

Version: 1.1



Born2beRoot

Your very first own server.

Summary

• • •

This document is a System Administration related exercise.

#Network

#Administration

#Unix

42

Intellectual Property Disclaimer

All content presented in this training module, including but not limited to texts, images, graphics, and other materials, is protected by intellectual property rights held by Association 42.

Terms of Use:

- **Personal use:** You are permitted to use the contents of this module solely for personal purpose. Any commercial use, reproduction, distribution, modification, or public display is strictly prohibited without prior written permission from Association 42.
- **Respect for Integrity:** You must not alter, transform, or adapt the content in any way that could harm its integrity.

Protection of Rights:

Any violation of these terms constitutes an infringement of intellectual property rights and may result in legal action. We reserve the right to take all necessary measures to protect our rights, including but not limited to claims for damages.

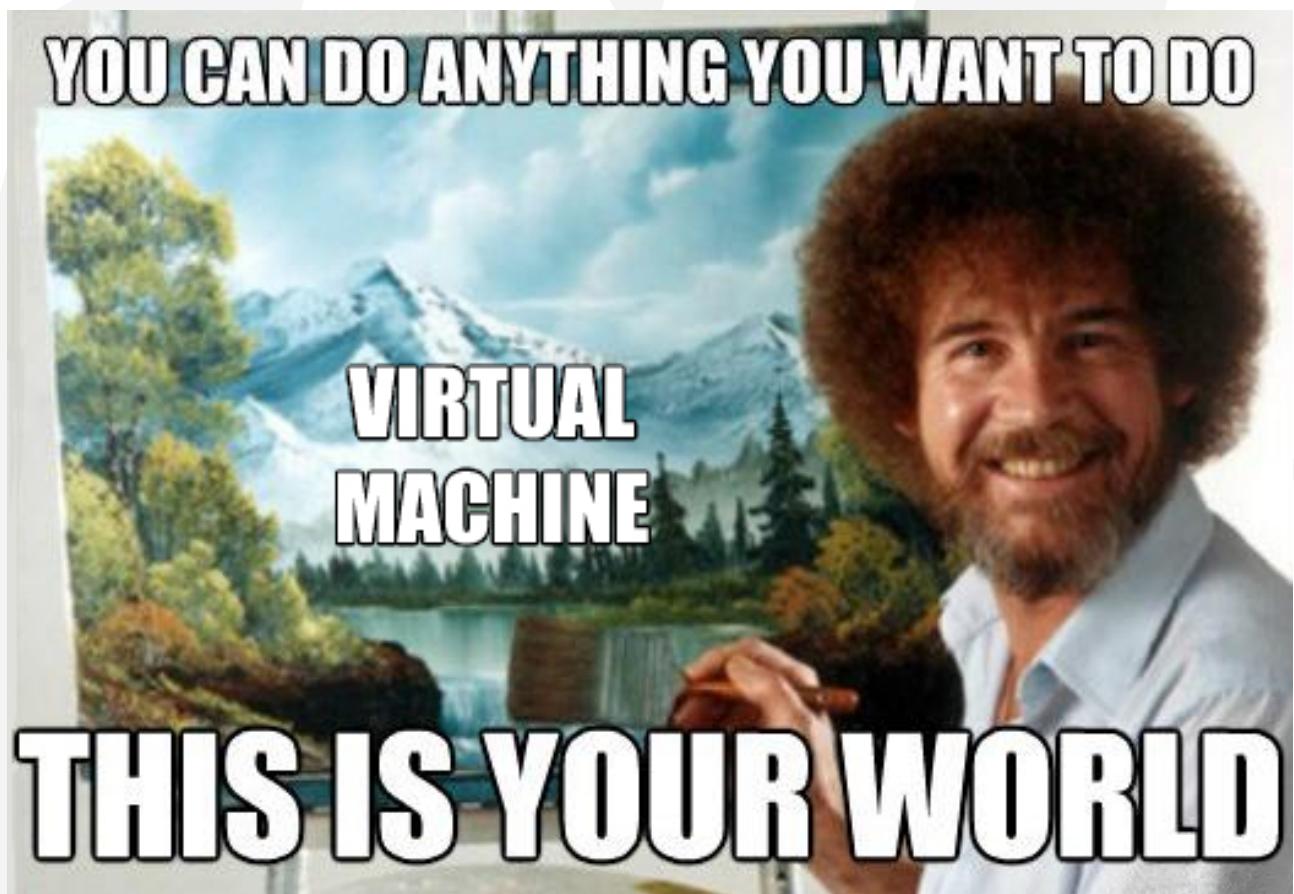
*For any questions regarding the use of the content or to obtain authorization, please contact:
legal@42.fr*

Contents

1 Preamble	1
2 Introduction	2
3 AI Instructions	3
4 General guidelines	5
5 Mandatory part	6
6 Readme Requirements	12
7 Submission and peer-review	13

Chapter 1

Preamble



Chapter 2

Introduction

This activity aims to introduce you to the wonderful world of virtualization.

You will create your first machine in VirtualBox (or UTM if you can't use VirtualBox) under specific instructions. Then, at the end of this activity, you will be able to set up your own operating system while implementing strict rules.

Chapter 3

AI Instructions

● Context

This activity is designed to help you discover the fundamental building blocks of your 42 training.

To properly anchor key knowledge and skills, it's essential to adopt a thoughtful approach to using AI tools and support.

True foundational learning requires genuine intellectual effort — through challenge, repetition, and peer-learning exchanges.

For a more complete overview of our stance on AI — as a learning tool, as part of the 42 training, and as an expectation in the job market — please refer to the dedicated FAQ on the intranet.

● Main message

- 👉 Build strong foundations without shortcuts.
- 👉 Really develop tech & power skills.
- 👉 Experience real peer-learning, start learning how to learn and solve new problems.
- 👉 The learning journey is more important than the result.
- 👉 Learn about the risks associated with AI, and develop effective control practices and countermeasures to avoid common pitfalls.

● Learner rules:

- You should apply reasoning to your assigned tasks, especially before turning to AI.
- You should not ask for direct answers to the AI.
- You should learn about 42 global approach on AI.

● Phase outcomes:

Within this foundational phase, you will get the following outcomes:

- Get proper tech and coding foundations.
- Know why and how AI can be dangerous during this phase.

● Comments and example:

- Yes, we know AI exists — and yes, it can solve your activities. But you're here to learn, not to prove that AI has learned. Don't waste your time (or ours) just to demonstrate that AI can solve the given problem.
- Learning at 42 isn't about knowing the answer — it's about developing the ability to find one. AI gives you the answer directly, but that prevents you from building your own reasoning. And reasoning takes time, effort, and involves failure. The path to success is not supposed to be easy.
- Keep in mind that during exams, AI is not available — no internet, no smartphones, etc. You'll quickly realise if you've relied too heavily on AI in your learning process.
- Peer learning exposes you to different ideas and approaches, improving your interpersonal skills and your ability to think divergently. That's far more valuable than just chatting with a bot. So don't be shy — talk, ask questions, and learn together!
- Yes, AI will be part of the curriculum — both as a learning tool and as a topic in itself. You'll even have the chance to build your own AI software. In order to learn more about our crescendo approach you'll go through in the documentation available on the intranet.

✓ Good practice:

I'm stuck on a new concept. I ask someone nearby how they approached it. We talk for 10 minutes — and suddenly it clicks. I get it.

✗ Bad practice:

I secretly use AI, copy some code that looks right. During peer evaluation, I can't explain anything. I fail. During the exam — no AI — I'm stuck again. I fail.

Chapter 4

General guidelines

- The use of VirtualBox (or UTM if you can't use VirtualBox) is mandatory.
- You only have to turn in a `signature.txt` file at the root of your repository. You must paste the signature of your machine's virtual disk in it. Go to Submission and peer-review for more information.
- The use of snapshots is forbidden. If you use them, your grade will be 0.

Chapter 5

Mandatory part

This activity consists of having you set up your first server by following specific rules.



Since it is a matter of setting up a server, you will install the minimum of services. For this reason, a graphical interface is of no use here. It is therefore forbidden to install X.org, or Wayland, or any other equivalent graphics server. Otherwise, your grade will be 0.

You must choose as an operating system either the latest stable version of Debian (no testing/unstable), or the latest stable version of Rocky. Debian is highly recommended if you are new to system administration.



Setting up Rocky is quite complex. Therefore, you don't have to set up KDump. However, SELinux must be running at startup and its configuration has to be adapted for the project's needs. AppArmor for Debian must be running at startup too.

You must create at least 2 encrypted partitions using LVM. Below is an example of a possible partitioning:

```
wil@wil:~$ lsblk
NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda        8:0    0   8G  0 disk 
└─sda1     8:1    0  487M 0 part /boot
└─sda2     8:2    0   1K  0 part 
└─sda5     8:5    0  7.5G 0 part 
  └─sda5_crypt 254:0  0  7.5G 0 crypt 
    ├─wil--vg-root 254:1  0  2.8G 0 lvm   /
    ├─wil--vg-swap_1 254:2  0  976M 0 lvm   [SWAP]
    └─wil--vg-home 254:3  0  3.8G 0 lvm   /home
sr0       11:0   1 1024M 0 rom 

wil@wil:~$ _
```



During the defense, you will be asked a few questions about the operating system you chose. For instance, you should know the differences between aptitude and apt, or what SELinux or AppArmor is. In short, understand what you use!

A SSH service will be running on the mandatory port 4242 in your virtual machine. For security reasons, it must not be possible to connect using SSH as root.



The use of SSH will be tested during the defense by setting up a new account. You must therefore understand how it works.

You have to configure your operating system with the UFW (or firewalld for Rocky) firewall and thus leave only port 4242 open in your virtual machine.



The example shows arbitrary disk sizes. You need to determine the appropriate size for each partition to ensure proper operation while avoiding unnecessary disk usage.



Your firewall must be active when you launch your virtual machine. For Rocky, you have to use firewalld instead of UFW.

- The `hostname` of your virtual machine must be your login ending with 42 (e.g., wil42). You will have to modify this `hostname` during your review.
- You have to implement a strong password policy.
- You have to install and configure `sudo` following strict rules.
- In addition to the root user, a user with your login as username has to be present.
- This user has to belong to the `user42` and `sudo` groups.



During the defense, you will have to create a new user and assign it to a group.

To set up a strong password policy, you have to comply with the following requirements:

- Your password has to expire every 30 days.
- The minimum number of days allowed before the modification of a password will be set to 2.
- The user has to receive a warning message 7 days before their password expires.
- Your password must be at least 10 characters long. It must contain an uppercase letter, a lowercase letter, and a number. Also, it must not contain more than 3 consecutive identical characters.

- The password must not include the name of the user.
- The following rule does not apply to the root password: The password must have at least 7 characters that are not part of the former password.
- Of course, your root password has to comply with this policy.



After setting up your configuration files, you will have to change all the passwords of the accounts present on the virtual machine, including the root account.

To set up a strong configuration for your `sudo` group, you have to comply with the following requirements:

- Authentication using `sudo` has to be limited to 3 attempts in the event of an incorrect password.
- A custom message of your choice has to be displayed if an error due to a wrong password occurs when using `sudo`.
- Each action using `sudo` has to be archived, both inputs and outputs. The log file has to be saved in the `/var/log/sudo/` folder.
- The TTY mode has to be enabled for security reasons.
- For security reasons too, the paths that can be used by `sudo` must be restricted. Example: `/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin`

Finally, you have to create a simple script called `monitoring.sh`. It must be developed in `bash`.

At server startup, the script will display some information (listed below) on all terminals every 10 minutes (take a look at `wall`). The banner is optional. No error must be visible.

Your script must always be able to display the following information:

- The architecture of your operating system and its kernel version.
- The number of physical processors.
- The number of virtual processors.
- The current available RAM on your server and its utilization rate as a percentage.
- The current available storage on your server and its utilization rate as a percentage.
- The current utilization rate of your processors as a percentage.
- The date and time of the last reboot.
- Whether LVM is active or not.
- The number of active connections.
- The number of users using the server.
- The IPv4 address of your server and its MAC (Media Access Control) address.
- The number of commands executed with the `sudo` program.



During the defense, you will be asked to explain how this script works. You will also have to interrupt it without modifying it. Take a look at `cron`.

Example of how this script is expected to work



Broadcast message from root@wil (tty1) (Sun Apr 25 15:45:00 2021):

```
#Architecture: Linux wil 4.19.0-16-amd64 #1 SMP Debian 4.19.181-1
(2021-03-19) x86_64 GNU/Linux
#CPU physical: 1
#vCPU: 1
#Memory Usage: 74/987MB (7.50%)
#Disk Usage: 1009/2Gb (49%)
#CPU load: 6.7%
#Last boot: 2021-04-25 14:45
#LVM use: yes
#Connections TCP: 1 ESTABLISHED
#User log: 1
#Network: IP 10.0.2.15 (08:00:27:51:9b:a5)
#Sudo: 42 cmd
```

Below are two commands you can use to check some of the subject's requirements:

For Rocky:

```
[root@wil will]# head -n 2 /etc/os-release
NAME="Rocky Linux"
VERSION="8.7 (Green Obsidian)"
[root@wil will]# sestatus
SELinux status:                 enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                   enforcing
Mode from config file:          enforcing
Policy MLS status:              enabled
Policy deny_unknown status:     allowed
Memory protection checking:     actual (secure)
Max kernel policy version:      33
[root@wil will]# ss -tunlp
Netid State Recv-Q Send-Q Local Address:Port    Peer Address:Port Process
tcp   LISTEN  0      128      0.0.0.0:4242          0.0.0.0:*      users:(("sshd",pid=28429,fd=6))
tcp   LISTEN  0      128      [::]:4242           [::]:*        users:(("sshd",pid=28429,fd=4))
[root@wil will]# firewall-cmd --list-service
ssh
[root@wil will]# firewall-cmd --list-port
4242/tcp
[root@wil will]# firewall-cmd --state
running
[root@wil will]# _
```

For Debian:

```
root@wil:~# head -n 2 /etc/os-release
PRETTY_NAME="Debian GNU/Linux 10 (buster)"
NAME='Debian GNU/Linux'
root@wil:/home/wil# /usr/sbin/aa-status
apparmor module is loaded.
root@wil:/home/wil# ss -tunlp
Netid State  Recv-Q Send-Q Local Address:Port  Peer Address:Port
tcp   LISTEN  0      128          0.0.0.0:4242      0.0.0.0:*      users:(("sshd",pid=523,fd=3))
tcp   LISTEN  0      128          [::]:4242        [::]:*       users:(("sshd",pid=523,fd=4))
root@wil:/home/wil# /usr/sbin/ufw status
Status: active

To                         Action      From
--                         -----      ---
4242                       ALLOW       Anywhere
4242 (v6)                   ALLOW       Anywhere (v6)
```

Chapter 6

Readme Requirements

A README.md file must be provided at the root of your Git repository. Its purpose is to allow anyone unfamiliar with the activity (peers, staff, recruiters, etc.) to quickly understand what the activity is about, how to run it, and where to find more information on the topic.

The README.md must include at least:

- The very first line must be italicized and read: *This activity has been created as part of the 42 curriculum by <login1>[, <login2>[, <login3>[...]]].*
 - A "**Description**" section that clearly presents the activity, including its goal and a brief overview.
 - An "**Instructions**" section containing any relevant information about compilation, installation, and/or execution.
 - A "**Resources**" section listing classic references related to the topic (documentation, articles, tutorials, etc.), as well as a description of how AI was used — specifying for which tasks and which parts of the activity.
- ➡ **Additional sections may be required depending on the activity** (e.g., usage examples, feature list, technical choices, etc.).

Any required additions will be explicitly listed below.

- A **Project description** section must also explain the choice of operating system (Debian or Rocky), with their respective pros and cons. It must indicate the main design choices made during the setup (partitioning, security policies, user management, services installed) as well as a comparison between:
 - Debian vs Rocky Linux
 - AppArmor vs SELinux
 - UFW vs firewalld
 - VirtualBox vs UTM



English is recommended; alternatively, you may use the main language of your campus.

Chapter 7

Submission and peer-review

You only have to turn in the expected README file, and a `signature.txt` file at the root of your Git repository. You must paste the signature of your machine's virtual disk in it. To get this signature, you first have to open the default installation folder (it is the folder where your VMs are saved):

- Windows: `%HOMEDRIVE%%HOMEPATH%\VirtualBox VMs\`
- Linux: `~/VirtualBox VMs/`
- MacM1: `~/Library/Containers/com.utmapp.UTM/Data/Documents/`
- MacOS: `~/VirtualBox VMs/`

Then, retrieve the signature from the ".vdi" file (or ".qcow2 for UTM'users) of your virtual machine in sha1 format. Below are 4 command examples for a `rocky_serv.vdi` file:

- Windows: `certUtil -hashfile rocky_serv.vdi sha1`
- Linux: `sha1sum rocky_serv.vdi`
- For Mac M1: `shasum rocky.utm/Images/disk-0.qcow2`
- MacOS: `shasum rocky_serv.vdi`

This is an example of what kind of output you will get:

- `6e657c4619944be17df3c31faa030c25e43e40af`



Please note that your virtual machine's signature will be altered after each review. To solve this problem, you can duplicate your virtual machine or use `save state`.



It is, of course, FORBIDDEN to turn in your virtual machine in your Git repository. During the review, the signature of the `signature.txt` file will be compared with the one of your virtual machine. If the two of them are not identical, your grade will be 0.



The use of snapshots is FORBIDDEN. During the review, if any snapshot is detected, your grade will be 0.