Phishing Website Detection System – Project Documentation

Date: November 16, 2025
Author: Laiba Saleem

Table of Contents

1. Executive Summary

The **Phishing Website Detection System** is a cybersecurity solution designed to protect users from phishing attacks. It leverages machine learning algorithms to analyze URLs in real-time and classify them as **legitimate** or **malicious**. The system is accessible via a **web application** and a **Chrome browser extension**, providing users with immediate threat alerts.

Key Features:

- Real-time URL analysis
- Machine learning powered (Multinomial Naive Bayes)
- RESTful API for integration
- Intuitive UI with visual feedback
- Robust URL preprocessing and error handling

Business Value:

- Reduces exposure to phishing threats
- Enhances security for individual and enterprise users
- Supports rapid detection without user browsing risk

2. Project Overview

Problem Statement:
Phishing attacks are increasingly sophisticated. Traditional methods often fail to detect newly created malicious sites. Users require instant and reliable detection.

Solution Approach:
The system uses ML models trained on labeled URL datasets to predict phishing attempts. It offers both a browser extension for real-time protection and a web app for manual checks.

Target Users:

- General internet users

- Organizations requiring employee protection
- Security professionals and developers

---

3. System Architecture

Architecture Overview:

```
User Interfaces (Chrome Extension)
          |
          ▼
      Flask Backend API
          |
          ▼
    URL Preprocessing → Feature Extraction → ML Model → Response Formatter
          |
          ▼
      JSON Response
```

Components:

- **Frontend:** HTML/CSS/JS web app, Chrome extension popup UI
- **Backend:** Flask API (`/api/check`) with URL preprocessing and ML inference
- **Machine Learning:** TF-IDF vectorizer + Multinomial Naive Bayes classifier
- **Dataset:** Labeled URLs (phishing vs legitimate)

4. Technical Specifications

Backend: Python 3.8+, Flask, Flask-CORS
Machine Learning: scikit-learn, pandas, numpy, NLTK
Frontend: HTML5, CSS3, JavaScript, Bootstrap, Tailwind CSS
Tools: Git, virtual environments, Jupyter notebooks
System Requirements: Windows/macOS/Linux, 2GB RAM minimum

5. Installation & Setup

1. **Clone Repository**

```
git clone
https://github.com/LaibaSaleem043/phishing-website-detection.git
cd phishing-website-detection
```

2. **Create Virtual Environment**

```
python -m venv venv
source venv/bin/activate  # macOS/Linux
venv\Scripts\activate     # Windows
```

3. **Install Dependencies**

```
pip install -r requirements.txt
```

4. **Download NLTK Data**

```
python -c "import nltk; nltk.download('punkt');
nltk.download('stopwords')"
```

5. **Run Flask Server**

```
python app.py
```

6. API Documentation

Base URL: `http://localhost:5000`
Endpoint: `/api/check` (POST)

Request:

```
{
  "url": "https://www.example.com"
}
```

Success Response:

```
{
  "success": true,
  "is_phishing": false,
  "message": "Website is safe",
  "url": "https://www.example.com",
  "cleaned_url": "example.com"
}
```

Error Response (400/500):

```
{
  "success": false,
  "error": "Invalid URL",
  "message": "Please provide a valid URL"
}
```

7. User Guide

Web App Usage:

1. Start Flask server
2. Visit `http://localhost:5000`
3. Enter URL → click Submit → view result

Chrome Extension Usage:

1. Enable Developer mode in Chrome (`chrome://extensions`)
2. Load unpacked extension (`chrome-extension` folder)

3. Check current page or input URL manually
4. API result displayed in popup

8. Development Details

Project Structure:

```
phishing-website-detection/
├── app.py
├── phishing.pkl
├── vectorizer.pkl
├── phishing_mnb.pkl
├── requirements.txt
├── templates/index.html
├── chrome-extension/
└── Dataset/phishing_site_urls.csv
```

Core Logic:

- `check_url_api()`: preprocesses URL → vectorizes → ML prediction → returns JSON

9. Machine Learning Model

- **Algorithm:** Multinomial Naive Bayes
- **Features:** TF-IDF vectorized URL strings
- **Training Dataset:** `phishing_site_urls.csv`
- **Performance:** ~86% accuracy on validation dataset

10. Testing & Quality Assurance

- Unit tests for preprocessing, ML inference, API endpoints

- Manual testing with known phishing/legitimate URLs
- Cross-browser extension testing

11. Deployment Guide

- Deploy Flask API on cloud server (AWS, GCP, Azure)
- Update Chrome extension API URL
- Ensure HTTPS for production
- Optional: Docker containerization

12. Troubleshooting

- **Error: Flask not starting** → check dependencies, virtual environment
- **Extension shows no result** → verify server running at specified URL
- **Model file missing** → ensure `phishing.pkl` and `vectorizer.pkl` exist

13. Security Considerations

- Validate all inputs to prevent injection
- Use HTTPS in production
- Limit API access if exposed externally
- Sanitize logs to avoid storing sensitive URLs

14. Future Enhancements

- Add deep learning models for higher accuracy
- Real-time monitoring of user browsing

- Browser plugins for Firefox, Edge
- Integration with email filters for phishing links

15. References / Appendix

- Dataset: Phishing Site URLs
- Flask Documentation: https://flask.palletsprojects.com/
- scikit-learn: https://scikit-learn.org/