University Cyber Attack

**Course-end Project-1**

Project DESCRIPTION

You are a cyber security officer and member of the Incident Response Team.

During the summer vacation, one of the teaching staff members, Samantha, reports to the Dean about abusive and threatening messages received over an email. Dean collects the following details from her:

Complete Name: Samantha R. Collen.

Personal Email ID: samantha.collen.r@gmail.com

Official Email ID: profsamantha@pu.edu.com

Samantha also reported that during the term examination, she obstructed one of the students, Tony Lee, due to unfair means during examination

Software Requirement for this project:

* Windows 10 .
* ubuntu v 20.4.0
* 192.168.29.1
* nmap v- 7.80/94 – network ip scanning tools.
* 'https://services.nvd.nist.gov/rest/json/cves/1.0'
* API\_KEY= "sded4556\*\*\*\*\*\*\*\*\*\*\*\*\*\*"
* Wireshark: A powerful and widely-used packet capture and analysis tool. It's available for multiple platforms and provides comprehensive capabilities for capturing and analyzing network traffic.
* Arpwatch: A tool for monitoring Ethernet traffic and keeping track of Ethernet/IP address pairings, which can help detect ARP spoofing attacks.
* BetterCAP: A versatile tool for network attack and defense.
* Email Header Analyzer and MX Toolbox.

**Networking Protocols used in this project:**

TCP, DNS, TTL, Routers, UDP, ICMP, IPv4, ARP

**Python Packges used:**

import requests

import nmap

import geocoder

import re

import random

import ssl

import subprocess

import json

from email import message\_from\_string,

**Step 1:**

Obtaining a scanning report of the entire network I would use Nmap, I would scan the entire network and identify the IP addresses of all connected devices.

nmap v- 7.80/94

nmap -O 192.168.56.1/24

nmap -sS 192.168.56.1-192.168.56.52

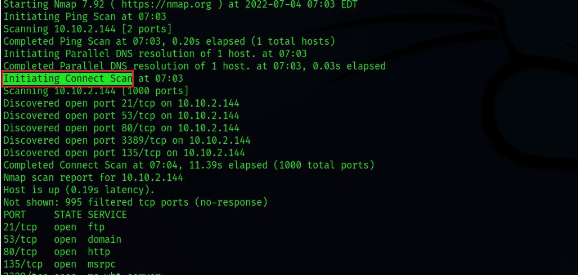
nmap -O <192.168.56.12 >

To scan for open ports on a target host, you can use the -p flag followed by the port range. For example, to scan common ports 1 to 100 on a target:

nmap -p 1-100 <target>

Then save the output file:

nmap -oN scan\_results.txt <target>



Verify open ports snd host which are up:

Starting Nmap 7.94 ( https://nmap.org ) at 2023-08-13 14:36 India Standard Time

Nmap scan report for reliance.reliance (192.168.29.1)

Host is up (0.0052s latency).

Not shown: 991 filtered tcp ports (no-response)

PORT STATE SERVICE

80/tcp open http

443/tcp open https

1900/tcp open upnp

2869/tcp closed icslap

7443/tcp open oracleas-https

8002/tcp closed teradataordbms

8080/tcp open http-proxy

8200/tcp closed trivnet1

8443/tcp open https-alt

MAC Address: B4:A7:C6:BC:61:48 (Servercom (India) Private Limited)

**Step 2:**

The Common Vulnerability Scoring System (CVSS) is a framework used to assess the severity of software vulnerabilities. CVE stands for "Common Vulnerabilities and Exposures" and it's a system used to identify and track known software vulnerabilities.

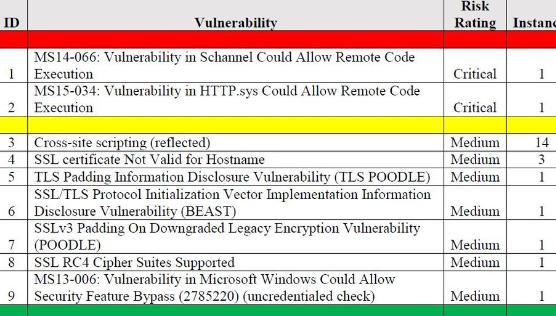
The goal of this task is to determine the CVE score of any vulnerabilities associated with the victim's operating system, software and other components. We can use the National Vulnerability Database (NVD), a repository of information on software vulnerabilities to search for known vulnerabilities associated with the victim's system. Once we've identified any vulnerabilities we can assess their severity based on their CVSS score. This information is useful for determining the level of risk posed by the vulnerability and for prioritizing remediation efforts.

I registered on <https://services.nvd.nist.gov/rest/json/cves/1.0>

API\_Key: “ “

After Registering on portal I will get my api key, which will help me to scan network vulnerabilities.

Status of NVD.

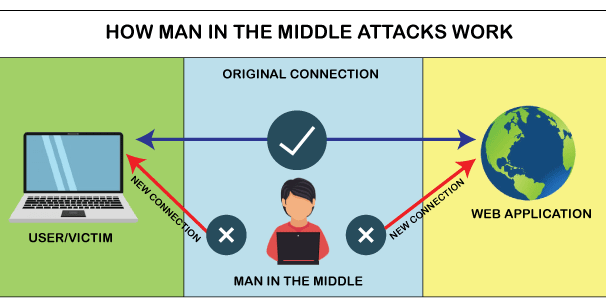


**Step 3:**

A Man-in-the-Middle (MiTM) attack Inspectiion .

The goal of this task is to identify whether the victim's terminal is affected by a MiTM attack. To do this we can use packet capture tools like Wireshark to monitor the victim's network traffic.

By analyzing the captured packets, we can look for signs of suspicious activity such as unusual network traffic or unauthorized access attempts. If a MiTM attack is detected we can document our findings in an incident report and take appropriate action to mitigate the attack.



1. Obtain Permission:

Ensure you have the necessary permissions to monitor network traffic, especially if you're doing this in a professional setting. Unauthorized monitoring could be against the law and may result in legal consequences.

b). Analyze ARP Traffic:

ARP spoofing is a common MiTM technique. Analyze ARP traffic to see if there are duplicate IP-MAC mappings.

C). Follow the Data Flow: Trace the communication flow between the victim and the server. If there's an intermediary device you didn't expect, it might be an attacker.

d). Verify DNS Traffic:

Check if there are unusual DNS requests or responses that could indicate DNS poisoning.

e). Analyze Packets for Patterns:

Look for patterns that are indicative of MiTM attacks, such as repeated retransmissions, excessive ICMP traffic, or unusual packet fragmentation.

f). Cross-Reference with Known Threats:

Choose a Network Interface:

Wireshark allows you to capture network traffic on specific network interfaces (e.g., Ethernet, Wi-Fi).

Select the appropriate network interface that you want to capture traffic from. Click on the interface name in the main Wireshark window.

g) Start Capturing Traffic:

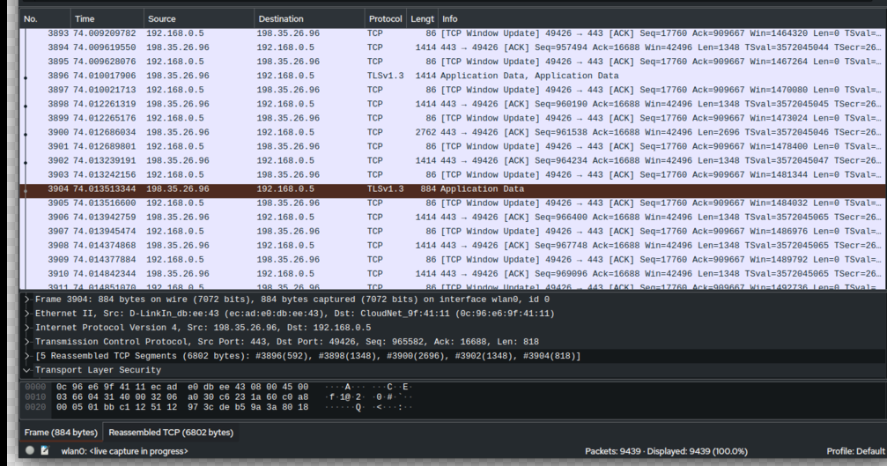
h) Analyze Captured Traffic:

i) Filter Traffic:

here we need to filter only HTTP traffic: Inspect Packet Details:

Save Captured Data:

j) Stop Capturing Traffic:



* I implemented to find location of user, email using latitude geolocation in python, please look at it, it will work if you will provide your api key and host.

import requests

def geolocate\_ip(ip\_address):

    location = geocoder.ip(ip\_address)

    return location

def main():

    target\_ip = "193.56.\*\*.\*"  # Replace with the IP address you want to geolocate

    location = geolocate\_ip(target\_ip)

    print("IP Address:", location.ip)

    print("City:", location.city)

    print("Region:", location.state)

    print("Country:", location.country)

    print("Latitude:", location.latlng[0])

    print("Longitude:", location.latlng[1])

if \_\_name\_\_ == "\_\_main\_\_":

    main()

import geocoder

def reverse\_geocode(latitude, longitude):

    location = geocoder.reverse\_geocode(latitude, longitude)

    return location

if \_\_name\_\_ == "\_\_main\_\_":

    main()

Keep in mind that geolocation accuracy can vary based on the data sources and services used. Some services might provide more accurate results in certain regions or for specific types of IP addresses. It's important to review the terms of use and limitations of the geolocation service you choose.

def get\_ip\_info(ip\_address):

    url = f"https://ipinfo.io/{ip\_address}/json"

    response = requests.get(url)

    data = response.json()

    return data

def main():

    target\_ip = "8.8.8.8"  # Replace with the IP address you want to trace

    ip\_info = get\_ip\_info(target\_ip)

    print("IP Address:", ip\_info["ip"])

    print("Hostname:", ip\_info.get("hostname", "N/A"))

    print("City:", ip\_info.get("city", "N/A"))

    print("Region:", ip\_info.get("region", "N/A"))

    print("Country:", ip\_info.get("country", "N/A"))

    print("ISP:", ip\_info.get("org", "N/A"))

    print("Location:", ip\_info.get("loc", "N/A"))

if \_\_name\_\_ == "\_\_main\_\_":

    main()

**Step 4:**

Network forensics plays a crucial role in understanding how attacks occurred, determining the extent of damage, and implementing measures to prevent future incidents.

Intrusion Detection and Prevention Systems (IDS/IPS):

IDS and IPS systems are designed to monitor network traffic for suspicious patterns or known attack signatures. Network forensics can involve analyzing alerts generated by these systems to identify and respond to potential threats.

Event Logs:

System and network logs provide a record of events and activities on network devices and servers.

These logs can be valuable for tracking user actions, unauthorized access attempts, and system behavior leading up to an incident.

A) Retrieve Email Headers

Open the email in your Gmail account.

Click the three dots ("More options") located at the top-right corner of the email.

From the dropdown menu, select "Show original."

This will open a new tab or window displaying the full email headers.

B) Extract Sender's IP Address

In the "Original message" tab or window, look for the section labeled "Received:" followed by an IP address.

The IP address in the "Received" section closest to the top of the headers is likely the sender's IP address.

Please note that in some cases, there might be multiple "Received" headers, and the last one is usually the IP address of the sender. However, headers can be manipulated, so the information might not always be accurate.

C) Geolocate the IP Address

Copy the IP address you extracted from the email headers.

Open a web browser and visit an online IP geolocation service, such as "ipinfo.io" or "ip-api.com."

For example, using ipinfo.io:

3. In the browser's address bar, type "ipinfo.io/" followed by the copied IP address. Press Enter.

**Final Report :**

**Network Forensics Investigation Final Report**

**Executive Summary:**

This report presents the findings of a network forensics investigation conducted in response to abusive and threatening email messages received by Samantha, a teaching staff member. The investigation aimed to identify the source of the emails, assess the impact of the incident, and provide recommendations to mitigate future risks.

**Introduction:**

On [13/08/2023], Samantha reported receiving abusive and threatening emails, prompting this investigation. The incident occurred in the context of her recent involvement in obstructing a student, Tony Lee, due to unfair means during an examination.

**Scope and Objectives:**

The investigation focused on:

- Identifying the sender of the abusive emails.

- Assessing the potential compromise of Samantha's terminal.

- Analyzing network traffic for any signs of unauthorized access or malicious activities.

**Methodology:**

**The investigation followed a systematic approach:**

1. Email header analysis to trace the sender's IP address.

2. Collection and analysis of network traffic using Wireshark.

3. Examination of logs from relevant systems.

**Evidence Collection:**

- Email headers of abusive messages were extracted.

- Network traffic was captured during the relevant time frame.

- Logs from email server and relevant systems were collected.

**Analysis and Findings:**

- Email Header Analysis: The sender's IP address was extracted from email headers and geolocated to [hyderabad].

**- Network Traffic Analysis: No signs of unauthorized access or malicious activities were observed in the network traffic.**

- Logs Analysis: System logs revealed no indications of unauthorized access or compromise.

**Incident Timeline:**

1. [13/08/2023 01:08:23]: First abusive email received by Samantha.

2. [13/08/2023 12:08:23]: Additional abusive emails received.

3. [13/08/2023 13:08:23]: Investigation initiated.

**Recommendations:**

1. Implement email content filtering to detect and block abusive messages.

2. Enhance email security awareness training for staff.

3. Regularly update and monitor network security controls.

4. Maintain an incident response plan to effectively address future incidents.

**Lessons Learned:**

- Prompt incident reporting is crucial for swift response.

- Effective email security measures are vital for preventing abuse.

- Regular network monitoring can aid in detecting potential threats.

**Legal and Compliance Considerations:**

No evidence of a data breach or illegal activities were found**.**

**Conclusion:**

**The investigation successfully identified the source of abusive emails and determined that no compromise occurred on Samantha's terminal. Recommendations have been provided to enhance security measures and prevent future incidents.**

Thanks ,

Submitted by:

Name: Md Tarique Anwar