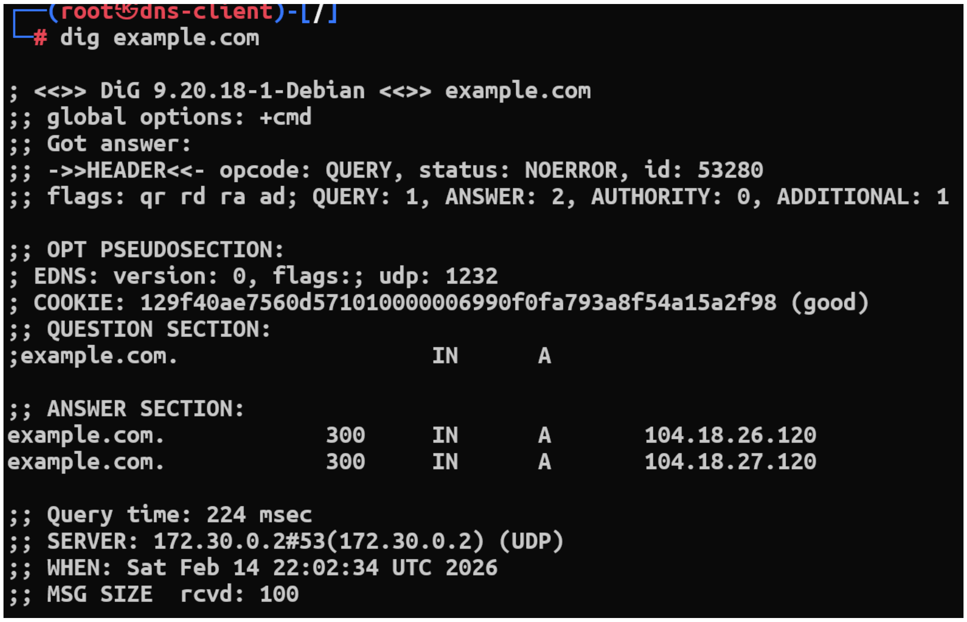
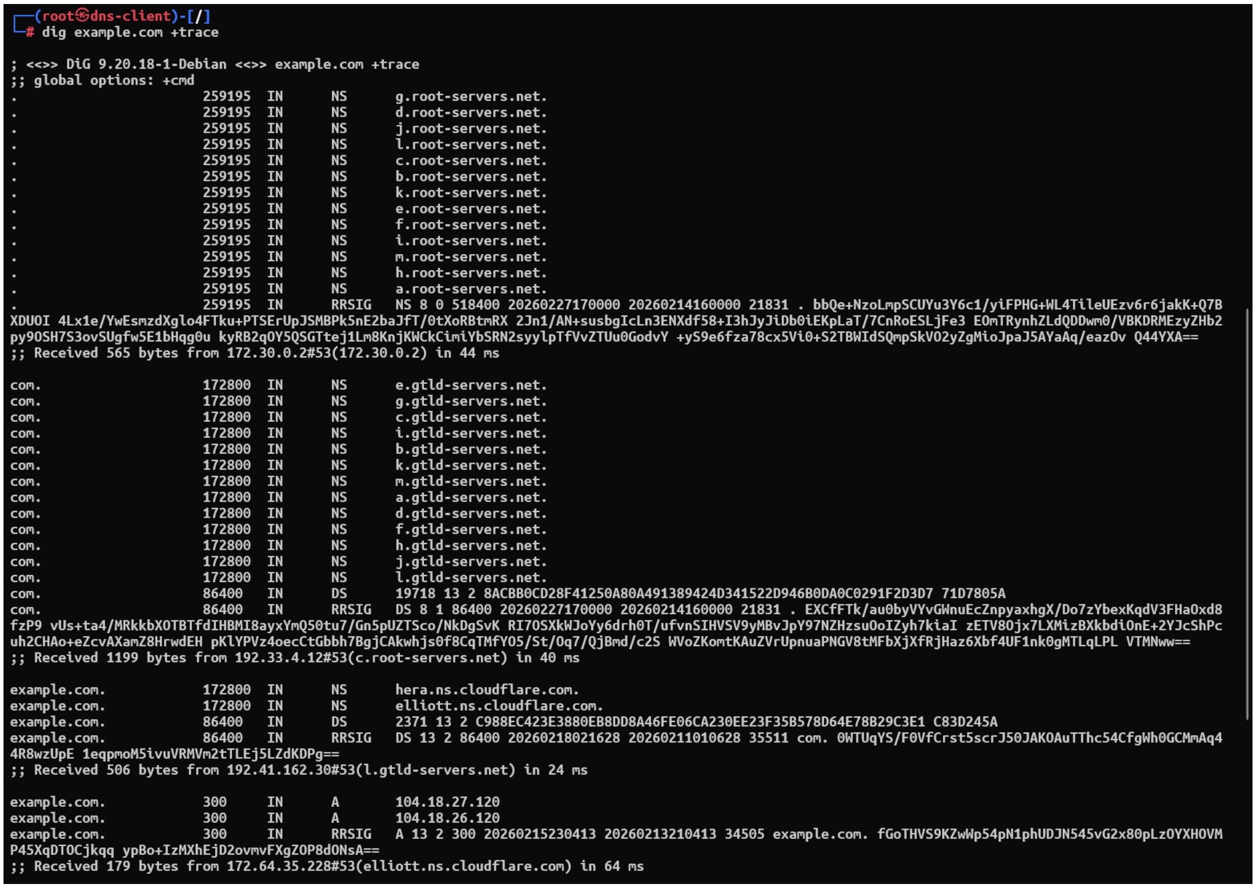
DNS Lab

- Docker environment for this lab.
- Server IP: `172.30.0.2`
- Client IP: `172.30.0.3`

Task 1.1: Exploring DNS Queries



```
┌──(root㉿dns-client)-[/]
└─# dig example.com

; <<>> DiG 9.20.18-1-Debian <<>> example.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53280
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 129f40ae7560d571010000006990f0fa793a8f54a15a2f98 (good)
;; QUESTION SECTION:
;example.com.                   IN      A

;; ANSWER SECTION:
example.com.            300     IN      A       104.18.26.120
example.com.            300     IN      A       104.18.27.120

;; Query time: 224 msec
;; SERVER: 172.30.0.2#53(172.30.0.2) (UDP)
;; WHEN: Sat Feb 14 22:02:34 UTC 2026
;; MSG SIZE  rcvd: 100
```



```
┌──(root㉿dns-client)-[/]
└─# dig example.com +trace

; <<>> DiG 9.20.18-1-Debian <<>> example.com +trace
;; global options: +cmd
.                       259195  IN      NS      g.root-servers.net.
.                       259195  IN      NS      d.root-servers.net.
.                       259195  IN      NS      j.root-servers.net.
.                       259195  IN      NS      l.root-servers.net.
.                       259195  IN      NS      c.root-servers.net.
.                       259195  IN      NS      b.root-servers.net.
.                       259195  IN      NS      k.root-servers.net.
.                       259195  IN      NS      e.root-servers.net.
.                       259195  IN      NS      f.root-servers.net.
.                       259195  IN      NS      i.root-servers.net.
.                       259195  IN      NS      m.root-servers.net.
.                       259195  IN      NS      h.root-servers.net.
.                       259195  IN      NS      a.root-servers.net.
.                       259195  IN      RRSIG   NS 8 0 518400 20260227170000 20260214160000 21831 . bbQe+NzoLmpSCUYu3Y6c1/yiFPHG+WL4TileUEzv6r6jakK+Q7B
XDUOI 4Lx1e/YwEsmzdXglo4FTku+PTSErUpJSMBPk5nE2baJfT/0tXoRBtmRX 2Jn1/AN+susbgIcLn3ENXdf58+I3hJyJiDb0iEKpLaT/7CnRoESLjFe3 EOmTRynhZLdQDDwm0/VBKDRMEzyZHb2
py9OSH7S3ovSUgfw5E1bHqg0u kyRB2qOY5QSGTtej1Lm8KnjKWCkCimiYbSRN2syylpTfVvZTUu0GodvY +yS9e6fza78cx5Vi0+S2TBWIdSQmpSkVO2yZgMioJpaJ5AYaAq/eazOv Q44YXA==
;; Received 565 bytes from 172.30.0.2#53(172.30.0.2) in 44 ms

com.                    172800  IN      NS      e.gtld-servers.net.
com.                    172800  IN      NS      g.gtld-servers.net.
com.                    172800  IN      NS      c.gtld-servers.net.
com.                    172800  IN      NS      i.gtld-servers.net.
com.                    172800  IN      NS      b.gtld-servers.net.
com.                    172800  IN      NS      k.gtld-servers.net.
com.                    172800  IN      NS      m.gtld-servers.net.
com.                    172800  IN      NS      a.gtld-servers.net.
com.                    172800  IN      NS      d.gtld-servers.net.
com.                    172800  IN      NS      f.gtld-servers.net.
com.                    172800  IN      NS      h.gtld-servers.net.
com.                    172800  IN      NS      j.gtld-servers.net.
com.                    172800  IN      NS      l.gtld-servers.net.
com.                    86400   IN      DS      19718 13 2 8ACBB0CD28F41250A80A491389424D341522D946B0DA0C0291F2D3D7 71D7805A
com.                    86400   IN      RRSIG   DS 8 1 86400 20260227170000 20260214160000 21831 . EXCfFTk/au0byVYvGWnuEcZnpyaxhgX/Do7zYbexKqdV3FHaOxd8
fzP9 vUs+ta4/MRkkbXOTBTfdIHBMI8ayxYmQ50tu7/Gn5pUZTSco/NkDgSvK RI7OSXkWJoYy6drh0T/ufvnSIHVSV9yMBvJpY97NZHzsuOoIZyh7kiaI zETV8Ojx7LXMizBXkbdiOnE+2YJcShPc
uh2CHAo+eZcvAXamZ8HrwdEH pKlYPVz4oecCtGbbh7BgjCAkwhjs0f8CqTMfYO5/St/Oq7/QjBmd/c2S WVoZKomtKAuZVrUpnuaPNGV8tMFbXjXfRjHaz6Xbf4UF1nk0gMTLqLPL VTMNww==
;; Received 1199 bytes from 192.33.4.12#53(c.root-servers.net) in 40 ms

example.com.            172800  IN      NS      hera.ns.cloudflare.com.
example.com.            172800  IN      NS      elliott.ns.cloudflare.com.
example.com.            86400   IN      DS      2371 13 2 C988EC423E3880EB8DD8A46FE06CA230EE23F35B578D64E78B29C3E1 C83D245A
example.com.            86400   IN      RRSIG   DS 13 2 86400 20260218021628 20260211010628 35511 com. 0WTUqYS/F0VfCrst5scrJ50JAKOAuTThc54CfgWh0GCMmAq4
4R8wzUpE 1eqpmoM5ivuVRMVm2tTLEj5LZdKDPg==
;; Received 506 bytes from 192.41.162.30#53(l.gtld-servers.net) in 24 ms

example.com.            300     IN      A       104.18.27.120
example.com.            300     IN      A       104.18.26.120
example.com.            300     IN      RRSIG   A 13 2 300 20260215230413 20260213210413 34505 example.com. fGoTHVS9KZwWp54pN1phUDJN545vG2x80pLzOYXHOVM
P45XqDTOCjkqq ypBo+IzMXhEjD2ovmvFXgZOP8dONsA==
;; Received 179 bytes from 172.64.35.228#53(elliott.ns.cloudflare.com) in 64 ms
```

1. Which DNS servers are contacted during resolution?

`172.30.0.2` , `192.33.4.12#53(c.root-servers.net)`, `192.41.162.30#53(l.gtld-servers.net)` , `172.64.35.228#53(elliott.ns.cloudflare.com)`

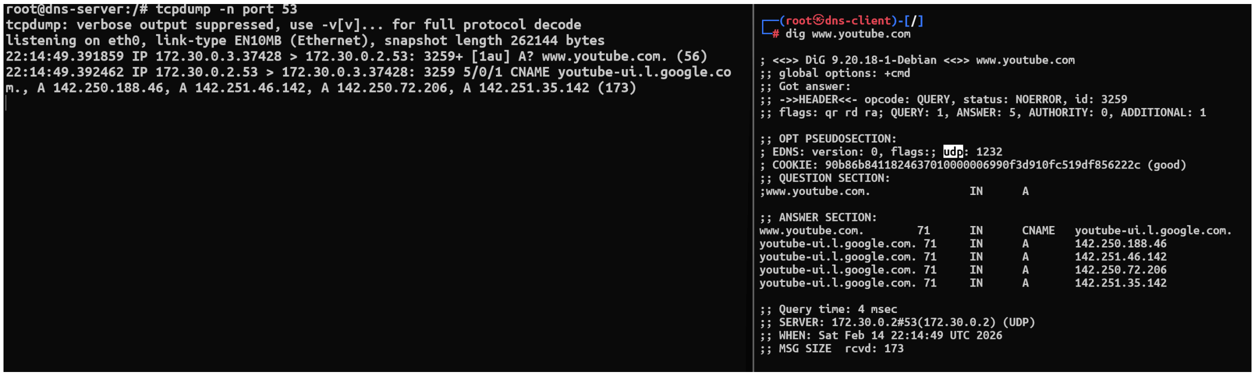
2. What information is returned in a DNS response?

   - The response returns resource records such as the queried name, record type (for example A), class, TTL, and the resolved IP address.

3. At which points could an attacker interfere with the process?

   - Because DNS commonly uses UDP and lacks strong authentication, an attacker can forge a fake response while the client is waiting, enabling spoofing and cache poisoning attacks against the hierarchical DNS infrastructure.

Task 2.1: Capturing DNS Traffic

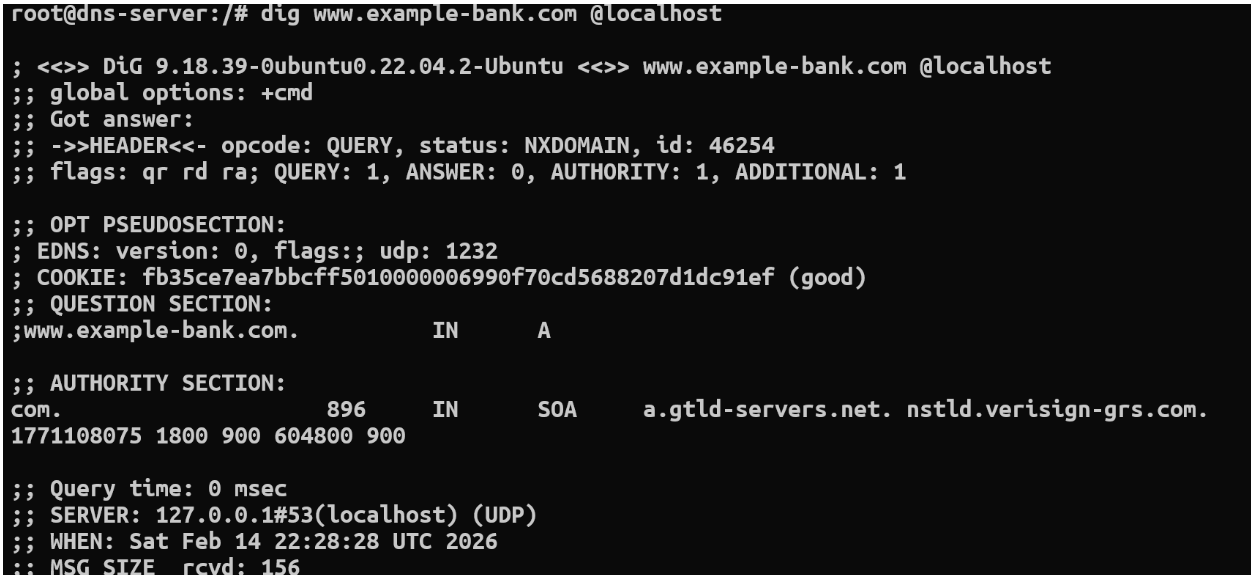


```
root@dns-server:/# tcpdump -n port 53
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:14:49.391859 IP 172.30.0.3.37428 > 172.30.0.2.53: 3259+ [1au] A? www.youtube.com. (56)
22:14:49.392462 IP 172.30.0.2.53 > 172.30.0.3.37428: 3259 5/0/1 CNAME youtube-ui.l.google.co
m., A 142.250.188.46, A 142.251.46.142, A 142.250.72.206, A 142.251.35.142 (173)
```

```
┌──(root㉿dns-client)-[/]
└─# dig www.youtube.com

; <<>> DiG 9.20.18-1-Debian <<>> www.youtube.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 3259
;; flags: qr rd ra; QUERY: 1, ANSWER: 5, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 90b86b84118246370100000006990f3d910fc519df856222c (good)
;; QUESTION SECTION:
;www.youtube.com.               IN      A

;; ANSWER SECTION:
www.youtube.com.        71      IN      CNAME   youtube-ui.l.google.com.
youtube-ui.l.google.com. 71     IN      A       142.250.188.46
youtube-ui.l.google.com. 71     IN      A       142.251.46.142
youtube-ui.l.google.com. 71     IN      A       142.250.72.206
youtube-ui.l.google.com. 71     IN      A       142.251.35.142

;; Query time: 4 msec
;; SERVER: 172.30.0.2#53(172.30.0.2) (UDP)
;; WHEN: Sat Feb 14 22:14:49 UTC 2026
;; MSG SIZE  rcvd: 173
```

1. Is DNS using TCP or UDP by default?
   - On the right side, there is a `udp` after the flags, confirming the transport protocol is UDP.
2. What fields appear in a DNS query and response?
   - `HEADER` : opcode, status, id, flags, QUERY, ANSWER, AITHORITY, ADDITIONAL
   - `OPT PSEUDOSECTION`: EDNS, version, flags, udp, COOKIE, QUESTION SECTION
   - `ANSER SECTION`: domain name, record type (i.e. CNAME, A), IP address
   - `Query time`, `Server`, `When` (timestamp), `MSG SIZE`
3. Why might DNS traffic be vulnerable to spoofing?
   - Since DNS server just response to the source IP address from the DNS query, lacking of authentication and randomization limits, it couldn't verify whether it is the one who has the source IP address send the request. Therefore, it is vulnerable to spoofing.

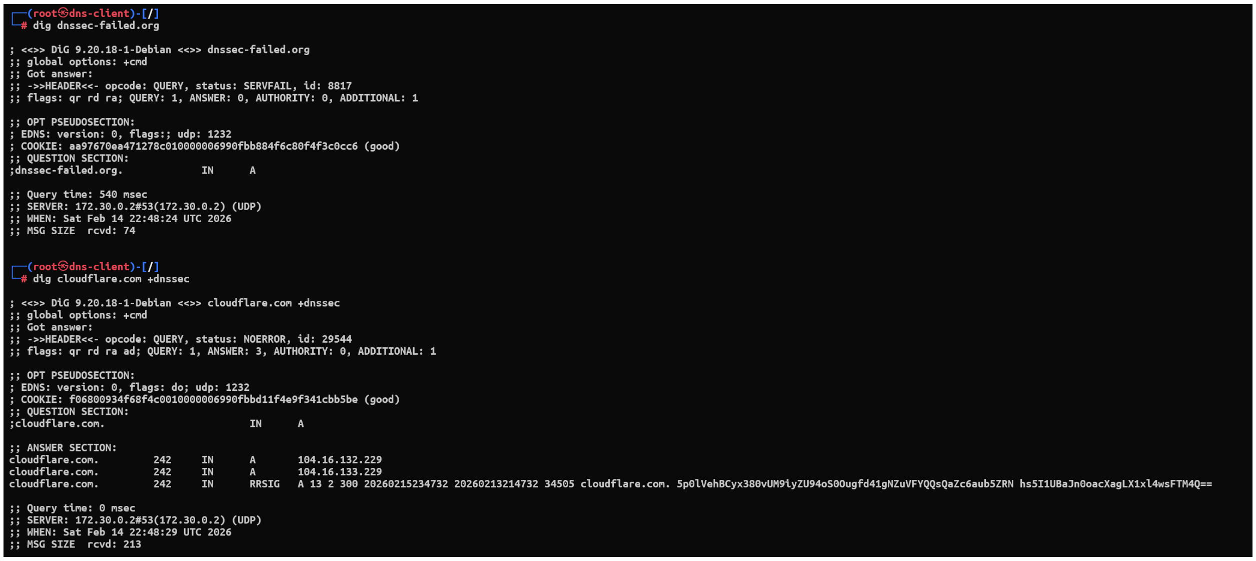
Task 3.1: Query the Local DNS Server



```
root@dns-server:/# dig www.example-bank.com @localhost

; <<>> DiG 9.18.39-0ubuntu0.22.04.2-Ubuntu <<>> www.example-bank.com @localhost
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 46254
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: fb35ce7ea7bbcff5010000006990f70cd5688207d1dc91ef (good)
;; QUESTION SECTION:
;www.example-bank.com.          IN      A

;; AUTHORITY SECTION:
com.                    896     IN      SOA     a.gtld-servers.net. nstld.verisign-grs.com.
1771108075 1800 900 604800 900

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(localhost) (UDP)
;; WHEN: Sat Feb 14 22:28:28 UTC 2026
;; MSG SIZE  rcvd: 156
```

1. Is the response what you would expect?
    - Yes, the status: `NXDOMAIN` means that the domain we ask for doesn’t exist.
2. Why is recursive DNS resolution risky if misconfigured?
    - Recursive DNS resolution is risky when misconfigured because an open or weakly protected resolver can be abused as both a pivot and an amplifier, it can be used to poison users’ view of the DNS, to participate in DDoS attacks, and to leak internal information to the Internet.
3. What security assumptions does cache poisoning break?
    - Cache poisoning breaks the assumption that the recursive resolver’s cache is a trustworthy reflection of the DNS hierarchy.

Task 4.1: Testing DNSSEC Validation



```
┌──(root㉿dns-client)-[/]
└─# dig dnssec-failed.org

; <<>> DiG 9.20.18-1-Debian <<>> dnssec-failed.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 8817
;; flags: qr rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: aa97670ea471278c010000006990fbb884f6c80f4f3c0cc6 (good)
;; QUESTION SECTION:
;dnssec-failed.org.             IN      A

;; Query time: 540 msec
;; SERVER: 172.30.0.2#53(172.30.0.2) (UDP)
;; WHEN: Sat Feb 14 22:48:24 UTC 2026
;; MSG SIZE  rcvd: 74


┌──(root㉿dns-client)-[/]
└─# dig cloudflare.com +dnssec

; <<>> DiG 9.20.18-1-Debian <<>> cloudflare.com +dnssec
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 29544
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 1232
; COOKIE: f06800934f68f4c0010000006990fbbd11f4e9f341cbb5be (good)
;; QUESTION SECTION:
;cloudflare.com.                        IN      A

;; ANSWER SECTION:
cloudflare.com.         242     IN      A       104.16.132.229
cloudflare.com.         242     IN      A       104.16.133.229
cloudflare.com.         242     IN      RRSIG   A 13 2 300 20260215234732 20260213214732 34505 cloudflare.com. 5p0lVehBCyx380vUM9iyZU94oS0Ougfd41gNZuVFYQQsQaZc6aub5ZRN hs5I1UBaJn0oacXagLX1xl4wsFTM4Q==

;; Query time: 0 msec
;; SERVER: 172.30.0.2#53(172.30.0.2) (UDP)
;; WHEN: Sat Feb 14 22:48:29 UTC 2026
;; MSG SIZE  rcvd: 213
```

1. What happens when DNSSEC validation fails?
    - A validating resolver will therefore return status: SERVFAIL instead of an A record when DNSSEC validation fails.
2. How does DNSSEC change the trust model of DNS?
    - It use RRSIG to verify the responded content adding a layer of authentication.
3. What types of attacks does DNSSEC prevent?
    - DNS response forgery and cache poisoning (attackers injecting fake records into resolvers' caches).

Reflection Questions

1. Why is DNS an attractive target for attackers?
    - Basically, just as I answer in Task1 question 3, it uses UDP which doesn't form a connection between devices, so an attacker can easily forge a legitimate response and spoof DNS replies. In addition, almost every Internet connection relies on DNS, so if an attacker can control or poison DNS, they can silently redirect many users to malicious sites, steal credentials, or disrupt services on a large scale.
2. Why is DNS security often overlooked in system design?

- Since it is a hierarchical system, we should consider a whole system rather than a single point, but in practice DNS is often treated as basic infrastructure that “just works” and is delegated to default ISP or cloud resolvers. As a result, designers may focus more on securing applications and firewalls while assuming DNS is trustworthy, overlooking configuration issues (like open resolvers, missing DNSSEC, or weak logging) that can impact the entire system.

3. Would you recommend running an in-house DNS server for an enterprise? Why or
why not?

    - No, if there is a customized device for a DNS server, you need to take extra resource to protect it, widening the attacking surface for attacker and adding operational overhead. Unless the enterprise has strong networking and security expertise to patch, monitor, and harden DNS (including DNSSEC and logging), it may be safer and more efficient to rely on well-managed external resolvers or dedicated DNS providers instead of running its own.