جامعة بيرزيت

BIRZEIT UNIVERSITY

Computer Systems Engineering

Computer Networks Lab

Report #2

Student:
Ameer Alkam
ID:
1120217

Supervisor:
Dr. Iyad Tumar
Teacher Assistant:
Odai Salman

Partners:
Omar Muhtaseb - 1122010
Sondus Samara - 1121775

April 27, 2016

Abstract

In this report we discuss two concepts of networks, Access Control Lists (ACL) and Virtual-LANs (VLANs). The first concept in which we deny or permit certain traffic to go through the network, for security, privacy, or any other reasons, by blocking certain sources or destinations, or doing more advanced analysis on the packets being transferred. The second concept is a method of creating logically separated networks, that mya or may not be in the same physical location, using switches, and connecting these VLANs using the Router on Stick (RoS) idea. We also demonstrate how to build and configure ACLs and VLANs.

Contents

Part I
Introduction

1 Access Lists

Access Control Lists, or ACLs, are methods of filtring the network traffic, by examining the packets based on some criteria, and deciding on it whether or not a router's interface will pass the traffic or drop it. The criteria used could be the source or destination address, the protocol used, or some other information. One of the main reasons of using an ACL is to provide more security to the network.

On CISCO routers, ACLs can either be named, or numbered, they use the Wildcard Mask (the inverse mask) instead of the traditional subnet mask. ACL is divided into two types, **Standard ACL** which only looks at the source IP of the packet, and filters on it's basis, and **Extended ACL** which filters on the basis of the source and destination IPs and port numbers. There's also an option that can be used for packets that do not fall under any explicit rules in the ACL, to **deny all traffic** that doesn't satisfy any of the rules set.

1.1 Configuring ACL

To configure ACL, we first must select an interface that the Access List applies to, then we define an Access List with the command ‘`ip access-group <number> {in|out}`‘, notice how when defining an Access Group we specify whether the rules for it apply to incomming traffic, or outgoing traffic. After that we add our ACL entires via the command ‘`access-list <number> {permit|deny} <source-IP> <source-wildcardmask>`‘.

To use extended mode, the command to add ACL entries takes the form ‘ `access-list <number> {permit|deny} <protocol> host <source-IP> host <destination-IP> {eq <port>}`‘, notice that `{eq <port>}` is optional and can be omitted. One small note is that to use Extended ACL the Access List number must be between 100 and 199, while using Standard ACL the Access List number is between 1 and 99.

To view ACL entries we can use the command ‘`show access-list`‘.

2 Virtual LANs

2.1 Switching

Switches, which are a Layer-2 devices in both the OSI and TCP/IP models, work by assigining the MAC address of any frames it recieves, to forward to other devices in the LAN, to the port it recieved it at, this creates a Switching Table, that it can later on use to forward the messages it recieves. When a switch recieves a frame to a device that isn't mapped on the switching table, it broadcasts the frame it recieved to all the devices connected to it, and waits for a response.

2.1.1 Switch Virtual Interface

A Siwtch Virtual Interface, SVI, is a VLAN of switch ports represening one switch port and a routing or bridging system. SVI can be used for switching, but it can also provide some Layer-3 functionality, using SVIs can ommit the need for a physical router, when connecting Virtual LANs together by providing the functionality given by the Router on Stick, more on Virtual LANs and Router on Stick later. SVIs are commonly found in special types of switches called Layer-3, or Multilayer, Switches.

2.2 VLANs

Virtual LANs, or VLANs, is the logical separation, on the Data-Link Layer, and grouping of the ports of the switch to create multiple independant LANs on one switch. Devices in VLANs are identified by the Tag ID of the specific VLAN their in, when a frame is sent, the switch first checks which ports are Tagged with the same ID, and then uses the switching table, or broadcasts over the port with the same tag, to decide which device is the destination.

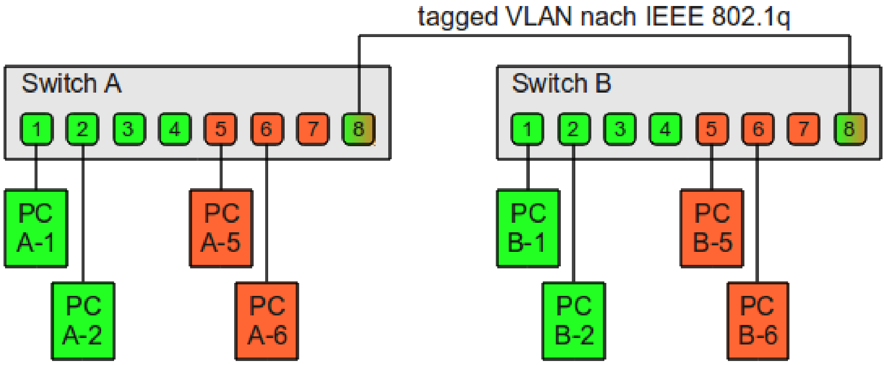


Figure 1: A schema showing a physical view of VLANs.

2.2.1 Trunks

Each port of the switch is, usually, assigned to one VLAN, and it drops any traffic it encounters that doesn't have the same VLAN tag as itself. This creates a problem when trying to connect two switches with VLANs enabled on each, since we need to connect one wire for each of the VLANs assigned at the switches. The main issue here is that a port only forwards traffic from and to it's own VLAN.

Trunks come in handy to solve the above issue, by allowing a port to forward any traffic it recieves to any known VLAN, regardless of it's VLAN tag, that way we can connect one Trunk between two switches, and have any device in any VLAN communicate with other devices from it's VLAN that is connected to the other switch.

2.2.2 RoS

Since each VLAN is treated as a seprate network, devices on any of the VLANs cannot communicate to one another, eventhough they maybe connected to the same switch. To connect the VLANs together we need to use a router and connect each VLAN to it, and have it route the traffic between the two. And due to the impracticallity of connecting evey VLAN individually to the router, we can use Trunks here to have only one link going to the router, that link must be a Trunk, and have all the traffic be forwarded through it. Such type of routers is called Router on Stick, or RoS for short.

To get this to work, that is, connect one Trunk that forwards the traffic of many VLANs, with different IP subnets, to one interface at the router, we use **sub-interfaces**, a sub-interface a part of a main interface of a router, that has a unique IP, and tag ID, and is used to route special types of traffic, the main interface does not need to have an IP address when using sub-interfaces.

Sub-interfaces combined with Trunks allow us to connect, using one router port, multiple VLANs with multiple IP subnets, and have the router translate the traffic between the different VLANs. That way when a switch recieves a traffic going to a VLAN not connected to it, it can send the traffic to the RoS, and have the router **decapsulate** that traffic, then **encapsulate** it using the destination VLAN tag ID, then send it to the switch where it can forward it back to the destination device.

2.3 Configuring VLANs

There are three main commands used to configure VLANs on a CISCO switch, `‘vlan <tag>‘` to define a new VLAN, with a tag ID `<tag>`, for the switch. To assign a VLAN to a port, we first go to the interface then use the command `‘ switchport access VLAN <tag>‘`, or the command `‘switchport mode trunk ‘` to set it as a trunk. We can view the VLANs that a switch has by using the command `‘show VLAN‘`

Part II

Procedure

1 Access Control List

Here we built the following topology,

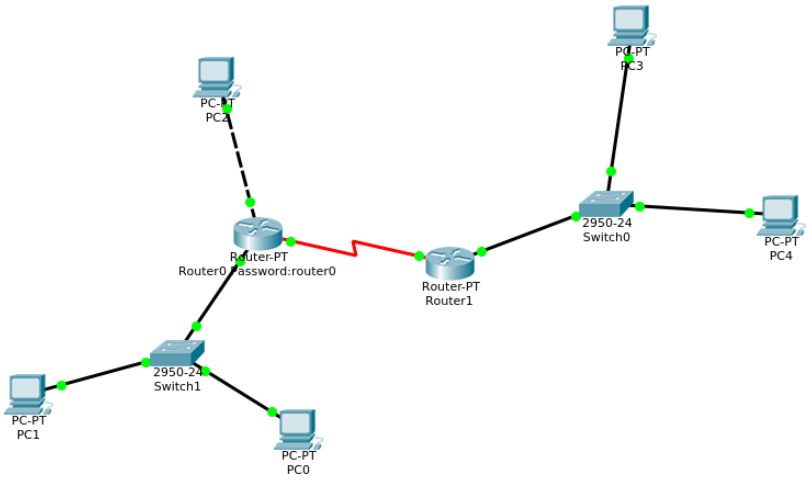


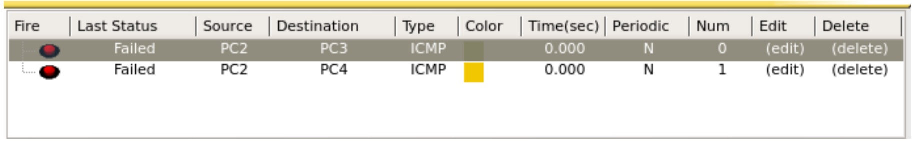
Figure 2: The topology used in ACL, table 1 shows the IP distribution.

Devices	IP address
PC0	193.168.1.3
PC1	193.168.1.2
PC2	193.168.0.2
PC3	193.168.2.2
PC4	193.168.2.3
Router0 ˜ Router1	10.0.0.1
Router0 ˜ PC2	193.168.0.1
Router0 ˜ PC0/1	193.168.1.1
Router1 ˜ Router0	10.0.0.2
Router1 ˜ PC3/4	193.168.2.1

Table 1: The IP addresses for the devices in the topology above.

We configured and enabled RIP on the routers. Then we started to test ACL use, we began by blocking PC2 from accessing 193.168.2.0/24 subnet, we used the commands shown bellow on Router 1, the first 2 commands create new ACL rules while the other two apply these rules to a specific interface as In or Out rules.

```
>access-list 10 deny host 193.168.0.2
>access-list 10 permit any

>int fa0/1
>ip access-group 10 in
```

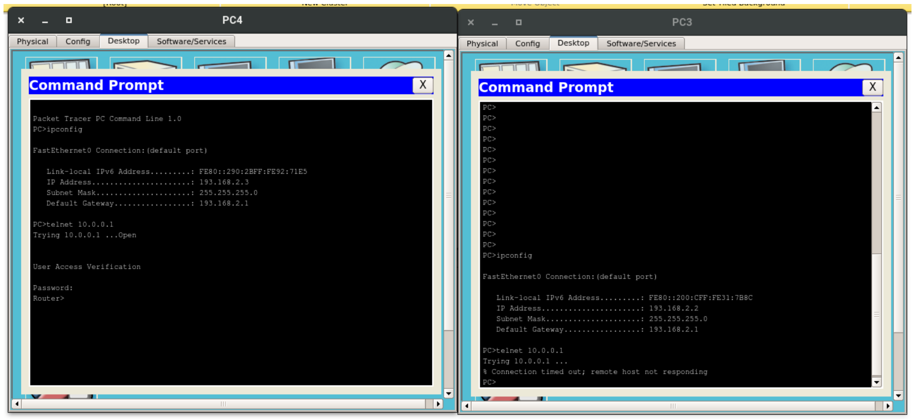


Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Failed	PC2	PC3	ICMP	▨	0.000	N	0	(edit)	(delete)
●	Failed	PC2	PC4	ICMP	🟨	0.000	N	1	(edit)	(delete)

Figure 3: The result of pinging PC3 and PC4 after blocking PC2 on Router1.

After that we applied Extended ACL to block PC3 from using the telnet protocl to access Router0. In Router0 we used the commands,

```
>access-list 130 deny tcp host 193.168.2.2 host
    10.0.0.1 eq telnet
>access-list 130 permit any any

>int se2/0
>ip access-group 130 in
```



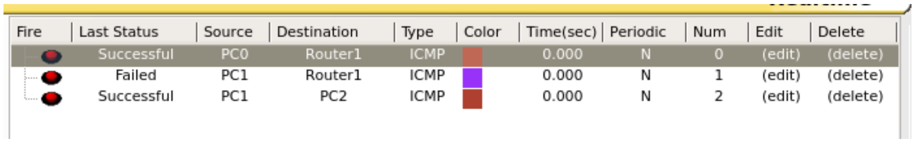
```
Packet Tracer PC Command Line 1.0
PC>ipconfig

FastEthernet0 Connection:(default port)

   Link-local IPv6 Address.........: FE80::290:2BFF:FE92:71E5
   IP Address......................: 193.168.2.3
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: 193.168.2.1

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...Open


User Access Verification

Password:
Router>
```

```
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>
PC>ipconfig

FastEthernet0 Connection:(default port)

   Link-local IPv6 Address.........: FE80::200:CFF:FE31:7B8C
   IP Address......................: 193.168.2.2
   Subnet Mask.....................: 255.255.255.0
   Default Gateway.................: 193.168.2.1

PC>telnet 10.0.0.1
Trying 10.0.0.1 ...
% Connection timed out; remote host not responding
PC>
```

Figure 4: Demonstrating that PC3 cannot telnet to Router0 after the commands.

The last thing we did was to block PC1 from accessing the serial connection between Router0 and Router1, the following commands were used,

```
>access-list 10 deny host 193.168.1.2
>access-list 10 permit any

>int se2/0
>ip access-group 10 out
```



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	PC0	Router1	ICMP	🟥	0.000	N	0	(edit)	(delete)
●	Failed	PC1	Router1	ICMP	🟪	0.000	N	1	(edit)	(delete)
●	Successful	PC1	PC2	ICMP	🟥	0.000	N	2	(edit)	(delete)

Figure 5: Showing the effect after blocking PC1 from accesing the serial connection.

2 VLANs

2.1 VLANs Part 1

We used the topology shown in figure 6 for the first part of the experiment.

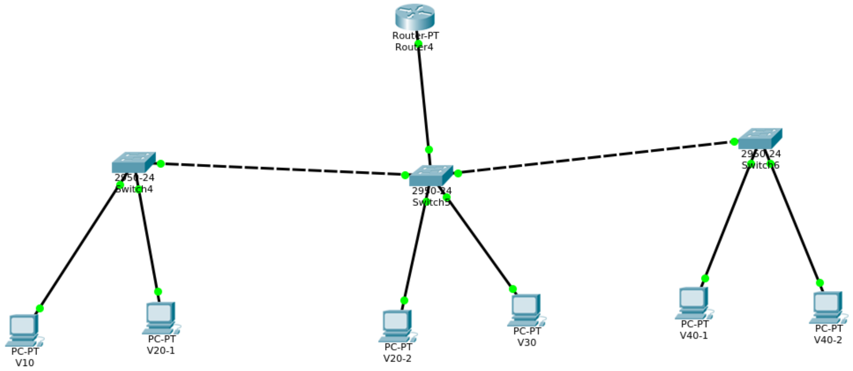


Figure 6: The topology used for the RoS part, the IPs of the devices are shown in the following figure 2.

Device Name	IP address	VLAN tag
V10	193.168.0.2	10
V20-1	193.168.1.2	20
V20-2	193.168.1.3	20
V30	193.168.2.2	30
V40-1	193.168.3.2	40
V40-2	193.168.3.3	40

Table 2: The IPs of the devices in the above topology.

The router has 4 sub-interfaces, with IPs shown in the following table.

Sub-Interface	IP
fa0/0.10	193.168.0.1
fa0/0.20	193.168.1.1
fa0/0.30	193.168.2.1
fa0/0.40	193.168.3.1

Table 3: Sub-interfaces of the router.

To configure the router's sub-interfaces we sued the following commands sequence,

```
>int fa0/0.10>ip address 193.168.0.1>int fa0/0.20>ip address
193.168.1.1>int fa0/0.30>ip address 193.168.2.1>int fa0/0.40>ip
address 193.168.3.1
```

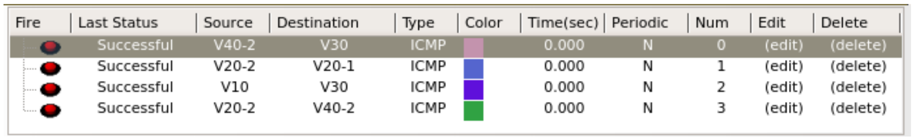
Table 4: The commands to configure the router.

And then the to configure the switches we used the commands shown in the follwoing table 5,

Switch	Commands
4	>vlan 10 >vlan 20 >int fa0/1 >switchport access vlan 10 >int fa0/2 >switchport access vlan 20 >int fa0/3 >switchport mode trunk
5	>vlan 10 >vlan 20 >vlan 30 >vlan 40 >int fa0/1 >switchport access vlan 20 >int fa0/2 >switchport access vlan 30 >int fa0/3 >switchport mode trunk >int fa0/4 >switchport mode trunk >int fa0/5 >switchport mode trunk
6	>vlan 40 >int fa0/1 >switchport access vlan 40 >int fa0/2 >switchport access vlan 40 >int fa0/3 >switchport mode trunk

Table 5: The commands used to configure the switches.

One note when configuring the switches, is that the middle switch in our topology, the one responcible for connecting with the RoS, needs to have full knowledge of all the VLANs out ther, in order to properly forward traffic from all VLANs to the RoS, without knowing this we struggled to get the network to work without any luck!



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	V40-2	V30	ICMP	🟥	0.000	N	0	(edit)	(delete)
●	Successful	V20-2	V20-1	ICMP	🟦	0.000	N	1	(edit)	(delete)
●	Successful	V10	V30	ICMP	🟪	0.000	N	2	(edit)	(delete)
●	Successful	V20-2	V40-2	ICMP	🟩	0.000	N	3	(edit)	(delete)

Figure 7: Ping commands between some of the devices after the configuration.

2.2 VLANs Part 2

In the second part of the experiment, we replaced the RoS with a Multilayer Switch, and kept the end hosts.

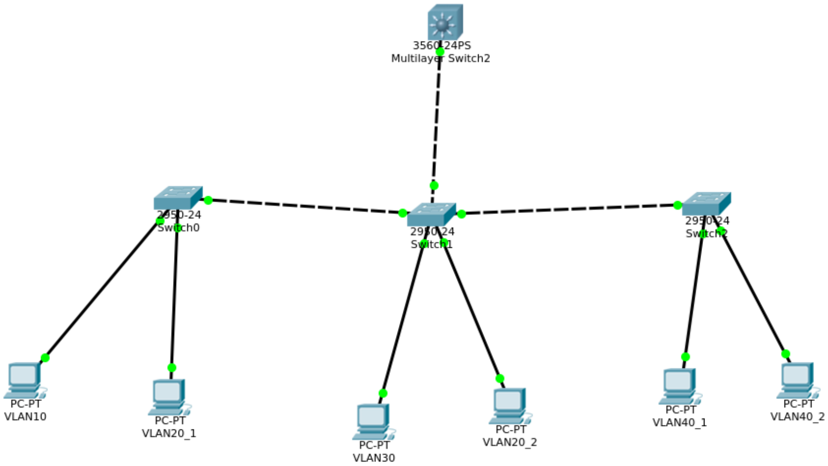
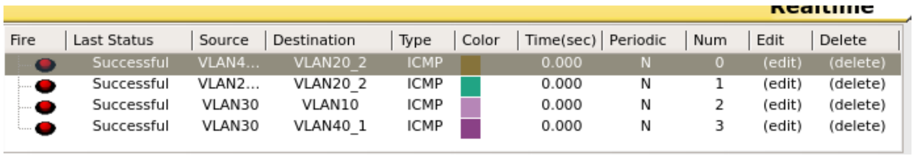


Figure 8: The topology with the Multilayer Switch instead of the RoS.

The devices used the same IPs as the previous part. The normal switches' configuration was unchanged, the Multilayer switch configuration was as follows.

```
>int fa0/1
>no switchport
>ip routing

>vlan 10
>switchport access vlan 10
>vlan 20
>switchport access vlan 20
>vlan 30
>switchport access vlan 30
>vlan 40
>switchport access vlan 40

>int vlan 10
>ipp add 193.168.0.1 255.255.255.0
>int vlan 20
>ipp add 193.168.1.1 255.255.255.0
>int vlan 30
>ipp add 193.168.2.1 255.255.255.0
>int vlan 40
>ipp add 193.168.3.1 255.255.255.0
```

Table 6: The commands to configure the Multilayer Switch.



Realtime

Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
●	Successful	VLAN4...	VLAN20_2	ICMP	🟫	0.000	N	0	(edit)	(delete)
●	Successful	VLAN2...	VLAN20_2	ICMP	🟩	0.000	N	1	(edit)	(delete)
●	Successful	VLAN30	VLAN10	ICMP	🟪	0.000	N	2	(edit)	(delete)
●	Successful	VLAN30	VLAN40_1	ICMP	🟪	0.000	N	3	(edit)	(delete)

Figure 9: The ping messages between some of the devices in the topology.

Part III
Conclusions

In these experiments, we began by discussing the uses and benefits of ACLs and how to configure them, from the security and privacy reasons of blocking certain traffic from reaching certain devices, to controlling the types of content being passed in the network. After that we began discussing the concept of VLANs, and how build logically sepaated networks, even using a single switch, how to connect these VLANs, what uses they have, from creating specialized networks for specific uses under an enterprise, to blocking broadcasts from reaching devices that mustn't recieve such broadcasts, and we encountered some deficulties in the setup of VLANs but we got over them evnetually with the help of our TA Odai.

List of Figures

List of Tables

References

[1] The lab manuals.

[2] Switch Virtual Interface | Wikipedia
`https://en.wikipedia.org/wiki/Switch_virtual_interface`