# Department of Computer Science

**EE353: Computer Networks**

**Class: BSCS7AB**

# Lab#5 : Domain Name System(DNS)

# Date: 14 OCT 2019

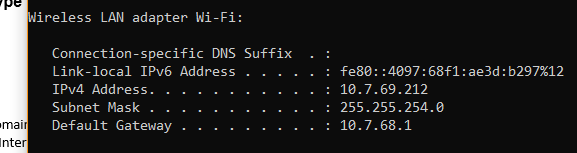
**Lab Engineer: Kaleem Ullah**

# Instructor: Dr. Muhammad Zeeshan

# ***Lab Title:*** Domain Name System(DNS)

**Name: M. Hasnain Naeem Regn. No.: 212728**

**IP Screen Shot:**



***Objective of this lab:***

As described in Section 2.5 of the textbook, 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. Much can go on“under the covers,” invisible to the DNS clients, 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.

Before beginning this lab, you’ll probably want to review DNS by reading Section 2.5 of the textbook. In particular, you may want to review the material on **local DNS servers**, **DNS** **caching**, **DNS records and messages**, and the **TYPE field** in the DNS record.

***Instructions:***

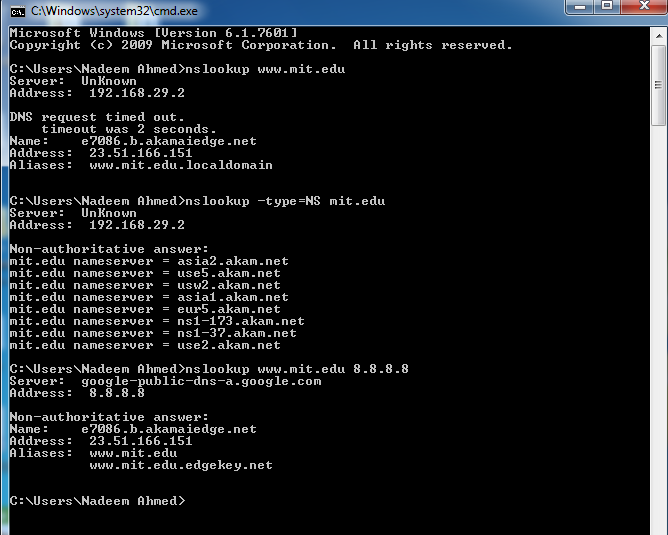
* *Read carefully before starting the lab.*
* *These exercises are to be done individually.*
* *You are supposed to provide the answers to the questions listed at the end of this document and upload the completed report to your course’s LMS site.*
* *Avoid plagiarism by copying from the Internet or from your peers. You may refer to source/ text but you must paraphrase the original work.*
* *Complete the lab half an hour before the lab ends.*
* *At the end of the lab, an online Quiz will be conducted to evaluate your understanding.*

1. **Introduction to DNS**

There are two ways to identify a host -- a hostname and an IP address.  People prefer the more mnemonic hostname identifier, while routers prefer fixed-length, hierarchically-structured IP addresses. In order to reconcile these different preferences, we need a directory service that translates hostnames to IP addresses. This is the main task of the Internet's **Domain Name System (DNS)**.  The DNS is (i) a distributed database implemented in a hierarchy of **name servers**and (ii) an application-layer protocol that allows hosts and name servers to communicate in order to provide the translation service.

1. **Introduction to NSLOOKUP**

In this lab, we’ll make extensive use of the *nslookup* tool, which is available in most Linux/Unix and Microsoft platforms today. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. To run it in Windows, open the Command Prompt and run *nslookup* on the command line.

In it is most basic operation, *nslookup* tool allows the host running the tool to query any specified DNS server for a DNS record. The queried DNS server can be a root DNS server, a top -level-domain DNS server, an authoritative DNS server, or an intermediate DNS server (see the textbook for definitions of these terms). To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from thatsame DNS server, and displays the result.

The above screenshot shows the results of three independent *nslookup* commands (displayed in the Windows Command Prompt). In this example, the client host is using Windows 7 and is connected to an ISP using ADSL modem. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is 192.168.29.2. Consider the first command:

1. **nslookup www.mit.edu**

In words, this command is saying “please send me the IP address for the host www.mit.edu”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the local DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address (of a CDN node hosting the site) www.mit.edu. Although the response came from the local DNS server, it is quite possible that this local DNS server iteratively contacted several other DNS servers to get the answer, as described in Section 2.5 of the textbook.

Now consider the second command:

1. **nslookup –type=NS mit.edu**

In this example, we have provided the option “-type=NS” and the domain “mit.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 servers for mit.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 (which is the default local DNS server) along with MIT nameservers (again hosted on Akamai CDN). Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. 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 MIT DNS server. Note that the answer does not include the IP addresses of the authoritative DNS servers for MIT.

Now finally consider the third command:

1. **nslookup www.mit.edu 8.8.8.8**

In this example, we indicate that we want the query sent to the Google public DNS server (well known IP 8.8.8.8) rather than to the default local DNS server (192.168.29.2). Thus, the query and reply transaction takes place directly between our querying host and Google public DNS server. In this example, the Google DNS server provided the IP address of the host [www.mit.edu](http://www.mit.edu) that is similar to the response in first nslookup.

Now that we have gone through a few illustrative examples, you are perhaps wondering about the general syntax of *nslookup* commands. The syntax is:

**nslookup –option1 –option2 host-to-find dns-server**

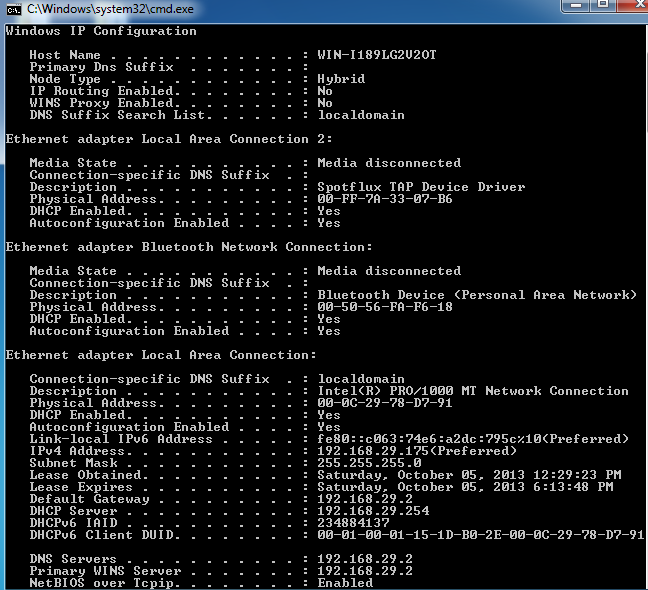
In general, *nslookup* can be run with zero, one, two or more options. And as we have seen in the above examples, the dns-server is optional as well; if it is not supplied, the query is sent to the default local DNS server.

You can also run the nslookup in interactive mode. Simply type nslookup and press enter. Now at the > prompt, type help to see the list of all options available.

1. **Introduction to IPCONFIG**

*ipconfig* (for Windows) and *ifconfig* (for Linux/Unix) are among the most useful littleutilities in your host, especially for debugging network issues. Here we’ll only describe *ipconfig*, although the Linux/Unix *ifconfig* is very similar. *ipconfig* can be used to showyour current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you can get all this information about your host simply by entering:

1. **Ipconfig /all**



into the Command Prompt, as shown in the following screenshot.

*ipconfig* is also very useful for managing the DNS information stored in your host. InSection 2.5 we learned that a host can cache DNS records it recently obtained. To see these cached records, after the prompt C:\> provide the following command:

1. **ipconfig /displaydns**

Each entry shows the remaining Time to Live (TTL) in seconds. To clear the cache, enter:

1. **ipconfig /flushdns**

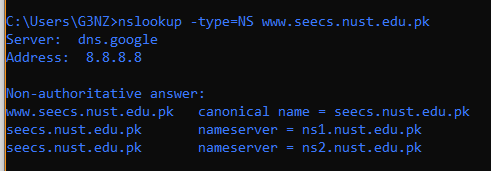
Flushing the DNS cache clears all entries.

***Steps for performing this lab:***

***Exercise 01: nslookup***

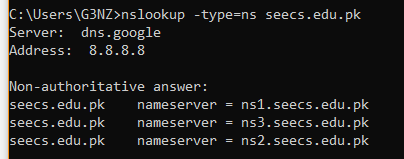
Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. You have to use 8.8.8.8 as your local DNS server for all these exercises. Do the following (and write down the results):

**1.1 Run *nslookup* to obtain the IP address of the Web server hosting www.seecs.nust.edu.pk.**



**1.2 Run *nslookup* to determine the authoritative DNS servers for domain seecs.edu.pk. Provide both the names of these DNS servers and also the IP addresses of one of the DNS servers.**

Nameservers point to the actual servers hosting the content of website. So, here authoritative DNS servers are actually the name servers. Which are:



**IP addresses of nameservers are:**

1. Name: ns1.seecs.edu.pk

Address: 111.68.101.5

1. Name: ns2.seecs.edu.pk

Address: 111.68.101.6

1. Name: ns3.seecs.edu.pk

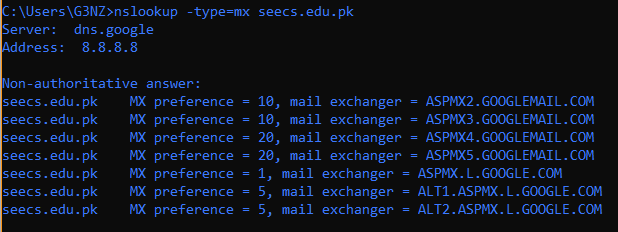
Address: 111.68.101.7

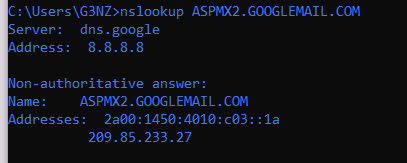
**1.3 Run nslookup to determine the mail servers for** [seecs.edu.pk](http://www.seecs.edu.pk)**. Provide both the names of these Mail servers and also the IP address of one of these Mail servers.**

**Names:**

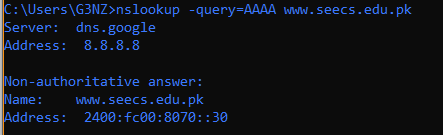
* ASPMX.L.GOOGLE.COM
* ALT1.ASPMX.L.GOOGLE.COM
* ALT2.ASPMX.L.GOOGLE.COM
* ASPMX2.GOOGLEMAIL.COM
* ASPMX3.GOOGLEMAIL.COM
* ASPMX4.GOOGLEMAIL.COM
* ASPMX5.GOOGLEMAIL.COM

**IP Address of ALT2.ASPMX.L.GOOGLE.COM:** 209.85.233.27





**1.4 Query the public DNS service provided by Google at 8.8.8.8 to query for the IPv6 address of www.seecs.edu.pk. Provide the IPv6 address. Note how this address is different from the IPv4 addresses that you were getting for the previous questions.**



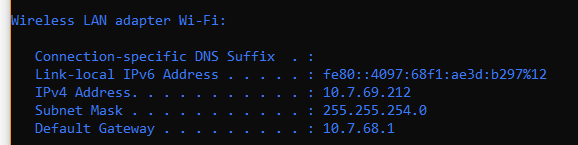
**IPv6 Address:** 2400:fc00:8070::30

**Difference:** Ipv4 uses 32 bit address and Ipv6 address uses 128 bit address.

***Exercise 02: Tracing DNS with Wireshark (while using browser)***

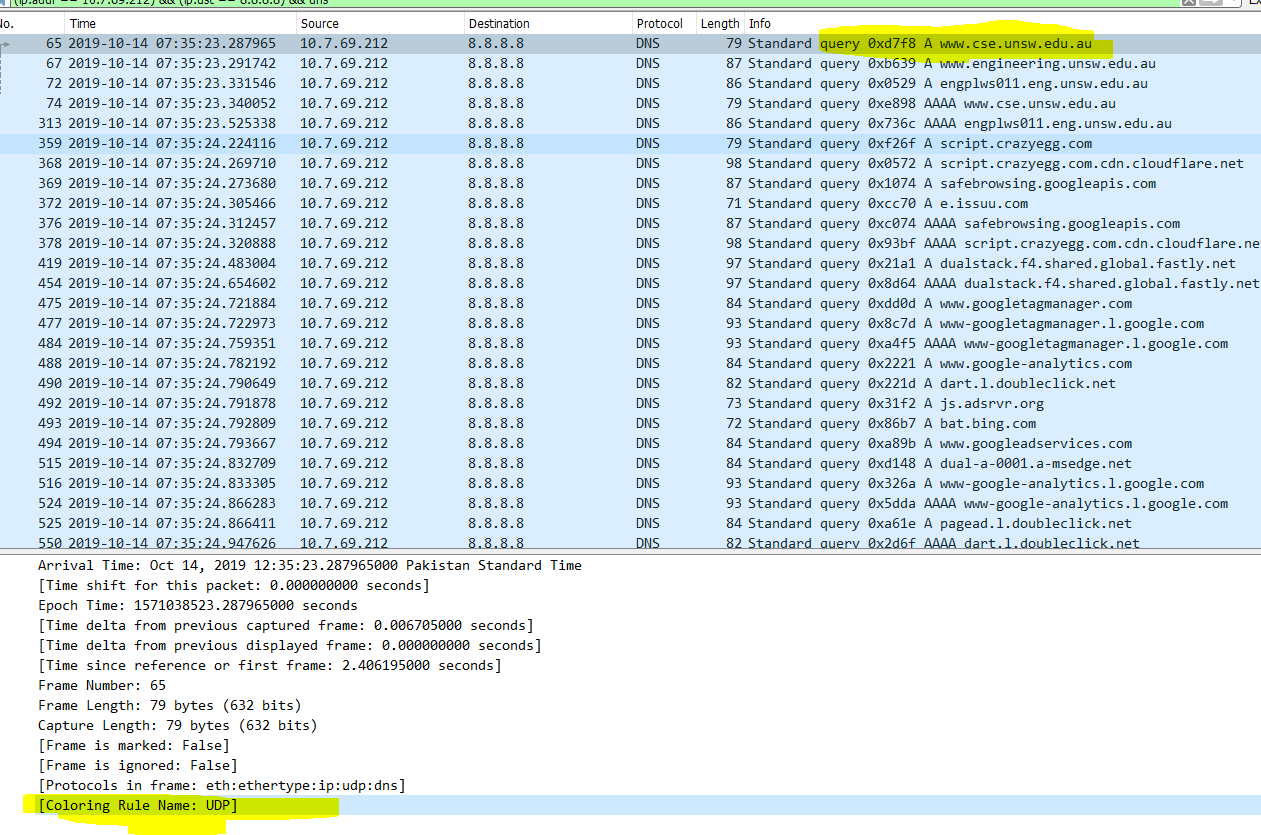
Now that we are familiar with *nslookup* and *ipconfig*, we’re ready to get down to some serious business. Let’s first capture the DNS packets that are generated by ordinary Web-surfing activity.

* Use *ipconfig* to empty the DNS cache in your host.
* Open your browser and empty your browser cache. (With Internet Explorer, go to Tools menu and select Internet Options; then in the General tab select Delete Files.)
* Open Wireshark and enter “ip.addr == your\_IP\_address” into the filter, where you obtain your\_IP\_address with ipconfig. This filter removes all packets that neither originate nor are destined to your host.



* Start packet capture in Wireshark.
* With your browser, visit the Web page: http://www.cse.unsw.edu.au
* Stop packet capture.
  1. **Locate the DNS query and response messages. Are these sent over UDP or TCP (i.e., what transport layer protocol is being used)?**

**UDP** is used as transport layer protocol.

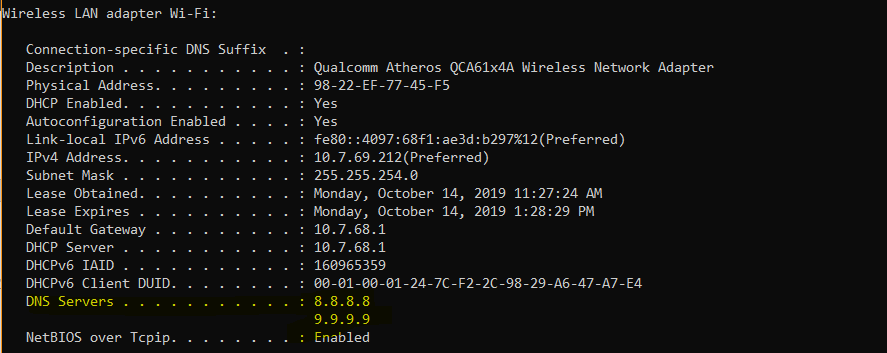


* 1. **To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same?**

**Local DNS Server IP:** 8.8.8.8

**Query DNS Server IP:** 8.8.8.8

Yes, both are same.

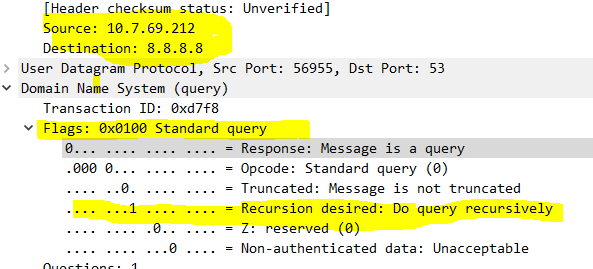




* 1. ***What* is contained in the flag field of your DNS request and response? Explain each set bit.**

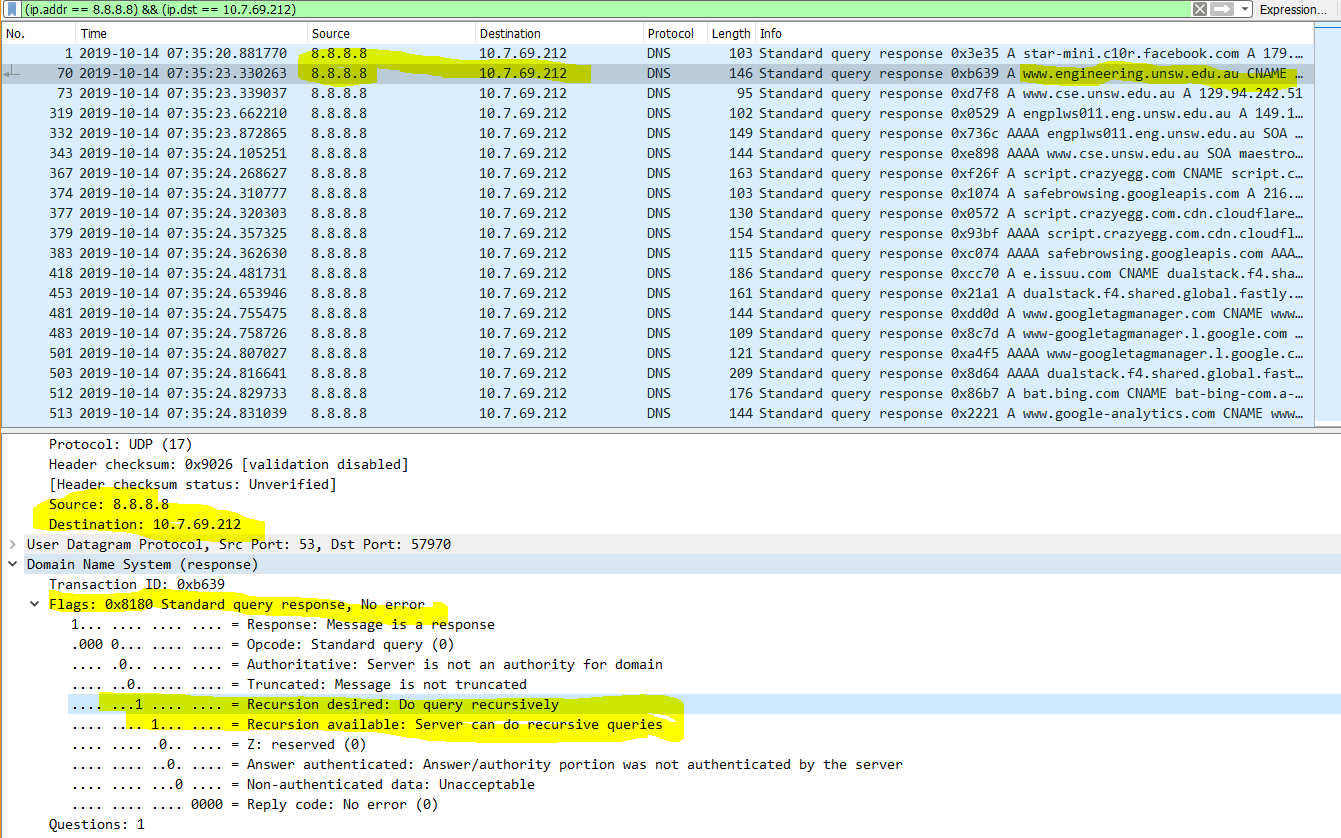
**REQUEST:**

**Query Used:** (ip.addr == 10.7.69.212) && (ip.dst == 8.8.8.8)



**RESPONSE:**

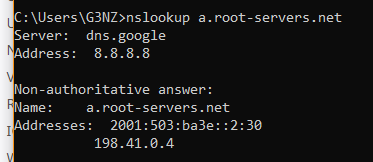
**Query Used:** (ip.addr == 8.8.8.8) && (ip.dst == 10.7.69.212)



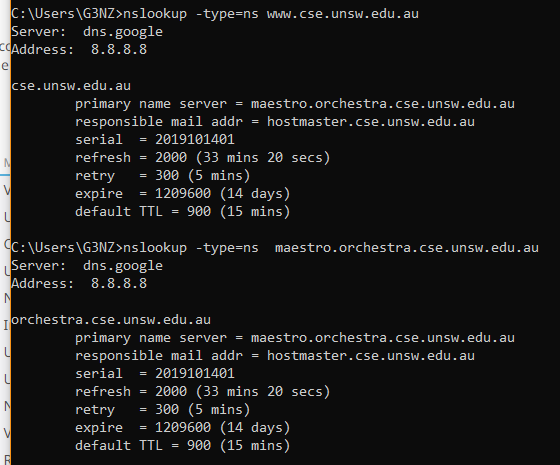
**Flag Bit Explanation:**

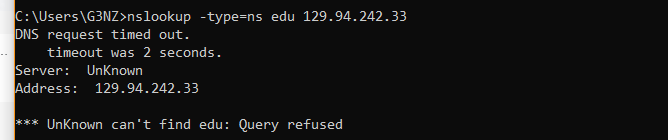
* 0th Bit: Indicates if the message is Request (0) or Response(1)
* 1-4th Bit: Query opcode
* 5th Bit: Indicates if the server is Authoritative (1) or Non-authoritative (0). It is meaningless in request header. In our case, we are receiving the packets from a Non-authoritative server with IP: 8.8.8.8, owned by Google.
* 6th Bit: Indicates if the message is truncated (1) or not truncated (0).
* 7th Bit: Indicates that the Recursion is desired or not for DNS query processing. In our case, it was sent in Request object that recursion is desired. Also, in Response object, query processing through recursion is desired,
* 8th Bit: Indicates if the Recursion is available for the query processing or not. In our case, Response flag object indicates that it is available.
* 9th Bit: 0 if reserved and 1 if not.
* 10th Bit: Answer authenticated if 1, otherwise it is unauthenticated.
* 11th Bit: Specifies whether to accept unauthenticated data or not. 0 means, unauthenticated data should be unacceptable.
  1. **Find out the IP address of at-least one ROOT name server.**

**IP address of a.root-servers.net:** 198.41.0.4



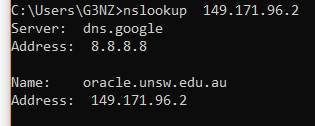
Tried to get to the root-server through nameservers of [www.cse.unsw.edu.au](http://www.cse.unsw.edu.au), but query was refused by its hosting server beethoven.orchestra.cse.unsw.edu.au with IP 129.94.242.33.





* 1. **Find out the fully qualified domain name for IP address 149.171.96.2**

**FQDN of given IP:** oracle.unsw.edu.au



## Conclusion:

Website URLs are resolved through a DNS server hierarchy. We can inspect the DNS servers used for a specific URL and its IP address through various tools to get familiar with the traceroute of a URL.  
For DNS packets, we can snif the packets used for communication with the DNS servers.