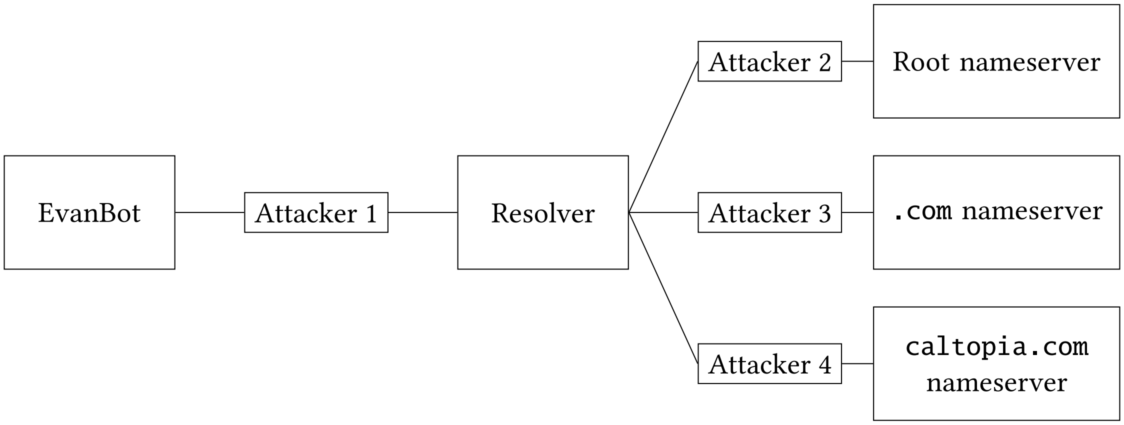
Weaver
Spring 2022

CS 161
Computer Security

Exam Prep 11

Q1 *Caltopia DNS* (21 points)

EvanBot is trying to determine the IP address of `caltopia.com` with DNS. However, some attackers on the network want to provide EvanBot with the wrong answer.



Assumptions:

- Each attacker is a man-in-the-middle (MITM) attacker between their two neighbors on the diagram above.
- No attackers can perform a Kaminsky attack.
- Standard DNS (not DNSSEC) is used unless otherwise stated.
- No private keys have been compromised unless otherwise stated.
- In each subpart, both EvanBot's cache and the local resolver's cache start empty.
- Each subpart is independent.

Clarification during exam: Assume that bailiwick checking is in use for this entire question.

In each subpart, EvanBot performs a DNS query for the address of `caltopia.com`.

Q1.1 (4 points) In this subpart only, assume the attackers only passively observe messages.

Which of the attackers would observe an `A` record with the IP address of `caltopia.com` as a result of EvanBot's query? Select all that apply.

■ (A) Attacker 1

□ (B) Attacker 2

□ (C) Attacker 3

■ (D) Attacker 4

□ (E) None of the above

□ (F) ——

Solution: The `A` type record is sent from the `caltopia.com` name server to the resolver, and then from the resolver to EvanBot.

Q1.2 (3 points) Which of the attackers can poison the local resolver's cached record for `cs161.org` by injecting a record into the additional section of the DNS response? Select all that apply.

Note: Attacker 1 has intentionally been left out as an answer choice.

■ (G) Attacker 2

□ (H) Attacker 3

□ (I) Attacker 4

□ (J) None of the above

□ (K) ——

□ (L) ——

Solution: `cs161.org` is in bailiwick for root, so Attacker 2 could add a record for `cs161.org` in the response from root.

However, `cs161.org` is not in bailiwick for `.com` or `caltopia.com`, so attackers 3 and 4 cannot add a record for `cs161.org` in the responses from `.com` or `caltopia.com`.

Q1.3 (4 points) Assume that the resolver and the name servers all validate DNSSEC, but EvanBot does not validate DNSSEC. Which of the attackers can poison EvanBot's cached record for `caltopia.com` by modifying the DNS response? Select all that apply.

■ (A) Attacker 1

□ (B) Attacker 2

□ (C) Attacker 3

□ (D) Attacker 4

□ (E) None of the above

□ (F) ——

Solution: Since the resolver and the name servers all validate DNSSEC, any attacker between the resolver and a name server can't do anything to inject malicious records. However, since EvanBot doesn't validate DNSSEC, Attacker 1 can inject a malicious `A` record.

Q1.4 (5 points) In this subpart only, assume the attackers only passively observe messages.

Assume that everyone validates DNSSEC. Which of the following records would Attacker 3 observe as a result of EvanBot's query? Select all that apply.

□ (G) `DS` record with hash of the `.com` name server's public KSK

■ (H) `DS` record with hash of the `caltopia.com` name server's public KSK

□ (I) `A` record with the IP address of `caltopia.com`

■ (J) `A` record with the IP address of the `caltopia.com` name server

■ (K) `DNSKEY` record with the `.com` name server's public KSK

□ (L) None of the above

Solution: The `.com` name server returns:

- A `DNSKEY` record with its public keys (option K)
- An `NS` record with the domain of the next name server (`caltopia.com`)
- An `A` record with the IP of the next name server (`caltopia.com`) (option J)
- A `DS` record with hash of the next name server's public KSK (option H)

Option (G) would be returned by `.com`'s parent (the root), so Attacker 2 would see this record, not Attacker 3.

Option (I) would be returned by the `caltopia.com` name server, so Attacker 4 would see this, not Attacker 3.

Q1.5 (3 points) Assume that everyone validates DNSSEC, and the `caltopia.com` name server's private KSK has been compromised (i.e. all attackers know the `caltopia.com` name server's private KSK). No other private keys have been compromised.

Can EvanBot trust that they received the correct IP address of `caltopia.com`?

○ (A) Yes, because the ZSK that signs the `A` record has not been compromised

○ (B) Yes, because the trust anchor (the root's KSK) has not been compromised

○ (C) No, because the compromised KSK can be used to sign a malicious `A` record

● (D) No, because the compromised KSK can be used to sign a fake ZSK that is used to sign a malicious `A` record

○ (E) ——

○ (F) ——

Solution: The chain of trust has been broken, so EvanBot can't trust that they received the correct IP address anymore.

The KSK is only used to sign ZSKs, so the attacker will have to sign a fake ZSK first, and then use the fake ZSK to sign the malicious `A` record.

Q1.6 (2 points) True or False: DNSSEC prevents Attacker 4 from learning the IP address of `caltopia.com`.

○ (G) True ● (H) False ○ (I) —— ○ (J) —— ○ (K) —— ○ (L) ——

Solution: DNSSEC provides no confidentiality over the DNSSEC records.

Q2 *Intrusion Detection Scenarios* **(12 points)**

For each scenario below, select the best detector or detection method for the attack.

Q2.1 (3 points) The attacker constructs a path traversal attack with URL escaping: `%2e%2e%2f%2e%2e%2f`.

○ (A) NIDS, because of interpretation issues

○ (B) NIDS, because of cost

● (C) HIDS, because of interpretation issues

○ (D) HIDS, because of cost

○ (E) —

○ (F) —

Solution: This path traversal attack is masked using percent encoding in URLs. A traditional NIDS might not recognize this since it is specific to HTTP servers, so a HIDS would be the best option here in order ot avoid the interpretation issues of percent encoding.

Q2.2 (3 points) The attacker is attacking a large network with hundreds of computers, and a detector must be installed as quickly as possible.

○ (G) NIDS, because of interpretation issues

● (H) NIDS, because of cost

○ (I) HIDS, because of interpretation issues

○ (J) HIDS, because of cost

○ (K) —

○ (L) —

Solution: A major advantage of NIDS is that they can be quickly installed in order to cover an entire network. Because of the time constraints, the NIDS would be the best in order to mitigate the time cost.

Q2.3 (3 points) The attacker constructs an attack that is encrypted with HTTPS.

○ (A) NIDS, because of interpretation issues

○ (B) NIDS, because of cost

● (C) HIDS, because of interpretation issues

○ (D) HIDS, because of cost

○ (E) —

○ (F) —

Solution: A NIDS is not able to decrypt data since it doesn't have the keys that are stored on the host. Thus, only the host can decrypt an interpret the requests, and a HIDS would be the best IDS to use here.

Q2.4 (3 points) The attacker constructs a buffer overflow attack using shellcode they found online in a database of common attacks.

● (G) Signature-based

○ (H) Specification-based

○ (I) Anomaly-based

○ (J) Behavioral

○ (K) ——

○ (L) ——

Solution: This shellcode is easily obtainable and has not been modified, so a signature that matches the exact shellcode would be most effective in detecting this attack.

Q3 *Election Security* (23 points)

The 2020 elections are coming up, and the United States Government has tasked you with securing the nation's voting machines!

Assume election headquarters are in a top-secret, undisclosed site. All incoming network requests pass through a network-based intrusion detection system (NIDS), as well as a firewall. Outside users can only access the server with HTTPS.

Q3.1 (3 points) Which of these attacks are **always** preventable in this setup? Assume the attacker is on-path. Select all that apply.

□ (A) RST Injection Attack

□ (B) SQL Injection Attack

□ (C) Reflected XSS Attack

■ (D) None of the Above

□ (E) ——

□ (F) ——

Q3.2 (3 points) Which of these attacks are **always** preventable in this setup? Assume the attacker is on-path. Select all that apply.

■ (G) SYN Flooding Attack

□ (H) DNS Spoofing Attack

□ (I) DDoS Attack

□ (J) None of the Above

□ (K) ——

□ (L) ——

Solution:

- RST Injection Attack - HTTPS doesn't prevent RST Injection attacks, so they're still a potential vulnerability
- SQL Injection Attack - these attacks are generally application-layer (so transport-layer security and firewalls don't protect against them)
- Reflected XSS Attack - same reasoning as above. Additionally, even if NIDS were capable of detecting these over HTTP, it wouldn't be able to see any payloads under HTTPS.
- SYN Flooding Attack - these attacks are preventable using SYN Cookies!
- DNS Spoofing Attack - none of the defenses prevent DNS Spoofing
- DDoS Attack - not much a NIDS can do here, unfortunately

Q3.3 (3 points) An attacker injects malicious code on a server inside the election headquarters that changes all submitted votes to one candidate. Which detection system is best suited to defend against this attacker?

● (A) HIDS ○ (C) Firewall ○ (E) —

○ (B) NIDS ○ (D) — ○ (F) —

Solution: Only a host-based system would be able to detect and/or prevent this attack from happening!

Q3.4 (3 points) An attacker realizes that the ballot boxes are running a vulnerable version of Linux, and uses a previously-known buffer overflow exploit. Which detection method is best suited to defend against this attacker?

○ (G) Anomaly-Based Detection ○ (J) Behavioral-Based Detection

● (H) Signature-Based Detection ○ (K) —

○ (I) Specification-Based Detection ○ (L) —

Solution: Signature-based detection approaches are primarily responsible for catching known attacks!

Q3.5 (5 points) Ben, a computer scientist at the top-secret site, has a HIDS installed on his work laptop. He decides to sign into his personal email account, claiming that HTTPS will protect the government from seeing his emails. Is he correct? Justify your answer in 1–2 sentences.

○ (A) Yes ○ (D) —

● (B) No ○ (E) —

○ (C) — ○ (F) —

Solution: Host-based intrusion detection systems are capable of reading data inbound/outbound HTTPS connections, so Ben's use of HTTPS doesn't really help him here.

We also accepted yes as an answer if it was justified by claiming he could use an email client that the HIDS didn't have access to

Q3.6 (3 points) You're discovered that an attacker has managed to connect to a service running inside our network from IP Address and is in the process of performing a DoS attack! Write a stateful firewall rule to block all traffic originating from the attacker. Our service is running on IP address (port 443).

Solution: drop * :*/ext -> :443/int