

Assignment F.1 - Home Pentesting Lab

💡 It is extremely helpful to have the means to investigate malware, vulnerabilities and practice your penetration testing in a controlled lab environment. In this assignment you will be shown a means in which to construct such a lab using VMWare Workstation on either a dedicated system or one used for your daily work.

This activity is divided into 4 milestones and is designed to be completed over a course of 2-3 weeks. It's important you don't wait until the last minute to tackle this assignment.

Assumptions

- You have a personal laptop or PC that is capable of running VMWare Workstation or Fusion
- This system has 16GiB or RAM or better and at least 100 GiB of spare space. External fast media can also work.

OR

- You have fast reliable removable storage and you are prepared to hang out in a CYBER lab to create your externally stored lab. You will need to pay particular attention to Virtual Machine versioning and networking so that you can move between workstations. You will need to be super careful about properly handling your external media.

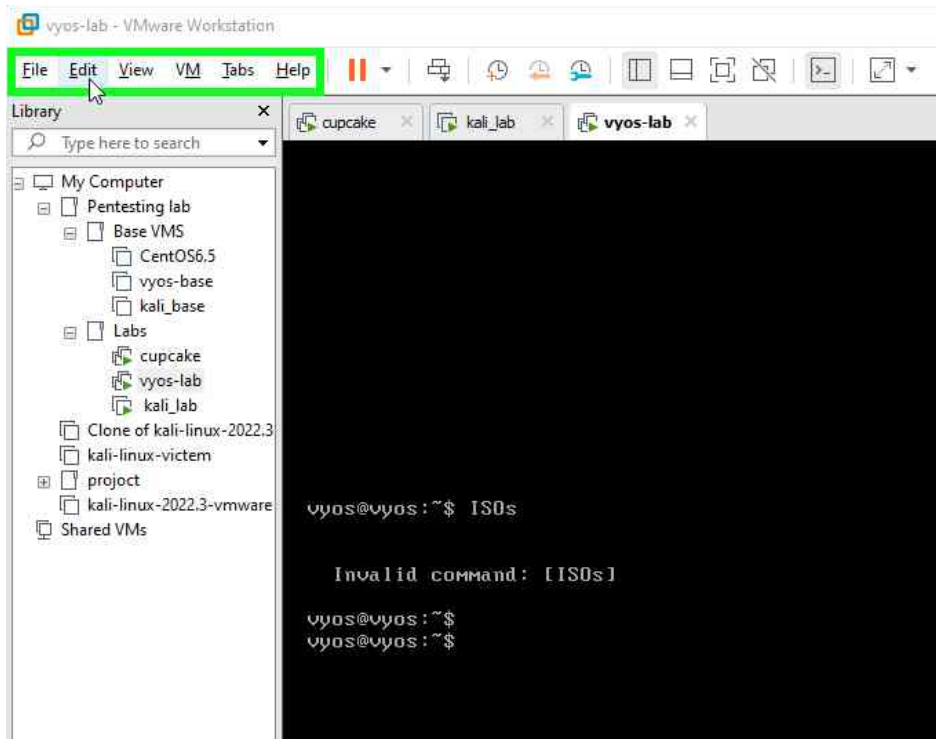
Milestone 1 - VMWare and Kali

Instructions

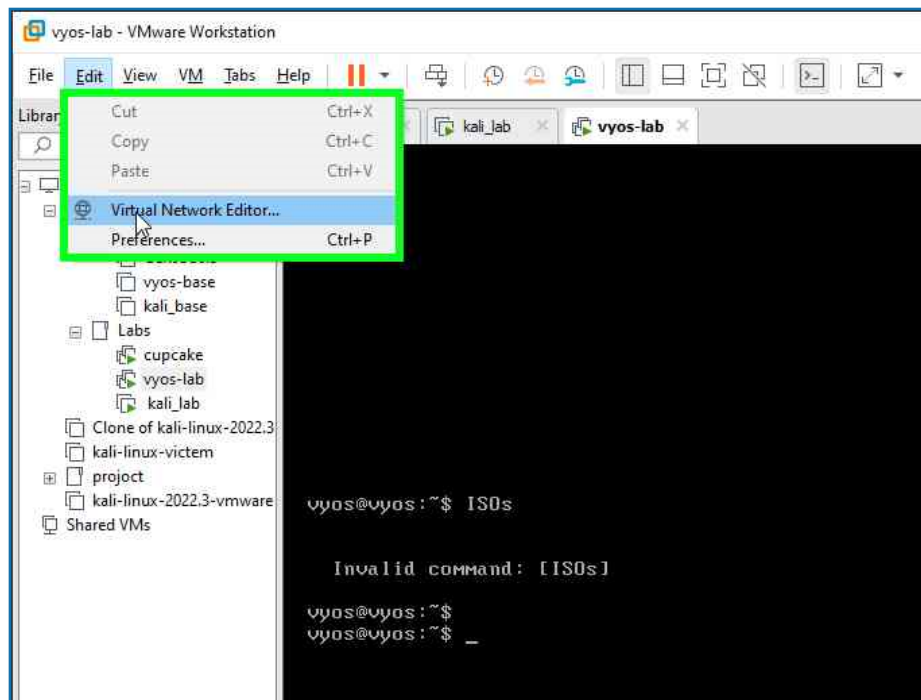
Watch the following [video](#)

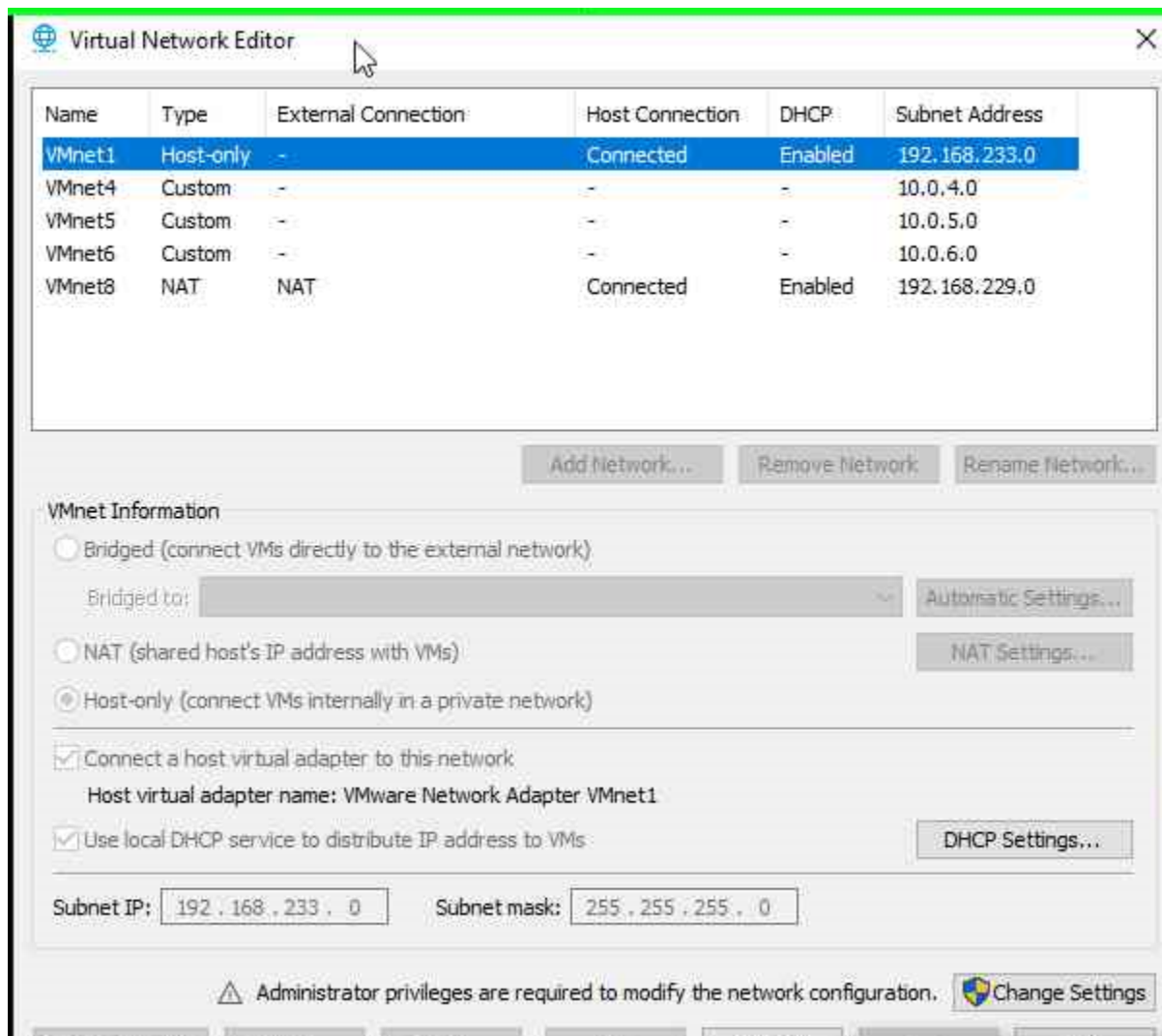
Network in VMware Workstation

Step 1: left click on "Edit (menu item)" in "VMware Workstation"



Step 2: left click on "Virtual Network Editor... (menu item)"





ISOs

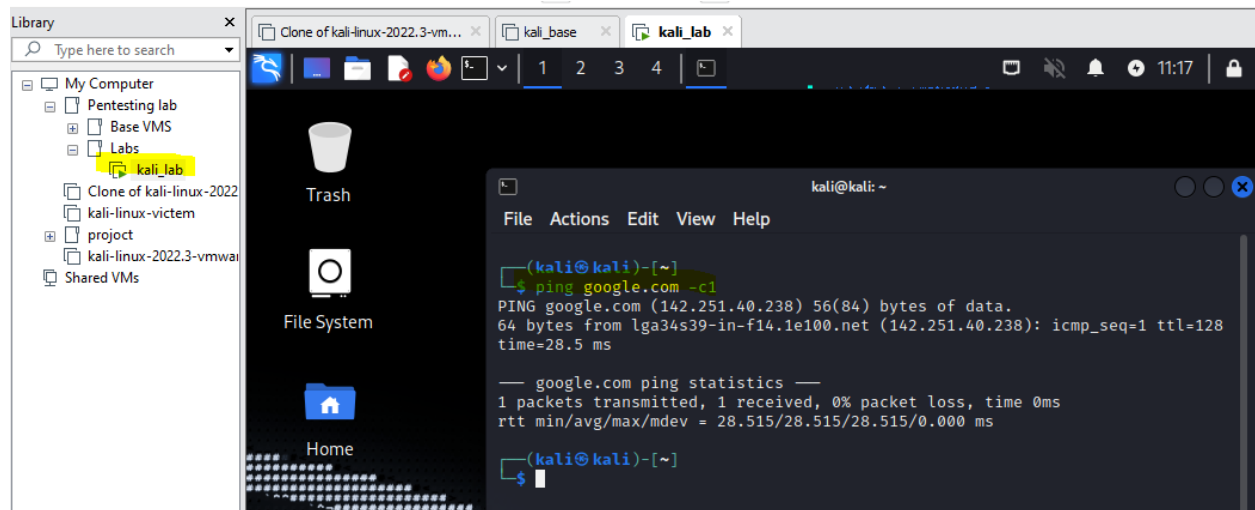
pull down the following ISO's or prebaked VMs to your host or fast external media.

- centos vulnerable [iso](#)
- xubuntu <https://xubuntu.org/>
- kali linux [iso](#) or [vm](#) (feel free to upgrade)
- vyos [iso](#) or [vm](#)
 - If you want to know how to build prepare your own ova, see [this](#)
 - The ova uses the vyos/Ch@mpl@1n!22 username/password combination.

Kali

- Create a Kali VM called kali-base. Get Kali installed using NAT Networking.
- Make a Linked Clone of kali-base called kali-lab

Deliverable 1. Provide a screenshot of your kali-lab vm pinging google.com similar to the one below



Milestone 2 - vyos

Instructions

Watch the following [video](#)

- Create a Base Image for vyos called vyos-base
- Make sure to clear hw-id's as instructed in the video
- Create a "Base" snapshot
- Create a Linked Clone called vyos-lab
- Change eth1 to be VMNET 5

Here are the vyos configuration commands from the video. Don't forget the configure, commit and save commands.

```
configure
set interfaces ethernet eth0 address '192.168.229.10/24'
set interfaces ethernet eth0 description 'Nat on VMware Host'
set interfaces ethernet eth1 address '10.0.5.2/24'
set interfaces ethernet eth1 description 'VMNET5-RANGE'
```

```

set protocols static route 0.0.0.0/0 next-hop 192.168.229.2
set service ssh listen-address '192.168.229.10'
set system name-server '192.168.229.2'
set service ssh listen-address 192.168.229.10
commit
save

```

💡 VYOS's nightly images are sometimes a moving target, with new features being introduced and old syntax being deprecated. If you run into an issue with the latest and greatest iso image, let your instructor know.

Deliverable 2. Provide a screenshot similar to the one below that displays your interfaces as well as a successful ping against google.com

```

Codes: S - State, L - Link, u - Up, D - Down, A - Admin Down
Interface      IP Address      S/L  Description
-----
eth0           192.168.229.10/24 u/u   Nat on VMNET8
eth1           10.0.5.2/24     u/u   Range on VMNET5
lo             127.0.0.1/8     u/u
              ::1/128
vyos@vyos:~$ ping google.com -c1

Invalid command: /usr/libexec/vyos/op_mode/ping.py google.com -c1 [-c1]

<nocomps>vyos@vyos:~$ ping google.com
PING google.com (142.250.80.110) 56(84) bytes of data.
64 bytes from lga34s36-in-f14.1e100.net (142.250.80.110): icmp_seq=1 ttl=128 time=28.1 ms
64 bytes from lga34s36-in-f14.1e100.net (142.250.80.110): icmp_seq=2 ttl=128 time=28.3 ms
64 bytes from lga34s36-in-f14.1e100.net (142.250.80.110): icmp_seq=3 ttl=128 time=28.6 ms
^C
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 28.093/28.343/28.617/0.214 ms
vyos@vyos:~$ _

```

Milestone 3 - the centos target

Instructions

Watch the following [video](#)

- Create a Base Image for vyos called centos-base
- Clear out UUIDs and MAC addresses from configurations as shown
- Install vmware tools on centos-base
- Create a linked clone called cupcake
- Configure ssh on vyos-lab
- Configure DHCP on vyos-lab
- Reboot cupcake and demonstrate that it is pingable by IP from vyos-lab

DHCP Configuration Commands from the Video

💡 Note, depending on the vyos release you are using, the command syntax may differ. Use the tab character to see the valid commands such as: `set service dhcp-server <TAB><TAB>`

```
configure
set service dhcp-server global-parameters 'local-address 10.0.5.2;'
set service dhcp-server shared-network-name DHCPPOOL authoritative
set service dhcp-server shared-network-name DHCPPOOL subnet 10.0.5.0/24
default-router '10.0.5.2'
set service dhcp-server shared-network-name DHCPPOOL subnet 10.0.5.0/24
domain-name 'range.local'
set service dhcp-server shared-network-name DHCPPOOL subnet 10.0.5.0/24
lease '86400'
set service dhcp-server shared-network-name DHCPPOOL subnet 10.0.5.0/24
range POOL1 start '10.0.5.50'
set service dhcp-server shared-network-name DHCPPOOL subnet 10.0.5.0/24
range POOL1 stop '10.0.5.100'
commit
save
```

Note, a DNS server is not required in the DHCP settings. We actually do not want our range to resolve external hostnames. The following shows a completed dhcp-server on vyos.

```
vyos@vyos# show service dhcp-server
global-parameters "local-address 10.0.5.2;"
shared-network-name DHCPPOOL {
    authoritative
    subnet 10.0.5.0/24 {
        default-router 10.0.5.2
        domain-name range.local
        lease 86400
        range POOL1 {
            start 10.0.5.50
            stop 10.0.5.100
        }
    }
}
```

Deliverable 3. Provide a screenshot of an ssh session with vyos-lab and subsequent ping to cupcake from vyos-lab. Similar to the one below.

```
vyos@vyos# ping 10.0.5.50
PING 10.0.5.50 (10.0.5.50) 56(84) bytes of data.
64 bytes from 10.0.5.50: icmp_seq=1 ttl=64 time=1.04 ms
64 bytes from 10.0.5.50: icmp_seq=2 ttl=64 time=0.343 ms
64 bytes from 10.0.5.50: icmp_seq=3 ttl=64 time=0.656 ms
^C
--- 10.0.5.50 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
rtt min/avg/max/mdev = 0.343/0.679/1.039/0.284 ms
[edit]
vyos@vyos#
```

Milestone 4 - VPN connectivity to the target network

Instructions

Watch the following [video](#)

Private key: qL0OcCaWKTTCntw3P2tjRFtG3ljtYfRjiiDQocdA9m8=

Public key: XF1OlkpqrgOZUGbwC4f1n8l/o53Ss8/95p/ciLwqjV8=

- Install Wireguard on Kali
- Create a keypair on kali

```
sudo apt install wireguard
cd /etc/wireguard
umask 077
```

```
wg genkey | tee privatekey | wg pubkey > publickey
```

- Create the default keypair on vyos

```
generate wireguard default-keypair
configure
set interfaces wireguard wg0 private-key default
```

- Create a peer for 10.0.99.100 on vyos

```
set interfaces wireguard wg0 address '10.0.99.1/24'
set interfaces wireguard wg0 peer namegoeshere allowed-ips '10.0.99.100/32'
set interfaces wireguard wg0 peer namegoeshere public-key keygoeshere
set interfaces wireguard wg0 port '51820'
commit
save
exit
exit
show interfaces wireguard wg0 public-key
```

wY0uOQvdLe57qRiUFBMYXg7RulkCs1vKaBvFZE9K/HE=

Deliverable 4. Demonstrate network connectivity from kali to the victim machine using the DHCP provided address on VMNET5. Provide a screenshot similar to the one below.

```
(kali㉿kali)-[~]
$ ssh -oHostKeyAlgorithms=+ssh-dss root@10.0.5.50
root@10.0.5.50's password:
Last login: Tue Nov 22 16:54:13 2022 from 10.0.99.100
[root@centos ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16436 qdisc noqueue state UNKNOWN
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP qlen 1000
    link/ether 00:0c:29:18:14:89 brd ff:ff:ff:ff:ff:ff
    inet 10.0.5.50/24 brd 10.0.5.255 scope global eth0
        inet6 fe80::20c:29ff:fe18:1489/64 scope link
            valid_lft forever preferred_lft forever
[root@centos ~]# exit
logout
Connection to 10.0.5.50 closed.

(kali㉿kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default q
len 1000
    link/ether 00:0c:29:34:e2:ba brd ff:ff:ff:ff:ff:ff
    inet 192.168.229.129/24 brd 192.168.229.255 scope global dynamic noprefixroute eth0
        valid_lft 1279sec preferred_lft 1279sec
    inet6 fe80::cb6a:dd2f:ee2b:b810/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state UNKNOWN group default
qlen 1000
    link/none
    inet 10.0.99.100/24 scope global wg0
        valid_lft forever preferred_lft forever
```

Deliverable 5. Comprehensive journaling and chronological reflections on this activity. Provide a video overview of all documentation authored and your reflections on this activity. Submit a link to your panopto or google drive based video (not a link to your wiki).

