King Fahd University of Petroleum and Mineral
The Department of Information and Computer Science

Experimentation and Analysis of Present Solutions: Phishpedia

Experimental Report for Phishapedia

Presented by
Fatimah Almusaid G202306890
Atika Alnaim G202306850
Rawan Alali G202307110

Supervised by
Dr. Waleed AlGobi

December 3, 2024

Agenda

Introduction

What is Phishpedia?

- A phishing detection tool using deep learning to analyze logos and webpage features.
- Protects users by identifying phishing websites based on visual elements.

Objective of Experimentation:

Test its performance and analyze effectiveness.

Environment Setup

Tools:

- Kali Linux on VirtualBox.
- Dependencies: Python3, pip3, net-tools, and Anaconda.

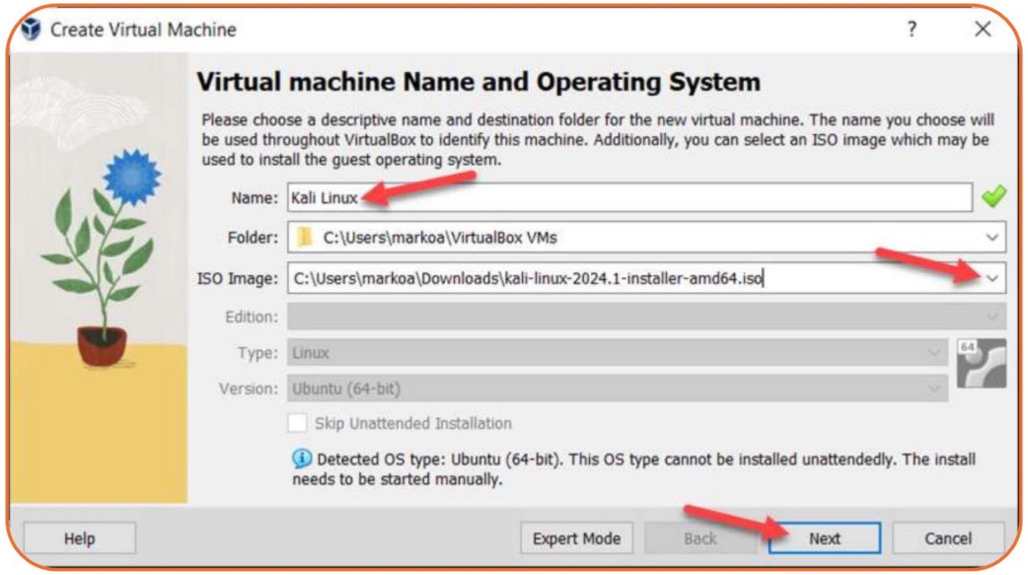
Installation Steps:

1. Download and install Kali Linux ISO.
2. Configure VirtualBox settings.
3. Install necessary dependencies.

Why This Setup?

To create a controlled environment for experimentation.

Environment Setup


Create Virtual Machine

Virtual machine Name and Operating System

Please choose a descriptive name and destination folder for the new virtual machine. The name you choose will be used throughout VirtualBox to identify this machine. Additionally, you can select an ISO image which may be used to install the guest operating system.

Name: Kali Linux

Folder: C:\Users\markoa\VirtualBox VMs

ISO Image: C:\Users\markoa\Downloads\kali-linux-2024.1-installer-amd64.iso

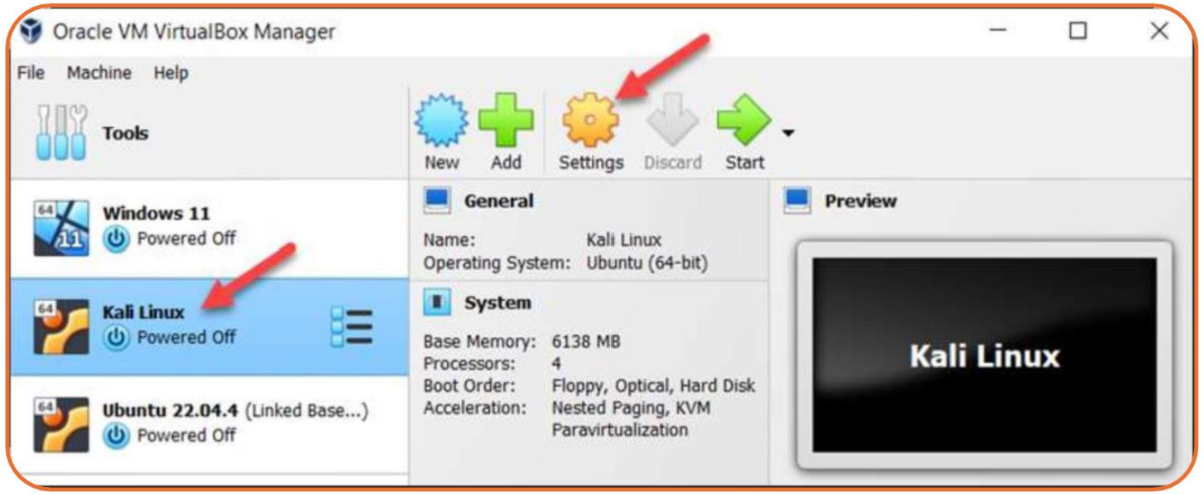
Edition:

Type: Linux

Version: Ubuntu (64-bit)

Skip Unattended Installation

Detected OS type: Ubuntu (64-bit). This OS type cannot be installed unattendedly. The install needs to be started manually.


Oracle VM VirtualBox Manager

Windows 11 Powered Off

Kali Linux Powered Off

Ubuntu 22.04.4 (Linked Base...) Powered Off

General

Name: Kali Linux
Operating System: Ubuntu (64-bit)

System

Base Memory: 6138 MB
Processors: 4
Boot Order: Floppy, Optical, Hard Disk
Acceleration: Nested Paging, KVM Paravirtualization

Preview

Phishpedia Overview

Key Features:

1. Logo Detection and Comparison:

a) Phishpedia identifies logos on websites and matches them to known logos from a reference dataset. This helps determine if a website is impersonating a legitimate brand.

2. Uses Faster-RCNN for Logo Detection:

a) Faster-RCNN (Faster Region-Based Convolutional Neural Network) is a deep learning model used for object detection.
b) It scans the webpage screenshot and identifies areas where logos are likely present.

3. Siamese Network for Brand Logo Matching:

a) After a logo is detected, the Siamese network compares it with known brand logos to see if it matches any legitimate brands.
b) A Siamese network works by analyzing the similarity between two inputs, making it suitable for identifying subtle differences in logos.

Phishpedia Overview

Core Components:

Scripts for Logo Recognition and Matching:

logo_recog.py: This script is responsible for detecting logos on a webpage. It uses the Faster-RCNN model to locate logos in screenshots.
logo_matching.py: Once a logo is detected, this script compares it with logos in the reference dataset using the Siamese network.

Configuration File (*configs.yaml*):

- This file defines important parameters for the models, such as:
    - The detection threshold (minimum confidence required for logo detection).
    - The matching threshold (confidence level required to declare a match).
- Fine-tuning these thresholds improves the accuracy of detection and reduces false positives.

Traditional Farmwork vs. Phishpedia

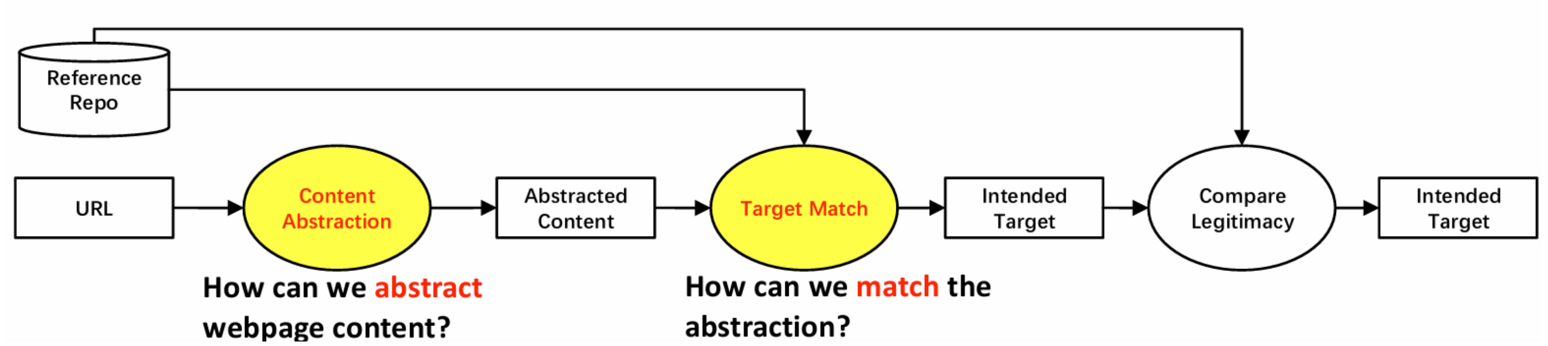


Fig1: Typical Framework for Phishing Identification

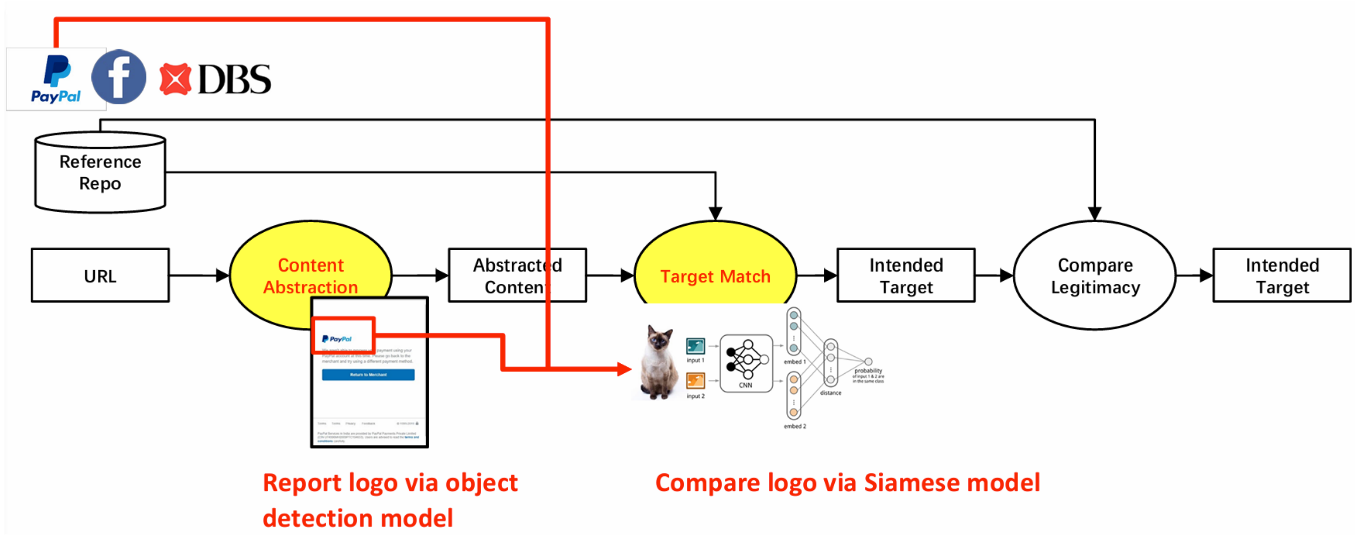
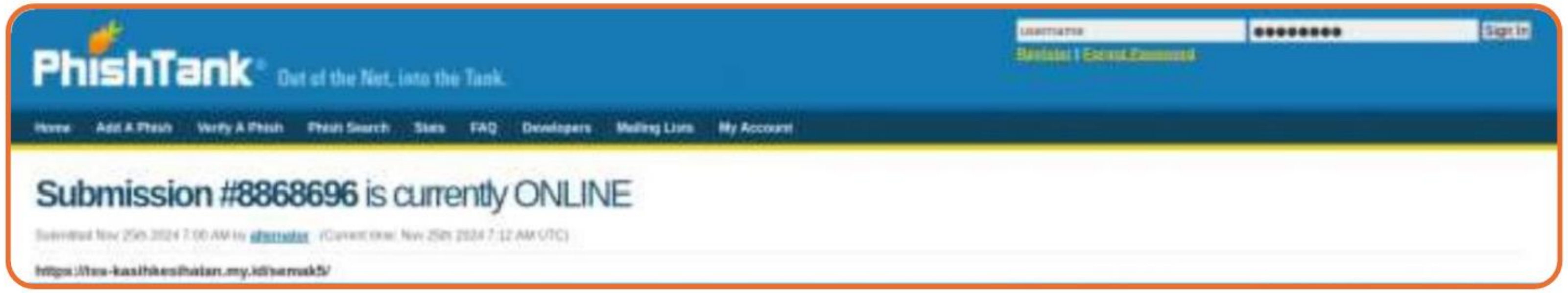


Fig2: Phishapedia Approach Overview

Example of Phishing Site

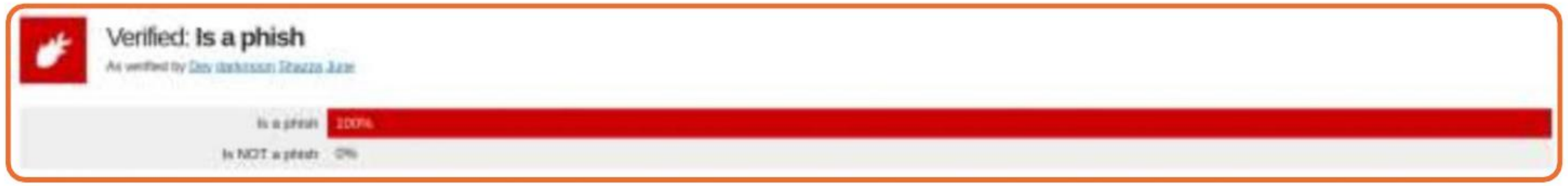
- Let's go to PhishTank and find some malicious websites reported as phishing. Here is an example where we can see that it has been reported as phishing, and the prediction score is 100%.



Submission #8868696 is currently ONLINE

Submitted Nov 25th 2024 7:00 AM by [illegible] (Current time: Nov 25th 2024 7:12 AM UTC)

https://tss-kasihkesihatan.my.id/semak5/



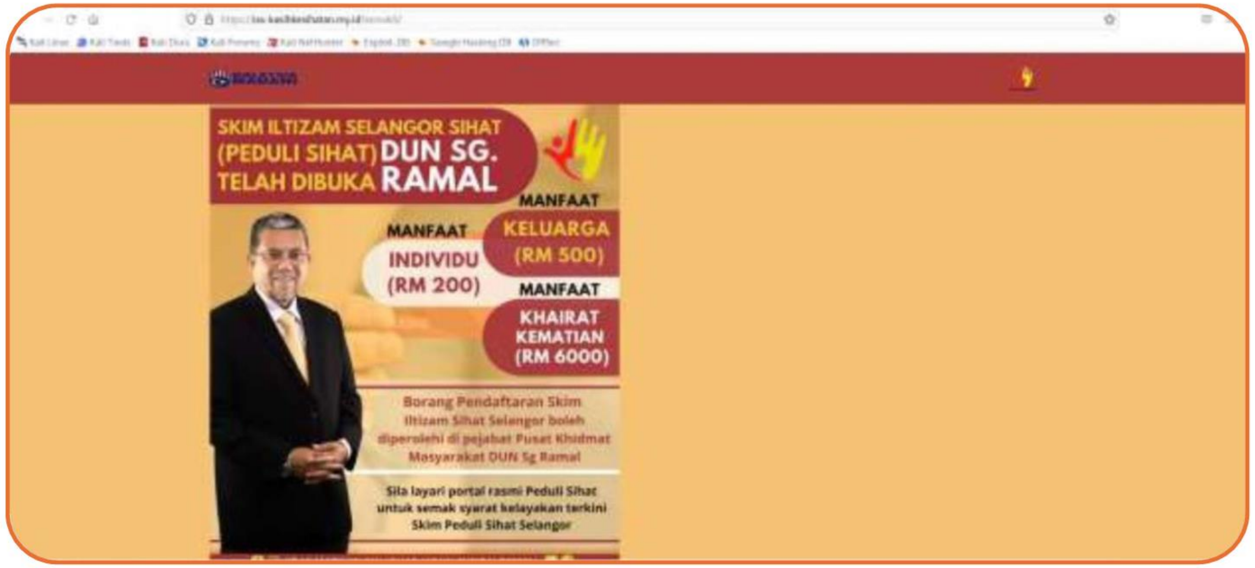
Verified: **Is a phish**

As verified by [illegible]

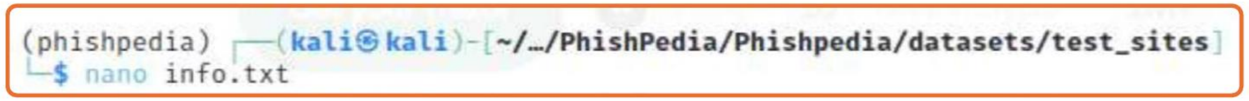
Is a phish	100%
Is NOT a phish	0%

Example of Phishing Site

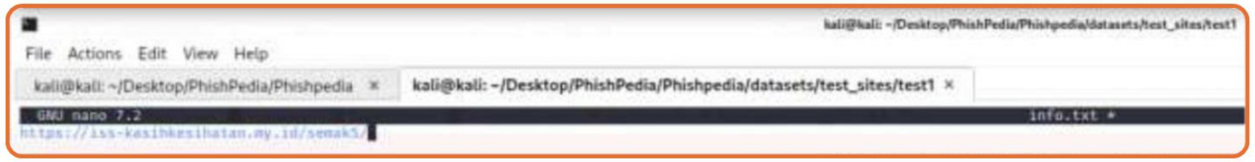
- Open in new tab


SKIM ILTIZAM SELANGOR SIHAT (PEDULI SIHAT) TELAH DIBUKA
DUN SG. RAMAL
MANFAAT INDIVIDU (RM 200)
MANFAAT KELUARGA (RM 500)
MANFAAT KHAIRAT KEMATIAN (RM 6000)
Borang Pendaftaran Skim Iltizam Sihat Selangor boleh diperolehi di pejabat Pusat Khidmat Masyarakat DUN Sg Ramal
Sila layari portal rasmi Peduli Sihat untuk semak syarat kelayakan terkini Skim Peduli Sihat Selangor

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



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



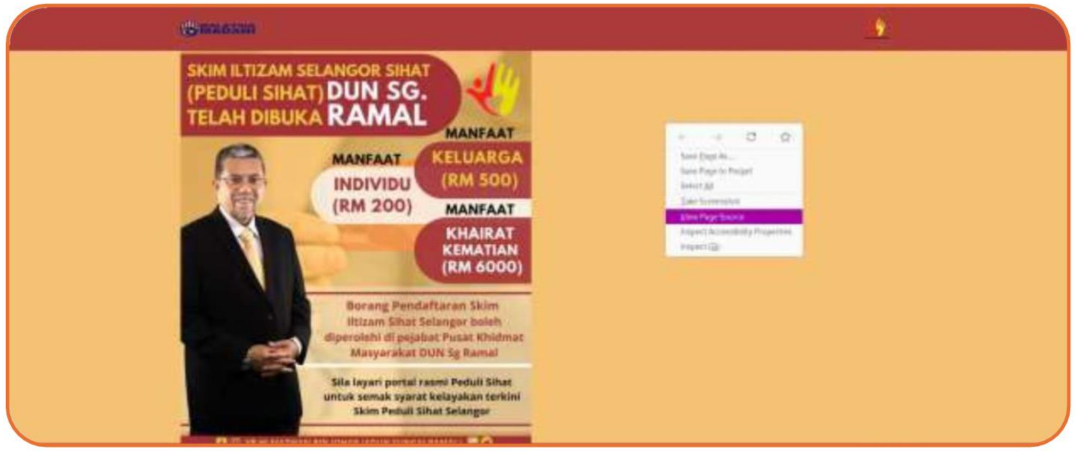
kali@kali: ~/Desktop/PhishPedia/Phishpedia/datasets/test_sites/test1

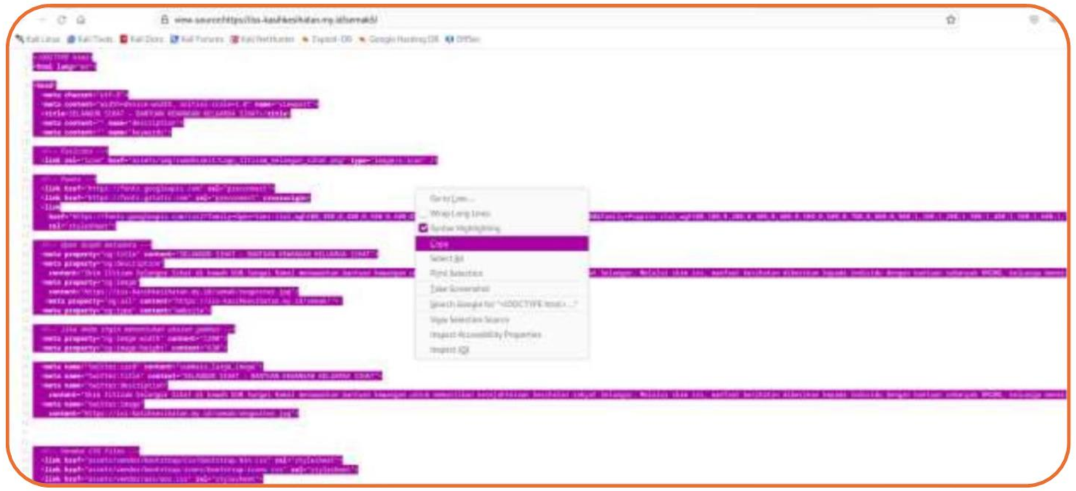
File Actions Edit View Help
kali@kali: ~/Desktop/PhishPedia/Phishpedia
kali@kali: ~/Desktop/PhishPedia/Phishpedia/datasets/test_sites/test1

```
 GNU nano 7.2                                                    info.txt *
https://iss-kasihkesihatan.my.id/semak5/
```

Example of Phishing Site

- 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.

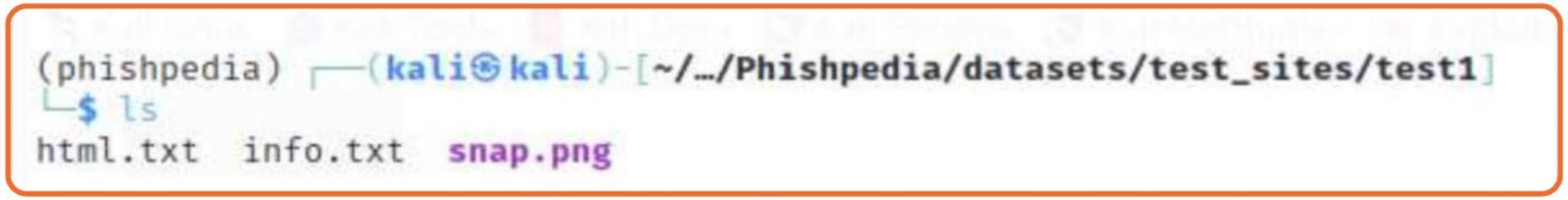

SKIM ILTIZAM SELANGOR SIHAT (PEDULI SIHAT) DUN SG. RAMAL TELAH DIBUKA
MANFAAT INDIVIDU (RM 200)
MANFAAT KELUARGA (RM 500)
MANFAAT KHAIRAT KEMATIAN (RM 6000)
Borang Pendaftaran Skim Iltizam Sihat Selangor boleh diperolehi di pejabat Pusat Khidmat Masyarakat DUN Sg Ramal
Sila layari portal rasmi Peduli Sihat untuk semak syarat kelayakan terkini Skim Peduli Sihat Selangor
View Page Source



```
[illegible]
```

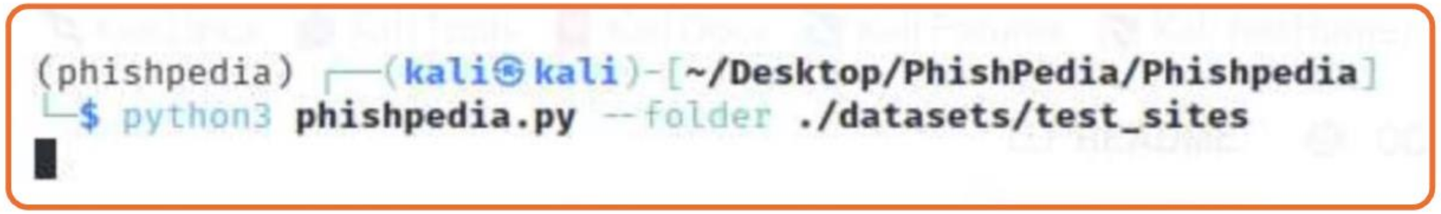
Example of Phishing Site

- Now, we have all three required files



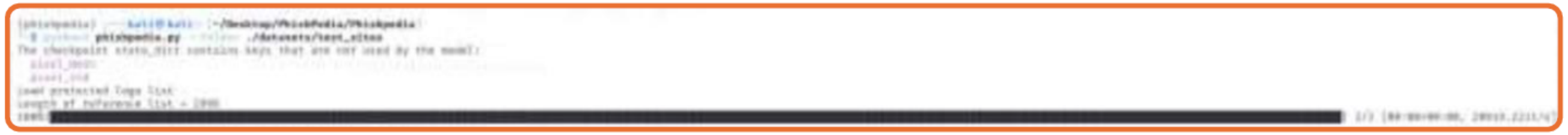
```
(phishpedia) ┌──(kali㉿kali)-[~/…/Phishpedia/datasets/test_sites/test1]
└─$ ls
html.txt   info.txt   snap.png
```

- Now, let’s run the tool



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

- Reported as 100% phishing

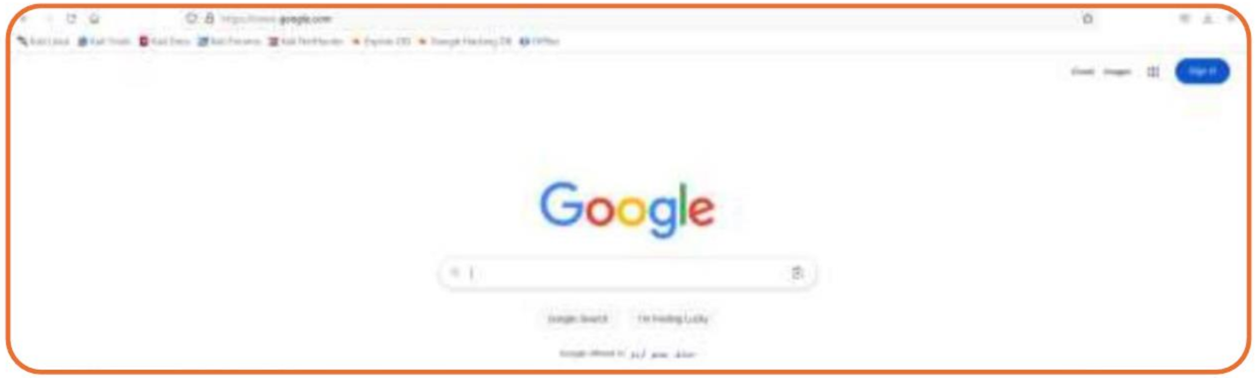


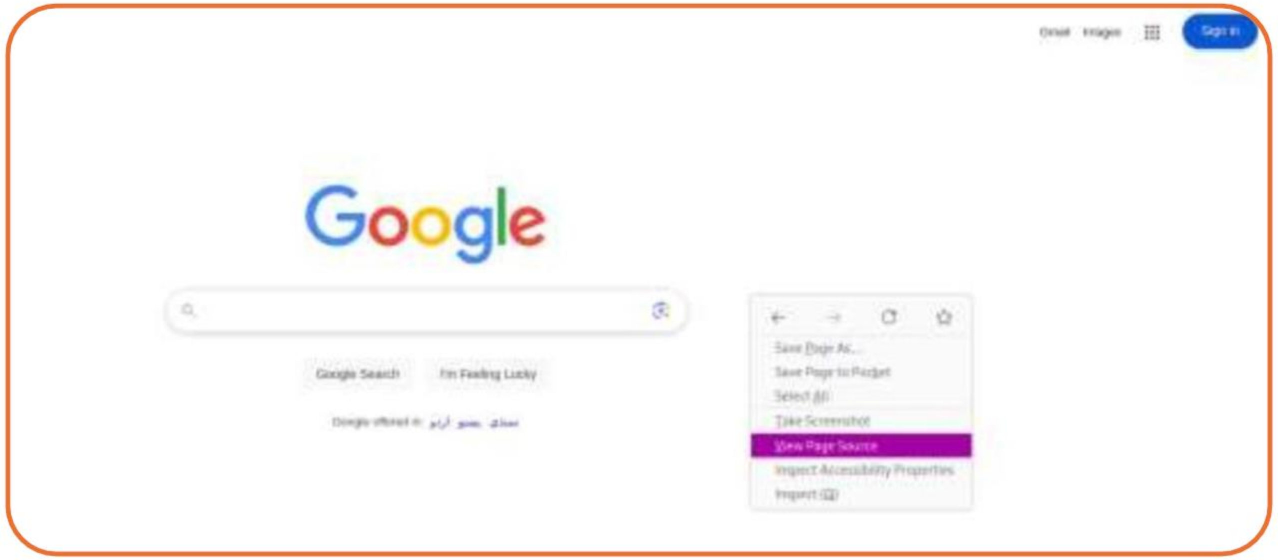
```
(phishpedia) ┌──(kali㉿kali)-[~/Desktop/PhishPedia/Phishpedia]
└─$ python3 phishpedia.py --folder ./datasets/test_sites
The checkpoint state_dict contains keys that are not used by the model:
  pixel_mean
  pixel_std
Load protected logo list
Length of reference list = 2000
100%|████████████████████████████████████████████████████████████| 1/1 [00:00<00:00, [illegible]]
```

- **Prediction: True**

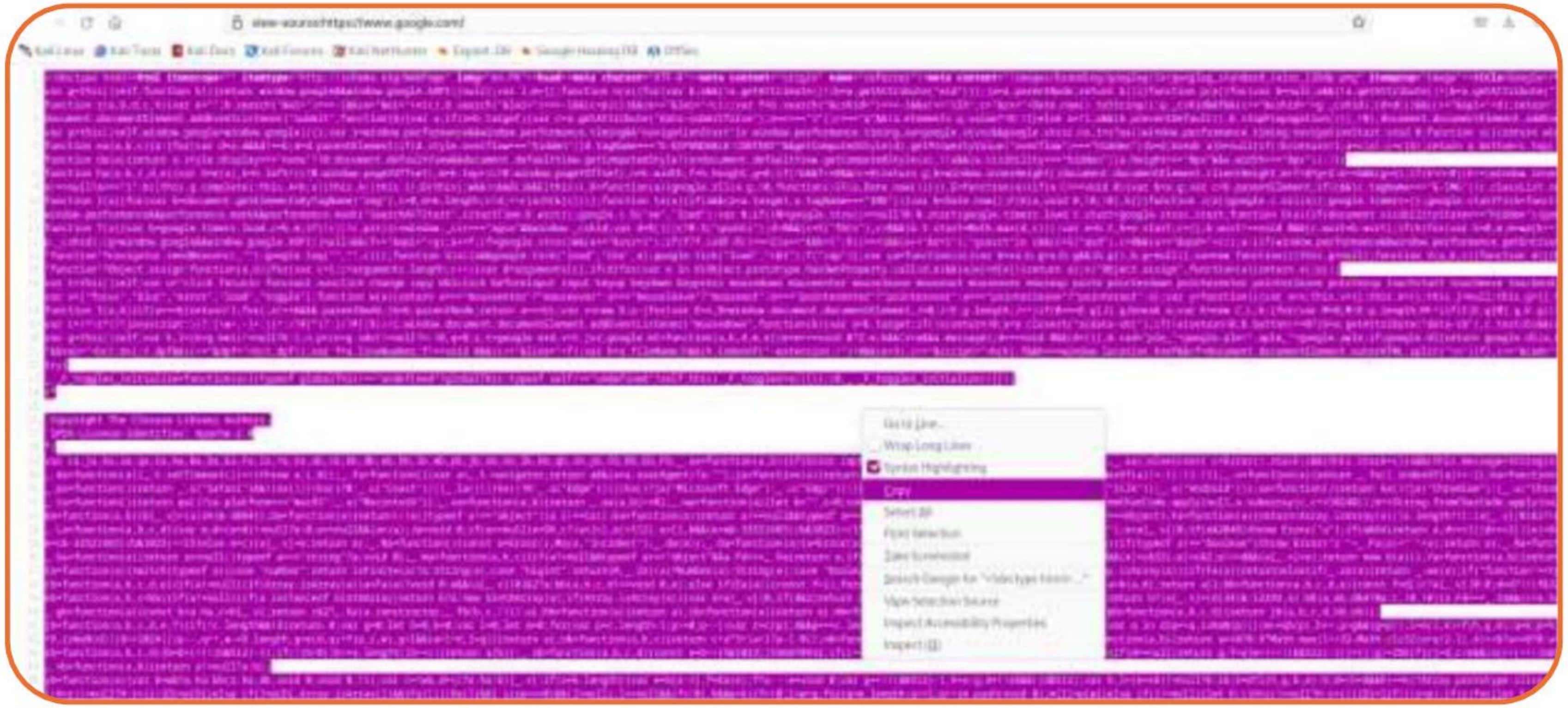
Example of Legitimate Site

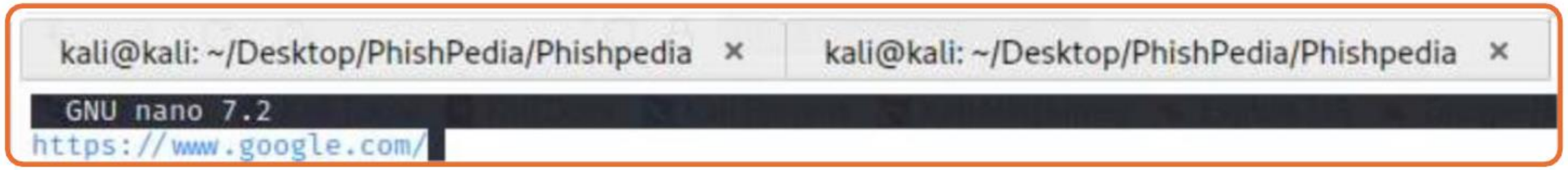
2.3.4.1. Site: Google.com





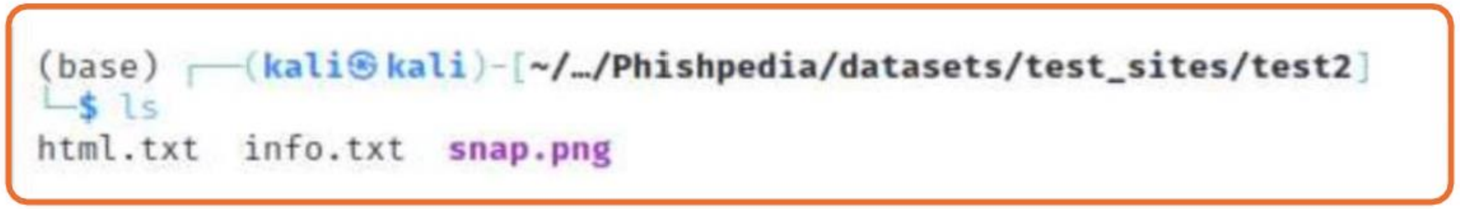
Example of Legitimate Site





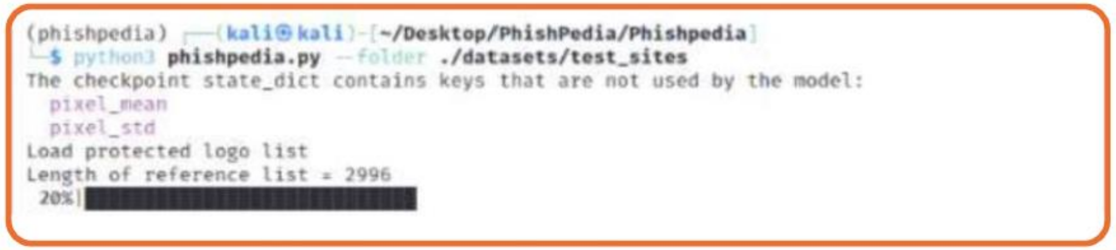
```
kali@kali: ~/Desktop/PhishPedia/Phishpedia    ×    kali@kali: ~/Desktop/PhishPedia/Phishpedia    ×
  GNU nano 7.2
https://www.google.com/
```

Example of Legitimate Site



```
(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**

Challenges and Limitations

- **Accuracy Concerns**
  False positives when detecting logos due to low detection thresholds.

- **Scalability Issues**
  Performance degradation with a growing brand reference list.

- **Dynamic Content Handling**
  Difficulty analyzing dynamically loaded webpage elements (e.g., JavaScript-rendered content).

- **Adaptation to Evolving Threats**
  Phishing tactics frequently evolve, requiring continuous updates to logo references and detection thresholds.

- **Environmental Constraints**
  Dependency on pre-trained models and large datasets for accurate detection.

Conclusion

learning and visual analysis to identify malicious websites with impressive accuracy. By focusing on logo detection and comparisons against trusted brand references, it provides a robust solution for identifying phishing attempts. Its reliability across a range of scenarios and domains underscores its potential as a critical component in modern cybersecurity frameworks. However, its limitations in scalability, detection of evolving phishing tactics, and handling dynamic content point to areas that require further development.

Future work

- **Enhancing Scalability**

Investigate ways to optimize performance for larger brand reference datasets to handle real-world scaling challenges.

- **Improving Detection of Evolving Threats**

Update algorithms to adapt to rapidly changing phishing strategies, including obfuscated content and dynamic elements.

- **User Behavior Analysis:**

Incorporate models that analyze user interaction patterns on webpages to detect phishing attempts, such as unusual clicks or form submissions.

- **Integration with Real-Time Systems**

Develop seamless integration with browsers and email clients for live phishing detection.

References

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

Kali Linux, Anaconda, and PyTorch documentation.