README.md

50.005 Lab 6: DNS Lab

Table of Contents

Overview

In NS Module 4, we learnt about the role of the Domain Name System (DNS) in Internet naming and addressing.

In this lab exercise, we will go deeper into DNS by using specialised network tools to perform and analyse DNS queries.

Part 1: Exploring DNS using `dig`

The Domain Information Groper (`dig`) is commonly used for performing DNS lookups

DNS Basics

Question 1

Using `dig` , find the IP address for `thyme.lcs.mit.edu.` What is the IP address?

Answer

- The IP address of `thyme.lcs.mit.edu.` was found to be `18.26.0.122` .

Outputs

> **Output for `dig thyme.lcs.mit.edu.`**
>
> ```
> users-MacBook-Pro:~ user$ dig thyme.lcs.mit.edu.
>
> ; <<>> DiG 9.10.6 <<>> thyme.lcs.mit.edu.
> ;; global options: +cmd
> ;; Got answer:
> ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 17154
> ;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1
>
> ;; OPT PSEUDOSECTION:
> ; EDNS: version: 0, flags:; udp: 512
> ;; QUESTION SECTION:
> ;thyme.lcs.mit.edu.             IN      A
>
> ;; ANSWER SECTION:
> thyme.lcs.mit.edu.      1799    IN      CNAME   mercury.lcs.mit.edu.
> mercury.lcs.mit.edu.    1799    IN      A       18.26.0.122
>
> ;; Query time: 535 msec
> ```

```
;; SERVER: 192.168.2.101#53(192.168.2.101)
;; WHEN: Thu Apr 16 02:05:51 +08 2020
;; MSG SIZE  rcvd: 84
```

Question 2

The `dig` answer for the previous question includes a record of type `CNAME`. What does `CNAME` mean?

Answer

- `CNAME` stands for **canonical name**. It is used to redirect a domain name to another domain name.
- For instance, `thyme.lcs.mit.edu.` has a `CNAME` record with value `mercury.lcs.mit.edu.`.
    - This means that the domain `thyme.lcs.mit.edu.` should be redirected to `mercury.lcs.mit.edu.`.

Question 3

What is the expiration time for the `CNAME` record?

Answer

- The expiration time for the `CNAME` record was **1799 seconds**.
    - This can be found in the second column of the `CNAME` record (See above).

Question 4

Run the following commands to find out what your computer receives when it looks up `ai` and `ai.` in the `mit.edu` domain.

- *`dig +domain=mit.edu ai`*
- *`dig +domain=mit.edu ai.`*

What are the two resulting IP addresses?

Answer

- The first command `dig +domain=mit.edu ai` did not result in an IP address.
    - An `SOA` record was returned instead.
        - `SOA` (**Start Of Authority**) records contain administrative information about a zone.
- The second command `dig +domain=mit.edu ai.` returned the IP address `209.59.119.34`.

Outputs

Output for `dig +domain=mit.edu ai`

```
users-MacBook-Pro:~ user$ dig +domain=mit.edu ai

; <<>> DiG 9.10.6 <<>> +domain=mit.edu ai
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 45448
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ai.mit.edu.                    IN      A
```

```
;; AUTHORITY SECTION:
ai.mit.edu.             1507    IN      SOA     auth-ns0.csail.mit.edu. bug-domain.csail.mit.edu. 35472 1800
300 604800 14400

;; Query time: 4 msec
;; SERVER: 192.168.2.100#53(192.168.2.100)
;; WHEN: Thu Apr 16 02:19:58 +08 2020
;; MSG SIZE  rcvd: 101
```

Output for `dig +domain=mit.edu ai.`

```
users-MacBook-Pro:~ user$ dig +domain=mit.edu ai.

; <<>> DiG 9.10.6 <<>> +domain=mit.edu ai.
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 39815
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;ai.                            IN      A

;; ANSWER SECTION:
ai.                     30905   IN      A       209.59.119.34

;; Query time: 3 msec
;; SERVER: 192.168.2.100#53(192.168.2.100)
;; WHEN: Thu Apr 16 02:20:32 +08 2020
;; MSG SIZE  rcvd: 47
```

Question 5

Why are the results for both queries different? Look up the manual for `dig` to find out what the `+domain` parameter does.

Based on the output of the two commands, what is the difference between the DNS searches being performed for `ai` and `ai.` ?

Answer

- `+domain` is a query option that modifies a query to search within a specific domain.
- By specifying `+domain` , `dig` will only search within the zone specified under `+domain` for a given domain name.
- Thus, the command `dig +domain=mit.edu ai` will search for the domain name `ai` under the zone `mit.edu` .
  - This happens to be the domain for the now-defunct MIT Artificial Intelligence Laboratory.
    - The MIT Artificial Intelligence Laboratory and the MIT Laboratory for Computer Science merged to form CSAIL on July 1, 2003.
    - Visiting `ai.mit.edu` on a web browser will actually redirect to the CSAIL website.
  - Since `dig +domain=mit.edu ai` only returned an `SOA` record, we can conclude that `ai.mit.edu` does not have any `A` records
- Meanwhile, the command `dig +domain=mit.edu ai.` will search for the domain name `ai.` under the zone `mit.edu` .
- However, because `ai.` ends with a `.` , it is treated as a **top-level domain** which is queried from the root domain `.` .

- `ai` happens to be the country code top-level domain for the country of Anguilla.
- `ai.` actually hosts a webpage containing links to Anguilla domain registration.
  - From the `A` record that was returned, we can see that this webpage is hosted at IP `209.59.119.34`.

Outputs

> **Output for `man dig`**
>
> ```
> NAME
>        dig - DNS lookup utility
>
> ...
>
> ...
>
> QUERY OPTIONS
>        dig provides a number of query options which affect the way in which
>        lookups are made and the results displayed...
>        ...
>
>        ...
>
>        +domain=somename
>            Set the search list to contain the single domain somename, as if
>            specified in a domain directive in /etc/resolv.conf, and enable
>            search list processing as if the +search option were given.
>
>        ...
>
> ...
>
> ...
> ```

Understanding Hierarchy

In the previous section, you ran `dig` without changing the default options.

This causes `dig` to perform a recursive lookup if the DNS server being queried supports it.

Now, you will trace the intermediate steps involved in a performing recursive query by beginning at a root server and manually going through the DNS hierarchy to resolve a host name.

You can obtain a list of all the root servers by running the command `dig . NS`.

Question 6

Use `dig` to query one of the DNS root servers for the IP address of `lirone.csail.mit.edu` without using recursion.

What is the command that you use to do this?

Answer

- I used the command `dig @a.root-servers.net. lirone.csail.mit.edu +norecurs`.

Outputs

> **Output for `dig @a.root-servers.net. lirone.csail.mit.edu +norecurs`**
>
> ```
> users-MacBook-Pro:~ user$ dig @a.root-servers.net. lirone.csail.mit.edu +norecurs
> ```

```
; <<>> DiG 9.10.6 <<>> @a.root-servers.net. lirone.csail.mit.edu +norecurs
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 48552
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lirone.csail.mit.edu.          IN      A

;; AUTHORITY SECTION:
...
edu.                    172800  IN      NS      a.edu-servers.net.
...

;; ADDITIONAL SECTION:
...
a.edu-servers.net.      172800  IN      A       192.5.6.30
...

;; Query time: 199 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
;; WHEN: Thu Apr 16 03:57:24 +08 2020
;; MSG SIZE  rcvd: 844
```

Question 7

Go through the DNS hierarchy from the root until you have found the IP address of `lirone.csail.mit.edu`.

You should disable recursion and follow the referrals manually.

Which commands did you use, and what address did you find?

Answer

- The commands I used were:
  i. `dig @a.root-servers.net. lirone.csail.mit.edu +norecurs`
  ii. `dig @a.edu-servers.net. lirone.csail.mit.edu +norecurs`
  iii. `dig @asia1.akam.net. lirone.csail.mit.edu +norecurs`
  iv. `dig @auth-ns0.csail.mit.edu. lirone.csail.mit.edu +norecurs`
- The IP address of `lirone.csail.mit.edu` was found to be `128.52.129.186`.

Outputs

Output for `dig @a.edu-servers.net. lirone.csail.mit.edu +norecurs`

```
users-MacBook-Pro:~ user$ dig @a.edu-servers.net. lirone.csail.mit.edu +norecurs

; <<>> DiG 9.10.6 <<>> @a.edu-servers.net. lirone.csail.mit.edu +norecurs
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 56161
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 8, ADDITIONAL: 1
```

```
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lirone.csail.mit.edu.          IN      A

;; AUTHORITY SECTION:
...
mit.edu.               172800  IN      NS      asia1.akam.net.
...

;; Query time: 42 msec
;; SERVER: 192.5.6.30#53(192.5.6.30)
;; WHEN: Thu Apr 16 04:01:23 +08 2020
;; MSG SIZE  rcvd: 216
```

Output for `dig @asia1.akam.net. lirone.csail.mit.edu +norecurs`

```
users-MacBook-Pro:~ user$ dig @asia1.akam.net. lirone.csail.mit.edu +norecurs

; <<>> DiG 9.10.6 <<>> @asia1.akam.net. lirone.csail.mit.edu +norecurs
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 43867
;; flags: qr; QUERY: 1, ANSWER: 0, AUTHORITY: 4, ADDITIONAL: 6

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;lirone.csail.mit.edu.          IN      A

;; AUTHORITY SECTION:
...
csail.mit.edu.          1800    IN      NS      auth-ns0.csail.mit.edu.
...

;; ADDITIONAL SECTION:
...
auth-ns0.csail.mit.edu. 1800    IN      A       128.30.2.123
...

;; Query time: 6 msec
;; SERVER: 95.100.175.64#53(95.100.175.64)
;; WHEN: Thu Apr 16 04:01:55 +08 2020
;; MSG SIZE  rcvd: 233
```

Output for `dig @auth-ns0.csail.mit.edu. lirone.csail.mit.edu +norecurs`

```
users-MacBook-Pro:~ user$ dig @auth-ns0.csail.mit.edu. lirone.csail.mit.edu +norecurs

; <<>> DiG 9.10.6 <<>> @auth-ns0.csail.mit.edu. lirone.csail.mit.edu +norecurs
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 27754
;; flags: qr aa; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
```

```
;lirone.csail.mit.edu.          IN      A

;; ANSWER SECTION:
lirone.csail.mit.edu.   1800    IN      A       128.52.129.186

;; Query time: 339 msec
;; SERVER: 128.30.2.123#53(128.30.2.123)
;; WHEN: Thu Apr 16 04:02:33 +08 2020
;; MSG SIZE  rcvd: 65
```

Understanding Caching

Question 8

Without using recursion, query your default DNS server for information about `www.dmoz.org` and answer the following questions.

- *What is the command that you used?*
- *Did your default server have the answer in its cache? How did you know?*
- *How long did the query take?*

Note: If the information was cached, find another host name that was not cached and complete all the questions in this section using that host.

Answer

- I used the command `dig www.who.org +norecurs` .
- My default server did not have the answer in its cache.
  - This is because there was no answer section in the `dig` response.
  - Instead, information about the top-level domain `org` was returned in the authority section.
- The query only took **4** milliseconds.
  - This is because information about common top-level domains such as `org` is usually cached by local DNS name servers.

Outputs

Output for `dig www.who.org +norecurs`

```
users-MacBook-Pro:~ user$ dig www.who.org +norecurs

; <<>> DiG 9.10.6 <<>> www.who.org +norecurs
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2275
;; flags: qr ra; QUERY: 1, ANSWER: 0, AUTHORITY: 6, ADDITIONAL: 10

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.who.org.                   IN      A

;; AUTHORITY SECTION:
...
org.                    31973   IN      NS      b0.org.afilias-nst.org.
...

;; ADDITIONAL SECTION:
```

```
...
b0.org.afilias-nst.org. 36357   IN      A       199.19.54.1
...

;; Query time: 4 msec
;; SERVER: 192.168.2.100#53(192.168.2.100)
;; WHEN: Thu Apr 16 04:33:45 +08 2020
;; MSG SIZE  rcvd: 394
```

Question 9

Query your default DNS server for information about the host in the previous question, using the recursion option this time.

How long did the query take?

Answer

- The query took **430** milliseconds this time.
- This time, an answer was returned.
    - The IP address of `www.who.org.` was found to be `158.232.12.119` .

Outputs

Output for `dig www.who.org`

```
users-MacBook-Pro:~ user$ dig www.who.org

; <<>> DiG 9.10.6 <<>> www.who.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61394
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.who.org.                   IN      A

;; ANSWER SECTION:
www.who.org.            21599   IN      A       158.232.12.119

;; Query time: 430 msec
;; SERVER: 192.168.2.100#53(192.168.2.100)
;; WHEN: Thu Apr 16 04:48:11 +08 2020
;; MSG SIZE  rcvd: 56
```

Question 10

Query your default DNS server for information about the same host without using recursion.

How long did the query take?

Has the cache served its purpose? Explain why.

Answer

- This time, the query only took **3** milliseconds.
- The cache has served its purpose.

- This is because now, the answer can be returned immediately without any need for recursion.

Outputs

> **Output for `dig www.who.org +norecurs` after caching**

```
users-MacBook-Pro:~ user$ dig www.who.org +norecurs

; <<>> DiG 9.10.6 <<>> www.who.org +norecurs
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32708
;; flags: qr ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;www.who.org.                   IN      A

;; ANSWER SECTION:
www.who.org.            20277   IN      A       158.232.12.119

;; Query time: 3 msec
;; SERVER: 192.168.2.100#53(192.168.2.100)
;; WHEN: Thu Apr 16 05:10:13 +08 2020
;; MSG SIZE  rcvd: 56
```

Part 2: Tracing DNS using Wireshark

Wireshark *is a powerful tool used to capture packets sent over a network and analyse the content of the packets retrieved.*

The file `dnsrealtrace.pcapng` *contains a trace of the packets sent and received when a web page is downloaded from a web server over the SUTD network.*

In the process of downloading the web page, DNS is used to find the IP address of the server.

Open the `dnsrealtrace.pcapng` *in Wireshark and answer the following questions.*

Question 1

Locate the DNS query and response messages. Are they sent over UDP or TCP?

Answer

- They are sent over UDP (See `User Datagram Protocol` in outputs).

Outputs

> **Output of `Frame 5` (DNS Query)**

```
Frame 5: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{EB24B36B-D34B-
4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: LiteonTe_f4:af:32 (20:68:9d:f4:af:32), Dst: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f)
Internet Protocol Version 4, Src: 192.168.81.41, Dst: 192.168.2.11
User Datagram Protocol, Src Port: 57763, Dst Port: 53
Domain Name System (query)
```

> **Output of `Frame 6` (DNS Response)**

```
Frame 6: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF_{EB24B36B-D34B-4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f), Dst: LiteonTe_f4:af:32 (20:68:9d:f4:af:32)
Internet Protocol Version 4, Src: 192.168.2.11, Dst: 192.168.81.41
User Datagram Protocol, Src Port: 53, Dst Port: 57763
Domain Name System (response)
```

Question 2

What is the destination port for the DNS query message?

What is the source port of the DNS response message?

Answer

- The destination port for the DNS query message is port `53`.
- The source port of the DNS response message is also port `53`.

Outputs

Output of `Frame 5` (DNS Query)

```
Frame 5: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{EB24B36B-D34B-4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: LiteonTe_f4:af:32 (20:68:9d:f4:af:32), Dst: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f)
Internet Protocol Version 4, Src: 192.168.81.41, Dst: 192.168.2.11
User Datagram Protocol, Src Port: 57763, Dst Port: 53
    Source Port: 57763
    Destination Port: 53
    Length: 42
    Checksum: 0x7230 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Timestamps]
Domain Name System (query)
```

Output of `Frame 6` (DNS Response)

```
Frame 6: 142 bytes on wire (1136 bits), 142 bytes captured (1136 bits) on interface \Device\NPF_{EB24B36B-D34B-4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f), Dst: LiteonTe_f4:af:32 (20:68:9d:f4:af:32)
Internet Protocol Version 4, Src: 192.168.2.11, Dst: 192.168.81.41
User Datagram Protocol, Src Port: 53, Dst Port: 57763
    Source Port: 53
    Destination Port: 57763
    Length: 108
    Checksum: 0x9358 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
    [Timestamps]
Domain Name System (response)
```

Question 3

What is the IP address to which the DNS query message was sent?

Use `ifconfig` to determine the IP address of your local DNS server. Are these two addresses the same?

Answer

- The destination IP of the DNS query message was `192.168.2.11` (See output below).
- The IP address of my local DNS server is `192.168.2.100`. It is not the same address as the above.
    - I was unable to find the IP address of my local DNS server using `ifconfig` (using MacOS Catalina).
    - However, I was able to find that information using `scutil --dns`.

Outputs

> **Output of `Frame 5` (DNS Query)**
>
> ```
> Frame 5: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface \Device\NPF_{EB24B36B-D34B-
> 4538-82BD-2835D4018C53}, id 0
> Ethernet II, Src: LiteonTe_f4:af:32 (20:68:9d:f4:af:32), Dst: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f)
> Internet Protocol Version 4, Src: 192.168.81.41, Dst: 192.168.2.11
>     0100 .... = Version: 4
>     .... 0101 = Header Length: 20 bytes (5)
>     Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
>     Total Length: 62
>     Identification: 0x2423 (9251)
>     Flags: 0x0000
>     Fragment offset: 0
>     Time to live: 128
>     Protocol: UDP (17)
>     Header checksum: 0x4207 [validation disabled]
>     [Header checksum status: Unverified]
>     Source: 192.168.81.41
>     Destination: 192.168.2.11
> User Datagram Protocol, Src Port: 57763, Dst Port: 53
> Domain Name System (query)
> ```

> **Output of `ifconfig | grep inet`**
>
> ```
> users-MacBook-Pro:~ user$ ifconfig | grep inet
>         inet 127.0.0.1 netmask 0xff000000
>         inet6 ::1 prefixlen 128
>         inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
>         inet6 fe80::844:495b:f724:944a%en0 prefixlen 64 secured scopeid 0x4
>         inet 10.12.98.8 netmask 0xffff0000 broadcast 10.12.255.255
>         inet6 fe80::c494:50ff:fed0:7ffa%awdl0 prefixlen 64 scopeid 0x9
>         inet6 fe80::c494:50ff:fed0:7ffa%llw0 prefixlen 64 scopeid 0xa
>         inet6 fe80::f15c:9a8:7e4:2212%utun0 prefixlen 64 scopeid 0xb
>         inet6 fe80::c774:be24:967c:8ed0%utun1 prefixlen 64 scopeid 0xc
> ```

> **Output of `scutil --dns | grep nameserver`**
>
> ```
> users-MacBook-Pro:~ user$ scutil --dns | grep nameserver
>   nameserver[0] : 192.168.2.100
>   nameserver[1] : 192.168.2.101
>   nameserver[0] : 192.168.2.100
>   nameserver[1] : 192.168.2.101
> ```

Question 4

Examine the second DNS query message. What type of DNS query is it?

Does the query message contain any answers?

Answer

- The second DNS query message is a type `A` query for the domain name `updatekeepalive.mcafee.com`.
- It does not contain any answers.

Outputs

> **Output of `Frame 11` (DNS Query)**

```
Frame 11: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface \Device\NPF_{EB24B36B-D34B-4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: LiteonTe_f4:af:32 (20:68:9d:f4:af:32), Dst: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f)
Internet Protocol Version 4, Src: 192.168.81.41, Dst: 192.168.2.11
User Datagram Protocol, Src Port: 64888, Dst Port: 53
Domain Name System (query)
    Transaction ID: 0xa7e6
    Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
    Queries
        updatekeepalive.mcafee.com: type A, class IN
            Name: updatekeepalive.mcafee.com
            [Name Length: 26]
            [Label Count: 3]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
    [Response In: 12]
```

Question 5

Examine the second DNS response message. How many answers are provided?

What does each of these answers contain?

Answer

- Two answers are provided in the second DNS response message for the domain name `updatekeepalive.mcafee.com`.
- The first answer is a `CNAME` record for `updatekeepalive.mcafee.com` with value `updatekeepalive.glb.mcafee.com`.
- The second answer is an `A` record for `updatekeepalive.glb.mcafee.com` with value `161.69.12.13`.

Outputs

> **Output of `Frame 12` (DNS Response)**

```
Frame 12: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface \Device\NPF_{EB24B36B-D34B-4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f), Dst: LiteonTe_f4:af:32 (20:68:9d:f4:af:32)
Internet Protocol Version 4, Src: 192.168.2.11, Dst: 192.168.81.41
User Datagram Protocol, Src Port: 53, Dst Port: 64888
Domain Name System (response)
    Transaction ID: 0xa7e6
    Flags: 0x8180 Standard query response, No error
    Questions: 1
```

```
    Answer RRs: 2
    Authority RRs: 0
    Additional RRs: 0
    Queries
        updatekeepalive.mcafee.com: type A, class IN
            Name: updatekeepalive.mcafee.com
            [Name Length: 26]
            [Label Count: 3]
            Type: A (Host Address) (1)
            Class: IN (0x0001)
    Answers
        updatekeepalive.mcafee.com: type CNAME, class IN, cname updatekeepalive.glb.mcafee.com
            Name: updatekeepalive.mcafee.com
            Type: CNAME (Canonical NAME for an alias) (5)
            Class: IN (0x0001)
            Time to live: 209 (3 minutes, 29 seconds)
            Data length: 22
            CNAME: updatekeepalive.glb.mcafee.com
        updatekeepalive.glb.mcafee.com: type A, class IN, addr 161.69.12.13
            Name: updatekeepalive.glb.mcafee.com
            Type: A (Host Address) (1)
            Class: IN (0x0001)
            Time to live: 3 (3 seconds)
            Data length: 4
            Address: 161.69.12.13
    [Request In: 11]
    [Time: 0.005536000 seconds]
```

Question 6

Locate a TCP SYN packet sent by your host subsequent to the above DNS response.

This packet opens a TCP connection between your host and the web server.

Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message?

Answer

- The destination of the SYN packet is `161.69.12.13`.
    - This corresponds to the value of the `A` record that was returned in the DNS response message.

Outputs

Output of `Frame 13` (TCP SYN)

```
Frame 13: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface \Device\NPF_{EB24B36B-D34B-
4538-82BD-2835D4018C53}, id 0
Ethernet II, Src: LiteonTe_f4:af:32 (20:68:9d:f4:af:32), Dst: CheckPoi_30:5d:5f (00:1c:7f:30:5d:5f)
Internet Protocol Version 4, Src: 192.168.81.41, Dst: 161.69.12.13
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 52
    Identification: 0x27a9 (10153)
    Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x13f7 [validation disabled]
    [Header checksum status: Unverified]
    Source: 192.168.81.41
```

```
    Destination: 161.69.12.13
Transmission Control Protocol, Src Port: 12056, Dst Port: 80, Seq: 0, Len: 0
```