version 4

IEWB-RS Technology Labs
Security

Brian Dennis, CCIE # 2210 (R&S / ISP Dial / Security / Service Provider)
Brian McGahan, CCIE# 8583 (R&S / Service Provider)

Copyright Information

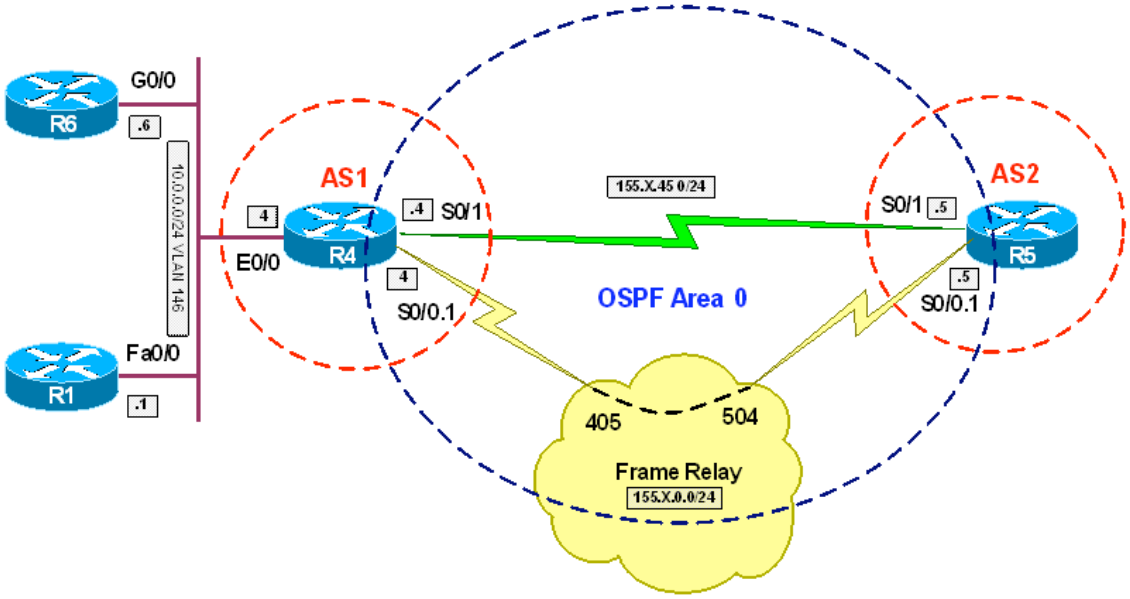
Disclaimer

The following publication, ***CCIE Routing and Switching Lab Workbook***, is designed to assist candidates in the preparation for Cisco Systems' CCIE Routing & Switching Lab exam. While every effort has been made to ensure that all material is as complete and accurate as possible, the enclosed material is presented on an "as is" basis. Neither the authors nor Internetwork Expert, Inc. assume any liability or responsibility to any person or entity with respect to loss or damages incurred from the information contained in this workbook.

This workbook was developed by Internetwork Expert, Inc. and is an original work of the aforementioned authors. Any similarities between material presented in this workbook and actual CCIE™ lab material is completely coincidental.

Traffic Filtering with Access Lists

Objective: Configure access-lists to permit FTP and WWW connections to specific servers. Do not block the necessary traffic (routing, etc)



Directions

- Configure routers as per the NAT scenario “Configuring Static NAT”
- The servers mentioned in the task have IP addresses 150.X.4.1 and 150.X.4.6 respectively (static NAT mappings)
- The task is to permit TCP connections to ports of FTP and WWW services. Either Passive or Active FTP should work
- In addition, no OSPF, BGP, Ping and Traceroutes should be prohibited either inbound or outbound
- Ping uses ICMP message types “echo” and “echo-reply”
- Tracroute (the UNIX variant, which IOS utilizes) uses by default UDP port range 33434 – 33464 to probe the network, and expects ICMP packets of type either “Time-Exceeded” or “Port-Unreachable” in response
- BGP uses port 179 to establish it’s connection and OSPF has IP protocol number 89
- Active FTP uses TCP port 21 for inbound connections and port 20 to make outbound connections (server to client)
- Passive FTP client opens data connections inbound to ports in range 123 – 65535 (client to server)
- Create extended access-list FROM_OUTSIDE on R4 to permit all connections mentioned beforehand
- Add “deny ip any any log” at the end of access-list to log all denied packets
- Apply this access-list to both Serial and FR interfaces on ingress

Final Configuration

```
SW1:
ip access-list extended FROM_OUTSIDE
 remark ==
 remark == Pings
 remark ==
 permit icmp any any echo
 permit icmp any any echo-reply
 remark ==
 remark == Inbound Traceroute
 remark ==
 permit udp any any range 33434 33464
 remark ==
 remark == Backscatter ICMP messages for outbound Traceroute
 remark ==
 permit icmp any any time-exceeded
 permit icmp any any port-unreachable
 remark ==
 remark == BGP Inbound/Outbound
 remark ==
 permit tcp host 150.1.5.5 eq bgp host 150.1.4.4
 permit tcp host 150.1.5.5 host 150.1.4.4 eq bgp
 remark ==
 remark == OSPF packets
 remark ==
 permit ospf any any
 remark ==
 remark == Active FTP
 remark ==
 permit tcp any host 150.1.4.1 range 20 21
 permit tcp any host 150.1.4.6 range 20 21
 remark ==
 remark == Passive FTP data connections
 remark ==
 permit tcp any host 150.1.4.1 range 1023 65535
 permit tcp any host 150.1.4.6 range 1023 65535
 remark ==
 remark == WWW
 remark ==
 permit tcp any host 150.1.4.1 eq 80
 permit tcp any host 150.1.4.6 eq 80
 remark ==
 remark == WWW
 remark ==
 deny ip any any log

interface Serial 0/1
 ip access-group FROM_OUTSIDE in
!
interface Serial 0/0.1
 ip access-group FROM_OUTSIDE in
```

Verification

```
R4#show ip access-lists
Extended IP access list FROM_OUTSIDE
    10 permit icmp any any echo
    20 permit icmp any any echo-reply
    30 permit udp any any range 33434 33464
```



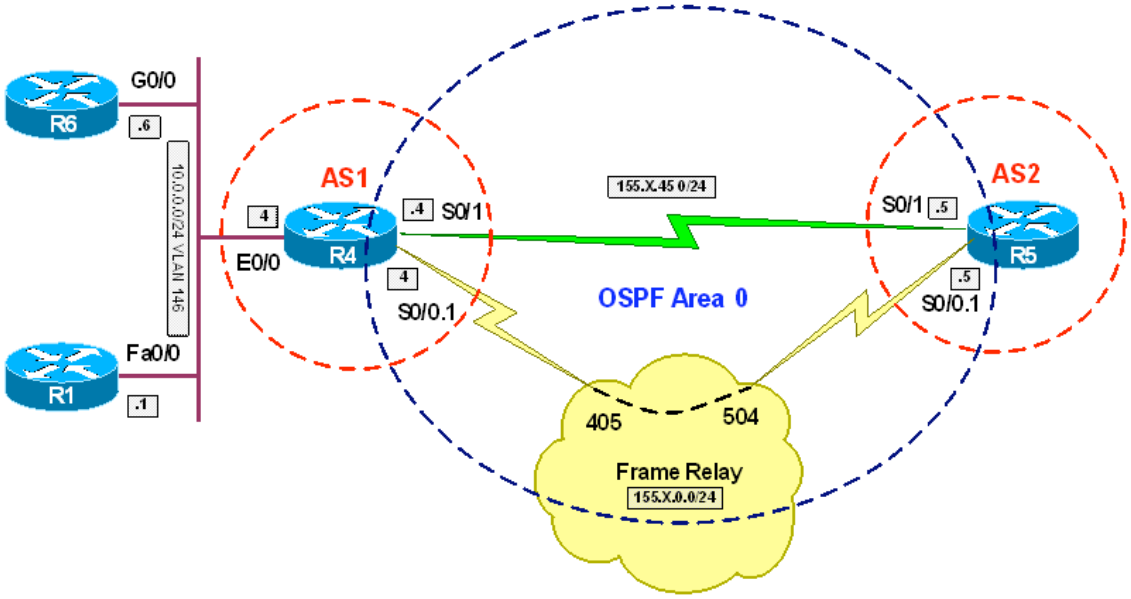
```
    40 permit icmp any any time-exceeded
    50 permit icmp any any port-unreachable
    60 permit tcp host 150.1.5.5 eq bgp host 150.1.4.4 (42 matches)
    70 permit tcp host 150.1.5.5 host 150.1.4.4 eq bgp
    80 permit ospf any any (80 matches)
    90 permit tcp any host 150.1.4.1 range ftp-data ftp
    100 permit tcp any host 150.1.4.6 range ftp-data ftp
    110 permit tcp any host 150.1.4.1 range 1023 65535
    120 permit tcp any host 150.1.4.6 range 1023 65535
    130 permit tcp any host 150.1.4.1 eq www
    140 permit tcp any host 150.1.4.6 eq www
    150 deny ip any any log

R5#ping 150.1.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/31/32 ms

R5#ping 150.1.4.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/44/44 ms

R5#ping 150.1.4.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/44/44 ms
R5#

R4#ping 150.1.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/45/48 ms

R1#traceroute 150.1.5.5

Type escape sequence to abort.
Tracing the route to 150.1.5.5

  1 10.0.0.4 4 msec 0 msec 4 msec
  2 155.1.0.5 24 msec *  20 msec
R1#

R5#traceroute 150.1.4.1

Type escape sequence to abort.
Tracing the route to 150.1.4.1

  1 155.1.45.4 16 msec
    155.1.0.4 24 msec *
R5#

R1#copy running-config flash:test.txt
Destination filename [test.txt]?
```



```
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0xC5CD)
910 bytes copied in 6.647 secs (137 bytes/sec)

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip http server
R1(config)#ftp-server enable
R1(config)#ftp-server topdir flash:

R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#no ip ftp passive
R5#copy ftp://150.1.4.1/test.txt null:
Accessing ftp://150.1.4.1/test.txt...
Loading test.txt !
[OK - 910/4096 bytes]

910 bytes copied in 2.560 secs (355 bytes/sec)

R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#ip ftp passive
R5(config)#do copy ftp://150.1.4.1/test.txt null:
Accessing ftp://150.1.4.1/test.txt...
Loading test.txt !
[OK - 910/4096 bytes]

910 bytes copied in 2.584 secs (352 bytes/sec)
R5(config)#

R5#telnet 150.1.4.1 80
Trying 150.1.4.1, 80 ... Open

R5#disc 1
Closing connection to 150.1.4.1 [confirm]

R5#telnet 150.1.4.1
Trying 150.1.4.1 ...
% Destination unreachable; gateway or host down

R4#show ip access-lists
Extended IP access list FROM_OUTSIDE
    10 permit icmp any any echo (12 matches)
    20 permit icmp any any echo-reply (15 matches)
    30 permit udp any any range 33434 33464 (3 matches)
    40 permit icmp any any time-exceeded
    50 permit icmp any any port-unreachable (6 matches)
    60 permit tcp host 150.1.5.5 eq bgp host 150.1.4.4 (135 matches)
    70 permit tcp host 150.1.5.5 host 150.1.4.4 eq bgp
    80 permit ospf any any (340 matches)
    90 permit tcp any host 150.1.4.1 range ftp-data ftp (376 matches)
    100 permit tcp any host 150.1.4.6 range ftp-data ftp
    110 permit tcp any host 150.1.4.1 range 1023 65535 (28 matches)
    120 permit tcp any host 150.1.4.6 range 1023 65535
    130 permit tcp any host 150.1.4.1 eq www (14 matches)
    140 permit tcp any host 150.1.4.6 eq www
    150 deny ip any any log (1 match)
```


Traffic Filtering with Reflexive Access-Lists

Objective: Configure router to use access-list technique that creates ACL entries for returning traffic dynamically



Directions

- The task is to permit inside hosts to access outside resources via telnet and HTTP. In addition, pings should be also permitted
- Configure routers as per the NAT scenario "Standard NAT with Overloading (PAT)"
- Create access-list OUTBOUND and permit outgoint TCP connections to port numbers 80 and 23. This connections should be reflected in access-list MIRROR
- Additionally, permit outbound ICMP of type echo and reflect it into access-list MIRROR
- Create access-list INBOUND. Evaluate access-list MIRROR in the beginning, and additionally permit OSPF traffic. Deny and log all other traffic
- Apply access-list OUTBOUND and INBOUND to both Serial and Frame-Relay interfaces, as egress and ingress filters respectively
- Note that router's traffic by default is not subject to reflective ACL inspections

Final Configuration

R4:

```
ip access-list extended OUTBOUND
 permit tcp any any eq 23 reflect MIRROR
 permit tcp any any eq 80 reflect MIRROR
 permit icmp any any echo reflect MIRROR
```



```
!
ip access-list extended INBOUND
 evaluate MIRROR
 permit ospf any any
 deny ip any any log
!
interface Serial 0/1
 ip access-group INBOUND in
 ip access-group OUTBOUND out
!
interface Serial 0/0.1
 ip access-group INBOUND in
 ip access-group OUTBOUND out
```

Verification

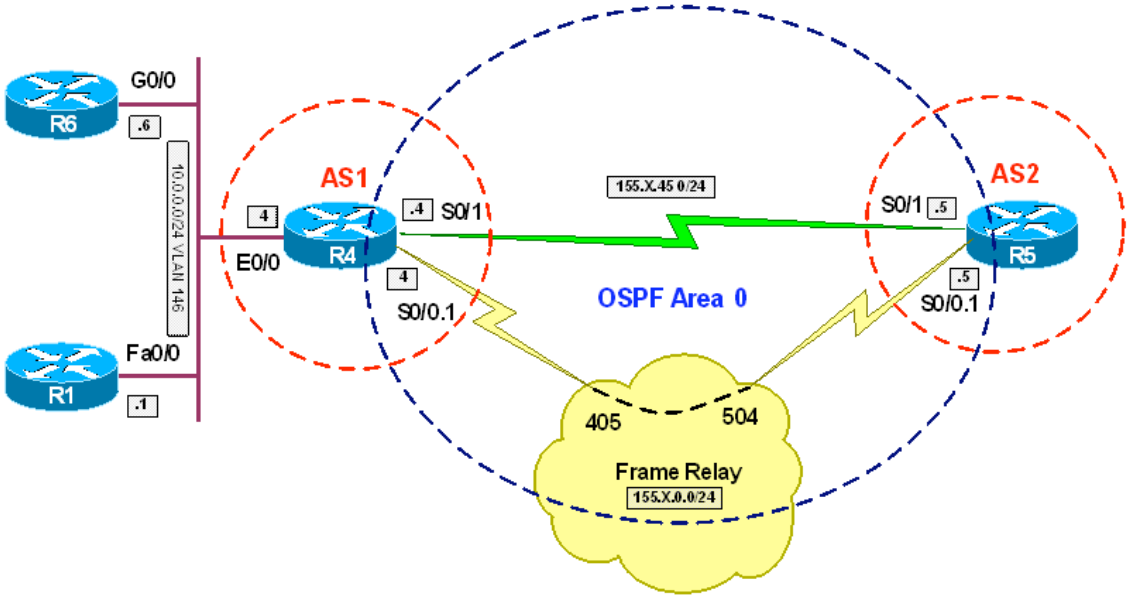
```
R1#telnet 150.1.5.5
Trying 150.1.5.5 ... Open

R5>

R4#show ip access MIRROR
Reflexive IP access list MIRROR
     permit tcp host 150.1.5.5 eq telnet host 150.1.4.4 eq 43992 (33 matches)
(time left 295)

R6#ping 150.1.5.5 size 1500 repeat 10

Type escape sequence to abort.
Sending 10, 1500-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
.!!!
Rack1AS>4
[Resuming connection 4 to r4 ... ]

R4#show ip acce MIRROR
Reflexive IP access list MIRROR
     permit icmp host 150.1.5.5 host 150.1.4.4  (29 matches) (time left 299)
     permit tcp host 150.1.5.5 eq telnet host 150.1.4.4 eq 43992 (33 matches)
(time left 252)

R4#telnet 150.1.5.5
Trying 150.1.5.5 ...
%SEC-6-IPACCESSLOGP: list INBOUND denied tcp 150.1.5.5(23) ->
155.1.45.4(21042), 1 packet
% Connection timed out; remote host not responding

R4#show ip bgp summary
BGP router identifier 150.1.4.4, local AS number 1
BGP table version is 3, main routing table version 3

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/PfxRcd
150.1.5.5       4     2      29      32        0    0    0 00:01:00 Active

R4#ping 150.1.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```



```
%SEC-6-IPACCESSLOGDP: list INBOUND denied icmp 150.1.5.5 -> 155.1.45.4 (0/0), 1
packet
%SEC-6-IPACCESSLOGP: list INBOUND denied tcp 150.1.5.5(179) ->
150.1.4.4(15332), 1 packet
```


Reflexive Access-Lists and Router-Generated Traffic

Objective: Configure router so that reflexive access-lists may evaluate router-generated traffic



Directions

- Configure routers as per the NAT scenario “Traffic Filtering with Reflexive Access-Lists”
- Add a line to access-list OUTBOUND to permit and reflect oubound connections to port 179
- Create extended access-list LOCAL_TRAFFIC and match TCP/ICMP traffic from any to any
- Create route-map LOCAL_POLICY; match access-list LOCAL_TRAFFIC and set interface Loopback0
- Apply route-map LOCAL_POLICY as local policy

Final Configuration

R4:

```
ip access-list ext OUTBOUND
 permit tcp any any eq 179 reflect MIRROR
!
ip access-list extended LOCAL_TRAFFIC
 permit tcp any any
 permit icmp any any
!
route-map LOCAL_POLICY
 match ip address LOCAL_TRAFFIC
 set interface Loopback0
!
ip local policy route-map LOCAL_POLICY
```

Verification

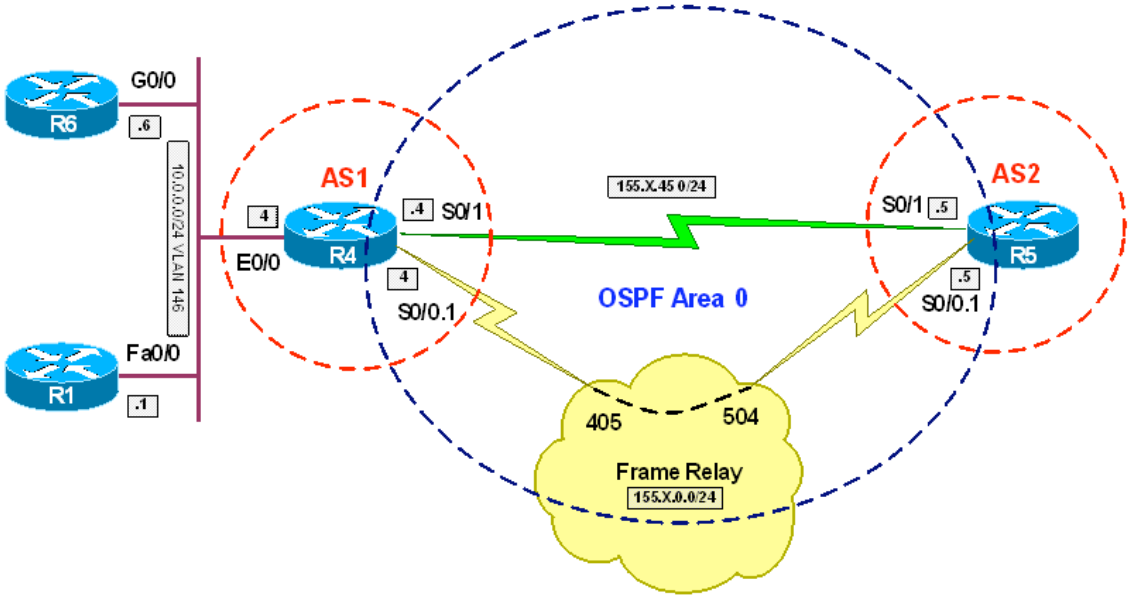
```
R4#ping 150.1.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/49/64 ms

R4#show ip access-lists MIRROR
Reflexive IP access list MIRROR
     permit icmp host 150.1.5.5 host 155.1.45.4  (8 matches) (time left 297)
     permit icmp host 150.1.5.5 host 155.1.0.4  (12 matches) (time left 297)

R4#telnet 150.1.5.5
Trying 150.1.5.5 ... Open

R5>exit

[Connection to 150.1.5.5 closed by foreign host]

R4#sh ip access-lists MIRROR
Reflexive IP access list MIRROR
     permit tcp host 150.1.5.5 eq telnet host 155.1.0.4 eq 47175 (78 matches)
(time left 1)
     permit icmp host 150.1.5.5 host 155.1.45.4  (8 matches) (time left 233)
     permit icmp host 150.1.5.5 host 155.1.0.4  (12 matches) (time left 233)

R4#sh ip bgp summary
BGP router identifier 150.1.4.4, local AS number 1
BGP table version is 2, main routing table version 2
1 network entries using 117 bytes of memory
1 path entries using 52 bytes of memory
2/1 BGP path/bestpath attribute entries using 248 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 441 total bytes of memory
BGP activity 2/1 prefixes, 2/1 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down State/PfxRcd
150.1.5.5       4     2      34      36        2    0    0 00:00:53        1
```



```
R4#show ip acce MIRROR
Reflexive IP access list MIRROR
     permit tcp host 150.1.5.5 eq bgp host 150.1.4.4 eq 34994 (21 matches)
(time left 241)

R4#show ip access-lists LOCAL_TRAFFIC
Extended IP access list LOCAL_TRAFFIC
    10 permit tcp any any (42 matches)
    20 permit icmp any any (29 matches)

R4#show route-map
route-map LOCAL_POLICY, permit, sequence 10
  Match clauses:
    ip address (access-lists): LOCAL_TRAFFIC
  Set clauses:
    interface Loopback0
  Policy routing matches: 70 packets, 4678 bytes
```

Configuring CBAC for Traffic Inspection

Objective: Configure router for outgoing traffic inspection and dynamic opening of ACL pinholes



Directions

- Configure routers as per the NAT scenario “Standard NAT with Overloading (PAT)”
- Create inspection rule named INSPECT to permit TCP based protocols.
- Additionally, permit FTP transactions to be performed through the firewall
- Next, configure this rule to permit ICMP and inspect router-generated TCP/ICMP traffic
- Create access-list INBOUND and permit OSPF with it. Block and log all other traffic
- Apply access-list INBOUND ingress on Serial and FR interfaces
- Apply inspection rules egress on Serial and FR interfaces

Final Configuration

R4:

```
ip inspect name INSPECT ftp
ip inspect name INSPECT icmp router-traffic
ip inspect name INSPECT tcp router-traffic
!
ip access-list ext INBOUND
 permit ospf any any
 deny ip any any log
!
interface Serial 0/1
 ip access-group INBOUND in
 ip inspect INSPECT out
!
interface Serial 0/0.1
```



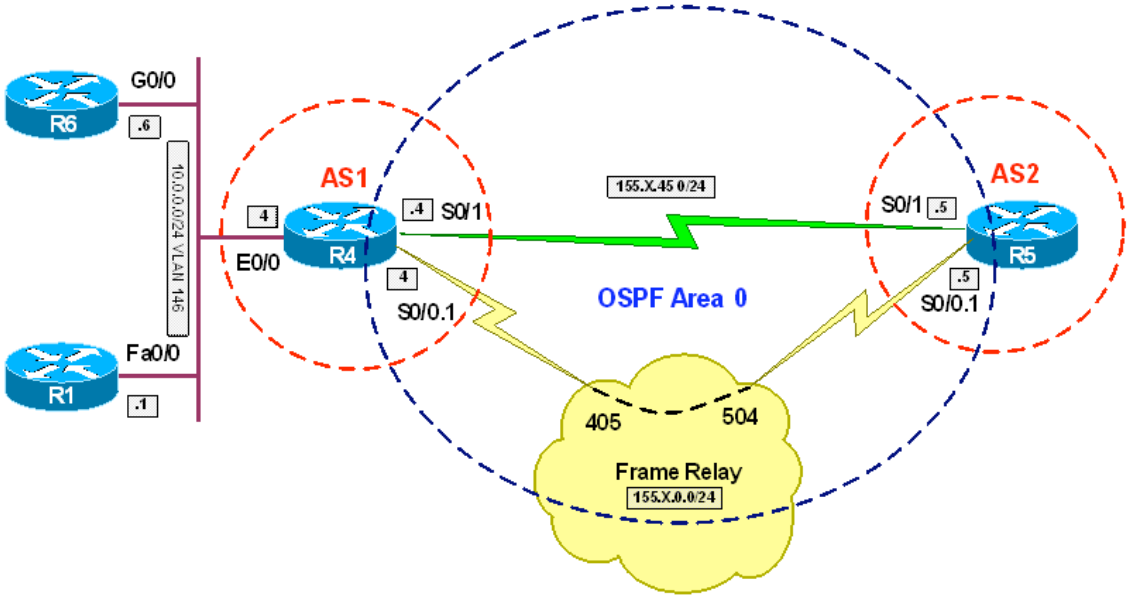
```
 ip access-group INBOUND in
 ip inspect INSPECT out
```

Verification

```
R4#show ip inspect config
Session audit trail is disabled
Session alert is enabled
one-minute (sampling period) thresholds are [400:500] connections
max-incomplete sessions thresholds are [400:500]
max-incomplete tcp connections per host is 50. Block-time 0 minute.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name INSPECT
    http alert is on audit-trail is off timeout 3600
    ftp alert is on audit-trail is off timeout 3600
    icmp alert is on audit-trail is off timeout 10
    telnet alert is on audit-trail is off timeout 3600
    router alert is on audit-trail is off timeout 30

R6#telnet 150.1.5.5
Trying 150.1.5.5 ... Open

R5>

R4#show ip inspect sessions
Established Sessions
 Session 650FF88C (10.0.0.6:54327)=>(150.1.5.5:23) tcp SIS_OPEN
 Session 650FFB04 (150.1.4.4:40087)=>(150.1.5.5:179) tcp SIS_OPEN

R4#ping 150.1.5.5

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 44/44/48 ms

R4#telnet 150.1.5.5
Trying 150.1.5.5 ... Open

R5>exit

[Connection to 150.1.5.5 closed by foreign host]

R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#ftp-server enable
R5(config)#ftp-server topdir flash:
R5(config)#do copy start flash:
Destination filename [r5-confg]? test.txt
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x10CB)
1668 bytes copied in 0.292 secs (5712 bytes/sec)

R6#copy ftp://150.1.5.5/test.txt null:
Accessing ftp://150.1.5.5/test.txt...
Loading test.txt !
[OK - 1668/4096 bytes]1668 bytes copied in 4.652 secs (359 bytes/sec)
```


Access Control with Dynamic ACLs (Lock & Key)

Objective: Configure router to authenticate users via telnet login session before granting them access to internal resources



Directions

- Configure routers as per the NAT scenario “Configuring Static NAT”
- The goal is to permit remote uses access the inside network and set the absolute and inactivity timeouts
- Configure VTY lines to authenticate incoming telnet sessions
- Create local username DYNACL with passwod CISCO
- Configure this user to have autocommand “**access-enable host timeout 5**”, therefore setting inactivity timeout to 5 minutes
- Create extended access-list INBOUND and permit OSPF, BGP and Telnet traffic with this list
- Create dynamic list entry ACCESS with timeout value of 10 (absolute timeout) and configure “**permit ip any any**” as dynamic rule
- Deny and log everything else with access-list INBOUND

Final Configuration

R4:

```
line vty 0 4
 login local
!
username DYNACL password CISCO
username DYNACL autocommand access-enable host timeout 5
!
ip access-list extended INBOUND
 permit ospf any any
 permit tcp any any eq bgp
 permit tcp any eq bgp any
 permit tcp any any eq telnet
 dynamic ACCESS timeout 10 permit ip any any
 deny ip any any log
!
interface Serial 0/1
 ip access-group INBOUND in
!
interface Serial 0/0.1
 ip access-group INBOUND in
```

Verification

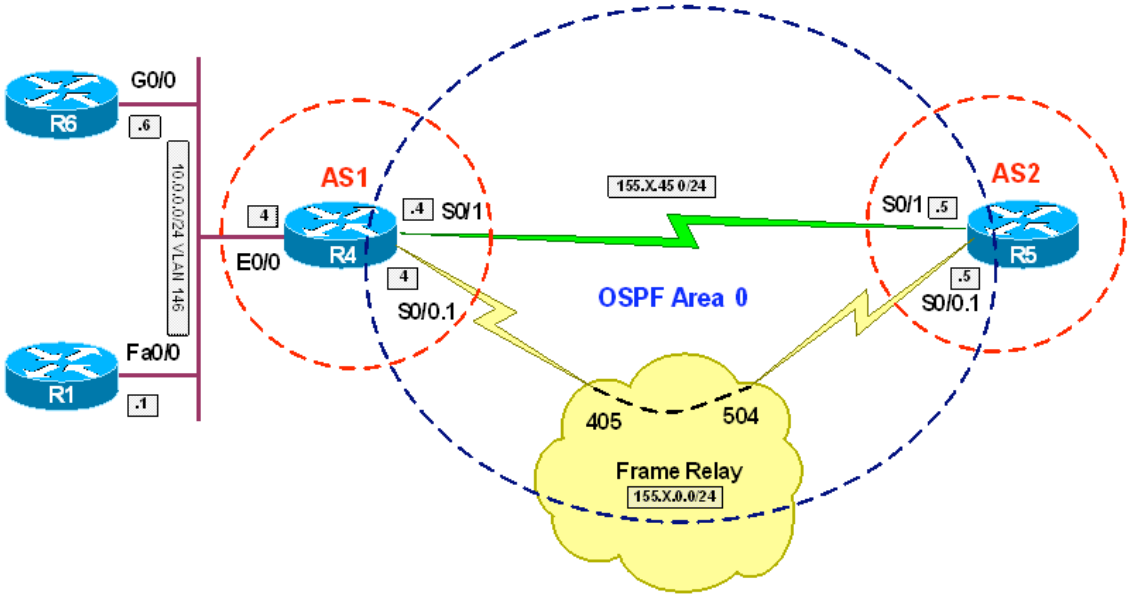
```
R5#ping 150.1.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.1, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

R5#ping 150.1.4.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
U.U.U
Success rate is 0 percent (0/5)

R5#telnet 150.1.4.4
Trying 150.1.4.4 ... Open


User Access Verification

Username: DYNACL
Password: CISCO
[Connection to 150.1.4.4 closed by foreign host]

R5#

R4#show ip acce
Extended IP access list INBOUND
    10 permit ospf any any (11 matches)
    20 permit tcp any any eq bgp
    30 permit tcp any eq bgp any (6 matches)
    40 permit tcp any any eq telnet (99 matches)
    50 Dynamic ACCESS permit ip any any
```



```
       permit ip host 155.1.45.5 any
    60 deny ip any any log (10 matches)
R4#

R5#ping 150.1.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/31/32 ms

R5#ping 150.1.4.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/44/44 ms

R4#show ip access-lists
Extended IP access list INBOUND
    10 permit ospf any any (49 matches)
    20 permit tcp any any eq bgp
    30 permit tcp any eq bgp any (24 matches)
    40 permit tcp any any eq telnet (99 matches)
    50 Dynamic ACCESS permit ip any any
       permit ip host 155.1.45.5 any (12 matches) (time left 132)
    60 deny ip any any log (10 matches)
```


Using NBAR to Filter Traffic

Objective: Configure router to filter traffic based on application-level criteria



Directions

- Configure routers as per the NAT scenario “Standard NAT with Overloading (PAT)”
- Make sure CEF is enabled globally on R4
- Create a map-class IMAGES on R4 on match any of HTTP URLs that are retrieving an image file (.gif, .jpeg, .jpg)
- Create policy-map DROP_IMAGES and configure it to drop any traffic in class IMAGES
- Apply the policy-map DROP_IMAGES ingress to the interfaces Serial 0/1 and Serial 0/0.1

Final Configuration

R4:

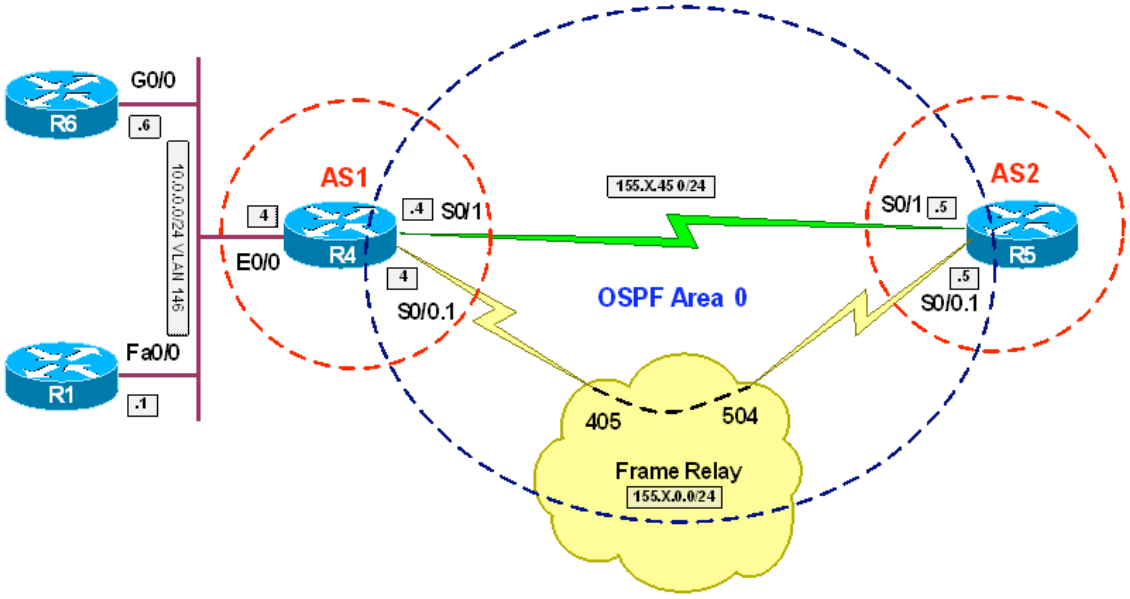
```
ip cef
class-map match-any IMAGES
 match protocol http url “*.gif”
 match protocol http url “*.jpeg|*.jpg”
!
policy-map DROP_IMAGES
 class IMAGES
   drop
!
interface Serial 0/1
 service-policy input DROP_IMAGES
!
interface Serial 0/0.1
 service-policy input DROP_IMAGES
```


Verification

```
R5#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R5(config)#ip http server
R5(config)#ip http path flash:
R5(config)#do copy start flash:test.gif
Destination filename [test.gif]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x10CB)
1668 bytes copied in 0.288 secs (5792 bytes/sec)
R5(config)#do copy start flash:test.jpg
Destination filename [test.jpg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x10CB)
1668 bytes copied in 0.300 secs (5560 bytes/sec)
R5(config)#do copy start flash:test.jpeg
Destination filename [test.jpeg]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x10CB)
1668 bytes copied in 0.288 secs (5792 bytes/sec)
R5(config)#do copy start flash:test.txt
Destination filename [test.txt]?
Erase flash: before copying? [confirm]n
Verifying checksum...  OK (0x10CB)
1668 bytes copied in 0.294 secs (5670 bytes/sec)

R1#copy http://150.1.5.5/test.txt null:
Loading http://150.1.5.5/test.txt !
1668 bytes copied in 2.496 secs (668 bytes/sec)

R1#copy http://150.1.5.5/test.gif null:
%Error opening http://150.1.5.5/test.gif (I/O error)

R1#copy http://150.1.5.5/test.jpeg null:
%Error opening http://150.1.5.5/test.jpeg (I/O error)

R1#copy http://150.1.5.5/test.jpg null:
%Error opening http://150.1.5.5/test.jpg (I/O error)

R4#show policy-map interface serial 0/0.1

 Serial0/0.1

  Service-policy input: DROP_IMAGES

    Class-map: IMAGES (match-any)
      24 packets, 4971 bytes
      5 minute offered rate 0 bps, drop rate 0 bps
      Match: protocol http url "*.jpeg|*.jpg"
        16 packets, 3314 bytes
        5 minute rate 0 bps
      Match: protocol http url "*.gif"
        8 packets, 1657 bytes
        5 minute rate 0 bps
      drop

    Class-map: class-default (match-any)
      70 packets, 7822 bytes
      5 minute offered rate 0 bps, drop rate 0 bps      Match: any
```


Using Policy-Based Routing to Filter Traffic

Objective: Configure router to filter traffic based on packet lengths



Directions

- Configure routers per the NAT scenario “Configuring Static NAT”
- The task is to permit small ICMP echo packets with L3 length up to 300 bytes
- Create extended access-list ICMP_ECHO and match ICMP echo packets
- Create route-map ICMP_CONTROL; with section 10 permit packets matching the access-list ICMP and having length 301-1500. Route this packets to Null0 interface
- Apply this route-map to Serial and FR interfaces

Final Configuration

R4:

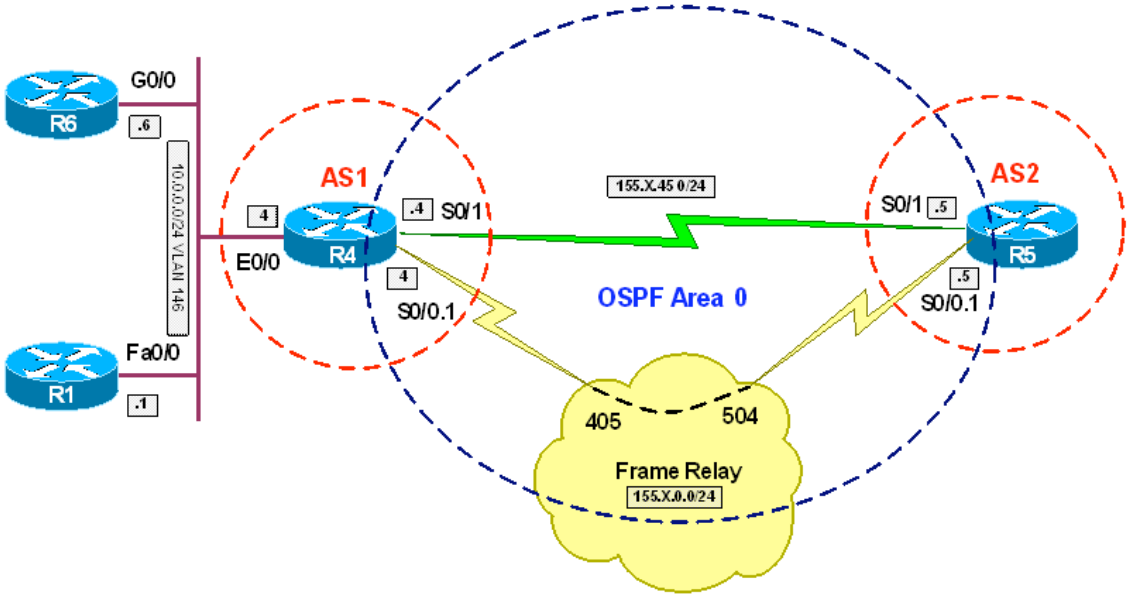
```
ip access-list extended ICMP_ECHO
 permit icmp any any echo
!
route-map ICMP_CONTROL permit 10
 match ip address ICMP_ECHO
 match length 301 1500
 set interface Null0
!
interface Serial 0/1
 ip policy route-map ICMP_CONTROL
!
interface Serial 0/0.1
 ip policy route-map ICMP_CONTROL
```


Verification

```
R4#show route-map
route-map ICMP_CONTROL, permit, sequence 10
  Match clauses:
    ip address (access-lists): ICMP_ECHO
    length 301 1500
  Set clauses:
    interface Null0
  Policy routing matches: 0 packets, 0 bytes

R5#ping 150.1.4.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 28/30/32 ms

R5#ping 150.1.4.6

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 44/44/44 ms

R5#ping 150.1.4.6 size 301

Type escape sequence to abort.
Sending 5, 301-byte ICMP Echos to 150.1.4.6, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R5#ping 150.1.4.1 size 301

Type escape sequence to abort.
Sending 5, 301-byte ICMP Echos to 150.1.4.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

R4#show route-map
route-map ICMP_CONTROL, permit, sequence 10
  Match clauses:
    ip address (access-lists): ICMP_ECHO
    length 301 1500
  Set clauses:
    interface Null0
  Policy routing matches: 10 packets, 3050 bytes
```


DoS Attacks Prevention with TCP Intercept

Objective: Configure router to intercept all TCP connections to Web Servers



Directions

- Configure routers as per the NAT scenario “Common Configuration”
- Create access-list 199 and match TCP connections to network 150.1.4.0/24 on port 80
- Configure TCP intercept to use list 199, and random drop mode
- Start clamping half-open sessions when their number reaches 1500
- Stop clamping half-open sessions when their number reaches 1200
- Set inactive connection timeout to one hour

Final Configuration

```
R4:
access-list 199 permit tcp any 150.1.4.0 0.0.0.255 eq 80
!
ip tcp intercept list 199
ip tcp intercept max-incomplete high 1500
ip tcp intercept max-incomplete low 1200
ip tcp intercept connection-timeout 3600
ip tcp intercept drop-mode random
```

Verification

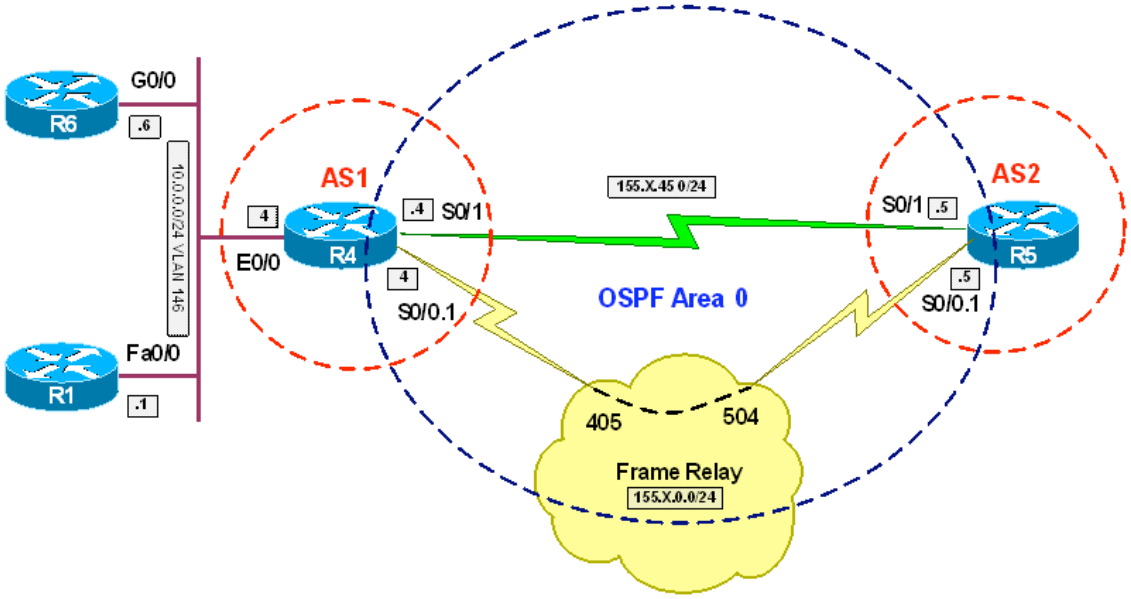
```
R4#debug ip tcp intercept
TCP intercept debugging is on

R5#telnet 150.1.4.100 80
Trying 150.1.4.100, 80 ... Open
```



```
[Connection to 150.1.4.100 closed by foreign host]
R5#

R4#
*Mar 19 21:29:42.391: INTERCEPT: new connection (155.1.45.5:20959 SYN ->
150.1.4.100:80)
*Mar 19 21:29:42.395: INTERCEPT(*): (155.1.45.5:20959 <- ACK+SYN
150.1.4.100:80)
*Mar 19 21:29:42.415: INTERCEPT: 1st half of connection is established
(155.1.45.5:20959 ACK -> 150.1.4.100:80)
*Mar 19 21:29:42.419: INTERCEPT(*): (155.1.45.5:20959 SYN -> 150.1.4.100:80)
*Mar 19 21:29:42.419: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:29:42.423: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:29:43.415: INTERCEPT(*): SYNSENT retransmit 1 (155.1.45.5:20959 SYN
-> 150.1.4.100:80)
*Mar 19 21:29:43.415: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:29:45.415: INTERCEPT(*): SYNSENT retransmit 2 (155.1.45.5:20959 SYN
-> 150.1.4.100:80)
*Mar 19 21:29:45.415: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:29:49.415: INTERCEPT(*): SYNSENT retransmit 3 (155.1.45.5:20959 SYN
-> 150.1.4.100:80)
*Mar 19 21:29:49.415: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:29:57.415: INTERCEPT(*): SYNSENT retransmit 4 (155.1.45.5:20959 SYN
-> 150.1.4.100:80)
*Mar 19 21:29:57.415: INTERCEPT: client packet dropped in SYNSENT
(155.1.45.5:20959 -> 150.1.4.100:80)
*Mar 19 21:30:13.415: INTERCEPT: SYNSENT retransmitting too long
(155.1.45.5:20959 <-> 150.1.4.100:80)
*Mar 19 21:30:13.415: INTERCEPT(*): (155.1.45.5:20959 <- RST 150.1.4.100:80)
```


Configuring TCP Intercept in Watch Mode

Objective: Configure router to monitor incoming TCP connections in order to prevent possible DoS attack



Directions

- Configure routers as per the NAT scenario “Common Configuration”
- Create access-list 199 and match TCP connections to network 150.1.4.0/24 on port 80
- Configure TCP intercept to use list 199, and random drop mode
- Configure TCP intercept not to proxy the incoming connections. However, router should reset connections that linger in half-open state for more than 15 seconds
- Start resetting half-open sessions when their number reaches 1500
- Stop resetting half-open sessions when their number reaches 1200

Final Configuration

R4:

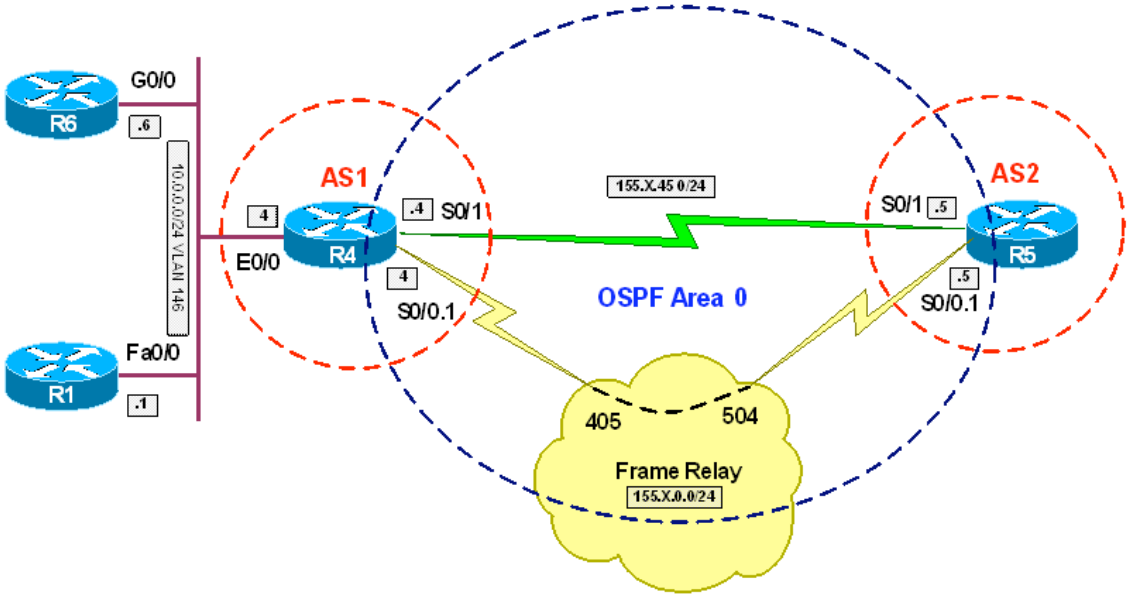
```
access-list 199 permit tcp any 150.1.4.0 0.0.0.255 eq 80
!
ip tcp intercept list 199
ip tcp intercept mode watch
ip tcp intercept watch-timeout 15
ip tcp intercept max-incomplete high 1500
ip tcp intercept max-incomplete low 1200
ip tcp intercept connection-timeout 3600
ip tcp intercept drop-mode random
```


Verification

```
R4#debug ip tcp intercept
TCP intercept debugging is on

R5#telnet 150.1.4.1 80
Trying 150.1.4.1, 80 ...
% Connection timed out; remote host not responding

R4#show logging
*Mar 20 13:09:49.087: INTERCEPT: new connection (155.1.45.5:33678 SYN ->
150.1.4.1:80)
*Mar 20 13:09:49.091: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:09:51.087: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:09:51.091: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:09:55.087: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:09:55.091: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:10:03.087: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:10:03.091: INTERCEPT: client packet passed in SYNSENT
(155.1.45.5:33678 -> 150.1.4.1:80)
*Mar 20 13:10:04.087: INTERCEPT: SYNSENT timing out (155.1.45.5:33678 <->
150.1.4.1:80)
*Mar 20 13:10:04.087: INTERCEPT(*): (155.1.45.5:33678 RST -> 150.1.4.1:80)
```


DoS Attacks Prevention with CBAC

Objective: Configure CBAC to defend against SYN-Flooding Attacks



Directions

- Configure routers as per the NAT scenario “Common Configuration”
- Advertise network 10.0.0.0/24 into OSPF on R4
- Create inspection rule DOS_MITIGATION and inspect TCP traffic
- Configure CBAC to start claming half-open sessions when their number reaches 1200 and stop on 1000 sessions
- Configure CBAC to start replacing half-open sessions when their rate exceeds 300 per minute and stop when it falls below 100
- Configure CBAC to block any host for 5 minutes when it has more then 50 half-open sessions

Final Configuration

R4:

```
ip inspect max-incomplete high 1200
ip inspect max-incomplete low 1000
ip inspect one-minute low 200
ip inspect one-minute high 300
ip inspect tcp max-incomplete host 50 block-time 5
!
ip inspect name DOS_MITIGATION tcp
!
interface Ethernet 0/0
 ip inspect DOS_MITIGATION out
!
router ospf 1
 redistribute connected subnets
```


Verification

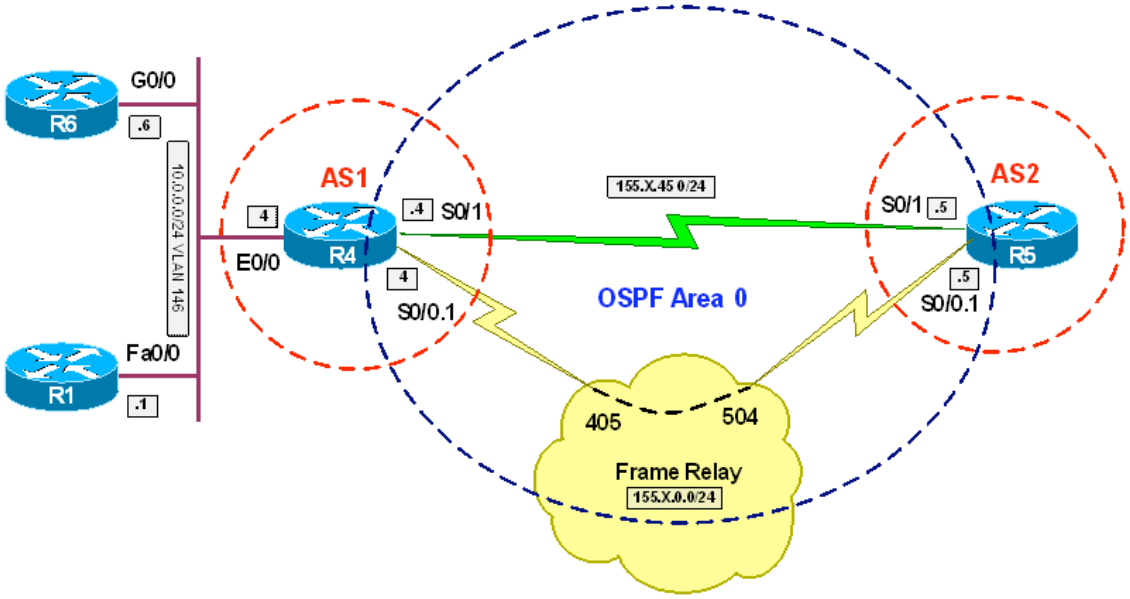
```
R4#show ip inspect all
Session audit trail is disabled
Session alert is enabled
one-minute (sampling period) thresholds are [200:300] connections
max-incomplete sessions thresholds are [1000:1200]
max-incomplete tcp connections per host is 50. Block-time 5 minutes.
tcp synwait-time is 30 sec -- tcp finwait-time is 5 sec
tcp idle-time is 3600 sec -- udp idle-time is 30 sec
dns-timeout is 5 sec
Inspection Rule Configuration
 Inspection name DOS_MITIGATION
    tcp alert is on audit-trail is off timeout 3600

Interface Configuration
 Interface Ethernet0/0
  Inbound inspection rule is not set
  Outgoing inspection rule is DOS_MITIGATION
    tcp alert is on audit-trail is off timeout 3600
  Inbound access list is not set
  Outgoing access list is not set

R4#debug ip inspect tcp
INSPECT TCP Inspection debugging is on

R4#debug ip inspect event
INSPECT special events debugging is on

R5#debug ip tcp transactions
TCP special event debugging is on

R1#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#int fa 0/0
R1(config-if)#shut

R5#telnet 10.0.0.1
Trying 10.0.0.1 ...
*Mar 20 15:45:27.171: TCP: Random local port generated 64972
*Mar 20 15:45:27.171: TCB65C81554 created
*Mar 20 15:45:27.171: TCB65C81554 setting property TCP_TOS (11) 65650030
*Mar 20 15:45:27.171: TCB65C81554 bound to UNKNOWN.64972
*Mar 20 15:45:27.171: TCP: sending SYN, seq 2073350221, ack 0
*Mar 20 15:45:27.171: TCP0: Connection to 10.0.0.1:23, advertising MSS 536
*Mar 20 15:45:27.175: TCP0: state was CLOSED -> SYNSENT [64972 -> 10.0.0.1(23)]
*Mar 20 15:45:29.175: 155.1.45.5:64972 <---> 10.0.0.1:23   congestion window
changes
*Mar 20 15:45:29.175: cwnd from 536 to 536, ssthresh from 65535 to 1072
*Mar 20 15:45:29.175: TCP0: timeout #1 - timeout is 4000 ms, seq 2073350221
*Mar 20 15:45:33.175: TCP0: timeout #2 - timeout is 8000 ms, seq 2073350221
*Mar 20 15:45:41.175: TCP0: timeout #3 - timeout is 16000 ms, seq 2073350221
*Mar 20 15:45:57.175: TCP0: timeout #4 - timeout is 29996 ms, seq 2073350221
*Mar 20 15:45:57.195: TCP0: bad seg from 10.0.0.1 -- Rst bit set.: port 64972
seq 0 ack 0 rcvnxt 0 rcvwnd 4128 len 0

R4#
CBAC sis 65086504 pak 64B21D7C SIS_CLOSED/LISTEN TCP SYN SEQ 2073350221 LEN 0
(155.1.45.5:64972) => (10.0.0.1:23)
CBAC sis 65086504 pak 64B23264 SIS_OPENING/SYNSENT TCP SYN SEQ 2073350221 LEN 0
(155.1.45.5:64972) => (10.0.0.1:23)
```



```
CBAC sis 65086504 L4 inspect result: SKIP packet 64B23264 (155.1.45.5:64972)
(10.0.0.1:23) bytes 0 ErrStr = Retransmitted Segment tcpCBAC sis 65086504 pak
650025A4 SIS_OPENING/SYNSENT TCP SYN SEQ 2073350221 LEN 0 (155.1.45.5:64972) =>
(10.0.0.1:23)
CBAC sis 65086504 L4 inspect result: SKIP packet 650025A4 (155.1.45.5:64972)
(10.0.0.1:23) bytes 0 ErrStr = Retransmitted Segment tcp
CBAC sis 65086504 pak 64B1FAA4 SIS_OPENING/SYNSENT TCP SYN SEQ 2073350221 LEN 0
(155.1.45.5:64972) => (10.0.0.1:23)
CBAC sis 65086504 L4 inspect result: SKIP packet 64B1FAA4 (155.1.45.5:64972)
(10.0.0.1:23) bytes 0 ErrStr = Retransmitted Segment tcp
CBAC sent a TCP pkt (10.0.0.1:23) tcp flag:0x4 -> 155.1.45.5:64972 seq 0 ack 0
wnd 4128
CBAC sent a TCP pkt (155.1.45.5:64972) tcp flag:0x4 -> 10.0.0.1:23 seq
2073350222 ack 0 wnd 0
CBAC sis 65086504 pak 6500724C SIS_CLOSED/LISTEN TCP SYN SEQ 2073350221 LEN 0
(155.1.45.5:64972) => (10.0.0.1:23)

CBAC sent a TCP pkt (10.0.0.1:23) tcp flag:0x4 -> 155.1.45.5:64972 seq 0 ack 0
wnd 4128
CBAC sent a TCP pkt (155.1.45.5:64972) tcp flag:0x4 -> 10.0.0.1:23 seq
2073350222 ack 0 wnd 0
```


Configuring Application Port-Mapping with CBAC

Objective: Configure router so that CBAC recognizes common application on non-standard port



Directions

- Configure routers per the NAT scenario “Configuring Static PAT”
- Create CBAC inspection rule named INSPECT_TELNET on R4 and configure it to inspect telnet protocol
- Create access-list 99 on R4 and match network 10.0.0.0/24 with it
- Map ports 1023 and 6023 as telnet ports for networks in list 99
- Apply inspection rule INSPECT_TELNET inbound on Serail and FR interfaces
- Create extended access-list INSIDE and deny everything with it
- Apply access-list INSIDE inbound to Ethernet interface on R4

Final Configuration

R4:

```
access-list 99 permit 10.0.0.0 0.0.0.255
!
ip inspect name INSPECT_TELNET telnet
ip port-map telnet port tcp 1023 list 99
ip port-map telnet port tcp 6023 list 99
!
interface Serial 0/1
 ip inspect INSPECT_TELNET in
!
interface Serial 0/0.1
 ip inspect INSPECT_TELNET in
!
ip access-list extended INSIDE
 deny   ip any any
```



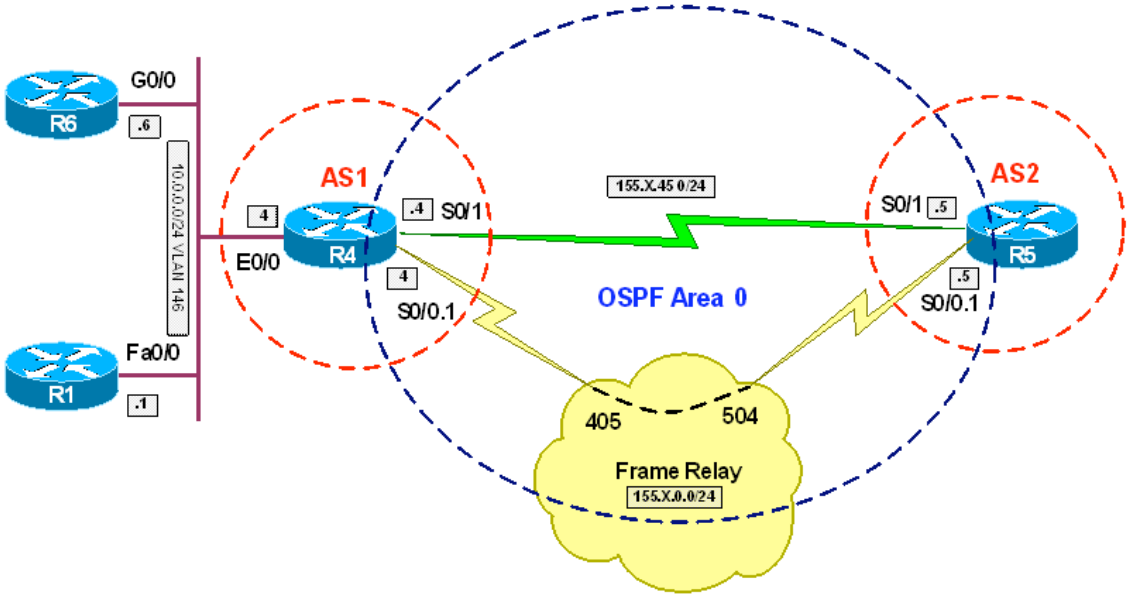
```
!
interface Ethernet 0/0
 ip access-group INSIDE in
```

Verification

```
R5#telnet 150.1.4.4 1023
Trying 150.1.4.4, 1023 ... Open

R1>

R4#show ip inspect sessions
Established Sessions
 Session 6507BE4C (155.1.45.5:17996)=>(10.0.0.1:23) telnet SIS_OPEN

R5#telnet 150.1.4.4 6023
Trying 150.1.4.4, 6023 ... Open

R6>

R4#show ip inspect sessions
Established Sessions
 Session 6507C0C4 (155.1.45.5:59812)=>(10.0.0.6:23) telnet SIS_OPEN
```


Using CAR for Smurf Attack Mitigation

Objective: Configure router to protect inside network from Smurf attack



Directions

- Configure routers as per the NAT scenario “Configuring Static NAT”
- Smurf attack is performed reflecting ICMP echo packets; Therefore, target network is bombed with a heavy flow of ICMP echo-reply packets
- Create extended access-list 100 on R4 and match ICMP echo-reply packets with it
- Create rate-limit rule to limit traffic matching access-list 100 to 64Kbps.
- Select Bc and Be values to accommodate sustained bursts length of 1 sec and 1.5 sec of traffic-rate:
  Bc=64000*1/8=8000 bytes
  and
  Be=Be*1.5=12000 bytes.
- This rate-limiting rules should be applied to Serial and FR interfaces

Final Configuration

R4:

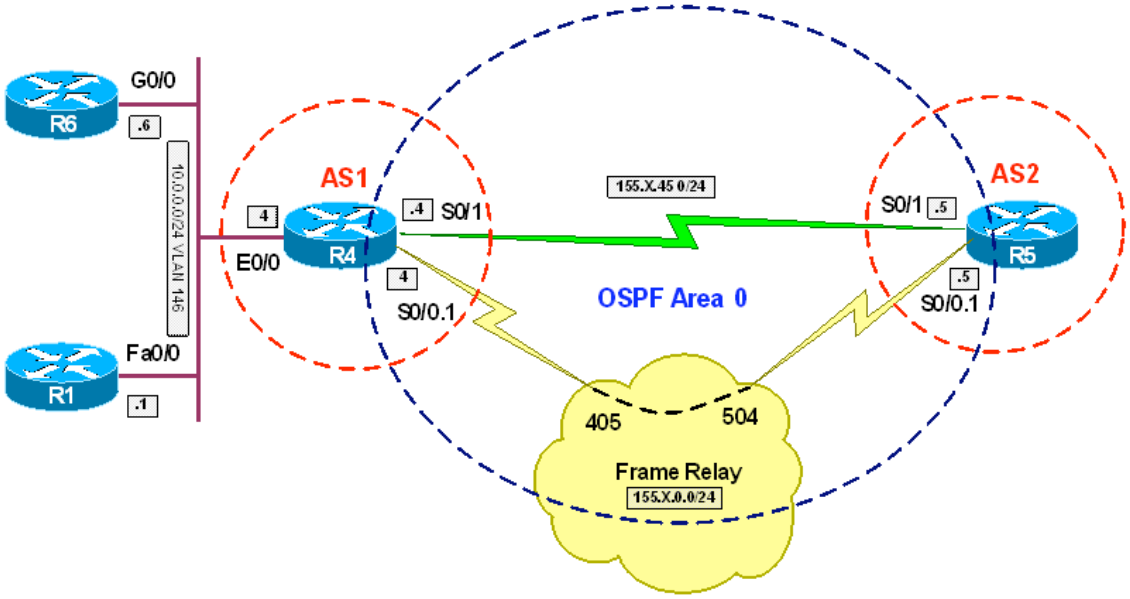
```
access-list 100 permit icmp any any echo-reply
!
interface Serial 0/1
 rate-limit input access-group 100 64000 8000 12000 conf tr exceed drop
!
interface Serial 0/0.1
 rate-limit input access-group 100 64000 8000 12000 conf tr exceed drop
```

Verification


```
R4#ping 150.1.5.5 size 500 repeat 100

Type escape sequence to abort.
Sending 100, 500-byte ICMP Echos to 150.1.5.5, timeout is 2 seconds:
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!
Success rate is 100 percent (100/100), round-trip min/avg/max = 192/195/208 ms

R4#show int se 0/1 rate
Serial0/1
  Input
    matches: access-group 100
      params:  64000 bps, 8000 limit, 12000 extended limit
      conformed 100 packets, 50400 bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: drop
      last packet: 134524ms ago, current burst: 0 bytes
      last cleared 00:03:26 ago, conformed 1000 bps, exceeded 0 bps

R4#show int se 0/0.1 rate
Serial0/0.1
  Input
    matches: access-group 100
      params:  64000 bps, 8000 limit, 12000 extended limit
      conformed 0 packets, 0 bytes; action: transmit
      exceeded 0 packets, 0 bytes; action: drop
      last packet: 672000ms ago, current burst: 0 bytes
      last cleared 00:03:31 ago, conformed 0 bps, exceeded 0 bps

R4#show ip acce 100
Extended IP access list 100
    10 permit icmp any any echo-reply (100 matches)
```


IP Address Spoofing Prevention with ACLs

Objective: Configure router to protect against IP address spoofing



Directions

- The task is to implement IP address filtering on outside to filter out RFC1918, RFC3330 networks
- Additionally, as per the recommendations of RFC2627, ingress filtering should be performed, to deny "illegal" IP addresses.
- That is, only "our" network is permitted from inside, and "our" networks are denied as sources on outside
- Also, it's a good idea to filter out ICMP redirect messages and disable IP source routing
- Pay special attention not to block source address 0.0.0.0 on inside interface, since DHCP usually uses it to send requests
- Pre-configure routes as per NAT scenario "Standard NAT Configuration".
- Apply your configurations to R4
- Create extended access-list OUTSIDE_IN
  - Filter out ICMP redirects and packets sourced from host 0.0.0.0
  - Filter out RFC 1918 networks
  - Filter out RFC 3330 networks
  - As per RFC 2627 deny packets sourced from "our" network 150.1.4.0/24
- Create extended access-list INSIDE_IN
  - Filter out ICMP redirects
  - Permit UDP packets from 0.0.0.0/32 to 10.0.0.4 (R4's address) port BOOTPs.
  - Permit network 10.0.0.0/24 as per RFC2627

    - Block and log everything else.
- Apply access-list INSIDE_IN ingress to Ethernet interface
- Apply access-list OUTSIDE_IN ingress to Serial and FR interfaces

Final Configuration

R4:

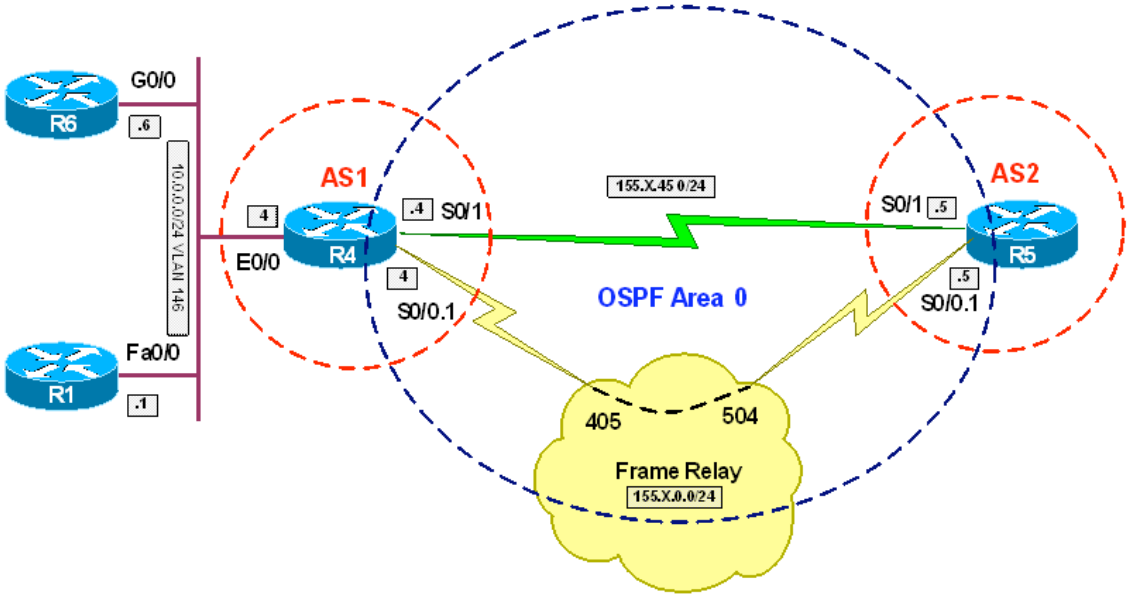
```
no ip source-route
!
ip access-list extended OUTSIDE_IN
 remark ==
 remark == Redirects may be used for spoofing
 remark ==
 deny icmp any any redirect
 remark ==
 remark == RFC 1918
 remark ==
 deny ip 10.0.0.0 0.255.255.255 any
 deny ip 172.16.0.0 0.15.255.255 any
 deny ip 192.168.0.0 0.0.255.255 any
 remark ==
 remark == RFC 3330
 remark ==
 deny ip host 0.0.0.0 any
 deny ip 224.0.0.0 31.255.255.255 any
 deny ip 127.0.0.0 0.255.255.255 any
 deny ip 169.254.0.0 0.0.255.255 any
 deny ip 192.0.2.0 0.0.0.255 any
 remark ==
 remark == RFC 2627
 remark ==
 deny ip 150.1.4.0 0.0.0.255 any
 remark ==
 remark == End of List
 remark ==
 permit ip any any

ip access-list extended INSIDE_IN
 deny icmp any any redirect
 permit ip 10.0.0.0 0.0.0.255 any
 permit udp host 0.0.0.0 host 10.0.0.4 eq bootps
 deny ip any any log

interface Ethernet 0/0
 ip access-group INSIDE_IN in
!
interface Serial 0/1
 ip access-group OUTSIDE_IN in
!
interface Serial 0/0.1
 ip access-group OUTSIDE_IN in
```


Verification

```
R4#sho ip access-lists
Standard IP access list INSIDE_NETWORK
    10 permit 10.0.0.0, wildcard bits 0.0.0.255
Extended IP access list INSIDE_IN
    10 deny icmp any any redirect
    20 permit ip 10.0.0.0 0.0.0.255 any
    30 permit udp host 0.0.0.0 host 10.0.0.4 eq bootps
    40 deny ip any any log

Extended IP access list OUTSIDE_IN
    10 deny icmp any any redirect
    20 deny ip 10.0.0.0 0.255.255.255 any
    30 deny ip 172.16.0.0 0.15.255.255 any
    40 deny ip 192.168.0.0 0.0.255.255 any
    50 deny ip host 0.0.0.0 any
    60 deny ip 224.0.0.0 31.255.255.255 any
    70 deny ip 127.0.0.0 0.255.255.255 any
    80 deny ip 169.254.0.0 0.0.255.255 any
    90 deny ip 192.0.2.0 0.0.0.255 any
    100 deny ip 150.1.4.0 0.0.0.255 any
    110 permit ip any any (26 matches)
```


Using uRPF to Prevent IP Address Spoofing

Objective: Configure router to prevent address spoofing using uRPF



Directions

- Configure routers as per the NAT scenario "Common Configuration"
- The task is to enable uRPF checks, yet exempt some networks from verification
- Additionally, all spoofing attempts should be logged
- Create additional interfaces Loopback1 on R5 with IP address 150.1.55.55/24 and Loopback2 with IP address 150.1.155.155/24
- Do not advertise the new Loopbacks into any routing protocol
- Disable BGP on R4 to stop receiving default route.
- Create access-list 100 on R4 and permit network 150.1.55.0/24. Deny and log everything else
- Configure uRPF on R4, using Serial and FR interfaces. Apply access-list 100 as uRPF ACL

Final Configuration

R4:

```
ip cef
access-list 100 permit ip 150.1.55.0 0.0.0.255 any
access-list 100 deny ip any any log
!
interface Serial 0/1
 ip verify unicast reverse 100
!
interface Serial 0/0.1
 ip verify unicast reverse 100
!
no router bgp 1
```



```
R5:
interface Loopback1
 ip address 150.1.55.55 255.255.255.0
!
interface Loopback2
 ip address 150.1.155.155 255.255.255.0
```

Verification

```
R5#ping 150.1.4.4 so lo1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
Packet sent with a source address of 150.1.55.55
.....
Success rate is 0 percent (0/5)

R4#sh ip acce
Extended IP access list 100
    20 permit ip 150.1.55.0 0.0.0.255 any (5 matches)
    30 deny ip any any log

R4#show ip int se 0/1
Serial0/1 is up, line protocol is up
  Internet address is 155.1.45.4/24
  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
  Multicast reserved groups joined: 224.0.0.5
  Outgoing access list is not set
  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
  IP fast switching is enabled
  IP fast switching on the same interface is enabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Feature Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  BGP Policy Mapping is disabled
  IP verify source reachable-via RX, allow default, ACL 100
  0 verification drops
  0 suppressed verification drops
```



```
R5#ping 150.1.4.4 so lo2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.4.4, timeout is 2 seconds:
Packet sent with a source address of 150.1.155.155
.....
Success rate is 0 percent (0/5)

R4#
%SEC-6-IPACCESSLOGDP: list 100 denied icmp 150.1.155.155 -> 150.1.4.4 (0/0), 1
packet

R4#show ip int se 0/1
Serial0/1 is up, line protocol is up
  Internet address is 155.1.45.4/24
  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
  Multicast reserved groups joined: 224.0.0.5
  Outgoing access list is not set
  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
  IP fast switching is enabled
  IP fast switching on the same interface is enabled
  IP Flow switching is disabled
  IP CEF switching is enabled
  IP CEF Feature Fast switching turbo vector
  IP multicast fast switching is enabled
  IP multicast distributed fast switching is disabled
  IP route-cache flags are Fast, CEF
  Router Discovery is disabled
  IP output packet accounting is disabled
  IP access violation accounting is disabled
  TCP/IP header compression is disabled
  RTP/IP header compression is disabled
  Policy routing is disabled
  Network address translation is disabled
  BGP Policy Mapping is disabled
  IP verify source reachable-via RX, allow default, ACL 100
  0 verification drops
  0 suppressed verification drops

R4#sh ip acce 100
Extended IP access list 100
    20 permit ip 150.1.55.0 0.0.0.255 any (5 matches)
    30 deny ip any any log (5 matches)
```

