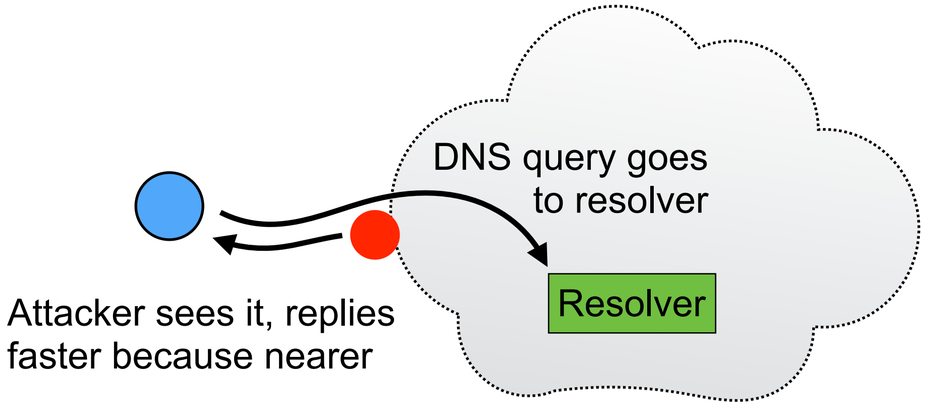
Methods for DNS Resolution

- DNS security
- DNS resolution over UDP, TLS, HTTPS, and QUIC

DNS Security (1/3)

- **DNS has historically been completely insecure**
  - Requests and responses delivered via unencrypted and unauthenticated protocol
  - Responses do not include a digital signature to verify authenticity of data
- Trivial to eavesdrop on who is looking up what name
- Trivial for on-path attackers, or malicious resolvers, to forge replies



e.g., possible if victim and attacker are in the same cafe, using insecure Wi-Fi

DNS Security (2/3)

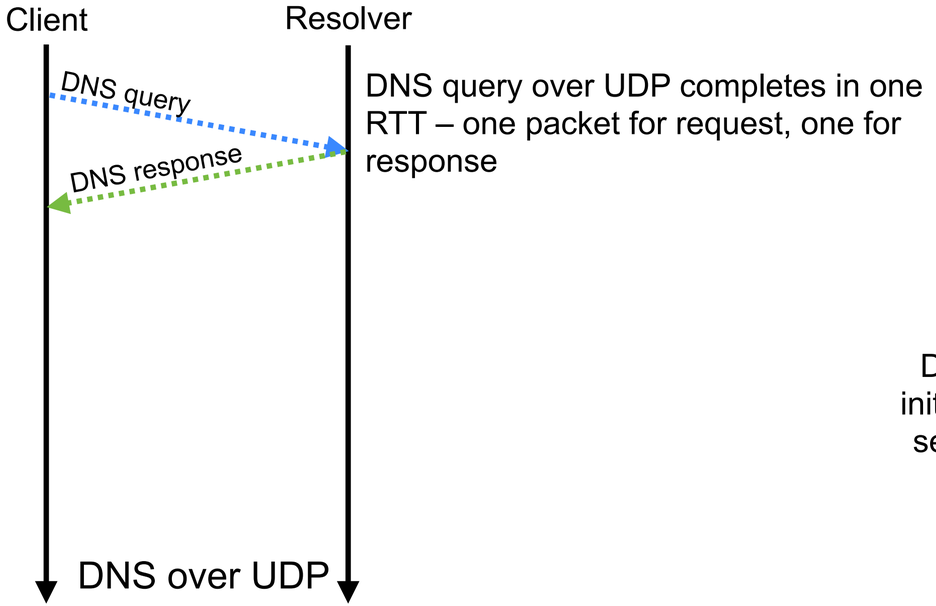
- Two approaches to securing DNS:
  - Transport security
    - Make DNS requests, and receive replies, over TLS (or some other secure channel)
    - Requests are responses are encrypted, so can’t be understood or modified by attacker
    - If you trust the resolver, this protects against attack

DNS Security (3/3)

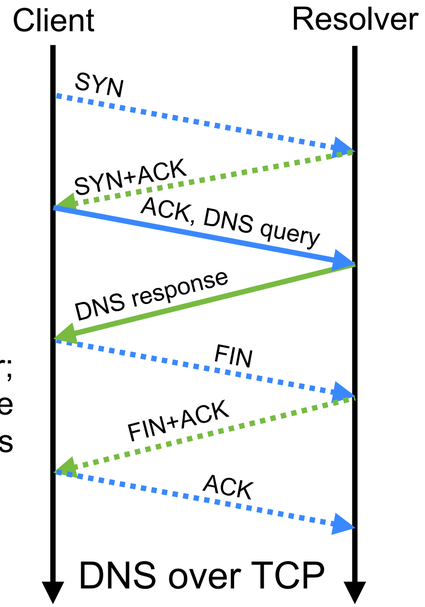
- Two approaches to securing DNS:
  - Transport security
  - Record security – DNSSEC
    - Add a digital signature to DNS responses that client can verify to check the data is valid
      - ICANN signs the root zone
      - Root servers sign information they provide about TLDs
      - TLDs sign information they provide about sub-domains
      - …
    - Allows a client to verify signatures back to the root, providing a chain of trust to demonstrate ownership of a domain – protects against malicious resolvers
    - Makes extensive use of public key cryptographic techniques – details are complex
    - Implemented, but not widely used
- **Need both transport and record security for fully secure DNS**

DNS Over UDP (1/4)

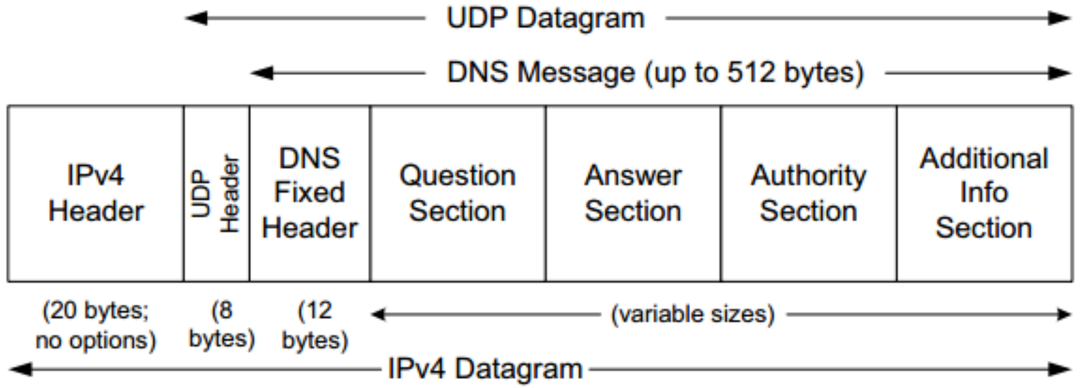
- DNS queries generally made over UDP port 53
  - Requests and responses are generally small enough to fit into a single packet
  - TCP reliability isn't needed – if no answer, retransmit the request
  - Congestion control isn't needed – can't adjust the rate you send a single packet



DNS over TCP exists, but is slower; initial and final TCP handshakes take several RTT and send extra packets

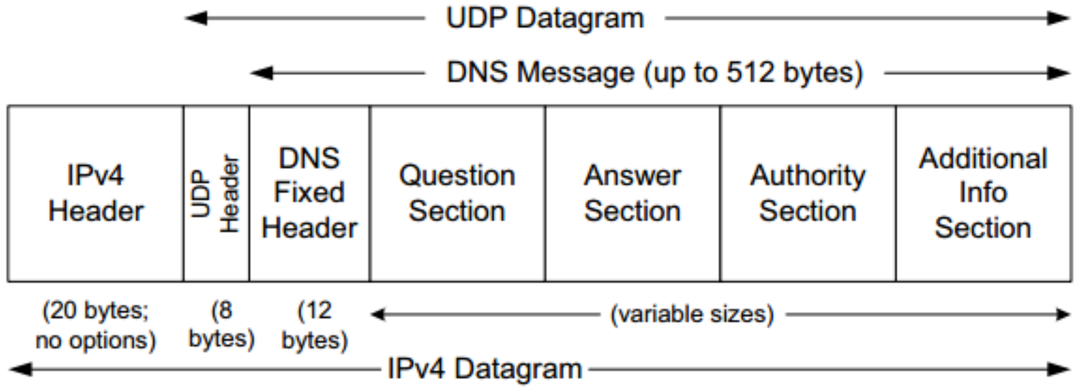


DNS Over UDP (2/4)



- Question section:
  - List of domain names and requested record type
    - e.g., what is the `AAAA` record for domain `csperkins.org`
  - Can include more than one question

DNS Over UDP (3/4)



- Answer, authority, and additional information sections:
  - List of domain names and record type, record data, and time-to-live
  - Answer section answers a question made in a previous request
    - e.g., the `AAAA` record for domain csperkins.org is 2a00:1098:0:86:1000::10 and it's valid for 1 hour
  - Authority describes where the answer came from

DNS Over UDP (4/4)

```
[stlinux02] > dig csperkins.org

; <<>> DiG 9.11.4-P2-RedHat-9.11.4-9.P2.el7 <<>> csperkins.org
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51409
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 5

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;csperkins.org.         IN A

;; ANSWER SECTION:
csperkins.org.    2681  IN A  93.93.131.127

;; AUTHORITY SECTION:
csperkins.org.    78278 IN NS ns2.mythic-beasts.com.
csperkins.org.    78278 IN NS ns1.mythic-beasts.com.

;; ADDITIONAL SECTION:
ns1.mythic-beasts.com.  70870  IN   A    45.33.127.156
ns2.mythic-beasts.com.  157301 IN   A    93.93.128.67
ns1.mythic-beasts.com.  70870  IN   AAAA 2600:3c00:e000:19::1
ns2.mythic-beasts.com.  157301 IN   AAAA 2a00:1098:0:80:1000::10

;; Query time: 0 msec
;; SERVER: 130.209.244.1#53(130.209.244.1)
;; WHEN: Wed Mar 04 18:26:53 GMT 2020
;; MSG SIZE  rcvd: 199
```

The `dig` tool on Linux or macOS performs DNS queries

DNS Over TLS (DoT)

- DNS over UDP is insecure
  - The packets are not encrypted or authenticated
  - Devices on the path between client and resolver can see DNS queries and responses – and can forge responses
- DNS over TLS solves this problem
  - DNS client opens a TCP connection to the resolver (port 853)
  - DNS client and resolver negotiate a TLS 1.3 session on the TCP connection
  - DNS client sends query, and receives response, over the TLS connection
    - DNS over TLS messages are formatted exactly the same as DNS over UDP, and contain exactly the same information – only difference is that they're sent over TLS not UDP
    - Slower and higher overhead than DNS over UDP – due to need to negotiate TCP and TLS –but more secure

DNS over HTTPS (DoH)

- DoH allows a client to send queries to a resolver using HTTPS
  - Can use with either GET or POST methods in HTTPS

```
GET /dns-query?dns=AAABAAABAAAAAAAAA3d3dwdleGFtcGxlA2NvbQAAAQAB HTTP/1.1
Accept: application/dns-message
```

Base-64 encoded version of the data that would be sent in a DNS-over-UDP request

Uses /dns-query as the URL path

```
POST /dns-query HTTP/1.1
Accept: application/dns-message
Content-type = application/dns-message
Content-length = 33

<33 bytes of UDP query, exactly as if sent in a UDP packet>
```

  - HTTP response has `Content-Type: application/dns-message` and contains the exact same data that would be sent in a UDP-based DNS response

DNS over QUIC (DoQ)

- Work in progress to define DNS over QUIC:
  - https://datatracker.ietf.org/doc/draft-huitema-quic-dnsoquic/
  - Same principle as DNS over TLS:
    - Client opens a QUIC connection to the resolver
      - Negotiates TLS security as part of the connection setup
    - Sends the request and receives the response over that connection
      - Requests and response contain exactly the same data as DNS over UDP – just sent via QUIC

Methods for DNS Resolution

- Increasingly many ways of making DNS queries:
  - DNS over UDP
  - DNS over TLS
  - DNS over HTTPS
  - DNS over QUIC
- The contents of the query and the response are **identical** in all cases
  - They change how the query is delivered to the resolver and how the response is returned, but not the contents of the messages
  - They change the security guarantees provided
  - They potentially gives clients more flexibility to query different resolvers

Methods for DNS Resolution

- DNS security
- DNS resolution over UDP, TLS, HTTPS, and QUIC