Access Control

Key concepts:

- Controlling the data flow within a network
- ACL (access control lists)

ACL

- A list of *permit* and *deny* statements
- Control network access through a given device (router, firewall, etc.)
- Main function: traffic *filtering*
- **Q:** What security services are provided by ACL?

- Enforce a security policy (partially)
- Need to be supplemented by other security mechanisms (cryptography, authentication, intrusion detection and prevention, etc.)
- Example: Would ACL alone enforce *confidentiality*?

ACL Applications

- ACLs tend to be used as the first line of defense for the network.
- Examples: ACLs on routers, switches, firewalls …

- Sample applications:
  - Filter routing information between routers
  - Secure/limit access (tty ports, Telnet, …) to devices (switches, routers, …)
  - Limit types of traffic to a device
  - Filter traffic through devices
  - etc.

Types of ACLs

- Standard ACLs
- Extended ACLs
- IP named ACLs
- Lock and key (Dynamic) ACLs
- Reflexive ACLs
- Established ACLs
- Time-based ACLs
- Distributed time-based ACLs

Subnet mask vs Inverse mask

- The **subnet mask** is used to *split* an IP address into the network address and the host address.
- The router uses the network id in the destination IP address to perform routing.
- Example:
  - IP address = 10.1.1.0, Subnet mask = 255.0.0.0
  - Network id = IP **AND** subnet mask = 10.0.0.0
  - Alternatively, 10.1.1.0***/8*** (The first 8 bits represent the network id.)

Subnet mask vs Inverse mask

- The **inverse mask** (aka the **wildcard mask**) is used in Cisco's IOS IP ACLs.

  Inverse mask = 255.255.255.255 – subnet mask

- Example:
  - IP address = 10.1.1.0, Subnet mask = 255.0.0.0
  - Inverse mask = 0.255.255.255 (1's bits means "don't care" when determining network id.)
  - Network id = 10.0.0.0
  - Alternatively, 10.1.1.0***/8*** (The first 8 bits represent the network id.)

Standard ACLs

- Inspects traffic by comparing the *source* address of the IP packets to the addresses configured in the ACL
- Sample commands of defining an ACL:

  Router(config)# access-list 1 permit 10.1.1.0 0.0.0.255

- Commands of applying an ACL to an interface:

  Router(config)# interface Serial0

  Router(config-if)# ip access-group 1 in

- All incoming traffic on the serial0 interface, except that from source 10.1.1.0/24, are blocked.

Extended ACLs

- To filter more-specific traffic based on the source address, the destination address, and specific protocols, ports, and flags
- Sample commands of defining an extended ACL:

  Router(config)# access-list 101 permit tcp any host 172.16.1.1 eq smtp
- Commands of applying an ACL to an interface:

  Router(config)# interface Serial0

  Router(config-if)# ip access-group 101 in
- Incoming SMTP traffic to host 172.16.1.1 are permitted on the Serial0 interface. (*Implicit deny*: If there's no other permit statement defined, all other traffic are denied.)

IP Named ACLs

- A standard or extended ACL may be given a name (instead of a number).
- Sample commands of defining an IP named ACL:

```
Router(config)# ip access-list standard myacl
                permit 192.16.1.0 0.0.0.255
                permit host 172.65.1.1
```

- Defines a named ACL *myacl that a*llows all traffic sourced from network 192.16.1.0/24 and host 172.65.1.1

Dynamic ACLs

- Aka the *Lock and Key* ACLs
- Allows the admin to set up a dynamic access that will allow *per-user* access control to a particular source/destination using an authentication mechanism
  - The ACL entry is only enabled for a specified period of time (*timeout* value).
  - Relies on either local authentication or server-based authentication (a TACACS+ or Radius server)
  - Example: A vty line on a router may be set up to allow the admin to telnet from a particular host to the router.

Reflexive ACLs

- Similar to the *Context Based Access Control (CBAC)*, more in ch. 5
- To restrict inbound traffic to those originated inside the router (*session control*)
- Must be defined as an extended named IP ACL
  - When defining the outbound ACL, use ‘reflect tcp_reflect’.
  - When defining the inbound ACL, use ‘evaluate tcp_reflect’.

Established ACLs

- The *established* keyword is added to a TCP extended ACL.
- The 3-way handshaking in TCP:

  SYN + SYN/ACK + ACK

- The router validates that a TCP packet belongs to an existing connection from an ongoing TCP session initiated earlier (by checking whether the packet has the ACK or RST bit set).
- Any TCP packet with an ACK/RST bit not set will be dropped.

Time-Based ACLs

- Access control based on time range (using the *time-range* keyword when defining an ACL)
  - Only works with IP and IPX numbered or named extended ACLs
  - Relies on the router's clock or synchronized with a NTP server
  - A time range may be *absolute* or *periodic*.
- Example: all IP traffic is being permitted through the network on weekdays during normal business hours.

Distributed Time-Based ACLs

- "**Distributed switching** is an architecture in which multiple processor-controlled switching units are distributed. There is often a hierarchy of switching elements, with a centralized *host switch* and with *remote switches* located close to concentrations of users."
  - source: http://en.wikipedia.org/wiki/Distributed_switching
  - The remote switches are close to the users and handles most of the switching tasks between the local users.
  - Therefore most traffic do not need to be forwarded to the host switch, which only handles complex tasks (e.g., conference calls) or calls involving non-local users.

Distributed Time-Based ACLs

- Distributed Time-Based ACL is a feature automatically enabled when the normal time-range ACL is configured on an interface in an upper-end router (e.g., the Cisco 7500 series).
- No commands are needed.
- Allows packets destined for an interface that is configured with time-based ACLs to be “distributed-switched” through that interface.

Turbo ACLs

- Compiles the ACLs into a set of lookup tables
- Packet headers are used to access these tables in a small, fixed number of lookups, independently of the existing number of ACL entries.
- Hash tables ?
- Beneficial when the number of entries grow larger in the ACLs.

rACLs

- *Receive* ACLs
- *Source:* *http://www.cisco.com/en/US/docs/ios/12_0s/feature/guide/ft_ipacl.html#wp1040112*
  - "You can configure an ACL that you have created to filter IPv4 or IPv6 traffic to process receive IP packets and reduce the CPU load on the route processor of unwanted traffic."
  - "In this way, you mitigate the adverse effects of **denial-of-service** attacks against the router."
  - "On a distributed platform, such as the Cisco 12000 series, the **IP receive ACL** filters traffic on the distributed line cards before IP packets are punted to the route processor."

rACLs

- Source: http://www.cisco.com/en/US/docs/solutions/Enterprise/Security/Baseline_Security/securbase.pdf
  - "a rACL is an access control list that controls the traffic sent by the various line cards to the RP on distributed architectures like the Cisco 1200 Series Routers."
  - "When configured, rACLs are first created on the RP, and then pushed to the line card CPUs. When packets enter the line cards, the packets are first sent to the line card CPU. Packets requiring processing by the RP are compared against the rACL before being sent to the RP."
  - "It should be noted that rACLs apply to traffic destined to the RP only, and does not affect transit traffic."

iACL

- Infrastructure Protection ACLs
- A concept/technique, not a feature
- Source: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801a1a55.shtml
  - "Data received by a router can be divided into two broad categories:
    - traffic that passes through the router via the forwarding path
    - traffic destined for the router via the receive path for route processor handling"
  - "In an effort to protect routers from various risks—both accidental and malicious—infrastructure protection ACLs should be deployed at network ingress points."
  - "These IPv4 and IPv6 ACLs deny access from external sources to all infrastructure addresses, such as router interfaces."
  - "At the same time, the ACLs permit routine transit traffic to flow uninterrupted and provide basic RFC 1918 , RFC 3330 , and anti-spoof filtering."

iACL

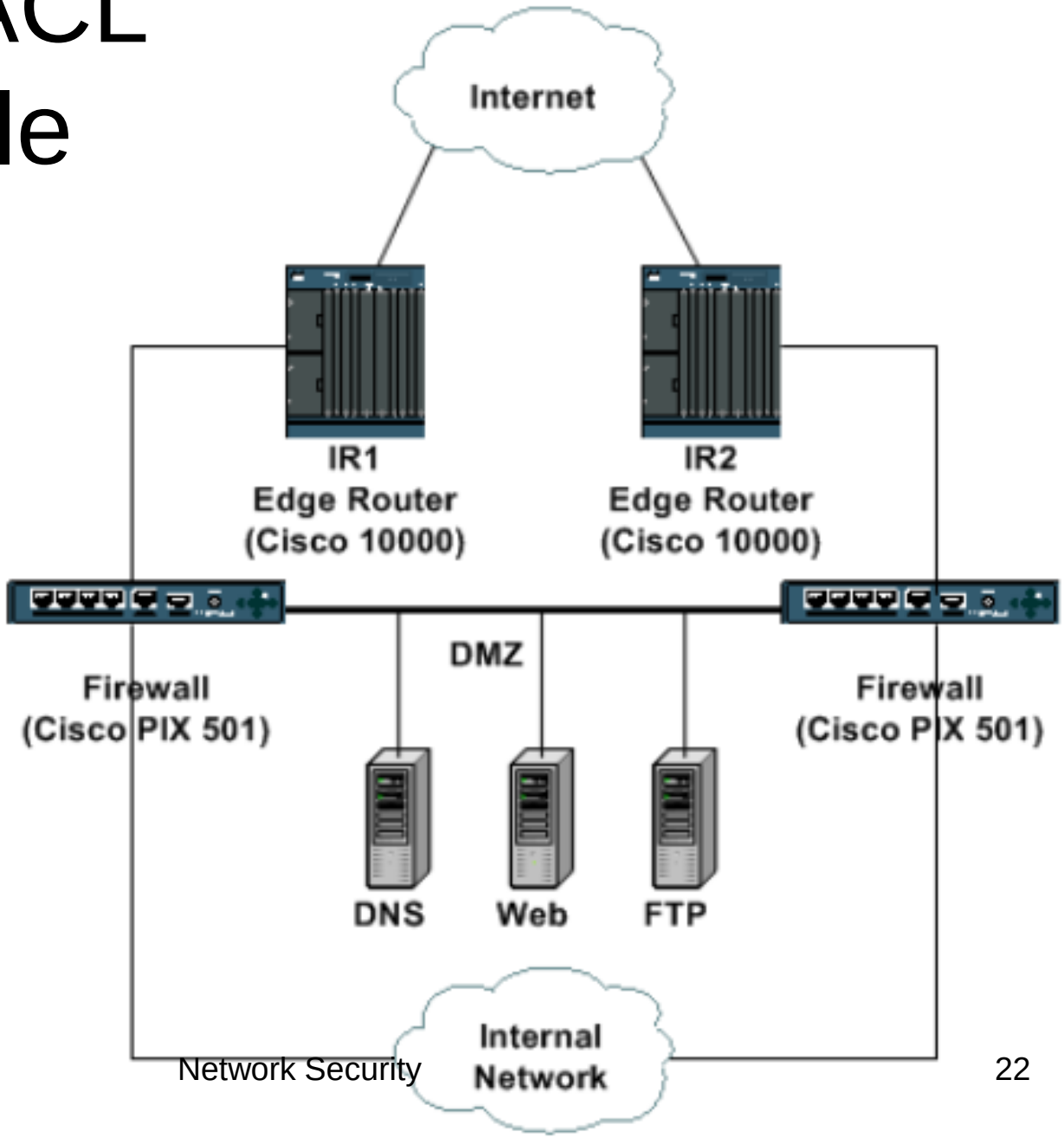
- <cont.>
  - “The route processor (RP) must handle certain types of data directly, most notably routing protocols, remote router access (such as Secure Shell [SSH]), and network management traffic such as Simple Network Management Protocol (SNMP).”
  - “In addition, protocols such as Internet Control Message Protocol (ICMP) and IP options can require direct processing by the RP.”
  - “All RPs have a performance envelope in which they operate. Excessive traffic destined for the RP can overwhelm the router. This causes high CPU usage and ultimately results in packet and routing protocol drops that cause a **denial of service**.”
  - “By filtering access to infrastructure routers from external sources, many of the external risks associated with a direct router attack are mitigated. Externally sourced attacks can no longer access infrastructure equipment. The attack is dropped on ingress interfaces into the autonomous system (AS).”

Transit ACL

- Also a concept/technique
- One of the many ways to increase network security by explicitly allowing legitimate traffic into the network
- Filtering is applied to control inbound traffic into the network and to block any unauthorized attempt at the edge of the network

Transit ACL example

- Source: http://www.cisco.com/en/US/tech/tk648/tk361/technologies_white_paper09186a00801afc76.shtml#intro
  - "Two edge routers, IR1 and IR2, provide direct connectivity to the Internet."



Transit ACL example

- <Cont.>
  - “Behind these two routers, a pair of firewalls (Cisco PIXes in this example) provides stateful inspection capabilities and access to both the internal network and the demilitarized zone (DMZ).”
  - “The DMZ contains public-facing services such as DNS and web; this is the only network accessible directly from the public Internet.”
  - “The internal network should never be accessed directly by the Internet, but traffic sourced from the internal network must be able to reach Internet sites.”

Transit ACL

- In general, a transit ACL is composed of four sections.
  1) Special-use address and anti-spoofing entries that deny illegitimate sources and packets with source addresses that belong within your network from entering the network from an external source

     Note: RFC 1918 defines reserved address space that is not a valid source address on the Internet. RFC 3330 defines special-use addresses that might require filtering. RFC 2827 provides anti-spoofing guidelines.
  2) Explicitly permitted return traffic for internal connections to the Internet
  3) Explicitly permitted externally sourced traffic destined to protected internal addresses
  4) Explicit **deny** statement

Classification ACL

- aka *characterization ACLs*
- Used as a diagnosis technique to identify potential source of a DoS attack
- Composed of all **permit** statements for the various protocols, ports, flags, etc. that could be sent to any of these three destinations: an infrastructure device, a public server in the protected zone, or any other device in the network

Debugging Traffic using ACLs

- Enable **debug** mode tend to consume lots of or all system resources
- Enable debug on an ACL restrict the debugging to be applied only to certain traffic
- Thus reduce the system overhead when debug is on