Peyrin Kao
Fall 2022

CS 161
Computer Security

Midterm

PRINT your name: ______________________, ______________________
(last) (first)

PRINT your student ID: ______________________

You have 110 minutes. There are 9 questions of varying credit (150 points total).

Question:	1	2	3	4	5	6	7	8	9	Total
Points:	3	24	15	28	27	14	7	13	19	150

For questions with **circular bubbles**, you may select only one choice.

○ Unselected option (completely unfilled)

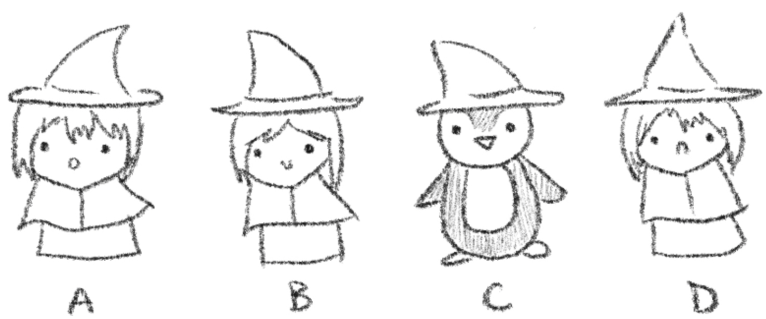
● Only one selected option (completely filled)

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

☐ You can select

■ multiple squares (completely filled)

Pre-exam activity (not graded, just for fun): Circle the imposter. (Difficulty: Extra Hard.)



Q1 *Honor Code* (3 points)

Read the following honor code and sign your name.

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.

SIGN your name: ______________________

Q2 *True/False* **(24 points)**

Each true/false is worth 2 points.

Q2.1 TRUE or FALSE: If an attacker controls `buf`, then the line `printf("%s", buf)` contains a format string vulnerability.

- ○ TRUE
- ○ FALSE

Q2.2 TRUE or FALSE: If we notice that the stack canary has been changed, but the SFP and RIP above the stack canary are unchanged, then it is safe to continue executing the program.

- ○ TRUE
- ○ FALSE

Q2.3 TRUE or FALSE: Non-executable pages, stack canaries, and ASLR are expensive defenses that should only be enabled if your program contains sensitive information.

- ○ TRUE
- ○ FALSE

Q2.4 TRUE or FALSE: If non-executable pages are enabled, it is impossible to execute shellcode on the heap.

- ○ TRUE
- ○ FALSE

Q2.5 Consider a variation of the IND-CPA game. Eve sends three plaintext messages to Alice. Alice randomly selects a message, encrypts it, and sends the encryption back to Eve.

TRUE or FALSE: If Eve cannot guess which message was encrypted with probability greater than $1/3$ (plus some negligible amount), then the scheme is IND-CPA secure.

- ○ TRUE
- ○ FALSE

Q2.6 TRUE or FALSE: If an encryption scheme is not deterministic, then it must be IND-CPA secure.

- ○ TRUE
- ○ FALSE

Q2.7 TRUE or FALSE: A hash function whose output always ends in 0 cannot be a collision-resistant hash function.

- ○ TRUE
- ○ FALSE

Q2.8 Consider a PRNG that takes in a seed and generates pseudorandom output by outputting SHA-2($\text{seed} \| x$), where x is a string of 0's that increases in length each time that output is generated (i.e. $x = 0$, then $x = 00$, then $x = 000$, etc.).

TRUE or FALSE: This is a secure PRNG.

○ TRUE ○ FALSE

Q2.9 TRUE or FALSE: In the Bitcoin protocol, digital signatures prevent an attacker from spending the same coin twice.

○ TRUE ○ FALSE

Q2.10 TRUE or FALSE: After a certificate authority places a certificate on a certificate revocation list, all clients immediately stop accepting the certificate as valid.

○ TRUE ○ FALSE

Q2.11 TRUE or FALSE: The shorter the expiration date on certificates, the longer the certification revocation list.

○ TRUE ○ FALSE

Q2.12 TRUE or FALSE: Since PRNGs are deterministic, their outputs are distinguishable from random to someone knows the internal state.

○ TRUE ○ FALSE

Q3 *Trilogy* (15 points)

Alice, Bob, and CodaBot each generate a random secret: a, b, and c, respectively. They then compute $g^a \bmod p$, $g^b \bmod p$, and $g^c \bmod p$, respectively, and publish these values to everyone else.

CodaBot also generates an RSA key pair SK_C and PK_C. Assume that everyone else knows CodaBot's correct public key.

CodaBot sends a message M to both Alice and Bob. CodaBot also wants to send some additional value(s) to Alice and Bob to ensure authenticity on the message. Eve is a passive eavesdropper who sees the messages CodaBot is sending.

For each scheme below, select who is able to verify that the message came from CodaBot.

Q3.1 (3 points) CodaBot sends M and $\mathsf{HMAC}(c, M)$.

- ○ Alice only
- ○ Alice and Bob only
- ○ Nobody
- ○ Bob only
- ○ Alice, Bob, and Eve

Q3.2 (3 points) CodaBot sends M and $\mathsf{HMAC}(g^{ac} \bmod p, M)$ and $\mathsf{HMAC}(g^{bc} \bmod p, M)$.

- ○ Alice only
- ○ Alice and Bob only
- ○ Nobody
- ○ Bob only
- ○ Alice, Bob, and Eve

Q3.3 (3 points) CodaBot sends M and $\mathsf{HMAC}(g^{c(a+b)}, M)$.

- ○ Alice only
- ○ Alice and Bob only
- ○ Nobody
- ○ Bob only
- ○ Alice, Bob, and Eve

Q3.4 (3 points) CodaBot sends $C = \mathsf{Enc}(g^{ac} \bmod p, M)$ and $\mathsf{HMAC}(g^{ac} \bmod p, C)$.

- ○ Alice only
- ○ Alice and Bob only
- ○ Nobody
- ○ Bob only
- ○ Alice, Bob, and Eve

Q3.5 (3 points) CodaBot sends $C = \mathsf{Enc}(g^{ac} \bmod p, M)$ and $\mathsf{Sign}(SK_C, C)$.

Assume Enc refers to an IND-CPA secure encryption function.

- ○ Alice only
- ○ Alice and Bob only
- ○ Nobody
- ○ Bob only
- ○ Alice, Bob, and Eve

Q4 *So, you want a secure key?* **(28 points)**

Alice wants to create a secure channel of communication with a server. From CS 161, Alice remembers that the best way of communicating is to somehow end up with a shared, symmetric key, but has no idea how this process works.

Assume that there exists a certificate authority (CA) actively sending certificates to many clients. Assume that Mallory, a MITM attacker, and Eve, an eavesdropper, can both exist in all communication channels for all subparts unless otherwise specified.

Q4.1 (2 points) Alice first wants to authenticate the CA before authenticating the server. She remembers that certificates provide authenticity, so she exchanges the following messages with the CA:

1. Alice queries the CA for the CA's public key and receives PK_{CA}.
2. Alice queries the CA for the server's public key and receives $\{\text{"The server's public key is } PK_S\text{"}\}_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK'_S **of Mallory's choosing** as the server's public key without being detected?

○ Yes ○ No

For the rest of this question, assume that instead of querying the CA for their public key, Alice has the CA's correct public key, PK_{CA}, hardcoded into her computer.

Q4.2 (5 points) When Alice queries the CA for the server's public key, the CA sends $\{\text{"The server's public key is } PK_S\text{"}\}_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK'_S, **not necessarily of Mallory's choosing**, as the server's public key without being detected?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

○ Yes ○ No

Q4.3 (5 points) When Alice queries the CA for the server's public key, the CA selects a random number x between 1 and 20 and sends $\{$"The server's public key is PK_S"$\|x\}_{SK_{\text{CA}}^{-1}}$.

Can Mallory trick Alice into accepting a different public key PK'_S, **not necessarily of Mallory's choosing**, as the server's public key without being detected?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

- ○ Yes
- ○ No

Q4.4 (4 points) Alice has received some public key PK_S from the CA, but she doesn't trust that PK_S belongs to the server. Which of the following messages can the server send to convince Alice that she is talking to the legitimate server? Select all that apply.

- ☐ The server sends $H(PK_S)$
- ☐ The server sends $\textsf{Sign}(SK_S, H(PK_S))$
- ☐ The server sends $H(SK_S||PK_S)$
- ☐ The server randomly generates a symmetric key K and sends $(\textsf{Sign}(SK_S, K), \textsf{HMAC}(K, PK_S))$
- ☐ None of the above

For the rest of this question, assume that Alice knows the server's correct public key PK_S.

Q4.5 (4 points) Alice recalls that one of the best ways to come up with a shared, symmetric key is to use a key exchange protocol, so she decides to use Diffie-Hellman. If Alice and the server use the Diffie-Hellman protocol to derive a shared key K, which of the following statements must be true? Select all that apply.

☐ Alice and the server will always derive the same key, K.

☐ Eve can recover the private keys of Alice and the server.

☐ The key exchange protocol provides forward secrecy against Eve.

☐ If Eve records the protocol and then compromises Alice's secret Diffie-Hellman component a, she can derive K.

☐ None of the above

For the rest of this question, assume that Alice has a public-private key pair (SK_A, PK_A). Assume that Alice knows the server's correct public key PK_S, in addition to the server knowing Alice's correct public key PK_A.

Q4.6 (4 points) Alice wants to try a modified version of the Diffie-Hellman key exchange which is as follows:

- Alice sends $(g^a, \mathsf{Sign}(SK_A, g^a))$
- The server sends $(g^s, H(g^s))$
- The shared key is derived as $K = g^{as}$

Can Mallory trick the either Alice or the server into deriving a key that is different from the one they would have derived if Mallory had not existed?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

○ Yes ○ No

Q4.7 (4 points) Alice gives up on independently deriving a shared key and instead decides to share keys instead. To share two randomly generated symmetric keys K_1 and K_2, Alice sends $C = (C_1, C_2)$, where

$$C_1 = \mathsf{PKEnc}(PK_S, K_1 || K_2) \qquad C_2 = \mathsf{HMAC}(K_2, C_1)$$

Can Mallory trick either Alice or the server into deriving a shared key that is different from the one they would have derived if the Mallory had not existed?

If you mark "Yes", provide an attack that would accomplish this goal. If you mark "No", explain why not in 2 sentences or fewer.

○ Yes ○ No

Q5 *AES-161* **(27 points)**

Alice has created a scheme called AES-161 to send messages to Bob securely in the presence of a man-in-the-middle attacker Mallory. Alice and Bob both share a symmetric key K that is secret from everyone else.

The encryption scheme for AES-161 is as follows:

$$C_1 = E_K(IV_1 \oplus M_1)$$
$$C_2 = E_K(C_1 \oplus IV_2 \oplus M_2)$$
$$C_i = E_K(C_{i-1} \oplus C_{i-2} \oplus M_i)$$

Q5.1 (3 points) Write the decryption formula of AES-161 for M_i, for $i > 2$.

Q5.2 (4 points) Is this scheme IND-CPA secure with randomly generated IVs? If you mark "Yes", provide a brief justification (10 words or fewer; no formal proof necessary). If you mark "No", provide a strategy to win the IND-CPA game with probability greater than $1/2$.

○ Yes ○ No

Q5.3 (4 points) Select all true statements for messages longer than 2 blocks. Assume that the PRNG is a secure, rollback-resistant PRNG that has been seeded once with a constant, public value.

- ☐ AES-161 is IND-CPA secure if both IV_1 and IV_2 are generated as $H(i)$ where i is a global, monotonically increasing counter that is incremented after every encryption.
- ☐ AES-161 is IND-CPA secure if IV_1 is generated by generating bytes from the PRNG and IV_2 is generated as $\mathsf{HMAC}(K_2, IV_1)$.
- ☐ AES-161 is IND-CPA secure if IV_1 is generated as $\mathsf{HMAC}(K_2, IV_2)$ and IV_2 is generated by generating bytes from the PRNG.
- ☐ AES-161 is IND-CPA secure if IV_1 is generated as $\mathsf{HMAC}(K_2, M_1)$ and IV_2 is $\mathsf{HMAC}(K_2, M_2)$.
- ☐ None of the above

Consider the following attack, called the FEI attack:

Given a ciphertext C of a known plaintext M, Mallory wishes to provide C' such that some subset of blocks of Mallory's choosing would be decrypted to M_i', where both i and M_i' are **any values of Mallory's choosing**. For other values of i, the corresponding M_i's **can be anything**.

For example, let's say Mallory wants to provide a C' so that the first and last blocks of an 8-block message are decrypted into values M_1' and M_8' of her choosing while blocks 2 through 7 are not necessarily values of her choosing. In other words, when Bob decrypts the ciphertext C', he will get

$$M_1' \| x_1 \| x_2 \| x_3 \| x_4 \| x_5 \| x_6 \| M_8'$$

where x_i refers to any value.

Q5.4 (6 points) Alice wishes to send a 3-block message M. Mallory wants to perform the FEI attack on the third block.

Provide a formula for all C_i' that differ from their corresponding C_i in terms of M_i, C_i, M_i', and C_i' for specific values of i. Your formula may also include any public values. You don't need to provide a formula for any $C_i' = C_i$.

Q5.5 (5 points) Assume that Alice is sending a 9-block message. What is the maximum number of blocks that Mallory can perform the FEI attack on?

Q5.6 (5 points) Assume that Alice is sending a 9-block message. Mallory wants to perform the FEI attack on the maximum number of blocks. You can pick which blocks the FEI attack is performed on.

Provide a formula for all C_i' that differ from their corresponding C_i in terms of M_i, C_i, M_i', and C_i' for specific values of i. Your formula may also include any public values. You don't need to provide a formula for any $C_i' = C_i$.

Q6 *Group Chat* **(14 points)**

Recall that the ElGamal scheme from lecture is used to send a message to a single recipient using their public key. We would like to modify this scheme to work in a group chat, where one person can send a message, anyone in the group chat can decrypt the message, and no one outside of the group chat can decrypt the message.

Consider a four-person group chat consisting of Alice, Bob, Charlie, and David. Their private keys are a, b, c, and d. Their public keys are $A = g^a \bmod p$, $B = g^b \bmod p$, $C = g^c \bmod p$, and $D = g^d \bmod p$, respectively, known to everyone (including people outside the group chat).

Q6.1 (4 points) EvanBot proposes an encryption scheme: the four people in the group chat exchange messages to derive a shared value $g^{a+b+c+d+r} \bmod p$. Alice sends the tuple $(g^r, M \times g^{a+b+c+d+r})$ to the group chat.

Is this a valid scheme, where everybody in the group chat can decrypt the message, and no one outside the group chat can decrypt the message? Briefly justify your answer.

○ Yes ○ No

Q6.2 (2 points) Now consider a general group chat consisting of m users, where each message is n bits long. Each user i has a private key a_i, known only to themselves, and a public key $A_i = g^{a_i} \bmod p$, known by everyone.

Is it possible for Alice (who is user $i = 0$) to send a single message of length no more than $O(n)$ that is decryptable by everyone in the group chat but no one outside of the group chat?

○ Yes ○ No

Q6.3 (4 points) Alice, Bob, Charlie, and David (with private keys a, b, c, and d) want to perform a shared key exchange to arrive at a shared key $g^{abcd} \bmod p$. What messages should they all send so that they all arrive at the same shared key, such that no eavesdropper can derive the value of the shared key?

Each message can only contain one value and one recipient, and each participant starts only knowing their private key. Use the format below. For example, if EvanBot is sending key g^e to Peyrin, it would be listed as follows:

Sender ___Evanbot___ Receiver ___Peyrin___ Message ___ g^e ___

Not all rows may be used.

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Sender ________ Receiver ________ Message ________

Q6.4 (4 points) Alice, Bob, Charlie, and David have executed the key exchange from the previous subpart. Now, Alice realizes that she doesn't like David and doesn't want him to see her messages to Bob and Charlie.

Is it possible for Alice to send **one** message in the group chat such that Bob and Charlie can read it, but not David? Your message may not scale with the number of users in the group chat.

Assume that Alice, Bob, Charlie, and David can perform any number of key exchanges amongst themselves, but cannot generate any new keys.

○ Yes ○ No

Briefly explain why or why not.

Q7 *Small Hulk* **(7 points)**

Consider the following vulnerable C code:

```c
void hulk(char *eyes) {
    char anger[16];
    strcpy(anger, eyes);
    printf("Hulk SMASH! %s\n", anger);
}
```

In this question, your goal is to delete the `smash.txt` file, which Hulk uses to smash his targets! Here are a few tools you can use:

- The `remove` standard C library method can be used to delete a file. The signature of the `remove` function is provided in the C appendix.
- The address of the `remove` function is `0xdeadbeef`.
- The address of `anger` is `0xffffdc10`.
- The string `"smash.txt"` exists in memory at `0xffffe644`.

Assume that **non-executable pages are enabled**, but all other memory safety defenses are disabled. Provide a string input to `eyes` that would delete `smash.txt`.

Q8 *Hulk Smash!* **(13 points)**

Assume that:

- For your inputs, you may use `SHELLCODE` as a 16-byte shellcode.
- If needed, you may use standard output as `OUTPUT`, slicing it using Python syntax.
- All x86 instructions are **4 bytes** long.
- For each provided code snippet, you run GDB once, and discover that:
  - The address of the RIP of the `hulk` method is `0xffffcd84`.
  - The address of a `ret` instruction is `0x080722d8`.

Consider the following function:

```c
int hulk(FILE *f, char *eyes) {
    void (* green_ptr)(void) = &green; //function pointer
    char buf[32];
    char str[28];
    fread(buf, 1, 32, f);
    printf("%s", buf);
    fread(buf, 4, 32, stdin);
    if (strlen(eyes) > 28) {
        return 0;
    }
    strncpy(str, eyes, sizeof(buf));
    return 1;
}
```

The following is the x86 code of `void green(void)`:

```
nop
nop
nop
ret
```

Assume that ASLR is enabled including the code section, but all other memory safety defenses are disabled.

Q8.1 (3 points) Fill in the following stack diagram, assuming that the program is paused after executing **Line 5**, including the arguments of `hulk` (the value in each row does not necessarily have to be four bytes long).

Stack

Q8.2 (10 points) Provide an input to each of the boxes below in order to execute `SHELLCODE`.

Provide a string value for `eyes` (argument to `hulk`):

Provide a string for the contents of the file that is passed in as the `f` argument of `hulk`:

Provide an input to the second `fread` in `hulk`:

Q9 *Brainf[REDACTED]* (19 points)

Consider the following code:

```c
void execute(char* commands, FILE *file) {
    int buf_ind = 0;
    int buf_len = 16;
    char buf[buf_len];
    size_t comm_ind = 0;
    while (commands[comm_ind]) {
        if (commands[comm_ind] == 'C') {
            buf_ind += 1;
        } else if (commands[comm_ind] == 'D') {
            buf_ind -= 1;
        } else if (commands[comm_ind] == 'E') {
            printf("%c", buf[buf_ind]);
        } else if (commands[comm_ind] == 'F') {
            printf("%x", &buf[buf_ind]);
        } else if (commands[comm_ind] == 'G') {
            fread(&buf[buf_ind], sizeof(char), 1, file);
        }
        /* assume you are provided two functions: min and max. */
        buf_ind = max(0, min(buf_len, buf_ind));
        comm_ind += 1;
    }
}
```

For this question, assume the following:

- You may use `SHELLCODE` as a 52-byte shellcode.
- Stack canaries are enabled, and all other memory safety defenses are disabled.
- If needed, you may use the standard output as `OUTPUT`, slicing it using Python syntax.
- The RIP of `execute` is located at `0xffffabcc`.
- The top of the stack is located at `0xffffffff`.
- `execute` is called from `main` with the proper arguments.

Q9.1 (4 points) Fill in the following stack diagram, assuming that the program is paused after executing **Line 6**, including the arguments of `execute` (the value in each row does not necessarily have to be four bytes long).

Stack

Q9.2 (12 points) We wish to construct a series of inputs that will cause this program to execute `SHELLCODE` that works 100% of the time.

Provide a string input to variable `commands` (argument to `execute`):

Provide a string for the contents of the file that is passed in as the `file` argument of `execute`:

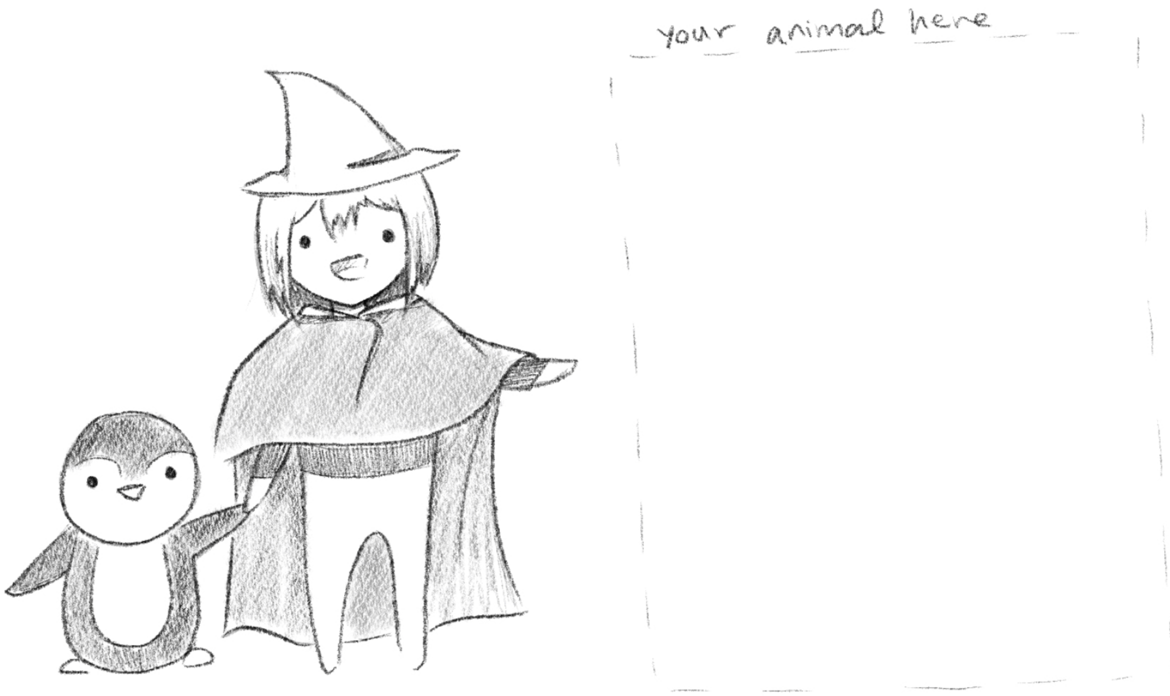
Q9.3 (3 points) If ASLR is now enabled, which of the following modifications to the provided code would allow you to execute `SHELLCODE` 100% of the time? Select all that apply.

- ☐ Line 10 is replaced with `scanf("%u", &buf_ind)`.
- ☐ `jmp *esp` is located in your code at `0xdeadbeef`.
- ☐ Line 14 is replaced with `comm_ind = getchar()`.
- ☐ None of the above

Nothing on this page will affect your grade in any way.

Activity: Zoo

EvanBot made a new friend at the zoo! What animal shall Bot befriend next?



Doodle

Congratulations for making it to the end of the exam! Feel free to leave any final thoughts, comments, feedback, or doodles here: