Attack surface of VMWare/Virtualbox/ESXi virtual switches

*Hilary Tesfay Gebreslassie 241696*

*Fabio Cangeri 249495*

## Virtual Switch

A virtual switch (or vSwitch) is a software program that enables one virtual machine to communicate with another with the same protocols used in physical switches, without the need for additional networking hardware. In general, vswitches can be interfaced to the physical network by simply associating them with one or more physical interfaces available in the hosts.

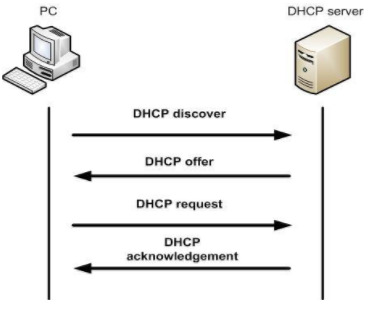
## vSwitch vs Physical Switch

The main differences are:

* Typology: vSwitch is a software component (implemented on a host machine) while physical switch is a hardware component (made up of ports, circuits and cables);
* Connectivity: Physical switches connect physical devices in a network using physical cables while vSwitches connect devices over virtual network interfaces and virtual network connections;
* Network Control: unlike physical switches, they do not need any phase of learning the MAC addresses for compiling the MAC-Address table, because they know in an authoritative way which devices are connected to each port.

## DHCP

When we enter a network there is an automatic configuration that allows us to obtain parameters to enter the network via DHCP. The parameters to enter the network are: a valid IP address, a valid DNS address of a DNS resolver, and a valid gateway address and we must know in which collision domain we are (i.e. know who our neighbors are and who are outside our collision domain domain).



Generally a device accepts the first offer that arrives from a DHCP server and DHCP does not implement any security machine, this means that any station that shares the collision domain with the attacking station can receive fake DHCP offers. If the attack is successful, the attacker can choose the victim's DNS, IP and gateway.

Generally the DHCP offer is unicast, and can also be broadcast.

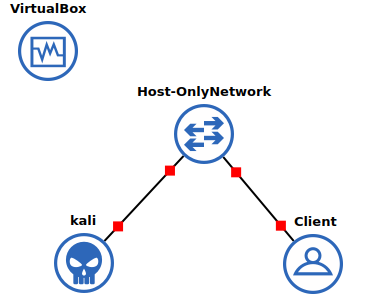
During the lease time, the client has the option to accept the offer and use the assigned IP address. If the client accepts the offer, the DHCP server notes the assignment and the client begins using the assigned IP address for the duration of the lease time. At the end of the lease time, the client must renew the IP address grant through a process known as DHCP Renewal or request a new IP address through a new DHCP request.

## DHCP starvation e spoofing attacks

In DHCP starvation, the attacking client sends so many DHCP Discover messages with fake MAC addresses that the DHCP pool fills up and the server is unable to serve valid clients. After performing the starvation attack, the attacker can set up a rogue DHCP server and start serving the victim machine with fake IP addresses. This way the attacker can perform a man in middle attack which captures the client's request by forwarding to the server and receives the response from the server and sends it to the client.

## Configuration

To perform this attack we used the VirtualBox virtual environment, version 6.1.38 hosted on a machine with Ubuntu 22.04.2 LTS. To simulate the vSwitch we used the Host Only Network adapter. To represent the client we used “lubuntu-22.04.2-desktop-amd64” (light version of Ubuntu) but it is not as relevant as the attacker where we mounted the OS “kali-linux-2023.2-virtualbox-amd64”.



## Why Host Only Network

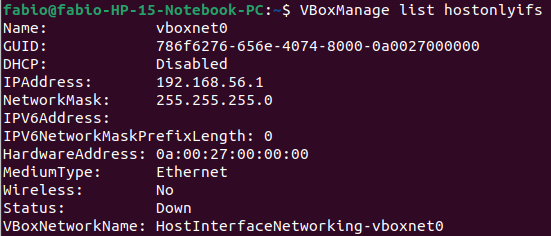
The VirtualBox Host Network Manager can be considered a virtual switch because it allows you to create and manage virtual networks within the VirtualBox environment. In the context of VirtualBox, Host Network Manager serves as a control interface for creating and configuring virtual networks. It provides similar functionality to a physical switch, such as the ability to define network parameters, assign IP addresses, configure DHCP services, and establish traffic routing rules. By creating virtual networks using Host Network Manager, you can connect multiple virtual machines to the same virtual network, allowing them to communicate with each other and with the host machine. This virtual switch feature allows for flexible network configurations and facilitates testing, development, and other scenarios where virtualized network environments are needed. It is important to note that while Host Network Manager provides virtual switch-like functionality within the VirtualBox environment, it operates at the host level and does not extend beyond the VirtualBox virtualization layer to interact with physical network switches.

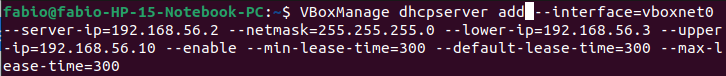
## Creation and configuration of the Host Only Network

Once you have created a virtual adapter in the Host Only Interface:



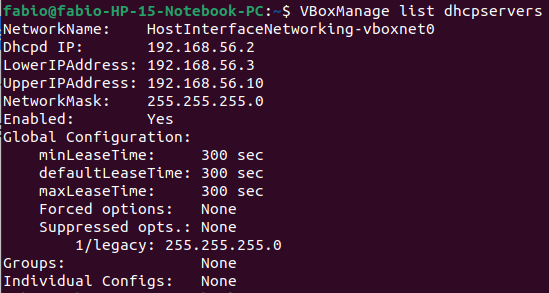
let's check that it was actually created:



Now let's move on to configuring the DHCP server (don't set the time lease in the demo):  


Among the parameters that we can set is that of the time lease where to simulate our attack with a low timing. Each device requires to renew its ip every 300 seconds. The time of 300 seconds is not very relevant, more than anything else the concept that when a user closes the computer, his IP will be assigned to another device, as it will not renew it, is useful.

Let's verify that our DHCP server is configured correctly:



Once this is done, connect each device to the Host Only Network from the settings.

## Run of the attack

From kali we run yersinia to perform the DHCP starvation attack: sudo yersinia -G.

We execute the attack until we fill the vSwitch requests (a few seconds) and then we don't close yersinia.

We try to connect a machine to get a new IP and see that it fails. To force a new computer to ask for a new IP: sudo dhcp -r <network adapter> (to remove its IP) and then sudo dhcp -v <network adapter> (to get a new IP).

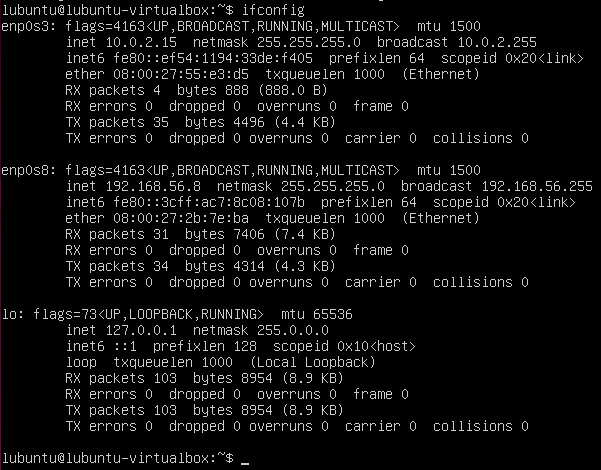
Through wireshark but also through Ettercap we can see the DHCP requests. Once our yersinia attack is closed we can see the MAC of the device that wants a new IP. When we close the Yersinia attack we have a few minutes.

At this time, when we stop our DHCP starvation attack, we listen to see if anyone is making a DHCP request. If he is advancing them we make a rogue DHCP server which will give him a fake IP address, DNS and gateway.

Ettercap (or via yersinia) is set like this:

| Naturally the parameters can be changed, what is important is the mask and the DNS server Ip which must be the attacker's IP in the collision domain. |  |
| --- | --- |

Now let's start a regular client. Let's run the ifconfig command to see a bit of your network cards.



As we can see, the "enp0s3" card is the one connected to the NAT that allows you to go to the internet, while "enp0s8" is the one connected to our vSwitch. Currently “enp0s8” has a valid IP address, assigned by the real DHCP server, i.e. our vSwitch. Let's take a closer look at the configuration of this card:



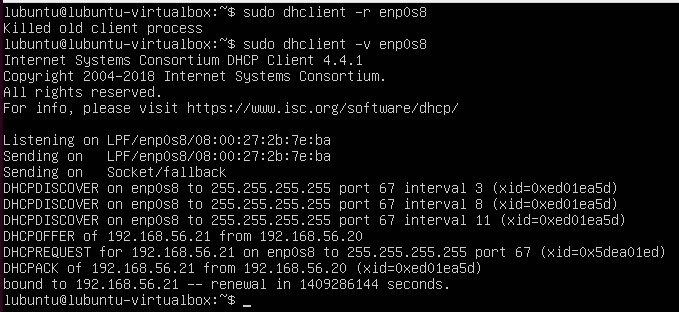
As we can see, the validity of the IP assigned by our DHCP will expire in 51 seconds. When the user closes his PC, we initiate our DHCP starvation attack, since the user's certificate expires and we take it with our DHCP attack. When he then turns the PC back on we see DHCP requests and then we start our rogue DHCP server to assign it our own IP.

We speed up this practice by executing the command sudo dhclient -r enp0s8 (which is used to delete the IP assigned to that network card) followed by the command sudo dhclient -v enp0s8 (to request a new IP).

When we perform the DHCP starvation attack, the user cannot find a valid IP:



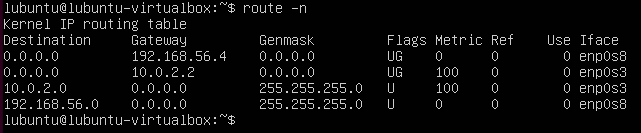
Now let's start our DHCP rogue server:



The user's device sends a series of unsuccessful DHCPdiscovers, the first because the legitimate DHCP server is under attack. Then starting our rogue DHCP server it is assigned an IP.

Let's do a traceroute test to verify the routing from that device to another host connected to the vSwitch. What we expect is that he quietly uses the vSwitch to communicate with his neighbors, i.e. the devices of the collision domain, instead to go on the internet the packets are routed on kali, because the DNS gateway entered via the rogue DHCP server imposes this broken and therefore we are in a passive man in the middle position currently. From here on you can go through an active attack.

Before seeing the traceroute we need to configure our kali machine. From the route table of the attacked client we note that each packet with a destination different from that of the collision domain of the network cards is sent either to 192.168.56.4 (our fake DNS) or to the network card connected to the internet (10.0.0.2) . Between the two, the one with the lowest metric is preferred, i.e. ours.

Now we have to forward all the packets on the eth0 card, i.e. the one connected to the internet on the kali machine, with the exception of the packets that have the IP belonging to our collision domain which must remain on the eth1 interface otherwise we don't use the vSwitch and we cannot reach the our neighbors.

First we need to enable traffic forwarding in the kernel. Let's open the file/etc/sysctl.conf using a text editor and make sure the following line is present and uncommented: net.ipv4.ip\_forward=1. We save with sudo sysctl -p.

Now we need to configure the iptable rule as described above.

sudo iptables -A FORWARD -i eth1 -o eth0 -m iprange ! --src-range 192.168.56.0-192.168.56.255 -j ACCEPT

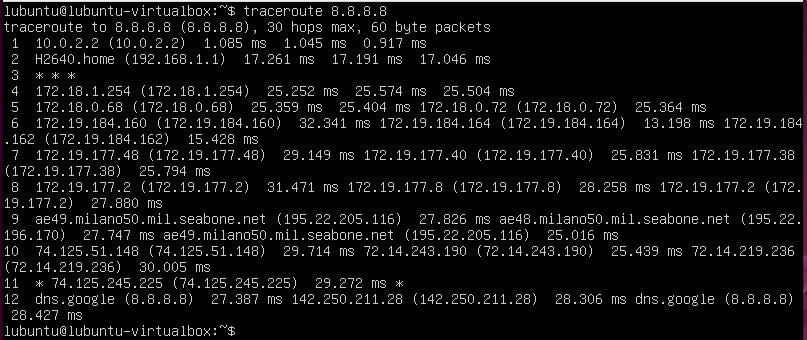
followed by the MASUADE to allow return traffic:

sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

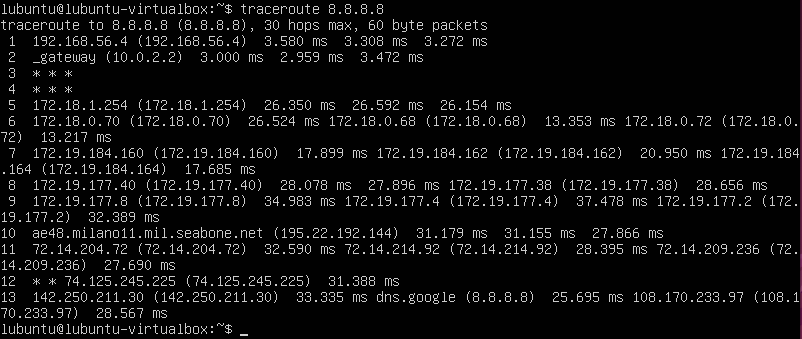
We save with this command, making the change persistent:

sudo iptables-save -f /etc/iptables/rules.v4

Now finally we can see the comparison of traceroutes before and after the attack



(Before: note the first node crossed which is the IP of the network card mounted on the client that allows you to go to the internet)



(After: Note the first node traversed which is kali's IP)

## Mitigation

There are several ways to mitigate this type of attack:

* Set static IPs from your machines being aware of the range of IPs that can be assigned by the vSwitch;
* Viewing the dhcp capture to see if we are under this attack;
* Checking our IP address;
* Setting a very high time for IP expiration, this means that only as soon as we attack the device could we be under attack.

## How I implement it

Runna kali, do the DHCP starvation attack; runnig lubuntu; perform the DHCP rouge server attack; You assign the IP to lubuntu; turn off kali and reopen it and do the iptables and then it goes.

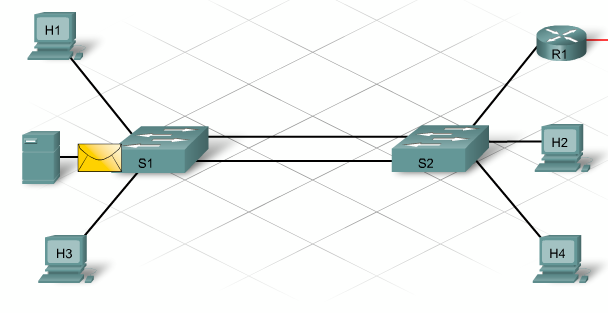
## Bibliography

[**VBoxManage manual**](https://www.virtualbox.org/manual/ch08.html#vboxmanage-hostonlynet)

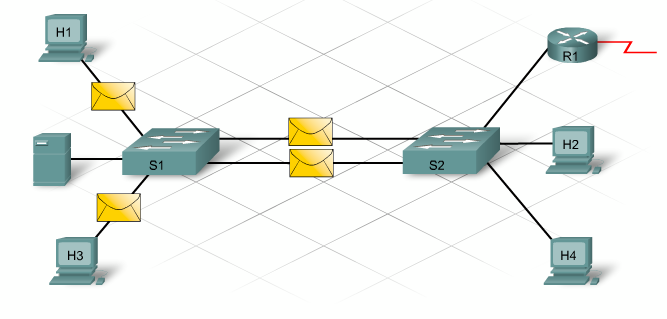
## Spanning Tree Protocol (SPT)

In many situations it is useful to connect multiple switches together, to ensure network expansion; it is also useful to have redundant connections, since if a cable connecting two switches breaks or the ports where it is connected have problems, communication between hosts on the different switches is lost.

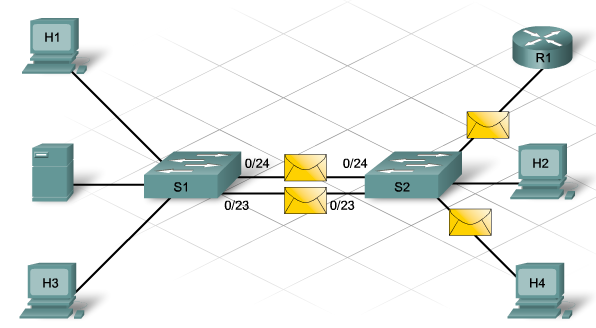
The server (on the left side) broadcasts a message:



Switch S1 will forward the frame on all ports except the incoming one:



Switch S2 receives 2 frames and processes them. What arrives on the 0/24 interface, being of the broadcast type, will be replicated on all ports except 0/24 (the arrival one). The same happens for the frame arrived on interface 0/23, replicated on all ports except 0/23 (the arrival one). At this point 2 identical frames restart towards the interfaces of switch S1:



switches with STP enabled check the network for the existence of cyclic paths (loops): if they are detected, some connection ports are blocked, thus breaking the ring generated by redundant paths. The other ports are left active for frame transmission.

STP defines a tree that traverses all switches in the network. To avoid switching loops, the Spanning Tree Protocol:

* forces some interfaces to a locked or standby state
* leaves other interfaces in the forwarding state
* configure the network by activating the path left in standby, in case the main path is no longer available

## How to exploit this attack?

[ESXI 5.0 Deficiency](https://kb.vmware.com/s/article/2047822)