A7011E Homework 1

Nico Ferrari (nicfer-0@student.ltu.se)

November 22, 2020

1. Notes:

In order to study OS security, we can divide the system in the following 3 layers: physical hardware (at the bottom), base OS with the privileged kernel code and at the top layer the user applications and utilities. Each layer has a surface of vulnerability which needs to be secured (hardening) and can be attacked from the layer below if they are not secured. Building and deploying a system must be a planned process which should consider the following steps:

- system security planning: basically performing risk management
- OS hardening: install the OS in a protected/isolated environment with minimal required functionalities and apply patches required and the proper configuration of it based on the requirements established during the assessment performed before, configure and install security controls and in the end security testing of the basic OS
- application secuirty: install required applications and configure and patch them in order to meet the security requirements (from step one) and possibly in a isolated/ secure environment as the OS. in this step also cryptographic pair of keys and cryptographic protocols used are configured.
- security maintenance: continuosly maintain the system secure (auditing, testing, backing up, recovery, test and apply patches to applications and OS...)

after planning what i need, i will install only the necessaries application, services and protocol in my systems, so i will not arrive to the point where i must remove unnecessary stuff, which could lead to security issues.

virtualization: running simulated environment in a layer abstracted from the actual resources, where the access to these resources is managed by the hypervisor. moreover, The hypervisor manage the VMs, their execution, executes privileged operations instead of host hardware and could be of type 1 or type 2.

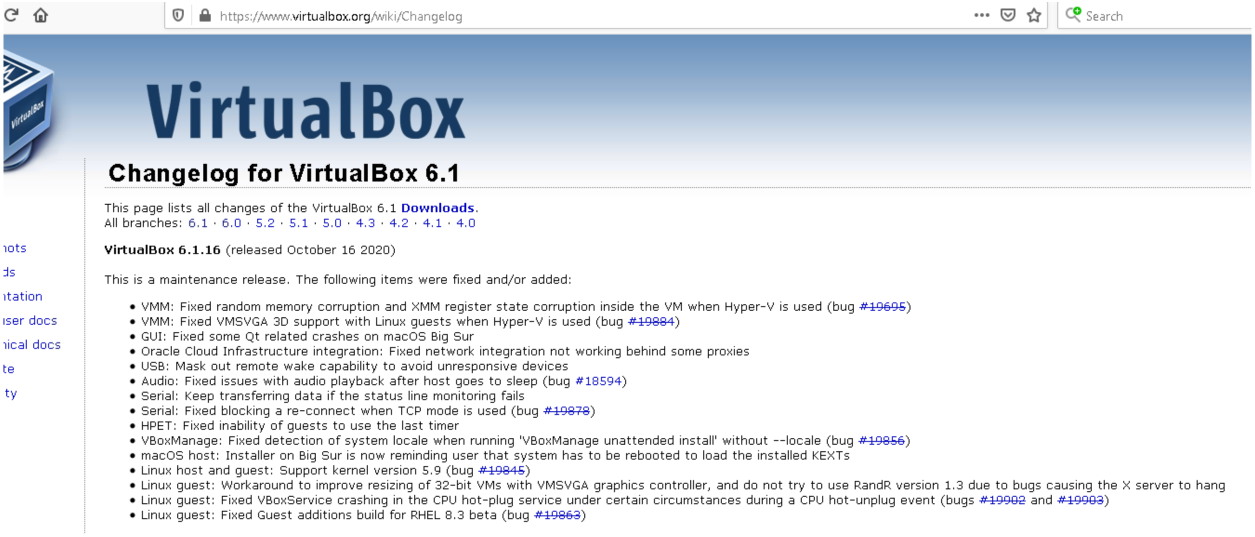
- type 1 (aka: bare metal): is one the top of the host hardware acting as an OS and on the top of it there are the VMs. Considere more efficient and secure due to the less amount of layers on the top of the hardware one. Examples: Xen, hyperV
- type 2: runs on the top of a host OS and uses its functions. This allow to run applications alongside the VMs. Examples: VMware workstation, VirtualBox.

Virtualisation brings different benefits in terms of consolidation by reducing the total number of servers required by an organization, reliability, security by containing some kind of attacks and applying different configurations to each VM. VMM manages the interaction with the hardware platform below and can be divided in type 1 VMM (implemented in ring-0) and type 2 VMM (implemented in ring 3) Virtualization brings downsides as well. it requires flexible policies to enable a secure increase of VMs; when the machine state is restored to a previous one the attacker can exploit older vulnerabilities; due to the mobility of a VM, it is more difficult to guarantee the security of the users of that VM; it is more difficult to uniquely identify a VM; become more difficult to delete confidential data due to the state saving . In addition to the vulnerabilities from the OSs, the suffer from vulnerabilities of the VMM layer.

2. Have you successfully completed Lab assignment (1)

THe first part of Lab 1 was to install Kali linux on a VM using VIrtualBox. As learned in this module, I have checked for new VirtualBox patches. A new version was released (V. 6.1.16) and contained some security bug fixes (Figure 1). I decided to install it and then I proceeded with the installation of the OS.

Figure 1:



Changelog for VirtualBox 6.1

This page lists all changes of the VirtualBox 6.1 **Downloads**.
All branches: 6.1 · 6.0 · 5.2 · 5.1 · 5.0 · 4.3 · 4.2 · 4.1 · 4.0

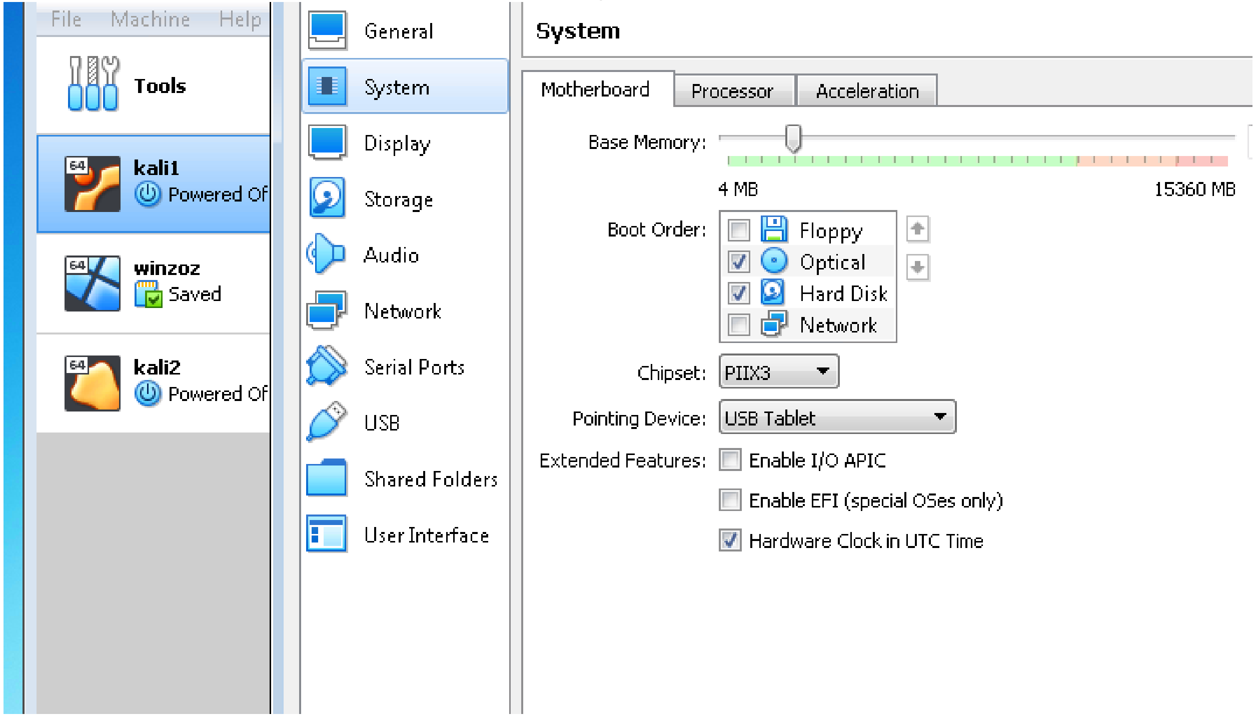
VirtualBox 6.1.16 (released October 16 2020)

This is a maintenance release. The following items were fixed and/or added:

- VMM: Fixed random memory corruption and XMM register state corruption inside the VM when Hyper-V is used (bug ~~#19695~~)
- VMM: Fixed VMSVGA 3D support with Linux guests when Hyper-V is used (bug ~~#19884~~)
- GUI: Fixed some Qt related crashes on macOS Big Sur
- Oracle Cloud Infrastructure integration: Fixed network integration not working behind some proxies
- USB: Mask out remote wake capability to avoid unresponsive devices
- Audio: Fixed issues with audio playback after host goes to sleep (bug #18594)
- Serial: Keep transferring data if the status line monitoring fails
- Serial: Fixed blocking a re-connect when TCP mode is used (bug ~~#19878~~)
- HPET: Fixed inability of guests to use the last timer
- VBoxManage: Fixed detection of system locale when running 'VBoxManage unattended install' without --locale (bug ~~#19856~~)
- macOS host: Installer on Big Sur is now reminding user that system has to be rebooted to load the installed KEXTs
- Linux host and guest: Support kernel version 5.9 (bug ~~#19845~~)
- Linux guest: Workaround to improve resizing of 32-bit VMs with VMSVGA graphics controller, and do not try to use RandR version 1.3 due to bugs causing the X server to hang
- Linux guest: Fixed VBoxService crashing in the CPU hot-plug service under certain circumstances during a CPU hot-unplug event (bugs ~~#19902~~ and ~~#19903~~)
- Linux guest: Fixed Guest additions build for RHEL 8.3 beta (bug ~~#19863~~)

In order to create a secure environment I disabled the NAT connection to in the VM configuration and then, after selecting the ISO to install (kali 2020.3, since the new version was not release yet), i started to follow the installation guidelines. USB support, floppy drive, I/O APIC and audio output have been disabled since not needed (Figure 2).

Figure 2:



System

Base Memory: 4 MB 15360 MB

Boot Order:
□ Floppy
☑ Optical
☑ Hard Disk
□ Network

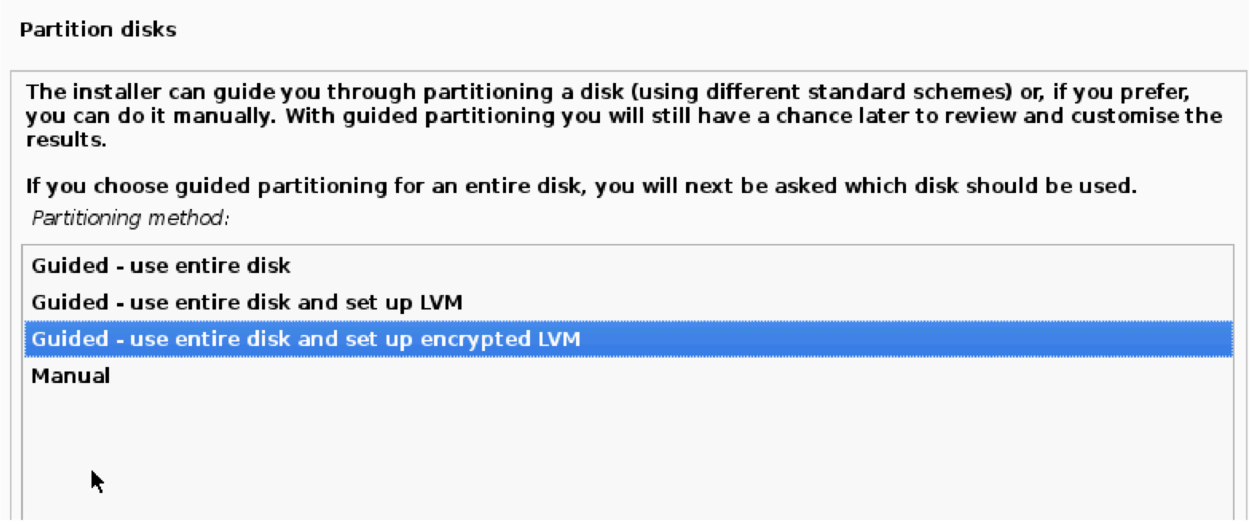
Chipset: PIIX3

Pointing Device: USB Tablet

Extended Features:
□ Enable I/O APIC
□ Enable EFI (special OSes only)
☑ Hardware Clock in UTC Time

in order to increase the security I also used an encrypted LVM. since the VM is just a file, encrypting it would add a security layer (Figure 3).

Figure 3:



Partition disks

The installer can guide you through partitioning a disk (using different standard schemes) or, if you prefer, you can do it manually. With guided partitioning you will still have a chance later to review and customise the results.

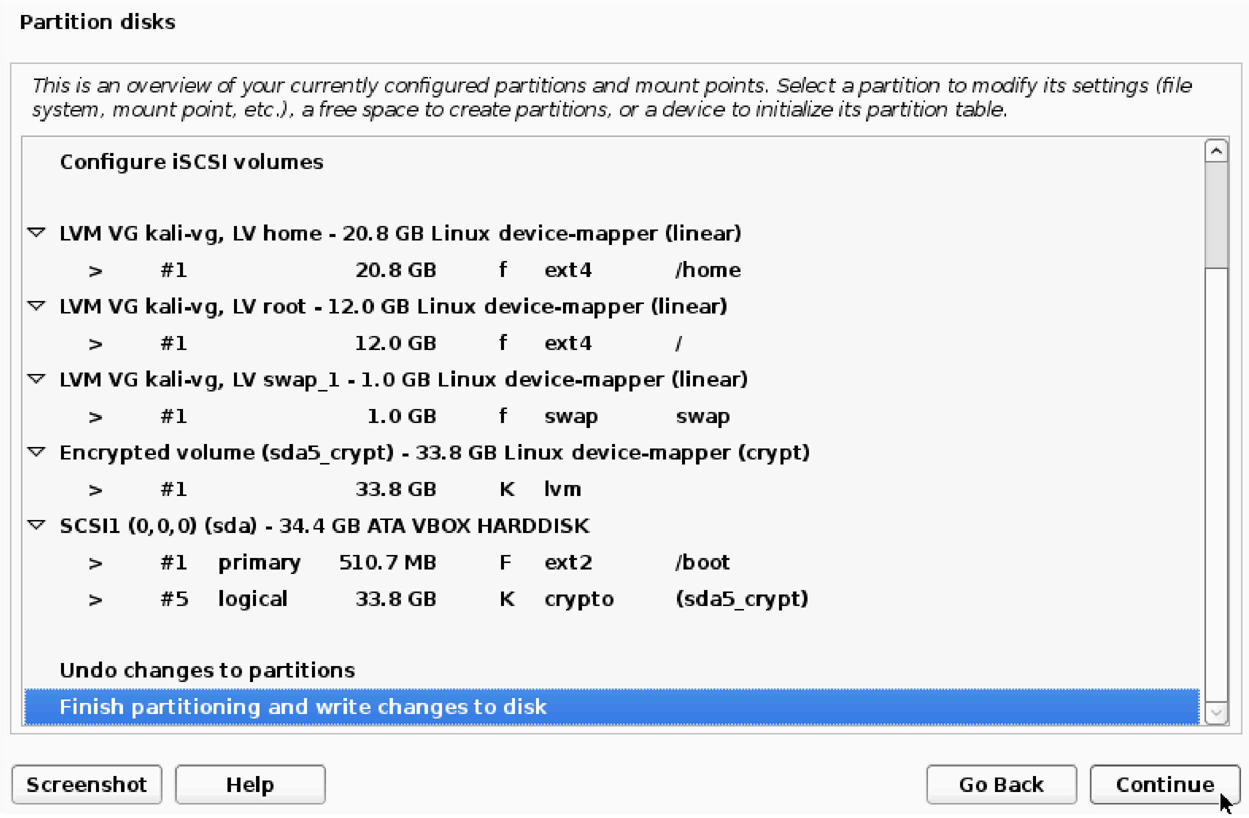
If you choose guided partitioning for an entire disk, you will next be asked which disk should be used.

Partitioning method:

Guided - use entire disk
Guided - use entire disk and set up LVM
Guided - use entire disk and set up encrypted LVM
Manual

I decided to dived the home partition from the rest in order to be able to perform special backups of the root partition in case the home partition gets compromised. Moreover, the rest of the system could be saved in a different harddrive than the home partition's one with different encryption in order to increase security of it. it would make easier to clone the system in the future for new users(Figure 4).

Figure 4:



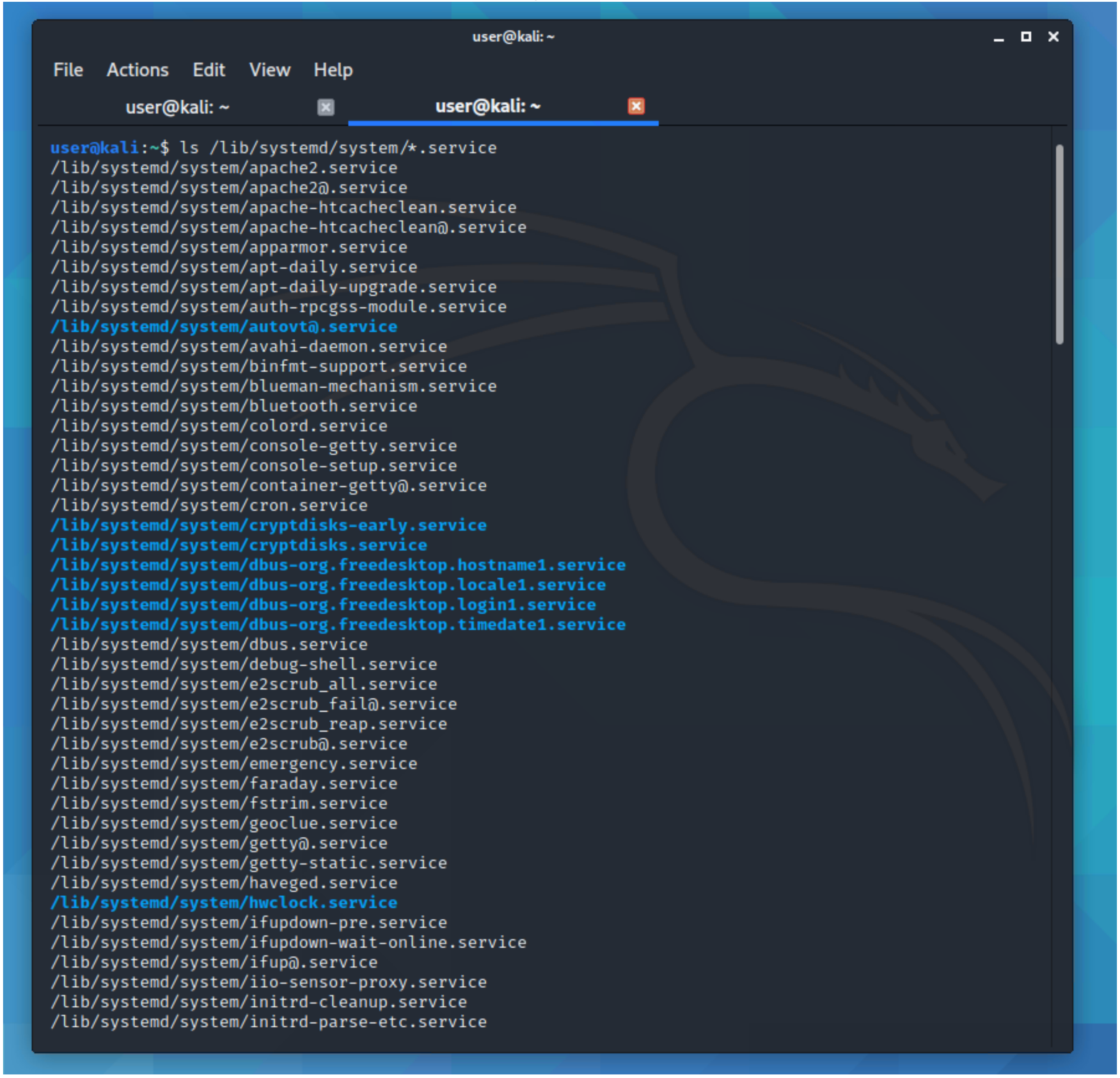
Partition disks

This is an overview of your currently configured partitions and mount points. Select a partition to modify its settings (file system, mount point, etc.), a free space to create partitions, or a device to initialize its partition table.

```
    Configure iSCSI volumes

▽  LVM VG kali-vg, LV home - 20.8 GB Linux device-mapper (linear)
      >     #1                    20.8 GB       f     ext4          /home
▽  LVM VG kali-vg, LV root - 12.0 GB Linux device-mapper (linear)
      >     #1                    12.0 GB       f     ext4          /
▽  LVM VG kali-vg, LV swap_1 - 1.0 GB Linux device-mapper (linear)
      >     #1                     1.0 GB       f     swap          swap
▽  Encrypted volume (sda5_crypt) - 33.8 GB Linux device-mapper (crypt)
      >     #1                    33.8 GB       K     lvm
▽  SCSI1 (0,0,0) (sda) - 34.4 GB ATA VBOX HARDDISK
      >     #1    primary       510.7 MB        F     ext2          /boot
      >     #5    logical         33.8 GB       K     crypto        (sda5_crypt)

    Undo changes to partitions
    Finish partitioning and write changes to disk
```

Instead of using the root users, i created a new user. Using by default the root user is not a good idea, beside security concerns, it is easier to damage the system (i.e. deleting important files which requires root powers). Once the system was installed i gave a look to the services offered by kali linux. I think that different hardening operations could be enabled in order to reduce the system to the functionalities we really need (from services like Apache to the usually useless and resource consuming Desktop Environment) (Figure 5). Since I do not know yet what the next labs will reserve for us, i decided to keep the entire system as it is.

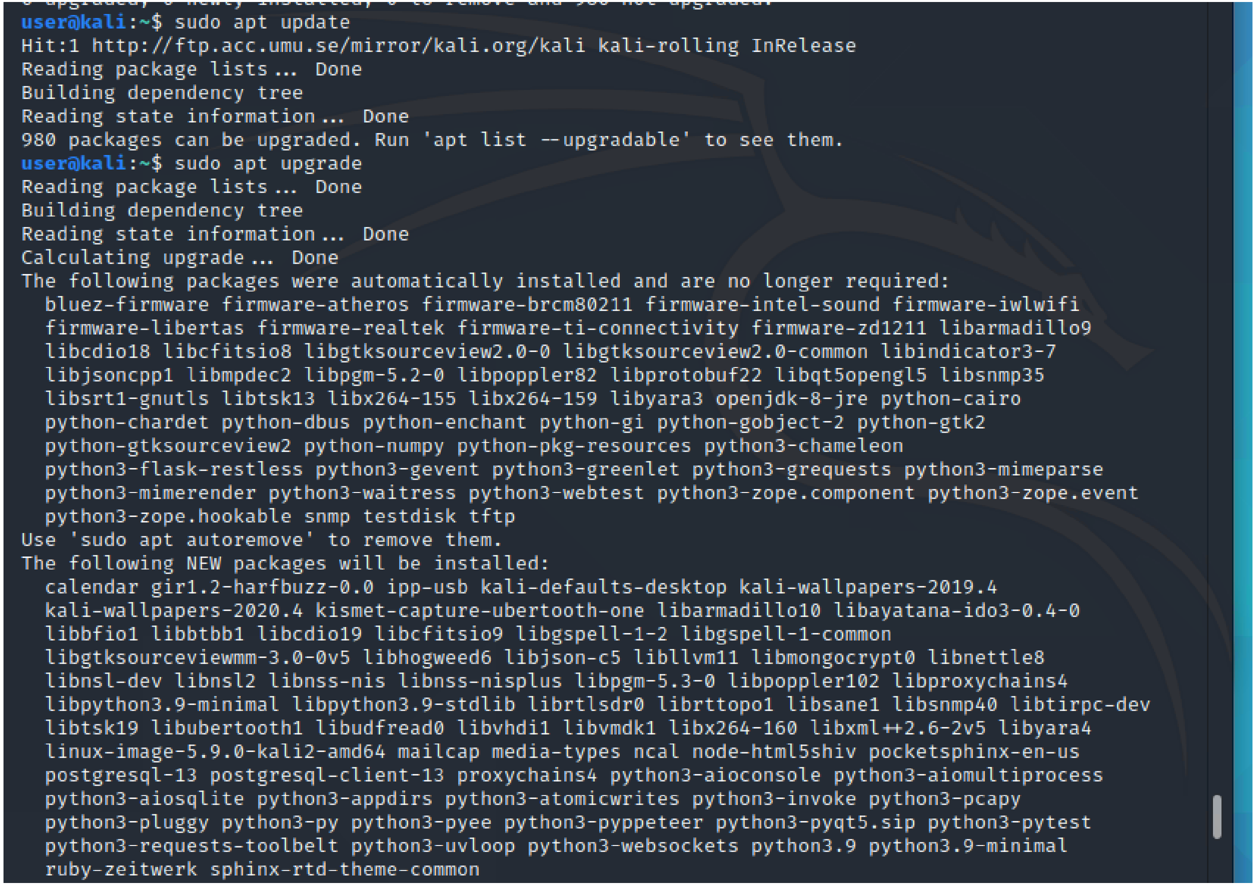
Figure 5:



```
user@kali:~$ ls /lib/systemd/system/*.service
/lib/systemd/system/apache2.service
/lib/systemd/system/apache2@.service
/lib/systemd/system/apache-htcacheclean.service
/lib/systemd/system/apache-htcacheclean@.service
/lib/systemd/system/apparmor.service
/lib/systemd/system/apt-daily.service
/lib/systemd/system/apt-daily-upgrade.service
/lib/systemd/system/auth-rpcgss-module.service
/lib/systemd/system/autovt@.service
/lib/systemd/system/avahi-daemon.service
/lib/systemd/system/binfmt-support.service
/lib/systemd/system/blueman-mechanism.service
/lib/systemd/system/bluetooth.service
/lib/systemd/system/colord.service
/lib/systemd/system/console-getty.service
/lib/systemd/system/console-setup.service
/lib/systemd/system/container-getty@.service
/lib/systemd/system/cron.service
/lib/systemd/system/cryptdisks-early.service
/lib/systemd/system/cryptdisks.service
/lib/systemd/system/dbus-org.freedesktop.hostname1.service
/lib/systemd/system/dbus-org.freedesktop.locale1.service
/lib/systemd/system/dbus-org.freedesktop.login1.service
/lib/systemd/system/dbus-org.freedesktop.timedate1.service
/lib/systemd/system/dbus.service
/lib/systemd/system/debug-shell.service
/lib/systemd/system/e2scrub_all.service
/lib/systemd/system/e2scrub_fail@.service
/lib/systemd/system/e2scrub_reap.service
/lib/systemd/system/e2scrub@.service
/lib/systemd/system/emergency.service
/lib/systemd/system/faraday.service
/lib/systemd/system/fstrim.service
/lib/systemd/system/geoclue.service
/lib/systemd/system/getty@.service
/lib/systemd/system/getty-static.service
/lib/systemd/system/haveged.service
/lib/systemd/system/hwclock.service
/lib/systemd/system/ifupdown-pre.service
/lib/systemd/system/ifupdown-wait-online.service
/lib/systemd/system/ifup@.service
/lib/systemd/system/iio-sensor-proxy.service
/lib/systemd/system/initrd-cleanup.service
/lib/systemd/system/initrd-parse-etc.service
```

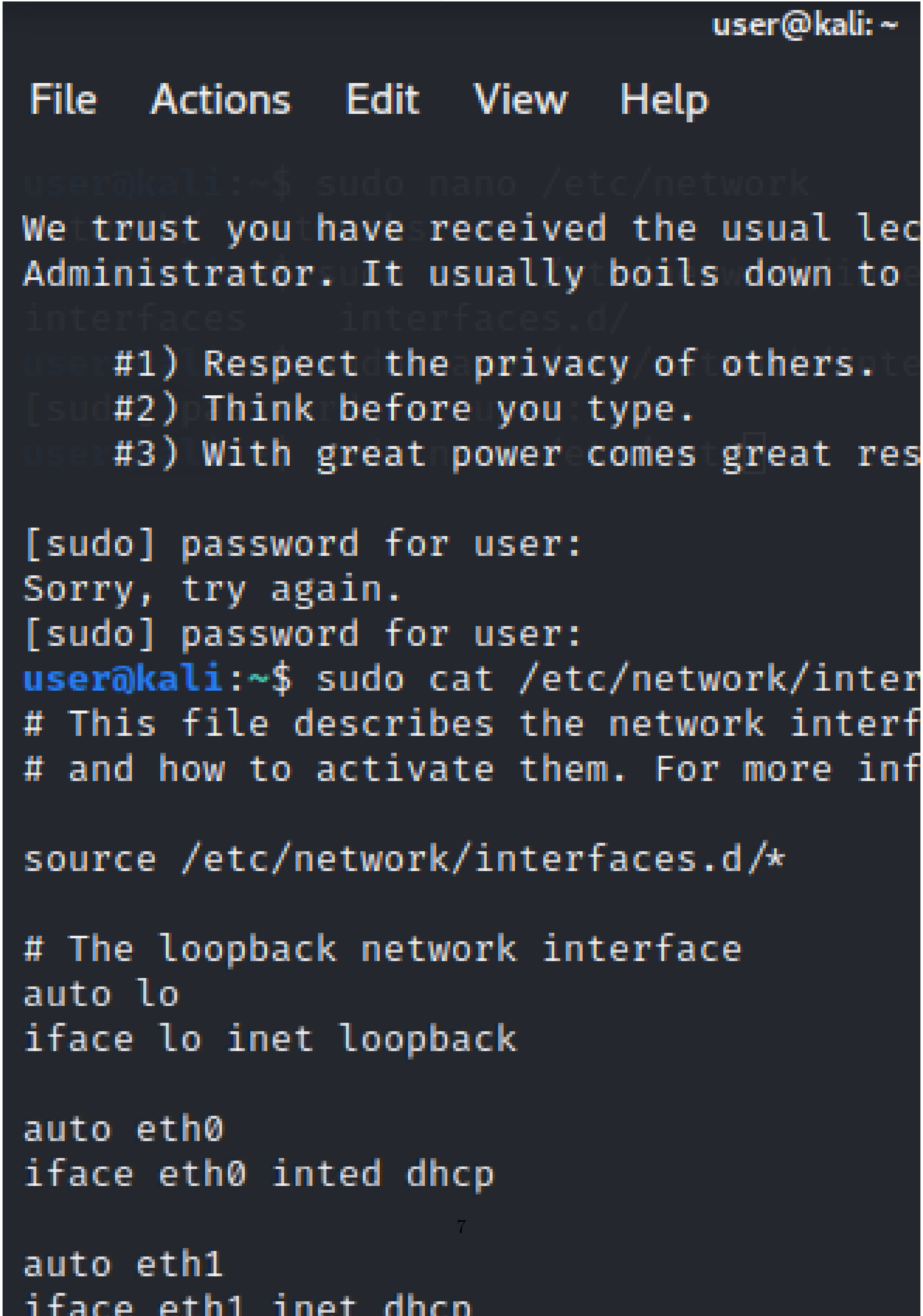
Now it is time to check for updates, so the VM nee to be attached to the NAT and then it is possible to run the commands to check for updates(Figure 6).

Figure 6:



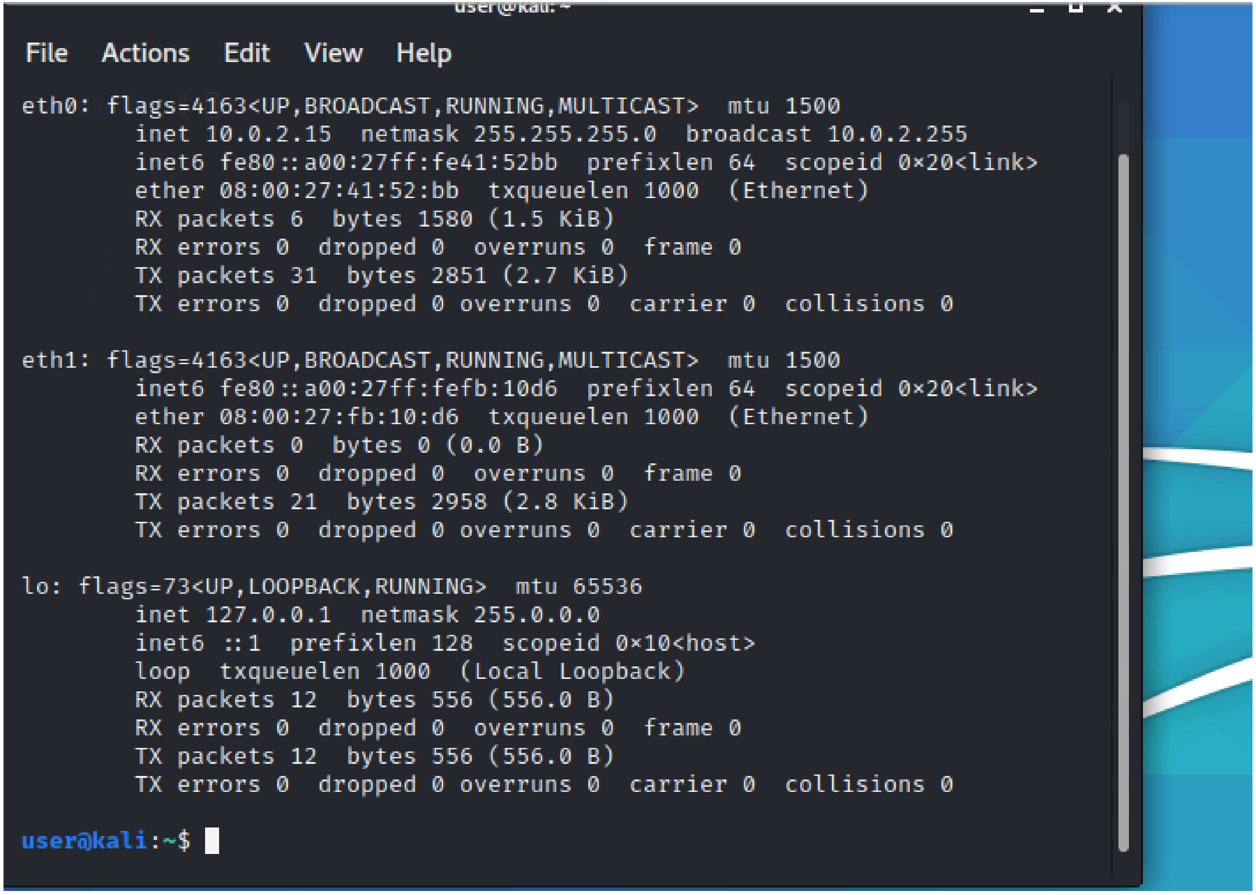
```
0 upgraded, 0 newly installed, 0 to remove and 980 not upgraded.
user@kali:~$ sudo apt update
Hit:1 http://ftp.acc.umu.se/mirror/kali.org/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
980 packages can be upgraded. Run 'apt list --upgradable' to see them.
user@kali:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  bluez-firmware firmware-atheros firmware-brcm80211 firmware-intel-sound firmware-iwlwifi
  firmware-libertas firmware-realtek firmware-ti-connectivity firmware-zd1211 libarmadillo9
  libcdio18 libcfitsio8 libgtksourceview2.0-0 libgtksourceview2.0-common libindicator3-7
  libjsoncpp1 libmpdec2 libpgm-5.2-0 libpoppler82 libprotobuf22 libqt5opengl5 libsnmp35
  libsrt1-gnutls libtsk13 libx264-155 libx264-159 libyara3 openjdk-8-jre python-cairo
  python-chardet python-dbus python-enchant python-gi python-gobject-2 python-gtk2
  python-gtksourceview2 python-numpy python-pkg-resources python3-chameleon
  python3-flask-restless python3-gevent python3-greenlet python3-grequests python3-mimeparse
  python3-mimerender python3-waitress python3-webtest python3-zope.component python3-zope.event
  python3-zope.hookable snmp testdisk tftp
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  calendar gir1.2-harfbuzz-0.0 ipp-usb kali-defaults-desktop kali-wallpapers-2019.4
  kali-wallpapers-2020.4 kismet-capture-ubertooth-one libarmadillo10 libayatana-ido3-0.4-0
  libbfio1 libbtbb1 libcdio19 libcfitsio9 libgspell-1-2 libgspell-1-common
  libgtksourceviewmm-3.0-0v5 libhogweed6 libjson-c5 libllvm11 libmongocrypt0 libnettle8
  libnsl-dev libnsl2 libnss-nis libnss-nisplus libpgm-5.3-0 libpoppler102 libproxychains4
  libpython3.9-minimal libpython3.9-stdlib librtlsdr0 librttopo1 libsane1 libsnmp40 libtirpc-dev
  libtsk19 libubertooth1 libudfread0 libvhdi1 libvmdk1 libx264-160 libxml++2.6-2v5 libyara4
  linux-image-5.9.0-kali2-amd64 mailcap media-types ncal node-html5shiv pocketsphinx-en-us
  postgresql-13 postgresql-client-13 proxychains4 python3-aioconsole python3-aiomultiprocess
  python3-aiosqlite python3-appdirs python3-atomicwrites python3-invoke python3-pcapy
  python3-pluggy python3-py python3-pyee python3-pyppeteer python3-pyqt5.sip python3-pytest
  python3-requests-toolbelt python3-uvloop python3-websockets python3.9 python3.9-minimal
  ruby-zeitwerk sphinx-rtd-theme-common
```

Figure 7:



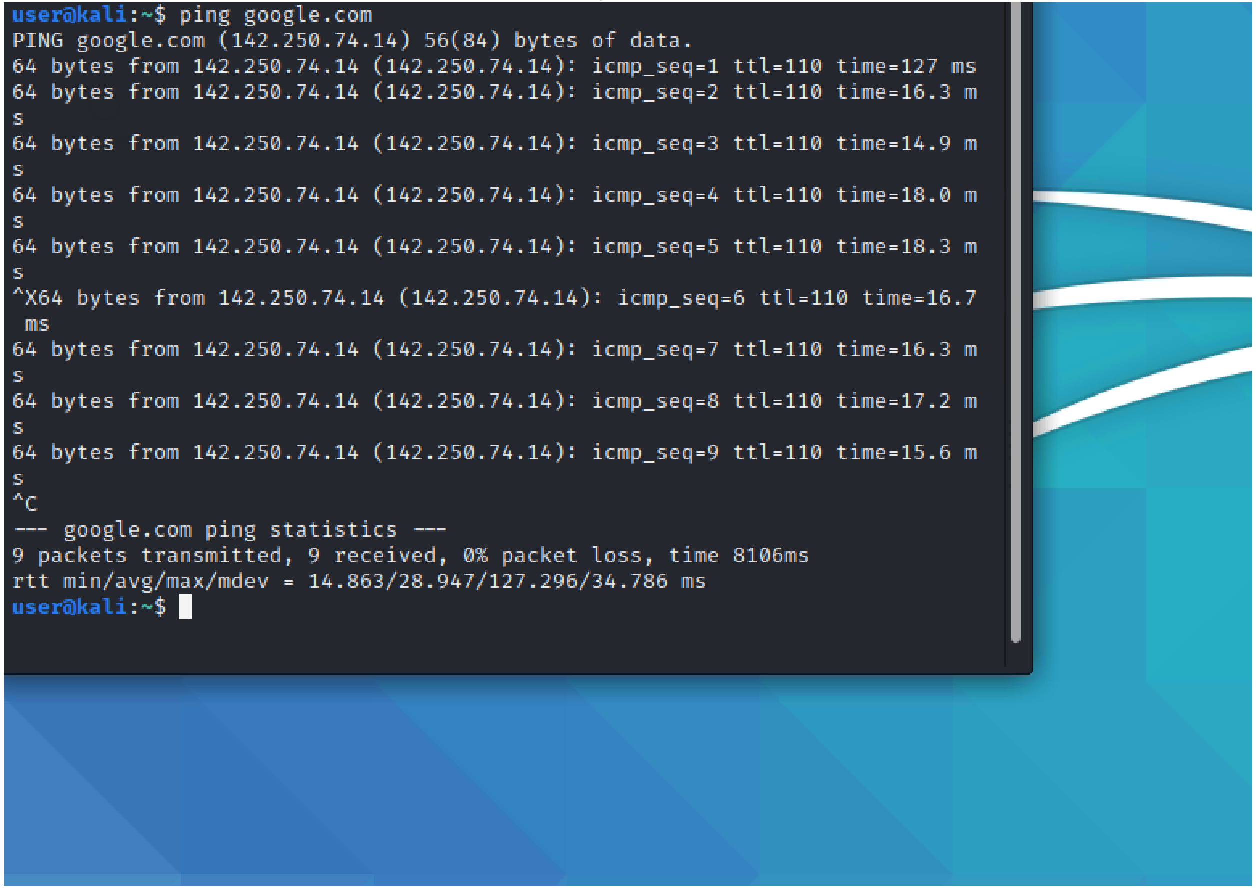
```
We trust you have received the usual lec
Administrator. It usually boils down to 

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great res

[sudo] password for user:
Sorry, try again.
[sudo] password for user:
user@kali:~$ sudo cat /etc/network/inter
# This file describes the network interf
# and how to activate them. For more inf

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inted dhcp

auto eth1
iface eth1 inet dhcp
```

Figure 8:



```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::a00:27ff:fe41:52bb  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:41:52:bb  txqueuelen 1000  (Ethernet)
        RX packets 6  bytes 1580 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 31  bytes 2851 (2.7 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet6 fe80::a00:27ff:fefb:10d6  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:fb:10:d6  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 21  bytes 2958 (2.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 12  bytes 556 (556.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 12  bytes 556 (556.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

user@kali:~$ 
```

Figure 9:



```
user@kali:~$ ping google.com
PING google.com (142.250.74.14) 56(84) bytes of data.
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=1 ttl=110 time=127 ms
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=2 ttl=110 time=16.3 m
s
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=3 ttl=110 time=14.9 m
s
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=4 ttl=110 time=18.0 m
s
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=5 ttl=110 time=18.3 m
s
^X64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=6 ttl=110 time=16.7
 ms
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=7 ttl=110 time=16.3 m
s
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=8 ttl=110 time=17.2 m
s
64 bytes from 142.250.74.14 (142.250.74.14): icmp_seq=9 ttl=110 time=15.6 m
s
^C
--- google.com ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8106ms
rtt min/avg/max/mdev = 14.863/28.947/127.296/34.786 ms
user@kali:~$ 
```

3. attack surfaces

nowdays, especially with this new situation due to the pandemy, Cloud computing is taking part of many daily activities and bringing many valuable benefits. The enabling component for cloud computing is virtualization, allowing different environments from abstract resources on a single host and migration of these environments between different hosts [8].

Virtualization introduces a software layer which lies between the Virtual Machines and the physical hardware. A Virtual Machine Manager or hypervisor manages the VMs, the resource allocation and controls the VMs' access to the underlying layer. Depending of its type , the hypervisor can be located directly on physical layer or on the top of the OS layer [1]. With this technology, security is an important concern, where each link between the different layers must establish a degree of trust. Virtualization doesn't bring more security compared to a system deployed directly over the physical layer. In fact, its security depends of the host OS(if present), the OS of the virtualized environment and the VMM itself, as studied in [5].

The hypervisor or VMM are the main target of the virtualized layer and, if compromised, it represent a security issue for all the VMs. For this reason, VMM represent a new attack surface which is unique to this technology. Due to its complexity (several thousands of lines of code), it is not possible to perform formal verification of the hypervisor's code to check for bugs [**Chen2011ProceedingsSecurity**] . VMs can exploit these possible bugs in the hypervisor in order to breach confidentiality, integrity and availability of the other VMs. Moreover, since the Hypervisor plays an inportant role through the whole lifetime of the VM, if we are in type 2 of hypervisor, the guest OS can exploit each communication with the VMs to perform an attack.

As shown in [6], also in Type 1 hypervisors many vulnerabilities have been identified. In type 1 hypervisor, the virtualization layers is directly over the physical layer, removing the attack surface created by the host OS. Some researches such as [2] showed some vulnerabilities affecting the most common type 1 hypervisors. from this research, the results show the VMM introduce several non-essential services and these may increase the attack surface significantly. Moreover it shows that most of the vulnerabilities are triggered from the guest VM, especially from the VM User-Space, which means that each user (also the ones with less privileges) can become a threat to the underlying hypervisor.

A really dangerous attack has been analyzed in [7], the escape attack, which exploits vulnerabilities of the VM operating system and allows the guest system to interact with the hypervisor and read files in the host system, including the system folder and other security-sensitive folders and is possible in both type-1 or type-2 hypervisors. This type of attacks represents a big threat in data isolation. Another type of attack affecting the attack surface composed by the VMM is Hypervisor Hyperjacking, where the intruder gets the control over the VMM, which creates the virtual environment within a virtual machine (VM) host. Collocating VMs on the same hypervisor platform, vulnerabilities can be exposed thanks to side channel attacks through cache, network traffic monitoring or via utilities to share files/data [3]. Thanks to these techniques it is possible to map the infrastructure and create a VM in the same Hypervisor platform of the targtet, and from there execute attacks to it. Also Network security becomes a problem in virtualized environment, since the VMs might share a single ethernet resource, each VM will have or share its own IP address through thecnologies such as Network Address Translation (NAT) but providing then certain degree of protection occlusion from IDS and IPS devices deployed on the LAN.

In [4] are shown several security threats, varying from network level threats to host, hypervisor and application level.

4. References

[1] "An Introduction to Virtualization". In: *Virtualization for Security*. Elsevier, 2009, pp. 1–43. DOI: 10.1016/b978-1-59749-305-5.00001-3.

[2] Audit Association for Computing Machinery. Special Interest Group on Security et al. *Cloud Computing '13 : proceedings of the 2013 International Workshop on Security in Cloud Computing : May 8, 2013, Hangzhou, China*, p. 70. ISBN: 9781450320672.

[3] Tyson T. Brooks, Carlos Caicedo, and Joon S. Park. "Security Vulnerability Analysis in Virtualized Computing Environments". In: *International Journal of Intelligent Computing Research* 3.4 (Dec. 2012), pp. 263–277. DOI: 10.20533/ijicr.2042.4655.2012.0034.

[4] Sanchika Gupta and Padam Kumar. "Taxonomy of Cloud Security". In: *International Journal of Computer Science, Engineering and Applications* 3.5 (Oct. 2013), pp. 47–67. ISSN: 22310088. DOI: 10.5121/ijcsea.2013.3505.

[5] Reihanah. Safavi-Naini, Audit Association for Computing Machinery. Special Interest Group on Security, and Macquarie University. *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security (ASIACCS '09) : Sydney, Australia, March 10-12, 2009*. Association for Computing Machinery, 2009. ISBN: 9781605583945.

[6] Sosp 11 Conference Committee and ACM Digital Library. *Sosp 11 Proceedings of the Twenty Third Acm Symposium on Operating Systems Principles*. ISBN: 9781450309776.

[7] Jiang Wu et al. "An access control model for preventing virtual machine escape attack". In: *Future Internet* 9.2 (June 2017). ISSN: 19995903. DOI: 10.3390/fi9020020.

[8] Guodong Zhu et al. "Detecting Virtualization Specific Vulnerabilities in Cloud Computing Environment". In: *IEEE International Conference on Cloud Computing, CLOUD*. Vol. 2017-June. IEEE Computer Society, Sept. 2017, pp. 743–748. ISBN: 9781538619933. DOI: 10.1109/CLOUD.2017.105.

5. thoughs about this week

I found the topic regarding virtualization really interesting. i have never studied these kind of vulnerabilities and threats and this opened my eyes in front of all these problems! beside that, I had problems with the internet connection and work so I was not able to 'enjoy' the lessons but had to to everything fast to at least finish the assignment after one week of delay. nevertheless, the recorded lesson had really important discussions, especially regarding the importance of the planning phase in the system deployment .