Department of Computer Science and Engineering, SVNIT, Surat

MTech I (2^{nd} semester)

CSE614: Core Elective II: Machine Learning for Security

Lab Assignment No 1:

Review of ML, NetSec and Introduction to ML for Security

SPRING Semester 2022-23

Dated Uploaded: 28^{th} December 2022

Instructions:

1. *Right from the time the assignment is uploaded the students must start implementing the assignments.*
2. *You can use any programming language or toolkit for your implementation.*
3. *All the assignments must be submitted in the form of a zip file containing the Program Source, the screenshot of the output that you obtained, the DataSet/Input Test data used and a ReadMe .txt file explaining what platform to use, what are the input parameters required for execution and how to execute it. Also write your conclusion in ReadMe file.*
4. *Perform usual error checking. Don't go overboard on this, but don't let your program die because of divide by zero.*
5. *Remember, your programs could be checked by a code-cheating program, so please follow the code of academic integrity.*
6. *There will be a viva for each assignment. This viva would be conducted for this assignment on a future date as specified.*
7. ***Maximum Points: Part A(100)***

---

1. Do the following before you start attempting the exercise questions, here:

   (a) Start up your web browser - not in the https mode.

   (b) Start up the Wireshark packet sniffer, but don't yet begin packet capture. Enter "http" (just the letters, not the quotation marks) in the display-filter-specification window, so that only captured HTTP messages will be displayed later in the packet-listing window. (We're only interested in the HTTP protocol here, and don't want to see the clutter of all captured packets).

   (c) Wait a bit more than one minute (we'll see why shortly), and then begin Wireshark packet capture.

   (d) Enter the following to your browser `http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html` Your browser should display the very simple, one-line HTML file.

   (e) Stop Wireshark packet capture.

   If you are unable to run Wireshark on a live network connection, you can download a packet trace that was created when the steps above were followed as follows: Download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file $http - ethereal - trace - 1$. Once you have downloaded the trace, you can load it into Wireshark and view the trace using the File pull down menu, choosing Open, and then selecting the $http - ethereal - trace - 1$ trace file. By looking at the information in the HTTP GET and response messages, write answer to each of the following questions. Take and include the snapshots of your screen wherever you feel it is necessary to support your answer or else where it is explicitly specified. Submit the pdf version of the file by email latest by the date specified in the lab.

   (a) What are the network interfaces available on your computer? Which network did you eventually select in your experiments.

   (b) Which application layer protocol is used in this case?

   (c) What are the other protocols used and displayed in the unfiltered packet listing window of wireshark, besides the one that you answered in Q(b)?

(d) What is the IPA of your machine? What is the IPA of the destination machine? Is there any way by which you can ascertain that the IPA of the destination indeed is the same as that you observed in wireshark? If so, how ?

(e) What is the class of the IPA of the source machine ? That of destination machine?

Select the first wireshark block i.e. "frame" in the packet-header details window. The packet-header-details window shows the details of the protocols associated with the selected packet. Note, however that the first Wireshark block – shown as "Frame" - is actually not a protocol, but it is a record that describes overall information about the packet, including when it was captured and how many bits long it is.

(f) How many bits were captured in this packet? At what time was this packet captured?

(g) What is the interface id used? What is the address of the interface?

The second block is "Ethernet". Note that you may have taken a trace on a computer using 802.11 yet still see an Ethernet block instead of an 802.11 block. Why? It happens because we asked Wireshark to capture traffic in Ethernet format on the capture options, so it converted the real 802.11 header into a pseudo-Ethernet header.

After the block "Ethernet" are shown blocks for different protocol layers i.e. IP, TCP, and HTTP. Note that the order of the blocks shown is from the bottom of the protocol stack upwards. This is because as packets are passed down the stack, the header information of the lower layer protocol is added to the front of the information from the higher layer protocol. That is, the lower layer protocols come first in the packet "on the wire."

For all the subsequent questions, you may have to expand appropriate block i.e. IP, TCP or HTTP and get the required information.

(h) Which packets are forming the TCP 3-way handshake for connection establishment ? What are the SYN and ACK in each of the three packets ?

(i) How long did it take from when the HTTP GET message was sent until the HTTP OK reply was received? (By default, the value of the Time column in the packet-listing window is the amount of time, in seconds, since Wireshark tracing began. To display the Time field in time-of-day format, select the Wireshark View pull down menu, then select Time Display Format, then select Time-of-day.)

(j) Print the two HTTP messages (GET and OK) referred to in question above. To do so, select Print from the Wireshark File command menu, and select the "Selected Packet Only" and "Print as displayed" radial buttons, and then click OK.

(k) What is the destination physical address of the first packet captured? What device does it belong to? Show where in the capture would you find this information.

(l) How many bytes of header does the first frame sent have? Show where in the capture would you find this information.

(m) By looking at the Ethernet header of a frame, can we determine if it contains an IP packet? Show where in the capture would you find this information.

(n) Is it possible to know if the first packet captured has TCP or UDP as transport protocol by looking at the IP header? Explain and show where in the capture would you find this information.

(o) In the SYN, ACK. What are the source and destination ports? Are these the same for the client and the server? Explain why.

(p) Why does the Server Hello message sent by the server have 1 as a relative sequence number and 185 as a relative acknowledgement number.

(q) Right-click a TCP capture → TCP preferences → Uncheck the box "Show relative sequence number." What is the first sequence number sent by the server to the client? Why is it not the 0 displayed by wireshark?

2. This exercise is a simple exercise that only requires you to capture the *tcpdump* traffic. The problem requires you to either use two virtual machines on your laptop or two different machines in the computer lab ask the administrator for the host name of both the machines, if so. Then run the *tcpdump* command on one machine say $PC1$ (saving the output for your lab report) so that it monitors all the packets that contain the IP address of $PC2$ only and none else. Next, open a new terminal window on $PC1$ and execute *ping* command to $PC2$. It may be necessary to press $Ctrl - C$ to terminate the *tcpdump* session. It may sometimes be best to simply redirect the output of *tcpdump* straight to a file and view it afterward with the more command or a text editor. Find out how can you do so. Run the command $\$tcpdump\ -enx\ -w\ exe2.out\&$ Do you see any output on the screen ? Why ?

3. This question is in continuation of the question no 3. Run the command *telnet remote host*. *remotehost* is the host name of either another virtual machine in your machine or it is the host name of any other machine in the network used in the lab (Ask the lab technical suport staff about the name of other machine). This command would generate some TCP traffic. After you login to the remote machine, terminate the *telnet* session and terminate the *tcpdump* program.
   Next, you will use *wireshark* to open the packet trace captured by *tcpdump* and analyze the captured packets.
   To do this, run $\$wireshark\ -r\ exe3.out\ \&$. The wireshark Graphical User Interface (GUI) will pop up and the packets captured by *tcpdump* will be displayed. For your report, you need to save any one of the packets that contain the link, IP, and TCP headers. Carry out the following instructions.

   - Click on a TCP packet from the list of captured packets in the wireshark window. Then go to the Edit menu and choose *Mark* Frame.
   - Go to the File menu and choose Print. In the Wireshark:Print dialog that pops up, check File, Plain Text, Expand all levels, Print detail and supress unmarked frames. Then, enter the output text file name, e.g., *headers.txt*, and click the OK button. The marked packet is now dumped into the text file, with a detailed list of the name and value of every field in all the three headers.

   Now answer the following questions:

   (a) Draw the format of the packet you saved, including the link, IP, and TCP headers, and identify the value of each field in these headers. Express the values in the decimal format.

   (b) What is the value of the protocol field in the IP header of the packet you saved? What is the use of the protocol field?

4. In a manner similar to the Exercise no 3, now run *tcpdump* to capture an ARP request and an ARP reply and then use wireshark to analyze the frames. If there are no arp requests and replies in the network, generate some using $arpinga - remote - machine$. After you see several ARP replies in the arping output, terminate the *arping* and the *tcpdump* program. Open the *tcpdump* trace using $\$wireshark\ -r\ exe4.out\ \&$. Print one ARP request and one ARP reply using wireshark. Now answer the following questions:

   (a) What is the value of the frame type field in an Ethernet frame carrying an ARP request and in an Ethernet frame carrying an ARP reply, respectively?

   (b) What is the value of the frame type field in an Ethernet frame carrying an IP datagram captured in the previous exercise?

   (c) What is the use of the frame type field?

5. Explain briefly the purposes of the following tcpdump expressions.

   (a) tcpdump udp port 520

   (b) tcpdump -x -s 120 ip proto 89

   (c) tcpdump -x -s 70 host ip addr1 and (ip addr2 or ip addr3)

   (d) tcpdump -x -s 70 host ip addr1 and not ip addr2

6. Start tcpdump in a command window to capture packets between your machine and a remote host using: $tcpdump\ -n\ -nn\ host\ your - host\ remote - host$. Execute any TCP utility, *telnet* for example - as in the problem before, in another command window. When you see a TCP packet in the *tcpdump* output, terminate *tcpdump* and save its output. Now answer the following question:

   (a) What are the port numbers used by the remote and the local computer?

   (b) Which machine0s port number matches the port number listed for telnet in the $/etc/services$ file? Note: In case telnet is not listed in the $/etc/services$ file, use *ssh* utility

7. Start *tcpdump* in one command window using $tcpdump\ -n\ -nn\ host\ your - host\ remote - host$. Then, telnet to the remote host from a second command window by typing *telnet remotehost*. Again issue the same command from a third command window. Now you are opening two telnet sessions to the same remote host simultaneously, from two different command windows. Check the port numbers being used on both sides of the two connections from the output in the tcpdump window. Save a TCP packet from each of the connections. Now answer the following questions:

(a) When you have two telnet sessions with your machine, what port number is used on the remote machine? Are both sessions connected to the same port number on the remote machine?

(b) What port numbers are used in your machine for the first and second telnet, respectively?

(c) What is the range of Internet-wide well-known port numbers? What is the range of well-known port numbers for Unix/Linux specific service? What is the range for a client port number? Compare your answer to the well-known port numbers defined in the /etc/services file. Are they consistent? In case they are not, try to discuss amongst peers and specify your view of the reason why they are not.
Note: In case *telnet* is not listed in the */etc/services* file, use *ssh*.

8. Execute the traceroute command with www/yahoo.com as argument. Write down the IP address of yahoo.com that was used for the trace route. Determine the number of iterations required to determine route. Enlist the IP addresses of all the machines between the source and the destination. What is the average round trip time of the packet that reached the destination ?

9. With respect to the question above, run *traceroute* on one window of your OS and run *tcpdump* on the other window. Analyze the output of *tcpdump*. Answer the following questionsmgiving appropriate hihglighted snapshots in support of your answer :

   (a) How many packets are send by traceroute in each iteration ? How can you prove this using the tcpdump output.

   (b) Consider one specific iteration of traceroute invocation/iteration. For this specific iteration, what are the individual round trip times of each of the three probes sent ? What is the average round trip time ? Does it match with the round trip time returned by traceroute ?

   (c) In each iteration of traceroute does it use the same port number for the destination ? IF yes, reason why and if no, then also argue why does it do so.

10. Download, study and analyze the NSL-KDD dataset (ref: `https://www.unb.ca/cic/datasets/nsl.html`). Is this a labeled dataset or unlabeled dataset ? Give reasons for your answer. How many samples are there in the training and that in the testing dataset ? Identify various attributes in this datasset. How many attack types are specified in this dataset ? Which ones ? Use the figure below to learn the attack types and different categories and identify them in the dataset. The study here prepares you with the groundwork to undertake the first problem in the next assignment.

Table 1. Attack types of DoS, R2L, U2R, Probe categories.

Attack class	**Attack type**
Dos	back, land, neptune, pod, smurf, teardrop
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster
U2R	buffer_overflow, loadmodule, perl, rootkit
Probe	ipsweep, nmap, portsweep, satan

Figure 1: KDD attack types