Lab 03: Network Segmentation part 2 - VLANs

Network Infrastructure Security (CSP)  
 2024-2025

© Henk Brouckxon, Nico Declerck, Chris Roets

# Introduction

## Lab concept

In the previous lab, we’ve used IPv4 subnetting to split the originally flat (= non-segmented) 10.0.0.0/24 network (Figure 2) into three separate subnets:

* A “Workstations” subnet for workstations that are used by employees
* A “Management” subnet for networking devices (e.g. switches) and administrative servers/workstations that should not be accessible for most employees.
* A “DMZ” (DeMilitarized Zone) subnet for servers that provide internet-facing services like webservers and mail servers.

Even though subnetting does provide a separation/segmentation between the different subnets at the IP level (OSI layer 3) and above, it should however be noted that it does **NOT** separate these networks at the ethernet level (OSI layers 1 and 2). Due to this, a hacker who gets access to the network (through wifi or an accessible ethernet port) would very easily be able to hop from one subnet to the other by configuring their device with an appropriate IP address and subnet mask within each targeted subnet. Subnetting should therefore not be considered as a secure way of segmenting the network, and additional measures are needed to improve the separation between the subnets.

Separation/segmentation of the subnets at OSI layer 2 could for example physically be achieved by assigning separate -not-interconnected - switches to each subnet, but especially for large networks that span a whole building or even a complete campus, this would be very inefficient. During this lab, we will therefore look into implementing the most frequently used (and very effective) method of network segmentation: Virtual LANs or VLANs. To allow some level of (strictly controlled[[1]](#footnote-2)) communication between the subnets, e.g. to get access to the internet or the servers from the Workstations subnet, we’ll also implement inter-VLAN routing in the firewall device.

## Learning goals

* VLAN based network segmentation
* Inter-VLAN routing

## Practicalities and prerequisites

You will need the following:

* A laptop/desktop 😊
* Access to your NIS environment, which is configured as expected at the end of lab 2.
* For this lab, you should start up the following devices in the NIS network:
  + Firewall
  + Switch
  + Workstation-01
  + Workstation-02
  + Server-01
  + Server-02
  + Adminserver
  + Adminstation

The “network” device should NOT be started up, since this will interfere with the operation of the switch, and may bring down your complete environment.

A diagram of a computer system

Description automatically generated

Figure 1 Equivalent physical layout for the NIS network

A computer icons on a black background

Description automatically generated

Figure 2 Logical layout of the NIS network at the start of lab 1 (flat network)

*Logical layout of the NIS network at the end of lab 2*

# Making sure everything is ok to start…

At the end of the previous lab, we changed one of the ports of the switch into a SPAN port, which copies the traffic that flows to/from one of the other ports to your adminstation device for monitoring. This however also causes the adminstation device to be excluded from all normal network access. Because we want to go back to normal network operation, it is therefore important to check that the SPAN port is no longer active.  
**remark**: a reboot of the “switch” device will probably reset the network and remove the SPAN port, but it’s safer to check now than to have to troubleshoot access problems later on.

1. Log into the switch host and remove all SPAN/Mirroring ports with the following command [1]:

ovs-vsctl clear bridge nis-lan mirrors

We are clearing the SPAN/Mirror ports we did previous lab, from the bridge (unlinking)

A screenshot of a computer

Description automatically generated

I guess the thing I did yesterday, it did not really save because of the amazing servers.

A black screen with white text

Description automatically generated

1. Verify the IP addresses and subnet configuration of the different hosts, using the table you defined in lab 2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **hostname** | **IP address** | **Subnet mask** | **Default Gateway** | **DNS** | **Operating System** |
| **workstation-01** | 10.0.0.2/26 | 255.255.255.192 | 10.0.0.1/26 | 10.0.0.5 | Alpine Linux v3.19 |
| **workstation-02** | 10.0.0.3/26 | 255.255.255.192 | 10.0.0.1/26 | 10.0.0.5 | Alpine Linux v3.19 |
| **switch** | 10.0.0.66/28 | 255.255.255.240 | 10.0.0.65/28 | 10.0.0.5 | Alpine Linux v3.19 |
| **adminserver** | 10.0.0.67/28 | 255.255.255.240 | 10.0.0.65/28 | 10.0.0.5 | Ubuntu 22.04.4 LTS |
| **adminstation** | 10.0.0.68/28 | 255.255.255.240 | 10.0.0.65/28 | 10.0.0.5 | Kali GNU/Linux Rolling 2024.2 |
| **server-01** | 10.0.0.82/29 | 255.255.255.248 | 10.0.0.81/29 | 10.0.0.5 | Alpine Linux v3.19 |
| **server-02** | 10.0.0.83/29 | 255.255.255.248 | 10.0.0.81/29 | 10.0.0.5 | Alpine Linux v3.19 |
| **firewall** | 10.0.0.5/24 | 255.255.255.0 | 192.168.16.1 | 127.0.0.1 + 192.168.16.5 | FreeBSD 14.1-RELEASE-p3 |

Table 1 Host documentation (after subnetting)

1. Verify that devices can ping each other within the same subnet (only use IP addresses to ping the hosts, DNS will not work for now), but not between different subnets.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A computer screen shot of a black screen

Description automatically generated

# Subnet hacking: hopping from one subnet to another

If everything is configured correctly, adminstation should only be able to connect/ping to the devices within its own subnet, i.e. adminserver and switch.

1. Ping from adminstation to server01 (only use IP addresses since DNS is not operational for now). Do you get a reply?

A screenshot of a computer

Description automatically generated

It says port unreachable (which makes sense, since we do not have a way to connect to the other one)

1. Change the IP address and subnet mask of adminstation in such a way that it becomes part of the “servers” subnet, and show that it can now ping to the server01 and server02 hosts (again, only use IP addresses). Insert a screenshot of these pings below.

Initial

A screenshot of a computer

Description automatically generated

Afterwards

A screenshot of a computer

Description automatically generated

Sudo systemctl restart NetworkManager

It works

A screenshot of a computer screen

Description automatically generated

1. Change the IP address and subnet mask of adminstation back to its original configuration

# Implementing VLANs in the “switch” device

As a first step, we’ll document how we will implement the VLANs in our network.

1. Assign a unique VLAN id number to each subnet, and add this in the Subnet table from lab 2:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **VLAN id** | **VLAN/Subnet name** | **Network address** | **prefix** | **netmask** |
| **1** | **Workstations** | 10.0.0.0 | /26 | 255.255.255.192 |
| **2** | **Management** | 10.0.0.64 | /28 | 255.255.255.240 |
| **3** | **DMZ(DeMilitarized Zone)** | 10.0.0.80 | /29 | 250.250.250.248 |
| **666** | **Unused VLAN** |  |  |  |

Table VLAN/Subnet table

1. Because VLANs work on OSI layer 2, configuration of the VLANs will not happen on the IP level (OSI layer 3), but the actual configuration of VLANs will be assigned to the physical ports of the switch. Use Figure 1, Table 1 and Table 2 to determine what VLAN should be assigned to each port of the switch. For now, do not include eth0 in your configuration since this will be a special case. Unused ports should be assigned to an unused VLAN id (typically id 666), called the “black hole” VLAN.

|  |  |  |
| --- | --- | --- |
| Port name | VLAN/Subnet Name | VLAN ID |
| eth0 | - | - |
| eth1 | Workstations | 1 |
| eth2 | Workstations | 1 |
| eth3 | DMZ | 3 |
| eth4 | DMZ | 3 |
| eth5 | Management | 2 |
| eth6 | Management | 2 |
| eth7 | Unused VLAN | 666 |
| eth8 | Unused VLAN | 666 |

Table 3 Port configuration table (omit eth0 for now)

1. Now log in to the switch host, and configure the VLAN for each of the ports of the switch.

**Hint:** use the command

ovs-vsctl set port <port-name> tag=<VLAN id>

I do not know if I am doing things right, but I think I have to just set ports with tags, and the tags are the ones I did.

A computer screen shot of a black screen

Description automatically generated

and check the configuration with

ovs-vsctl show

these ports are called “VLAN access ports” because they give the hosts access to the different VLANs

A computer screen shot of a black screen

Description automatically generated

# Subnet hacking part 2: hopping from one subnet to another with VLANs?

1. Change the IP address and subnet mask of adminstation in such a way that it becomes part of the “servers” subnet, as in question 5. DO NOT change the VLAN id of its access port! Can it again ping to the server01 and server02 hosts (again, only use IP addresses)?

Does not seem like I can do it after tagging

A screenshot of a computer screen

Description automatically generated

As like the OpenVSwitch knows that the device is supposed to be part of another segment, hence it disallows things like that. So last time I could “teleport”myself because the systems were not physically wired. Now it is another way.

1. Change the IP address and subnet mask of adminstation back to its original configuration in the management subnet.

A screenshot of a computer program

Description automatically generated

Done

# VLAN Trunk, Inter-VLAN routing and other configurations

In the configuration up to here, we focused on separating the different subnets, which causes communication between subnets (and with the internet) to no be longer possible. To restore this communication, we’ll use the firewall device as a router that manages the inter-VLAN traffic, and that later will be able to restrict access by using firewall rules. To this end, we’ll configure a virtual interface for each VLAN/subnet in the firewall device, and add the suitable IP address and subnet mask to these interfaces. For efficiency, a single interface (a VLAN trunk) will be used to handle the traffic ommunication between the router and the switch.

1. Extend the host documentation table with extra virtual interfaces in the firewall device, such that it has an interface in each VLAN/subnet, with suitable IP addresses and subnet masks.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **hostname** | **IP address** | **Subnet mask** | **Default Gateway** | **DNS** | **Operating System** |
| **Workstation01** | 10.0.0.2 | 255.255.255.192 | 10.0.0.1 | 10.0.0.0.1 | Alpine Linux v3.19 |
| **Workstation02** | 10.0.0.3 | 255.255.255.192 | 10.0.0.1 | 10.0.0.0.1 | Alpine Linux v3.19 |
| **Server01** | 10.0.0.82 | 255.255.255.248 | 10.0.0.81 | 10.0.0.81 | Alpine Linux v3.19 |
| **Server02** | 10.0.0.83 | 255.255.255.248 | 10.0.0.81 | 10.0.0.81 | Alpine Linux v3.19 |
| **Adminserver** | 10.0.0.67 | 255.255.255.240 | 10.0.0.65 | 10.0.0.65 | Ubuntu 22.04.4 LTS |
| **adminstation** | 10.0.0.68 | 255.255.255.240 | 10.0.0.65 | 10.0.0.65 | Kali GNU/Linux Rolling 2024.2 |
| **Switch** | 10.0.0.66 | 255.255.255.240 | 10.0.0.65 | 10.0.0.65 | Alpine Linux v3.19 |
| **opnsense** | 10.0.0.5 | 255.255.255.0 | 192.168.16.1 | 192.168.16.5 + 127.0.0.1 | FreeBSD 14.1-RELEASE-p3 |

For the sake of being safer and everything, I will change the DNS on the machines to be the default gateway.

To update on Alpine:

* Edit /etc/resolv.conf
* Change nameserver, and save
* Doas service networking restart

1. Extend the port configuration table for the switch, with eth configured as a trunk for the suitable VLAN ids.

|  |  |  |
| --- | --- | --- |
| Port name | VLAN/Subnet Name | VLAN ID |
| Eth0 | - | [1,2,3] |
| Eth1 | Workstations | 1 |
| Eth2 | Workstations | 1 |
| Eth3 | DMZ | 3 |
| Eth4 | DMZ | 3 |
| Eth5 | Management | 2 |
| Eth6 | Management | 2 |
| Eth7 | Unused VLAN | 666 |
| Eth8 | Unused VLAN | 666 |
|  |  |  |

Table 4 Port configuration table (including eth0)

1. Now log in to the switch and configure its port eth0 as a trunk port for the applicable VLAN ids

doas ovs-vsctl set port eth0 trunks=1,2,3

A screen shot of a computer

Description automatically generated

1. On the side of the firewall/router, we’ll of course also need to configure the suitable port as a VLAN trunk for the same VLAN ids. Log in to the firewall’s web interface (you may need to temporarily reconfigure the ports of the switch and/or the IP address of adminstation), and add the appropriate virtual interfaces to the port that connects to the switch. For each virtual interface, this will be a multi-step process:
   1. Create a VLAN with the suitable ID on the router:   
      Interfaces -> Other types -> VLAN -> Add
   2. Create a new virtual interface:  
      Interfaces -> Assignments -> New interface
   3. Enable the interface.

Now I guess it is the time to go into the OPNsense and create new VLANs?

Interfaces > Other Types > VLAN

Add new Interface

A screenshot of a computer

Description automatically generated

I set the parent to LAN (as we need this NIS network, right?) and also the tag, and description (i left the Device untouched)

The go to Assignments, and add the new interfaces.

A screenshot of a computer

Description automatically generated

After you add all of them, then to enable you need to go Interfaces > [Name of Interface] > Click Enable Interface > Ipv4 configuration set to ipv4 and set ipv4 address to the ip address of gateway (default) within the network. For example the DMZVLAN will have 10.0.0.81.

Repeat this for every of them, and then check the overview.





It even understands the network, due to the netmask you set up.

Also here comes the part why did I set the default gateway for all the machines.

1. Test and confirm that the communication between the different VLANs and subnets works.

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

1. Log in to the switch from your adminstation, and run the command

ovs-vsctl show

Make a screenshot of the output of this command, that also shows your hostname and personal account, and paste this screenshot here.

Spent like 25 minutes trying to fix the issues, since I could not ping from the switch machine, or to it.

The issues:

* Forgot to set the default gateway (apart from DNS) in the /etc/network/interfaces
* The port nis-lan can also have a tag, and since this machine is supposed to be in the management, then it should have the Management tag.

A computer screen shot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

# Optional: listening in on the VLAN Trunk

1. Configure the port of the switch device where adminstation is connected to be a SPAN/Mirror port for the traffic (incoming and outgoing) that passes through the trunk (eth0) port, and capture this traffic using wireshark. Can you distinguish the traffic that corresponds to the different VLANs in the VLAN trunk? Can you see why a VLAN trunk port is sometimes also called a “tagged” port?

Sorry but I am too dead to do this. Have a nice day/evening/morning.

But from the Computer Networks, as far as I remember, the VLANs also like tag the packets they send to like where they are directed and from where they come.

1. Remove all SPAN/Mirroring ports with the following command [1]:

ovs-vsctl clear bridge nis-lan mirrors

# Bibliography

|  |  |
| --- | --- |
| [1] | "OpenVSwitch," [Online]. Available: https://www.openvswitch.org/. [Accessed 10 2024]. |
| [2] | "OpenVSwitch - basic configuration FAQ," [Online]. Available: https://docs.openvswitch.org/en/latest/faq/configuration/. [Accessed 09 2024]. |
| [3] | "OpenVSwitch: ovs-vsctl man page," 21 06 2024. [Online]. Available: https://www.openvswitch.org/support/dist-docs/ovs-vsctl.8.txt. |

1. In the next lab, we will implement firewalls to control the traffic between the different subnets! [↑](#footnote-ref-2)