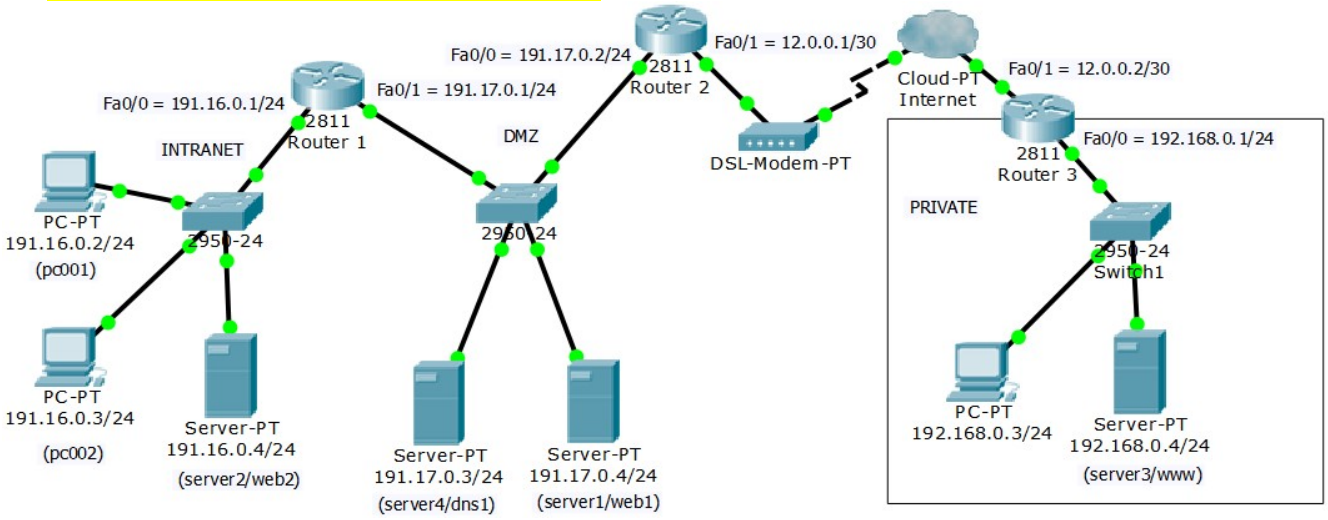
Redes de Computadores (RCOMP) – 2017/2018 Laboratory Class Script – PL22	
• Practical exercises – DNS, NAT and ACLs. • Reflexive ACLs.	• Cisco Packet Tracer

1. Practical exercise – DNS and NAT

Download the following Packet Tracer diagram.



All IP addresses and routeing tables are already settled. All end nodes are using DNS server 191.17.0.3, however, the DNS service is not active yet. The right side network is private, therefore, is not known to the internet, namely by Router 2. Nevertheless, NAT configuration on Router 3 is not yet deployed, you may check communications with the right side network are not possible for now.

1.1. Setup NAT on Router 3 to meet the following objectives:

a) Configure dynamic NAT with overload to allow all private nodes access to the internet.

```
(config)#interface Fa0/0
(config-if)#ip nat inside
(config)#interface Fa0/1
(config-if)#ip nat outside
(config)#no access-list 5
(config)#access-list 5 permit 192.168.0.0 0.0.0.255
(config)#ip nat inside source list 5 interface Fa0/1 overload
```

b) Configure static NAT to allow internet access to the HTTP service on private server 192.168.0.4. The internet access should be available at public address 12.0.0.2, TCP port number 80.

```
(config)#ip nat inside source static tcp 192.168.0.4 80 12.0.0.2 80
```

1.2. Thoroughly check connectivity.

a) Perform ICMP echo request tests between all end nodes, you will check private nodes are not accessible by public nodes, but elsewhere tests will be successful, including echo requests from private nodes to public nodes.

b) Now, go to pc001 (191.16.0.2) and open the Web Browser

Access URL http://191.16.0.4

Access URL http://191.17.0.4

Access URL http://192.168.0.4 (this will fail because the server address is private)

Access URL http://12.0.0.2 (thanks to static NAT the private server is available here)

c) Now, go to the right side private PC (192.168.0.3) and open the Web Browser

Access URL http://191.16.0.4

Access URL http://191.17.0.4

Access URL http://192.168.0.4

Access URL http://12.0.0.2 (this will fail)

The last test fails because packets are entering through an inside interface and after static NAT they would be exiting through an equally inside interface, we already know NAT is not applied it this case.

1.3. Activate and configure DNS service on server 191.17.0.3/24.

Create **A records** for names: pc001, pc002, server1, server2, server3 and server4

Attention: because the DNS server is public, server3 **A record** must be the public address (12.0.0.2).

Create **CNAME records** for names: web1, web2, dns1 and www.

1.4 Check communications again, but now using DNS names instead of IP addresses

a) Go to pc001 (191.16.0.2), open the Command Prompt and run:

```
ping web1
ping web2
ping dns
ping www
```

b) Yet on pc001 (191.16.0.2), open the Web Browser and access:

- http://web1
- http://web2
- http://www

Now, the same tests from the private network.

c) On the right side private PC (192.168.0.3) and open the Command Prompt and run:

```
ping web1
ping web2
ping dns
ping www
```

d) On the same PC (192.168.0.3), open now the Web Browser and access:

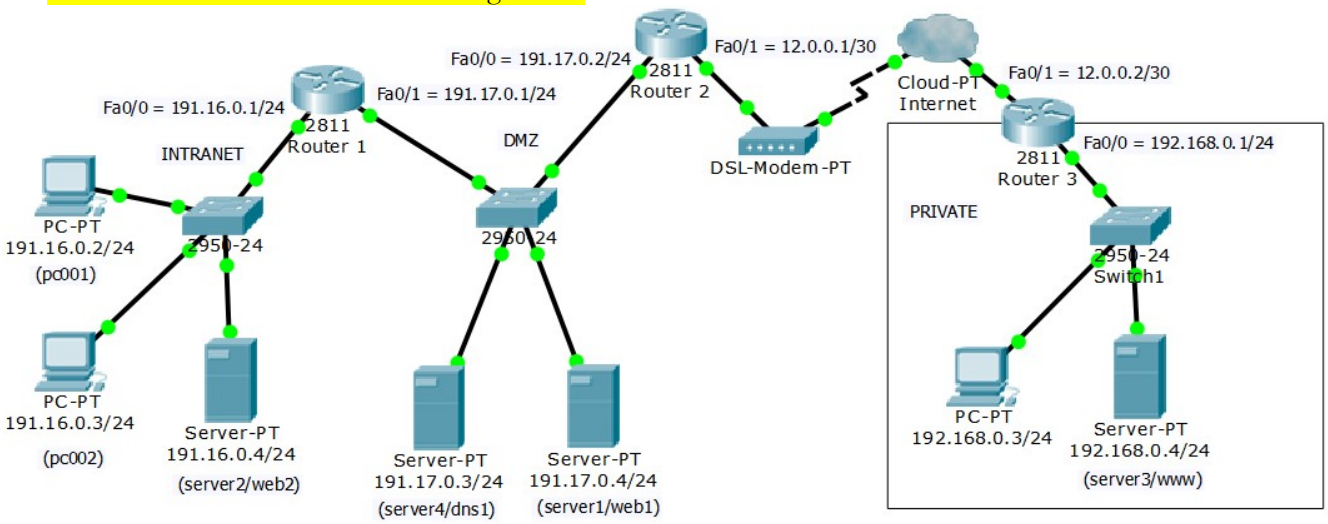
- http://web1
- http://web2
- http://www

We already knew this last test was fated to fail because the name resolves to address 12.0.0.2.

Nodes on the right side private network are not able to access the local web server by DNS name. How could this issue be solved?

2. Practical exercise – Access Control Lists

Continue with the last exercise's configuration.



The left side networks' administrator (INTRANET and DMZ) is required to enforce traffic access policies as follows:

- As far as possible, all IP spoofing must be blocked.

- The INTERNET may:

- Access HTTP service (TCP/80) on server1 and server2. Response traffic must also be allowed.

- Access DNS service (UDP/53) on server4. Response traffic must also be allowed.

- Send ICMP echo requests to nodes 191.16.0.4 and 191.17.0.4. Response traffic must also be allowed.

- For the INTRANET:

- All network nodes (191.16.0.1 up to 191.16.0.254) may access HTTP service (TCP/80) on server1 and DNS service (UDP/53) on server4. Response traffic must also be allowed.

- All network nodes (191.16.0.1 up to 191.16.0.254) may access HTTP service (TCP/80) on every internet node. Response traffic must also be allowed.

- Nodes 191.16.0.2, 191.16.0.6, 191.16.0.34 and 191.16.0.38 may send ICMP echo requests to every internet node. Response traffic must also be allowed.

- DMZ:

- Node 191.17.0.3 may access the DNS service (UDP/53) on every internet node. Response traffic must also be allowed.

- All other traffic is to be blocked.

Bear in mind that under this administrator's point of view, whatever nodes are beyond Router 2 interface Fa0/1 are internet nodes. Internet nodes addresses are all possible addresses except for those belonging to local networks.

2.1. Designing access lists

When designing access lists to a complex access policies scenario, many alternative solutions are possible, yet best solutions are usually achieved by blocking traffic as close to the source as possible. This implies enforcing ACL on input traffic.

One simple systematic approach is traversing every router's interface, and for each, analyse which traffic should be let in and which should be blocked.

So in this specific scenario we have four interfaces to analyse inbound traffic:

Router 1 – Interface Fa0/0 – Inbound (INTRANET)

```
no access-list 100
access-list 100 permit tcp host 191.16.0.4 eq 80 any established
access-list 100 permit tcp 191.16.0.0 0.0.0.255 host 191.17.0.4 eq 80
access-list 100 deny tcp any 191.17.0.0 0.0.0.255 eq 80
access-list 100 permit tcp 191.16.0.0 0.0.0.255 any eq 80
```

1st line - Allows response traffic for every HTTP access to server2.

2nd line - Allows HTTP access to server1.

3rd and 4th lines - Allows HTTP access from all nodes to all other nodes (internet), except for the DMZ.

```
(…)
access-list 100 permit udp 191.16.0.0 0.0.0.255 host 191.17.0.3 eq 53
access-list 100 permit icmp host 191.16.0.4 any echo-reply
access-list 100 deny icmp any 191.17.0.0 0.0.0.255 echo
access-list 100 permit icmp 191.16.0.2 0.0.0.36 any echo
```

1st line - Allows every node access to DNS service on server4.

2nd line - Allows echo replies (response traffic to internet echo requests to node 191.16.0.4).

3rd and 4th lines - Allows echo requests from nodes 191.16.0.2, 191.16.0.6, 191.16.0.34 and 191.16.0.38 to all nodes (internet) except for the DMZ.

Router 1 – Interface Fa0/1 – Inbound (DMZ)

```
no access-list 105
access-list 105 deny ip 191.16.0.0 0.0.0.255 any
access-list 105 permit tcp any eq 80 191.16.0.0 0.0.0.255 established
access-list 105 permit udp host 191.17.0.3 eq 53 191.16.0.0 0.0.0.255
access-list 105 deny ip 191.17.0.0 0.0.0.255 any
access-list 105 permit tcp any host 191.16.0.4 eq 80
access-list 105 permit icmp any 191.16.0.2 0.0.0.36 echo-reply
access-list 105 permit icmp any host 191.16.0.4 echo
```

1st line – Block external spoofing.

2nd line - Allows response traffic for every HTTP access.

3rd line - Allows response traffic for DNS requests to server4.

4th line - Block all other traffic from DMZ.

5th line – Allow internet HTTP access to server2.

6th line – Allow ICMP echo replies to nodes 191.16.0.2, 191.16.0.6, 191.16.0.34 and 191.16.0.38.

7th line – Allow ICMP echo requests to server2.

Router 2 – Interface Fa0/0 – Inbound (DMZ)

```
no access-list 100
access-list 100 permit tcp 191.16.0.4 0.1.0.0 eq 80 any established
access-list 100 permit udp host 191.17.0.3 eq 53 any
access-list 100 permit udp host 191.17.0.3 any eq 53
access-list 100 permit icmp 191.16.0.4 0.1.0.0 any echo-reply
access-list 100 deny ip 191.17.0.0 0.0.0.255 any
access-list 100 permit tcp 191.16.0.0 0.0.0.255 any eq 80
access-list 100 permit icmp 191.16.0.2 0.0.0.36 any echo
```

1st line – Allow HTTP responses from server1 and server2.

2nd line - Allows DNS response traffic from server4.

3rd line – Allows DNS requests from server4.

4th line - Allow ICMP echo replies from nodes 191.16.0.4 and 191.17.0.4.

5th line - Block all other traffic from DMZ.

6th line – Allow HTTP access from INTRANET.

7th line – Allow ICMP echo requests from nodes 191.16.0.2, 191.16.0.6, 191.16.0.34 and 191.16.0.38.

Router 2 – Interface Fa0/1 – Inbound (INTERNET)

```
no access-list 110
access-list 110 deny ip 191.16.0.0 0.1.0.255 any
access-list 110 permit tcp any 191.16.0.4 0.1.0.0 eq 80
access-list 110 permit udp any host 191.17.0.3 eq 53
access-list 110 permit udp any eq 53 host 191.17.0.3
access-list 110 permit icmp any 191.16.0.4 0.1.0.0 echo
access-list 110 deny ip any 191.17.0.0 0.0.0.255
access-list 110 permit tcp any eq 80 191.16.0.0 0.0.0.255 established
access-list 110 permit icmp any 191.16.0.2 0.0.0.36 echo-reply
```

1st line – Block external spoofing.

2nd line – Allow HTTP access to server1 and server2.

3rd line – Allow DNS access to server4.

4th line – Allows DNS replies to server4.

5th line – Allow ICMP echo requests to server1 and server2.

6th line – Block all other traffic to the DMZ.

7th line – Allow HTTP response traffic to all INTRANET nodes.

8th line – Allow ICMP echo replies to nodes 191.16.0.2, 191.16.0.6, 191.16.0.34 and 191.16.0.38.

2.2. Deploy defined access lists

Router 1:

```
interface Fa0/0
ip access-group 100 in
interface Fa0/1
ip access-group 105 in
```

Router 2:

```
interface Fa0/0
ip access-group 100 in
interface Fa0/1
ip access-group 110 in
```

2.3. Make some tests to partially validate the solution

a) Access from the internet.

Send ICMP echo requests from **server3** to:

- server1
- server2
- server4
- pc001

Open the Web Browser on **server3** and access URLs:

- http://web1
- http://web2

Check if test results match enforced access policies. Notice that by using hostnames (web1 and web2) the DNS service will be used.

b) Access to the internet.

Go to pc001 (191.16.0.2)

- open the Command Prompt and run:

```
ping www
```

- open the Web Browser and access: **http://www**

Repeat same tests on pc002 (191.16.0.3)

- open the Command Prompt and run:

```
ping www
```

- open the Web Browser and access: **http://www**

Check if test results match enforced access policies. Again, notice that by using a hostname (www) the DNS service will be used.

c) Perform further tests and see if results are as expected.

There are some issues with these access lists configuration

Spoofing is not entirely avoided – Nodes at the DMZ may send traffic to the INTERNET impersonating INTRANET nodes. Also, nodes at the DMZ may send traffic to the INTRANET impersonating INTERNET nodes. Nevertheless, there is no way to overcome this issue because in this scenario the DMZ is a transit network. Incoming traffic from transit networks may or not be originated on the network, thus some cases of spoofing cannot be prevented.

Allowed response traffic exposes nodes – these ACLs have static rules allowing response traffic, attackers may use these rules to reach internal nodes. This happens because the traffic is always allowed whether or not there was a request matching the allowed response. Reflexive ACLs can be used to improve this security flaw.

2.4. Reflexive access lists (not supported in Packet Tracer)

Reflexive ACLs are dynamically created to allow response traffic. They are valid only within named extended access lists.

On a named extended access list permit rule, the additional parameter **reflect** can be used:

```
(config-ext-nacl)# permit (…) reflect ACL-NAME [ timeout SEC ]
```

If a permit rule has the reflect parameter, **whenever a packet matches the rule**, a temporary permit rule is created and added to the named extended access list ACL-NAME (if it does not exist, it's created) this is called a reflexive ACL.

- The temporary permit rule created in access list ACL-NAME, only lasts for SEC seconds, 300 seconds by default. This can also be settled by the **ip reflexive-list timeout SEC** command.
- The created temporary permit rule criterions are:
    - Same protocol as the matched packet.
    - Source address equal to the matched packet destination address.
    - Destination address equal to the matched packet source address.
    - For UDP/TCP:
        - Destination port number equal to the matched packed source port number.
        - Source port number equal to the matched packed destination port number.

Reflexive ACLs are intended to allow, case by case, response traffic for each allowed request, therefore the reflexive ACL must be enforced in the opposite direction. This can be achieved by including the reflexive ACL into a named extended ACL that is being enforced in the opposite direction.

In an extended named ACL, another extended named ACL can be included by using the **evaluate** command:

```
(config-ext-nacl)# evaluate ACL-NAME
```

Even though this can't be tested in Packet Tracer, here is how we could use reflexive ACLs to solve the previous problem.

Router 1 – Interface Fa0/0 – Inbound (INTRANET)

```
ip access-list extended intra-in
 evaluate intra-replies
 permit tcp 191.16.0.0 0.0.0.255 host 191.17.0.4 eq 80 reflect dmz-replies
 deny tcp any 191.17.0.0 0.0.0.255 eq 80
 permit tcp 191.16.0.0 0.0.0.255 any eq 80 reflect dmz-replies
 permit udp 191.16.0.0 0.0.0.255 host 191.17.0.3 eq 53 reflect dmz-replies
 deny icmp any 191.17.0.0 0.0.0.255 echo
 permit icmp 191.16.0.2 0.0.0.36 any echo reflect dmz-replies
```

Router 1 – Interface Fa0/1 – Inbound (DMZ)

```
ip access-list extended dmz-in
 deny ip 191.16.0.0 0.0.0.255 any
 evaluate dmz-replies
 deny ip 191.17.0.0 0.0.0.255 any
 permit tcp any host 191.16.0.4 eq 80 reflect intra-replies
 permit icmp any host 191.16.0.4 echo reflect intra-replies
```

Router 2 – Interface Fa0/0 – Inbound (DMZ)

```
ip access-list extended dmz-in
 evaluate dmz-replies
 permit udp host 191.17.0.3 any eq 53 reflect intra-replies
 deny ip 191.17.0.0 0.0.0.255 any
 permit tcp 191.16.0.0 0.0.0.255 any eq 80 reflect intra-replies
 permit icmp 191.16.0.2 0.0.0.36 any echo reflect intra-replies
```

Router 2 – Interface Fa0/1 – Inbound (INTERNET)

```
ip access-list extended inter-in
 deny ip 191.16.0.0 0.1.0.255 any
 evaluate intra-replies
 permit tcp any 191.16.0.4 0.1.0.0 eq 80 reflect dmz-replies
 permit udp any host 191.17.0.3 eq 53 reflect dmz-replies
 permit icmp any 191.16.0.4 0.1.0.0 echo reflect dmz-replies
```