ACLs – Access Control Lists

ACL Terminology

- **ACL**: list of permit/deny statements (similar to a firewall)
- **Established connection**: TCP only; refers to preexisting connections
- **Operator**: keyword used to filter port numbers
- **Protocol**: standard method of transmitting and processing various kinds of information
- **Statement**: a line of an ACL

Wildcard Masks

- Calculated as an inverted subnet mask
- Bits do not need to be continuously 1 or 0 (i.e., 0.0.255.0 is a valid wildcard mask)
- 0s mean the corresponding bit in the IP address mu match
- 0.0.0.0 indicates the IP must match exactly
- 255.255.255.255 indicates any IP is acceptable
- 0.0.255.0 indicates only the 1st two octets and the last octet must match
- Wildcard masks do not have to be contiguous
- EX: 0.255.0.255 does not have a subnet equivalent

How an ACL Works

- Drops packets that do not match specific statements
- ORDER MATTERS! The router goes through the statements sequentially until it finds a match; if none are found, the packet is dropped
- Multiple lists can be created on a router
- Lists must be applied to an interface/line to take effect
- Uses wildcard masks

Standard vs. Extended ACLs

Standard

- Only filters based on source addresses
- Applied close to destination port
- Uses numerical identifiers 1-99

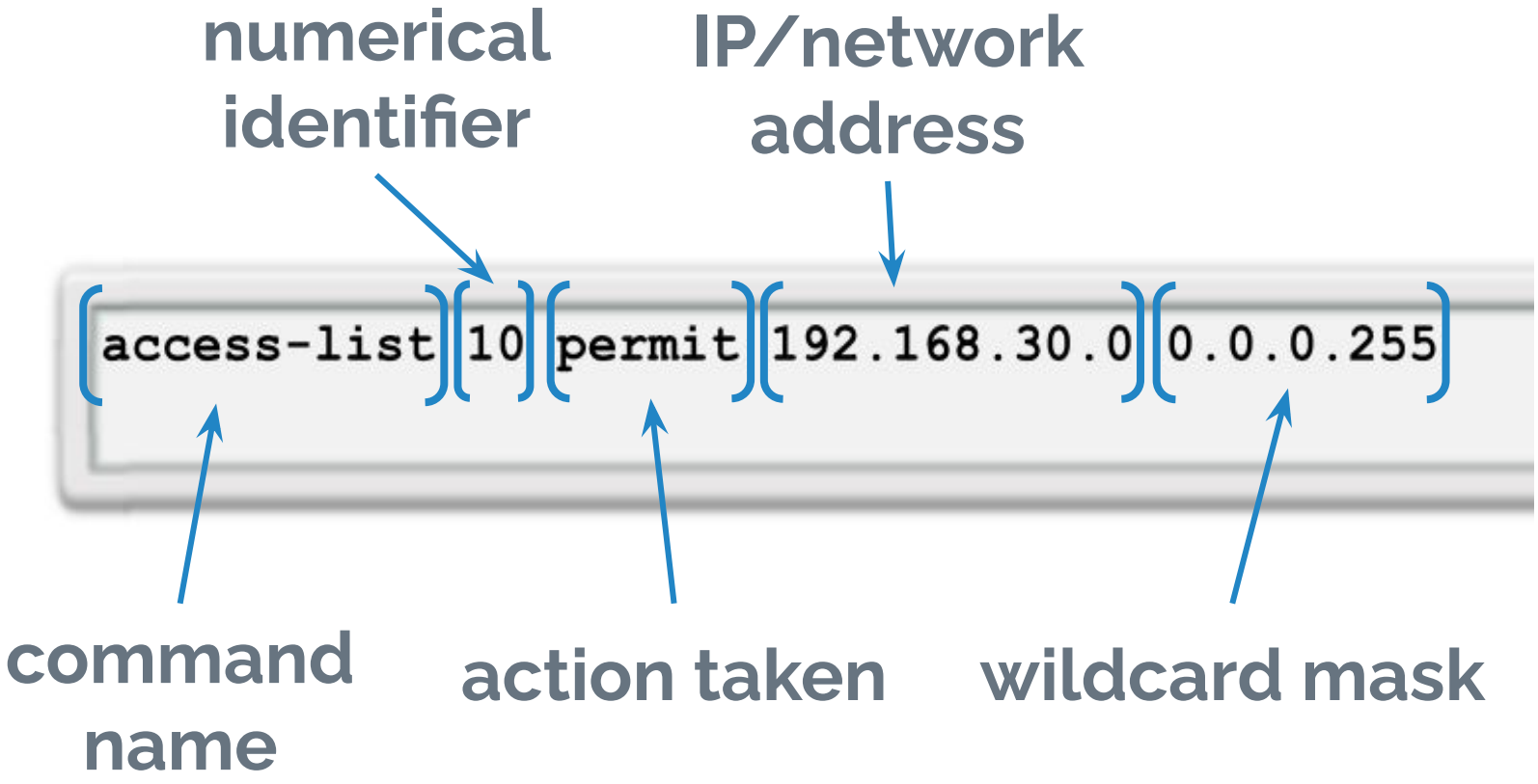
Extended

- Filters based on source and destination address, protocol types, port numbers, and/or established connections
- Applied close to source port
- Uses numerical identifiers 100-199

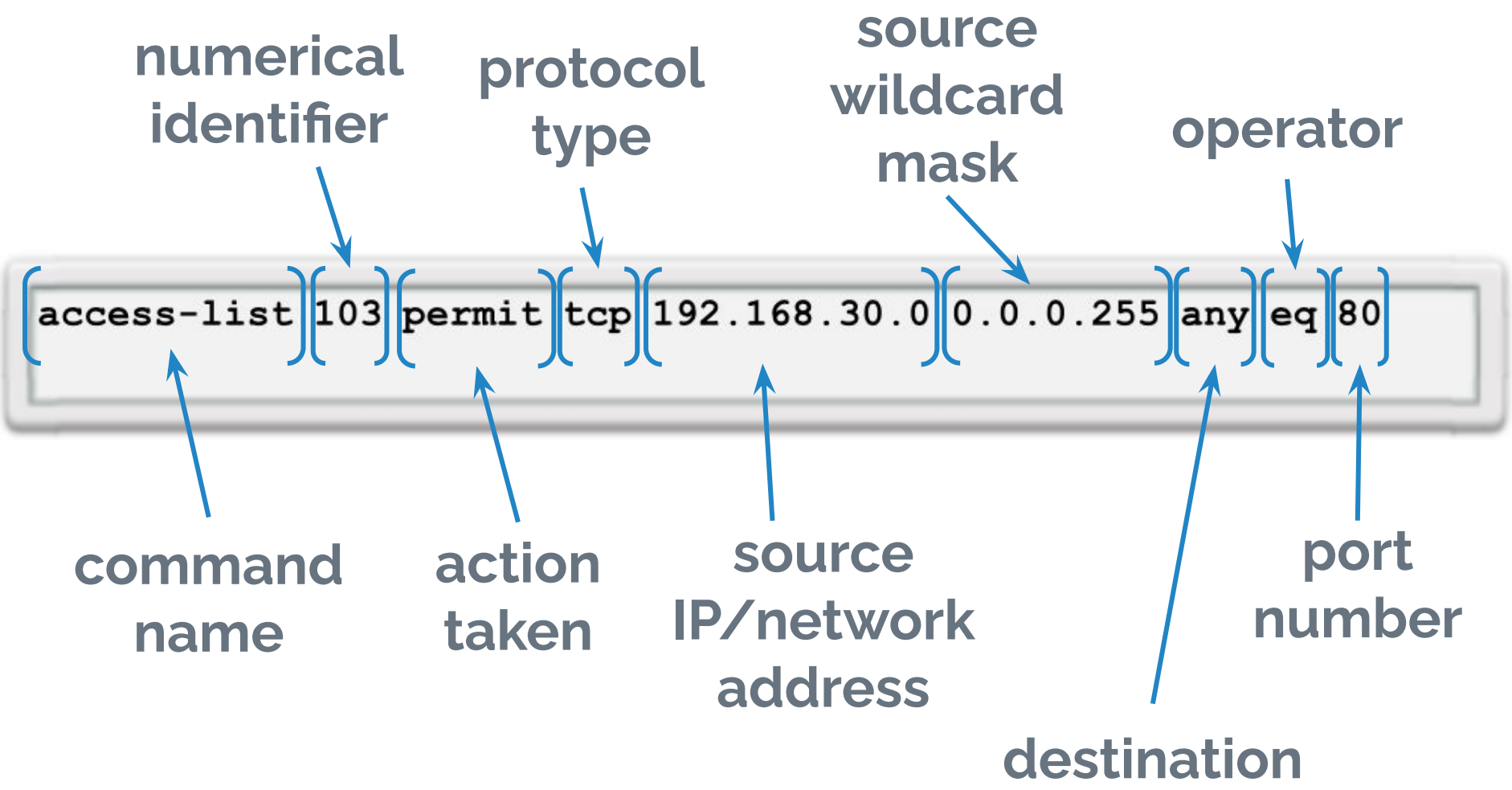
How an ACL Works (Cont.)

- Source/destination IP and wildcard can be replaced with keyword “any” to allow all traffic
- Numerically-identified ACLs cannot be modified; modifications must be made by deleting the list and redoing it
- Remarks (comments) can be added within an ACL to help explain what statements do
- Can only have one ACL per protocol, per direction, and per interface

Anatomy of a Standard ACL



Anatomy of an Extended ACL



Extended ACL Arguments

- Protocol
  - **ip** includes all protocols
  - Specific protocols include **ahp**, **eigrp**, **esp**, **gre**, **icmp**, **ospf**, **tcp**, and **udp**
- Operator (optional)
  - Operators indicate which port numbers are allowed
  - Must be used with a specific protocol
  - Equal (**eq**), not equal (**neq**), greater than (**gt**), less than (**lt**), and **range**
- Port Number (optional)
  - Used in conjunction with an operator
  - Can be numerical or named (e.g. **21** or **ftp**)

Applying ACLs to Interfaces/Lines

Step Description	Placement	Command	Notes
apply an access list to an interface	router interface	ip access-group [name/ACL #] [in/out]	"in/out" applies the ACL to inbound/outbound traffic apply close to destination for standard ACLs; close to source for extended ACLs
apply an access list to a vty line	all vty lines of a router	ip access-class [name/ACL #] [in/out]	"in/out" applies the ACL to inbound/outbound traffic

Named ACLs

- Configuration is slightly different from numerically-identified ACLs
- List must be created first
- Creation of a list enters the named ACL configuration mode, where statements can be placed
- Standard or extended is specified with a keyword
- Named ACLs must still be applied to interfaces/lines

Named ACLs (Cont.)

- Cannot start with a number
- Cannot include spaces
- Names are caps-sensitive
- Typically written in all caps (not required)
- Statements can be removed using the "no" keyword

Example:

```
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
```

Named ACL Configuration

Process	Step Description	Placement	Command	Category	Notes
ACL (standard named)	create/configure a named access list	router config	ip access-list standard [name]	requirement	name cannot begin with a number
	add permit/deny statements	ACL config	[permit/deny] [source ip] [source wildcard]	requirement	the source IP and wildcard can be replaced with "any" to allow all traffic implied "deny any" at the end of all statements subnet mask used for ASA instead of wildcard
	add a remark	ACL config	remark [remark]	optional	100 character limit
ACL (extended named)	create/configure a named access list	router config	ip access-list extended [name]	requirement	name cannot begin with a number
	add permit/deny statements	ACL config	[permit/deny] [protocol] [source ip] [source wildcard] {optional: [operator] [port #]} [destination ip] [destination wildcard] {optional: [operator] [port #]}	requirement	protocols include ahp, eigrp, esp, gre, icmp, ospf, tcp, udp; ip includes all operators include lt, gt, eq, neq, range (cannot use with ip protocol) add "established" to the end of the command to only permit established connections replace an ip/wildcard with "any" to allow all traffic subnet mask used for ASA instead of wildcard
	add a remark	ACL config	remark [remark]	optional	100 character limit

Inbound vs. Outbound Placement

Inbound

- Packets are processed before route is determined
- Efficient because it saves the overhead of routing lookups if packet is discarded
- Best used to filter packets when there is only one source network

Outbound

- Incoming packets are routed to the outbound interface, then processed through the outbound ACL
- Best used when the same filter will be applied to packets from multiple inbound interfaces to the same outbound interface

Inbound and Outbound ACLs

An inbound ACL filters packets coming into a specific interface and before they are routed to the outbound interface.

An outbound ACL filters packets after being routed, regardless of the inbound interface.

Packet Tracer Labs

RSE 9.2.1.10
RSE 9.3.2.11
RSE 9.4.2.8

Credits

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by SlidesCarnival
- Photographs by Unsplash