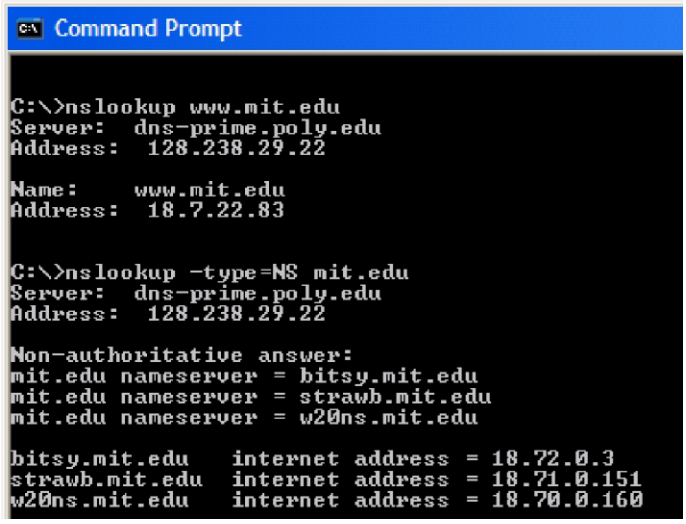
CSC 433: DNS Lab

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.

1. nslookup

In this lab, we'll make extensive use of the *nslookup* tool, available in most platforms today. 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. To accomplish this task, *nslookup* sends a DNS query to the specified DNS server, receives a DNS reply from that same DNS server, and displays the result.



The above screenshot shows the results of two independent *nslookup* commands (displayed in the Windows Command Prompt). In this example, the client host is located on the campus of Polytechnic University in Brooklyn, where the default local DNS server is dns-prime.poly.edu. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is dns-prime.poly.edu. Consider the first command:

```
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 DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of www.mit.edu.

Now consider the second command:

```
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 for mit.edu".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 three MIT nameservers. 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. Finally, the answer also includes the IP addresses of the authoritative DNS servers at MIT.

Now that we have provided an overview of *nslookup*, it is time for you to test drive it yourself. Do the following (and write down the results):

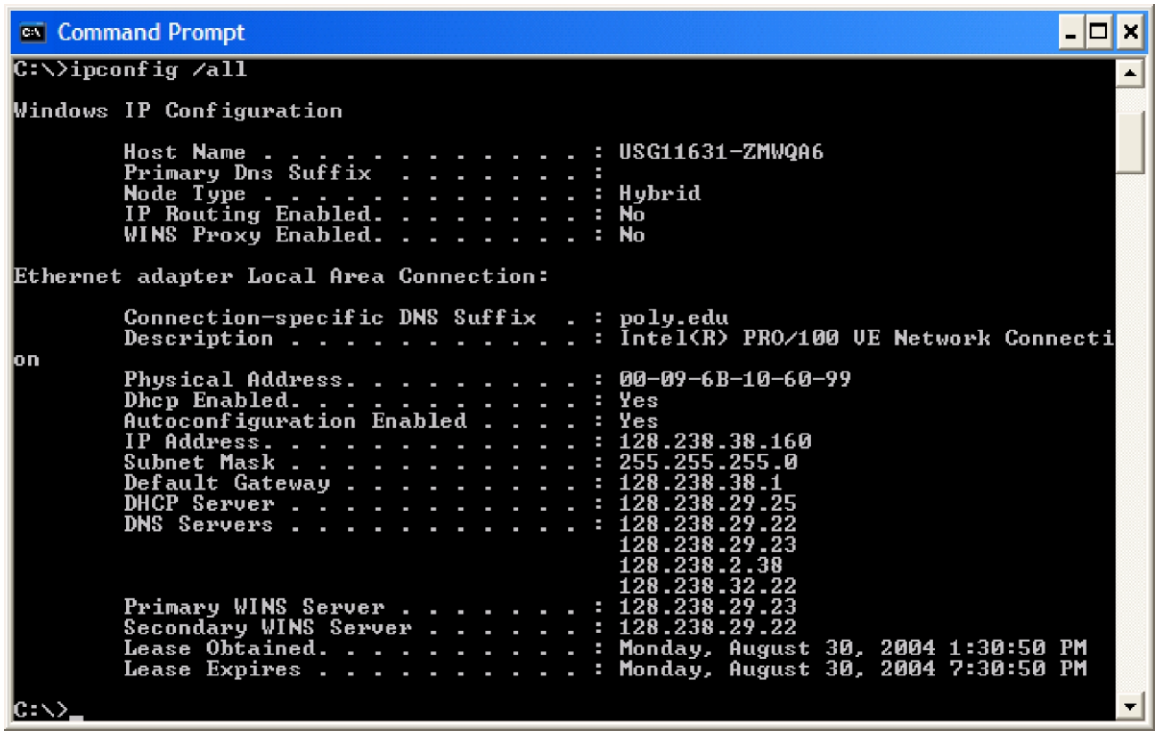
1. Run *nslookup* to obtain the IP address of www.sans.org
2. Run *nslookup* to determine the authoritative DNS servers for www.sans.org.

2. ipconfig

ipconfig (for Windows) and *ifconfig* (for Linux/Unix) are among the most useful little utilities 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 show your current TCP/IP information, including your address, DNS server addresses, adapter type and so on. For example, if you want all this information about your host simply by entering

```
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. A host can cache DNS records it recently obtained. To see these cached records, after the prompt C:\> provide the following command:

```
ipconfig  /displaydns
```

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

enter

```
ipconfig  /flushdns
```

Flushing the DNS cache clears all entries and reloads the entries from the hosts file.

3. Tracing DNS with Wireshark

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.ietf.org
- Stop packet capture.

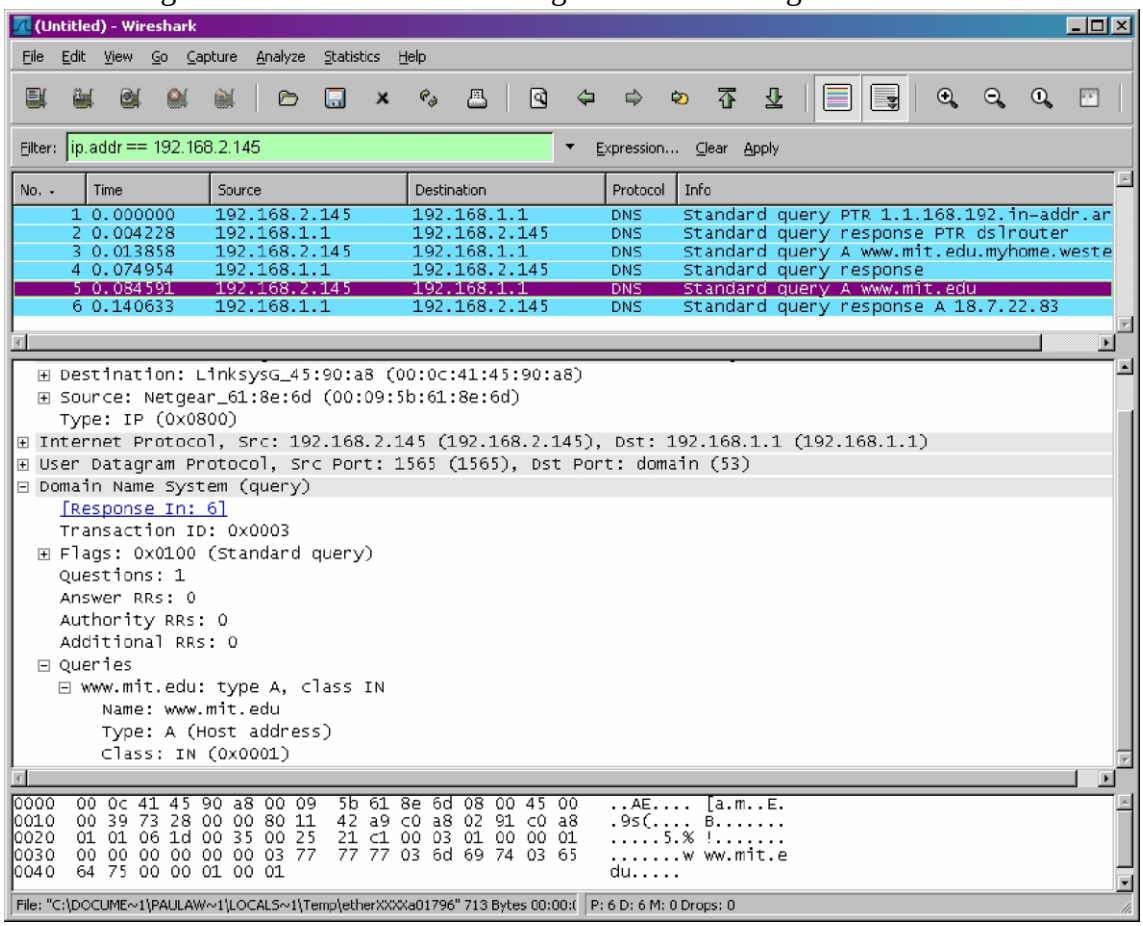
Answer the following questions:

3. Locate the DNS query and response messages. Are then sent over UDP or TCP?
4. What is the destination port for the DNS query message? What is the source port of DNS response message?
5. 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?
6. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
7. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?
8. Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?
9. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now let's play with *nslookup*.

- Start packet capture.
- Do an *nslookup* on www.mit.edu
- Stop packet capture.

You should get a trace that looks something like the following:



We see from the above screenshot that *nslookup* actually sent three DNS queries and received three DNS responses. For the purpose of this assignment, in answering the following questions, ignore the first two sets of queries/responses, as they are specific to *nslookup* and are not normally generated by standard Internet applications. You should instead focus on the last IPv4 query and response messages.

---

10. What is the destination port for the DNS query message? What is the source port of DNS response message?
11. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
12. Examine the DNS query message. What "Type" of DNS query is it? Does the query message contain any "answers"?
13. Examine the DNS response message. How many "answers" are provided? What do each of these answers contain?