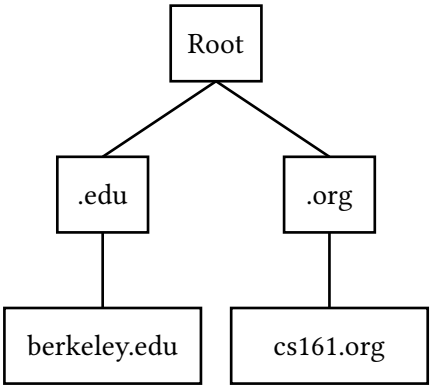
CS161 Fall 2026 — Introduction to Computer Security — Exam Prep 7

Q1 *Networking and Web: DNS* ❓ **(10 points)**

In this question, consider the DNS hierarchy to the right.

Unless otherwise stated, assume normal DNS is being used, without DNSSEC or DNS over HTTPS.

A student is performing a DNS lookup, and Mallory is an in-path attacker between the student and the recursive resolver.



Q1.1 (2 points) Which attacks could Mallory carry out against the student's DNS lookup for `cs161.berkeley.edu`? Select all that apply.

- ☐ Observe which domain name the student is looking up.
- ☐ Spoof a DNS reply that maps the domain to an attacker-controlled IP address.
- ☐ Decrypt the student's HTTPS traffic after the browser has verified the server's TLS certificate.
- ☐ Directly read cookies stored on the real `cs161.berkeley.edu` server.
- ◯ None of the above

Q1.2 (2 points) Now suppose DNS over HTTPS is in use. Which attacks could Mallory carry out against the student's DNS lookup for `cs161.berkeley.edu`? Select all that apply.

- ☐ Observe which domain name the student is looking up.
- ☐ Spoof a DNS reply that maps the domain to an attacker-controlled IP address.
- ☐ Decrypt the student's HTTPS traffic after the browser has verified the server's TLS certificate.
- ☐ Directly read cookies stored on the real `cs161.berkeley.edu` server.
- ◯ None of the above

This content is protected and may not be shared, uploaded, or distributed.

For the rest of the question, suppose Mallory has successfully executed an attack which gives her control of the `cs161.berkeley.edu` domain.

Q1.3 (2 points) Suppose a student has the following five cookies in their browser, with `Path=/`, `Secure=false` and `HttpOnly=true` on all five cookies.

After Mallory's attack, which cookies can Mallory read? Select all that apply.

- ☐ `Domain=sp26.cs161.org`
- ☐ `Domain=cs161.berkeley.edu`
- ☐ `Domain=cs161.org`
- ☐ `Domain=berkeley.edu`
- ☐ `Domain=www.mallory.com`
- ◯ None of the above

Q1.4 (2 points) After Mallory's attack, what strategies can Mallory use to read the cookies selected in the previous subpart? Consider each option independently.

- ☐ Mallory tricks the student into clicking some links she supplies.
- ☐ Mallory tricks the student into running some JavaScript she supplies.
- ☐ Mallory's attack allows her to directly read the cookies stored on the CS 161 server.
- ◯ None of the above

Q1.5 (1 point) Suppose `Secure=true` is set on the cookie with `Domain=cs161.berkeley.edu` above.

After Mallory's attack, can Mallory read this cookie?

- ◯ Yes, even without a certificate mapping `cs161.berkeley.edu` to her public key.
- ◯ Yes, but only with a certificate mapping `cs161.berkeley.edu` to her public key.
- ◯ No, even with a certificate mapping `cs161.berkeley.edu` to her public key.

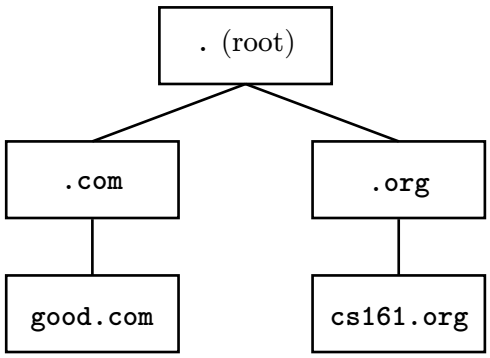
Q1.6 (1 point) After Mallory's attack, a student visits `http://cs161.berkeley.edu` in their browser and receives some JavaScript created by Mallory. Is this an XSS attack?

- ◯ Yes, because the JavaScript was sent by the student and reflected back to the student.
- ◯ Yes, because the JavaScript runs with the origin of `cs161.berkeley.edu`.
- ◯ No, because the JavaScript runs with the origin of `www.mallory.com`.
- ◯ No, because this is a CSRF attack, not an XSS attack.

This content is protected and may not be shared, uploaded, or distributed.

Q2 *Networking: GooDNS* 🤫 **(9 points)**

Consider this DNS hierarchy, where each box represents a name server:



EvanBot has the following records cached:

Record 1:	`org.`	`NS`	`a.org-servers.net`
Record 2:	`a.org-servers.net`	`A`	`192.7.14.21`
Record 3:	`com.`	`NS`	`a.com-servers.net`
Record 4:	`a.com-servers.net`	`A`	`192.6.16.161`
Record 5:	`evil.com`	`A`	`192.5.55.555`

In Q2.1 to Q2.3, each subpart continues on from previous subparts, i.e. records received in one subpart can be cached for later subparts.

Q2.1 (1 point) How many DNS requests does EvanBot need to make to learn the IP address of `www.cs161.org`?

◯ 0 ◯ 2 ◯ 3

◯ 1

Q2.2 (1 point) Record 1 is expired and removed from the cache.

How many DNS requests does the EvanBot need to make to learn the IP address of `www.cs161.org`?

◯ 0 ◯ 2 ◯ 3

◯ 1

Q2.3 (1 point) How many DNS requests does EvanBot need to make to learn the IP address of `not.good.com`?

◯ 0 ◯ 2 ◯ 3

◯ 1

This content is protected and may not be shared, uploaded, or distributed.

The rest of the question is independent of earlier subparts.

Q2.4 (1 point) Which of these best describes why an attacker would use the Kaminsky attack, instead of some other cache poisoning attack?

- ◯ The attacker is on-path; the Kaminsky attack only works for on-path attackers.
- ◯ Unlike other DNS attacks, the Kaminsky attack can poison a cache shared by many users.
- ◯ The attacker is off-path; the Kaminsky attack guarantees that the attacker will guess correctly.
- ◯ The attacker is off-path; the Kaminsky allows the attacker to make more guesses.

Q2.5 (2 points) Suppose the attacker can place HTML on a website that the victim will visit.

Which HTML snippets can help the attacker poison the cache for `www.google.com` (using the Kaminsky attack)? Select all that apply.

- ☐ `<img src="false1.google.com">`
- ☐ `<img src="false1.google.com/image.png">`
- ☐ `<img src="false1.cs161.org">`
- ☐ `<img src="www.google.com">`
- ◯ None of the above

Q2.6 (1 point) When source port randomization is enabled, what is the approximate probability that an off-path attacker successfully spoofs a DNS response?

- ◯ $1/2^{16}$
- ◯ $1/2^{64}$
- ◯ 1
- ◯ 0
- ◯ $1/2^{32}$

Q2.7 (1 point) When executing a Kaminsky attack, what should be the source IP in the attacker's spoofed DNS response?

- ◯ Attacker's IP address
- ◯ Resolver's IP address
- ◯ Name server's IP address
- ◯ The source IP field can be left blank.

Q2.8 (1 point) What is the primary reason DNSSEC does not provide confidentiality?

- ◯ The trust anchor model used by DNSSEC is incompatible with encryption protocols.
- ◯ Confidentiality would make the DNS query process too slow.
- ◯ DNS data is considered public information.
- ◯ Implementing encryption would require a new set of DNS record types, which is not feasible.

This content is protected and may not be shared, uploaded, or distributed.