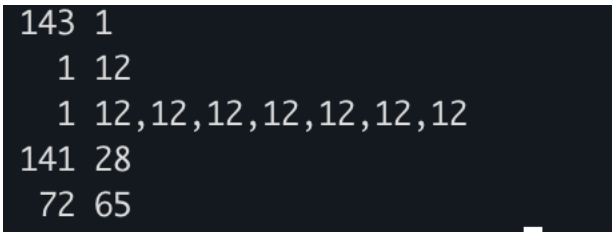
Basic Task for Computer Networking

By Raghav Dhabe

1. What types of traffic (HTTP, DNS, FTP, etc.) are present?
The types of traffic present are HTTP, TCP, DNS, mDNS, UDP, IPV4,IPV6, Ethernet.

2. How many DNS queries were made in total?
358 DNS queries were made in total. (Did it using first selecting protocol DNS, then for choosing only queries and not responses, applied the dns.flags.response == 0 on it)

3. What types of DNS queries were made?
A,AAAA, HTTPS, PTR, and interestingly enough, there was a data packet that had 7 DNS queries all of the code 12 meaning 12 PTR type of DNS queries made in 1 DNS packet.



```
143 1
  1 12
  1 12,12,12,12,12,12,12
141 28
 72 65
```

Code	Type	Meaning
1	A	IPv4 address lookup using the domain name
28	AAAA	IPv6 address lookup using domain name
65	HTTPS	Ask DNS server on how to securely connect to the domain
12	PTR	Reverse DNS lookup(looking up the domain by using the IP address)
12,12,12,12,12,12, 12		1 DNS packet making 7 PTR type queries

Run in terminal:

```
tshark -r common_task_1.pcapng -Y "dns.flags.response == 0" -T fields -e
dns.qry.type | sort | uniq -c
```

Using tshark in cli, we can read the file, apply filter, and output the select fields we care about from that.
Sort command is then used to sort the queries and then check all the unique queries present.

4. What is a Loopback Interface?
Loopback Interface is a virtual network or a virtual path that a machine can use to send data and receive it itself. An example would be hosting a website using your laptop as the server, and accessing that website on the same laptop. It is used for testing, debugging.
The loopback interface basically allows you to stimulate a virtual network(yes, *virtual network*) which allows you to test the network(the protocols and the data packets being sent and received) by yourself.

5. How many .txt files were requested? List their names.
First, we apply : {http.request.method == "GET" && http.request.uri contains ".txt"} (without the curly brackets) as a filter in Wireshark to first find the http files which have been requested(hence the GET), we then do an AND operation of that with a condition of the http request containing .txt files.
After this, from the PCAP files, we got 3 files : decoy1.txt, decoy2.txt, and encoded.txt

6. One .txt file contains base64-encoded content. Identify and decode it.
What does it contain?

Upon checking the file encoded.txt, (using follow->HTTP stream)
the base-64 encoded string is found
after decoding using a cmd line
echo 'RkxBR3tzcGlkM3JfbmV0d29ya19tYXN0ZXJ9Cg==' | base64 -d
I got the string:
FLAG{spid3r_network_master}

7. Was any attempt made to distract the analyst using decoy files? Explain.
Yes, there were 2 decoy files used to distract the analyst.
Upon checking decoy1.txt file, the text displayed "This is just a decoy"
decoy2.txt displayed "Nothing to see here"

8. Are there any known ports being used for uncommon services?
HTTPS is a DNS type in port 53 for a few packets, which is uncommon.

And there seem to be a lot of packets from port 53 having DNS type as HTTPS
And when checking the TCP port 8000, which is commonly used, it has 6 packets which use http protocol as well, which isn't uncommon(can be used for local dev servers) but it is non-standard, worth checking out the content.

9. How many HTTP GET requests are visible in the capture?
There are only 3 HTTP GET *requests* visible. Total 6 HTTP connections are seen in the file. (Again, just by applying the filter http.request.method == “GET”, we get the http get requests)

10. What User-Agent was used to make the HTTP requests?
User-Agent: curl/8.5.0
(using follow->HTTP stream)