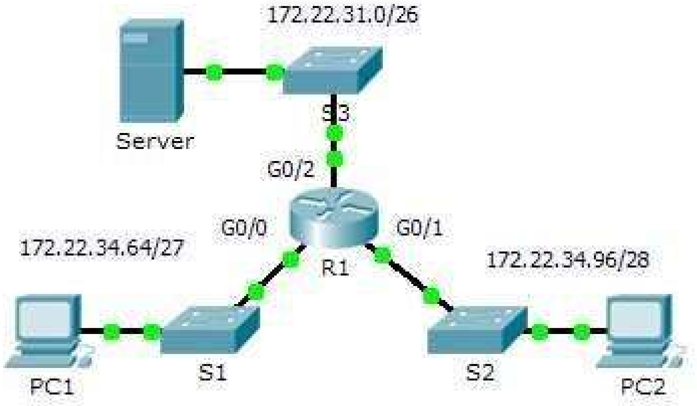
Practical 3: Configuring Extended ACLs - Scenario 1

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	172.22.34.65	255.255.255.224	N/A
	G0/1	172.22.34.97	255.255.255.240	N/A
	G0/2	172.22.34.1	255.255.255.192	N/A
Server	NIC	172.22.34.62	255.255.255.192	172.22.34.1
PC1	NIC	172.22.34.66	255.255.255.224	172.22.34.65
PC2	NIC	172.22.34.98	255.255.255.240	172.22.34.97

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Configure, Apply and Verify an Extended Named ACL

Background / Scenario

Two employees need access to services provided by the server. **PC1** needs only FTP access while **PC2** needs only web access. Both computers are able to ping the server, but not each other.

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Step 1: Configure an ACL to permit FTP and ICMP.

a. From global configuration mode on **R1**, enter the following command to determine the first valid number for an extended access list.

```
R1(config)# access-list ?
  <1-99>     IP standard access list
  <100-199>  IP extended access list
```

b. Add **100** to the command, followed by a question mark.

```
R1(config)#  access-list  100  ?
deny      Specify packets to reject
permit   Specify packets to forward
remark  Access list entry comment
```

c. To permit FTP traffic, enter **permit,** followed by a question mark.

```
R1(config)# access-list 100 permit ?
ahp    Authentication Header Protocol
eigrp  Cisco's EIGRP routing protocol
esp    Encapsulation Security Payload
gre    Cisco's GRE tunneling   icmp
Internet Control Message Protocol   ip
Any Internet Protocol   ospf   OSPF
routing protocol   tcp    Transmission
Control Protocol   udp    User Datagram
Protocol
```

d. This ACL permits FTP and ICMP. ICMP is listed above, but FTP is not, because FTP uses TCP. Therefore,enter **tcp** to further refine the ACL help.

```
R1(config)# access-list 100 permit tcp ?
  A.B.C.D  Source address   any
Any source host    host      A
single source host
```

e. Notice that we could filter just for **PC1** by using the **host** keyword or we could allow **any** host. In this case, any device is allowed that has an address belonging to the 172.22.34.64/27 network. Enter the network address, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 ?
  A.B.C.D  Source wildcard bits
```

f. Calculate the wildcard mask determining the binary opposite of a subnet mask.

```
11111111.11111111.11111111.11100000 = 255.255.255.224
00000000.00000000.00000000.00011111 = 0.0.0.31
```

g. Enter the wildcard mask, followed by a question mark.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 ?
  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   range    Match only
packets in the range of port numbers
```

h. Configure the destination address. In this scenario, we are filtering traffic for a single destination, which is the server. Enter the **host** keyword followed by the server's IP address.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62
?
  dscp            Match packets with given dscp value    eq
Match  only  packets  on  a  given  port  number     established
established   gt            Match only packets with a greater
```

```
port number   lt          Match only packets with a lower port
number   neq          Match only packets not on a given port
number   precedence   Match packets with given precedence value
range        Match only packets in the range of port numbers
  <cr>
```

i. Notice that one of the options is **<cr>** (carriage return). In other words, you can press **Enter** and the statement would permit all TCP traffic. However, we are only permitting FTP traffic; therefore, enter the **eq** keyword, followed by a question mark to display the available options. Then, enter **ftp** and press **Enter**.

```
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62
eq ?
  <0-65535>  Port number   ftp        File
Transfer Protocol (21)   pop3       Post Office
Protocol v3 (110)   smtp       Simple Mail
Transport Protocol (25)   telnet     Telnet (23)
  www        World Wide Web (HTTP, 80)
R1(config)# access-list 100 permit tcp 172.22.34.64 0.0.0.31 host
172.22.34.62 eq ftp
```

j. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC1** to **Server**. Note that the access list number remains the same and no particular type of ICMP traffic needs to be specified.

```
R1(config)# access-list 100 permit icmp 172.22.34.64 0.0.0.31 host
172.22.34.62
```

k. All other traffic is denied, by default.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that ACL 100 applies to is inbound from the network connected to Gigabit Ethernet 0/0 interface. Enter interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/0
```

```
R1(config-if)# ip access-group 100 in
```

Step 3:

Verify the ACL implementation.

a. Ping from **PC1** to **Server**. If the pings are unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC1** to **Server**. The username and password are both **cisco**.

```
PC> ftp 172.22.34.62
```

c. Exit the FTP service of the **Server**.

```
ftp> quit
```

d. Ping from **PC1** to **PC2**. The destination host should be unreachable, because the traffic was not explicitly permitted.

Part 2: Configure, Apply and Verify an Extended Named ACL

Step 1: Configure an ACL to permit HTTP access and ICMP.

a. Named ACLs start with the **ip** keyword. From global configuration mode of **R1**, enter the following command, followed by a question mark.

```
R1(config)# ip access-list ?
```

```
  extended       Extended   Access    List
standard  Standard Access List
```

b. You can configure named standard and extended ACLs. This access list filters both source and destination IP addresses; therefore, it must be extended. Enter **HTTP_ONLY** as the name. (For Packet Tracer scoring, the name is case-sensitive.)

```
R1(config)# ip access-list extended HTTP_ONLY
```

c. The prompt changes. You are now in extended named ACL configuration mode. All devices on the **PC2** LAN need TCP access. Enter the network address, followed by a question mark.

```
R1(config-ext-nacl)# permit tcp 172.22.34.96 ?
  A.B.C.D  Source wildcard bits
```

d. An alternative way to calculate a wildcard is to subtract the subnet mask from 255.255.255.255.

```
  255.255.255.255
- 255.255.255.240
-----------------
=   0.  0.  0. 15
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 ?
```

e. Finish the statement by specifying the server address as you did in Part 1 and filtering **www** traffic.

```
R1(config-ext-nacl)# permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www
```

f. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC2** to **Server**. Note: The prompt remains the same and a specific type of ICMP traffic does not need to be specified.

```
R1(config-ext-nacl)# permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62
```

g. All other traffic is denied, by default. Exit out of extended named ACL configuration mode.

Step 2: Apply the ACL on the correct interface to filter traffic.

From **R1**'s perspective, the traffic that access list **HTTP_ONLY** applies to is inbound from the network connected to Gigabit Ethernet 0/1 interface. Enter the interface configuration mode and apply the ACL.

```
R1(config)# interface gigabitEthernet 0/1
R1(config-if)# ip access-group HTTP_ONLY in Step
```

3: Verify the ACL implementation.

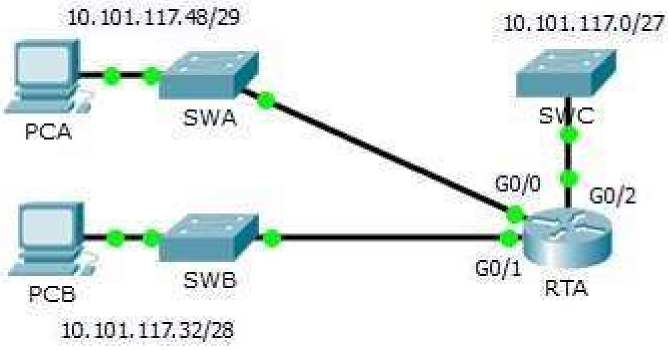
a. Ping from **PC2** to **Server**. The ping should be successful, if the ping is unsuccessful, verify the IP addresses before continuing.

b. FTP from **PC2** to **Server**. The connection should fail.

c. Open the web browser on **PC2** and enter the IP address of **Server** as the URL. The connection should be successful.

Practical 3: Configuring Extended ACLs - Scenario 2

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
RTA	G0/0	10.101.117.49	255.255.255.248	N/A
	G0/1	10.101.117.33	255.255.255.240	N/A
Fa4/0	G0/2	10.101.117.1	255.255.255.224	N/A
PCA	NIC	10.101.117.51	255.255.255.248	10.101.117.49
PCB	NIC	10.101.117.35	255.255.255.240	10.101.117.33
SWA	VLAN 1	10.101.117.50	255.255.255.248	10.101.117.49
SWB	VLAN 1	10.101.117.34	255.255.255.240	10.101.117.33
SWC	VLAN 1	10.101.117.2	255.255.255.224	10.101.117.1

Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Reflection Questions

Background / Scenario

In this scenario, devices on one LAN are allowed to remotely access devices in another LAN using the SSH protocol. Besides ICMP, all traffic from other networks is denied.

The switches and router have also been pre-configured with the following:

- Enable secret password: **ciscoenpa55**
- Console password: **ciscoconpa55**
- Local username and password: **Admin** / **Adminpa55**

Packet Tracer - Configuring Extended ACLs - Scenario 2

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Configure, apply and verify an ACL to satisfy the following policy:

- SSH traffic from devices on the 10.101.117.32/28 network is allowed to devices on the 10.101.117.0/27 networks.
- ICMP traffic is allowed from any source to any destination.
- All other traffic to 10.101.117.0/27 is blocked.

Step 1: Configure the extended ACL.

a. From the appropriate configuration mode on **RTA**, use the last valid extended access list number to configure the ACL. Use the following steps to construct the first ACL statement:

1) The last extended list number is 199.
2) The protocol is TCP.
3) The source network is 10.101.117.32.
4) The wildcard can be determined by subtracting 255.255.255.240 from 255.255.255.255.
5) The destination network is 10.101.117.0.
6) The wildcard can be determined by subtracting 255.255.255.224 from 255.255.255.255.
7) The protocol is SSH (port 22).

What is the first ACL statement?

```
access-list 199 permit tcp 10.101.117.32 0.0.0.15 10.101.117.0 0.0.0.31 eq 22
```

b. ICMP is allowed, and a second ACL statement is needed. Use the same access list number to permit all ICMP traffic, regardless of the source or destination address. What is the second ACL statement? (Hint: Use the **any** keywords)

```
access-list 199 permit icmp any any
```

c. All other IP traffic is denied, by default.

Step 2: Apply the extended ACL.

The general rule is to place extended ACLs close to the source. However, because access list 199 affects traffic originating from both networks 10.101.117.48/29 and 10.101.117.32/28, the best placement for this ACL might be on interface Gigabit Ethernet 0/2 in the outbound direction. What is the command to apply ACL 199 to the Gigabit Ethernet 0/2 interface?

```
ip access-group 199 out
```

Step 3: Verify the extended ACL implementation.

a. Ping from **PCB** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

b. SSH from **PCB** to **SWC**. The username is **~~Admin~~**, and the password is **~~Adminpa55~~**.

```
PC> ssh -l Admin 10.101.117.2
```

c. Exit the SSH session to **SWC**.

d. Ping from **PCA** to all of the other IP addresses in the network. If the pings are unsuccessful, verify the IP addresses before continuing.

e. SSH from **PCA** to **SWC**. The access list causes the router to reject the connection.

f. SSH from **PCA** to **SWB**. The access list is placed on **G0/2** and does not affect this connection. The username is **Admin**, and the password is **Adminpa55**.

g. After logging into **SWB**, do not log out. SSH to **SWC** in privileged EXEC mode.

```
SWB# ssh -l Admin 10.101.117.2
```

Part 2: Reflection Questions

1. How was PCA able to bypass access list 199 and SSH to SWC?

   Two steps were used: First, PCA used SSH to access SWB. From SWB, SSH was allowed to SWC.

2. What could have been done to prevent PCA from accessing SWC indirectly, while allowing PCB SSH access to SWC?

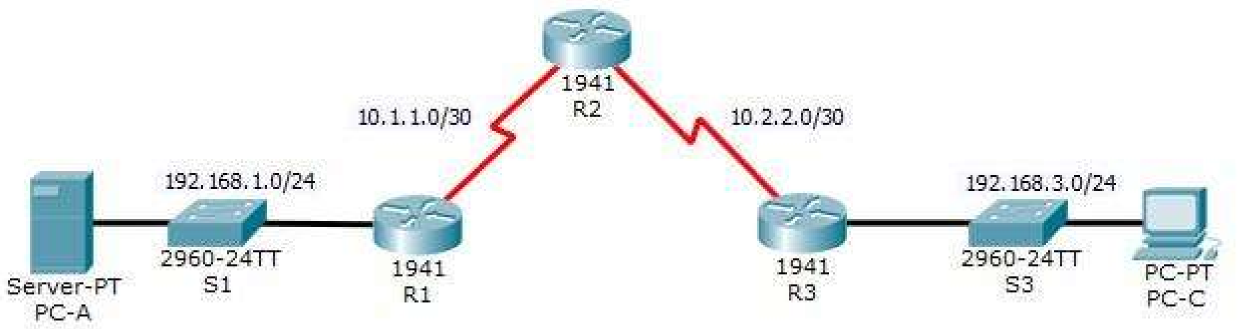
   Because it was requested to block all traffic to 10.101.117.0/27 except SSH traffic originating from 10.101.117.32/28 the access list could be written as is. Instead of applying the ACL to G0/2 outbound apply the same ACL to both G0/0 and G0/1 inbound.

Suggested Scoring Rubric

Activity Section	**Question Location**	**Possible Points**	**Earned Points**
Part 1: Configure, Apply and Verify an Extended Numbered ACL	Step 1a	4	
	Step 1b	4	
	Step 2	4	
	Part 1 Total	**12**	
Part 2: Reflection Questions	Question 1	4	
	Question 2	4	
	Part 2 Total	**8**	
	Packet Tracer Score	**80**	
	Total Score	**100**	

Practical 4: Configure IP ACLs to Mitigate Attacks.

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway	Switch Port
R1	G0/1	192.168.1.1	255.255.255.0	N/A	S1 F0/5
	S0/0/0 (DCE)	10.1.1.1	255.255.255.252	N/A	N/A
R2	S0/0/0	10.1.1.2	255.255.255.252	N/A	N/A
	S0/0/1 (DCE)	10.2.2.2	255.255.255.252	N/A	N/A
	Lo0	192.168.2.1	255.255.255.0	N/A	N/A
R3	G0/1	192.168.3.1	255.255.255.0	N/A	S3 F0/5
	S0/0/1	10.2.2.1	255.255.255.252	N/A	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1	S1 F0/6
PC-C	NIC	192.168.3.3	255.255.255.0	192.168.3.1	S3 F0/18

Objectives

- Verify connectivity among devices before firewall configuration.
- Use ACLs to ensure remote access to the routers is available only from management station PC-C.
- Configure ACLs on R1 and R3 to mitigate attacks.
- Verify ACL functionality.

Background/Scenario

Access to routers R1, R2, and R3 should only be permitted from PC-C, the management station. PC-C is also used for connectivity testing to PC-A, which is a server providing DNS, SMTP, FTP, and HTTPS services.

Standard operating procedure is to apply ACLs on edge routers to mitigate common threats based on source and destination IP address. In this activity, you will create ACLs on edge routers R1 and R3 to achieve this goal. You will then verify ACL functionality from internal and external hosts.

The routers have been pre-configured with the following:

o Enable password: **ciscoenpa55** o

Password for console: **ciscoconpa55**

o SSH logon username and password:

SSHadmin/ciscosshpa55 o IP addressing o

Static routing

Part 1: Verify Basic Network Connectivity

Verify network connectivity prior to configuring the IP ACLs.

Step 1: From PC-A, verify connectivity to PC-C and R2.

a. From the command prompt, ping **PC-C** (192.168.3.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. When finished, exit the SSH session. SERVER> **ssh -l SSHadmin 192.168.2.1**

Step 2: From PC-C, verify connectivity to PC-A and R2.

a. From the command prompt, ping **PC-A** (192.168.1.3).

b. From the command prompt, establish an SSH session to **R2** Lo0 interface (192.168.2.1) using username **SSHadmin** and password **ciscosshpa55**. Close the SSH session when finished. PC> **ssh -l SSHadmin 192.168.2.1**

c. Open a web browser to the **PC-A** server (192.168.1.3) to display the web page. Close the browser when done.

Part 2: Secure Access to Routers

Step 1: Configure ACL 10 to block all remote access to the routers except from PC-C. Use

the **access-list** command to create a numbered IP ACL on **R1**, **R2**, and **R3**.

```
R1(config)# access-list 10 permit host 192.168.3.3
R2(config)# access-list 10 permit host 192.168.3.3
R3(config)# access-list 10 permit host 192.168.3.3
```

Step 2: Apply ACL 10 to ingress traffic on the VTY lines. Use the **access-class**

command to apply the access list to incoming traffic on the VTY lines.

```
R1(config-line)# access-class 10 in
R2(config-line)# access-class 10 in
R3(config-line)# access-class 10 in
```

Step 3: Verify exclusive access from management station PC-C.

a. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

```
PC> ssh –l SSHadmin 192.168.2.1
```

b. Establish an SSH session to 192.168.2.1 from **PC-A** (should fail).

Part 3: Create a Numbered IP ACL 120 on R1

Create an IP ACL numbered 120 with the following rules:

○ Permit any outside host to access DNS, SMTP, and FTP services on server

PC-A. ○ Deny any outside host access to HTTPS services on **PC-A.** ○

Permit **PC-C** to access **R1** via SSH.

Note: Check Results will not show a correct configuration for ACL 120 until you modify it in Part 4.

Step 1: Verify that PC-C can access the PC-A via HTTPS using the web browser.

Be sure to disable HTTP and enable HTTPS on server **PC-A**.

Step 2: Configure ACL 120 to specifically permit and deny the specified traffic. Use

the **access-list** command to create a numbered IP ACL.

```
R1(config)# access-list 120 permit udp any host 192.168.1.3 eq domain
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq smtp
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq ftp
R1(config)# access-list 120 deny tcp any host 192.168.1.3 eq 443
R1(config)# access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22
```

Step 3: Apply the ACL to interface S0/0/0. Use the **ip access-group** command to apply the

access list to incoming traffic on interface S0/0/0.

```
R1(config)# interface s0/0/0
R1(config-if)# ip access-group 120 in
```

Step 4: Verify that PC-C cannot access PC-A via HTTPS using the web browser. Part

4: Modify an Existing ACL on R1

Permit ICMP echo replies and destination unreachable messages from the outside network (relative to **R1**). Deny all other incoming ICMP packets.

Step 1: Verify that PC-A cannot successfully ping the loopback interface on R2.

Step 2: Make any necessary changes to ACL 120 to permit and deny the specified traffic. Use

the **access-list** command to create a numbered IP ACL.

```
R1(config)# access-list 120 permit icmp any any echo-reply
R1(config)# access-list 120 permit icmp any any unreachable
R1(config)# access-list 120 deny icmp any any
R1(config)# access-list 120 permit ip any any
```

Step 3: Verify that PC-A can successfully ping the loopback interface on R2. Part

5: Create a Numbered IP ACL 110 on R3

Deny all outbound packets with source address outside the range of internal IP addresses on **R3**.

Step 1: Configure ACL 110 to permit only traffic from the inside network.
Use

the **access-list** command to create a numbered IP ACL.

```
R3(config)# access-list 110 permit ip 192.168.3.0 0.0.0.255 any
```

Step 2: Apply the ACL to interface G0/1.
Use the **ip access-group** command to apply the

access list to incoming traffic on interface G0/1.

```
R3(config)# interface g0/1
R3(config-if)# ip access-group 110 in
```

Part 6: Create a Numbered IP ACL 100 on R3

On **R3**, block all packets containing the source IP address from the following pool of addresses: any RFC 1918 private addresses, 127.0.0.0/8, and any IP multicast address. Since **PC-C** is being used for remote administration, permit SSH traffic from the 10.0.0.0/8 network to return to the host **PC-C**.

Step 1: Configure ACL 100 to block all specified traffic from the outside network.

You should also block traffic sourced from your own internal address space if it is not an RFC 1918 address. In this activity, your internal address space is part of the private address space specified in RFC 1918. Use the **access-list** command to create a numbered IP ACL. **`access-list 100 permit tcp 10.0.0.0`**

```
R3(config)#

0.255.255.255 eq 22 host
192.168.3.3
R3(config)# access-list 100 deny ip 10.0.0.0 0.255.255.255 any
R3(config)# access-list 100 deny ip 172.16.0.0 0.15.255.255 any
R3(config)# access-list 100 deny ip 192.168.0.0 0.0.255.255 any
R3(config)# access-list 100 deny ip 127.0.0.0 0.255.255.255 any
R3(config)# access-list 100 deny ip 224.0.0.0 15.255.255.255 any R3(config)#
access-list 100 permit ip any any
```

Step 2: Apply the ACL to interface Serial 0/0/1.
Use the **ip access-group** command to apply the

access list to incoming traffic on interface Serial 0/0/1.

```
R3(config)# interface s0/0/1
R3(config-if)# ip access-group 100 in
```

Step 3: Confirm that the specified traffic entering interface Serial 0/0/1 is handled correctly.

a. From the PC-C command prompt, ping the PC-A server. The ICMP echo replies are blocked by the ACL since they are sourced from the 192.168.0.0/16 address space.

b. Establish an SSH session to 192.168.2.1 from **PC-C** (should be successful).

Step 4: Check results.

Your completion percentage should be 100%. Click **Check Results** to see feedback and verification of which required components have been completed.

!!!Script for R1

```
access-list 10 permit host 192.168.3.3
line vty 0 4
 access-class 10 in
access-list 120 permit udp any host 192.168.1.3 eq domain
access-list 120 permit tcp any host 192.168.1.3 eq smtp
access-list 120 permit tcp any host 192.168.1.3 eq ftp access-
list 120 deny tcp any host 192.168.1.3 eq 443 access-list 120
permit tcp host 192.168.3.3 host 10.1.1.1 eq 22 interface
s0/0/0  ip access-group 120 in
access-list 120 permit icmp any any echo-reply
access-list 120 permit icmp any any unreachable
access-list 120 deny icmp any any access-list
120 permit ip any any
```

!!!Script for R2

```
access-list 10 permit host 192.168.3.3

line vty 0 4

 access-class 10 in
```

!!!Script for R3

```
access-list 10 permit host 192.168.3.3
line vty 0 4
 access-class 10 in
access-list 100 permit tcp 10.0.0.0 0.255.255.255 eq 22 host 192.168.3.3
access-list 100 deny ip 10.0.0.0 0.255.255.255 any access-list
100 deny ip 172.16.0.0 0.15.255.255 any access-list 100 deny
ip 192.168.0.0 0.0.255.255 any access-list 100 deny ip
127.0.0.0 0.255.255.255 any access-list 100 deny ip 224.0.0.0
15.255.255.255 any
access-list 100 permit ip any any
interface s0/0/1  ip access-group
100 in
access-list 110 permit ip 192.168.3.0 0.0.0.255 any
interface g0/1  ip access-group 110 in
```