**LAB12 wireless security coding**

**Class: M03 student ID: B2111933 Name: Truong Dang Truc Lam**

| Items | Resource |
| --- | --- |
| VM type | VirtualBox VM |
| OS | Kali Linux 2023.4 (Debian-based) |
| IP/URL | 192.168.x.x (local IP assigned to VM) |
| Language,version | Python 3.6.x (upgrade to ≥3.8 recommended) |
| framework |  |
| Libraries | scapy, cryptography, socket, struct, logging |
| Software tool | Wireshark, ifconfig/iproute2, scapy |
| Protocol; | MQTT, CoAP |
| Message broker | Mosquitto |
| Software tool | VirtualBox, Wireshark, Python, Terminal, tcpdump |
| Container | Docker |
| Code Reference |  |
|  |  |

Select one exercise from 1, 2 types

**1. Survey a wireless security pen-test process (2504-11)**

1. Software setup environment/IDE (multiple choice possible)
2. Design software Architecture
3. Explain your pen-test process (write used resource)

**2. Coding a wireless security pen-test program(select one pentest) (2504-11)**

1. Clone one coding model from slide or Ref.site

[Pentest #3] The MAC address flooding attacks

1. Software setup environment/IDE (multiple choice possible)

Kali Linux Terminal – main execution environment

VS Code – Python script editing

Python 3.6 (via python3) – scripting language used

Scapy – packet crafting and sending

Wireshark – packet sniffing and traffic analysis

VirtualBox – to run Kali Linux VM

Net-tools / iproute2 – to check and configure network interfaces (e.g., ifconfig, ip)

tcpdump – for verifying packet transmission

1. Design software Architecture

The software architecture of this project is designed for simplicity and effectiveness in performing a Layer 2 network attack. It is developed in Python and runs in a Kali Linux virtual machine hosted on VirtualBox.

The core logic is implemented in a script called cam\_flood.py, which uses the Scapy library to craft and send Ethernet frames. These packets contain randomly generated MAC and IP addresses to simulate fake hosts on the network.

The attack targets the CAM table of a network switch by overwhelming it with spoofed packets. This causes the switch to enter a fail-open mode, flooding all ports and creating potential for eavesdropping or denial of service.

The system operates through the Ethernet interface (enp0s3) and is executed from the Kali terminal. Tools like Wireshark and tcpdump are used optionally to observe traffic and verify the impact.

1. Explain your coding process (write used resource)

Because my PC use Ethernet for Network, so I will attack through Ethernet instead:

from scapy.all import \*

import logging

# Suppress Scapy warnings

logging.getLogger("scapy.runtime").setLevel(logging.ERROR)

# Function to generate a list of Ethernet/IP packets with random MAC/IP addresses

def generate\_packets():

packet\_list = []

for \_ in range(1000):

packet = Ether(src=RandMAC(), dst=RandMAC()) / IP(src=RandIP(), dst=RandIP())

packet\_list.append(packet)

return packet\_list

# Function to send packets over Ethernet

def cam\_overflow(packet\_list):

sendp(packet\_list, iface='enp0s3', verbose=False)

if \_\_name\_\_ == '\_\_main\_\_':

print("[\*] Generating and sending CAM overflow packets via enp0s3...")

packets = generate\_packets()

cam\_overflow(packets)

print("[+] Sent 1000 packets successfully.")

1. Execute your process and explain

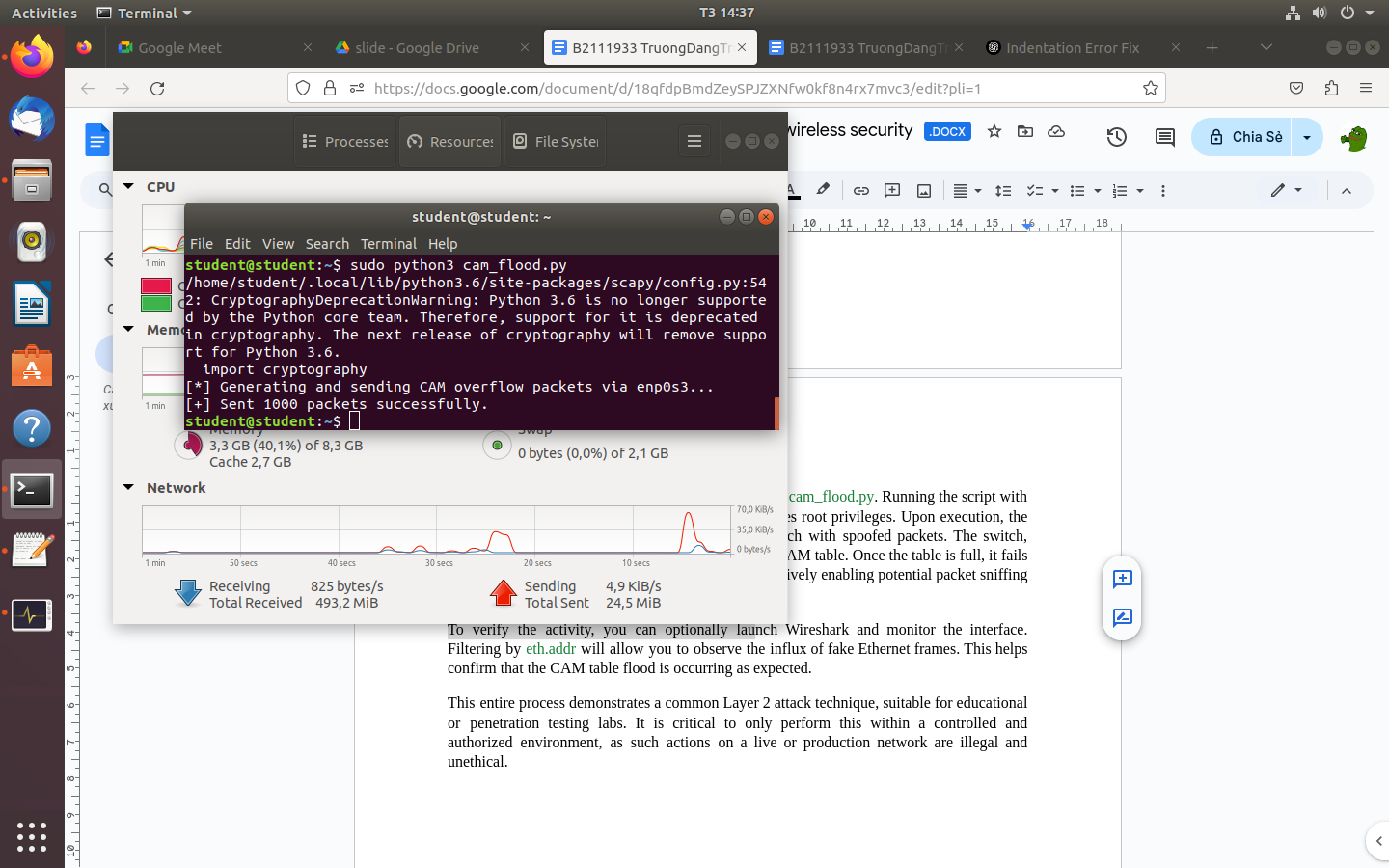
To execute the CAM table overflow attack, first ensure you're working inside a Kali Linux environment—ideally a virtual machine such as VirtualBox. Begin by opening a terminal and checking your active network interface using the ip a command. Identify the wired interface, which is typically named something like enp0s3. This interface will be used to send spoofed packets onto the local network.

Next, create a Python script named cam\_flood.py using a text editor such as nano. Inside this file, write the attack logic using the Scapy library. The script generates a large number of Ethernet frames with randomly spoofed MAC and IP addresses. These packets are designed to mimic many different hosts and are crafted using RandMAC() and RandIP() provided by Scapy. The generated packets are stored in a list and later transmitted using the sendp() function over the identified Ethernet interface.

After saving the script, execute it using sudo python3 cam\_flood.py. Running the script with sudo is necessary because sending raw packets requires root privileges. Upon execution, the script will rapidly flood the connected network switch with spoofed packets. The switch, attempting to learn each fake MAC address, fills its CAM table. Once the table is full, it fails open and begins broadcasting traffic to all ports, effectively enabling potential packet sniffing or network disruption.

To verify the activity, you can optionally launch Wireshark and monitor the interface. Filtering by eth.addr will allow you to observe the influx of fake Ethernet frames. This helps confirm that the CAM table flood is occurring as expected.

This entire process demonstrates a common Layer 2 attack technique, suitable for educational or penetration testing labs. It is critical to only perform this within a controlled and authorized environment, as such actions on a live or production network are illegal and unethical.



Flooding attack