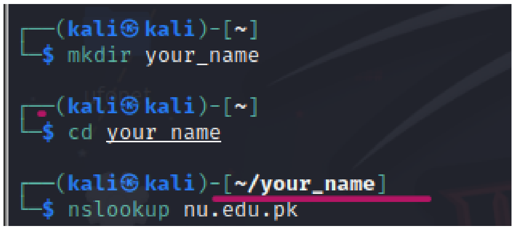
CY-3001 NETWORKS AND CYBERSECURITY II FALL 2023
ASSIGNMENT 1: Networking
Total Marks: 50

INSTRUCTIONS:

1. You should submit a zip containing your report and code file and use the naming convention i201234_A1.zip
2. To ensure that your work is not plagiarized; create a directory with your name and move into it, run all commands here and make sure that your name is visible in each screenshot.
3. In case of your name not being visible, your assignment will be treated as a plagiarism case and you will not be awarded any marks.



```
┌──(kali㉿kali)-[~]
└─$ mkdir your_name

┌──(kali㉿kali)-[~]
└─$ cd your_name

┌──(kali㉿kali)-[~/your_name]
└─$ nslookup nu.edu.pk
```

QUESTION# 1

The Domain Name System (DNS) translates hostnames to IP addresses, fulfilling a critical role in the Internet infrastructure. In this lab, we'll take a closer look at the client side of DNS. Recall that the client's role in the DNS is relatively simple – a client sends a *query* to its local DNS server, and receives a *response* back. As discussed in class, much can go on "under the covers," invisible to a DNS client, as the hierarchical DNS servers communicate with each other to either recursively or iteratively resolve the client's DNS query. From the DNS client's standpoint, however, the protocol is quite simple – a query is formulated to the local DNS server and a response is received from that server.

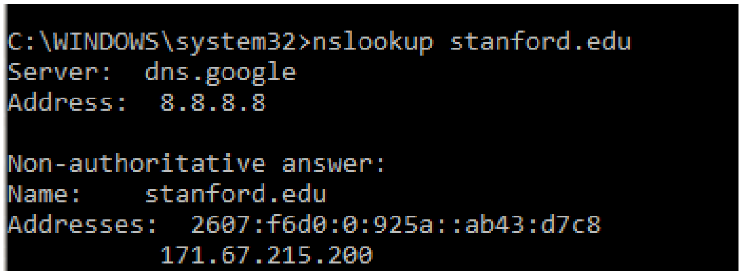
1**. nslookup [15]**

Let's start our investigation of the DNS by examining the nslookup command, which will invoke the underlying DNS services to implement its functionality. The nslookup command is available in most Microsoft, Apple IOS, and Linux operating systems. To run nslookup you just type the nslookup command on the command line in a DOS window, Mac IOS terminal window, or Linux shell.

In its most basic operation, nslookup allows the host running nslookup to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top-level-domain

(TLD) DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). For example, nslookup can be used to retrieve a "Type=A" DNS record that maps a hostname (e.g., www.nyu.edu) to its IP address. To accomplish this task, nslookup sends a DNS query to the specified DNS server (or the default local DNS server for the host on which nslookup is run, if no specific DNS server is specified), receives a DNS response from that DNS server, and displays the result.

Let's take nslookup out for a spin! We'll first run nslookup on the Linux command line on the www.mit.edu host, where the local name server is named dns.google (which has an IP address 8.8.8.8). Let's try nslookup in its simplest form:



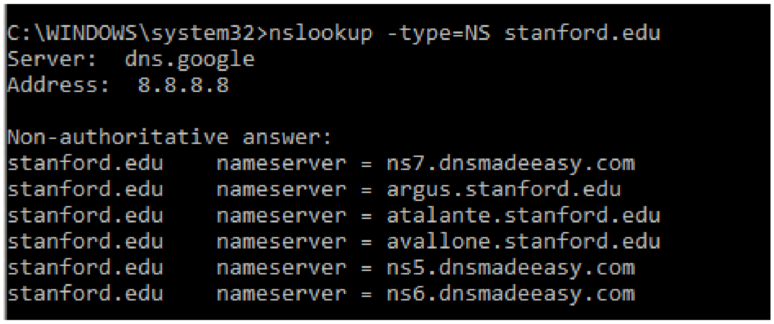
```
C:\WINDOWS\system32>nslookup stanford.edu
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
Name:    stanford.edu
Addresses:  2607:f6d0:0:925a::ab43:d7c8
          171.67.215.200
```

Figure 1: the basic nslookup command

In the first command the nslookup command is given one argument, a hostname (www.standford.edu). In words, this command is saying "please send me the IP address for the host www.stanford.edu." As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer – in this case the local DNS server 'dns.google'; and (2) the answer itself, which is the host name and IP address of www.stanford.edu. You may have noticed that there are two addresses provided for www.stanford.edu. The first (171.67.215.200) is an IPv4 address in the familiar-looking dotted decimal notation; the second (2607:f6d0:0:925a::ab43:d7c8) is a longer and more complicated looking IPv6 address.

Although the response came from the local DNS server (with IP address 8.8.8.8), it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer. Non-authoritative answer simply means the answer is not fetched from the authoritative DNS server for the queried domain name.

In addition to using nslookup to query for a DNS "Type=A" record, we can also use nslookup to nslookup to query for a "TYPE=NS" record, which returns the hostname (and its IP address) of an authoritative DNS server that knows how to obtain the IP addresses for hosts in the authoritative server's domain.



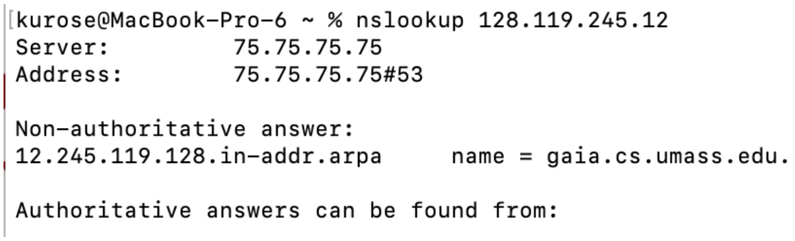
```
C:\WINDOWS\system32>nslookup -type=NS stanford.edu
Server:  dns.google
Address:  8.8.8.8

Non-authoritative answer:
stanford.edu    nameserver = ns7.dnsmadeeasy.com
stanford.edu    nameserver = argus.stanford.edu
stanford.edu    nameserver = atalante.stanford.edu
stanford.edu    nameserver = avallone.stanford.edu
stanford.edu    nameserver = ns5.dnsmadeeasy.com
stanford.edu    nameserver = ns6.dnsmadeeasy.com
```

Figure 2: using nslookup to find the authoritative name servers for the stanford.edu domain

In the example in Figure 2, we've invoked nslookup with the option "-type=NS" and the domain "stanford.edu". This causes nslookup to send a query for a type-NS record to the default local DNS server. In words, the query is saying, "please send me the host names of the authoritative DNS for nyu.edu". (When the –type option is not used, *nslookup* uses the default, which is to query for type A records.) The answer, displayed in the above screenshot, first indicates the DNS server that is providing the answer along with six STANFORD DNS name servers. Each of these servers is indeed an authoritative DNS server for the host. However, nslookup also indicates that the answer is "non-authoritative," meaning that this answer came from the cache of some server rather than from an authoritative NYU DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers. (Even though the type-NS query generated by nslookup did not explicitly ask for the IP addresses, the local DNS server returned these "for free" and *nslookup* displays the result.)

nslookup has a number of additional options beyond "-type=NS" that you might want to explore. Here's a site with screenshots of ten popular nslookup uses: https://www.cloudns.net/blog/10-most-used-nslookup-commands/ and here are the "man pages" for nslookup: https://linux.die.net/man/1/nslookup.

Lastly, we sometimes might be interested in discovering the name of the host associated with a given IP address, i.e., the reverse of the lookup shown in Figure 1 (where the host's name was known/specified and the host's IP address was returned). nslookup can also be used to perform this so-called "reverse DNS lookup." In Figure 3, for example, we specify an IP address as the nslookup argument (128.119.245.12 in this example) and nslookup returns the host name with that address (gaia.cs.umass.edu in this example)



```
[kurose@MacBook-Pro-6 ~ % nslookup 128.119.245.12
Server:         75.75.75.75
Address:        75.75.75.75#53

Non-authoritative answer:
12.245.119.128.in-addr.arpa     name = gaia.cs.umass.edu.

Authoritative answers can be found from:
```

Figure 3: using nslookup to perform a "reverse DNS lookup"

Now that we've provided an overview of nslookup, it's time for you to test drive it yourself. Do the following (and write down the results[1]).

1. Run nslookup to obtain the IP address of the web server for the FAST-NU. What is the IP address of ***nu.edu.pk*? [1]**

2. Run nslookup to obtain the IP address of the web server for the Flex. What is the IP address of ***flex.nu.edu.pk*? [1].**

3. What is the IP address of the DNS server that provided the answer to your nslookup command in question 1 above? Also, write the command to get this answer. [2]

4. Did the answer to your nslookup command in question 1 above come from an authoritative or non-authoritative server? [1]

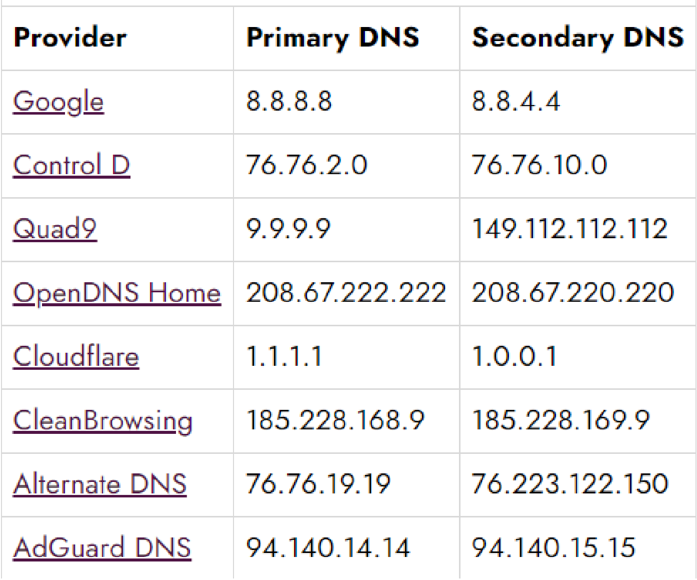
5. What is the IP address and name of the Authoritative server of nu.edu.pk? Also, write the command to get this answer. [2]

6. MX records store all relevant Mail Exchange server data. This information is used to route all email requests for the domain to the appropriate mail server. Use the type mx for the domain 'nu.edu.pk' and list down the mail servers used by the nu domain [2].

7. Use the nslookup command to determine the name of the authoritative name server for the iit.ac.in domain. What is that name? (If there are more than one authoritative servers, what is the name of the first authoritative server returned by nslookup)? If you had to find the IP address of that authoritative name server, how would you do so? [2]

8. In this example, we indicate that we want to the query sent to a DNS server (choose one from the list below) rather than to the default DNS server (dns-dns.google). Thus, the query and reply transaction takes place directly between our querying host and the other DNS server. Write the command for this lookup and also the name/IP of the DNS server used. [4]

   (syntax: lookup domain_name DNS_server_ip/name)



Provider	Primary DNS	Secondary DNS
Google	8.8.8.8	8.8.4.4
Control D	76.76.2.0	76.76.10.0
Quad9	9.9.9.9	149.112.112.112
OpenDNS Home	208.67.222.222	208.67.220.220
Cloudflare	1.1.1.1	1.0.0.1
CleanBrowsing	185.228.168.9	185.228.169.9
Alternate DNS	76.76.19.19	76.223.122.150
AdGuard DNS	94.140.14.14	94.140.15.15

2. BIND 9 [10]

“dig” is a robust command-line tool developed by BIND for querying DNS nameservers. It can identify IP address records, record the query route as it obtains answers from an authoritative nameserver, diagnose other DNS problems.

How to install BIND 9: https://phoenixnap.com/kb/dig-windows

1. Use dig command for nu.edu.pk and analyses the query and answer sections. It should provide information on the record type a DNS server returns. Provide a screenshot. [2]

2. Lets query our DNS server for information on the Root servers. Just type ‘dig ns’ and should see the list of root servers. If this does not work then use the dig command for dns.google (dig 8.8.8.8). You should see a list of root servers comes for the dns.google. Provide your screenshot [2]

3. Lets directly query about ‘nu.edu.pk’ from its authoritative server and see the output. Provide the screenshot [2]

4. Getting the list of TLD servers for .edu (provide a screenshot) [2]

5. Getting the list of ‘com.pk’ servers from ‘a.root-servers.net (provide a screenshot) [2]

QUESTION # 2

Introduction

In this task we will use existing software to examine remote machines and local traffic as well as play the part of a network attacker. Parts one and two should show you how a simple port scan can reveal a large amount of information about a remote server, as well as teach you how to use Wireshark to closely monitor and understand network traffic observable by your machine.

Part 1: Port scanning [17]

Port scanning is a method that can be used by an attacker to probe which ports are open on a given host, learning details about which software the server is running on publically-addressable interfaces. With this information, an attacker gains a better understanding of where and how to attack the victim server. Port scanning takes advantage of conventions in TCP and ICMP that seek to provide a sender with (perhaps too much!) information on why their connection failed.

In this part, you will use the nmap tool (https://nmap.org; https://en.wikipedia.org/wiki/Nmap) to scan the server **scanme.nmap.org**. By doing so, you should be able to see the powerful information that a simple scan can reveal. In your scan, make sure to:

1. Only scan **scanme.nmap.org!** Do not scan any other servers. You should only scan a

server if you have explicit permission from the server operator to do so.

2. Record the traffic with Wireshark (see part 2)
3. Use a TCP Connect scan . (Hint: Read the nmap man pages to find the appropriate flag to use.)
4. Enable OS detection, version detection, script scanning, and traceroute. (Hint: This is asingle flag.)
5. Do a quick scan (-T4).
6. Scan all ports.

When you get the result, report the following about the target server (scanme.nmap.org) based on the results of the scan:

1. What is the full command you used to run the TCP port scan (including arguments)? [2]

2. What is the full command you used to run the UDP port scan (including arguments)? [2]

3. What is the IP address of scanme.nmap.org? Please provide a screenshot of your scan. [2]

4. What ports are open on the target server? What applications are running on those ports? (For this part, you only need to report the service name printed by nmap.) [3]

Port	Service

5. The target machine is also running a webserver. What webserver software and version is being used? What ports does it run on? Please provide the full command and the screenshot of your scan. [3]

6. Find the IP of the **nu.edu.pk** web server. Then use the nslookup on this IP to find out the name of the server name. Please provide the full command and the screenshot of your scan. [5]

- ○ Check the Network Tab.
- ○ Reload the page.
- ○ View the Remote Address, under the Resource's Headers › General.

Part 2: Wireshark packet sniffing [13]

Wireshark is a tool to monitor local network traffic. Wireshark has access to complete header information of all packets on a monitored interface and presents a helpful GUI for understanding the structure of different protocols. Because of this it can be a valuable debugging tool for networking projects, as you will see in part 4.

A) Use the Wireshark packet analyzer (https://www.wireshark.org/; https://en.wikipedia.org/wiki/Wireshark) to examine the traffic generated by nmap duringthe scan in Part 1. You will need to start Wireshark and record traffic on the interface nmap willuse to scan before actually running the scan. The VM we provide has wireshark installed, just run**wireshark**. You can also install and run wireshark locally.

When you get the result, take a look at the Wireshark capture. Use Wireshark's filtering functionality to look at how nmap scans a single port. Report the following about the target server based on the results of the scan:

1. Write down the IP address of your machine and the IP address of destination server. Also, provide a screenshot with Highlighted IP addresses [2]

2. What does it mean for a port on scanme.nmap.org to be “closed?” More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “closed?” Justify your answer with the screenshots from the captured file in wireshark. [4].

3. What does it mean for a port on scanme.nmap.org to be “filtered?” More specifically, what is the TCP packet type, if any, the server gives in response to a SYN packet sent to port that is “filtered?” [hint: look for port: 25 with full TCP handshake (-sT), -p can scan only a specific port for specified host). Justify your answer with the screenshots from the captured file in wireshark. [4]

4. In addition to performing an HTTP GET request to the webserver, what other http request types does nmap send? [3]

Once again, please answer all questions briefly; no response should take more than three sentences.

QUESTION #3 [10]

Now that you know what port scanning is and have used Nmap. Your next task is to implement a basic port scanner that performs the following scans:

- XMAS Scan
- SYN/Stealth Scan
- FIN Scan
- NULL Scan
- ACK Scan

Research the different flags that must be on in the packet and what particular response indicates that a port is open. Along with the code, attach a screenshot of the network traffic (using Wireshark) during the port scan.