OS needed

Kali linux - Attacker launching the DDoS attack using hping3.

Windows - as the victim.

Ubuntu - Monitoring and running the detection script. Watch the console output on the Ubuntu monitoring machine for potential DDoS detection messages. Check the configured email inbox for any alerts sent by the script.

Checklist:

1. **Virtual Machine Setup**
    - Windows Machine (victim)
    - Ubuntu Machine (monitoring)
    - Kali Linux Machine (attacker)
2. **Network Configuration**
    - Ensure all VMs can communicate on the same virtual network.
3. **Ubuntu Machine Preparation**
    - System update and package installation
    - Pyshark and SMTP setup
    - Create and configure ddos_detection.py script
4. **Kali Linux Preparation**
    - System update and hping3 installation
    - Launch DDoS attack using hping3
5. **Running the Script**
    - Ensure the script is executable
    - Execute the script on the Ubuntu machine
6. **Monitoring and Alerts**
    - Monitor console output for detection messages
    - Check email inbox for alert notifications
7. **Testing and Validation**
    - Simulate different network traffic scenarios
    - Adjust thresholds and intervals as needed
    - Document configurations and changes

Software Requirements

1. Windows Machine (Victim)

- **Operating System**: Windows 10/11 (or any supported version)
- **Network Configuration**: Ensure the machine is accessible over the network.

2. Ubuntu Machine (Monitoring)

- **Operating System**: Ubuntu 20.04 LTS (or any supported version)
- **Python**: Python 3.x
- **Packages**: Pyshark, Tshark, Secure-SMTPLib
- **Network Configuration**: Ensure the machine is accessible over the network.

3. Kali Linux Machine (Attacker)

- **Operating System**: Kali Linux
- **Network Testing Tool**: hping3
- **Network Configuration**: Ensure the machine is accessible over the network.

Detailed Steps for Execution

Step 1: Setting Up Virtual Machines

1. **Install Virtual Machines**
    - Use virtualization software like VMware, **VirtualBox**, or any other preferred tool.
    - Create three VMs: one for Windows, one for Ubuntu, and one for Kali Linux.
2. **Network Configuration**
    - Ensure all VMs are on the same virtual network (e.g., **bridged** or internal network) to allow communication between them.

Step 2: Preparing the Ubuntu Monitoring Machine

- **Update System**

  sudo apt-get update
  sudo apt-get upgrade

- **Install pip3 (if not already installed)& pyshark**

  sudo apt install python3-pip
  pip3 install pyshark

- **Install Python and Create Virtual Environment**
  sudo apt-get install python3-venv
  python3 -m venv venv
  source venv/bin/activate *--after this command ubuntu run on virtual python environment*

- **Install Pyshark and Secure-SMTPLib**
  pip install pyshark secure-smtplib

- **Run Your Script**:
  python dos_detection.py

After this the Python script starts Executing and starts monitoring the packets from attacker

The python script as follows:
Run the script on Ubuntu by using the terminal
"nano dos_detection.py"
Save it by ctrl+u (write out) and exit by ctrl+x

```
import pyshark
import smtplib
from email.mime.text import MIMEText
from datetime import datetime

# Define the email parameters
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT = 587
EMAIL_ADDRESS = 'alerts.batch5@gmail.com'
EMAIL_PASSWORD = 'csbthxfhprmnewzx'  # Use your generated app password
TO_EMAIL = 'purijagannadh13@gmail.com'

# Color codes
COLOR_RESET = '\033[0m'
COLOR_RED = '\033[91m'
COLOR_GREEN = '\033[92m'
COLOR_YELLOW = '\033[93m'

# Function to send an email alert
def send_email_alert(attacker_ip, packet_count):
    subject = "Alert: Potential DDoS Attack Detected"
    detection_time = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
    body = (f"Dear Network Administrator,\n\n"
            f"Our monitoring system has detected a potential DDoS attack.\n\n"
            f"Details:\n"
            f"- Attacker IP: {attacker_ip}\n"
            f"- Packets per second: {packet_count}\n"
            f"- Detection Time: {detection_time}\n\n"
            f"Please review the network traffic and take necessary actions to mitigate the threat.\n\n"
            f"Best regards,\n"
            f"Network Security System")

    msg = MIMEText(body)
    msg['Subject'] = subject
    msg['From'] = EMAIL_ADDRESS
    msg['To'] = TO_EMAIL

    try:
        with smtplib.SMTP(SMTP_SERVER, SMTP_PORT) as server:
            server.starttls()
            server.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            server.sendmail(EMAIL_ADDRESS, TO_EMAIL, msg.as_string())
        print(f"{COLOR_GREEN}Email alert sent for attacker IP: {attacker_ip}, Packet count:
{packet_count}{COLOR_RESET}")
```

```
    except Exception as e:
        print(f"{COLOR_RED}Failed to send email: {e}{COLOR_RESET}")

# Define a threshold value for packet count
threshold = 20

# Create a live capture object
capture = pyshark.LiveCapture(interface='enp0s3')

# Dictionary to track packet counts per IP
packet_counts = {}

# Print initial message
print(f"{COLOR_YELLOW}Monitoring network traffic for DoS attack...{COLOR_RESET}")

# Start capturing packets
for packet in capture.sniff_continuously():
    try:
        if 'IP' in packet:
            ip_src = packet.ip.src

            # Increment packet count for the source IP
            if ip_src in packet_counts:
                packet_counts[ip_src] += 1
            else:
                packet_counts[ip_src] = 1

            # Check if the packet count exceeds the threshold
            if packet_counts[ip_src] > threshold:
                message = (f"{COLOR_RED}Potential DoS attack detected from IP: {ip_src}. "
                           f"Packet count: {packet_counts[ip_src]}{COLOR_RESET}")
                print(message)
                send_email_alert(ip_src, packet_counts[ip_src])

                # Continue sending alerts for each additional packet
                threshold = packet_counts[ip_src] + 1  # Update threshold to always alert for additional packets

    except AttributeError:
        # Some packets might not have an IP layer, so we skip them
        Pass
```

Interface: *Ensure the network interface (eth0) matches the interface on your Ubuntu monitoring machine.*
Type the following command and press Enter:

ip a
Look for the interface that has an IP address assigned to it and is in the UP state.
The interface name will typically be something like eth0, eth1, ens33, or ***enp0s3.***

Step 3: Preparing the Kali Linux Attacker Machine

- **Update System**
  sudo apt-get update
  sudo apt-get upgrade

- **Install hping3**
  sudo apt-get install hping3

- **Launch the DoS Attack**:
  sudo hping3 -S --flood -p 80 –rand-source ***<Windows_IP>***

Step 4: Preparing the Windows Victim Machine

No specific setup is needed unless you want to install network monitoring tools for observing the attack effects

After Executing the Python Script in UBUNTU
Monitor the Output

- **Watch for Alerts**: The script will print a message if a potential DDoS attack is detected, including the attacker's IP address.
- **Real-Time Monitoring**: As the script runs, it will check for incoming packets and identify any suspicious activity.

Check IP Addresses:

- **Windows Machine**: Find the IP address by opening Command Prompt and typing:
  ipconfig
- **Ubuntu Machine**: Check the IP address by typing in the terminal:
  ip a
- **Kali Linux Machine***: Check the IP address by typing in the terminal:*
  Ip a

Verifying Network Configuration

1. **Check IP Addresses**:
   - After setting up the internal network, check the IP addresses assigned to each VM to ensure they are within the same subnet.
2. **Ping Test**:
   - From each VM, try to ping the other VMs to verify network connectivity. For example, from the Ubuntu VM, you can use:

   Ping <IP_of_Windows_VM> if in ubuntu
   ping <IP_of_Kali_Linux_VM>

Kali IP - 192.168.185.251
windows IP - 192.168.185.223
Ubuntu IP - 192.168.185.196

new mail id
alerts.batch5@gmail.com
Batch#05
csbt hxfh prmn ewzx