Time-based ACL

What is an ACL (Access Control List)?

Access Control List: is used for packet filtering where we define a list of restrictions or conditions (permit/deny) in a sequential order, where the traffic received is compared with our list and functions according to the restriction set and decides whether the traffic is allowed or denied to go in or out of a specified network.

What is a Time-based ACL?

Time-based ACL: is a type of an access list which functions the same way as a typical ACL but the difference is, that it's based on a specified time period (Periodic/Absolute) for it to operate, it's useful when we want to place restrictions on an outbound/inbound traffic on a **daily basis** (Periodic) or set it for a **one time use** on a particular time of the day (Absolute), Time-based ACL is an additive feature to an ACL which grants the network admin more control over the network.

Scenarios

Scenario using Periodic keyword:

Time-based ACL is useful in a scenario where a company would want to **deny** their employees from internet access during their working hours (Monday – Friday | 9am – 5pm) but is then **allowed** as soon as the time period is over (Saturday – Sunday) or any time before/after 9 am and 5 pm, restrictions is then automatically reapplied in the following week ⬚Periodic (on a weekly basis)

Scenario using Absolute keyword:

The company could make use of the time-based ACL by **permitting** access to the internet for a specified amount of time and goes back to **denying** internet access as soon as the time we set runs out example: during lunch breaks ⬚Absolute (one-time use) has a start time and end time

Local clock vs NTP (Network Time Protocol):

Using the local clock for a time-based ACL: using the local clock to set a time-based ACL may cause issues as the user may

bypass the ACL restrictions by either setting the devices local time to a valid time (where internet access is allowed), or resetting the router which resets the routers local clock and would then grant him access to the internet or even worse it may remove any preconfigured access lists if the configs were not saved in the start up configs.

To mitigate this issue we can use NTP (network time protocol): which is a protocol used to unify and synchronize the time clock on all devices, where access list restrictions is compared with the defined time by the NTP instead of the local devices clock, where even if the user tried to change the local devices clock it wouldn't matter, or even in the case of restarting the router the NTP adjusts devices to the correct time as soon as it powers up again with the start up configs.

## How to configure a time-based ACL (periodic)

In this example: we want to **deny** icmp (ping) from PC1 & PC2 (network: 192.168.1.0 /24) to PC3 & PC4 (network: 192.168.2.0 /24) on weekdays ⬚ (Monday – Friday) from 09:00 to 17:00 (9am to 5pm) and automatically allow icmp (ping) after that time period.

Useful commands:

Router# show clock

Router# clock set 00:00:00(hh:mm:ss) 1 mar 2002 (dd mm yyyy)

Router# show access-lists
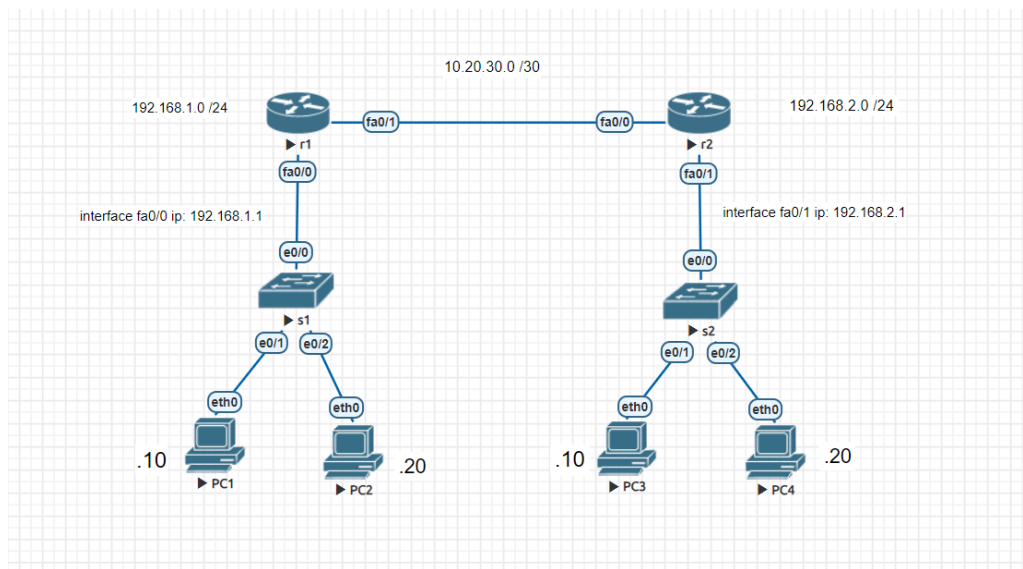

```
Router(config)#ip access-list exten
Router(config)#ip access-list extended 100
```

```
Router(config-ext-nacl)#deny icmp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
```

```
Router(config-ext-nacl)#$ 0.0.0.255 192.168.2.0 0.0.0.255 time-range time
```

```
Router(config-ext-nacl)#permit ip any any
```

```
Router(config)#time-range time
```

```
Router(config-time-range)#periodic weekdays 09:00 to 17:00

Router(config)#interface fastEthernet 0/0

Router(config-if)#ip access-group 100 in
```

Testing our ACL when the time is set to 2 pm on a Friday during working hours (communication administratively prohibited):

Testing our ACL when the time is set to 12 am on a Saturday (Success):

Testing our ACL when the time is set to 8 pm on a Friday (Success):

```
Router#
Router#
Router#
Router#
Router#
Router#clock set 20:00:00 1 mar 2002
Router#
Mar  1 20:00:00.003: %SYS-6-CLOCKUPDATE: System clock has been updated from 20:00:15 UTC Fri M
ar 1 2002 to 20:00:00 UTC Fri Mar 1 2002, configured from console by console.
Router#
```

```
pc1> ping 192.168.2.10

84 bytes from 192.168.2.10 icmp_seq=1 ttl=62 time=41.081 ms
84 bytes from 192.168.2.10 icmp_seq=2 ttl=62 time=41.280 ms
84 bytes from 192.168.2.10 icmp_seq=3 ttl=62 time=29.779 ms
84 bytes from 192.168.2.10 icmp_seq=4 ttl=62 time=38.326 ms
84 bytes from 192.168.2.10 icmp_seq=5 ttl=62 time=28.245 ms

pc1>
```

```
Router#show ip access-lists
Extended IP access list 100
    10 deny icmp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255 time-range time (inactive) (45 matches)
    20 permit ip any any (10 matches)
```