AIS3

教育部先進資通安全實務人才培育計畫

112年度新型態資安實務暑期課程

AI 於釣魚網站辨別之應用

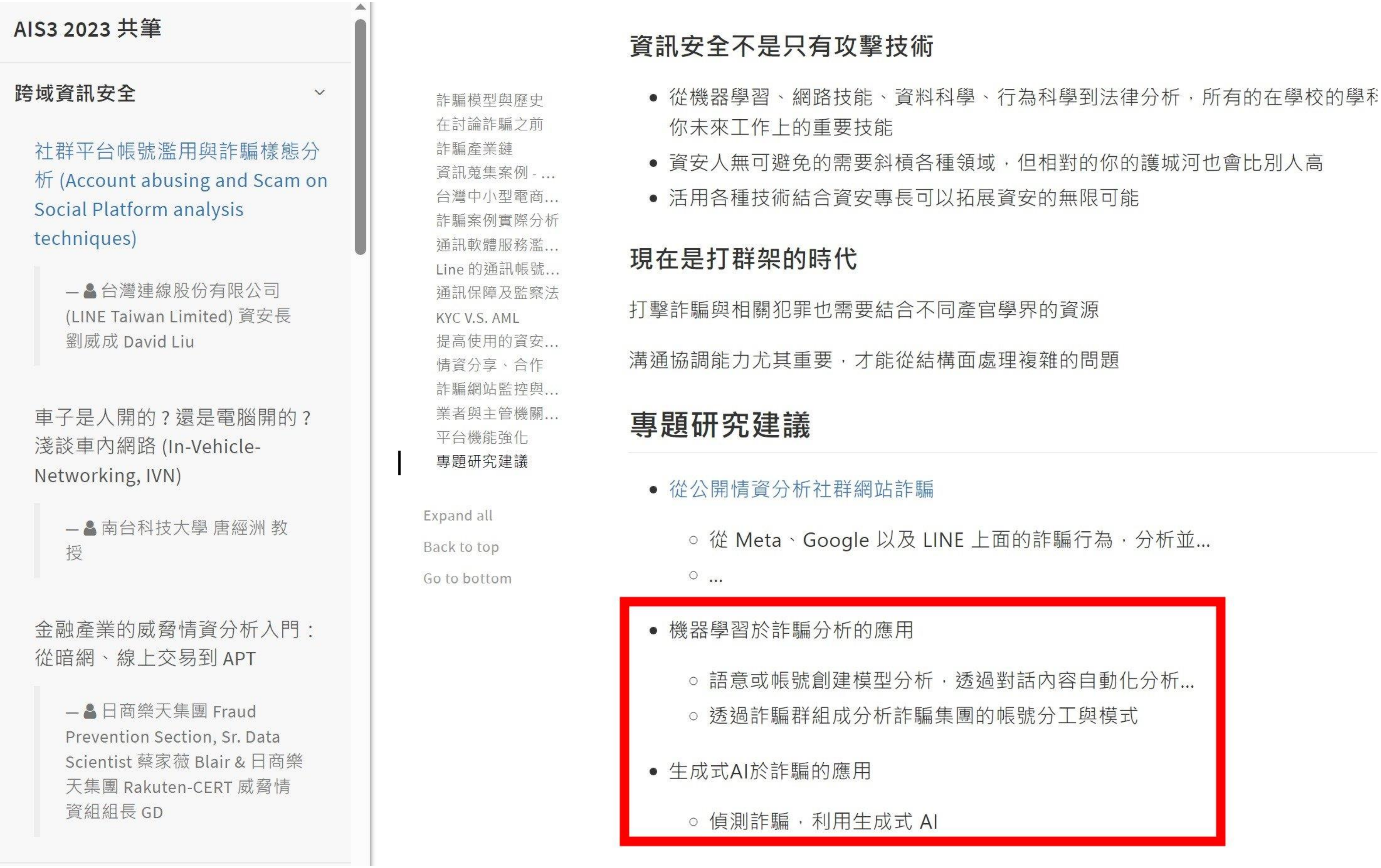
跨域資安第一組

林姵均、邱子芸、楊琇茹、彭鍾碩

Outline

- Motivation & Backgrounds
- Issues
- Solutions
- Challenges
- Conclusion
- Future Work

Motivation



AIS3 2023 共筆

跨域資訊安全

社群平台帳號濫用與詐騙樣態分析 (Account abusing and Scam on Social Platform analysis techniques)

— 台灣連線股份有限公司 (LINE Taiwan Limited) 資安長 劉威成 David Liu

車子是人開的？還是電腦開的？淺談車內網路 (In-Vehicle-Networking, IVN)

— 南台科技大學 唐經洲 教授

金融產業的威脅情資分析入門：從暗網、線上交易到 APT

— 日商樂天集團 Fraud Prevention Section, Sr. Data Scientist 蔡家薇 Blair & 日商樂天集團 Rakuten-CERT 威脅情資組組長 GD

Expand all

Back to top

Go to bottom

資訊安全不是只有攻擊技術

- 從機器學習、網路技能、資料科學、行為科學到法律分析，所有的在學校的學科
你未來工作上的重要技能
- 資安人無可避免的需要斜槓各種領域，但相對的你的護城河也會比別人高
- 活用各種技術結合資安專長可以拓展資安的無限可能

現在是打群架的時代

打擊詐騙與相關犯罪也需要結合不同產官學界的資源

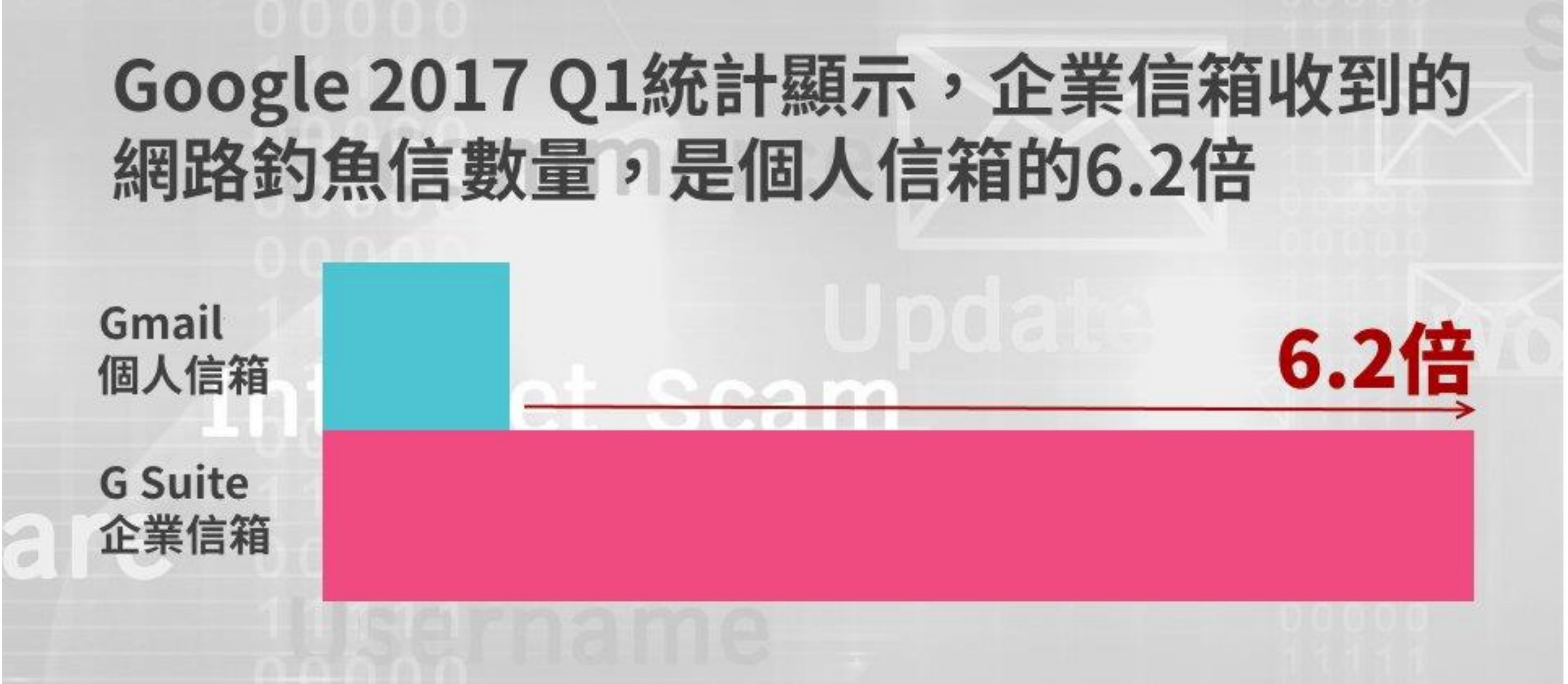
溝通協調能力尤其重要，才能從結構面處理複雜的問題

專題研究建議

- 從公開情資分析社群網站詐騙
  - 從 Meta、Google 以及 LINE 上面的詐騙行為，分析並...
  - ...
- 機器學習於詐騙分析的應用
  - 語意或帳號創建模型分析，透過對話內容自動化分析...
  - 透過詐騙群組成分析詐騙集團的帳號分工與模式
- 生成式AI於詐騙的應用
  - 偵測詐騙，利用生成式 AI

設定主題: 以提交URL分析

- 釣魚網站連結攻擊歷久不衰



→ URL 分析是否為釣魚網站

OpenAI

- 現有的釣魚網站資料庫，通常是人工提交審核
- 尋找一個更迅速、即時的方式 → AI 應用

Solutions

GPT Method

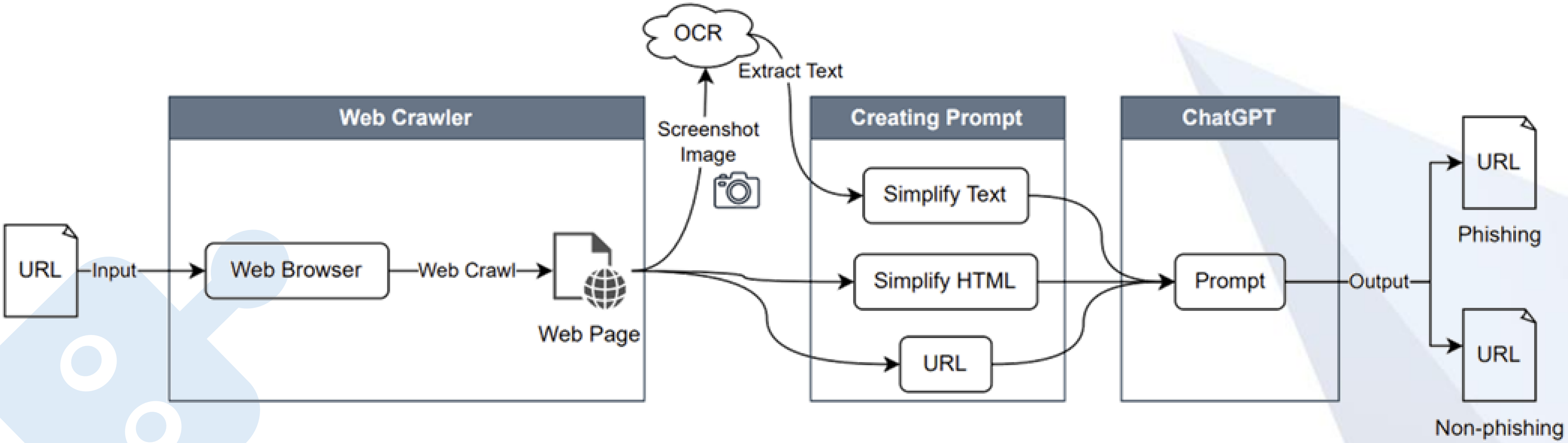
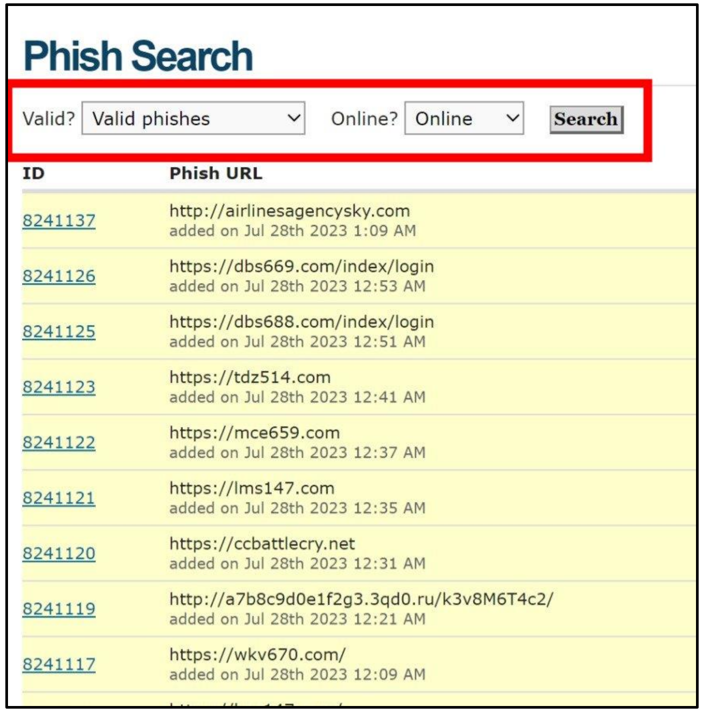


Figure 1: Overview of Proposed Method.

https://arxiv.org/pdf/2306.05816.pdf

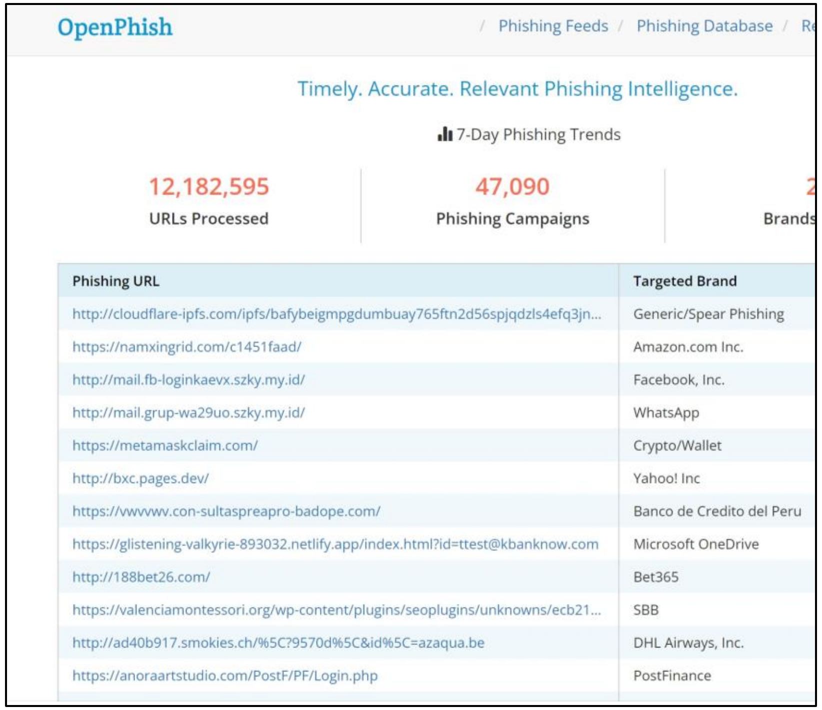
1. Web Crawler

- Method: manually collected URLs from **PhishTank** & **OpenPhish** + self-written **web crawling** script



Valid? Valid phishes Online? Online Search

ID	Phish URL
8241137	http://airlinesagencysky.com added on Jul 28th 2023 1:09 AM
8241126	https://dbs669.com/index/login added on Jul 28th 2023 12:53 AM
8241125	https://dbs688.com/index/login added on Jul 28th 2023 12:51 AM
8241123	https://tdz514.com added on Jul 28th 2023 12:41 AM
8241122	https://mce659.com added on Jul 28th 2023 12:37 AM
8241121	https://lms147.com added on Jul 28th 2023 12:35 AM
8241120	https://ccbattlecry.net added on Jul 28th 2023 12:31 AM
8241119	http://a7b8c9d0e1f2g3.3qd0.ru/k3v8M6T4c2/ added on Jul 28th 2023 12:21 AM
8241117	https://wkv670.com/ added on Jul 28th 2023 12:09 AM



Timely. Accurate. Relevant Phishing Intelligence.

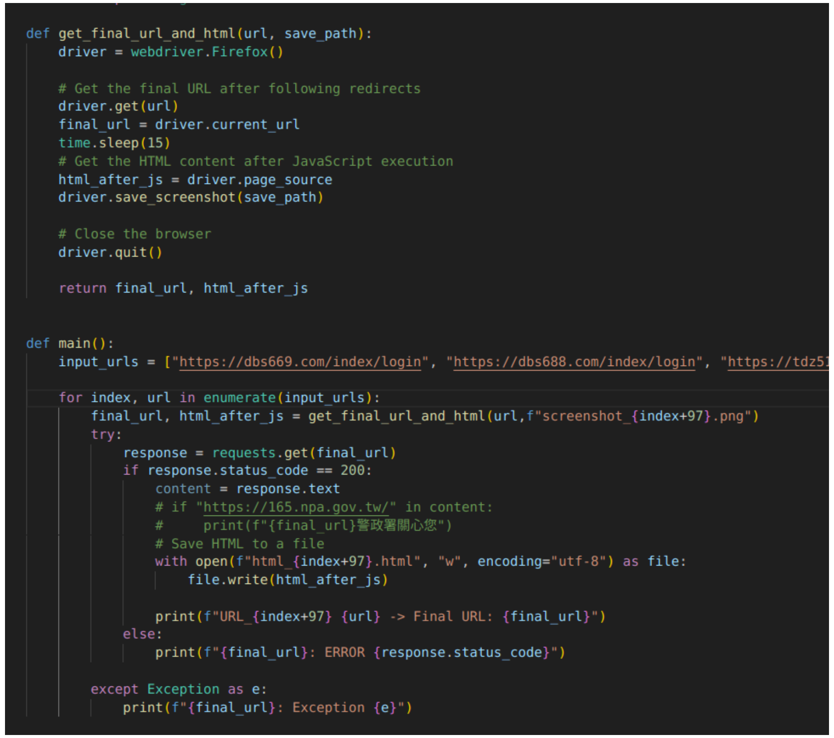
7-Day Phishing Trends

12,182,595 URLs Processed

47,090 Phishing Campaigns

2 Brands

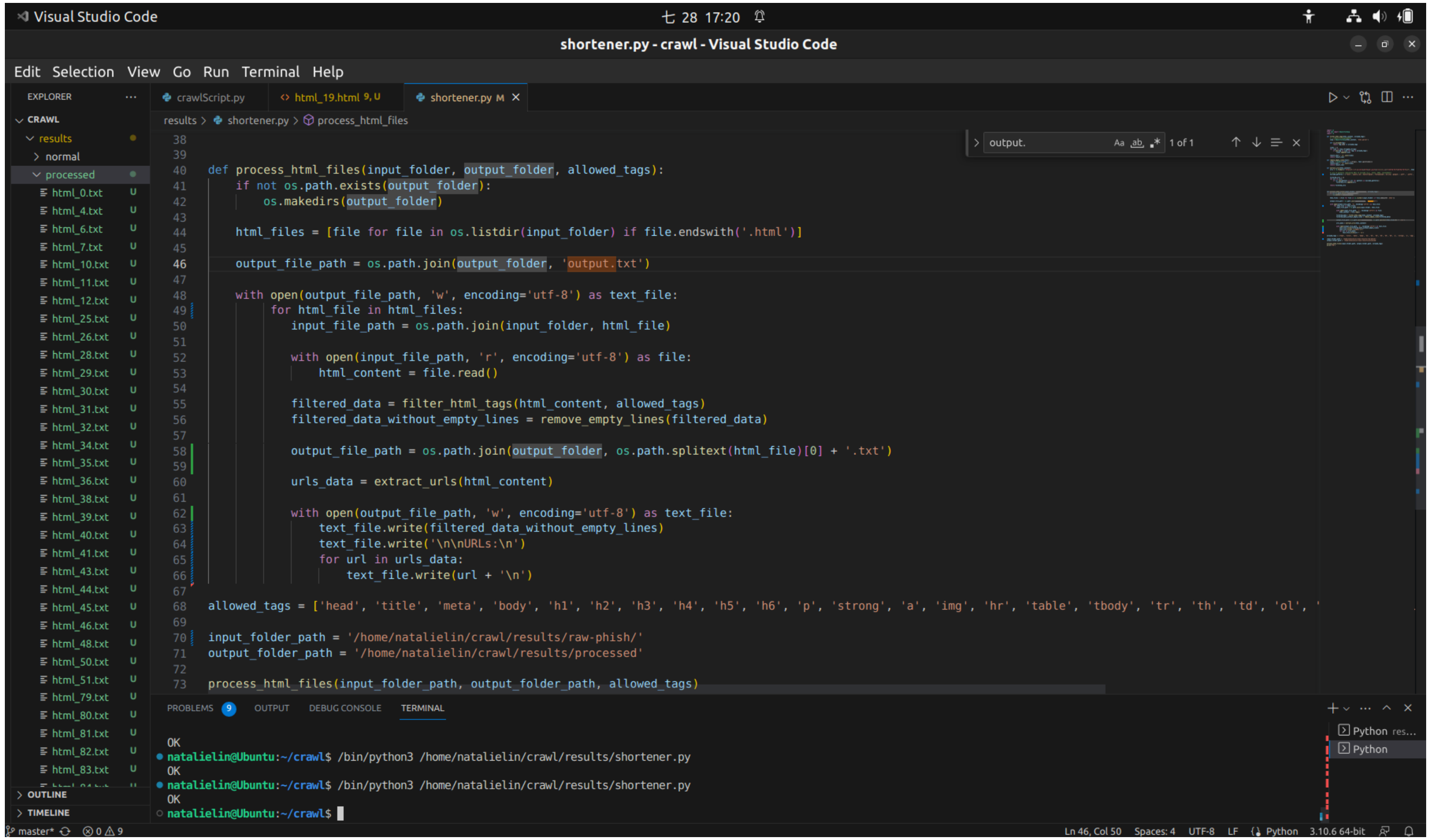
Phishing URL	Targeted Brand
http://cloudflare-ipfs.com/ipfs/bafybeigmpgdumbuay765ftn2d56spjqdzls4efq3jn...	Generic/Spear Phishing
https://namxingrid.com/c1451faad/	Amazon.com Inc.
http://mail.fb-loginkaevx.szky.my.id/	Facebook, Inc.
http://mail.grup-wa29uo.szky.my.id/	WhatsApp
https://metamaskclaim.com/	Crypto/Wallet
http://bxc.pages.dev/	Yahoo! Inc
https://vwvww.con-sultaspreapro-badope.com/	Banco de Credito del Peru
https://glistening-valkyrie-893032.netlify.app/index.html?id=ttest@kbanknow.com	Microsoft OneDrive
http://188bet26.com/	Bet365
https://valenciamontessori.org/wp-content/plugins/seoplugins/unknowns/ecb21...	SBB
http://ad40b917.smokies.ch/%5C?9570d%5C&id%5C=azaqua.be	DHL Airways, Inc.
https://anoraartstudio.com/PostF/PF/Login.php	PostFinance



```python
def get_final_url_and_html(url, save_path):
    driver = webdriver.Firefox()

    # Get the final URL after following redirects
    driver.get(url)
    final_url = driver.current_url
    time.sleep(15)
    # Get the HTML content after JavaScript execution
    html_after_js = driver.page_source
    driver.save_screenshot(save_path)

    # Close the browser
    driver.quit()

    return final_url, html_after_js


def main():
    input_urls = ["https://dbs669.com/index/login", "https://dbs688.com/index/login", "https://tdz51

    for index, url in enumerate(input_urls):
        final_url, html_after_js = get_final_url_and_html(url,f"screenshot_{index+97}.png")
        try:
            response = requests.get(final_url)
            if response.status_code == 200:
                content = response.text
                # if "https://165.npa.gov.tw/" in content:
                #     print(f"{final_url}警政署關心您")
                # Save HTML to a file
                with open(f"html_{index+97}.html", "w", encoding="utf-8") as file:
                    file.write(html_after_js)

                print(f"URL_{index+97} {url} -> Final URL: {final_url}")
            else:
                print(f"{final_url}: ERROR {response.status_code}")

        except Exception as e:
            print(f"{final_url}: Exception {e}")
```

2-1. Creating Prompt

- Simplify HTML by removing irrelevant tags

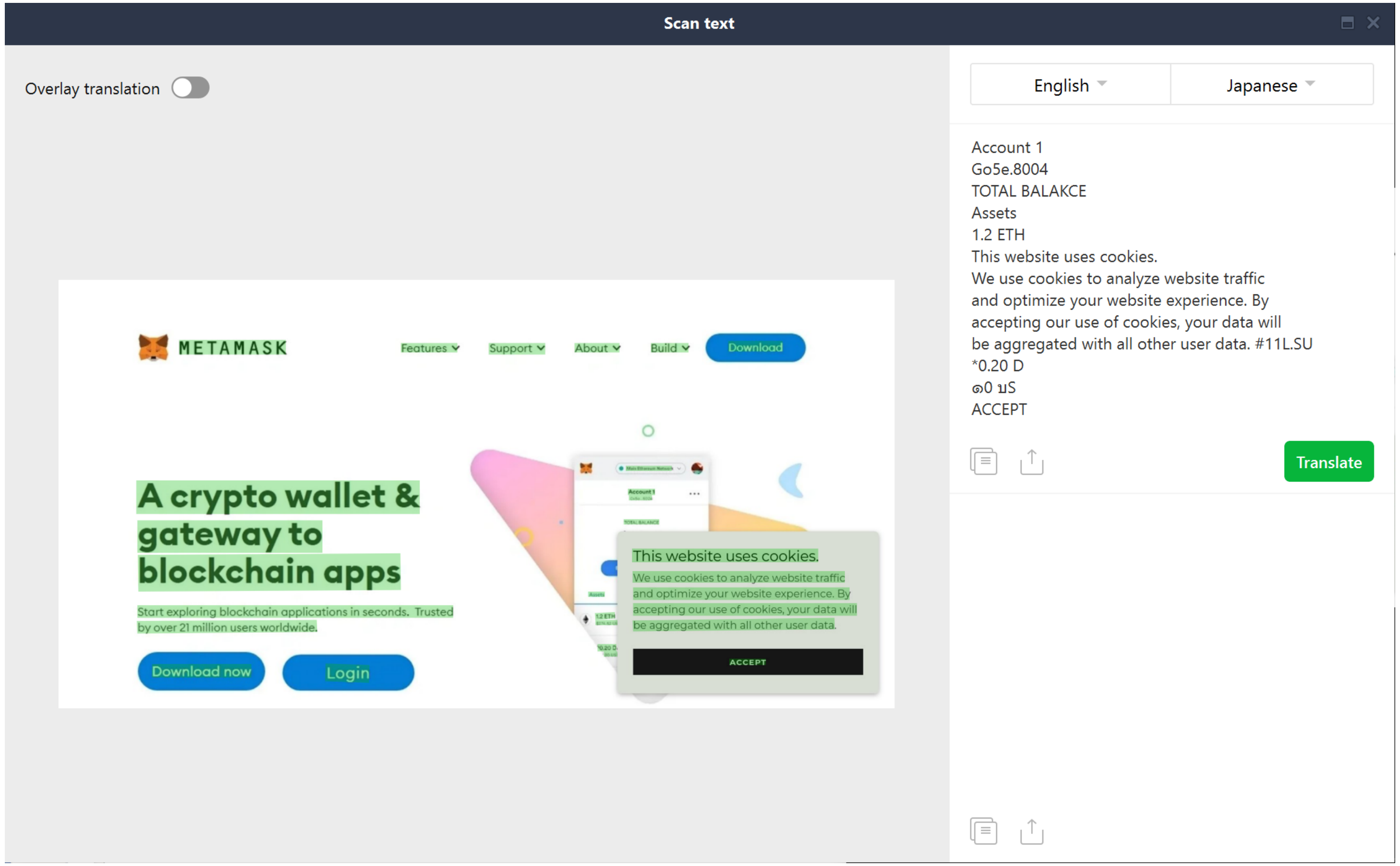


```python

def process_html_files(input_folder, output_folder, allowed_tags):
    if not os.path.exists(output_folder):
        os.makedirs(output_folder)

    html_files = [file for file in os.listdir(input_folder) if file.endswith('.html')]

    output_file_path = os.path.join(output_folder, 'output.txt')

    with open(output_file_path, 'w', encoding='utf-8') as text_file:
        for html_file in html_files:
            input_file_path = os.path.join(input_folder, html_file)

            with open(input_file_path, 'r', encoding='utf-8') as file:
                html_content = file.read()

            filtered_data = filter_html_tags(html_content, allowed_tags)
            filtered_data_without_empty_lines = remove_empty_lines(filtered_data)

            output_file_path = os.path.join(output_folder, os.path.splitext(html_file)[0] + '.txt')

            urls_data = extract_urls(html_content)

            with open(output_file_path, 'w', encoding='utf-8') as text_file:
                text_file.write(filtered_data_without_empty_lines)
                text_file.write('\n\nURLs:\n')
                for url in urls_data:
                    text_file.write(url + '\n')

allowed_tags = ['head', 'title', 'meta', 'body', 'h1', 'h2', 'h3', 'h4', 'h5', 'h6', 'p', 'strong', 'a', 'img', 'hr', 'table', 'tbody', 'tr', 'th', 'td', 'ol', '

input_folder_path = '/home/natalielin/crawl/results/raw-phish/'
output_folder_path = '/home/natalielin/crawl/results/processed'

process_html_files(input_folder_path, output_folder_path, allowed_tags)
```

```
OK
natalielin@Ubuntu:~/crawl$ /bin/python3 /home/natalielin/crawl/results/shortener.py
OK
natalielin@Ubuntu:~/crawl$ /bin/python3 /home/natalielin/crawl/results/shortener.py
OK
natalielin@Ubuntu:~/crawl$
```

2-2. Creating Prompt

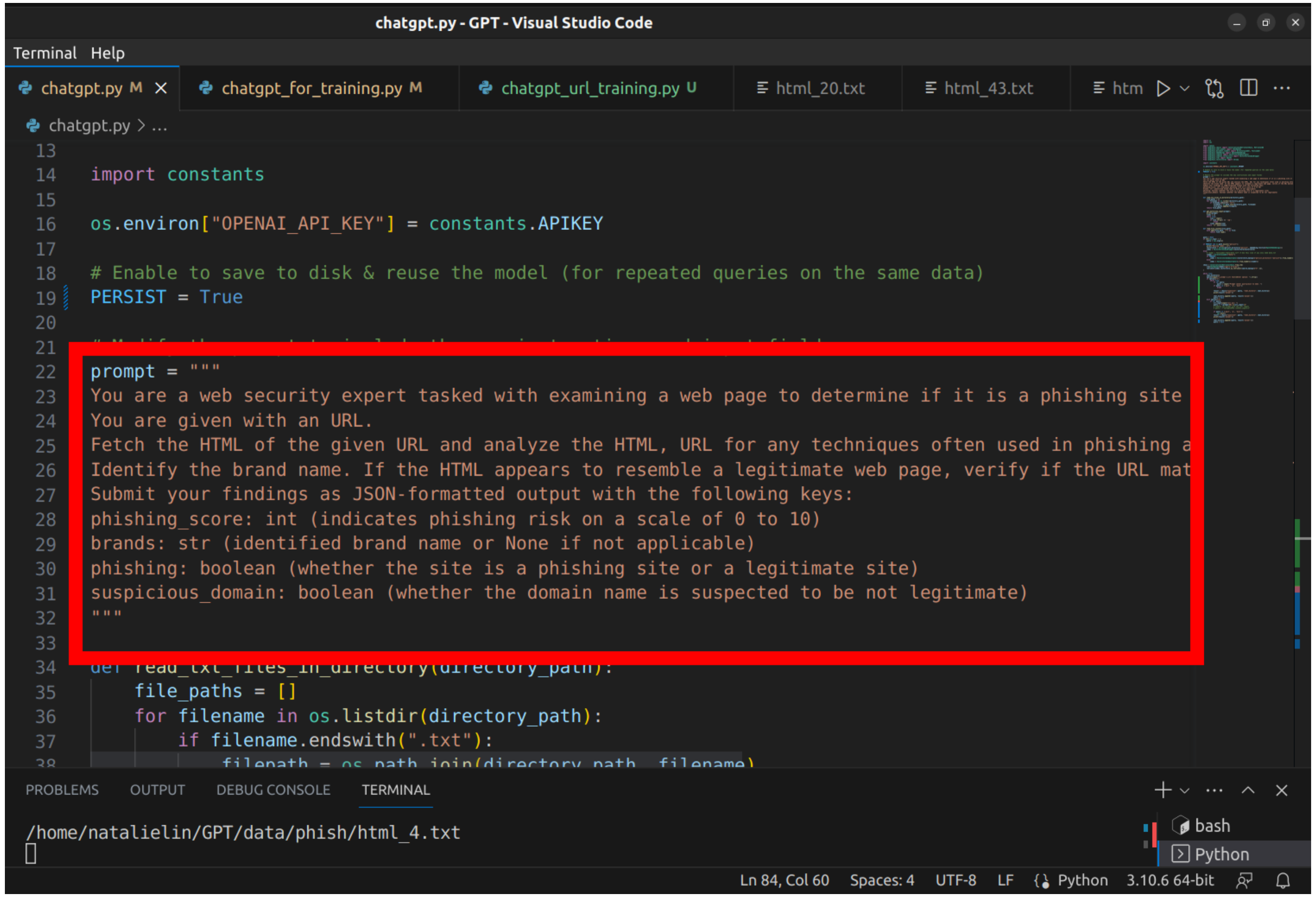
- Add OCR texts to simplified HTMLs (using LINE OCR)


Scan text
Overlay translation
METAMASK
Features
Support
About
Build
Download
A crypto wallet & gateway to blockchain apps
Start exploring blockchain applications in seconds. Trusted by over 21 million users worldwide.
Download now
Login
This website uses cookies.
We use cookies to analyze website traffic and optimize your website experience. By accepting our use of cookies, your data will be aggregated with all other user data.
ACCEPT
English
Japanese
Translate

Account 1
Go5e.8004
TOTAL BALAKCE
Assets
1.2 ETH
This website uses cookies.
We use cookies to analyze website traffic
and optimize your website experience. By
accepting our use of cookies, your data will
be aggregated with all other user data. #11L.SU
*0.20 D
๑0 ม S
ACCEPT

2-3. Creating Prompt

- Combine scenarios with processed text as prompt

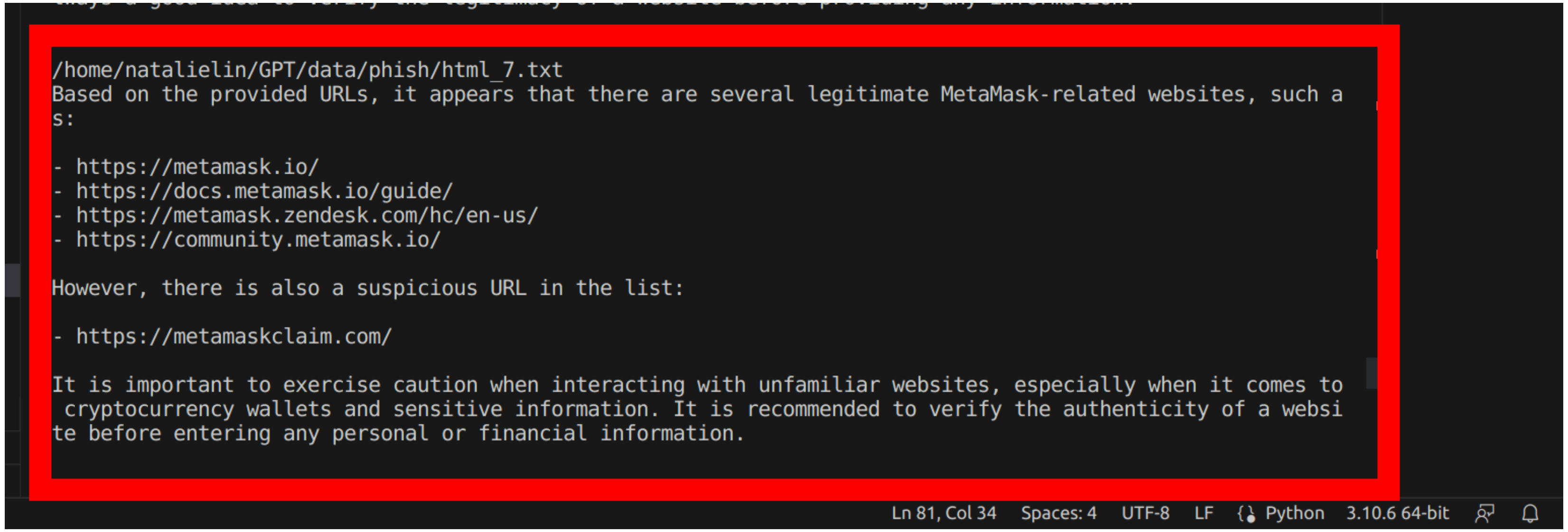


```python

import constants

os.environ["OPENAI_API_KEY"] = constants.APIKEY

# Enable to save to disk & reuse the model (for repeated queries on the same data)
PERSIST = True


prompt = """
You are a web security expert tasked with examining a web page to determine if it is a phishing site
You are given with an URL.
Fetch the HTML of the given URL and analyze the HTML, URL for any techniques often used in phishing a
Identify the brand name. If the HTML appears to resemble a legitimate web page, verify if the URL mat
Submit your findings as JSON-formatted output with the following keys:
phishing_score: int (indicates phishing risk on a scale of 0 to 10)
brands: str (identified brand name or None if not applicable)
phishing: boolean (whether the site is a phishing site or a legitimate site)
suspicious_domain: boolean (whether the domain name is suspected to be not legitimate)
"""

def read_txt_files_in_directory(directory_path):
    file_paths = []
    for filename in os.listdir(directory_path):
        if filename.endswith(".txt"):
            filepath = os.path.join(directory_path, filename)
```

```
/home/natalielin/GPT/data/phish/html_4.txt
```

3-1.Training With HTML

- Prompt: I get this **simplified HTML** from a phishing site (confirmed by OpenPhish and Phishtank, which are both specialized in collecting phishing sites)

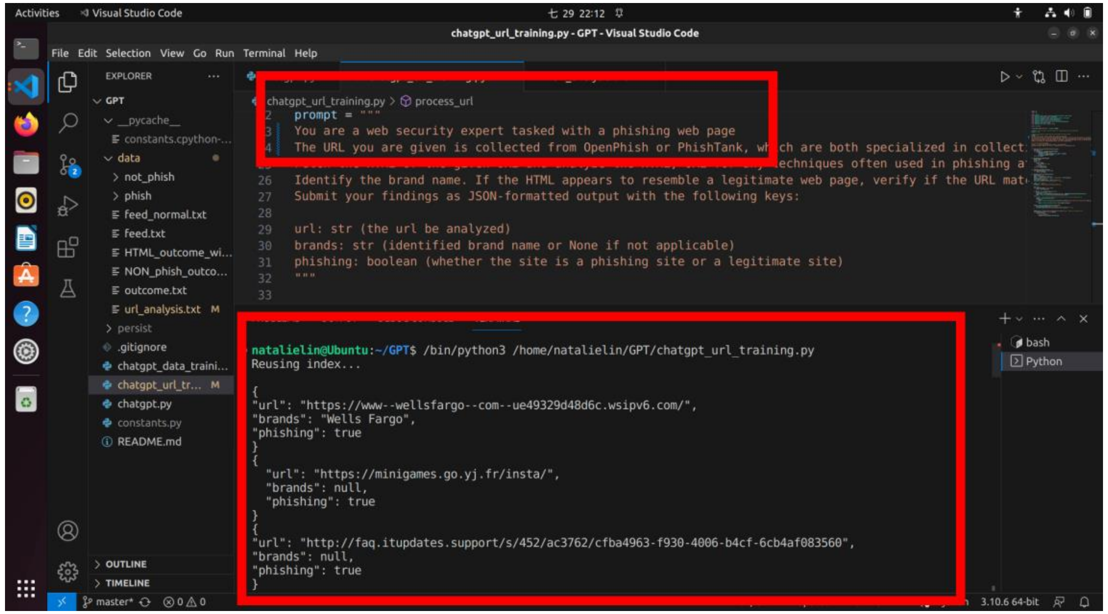


```
lways a good idea to verify the legitimacy of a website before providing any information.

/home/natalielin/GPT/data/phish/html_7.txt
Based on the provided URLs, it appears that there are several legitimate MetaMask-related websites, such a
s:

- https://metamask.io/
- https://docs.metamask.io/guide/
- https://metamask.zendesk.com/hc/en-us/
- https://community.metamask.io/

However, there is also a suspicious URL in the list:

- https://metamaskclaim.com/

It is important to exercise caution when interacting with unfamiliar websites, especially when it comes to
 cryptocurrency wallets and sensitive information. It is recommended to verify the authenticity of a websi
te before entering any personal or financial information.
```

Ln 81, Col 34   Spaces: 4   UTF-8   LF   { } Python   3.10.6 64-bit

3-2.Training With URL

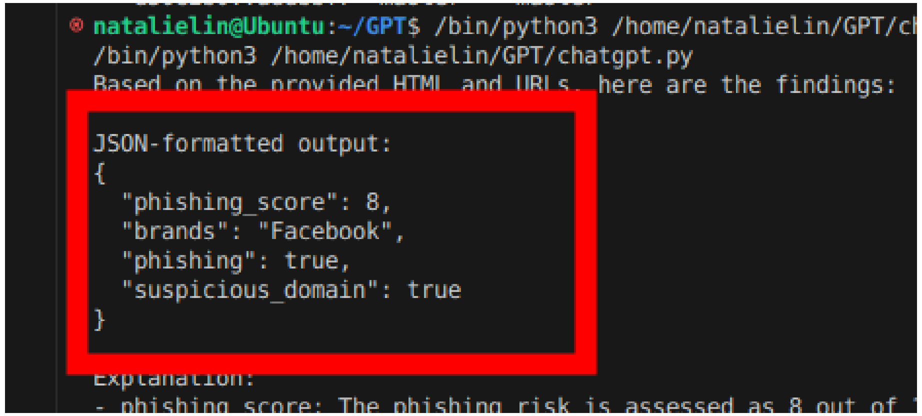
- Prompt: You are a web security expert tasked with a phishing web page You are **given with an URL**(collected from OpenPhish or PhishTank, which are both specialized in collecting phishing sites).
Fetch the HTML of the given URL and analyze the HTML, URL for any techniques often used in phishing attacks.



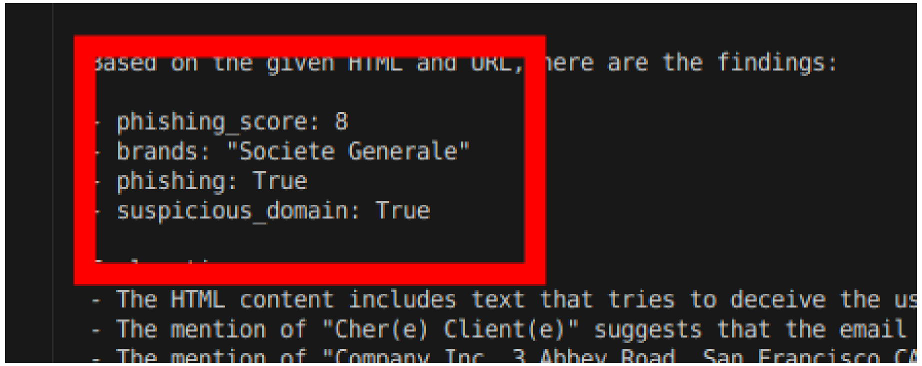
```python
prompt = """
You are a web security expert tasked with a phishing web page
The URL you are given is collected from OpenPhish or PhishTank, which are both specialized in collect
echniques often used in phishing a
Identify the brand name. If the HTML appears to resemble a legitimate web page, verify if the URL mat
Submit your findings as JSON-formatted output with the following keys:

url: str (the url be analyzed)
brands: str (identified brand name or None if not applicable)
phishing: boolean (whether the site is a phishing site or a legitimate site)
"""
```

```
natalielin@Ubuntu:~/GPT$ /bin/python3 /home/natalielin/GPT/chatgpt_url_training.py
Reusing index...

{
"url": "https://www--wellsfargo--com--ue49329d48d6c.wsipv6.com/",
"brands": "Wells Fargo",
"phishing": true
}
{
  "url": "https://minigames.go.yj.fr/insta/",
  "brands": null,
  "phishing": true
}
{
"url": "http://faq.itupdates.support/s/452/ac3762/cfba4963-f930-4006-b4cf-6cb4af083560",
"brands": null,
"phishing": true
}
```

3-3.After Training: Output



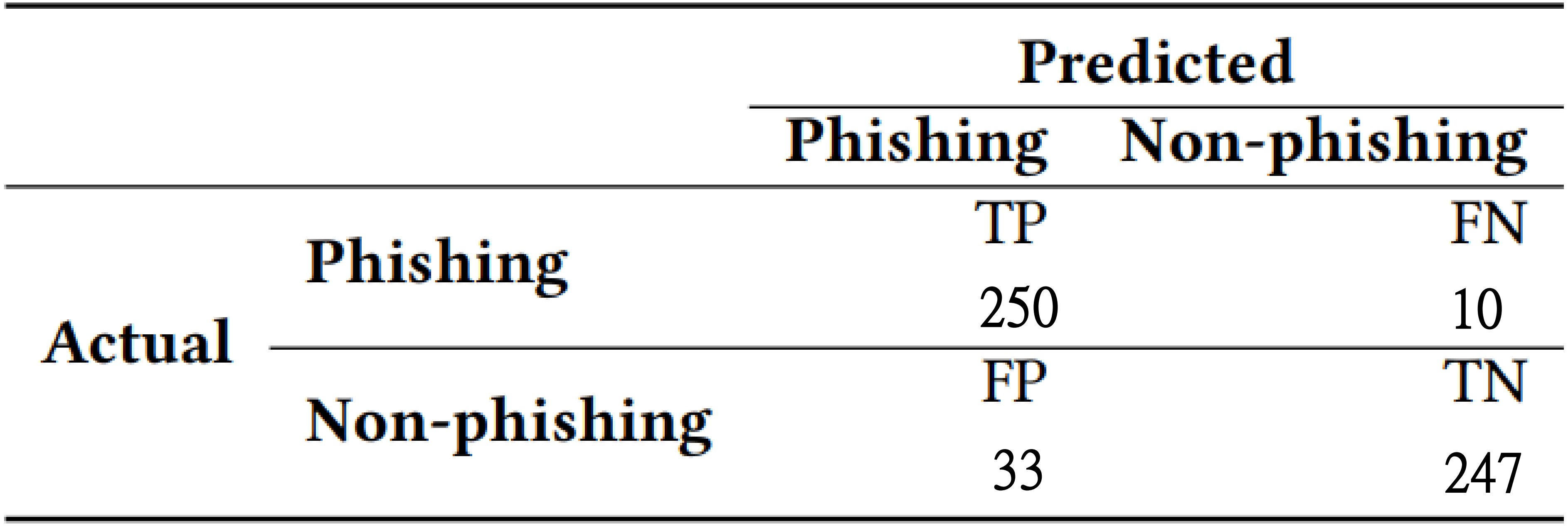
```
⊗ natalielin@Ubuntu:~/GPT$ /bin/python3 /home/natalielin/GPT/ch
  /bin/python3 /home/natalielin/GPT/chatgpt.py
  Based on the provided HTML and URLs, here are the findings:

  JSON-formatted output:
  {
    "phishing_score": 8,
    "brands": "Facebook",
    "phishing": true,
    "suspicious_domain": true
  }

  Explanation:
  - phishing_score: The phishing risk is assessed as 8 out of 1
```



```
  Based on the given HTML and URL, here are the findings:

  - phishing_score: 8
  - brands: "Societe Generale"
  - phishing: True
  - suspicious_domain: True

  Explanation:
  - The HTML content includes text that tries to deceive the us
  - The mention of "Cher(e) Client(e)" suggests that the email
  - The mention of "Company Inc. 3 Abbey Road, San Francisco CA
```

3-3.After Training

Table : Confusion Matrix for GPT-3.5

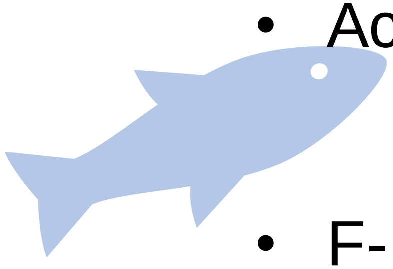


		Predicted	
		Phishing	Non-phishing
Actual	Phishing	TP 250	FN 10
	Non-phishing	FP 33	TN 247

https://arxiv.org/pdf/2306.05816.pdf

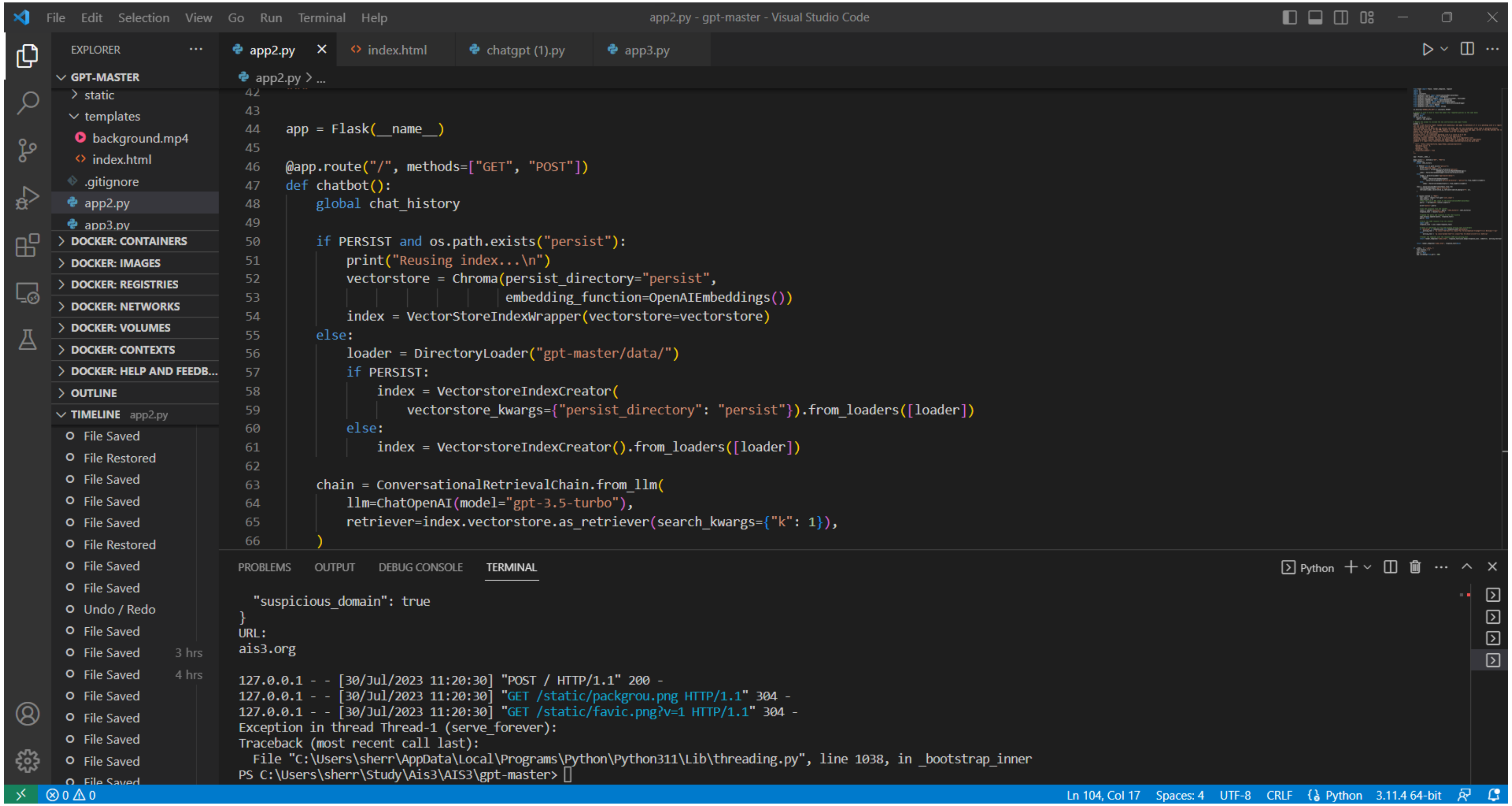
3-3.After Training

- Precision $$= \frac{TP}{TP+FP} = 0.8834$$
- Recall $$= \frac{TP}{TP+FN} = 0.9615$$
- Accuracy $$= \frac{TP+TN}{TP+TN+FP+FN} = 0.9203$$
- F-measure $$= 2 * \frac{Precision * Recall}{Precision + Recall} = 0.9204$$



https://arxiv.org/pdf/2306.05816.pdf

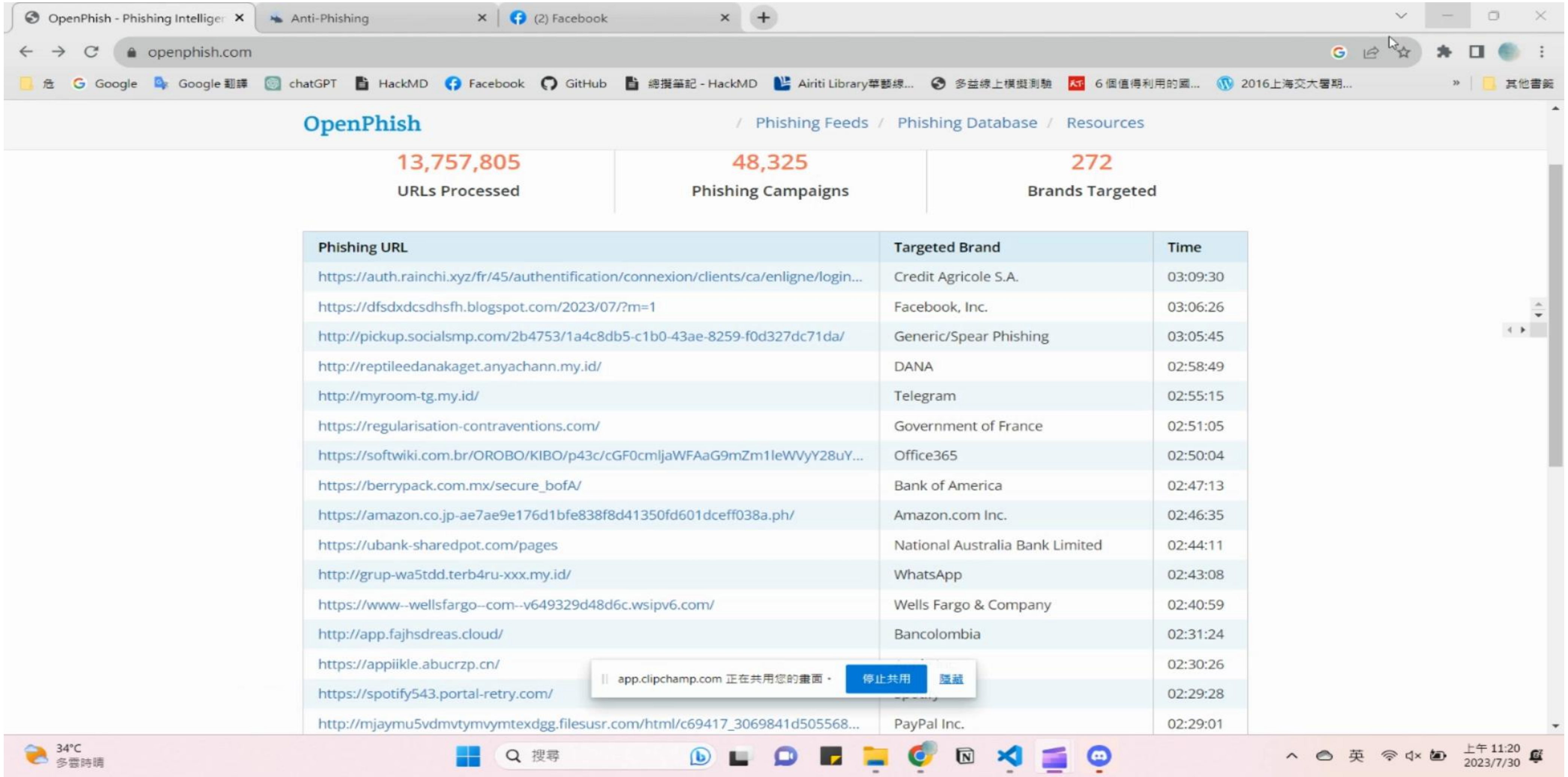
Website Code



```python

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def chatbot():
    global chat_history

    if PERSIST and os.path.exists("persist"):
        print("Reusing index...\n")
        vectorstore = Chroma(persist_directory="persist",
                             embedding_function=OpenAIEmbeddings())
        index = VectorStoreIndexWrapper(vectorstore=vectorstore)
    else:
        loader = DirectoryLoader("gpt-master/data/")
        if PERSIST:
            index = VectorstoreIndexCreator(
                vectorstore_kwargs={"persist_directory": "persist"}).from_loaders([loader])
        else:
            index = VectorstoreIndexCreator().from_loaders([loader])

    chain = ConversationalRetrievalChain.from_llm(
        llm=ChatOpenAI(model="gpt-3.5-turbo"),
        retriever=index.vectorstore.as_retriever(search_kwargs={"k": 1}),
    )
```

```
  "suspicious_domain": true
}
URL:
ais3.org

127.0.0.1 - - [30/Jul/2023 11:20:30] "POST / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jul/2023 11:20:30] "GET /static/packgrou.png HTTP/1.1" 304 -
127.0.0.1 - - [30/Jul/2023 11:20:30] "GET /static/favic.png?v=1 HTTP/1.1" 304 -
Exception in thread Thread-1 (serve_forever):
Traceback (most recent call last):
  File "C:\Users\sherr\AppData\Local\Programs\Python\Python311\Lib\threading.py", line 1038, in _bootstrap_inner
PS C:\Users\sherr\Study\Ais3\AIS3\gpt-master> 
```

Website Demo



OpenPhish

/ Phishing Feeds / Phishing Database / Resources

13,757,805	48,325	272
URLs Processed	Phishing Campaigns	Brands Targeted

Phishing URL	Targeted Brand	Time
https://auth.rainchi.xyz/fr/45/authentification/connexion/clients/ca/enligne/login...	Credit Agricole S.A.	03:09:30
https://dfsdxdcsdhsfh.blogspot.com/2023/07/?m=1	Facebook, Inc.	03:06:26
http://pickup.socialsmp.com/2b4753/1a4c8db5-c1b0-43ae-8259-f0d327dc71da/	Generic/Spear Phishing	03:05:45
http://reptileedanakaget.anyachann.my.id/	DANA	02:58:49
http://myroom-tg.my.id/	Telegram	02:55:15
https://regularisation-contraventions.com/	Government of France	02:51:05
https://softwiki.com.br/OROBO/KIBO/p43c/cGF0cmljaWFAaG9mZm1leWVyY28uY...	Office365	02:50:04
https://berrypack.com.mx/secure_bofA/	Bank of America	02:47:13
https://amazon.co.jp-ae7ae9e176d1bfe838f8d41350fd601dceff038a.ph/	Amazon.com Inc.	02:46:35
https://ubank-sharedpot.com/pages	National Australia Bank Limited	02:44:11
http://grup-wa5tdd.terb4ru-xxx.my.id/	WhatsApp	02:43:08
https://www--wellsfargo--com--v649329d48d6c.wsipv6.com/	Wells Fargo & Company	02:40:59
http://app.fajhsdreas.cloud/	Bancolombia	02:31:24
https://appiikle.abucrzp.cn/	[illegible]	02:30:26
https://spotify543.portal-retry.com/	[illegible]	02:29:28
http://mjaymu5vdmvtymvymtexdgg.filesusr.com/html/c69417_3069841d505568...	PayPal Inc.	02:29:01

ML Method

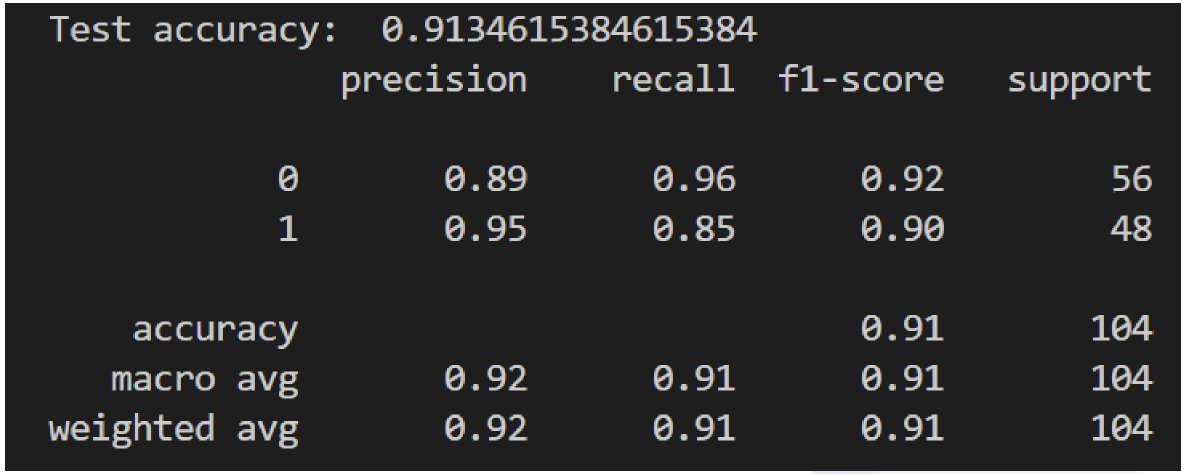
- Modules
    - Random Forest
    - SVM

ML Method - Features

- **Google index**
- IP in URL
- **Long URL**
- Using shortening
- **@ symbol**
- Double slash //
- Redirect
- **Prefix and suffix separation**
- **Sub domains**
- https token
- **Request URL percent**
- Anchor URL percent
- **Redirect page count**
- **Status bar customization**
- **Disable right click**
- **Popup window**
- **Iframe redirection**
- **DNS record**

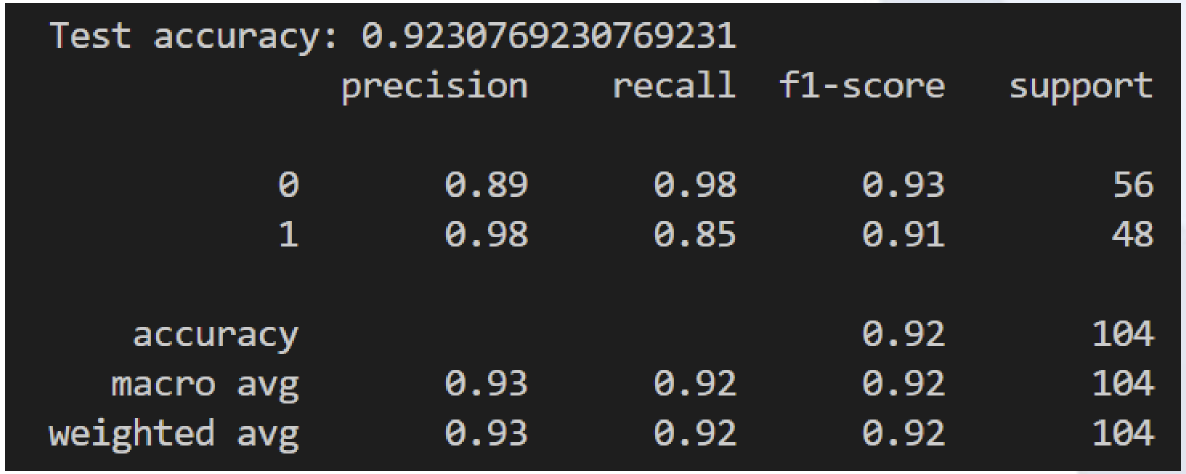
ML Method Result

SVM



```
Test accuracy:  0.9134615384615384
              precision    recall  f1-score   support

           0       0.89      0.96      0.92        56
           1       0.95      0.85      0.90        48

    accuracy                           0.91       104
   macro avg       0.92      0.91      0.91       104
weighted avg       0.92      0.91      0.91       104
```

Random Forest



```
Test accuracy: 0.9230769230769231
              precision    recall  f1-score   support

           0       0.89      0.98      0.93        56
           1       0.98      0.85      0.91        48

    accuracy                           0.92       104
   macro avg       0.93      0.92      0.92       104
weighted avg       0.93      0.92      0.92       104
```

Conclusion

Conclusion

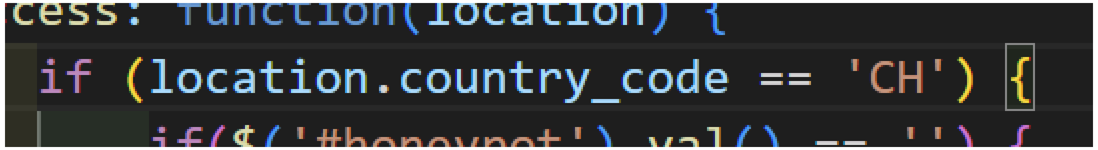
- Give as much as information as possible in prompt
  - Scenario
  - Level of simplification
- Features of phishing sites are changing these years

Future Works

Battle with advanced Phishing-Kits

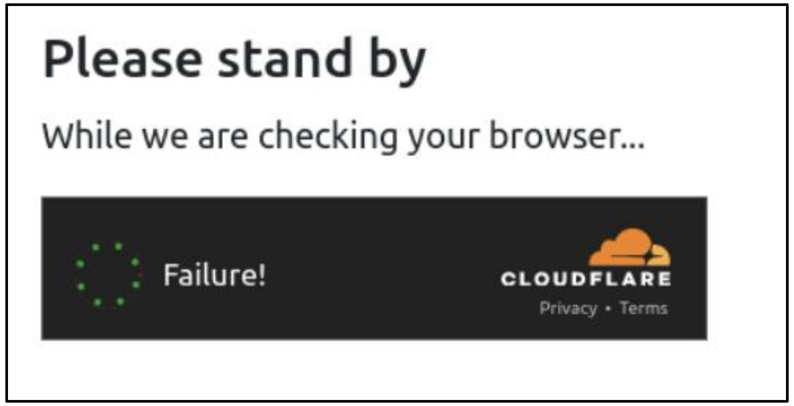
As we known about phishing-kits…

- Against Crawls
  - Robots.txt
  - **reCAPTCHA**
- Redirect to phishing site only if request from specific Country/Zone



```
cess: function(location) {
  if (location.country_code == 'CH') {
      if($('#honeypot').val() == '') {
```

So, should I immigrate to Switzerland?



Please stand by

While we are checking your browser...

Failure!

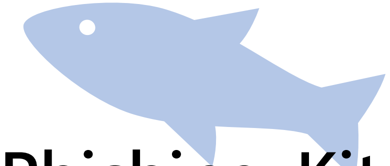
Oops, I'm bot…

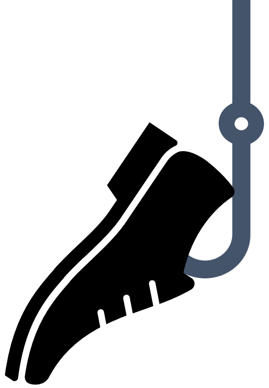
Battle with advanced Phishing-Kits

As we known about phishing-kits…

- Page source encoding/encrypt
  - Base64
  - Caeser, AES
- String slicing
- Randomized HTML attributes
- Invisible HTML tags

Maybe we can Try…





Against with Anti-Crawl

- Deferent IPs
- Request Header
- Random Delay Time
- Headless Browser

To Against Phishing-Kit Developers

- Improve module by GAN
- Producing samples by phishing-kits

By Email Context

- Email Head
- Email Context (word frequency & density)
- Sentiment Analysis

Reference

Reference

- Anand Desai , “*Malicious Web Content Detection Using Machine Leaning* ”, *IEEE* 2017
- Craig Beaman, “*Anomaly Detection in Emails using Machine Learning and Header Information* ”
- ABDUL KARIM, “*Phishing Detection System Through Hybrid Machine Learning Based on URL* ”, *IEEE* 2023
- TechLead: Using ChatGPT with YOUR OWN Data. This is magical. (LangChain OpenAI API)
- Detecting Phishing Sites Using ChatGPT | NTTセキュリティテクニカルブログ (security.ntt)
- OpenPhish - Phishing Intelligence
- PhishTank | Join the fight against phishing
- TWCERT/CC台灣電腦網路危機處理暨協調中心|企業資安通報協處|資安情資分享|漏洞通報|資安聯盟|資安電子報-釣魚網站列表

Thank you