Laboratory Class Script – PL19

• Static firewall. • Access Control Lists	• Cisco Packet Tracer

1. Static firewall

Static firewalls permit or deny individual packets based on static rules defined by the network administrator. Other names for this firewall type are **First generation firewall**, **Packet Filter**, **Stateless Firewall, Screening Router** or yet **Network Layer Firewall**.

Criterions for packet matching (and consequent permit or deny) are relative to layer three and four properties: IP source and destination node addresses (IPv4/IPv6), payload type (e.g. ICMP, UDP, TCP), source and destination port numbers (for UDP and TCP payloads only), message types for ICMP payloads, etc.

Static firewalls react the same way to a given packet regardless of context, this makes them inadequate against DoS attacks. The problem is, nothing allows it to distinguish between a packet from a licit access to the service and packet from a DoS attack to the service. A different type of firewall is required to handle DoS attacks, they are called dynamic or stateful.

Although they are not able to block some DoS attacks, static firewalls can do a lot in behalf of network security. On the top of the list of attacks that a static firewall can avoid are IP spoofing attacks.

1.1. IP spoofing

IP spoofing means forging the packet source IP address, this can be used for several illicit purposes, including concealing DoS attacks (Distributed DoS) to prevent them from being detected by dynamic firewalls. IP spoofing is also used to mislead static firewalls making them believe a packet is coming from an authorised source node address, and thus, allow access to some restricted services.

The measures to be taken in order to prevent IP spoofing, depend whether traffic is coming from known networks, or if traffic is going to known networks. **Local networks' addresses are known, but internet networks' addresses are unknown.**

Avoiding **internal spoofing**: any traffic going to the internet must have source addresses belonging to one of the known local networks.

Avoiding **external spoofing**: any traffic coming from the internet must have source addresses not belonging to any of the known local networks.

1.2. Access control

Although end-nodes should have local firewalls, firewalls are most effective in routers because they can be used by network administrators to enforce access rules for the entire infrastructures. Under the security level point of view, internet connected infrastructures should be split in at least three zones:

Intranet - Local users' networks (known networks) – from where some attacks are expected.

Internet – All other networks (unknown addresses networks) – from where most attacks are expected.

Demilitarized Zone (DMZ) – Local isolated and trusted network – no attacks are expected form here.

The DMZ must be isolated from other networks by routers and firewalls, no user workstations are allowed here, only servers. Firewalls must be deployed between security level zones.

The recommended access policy for any firewall is: block all traffic except for explicitly required traffic.

1.3. Selecting the access control deployment place

Packets travel throughout the network infrastructure passing through several routers, access control can be enforced in any of them. A decision must be taken about where (in which router) access control should be enforce for each purpose. For each access policy to be implemented, there will usually be several alternative options regarding the deployment router, there's no absolute rule for a decision because specific environments can lead to different optimal solutions. The optimal solution is the one that results in fewer rules and also avoids unnecessary traffic within the infrastructure.

Despite the inexistence of an absolute rule we can, nevertheless, point out one general rule: block traffic as soon as possible.

The interpretation of this general rule is, block incoming traffic as close to its source as possible. In other words, when it's being received by the closest to the source router, if possible, the router that is directly connected to the source network.

Yet, this is not a golden rule, specific circumstances may turn this not to be the best option.

2. Static firewall setup in Cisco IOS

In Cisco IOS we setup firewall rules by creating access control lists (ACL) and assigning them to incoming our outgoing traffic in specific router's interfaces.

An access list is a sequence of rules, each rule sets a match criterion and an action (permit or deny). Each individual packet is sequentially confronted with every rule on the ACL until a match is found. If a match is found, the action is executed and the analysis ends for that packet. Therefore, the rules' order on the ACL is pretty relevant.

If no matching rule is found on the ACL the packet is denied by default.

While until no ACL is deployed all packets are allowed to pass, once an ACL is deployed, only packets matching a permit rule without previously matching a deny rule will be allowed.

2.1. Numbered standard access lists

They are identified by a number between one and 99. The only matching criterion for standard access list rules is the source IP address. To **add (append) a rule** to a numbered standard access list the following command is used:

```
access-list NN ACTION SOURCE-IP-ADDRESS-SPECIFICATION
```

Where NN represents the access list identifier (1 up to 99) and action is either **permit** or **deny**. Numbered access lists are not editable, the only two available operations are adding (appending) a rule and removing the entire ACL (remove all rules) by using the following command:

```
no access-list NN
```

Each added rule will match packets with source addresses conforming to SOURCE-IP-ADDRESS-SPECIFICATION, this can be either:

```
host DOT-DECIMAL-IP-ADDRESS

any

DOT-DECIMAL-IP-ADDRESS DOT-DECIMAL-WILDCARD
```

The first form matches a single IP address (DOT-DECIMAL-IP-ADDRESS), the second, matches any IP address. The last form matches all addresses equal to DOT-DECIMAL-IP-ADDRESS but bits with one value in DOT-DECIMAL-WILDCARD are ignored, this means it only compares bits with zero value in DOT-DECIMAL-WILDCARD.

2.2. Numbered extended access lists

They are identified by a number between 100 and 199. Again, numbered access lists are not editable, the only two available operations are adding (appending) a rule and removing the entire ACL (remove all rules). But, unlike standard access lists, several other criterions can be now used beyond the packet's source IP address. **Now, a packet matches a rule only if it matches at same time all criterions on the rule**. Required criterions are now the protocol identifier (IP packet payload type), the source address and the destination address:

```
access-list NNN ACTION PROTOCOL SOURCE-IP-ADDRESS DESTINATION-IP-ADDRESS
```

Where NNN represents the access list identifier, now a number between 99 and 199. The PROTOCOL identifier can be either **ip**, which matches any payload type, or a payload identifier (upper layer protocol) like, for instance, **udp**, **tcp** or **icmp**. Both the SOURCE-IP-ADDRESS and DESTINATION-IP-ADDRESS specifications can use any of the three forms mentioned earlier.

Additional optional criterions may be available depending on the matching protocol. For UDP and TCP a source port number criterion may be added to the source IP address and a destination port number criterion may be added to the destination IP address

For the ICMP protocol, a message type identifier criterion may be added. For TCP protocol, the **established** criterion may also be used to match only packets regarding an already established TCP connection.

2.3. Named access lists

Named access lists are identified by names and not numbers, but they work pretty the same as numbered ACLs. The biggest advantage is they can be edited. The command used to create a new, or edit an existing, named access list is:

```
ip access-list standard|extended ACL-NAME
```

This will make CLI switch to a specific configuration level where rule can entered, each with a sequence number. In no sequence number is specified for a rule, it's appended, starting with number 10 and with a step of 10. Depending on being standard or extended rules are entered the same way as with numbered access lists, except that the **access-list** command is omitted.

For instance to insert a permit rule in ACL position 305 we would enter:

```
305 permit ...
```

This will fail if there's already a rule in position 305, we must first remove it:

```
no 305
```

At any time, we can renumber the rules in the named access list:

```
ip access-list resequence ACL-NAME STARTING-NUMBER STEP
```

2.4. Appling defined access lists

Just defining an access list has no immediate practical effect. Is must be applied, either to incoming or outgoing traffic of a router's interface. Until one access list is applied to an interface traffic, all traffic is permitted, as soon as we apply one, only traffic permitted by the ACL is allowed. Incoming and outgoing traffics are independent, for instance, applying an access list to incoming traffic does not affect outgoing traffic.

To apply an access list to some interface's traffic we must first enter that interface configuration level:

```
interface INTERFACE-NAME
```

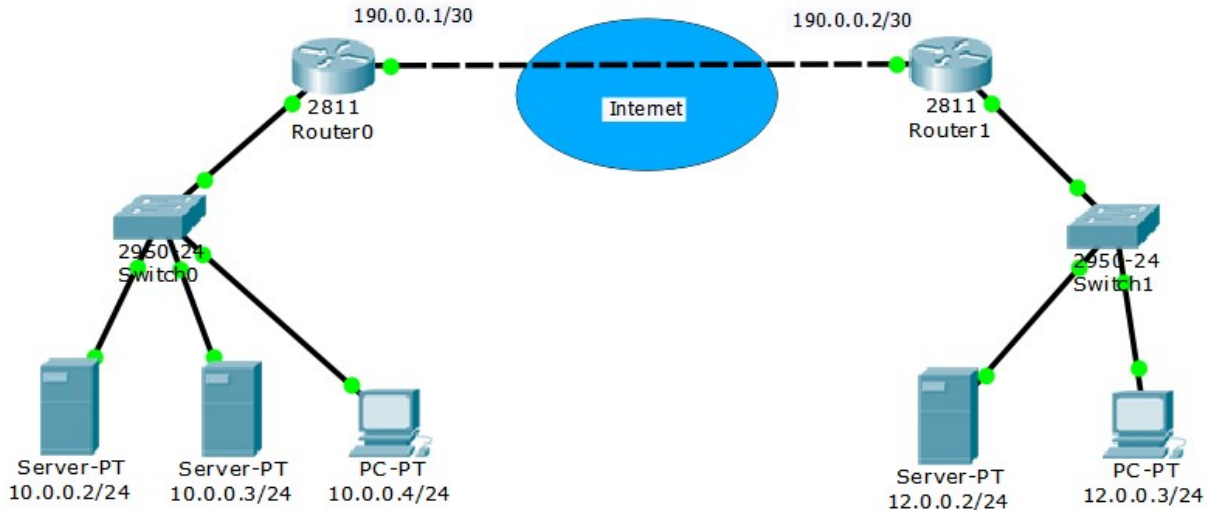
Once in interface configuration level, we can then apply one ACL to incoming traffic and another ACL to outgoing traffic:

```
ip access-group ACL-IDENTIFIER in|out
```

Where ACL-IDENTIFIER is either a number or a name. We may apply one ACL to incoming traffic and another ACL to outgoing traffic. But only one ACL to each traffic type.

3. Practical exercise

Create the following diagram on Packet Tracer:
(This diagram is available for download with already settled IP addresses and routing information)



Even though for the purpose of this exercise the internet is just a point to point connection, we will handle it as being the internet. Therefore, Router 0 is not aware of the right side network, it just known the default-gateway to use is 190.0.0.2 and everything beyond that is the internet (unknown addresses). As well, Router 1 is not aware of the left side network, it just known the default-gateway to use is 190.0.0.1 and everything beyond that is the internet (unknown addresses).

a) **If not already done, settle nodes IP addresses and routing.**

   Perform ICMP echo request tests to be sure all end nodes can communicate with each other.

b) **Now let's check IP spoofing is not being blocked.**

   In simulation mode.

   Select the Add Complex PDU tool and click on PC 12.0.0.3

   - Define as destination IP address: 10.0.0.2 (the leftmost server)

   - Define as source IP address: 10.0.0.3 (this is IP spoofing)

   - Define sequence number as 1 and one shot time 5 seconds.

   Start the simulation.

   You will see the request arriving to server 10.0.0.2, without it realizing the source address is forged, therefore, it sends a reply to 10.0.0.3.

   Server 10.0.0.3 ultimately discards the received echo reply because it hasn't asked for it in the first place.

   The point is, imagine our access policy was blocking any external access to 10.0.0.3, the attacker at 12.0.0.2 has overcome that policy and could for instance perform an echo requests DoS attach to 10.0.0.3.

Regarding the presented scenario, we can highlight **two security faults** over IP spoofing:

> **First** – the right side network administrator is not preventing **internal spoofing** as he should. He is allowing packets from his internal network outgoing to the internet with forged IP source addresses.

Second – the left side network administrator is failing to protect his own networks from **external spoofing**. He is not blocking incoming packets from the internet with forged source addresses belonging to his internal networks.

c) Fix all IP spoofing issues on this scenario.

Because it's all about packets' IP source addresses, IP spoofing can be prevented by using **standard** access lists, nevertheless, they may have to be later convert to **extended** if additional access policies are to be enforced.

Starting with the left side router administrator (Router 0):

`(config)#no access-list 1`
`(config)#access-list 1 permit 10.0.0.0 0.0.0.255`
 	
`(config)#interface Fa0/1`
`(config-if)#ip access-group 1 in`
	To avoid internal spoofing we create the numbered standard access list 1. It allows packets with a source address belonging to the internal network (10.0.0.0/24). Remaining traffic is, by default, blocked.
	
	Apply it to incoming traffic on interface Fa0/1.

`(config)#no access-list 2`
`(config)#access-list 2 deny 10.0.0.0 0.0.0.255`
`(config)#access-list 2 permit any`
 	
`(config)#interface Fa0/0`
`(config-if)#ip access-group 2 in`
	To avoid external spoofing we create the numbered standard access list 2. It blocks any packet with a source address belonging to the internal network (10.0.0.0/24). Remaining traffic must be explicitly permitted, otherwise it would be blocked by default.
	
	Apply it to incoming traffic on interface Fa0/0.

Now, the same principles for the right side router administrator (Router 1):

`(config)#no access-list 10`
`(config)#access-list 10 permit 12.0.0.0 0.0.0.255`
 	
`(config)#interface Fa0/1`
`(config-if)#ip access-group 10 in`
	To avoid internal spoofing we create the numbered standard access list 10. It allows packets with a source address belonging to the internal network (12.0.0.0/24). Remaining traffic is, by default, blocked.
	
	Apply it to incoming traffic on interface Fa0/1.

`(config)#no access-list 20`
`(config)#access-list 20 deny 12.0.0.0 0.0.0.255`
`(config)#access-list 20 permit any`
 	
`(config)#interface Fa0/0`
`(config-if)#ip access-group 20 in`
	To avoid external spoofing we create the numbered standard access list 20. It blocks any packet with a source address belonging to the internal network (12.0.0.0/24). Remaining traffic must be explicitly permitted, otherwise it would be blocked by default.
	
	Apply it to incoming traffic on interface Fa0/0.

This settles it for IP spoofing blocking.

d) Test again sending IP spoofing ICMP echo request as done in letter b), packets with forged source IP addresses don't even get out to the internet.

Notice, though, that in fact, this does not solve every IP spoofing issue. For local traffic, not passing through any firewall there is yet a flaw.

e) Let's see this flaw.

In simulation mode.

Select the Add Complex PDU tool and click on PC 10.0.0.4

- Define as destination IP address: 10.0.0.2 (the leftmost server)

- Define as source IP address: 10.0.0.3 (this is IP spoofing)

- Define sequence number as 1 and one shot time 5 seconds.

Start the simulation.

Again you will see the request arriving to server 10.0.0.3, without it realizing the source address is forged, therefore, it sends a reply to 10.0.0.2.

Solving this would require a separate network with its own firewall for each node, so this is as far as network administrators can go. This must be handled by the node local firewall administrator, not the network administrator.

f) Now let's add additional traffic restrictions.

First let's start by checking some accesses with present configuration. As far as source addresses are not forged, all accesses are allowed. All these servers have an HTTP service and also a client (Web Browser).

Select a server, then the Desktop tab and finally select Web Browser, enter the server IP address to be accessed by HTTP and you will see the main HTML page displayed. As things stand any node can access any HTTP server.

We are now going to enforce several restrictions for the left side router, the right side router will remain with no further restrictions beyond spoofing blocking.

Access policy for the left side administrator:

- Allow ICMP echo requests (echo message type) from the internet to server 10.0.0.2 and respective ICMP echo replies (echo-reply message type).
- Allow any node on network
- Allow HTTP (TCP port 80) access from the internet to server 10.0.0.3 and reply traffic back to the internet.
- All other traffic is to be blocked.

Previously defined ACLs for Router 0 will be now useless because defined criterions are not compatible with standard access lists, nevertheless, IP spoofing measures must still be effective. Old access lists can be erased or will simply be ignored when we assign the new ones to interfaces traffic.

Internet incoming traffic on Router 0 (traffic from internet to the local network)

A solution with a numbered extended access list applied to Fa0/0 incoming traffic:

```
(config)#no access-list 100
(config)#access-list 100 deny ip 10.0.0.0 0.0.0.255 any
(config)#access-list 100 permit icmp any host 10.0.0.2 echo
(config)#access-list 100 permit tcp any host 10.0.0.3 eq 80

(config)#interface Fa0/0
(config-if)#ip access-group 100 in
```

Intranet incoming traffic on Router 0 (traffic from the local network to the internet)

A solution with a named extended access list applied to Fa0/1 incoming traffic:

```
(config)#ip access-list extended FROM-LOCAL
(config-ext-nacl)#permit icmp host 10.0.0.2 any echo-reply
(config-ext-nacl)#permit tcp host 10.0.0.3 eq 80 any established

(config)#interface Fa0/1
```

```
(config-if)#ip access-group FROM-LOCAL in
```

g) Now let's test it.

Try to establish a relation between each test result and the enforced access lists.

- Send ICMP echo requests from server 10.0.0.2 to servers 10.0.0.3 and 12.0.0.2
- Send ICMP echo requests from server 12.0.0.2 to servers 10.0.0.2 and 10.0.0.3
- Try other ICMP echo requests, including to routers.
- Open the Web Browser in server 10.0.0.2, try opening URLs http://10.0.0.3 and http://12.0.0.2
- Open the Web Browser in server 12.0.0.2, try opening URLs http://10.0.0.2 and http://10.0.0.3
- Open the Web Browser in server 10.0.0.3, try opening URLs http://10.0.0.2 and http://12.0.0.2