

For any logistics emergencies during the exam, email [cs161-staff@berkeley.edu](mailto:cs161-staff@berkeley.edu) or text (252) 410-1123.

For questions with **circular bubbles**, you may select exactly *one* choice on Gradescope.

- Unselected option
- Only one selected option

For questions with **square checkboxes**, you may select *one* or more choices on Gradescope.

- You can select
- multiple squares

For questions with a **large box**, you need to write your answer in the text box on Gradescope.

There is an appendix at the end of this exam, containing descriptions of all C functions used on this exam.

You have 170 minutes, plus a 10-minute buffer for distractions or technical difficulties, for a total of 180 minutes. There are 11 questions of varying credit (200 points total).

The Gradescope answer sheet assignment has a time limit of 180 minutes. Do not click "Start Assignment" until you're ready to start the exam. The password to decrypt the PDF is at the top of the answer sheet.

The exam is open note. You can use an unlimited number of handwritten cheat sheets, but you must work alone.

Clarifications will be posted at <https://cs161.org/clarifications>.

### Q1 **MANDATORY** – *Honor Code*

(5 points)

Read the following honor code and type your name on Gradescope.

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, at minimum, negative points on the exam and a corresponding notch not on Nick's staff but on his Stanley Fubar demolition tool.

**This is the end of Q1. Leave the remaining subparts of Q1 blank on Gradescope, if there are any. Proceed to Q2 on your answer sheet.**

**Q2 True/false****(34 points)**

Each true/false is worth 2 points.

Q2.1 TRUE or FALSE: A cookie with the Secure flag set cannot be exploited in an XSS attack.

- TRUE  FALSE

Q2.2 TRUE or FALSE: A stack canary is placed above the local variables but below the rip of a given stack frame in order to defend against buffer overflow vulnerabilities.

*Clarification during exam:* “below the rip” means “somewhere below the rip,” not necessarily directly below the rip.

- TRUE  FALSE

Q2.3 TRUE or FALSE: Secure cryptographic hash functions provide IND-CPA confidentiality on a message because they are irreversible.

- TRUE  FALSE

Q2.4 Let  $E$  be an IND-CPA secure encryption scheme, and  $E'(x) = E(x) \parallel \text{len}(x)$ . In other words,  $E'(x)$  is the ciphertext  $E(x)$  concatenated with the length of the plaintext  $x$ .TRUE or FALSE:  $E'$  is IND-CPA secure.

- TRUE  FALSE

Q2.5 TRUE or FALSE: Modern, freely-available computer vision programs have become powerful enough to make CAPTCHAs obsolete.

- TRUE  FALSE

Q2.6 TRUE or FALSE: The Great Firewall of China can inject TCP RST packets to censor connections.

- TRUE  FALSE

Q2.7 TRUE or FALSE: Modern systems enable stack canaries, W^X, ASLR, and pointer authentication to defend against buffer overflow attacks. This is an example of defense-in-depth.

- TRUE  FALSE

Q2.8 TRUE or FALSE: It is possible to inspect encrypted HTTPS traffic with a HIDS.

- TRUE  FALSE

Q2.9 TRUE or FALSE: It is easier for an off-path attacker to inject messages into a TCP connection if the initial sequence numbers were derived from the current time (with second precision) than if the initial sequence numbers were generated randomly.

- TRUE  FALSE

Q2.10 TRUE or FALSE: TLS is vulnerable to RST injection attacks during the handshake, but not after the handshake is completed.

TRUE  FALSE

Q2.11 TRUE or FALSE: Input sanitation helps defend against some SQL injection and XSS attacks.

TRUE  FALSE

Q2.12 TRUE or FALSE: Randomizing the source IP and port is a common defense against DNS spoofing.

TRUE  FALSE

Q2.13 TRUE or FALSE: Of the security principles covered in class, two factor authentication is best described as an example of defense in depth.

TRUE  FALSE

Q2.14 TRUE or FALSE: DNS (without DNSSEC) is secure against an on-path attacker, but not a MITM attacker.

TRUE  FALSE

Q2.15 TRUE or FALSE: ASLR prevents all buffer overflow attacks.

TRUE  FALSE

Q2.16 TRUE or FALSE: Log analysis is effective at detecting attacks in real-time.

TRUE  FALSE

Q2.17 TRUE or FALSE: Clickjacking can help an attacker execute reflected XSS attacks.

TRUE  FALSE

Q2.18 TRUE or FALSE: EvanBot is a real bot. (0 points)

TRUE  FALSE

**This is the end of Q2. Leave the remaining subparts of Q2 blank on Gradescope, if there are any. Proceed to Q3 on your answer sheet.**

### Q3 Indirection

(24 points)

Consider the following vulnerable C code:

```
1 #include <stdlib.h>
2 #include <string.h>
3
4 struct log_entry {
5     char title [8];
6     char *msg;
7 };
8
9 void log_event(char *title , char *msg) {
10     size_t len = strlen(msg, 256);
11     if (len == 256) return; /* Message too long. */
12     struct log_entry *entry = malloc(sizeof(struct log_entry));
13     entry->msg = malloc(256);
14     strcpy(entry->title , title);
15     strncpy(entry->msg, msg, len + 1);
16     add_to_log(entry); /* Implementation not shown. */
17 }
```

Assume you are on a little-endian 32-bit x86 system and no memory safety defenses are enabled.

Q3.1 (3 points) Which of the following lines contains a memory safety vulnerability?

- (A) Line 10                       (D) Line 15
- (B) Line 13                       (E) —
- (C) Line 14                       (F) —

Q3.2 (3 points) Seeing an opportunity to exploit this program, you fire up GDB and step into the `log_event` function. Give a GDB command that will show you the address of the rip of the `log_event` function. (Abbreviations are fine.)

Enter your answer in the text box on Gradescope.

- (G) —     (H) —     (I) —     (J) —     (K) —     (L) —

Q3.3 (3 points) Fill in the numbered blanks on the following stack and heap diagram for `log_event`. Assume that lower-numbered addresses start at the bottom of both diagrams.

*Clarification during exam:* The stack diagram shown is incorrect. The values `1` and `title` should be swapped on the stack.

Stack	Heap
title	3
1	2
rip	
sfp	
len	
entry	

- (A) 1 = entry->msg    2 = entry->title    3 = msg  
 (B) 1 = entry->msg    2 = msg    3 = entry->title  
 (C) 1 = msg    2 = entry->title    3 = entry->msg  
 (D) 1 = msg    2 = entry->msg    3 = entry->title  
 (E) —  
 (F) —

Using GDB, you find that the address of the rip of `log_event` is `0xbfffe0f0`.

Let `SHELLCODE` be a 40-byte shellcode. Construct an input that would cause this program to execute shellcode. Write all your answers in Python 2 syntax (just like Project 1).

Q3.4 (6 points) Give the input for the `title` argument.

*Enter your answer in the text box on Gradescope.*

- (G) —     (H) —     (I) —     (J) —     (K) —     (L) —

Q3.5 (6 points) Give the input for the `msg` argument.

*Enter your answer in the text box on Gradescope.*

- (A) —     (B) —     (C) —     (D) —     (E) —     (F) —

Q3.6 (3 points) Which of the following defenses on their own would prevent your exploit?

Note: If stack canaries are enabled, you can assume `0xbfffe0f0` is still the correct address of the RIP.

(G) Stack canaries

(J) None of the above

(H)  $W^X$

(K) —

(I) ASLR

(L) —

**This is the end of Q3. Leave the remaining subparts of Q3 blank on Gradescope, if there are any. Proceed to Q4 on your answer sheet.**

**Q4 Malcode****(12 points)**

Q4.1 (3 points) Malcode X spreads by making a copy of its own binary on another machine and executing it. Which intrusion detection technique is best for detecting this malcode?

- (A) Signature-based detection                       (D) Behavioral detection  
 (B) Anomaly-based detection                       (E) —  
 (C) Specification-based detection                       (F) —

Q4.2 (3 points) Malcode X connects to other machines using TLS. Which intrusion detection method is best for detecting this malcode?

**Select one option, and briefly justify your answer (1 sentence) in the text box.**

- (G) NIDS     (H) HIDS     (I) —     (J) —     (K) —     (L) —

Q4.3 (3 points) Malcode Y spreads by encrypting its binary, copying the encrypted binary and a decryption script to another machine, and executing the decryption script to run the malcode. The encryption key and the IV/nonce (if needed) are randomly generated each time the malcode replicates. Which encryption schemes would cause every copy of the malcode to look different?

*Clarification during exam:* “Cause every copy of the malcode to look different” means that the encrypted copies of the malcode differ in at least 1 byte.

- (A) AES-ECB                       (C) AES-CTR                       (E) —  
 (B) AES-CBC                       (D) None of the above                       (F) —

Q4.4 (3 points) Malcode Z spreads the same way as Malcode Y. However, instead of randomly generating the encryption key and the IV/nonce (if needed), they are hard-coded into the binary and the decryption script. Which encryption schemes would cause every copy of the malcode to look different?

- (G) AES-ECB                       (I) AES-CTR                       (K) —  
 (H) AES-CBC                       (J) None of the above                       (L) —

**This is the end of Q4. Leave the remaining subparts of Q4 blank on Gradescope, if there are any. Proceed to Q5 on your answer sheet.**

**Q5 CalCentral Security**

**(20 points)**

Given your performance as a skilled attacker of the UnicornBox website, university administrators have asked you to assess the security of the CalCentral platform.

The CalCentral website is set up as follows:

- CalCentral is located at `https://calcentral.berkeley.edu/`.
- The Central Authentication Service (CAS) is located at `https://auth.berkeley.edu/`.
- CalCentral uses session tokens stored in cookies for authentication, similar to Project 3. The session token cookie has domain `berkeley.edu`, and the `Secure` and `HttpOnly` flags are set.
- CalCentral does **not** use CSRF tokens or any form of CSRF protection.

Each subpart is independent.

Q5.1 (3 points) When a user attempts to sign in on CalCentral, the CAS login portal appears in a pop-up window.

TRUE OR FALSE: Because CalCentral and CAS have the same origin, CAS can update the CalCentral webpage when a user signs in successfully.

- (A) True, because CalCentral and CAS are managed by the same organization.
- (B) True, because windows with the same origin can interact with each other.
- (C) False, because pop-up windows can never affect other windows, regardless of the origin.
- (D) False, because CalCentral and CAS don't have the same origin.
- (E) —
- (F) —

Q5.2 (3 points) When a user attempts to sign in on CalCentral, the CAS login portal appears in an `i` frame embedded on the CalCentral page.

TRUE OR FALSE: This design allows CalCentral to modify the text field on the CAS website to autofill the username field.

- (G) True, because CalCentral and CAS are managed by the same organization.
- (H) True, because the inner frame is loaded with the same origin of the outer frame.
- (I) False, because Javascript is needed to autofill form fields.
- (J) False, because the outer frame cannot affect the contents of the inner frame.
- (K) —
- (L) —



Q5.3 (3 points) If a user is logged into CalCentral (has a valid session token cookie), a GET Request to `https://calcentral.berkeley.edu/api/photo/` will contain a response with their CalCentral photo. The website `https://evil.com/` loads an image with the following HTML snippet:

```
<image src="https://calcentral.berkeley.edu/api/photo/">
```

TRUE OR FALSE: If a user is currently signed into CalCentral, the `https://evil.com/` website will be able to successfully display their photo.

- (A) True, because the browser attaches the session token in the request to CalCentral.
- (B) True, because the referer in the request is `https://calcentral.berkeley.edu`.
- (C) False, because the browser does not attach the session token in the request to CalCentral.
- (D) False, because the referer in the request is `https://evil.com`.
- (E) —
- (F) —

Q5.4 (3 points) You find a reflected XSS vulnerability on CAS. `https://berkeley.edu` has a footnote that says “UC Berkeley.”

TRUE OR FALSE: Using this vulnerability, you can cause the victim to see “CS 161 Enterprises” in the footnote when they visit `https://berkeley.edu`.

*Clarification during exam:* The footnote on `https://berkeley.edu/` is part of the static HTML page.

- (G) True, because the script runs with the same origin as `https://berkeley.edu`.
- (H) True, because XSS subverts the same-origin policy.
- (I) False, because the script runs with a different origin from `https://berkeley.edu`.
- (J) False, because the script only affects the browser’s local copy of the site.
- (K) —
- (L) —

Q5.5 (3 points) You find a stored XSS vulnerability on CalCentral.

TRUE OR FALSE: Using this vulnerability, you can cause the victim to load CalCentral with the “My Academics” button changed to link to `https://evil.com/`.

- (A) True, because Javascript on a page can change that page’s HTML
- (B) True, because CalCentral does not implement CSRF tokens.
- (C) False, because Javascript on a page cannot change that page’s HTML

(D) False, because `https://evil.com` has a different origin from CalCentral

(E) —

(F) —

Q5.6 (5 points) When a GET Request is made to `https://calcentral.berkeley.edu/api/classes/`, the server checks if the request's cookies contain the user's CalCentral session token. If a valid session cookie is found, the response contains a list of that user's classes. Otherwise, the server responds with `Error`.

Assume that you control each of the domains below. Select all domains where you'd be able to retrieve the class list of a victim who's currently signed in to CalCentral.

*Hint: Recall that CalCentral does not use CSRF tokens or any form of CSRF protection.*

*Clarification during exam:* "Retrieve the class list" means that the attacker is able to learn the class list.

(G) `https://evil.edu/`

(H) `https://berkeley.edu/`

(I) `https://auth.berkeley.edu/`

(J) `https://evil.calcentral.berkeley.edu/`

(K) `http://calcentral.berkeley.edu/`

(L) None of the above

**This is the end of Q5. Leave the remaining subparts of Q5 blank on Gradescope, if there are any. Proceed to Q6 on your answer sheet.**

**Q6 To Believe or Not To Believe**

**(15 points)**

You are a detective at the Universal Conflict-resolution Bureau (UCB). You have been presented with a new case: Alice claims that Bob agreed to pay her \$100. As evidence, she has a message from Bob, “I, Bob, owe Alice \$100,” along with some cryptography applied to the message.

Decide whether each piece of cryptographic evidence below is sufficient to believe her claim that this message is from Bob.

- $m$  is the message from Bob.
- PK, SK is a public-private key pair.
- MAC is a cryptographically secure message authentication code function.
- $k_1$  and  $k_2$  is a secret key shared between Alice and Bob.

*Clarification during exam:*  $k_1$  and  $k_2$  are both secret keys shared between Alice and Bob.

- H is a cryptographically secure hash function.
- $\text{Sign}(\text{SK}, m)$  is a digital signature algorithm signing a message  $m$  with secret key SK.
- Enc, Dec is an IND-CPA secure symmetric encryption scheme.

*Clarification during exam:* For all parts, you verify any signature you are presented with against the corresponding public key.

*Clarification during exam:* For all parts, Alice is not capable of stealing other people’s private keys.

*Clarification during exam:* All subparts are independent.

Q6.1 (3 points) Alice presents you with  $\text{Sign}(\text{SK}, m)$  and PK.

You obtain  $\text{Sign}(\text{SK}_{CA}, \text{“Bob’s public key is PK”})$  from a certificate authority you trust.  $\text{SK}_{CA}$  is the secret key of the CA, and you know the corresponding public key.

- |  |                             |
|--|-----------------------------|
| <input type="radio"/> (A) $m$ must be from Bob.            | <input type="radio"/> (D) — |
| <input type="radio"/> (B) $m$ is not necessarily from Bob. | <input type="radio"/> (E) — |
| <input type="radio"/> (C) —                                | <input type="radio"/> (F) — |

Q6.2 (3 points) Alice presents you with  $H(m)$ .

- |  |                             |
|--|-----------------------------|
| <input type="radio"/> (G) $m$ must be from Bob.            | <input type="radio"/> (J) — |
| <input type="radio"/> (H) $m$ is not necessarily from Bob. | <input type="radio"/> (K) — |
| <input type="radio"/> (I) —                                | <input type="radio"/> (L) — |

Q6.3 (3 points) Alice presents you with  $\text{MAC}(k_1, m)$  and the secret key  $k_1$ .

- (A)  $m$  must be from Bob.
- (B)  $m$  is not necessarily from Bob.
- (C) —
- (D) —
- (E) —
- (F) —

Q6.4 (3 points) Alice presents you with  $\text{MAC}(k_1, \text{Enc}(k_2, m))$  and the secret keys  $k_1$  and  $k_2$ .

- (G)  $m$  must be from Bob.
- (H)  $m$  is not necessarily from Bob.
- (I) —
- (J) —
- (K) —
- (L) —

Q6.5 (3 points) Alice presents you with  $\text{Sign}(\text{SK}, m)$ , PK.

Additionally, Alice generates a certificate with her public key,  $\text{Sign}(\text{SK}_{\text{Alice}}, \text{"Bob's public key is PK"})$  and presents you with the certificate and her public key  $\text{PK}_{\text{Alice}}$ .

*Clarification during exam:* Alice generates a certificate with her private key, not her public key.

- (A)  $m$  must be from Bob.
- (B)  $m$  is not necessarily from Bob.
- (C) —
- (D) —
- (E) —
- (F) —

**This is the end of Q6. Leave the remaining subparts of Q6 blank on Gradescope, if there are any. Proceed to Q7 on your answer sheet.**

**Q7 DKIM**

**(20 points)**

DKIM (DomainKeys Identified Mail) is an email integrity scheme that prevents an attacker from forging the sender (From:) field and thus spoofing an email.

In DKIM, each mail server has a domain, such as 1e100.net, and many public-private key pairs for signing emails. Each key pair has a unique numerical key ID.

To send an email, the sending mail server first verifies the sender’s identity. Then the mail server attaches a DKIM header to the email. The header contains a signature on the email, the server’s domain, and the ID of the key used to generate the signature.

To verify a DKIM header, you need to look up a public key to verify the signature. The public keys are stored using DNSSEC. To find a DKIM public key with domain {DOMAIN} and key ID {KEY\_ID}, you make a DNSSEC query for {Key\_ID}.\_domainkey.{DOMAIN} and receive a TXT type record with the public key.

For example, the name server for 20161025.\_domainkey.1e100.net has a TXT record with a public key. The public key has ID 20161025 and is used by the 1e100.net mail server.

*Clarification during exam:* Assume that \_domainkey subdomain directly issues TXT records for {KEY\_ID}.\_domainkey. For example, the \_domainkey.1e100.net nameserver directly issues TXT records for 20161025.\_domainkey.1e100.net.

Q7.1 (3 points) Which attacks on their own would help you learn how many DKIM keys a mail server is using?

- (A) Zone enumeration with NSEC
- (B) Zone enumeration with NSEC3
- (C) Kaminsky attack
- (D) None of the above
- (E) —
- (F) —

Consider making a DNSSEC query to obtain the public key with ID 12875102 for the domain stanfraud.com. Assume that you start with an empty cache (except the hardcoded KSK of the root). Assume that the \_domainkey subdomain uses a different KSK and ZSK from its parent domain.

For each of the next 3 subparts, list the records of each type in your cache after the query. Your answer should be a list of domains, e.g. google.com, . (root), org, \_domainkey.stanfraud.com.

Q7.2 (3 points) DS type record(s) with hash(es) of the KSKs of the following name server(s):

*Enter your answer in the text box on Gradescope.*

- (G) —
- (H) —
- (I) —
- (J) —
- (K) —
- (L) —

Q7.3 (3 points) DNSKEY type record(s) with the public key(s) (either ZSK or KSK) of the following name server(s):

Enter your answer in the text box on Gradescope.

(A) —  (B) —  (C) —  (D) —  (E) —  (F) —

Q7.4 (3 points) TXT type record(s) with the public key(s) of the following name server(s) or mail server(s):

Enter your answer in the text box on Gradescope.

(G) —  (H) —  (I) —  (J) —  (K) —  (L) —

Q7.5 (5 points) Which of the following private keys, if stolen on their own, would let an attacker create DKIM signatures as the 1e100.net mail server? Select all that apply.

Clarification during exam: The DKIM key refers to the private key in the DKIM key pair.

- |  |  |
|--|--|
| <input type="checkbox"/> (A) ZSK for the net name server           | <input type="checkbox"/> (D) A DKIM key for the 1e100.net domain     |
| <input type="checkbox"/> (B) KSK for the 1e100.net name server     | <input type="checkbox"/> (E) ZSK for the _domainkey.1e100.net domain |
| <input type="checkbox"/> (C) KSK for the web.1e100.net name server | <input type="checkbox"/> (F) None of the above                       |

Q7.6 (3 points) Consider 2 versions of DKIM. In Version A, mail servers use the same key pairs indefinitely. In Version B, each week, every mail server switches all of its key pairs and publishes the old private keys.

You send an email with a valid DKIM signature, and someone else publishes that email months later. Which version(s) of DKIM would let you credibly deny that you sent that email?

Assume that nobody compromises your email account and nobody steals any private signing keys in DNSSEC and DKIM.

Clarification during exam: For Version B of DKIM, assume that the mail server publishes old public keys along with the old private keys.

(G) Neither version

(J) Both Version A and Version B

(H) Version A only

(K) —

(I) Version B only

(L) —

**This is the end of Q7. Leave the remaining subparts of Q7 blank on Gradescope, if there are any. Proceed to Q8 on your answer sheet.**

**Q8 ACK Flood****(12 points)**

Recall that a SYN flooding attack overwhelms a server by sending a huge number of TCP packets with the SYN flag set.

ACK flooding is another DoS attack based on TCP, where the attacker sends a huge number of packets with the ACK flag set. This causes the server to exhaust resources trying to match each ACK packet to an existing TCP connection.

Q8.1 (4 points) Which of the following can be overwhelmed by an ACK flooding attack? Select all that apply.

- (A) Intermediate layer 3 routers (also called autonomous systems)
- (B) DNS name servers that only respond to DNS queries
- (C) HTTP web servers that only respond to HTTP requests
- (D) HTTPS web servers that only respond to HTTP and HTTPS requests
- (E) None of the above
- (F) —

Q8.2 (3 points) Which of the following could effectively defend against ACK flooding attacks while allowing legitimate connections? Select all that apply.

Hint: Recall that the ACK flag is used both in the TCP handshake and in sending messages over TCP.

- (G) Logging
- (H) NIDS
- (I) HIDS
- (J) None of the above
- (K) —
- (L) —

Q8.3 (5 points) Which field(s) in the packet headers must be set to a fixed value (cannot be random garbage) to target a specific process on a specific server?

- (A) Source IP
- (B) Destination IP
- (C) Source port
- (D) Destination port
- (E) Sequence number
- (F) None of the above

**This is the end of Q8. Leave the remaining subparts of Q8 blank on Gradescope, if there are any. Proceed to Q9 on your answer sheet.**



**Q9 Pre-Master Secrets****(15 points)**

Recall that in TLS, the client random value  $R_B$ , the server random value  $R_S$ , and the pre-master secret  $PS$  are used to derive the symmetric keys. This question considers several possible algorithms for deriving the symmetric keys.

For simplicity, suppose that we always generate one master key from  $R_B$ ,  $R_S$ , and  $PS$ , and then we use a hash-based key derivation function (HKDF) to generate the four symmetric keys from the one master key.

Assume that  $R_B$ ,  $R_S$ , and  $PS$  are all randomly generated integers between 0 and  $2^{256}$  unless otherwise stated. For each master key derivation scheme below, select all attacks a man-in-the-middle can do.

*Clarification during exam:* The attacker learns the value of the master key as long as the attacker knows the key derived by the client and the key derived by the server.

*Clarification during exam:* For all parts, assume that the MITM has observed previous handshakes between the client and the server.

*Clarification during exam:* Assume that all addition operations are performed on unsigned, 256-bit integers. The output is taken modulo  $2^{256}$ .

*Clarification during exam:* It is okay for the TLS handshake to fail, as long as the client/server derived a previously-used master key.

Q9.1 (3 points) Standard RSA TLS is used. To generate the master key, ignore  $PS$ , and add the other two values together: Master key =  $R_B + R_S$ .

- (A) Learn the value of the master key
- (B) Make the server derive a previously-used master key in a future handshake
- (C) Make the client derive a previously-used master key in a future handshake
- (D) None of the above
- (E) —
- (F) —

Q9.2 (3 points) Standard RSA TLS is used. To generate the master key, ignore  $R_B$ , and add the other two values together: Master key =  $R_S + PS$ .

- (G) Learn the value of the master key
- (H) Make the server derive a previously-used master key in a future handshake
- (I) Make the client derive a previously-used master key in a future handshake
- (J) None of the above
- (K) —
- (L) —

Q9.3 (3 points) Standard RSA TLS is used. To generate the master key, add the three values together: Master key =  $R_B + R_S + PS$ .

- (A) Learn the value of the master key
- (B) Make the server derive a previously-used master key in a future handshake
- (C) Make the client derive a previously-used master key in a future handshake
- (D) None of the above
- (E) —
- (F) —

Q9.4 (3 points) A buggy version of RSA TLS is used. Instead of generating  $R_S$  randomly, the server increments  $R_S$  by 1 for every new connection. To generate the master key, add the three values together: Master key =  $R_B + R_S + PS$ .

- (G) Learn the value of the master key
- (H) Make the server derive a previously-used master key in a future handshake
- (I) Make the client derive a previously-used master key in a future handshake
- (J) None of the above
- (K) —
- (L) —

Q9.5 (3 points) A buggy version of RSA TLS is used. Instead of generating  $R_S$  randomly, the server increments  $R_S$  by 1 for every new connection. To generate the master key, hash the concatenation of the three values: Master key =  $H(R_B || R_S || PS)$ .

- (A) Learn the value of the master key
- (B) Make the server derive a previously-used master key in a future handshake
- (C) Make the client derive a previously-used master key in a future handshake
- (D) None of the above
- (E) —
- (F) —

**This is the end of Q9. Leave the remaining subparts of Q9 blank on Gradescope, if there are any. Proceed to Q10 on your answer sheet.**

**Q10 Is EvanBot Real?****(22 points)**

The website `isevanbotreal.com` tells users whether EvanBot is real. However, the website creator doesn't want EvanBot to see the answer, so they require users to fill out a CAPTCHA before showing the answer.

The web server stores a database of CAPTCHA images and their corresponding text, where each CAPTCHA is given a unique ID.

```
1 CREATE TABLE captchas (  
2     id INTEGER UNIQUE ,  
3     answer TEXT  
4 );
```

When a client makes a GET request to `isevanbotreal.com/new`, the server returns a random CAPTCHA image and a cookie containing the ID number of that image. To submit the CAPTCHA, the client makes a POST request to `isevanbotreal.com/submit` with the answer text and the cookie containing the ID number of the image.

Q10.1 (5 points) To verify a submitted CAPTCHA, the server runs the following SQL query, replacing `$id` and `$text` with the cookie value and submitted answer text, respectively:

```
SELECT * FROM captchas WHERE id = $id and answer = '$text'
```

If zero rows are returned, the server returns `Incorrect CAPTCHA`. If more than zero rows are returned, the server returns the answer (`Yes`).

Provide an input for `$text` that would cause the server to return `Yes`, regardless of what the actual CAPTCHA text says.

If needed, you can use `$id` or `$text` to represent the value of user input.

*Enter your answer in the text box on Gradescope.*

(A) —     (B) —     (C) —     (D) —     (E) —     (F) —

Q10.2 (5 points) Consider a modification to the verification process. If zero rows are returned, the server returns `Incorrect CAPTCHA`, as before. If exactly one row is returned, the server returns `Yes`. If more than one row is returned, the server returns `Server error`.

Provide an input for `$text` that would cause the server to return `Yes`, regardless of what the actual CAPTCHA text says.

If needed, you can use `$id` or `$text` to represent the value of user input.

*Enter your answer in the text box on Gradescope.*

(G) —     (H) —     (I) —     (J) —     (K) —     (L) —

Q10.3 (3 points) You want to create a malicious link with domain `isevanbotreal.com`. Any user who requests a CAPTCHA and then clicks on your link will receive the answer `Yes`. Is it possible to create such a link?

- (A) Yes, by exploiting an XSS vulnerability (if one exists) on `isevanbotreal.com`.
- (B) Yes, the link is `isevanbotreal.com/submit`.
- (C) No, because links make GET requests, and submissions are made through POST requests.
- (D) No, because links cannot be used for SQL injection.
- (E) —
- (F) —

Q10.4 (3 points) Consider the following Javascript pseudocode:

```
1 // make a GET request for a new CAPTCHA
2 fetch('http://isevanbotreal.com/new');
3
4 injection = ... // a correct SQL injection exploit from subpart 2
5
6 // send a POST request with an input that always solves the
  CAPTCHA
7 response = fetch('http://isevanbotreal.com/submit',
8                 {method: 'POST', body: injection});
9
10 // Display the response
11 alert(response);
```

Does this code successfully display the answer (`Yes`) without solving the CAPTCHA?

- (G) Yes, because the browser attaches the ID cookie in the POST request
- (H) Yes, because the script uses HTTP, not HTTPS
- (I) No, because SQL injection cannot work over Javascript
- (J) No, because the ID cookie is not sent to the server in the POST request
- (K) —

(L) —

Q10.5 (3 points) Which of these defenses would stop your exploit in subpart 2? Select all that apply.

(A) Parameterized SQL

(B) Return the ID as a hidden form field instead of a cookie

(C) Set the HttpOnly flag on the ID cookie

(D) None of the above

(E) —

(F) —

Q10.6 (3 points) Suppose the website implements a defense so that your attack in subpart 2 is no longer possible. There are no other vulnerabilities, and the CAPTCHAs are not solvable by machines, so the only way to receive the answer Yes is for a human to legitimately solve a CAPTCHA. Can a non-human still find a way to solve the CAPTCHAs?

(G) Yes, by using a CAPTCHA solving service

(H) Yes, by brute-forcing the CAPTCHA

(I) No, because CAPTCHA solving services are prohibitively expensive

(J) No, because CAPTCHAs always defend against non-human attackers

(K) —

(L) —

**This is the end of Q10. Leave the remaining subparts of Q10 blank on Gradescope, if there are any. Proceed to Q11 on your answer sheet.**

**Q11** *Hackerman Visits the Voting Booth* (21 points)

Your sketchy friend Jared asks you to use your CS 161 skills to help him rig some sort of election. He hands you a business card with credentials for a Russian supercomputer.

Armed with massive computing power, you show up to the Caltopia polling center. It has a Wi-Fi network secured with standard WPA2-PSK.

Q11.1 (5 points) You observe a WPA 4-way handshake. Which values from the handshake are needed to perform a brute-force search for the Wi-Fi password? Select all that apply.

- (A) ANonce
- (B) SNonce
- (C) The router's MAC address
- (D) The client's MAC address
- (E) The MICs
- (F) None of the above

Q11.2 (4 points) What can you do after successfully brute-forcing the Wi-Fi password? Select all that apply.

- (G) Perform on-path network attacks against victims in the same Wi-Fi network
- (H) Decrypt network traffic encrypted with the PTK of a user who joins the network after you
- (I) Decrypt network traffic encrypted with the GTK
- (J) Decrypt TLS network traffic
- (K) None of the above
- (L) —

Q11.3 (3 points) Which defenses would stop your attack? Select all that apply.

*Clarification during exam:* Assume that the Russian supercomputer is able to brute-force the password in in roughly an hour.

- (A) Changing the Wi-Fi password every day
- (B) Using WPA2-Enterprise
- (C) A modern NIDS system
- (D) None of the above
- (E) —
- (F) —

You arrive at the New Blackwell City polling center. It also has a Wi-Fi network secured with standard WPA2-PSK.

You walk up to a poll worker, claim that you're a fellow poll worker, and ask for the Wi-Fi password. They write the password on a post-it note and give it to you.

Q11.4 (3 points) Which security principle is most closely related to your experience at this polling place?

(G) Consider Shannon's maxim

(J) Consider human factors

(H) Least privilege

(K) Defense in depth

(I) Security is economics

(L) Time of check to time of use

At the Campanile City polling center, you see a DHCP Discover message broadcast to everyone.

Assume your computer has IP address 10.10.10.142, and the network's router and DHCP server have IP address 10.10.10.5. Assume that there are no other machines on the network. Assume there are no reserved or private IP addresses.

You want to return a malicious DHCP Offer that would make you a MITM. What values of the assigned IP address and the gateway IP address could you use in your response?

Q11.5 (3 points) Assigned IP address:

*Enter your answer in the text box on Gradescope.*

(A) —

(B) —

(C) —

(D) —

(E) —

(F) —

Q11.6 (3 points) Gateway IP address:

*Enter your answer in the text box on Gradescope.*

(G) —

(H) —

(I) —

(J) —

(K) —

(L) —

**This is the end of Q11. Leave the remaining subparts of Q11 blank on Gradescope, if there are any. You have reached the end of the exam.**

## C Function Definitions

```
size_t strlen(const char *s, size_t maxlen);
```

The `strlen()` function returns the number of characters in the string pointed to by `s`, excluding the terminating null byte (`'\0'`), but at most `maxlen`. In doing this, `strlen()` looks only at the first `maxlen` characters in the string pointed to by `s` and never beyond `s+maxlen`.

```
char *strcpy(char *dest, const char *src);
```

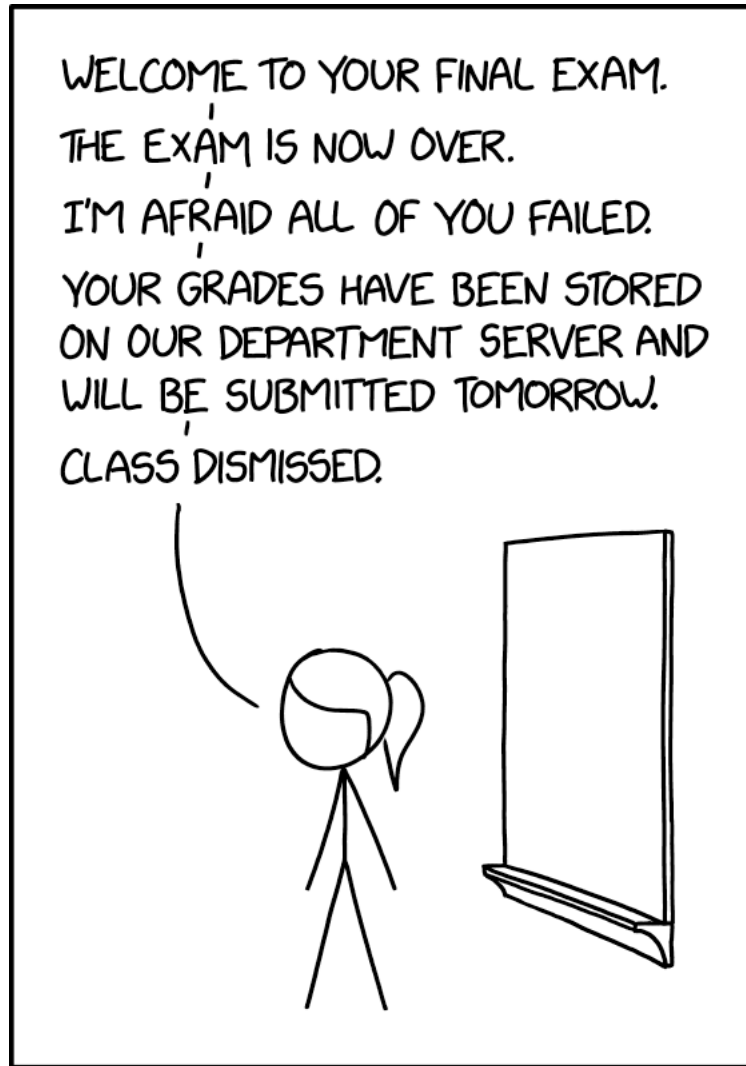
The `strcpy()` function copies the string pointed to by `src`, including the terminating null byte (`'\0'`), to the buffer pointed to by `dest`. The strings may not overlap, and the destination string `dest` must be large enough to receive the copy.

```
char *strncpy(char *dest, const char *src, size_t n);
```

The `strncpy()` function is similar, except that at most `n` bytes of `src` are copied. Warning: If there is no null byte among the first `n` bytes of `src`, the string placed in `dest` will not be null-terminated.

If the length of `src` is less than `n`, `strncpy()` writes additional null bytes to `dest` to ensure that a total of `n` bytes are written.





## CYBERSECURITY FINAL EXAMS

“Final Exam” by Randall Munroe is licensed under [CC BY-NC 2.5](https://creativecommons.org/licenses/by-nc/2.5/).