MALICIOUS URL DETECTOR

REPORT

Name: Shashank Molugu

Roll No:1601-23-737-052

Date: 01/09/2025

Github Repository: https://github.com/ShashankM18/CyberSecurity-Assignment1

ABSTRACT

Downloading software from the internet is a common activity, but many users fall victim to malicious links disguised as legitimate download sources. Attackers often use techniques such as domain spoofing, typosquatting, or malicious top-level domains (TLDs) to trick users into downloading harmful files.

This project proposes a Verified Download Link Tool that checks whether a given software download link is official/safe or suspicious using a set of lightweight cybersecurity heuristics. The system verifies links based on an official domain whitelist, suspicious TLD detection, HTTPS checks, typosquatting similarity, and shady path keywords.

The tool is implemented in Python with a Gradio-based user interface, allowing users to input any download URL and instantly receive an analysis. The system provides verdicts such as Likely Official, Unknown, or Suspicious along with a risk score and reasoning. This project demonstrates how simple, rule-based detection can enhance user awareness and safety in software downloads.

APPROACH

1. Problem Definition:
   - Determine whether a software download link is safe or suspicious by analyzing the domain, protocol, and structure of the URL.

2. Heuristic Rules Implemented:
   - Whitelist Matching: Official domains list (e.g., python.org, videolan.org).
   - HTTPS Verification: Non-HTTPS links are flagged as risky.
   - Suspicious TLDs: Domains with risky TLDs (.zip, .xyz, .top).
   - Typosquatting Detection: Levenshtein distance to detect similar-looking domains.
   - Path Keyword Check: Flags keywords like 'crack', 'serial', 'license-key'.

3. Implementation:
   - Core logic in Python.
   - Configurable whitelist.yml and suspicious TLD list.
   - Exportable results in JSON/CSV format.

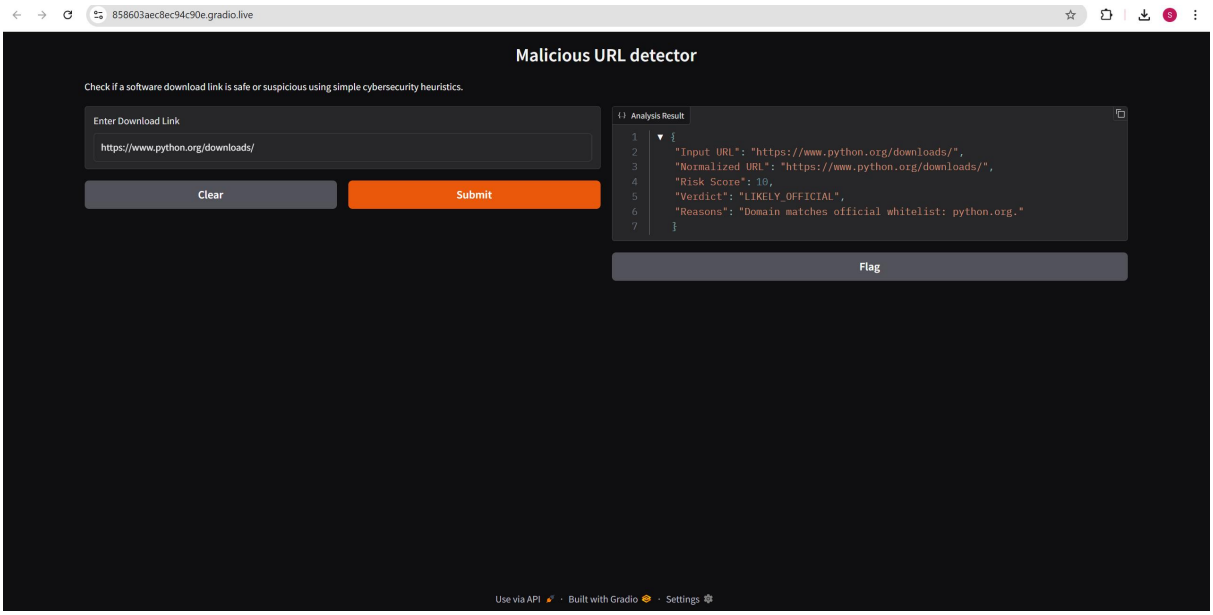
4. User Interface:
   - Integrated with Gradio for an interactive browser-based UI.
   - Users can enter a URL and view structured results.
   - Gradio generates a public shareable link for demos.

RESULTS

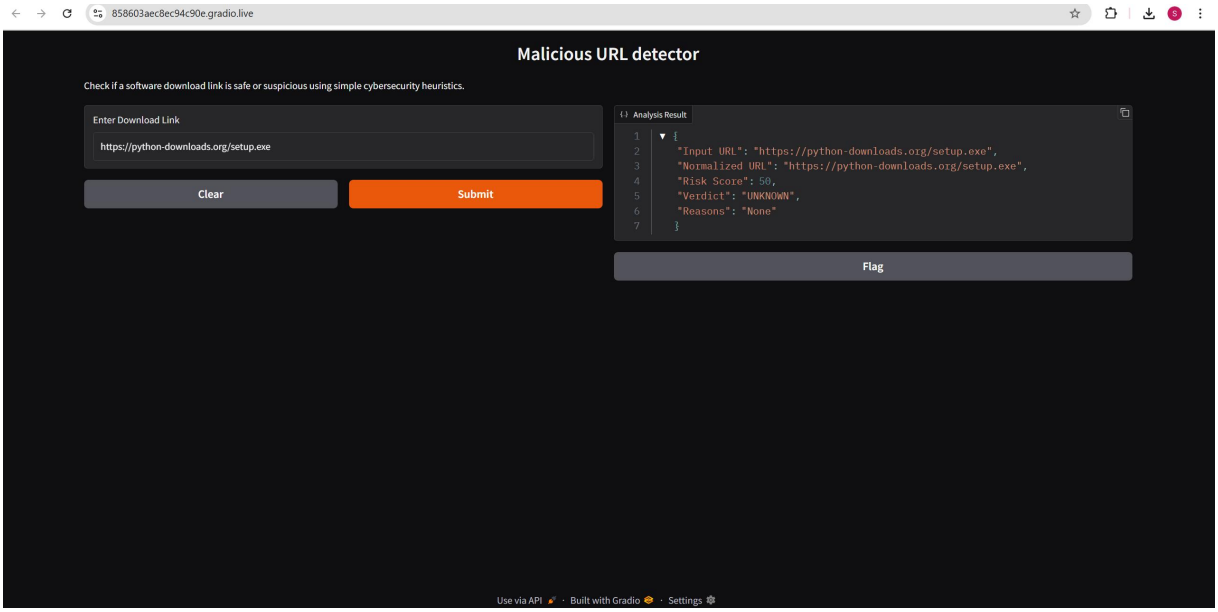
The Gradio UI made the system interactive and easy to use.

Sample test results:

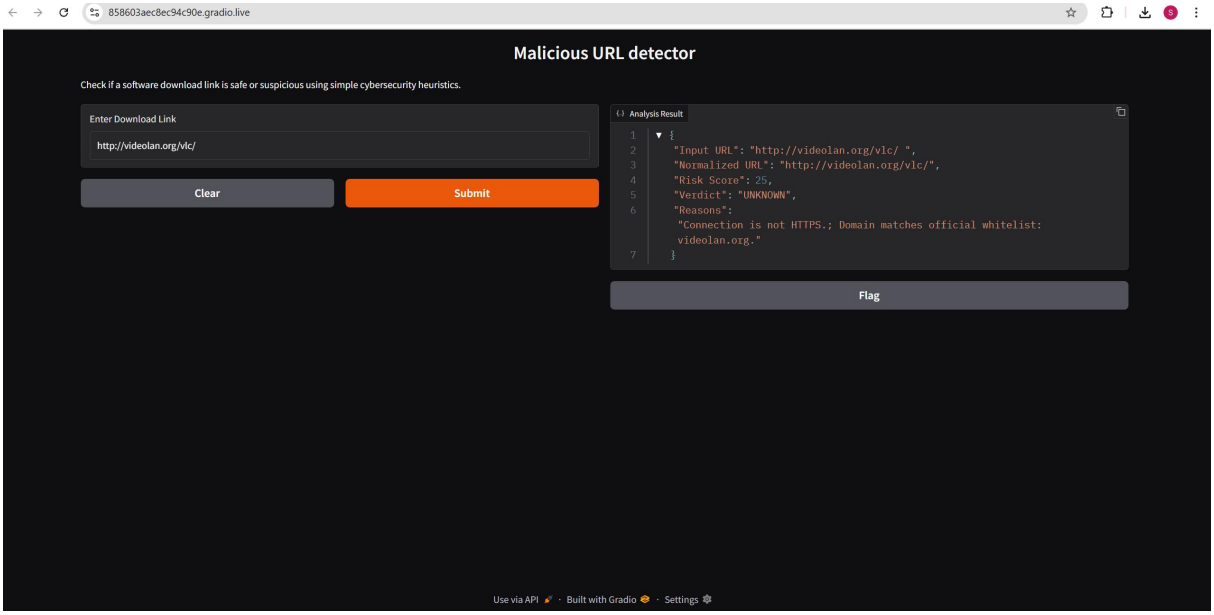
CASE 1:
- Input: https://www.python.org/downloads/ → Verdict: Likely Official (whitelist match, HTTPS).



CASE 2:
- Input: https://python-downloads.org/setup.exe → Verdict: Suspicious (typosquatting, non-whitelisted domain).



CASE 3:
- Input: http://videolan.org/vlc/ → Verdict: Unknown (official domain but non-HTTPS).

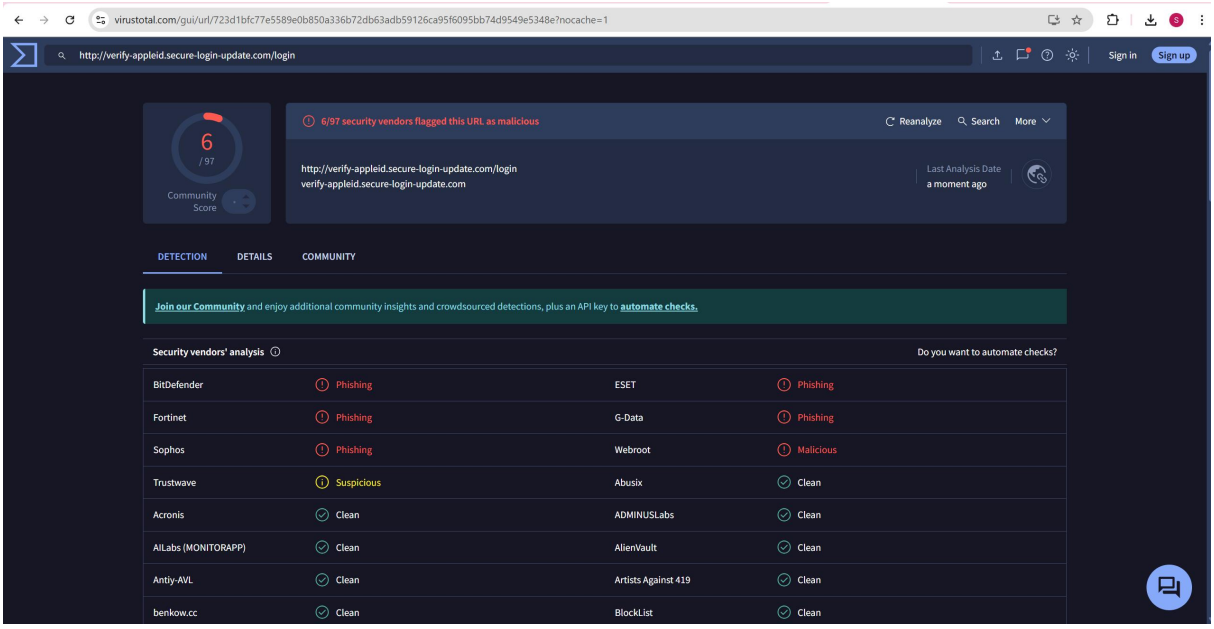


This shows the tool effectively detects risky download links using lightweight heuristics.

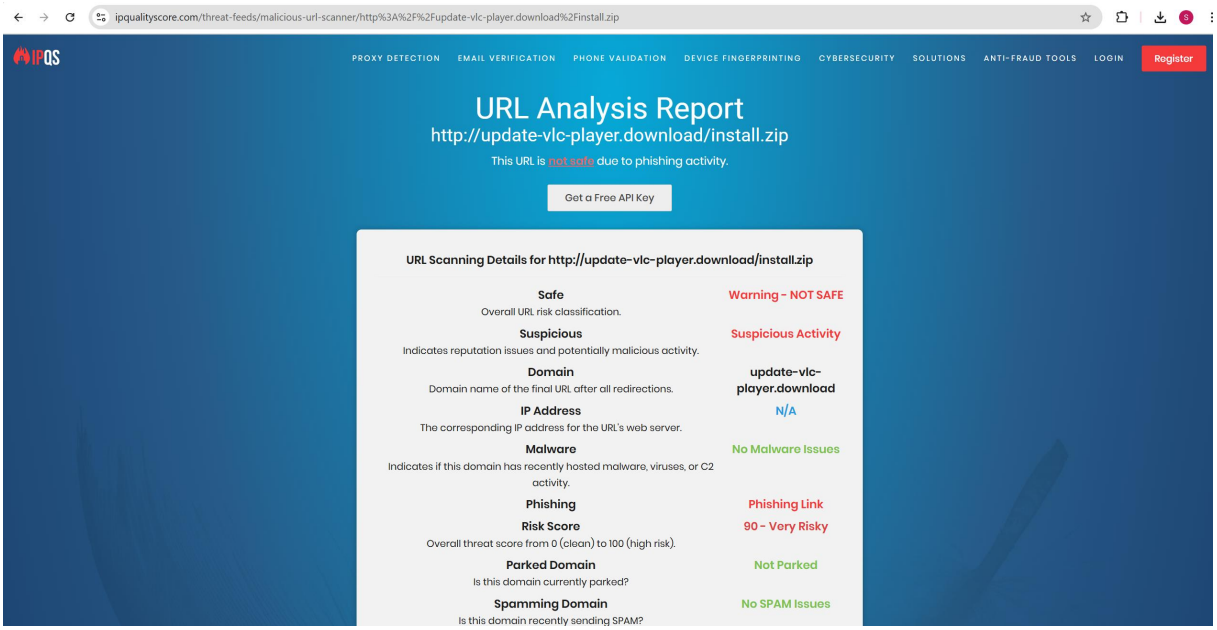
EXISTING TOOLS EXPLORED

During this project, similar tools and platforms were explored for comparison:

VirusTotal – Multi-engine URL and file scanner.



IPQS – Check suspicious links with the IPQS malicious URL scanner. Real-time results detect phishing links and malware domains with accurate, deep machine learning analysis.



Compared to these, the proposed tool is lightweight, open-source, beginner-friendly, and specifically targeted at software download verification.

CONCLUSION

This project demonstrates how a simple, rule-based approach can help users verify whether software download links are safe. By combining domain whitelist matching, TLD analysis, HTTPS checks, and keyword detection, the tool provides quick and understandable verdicts.

Integration with Gradio makes it user-friendly and shareable, allowing anyone to test URLs without technical expertise.

Future improvements include integrating threat intelligence feeds (PhishTank, URLHaus) and adding checksum verification of downloaded files for stronger assurance.