# Experimentation and Analysis of Present Solutions: Phishpedia

***A Project Report***

This report represents the submission of R3, as required for the fulfillment of graduation requirements

## Experimental Report for Phishapedia


By

**Fatimah Almusaid** G202306890

**Atika Alnaim** G202306850

**Rawan Alali** G202307110

Supervised by

**Dr. Waleed AlGobi**

***King Fahd University of Petroleum and Mineral***

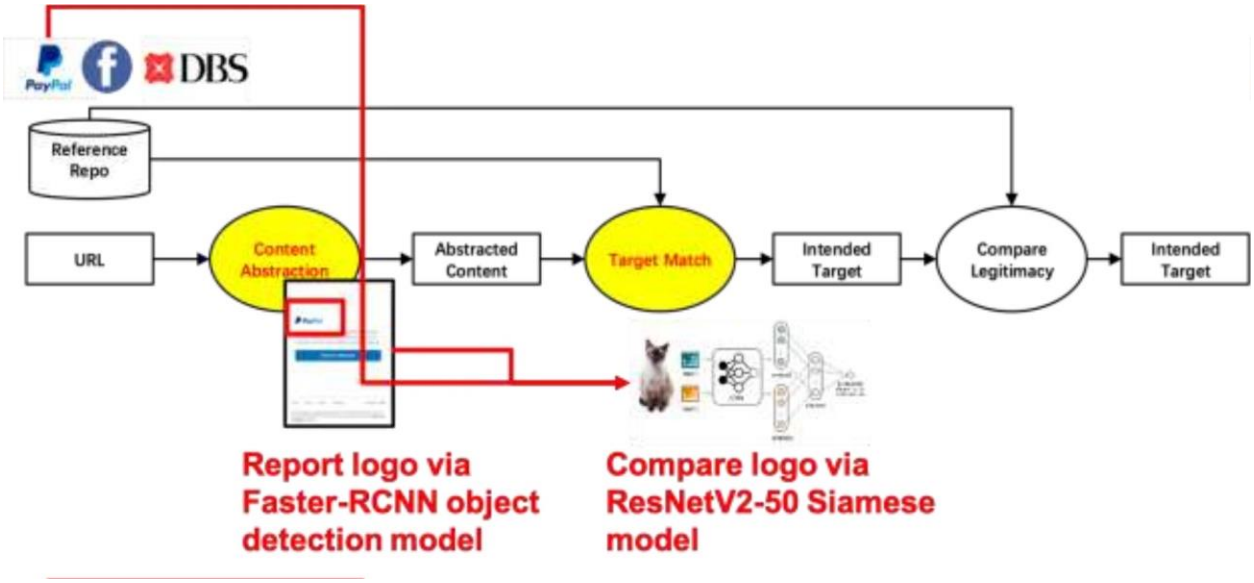
***The Department of Information and Computer Science***


*December 3, 2024*

# Table of Contents

# Experimentation and Analysis of Present Solutions: Phishpedia

PhishPedia is a phishing detection tool that leverages deep learning to identify phishing websites based on visual elements. By analyzing logos and webpage features, PhishPedia can effectively distinguish between legitimate and phishing sites, helping protect users from fraudulent attacks. It integrates pre-trained models to compare suspected phishing pages against known brand assets, offering an efficient solution for proactive threat identification.




## 1. Preparing to Install Kali Linux on VirtualBox

To create and prepare a virtual machine for Kali Linux, you must load an ISO file and configure virtual hardware, such as memory, CPU cores, and hard disks. Follow the steps below to complete these actions.

### 1.1. Download Kali Linux ISO Image

Kali Linux offers ISO images for 32-bit, 64-bit, and ARM64 architectures. To download an ISO file:

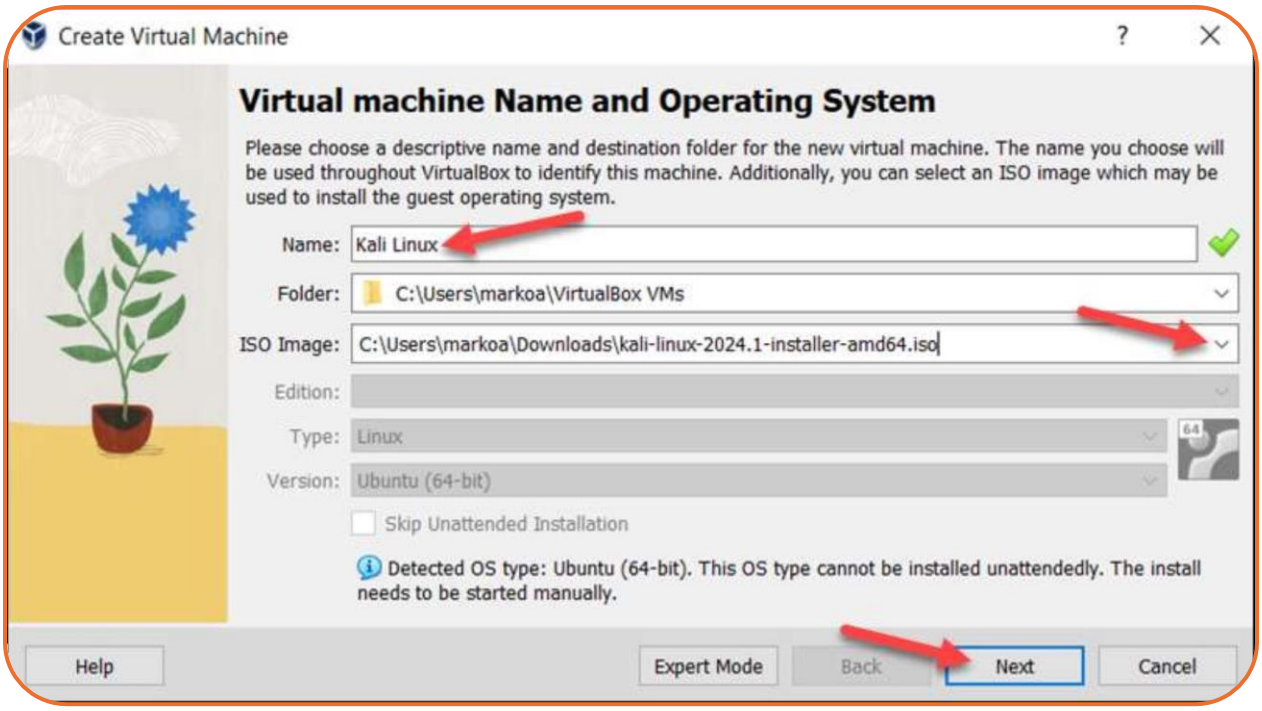
- Visit the installer section of the Kali Linux official website.
- Select the system architecture of the host OS and download the ISO file by clicking the button in the bottom-left corner of the installer card.


## 1.2. Create Kali Linux VirtualBox Instance

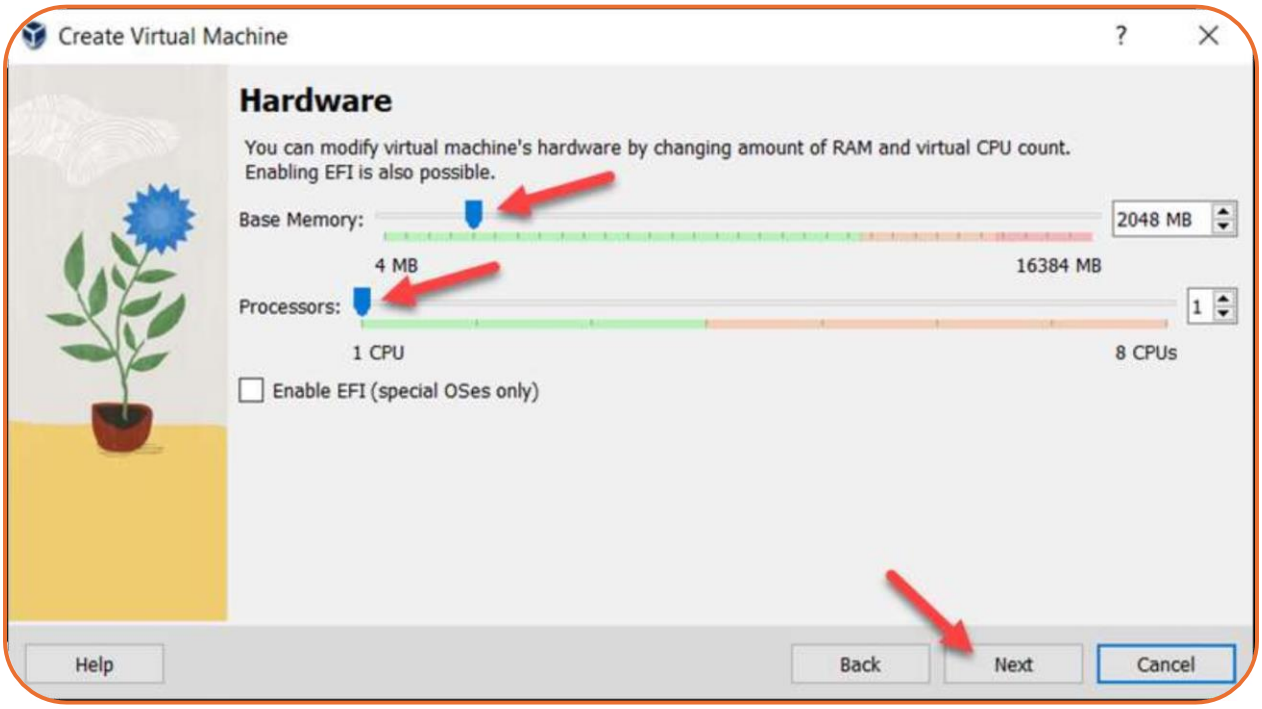
Create a new virtual machine and configure it to run Kali Linux. Proceed with the steps below to correctly set up a Kali Linux VM in VirtualBox:

- Launch **VirtualBox Manager** and click the **New** icon.

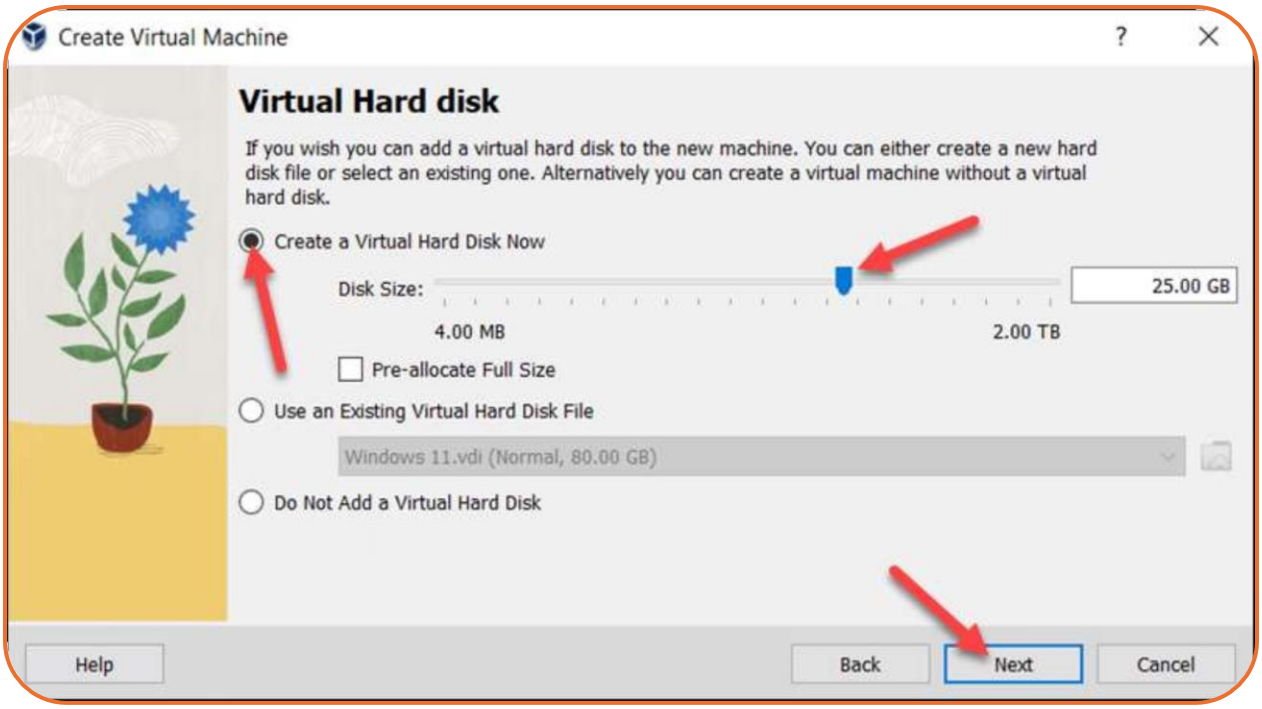

- Specify a name for the VM and provide the path to the ISO image. Select **Next**.



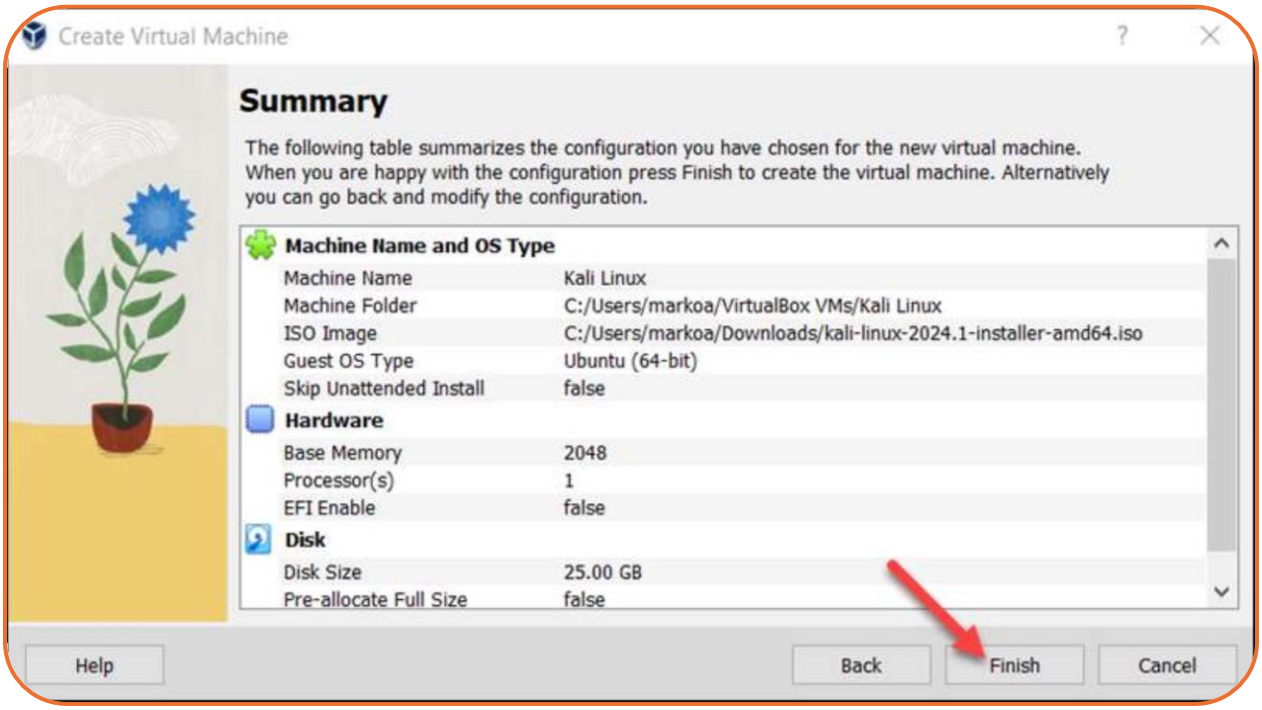

- Select the amount of memory and the number of virtual CPUs to allocate to the VM. The minimum recommended values for Kali Linux are **2 GB of RAM** and **1 CPU**. Select **Next** when you finish setting up the VM hardware.




- Create a virtual hard disk for the new VM. The recommended hard disk size is at least **25 GB**. Alternatively, you can use an existing virtual hard disk file or decide not to add one. Click **Next** to proceed to the next step.




- Review the new VM setup on the **Summary** page. Select **Finish** to create the virtual machine.

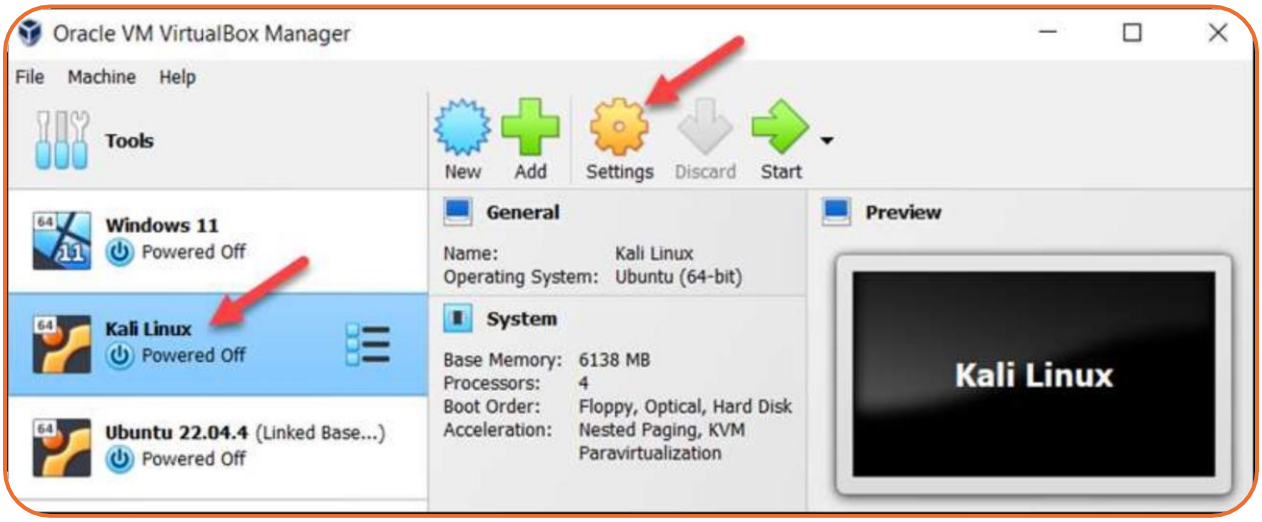



- The VM appears on the list in VirtualBox Manager.

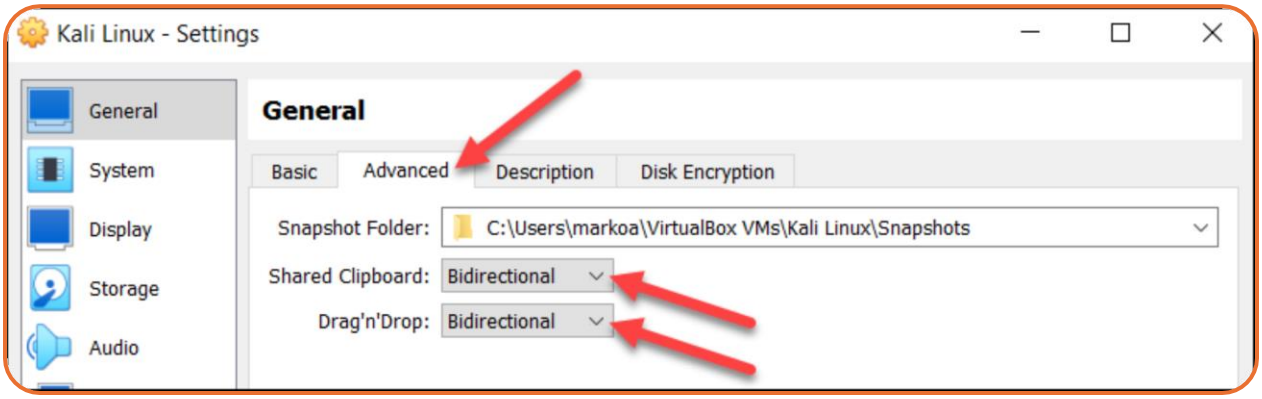
## 1.3. Configure Virtual Machine Settings and Start VM

Before starting the VM and beginning the installation process, follow the steps below to perform additional adjustments to the VM:

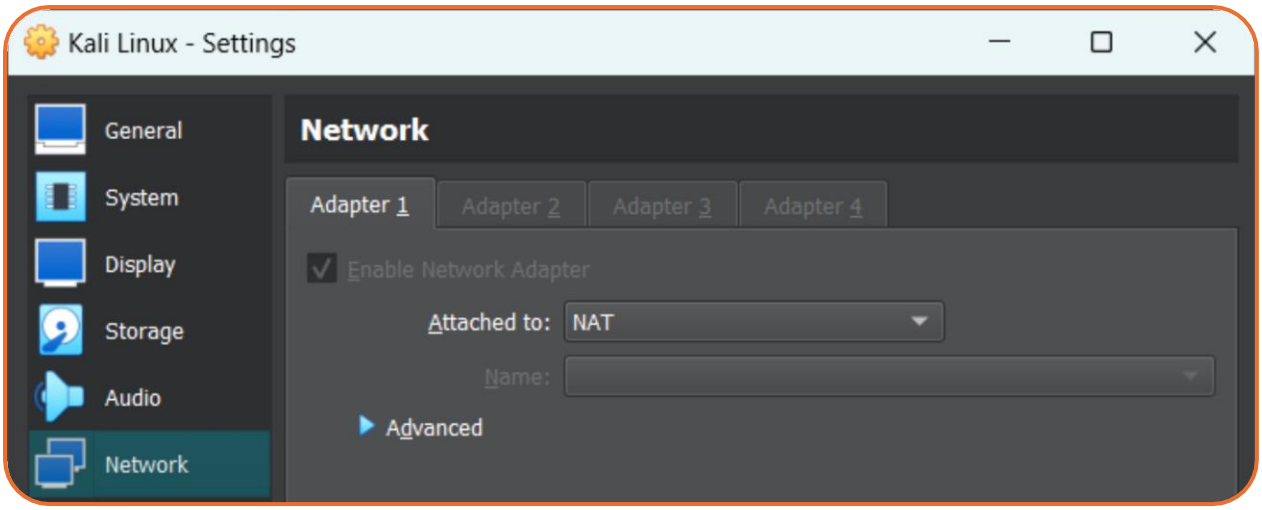
- Select the Kali Linux VM and click the **Settings icon**.



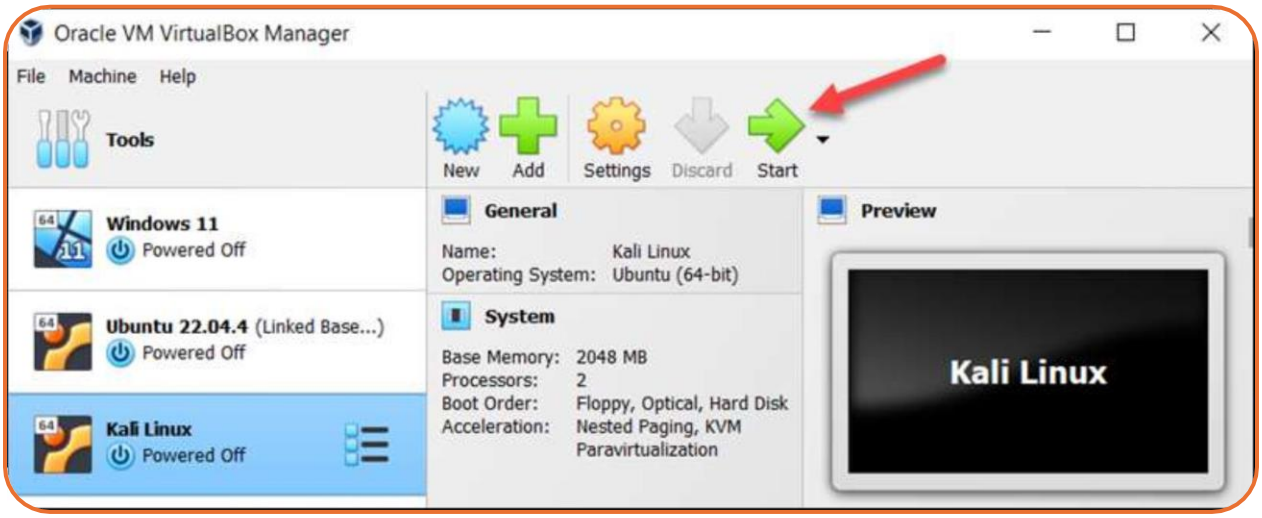

- Select the **Advanced** tab in the **General** section and change the **Shared Clipboard** and **Drag'n'Drop** settings to **Bidirectional**. This feature allows the host and the guest machine to exchange files.




- **Warning**: Connecting the host and VM clipboards breaks VM isolation. Use this option with caution.
- Select **Network** from the menu on the left side. Change the **Attached to** field to **NAT Adapter**. Select **OK** at the bottom of the window to return to the main window.



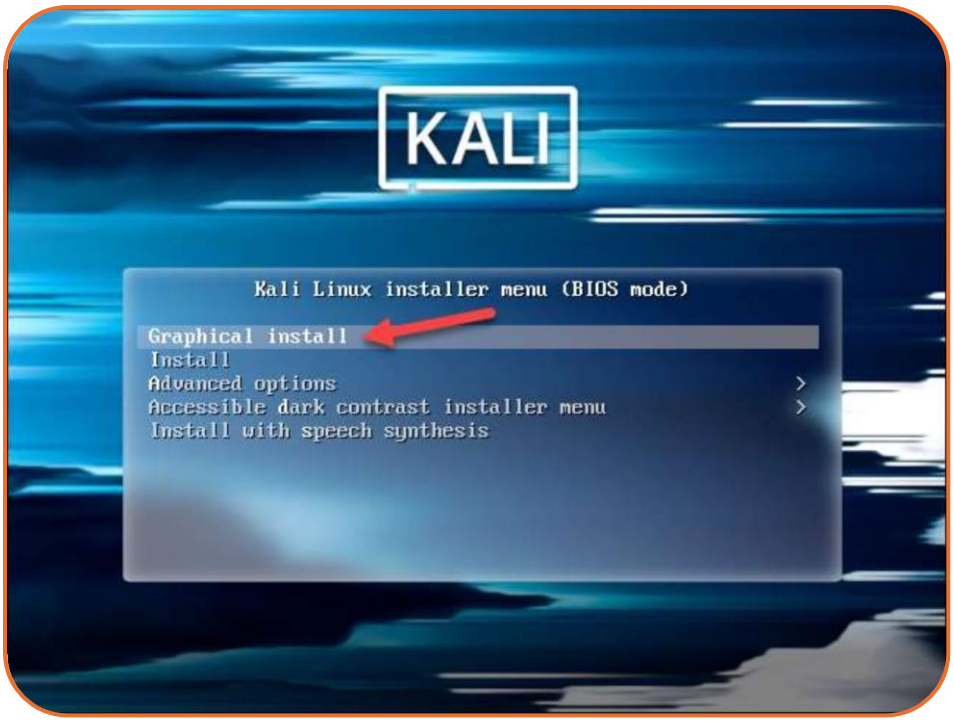

- 4. Click **Start** to begin installing Kali Linux.




- Kali Linux uses the Debian installer to set up the operating system. The sections below provide a detailed walkthrough of the installer and offer advice on configuring Kali Linux.

## 1.4. Perform Initial Configuration

- When the new VM is started, the Kali Linux installer menu appears. Start the installation procedure by following the steps below:
- Select the **Graphical install** option.

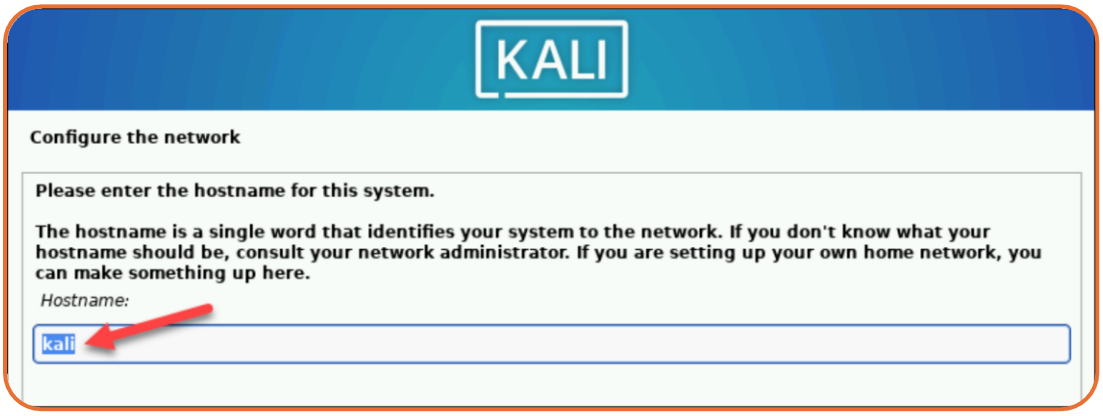



- Choose the system's **default language**, which will also be used during installation.
- Find and select your **country** from the list or choose **other**.
- Decide which **keyboard mapping** to use.

## 1.5. Configure Host, User, and Time Zone

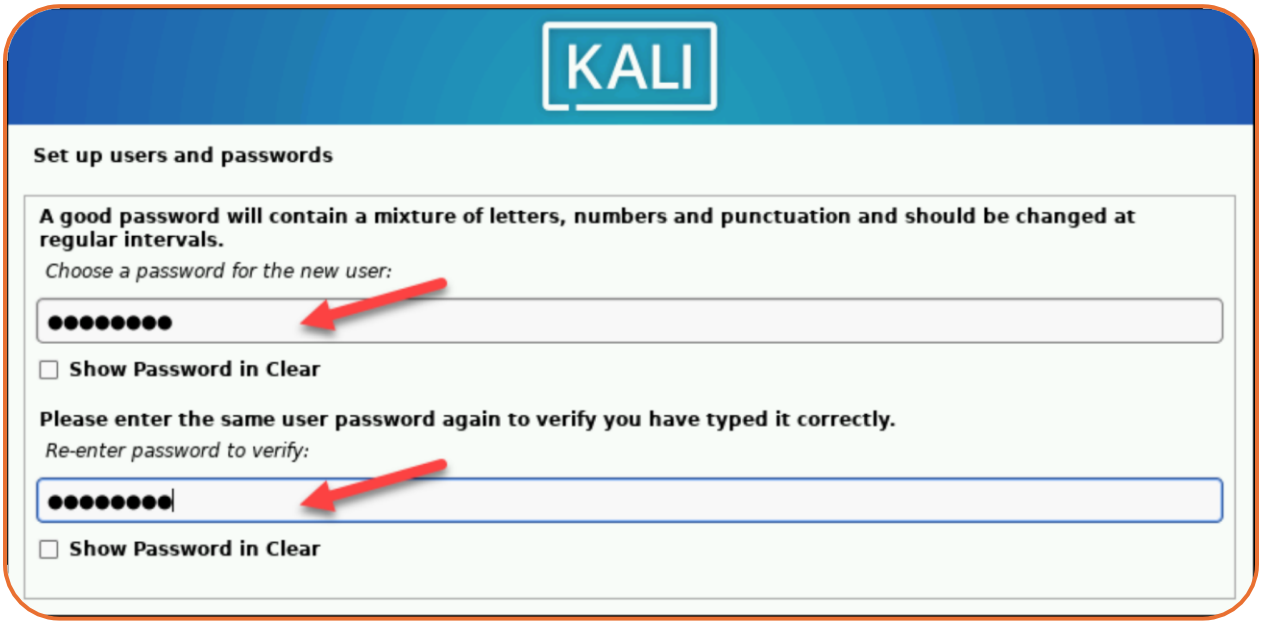
The following installer steps set up the hostname and domain of the system and configure the user:

- In the Configure the network section, enter a system hostname.




- Type a **domain name** that the OS will use to identify the VM within a network. Specifying a domain name is not necessary if the VM is not part of an extensive local network.


- Create a **user account** by providing the user's full name and username.
- Create a strong password for the user account.

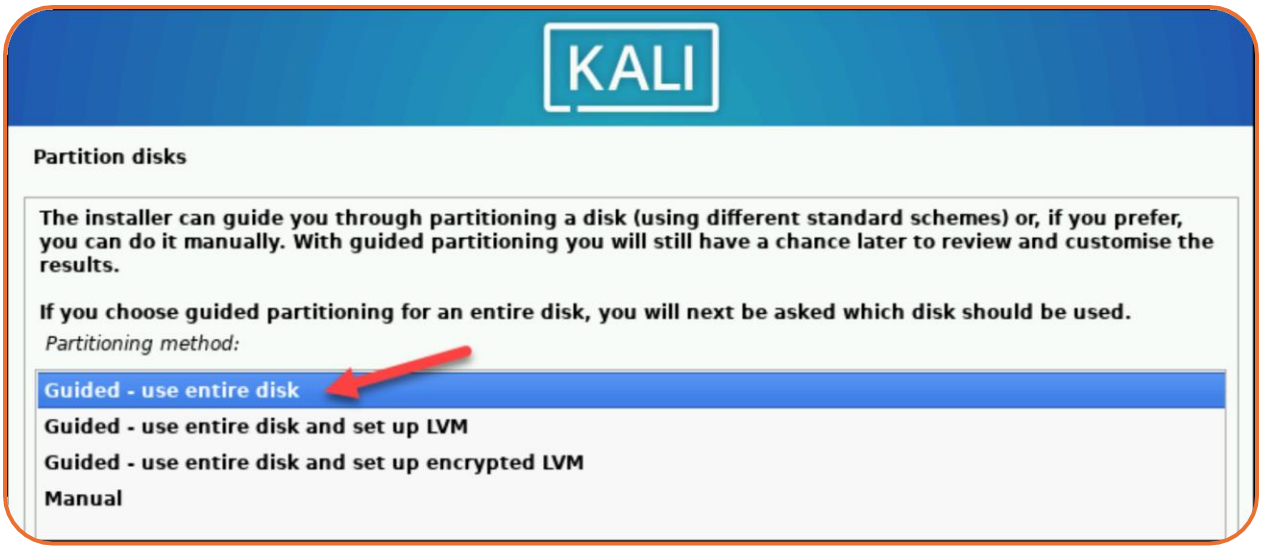



- 5. Select the correct **time zone** from the available options.

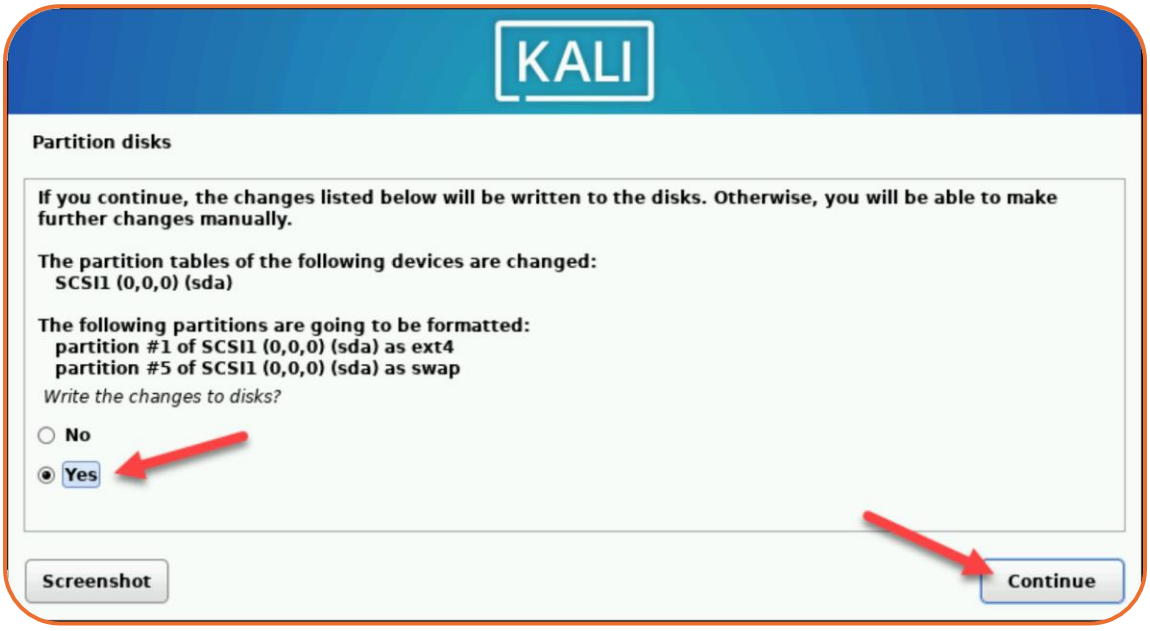
## 1.6. Create Hard Disk Partitions

Proceed with the following steps to create a bootable partition on the virtual hard disk:

- Select how to partition the hard disk. The default option is **Guided - use entire disk**.




- Select the disk you want to use for partitioning. The only available option is the disk created during the VM creation.
- Select the **partitioning scheme**. The default option is **All files in one partition**.
- The wizard provides an overview of the configured partitions. Ensure that the **Finish partitioning and write changes to disk** option is selected.
- Confirm the choice by selecting **Yes** on the next screen.

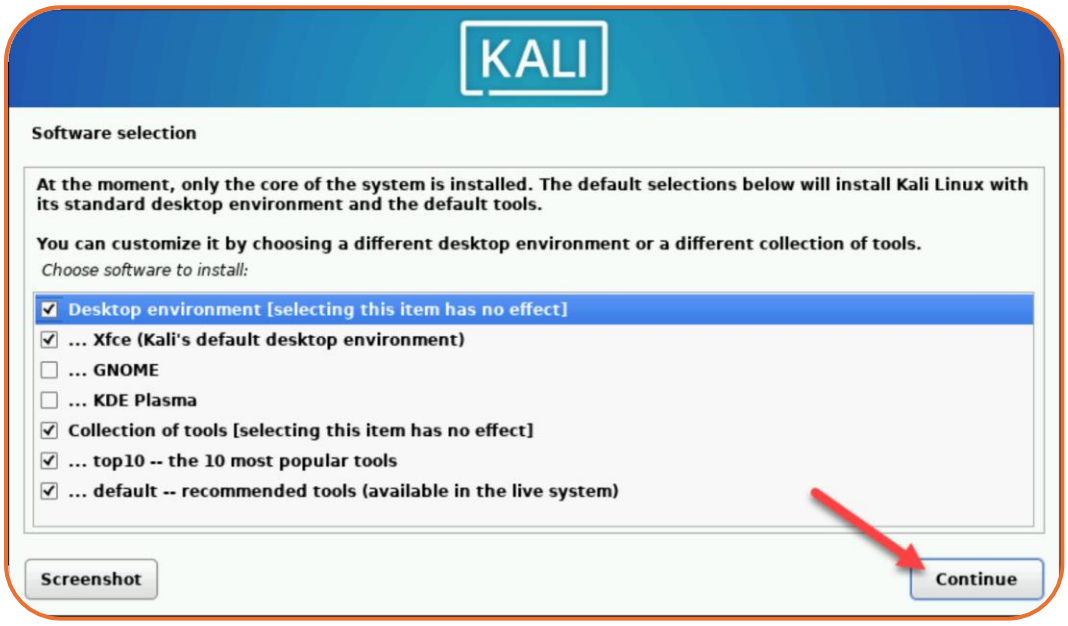



- The wizard starts installing Kali.

## 1.7. Customize Kali Linux Installation

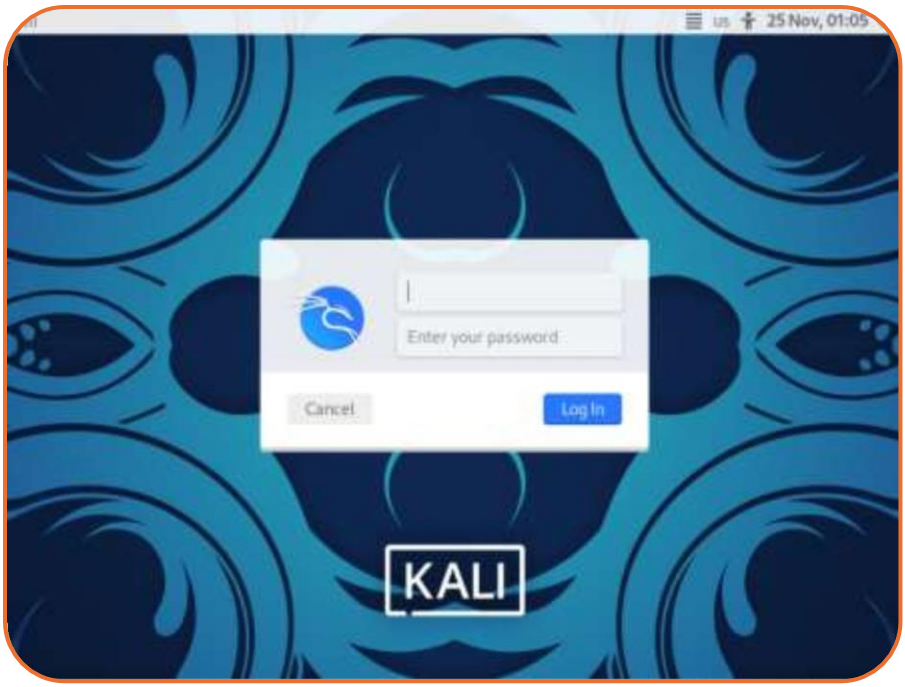
After installing the system's core, Kali enables users to customize the OS further. Choose the components to install by executing the following steps:

- Select the desktop environment and the tools you want, or click **Continue** to proceed with the default options.




- Select whether you want to use a network mirror.
- If you use an **HTTP proxy**, enter the necessary information. Otherwise, leave the field blank.
- Install **the GRUB bootloader** on the hard disk. Select **Yes** and **Continue**.
- Select a boot-loader device to ensure the newly installed system is bootable.


- When Kali finishes installing, the *Installation is complete* message appears.
- Click **Continue** to reboot your VM. After rebooting, the Kali login screen appears.
- Enter the username and password created in the previous steps.
- The Kali Linux desktop appears on the screen.



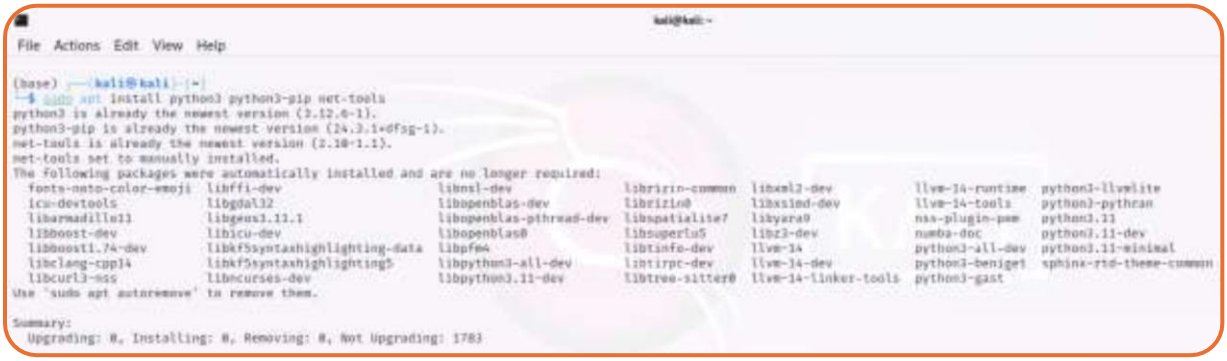


## 2. Setup Documentation

Once the virtual machine is installed, we will proceed to update it and then install the necessary tools and libraries to run the application.

- Updating apt repos.

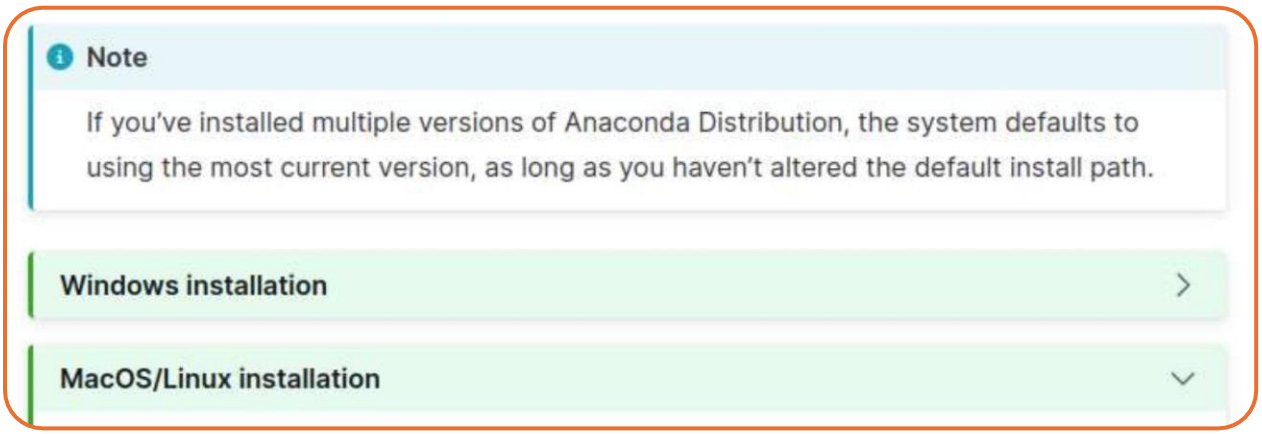

- Installing python3, pip3, and net-tools etc. with other necessary dependencies.




Anaconda is an open-source software that contains Jupyter, spyder, etc that are used for large data processing, data analytics, and heavy scientific computing. Anaconda works for R and python programming language. Spyder (sub-application of Anaconda) is used for python. Opencv for python will work in spyder. Package versions are managed by the package management system called conda.

To begin working with Anaconda, one must get it installed first. Follow the below instructions to Download and install Anaconda on your system:

- Visit this link and navigate to section https://docs.anaconda.com/anaconda/install/ **Installing Anaconda Distribution**
- Then follow steps provided for Linux Debian distribution as shown in figures below.



Debian | RedHat | ArchLinux | OpenSuse/SLES | Gentoo

```
dr2 libxrandr2 libxss1 libxcursor1 libxcomposite1 libasound2 libxi6 libxtst6
```

1. Download the latest version of Anaconda Distribution by opening a terminal and running one of the following commands (depending on your Linux architecture):

Linux x86 | AWS Graviton2/ARM64 | IBMZ/LinuxOne/s390x

```
-O https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh
```

**To install a different version** >

2. (Recommended) Verify the integrity of your installer to ensure that it was not corrupted or tampered with during download.

**How do I verify my installer's integrity?** >

3. Install Anaconda Distribution by running one of the following commands (depending on your Linux architecture):

Linux x86 | AWS Graviton2/ARM64 | IBMZ/LinuxOne/s390x

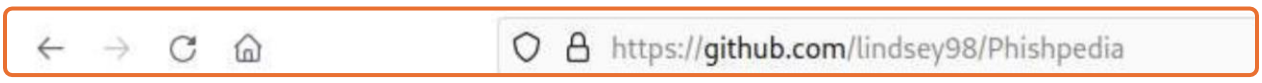
```
bash ~/Anaconda3-2024.10-1-Linux-x86_64.sh
```

4. Press return to review the license agreement. Then press and hold return to scroll.
5. Enter `yes` to agree to the license agreement.
6. Press return to accept the default install location (`PREFIX=/Users/<USER>/anaconda3`), or enter another file path to specify an alternate installation directory. The installation might take a few minutes to complete.
7. Choose an initialization options:
   - Yes - `conda` modifies your shell configuration to initialize conda whenever you open a new shell and to recognize conda commands automatically.
   - No - `conda` will not modify your shell scripts. After installation, if you want to initialize, you must do so manually. For more information, see Manual Shell Initialization.
8. The installer finishes and displays, "Thank you for installing Anaconda3!"
9. Close and re-open your terminal window for the installation to fully take effect, or use the following command to refresh the terminal:

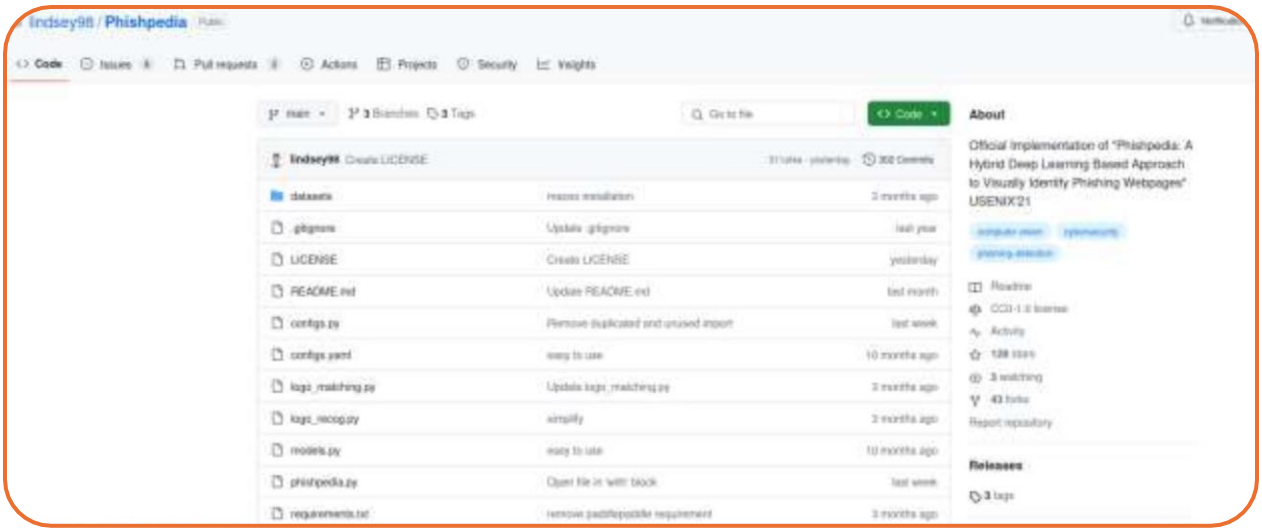
```
source ~/.bashrc
```

- Once the installation is complete, now verify the installation by running the command shown in the image below.


- This confirms that the Anaconda has been successfully installed, and we can now use it for further activities.
- Next, let's download Phishpedia and configure it.
- Open Firefox on Kali and search for Phishpedia.




- It will show GitHub repository link, open it




- Now clone the Phishpedia using this command

1. Create a local clone of Phishpedia

```
git clone https://github.com/lindsey98/Phishpedia.git
```

```
(base) ┌──(kali㉿kali)-[~]
└─$ git clone https://github.com/lindsey98/Phishpedia.git
Cloning into 'Phishpedia'...
remote: Enumerating objects: 1418, done.
remote: Counting objects: 100% (525/525), done.
remote: Compressing objects: 100% (297/297), done.
remote: Total 1418 (delta 298), reused 363 (delta 221), pack-reused 893 (from 1)
Receiving objects: 100% (1418/1418), 4.06 MiB | 2.09 MiB/s, done.
Resolving deltas: 100% (834/834), done.
```

## 2.1. Prerequisites

```
scipy
tldextract
opencv-python
pandas
numpy
tqdm
Pillow==8.4.0
pathlib
fvcore
pycocotools
scikit-learn
lxml
gdown
memory-profiler
psutil
```

- Setup the Phishpedia conda environment.
  - In this step, we would be installing the core dependencies of Phishpedia such as pytorch, and detectron.
  - In addition, we would also download the model checkpoints and brand reference list. This step may take some time.

```
(base) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ chmod +x ./setup.sh
export ENV_NAME="phishpedia"
./setup.sh
Creating new Conda environment: phishpedia with Python 3.8
Channels:
 - defaults
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: /home/kali/anaconda3/envs/phishpedia

  added / updated specs:
    - python=3.8
```

```
The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    pip-24.2                   |   py38h06a4308_0         2.2 MB
    python-3.8.20              |       he870216_0        23.8 MB
    setuptools-75.1.0          |   py38h06a4308_0         1.7 MB
    wheel-0.44.0               |   py38h06a4308_0         108 KB
    ------------------------------------------------------------
                                           Total:        27.8 MB

The following NEW packages will be INSTALLED:

  _libgcc_mutex      pkgs/main/linux-64::_libgcc_mutex-0.1-main
  _openmp_mutex      pkgs/main/linux-64::_openmp_mutex-5.1-1_gnu
  ca-certificates    pkgs/main/linux-64::ca-certificates-2024.9.24-h06a4308_0
  ld_impl_linux-64   pkgs/main/linux-64::ld_impl_linux-64-2.40-h12ee557_0
  libffi             pkgs/main/linux-64::libffi-3.4.4-h6a678d5_1
  libgcc-ng          pkgs/main/linux-64::libgcc-ng-11.2.0-h1234567_1
  libgomp            pkgs/main/linux-64::libgomp-11.2.0-h1234567_1
  libstdcxx-ng       pkgs/main/linux-64::libstdcxx-ng-11.2.0-h1234567_1
  ncurses            pkgs/main/linux-64::ncurses-6.4-h6a678d5_0
  openssl            pkgs/main/linux-64::openssl-3.0.15-h5eee18b_0
  pip                pkgs/main/linux-64::pip-24.2-py38h06a4308_0
  python             pkgs/main/linux-64::python-3.8.20-he870216_0
  readline           pkgs/main/linux-64::readline-8.2-h5eee18b_0
  setuptools         pkgs/main/linux-64::setuptools-75.1.0-py38h06a4308_0
  sqlite             pkgs/main/linux-64::sqlite-3.45.3-h5eee18b_0
  tk                 pkgs/main/linux-64::tk-8.6.14-h39e8969_0
```

- It will ask to select an option, select “A”

```
resnetv2_rgb_new.pth.tar already exists. Skipping download.
expand_targetlist.zip already exists. Skipping download.
faster_rcnn.yaml already exists. Skipping download.
rcnn_bet365.pth already exists. Skipping download.
domain_map.pkl already exists. Skipping download.
Archive:  expand_targetlist.zip
replace expand_targetlist/expand_targetlist/.DS_Store? [y]es, [n]o, [A]ll, [N]one, [r]ename: A
```

```
 95%|          | 313M/330M [02:08<00:07, 2.34MB/s]
 95%|          | 314M/330M [02:09<00:07, 2.30MB/s]
 95%|          | 315M/330M [02:09<00:05, 2.78MB/s]
 95%|          | 315M/330M [02:09<00:04, 3.03MB/s]
 96%|          | 316M/330M [02:09<00:05, 2.65MB/s]
 96%|          | 316M/330M [02:09<00:05, 2.76MB/s]
 96%|          | 317M/330M [02:10<00:04, 2.67MB/s]
 96%|          | 317M/330M [02:10<00:05, 2.54MB/s]
 96%|          | 318M/330M [02:10<00:05, 2.41MB/s]
 96%|          | 318M/330M [02:10<00:04, 2.40MB/s]
 97%|          | 319M/330M [02:10<00:04, 2.41MB/s]
 97%|          | 319M/330M [02:11<00:04, 2.41MB/s]
 97%|          | 320M/330M [02:11<00:04, 2.37MB/s]
 97%|          | 320M/330M [02:11<00:04, 2.35MB/s]
 97%|          | 321M/330M [02:11<00:03, 2.30MB/s]
 97%|          | 321M/330M [02:11<00:03, 2.75MB/s]
 98%|          | 322M/330M [02:12<00:03, 2.44MB/s]
 98%|          | 322M/330M [02:12<00:02, 2.90MB/s]
 98%|          | 323M/330M [02:12<00:02, 2.89MB/s]
 98%|          | 323M/330M [02:12<00:02, 2.68MB/s]
 98%|          | 324M/330M [02:13<00:02, 2.38MB/s]
 98%|          | 325M/330M [02:13<00:02, 2.15MB/s]
 98%|          | 325M/330M [02:13<00:02, 2.01MB/s]
 99%|          | 326M/330M [02:14<00:02, 1.63MB/s]
 99%|          | 327M/330M [02:14<00:01, 2.45MB/s]
 99%|          | 327M/330M [02:14<00:01, 2.40MB/s]
 99%|          | 328M/330M [02:14<00:00, 2.68MB/s]
 99%|          | 328M/330M [02:14<00:00, 2.27MB/s]
100%|          | 329M/330M [02:15<00:00, 2.41MB/s]
100%|          | 330M/330M [02:15<00:00, 2.71MB/s]
100%|          | 330M/330M [02:15<00:00, 2.44MB/s]

Attempting to download domain_map.pkl (Attempt 1/3)...
/home/kali/anaconda3/envs/phishpedia/lib/python3.8/site-packages/gdown/__m
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=1qSdkSSoCYUkZMKs44Rup_1DPBxHnEKl1
To: /home/kali/Desktop/PhishPedia/Phishpedia/models/domain_map.pkl

  0%|          | 0.00/213k [00:00<?, ?B/s]
100%|          | 213k/213k [00:00<00:00, 776kB/s]
100%|          | 213k/213k [00:00<00:00, 776kB/s]
```

- Once installed then we will activate the virtual environment

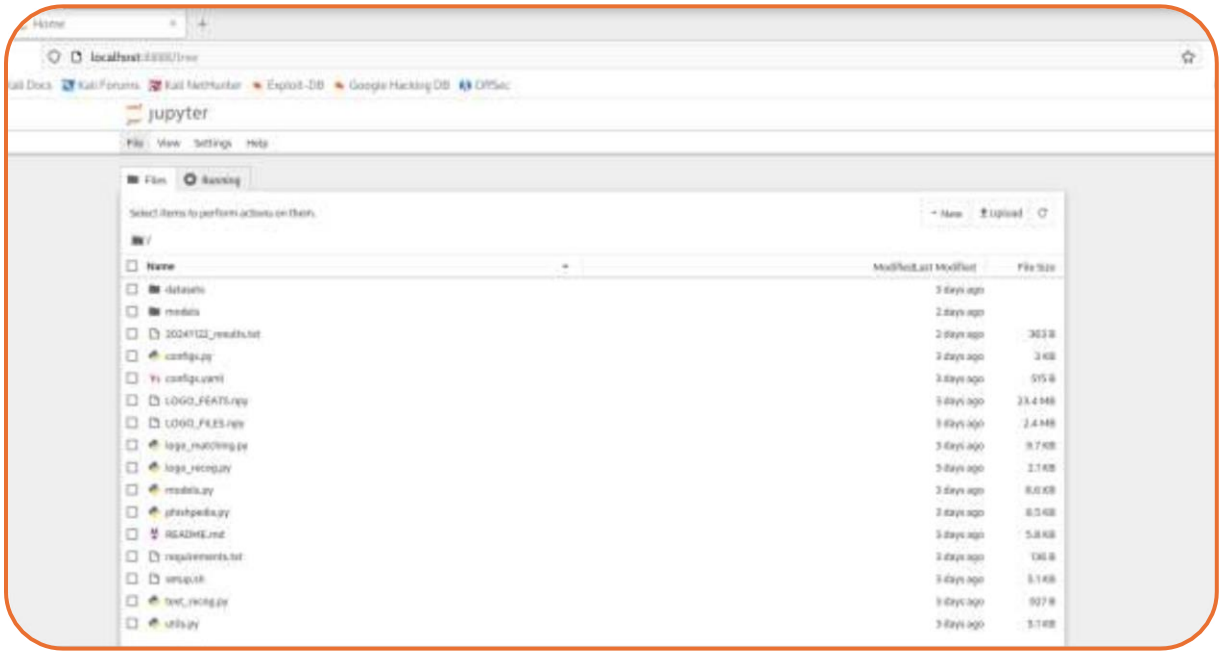
```
(base) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ conda activate phishpedia

(phishpedia) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$
```

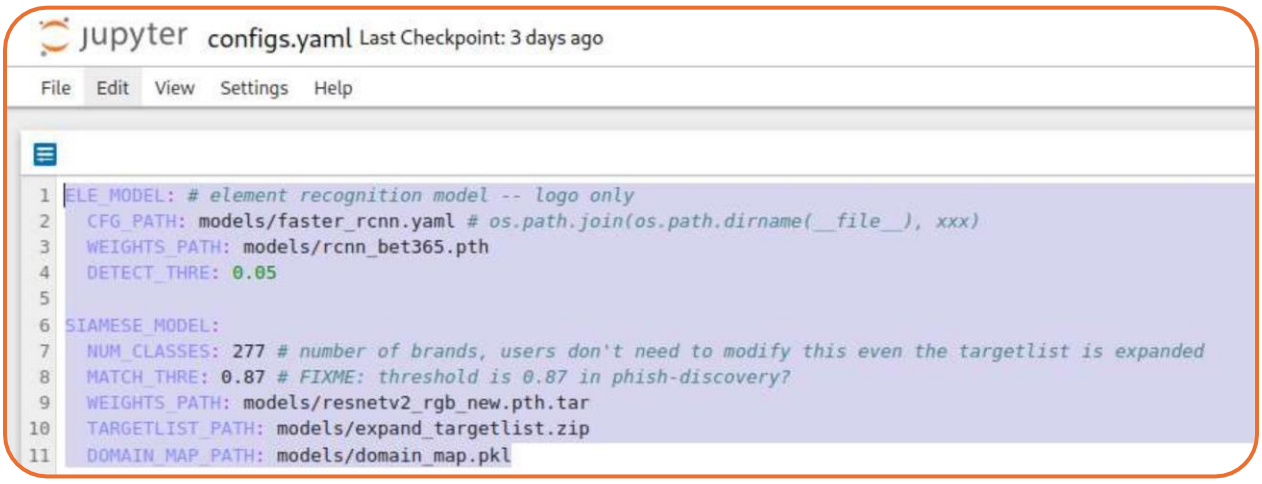

- Once installed and configured, we can use the tool for making predictions.
- To better understand its functionality, we will use real-world examples from PhishTank, a large database of reported phishing URLs.
- We will pass various links and details from PhishTank to the tool and verify the outcomes.

## 2.2. Understanding the code

After navigating to the folder in terminal we type the command “jupyter notebook” and it will open an interactive window in the browser where we can see all our files of the tool



- Phishpedia is a phishing identification system that utilizes deep learning to visually detect phishing webpages. The repository contains several key files and directories:
    - **README.md**: Provides an overview of Phishpedia, including its purpose, features, and instructions for setup and usage.
    - **phishpedia.py**: The main script that orchestrates the phishing detection process by integrating various components.
    - **logo_recog.py**: Contains functions for detecting logos on webpages, a crucial step in identifying potential phishing sites.
    - **logo_matching.py**: Implements the logic for matching detected logos against a reference set to determine the targeted brand.
    - **text_recog.py**: Handles text recognition tasks, such as identifying credential input fields, to assess phishing intent.
    - **models.py**: Defines the deep learning models used for logo detection and matching.
    - **configs.yaml**: A configuration file specifying parameters and settings for the system's operation.
    - **requirements.txt**: Lists the Python dependencies required to run Phishpedia.
    - **setup.sh**: A shell script to automate the setup process, including environment preparation and dependency installation.
    - **datasets/**: A directory intended to store datasets used for training and testing the system.
    - **expand_targetlist/**: Contains reference images and data for various brands, aiding in the logo matching process.
- These components collectively enable Phishpedia to detect phishing websites by analyzing visual elements and comparing them to known brand assets.
- For example we can open "*configs.yaml*" file by clicking on it and it will open in a new tab.




- *ELE_MODEL: Element Recognition Model (Logo Detection)*
    - **CFG_PATH**: Specifies the path to the configuration file (faster_rcnn.yaml) for the Faster R-CNN model used to recognize elements, specifically logos. This helps to set up the structure of the model.

- **WEIGHTS_PATH**: Indicates the path to the pre-trained model weights (rcnn_bet365.pth). These weights allow the model to recognize logos accurately based on prior training.
- **DETECT_THRE**: Detection threshold value (0.05). This controls the minimum confidence level required for a logo to be considered detected. Lower values make the model more sensitive to detecting logos, but it may result in more false positives.

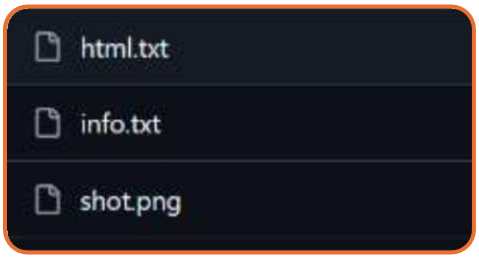
- *SIAMESE_MODEL: Brand Logo Matching Model*
  - **NUM_CLASSES**: Specifies the number of brand classes (277). This represents the number of different brands the model can recognize. Users are advised not to change this even if new targets are added to the list.
  - **MATCH_THRE**: Matching threshold value (0.87). This is the threshold for determining whether a detected logo matches a known brand. A higher value ensures stricter matching to reduce false positives.
  - **WEIGHTS_PATH**: Path to the pre-trained weights for the Siamese network (resnetv2_rgb_new.pth.tar). This model is used for comparing logos to reference images.
  - **TARGETLIST_PATH**: Path to the zip file containing the reference logos (expand_targetlist.zip). This helps in matching detected logos with known brand assets.
  - **DOMAIN_MAP_PATH**: Path to a domain-brand mapping file (domain_map.pkl). This helps in linking detected logos to their respective brand domains for better identification.
- In summary, the *configs.yaml* file provides paths to important models, weights, and reference data, as well as critical threshold parameters to fine-tune the detection and matching processes in Phishpedia. This helps control how logos are detected, matched, and associated with potential phishing attempts.
- Similarly, all other files can be viewed in this way.

## 2.3. Running the Tool

- To run the tool, we need to use Python (which we installed earlier) and the phishpedia.py file. Then, provide the path to the folder containing the subfolders that we want to test.

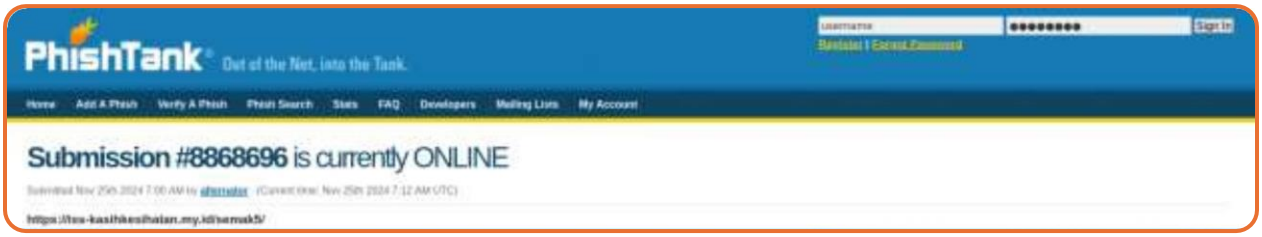
```
python phishpedia.py --folder <folder you want to test e.g. ./datasets/test_sites>
```

- In the folder, each sub-folder must contain at least three items that the model requires as input:
  1. html.txt: Contains the HTML code of the phishing page.
  2. info.txt: Contains the URL of the phishing website.
  3. shot.png: A screenshot of the phishing site.



### 2.3.1. Test 01

- Let's go to PhishTank and find some malicious websites reported as phishing. Here is an example where we can see that it has been reported as phishing, and the prediction score is 100%.

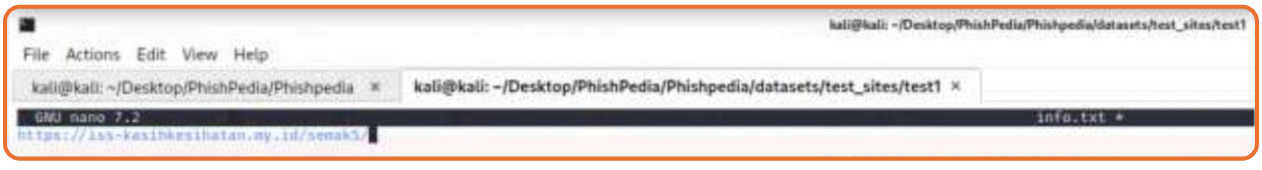




- Open in new tab

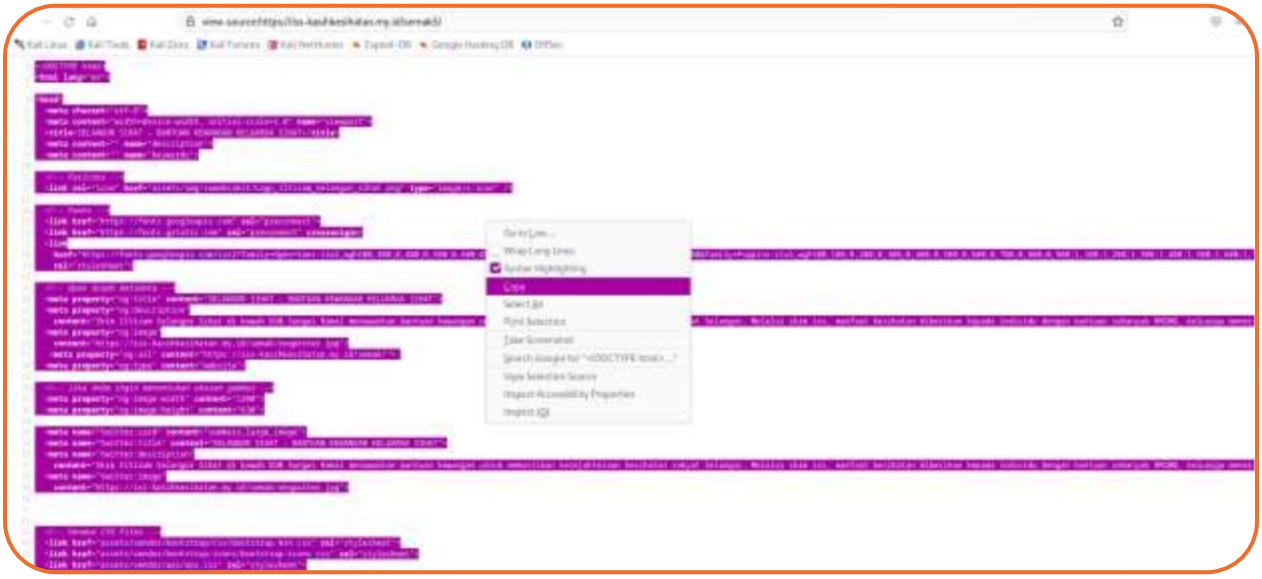

- Create file named “info.txt” and add link to it

```
(phishpedia) ┌──(kali㉿kali)-[~/…/PhishPedia/Phishpedia/datasets/test_sites]
└─$ nano info.txt
```




- Take a screenshot of the webpage and add it to the test1 folder.
- Next, create a file named html.txt and paste the page source into it.

- Right-click on the webpage, select "View Page Source," and copy the content into the file.

- Now, we have all three required files

```
(phishpedia) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test1]
└─$ ls
html.txt  info.txt  snap.png
```

- Now, let's run the tool

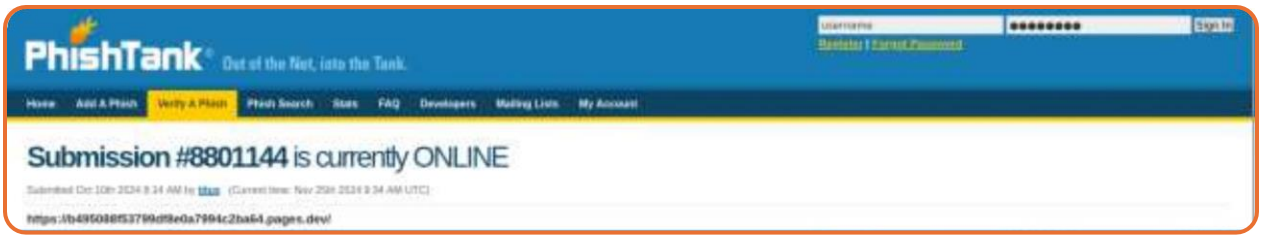
```
(phishpedia) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ python3 phishpedia.py --folder ./datasets/test_sites
```

- Reported as 100% phishing

- **Prediction: True**

### 2.3.2. Test 02

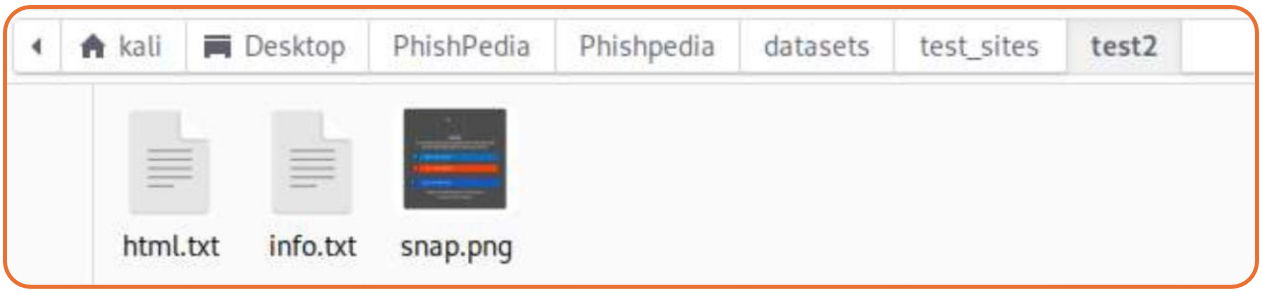
- Let's now search for top brands





- Open in new tab


```
(base) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test2]
└─$ nano info.txt
```



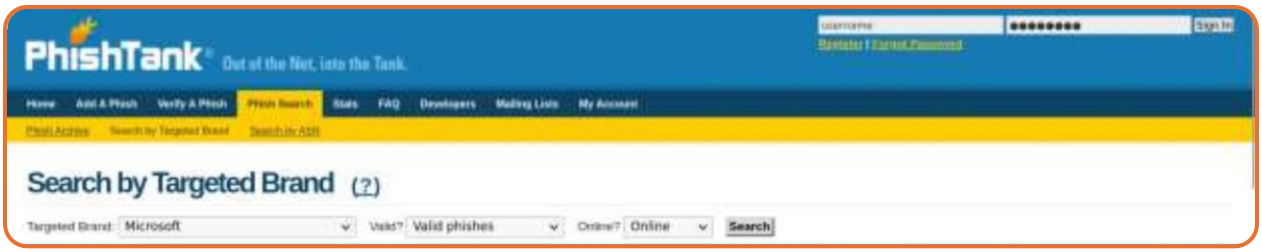

- Now, let's run the tool

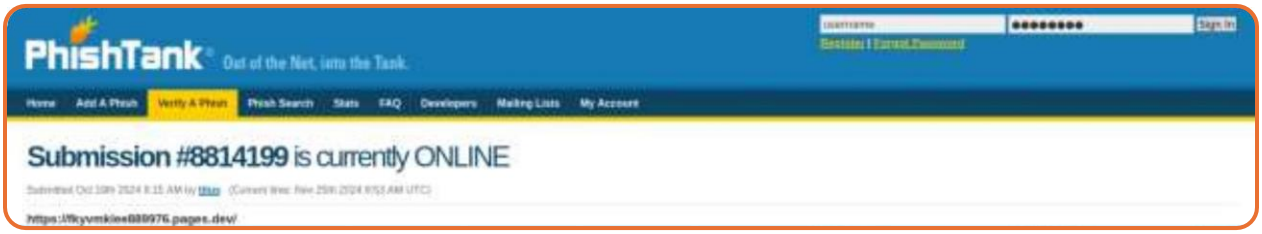
```
(phishpedia) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ python3 phishpedia.py --folder ./datasets/test_sites
```

- Reported as 100% phishing

- **Prediction: True**

### 2.3.3. Test 03

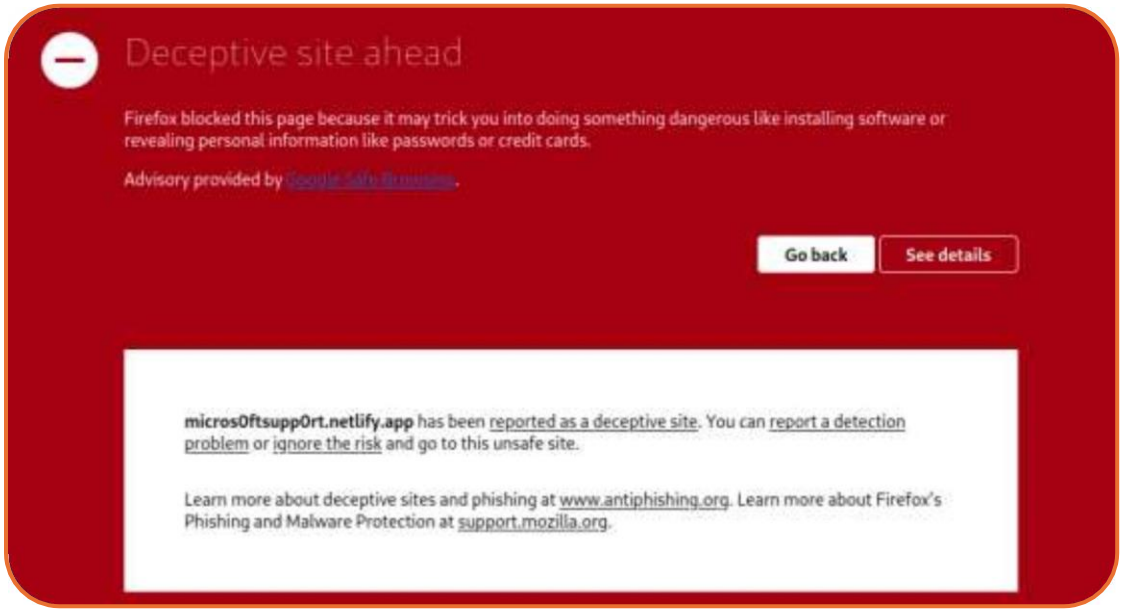


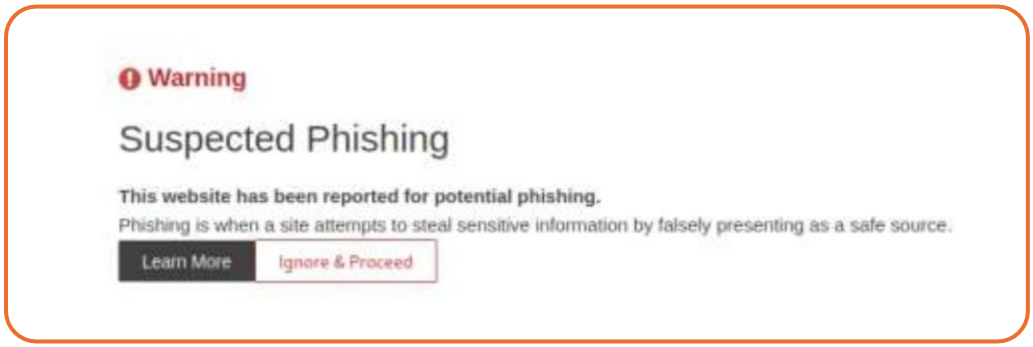


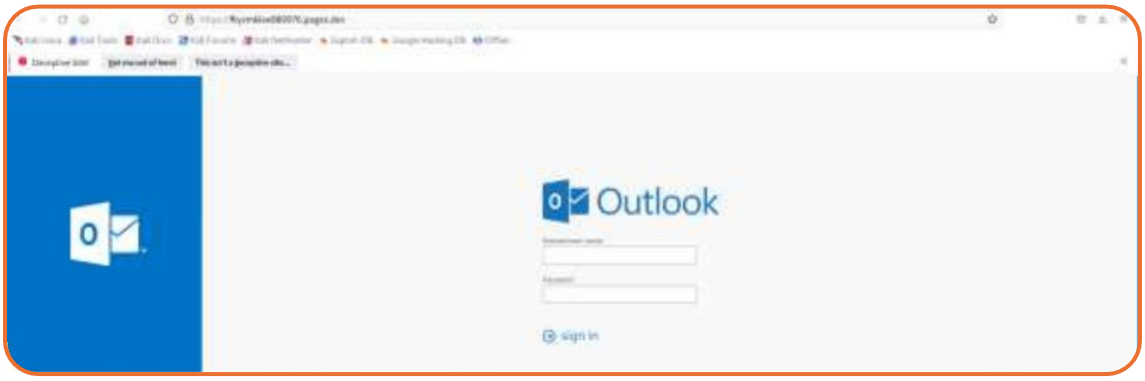
- Open in new tab


- It says a deceptive, let's proceed ahead. Click See Details and Ignore Risk option




- Click on Ignore & Proceed



```
(phishpedia) ┌──(kali㉿kali)-[~/…/PhishPedia/Phishpedia/datasets/test_sites]
└─$ nano info.txt
```


```
(base) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test2]
└─$ nano html.txt
```

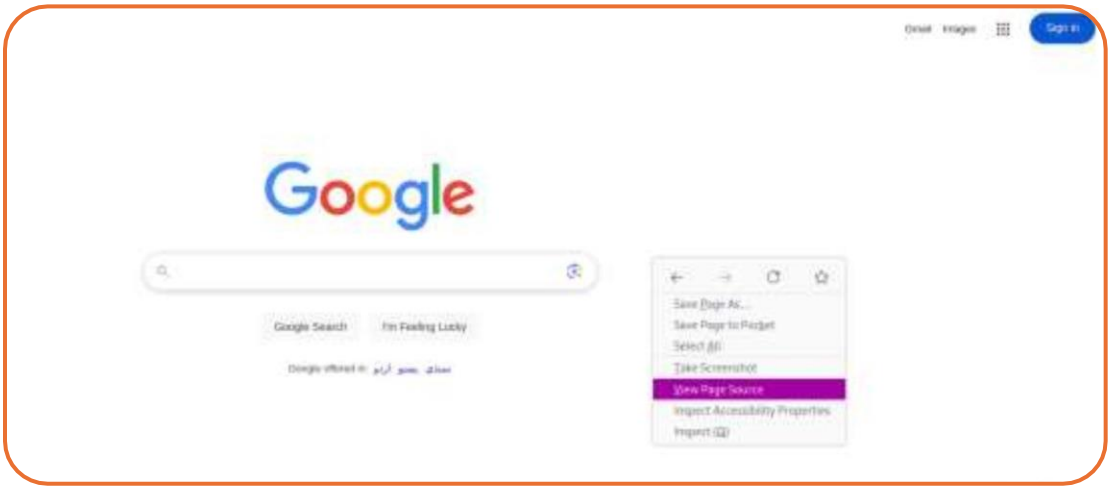
```
(base) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test2]
└─$ ls
html.txt  info.txt  snap.png
```

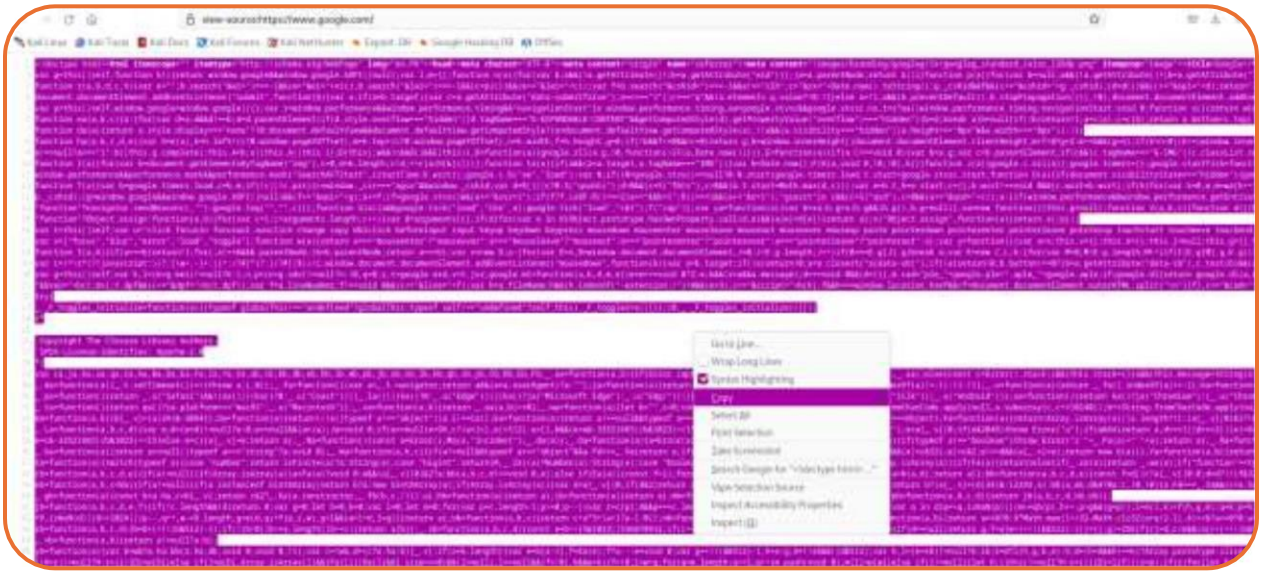
- Now let's run the tool
- Reported as 100% phishing

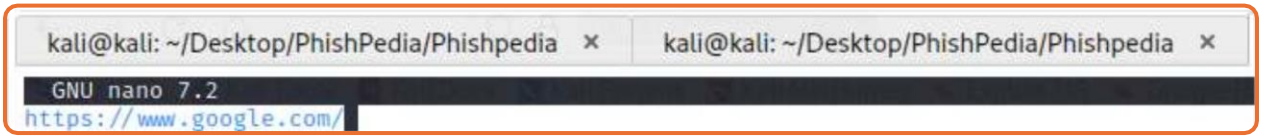
- **Prediction: True**

### 2.3.4. Test 04 {Legitimate Site}

#### *2.3.4.1. Site: Google.com*



```
(base) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test2]
└─$ ls
html.txt  info.txt  snap.png
```

- Let's run the tool now
- Reported as 20% phishing

```
(phishpedia) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ python3 phishpedia.py --folder ./datasets/test_sites
The checkpoint state_dict contains keys that are not used by the model:
  pixel_mean
  pixel_std
Load protected logo list
Length of reference list = 2996
 20%|
```

- **Prediction: True**

## GitHub Repository Citation:

Lindsey98. *Phishpedia: Phishing Detection Tool Using Deep Learning.* GitHub. Available at: https://github.com/lindsey98/Phishpedia.