ID: 116162

Identifikácia phishingových web stránok použitím ML

Bc. Michael John Čverčko

Zámer:

Identifikácia phishingových web stránok použitím ML. Vytvorím ML model na detektovanie phishingových web stránok pomocou analýzy url patternov, obsahu web stránky a jej metadáta. Zber dostupných datasetov, natrainujem model a spravím evaluáciu modelu.

Analýza problémovej oblasti a existujúcich riešení

Phishingové web stránky sú zamerané na podvodné získavanie citlivých informácií, ako sú prihlasovacie údaje, čísla kreditných kariet alebo osobné údaje. Tieto stránky často napodobňujú vzhľad a správanie legitímnych webov čím obete sú oklamané a nevedomo vložia ich citlivé dáta na stránku. Takto sa útočník dostane k citlivým dát použiíeľov.

Medzi najhlavnejšie výzvy v boji proti phishingu sú:

Rýchle prispôsobenie phishingových stránok: Phishingové stránky sa menia, sú krátkodobo aktívne a využívajú rôzne techniky, ako skrátené URL a dynamický obsah.

Komplexnosť dát: Identifikácia phishingových web stránok vyžaduje kombináciu analýzy URL, obsahu a metadát.

Škálovateľnosť: Detekčný systém musí byť rýchly a schopný analyzovať veľké množstvo prístupov v reálnom čase.

Existujúce riešenia ako odhaliť phishing:

Pravidlové prístupy: Využívajú preddefinované heuristiky, ako je dĺžka URL, neprítomnosť HTTPS, alebo podozrivé slová v URL ake je buď "login" alebo "verify". Problémy pri tomto riešenie je nízka schopnosť detekcie nových, doteraz neznámych útokov.

Blacklisty a whitelisty: Systémy ako Google Safe Browsing alebo OpenPhish poskytujú databázy známych phishingových a legitímnych stránok.
Problémy pri tomto riešenie je potrebná častá aktualizácia zoznamu čo spôsobuje, že je to neefektívne voči novým phishingovým stránkam.

Strojové učenie (Machine Learning [ML]): Využíva automatickú analýzu vzorov a metadát na identifikáciu phishingových stránok. Môže sa využívať Supervised Learning (učenie s učiteľom), čo znamená, že model je trénovaný na označených dátach (napr. phishing/legitímne stránky), alebo Unsupervised Learning (učenie bez učiteľa), ktoré umožňuje detekciu skupín podobných stránok bez predchádzajúceho označenia.

Najpoužívanejšie prístupy pre ML:

URL Analysis: Extrahuje vlastnosti ako počet znakov, prítomnosť špecifických slov alebo doménové atribúty.

Content Analysis: Analyzuje HTML obsah a JavaScript, aby zistil podozrivé formy alebo kód.

Metadata Analysis: Zahŕňa SSL/TLS certifikáty, WHOIS dáta, vek domény.

Hybridné prístupy: Kombinujú pravidlá, blacklisty a ML pre komplexnú ochranu. Usecase by bol ako emailový filter založený na pravidlách detekuje najzrejmejšie hrozby, zatiaľ čo ML analyzuje menej zrejmé prípady.

Moje kľúčové poznatky:

Pri detekcií phishingových web stránok s použitím ML je najlepší Random Forest a Gradient Boosted Trees z dôvodu, že dosahujú najvyššiu presnosť pri tabulárnych dátach. https://app.dimensions.ai/details/publication/pub.1160309483

Analýza URL je najvýznamnejší faktor. Použitie textového spracovania (TF-IDF) na analýzu znakov URL.

Identifikácia phishingových stránok na základe obsahu, ako sú podozrivé formuláre, JavaScript kódy a odkazy na externé zdroje.

WHOIS dáta poskytujú dôležité informácie o doméne, ako je vek, registrátor, alebo anonymný vlastník.

Identifikované medzery:

Nedostatočné pokrytie nových phishingových techník.Nie všetky stránky majú dostupné metadáta (napr. WHOIS). Efektivita a rýchlosť modelov v reálnom čase.

Návrh riešenia problému

Idem natrénovať model, aby vedel rozoznať phishingové stránky. Systém s využitím tohto modelu bude schopný automaticky analyzovať URL, obsah a metadáta stránok, spoľahlivo identifikovať phishingové stránky s vysokou presnosťou a bude dostatočne rýchly na použitie v reálnom čase.

Architektúra

Komponenty systému

1. Zber dát:
    - Získanie označených dátových súborov obsahujúcich phishingové a legitímne stránky.
    - Zdroj dát: Grega Vrbančič's Phishing Dataset https://github.com/GregaVrbancic/Phishing-Dataset
2. Predspracovanie dát:
    - Čistenie dát: Odstránenie duplicitných záznamov, neplatných URL.
    - Extrahovanie features:
        - URL-based features: DÍžka URL, počet špeciálnych znakov, podozrivé slová.
        - Content-based features: Viditeľný text, HTML kód, JavaScript.
        - Metadata-based features: SSL certifikát, vek domény, registrátor.
3. Model strojového učenia:
    - Algoritmy: Random Forest
    - Typ učenia: Supervised Learning.
4. Vyhodnocovanie a testovanie:
    - Použitie metrik ako presnosť, precision, recall, F1-score.
5. Nasadenie:
    - API pre detekciu phishingu v reálnom čase cez Flask
    - Možné integrovať do bezpečnostných systémov alebo webových prehliadačov.

Porovnanie útokov

1.

Typ útoku

Phishing cez URL

Charakteristika

URL obsahuje podozrivé slová, znaky alebo skrátené odkazy.

Stratégia detekcie

Analýza URL vzorov (napr. dĺžka, TLD, znaky).

2.

Typ útoku

Phishing cez obsah

Charakteristika

Stránka obsahuje podvodné formuláre alebo kopíruje legitímny vzhľad.

Stratégia detekcie

Analýza HTML a JavaScript obsahu.

3.

Typ útoku

Phishing cez email

Charakteristika

Odkazy v emailoch vedú na phishingové stránky.

Stratégia detekcie

Detekcia phishingového odkazu z emailov.

4.

Typ útoku

Man-in-the-Middle

Charakteristika

Zachytávanie komunikácie medzi používateľom a legitímnou stránkou.

Stratégia detekcie

Kombinácia SSL analýzy a dynamických detekčných prvkov.

Metóda testovania

Rozdelím dataset na 80 % tréningové dáta a 20 % testovacie dáta.

Použijem presnosť (accuracy), precíznosť (precision), citlivosť (recall) a F1 skóre na vyhodnotenie modelu.

Natrénuj model Random Forest na tréningových dátach.

Vyhodnotím model na testovacích dátach pomocou uvedených metrík.

Matrica zámien (confusion matrix) zobrazí správne pozitívne, správne negatívne, falošné pozitívne a falošné negatívne predikcie.

Hodnoty metrík (napr. presnosť, precíznosť, citlivosť a F1 skóre).

Opis funkcií a použitia prototypu

GIT REPO: https://github.com/JohnZeki/ML-PHISHING/tree/main

```python
# ================================
# Imports and Data Preparation
# ================================

import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix
import pickle  # To save the model
from flask import Flask, request, jsonify  # Flask for API deployment

```

```python
# ================================
# Data Collection and Sanitation
# ================================

# Load the dataset
df = pd.read_csv('dataset_full.csv')

# Separate features and labels
X = df.iloc[:, :-1]  # All columns except the last one
y = df.iloc[:, -1]  # The last column as the target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

```

```python
# ================================
# Model Training
# ================================

# Train the Random Forest model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Evaluate the model on the test set
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Print metrics
print("Presnosť:", accuracy)
print("Preciznosť:", precision)
print("Citlivosť:", recall)
print("F1 skóre:", f1)
print("Matrica zámien:\n", confusion_matrix(y_test, y_pred))

# Save the trained model
with open('phishing_model.pkl', 'wb') as file:
    pickle.dump(model, file)
```

```python
# ================================
# Flask API for Deployment
# ================================

# Load the trained model
with open('phishing_model.pkl', 'rb') as file:
    loaded_model = pickle.load(file)

# Create Flask application
app = Flask(__name__)


# Define the prediction endpoint
@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    features = pd.DataFrame([data])  # Expecting data as a dictionary of feature values
    prediction = loaded_model.predict(features)[0]
    return jsonify({"is_phishing": bool(prediction)})


if __name__ == '__main__':
    app.run(debug=True)
```

```
(venv) PS C:\Users\micha\PycharmProjects\ML-phishing> waitress-serve --port=5000 app:app
Presnosť: 0.969825155104343
Precíznosť: 0.9519184069936862
Citlivosť: 0.9610983981693364
F1 skóre: 0.9564863765758438
Matrica zámien:
 [[11315   297]
 [  238  5880]]
INFO:waitress:Serving on http://0.0.0.0:5000
```

Na základe dostupného dátového setu hodnotím model veľmi pozitívne na trénovacej sade.

Presnosť (Accuracy):

0.9698 (96,98 %) označuje celkové percento správnych predikcií, či už phishingových alebo legitímnych URL.

Precíznosť (Precision):

0.9519 (95,19 %) je percento správne predikovaných phishingových URL zo všetkých URL, ktoré model označil ako phishingové.

Citlivosť (Recall):

0.9610 (96,10 %) označuje percento skutočných phishingových URL, ktoré model správne identifikoval.

F1 skóre:

0.9564 (95,64 %) je harmonický priemer precíznosti a citlivosti, ktorý poskytuje vyvážené hodnotenie výkonu modelu.

Confusion matrix

Riadky reprezentujú skutočné triedy:
Riadok 1: Skutočné legitímne URL.
Riadok 2: Skutočné phishingové URL.

Stĺpce reprezentujú predikované triedy:
Stĺpec 1: Predikované ako legitímne.
Stĺpec 2: Predikované ako phishingové.

11315: Správne negatívne (True Negatives) – legitímne URL správne klasifikované ako legitímne.
297: Falošne pozitívne (False Positives) – legitímne URL nesprávne klasifikované ako phishingové.
238: Falošne negatívne (False Negatives) – phishingové URL nesprávne klasifikované ako legitímne.
5880: Správne pozitívne (True Positives) – phishingové URL správne klasifikované ako phishingové.