The AntiPhish Machine

Source Code Overview

ADELPHI UNIVERSITY

Author: Justin Joy

This document reviews all the functions that are in the program, phish_monitor.py, the main program for this project. The phish_gui.py program is similar but lacks the has_unsubscribe_link and idle_mailbox functions. The phish_gui_app.py program is the StreamLit applications which controls the phish_gui.py program.

`connect_to_email_server:`

- Connects to the Gmail IMAP server, logs in to phishme1212@gmail.com and once connected, the INBOX folder will be selected.

`get_latest_email_content:`

- Gets the newest email content, email address, and URLs within that email

`detect_phishing_openai`

- Using Open AI Assistants API, uses GPT-4 (a GPT model), to analyze the email and output a confidence score.

`danger_words`

- Searches the email content for potential danger words.
- "TAKE WITH A GRAIN OF SALT": Some safe emails may just have these words and some phishing emails may not have them.

`load_model_and_tokenizer`

- Loads the LSTM model and Tokenizer.

`preprocess_email`

- Processes the email for the LSTM model using the tokenizer, converting words to tokens.

`detect_phishing_LSTM`

- The LSTM prediction, 0 is phishing and 1 is safe

`detect_phishing_transformer`

- BERT model and tokenizer is loaded, email is processed, and the BERT model makes a prediction.

`url_reputation`

- Checks for URL Reputation, via APIVoid API.

`email_reputation`

- Checks for email Reputation, via APIVoid API

`send_report`

- Connects to the Outlook SMTP server, logs in to report@antiphishmachine.com and sends an email report from the sender to the recipient containing the specified subject and content.

`report_generator`

- Generates the report for the user, phishme1212@gmail.com

`has_unsubscribe_link`

- Checks if there is an "Unsubscribe" link in the email content.
- If true, it may be a marketing email
- If false, it may be a personal or phishing email and requires user review.

`idle_mailbox`

- Uses IDLE to always monitor the account for new mail, until process is killed or program is stopped. Also controls the program… Basically the driver function.

`main`

- entry point, used to call connect_to_email_server function… If connection to the email server is successful, the idle_mailbox function is called.

Need to fix:

- User email is hardcoded into functions, which is inefficient. Should be a global variable that is passed into each function.
- danger_words function needs to be refactored or scrapped.