COMET

A DEBIAN-TOOL

Genesis

For any ethical hacking, many people starts with Kali Tools. The fastest and easiest way is to download an ISO and run this image with a virtual-environment tool.

Virtual Machines could be installed on many systems.

But. Yes, it's a But at this point. But it is not the correct way. I don't have any judgement.

My proposition is this tool, made in a logical thinking. It's a fact.

Many professional workers share to not use Kali as a professional tool.
To understand exactly the explanation, listen three main points.

1. **Best capacity**. Kali Tool in a Virtual Machine is not enough sufficient in a high technical point. If you reply Kali is also available in pods: yes, you have right.

2. **Professional Work**. Kali is not enough for each customers. You can create easily one pod for each customer. You have one environment and you can work inside without a fear to destroy or to mix data between customers. With only One Virtual Machine, data has a high rate to be mixed. Keep in mind this mantra: create a new environment as a start then execute your work procedure and finaly write the document procedure as your final delivery.

3. **Free**. Stay Free. Free of the cost of any Licence. Free for choosing your own tool.
You can learn as personal way and then apply your lessons in a profesional way. With a same tool. You learn your tool all the time during your journey.
Free? Okay, but some will reply I use Kali Tools, so I'm not fully Free. What? Not free? But Kali is free as this quote in their website: << Kali Linux is an *open-source ". Free to use. Perhaps is it possible to find for each shell a free shell which is not from Kali? But why? What is the goal?* :)
*This tool **Comet** could be upgraded with a collaborating work! Better than to be angry, have a hunger for curiosity. Doing a proposition is better than only sharing a complaint.*
The Mind is also to use all layers available as "Open Source" as possible. At least, as Free as possible.
Samples: Notepad++, LibreOffice, GIMP, Podman, Debian, Kali Linux.

4. Your own tool. It is not a main point. It's an advice. Learn Docker, understand any process. And then switch to Podman. Download any Pod and learn deeply how it is functional. Copy and modify any Container, any Pod. It's the best way to learn. It's a flexible tool.

Try it!. The mind is to have a special dedicated environment: an universe. In this universe, some asteroid can crash some planets. As an attacker.

Come to Comet and Welcome in my Universe !

[@ r00t4M0NK] *https://github.com/r00t4M0NK/*

Run the Comet Pod

[INSTALL] [WINDOWS] [PODMAN]

Why Windows? It's enough as a standard and easy to find when you buy a computer.

Up to you to change this setting and update this document with your own choice.

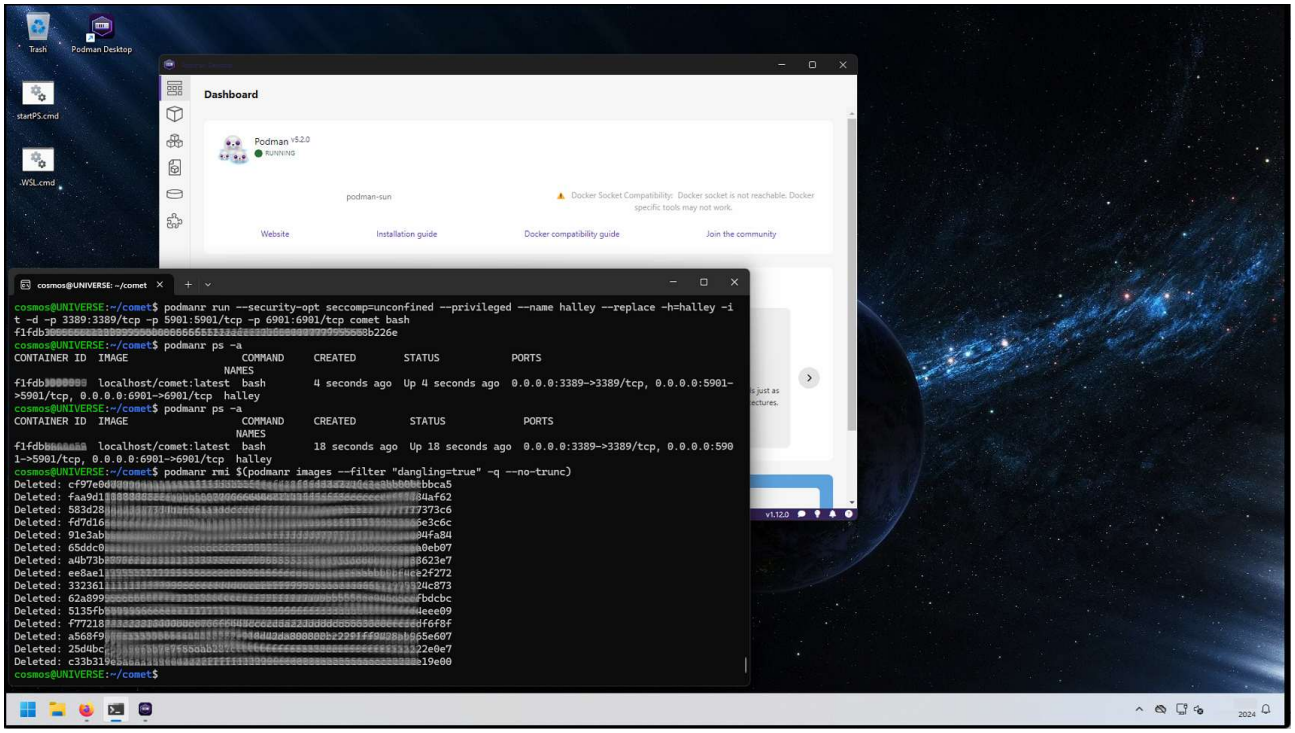
Here it's an example with a choice Windows/Podman (but it's possible to change and choose Linux/Docker)

- Step 1. In WSL2, create your work directory. As mkdir -p /home/<your_user>/comet (with correct rights as 755, correct user, up to you)
- Step 2. Follow instructions inside **Dockerfile**. Chapter "DOWNLOAD DIRECTLY" is an easy-way to set up fast your environment: Build, Clean, Run.
- Step 3. Check the Container which is up.
- Step 4. To add the CyberSecurity Layer (CSL), refer to /tools/bin/cyberhacktoolsinstall.sh (check the GitHub) in chapter "For Downloading parts" with 10 steps.
- Step 5. Enjoy.

STEP 2

Here, podman means "podman –remote" as WSL is linked to Podman-Machine & Podman-Desktop.
Do first the install, set all settings and check.
You need to to acquire your first skill in working in full autonomy.

If you need to see a sample, here it's a machine Windows with Podman:



Note: the command "run" is right for VPN & NMAP. Refer comments in the code for this point.
And the full command is decribed in this document below.

In my case, the using is WSL-user with « sudo docker » for a Docker build, and WSL-user whit « podman », standing for « podman --remote »
In all cases, install is done with root or « sudo apt install »

[TOOLS]
WSL-root> apt-get install -y curl dos2unix

[DOWNLOAD]
WSL-user> curl https://raw.githubusercontent.com/r00t4M0NK/BFHBP_pbc/refs/heads/main/Dockerfile > Dockerfile
WSL-user> dos2unix Dockerfile

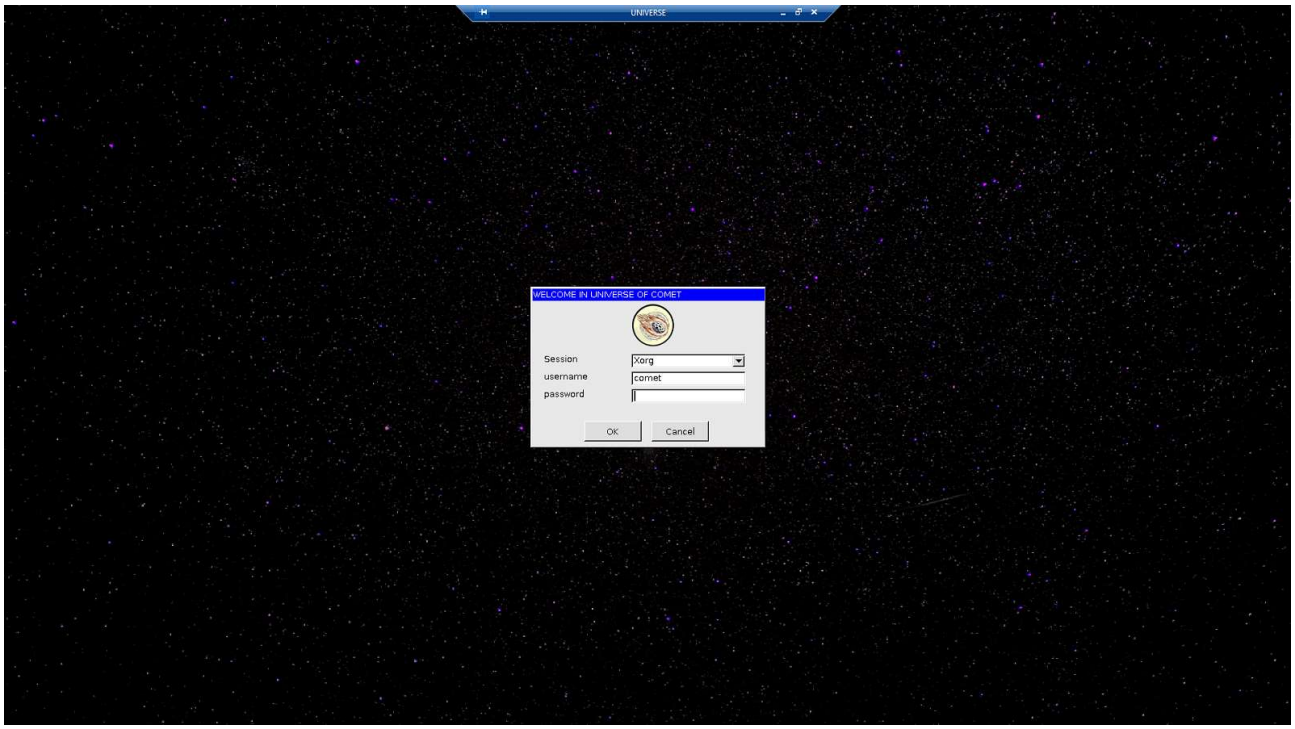
[BUILD]
WSL-user> podman build --rm -t comet .

[CLEAN]
WSL-user> podman rmi $(podman images --filter "dangling=true" -q –no-trunc)

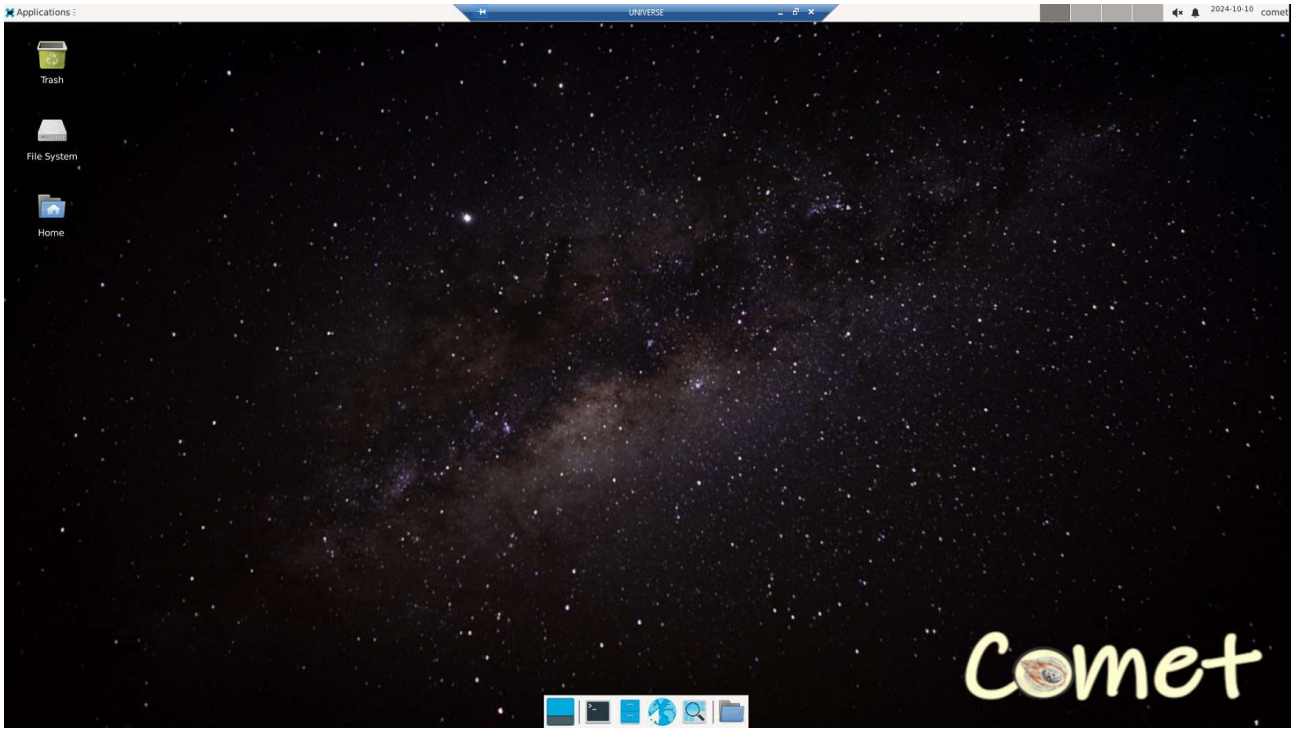
[RUN]
WSL-user> podman run --replace --security-opt seccomp=unconfined --privileged --name halley --replace -h=halley -it -d -p 3389:3389/tcp -p 5901:5901/tcp -p 6901:6901/tcp --cap-add=NET_ADMIN comet bash

STEP 3

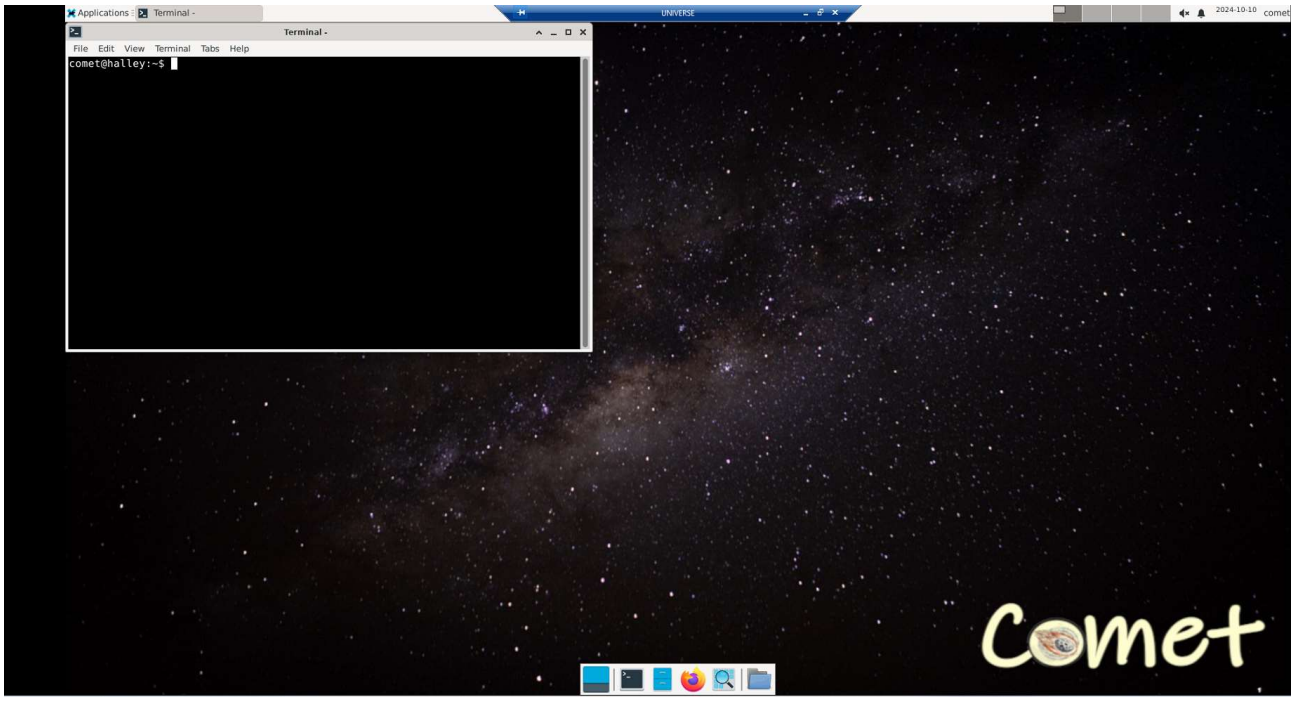
Connect through RDP to your server.



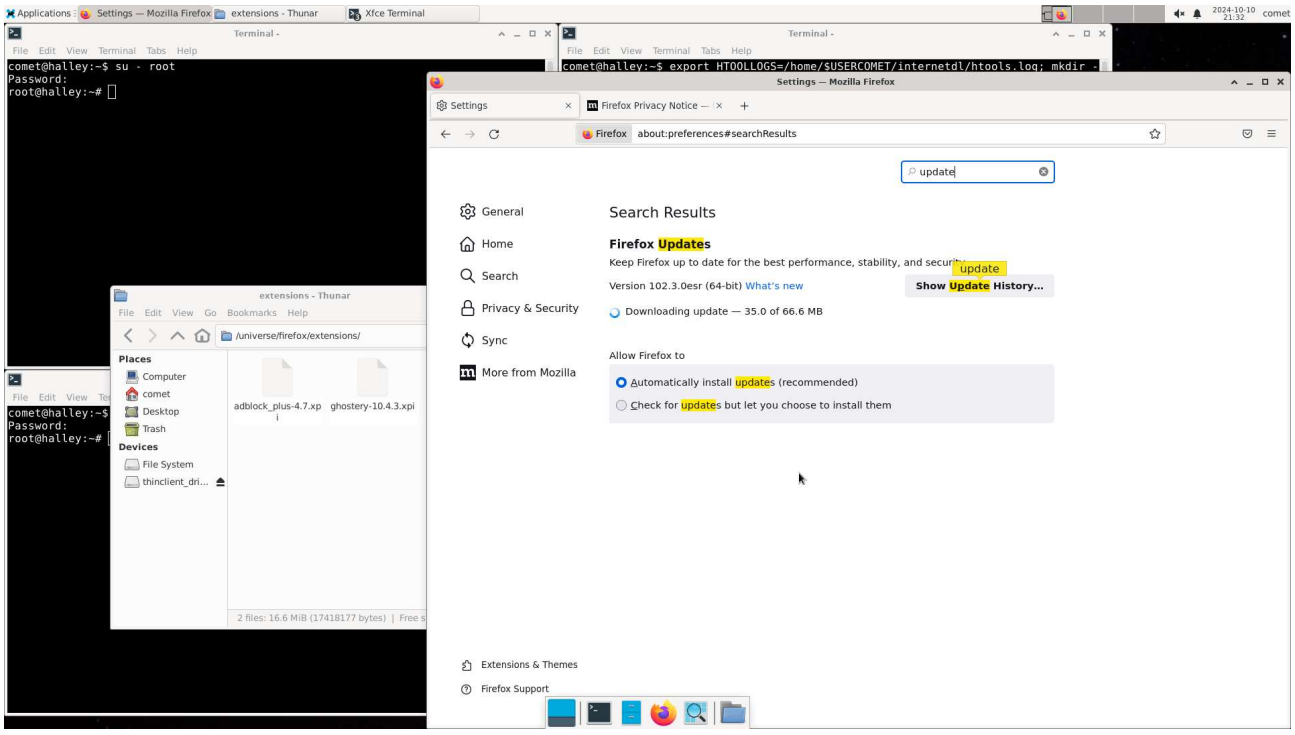
First Connection shows a default browser:



Start a Terminal and see the update for Firefox (needs an additional work to have this same view through VNC):



Update Firefox and prepare to install CSL:

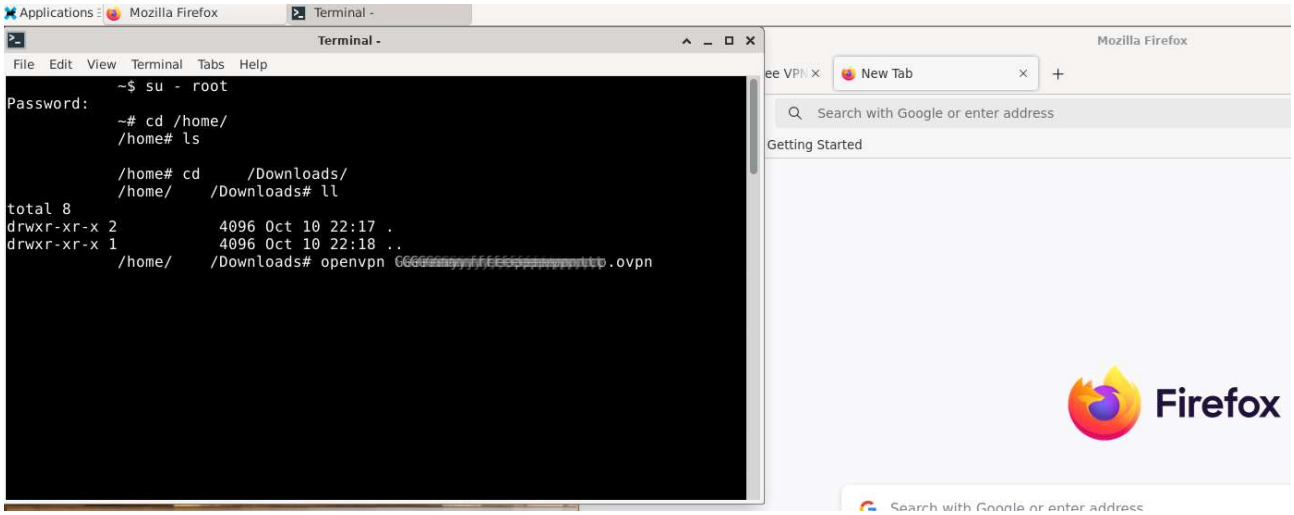


When Firefox is updated, drag&drop extensions inside it to use them as filters.

install CSL (CyberSecurity Layer) ⇒ refer to the GitHub (see above)

Run the VPN

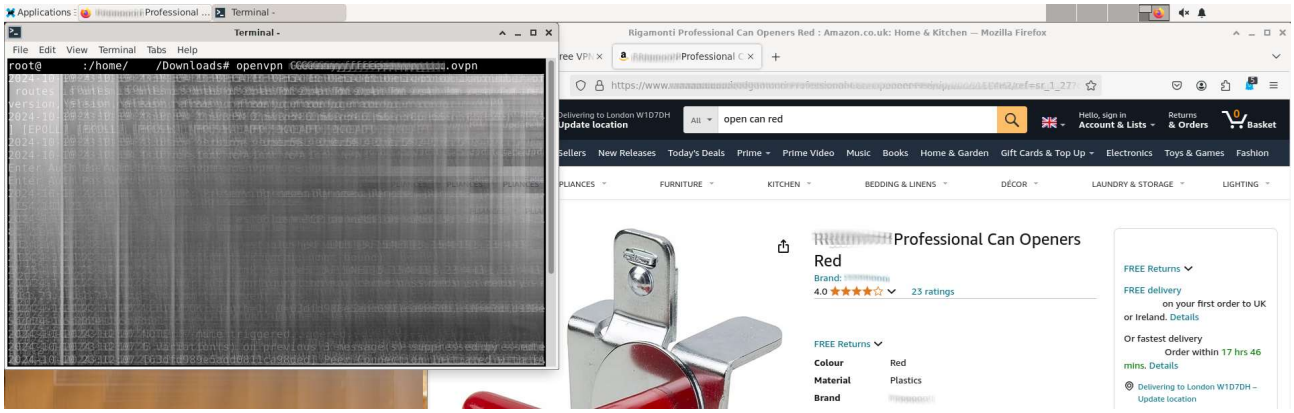
Start OpenVPN in the machine pod (example with Docker):



Start OpenVPN in the machine host (example with Podman)
WSL-host-root> openvpn <file>

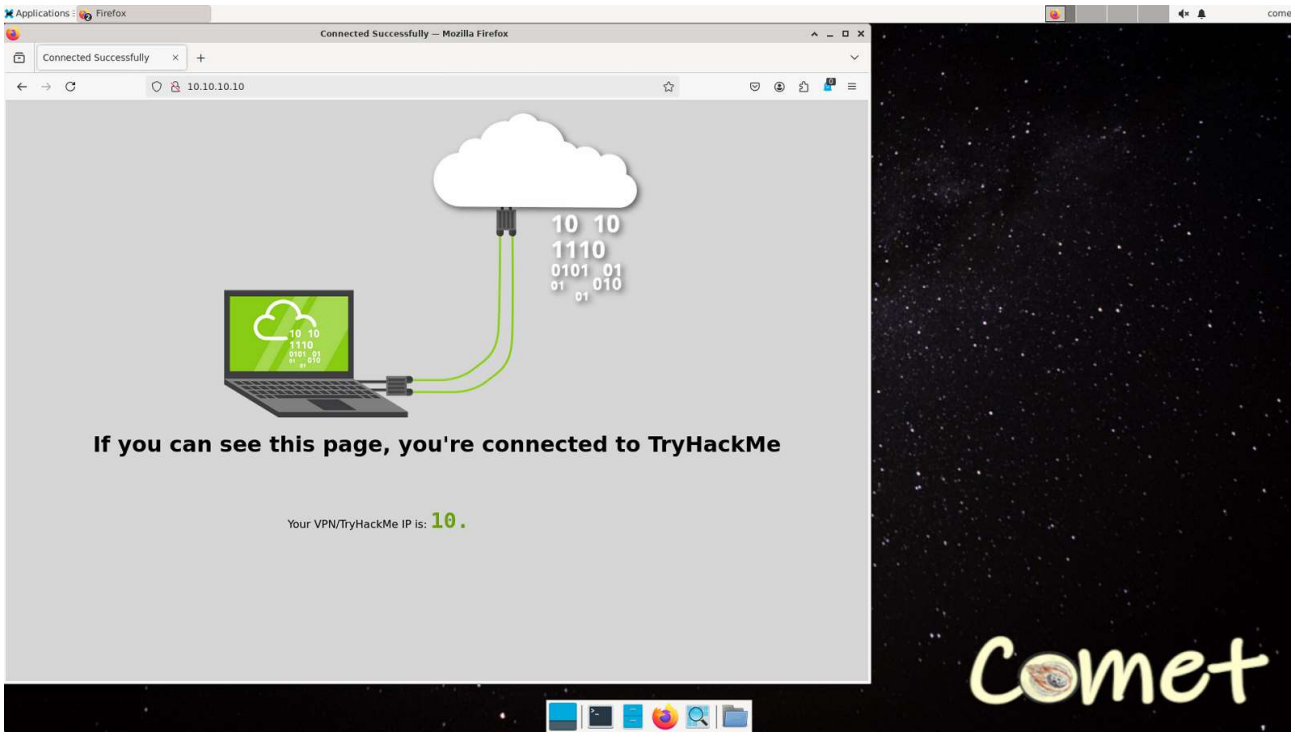
Then enter user/password, if needed.
Or use this line (you can apply it too for a Docker environment):
WSL-root> export CONFIG_FILE=<filesetting_xxx.ovpn>; export VPN_USER=<user>; export VPN_PASSWORD=<****>; bash -c "openvpn --config "'"$CONFIG_FILE"' --auth-user-pass <(echo -e "$VPN_USER"'"\n"'"$VPN_PASSWORD"')"

Then check inside the Pod before using this environment as a public show from your computer:
(here, Docker)



Or use any tool which show your IP & your informations.

Start OpenVPN in the machine host (example with Podman & TryHackMe):



Install CSL

To install the CyberSecurity Layer (script "cyberhacktoolsinstall.sh") :

- download the script and open with your favourite text reader (do these 2 actions the way you want).
- read instructions in this shell and learn the experience with cases already tested for you
- to run fast the install: open 3 terminal inside the pod: 1 with user and 2 with root
- target each "################# [STEP]": copy the line-instruction, and paste into the right terminal
- Step 4 is a control-step: up to you to run or not
- "terminal user" is for the first step
- "terminal root #1" is for all other steps
- "terminal root #2" is to control: change dir to /home/<user>/internetdl and tail the log file

One install could be "fast". But if you fail and restart everything, it can be a long time.

In order to save, do in WSL (podman as "podman –remote"):

```
WSL-user> podman stop halley
       PS> podman machine ls
       PS> podman machine stop podman-<machine>
       PS> wsl --shutdown
       PS> wsl --export podman-<machine> podman.tar
       PS> mv podman.tar <podman_<nameYouDecide>.tar
       PS> wsl
```

Meaning:
- stop the pod
- list machines
- stop the machine podman
- close WSL properly
- export the machine podman
- move and rename the machine into you save directory (private, as you decide)
- start again

/!\ Be carefull with documents. Sometimes, when an author writes the command, the editor sofware could replace some dash by a "picture-dash": this char is not executed inside a terminal. Have habits to check any command before executing.

Summary

- you need to install a container-software (Docker or Podman)
    * my choice to use a Desktop-Sofware it’s to have a graphical view: easy to share to a customer
    * my choice to use WSL is to have an unix-shell: commands are easy to share to you
    * my not choice to use Windows it’s for buying a machine hardware and have a ready-system to use
    * install the software in your system (Desktop App included)
    * install WSL (in my case, I choose Debian for WSL distro) if you use a Windows system
    * work on settings for sending command from WSL to the Desktop software (network skills)
    * help yourself with exprience and commands described in comment in Dockerfile
- when product is installed and work (you can create image, or list elements with WSL and you see all result in the Desktop App too), you are ready for the Comet-sofware part!
- in WSL, create the work-directory and prepare the Dockerfile
- follow instructions to download the source-file from GitHub, build the image, and do the clean part
- follow instructions to run your Comet-Pod
- connect to the pod: update the Firefox Browser, add extension if you want or use yours
- install CSL to add the "Ethical Hacking" Layer in this software
- stop pod, stop wsl, and do an export (export machine in case of Podman) & restart to have pod up

You are ready with your environment to learn and test your skills with differents learning platforms.

CONCLUSION

[THE TOOL]

What can do this tool ?

- On the start, it’s near to be a ready-tool to learn Ethical Hacking. You have your own machine, with your own settings, and you plug it into any learning-platforms which has created their own network to secure the perimeter and for the company and for the "user".
- **Comet** can be changed into another tool: a desktop environment (to use internet tools without any personal informations and share your screen to public), an anonymous environment (but it’s needed to add another layer on **Comet**), a streaming server environment (not tested), a gaming environment as a server (not tested), and so on.
- **Comet** can be a Full Free Tool, depending of your choice of components. Up to you to check them and build for your own reasons. The main limitation is to stay in Legal Actions. You can learn with the Tool and apply your lessons in a business work with the same tool, wihtout a cost licence.
- **Comet** is generic as posible it is. But Differences can be found between your choice, as Docker or Podman. As many software component, this tool requires sometimes some work and you need to understand what you do. Example: mechanism of VPN is not the same if you run as a Docker Container or as a Podman Pod. It’s Portable (**Comet** can run from many systems as Linux, Mac, Windows). And all test cases cannot be fully tested. Help yourself with all technical comments available into 3 files: the Dockerfile, the script CSL file and the Tips file.
- Enjoy !!