Nicholas Weaver Spring 2022 — CS 161 Computer Security — Midterm

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

PRINT your student ID: _____

You have 110 minutes. There are 8 questions of varying credit (149 points total).
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)

Q1 *Honor Code* **(1 point)**

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 and a corresponding notch on Nick's Stanley Fubar demolition tool.

SIGN your name: _____

Q2 *True/false* **(30 point)**

Each true/false is worth 2 points.

Q2.1 True or False: In a big-endian system, consecutive bytes are written from high to low addresses, rather than from low to high addresses.

○ True ○ False

Q2.2 True or False: It's impossible to reseed a secure PRNG with an input that would cause it to lose entropy.

○ True ○ False

Q2.3 True or False: If an attacker compromises the internal state of a secure PRNG, and then the PRNG is reseeded with a high-entropy input that the attacker doesn't know, the attacker is no longer able to predict future outputs of the PRNG.

○ True ○ False

Q2.4 True or False: Using a memory safe language is the only way to defend against all serialization vulnerabilities.

○ True ○ False

Q2.5 Consider the following symmetric encryption scheme for encrypting and decrypting messages: $\mathsf{Enc}(K, M) = M^K$, and $\mathsf{Dec}(K, C) = C^{\frac{1}{K}}$. Assume that the key K, message M, and ciphertext C are positive integers.

True or False: This scheme is IND-CPA secure.

○ True ○ False

Q2.6 True or False: In Diffie-Hellman, an attacker who sees $g^a \bmod p$ and b has enough information to learn the shared secret.

○ True ○ False

Q2.7 True or False: A secure block cipher will always map the same input to multiple outputs, since passing the same plaintext through a block cipher will yield different ciphertexts.

○ True ○ False

Q2.8 EvanBot designs a new variant of Diffie-Hellman that requires a second public parameter g_2. EvanBot's scheme is secure using any constant, but everyone must use the same constant.

True or False: EvanBot should explain to everyone how g_2 is generated.

○ True ○ False

Q2.9 During a rocket launch, the launch mechanism depends on the precise positioning of the Earth and the local weather at launch time.

True or False: A possible communications delay between a weather beacon located on the ground and the rocket launch mechanism may present a time-of-check-to-time-of-use issue.

○ True ○ False

Q2.10 Consider a modification to the one-time pad scheme for encrypting a fixed-size message M using a secret key K and initialization vector IV, as described by the following algorithm:

$$C = (IV, K \oplus M \oplus \text{SHA-256}(IV))$$

True or False: If we choose a unique, random IV each time we encrypt a message, then this scheme is IND-CPA secure even when we reuse a key K.

○ True ○ False

Q2.11 True or False: Defense in depth is recommended when protecting legacy C code from memory safety vulnerabilities.

○ True ○ False

Q2.12 Consider a development tool where developers are prompted with two options when setting up a new project: a memory-unsafe language (the default option) and a memory-safe language.

True or False: This is a violation of Shannon's Maxim.

○ True ○ False

Q2.13 Alice wants to verify that a public key, PK_B, belongs to Bob, and she knows that the public key PK_{CA} belongs to a trusted certificate authority. She receives a message from Bob: $\{\text{“}PK_B \text{ belongs to Bob”}\}_{SK_{\text{CA}}^{-1}}$.

True or False: Alice can now trust that PK_B belongs to Bob.

○ True ○ False

Q2.14 Alice wants to verify that a public key, PK_B, belongs to Bob, and she knows that the public key PK_{CA} belongs to a trusted certificate authority. She receives a message from Mallory (whose public key is PK_M): $\left\{\{\text{“}PK_B \text{ belongs to Bob”}\}_{SK_{\text{CA}}^{-1}}\right\}_{SK_M^{-1}}$.

True or False: Alice can now trust that PK_B belongs to Bob.

○ True ○ False

Q2.15 True or False: Salting password hashes with an n-bit salt increases the difficulty of conducting an offline, brute-force attack on all users by a factor of n. Assume that each hash computation itself runs in constant time.

○ True ○ False

Q2.16 (0 points) True or False: Batman is EvanBot.

○ True ○ False

Q3 *The Joker's Schemes* **(17 point)**

The Joker has decided to evaluate the following encryption scheme. Assume the block cipher uses an n-bit block size, and the scheme uses a $2n$-bit $IV = IV_1 \| IV_2$, where IV_1 and IV_2 are each n bits:

$$C_1 = P_1 \oplus E_K(IV_1)$$
$$C_2 = P_2 \oplus E_K(IV_2 \oplus C_1)$$
$$C_i = P_i \oplus E_K(C_{i-2} \oplus C_{i-1})$$

Assume that the IV is sent along with the ciphertext in all instances.

Clarification issued during exam: Assume all IVs are generated per message.

Q3.1 (2 points) Write a formula to decrypt P_i (for $i > 2$) using this scheme.

Q3.2 (4 points) Is this scheme IND-CPA secure with a randomly generated IV? If you put yes, provide a brief justification (no formal proof necessary). If you put no, provide a strategy to win the IND-CPA game with probability greater than $\frac{1}{2}$.

○ Yes ○ No

Q3.3 (4 points) Which of the following methods of choosing IV results in an IND-CPA secure scheme? Select all that apply.

- ☐ $IV = 0^{2n}$ (the bit 0 repeated $2n$ times)
- ☐ $IV = H(i)$, where i is a monotonically increasing counter that increments for each message and H is a cryptographic hash function that outputs $2n$ bits
- ☐ $IV = IV_1 \| IV_2$, where IV_1 is a randomly chosen, n-bit number and $IV_2 = 0^n$
- ☐ $IV = IV_1 \| IV_2$, where $IV_1 = 0^n$ and IV_2 is a randomly chosen, n-bit number
- ☐ None of the above

Q3.4 (4 points) The Joker encrypts a 5-block long message and sends it to the Mob. Batman intercepts the encrypted message and changes the second block of the cipher text C_2. Which of the following blocks of plaintext no longer decrypts to its original value? Select all that apply.

- ☐ P_1
- ☐ P_2
- ☐ P_3
- ☐ P_4
- ☐ P_5
- ☐ None of the above

Q3.5 (3 points) Which of the following statements are true about this encryption scheme? Select all that apply.

- ☐ Encryption is parallelizable
- ☐ Decryption is parallelizable
- ☐ If C is the ciphertext of M, then $C' = C \oplus x$ decrypts to the plaintext $M \oplus x$
- ☐ None of the above

Q4 *The Red Hood* **(15 point)**

Jason Todd decides to launch a communications channel in order to securely communicate with the Red Hood Gang over an insecure channel. Jason wants to test different schemes in his attempt to attain confidentiality and integrity.

Notation:

- M is the message Jason sends to the recipient.
- K_1, K_2, and K_3 are secret keys known to only Jason and the recipient.
- ECB, CBC, and CTR represent block cipher encryption modes for a secure block cipher.
- Assume that CBC and CTR mode are called with randomly generated IVs.
- H is SHA2, a collision-resistant, one-way hash function.
- HMAC is the HMAC construction from lecture.

Decide whether each scheme below provides confidentiality, integrity, both, or neither. For all question parts, the ciphertext is the value of C; t **is a temporary value that is not sent as part of the ciphertext.**

Q4.1 (3 points)

$$t = \mathsf{CBC}(K_1, M) \qquad C_1 = \mathsf{ECB}(K_2, t) \qquad C_2 = \mathsf{HMAC}(K_3, t) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q4.2 (3 points)

$$t = \mathsf{ECB}(K_1, M) \qquad C_1 = \mathsf{CBC}(K_2, t) \qquad C_2 = \mathsf{HMAC}(K_3, t) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q4.3 (3 points)

$$C_1 = \mathsf{ECB}(K_1, M) \qquad C_2 = H(K_2 \| C_1) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q4.4 (3 points) For this subpart only, assume that i a monotonically, increasing counter incremented per message.

$$C_1 = \mathsf{CTR}(K_1, M) \qquad C_2 = \mathsf{HMAC}(i, H(C_1)) \qquad C = (C_1, C_2)$$

Clarification issued during exam: Assume that the counter, i, starts at 0.

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q4.5 (3 points) For this subpart only, assume that the block size of block cipher is n, the lengths of K_1 and K_2 are n, the length of M must be $2n$, and the length of the hash produced by H is $2n$.

$$C_1 = \mathsf{CBC}(K_1, K_2) \qquad C_2 = M \oplus C_1 \oplus H(C_1) \qquad C = (C_1, C_2)$$

- ○ Confidentiality only
- ○ Integrity only
- ○ Both confidentiality and integrity
- ○ Neither confidentiality nor integrity

Q5 *Ra's Al Gamal* **(10 point)**

Recall the ElGamal scheme from lecture:

- KeyGen$() = (b, B = g^b \bmod p)$
- Enc$(B, M) = (C_1 = g^r \bmod p, C_2 = B^r \times M \bmod p)$

Q5.1 (3 points) Is the ciphertext (C_1, C_2) decryptable by someone who knows the private key b? If you answer yes, provide a decryption formula. You may use C_1, C_2, b, and any public values.

○ Yes ○ No

Q5.2 (4 points) Consider an adversary that can efficiently break the discrete log problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

○ Yes ○ No

Q5.3 (3 points) Consider an adversary that can efficiently break the Diffie-Hellman problem. Can the adversary decrypt the ciphertext (C_1, C_2) without knowledge of the private key? If you answer yes, briefly state how the adversary can decrypt the ciphertext.

○ Yes ○ No

Q6 *Probability of MANBAT* **(21 point)**

Consider the following vulnerable C code:

```
1 void unsafe() {
2     char buffer[8];
3     gets(buffer);
4 }
5
6 int main(void) {
7     unsafe();
8     return 0;
9 }
```

Assume that there is no compiler padding or additional saved registers in all subparts. Also assume that `SHELLCODE` represents 8-byte shellcode.

You run this code in GDB once and discover that the address of `buffer` is `0x3a9d2800`.

Q6.1 (3 points) *Suppose that no memory safety defenses are enabled.* Which of the following exploits would cause shellcode to execute? Select all that apply.

- ☐ `'A' * 12 + '\x10\x28\x9d\x3a' + SHELLCODE`
- ☐ `SHELLCODE + 'A' * 4 + '\x00\x28\x9d\x3a'`
- ☐ `'A' * 4 + SHELLCODE + '\x04\x28\x9d\x3a'`
- ☐ None of the above

For the rest of this question, if ASLR is enabled, each segment of memory is exactly `0x10000` bytes long, and each starting address always has `0x0000` as the lower (least significant) bits and has 16 random upper (most significant) bits. For example, the stack segment might be located between addresses `0x3a9d0000` to `0x3a9e0000`, but it will not be located between addresses `0x3a9d0100` to `0x3a9e0100`, because the bottom 16 bits are not all zeros.

With ASLR enabled, you run the program in GDB three times and print the address of `buffer` each time. You see the following addresses: `0xef062800`, `0x2aec2800`, and `0x10702800`.

Q6.2 (3 points) *Suppose that ASLR is enabled, and no other memory safety defenses are enabled.* Consider the following exploit: `'A' * 12 + '\x10\x28\x9d\x3a' + SHELLCODE`.

What is the approximate probability that the above exploit will work on any given execution of the program?

- ○ 0
- ○ $1/2^{16}$
- ○ $1/2$
- ○ $1/2^{32}$
- ○ $1/2^{4}$
- ○ 1

Q6.3 (3 points) *Suppose that ASLR and stack canaries are enabled, and no other memory safety defenses are enabled.* Assume that the stack canary is four completely random bytes (no null byte).

Consider the following exploit: `'A' * 16 + '\x14\x28\x9d\x3a' + SHELLCODE`.

What is the approximate probability that the above exploit will work on any given execution of the program?

- ○ 0
- ○ $1/2^{48}$
- ○ $1/2^{16}$
- ○ $1/2^{16\times 32}$
- ○ $1/2^{32}$
- ○ 1

Q6.4 (4 points) Which of the following additional vulnerabilities in the code would increase the probability of success of your exploit from the previous part? Select all that apply.

Clarification during exam: This question part has been dropped. Everyone will receive points on this question part.

- ☐ A vulnerability that lets you read any memory in the program
- ☐ A vulnerability that lets you read any memory from the stack only
- ☐ A bug that causes one byte of the stack canary to always be `0x61`
- ☐ A null byte in the stack canary
- ☐ None of the above

Q6.5 (3 points) *Suppose that ASLR, stack canaries, and non-executable pages are enabled, and no other memory safety defenses are enabled.* Assume that the stack canary is four completely random bytes (no null byte).

Consider the following exploit: `'A' * 16 + '\x14\x28\x9d\x3a' + SHELLCODE`

What is the approximate probability that the above exploit will work on any given execution of the program?

- ○ 0
- ○ $1/2^{48}$
- ○ $1/2^{16}$
- ○ $1/2^{16\times 32}$
- ○ $1/2^{32}$
- ○ 1

Q6.6 (3 points) Consider the following modified version of the original code:

```
1  char buffer[8];
2
3  void unsafe() {
4      gets(buffer);
5  }
6
7  int main(void) {
8      unsafe();
9      return 0;
10 }
```

Suppose that ASLR is enabled, and no other memory safety defenses are enabled.

As before, you run GDB and discover that the address of `buffer` is `0x3a9d2800`.

Consider the following exploit: `'A' * 12 + '\x10\x28\x9d\x3a' + SHELLCODE`.

What is the approximate probability that the above exploit will cause malicious shellcode to execute on any given execution of the program?

- ○ 0
- ○ $1/2^{16}$
- ○ $1/2$
- ○ $1/2^{32}$
- ○ $1/2^{4}$
- ○ 1

Q6.7 (2 points) Generally, if an exploit succeeds with probability $1/2^{20}$, an attacker might try the exploit 2^{20} times until it succeeds. However, whether an attacker is willing to try 2^{20} times depends on whether the code has a timeout after each failed attempt, and whether the code is even worth attacking in the first place.

Which security principle or example is most relevant to this situation?

- ○ Least privilege
- ○ Don't rely on security through obscurity
- ○ Fail-safe defaults
- ○ Know your threat model
- ○ Time-of-check to time-of-use

Q7 *Robin* **(29 point)**

Consider the following code snippet:

```
1  void robin(void) {
2      char buf[16];
3      int i;
4
5      if (fread(&i, sizeof(int), 1, stdin) != 1)
6          return;
7
8      if (fgets(buf, sizeof(buf), stdin) == NULL)
9          return;
10
11     _________
12 }
```

Clarification issued during exam: `fread` returns the number of members read. `fgets` returns NULL if an error occurs.

Clarification issued during exam: If you put "possible," writing "not needed" for a line means that any input to that line would work for the exploit.

Assume that:

- There is no compiler padding or additional saved registers.
- The provided line of code in each subpart compiles and runs.
- `buf` is located at memory address `0xffffd8d8`
- Stack canaries are enabled, and all other memory safety defenses are disabled.
- The stack canary is four completely random bytes (**no null byte**).

For each subpart, mark whether it is possible to leak the value of the stack canary. If you put possible, provide an input to Line 5 and an input to Line 8 that would leak the canary. If the line is not needed for the exploit, you must write "Not needed" in the box.

Write your answer in Python 2 syntax (just like in Project 1).

Q7.1 (3 points) Line 11 contains `printf("%x", buf[i]);`.

Clarification during exam: This question part has been dropped. Everyone will receive points on this question part.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.2 (3 points) Line 11 contains `printf("%s", buf[i]);`.

Clarification during exam: This question part has been dropped. Everyone will receive points on this question part.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.3 (3 points) Line 11 contains `gets(buf);`.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.4 (3 points) Line 11 contains `printf("%s", buf);`.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.5 (5 points) **For this subpart only, enter an input that allows you to leak a single character from memory address `0xffffd8d7`. Mark "Not possible" if this is not possible.** Line 11 contains `printf("%c", buf[i]);`.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.6 (6 points) Line 11 contains `printf(buf);`.

○ Possible ○ Not possible

Line 5:

Line 8:

Q7.7 (6 points) Line 11 contains `printf(i);`.

○ Possible ○ Not possible

Line 5:

Line 8:

Q8 *Copperhead* **(26 point)**

Consider the following vulnerable C code:

```
1  struct vtable {
2      void (*hiss)(void);  // function pointer
3      void (*snack)(void); // function pointer
4      void (*bite)(void);  // function pointer
5  };
6
7  struct snake {
8      char name[16];
9      struct vtable *vtable_ptr;
10 };
11
12 void snake_bite(void) { printf("ouch!\n"); }
13 void viper_bite(void) { printf("ouch + venom!\n"); }
14
15 int main(void) {
16     struct vtable snake_vtable = { NULL, NULL, snake_bite };
17     struct vtable viper_vtable = { NULL, NULL, viper_bite };
18     struct snake copperhead;
19     char venom[32];
20     int i;
21
22     /* Make copperhead. */
23     copperhead.vtable_ptr = &viper_vtable;
24     fgets(venom, sizeof venom, stdin);
25     for (i = 0; i <= 16; i++) {
26         copperhead.name[i] = venom[i];
27     }
28
29     copperhead.vtable_ptr->bite();
30
31     return 0;
32 }
```

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. Assume there are **no memory safety defenses enabled**.

Q8.1 (5 points) Fill in the following stack diagram, assuming that the program is paused at **Line 22**. There are no extra rows. Each row should contain one struct member or variable from the following (not all options will be used):

`copperhead.name`	`copperhead.vtable_ptr`	address of `fgets`
`i`	address of `printf`	`snake_vtable.bite`
`snake_vtable.hiss`	`snake_vtable.snack`	`venom`
`viper_vtable.bite`	`viper_vtable.hiss`	`viper_vtable.snack`

Stack

`RIP of main`
`SFP of main`

Q8.2 (3 points) Which of the following lines contains a memory safety vulnerability?

- ○ Line 12
- ○ Line 16
- ○ Line 19
- ○ Line 23
- ○ Line 24
- ○ Line 25

Q8.3 (10 points) Assume that, at Line 22, the value of the ESP is `0xffff9308`. Provide an input to the program that will cause a malicious 8-byte shellcode to be executed. You may reference the variable `SHELLCODE` in your exploit. Write your answer in Python 2 syntax (just like in Project 1).

Q8.4 (3 points) Assume an attacker has successfully carried out the above exploit. At what point will execution jump to shellcode?

- ○ When Line 24 is executed
- ○ When Line 26 is executed
- ○ When Line 29 is executed
- ○ When `snake_bite` returns
- ○ When `viper_bite` returns
- ○ When `main` returns

Q8.5 (5 points) Which of the following memory safety defenses would prevent an attacker from executing the exploit above? Select all that apply.

- ☐ Stack canaries
- ☐ ASLR
- ☐ Pointer authentication (assuming a 64-bit system)
- ☐ Non-executable pages
- ☐ Rewriting the code in a memory-safe language
- ☐ None of the above