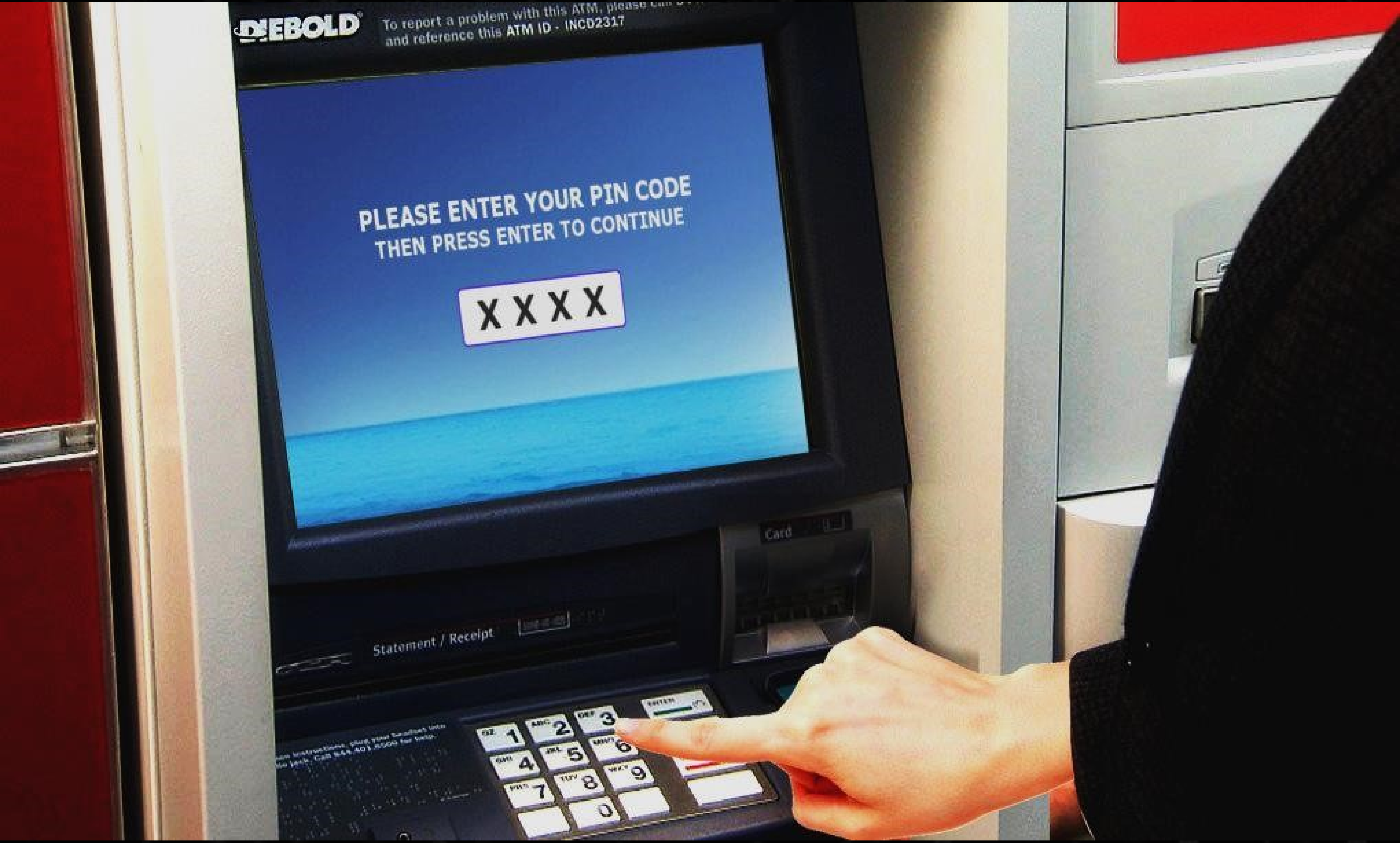
TSA

3
ESC
6
▲
9
▼
▶
M/OK


DIEBOLD
To report a problem with this ATM, please call
and reference this ATM ID - INCD2317
PLEASE ENTER YOUR PIN CODE
THEN PRESS ENTER TO CONTINUE
X X X X
Card
Statement / Receipt
ENTER
1
ABC 2
DEF 3
GHI 4
JKL 5
MNO 6
PRS 7
TUV 8
WXY 9
0

POLICE

CO 325: Computer and Network Security

Network Access Control

What is Access Control?

- System that enables an authority to control access
  - By *users* (subjects)
  - To *system resources* (objects)
  - Based on a *security policy* (access control matrix)
- Examples
  - Lock on a car door or file cabinet
  - Guest list for entrance to an event
  - PIN on an ATM cash machine
  - Password for logging in to a computer account
  - Access control list applied at a firewall

Access Control in Computer Security

- Authentication
  - Confirming *identity* of the subject
  - Based on what you *know* (e.g., PIN, password), what you *have* (e.g., smart card), what you *are* (e.g., iris, fingerprint, voice), or *where* you are (e.g., inside firewall)
- Authorization
  - Determining *what* the subject can do
  - E.g., read/write/execute, or accept/deny
- Accountability
  - Associating a subject with its actions
  - To detect and/or recreate security violations
  - E.g., audit trails of failed login attempts or blocked traffic

Access Control Matrix

- Representation of access control policy
  - *Columns*: objects (e.g., file, directory, printer, link)
  - *Rows*: subjects (e.g., user, process, threads)
  - *Entry*: set of access operations
- A request (*o*, *s*, *a*) is granted if access operation *a* belongs to the entry for subject *s* and object *o*

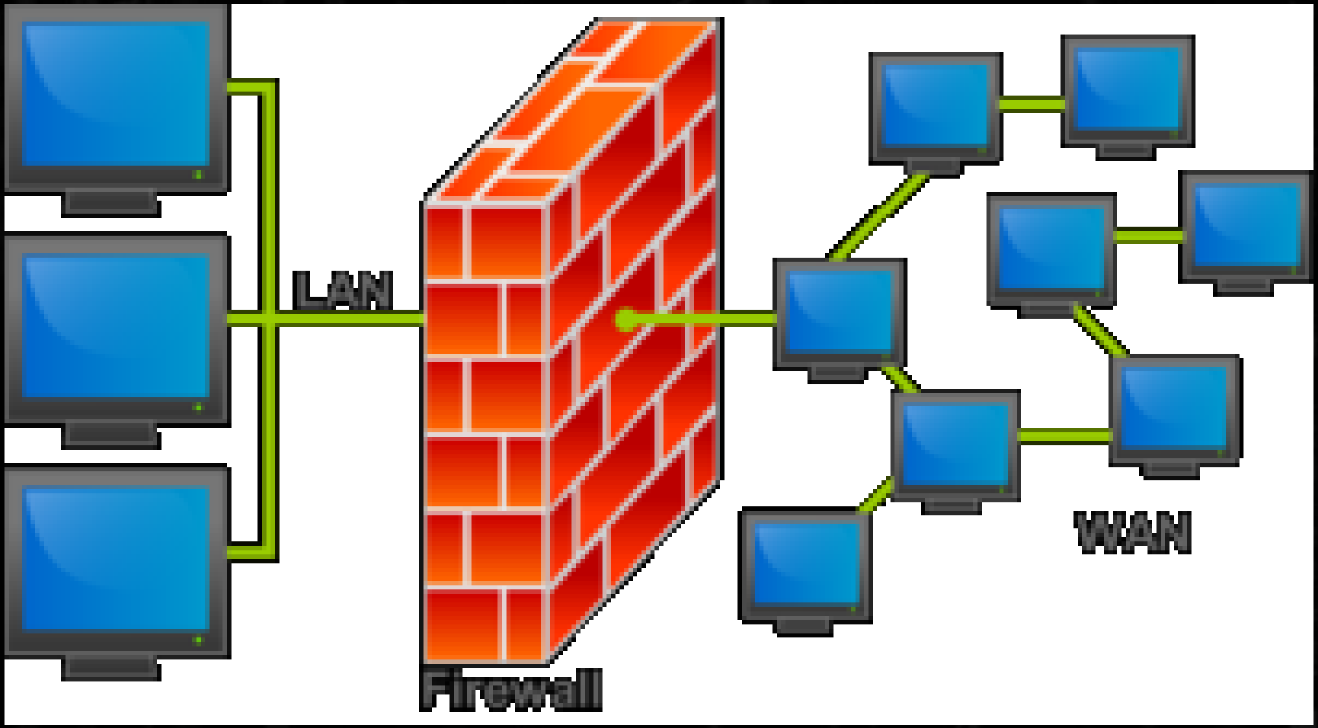
Objects **Subjects**	trash	a.out	allfiles.txt
e11511	{r,w}	{r,w,x}	{r,w}
e12451		{r,x}	{r}

Two Implementation Approaches

- Access-control list
  - Focuses on the *object* (i.e., a column in the matrix)
  - Analogous to a "guest list" of the invited guests
- Capability list
  - Focuses on the *subject* (i.e., a row in the matrix)
  - Analogous to issuing keys for unlocking a file cabinet

Objects **Subjects**	trash	a.out	allfiles.txt
e11511	{r,w}	{r,w,x}	{r,w}
e12451		{r,x}	{r}

Access Control Lists: Firewalls



Stateless Packet Filters (1st Gen)

- Filter based on information contained in the packet
  - E.g., IP addresses, protocol, port numbers, …
  - Well-known TCP/UDP ports for applications
- Access Control List: <pattern, permit/deny> rules
  - Process rules in order till encountering a match
  - Analogous to the if-elseif-else programming construct

Src=1.2.3.4, Dest=5.6.7.8	Deny
Dest=1.2.3.*	Allow
Dest=1.2.3.8, Dport!=53	Deny
Src=1.2.3.7, Dport=100	Allow
Dport=100	Deny

Stateful Filters (2^{nd} Gen)

- Maintains state for each ongoing connection
  - IP address, port numbers, sequence numbers, …
  - And times out after a period of inactivity
- Avoids repeating lengthy rule processing
  - CPU-intensive check only for the first packet
  - Cache of the result for the remaining packets
- Allows policies based on state of the connection
  - E.g., only allow incoming packets for established connections (to prevent unsolicited connections)

Application-Level Filters (3rd Gen)

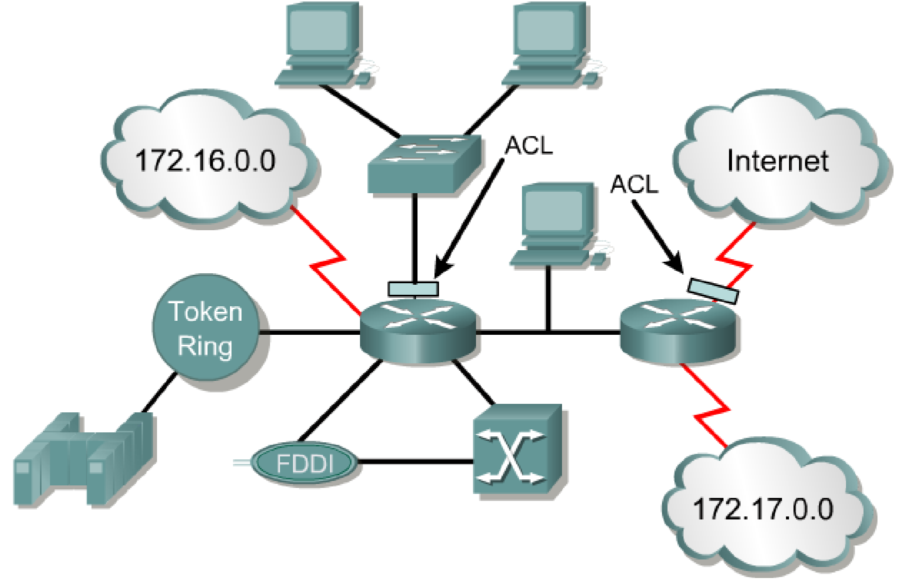
- Proxies traffic before forwarding to client or server
  - For particular applications
  - E.g., Web server, database server, …
- Understands the applications
  - Parses message contents
  - E.g., URL, domain name, SQL query, Google search, …
- Enabling richer access-control policies
  - Preventing SQL injection attacks
  - Blocking access to certain sites or URLs
  - Blocking searches on particular search terms
  - Blocking BitTorrent on non-well-known ports

Where Do Firewalls Run?

- On the end host
  - Can include information from application system calls
  - Scalability and customization with a firewall per host
- Just in front of the host(s)
  - Share the firewall between a group of related hosts
  - Reduces traffic and CPU load on the end hosts
- At the gateway to the Internet
  - Share the firewall across an entire organization
  - Avoid wasting resources inside the organization
- On the router itself
  - Avoid the cost of buying and supporting another box

A HIGH LEVEL VIEW of ACLs

What are ACLs?



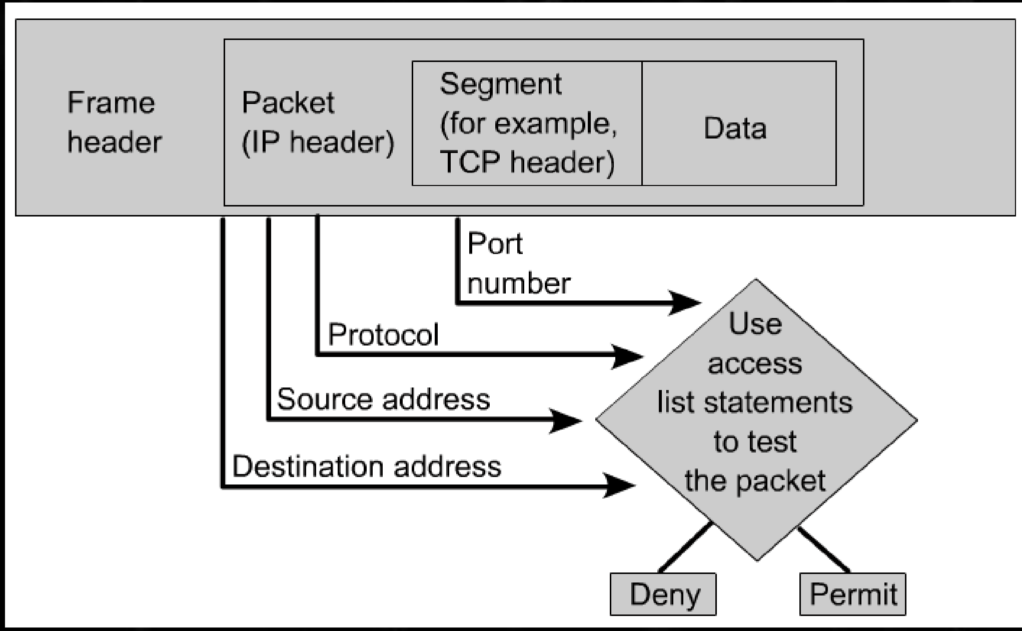
- An access list is a sequential series of commands or filters.
- These lists tell the router what types of packets to:
  - accept or
  - deny
- Acceptance and denial can be based on specified conditions.
- ACLs applied on the router or firewall's interfaces.

Reasons to create ACLs

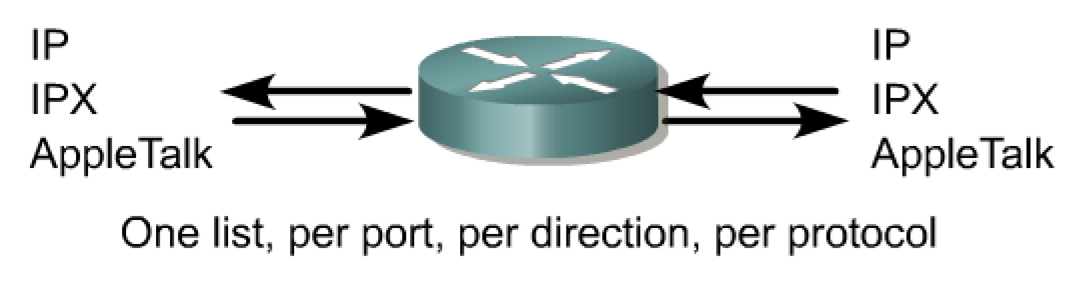
- **Limit network traffic:** hence increase network performance
- **Provide traffic flow:** limit traffic through the network
- **Provide security**
- ACLs establish
  - which traffic is blocked
  - which traffic is not blocked

What are ACLs?

- The router/firewall examines each packet to determine whether to forward or drop it, based on the conditions specified in the ACL.
- *Some* ACL decision points are:
  - IP source address
  - IP destination addresses
  - UDP or TCP protocols
  - upper-layer (TCP/UDP) port numbers



What are ACLs?

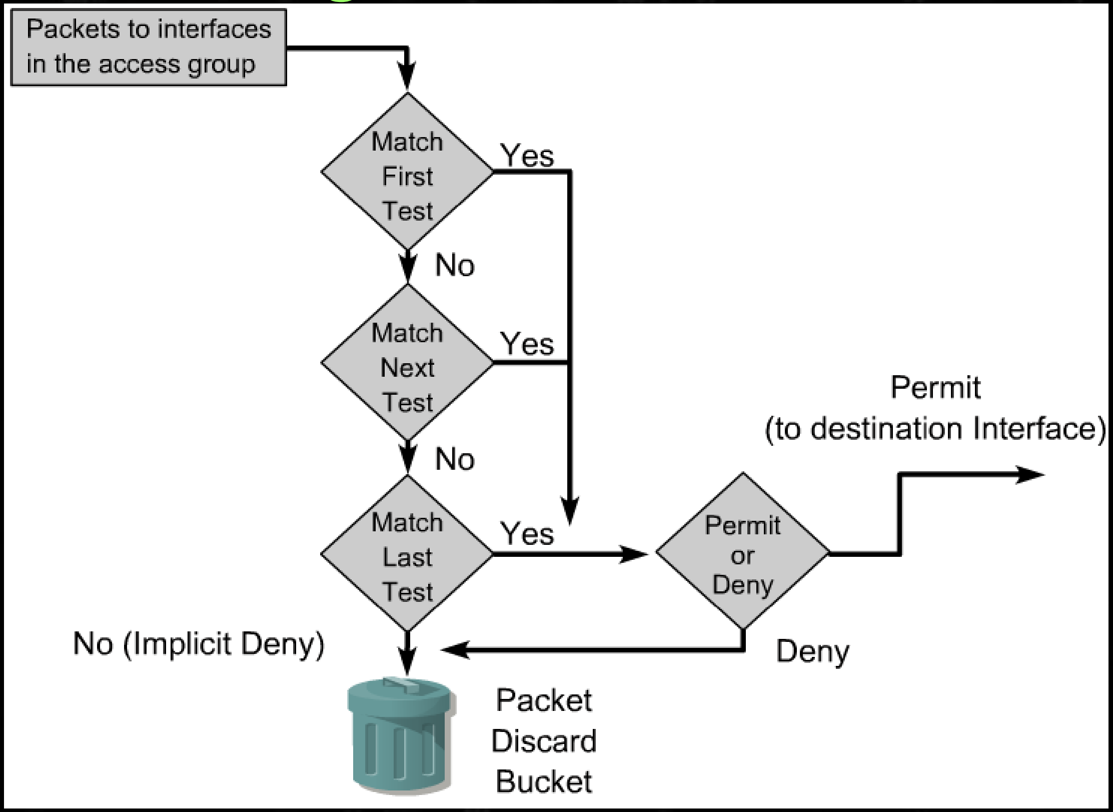


One list, per port, per direction, per protocol

- ACLs must be defined on a:
  - per-protocol (IP, IPX, AppleTalk)
  - per direction (in or out)
  - per port (interface) basis.
- ACLs control traffic in one direction at a time on an interface.
- A separate ACL would need to be created for each direction, one for inbound and one for outbound traffic.
- Finally every interface can have multiple protocols and directions defined.

How ACLs work

- An ACL is a group of statements that define whether packets are accepted or rejected coming into an interface or leaving an interface.



How ACLs work

- **ACLs** operate in sequential, logical order.
- They evaluate packets from the **top down.**
- Once there is an access list statement **match**, the packet skips the rest of the statements.
  - If a condition **match is true**, the packet is *permitted* or *denied.*
- There can be **only one access list** per protocol per interface.
- There's an implicit "deny any" at the end of every ACL
- When first learning how to create ACLs, it is a good idea to add the **implicit deny** at the end of ACLs to reinforce the dynamic presence of the command line..

Two types of ACLs

- Standard IP ACLs
  - Can only filter on source IP addresses
- Extended IP ACLs
  - Can filter on:
    - Source IP address
    - Destination IP address
    - Protocol (TCP, UDP)
    - Port Numbers (Telnet – 23, http – 80, etc.)
    - *and other parameters*

Creating ACLs – 2 Steps

1. Define an ACL

```
ciscoasa(config)# access-list any2host
extended permit ip any host 192.168.100.10
```

2. Apply the ACL to an Interface

```
ciscoasa(config)# access-group any2host in
interface outside
```

From Cisco Web Site

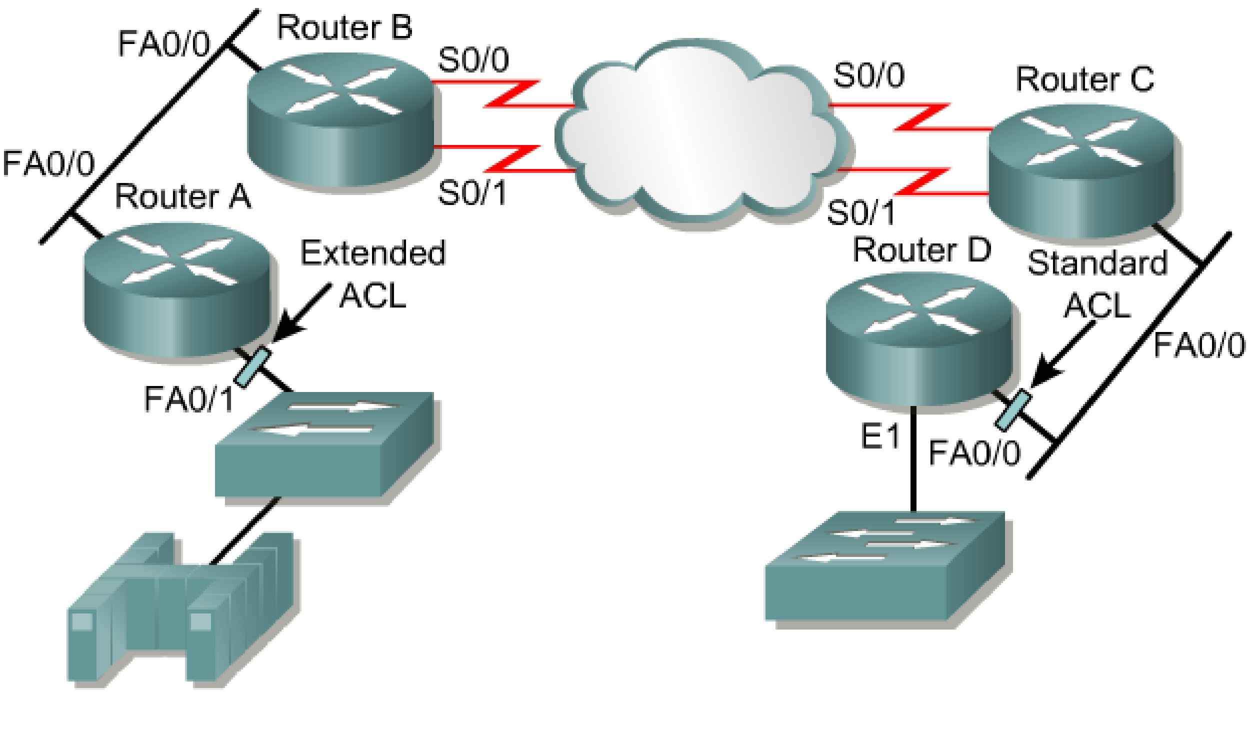
Applying ACLs

- You can define ACLs without applying them.
- However, the ACLs will have no effect until they are applied to the firewall or router's interface.
- It is a good practice to apply the Standard ACLs on the interface closest to the destination of the traffic and Extended ACLs on the interface closest to the source. (coming later!)

Defining In, Out

- **Out** - Traffic that has already been routed by the firewall/router and is leaving the interface
- **In** - Traffic that is arriving on the interface and which will be routed by the firewall/router.

Placing an ACL



- Standard ACLs should be placed close to the destination
- Extended ACLs should be placed close to the source

Intra-Firewall Configuration Errors

- Shadowing (error)
  - All packets matching a rule would match an earlier rule
  - E.g.: rule 4 is shadowed by rule 2
- Generalization (warning)
  - A more-specific earlier rule takes a different action
  - E.g.: rule 7 is a generalization of rule 4
- Correlation (warning)
  - Partially overlapping rules with different actions
  - E.g.: rules 2 and 6

```
1. deny tcp 10.1.1.0/25 any
2. accept udp any 192.168.1.0/24
3. deny tcp 10.1.1.128/25 any
4. deny udp 172.16.1.0/24 192.168.1.0/24
5. accept tcp 10.1.1.0/24 any
6. deny udp 10.1.1.0/24 192.168.0.0/16
7. accept udp 172.16.1.0/24 any
```

Intra-Firewall Inefficiencies

- Verbosity
  - A set of rules can be summarized with fewer rules
  - E.g., rules 5-8 with “deny udp 10.1.1.0/24 any”
- Redundancy
  - Removing a rule does not change any actions
  - E.g., rule 3 is redundant with rule 2
  - E.g., rules 5, 6, 7, and 8 are redundant with rule 9

```
1. accept tcp 192.168.1.1/32 172.16.1.1/32
2. accept tcp 10.0.0.0/8 any
3. accept tcp 10.2.1.0/24 any
4. deny tcp any any
5. deny udp 10.1.1.0/26 any
6. deny udp 10.1.1.64/26 any
7. deny udp 10.1.1.128/26 any
8. deny udp 10.1.1.192/26 any
9. deny udp any
```

Inter-Firewall Configuration Errors

- Enterprises often have multiple firewalls
  - E.g., on most/all of the end hosts
  - E.g., on many network interfaces
  - E.g., on multiple firewalls inside the network
- Together, they should realize a single policy
  - One high-level policy, with a distributed realization
- Challenges
  - Consistent configuration of the many firewalls
  - Ensuring the system works even when routing changes

Beyond Today's Low-Level Policies

- Expressed in terms of network identifiers
  - E.g., MAC and IP addresses, port numbers, …
  - Should express policies based on *names*
- Doesn't capture changes in user's status
  - E.g., machines becomes infected…
  - These changes should affect access control
- Puts policy in the wrong place (i.e., the network)
  - End-host is a better place to enforce policy
  - Network should only stop denial-of-service attacks

The Network is the Wrong Place?

- Network-based access control does not scale
  - Especially as link speeds continue to grow
- Volatility of network identifiers
  - End-host identifiers change as hosts move
  - Forcing the network to track a lot of churn
- Poor visibility into application information
  - Forcing a reliance on TCP/UDP port numbers
  - Requiring (expensive) deep-packet inspection
- Limited sphere of influence
  - Remote users are the norm, not an exception
  - Often forced through special firewalls, VPNs, …

Alternate Approach: Capabilities

- Who should be in charge?
  - Destination knows which traffic is legitimate
  - Network can shed load before it is excessive
- Division of labor
  - Network filtering based on destination control
  - Explicit authorization that the network can check
  - Packets carrying “capabilities”
- Solution
  - Tokens: short-lived capabilities carried on packets
  - Servers: grant tokens based on policies
  - Routers: filter packets lacking the right token

Discussion

- Where should access control be performed?
  - Host OS, hypervisor switch, network elements, …
- What should be the "subject" of access control?
  - The location of the computer
  - The computer
  - The person using the computer
- Access control vs. capabilities?
- What about inter-domain access control?
  - Use capabilities?
  - Coordinate ACLs across domains?