DNS and DNSSEC

Question 1 ***True/false***

Q1.1 Suppose we increase the entropy of the DNS ID field to 128 bits. It is infeasible for an on-path adversary to spoof a DNS answer.

○ True ● False

Solution: False. The adversary is on-path so they can see the QID in plaintext.

Q1.2 True or False: A DNS lookup for `en.wikipedia.org` will always force the recursive resolver to send at least 3 DNS queries.

○ True ● False

Solution: False. Answers could be cached.

Q1.3 Suppose we increase the entropy of the DNS ID field to 128 bits. It is infeasible for an on-path adversary to spoof a DNS answer.

○ True ● False

Solution: False. The adversary is on-path so they can see the QID in plaintext.

Q1.4 True or False: There is nothing a man-in-the-middle attacker (MITM) can do to interfere with a DNSSEC query.

○ True ● False

Solution: False. The MITM could do a DoS attack by dropping responses.

Q1.5 TRUE or FALSE: Destination port randomization could be implemented to increase the security of DNS without breaking the DNS protocol shown in lecture.

○ TRUE ● FALSE

Solution: False. The destination port needs to be well-known so requests can be sent.

Q1.6 TRUE or FALSE: In DNSSEC, if the root key is compromised, then no DNS records can be trusted.

● TRUE ○ FALSE

Solution: True. DNSSEC relies on the root server being trustworthy.

Question 2

Alice is using a DNS resolver to perform a DNS lookup for `www.google.com`. A single, valid nameserver is authoritative for each of the following zones:

Zone	*Nameserver*
`.`	`a.root-servers.net`
`.com`	`a.gtld-servers.net`
`google.com`	`ns1.google.com`

Assume no other legitimate clients will query the resolver (but the adversary can query it if they wish), the resolver's cache is initially empty, and the resolver uses iterative querying.

Assume that in DNSSEC, no one will accept a record unless it has a valid signature.

The attacker is on-path between the resolver and `ns1.google.com`, but off-path to the other name servers. The attacker also knows when Alice makes a request. **Assume DNS uses a static source port known to the attacker**.

For each part, select all of the records that the attacker can poison.

Q2.1 Standard DNS is used.

■ (A) Alice's cached A record for `www.google.com`

■ (B) Resolver's cached NS record for `.com`

■ (C) Resolver's cached NS record for `google.com`

☐ (D) Resolver's cached NS record for `.`

☐ (E) ——

☐ (F) ——

Solution: The adversary can spoof the DNS response from the resolver to the client even though they are off-path since we are using vanilla DNS and the source port isn't randomized (poisoning Alice's A record). Furthermore, the adversary can mount a Kaminsky attack against the resolver by querying the resolver directly; since the attacker can predict the source port, this will be successful. So, they can poison the cache for all of the NS records except the root since this is hardcoded. The adversary can also use this method to poison the A record for www.google.com cached on the resolver, and when Alice queries it, the resolver will respond to the client with the poisoned cache entry.

Q2.2 Standard DNS is used. Also, the resolver has a hardcoded NS record that maps the `google.com` zone to `ns1.google.com`, and a hardcoded A record with the IP address of `ns1.google.com`.

■ (G) Alice's cached A record for `www.google.com`

■ (H) Resolver's cached NS record for `.com`

☐ (I) Resolver's cached NS record for `google.com`

☐ (J) —

☐ (K) —

☐ (L) —

Solution: Similar to above, the adversary can poison everything in the resolver's cache, except the hardcoded records.

Q2.3 The resolver and nameservers use DNSSEC, and Alice uses standard DNS.

■ (A) Alice's cached A record for `www.google.com`

☐ (B) Resolver's cached NS record for `.com`

☐ (C) Resolver's cached NS record for `google.com`

☐ (D) —

☐ (E) —

☐ (F) —

Solution: Same reasoning as above for the first option. The adversary can't poison the cache for any of the NS records since that would require forging a signature.

Q2.4 The resolver and nameservers use DNSSEC, and Alice uses standard DNS. The adversary compromises `a.gtld-servers.net`.

■ (G) Alice's cached A record for `www.google.com`

■ (H) Resolver's cached NS record for `.com`

■ (I) Resolver's cached NS record for `google.com`

☐ (J) ——

☐ (K) ——

☐ (L) ——

Solution: Same reasoning as above for the first option. Controlling the `.com` domain allows the attacker to poison the resolver's cached NS record for `google.com` and `.com` since they have the private signing key and both of these domains are in-bailiwick. However, the fact that `.com` is in-bailiwick was considered outside the scope of the course and so this option wasn't graded.

Q2.5 The resolver and nameservers use DNSSEC, and Alice uses standard DNS. The adversary compromises `ns1.google.com`.

■ (A) Alice's cached A record for `www.google.com`

☐ (B) Resolver's cached NS record for `.com`

■ (C) Resolver's cached NS record for `google.com`

☐ (D) ——

☐ (E) ——

☐ (F) ——

Solution: Controlling the `google.com` nameserver allows the attacker to poison the final result which they were already able to do. Same as the previous question, the attacker can also poison the cache for `google.com` but this option wasn't graded. Bailiwick rules stop them from poisoning the cache for higher zones.

Q2.6 All parties use standard DNS, but the resolver and Alice encrypt their DNS messages with TLS.

■ (G) Alice's cached A record for `www.google.com`

■ (H) Resolver's cached NS record for `.com`

■ (I) Resolver's cached NS record for `google.com`

☐ (J) ——

☐ (K) ——

☐ (L) ——

Solution: The attacker can perform the Kaminsky attack to poison the `.com` + `google.com` NS records and on-path spoofing for the final DNS result.

Q2.7 All parties use standard DNS, but Alice, the resolver, and `ns1.google.com` encrypt their DNS messages with TLS.

■ (A) Alice's cached A record for `www.google.com`

■ (B) Resolver's cached NS record for `.com`

■ (C) Resolver's cached NS record for `google.com`

☐ (D) ——

☐ (E) ——

☐ (F) ——

Solution: The attacker can use the Kaminsky attack to poison the NS records for the `.com` zone since the root nameserver and `.com` nameservers don't use TLS, and then can subsequently poison the `google.com` NS record and the final result.

Q2.8 All parties use standard DNS, but everyone encrypts their DNS messages with TLS.

☐ (G) Alice's cached A record for `www.google.com`

☐ (H) Resolver's cached NS record for `.com`

☐ (I) Resolver's cached NS record for `google.com`

☐ (J) —

☐ (K) —

☐ (L) —

Solution: The attacker hasn't compromised any of the nameservers so they can't do anything here. TLS prevents an on-path adversary from tampering with the messages between the parties.

Q2.9 Alice and the resolver use standard DNS, but encrypt their DNS messages with TLS. The resolver and nameservers use DNSSEC.

☐ (A) Alice's cached A record for `www.google.com`

☐ (B) Alice's cached NS record for `google.com`

☐ (C) Resolver's cached NS record for `.com`

☐ (D) Resolver's cached NS record for `google.com`

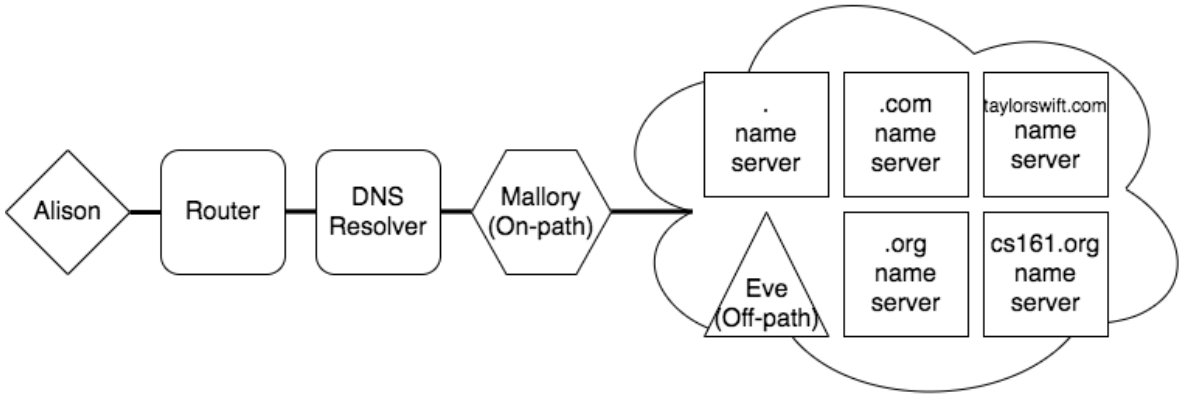
☐ (E) —

☐ (F) —

Solution: The attacker can't poison any caches here due to DNSSEC, and can't compromise the client's connection with the resolver due to TLS.

Question 3 ***I Knew UDP Was Trouble***

In the following diagram, Alison is connected to the network through her local router, which is connected to the local DNS resolver, which in turn uses iterative queries to resolve domains. Ports and the random UDP ID numbers are 16 bits, and DNS queries use 53 as both the source and destination ports. Mallory is an on-path attacker, while Eve is an off-path attacker. `cs161.org`, `.org`, `.com`, and the root domain support DNSSEC, but `taylorswift.com` does not. DNS caches always start empty. Each subpart is independent.



Q3.1 Which of the following entities, if malicious, could poison Alison's DNS resolver's cache for `taylorswift.com`?

- ■ (A) Mallory
- ■ (B) Name server for `.`
- ■ (C) Name server for `.com`
- ■ (D) Name server for `.org`
- ■ (E) Name server for `taylorswift.com`
- □ (F) None of the above

Solution: Every entity in the network can either directly modify a response or spoof a packet.

Q3.2 Which of the following entities, if malicious, could poison Alison's DNS resolver's cache for `cs161.org`?

☐ (G) Mallory

■ (H) Name server for `.`

☐ (I) Name server for `.com`

■ (J) Name server for `.org`

☐ (K) Name server for `taylorswift.com`

☐ (L) None of the above

Solution: DNSSEC prevents spoofing attacks and in-path attacks, but if a name server is malicious, it could change the response and still sign it. The resolver can directly change the response.

Q3.3 Which of the following actions would be effective in preventing Mallory from having a non-negligible probability of being able to poison the cache for `taylorswift.com`?

■ (A) Using TLS for all DNS queries

■ (B) Using DNSSEC for `taylorswift.com`

☐ (C) Using TCP instead of UDP for the DNS query

☐ (D) Source port randomization

☐ (E) None of the above

☐ (F) ——

Solution: TLS and DNSSEC authenticate the records. Name servers are not assumed to be malicious.

Q3.4 Which of the following actions would be effective in preventing Eve from having a non-negligible probability of being able to poison the cache for `taylorswift.com`?

■ (G) Using TLS for all DNS queries

■ (H) Using DNSSEC for `taylorswift.com`

■ (I) Using TCP instead of UDP for the DNS query

■ (J) Source port randomization

☐ (K) None of the above

☐ (L) ——

Solution: Same as the previous part, and also randomizing the source port is enough to prevent blind spoofing. TCP helps because Eve would have to guess the TCP sequence numbers to inject a forged response into the TCP connection.

Q3.5 Which of the following actions would be effective in preventing a malicious `.com` name server from having a non-negligible probability of being able to poison the cache for `taylorswift.com`?

☐ (A) Using TLS for all DNS queries

☐ (B) Using DNSSEC for `taylorswift.com`

☐ (C) Using TCP instead of UDP for the DNS query

☐ (D) Source port randomization

■ (E) None of the above

☐ (F) ——

Solution: If the name server itself is malicious, it would be able to poison the cache no matter what.