**[MANDATORY]**

1. **Installing Sudo**

**DEF** Sudo, acronimo di Super User DO, letteramente “esegui come super utente”, e’ un programma per i sistemi operativi Unix e Unix-like cge, con dei vincoli, permette di eseguire altri programmi assumendo l’identita’ (e di conseguenza anche i privilegi) di altri utenti. Su Ubuntu e’ preinstallato (si dice nativo), mentre in Debian (quello che ci serve per la la nostra VM bisogna installarlo).

1. First type su - to login in as the root user.

2. Then type apt-get update -y

3. Then type apt-get upgrade -y

4. Then type apt install sudo

5. Then type sudo usermod -aG sudo [your\_username] to add user in the sudo group

(To check if user is in sudo group, type getent group sudo).

6. Type sudo visudo to open sudoers file

7. Lastly find - # User privilege specification, type [your\_username] ALL=(ALL:ALL) ALL in addiction to root

8. Ctrl+X to EXIT and Y to Save

1. **Installing and Configuring SSH (Secure Shell Host)**

**DEF  
**SSH (Secure Shell)**** è un protocollo di rete che fornisce un accesso sicuro ai computer remoti perché crittografa tutte le comunicazioni tra client e server. In altre parole, SSH consente di costruire un cosiddetto tunnel SSH, a un’estremità del quale i dati vengono crittografati e inviati e all’altra estremità vengono ricevuti e decrittati. I dati si spostano attraverso il tunnel crittografati.

1. Type sudo apt install openssh-server and Type Y to confirm installation

2. Type sudo systemctl status ssh to check SSH Server Status

3. Type sudo nano /etc/ssh/sshd\_config

4. Find this line #Port22

5. Change the line to Port 4242 without the # (Hash) in front of it

6. Save and Exit

7. Then type sudo grep Port /etc/ssh/sshd\_config to check if the port settings are right

if is corretc it return   
 “Port 4242

#GatewayPorts no

root@your\_name42:/home/yourname#”

8. Lastly type sudo service ssh restart to restart the SSH Service

1. **Installing and Configuring UFW (Uncomplicated Firewall)**

**DEF Uncomplicated Firewall** (ufw) è un [firewall](https://it.wikipedia.org/wiki/Firewall" \o "Firewall) ideato per gestire le [iptables](https://it.wikipedia.org/wiki/Iptables" \o ") in maniera semplificata, attraverso un'[interfaccia a riga di comando](https://it.wikipedia.org/wiki/Interfaccia_a_riga_di_comando" \o "Interfaccia a riga di comando). È un [software libero](https://it.wikipedia.org/wiki/Software_libero" \o "Software libero) scritto in [python](https://it.wikipedia.org/wiki/Python" \o "Python) per i [sistemi operativi](https://it.wikipedia.org/wiki/Sistema_operativo" \o "Sistema operativo) [GNU/Linux](https://it.wikipedia.org/wiki/GNU/Linux" \o "GNU/Linux).

Uncomplicated Firewall ha un'[interfaccia grafica](https://it.wikipedia.org/wiki/Interfaccia_grafica" \o "Interfaccia grafica) chiamata *GUI for Uncomplicated Firewall* (gufw) con regole predefinite per bloccare o autorizzare i più comuni servizi di rete.

iptables --> https://it.wikipedia.org/wiki/Netfilter

1. First type apt install ufw to install UFW and type Y to confirm installation

2. Type sudo ufw enable to inable UFW

3. Type sudo ufw status numbered to check the status of UFW

if true returns: ---> Status: active

1. Type sudo ufw allow 4242 to configure the Port Rules

returns: Rule added

Rule added (v6)

1. Lastly Type sudo ufw status numbered to check the status of UFW 4242 Port

IMPORTANT

to remove port rule:

sudo ufw delete allow <port>

sudo ufw delete deny <port>

1. **Connecting to SSH**

1. Go to your Virtual Box Program

2. Click on your Virtual Machine and select Settings

3. Click Network then Adapter 1 (attached to NAT) then Advanced and then click on Port Forwarding

4. Click on PLUS GREEN BUTTON on the right Adds news port forwarding rule

4. Change the Host Port to 4243 and Guest Port to 4242

5. Then head back to your Virtual Machine

6. Type sudo systemctl restart ssh to restart your SSH Server

7. Type sudo service sshd status to check your SSH Status

8. Open a terminal(on client) and type the following ssh your\_username@127.0.0.1 -p 4243

9. In case an error occurs, then type rm ~/.ssh/known\_hosts in your Terminal and then retype ssh your\_username@127.0.0.1 -p 4243

type Yes to confirm connection to SSH

10. Lastly type exit to quit your SSH Terminal Connection

PS: on 42 school computer, you have to use the Host Port 4243 and Guest Port to 4242, then ssh your\_username@127.0.0.1 -p 4242 in terminal to connect.

1. **Password Policy**

You can do it by your terminal (4. Connectin to SSH ---> 8. Open terminal etc...)  
  
Edit /etc/login.defs

cd /etc to go in directory  
 sudo nano login.defs to open file  
 then find "password aging controls". Modify them as per subject instructions:

PASS\_MAX\_DAYS 30

PASS\_MIN\_DAYS 2

PASS\_WARN\_AGE 7

these changes aren't automatically applied to existing users, so use chage command to

modify for any users and for root:

$ sudo chage -M 30 <username/root>

$ sudo chage -m 2 <username/root>

$ sudo chage -W 7 <username/root>

Use chage -l <username/root> to check user settings.

Install password quality verification library:

$ sudo apt install libpam-pwquality

Then, edit the /etc/security/pwquality.conf file like so:

cd /etc/security to go in directory

sudo nano pwquality.conf to open file

# Number of characters in the new password that must not be present in the

# old password.

difok = 7

# The minimum acceptable size for the new password (plus one if

# credits are not disabled which is the default)

minlen = 10

# The maximum credit for having digits in the new password. If less than 0

# it is the minimun number of digits in the new password.

dcredit = -1

# The maximum credit for having uppercase characters in the new password.

# If less than 0 it is the minimun number of uppercase characters in the new

# password.

ucredit = -1

# ...

# The maximum number of allowed consecutive same characters in the new password.

# The check is disabled if the value is 0.

maxrepeat = 3

# ...

# Whether to check it it contains the user name in some form.

# The check is disabled if the value is 0.

usercheck = 1

# ...

# Prompt user at most N times before returning with error. The default is 1.

retry = 3

# Enforces pwquality checks on the root user password.

# Enabled if the option is present.

enforce\_for\_root

# ...

Change user passwords to comply with password policy:

$ sudo passwd <user/root>

1. **Users and Groups**
   1. **Create Groups**

1. First type sudo groupadd user42 to create a group

2. Then type sudo groupadd evaluating to create an evaluating group

3. Lastly type getent group to check if the group has been created

* 1. **Create a new user and add user in groups**

1. First type cut -d: -f1 /etc/passwd to check all local users

2. Type sudo adduser [new\_username] to create a username then press ENTER for the default choses

3. Type sudo usermod -aG user42 [your\_username] to add your\_username in user42 group

4. Type sudo usermod -aG evaluating [your\_new\_username]

5. Type getent group user42 to check if the user is the group

6. Type getent group evaluating to check the group

7. Type groups to see which groups the user account belongs to

8. Lastly type sudo chage -l [your\_new\_username] to check if the password rules are working in users

**6.3 Creating sudo.log**

1. First type cd ~/../ to ho in your /home

2. Then type cd /var/log

3. Then type sudo mkdir sudo (if it already exists, then continue to the next step).

4. Then type cd /var/log/sudo

5. Then type sudo touch sudo.log to create file.log

6. Then type cd ~/../ to ESC and return in your /home

**6.4 Configuring Sudoers Group**

1. First type sudo nano /etc/sudoers to open sudoers file

2. Now edit your sudoers file to look like the following by adding in all of the defaults in the

image below -

Defaults env\_reset

Defaults mail\_badpass

Defaults secure\_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"

Defaults badpass\_message="Password is wrong, please try again!"

Defaults passwd\_tries=3

Defaults logfile="/var/log/sudo/sudo.log"

Defaults log\_input, log\_output

Defaults requiretty

1. **Monitoring Script**
2. Then type cd /usr/local/bin
3. Then type sudo touch monitoring.sh
4. Lastly type sudo chmod 777 monitoring.shc  
   4. open monitoring.txt (the other file that i send you)

5 Copy text from it

1. open monitoring.sh typing sudo nano monitoring.sh
2. Paste the text
3. Ctrl + X to save
4. Y to confirm saving
5. **Crontab Configuation**
6. Type sudo crontab -e
7. Then press 1 to choose nano for modify the crontab
8. Lastly at the end of the crontab, type the following line:  
    \*/10 \* \* \* \* /usr/local/bin/monitoring.sh | wall  
   this means that every 10 mins, this script will show

to test your script runs type:

sudo /usr/local/bin/monitoring.sh

1. **Signature.txt**

Warning: before you generate a signature number, turn off your Virtual Machine.

1. Go into .vdi file location

2. Type sha1sum VirtualBox.vdi or whatever your Virtual Machine is called (This can take from a few seconds to 5 mins).

3. Copy the output number and create a signature.txt file and paste that number in the file.

4. Now you submit the signature.txt file with the output number in it.

**[BONUS]**

**Part 1**

For the Born2beroot bonuses, we have to install WordPress with Lighttpd, MariaDB and PHP. We also have to install another service of our own choice, and justify that choice.

1. **Installing WordPress**

**1.1. Installing PHP**

1.1.1. Add a different ATP repositiry. A Sury,s repository.  
  
  
$ sudo apt update

$ sudo apt install curl

$ sudo curl -sSL https://packages.sury.org/php/README.txt | sudo bash -x

$ sudo apt update

1.1.2. Install PHP 8.1+ version.

$ sudo apt install php8.1

$ sudo apt install php-common php-cgi php-cli php-mysql

1.1.3. Check php version

$ php -v

**1.2. Installing Lighttpd**

1.2.1. If Apache may be installed Uninstall it because it conflicts whit lighttpd.

$ systemctl status apache2

$ sudo apt purge apache2

1.2.2. Install Lighttpd

$ sudo apt install lighttpd

1.2.3. Check version, start, enable lighttpd and check status

$ sudo lighttpd -v

$ sudo systemctl start lighttpd

$ sudo systemctl enable lighttpd

$ sudo systemctl status lighttpd

1.2.4. Allow http port (port 80) through UFW

$ sudo ufw allow http

$ sudo ufw status

1.2.5. Forward host port 8080 to guest port 80 in VirtualBox

1. Go to VM >> Setting >> Network >> Advanced >> Adapter 1 >> Port Forwarding
2. Add rule for host port 8080 to forward to guest port 80 (Vedi 4.4. Connecting SSH)

To test Lighttpd, go to host machine browser and type in address http://127.0.0.1:8080 or http://localhost:8080. You should see a Lighttpd "placeholder page".

1.2.6. Activate lighttpd FastCGI module

1. Back in your VM
2. $ sudo lighty-enable-mod fastcgi
3. $ sudo lighty-enable-mod fastcgi-php
4. $ sudo service lighttpd force-reload

To test php is working with lighttpd, create a file in /var/www/html named info.php. In that php file, write:

<?php

phpinfo();

?>

Save and go to your browser and type <http://127.0.0.1:8080/info.php>

You should get a page with PHP information.

**1.3. Installing MariaDB**

1.3.1. Installation

$ sudo apt install mariadb-server

1.3.2. Start, Enable and check MariaDB status

$ sudo systemctl start mariadb

$ sudo systemctl enable mariadb

$ systemctl status mariadb

1.3.3. MySQL secure installation and answer the questions like so:

$ sudo mysql\_secure\_installation

Enter current password for root (enter for none): <Enter>

Switch to unix\_socket authentication [Y/n]: Y

Set root password? [Y/n]: Y

New password: 101Asterix!

Re-enter new password: 101Asterix!

Remove anonymous users? [Y/n]: Y

Disallow root login remotely? [Y/n]: Y

Remove test database and access to it? [Y/n]: Y

Reload privilege tables now? [Y/n]: Y

1.3.4. Restart MariaDB service:

$ sudo systemctl restart mariadb

1.3.5. Enter MariaDB interface:

$ mysql -u root -p

1.3.6. Enter MriaDB root password, then create a database for WordPress

MariaDB [(none)]> CREATE DATABASE wordpress\_db;

MariaDB [(none)]> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'WPpassw0rd';

MariaDB [(none)]> GRANT ALL ON wordpress\_db.\* TO 'admin'@'localhost' IDENTIFIED BY 'WPpassw0rd' WITH GRANT OPTION;

MariaDB [(none)]> FLUSH PRIVILEGES;

MariaDB [(none)]> EXIT;

1.3.7. to Check that database was created successfully, go back into MariaDB interface:

$ mysql -u root -p

and show databases:

MariaDB [(none)]> show databases;

You should see something like this:  
  
+-------------------------------+

| Database |

+-------------------------------+

| information\_schema |

| mysql |

| performance\_schema |

| wordpress\_db |

+------------------------------+

**1.4. Installing WordPress**

1.4.1. Install two tools

$ sudo apt install wget

$ sudo apt install tar

$ sudo apt autoremove to remove apacha2 packages

(ASSICURATI DI ESSERE IN /var/www/html)

1.4.2. Download the latest version of WordPress

$ wget <http://wordpress.org/latest.tar.gz>

1.4.3. Extract it

$ tar -xzvf latest.tar.gz

1.4.4. Move it to /var/www/html/

$ sudo mv wordpress/\* /var/www/html/

1.4.5. Remove file.tar and empy wordpress/ directory

$ rm -rf latest.tar.gz wordpress/

1.4.6. Create WordPress configuration file:

$ sudo mv /var/www/html/wp-config-sample.php /var/www/html/wp-config.php

1.4.7. Edit wp-config.php

$ sudo nano /var/www/html/wp-config.php

then insert with database info:

<?php

/\* ... \*/

/\*\* The name of the database for WordPress \*/

define( 'DB\_NAME', 'wordpress\_db' );

/\*\* Database username \*/

define( 'DB\_USER', 'admin' );

/\*\* Database password \*/

define( 'DB\_PASSWORD', 'WPpassw0rd' );

/\*\* Database host \*/

define( 'DB\_HOST', 'localhost' );

Exit with Ctrl + X and save with Y

1.4.8. Change permissions of WordPress directory to grant rights to web server and restart lighttpd

$ sudo chown -R www-data:www-data /var/www/html/

$ sudo chmod -R 755 /var/www/html/

$ sudo systemctl restart lighttpd

1.4.9. Come in browser, connect to <http://127.0.0.1:8080> and finish WordPress installation

1.4.9.1 Setting   
 Chose language, Site Title, Username ecc..

Save

You installed Wordpress!

**Part 2**

for the second part of bonus, i chose to install Fail2ban as a security measure for SSH againt brute force attacks.

1. Install Fail2ban

$ sudo apt install fail2ban

$ sudo systemctl start fail2ban

$ sudo systemctl enable fail2ban

$ sudo systemctl status fail2ban

1. Create a file jail.local

$ touch jail.local

1. Copy in jail.conf

$ sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

1. Edit jail.local

$ sudo nano /etc/fail2ban/jail.local

* 1. Find line “SSH servers” heading and modify like this:

[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:

# normal (default), ddos, extra or aggressive (combines all).

# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.

# mode = normal

enabled = true

maxretry = 3

findtime = 10m

bantime = 1d

port = 4242

logpath = %(sshd\_log)s

backend = %(sshd\_backend)s

1. Restart fail2ban

$ sudo systemclt restart fail2ban

non riesco ad attivarlo