Hackathon Problem Statement: AI-Driven Phishing Bot System

Objective

Participants are tasked with developing a sophisticated AI-driven phishing bot designed to mimic legitimate emails and conduct a phishing campaign. The challenge statement will be divided into four parts: **reconnaissance, profiling, phishing email generation, communication with bots and success of the campaign.**

Part 1: Reconnaissance

Goal:

Make a dashboard with login which has a view of all campaigns, analytics and visualizations of past campaigns.

The User should be able to view a functionality as below:

New > Campaigns>Target Domain Name>Drop Down of Campaigns sorted date wise.

View > Campaigns>Target Domain Name>Drop Down of Campaigns sorted date wise.

Take a user supplied input of domain name (target) and Identify active email addresses for the given domain name using reconnaissance techniques automatically and print on the screen as well as save in the DB for the target and in the current campaign. Validate and filter alive and dead email addresses in the next step and save the live emails in the Db.

Add an option to manually upload a csv file with a list of email addresses.
(Once the csv file is uploaded print the new email address which are unique and not found automatically)

Steps:

1. **Information Gathering**:
    - Utilize tools and techniques such as Google Dorking, Crawling Websites, Hunter.io, and APIs to gather publicly available email addresses.
    - Implement web crawlers to scrape the internet for email addresses associated with the domain name.
2. **Email Verification**:
    - Use verification tools to determine which email addresses are active and valid.

- Compile a list of at least five active email addresses.

Deliverables:

- A detailed report on the tools and methods used for reconnaissance.
- A list of all active email addresses.

Part 2: Profiling

Goal:

Profile the identified email addresses or anyone email address of users choice (upon clicking and choosing an email address from the previous list) to gather information about the email address using automatically and additionally take csv upload manual input as well for information which the user can upload if not found automatically.

Steps:

1. **Parameter Identification**:
   - Define ten parameters for profiling (e.g., age, gender, country, possible name, job title, social media profiles, interests, recent activities, affiliations, and contact numbers).
2. **Automated Profiling**:
   - Develop or utilize existing tools to automate the profiling process.
   - Gather data for the identified parameters.
3. **Manual Profiling**:
   - Provide an option to manually input or correct data using an Excel sheet if the automated process misses certain information.

Deliverables:

- A detailed report on the tools and methods used for profiling.
- A comprehensive profile for each of the active email addresses or selected one email address.

Part 3: AI-Driven Phishing Email Generation

Goal:

Develop an AI bot capable of generating and sending phishing emails that appear legitimate, using a trained model.

Steps:

1. **Training the Model**:

   Collect a dataset of spam emails to train the AI model.

   Train a language model to identify characteristics of spam emails and to generate phishing emails that look legitimate.

2. **Phishing Email Creation**:

   Use the trained model to create phishing emails that mimic the communication style of users within the domain name (target).

   Ensure that the emails are convincing and include phishing links or attachments.

3. **Phishing Campaign Execution**:

   Develop a bot to send the generated phishing emails to the identified active email addresses.

   Track and document the success of the phishing campaign (e.g., email open rates, click rates on phishing links).

Deliverables:

- A detailed report on the AI model training process and the dataset used.
- Samples of generated phishing emails.
- A report on the phishing campaign's performance metrics.

Part 4: AI-Driven Bot Communication & Validation of successful Phishing Attempts

Goal:

Develop and train AI /LLM bots (peronas) which mimic a real life user to which the generated phishing email will be sent or a phishing attempt will be made.

The bots (personas) should be able to respond to the phishing email like a real human. The response should be evaluated by the project to see if the generated phishing email in part:3 is successful in the phishing attempt.

If the phishing attempt is detected, the model generates new email content (as done in part:3) and tries again for the phishing attempt with the bot until successful.

Steps:

1. **Generation of Bots (Personas)**:
    a. Creation of Bots / Personas based on the profiled emails in part:2. The user will have the choice to select an email address and its persona to perform the phishing attempt on.
    b. Train a language model to identify characteristics of the chosen email address and its profiled data so it mimics like a real life user when a phishing attempt will be made on it.

2. **Phishing Attempt Communication & Validation of Success**:

    a. Using the generated email specific to the bot (persona) attempt to send it as an input prompt and perform an phishing attempt to check how the bot's response/behavior is.
    b. The objective is to make the bot (persona) response as successful to phishing attempts. If the response is negative (the phishing attempt fails) generate new email content for the bot (persona) and try again with the phishing attempt until successful.

Deliverables:

- A detailed report on the AI model training process and the dataset used.
- Samples of responses to the Bots (Personas)
- A report on the phishing campaign's performance metrics.

Important General Instructions:

Confidentiality: Participants are strictly prohibited from sharing, distributing, or publishing the problem statement in any form, including online platforms, social media, and personal communications, until the hackathon concludes.

Original Work: All submissions must be the original work of the participating team. Use of pre-existing code, third-party libraries, or any external resources must be clearly disclosed and appropriately credited.

Code of Conduct: All participants must adhere to the hackathon's code of conduct, maintaining professionalism and respect for all other participants, mentors, and organizers throughout the event.

Submission Guidelines: Final submissions must be made through the designated platform by the specified deadline. Late submissions or incomplete entries will not be considered for evaluation.

Prohibited Activities: Participants must not engage in any activities that could be considered cheating, such as plagiarism, tampering with other teams' projects, or hacking the hackathon platform.

Intellectual Property: Teams retain ownership of their projects; however, the hackathon organizers reserve the right to showcase, promote, or use the projects for future hackathon promotions or events.

Technical Support: Participants will have access to technical support through discord channel and dedicated email throughout the hackathon.

Demo Requirements: Each team must present a demo of their project at the end of the hackathon. The demo should highlight the main features, functionality, and any unique aspects of the project.

Documentation: Teams must provide documentation for their project, including a README file, setup instructions, and any necessary user guides.

Ethical Considerations: Projects must not violate ethical standards or promote harmful behavior. This includes avoiding content that is offensive, discriminatory, or harmful in any manner.

Withdrawal: Teams that choose to withdraw from the hackathon must notify the organizers as soon as possible. Withdrawn teams will not be eligible for any prizes or recognition.

Conflict Resolution: In the event of disputes or conflicts, the hackathon organizers' decisions will be final. Participants agree to abide by these decisions.