

Sus-detector: Phishing Website Detection System Using URL and SSL Features

Course Title: Computer Security Lab

Course Code: CSE-4744

Institution: Department of CSE, IIUC

Team Members

- C213078 - Shehabudowllah Rakib
- C213081 - Md. Asiful Islam
- C213094 - Kazi Irfanul Karim

Instructor: Md. Ziaul Haque, Adjunct Faculty, Dept of CSE, IIUC

1. Introduction and Background

Phishing attacks are a major threat in today's digital landscape, where malicious websites imitate legitimate platforms to trick users into revealing sensitive information. These attacks often rely on deceptive URLs, fake certificates, and misleading web behaviors to gain users' trust.

Sus-detector is a lightweight web-based system that detects potentially harmful websites by analyzing the structure and security characteristics of a given URL. It evaluates various factors, including SSL certificate presence, domain details, and redirection patterns, to determine whether a site is safe for users. This proactive approach enhances online safety by identifying phishing attempts before users fall victim.

2. System Architecture / Design

The architecture of the system is modular and consists of the following components:

- **Frontend (User Interface):** Developed using HTML/CSS for simple input and result display.
- **Flask Backend:** Handles routing, feature extraction, and model integration.
- **Feature Extraction Module:** Extracts 30 URL-based and certificate-based features.
- **ML Model Module:** Pre-trained machine learning model performs the classification.
- **Prediction Output:** Returns "Legitimate" or "Phishing" based on prediction.

Workflow Summary:

1. User inputs a website URL.
 2. System checks for SSL/TLS certificate.
 3. 30 features are extracted from the URL and web metadata.
 4. Features are passed into the ML model for classification.
 5. Result is displayed to the user.
-

3. Implementation Details

The project was developed using the following technologies:

- **Language & Framework:** Python with Flask
- **Frontend:** HTML, CSS (basic)
- **Machine Learning:** Tested several pre-trained machine learning model and found Gradient Boosting Classifier correctly classify URL up to 97.4% respective classes
- **Feature Extraction:** Custom Python scripts to gather lexical and behavioral indicators

Features extracted include:

- **UsingIp()** – Checks if the URL uses an IP address instead of a domain, which is a common trait in phishing.
- **longUrl()** – Flags very long URLs which are often used to obfuscate intent.
- **shortUrl()** – Detects use of URL shortening services which can hide malicious links.
- **symbol()** – Identifies use of '@' symbol in URLs, often used to redirect users unknowingly.
- **redirecting()** – Checks for excessive "/" which may indicate multiple redirections.
- **prefixSuffix()** – Flags hyphens in domain names (e.g., www.bank-secure.com), commonly used in phishing.
- **SubDomains()** – Looks for excessive subdomains used to mimic legitimate sites.
- **Hppts()** – Detects if "https" is part of the URL path rather than indicating true security.
- **DomainRegLen()** – Short domain registration length may indicate a throwaway phishing site.

- **Favicon()** – Checks if the favicon is loaded from a different domain, which can be suspicious.
- **NonStdPort()** – Flags URLs using ports other than 80 or 443, which may indicate malicious intent.
- **HTTPSDomainURL()** – Checks if HTTPS is correctly implemented and part of the main domain.
- **RequestURL()** – Measures whether external objects (e.g., images, scripts) are loaded from different domains.
- **AnchorURL()** – Evaluates anchor tags; phishing sites often redirect anchors to suspicious domains.
- **LinksInScriptTags()** – Detects links embedded in script tags rather than visible text.
- **ServerFormHandler()** – Checks if form data is submitted to unknown or untrusted servers.
- **InfoEmail()** – Flags presence of “mailto:” links that could harvest user email.
- **AbnormalURL()** – Identifies URLs that do not match standard domain formats or structures.
- **WebsiteForwarding()** – Flags excessive HTTP redirects which are used to hide final destinations.
- **StatusBarCust()** – Detects manipulation of the browser status bar, a trick to mislead users.
- **DisableRightClick()** – Checks if right-click functionality is disabled to hide phishing intent.
- **UsingPopupWindow()** – Identifies use of popup windows, commonly used in social engineering.
- **IframeRedirection()** – Detects presence of iframes that could redirect users or hide malicious content.
- **AgeofDomain()** – Very young domains are often red flags for phishing sites.
- **DNSRecording()** – Determines if the domain has valid DNS records; phishing domains often don't.
- **WebsiteTraffic()** – Evaluates popularity and traffic of the domain using third-party sources.
- **PageRank()** – Low or zero PageRank is a sign of untrustworthy or unknown websites.
- **GoogleIndex()** – Flags websites that are not indexed by Google as potentially suspicious.

- **LinksPointingToPage()** – A low number of external backlinks may signal an untrusted domain.
- **StatsReport()** – Uses aggregated online threat intelligence reports (e.g., PhishTank, VirusTotal).

The dataset used includes both legitimate and phishing URLs with labeled classifications, preprocessed and split for training and testing purposes.

4. Screenshots / Outputs

Sus-Detector

[Home](#)[About](#)[Help](#)

Download Guide

Advanced AI-Powered Detection

Detect Phishing Websites Before They Detect You

Protect yourself from malicious websites with our AI-powered phishing detection system. Simply enter a URL and get instant security analysis.

Enter website URL (https://)

Scan Now

97.4%
Accuracy Rate

1.1k+
URLs Scanned

24/7
Protection

Frequently Asked Questions

Everything you need to know about phishing and how to protect yourself

What is phishing?

Why should I care about phishing?

Types of Phishing Attacks

How to protect yourself?

About Sus-Detector

Sus-Detector is an innovative project that leverages machine learning algorithms to identify and prevent phishing attacks. Our system analyzes **SSL/TLS Certificates**, **URL patterns**, **website content**, and **behavioral indicators** to provide real-time protection against malicious websites.

Development Team

Shehabudowilah
Rakib
C213078

Md. Asifur Islam
C213081

Kazi Irfanul Karim
C213094

Course Information

Course: CSE-4744 - Computer Security Lab

Instructor: Md. Ziaul Haque, Adjunct Faculty, Dept of CSE, IJUC


Take Google's Phishing Test

© 2024 Sus-Detector: Computer Security Lab Project - All Rights Reserved

🛡️ Advanced AI-Powered Detection

Detect Phishing Websites Before They Detect You

Protect yourself from malicious websites with our AI-powered phishing detection system. Simply enter a URL and get instant security analysis.




Enter website URL (https://e)

Scan Now 🔍

https://web.whatsapp.com/

Website is Safe to use

This website appears safe to use.

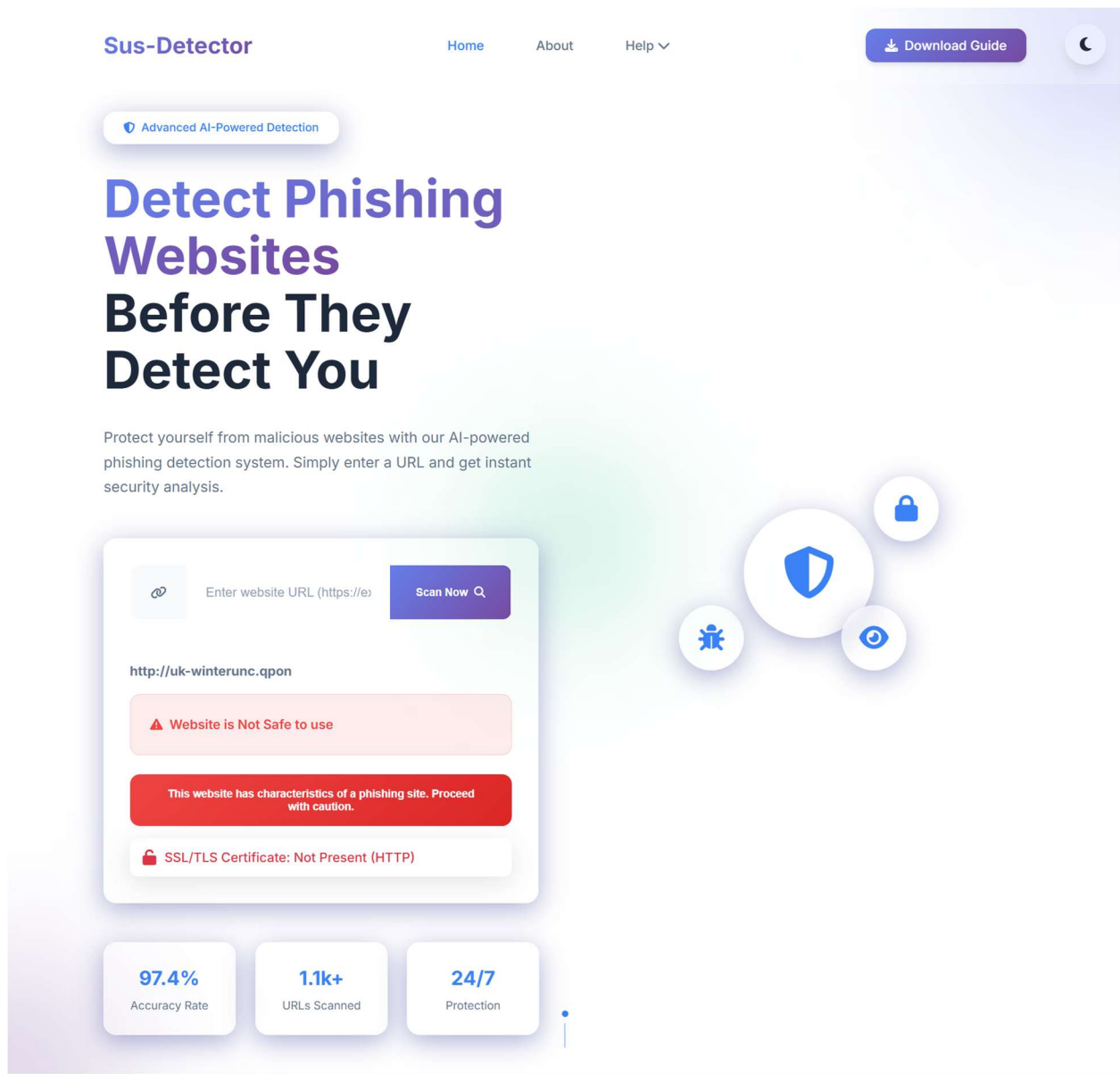
 SSL/TLS Certificate: Present (HTTPS)



97.4%
Accuracy Rate

1.1k+
URLs Scanned

24/7
Protection



5. Security Analysis

Sus-detector applies multiple layers of verification beyond simple keyword matching. It checks for structural inconsistencies in the URL and verifies certificate presence and domain history, making it resilient against common phishing techniques such as:

- IP-based URLs
- Suspicious redirects
- Fake SSL implementation

- Excessive subdomain usage
- Abnormal domain age

However, the system relies on static feature extraction and may not detect zero-day phishing attacks or dynamically changing threats without periodic model retraining.

6. Conclusion and Future Work

Sus-detector successfully demonstrates how machine learning combined with URL and SSL analysis can serve as an effective phishing detection method. The system offers a web-based, lightweight interface suitable for real-time URL checks.

Future enhancements:

- Integration with a real-time browser plugin
- Deep learning model for dynamic page content analysis
- Periodic model retraining using recent datasets
- Blacklist/whitelist integration