Assignment 02 -- DNS

1. **How can DNS be used to load balance services? Give a concrete explanation for google.com**

Load balancing is the practice of configuring any domain which is a part of the Domain Name system such that client requests are sent directly to be handled by other backend server machines. DNS is used to perform load distribution among replicated servers like Web servers. For busy sites like *facebook.com* client requests are replicated over multiple servers, with each running on a different end system having a different IP address. For replicated web servers a set of IP address is associated to one canonical host name and the DNS database contains this set of IP addresses. When any client requests a DNS query for a name which associated to a address or set of addresses , the server replies with a set of IP addresses in a round robin method for every address in the set.

DNS load balancing by google.com

```
vishdesai@DESKTOP-K19L02N:~$ nslookup ebay.com
Server:         172.24.32.1
Address:        172.24.32.1#53

Non-authoritative answer:
Name:   ebay.com
Address: 66.211.172.37
Name:   ebay.com
Address: 64.4.253.77
Name:   ebay.com
Address: 216.113.179.53
Name:   ebay.com
Address: 66.211.175.229
Name:   ebay.com
Address: 216.113.181.253
Name:   ebay.com
Address: 209.140.148.143
```

```
vishdesai@DESKTOP-K19L02N:~$ nslookup google.com
Server:         172.24.32.1
Address:        172.24.32.1#53

Non-authoritative answer:
Name:   google.com
Address: 142.251.40.78
Name:   google.com
Address: 2607:f8b0:4000:81a::200e
```

Unlike other domains like ebay.com or amazon.com, Google use its own load balancing algorithm and when requested it redirects only one IP address. This is called Anycast DNs which is a traffic routing algorithm used to deliver the contents of the website swiftly and uses stand-alone IP addresses on multiple nodes. Anycast DNS is resilient to DDoS attacks since the traffic is spread across multiple networks and swarming this network with thousands of requests would not overwhelm one server, but the load is distributed amongst many.

2. **DNS has been around since 1985 and the core protocol is still being used today. What is the inherent weakness of DNS (as of RFC1035, excluding DNSSEC)? Give an example of how an attacker might utilize it.**

   The inherent weaknesses of DNS include its connectionless UDP and its recursive iterative approach amongst many. A DNS server configured to support a recursive resolution is vulnerable to the DOS (Denial of Service) attack. The recursive requests are sent to the server until it gets any reply from the server. A recursive DNS server acts as the medium connecting the users to an Authoritative DNS servers. Recursive DNS acts as a critical link for users trying to access websites like amazon.com etc. and hence attackers use this inherent weakness to launch DDoS attacks which are also known as DNS Amplification attacks. The main reason for these attacks is because of the way servers have been deployed. They are deployed in a stand-alone manner(unicast) with no fallout plans for redundancy.

3. **Perform a manual iterative DNS query for mail-relay.iu.edu with dig starting from the root servers. List all commands and their outputs and explain why you issued every command. Do not use tracing features (dig +trace) for your final write-down.**

   DNS iterative query

   An iterative query is a request for the website name that the DNS server replies with the IP address. They are also known as non-recursive queries, When any DNS client requests the server for any website name, the serves sends its best feasible answer else it refers the request to a lower-level DNS i.e., lower-level server is delegated as a higher-level server to be *Authoritative*.

   About the commands

   dig @b-root-servers.net edu q- A

   This command is used to access the root server to obtain a list of servers that are Authority for the domain edu.

```
1.  vishdesai@DESKTOP-K19L02N:/$ dig @b.root-servers.net edu q-A
2.  ;; BADCOOKIE, retrying.
3.  ; <<>> DiG 9.16.1-Ubuntu <<>> @b.root-servers.net edu q-A
4.  ; (2 servers found)
5.  ;; global options: +cmd
6.  ;; Got answer:
```

```
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33745
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 27
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 5e29d22a12f0bb3f01000000614402c321731bb0d61af13b (good)
;; QUESTION SECTION:
;edu.                           IN      A

;; AUTHORITY SECTION:
edu.                    172800  IN      NS      a.edu-servers.net.
edu.                    172800  IN      NS      b.edu-servers.net.
edu.                    172800  IN      NS      c.edu-servers.net.
edu.                    172800  IN      NS      d.edu-servers.net.
edu.                    172800  IN      NS      e.edu-servers.net.
edu.                    172800  IN      NS      f.edu-servers.net.
edu.                    172800  IN      NS      g.edu-servers.net.
edu.                    172800  IN      NS      h.edu-servers.net.
edu.                    172800  IN      NS      i.edu-servers.net.
edu.                    172800  IN      NS      j.edu-servers.net.
edu.                    172800  IN      NS      k.edu-servers.net.
edu.                    172800  IN      NS      l.edu-servers.net.
edu.                    172800  IN      NS      m.edu-servers.net.
;; Query time: 50 msec
;; SERVER: 199.9.14.201#53(199.9.14.201)
;; WHEN: Thu Sep 16 22:51:46 EDT 2021
;; MSG SIZE  rcvd: 855

```

dig @a.edu-servers.net www.indiana.edu q-A

This command is used to run the same query for the NS records we obtain from the root servers i.e., edu namespace!

```
vishdesai@DESKTOP-K19L02N:/$ dig @a.edu-servers.net www.indiana.edu q-A
; <<>> DiG 9.16.1-Ubuntu <<>> @a.edu-servers.net www.indiana.edu q-A
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 4801
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 3, ADDITIONAL: 6
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.indiana.edu.               IN      A

;; AUTHORITY SECTION:
indiana.edu.            172800  IN      NS      dns1.iu.edu.
indiana.edu.            172800  IN      NS      dns2.iu.edu.
indiana.edu.            172800  IN      NS      dns3.iu.edu.

;; ADDITIONAL SECTION:
dns1.iu.edu.            172800  IN      A       134.68.220.8
dns1.iu.edu.            172800  IN      AAAA    2001:18e8:3:220::10
dns2.iu.edu.            172800  IN      A       129.79.1.8
dns2.iu.edu.            172800  IN      AAAA    2001:18e8:2:8::10
dns3.iu.edu.            172800  IN      A       52.23.85.80

;; Query time: 50 msec
```

```
;; SERVER: 192.5.6.30#53(192.5.6.30)
;; WHEN: Thu Sep 16 22:54:37 EDT 2021
;; MSG SIZE  rcvd: 208

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21844
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
```

dig @dns1.iu.edu mail-relay.iu.edu q-A

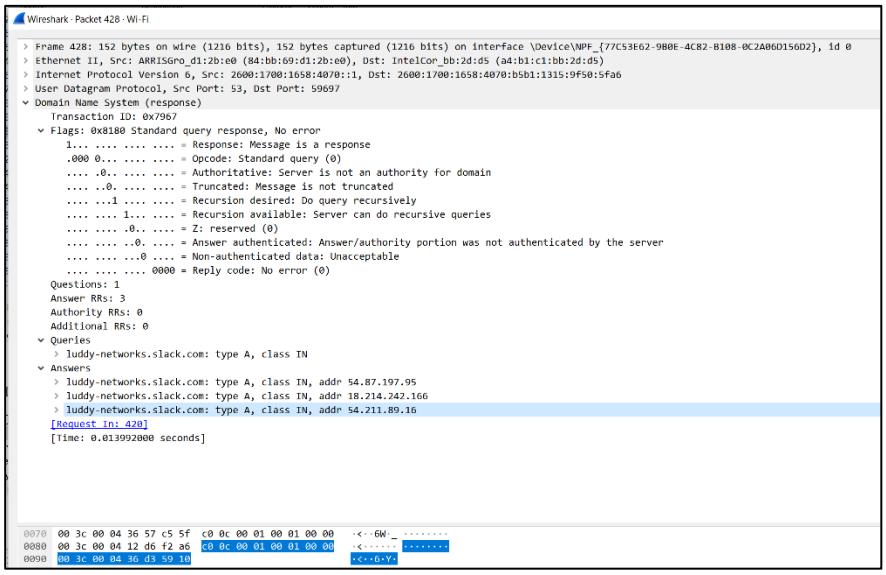
This command is used to access the mail servers i.e., mail-relay.iu.edu

```
vishdesai@DESKTOP-K19L02N:/$ dig @dns1.iu.edu mail-relay.iu.edu q-A
; <<>> DiG 9.16.1-Ubuntu <<>> @dns1.iu.edu mail-relay.iu.edu q-A
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 2930
;; flags: qr aa rd; QUERY: 1, ANSWER: 2, AUTHORITY: 3, ADDITIONAL: 6
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 564b3d9a5eb8e18e25a7b19b614403f9b0ff52ecdfbe60a7 (good)
;; QUESTION SECTION:
;mail-relay.iu.edu.             IN      A

;; ANSWER SECTION:
mail-relay.iu.edu.      300     IN      A       134.68.220.21
mail-relay.iu.edu.      300     IN      A       129.79.1.63

;; AUTHORITY SECTION:
iu.edu.                 3600    IN      NS      dns1.iu.edu.
iu.edu.                 3600    IN      NS      dns2.iu.edu.
iu.edu.                 3600    IN      NS      dns3.iu.edu.

;; ADDITIONAL SECTION:
dns1.iu.edu.            3600    IN      A       134.68.220.8
dns2.iu.edu.            3600    IN      A       129.79.1.8
dns3.iu.edu.            3600    IN      A       52.23.85.80
dns1.iu.edu.            3600    IN      AAAA    2001:18e8:3:220::10
dns2.iu.edu.            3600    IN      AAAA    2001:18e8:2:8::10

;; Query time: 20 msec
;; SERVER: 134.68.220.8#53(134.68.220.8)
;; WHEN: Thu Sep 16 22:56:56 EDT 2021
;; MSG SIZE  rcvd: 267

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 39960
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
; COOKIE: 564b3d9a5eb8e18ed09bd0a6614403f9e37d0f47ad391a30 (good)
;; QUESTION SECTION:
;q-A.                           IN      A

;; Query time: 30 msec
```

```
;; SERVER: 134.68.220.8#53(134.68.220.8)
;; WHEN: Thu Sep 16 22:56:56 EDT 2021
;; MSG SIZE  rcvd: 60
```

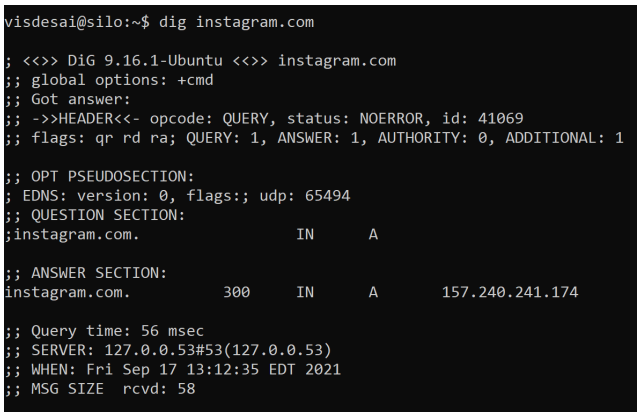
We make use of manual iterative DNS queries because they are much faster as the answers to the queries already exist in the cache and also it gives the DNS request client much greater control. It makes use of referrals which has a list of name servers for the next higher level in the hierarchical structure of DNS.

4. **You are sitting in a coffee shop and are connected to a public WLAN. You fire up Wireshark and start sniffing the traffic of other customers. You notice that all of their traffic is over https so you cannot simply read it. You also notice something striking about the DNS traffic, what is it and what are the implications?**



From the above, in the Domain Name System (response) we can see the domain name, source ports, destination ports, queries, answers and also the response time. The request data structure informs the server of the type of packet (query), number of questions it contains and also the data in the queries while HTTP traffic or TLS traffic does not the display domain name, queries, answers nor the response time.

5. **Suppose that IU has an internal DNS cache. You are an ordinary user (no network admin). Can you determine (and if yes, how) if a given external website was recently accessed?**



Yes, by using the dig command to lookup an external website like instagram.com would return a query time of 56 msec on the first try but executing the commands again would return a query time of 0 msec since the name servers were already stored in the DNS cache. Through this way, we are able to determine if an external website was acessed or not.