Problem 1 The time-to-live (TTL) field of DNS responses is a 32-bit value that tells recursive resolvers how long (in seconds) they may cache and rely on the response. For records that change infrequently, such as CNAME redirections, the TTL field is usually set as high as 86400, or one day. In some cases, however, DNS servers set TTL much lower—as low as 30, or half a minute.

1. Suppose example.com is a popular site with thousands of Web server machines, each with its own IP address. Why might example.com’s name servers wish to answer each DNS A-record (IP address) lookup with a few different Web server IP addresses? Why might example.com’s name servers wish to set a very short TTL in these responses?

2. Now suppose that example.com’s has Web servers in many data centers around the world. Why might example.com’s name servers wish to answer DNS A-record lookups with Web server IPs geographically close to the client making the request? Why might example.com’s name servers wish to set a very short TTL in these responses?

3. Suppose an attacker wishes to mount a distributed denial of service attack against example.com. What part of example.com’s infrastructure should the attacker target? Why? Be specific.

---

\[1\] Databases mapping IP addresses to approximate geographic location are commercially available.
Problem 2 To simplify certificate deployment, TLS supports wildcard certificates. For example, many of UCSD's TLS servers use a single wildcard cert for "*.ucsd.edu":

Suppose that https://blink.ucsd.edu/, a server used by the University for many sensitive HR applications (IP: 132.239.180.101), and https://cseweb.ucsd.edu/, a server that hosts home pages for CS department members (IP: 132.239.8.67), both present UCSD's wildcard cert for TLS connections. Suppose further that the Apache server installed on https://cseweb.ucsd.edu/ is configured to support only that virtual host, and will serve content from the cseweb.ucsd.edu site regardless of what HTTP Host: header the client sends.

Alice Attacker is a graduate student in the CSE department, and has a home page at https://cseweb.ucsd.edu/~attacker/, where she can place arbitrary content. In addition, Alice Attacker has set up a rogue UCSD-GUEST access point on campus, allowing her to act as an in-path network attacker against any user who connects to that access point.

**Explain how Alice can completely undermine origin isolation for logged-in users of https://blink.ucsd.edu/ who connect to her access point**, by injecting JavaScript of her choice into the https://blink.ucsd.edu origin in their browsers. Be specific.

---

This is not the actual configuration of these servers. Do not attempt to attack blink.ucsd.edu, or any other server, without permission from its owners.
**Problem 3** The public-key pinning HTTP security header specifies the keys that can appear in that server’s certificate chain in the future. A site can use the header to list trusted CA keys that are allowed to issue certificates for its domain, or even trusted server public keys independent of CA. The header can be used to protect a site from malicious or compromised CAs. For example, the header

```
1 Public-Key-Pins: max-age=5184000;
2  pin-sha256="MOAuXUqLFrSHaKN50thH3DSenDECM5rRGQK5ggcVx5g=";
3  pin-sha256="315eGe9bkPs4g9Qi7a+0jGZgv420JL2XXTV7Q10ueII="
```

instructs the client for the next 60 days to accept only certificate chains for the site where the SHA-256 hash of some public key in the chain, base64-encoded, is MOAu... or 315e... The header must always specify at least two keys, at least one of them not be in the current certificate chain (to provide a recovery mechanism should the main key be lost). For security, key pins are maintained even if a user clears his cookies or switches to private browsing mode.

Suppose that an attacker compromises the server hosting https://victim.com/, which has a valid TLS certificate but doesn’t currently use key pinning.

The attacker then: (1) uses her own machines to generate two new keypairs, key\textsubscript{A} and key\textsubscript{B}; (2) obtains a certificate from Let’s Encrypt for victim.com with key\textsubscript{A}; (3) loads key\textsubscript{A} and the corresponding new certificate on the compromised victim.com server; (4) configures the victim.com server to send a public-key pinning header listing only key\textsubscript{A} and key\textsubscript{B}, with a one-year lifetime; and (6) after a few days, destroys all copies of the key\textsubscript{A} keypair, leaving the site inoperable.

At this point, all copies of key\textsubscript{A} are gone, and the attacker remains in possession of the key\textsubscript{B} keypair. Now she contacts the victim.com site’s owners.

**What is the attacker’s goal?** What will her message to the site’s owners say? What, if anything, can the site’s owners do now to recover?

---

\[ Because the attacker can place arbitrary content on the victim.com site, she can pass Let’s Encrypt’s automated checks. \]