```python
from flask import Flask, request, render_template, redirect, url_for, flash
import os
import subprocess
import yara
import PyPDF2
from werkzeug.utils import secure_filename
import google.generativeai as genai

app = Flask(__name__)
app.secret_key = "your_secret_key"
app.config['UPLOAD_FOLDER'] = "uploads"
os.makedirs(app.config['UPLOAD_FOLDER'], exist_ok=True)

# YARA rule for basic PDF threats
YARA_RULES = r"""
rule SuspiciousPDF {
    meta:
        description = "Detects suspicious PDFs with JavaScript or embedded files"
    strings:
        $javascript = /\/JavaScript|\/JS/
        $embedded = /\/EmbeddedFile/
    condition:
        $javascript or $embedded
}
"""

def scan_pdf_with_yara(file_path):
    rules = yara.compile(source=YARA_RULES)
    with open(file_path, "rb") as file:
        matches = rules.match(data=file.read())
    return matches

def analyze_with_pdfid(file_path):
    try:
        result = subprocess.run(
            ["python3", "pdfid.py", file_path],
            capture_output=True,
            text=True
        )
        if result.returncode == 0:
            return result.stdout
        else:
            return "Error running PDFiD."
    except Exception as e:
        return f"Exception while running PDFiD: {e}"

# Configure Gemini API with your API key
genai.configure(api_key="AIzaSyBfU2nHcesPIYnBTp4_w6N6CG-K1h2Y5VA")

def analyze_message_with_gemini(message):
    """
    Uses Gemini AI to analyze a message for phishing and suspicious behavior.
    """
    try:
        prompt = (
            f"Analyze the following text and determine if it contains any suspicious elements,
"
            f"potential phishing indicators, or fraudulent attempts. Provide a detailed
analysis:\n\n"
            f"{message}"
        )
        model = genai.GenerativeModel("gemini-1.5-flash")
        response = model.generate_content(
```

```
            prompt
        )
        return response.text  # Return the text result
    except Exception as e:
        return f"Error analyzing the message: {e}"


@app.route('/')
def index():
    return render_template("index.html")

@app.route('/upload', methods=['POST', 'GET'])
def upload_file():
    if request.method == 'POST':
        if 'pdf_file' not in request.files:
            flash("No file uploaded!")
            return redirect(url_for('upload_file'))

        file = request.files['pdf_file']
        if file.filename == '':
            flash("No selected file!")
            return redirect(url_for('upload_file'))

        filename = secure_filename(file.filename)
        file_path = os.path.join(app.config['UPLOAD_FOLDER'], filename)
        file.save(file_path)

        yara_matches = scan_pdf_with_yara(file_path)
        pdfid_output = analyze_with_pdfid(file_path)

        try:
            with open(file_path, "rb") as pdf_file:
                pdf_reader = PyPDF2.PdfReader(pdf_file)
                metadata = pdf_reader.metadata
                num_pages = len(pdf_reader.pages)
        except Exception as e:
            metadata = {"Error": str(e)}
            num_pages = "Unknown"

        report = {
            "filename": filename,
            "metadata": metadata,
            "num_pages": num_pages,
            "yara_matches": [match.rule for match in yara_matches],
            "pdfid_output": pdfid_output
        }

        return render_template("pdfscan.html", report=report)

    return render_template("upload.html")

@app.route('/message', methods=['GET', 'POST'])
def analyze_message():
    if request.method == 'POST':
        message = request.form.get('message')
        if not message:
            flash("No message provided!")
            return redirect(url_for('analyze_message'))

        analysis = analyze_message_with_gemini(message)
        return render_template("message_result.html", message=message, analysis=analysis)

    return render_template("message_input.html")
```

```
if __name__ == '__main__':
    app.run(debug=True)
```