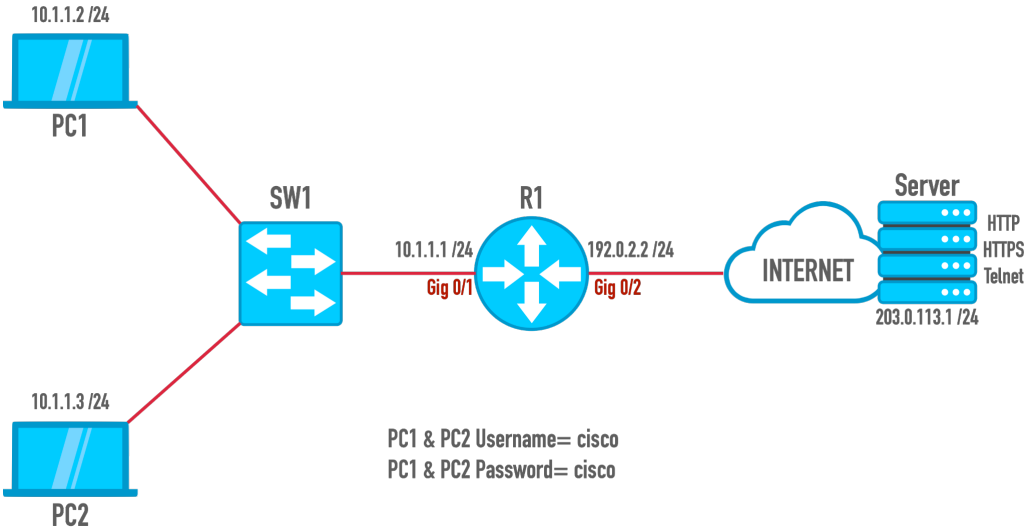
Extended Numbered ACL

Topology



Initial Configuration Commands

PC1:

```
sudo hostname PC1
sudo ifconfig eth0 10.1.1.2 netmask 255.255.255.0 up
sudo route add default gw 10.1.1.1
```

PC2:

```
sudo hostname PC2
sudo ifconfig eth0 10.1.1.3 netmask 255.255.255.0 up
sudo route add default gw 10.1.1.1
```

SW1:

```
enable
conf t
no ip domain-lookup
logging console
line con 0
logging synchronous
```

```
exec-timeout 0 0
hostname SW1
end
copy run star
```

R1:

```
enable
conf t
host R1
no banner motd
no banner login
no banner exec
no banner incoming
line vty 0 15
password cisco
login
exec-timeout 0 0
transport input telnet
line con 0
logging synchronous
exit
no ip domain-lookup
ipv6 unicast-routing
int gig 0/1
ip address 10.1.1.1 255.255.255.0
ipv6 address 2000:2::1/64
no shutdown
int gig 0/2
ip address 192.0.2.2 255.255.255.0
ipv6 address 2000:1::1/64
no shutdown
exit
ipv6 router rip ROUTE
int gig 0/1
ipv6 rip ROUTE enable
exit
int gig 0/2
ipv6 rip ROUTE enable
exit
router ospf 1
network 0.0.0.0 255.255.255.255 area 0
```

```
end
copy run star
```

SERVER:

```
enable
conf t
host SERVER
no banner motd
no banner login
no banner exec
no banner incoming
line vty 0 15
password cisco
login
exec-timeout 0 0
transport input telnet
line con 0
logging synchronous
int lo0
ip address 203.0.113.1 255.255.255.0
ipv6 address 2000:A::1/64
exit
no ip domain-lookup
ipv6 unicast-routingipv6 address 2000:1::2/64
int gig 0/1
ip address 192.0.2.1 255.255.255.0
no shutdown
exit
ipv6 router rip ROUTE
int gig 0/1
ipv6 rip ROUTE enable
exit
int lo0
ipv6 rip ROUTE enable
exit
router ospf 1
network 0.0.0.0 255.255.255.255 area 0
exit
ip http server
ip http secure-server
```

```
end
copy run star
```

Lab Tasks

- Ping the server from both PC1 and PC2 to make sure we can reach the server.
- PC1 should not be allowed to contact the server using Telnet.
- Allow PC1 and PC2 to connect to the server using all other ports by specifying the subnet of both PC's.
- Apply the ACL going into Gig 0/1.
- Take a look at the ACL that we just created.
- Telnet to the server by using the port numbers of HTTPS, HTTP, and Telnet and see if we are able to connect to the server on PC1. Confirm that PC1 is not able to Telnet to the server.
- Telnet to the server by using the port numbers of HTTPS, HTTP, and Telnet and see if we are able to connect to the server on PC2. Confirm that PC2 is still able to telnet to the server.

Solution

Step 1: Ping the server from both PC1 and PC2 to make sure we can reach the server.

PC1

```
PC1 login: cisco
Password: cisco

PC1:~$ ping 203.0.113.1
PING 203.0.113.1 (203.0.113.1): 56 data bytes
64 bytes from 203.0.113.1: seq=0 ttl=42 time=25.585 ms
64 bytes from 203.0.113.1: seq=1 ttl=42 time=12.622 ms
64 bytes from 203.0.113.1: seq=2 ttl=42 time=10.649 ms
64 bytes from 203.0.113.1: seq=3 ttl=42 time=10.265 ms
^C
--- 203.0.113.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 10.265/14.780/25.585 ms
```

PC2

```
PC2 login: cisco
Password: cisco

PC2:~$ ping 203.0.113.1
```

```
PING 203.0.113.1 (203.0.113.1): 56 data bytes
64 bytes from 203.0.113.1: seq=1 ttl=42 time=12.431 ms
64 bytes from 203.0.113.1: seq=2 ttl=42 time=10.929 ms
64 bytes from 203.0.113.1: seq=3 ttl=42 time=10.526 ms
64 bytes from 203.0.113.1: seq=4 ttl=42 time=11.563 ms
^C
--- 203.0.113.1 ping statistics ---
5 packets transmitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 10.526/11.362/12.431 ms
```

Step 2: PC1 should not be allowed to contact the server using Telnet.

```
R1>en
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 100 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

R1(config)#access-list 100 deny tcp host 10.1.1.2 host 203.0.113.1 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)#access-list 100 deny tcp host 10.1.1.2 host 203.0.113.1 eq 23
```

Step 3: Allow PC1 and PC2 to connect to the server using all other ports by specifying the subnet of both PC's.

```
R1(config)#access-list 100 permit ip 10.1.1.0 0.0.0.255 any
```

Step 4: Apply the ACL going into Gig 0/1.

```
R1(config)#int gig 0/1
R1(config-if)#ip access-group 100 in
R1(config-if)#end
```

Step 5: Take a look at the ACL that we just created.

R1**#show access-list**
Extended IP access list 100
  10 deny tcp host 10.1.1.2 host 203.0.113.1 eq telnet
  20 permit ip 10.1.1.0 0.0.0.255 any

(## Notice how we put our most specific ACL at the top)

Step 6: Telnet to the server by using the port numbers of HTTPS, HTTP, and Telnet and see if we are able to connect to the server on PC1. Confirm that PC1 is not able to Telnet to the server.

HTTPS
PC1:~$ **telnet 203.0.113.1 443**
Connected to 203.0.113.1

HTTP
PC1:~$ **telnet 203.0.113.1 80**
Connected to 203.0.113.1

Telnet
PC1:~$ **telnet 203.0.113.1**
telnet: can't connect to remote host (203.0.113.1): Host is unreachable

(## Notice how we are able to connect to the SERVER via HTTPS and HTTP, but we are not able to connect to the SERVER via Telnet because of our ACL.)

Step 7: Telnet to the server by using the port numbers of HTTPS, HTTP, and Telnet and see if we are able to connect to the server on PC2. Confirm that PC2 is still able to telnet to the server.

HTTPS
PC2:~$ **telnet 203.0.113.1 443**
Connected to 203.0.113.1

HTTP
PC2:~$ **telnet 203.0.113.1 80**
Connected to 203.0.113.1

Telnet
PC2:~$ **telnet 203.0.113.1**
Connected to 203.0.113.1

(## Notice how we are able to connect to the SERVER on PC2 via HTTPS, HTTP, and Telnet.)