**MALICIOUS URL ANALYZING TOOL**

**Logo

Description automatically generated**

**Submitted to: Prof. B. Mohinder Singh**

**Submitted by: Kshitij Pathak(22BCE20232)**

**:Adham Ansari(22BCE8873)**

**SLOT: L14+L15**

**Abstract**

The URL Analysis Tool is developed to identify and assess the safety of web URLs by evaluating them using multiple threat intelligence sources. It helps detect malicious URLs, phishing threats, and suspicious domains. The tool offers a graphical interface for easy access and provides verdicts based on vendor reputation and threat severity. In an era where cyber threats are increasing at an alarming rate, this tool serves as a proactive defense mechanism against potential threats posed by unsafe URLs.

**Introduction**

With the increasing digital threats such as phishing, malware, ransomware, and drive-by downloads, malicious URLs have become one of the most common entry points for cyber-attacks. Even tech-savvy users can fall victim to seemingly legitimate links that mask harmful payloads. Thus, a secure and reliable system to analyze and verify the authenticity of URLs is essential for individual users, educational institutions, and cybersecurity professionals.

This project focuses on URL security by using multiple API services from trusted threat intelligence platforms and analyzing domain metadata, DNS records, and WHOIS data. The tool simplifies the entire process through a user-friendly GUI, ensuring accessibility for users from both technical and non-technical backgrounds.

**Existing or Related Tools**

* **Google Safe Browsing:** Flags URLs associated with phishing and malware. Integrated into many browsers for background checks.
* **VirusTotal:** Aggregates results from over 70 antivirus scanners and domain blacklisting services.
* **PhishTank:** A community-based phishing verification tool providing a publicly accessible phishing database.
* **URLVoid:** Offers basic URL reputation and blacklist checks.

**Drawbacks of Existing Tools**

* Most tools are browser-based or cloud-based and require continuous internet access.
* Limited to a single detection engine, increasing the risk of false positives or negatives.
* Require technical knowledge or browser plugins to operate effectively.
* Lack of consolidated threat intelligence from multiple trusted sources.
* Some tools have a complex interface, making them less suitable for non-technical users.

**Proposed Tool**

**What it's used for:** The tool is designed for real-time evaluation of web URLs to determine if they pose a potential threat. It consolidates data from various APIs, performs domain analysis, and provides a final verdict along with recommendations. This makes it a valuable addition to cybersecurity assessments and educational demonstrations.

**Core Methodology (Algorithm/Pseudocode):**

1. Take URL input from the user

2. Check if the format is valid

3. Use APIs to analyze:

- Google Safe Browsing

- VirusTotal

- IOC Search

4. Extract domain/IP using `whois`, `socket`, and DNS lookup

5. Retrieve domain age and registration details

6. Consolidate data:

- Vendor reputation score

- Threat type and severity

7. Generate a final verdict (Safe or Malicious)

8. Display results in GUI with recommendation

**Explanation:** The algorithm starts by validating the format of the input URL. Once validated, it sends requests to threat intelligence APIs which return various reports and flags. The domain analysis modules then extract the WHOIS record to determine domain ownership, age, and other metadata. The DNS and IP lookup modules provide insight into the hosting location and associated risk. All these data points are cross-verified to avoid false positives. The final verdict is derived based on two primary criteria: the **reputation of the vendors** who flagged the URL, and the **severity of the threat** (e.g., phishing, malware, redirector, etc.).

**System Requirement Specification (Minimum)**

* **Operating System:** Windows 10 or higher (64-bit)
* **RAM:** Minimum 4GB (8GB recommended for smooth performance)
* **Python Version:** 3.8 or above
* **Required Python Libraries:**
  + requests
  + tkinter
  + whois
  + socket
  + json
* **Internet Connection:** Required to connect to external APIs
* **Screen Resolution:** Minimum 1280x720 for proper GUI display

**Results and Discussions**

**Input Screenshot:** [Insert Screenshot of GUI URL input here]

**Output Screenshot:** [Insert Screenshot showing the results - Safe/Malicious verdict with analysis]

**Analysis:** The screenshots demonstrate the working of the tool with different types of URLs. For legitimate URLs, the APIs return clean reports with no threats, a high reputation score, and long domain age. For malicious URLs, the tool highlights flagged reports, notes domain abnormalities (such as recently registered domains or suspicious DNS configurations), and offers a clear verdict for the user.

**Verdict Criteria:** The final safety decision of the tool is based on two major factors:

* **Vendor Reputation**: Trusted vendors like Google Safe Browsing, VirusTotal, and others provide detection data. If multiple vendors mark the URL as unsafe, the verdict leans towards 'Malicious'.
* **Threat Severity**: Each threat report includes the type of threat detected. URLs marked for phishing, malware delivery, or scams are treated as high risk.

This combination improves the accuracy of results and ensures reliability for end users.

**Conclusion**

The URL Analysis Tool efficiently identifies potentially dangerous URLs by combining various threat intelligence resources. It simplifies threat detection for non-technical users through a graphical interface. By considering vendor reputation and threat severity, the tool provides a balanced verdict system that reduces false positives. This project demonstrates how integrating multiple security services with clear algorithms can provide a robust URL screening solution.

Further enhancements may include real-time browser extensions, visual data dashboards, and AI-based detection models to identify zero-day threats or unreported phishing attempts.

**References**

* Google Safe Browsing API: https://developers.google.com/safe-browsing
* VirusTotal API: https://www.virustotal.com/gui/home/search
* IOC Search: https://www.iocsearch.com/
* WHOIS Lookup: https://who.is
* URLVoid: https://www.urlvoid.com/

**GitHub Repository:** https://github.com/6tapped/URL-Analyzer