▶ The “audit-trail” keyword builds an audit trail of the specified events occurring in the firewall
  - ▶ A chronological sequence of audit records.
- ▶ If unspecified, the “alert” and “audit-trail” values are set accordingly to the following commands:

```
R2(config)#no ip inspect audit-trail
```

```
R2(config)#no ip inspect alert-off
```

- ▶ The “timeout” value overrides the default TCP and UDP timeout values

# Configuring CBAC – Alerts and audits

---

- ▶ CBAC has 2 types of logging functions: alerts and audits
- ▶ Alerts – messages concerning CBAC operation
  - ▶ Alert on low resources
  - ▶ Alert on detected DoS attack
  - ▶ Enabled by default and displayed on the console. Disable with:  
`R(config)#ip inspect alert-off`
- ▶ Alert example: SMTP attack attempt:  
`%FW-4-SMTP_INVALID_COMMAND: Invalid SMTP command from initiator  
(209.165.201.5:49387)`

# Configuring CBAC – alerts and audits

---

- ▶ Audits keep track of connections inspected by CBAC.
- ▶ Display messages when the router adds or removes an entry from the state table
- ▶ The audit record gives basic statistical info about the connection
- ▶ Disabled by default, enable with:
- ▶ Example audit message for initiating a Telnet connection:

```
R(config)#ip inspect audit-trail
```

```
%FW-6-SESS_AUDIT_TRAIL: tcp session  
initiator (192.168.1.2:32782) sent 22 bytes  
responder (209.165.201.1:23) sent 200 bytes
```

# Configuring CBAC

---

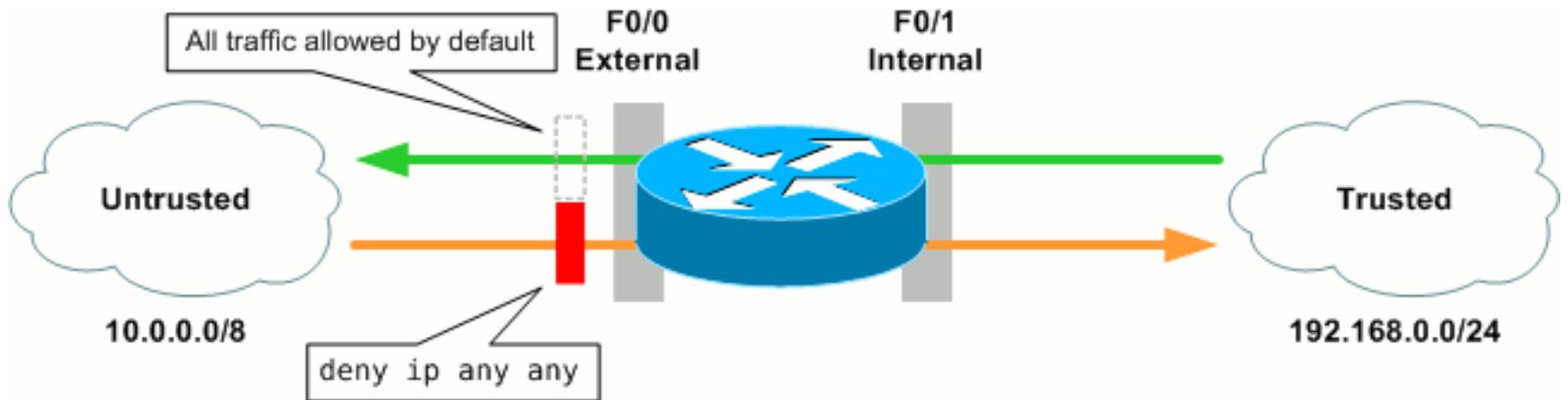
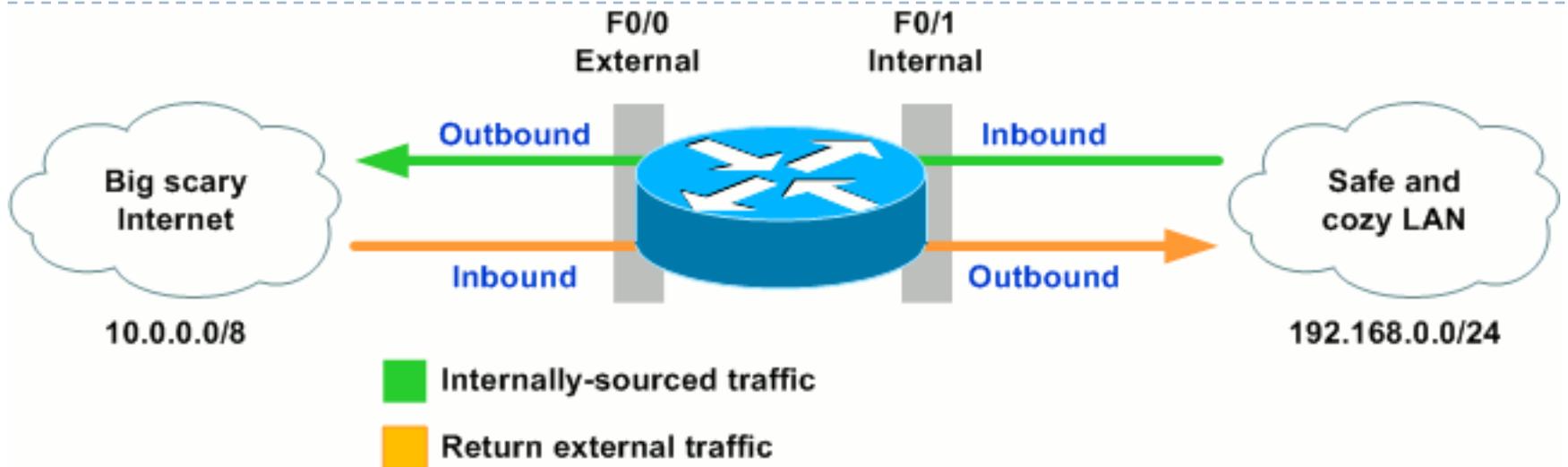
- ▶ Applying the inspection rule to an interface:

```
R2(config)#int Serial 0/0/0
```

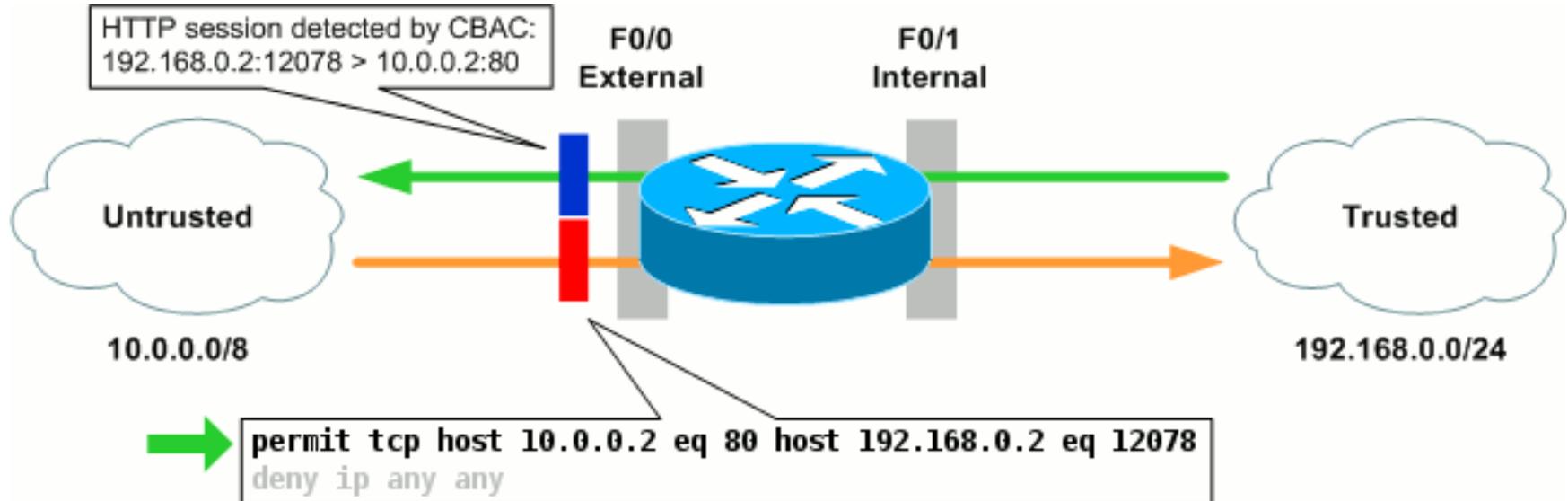
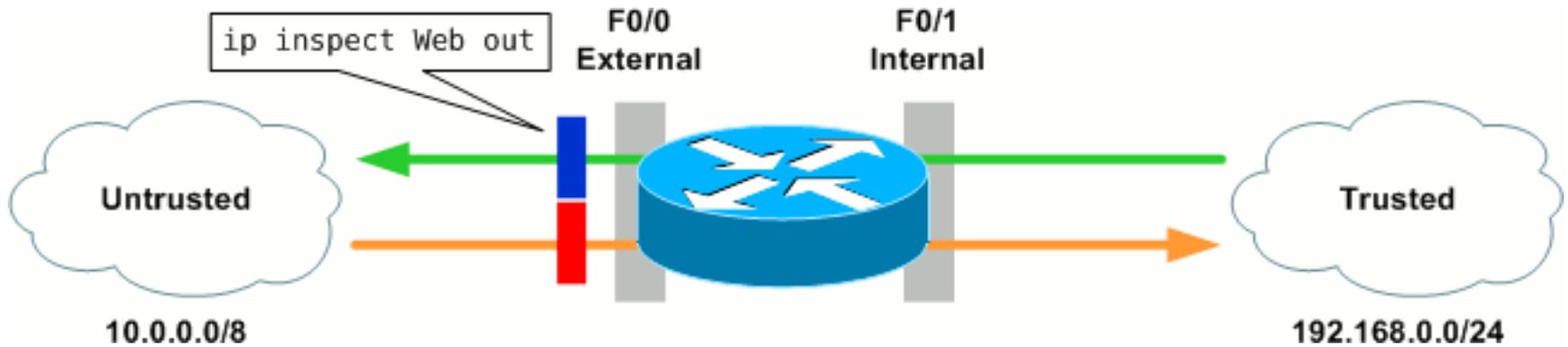
```
R2(config-if)#ip inspect FWRULE out
```

- ▶ General rules when applying ACLs and inspection rules:
  - ▶ On the “inside” interface, use an ACL that permits only allowed traffic to leave the network
  - ▶ On the “outside” interface apply the inspection rule on outbound / on the “inside” interface apply the rule on inbound
  - ▶ On the “outside” interface, will be created a dynamic ACL that permits the reply traffic (on inbound), the traffic to be inspected by CBAC

# CBAC example



# CBAC example



# Verifying CBAC

---

```
R2#show ip inspect config
```

```
Session audit trail is enabled
```

```
Session alert is enabled
```

```
one-minute (sampling period) thresholds are [400:500] connections
```

```
max-incomplete sessions thresholds are [400:500]
```

```
max-incomplete tcp connections per host is 50. Block-time 0 minute.
```

```
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
```

```
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
```

```
dns-timeout is 5 sec
```

```
Inspection Rule Configuration
```

```
Inspection name FWRULE
```

```
  bittorrent alert is on audit-trail is on timeout 3600
```

```
  edonkey alert is on audit-trail is on timeout 3600
```

```
  ftp alert is on audit-trail is on timeout 3600
```

```
  pop3 alert is on audit-trail is on timeout 3600
```

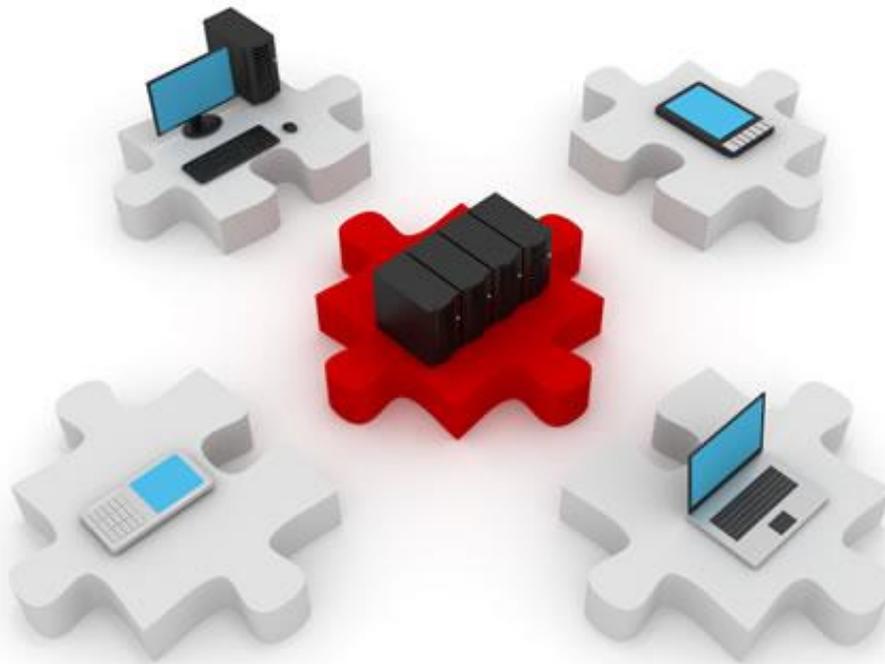
```
  smtp max-data 20000000 alert is on audit-trail is on timeout 3600
```

```
  http alert is on audit-trail is on timeout 3600
```

```
  https alert is on audit-trail is on timeout 3600
```

```
  ssh alert is on audit-trail is on timeout 3600
```

```
  irc alert is off audit-trail is off timeout 3600
```



# Windows Firewall

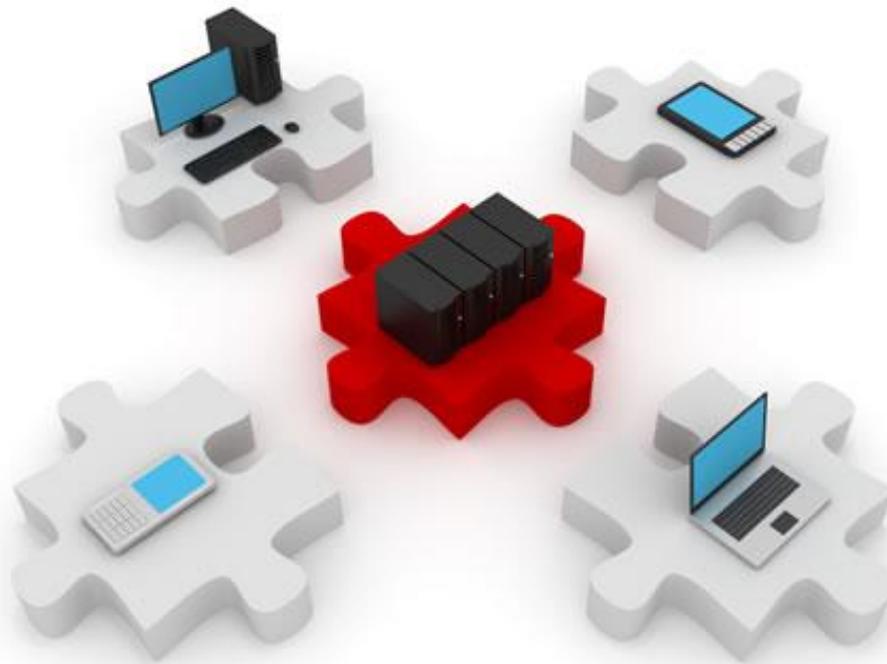
An entirely new approach to security

# Windows Firewall design and deployment

---



*\*The chain is no weaker than its strongest link\**  
Photo by TôHell, 2003-09-23 in Slagsta, SE



Nope, just kidding :)

THE END!!!