SMART INDIA HACKATHON 2023

PS Code: SIH1454

Problem Statement Title:
Create an intelligent system using AI/ML to detect phishing domains that imitate the look and feel of genuine domains.

Team Name: Byte Busters

Team Leader Name: Bhuvan Deep Sagar

Institute Name: Netaji Subhas University of Technology

Project Description

- The most used method for compromising users globally is the phishing attack. Phishing links or websites are disseminated to target users through various channels, such as email and SMS. Some domains even host imitation login pages resembling the intended websites' legitimate login pages. Attempting to log in on these deceptive pages can compromise user login credentials and the potential download of malicious software onto their computers. This problem aims to detect such phishing domains among recently registered websites using publicly available databases, like the WHOIS Database, which provides a roster of newly registered domains.
- The system uses the following techniques:

1. Backend code/content similarity in web pages.
2. Web page image analysis (i.e., analysis between genuine and phishing site web page images; the more similarity, the better the probability score of being a lookalike phishing site).

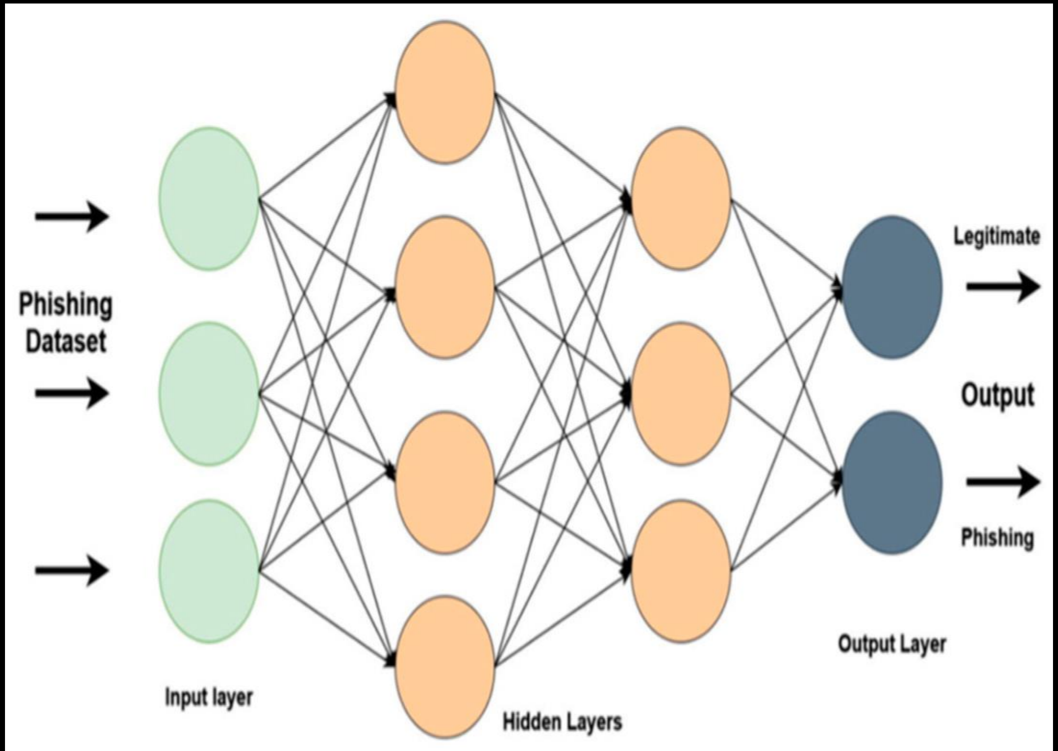
- The evaluation would be based on:

1. Probability scores of phishing domains on their proximity to the genuine domain.
2. Ability to detect new phishing domains in a reasonable time.
3. Ease of use and flexibility of output formats.

Tech Stack Used

1. **Programming Languages**:
   - **Python (**due to its extensive libraries and frameworks for ML, natural language processing, and data analysis.)
2. **Machine Learning and AI Frameworks**:
   - **Scikit-Learn** (A popular machine learning library for classification and feature selection tasks.)
   - **TensorFlow (** Deep learning frameworks for building and training neural networks, which can be useful for more advanced detection models.)
3. **Data Processing and Analysis**:
   - **Pandas** (For data manipulation and analysis.)
   - **NumPy** (For numerical computing.)
4. **Natural Language Processing (NLP)**:
   - **NLTK (Natural Language Toolkit)** (NLP libraries for text analysis can help identify phishing characteristics in text.)
5. **Web Scraping**:
   - **Beautiful Soup** or **Scrapy** (For extracting data from websites, including WHOIS data.)

6. **Database**:
   - **SQL or NoSQL database**: To store and manage historical data on registered domains, if necessary.
7. **Cloud Services**:
   - **AWS, Azure, or Google Cloud**: These cloud platforms offer scalable infrastructure and machine learning services that can be utilized for training and deploying AI/ML models.
8. **Version Control**:
   - **Git**: For version control and collaboration among team members.
9. **Web Development (Optional)**:
   - **Flask or Django**: If you want to create a web-based interface for users to interact with the system.
10. **Deployment**:
    - **Docker**: For containerization of the application.
    - **Kubernetes**: For container orchestration.
    - **Heroku, AWS Elastic Beanstalk, or similar**: For deploying web applications if needed.
11. **Monitoring and Logging**:
    - **ELK Stack (Elasticsearch, Logstash, Kibana)** or similar tools for monitoring and logging the system's behavior and performance.
12. **Security**:
    - Consider implementing security best practices, as this project deals with sensitive information. Regularly update libraries and dependencies to patch vulnerabilities.

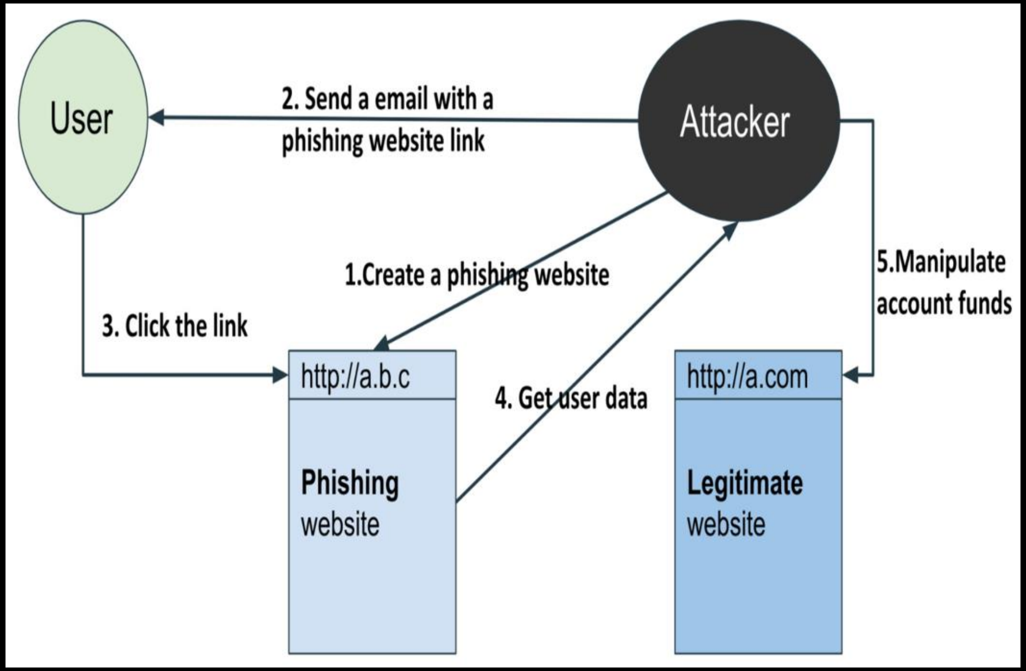


13. **Collaboration and Communication**:
    - Tools like **Slack** or **Microsoft Teams** (for team communication and collaboration.)
14. **Testing**:
    - Use unit testing and potentially integration testing frameworks (to ensure the system's reliability)
15. **Documentation**:
    - Tools like **Sphinx** (for generating documentation)
16. **Project Management**:
    - Tools like **Jira, Trello, or Asana** (to manage project tasks and timelines)

Use Cases

These use cases highlight the versatility and importance of an intelligent system for detecting phishing domains. It can be applied across various domains to enhance cybersecurity and protect users and organizations from the ever-evolving threat of phishing attacks.

1. **Email Security**:
   - The system can be integrated into email security solutions to scan incoming emails for suspicious links and domains. It can prevent phishing emails from reaching users' inboxes by identifying and blocking malicious links.
2. **Web Browsing Protection**:
   - Users can install browser extensions or add-ons that utilize the system to warn them when they visit potentially phishing websites. This provides real-time protection while browsing the web.
3. **Endpoint Security**:
   - The system can be integrated into endpoint security solutions, such as antivirus software, to scan and block attempts to download malicious payloads from phishing websites.
4. **Corporate Network Security**:
   - Enterprises can deploy the system at the network level to detect and block phishing domains, safeguarding employees and corporate data from phishing attacks.
5. **User Education and Training**:
   - The system can be used to educate users about phishing threats by providing real-world examples. When users attempt to visit a phishing site, they can receive educational messages about the dangers of phishing.



6. **Security Research**:
    - Security researchers and organizations can use the system to gather data on phishing trends and techniques. It can help understand how phishing attacks evolve and identify new threat vectors.
7. **WHOIS Database Analysis**:
    - The system can automate analyzing WHOIS data to identify newly registered domains that may be suspicious. This can be valuable for cybersecurity professionals and organizations monitoring domain registrations.
8. **Incident Response**:
    - In the event of a successful phishing attack, the system can assist in incident response by providing information about the malicious domains and helping organizations take corrective actions.
9. **Government and Law Enforcement**:
    - Government agencies and law enforcement can use the system to proactively identify and take down phishing websites, reducing cybercrime.
10. **Phishing Prevention Services**:
    - Companies specializing in cybersecurity services can offer phishing prevention to clients, using the intelligent system as part of their offering.
11. **Continuous Learning and Improvement**:
    - The system can continuously learn and adapt to new phishing techniques and patterns, improving its accuracy.
12. **Third-Party Integration**:
    - Third-party security vendors can integrate the system into their products, enhancing the overall security posture of their customers.

Dependencies

1. **Data Sources**:
   - Availability of reliable and up-to-date data sources, such as WHOIS databases and domain registration records, is crucial for the system to identify newly registered domains accurately.
2. **Data Quality**:
   - The data quality used for training and testing the machine learning models is essential. Inaccurate or incomplete data can lead to false positives or negatives in phishing detection.
3. **Access to Computational Resources**:
   - Appropriate computational resources, including high-performance servers and GPUs, are required to efficiently train and deploy machine learning models.
4. **Expertise in AI/ML**:
   - The project requires expertise in machine learning and artificial intelligence to develop effective phishing detection models.
5. **Security Knowledge**:
   - Understanding cybersecurity and phishing techniques is crucial to designing effective detection algorithms.
6. **Web Scraping Tools**:
   - Dependence on web scraping tools for collecting WHOIS and other relevant data. Changes in the structure of websites can break scraping routines.
7. **External APIs**:
   - If the system relies on external services or APIs (e.g., for real-time updates or threat intelligence feeds), their availability and reliability are dependencies.
8. **Regulatory Compliance**:
   - Compliance with data privacy and security regulations, such as GDPR, HIPAA, or industry-specific standards, may be necessary, depending on the project's scope and data handling.

Show-stoppers

1. **Data Unavailability**:
   - If access to WHOIS databases or other critical data sources becomes restricted or limited, it could severely impact the system's ability to detect phishing domains.
2. **Inadequate Training Data**:
   - Lack of a sufficient and diverse training dataset can hinder the development of effective machine learning models. Phishing patterns evolve rapidly, so staying up-to-date is critical.
3. **Model Performance**:
   - If the machine learning models do not perform well in real-world scenarios, the system can be ineffective at detecting phishing domains, leading to false positives or missed threats.
4. **False Positives/Negatives**:
   - Striking the right balance between false positives and false negatives is a significant challenge. High false positive rates can lead to user frustration, while high false negatives can put users at risk.
5. **Scalability**:
   - As the number of newly registered domains increases over time, the system must be able to scale to handle the growing workload efficiently.
6. **Maintenance and Updates**:
   - Failure to regularly update and maintain the system, including its data sources and machine learning models, can decrease effectiveness over time.
7. **Legal and Ethical Issues**:
   - Violating legal or ethical data collection, usage, or sharing boundaries can lead to legal actions and reputational damage.
8. **Budget Constraints**:
   - Lack of adequate funding or resources can hinder the project's progress and limit the ability to acquire necessary tools and data sources.
9. **Adaptation to New Phishing Techniques**:
   - Phishing techniques are constantly evolving. Failure to adapt the system to new tactics and evasion methods can make it obsolete.
10. **User Acceptance**:
    - If users do not trust or use the system, its effectiveness in protecting against phishing attacks can be compromised.

Team Member Details

Team Leader Name: Bhuvan Deep Sagar	Stream: CSE	Year: III	Branch: B.Tech
Team Member 1 Name: Vievaan Khattar	Stream: CSE	Year: III	Branch: B.Tech
Team Member 2 Name: Zaid Khan	Stream: CSE	Year: III	Branch: B.Tech
Team Member 3 Name: Jigyaasa Meena	Stream: CSE	Year: III	Branch: B.Tech
Team Member 4 Name: Purvi Rathore	Stream: CSE	Year: III	Branch: B.Tech
Team Member 5 Name: Iptisha Dugtal	Stream: CSE	Year: III	Branch: B.Tech