Homework 2
CS 494: Network Security

September 20, 2023

In this homework assignment, you will learn the concepts and tools for passive and active network analysis. The due date for this assignment is **October 4, 2023 at 11:59pm local time**. The required files for this assignment are contained in `hw2_files.zip` which can be downloaded from Piazza. Make sure you read the handout before starting.

You will need to submit your answers in a specified format to receive complete credit. All the text-based responses should go in a single file named `hw2.txt`. This file should be formatted in the following way:

```
1:  the answer to question 1
2:  the answer to question 2
```

Answers to questions that require other than a text-based response (e.g. an image file, or code), should be submitted as part of a zip archive containing your `hw2.txt` and other files. The final zip file is the one file you are required to submit. Refer to **section 6** for detailed submission instructions.

1 Getting Started

For this course, we will use Piazza for all course related questions, concerns, and logistics. If you have any questions, you should refer to the class Piazza. You will make your submissions and be able to view your grades via Gradescope.

You are required to have a POSIX environment for the homework assignments and it is highly recommended you get it set up early on. For the primary programming language, we will use Python 3.5 or above. Please make sure that you are using Python 3.5+ for all assignments. You are responsible for making sure your assignments work with the correct version. If you are unsure what version of python you are using, you can always check by typing `python3 --version` in the command line. You can get the latest version of Python by going to the Python website and installing the appropriate download. If you have any questions or problems in the installation process, contact the TA.

2 Wireshark (12 points)

In this first part of the assignment, you will learn about Wireshark, and its packet filtering and capturing features. You can install the latest version from link. `hw2_files.zip` provided contains two packet traces which you will use for this part by loading them into Wireshark.

To answer questions 1 to 6 use `trace1.pcap`, for questions 7 to 9 use `trace2.pcap`.

1. What are the total number of packets sent by the host `198.105.254.25`?

2. What is the source MAC address of the machine generating the ARP packets?

3. There is an HTTP connection with the host `143.244.131.1`. What is the cookie value in the header?

4. There is one DNS query to a `.edu` domain. What is the domain name in the query?

5. In a different HTTP connection, there is a login attempt with the query including `username` and `password`. What are the values being queried? The username is cleartext while the password is `hashed`. Use the tool and input files (dictionary.txt) of Assignment-1 to retrieve the original value of the hashed password.

6. Extract the contents of the JPEG image corresponding the GET request for `/images/image.jpeg` to a destination host. Save the image as `6.jpg` and include it the final zip archive.

(Reminder: Use `trace2.pcap` for questions 7 to 9)

7. What is the POP username?

8. What is the POP password?

9. How many email messages are there in the email account?

10. In this task, you use Wireshark to capture live traffic and save it as a packet trace. Using `curl`, make a request to `example.com`. Capture and save all relevant HTTP and DNS traffic and save it to a file named `10.pcap`. You should filter out any packets not a part of the request.

(The dataset, and part of the task is originally from UW Madison CS 642 taught by Thomas Ristenpart in 2014. Thanks!)

3 TCPDUMP (4 points)

Similar to Wireshark, TCPDUMP is a command line tool for passive network analysis. You can download and read about TCPDUMP here. For this part of the assignment you will only use `trace1.pcap`. To load the file into TCPDUMP you can use `tcpdump -n -r trace1.pcap`. To answer the next four questions, you are required to provide the one liner tcpdump commands to filter the relevant packets (in each of the given scenarios) from the trace.

11. All packets sent from the host `131.253.40.84`.

12. Traffic received at port `80` by all the hosts.

13. Command to display only the ARP packets in the trace.

14. Filter all SSL traffic to the host `74.125.225.81`. *Hint: Remember the port used in SSL.*

4 Nmap (3 points)

Nmap (or Zenmap) is an open source tool widely used to discover open ports, and hosts on the Internet and to also create a network map. In this task of the assignment, you will use Nmap to scan certain hosts and evaluate what services they are running. You can read about and install the version of Nmap compatible with your OS from here.

As performing a network scan on a large IP range or other websites can alert intrusion detection systems, for this assignment, you will use `scannme.insecure.org` as a sample host to perform the scans. Using Nmap, answer questions 15 to 17.

15. List all the open TCP ports open on the host. Provide your responses as comma separated values in a single line.

16. What is the Nmap command to discover open UDP ports?

17. Nmap also provides a guess of OS running on the host. From your scan, what OS did Nmap detect on the host?

5 Packet Spoofing (7 points)

For the task of this assignment, you will learn how to create your own packets and send them over the network Essentially, you will use the Python's scapy module to write a spoofer. Below is a list of requirements that your code must meet to receive full credit.

- All your code should contained as part of a single function defined as `send_packet`
- Your function should take in 4 arguments `src_ip, dst_ip, dst_port, payload`
- The `send_packet` function should create a spoofed UDP packet with the payload and send it over to the destination IP and port specified
- You should ensure that the payload does not exceed 150 bytes. If it does, your function gracefully exits.
- For this task, you are only allowed to use the Python scapy module
- Save the file as `spoofer.py` and include it in the zip archive.

6 Sumbission

In total, you will submit the following 4 files in a compressed zip archive named `hw2.zip` and upload it to Gradescope:

1. `hw2.txt`: The file that contains all the text based responses. Make sure your answers match the questions numbers provided in the handout.
2. `6.jpg`: The image extracted in question 6
3. `10.pcap`: The captured pcap in question 10
4. `spoofer.py`: The file containing the code for the packet spoofing task .