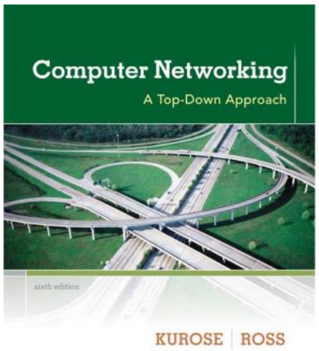
CS 447: Network and Data Communication

Wireshark Lab #03: DNS



You may refer to sections 2.5 in the textbook as a review, before beginning the lab. 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**.

As described in Section 2.5 of the text[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 shown in Figures 2.21 and 2.22 in the textbook, 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.

❖ 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 that same DNS server, and displays the result.

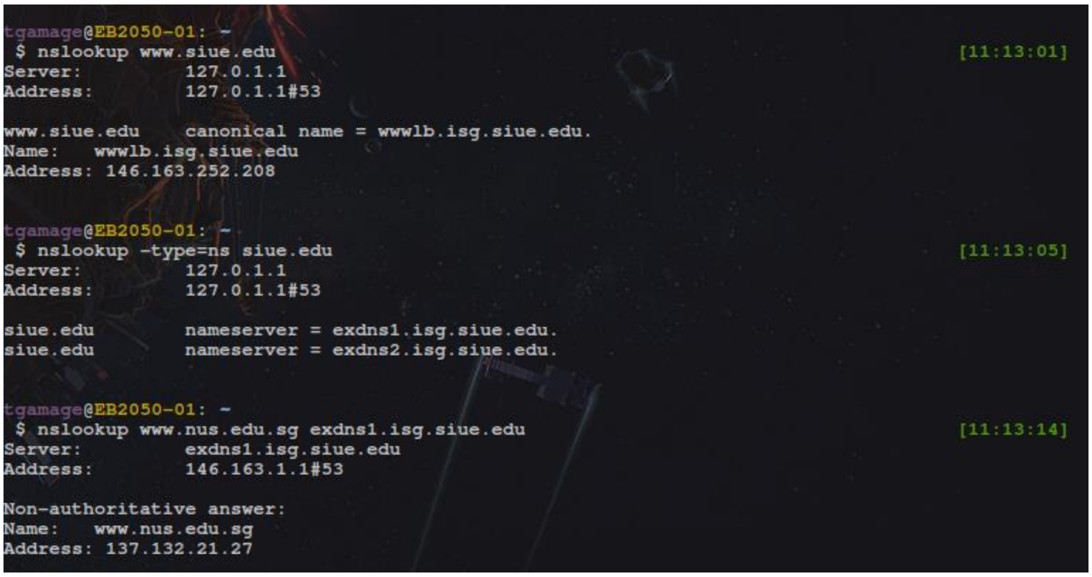
The screenshot below shows the results of three independent *nslookup* commands (displayed in the Linux Command Prompt). In this example, the client host is located on the SIUE campus, where the default local DNS server is localhost. When running *nslookup*, if no DNS server is specified, then *nslookup* sends the query to the default DNS server, which in this case is 127.0.0.1. Consider the first command:

```
nslookup www.siue.edu
```

In words, this command is saying "please send me the IP address for the host www.siue.edu". As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP

[1] References to figures and sections are for Computer Networking, A Top-down Approach, 6th ed., J.F. Kurose and K.W. Ross, Addison-Wesley/Pearson, 2012.

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



```
tgamage@EB2050-01: ~
 $ nslookup www.siue.edu                                                    [11:13:01]
Server:         127.0.1.1
Address:        127.0.1.1#53

www.siue.edu    canonical name = wwwlb.isg.siue.edu.
Name:   wwwlb.isg.siue.edu
Address: 146.163.252.208


tgamage@EB2050-01: ~
 $ nslookup -type=ns siue.edu                                               [11:13:05]
Server:         127.0.1.1
Address:        127.0.1.1#53

siue.edu        nameserver = exdns1.isg.siue.edu.
siue.edu        nameserver = exdns2.isg.siue.edu.


tgamage@EB2050-01: ~
 $ nslookup www.nus.edu.sg exdns1.isg.siue.edu                              [11:13:14]
Server:         exdns1.isg.siue.edu
Address:        146.163.1.1#53

Non-authoritative answer:
Name:   www.nus.edu.sg
Address: 137.132.21.27
```

Now consider the second command:

```
nslookup –type=ns siue.edu
```

In this example, we have provided the option "-type=NS" and the domain "siue.edu". This causes *nslookup* to send a query for a type-NS record to the default local DNS server. In other words, the query is saying, "please send me the host names of the authoritative DNS for siue.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 two SIUE nameservers. Each of these servers is indeed an authoritative DNS server for the hosts on the MIT campus. Note that, if the *nslookup* output indicates that the answer is "non-authoritative," that means that this answer came from the cache of some server rather than from an authoritative DNS server. Finally, the answer also includes the IP addresses of the authoritative DNS servers at SIUE. (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.)

Now finally consider the third command:

```
nslookup www.nus.edu.sg exdns1.isg.siue.edu
```

In this example, we indicate that we want to the query sent to the DNS server exdns1.isg.siue.edu rather than to the default DNS server (127.0.0.1). Thus, the query and reply transaction takes place directly between our querying host and exdns1.isg.siue.edu. In this example, the DNS server exdns1.isg.siue.edu provides the IP address of the host www.nus.edu.sg, which is a web server at the National University of Singapore.

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

```
nslookup [–option] [host-to-find | -(interactive)] [dns-server]
```

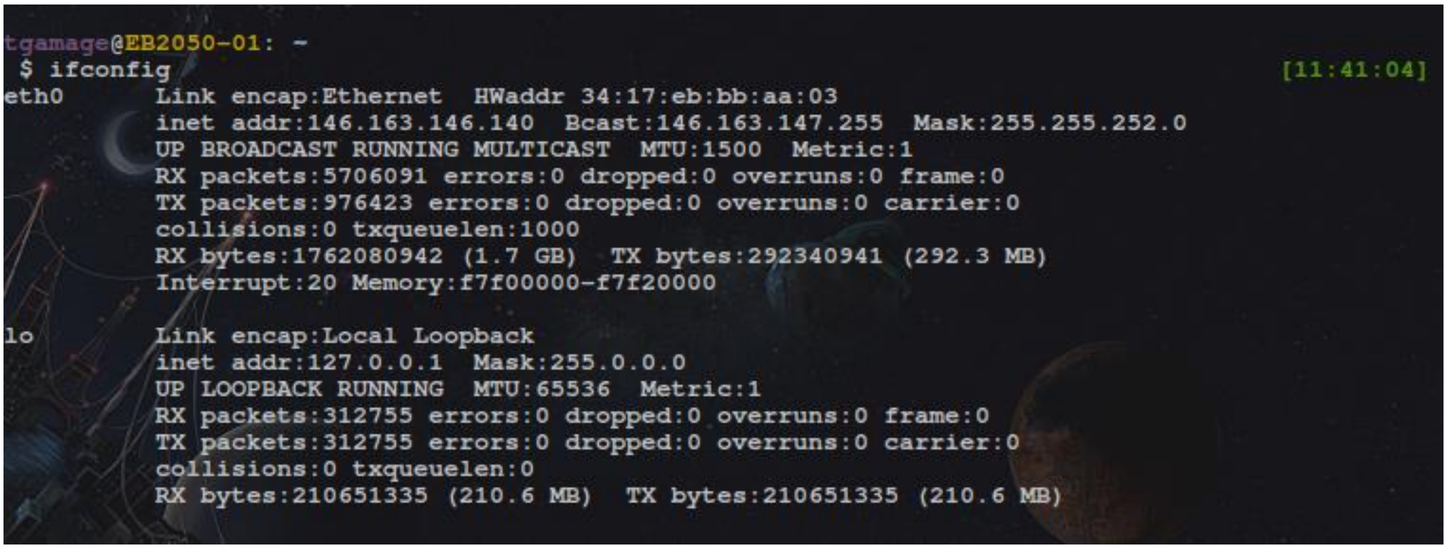
read the man page (type man nslookup on a Linux command prompt) to learn more about this utility. Also, learn about these other Linux utilities – *dig, host* – used for dns lookup.

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 a Web server in Asia. What is the IP address of that server?
2. Run *nslookup* to determine the authoritative DNS servers for a university in Europe.
3. Run *nslookup* so that one of the DNS servers obtained in Question 2 is queried for the mail servers for Yahoo! mail. What is its IP address?

❖ Ifconfig (ipconfig for windows)

ifconfig (for Linux/Unix) and *ipconfig* (for Windows) are among the most useful little utilities in your host, especially for debugging network issues. Here we'll only describe *ifconfig,* although the windows version is very similar. *Ifconfig* can be used to show your current TCP/IP information, including your address, MTU, DNS server address and adapter type (in windows) and so on.



```
tgamage@EB2050-01: ~
 $ ifconfig                                                                          [11:41:04]
eth0      Link encap:Ethernet  HWaddr 34:17:eb:bb:aa:03
          inet addr:146.163.146.140  Bcast:146.163.147.255  Mask:255.255.252.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5706091 errors:0 dropped:0 overruns:0 frame:0
          TX packets:976423 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1762080942 (1.7 GB)  TX bytes:292340941 (292.3 MB)
          Interrupt:20 Memory:f7f00000-f7f20000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:312755 errors:0 dropped:0 overruns:0 frame:0
          TX packets:312755 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:210651335 (210.6 MB)  TX bytes:210651335 (210.6 MB)
```

The above figure shows the output of running the command `ifconfig` on a Linux machine. Also try `nm-tool` (on Linux)

❖ Tracing DNS with Wireshark

Now that we are familiar with nslookup and ifconfig, 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.

- Flush your local DNS cache
  - On Windows this can be accomplished using *ipconfig /flushdns*
  - On Linux first install *nscd*
    - *`sudo apt-get install nscd`* (if not installed by default)
    - *`sudo nscd restart`* or *`sudo nscd –i hosts`*
- 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.

If you are unable to run Wireshark on a live network connection, you can download a packet trace file that was captured while following the steps above on one of the author’s computers[2]. Answer the following questions. Whenever possible, when answering a question below, you should hand in a printout of the packet(s) within the trace that you used to answer the question asked. Annotate the printout[3] to explain your answer. To print a packet, use *File→Print*, choose *Selected packet only*, choose Packet summary line, and select the minimum amount of packet detail that you need to answer the question.

4. Locate the DNS query and response messages. Are then sent over UDP or TCP?
5. What is the destination port for the DNS query message? What is the source port of DNS response message?
6. 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?
7. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
8. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
9. 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?
10. This web page contains images. Before retrieving each image, does your host issue new DNS queries?

Now let’s play with *nslookup*[4]

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

11. What is the destination port for the DNS query message? What is the source port of DNS response message?

[2] Download the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip and extract the file dns¬ethereal-trace-1. The traces in this zip file were collected by Wireshark running on one of the author’s computers, while performing the steps indicated in the Wireshark lab. 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 dns-ethereal-trace-1 trace file.

[3] What do we mean by “annotate”? If you hand in a paper copy, please highlight where in the printout you’ve found the answer and add some text (preferably with a colored pen) noting what you found in what you’ve highlight. If you hand in an electronic copy, it would be great if you could also highlight and annotate

[4] If you are unable to run Wireshark and capture a trace file, use the trace file dns-ethereal-trace-2 in the zip file http://gaia.cs.umass.edu/wireshark-labs/wireshark-traces.zip

12. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
13. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
14. Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?
15. Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

```
nslookup –type=ns mit.edu
```

Answer the following questions:

16. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server?
17. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
18. Examine the DNS response message. What MIT nameservers does the response message provide? Does this response message also provide the IP addresses of the MIT namesers?
19. Provide a screenshot.

Now repeat the previous experiment, but instead issue the command:

`nslookup www.nus.edu.sg 8.8.8.8` (8.8.8.8 is the google’s public dns server)

Answer the following questions:

20. To what IP address is the DNS query message sent? Is this the IP address of your default local DNS server? If not, what does the IP address correspond to?
21. Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?
22. Examine the DNS response message. How many “answers” are provided? What does each of these answers contain?
23. Provide a screenshot.