Dhanashree Badgujar

Phishing scanner

Overview:

The code is a Phishing Link Scanner. It checks a given URL to determine if it's potentially a phishing website or not. The scanner does this by: -

1. Checking the URL for suspicious patterns, hence the code: -

```
def is_suspicious_url(url):
  suspicious_patterns = [
    r"\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}",  # IP address in URL
    r"\.xyz",  # Certain TLDs often used by phishing sites
    r"\.ru",   # .ru is often associated with phishing
    r"login",  # Common word used in phishing sites
  ]
  for pattern in suspicious_patterns:
    if re.search(pattern, url):
      return True
  return False
```

2. Looking at the domain's registration details., hence the code (function): -

```
def get_domain_info(url):
  domain = urlparse(url).netloc
  try:
    w = whois.whois(domain)
    if w.creation_date is None or (type(w.creation_date) == list and
len(w.creation_date) == 0):
      return False  # New domain, possibly suspicious
    return True
  except Exception as e:
    return False  # Could not retrieve domain info, likely suspicious
```

3. Checking if the URL is flagged by a reputable service (PhishTank).

```
# Check if the URL is in PhishTank (API check)
def check_with_phishtank(url):
  phishtank_api_url = "https://checkurl.phishtank.com/checkurl/"
  headers = {"Content-Type": "application/x-www-form-urlencoded"}
  data = {"url": url, "format": "json", "app_key": "your_api_key_here"}
  response = requests.post(phishtank_api_url, data=data, headers=headers)
  result = response.json()
```

```
    if result.get('results', {}).get('in_databse', False):
        print("PhishTank: This URL is flagged as phishing.")
        return True
    else:
        print("PhishTank: This URL seems safe.")
        return False
```

Analysing the content of the webpage for signs of phishing (e.g. fake forms).

```
def analyze_page(url):
    try:
        # Fetch the page content
        response = requests.get(url)
        soup = BeautifulSoup(response.content, 'html.parser')

        # Look for phishing signs in the html tags(e.g: fake forms, excessive redirects)
        forms = soup.find_all('form')
        for form in forms:
            action = form.get('action', '')
            if 'login' in action or 'submit' in action:
                print("Suspicious form found in page!")
                return True
        return False
    except Exception as e:
        print(f"Error fetching page: {e}")
        return True
```

Finally the main function, where it executes each of these functions: -

```
# Final phishing link scanner function
def phishing_link_scanner(url):
    print(f"Analyzing URL: {url}")
    # Step 1: Check the URL pattern
    if is_suspicious_url(url):
        print("Suspicious URL pattern detected!")
        return True
    # Step 2: Check domain registration information
    if not get_domain_info(url):
        print("Suspicious domain registration detected!")
        return True
    # Step 3: Check URL with PhishTank
    if check_with_phishtank(url):
        return True

    # Step 4: Analyze the page for suspicious content
    if analyze_page(url):
        print("Suspicious content found on the page!")
        return True

    print("The URL seems safe.")
    return False
```

The code for testing these functions is: -

```
test_url = input("Enter URL to check: ")
result = phishing_link_scanner(test_url)
if result:
  print("Warning! This is a phishing attempt.")
else:
  print("The URL appears to be safe.")
```

which might generate this o/p: -

```
Enter URL to check:  http://example.com
Analyzing URL: http://example.com
Suspicious domain registration detected!
Warning! This is a phishing attempt.
```

It prompts you to enter a URL.
Then, it runs all the checks on the URL and tells you whether it's safe or a phishing attempt.

Summary:

The Scanner looks at patterns in the URL, domain registration info, page content, and checks with PhishTank to see if the URL is a phishing attempt.

It helps protect you from visiting malicious websites that might try to steal your personal information.