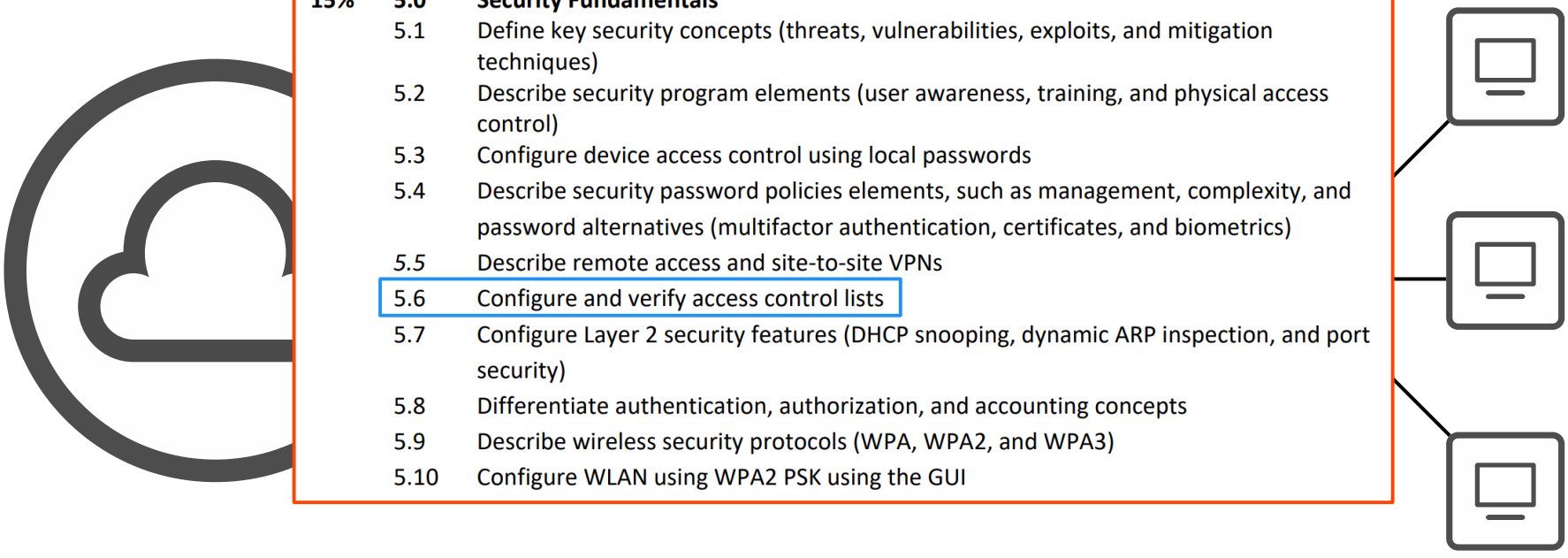
JEREMY'S IT LAB

CCNA 200-301 Day 35

Extended Access Control Lists

15%	**5.0**	**Security Fundamentals**
	5.1	Define key security concepts (threats, vulnerabilities, exploits, and mitigation techniques)
	5.2	Describe security program elements (user awareness, training, and physical access control)
	5.3	Configure device access control using local passwords
	5.4	Describe security password policies elements, such as management, complexity, and password alternatives (multifactor authentication, certificates, and biometrics)
	5.5	Describe remote access and site-to-site VPNs
	5.6	Configure and verify access control lists
	5.7	Configure Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)
	5.8	Differentiate authentication, authorization, and accounting concepts
	5.9	Describe wireless security protocols (WPA, WPA2, and WPA3)
	5.10	Configure WLAN using WPA2 PSK using the GUI



Things we'll cover

- Another way to configure numbered ACLs
- Editing ACLs
- Extended numbered and named ACLs

Configuring numbered ACLs with subcommands

- In Day 34, you learned that numbered ACLs are configured in global config mode:

```
R1(config)# access-list 1 deny 192.168.1.1
R1(config)# access-list 1 permit any
```

- You learned that named ACLs are configured with subcommands in a separate config mode:

```
R1(config)# ip access-list standard BLOCK_PC1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

- However, in modern IOS you can also configure numbered ACLs in the exact same way as named ACLs:

```
R1(config)# ip access-list standard 1
R1(config-std-nacl)# deny 192.168.1.1
R1(config-std-nacl)# permit any
```

- This is just a different way of configuring numbered ACLs. However, in the running-config the ACL will display as if it was configured using the traditional method.

Configuring numbered ACLs with subcommands

```
R1(config)#ip access-list standard ?
  <1-99>        Standard IP access-list number
  <1300-1999>   Standard IP access-list number (expanded range)
  WORD          Access-list name

R1(config)#ip access-list standard 1
R1(config-std-nacl)#deny 192.168.1.1
R1(config-std-nacl)#permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 permit any
R1(config-std-nacl)#
```

Advantages of named ACL config mode

- You can easily delete individual entries in the ACL with **no** *entry-number*.

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#no 30
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
```

Advantages of named ACL config mode

```
R1(config)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.3.0, wildcard bits 0.0.0.255
    40 permit any
R1(config)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.3.0 0.0.0.255
access-list 1 permit any
R1(config)#no access-list 1 deny 192.168.3.0 0.0.0.255
R1(config)#do show access-lists
R1(config)#do show running-config | section access-list
R1(config)#
```

When configuring/editing numbered ACLs from global config mode, you can't delete individual entries, you can only delete the entire ACL!

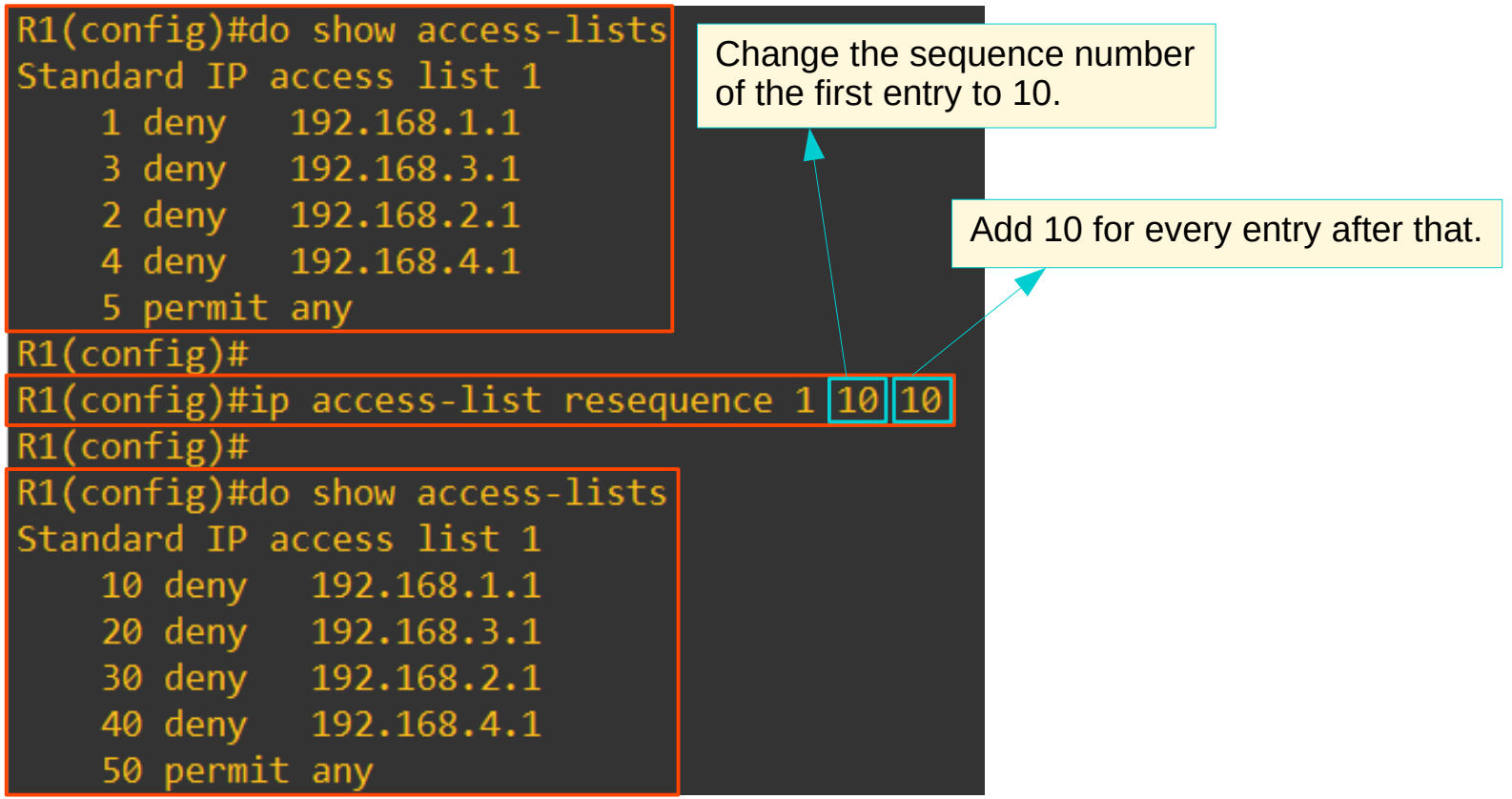
Advantages of named ACL config mode

- You can easily delete individual entries in the ACL with **no** *sequence-number*.
- You can insert new entries in between other entries by specifying the sequence number.

```
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#30 deny 192.168.2.0 0.0.0.255
R1(config-std-nacl)#
R1(config-std-nacl)#do show access-lists
Standard IP access list 1
    10 deny   192.168.1.1
    20 deny   192.168.1.2
    30 deny   192.168.2.0, wildcard bits 0.0.0.255
    40 permit any
R1(config-std-nacl)#
R1(config-std-nacl)#do show running-config | section access-list
access-list 1 deny   192.168.1.1
access-list 1 deny   192.168.1.2
access-list 1 deny   192.168.2.0 0.0.0.255
access-list 1 permit any
```

Resequencing ACLs

- There is a *resequencing* function that helps edit ACLs.
- The command is **`ip access-list resequence`** *`acl-id starting-seq-num increment`*



Extended ACLs

- Extended ACLs function mostly the same as standard ACLs.
- They can be numbered or named, just like standard ACLs.
  → Numbered ACLs use the following ranges: 100 – 199, 2000 – 2699
- They are processed from top to bottom, just like standard ACLs.
- However, the can match traffic based on more parameters, so they are more precise (and more complex) than standard ACLs.
- We will focus on matching based on these main parameters: **Layer 4 protocol/port**, **source address**, and **destination address**.

```
R1(config)# access-list number {permit | deny} protocol src-ip dest-ip
```

```
R1(config)# ip access-list extended {name | number}
R1(config-ext-nacl)# [seq-num] {permit | deny} protocol src-ip dest-ip
```

Matching the protocol

```
R1(config)#ip access-list extended EXAMPLE
R1(config-ext-nacl)#deny ?
  <0-255>          An IP protocol number
  ahp              Authentication Header Protocol
  eigrp            Cisco's EIGRP routing protocol
  esp              Encapsulation Security Payload
  gre              Cisco's GRE tunneling
  icmp             Internet Control Message Protocol
  igmp             Internet Gateway Message Protocol
  ip               Any Internet Protocol
  ipinip           IP in IP tunneling
  nos              KA9Q NOS compatible IP over IP tunneling
  object-group     Service object group
  ospf             OSPF routing protocol
  pcp              Payload Compression Protocol
  pim              Protocol Independent Multicast
  sctp             Stream Control Transmission Protocol
  tcp              Transmission Control Protocol
  udp              User Datagram Protocol
```

1: ICMP
6: TCP
17: UDP
88: EIGRP
89: OSPF

Matching the source/destination IP address

```
R1(config-ext-nacl)#deny tcp ?
  A.B.C.D        Source address
  any            Any source host
  host           A single source host
  object-group   Source network object group

R1(config-ext-nacl)#deny tcp any ?
  A.B.C.D        Destination address
  any            Any destination host
  eq             Match only packets on a given port number
  gt             Match only packets with a greater port number
  host           A single destination host
  lt             Match only packets with a lower port number
  neq            Match only packets not on a given port number
  object-group   Destination network object group
  range          Match only packets in the range of port numbers

R1(config-ext-nacl)#deny tcp any 10.0.0.0 ?
  A.B.C.D  Destination wildcard bits

R1(config-ext-nacl)#deny tcp any 10.0.0.0 0.0.0.255
R1(config-ext-nacl)#
```

In extended ACLs, to specify a /32 source or destination you have to use the **host** option or specify the wildcard mask.
You can't just write the address without either of those.

Deny all packets that encapsulate a TCP segment, from any source, to destination 10.0.0.0/24.

Extended ACL entry practice (1)

1. Allow all traffic

```
R1(config-ext-nacl)#permit ip any any
```

2. Prevent 10.0.0.0/16 from sending UDP traffic to 192.168.1.1/32

```
R1(config-ext-nacl)#deny udp 10.0.0.0 0.0.255.255 host 192.168.1.1
```

3. Prevent 172.16.1.1/32 from pinging hosts in 192.168.0.0/24

```
R1(config-ext-nacl)#deny icmp host 172.16.1.1 192.168.0.0 0.0.0.255
```

Matching the TCP/UDP port numbers

- When matching TCP/UDP, you can optionally specify the source and/or destination port numbers to match.

```
R1(config-ext-nacl)#deny tcp src-ip  eq    src-port-num   dest-ip  eq    dst-port-num
                                     gt                             gt
                                     lt                             lt
                                     neq                            neq
                                     range                          range
```

- **`eq 80`** = equal to port 80
- **`gt 80`** = greater than 80 (81 and greater)
- **`lt 80`** = less than 80 (79 and less)
- **`neq 80`** = NOT 80
- **`range 80 100`** = from port 80 to port 100

TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

Matching the TCP/UDP port numbers

```
R1(config-ext-nacl)#deny tcp any host 1.1.1.1 eq ?
  <0-65535>     Port number
  bgp           Border Gateway Protocol (179)
  chargen       Character generator (19)
  cmd           Remote commands (rcmd, 514)
  daytime       Daytime (13)
  discard       Discard (9)
  domain        Domain Name Service (53)
  drip          Dynamic Routing Information Protocol (3949)
  echo          Echo (7)
  exec          Exec (rsh, 512)
  finger        Finger (79)
  ftp           File Transfer Protocol (21)
  ftp-data      FTP data connections (20)
  gopher        Gopher (70)
  hostname      NIC hostname server (101)
  ident         Ident Protocol (113)
  irc           Internet Relay Chat (194)
  klogin        Kerberos login (543)
  kshell        Kerberos shell (544)
  login         Login (rlogin, 513)
  lpd           Printer service (515)
  nntp          Network News Transport Protocol (119)
  onep-plain    ONEP Cleartext (15001)
  onep-tls      ONEP TLS (15002)
  pim-auto-rp   PIM Auto-RP (496)
  pop2          Post Office Protocol v2 (109)
  pop3          Post Office Protocol v3 (110)
  smtp          Simple Mail Transport Protocol (25)
  sunrpc        Sun Remote Procedure Call (111)
  tacacs        TAC Access Control System (49)
  talk          Talk (517)
  telnet        Telnet (23)
  time          Time (37)
  uucp          Unix-to-Unix Copy Program (540)
  whois         Nicname (43)
  www           World Wide Web (HTTP, 80)
```

```
R1(config-ext-nacl)#deny tcp any host 1.1.1.1 eq 80
```

→ Deny all packets destined for IP address 1.1.1.1/32, TCP port 80.

After the destination IP address and/or destination port numbers, there are many more options you can use to match (not necessary for the CCNA).
Some examples:

- **`ack`**: match the TCP ACK flag
- **`fin`**: match the TCP FIN flag
- **`syn`**: match the TCP SYN flag
- **`ttl`**: match packets with a specific TTL value
- **`dscp`**: match packets with a specific DSCP value

If you specify the protocol, source IP, source port, destination IP, destination port, etc, a packet must match ALL of those values to match the ACL entry. Even if it matches all except one of the parameters, the packet won't match that entry of the ACL.

Extended ACL entry practice (2)

1. Allow traffic from 10.0.0.0/16 to access the server at 2.2.2.2/32 using HTTPS.

```
R1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.255.255 2.2.2.2 0.0.0.0 eq 443
```

2. Prevent all hosts using source UDP port numbers from 20000 to 30000 from accessing the server at 3.3.3.3/32.

```
R1(config-ext-nacl)#deny udp any range 20000 30000 host 3.3.3.3
```

3. Allow hosts in 172.16.1.0/24 using a TCP source port greater than 9999 to access all TCP ports on server 4.4.4.4/32 except port 23.

```
R1(config-ext-nacl)#permit tcp 172.16.1.0 0.0.0.255 gt 9999 host 4.4.4.4 neq 23
```

Extended ACLs

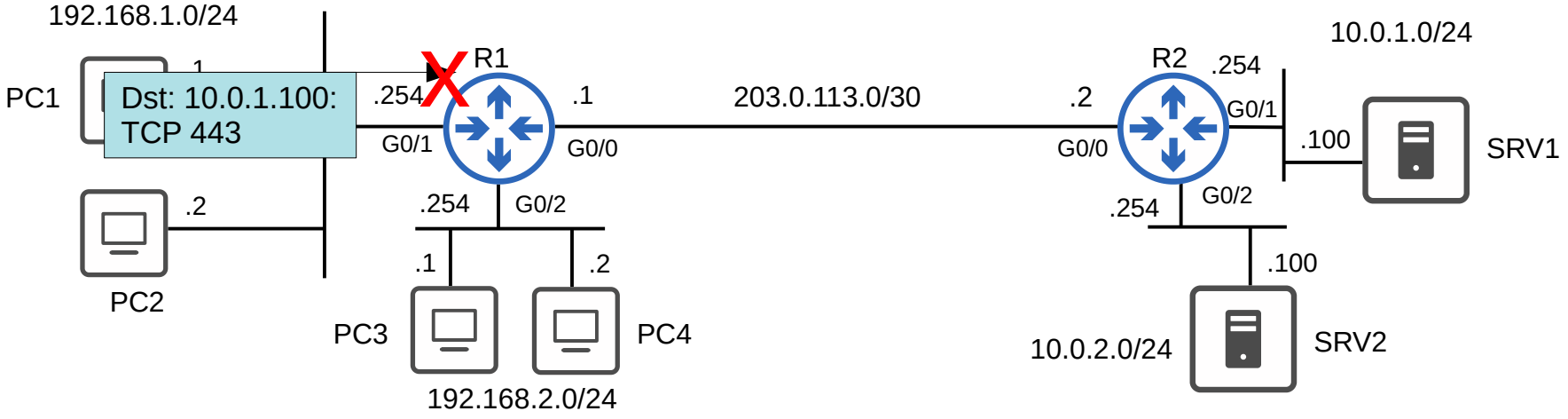
```
R1(config)#ip access-list extended HTTP_SRV1
R1(config-ext-nacl)#deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/1
R1(config-if)#ip access-group HTTP_SRV1 in
```

Extended ACLs should be applied as close to the source as possible, to limit how far the packets travel in the network before being denied.

(Standard ACLs are less specific, so if they are applied close to the source there is a risk of blocking more traffic than intended)

Requirements:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1.
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24.
- None of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 or 10.0.2.0/24.

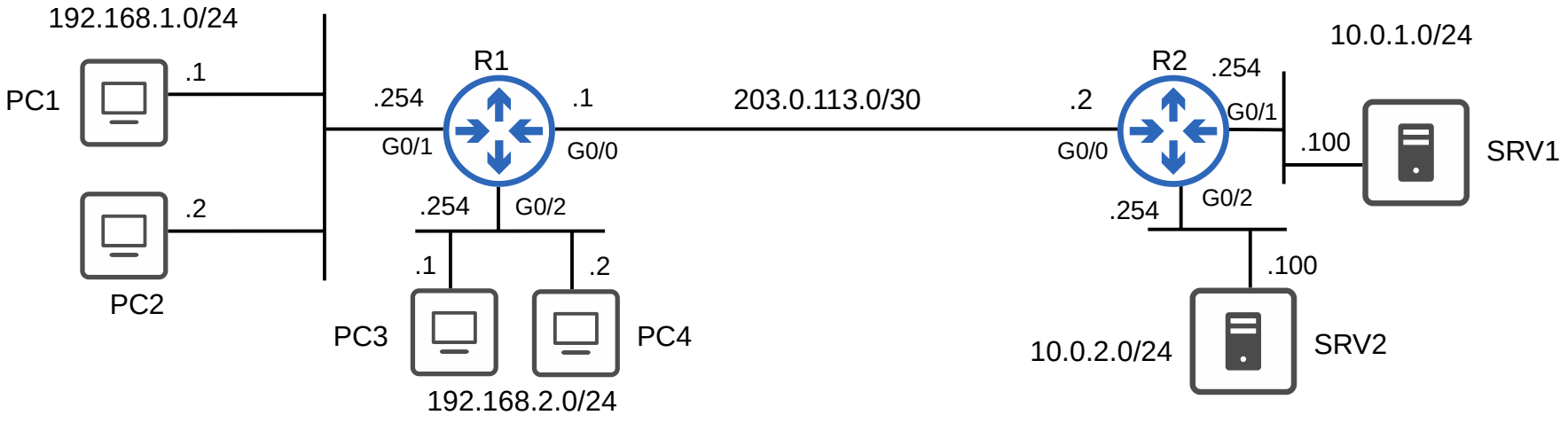


Extended ACLs

```
R1(config)#ip access-list extended BLOCK_10.0.2.0/24
R1(config-ext-nacl)#deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group BLOCK_10.0.2.0/24 in
```

Requirements:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1.
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24.
- None of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 or 10.0.2.0/24.

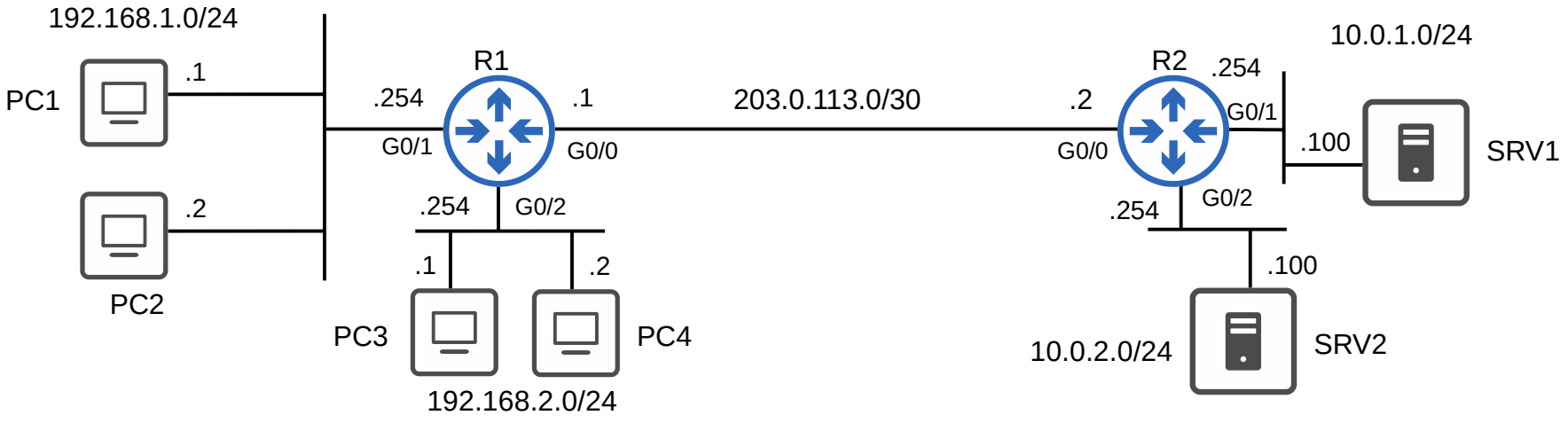


Extended ACLs

```
R1(config)#ip access-list extended BLOCK_ICMP
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
R1(config-ext-nacl)#deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/0
R1(config-if)#ip access-group BLOCK_ICMP out
```

Requirements:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1.
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24.
- None of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 or 10.0.2.0/24.

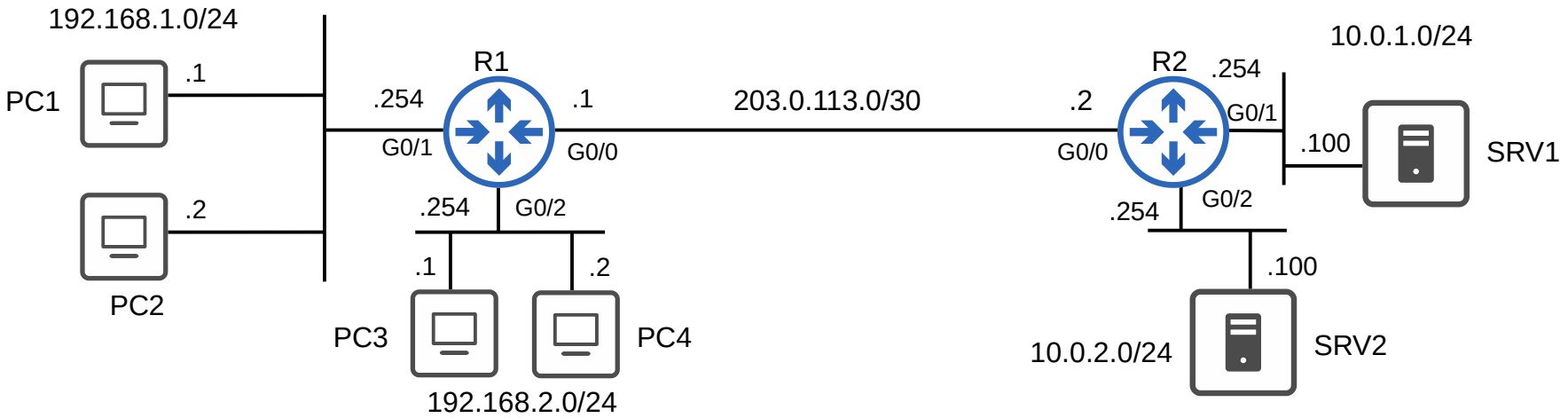


Extended ACLs

```
R1#show access-lists
Extended IP access list BLOCK_10.0.2.0/24
    10 deny ip 192.168.2.0 0.0.0.255 10.0.2.0 0.0.0.255
    20 permit ip any any
Extended IP access list BLOCK_ICMP
    10 deny icmp 192.168.1.0 0.0.0.255 10.0.1.0 0.0.0.255
    20 deny icmp 192.168.1.0 0.0.0.255 10.0.2.0 0.0.0.255
    30 deny icmp 192.168.2.0 0.0.0.255 10.0.1.0 0.0.0.255
    40 permit ip any any
Extended IP access list HTTP_SRV1
    10 deny tcp 192.168.1.0 0.0.0.255 host 10.0.1.100 eq 443
    20 permit ip any any
```

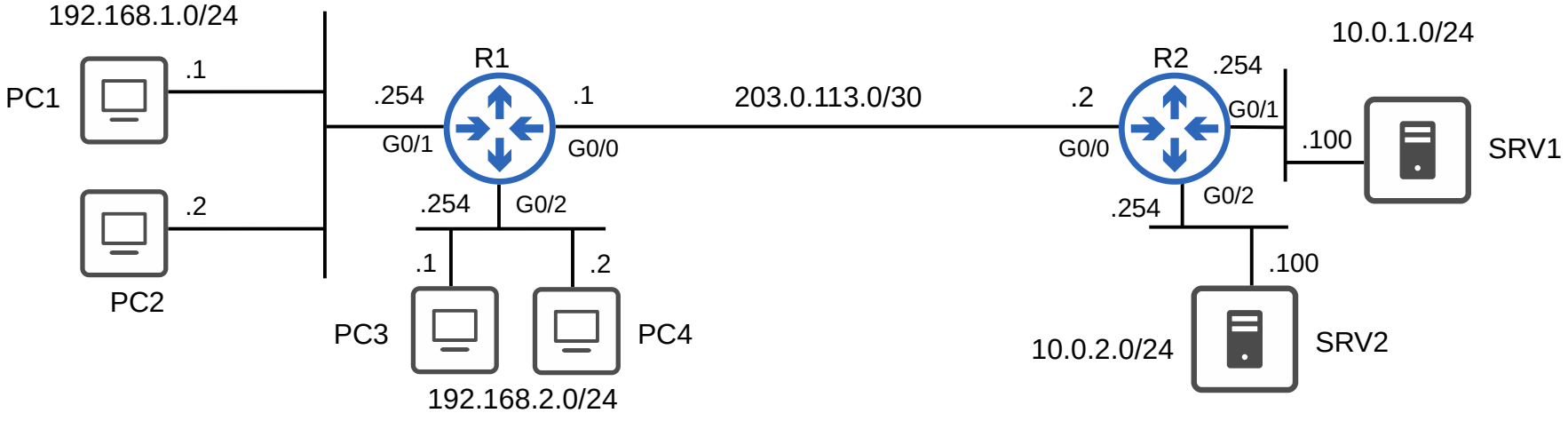
Requirements:

- Hosts in 192.168.1.0/24 can't use HTTPS to access SRV1.
- Hosts in 192.168.2.0/24 can't access 10.0.2.0/24.
- None of the hosts in 192.168.1.0/24 or 192.168.2.0/24 can ping 10.0.1.0/24 or 10.0.2.0/24.



Extended ACLs

```
R1#show ip interface g0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet address is 203.0.113.1/30
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is BLOCK_ICMP
  Inbound  access list is not set
  Proxy ARP is enabled
  Local Proxy ARP is disabled
  Security level is default
  Split horizon is enabled
  ICMP redirects are always sent
  ICMP unreachables are always sent
  ICMP mask replies are never sent
```



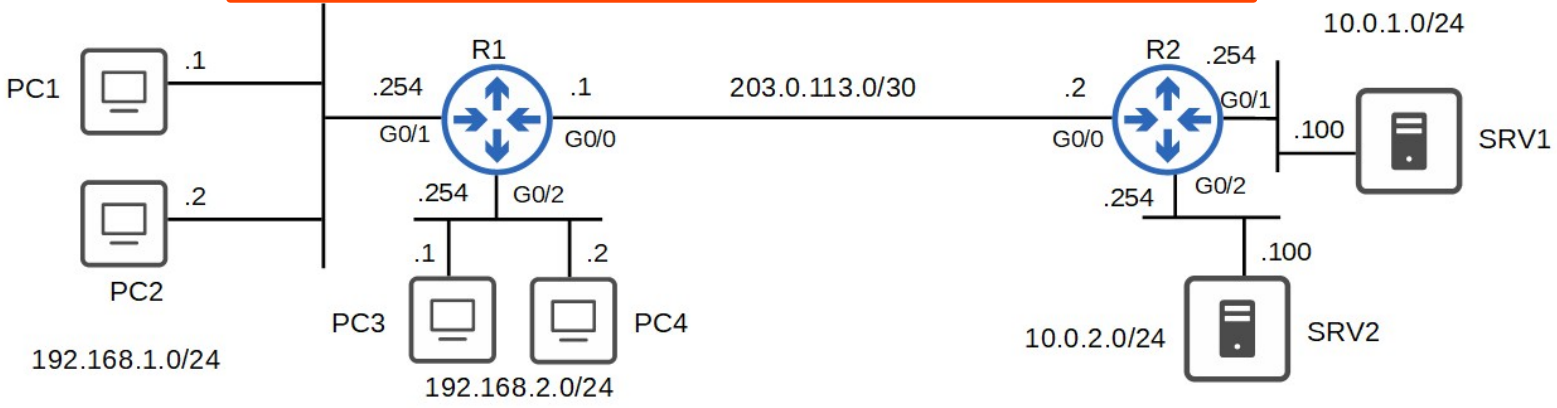
Things we covered

- Another way to configure numbered ACLs
- Editing ACLs
- Extended numbered and named ACLs

Quiz 1

Which ACL, when applied outbound on R1's G0/0, permits only PC1 to access the TFTP server on SRV1?

```
R1#show access-lists
Extended IP access list 100
    10 permit tcp host 192.168.1.1 host 10.0.1.100 eq 69
    20 permit ip any any
Extended IP access list 101
    10 permit udp host 192.168.1.1 host 10.0.1.100 eq tftp
    20 permit ip any any
Extended IP access list 102
    10 permit udp host 192.168.1.1 eq tftp host 10.0.1.100
    20 deny udp any eq tftp host 10.0.1.100
    30 permit ip any any
Extended IP access list 103
    10 permit udp host 192.168.1.1 host 10.0.1.100 eq tftp
    20 deny udp any host 10.0.1.100 eq tftp
    30 permit ip any any
```



Quiz 2

What effect will the following command have on ACL 1?

```
R1(config)#no access-list 1 deny 10.0.2.0 0.0.0.255
```

```
R1#show running-config | section access-list
access-list 1 deny   192.168.1.0 0.0.0.255
access-list 1 deny   10.0.2.0 0.0.0.255
access-list 1 deny   10.0.3.0 0.0.0.255
access-list 1 permit any
```

a) Traffic from 10.0.2.0/24 will be permitted.

b) ACL 1 will be deleted.

c) The command will not work (it will be rejected).

d) Traffic to 10.0.2.0/24 will be permitted.

Quiz 3

Which command was used to resequence ACL 199?

Before:

```
Extended IP access list 199
    1 permit ip 10.0.0.0 0.255.255.255 host 172.16.2.2
    2 permit udp host 1.2.3.4 host 8.8.8.8 eq domain
    3 permit tcp host 1.2.3.4 host 8.8.8.8 eq domain
    4 permit ip 192.168.2.0 0.0.0.255 host 1.1.1.1
    5 deny ip any any
```

After:

```
Extended IP access list 199
    5 permit ip 10.0.0.0 0.255.255.255 host 172.16.2.2
    15 permit udp host 1.2.3.4 host 8.8.8.8 eq domain
    25 permit tcp host 1.2.3.4 host 8.8.8.8 eq domain
    35 permit ip 192.168.2.0 0.0.0.255 host 1.1.1.1
    45 deny ip any any
```

a) `R1(config)#ip access-list extended resequence 199 5 10`

b) `R1(config)#ip access-list resequence 199 5 15 25 35 45`

c) `R1(config)#ip access-list resequence 199 5 10`

d) `R1(config)#ip access-list resequence extended 199 5 10`

Quiz 4

Which of the following ACLs would prevent R1 from forwarding OSPF packets out of G0/2?

```
R1(config)#ip access-list extended 110
R1(config-ext-nacl)#deny 89 any any
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group 110 in
```

```
R1(config)#ip access-list extended 111
R1(config-ext-nacl)#deny 88 any any
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group 111 in
```

```
R1(config)#ip access-list extended 112
R1(config-ext-nacl)#deny 89 any any
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group 112 out
```

```
R1(config)#ip access-list extended 113
R1(config-ext-nacl)#deny 88 any any
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/2
R1(config-if)#ip access-group 113 out
```

Quiz 5

ACL 150 isn't having the intended effect. How can it be fixed to deny HTTP and HTTPS traffic from 192.168.1.0/24 to 10.0.2.0/24, but allow other traffic? (select two)

```
R1(config)#ip access-list extended 150
R1(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255
10.0.2.0 0.0.0.255 eq 80
R1(config-ext-nacl)#deny udp 192.168.1.0 0.0.0.255
10.0.2.0 0.0.0.255 eq 443
R1(config-ext-nacl)#permit ip any any
R1(config-ext-nacl)#interface g0/1
R1(config-if)#ip access-group 150 out
```

a) Swap the source/destination IPs

b) Move the `permit ip any any` statement to the beginning of the ACL

c) Apply the ACL inbound on G0/1, not outbound

d) Apply the ACL inbound on G0/0, not G0/1

e) The protocol should be TCP, not UDP

f) The port numbers should be 88 and 404