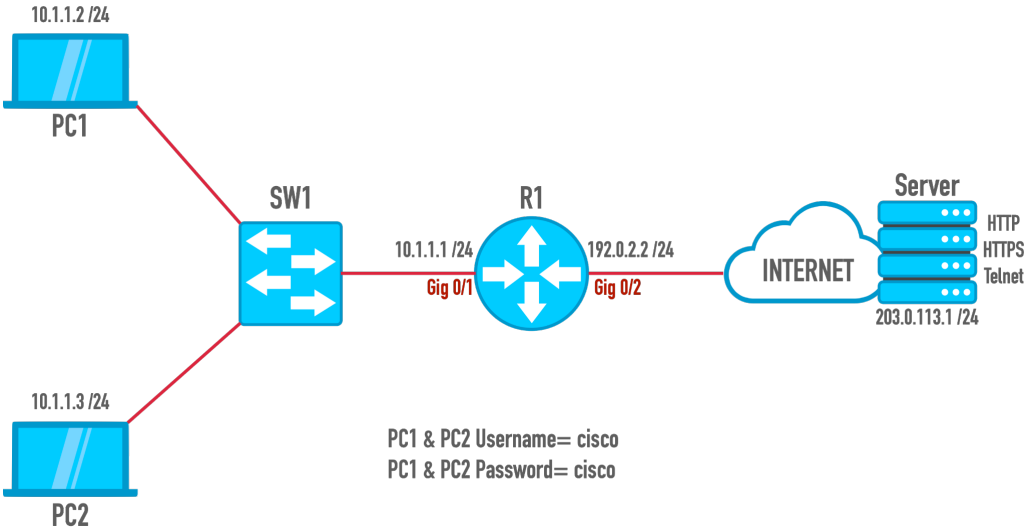
Standard Numbered ACL

Topology



Initial Configuration Commands

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 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
exit
no ip domain-lookup
ipv6 unicast-routing
int gig 0/1
ip address 192.0.2.1 255.255.255.0
ipv6 address 2000:1::2/64
no shutdown
int lo0
ip address 203.0.113.1 255.255.255.0
ipv6 address 2000:A::1/64
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.
- Permit all IP traffic from PC1 to the Server.
- Deny all IP traffic from PC2 to the Server.
- Ping the server from PC1 to make sure we can still reach the server. Ping the server from PC2 and make sure our traffic from PC2 gets denied.

Solution

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

PC1

```
inserthostname-here login: cisco
Password: cisco

inserthostname-here:~$ 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=2.763 ms
64 bytes from 203.0.113.1: seq=1 ttl=42 time=2.643 ms
64 bytes from 203.0.113.1: seq=2 ttl=42 time=2.889 ms
64 bytes from 203.0.113.1: seq=3 ttl=42 time=2.801 ms
^C
--- 203.0.113.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2.643/2.774/2.889 ms
```

PC2

```
inserthostname-here login: cisco
Password: cisco

inserthostname-here:~$ 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=4.631 ms
64 bytes from 203.0.113.1: seq=1 ttl=42 time=2.483 ms
64 bytes from 203.0.113.1: seq=2 ttl=42 time=2.738 ms
^C
--- 203.0.113.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 2.483/3.284/4.631 ms
```

Step 2: Permit all IP traffic from PC1 to the Server.

```
R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list ?
  <1-99>              IP standard access list
  <100-199>           IP extended access list
  <1100-1199>         Extended 48-bit MAC address access list
  <1300-1999>         IP standard access list (expanded range)
  <200-299>           Protocol type-code access list
  <2000-2699>         IP extended access list (expanded range)
  <2700-2799>         MPLS access list
  <300-399>           DECnet access list
  <700-799>           48-bit MAC address access list
  compiled            Enable IP access-list compilation
  dynamic-extended    Extend the dynamic ACL absolute timer
  rate-limit          Simple rate-limit specific access list

R1(config)#access-list 1 ?
  deny    Specify packets to reject
  permit  Specify packets to forward
  remark  Access list entry comment

R1(config)#access-list 1 permit host 10.1.1.2
```

Step 3: Deny all IP traffic from PC2 to the Server.

```
R1#conf t
R1(config)#access-list 1 deny host 10.1.1.3
R1(config)#int gig 0/1
R1(config-if)#ip access-group 1 in
R1(config-if)#end
R1#show access-list
Standard IP access list 1
    10 permit 10.1.1.2
    20 deny   10.1.1.3
```

Step 4: Ping the server from PC1 to make sure we can still reach the server. Ping the server from PC2 and make sure our traffic from PC2 gets denied.

PC1

```
inserthostname-here:~$ 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=2.594 ms
64 bytes from 203.0.113.1: seq=1 ttl=42 time=3.088 ms
```

```
64 bytes from 203.0.113.1: seq=2 ttl=42 time=2.818 ms
64 bytes from 203.0.113.1: seq=3 ttl=42 time=2.772 ms
^C
--- 203.0.113.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 2.594/2.818/3.088 ms
```

(##Notice how packets from PC1 are making it to the server)

PC2

```
inserthostname-here:~$ ping 203.0.113.1
PING 203.0.113.1 (203.0.113.1): 56 data bytes
^C
--- 203.0.113.1 ping statistics ---
6 packets transmitted, 0 packets received, 100% packet loss
```

(##Notice how packets from PC2 are not making it to the server)