Network Security II

Question 1 ***DNS Walkthrough*** ()

Your computer sends a DNS request for "www.google.com"

Q1.1 Assume the DNS resolver receives back the following reply:

```
com. NS a.gtld-servers.net
a.gtld-servers.net A 192.5.6.30
```

Describe what this reply means and where the DNS resolver would look next.

Q1.2 If an off-path adversary wants to poison the DNS cache, what values does the adversary need to guess?

Q1.3 Why not use cryptography to make the DNS connection secure?

Question 2 ***DNS*** **(14 min)**

Q2.1 Alice wants to access Berkeley's diversity advancement project DARE, `dare.berkeley.edu`. Her laptop connects to a wireless access point (AP).

Alice worries that a hacker attacks the DNS protocol when her laptop is looking for the IP address of `dare.berkeley.edu`. Assume that DNSSEC is not in use.

◇ **Question:** Which of the following can attack the DNS protocol and have Alice's browser obtain an incorrect IP address for DARE? (Select 0 to 8 options.)

- ☐ The laptop's operating system.
- ☐ The laptop's network interface controller.
- ☐ The wireless access point.
- ☐ An on-path attacker on the local network.
- ☐ The local DNS resolver of the network.
- ☐ The root DNS servers.
- ☐ `berkeley.edu`'s DNS nameservers.
- ☐ An on-path attacker between the local DNS resolver and the rest of the Internet.

Q2.2 Now assume that `berkeley.edu` implements DNSSEC and Alice's recursive resolver (but not her client) validates DNSSEC.

◇ **Question:** Which of the following can attack the DNS protocol and have Alice's browser obtain an incorrect IP address for DARE? (Select 0 to 8 options.)

- ☐ The laptop's operating system.
- ☐ The laptop's network interface controller.
- ☐ The wireless access point.
- ☐ An on-path attacker on the local network.
- ☐ The local DNS resolver of the network.
- ☐ The root DNS servers.
- ☐ `berkeley.edu`'s DNS nameservers.
- ☐ An on-path attacker between the local DNS resolver and the rest of the Internet.

Q2.3 An attacker wants to poison the local DNS resolver's cache using the Kaminsky attack. We assume that the resolver does not use source port randomization, so the attacker will likely succeed.

In the Kaminsky attack, the attacker asks the resolver for a *non-existing* subdomain of UC Berkeley, *e.g.*, `stanford.berkeley.edu`, instead of asking for an *existing* domain like `dare.berkeley.edu`.

◇ **Question:** What is the advantage of asking for a non-existent domain compared to asking for an existing domain? (answer within 10 words)

_____

_____

Question 3 ***NSEC*** **()**

In class, you learned about DNSSEC, which uses signature chains to ensure authentication for DNS results. Recall that in the case of a negative result (the name requested doesn't exist), the nameserver returns a signed pair of domains that are alphabetically before and after the requested name.

For example, suppose the following names exist in `google.com` when it's viewed in alphabetical order:

```
...
a-one-and-a-two-and-a-three-and-a-four.google.com
a1sauce.google.com
aardvark.google.com
...
```

In this ordering, `aaa.google.com` would fall between `a1sauce.google.com` and `aardvark.google.com`. So in response to a DNSSEC query for `aaa.google.com`, the name server would return an NSEC RR that in informal terms states "the name that in alphabetical order comes after `a1sauce.google.com` is `aardvark.google.com`", along with a signature of that NSEC RR made using `google.com`'s key.

Q3.1 DNS attacks we previously saw in class caused victims to unknowingly visit an attacker-controlled domain. Since receiving a negative result back from a nameserver causes a client to raise an error rather than visit a domain, why is a signature still necessary? What attack becomes possible without one?

Q3.2 A startup, `ThoughtlessSecurity`, decides to modify DNSSEC to only return a signature of the *requested domain* on a negative result. They claim that this change will drastically reduce the packet-size of a negative result.

A company implements `ThoughtlessSecurity`'s product on their nameserver. What attack is now possible? Specify exactly how an attacker could execute this attack.

Q3.3 Using the originally-described DNSSEC protocol, describe how an attacker can enumerate all domain names

Q3.4 A new startup, `ThoughtfulSecurity` wants to use a hash function to hinder this enumeration process and start by taking the hash of each existing domain. How can they use hashes to provide authenticated negative results?

Q3.5 How does this method help prevent enumeration attacks? Which properties does the hash function need to have?

Q3.6 Describe how an adversary with access to a dictionary might still be able to perform an enumeration attack. What conditions must hold true for the domain names?