Francesco PALMIERI
Università di Salerno

Securing Internet communications

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it

License & Disclaimer

License Information

Disclaimer

- We disclaim any warranties or representations as to the accuracy or completeness of this material.
- Materials are provided “as is” without warranty of any kind, either express or implied, including without limitation, warranties of merchantability, fitness for a particular purpose, and non-infringement.
- Under no circumstances shall we be liable for any loss, damage, liability or expense incurred or suffered which is claimed to have resulted from use of this material.

Topics

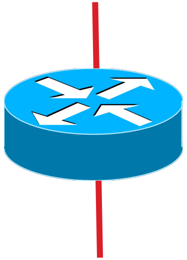
- Security enforcement devices
- Traffic filtering strategies and policies
- Network access control: techniques and tools
- Implementing simple network access control policies

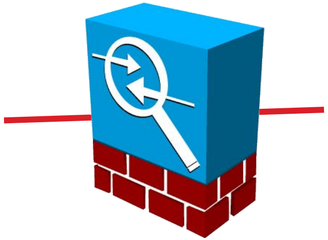
Current Topic

- Security enforcement devices
- Traffic filtering strategies and policies
- Network access control: techniques and tools
- Implementing simple network access control policies

Security enforcement devices

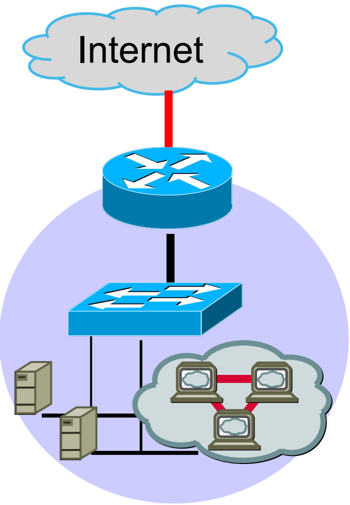
- In a common security architecture in order to implement security policies we can rely on two perimeter control devices:
  - **Routers** (typically those located on the neywork border)
  - **Firewalls**





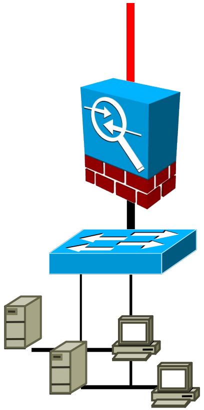
Border Router

- The border router is the first barrier protecting its internal network
  - difficult to circumvent by malicious end-users
- It allows the centralization of a good number of security checks
- Its protection is fundamental
  - a compromise may open access to the internal network
  - an inadequate filtering policy can expose the internal network to attacks
  - corruption of routing tables can cause disruptions and unauthorized access to data
- A properly configured router can minimize effects from internal sites compromised by attacks or hostile activities (insider threats)



Firewall

- Firewall is an english term with original meaning of a fire isolation barrier
- It is the main passive perimeter defense component
- It has security enforcement tasks, in the broadest sense of the term, with the aim of controlling traffic between two or more networks:
  - allowing only what is specifically authorized by the security policy
  - detecting and reporting any attempts to violate the security policy
  - possibly carrying out additional auditing and accounting functions
  - it can also connect at the link or network layer two or more network segments



Why installing a firewall

- To implement a security policy:
  - Able to allow controlled access to systems or services of a protected network:
    - Authorized users only
    - Only to authorized systems
  - Able to allow users and systems of a protected network to access the systems and services of an outside (untrusted) network in a controlled way:
    - only if the risk is acceptable
    - recording all their activities

Firewall: pros

- Centralization of security policies
  - Can result in a Single point of failure (can be a disadvantage)
- Relying on a special purpose solution able to optimize traffic filtering operations (through appropriate HW)
- Ability to inspect traffic from data link to application layers
- Stateful control of sessions

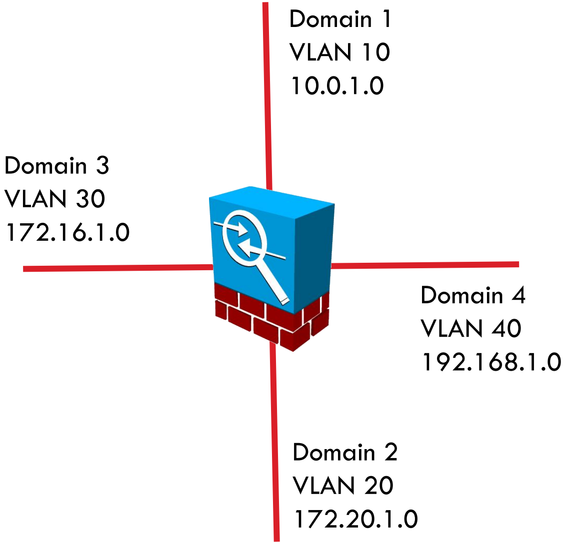
Firewall: cons

- Difficulty in coping with with non-trivial protocols
- Performance / throughput
  - It can turn into a bottleneck
  - user perception can be negative due to service limita
- Complex management
  - Configuration requirs specialization
  - Verification and analysis of logs is not straightforwar

- Excessive sense of trust and internal insecurity
- High costs for performance beyond Gigabit

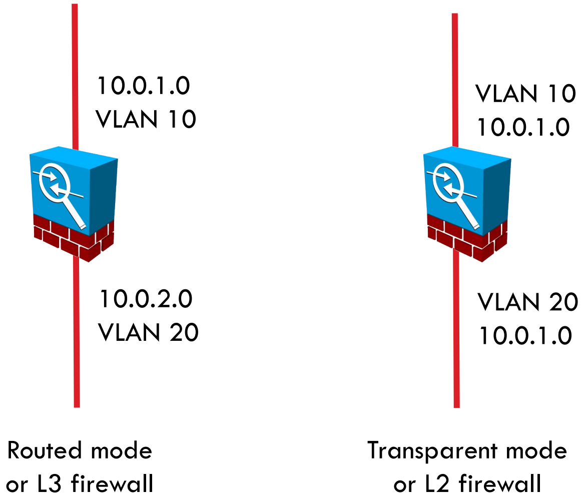
Implementation and basic functions

- Network device with at least 2 network interfaces
  - Each interface identifies a **separate security domain** on a different network segment (VLAN)
- Can remap IP addresses (NAT)
- Filters traffic between different zones/domains through predefined rules (access control policies)
- It can mediate access to specific applications for control and inspection purposes:
  - Proxy service access
  - Content filtering (selective content filtering)
  - Deep packet inspection and traffic analysis
  - Endorce bandwidth limitations on specific traffic types



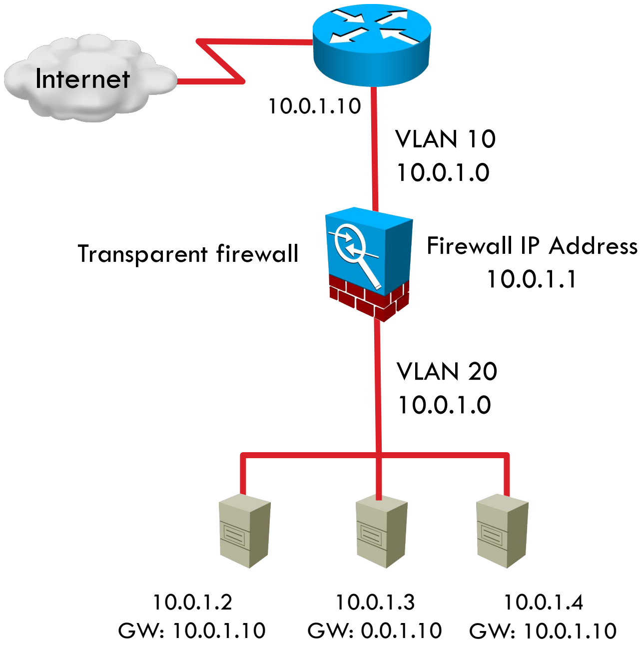
Firewall: operating modes

- A firewall can operate in two ways:
  - **Routed**: Operates at level 3, segmenting different networks based on IP addresses
  - **Transparent**: Operates at level 2, segmenting on MAC address basis
- A routed firewall looks like a layer 3 device and needs an IP address/network on each interface associated to a segment
  - Routes IP/IPv6 traffic between the various interfaces
  - Supports the most common routing protocols



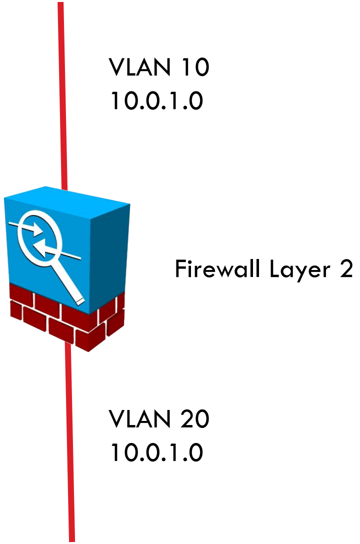
Transparent mode operations

- Layer 3 traffic must be explicitly allowed to pass through the firewall
  - However it performs packet screening/filtering from network to application layers
- The segments connected to the interfaces must be on the same layer 3 subnet
- The firewall IP address must not be configured as the default gateway for connected devices
  - Devices must point to the router ahead of the firewall (passed through transparently)
  - Each interface identifies a different segment/VLAN even if associated with the same IP network (the firewall bridges different segments)



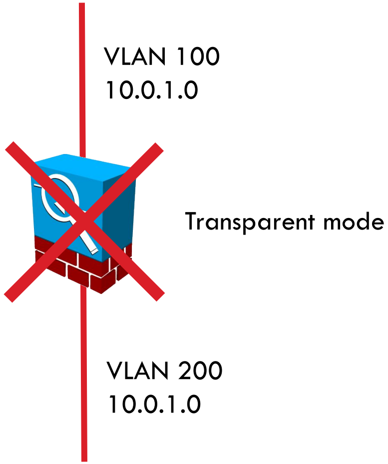
Transparent mode benefits

- Flexible, integrated and easy to manage:
  - IP-level redirection not required
  - No NAT to configure
  - Routing and redirection problems cannot occur (it does not perform routing)
- Totally invisible from the outside
- Greater robustness



Transparent mode unsupported features

- The following features are typically not supported by a firewall in transparent mode:
  - NAT
  - Routing protocols (e.g. OSPF, RIP, BGP)
  - IP / IPv6
  - DHCP relay
  - QoS
  - Multicast
  - VPN termination

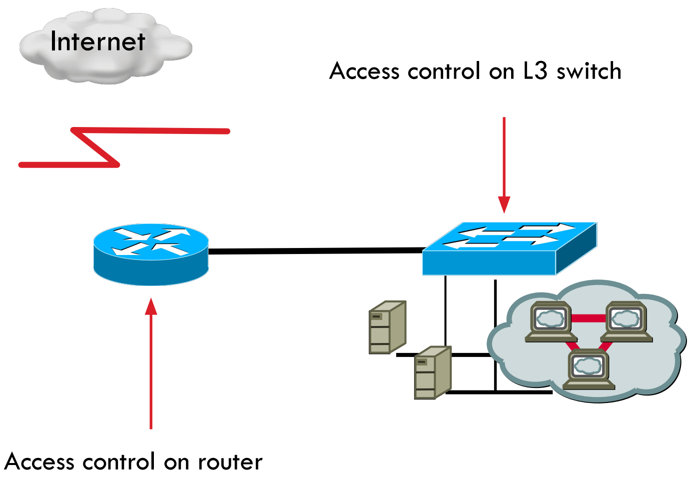


Current Topic

- Security enforcement devices
- Traffic filtering strategies and policies
- Network access control: techniques and tools
- Implementing simple network access control policies

Traffic filtering: putting controls on border router

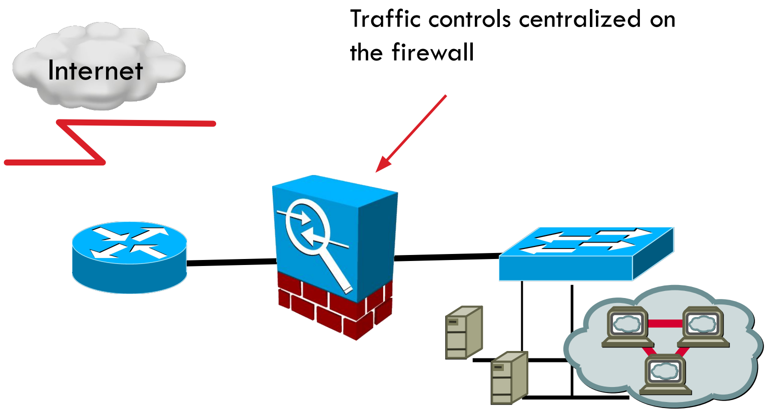
- Router and Switch Layer 3 devices provide simple access control mechanisms
  - Based on stateless traffic filtering
  - Filters based on IP address and TCP/UDP Ports
- The use of complex controls with a significant number of filtering clauses, entails a certain increase in CPU load in the forwarding activity.



  - can be acceptable if the Router or L3 Switch provide hardware implementation of access control mechanisms
- The truly advantage is that such devices are already present in any network, partitioning it in a natural way

Traffic filtering: putting controls on firewalls

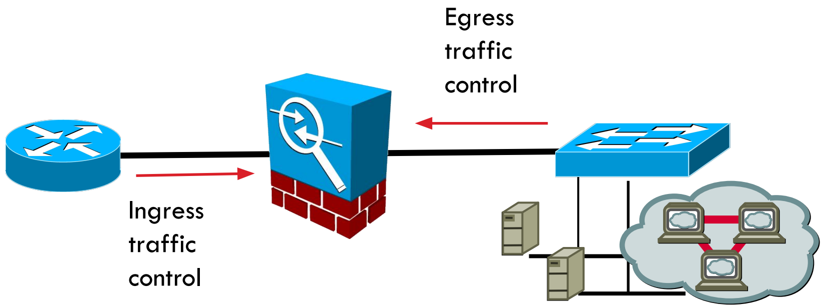
- The introduction of a firewall reduces the CPU burden associated to traffic control/filtering activity on routers or L3 switches
- The centralization of control policies on the firewall constitutes a significant advantage from the management perspective:
  - reduces the configuration complexity
  - centralizes the management of logics and filtering problems
  - It allows you to simultaneously protect thousands of machines



- This policy does not scale in the presence of large traffic volumes and becomes a performance bottleneck that can be exploited to create DoS

Chosing the right filtering location

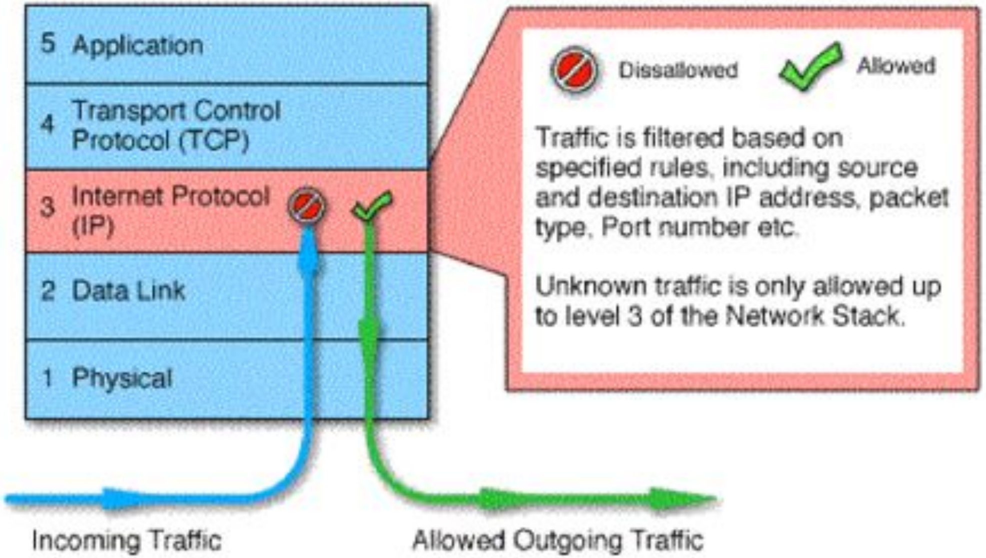
- **Ingress** filtering:
  - natively knows from which interface packets are coming in
  - provides protection of internal networks
- **Egress** filtering:
  - enables checking also the locally generated traffic exiting the domain
  - blocks what should not come out



- The controls should be located **as close as possible** to the traffic origin

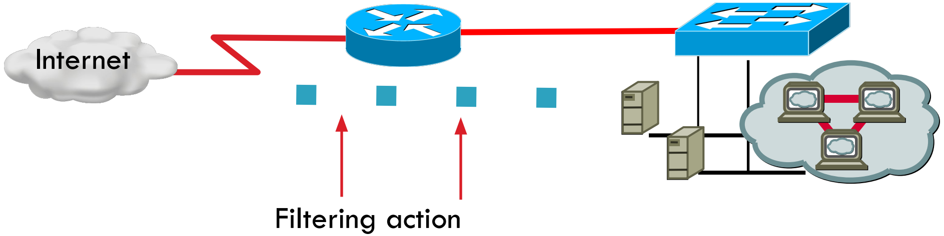
Filtering Parameters

- IP Header
  - source address
  - sestination address
  - protocol
  - flags, options (source routing)
- TCP/UDP Header
  - source port
  - destination ports
  - flags TCP (SYN, ACK)



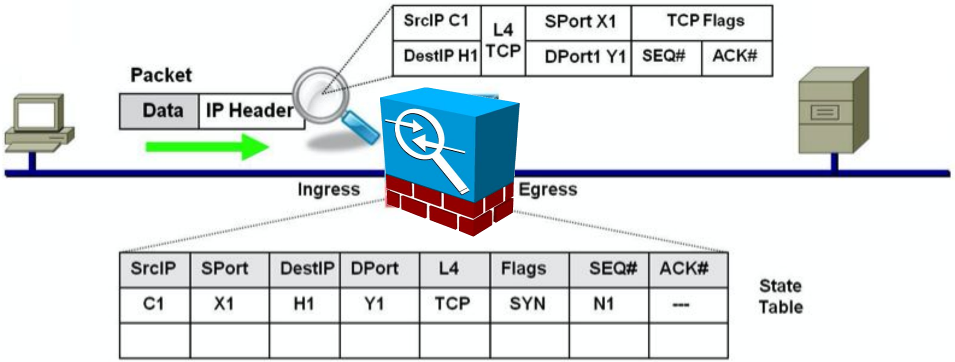
Stateless filtering (router)

- Based only on IP addresses, TCP / UDP ports (source and destination), and protocol
- Controls are carried out independently one packet at a time (no memory/state)
- There is no perception of the flow of packets belonging to an end to end connection
- Packets can also come from interfaces other than the one on which they exit (asymmetric forwarding phenomena are tolerated)



Stateful filtering (firewall)

- When a new connection is established, if the filtering rules do not block it, then the related information are used to add an entry (**session**) of a **connection status table**.
- Subsequent incoming packets will be handled according to their belonging to one of the active connections (or data flow sessions) whose status is saved in the table.
- When the connection is terminated, the corresponding entry in the table is deleted
- The table contains:
  - Unique session ID
  - Connection status (*handshaking*, *established*, *closing*)
  - Packet sequencing information
  - Source and destionation addresses/ports
  - Network interfaces used



Filtering (access control) policies

- Before defining any filtering policy aimed at performing access control, a careful preliminary assessment must be made, by considering:
  - Who needs access?
  - When and how?
  - From where?
  - At which time/date?
  - What services does it need?
  - What protocols does it use?
  - What QoS (e.g bandwidth) does it require?

Filtering (access control) policies

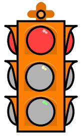
A firewall (or router) can operate in two diametrically opposite ways:

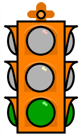
- **Deny All**: Anything that isn't specifically allowed is denied. **High security**
  - Block all traffic and each service must be implemented on a case-by-case basis
  - More conservative policy in terms of protection
  - the number of choices available to the user is limited

- **Allow All**: Anything that is not specifically denied is allowed. **Ease of management**
  - Forward all traffic and each malicious service must be closed on a case-by-case basis
  - Increasing difficulties in guaranteeing security as the network grows.
  - Rarely used in security schemes, however it is may cover several specific cases

Selective traffic filtering

Service	Port	Protocol
echo	7	TCP/UDP
discard	9	TCP/UDP
systat	11	TCP/UDP
daytime	13	TCP/UDP
netstat	15	TCP
quotd	17	TCP/UDP
chargen	19	TCP/UDP
ftp-data	20	TCP
ftp	21	TCP
ssh	22	TCP/UDP
telnet	23	TCP
smtp	25	TCP
time	37	TCP/UDP
rlp	39	TCP/UDP
whois	43	TCP/UDP
tacacs	49	TCP/UDP
domain	53	TCP
whois++	63	TCP/UDP
bootp	67-68	UDP
tftp	69	UDP
gopher	70	TCP
finger	79	TCP
http	80	TCP
link	87	TCP
supdup	95	TCP
pop2	109	TCP
pop3	110	TCP
sunrpc	111	TCP/UDP
auth	113	TCP/UDP
nntp	119	TCP
ntp	123	TCP/UDP
nbios-ns	137	TCP/UDP

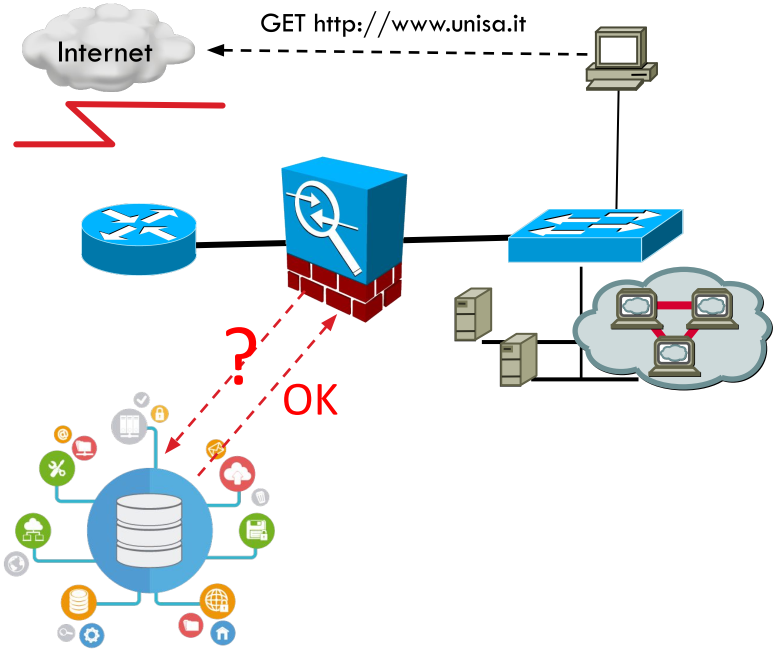




Service	Port	Protocol
nbios-dgm	138	TCP/UDP
nbios-ssn	139	TCP/UDP
imap	143	TCP
NeWS	144	TCP
snmp	161	UDP
snmptrap	162	UDP
xdmcp	177	UDP
irc	194	TCP/UDP
wais/Z39.50	210	TCP
imap3	220	TCP
ldap	389	TCP/UDP
netware-ip	396	TCP/UDP
rmt	411	TCP
https	443	TCP
exec	512	TCP
biff	512	UDP
login	513	TCP
who	513	UDP
shell	514	TCP
syslog	514	UDP
printer	515	TCP/UDP
talk/ntalk	517-518	TCP/UDP
route	520	UDP
timed	525	TCP/UDP
uucp	540-541	TCP
mountd	635	TCP/UDP
wins	1512	TCP/UDP
radius-old	1645-1646	UDP
radius	1812-1813	UDP
openwin	2000	TCP
NFS	2049	TCP/UDP
X11	6000-6063	TCP

- It can be a good practice blocking or selectively filtering potentially dangerous services
- Only allow access to an extremely limited number of services (e-mail, www, ftp) provided by specific and possibly controlled hosts

Content filtering



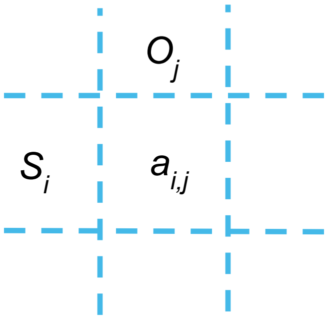
- Filtering unwanted, objectionable, and harmful content through URL inspection
- Requires the use of third party and always up-to-date knowledge bases
  - Resource classification DB
  - Categorization engines
- The firewall performs payload inspection and before admitting the session checks the content type against local policies
  - e.g. block gambling, drugs, crime-related URLs

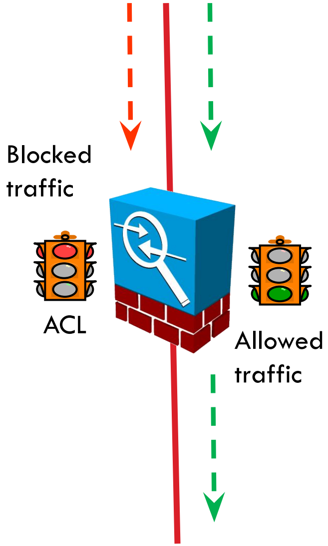
Current Topic

- Security enforcement devices
- Traffic filtering strategies and policies
- Network access control: techniques and tools
- Implementing simple network access control policies

Traffic filtering mechanisms: ACL

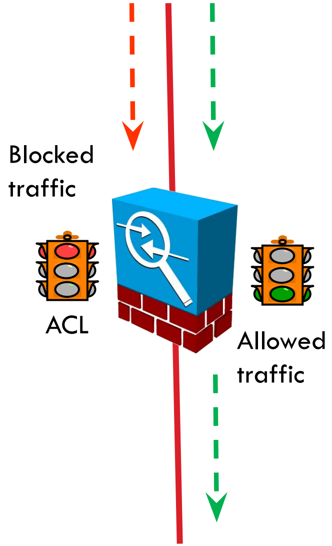
- The simplest and most immediate way to implement security schemes and policies is traffic filtering (packet filtering)
- Routers, switches and firewalls support lists of filtering (or access control) rules known as ACLs: **Access control Lists**
- An ACL represents a column in the Lampson's **access control matrix**, where:
  - S_i: j-th subject to be controlled (e.g. a netblock)
  - O_j: j-th object to be protected (e.g. an interface)
  - $a_{i,j}$: access rigths of S_i on O_j (e.g. permit or deny)





Traffic filtering mechanisms: ACL

- Filtering rules can be applied at:
  - **data link** layer (based on MAC addresses)
  - **network** layer (based on IP addresses)
  - **trasport** layer (based on ports or protocol)
- Additional elements may be checked:
  - Date and time of application
  - Session flags or status (established, closing etc.)
- Each packet received is compared with each rule, in the order in which it appears on the list, to decide if it has to be forwarded or dropped
  - The application of controls takes place on an interface basis
  - Eligible actions are **permit** (or allow) and **deny** (or drop)
  - The direction of application of the controls (**inbound** or **outbound**) is significant and defines the origin of the traffic concerned

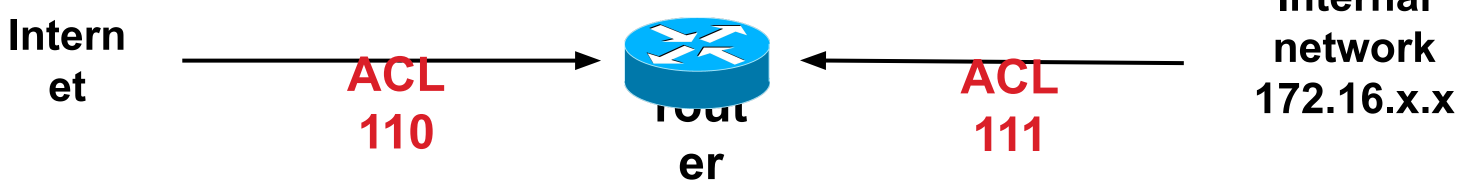


ACL Syntax

- Only one ACL can be applied to an interface in each specific direction (inbound or outbound):

```
interface ethernet 0
  ip access-group 110 in
  ip access-group 111 out
```

- In the example below, the ACL 110 and 111 are applied respectively to the input and output on the border interface that connects a router to the outside world



ACL Syntax

- An ACL is made up of rules scanned in sequence until the first match

list identifier protocol destination address/port

```
access-list 110 permit tcp any gt 1024 172.16.0.0 0.0.255.255 eq 23
```

action src addr/port

- The masks associated with the addresses are in "reverse dotted mask" format or “/msklen” in format (e.g. `0.0.0.255` is equal to `/24`)

ACL Syntax

- **Long search**: the search is carried out until there is a matching (rule with permission or denial found) or until the list is finished;
- Efficiency depends on the **order**: the most frequent matching element should be the first in the list
- Removing a permission may be without effect
- The **any** option replaces 0.0.0.0 as the IP address and 255.255.255.255 as the wildcard mask. It results in a matching with any compared address.
- The **host** option replaces 0.0.0.0 as a mask. This mask requires that all the bits of the address match. Compare exactly one address.

ACL Syntax

- Every ACL terminates with an implicit "`deny any any`" clause
- It is possible to use relational operators in ACLs: `eq neq, gt, lt:`

```
access-list 110 deny tcp 192.168.1.0 0.0.0.255 any eq www
access-list 110 deny tcp any eq ftp 192.168.1.25
```

- ACLs can be assigned logical names

```
ip access-list extended allowt permit tcp host 192.132.34.17  any eq 23
```

- It is possible to define rules in ACL that can be activated on a date/time basis, specifying a "time-range" of validity and a periodic or absolute scope

```
time-range no-http periodic weekdays 8:00 to 18:00
access-list 110 deny tcp any any eq http time-range no-http
```

ACL Syntax

- The "established" clause at the end of a rule identifies all TCP connections that have passed the setup phase (3 way handshake)

```
access-list 110 permit tcp any any estabilished
```

  - allows you to block all incoming traffic from the outside, with the exception of return TCP traffic, due to a TCP session started from the inside.
  - checks, on incoming TCP packets, the presence of the TCP ACK or RST flags:
    - if they are present, traffic is allowed,
    - otherwise it is assumed that the traffic has been generated from the outside and will be blocked.

ACLs on Linux: iptables

- Simple ACLs can be implemented as well under linux with iptables
- Iptables is used to set up, maintain, and inspect the tables of IPv4 packet filter rules in the Linux kernel.
- Several different tables may be defined. Each table contains a number of built-in chains and may also contain user-defined chains.
- chain = list of rules which can match a set of packets
  - each rule specifies criteria for a packet and an associated target, namely what to do with a packet that matches the pattern
  - We are interested in the FORWARD built-in chain:
    - packets that have been routed and were not for local delivery will traverse this chain.

ACL Syntax: iptables

- It is possible a new user-defined chain by the given name

```
iptables –N acl111
```

- … and apply on specific inbound or outbound interfaces

```
iptables -A FORWARD -i eth1 -o eth0 -j acl110
```

- Also a default policy for the chain can be specified

```
iptables -P FORWARD DROP
```

- Possible targets are
  - accept = let the packet through
  - drop = drop the packet on the floor

ACL Syntax: iptables

- Syntax is very intuitive and based on traditional shell command-line

list identifier protocol destination address/port

```
iptables -A myacl -p tcp -s 172.16.1.0/24 -d 192.168.1.1 --dport domain -j ACCEPT
```

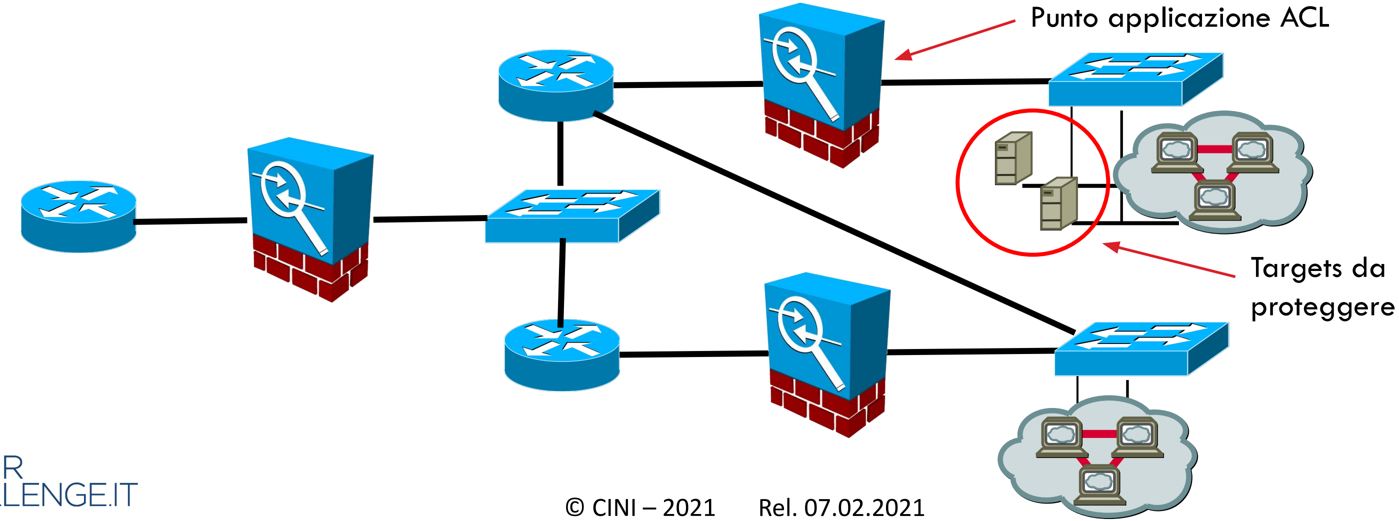
src addr/mask action

- very similar to traditional ACLs and also provides an “established” facility

```
iptables -A acl110 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Applying ACL in the correct places

- ACLs should be placed **as close as possible** to the **target** to be protected
- This allows to **restrict** the **size** of the **security domain** in order to increase the effectiveness of the filtering policies implemented and make the solution more **scalable**



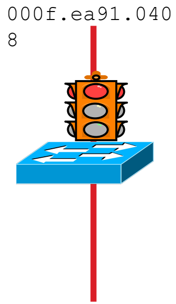
ACL applied on a switch

- Filtering can be applied also at the link layer
  - e.g. by authorizing only one host to traverse an interface

```
mac access-list mac-01
  permit host 00c0.4f00.0407 any
```

```
interface eth 1
  mac port access-group mac-01 in
```

```
iptables -A FORWARD -i eth1 -m mac --mac-source 00:C0:4F:00:04:07 -j ACCEPT
```



- It can be helpful to completely lock the mac of a compromised host

```
mac-address-table static 000f.ea91.0408 vlan 1 drop
```

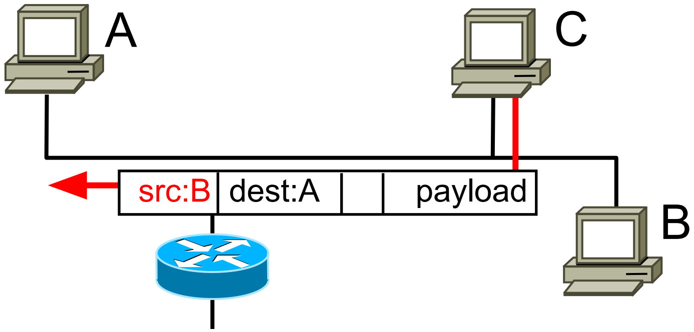
```
iptables -A FORWARD -m mac --mac-source 00:0F:EA:91:04:08 -j DROP
```

Current Topic

- Security enforcement devices
- Traffic filtering strategies and policies
- Network access control: techniques and tools
- Implementing simple network access control policies

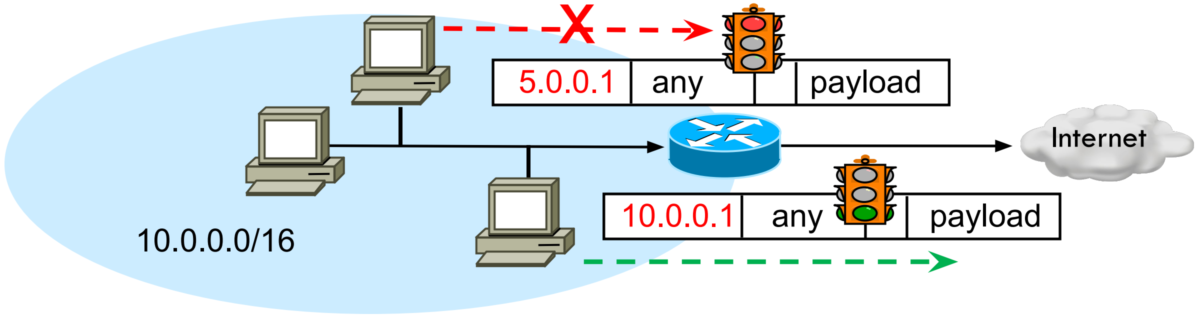
IP address spoofing

- The source IP address is currently the only mechanism for identifying the source available on the Internet
- The falsification/forgery of this data is the basis of most of the attacks and hostile actions
- Spoofing consists in falsifying the source address
  - Any user is able to generate IP packets with any value of the fields provided by the protocol structure
  - Therefore it is immediate to change the source address of the IP packets to prevent any form of identification
  - The result is that C in attacking A assumes the identity of B



Inbound anti-spoofing filtering

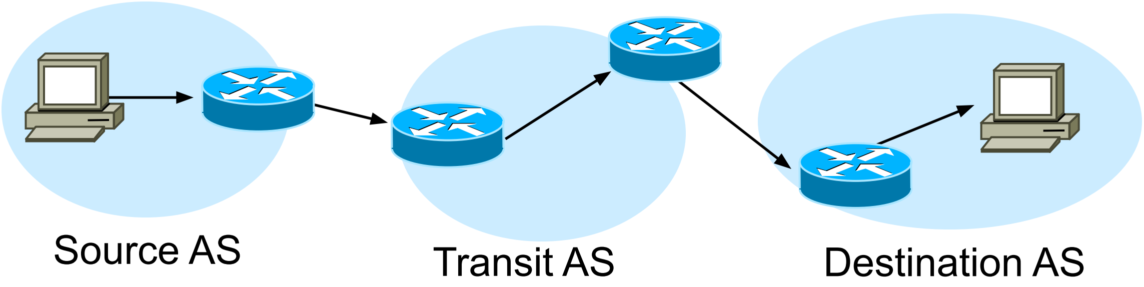
- Solution: checking and enforcing the correctness of the origin of the generated packets



- Inbound filtering policy (RFC 2827, 2000): A border router forwards only packets with legitimate source addresses

Practical implementation problems

- It is necessary that all the organizations involved and the transit ISPs do it
- Everything is based on a collaboration and trust logic working at a global level
  - If 10% of ISPs do not implement it, it is ineffective
- Another solution: enforcing / IP validation of sources at AS peering level



- A packet can only pass if the transit AS validates the source

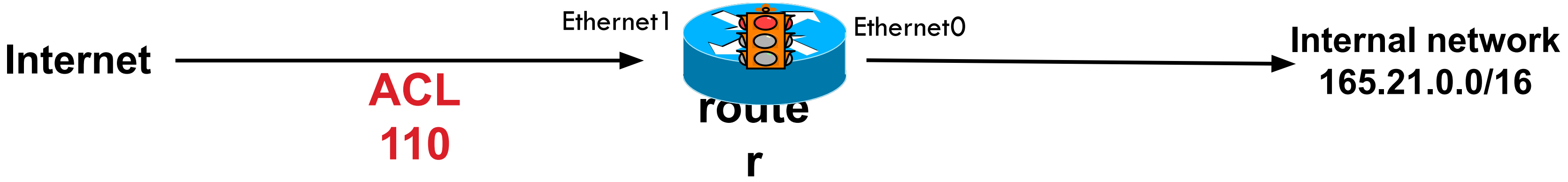
Inbound Anti-Spoofing filters

- The easiest way to protect yourself is to discard all incoming traffic with inadmissible source addresses with respect to the traffic origin

```
interface ethernet 1
    ip access-group 110 in
```

```
# blocca traffico spoof entrante da eth1
iptables -A FORWARD -i eth1 …
```

- Block all traffic with source addresses 165.21.0.0/16 if coming from outside (they are my internal addresses!)



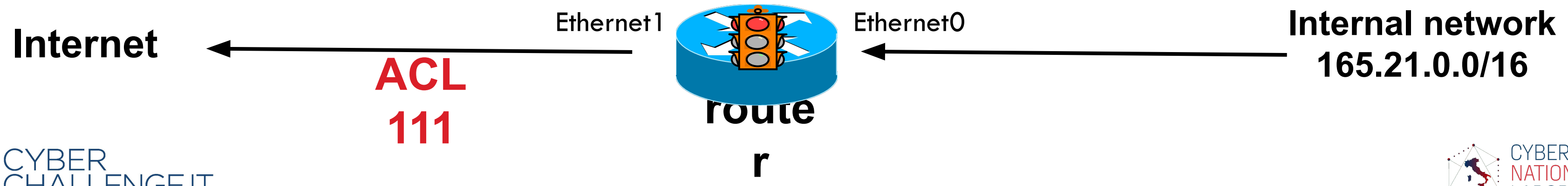
Outbound Anti-Spoofing filters

- To also prevent voluntary or involuntary spoofing from the inside of your network to the outside, similar filtering must be applied to outbound traffic

```
interface ethernet 1
      ip access-group 111 out
```

```
# non inoltrare il traffico spoof da eth0
iptables -A FORWARD -i eth0 …
```

- Block any outgoing packet with source address that does not fall on the network 165.21.0.0/16



Anti-Spoofing ACLs

Inbound Anti spoofing

! Block traffic from the outside with internal source addresses:
```
access-list 110 deny ip 165.21.0.0 0.0.255.255 any log
access-list 110 permit ip any any
```

iptables -A FORWARD -i eth1 -s 165.21.0.0 /16 -j DROP

Outbound Anti spoofing

! Block outgoing traffic with foreign source IPs:
```
access-list 111 permit ip 165.21.0.0 0.0.255.255 any
access-list 111 deny ip any any log
```

iptables -A FORWARD -i eth0 -s ! 165.21.0.0 /16 -j DROP

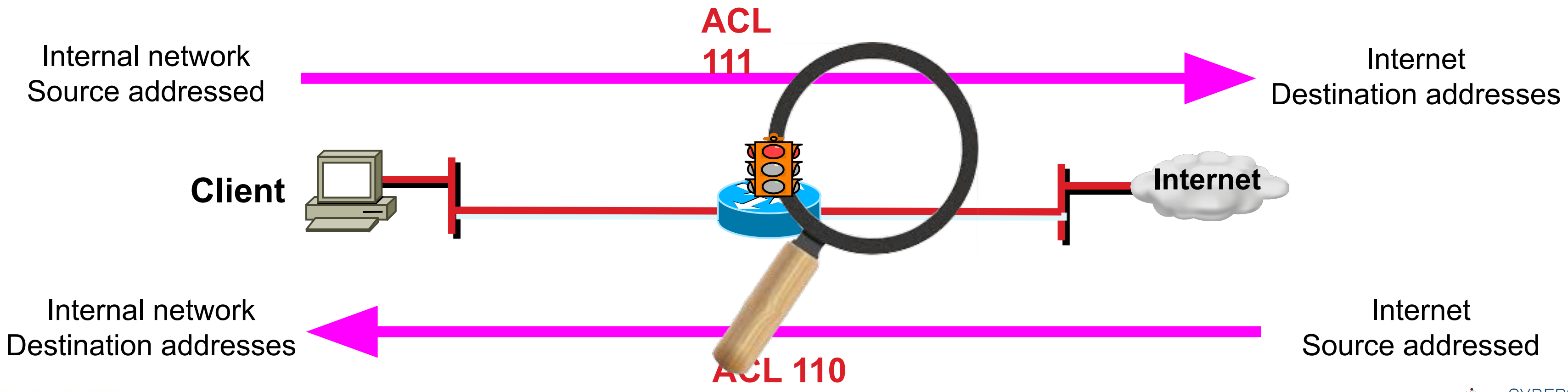
Definition of a simple access control policy

Example (in a SOHO scenario)

- Allow outgoing use of all TCP-based services (www, e-mail, etc.)
- Only allow incoming access to an extremely limited number of TCP services (e-mail, www) provided by a specific and controlled host
- Also allow incoming traffic related to sessions open from within (watch out for ftp!)
- Allow Ping and Traceroute from inside and not from outside
- Allow DNS through an internal server

Applying access controls

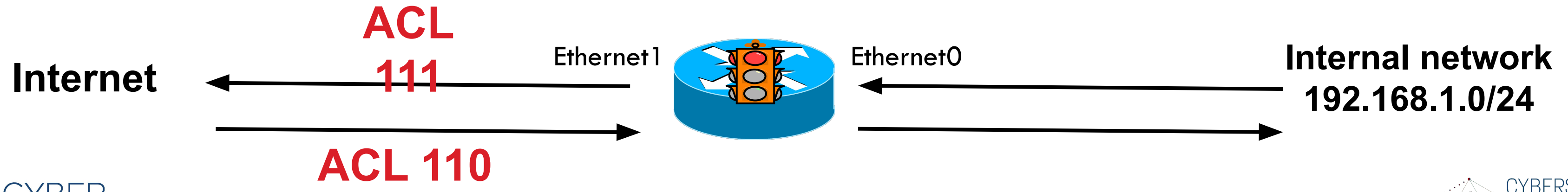
- ACL-based filtering can be implemented on any border device
- In our simple example, it is effective to act at the "border router" level, which separates the two distinct security domains (inside, outside) and on which it is possible to centrally control the traffic that flows between these domains.
- We only need 2 ACLs (110 and 111) to be applied at the input and output respectively



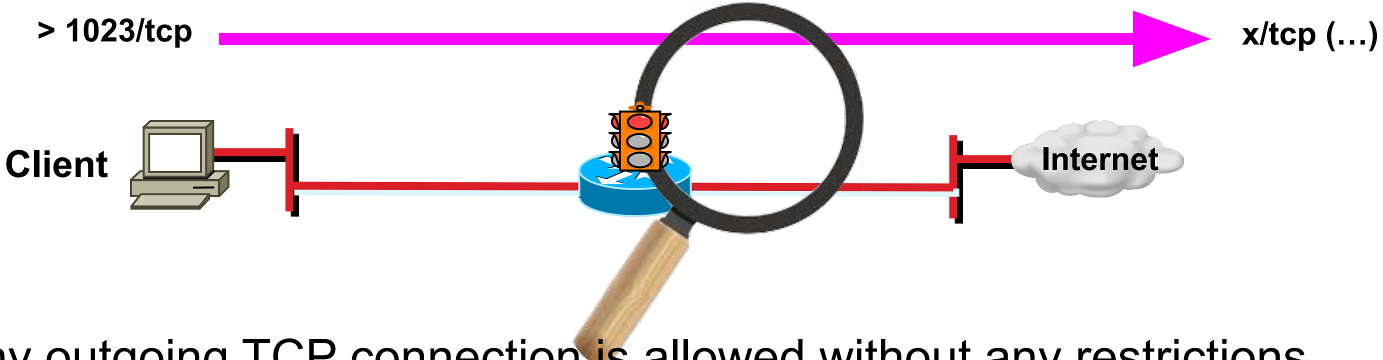
Applying access controls

```
interface ethernet 0
    ip access-group 110 in
    ip access-group 111 out
```

```
iptables -P FORWARD DROP
iptables -N acl110
iptables -N acl111

iptables -A FORWARD -i eth1 -o eth0 -j acl110
iptables -A FORWARD -i eth0 -o eth1 -j acl111
```



Outgoing traffic

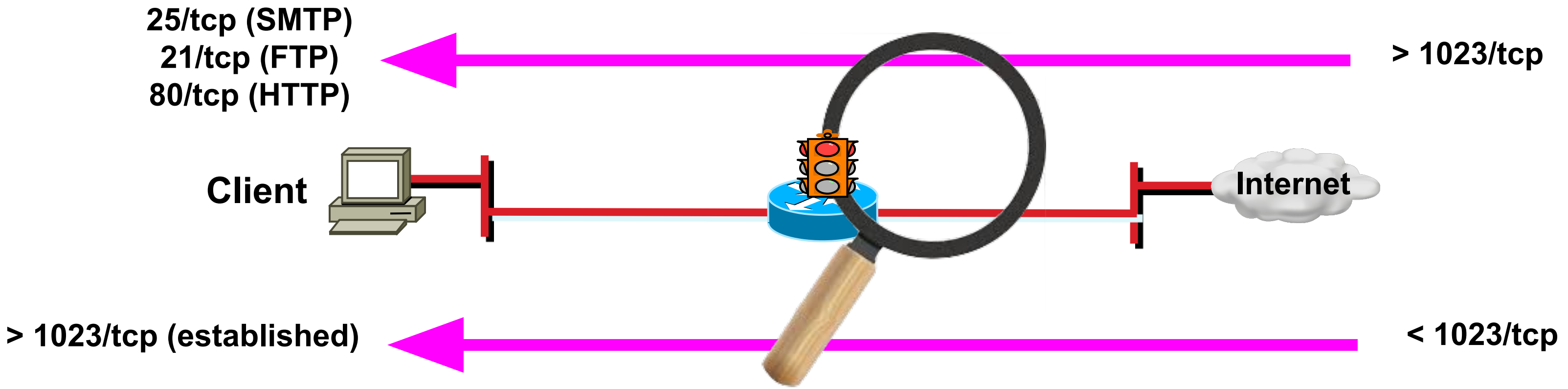


- Any outgoing TCP connection is allowed without any restrictions
- UDP and raw connections are implicitly blocked

```
access-list 111 permit tcp 192.168.1.0 0.255.255.255 any
```

```
iptables -A acl111 -p tcp -s 192.168.1.0/24 -j ACCEPT
```

Incoming traffic



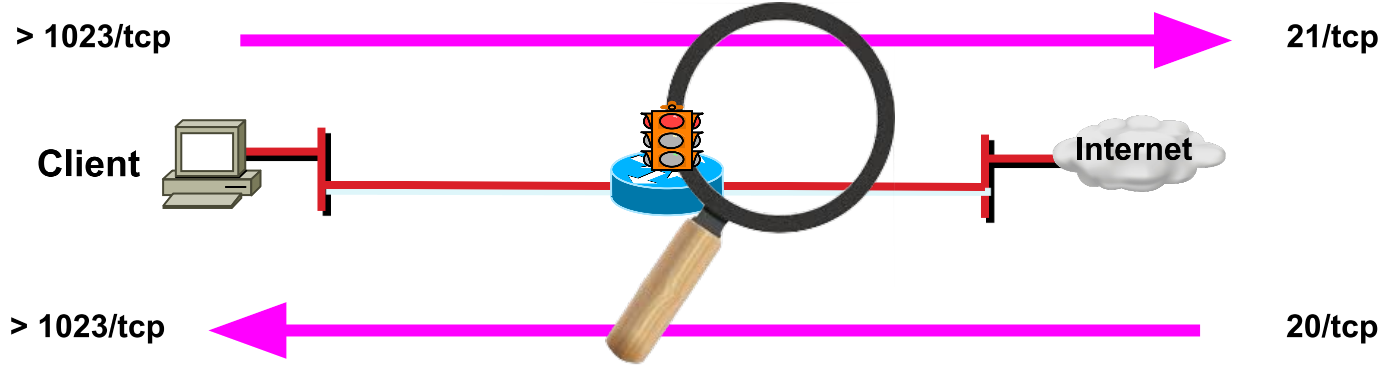
- Access to internal services must be controlled and allowed only to the hosts providing some services
- Backward incoming traffic (from outside to inside) must be allowed only if related to connections already open from inside (established)

Incoming traffic

```
! Block incoming spoofed addresses
access-list 110 deny ip 192.168.1.0 0.255.255.255
! Allow TCP sessions only toward internal service provider host
access-list 110 permit tcp any host 192.168.1.1 eq 25
access-list 110 permit tcp any host 192.168.1.1 eq 80
access-list 110 permit tcp any host 192.168.1.1 eq 80
! Only TCP return traffic of sessions open from within
access-list 110 permit tcp any 192.168.1.0 0.0.0.255 established
```

```
iptables -A acl110  -s 192.168.1.0/24 -j DROP
iptables -A acl110 -p tcp -d 192.168.1.1 --dport www -j ACCEPT
iptables -A acl110 -p tcp -d 192.168.1.1 --dport smtp -j ACCEPT
iptables -A acl110 -p tcp -d 192.168.1.1 --dport ftp -j ACCEPT
iptables -A acl110 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

FTP problems

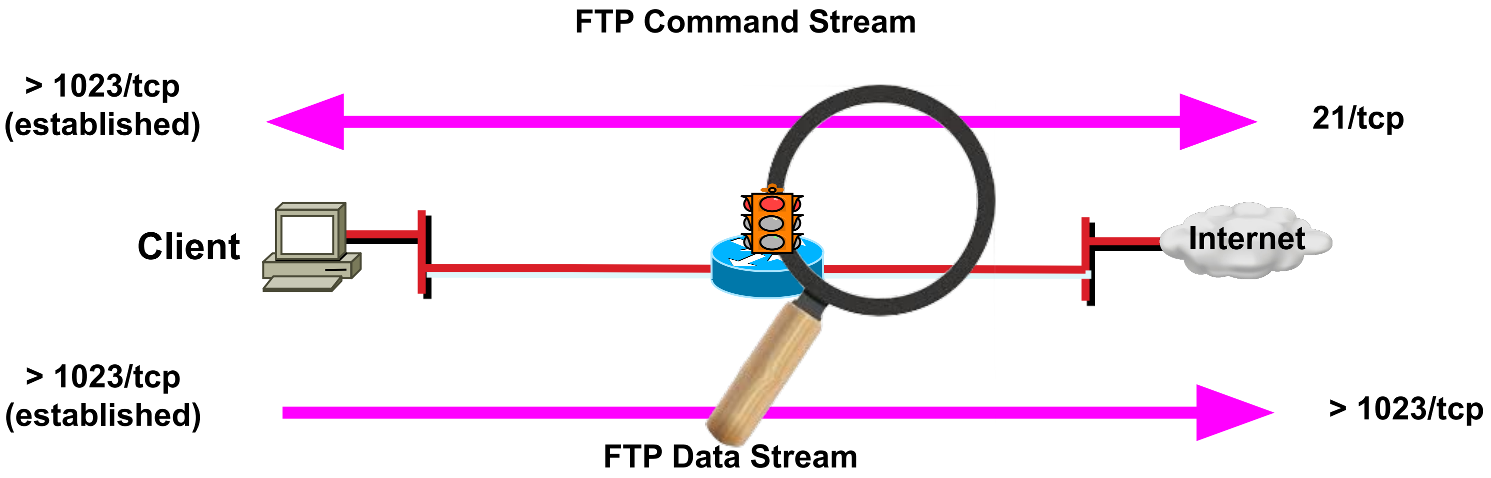


- After opening the connection to the control channel from the inside, it is necessary for each transfer to guarantee the possibility of opening the data connections backwards

```
access-list 110 permit tcp any eq 20 192.168.1.0 0.0.0.255 gt 1023
```

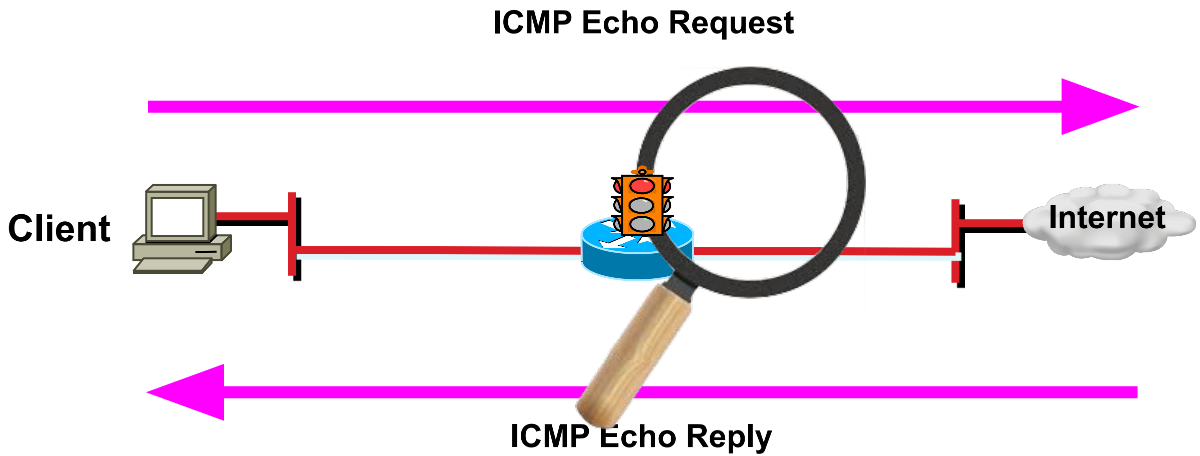
```
iptables -A acl110 -p tcp --sport 20 --match multiport --dports 1024:65535 -j ACCEPT
```


Passive mode FTP



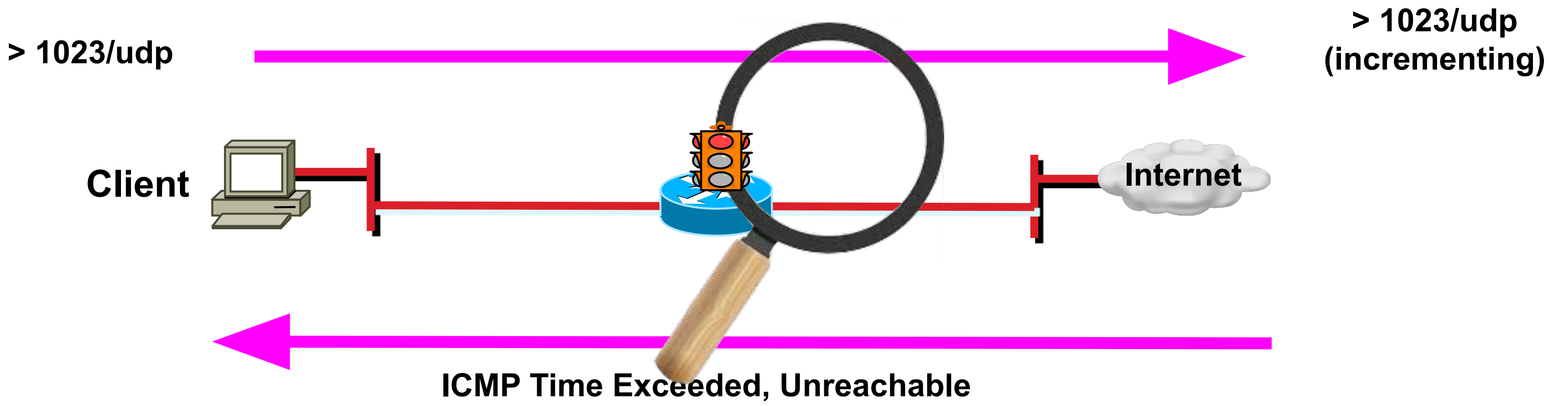
- Passive mode FTP eliminates the need to access port 20 backwards and solves the problem
- To prevent the problem you can use a firewall (it does "stateful" filtering)

Ping



- To ensure the functionality of the ping sessions started from within, allow ICMP echo reply messages backwards in response to echo request messages
- Ping initiated from the outside is inhibited

Traceroute



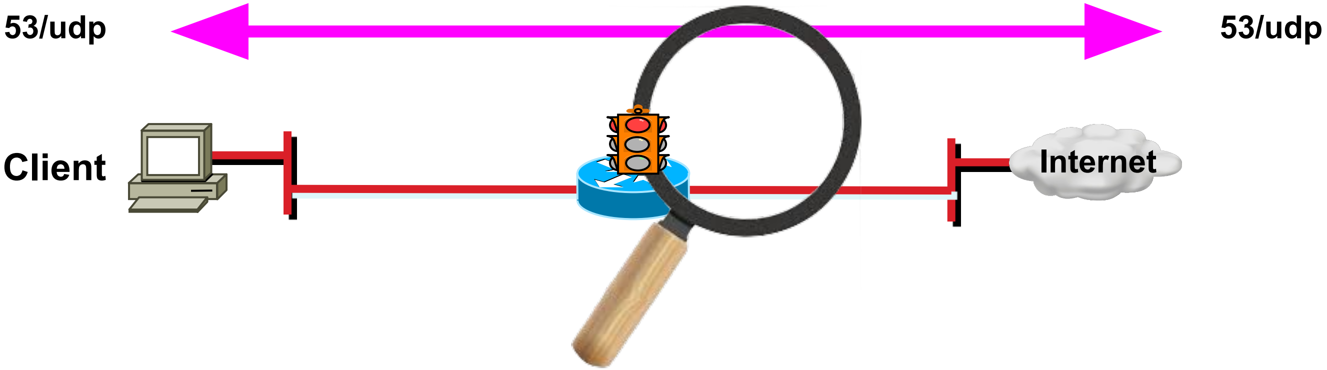
- To ensure the functionality of the traceroute started from the inside, ICMP time exceeded messages (intermediate steps) and ICMP port unreachable messages (final condition) must be allowed backwards
- Traceroute initiated from the outside is inhibited

Ping and Traceroute filters

```
access-list 110 permit icmp any 192.168.1.0 0.0.0.255  echo-reply
access-list 110 permit icmp any 192.168.1.0 0.0.0.255  time-exceeded
access-list 110 permit icmp any 192.168.1.0 0.0.0.255  unreachable

access-list 111 permit icmp 192.168.1.0 0.0.0.255 any echo
access-list 111 permit udp  192.168.1.0 0.0.0.255 gt 1023 any gt 1023
```

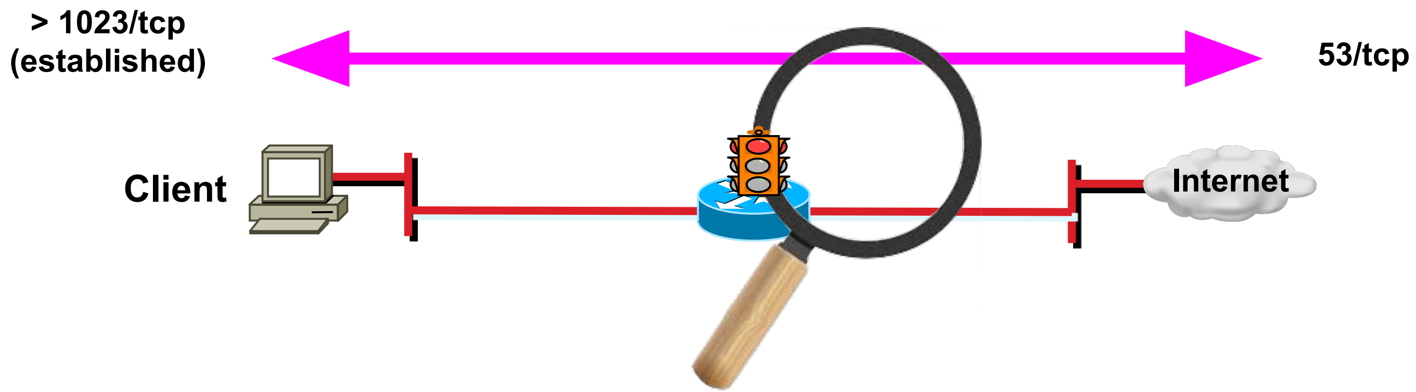
```
iptables -A acl110 -p icmp --icmp-type echo-reply -s 0/0 -d 192.168.1.0/24 -j ACCEPT
iptables -A acl110 -p icmp --icmp-type destination-unreachable -s 0/0 -d 192.168.1.0/24 -j ACCEPT
iptables -A acl110 -p icmp --icmp-type time-exceeded -s 0/0 -d 192.168.1.0/24 -j ACCEPT

iptables -A acl111 -p icmp --icmp-type echo-request -s 192.168.1.0/24 -j ACCEPT
iptables -A acl111 -p udp --match multiport --sports 1024:65535 --match multiport --dports
    1024:65535 -j ACCEPT
```

DNS: Queries



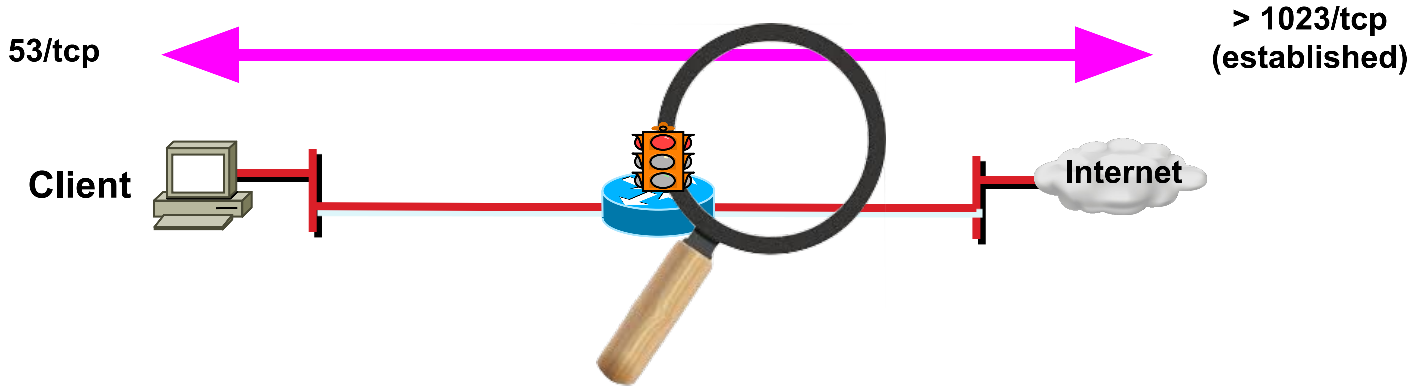
- DNS requests and responses must be guaranteed (UDP messages in both directions to port 53)

DNS: bulk responses



- ... the same goes for bulk replies on TCP sessions (only if started from within)
- In any case, external queries based on TCP / 53 must be blocked
- DNS queries must be admitted only to the DNS server

DNS: zone transfers



- … As well as in the opposite direction for zone transfers on TCP sessions
- Transfers should only be allowed to authorized hosts

DNS: filtering rules

```
! DNS Requests or responses  DNS to the internal server
access-list 110 permit udp any host 192.168.1.1 eq 53
! Zone transfers
access-list 110 permit tcp host 172.16.1.1 host 192.168.1.1 eq 53

! Server to server queries
access-list 111 permit udp host 192.168.1.1 eq 53 any eq 53
```

```
iptables -A acl110 -p udp -d 192.168.1.1 --dport domain -j ACCEPT
iptables -A acl110 -p tcp -s 172.16.1.1 -d 192.168.1.1 --dport domain -j ACCEPT

iptables -A acl111 -p udp --dport 53 -j ACCEPT
```

Francesco PALMIERI
Università di Salerno

Securing Internet communications

CYBER CHALLENGE.IT

CYBERSECURITY NATIONAL LABORATORY

https://cybersecnatlab.it