CYBER SECURITY FUNDAMENTALS

ASSIGNMENT -8

NAME: B. SHANMUKH

Reg.no: 282023-024

Question – 1

Imagine you are a cybersecurity analyst working for a large multinational corporation. One morning, your team receives an urgent report about a potential security breach in the company's network. The IT department has noticed unusual network activity originating from a particular IP address. Your team has been tasked with investigating this incident to determine if it poses a threat to the organization's network security.

Assignment Question:

1. **Using the Python library Scapy, analyze the network packets associated with the suspicious IP address provided.**

Expected Procedure:

1. **A detailed explanation of how Scapy can be utilized to capture and dissect network packets**

2. **A step-by-step breakdown of the process you followed to capture and analyze the network traffic.**

3. **Identification and interpretation of any suspicious or anomalous network behavior observed in the captured packets.**

4. **Recommendations for mitigating the identified security risks and securing the network against similar threats in the future.**

Expected Code:

1. **Write a python code to Network Packet Analysis with Scapy.**

ANS:

1. Install Scapy: If you haven't already, install the Scapy library using pip:

pip install scapy

2. Write Python code to capture and analyze network packets:

```
from scapy.all import *

# Define the suspicious IP address
suspicious_ip = "x.x.x.x"
  # Replace x.x.x.x with the actual suspicious IP address

# Define a function to analyze packets
def packet_analysis(packet):
    if IP in packet and packet.haslayer(TCP):
 # Ensure the packet has IP and TCP layers
        if packet[IP].src == suspicious_ip or packet[IP].dst == suspicious_ip:
 # Check if the suspicious IP is either source or destination
            print("Suspicious Packet Found:")
            print(packet.summary())  # Print a summary of the suspicious packet
            print(packet.show())
 # Print detailed information about the suspicious packet

# Start packet sniffing
print("Starting packet capture...")
sniff(prn=packet_analysis, store=0)
# Call packet_analysis function for each captured packet
```

- This code snippet sets up a packet sniffer using Scapy. It captures packets and checks if the suspicious IP address is either the source or destination of the packet. If a packet matches this condition, it prints a summary and detailed information about the suspicious packet.

- To run this code, save it to a Python file (e.g., analyze_packets.py) and execute it in a terminal or command prompt:

python analyze_packets.py

Question – 2

Imagine you are working as a cybersecurity analyst at a prestigious firm. Recently, your company has been experiencing a surge in cyber attacks, particularly through phishing emails and websites. These attacks have not only compromised sensitive information but also tarnished the reputation of the company.
In light of these events, your team has been tasked with developing a robust solution to detect and mitigate phishing websites effectively. Leveraging your expertise in Python programming and cybersecurity, your goal is to create a program that can accurately identify phishing websites based on various features and indicators.

Assignment Task:

Using the Python programming language, develop a phishing website detection system that analyzes website characteristics and determines the likelihood of it being a phishing site.

Expected Procedure:

1. **Accept 2 web URL. One real and another one phishing**
2. **Analyze the data from both the websites.**
3. **Identify the phishing site.**

Expected Code:

- **Phishing Website Detection with Python**

ANS:

To develop a phishing website detection system in Python, we can utilize various features and indicators commonly associated with phishing websites. One approach is to analyze characteristics such as domain age, SSL certificate validity, presence of suspicious links or forms, and similarity to known phishing templates. Below is a simple implementation of such a system:

```
import requests
from bs4 import BeautifulSoup
import whois
import datetime
import re

def get_html(url):
    try:
        response = requests.get(url)
        return response.text
    except Exception as e:
        print("Error fetching URL:", e)
        return None

def get_domain_age(domain):
    try:
        whois_info = whois.whois(domain)
        creation_date = whois_info.creation_date
        if isinstance(creation_date, list):
            creation_date = creation_date[0]
        return (datetime.datetime.now() - creation_date).days
    except Exception as e:
        print("Error getting domain age:", e)
        return None

def check_ssl_validity(url):
    try:
        response = requests.get(url)
        return response.status_code == 200 and "https://" in response.url
    except Exception as e:
        print("Error checking SSL validity:", e)
        return False

def check_hyperlinks(html):
    try:
        soup = BeautifulSoup(html, 'html.parser')
        links = soup.find_all('a', href=True)
        suspicious_links = [link['href'] for link in links if 'javascript:' in link['href'] or
'mailto:' in link['href']]
        return len(suspicious_links) > 0
    except Exception as e:
        print("Error checking hyperlinks:", e)
        return False

def check_forms(html):
    try:
        soup = BeautifulSoup(html, 'html.parser')
        forms = soup.find_all('form')
        return len(forms) > 0
    except Exception as e:
        print("Error checking forms:", e)
        return False
```

```python
def analyze_website(url):
    html = get_html(url)
    if html is None:
        return False

    domain = re.search(r'https?://([A-Za-z_0-9.-]+).*', url).group(1)

    domain_age = get_domain_age(domain)
    ssl_validity = check_ssl_validity(url)
    hyperlinks = check_hyperlinks(html)
    forms = check_forms(html)

    # Example heuristic: if domain age is less than 30 days, consider it suspicious
    if domain_age is not None and domain_age < 30:
        return True

    # Example heuristic: if SSL certificate is invalid or missing, consider it suspicious
    if not ssl_validity:
        return True

    # Example heuristic: if suspicious hyperlinks or forms are found, consider it suspicious
    if hyperlinks or forms:
        return True

    return False

# Example usage
real_website_url = "https://example.com"
phishing_website_url = "https://example-phishing.com"

real_website_result = analyze_website(real_website_url)
phishing_website_result = analyze_website(phishing_website_url)

print("Real website:", "Phishing" if real_website_result else "Legitimate")
print("Phishing website:", "Phishing" if phishing_website_result else "Legitimate")
```

In this code:

- We define functions to extract HTML content, retrieve domain age, check SSL validity, and analyze hyperlinks and forms presence.
- We then use these functions to analyze given URLs and determine the likelihood of them being phishing websites based on certain heuristics.
- Heuristics could include factors like domain age, SSL certificate validity, presence of suspicious links or forms, etc.
- This is a basic implementation and can be extended with more sophisticated features and machine learning models for better accuracy.