CS 161 Spring 2021 Computer Security

Final Exam

INSTRUCTIONS

This is your exam. Complete it either at exam.cs61a.org or, if that doesn't work, by emailing course staff with your solutions before the exam deadline.

This exam is intended for the student with email address `tianqiy@berkeley.edu`. If this is not your email address, notify course staff immediately, as each exam is different. Do not distribute this exam PDF even after the exam ends, as some students may be taking the exam in a different time zone.

For questions with **circular bubbles**, you should select exactly *one* choice.

○ You must choose either this option

○ Or this one, but not both!

For questions with **square checkboxes**, you may select *multiple* choices.

□ You could select this choice.

□ You could select this one too!

You may start your exam now. Your exam is due at 02:40PM Pacific Time. Go to the next page to begin.

Preliminaries

You can complete and submit these questions before the exam starts.

(a) **(5.0 pt)** Read the following honor code and sign your name. *Failure to do so will result in a grade of 0 for this exam.*

> I understand that I may not collaborate with anyone else on this exam, or cheat in any way. I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that academic misconduct will be reported to the Center for Student Conduct and may further result in partial or complete loss of credit.

(b) What is your student ID number?

1. **True/false**

Each true false is worth 2 points.

(a) **(2.0 pt)** True or false: If a pseudorandom number generator (pRNG) is secure, then an attacker who only sees the output of the pRNG is unable to learn its internal state.

○ True

○ False

(b) **(2.0 pt)** True or false: A primary advantage of a host-based intrusion detection system (HIDS) over a network-based intrusion detection system (NIDS) is that traffic can be analyzed in plaintext, since the host can access decrypted TLS traffic.

○ True

○ False

(c) **(2.0 pt)** True or false: Sending all DNS requests and responses over TLS/HTTPS (DNS over HTTPS) can be used as an effective defense against censorship by preventing censors from knowing what websites you are visiting.

○ True

○ False

(d) **(2.0 pt)** True or false: Alice decides to use Tor to protect herself from tracking and surveillance online. The Tor circuit contains three Tor nodes: an entry node, a relay node, and an exit node. Assume the nodes do not collude. The exit node knows Alice's IP address but not the domain of the website she is visiting.

○ True

○ False

(e) **(2.0 pt)** True or false: Specification-based detection uses a blacklist.

○ True

○ False

(f) **(2.0 pt)** Consider two different detectors with the same false positive rate and false negative rate. Assume that false negatives and false positives are equally costly.

True or false: A website with a high volume of users but a low volume of attacks would benefit more from placing the detectors in series rather than in parallel.

○ True

○ False

(g) **(2.0 pt)** True or false: For organizations with a large number of network devices, network-based intrusion detection systems (NIDS) are easier to deploy and manage than host-based intrusion detection systems (HIDS).

○ True

○ False

(h) (2.0 pt) True or false: WPA2 is a protocol that translates IP addresses to MAC addresses.

○ True

○ False

(i) (2.0 pt) True or false: The UDP protocol guarantees that packets are delivered to the destination server by detecting dropped packets and retransmitting them until they are acknowledged.

○ True

○ False

(j) (2.0 pt) True or false: Publicly accessible stairs, walkways, and elevators can be considered part of the physical equivalent of a trusted computing base for airport security.

○ True

○ False

(k) (2.0 pt) True or false: Clickjacking refers to a class of attacks where the attacker manipulates the user interface of a website to convince the user to click something that they did not intend to click on.

○ True

○ False

(l) (2.0 pt) True or false: Cryptographically secure MACs can be constructed using secure cryptographic hash functions.

○ True

○ False

(m) (2.0 pt) True or false: Argon2 and PBKDF2 are appropriate algorithms to use when hashing and storing passwords in a database.

○ True

○ False

(n) (2.0 pt) True or false: All forms of two-factor authentication (2FA) are resistant to phishing attacks.

○ True

○ False

(o) (2.0 pt) True or false: One-time pads, as long as they are used correctly, are secure against an adversary with infinite computational power.

○ True

○ False

(p) (2.0 pt) True or false: Logging is a method of intrusion detection in which server log files are preserved so they can be asynchronously scanned to detect malicious activity.

○ True

○ False

(q) **(2.0 pt)** True or false: Signature-based intrusion detection systems are good at identifying novel network attacks that have not been previously seen.

○ True

○ False

(r) **(2.0 pt)** True or false: TLS is able to prevent on-path attackers from learning metadata about your communications (e.g. request and response times, message length) by encrypting communications from a client to a server.

○ True

○ False

(s) **(2.0 pt)** True or false: When analyzing a cryptographic hashing scheme, preimage resistance (one-way) implies collision resistance.

○ True

○ False

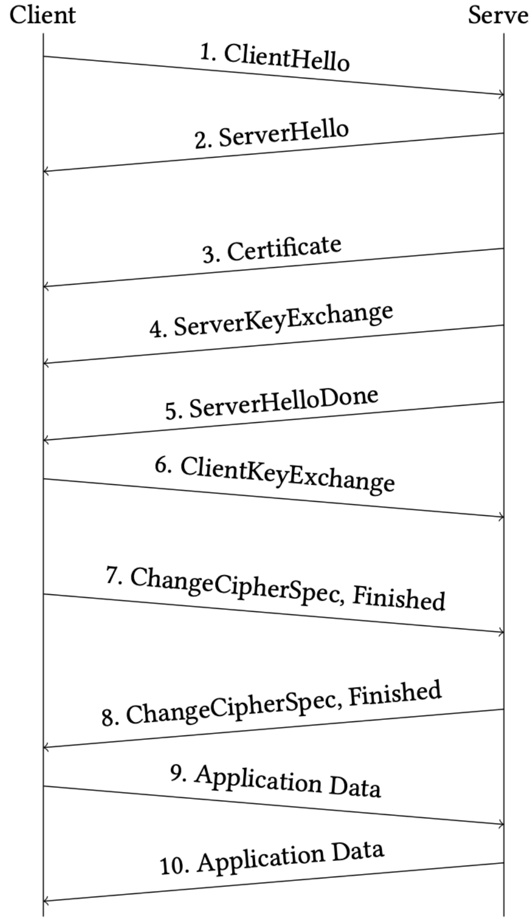
(t) **(0.0 pt)** True or false: `EvanBot is a real bot.`

○ True

○ False

2. Mutuality

Recall the TLS handshake:



1. Client sends 256-bit random number R_b and supported ciphers

2. Server sends 256-bit random number R_s and chosen cipher

3. Server sends certificate

4. DH: Server sends $\{g, p, g^a \bmod p\}_{K_{\text{server}}^{-1}}$

5. Server signals end of handshake

6. DH: Client sends $g^b \bmod p$
   RSA: Client sends $\{PS\}_{K_{\text{server}}}$
Client and server derive cipher keys C_b, C_s and integrity keys I_b, I_s from R_b, R_s, PS

7. Client sends MAC(dialog, I_b)

8. Server sends MAC(dialog, I_s)

9. Client data takes the form $\{M_1, \text{MAC}(M_1, I_b)\}_{C_b}$

10. Server data takes the form $\{M_2, \text{MAC}(M_2, I_s)\}_{C_s}$

Diagram of the TLS handshake

In TLS, we verify the identity of the server, but not the client. How would we modify TLS to also verify the identity of the client?

(a) (3.0 pt) Which of these additional values should the client send to the server?

○ A certificate with the client's private key, signed by a certificate authority's private key

○ A certificate with the client's public key, signed by a certificate authority's private key

○ A certificate with the client's public key, signed by the server's private key

○ A certificate with the client's public key, signed by the client's private key

(b) (3.0 pt) How should the client send the premaster secret in RSA TLS?

○ Encrypted with the server's public key, signed by a certificate authority's private key

○ Encrypted with the client's public key, signed by a certificate authority's private key

○ Encrypted with the server's public key, signed by the client's private key

○ Encrypted with the client's public key, signed by the server's private key

(c) (3.0 pt) EvanBot argues that the key exchange protocol in Diffie-Hellman TLS doesn't need to be changed to support client validation. Is EvanBot right?

○ Yes, because the server has already received and verified the client's certificate

○ No, the client must additionally sign their part of the Diffie-Hellman exchange with the certificate authority's private key

○ No, the client must additionally sign their part of the Diffie-Hellman exchange with the client's private key

○ Yes, because only the client knows the secret a, so the server can be sure it's talking to the legitimate client

(d) (2.0 pt) True or false: The server can be sure that they're talking to the client (and not an attacker impersonating the client) immediately after the client and server exchange certificates.

○ False

○ True

(e) (3.0 pt) At what step in the TLS handshake can both the client and server be sure that they have derived the same symmetric keys?

○ Immediately after the TCP handshake, before the TLS handshake starts

○ Immediately after the ClientHello and ServerHello are sent

○ Immediately after the client and server exchange certificates

○ Immediately after the client and server verify signatures

○ Immediately after the MACs are exchanged and verified

(f) (4.0 pt) Which of these keys, if stolen individually, would allow the attacker to impersonate the client? Select all that apply.

□ Private key of the server

□ Private key of a certificate authority

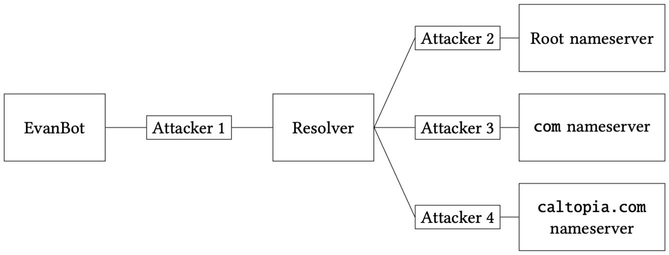
□ Private key of the client

□ Public key of a certificate authority

□ None of the above

3. Caltopia DNS

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.



DNS network layout. Attacker 1 is between the client and resolver. Attacker 2 is between the resolver and root nameserver. Attacker 3 is between the resolver and .com namserver. Attacker 4 is between the resolver and caltopia.com nameserver.

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.

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

(a) **(4.0 pt)** 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.

□ Attacker 1

□ Attacker 2

□ Attacker 3

□ Attacker 4

□ None of the above

(b) **(3.0 pt)** 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.

□ Attacker 2

□ Attacker 3

□ Attacker 4

□ None of the above

(c) (4.0 pt) 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.

□ Attacker 1

□ Attacker 2

□ Attacker 3

□ Attacker 4

□ None of the above

(d) (5.0 pt) 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.

□ `A` record with the IP address of the `caltopia.com` name server

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

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

□ `DNSKEY` record with the `.com` name server's public KSK

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

□ None of the above

(e) (3.0 pt) 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`?

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

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

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

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

(f) (2.0 pt) True or false: DNSSEC prevents Attacker 4 from learning the IP address of `caltopia.com`.

○ True

○ False

4. UnicornBox v2

UnicornBox decides to implement 2-factor authentication (2FA).

The server stores a table of active codes with the following schema:

```
CREATE TABLE IF NOT EXISTS codes (
    username TEXT,
    code TEXT,
    -- Additional fields not shown.
);
```

When a user wants to log in:

(a) The user logs in by making a POST request with their username and password.

(b) The server randomly generates a 10-digit numerical code and stores it in the codes table.

(c) The server sets a cookie with name = `login_user_cookie` and value = the user's username in the user's browser. The server also sends a text to the user's phone with the code.

(d) The user makes a GET request to `https://unicornbox.com/confirm?code=$code`, where `$code` is the code that was entered.

(e) The server runs the SQL query `SELECT username FROM codes WHERE    code = '$code'`, where `$code` is the value submitted by the user.

(f) The server checks that the value returned by the SQL query matches the username sent in the `login_user_cookie` cookie in the request submitted by the user.

(a) (5.0 pt) Assume that `evan` is the name of an account in CalCentral with an entry in the `codes` table.

Construct an input for `$code` that would cause the SQL query in step 5 to return `evan`.

(b) (4.0 pt) How can you log in as `evan` without knowing their password? You may use `PAYLOAD` to reference your answer from the previous part.

Hint: You will need 2 steps. List both.

(c) (4.0 pt) Which of these defenses would stop your exploit from above? Select all that apply.

- □ Putting the hash of the username in the cookie instead of the username
- □ Rate limiting requests to the UnicornBox server
- □ Using a 20-digit code instead of a 10-digit code
- □ Using SQL prepared statements
- □ None of the above

(d) (2.0 pt) Consider a modification to Steps 5 and 6. If there are any rows returned by the SQL query, then the verification succeeds without checking the value of the returned username. However, the server returns an error without executing the query if the format of the code is not exactly 10 numerical digits.

True or false: The modified scheme is no longer exploitable using SQL injection.

- ○ True
- ○ False

(e) (2.0 pt) Briefly (1 sentence) justify your answer from the previous part.

5. Plaintext Feedback

Consider the "plaintext feedback" (PFB) mode where the encryption formula for ciphertext block C_i is given as follows:

$$C_0 = IV$$
$$C_i = E(K, P_i) \oplus C_{i-1}$$

E is AES encryption and D is AES decryption.

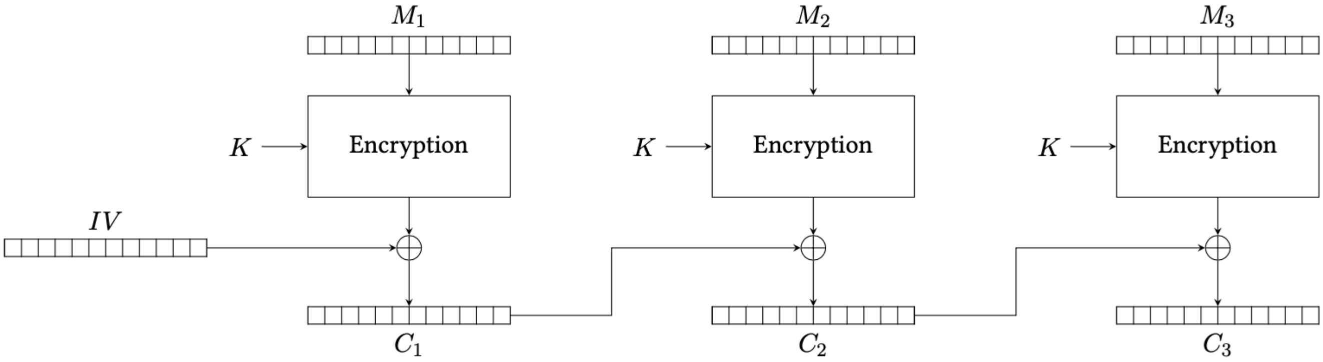


Diagram of the PFB mode of operation

(a) **(3.0 pt)** Which of these is the corresponding decryption equation?

○ $P_i = E(K, C_i \oplus P_{i-1})$

○ $P_i = E(K, P_i \oplus P_{i-1})$

○ $P_i = D(K, C_i \oplus P_{i-1})$

○ $P_i = E(K, C_i \oplus C_{i-1})$

○ $P_i = D(K, C_i \oplus C_{i-1})$

○ $P_i = D(K, P_i \oplus P_{i-1})$

(b) **(3.0 pt)** Alice and Bob are communicating using PFB mode. Alice encrypts and sends a 10-block message encrypted using PFB. Bob receives the message, but the 6th ciphertext block C_6 is lost in transmission. Which blocks of plaintext can Bob recover? Assume Bob is aware that C_6 was lost in transmission.

○ Bob can recover all blocks except for P_6 and P_7.

○ Bob can recover all blocks up to and including P_5, but no block after that.

○ Bob can recover all blocks of the message.

○ Bob can recover all blocks up to and including P_6, but no block after that.

○ Bob cannot recover any block of the message.

○ Bob can recover all blocks except for P_6.

(c) (3.0 pt) PFB mode is not IND-CPA secure. To prove this, the adversary will win the IND-CPA game against the challenger as follows:

First, the adversary sends two messages, P and P'. The first message P is 3 unique, randomly generated blocks, $P = P_1 \| P_2 \| P_3$. Which of the following values of P' would allow the adversary to win the IND-CPA game?

○ $P' = P'_1 \| P'_1 \| P'_1$, where P'_1 is a randomly generated block

○ $P' = P'_1 \| P'_2 \| P'_3$, where P'_i is the same as P_i, but with the last bit flipped

○ $P' = P'_1 \| P'_2 \| P'_3$, where P'_i is the same as P_i, but every bit flipped

○ $P' = P_1 \| P_2 \| P_3$

○ $P' = P'_1 \| P'_2 \| P'_3$, where P'_1, P'_2, and P'_3 are unique, randomly generated blocks

(d) (3.0 pt) The challenger sends back a ciphertext $C = C_0 \| C_1 \| C_2 \| C_3$, which is an encryption of either P or P'. Describe a strategy that the adversary should use to deduce whether P or P' was encrypted that would allow them to win the IND-CPA game with probability greater than $\frac{1}{2}$.

(e) (3.0 pt) Which of the following are true about PFB mode? Select all that apply.

□ The plaintext must be padded to a multiple of the block length

□ PFB provides integrity

□ Decryption is parallelizable

□ None of the above

6. TC sPeedy

(a) (3.0 pt) To improve the speed of TCP, Alice suggests modifying the TCP protocol to allow data to be sent in the SYN and SYN-ACK packets during the 3-way handshake. The data in the SYN packet is immediately accepted by the server during the initial handshake (before the 3-way handshake finishes).

Which of the following attacks are possible on this modified scheme? Select all that apply.

□ An off-path attacker can fool the server into accepting some spoofed data.

□ An off-path attacker can reliably execute a RST injection attack.

□ An off-path attacker can reliably inject packets after a connection has been established.

□ None of the above

(b) (2.0 pt) Alice notices that her modified scheme may be vulnerable to a DoS attack where the attacker sends a large data payload in the SYN packet without completing the TCP handshake. She proposes including SYN cookies as part of her modification.

True or false: SYN cookies provide a valid defense against the proposed DoS attack.

○ True

○ False

(c) (4.0 pt) Alice uses her modified 3-way handshake to form a TCP connection with a server. Assume that source port randomization is not in use.

What fields would an **on-path** attacker have to guess in order to inject some data from Alice's client to the server?

□ Client IP address and port

□ Server sequence number

□ Client sequence number

□ Server IP address and port

□ None of the above

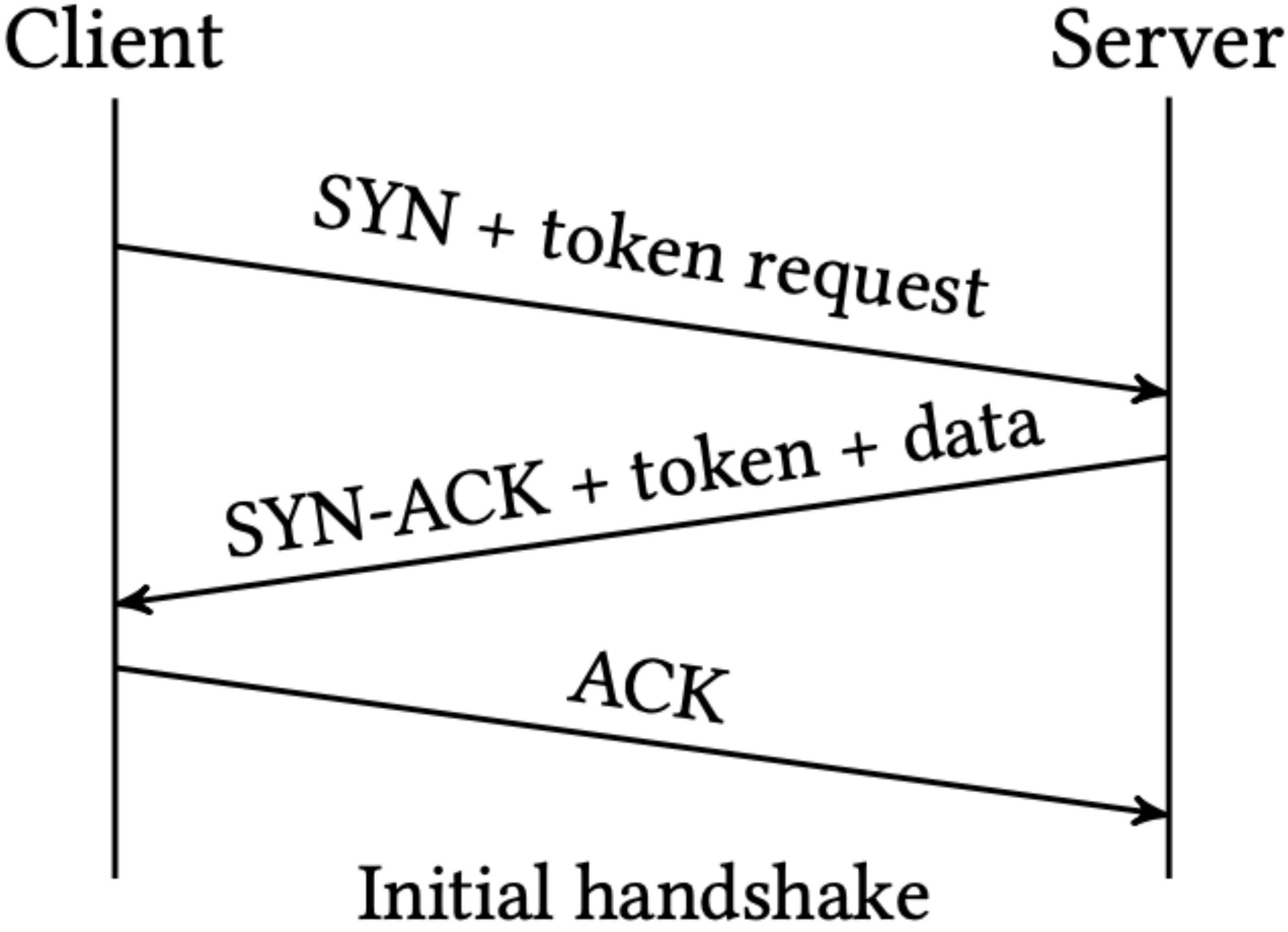


Diagram of the initial connection of the TFO protocol

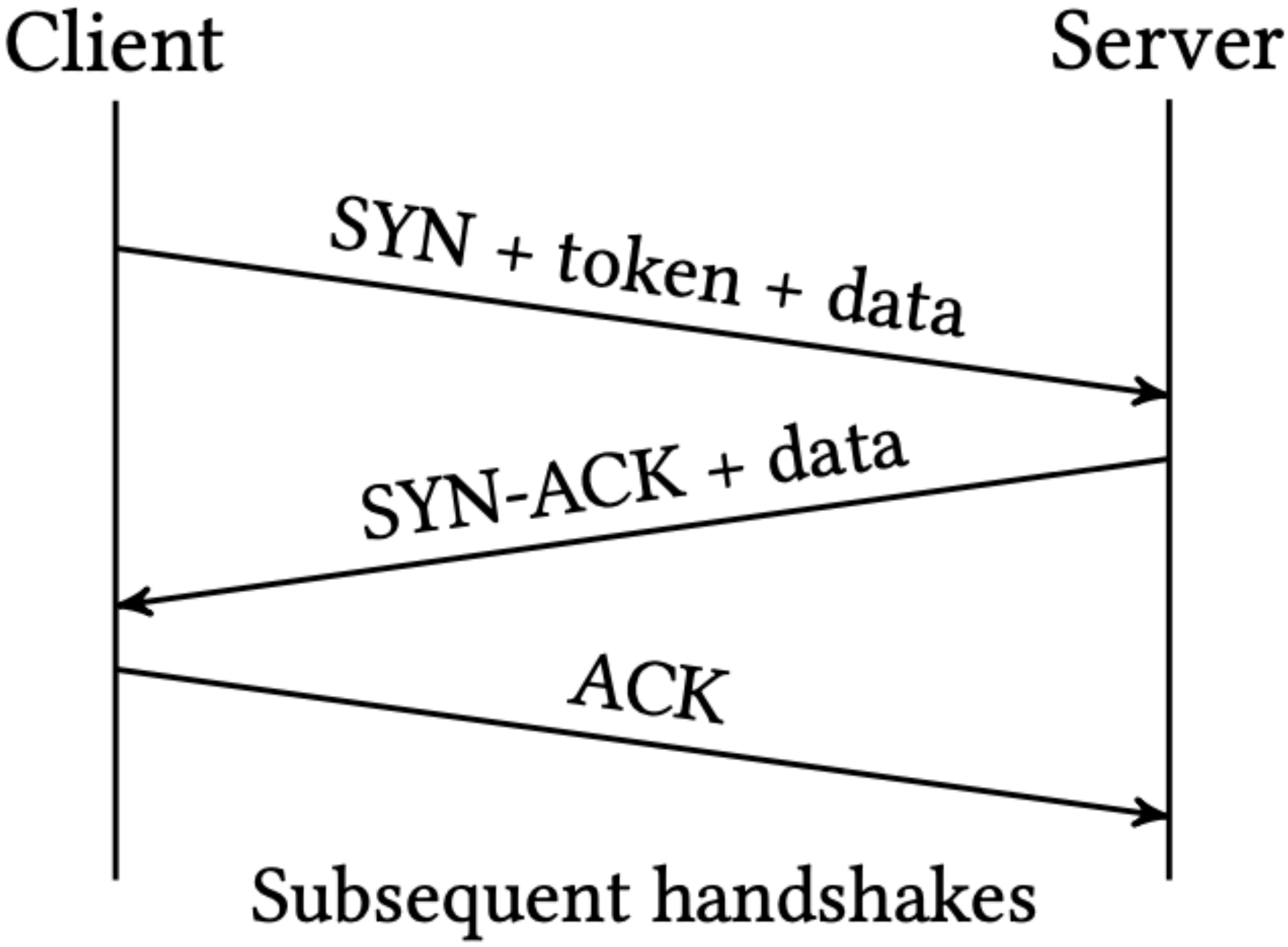


Diagram of subsequent connections of the TFO protocol

(d) (3.0 pt) Alice modifies her protocol to use a cryptographic token. When a client and server connect for the first time:

i. The client sends a SYN packet with a token request.

ii. The server generates a token with a MAC function using a key known only to the server and responds with a SYN-ACK packet to the client containing the token. The client and server both store the token.

iii. The client responds with an ACK packet, as in normal TCP.

In subsequent connections, the client skips the 3-way handshake by sending the SYN packet with both the token and data (similar to Alice's modification from previous parts). The server verifies the value of the token and acknowledges both the SYN and the data. The server may begin sending data to the client before receiving the client's ACK as part of the handshake. The server rejects the SYN and data if the token is invalid.

Here are diagrams detailing the protocol:

Initial connection:

Subsequent connections:

Which of the following attacks on TCP becomes more difficult with the addition of the token? Select all that apply.

□ RST injection

□ MITM hijacking

□ Blind hijacking

□ None of the above

(e) **(3.0 pt)** A major issue with this protocol is that it is vulnerable to replay attacks, as an adversary can spoof a connection by replaying the token. A potential workaround is to modify the TTL (time to live) of the token. Name **one** benefit and **one** drawback of using a shorter TTL rather than a longer TTL.

7. Full-Stack Security

Examtool is a test-taking website located at `https://exam.cs161.org/`. Assume that all network connections are made over HTTPS, unless otherwise specified.

Examtool uses session tokens for user authentication. Session tokens are stored as cookies with `Domain=exam.cs161.org` and no other cookie attributes (no `Secure` flag, no `HttpOnly` flag, `Path=/`).

When a student fills out or changes an answer, their browser makes a POST request to `https://exam.cs161.org/submit_que` with the student's updated answers.

(a) (5.0 pt) Which of the following attacks could allow an adversary to read the session token cookie? Select all that apply.

□ Stored XSS attack at `https://exam.cs161.org/`

□ Reflected XSS attack at `https://exam.cs161.org/`

□ Root access to the Wi-Fi access point that the student is using

□ Root access to another device on the same Wi-Fi network that the student is using

□ Exploitable buffer overflow vulnerability in browser

□ None of the above

(b) (4.0 pt) For a question on an exam, Alice first submits "A" and then later changes her answer and submits "B". What could a MITM attacker between Alice's computer and the `exam.cs161.org` server do? Select all that apply.

□ Perform a replay attack to restore Alice's saved answer to "A"

□ Modify Alice's submitted answer choice to "C"

□ Run JavaScript in Alice's browser

□ Perform a DoS attack to prevent Alice from submitting an answer choice

□ None of the above

(c) (4.0 pt) Suppose the MITM attacker has identified a vulnerability in HTTPS that allows them to arbitrarily read and modify data in transit without detection. Alice submits another answer. What could a MITM attacker between Alice's computer and the `exam.cs161.org` server do? Select all that apply.

□ Redirect Alice's browser to `https://evil.com/`

□ Change Alice's answer choice without detection

□ Set cookie values for the page at `https://exam.cs161.org/`

□ Access any file on Alice's computer

□ None of the above

(d) (2.0 pt) This subpart and following subparts are independent of previous subparts.

An instructor uploads an exam to Examtool by applying some cryptography to the exam and sending it over an insecure channel.

Assumptions:

- m is the message to encrypt (i.e. the exam).
- $\|$ is concatenation.
- k_1 and k_2 are two different secret keys known only to the Examtool server and the instructor.
- $E(k, m)$ is the encryption function of an IND-CPA secure symmetric encryption scheme.
- $\text{MAC}(k, m)$ is a secure MAC function.

For each pair of cryptographic schemes, select the scheme with fewer potential vulnerabilities.

Select the more secure scheme:

○ $C = C_1 \| C_2$, where $C_1 = E(k_1, m)$ and $C_2 = \text{MAC}(k_1, C_1)$

○ $C = C_1 \| C_2$, where $C_1 = E(k_1, m)$ and $C_2 = \text{MAC}(k_2, C_1)$

(e) (2.0 pt) Select the more secure scheme:

○ $C = C_1 \| C_2$, where $C_1 = E(k_1, m)$ and $C_2 = \text{MAC}(k_2, C_1)$

○ $C = E(k_1, m || \text{MAC}(k_2, m))$

8. "Bank-Grade" Security

Bear Bank is using a third-party analytics service called ABtesters. To use it, the bank website includes a tag to load the ABtesters JavaScript library.

Bear Bank's website is located at `https://bearbank.com` and contains the following HTML:

```
<script src="https://cdn.abtesters.com/lib.js"></script>
<form name="login" action="/login" method="POST">
  <input type="text" name="username" />
  <input type="password" name="password" />
</form>
```

(a) (5.0 pt) In the same-origin policy, which of the following are used in determining the origin of an HTTP webpage? Select all that apply.

□ Server port

□ None of the above

□ Request path

□ Domain name

□ Server IP

□ Protocol

(b) (3.0 pt) Bear Bank is concerned that the ABtesters JavaScript library could steal customer passwords from the login form if the JavaScript library were compromised. Is this a valid concern?

○ No, because the ABtesters JavaScript library can only execute specific JavaScript functions required for its basic functionality.

○ Yes, because the ABtesters JavaScript library executes with the origin of Bear Bank's webpage.

○ No, because `https://cdn.abtesters.com` uses a certificate that is signed for different domain name.

○ Yes, because the ABtesters JavaScript library executes with the origin of ABtester's webpage.

(c) (3.0 pt) Bear Bank decides to move the login form to `https://auth.bearbank.com` and embed it on the homepage (`https://bearbank.com/`) in an iframe.

Can the ABtesters JavaScript library running on Bear Bank's homepage steal customer passwords from the login form in the iframe?

○ No, because the ABtesters JavaScript library has a different origin than the login form.

○ Yes, because the ABtesters JavaScript library can execute any JavaScript it wants on the Bear Bank's homepage.

○ No, because the ABtesters JavaScript library is not developed by Bear Bank itself.

○ Yes, because the ABtesters JavaScript library is running on the same page as the iframe.

(d) (3.0 pt) After a user successfully logs into their account, Bear Bank's website sets a `session_token` cookie to track the user's logged in status and allows users to transfer transfer funds by making a GET request to `https://bearbank.com/transfer`.

Which of the following cookie attributes would cause the `session_token` cookie to be sent in a request to `https://bearbank.com/transfer`? Select all that apply.

□ Domain=auth.bearbank.com; Path=/login; HttpOnly; Secure

□ Domain=bearbank.com; Path=/transfer; Secure

□ Domain=bearbank.com; Path=/transactions

□ None of the above

(e) (3.0 pt) Bear Bank realizes that there are no CSRF protections on the transfer form, which means attackers can steal money from users' accounts.

Which of the following methods are are reliable defenses against CSRF attacks? Select all that apply.

□ Move the transfer form to an iframe hosted at `https://transfer.bearbank.com`

□ Check the referrer header on the server when processing the transfer form submission

□ Add a random CSRF token to the transfer form each time the page loads

□ None of the above

(f) (4.0 pt) The following subparts are independent of the previous subparts.

Tree Bank is a different bank considering alternative security methods. Once a user is logged in, they can send HTTP requests to Tree Bank to make transactions. Each request contains a session token set by the server when the user first logged in. The requests do not contain any counters or timestamps. The requests are sent over HTTP (not HTTPS).

Eve is an on-path attacker.

Eve observes a single request from EvanBot to Tree Bank, which contains a transaction. What can Eve do? Select all that apply.

□ Learn the contents of EvanBot's transaction

□ Repeat EvanBot's transaction

□ Learn EvanBot's session token

□ Learn EvanBot's password

□ None of the above

(g) (4.0 pt) Assume that the user knows Tree Bank's public key, and Tree Bank's corresponding private key has not been compromised.

Suppose that Tree Bank requires that the user encrypt the entire HTTP request (including the transaction and token) with the ElGamal scheme from lecture before sending it to the bank.

Eve observes a single encrypted request from EvanBot to Tree Bank, which contains a transaction. What can Eve do? Select all that apply.

□ Repeat EvanBot's transaction

□ Learn EvanBot's password

□ Learn the contents of EvanBot's transaction

□ Learn EvanBot's session token

□ None of the above

(h) (3.0 pt) What is the best way for the bank to defend against Eve's attacks, and what concept best describes the design flaw that allowed Eve to compromise EvanBot's requests?

○ Use TLS. Security is economics.

○ Use TLS. Least privilege.

○ Use TLS. Don't build your own crypto.

○ Use DNSSEC. Security is economics.

○ Use DNSSEC. Least privilege.

○ Use DNSSEC. Don't build your own crypto.

9. Storefront

Definitions for the relevant C functions are given below:

```
int strncmp(const char *s1, const char *s2, size_t n);

       The  strncmp()  function  compares  the  first (at most) n bytes of two
       strings s1 and s2. It returns  an  integer  less  than,  equal  to,  or
       greater  than  zero  if  s1 is found, respectively, to be less than, to
       match, or be greater than s2.

char *fgets(char *s, int size, FILE *stream);

       fgets()  reads in at most one less than size characters from stream and
       stores them into the buffer pointed to by s.  Reading  stops  after  an
       EOF  or a newline.  If a newline is read, it is stored into the buffer.
       A terminating null byte ('\0') is stored after the  last  character  in
       the buffer
```

Consider the following vulnerable C code:

```
void copy_string(char *dst, const char *src, size_t n) {
    for (size_t i = 0; i < n + 1; i++) {
        dst[i] = src[i];
        if (src[i] == '\0') {
            break;
        }
    }
}

void add_to_store(char *lst) {
    char listing[256];

    copy_string(listing, lst, 256);

    printf("Contacting server to add: %s...\n", listing);
    contact_server_and_wait(listing); // Implementation not shown.
}

void invoke(char *lst) {
    add_to_store(lst);
}

int main(void) {
    char buf[4096];
    do {
        fgets(stdin, buf, 4096);
        invoke(buf);
    } while (strcmp(buf, "exit") != 0);
    return 0;
}
```

Assume you are on a little-endian 32-bit x86 system. Assume that there is no compiler padding or saved additional registers in all questions. For the first four parts, assume that **no memory safety defenses** are enabled.

(a) (3.0 pt) Which of the following memory safety vulnerabilities is present in this code?

○ Format string vulnerability

○ Signed/unsigned vulnerability

○ Off-by-one

○ None of the above

(b) (3.0 pt) Which of the following values on the stack can be partially or completely overwritten by the call to `copy_string` at line 13? Select all that apply.

Hint: Draw a stack diagram.

□ SFP of `add_to_store`

□ RIP of `add_to_store`

□ `listing`

□ None of the above

(c) (5.0 pt) Assume that the address of `listing` is 0xf2e6f630. Construct an input at Line 26 that would allow an attacker to execute malicious shellcode. You may reference the variable `SHELLCODE` as a 28-byte shellcode in your answer. Write your answer in Python 2 syntax (just like in Project 1).

(d) (3.0 pt) Your exploit from above may not necessarily work with all possible addresses of `listing`. Provide **one** such address that would prevent your exploit from working. Write your answer in a format like `0xdeadbeef`.

(e) (3.0 pt) Which of the following techniques could an attacker use to execute malicious shellcode if W^X and no other defenses are enabled? Select all that apply.

□ Server-side request forgery

□ ret2esp

□ Return-oriented programming

□ None of the above

(f) (2.0 pt) True or false: Stack canaries with no other defenses would prevent an attacker from executing malicious shellcode in this code (not necessarily using your exploit from above). Assume that all 4 bytes of the stack canary are randomized.

○ True

○ False

(g) (2.0 pt) Justify your answer from the previous part.

(h) (2.0 pt) True or false: ASLR with no other defenses would prevent an attacker from executing malicious shellcode in this code (not necessarily using your exploit from above).

○ True

○ False

(i) (2.0 pt) Justify your answer from the previous part.

10. Cat

(a) **(0.0 pt)** What is the name of Nick's gray cat?

No more questions.