CS 161 Spring 2023 — Introduction to Computer Security — Final

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

PRINT your student ID: _____

There are 11 questions of varying credit (200 points total).

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Points:	2	36	21	15	24	19	30	17	14	22	0	200

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 (for fun, not graded):
Karaoke time! What are your favorite lyrics?

Q1 *Honor Code* (2 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* **(36 points)**

Each true/false is worth 2 points.

Q2.1 When writing the final exam for CS 161, EvanBot decides to share the document with CodaBot, even though CodaBot isn't involved in writing the exam.

TRUE or FALSE: EvanBot is violating the "least privilege" security principle.

○ TRUE ○ FALSE

Q2.2 TRUE or FALSE: MACs are a good example of the "Detect if you can't prevent" security principle.

○ TRUE ○ FALSE

For the next two subparts: Consider a system where ASLR is enabled. You open GDB and find that the address of the RIP of the `foo` stack frame is `0xffff3820`.

Q2.3 TRUE or FALSE: It's possible that the address of the SFP of the same stack frame `foo` is `0xffff3824` on the **same** run of the program.

○ TRUE ○ FALSE

Q2.4 TRUE or FALSE: It's possible that the address of the SFP of the same stack frame `foo` is `0xffff3824` on a **different** run of the program.

○ TRUE ○ FALSE

Q2.5 TRUE or FALSE: Having stack canaries start with a null terminator helps prevent functions like `printf` from reading the canary on the stack, but reduces the number of bruteforce attempts required to guess the canary value.

○ TRUE ○ FALSE

Q2.6 Second-preimage resistance is a property of a cryptographic hash function H: Given $H(a)$, it is computationally hard to find $b \neq a$ such that $H(a) = H(b)$.

TRUE or FALSE: All second-preimage resistant hash functions are also collision resistant.

○ TRUE ○ FALSE

Q2.7 TRUE or FALSE: Symmetric key encryption is generally slower than public-key encryption.

○ TRUE ○ FALSE

Q2.8 TRUE or FALSE: The security of a PRNG is limited by the entropy of its initial seed, assuming the PRNG is never reseeded.

○ TRUE ○ FALSE

Q2.9 TRUE or FALSE: Servers often hash passwords using slow hash functions in order to stop brute force attacks.

○ TRUE ○ FALSE

Q2.10 True or False: A cookie set with `Domain=boogle.com` will be sent to `auth.boogle.com`.

○ True ○ False

Q2.11 True or False: Implementing a secure escaping policy to prevent XSS is easier than using parameterized SQL.

○ True ○ False

Q2.12 True or False: Clickjacking attacks are only possible if a user is logged in and has a session token cookie.

○ True ○ False

Q2.13 True or False: Accepting only the first ARP response for each ARP request is a good way to defend against ARP spoofing attacks.

○ True ○ False

Q2.14 True or False: An on-path attacker who knows the WiFi password can always eavesdrop on new WPA2 connections.

○ True ○ False

Q2.15 True or False: WPA2-Enterprise involves connecting to a third-party authentication server, separate from the Access Point.

○ True ○ False

Q2.16 True or False: In networking, "best effort" means that we make the best effort possible to ensure the packet has reached its destination, often by using SEQ/ACK numbers.

○ True ○ False

Q2.17 True or False: A firewall that blocks all inbound packets will prevent against all network attacks.

○ True ○ False

Q2.18 EvanBot's computer has been infected with a new virus that has not been seen before.

True or False: Behavioral detection can be used to detect the presence of this virus.

○ True ○ False

Q2.19 (0 points) True or False: `EvanBot is a real bot.`

○ True ○ False

Q3 *Memory Safety: No Doubt* **(21 points)**

Consider the following code:

```
1  void to_it(char *buf) {
2      fgets(&buf, 30, stdin);
3      fgets(buf, 30, stdin);
4      return;
5  }
6  int main() {
7      char arr[60];
8      gets(arr);
9      /* printf("Cool cool cool: %x"); */
10     to_it(arr);
11     return 0;
12 }
```

Assumptions:

- You may use `SHELLCODE` as a 40-byte shellcode.
- The RIP of `to_it` is located at `0xffffde20`.
- **Stack canaries are enabled**, but all other memory safety defenses are disabled.

Q3.1 (4 points) Which values can an attacker overwrite (or partially overwrite) using the `fgets` on **Line 2**? Select all that apply.

- ☐ `buf`
- ☐ RIP of `to_it`
- ☐ None of the above
- ☐ `arr[60]`
- ☐ SFP of `to_it`

Provide inputs to execute shellcode.

Q3.2 (4 points) Input to `gets` on Line 8:

Q3.3 (4 points) Input to `fgets` on Line 2:

Q3.4 (4 points) Input to `fgets` on Line 3:

The code, reprinted for your convenience:

```
1  void to_it(char *buf) {
2      fgets(&buf, 30, stdin);
3      fgets(buf, 30, stdin);
4      return;
5  }
6  int main() {
7      char arr[60];
8      gets(arr);
9      /* printf("Cool cool cool: %x"); */
10     to_it(arr);
11     return 0;
12 }
```

Q3.5 (5 points) Now, assume that **ASLR is enabled**, but all other memory safety defenses (including stack canaries) are disabled.

Is it possible to construct an exploit that always executes shellcode?

- ○ Yes, without uncommenting Line 9
- ○ Yes, but only if Line 9 is uncommented
- ○ No, even if Line 9 is uncommented

Briefly justify your answer. For full credit, you should explain why Line 9 does or does not help.

Q4 *Memory Safety: Andor, or XOR?* **(15 points)**

(intro text just for fun) Cassian Andor has asked you to exploit an Empire system. If you can run his `SHELLCODE`, you will leak the plans for the Death Star and save the rebel alliance!

Consider the following code:

```
1  void galaxy(char *clone) {
2      char droid[64];
3      int i = 0;
4      int j = 0;
5      char force[24];
6      gets(droid);
7      gets(force);
8      gets(clone);
9
10     while (0 <= i && i < 24 && 0 <= j){
11         if (clone[i] == 0x54) {
12             clone[i] = force[i] ^ droid[j];
13         } else {
14             clone[i] = force[i] ^ clone[i];
15         }
16         i++;
17         j++;
18     }
19 }
20
21 int main() {
22     char rebel[16];
23     galaxy(rebel);
24     return 0;
25 }
```

Stack at Line 5

RIP of `main`
SFP of `main`
(1)
(2)
(3)
RIP of `galaxy`
SFP of `galaxy`
(4)
`droid`
`i`
`j`
`force`

Assumptions:

- You may use `SHELLCODE` as a 63-byte shellcode.
- `droid` is located at `0xf0e13370`.
- **Stack canaries are enabled**, but all other memory safety defenses are disabled.

Q4.1 (3 points) What values go in the blanks in the stack diagram above?

○ (1) canary	(2) `clone`	(3) `rebel`	(4) canary
○ (1) canary	(2) `rebel`	(3) `clone`	(4) canary
○ (1) `rebel`	(2) canary	(3) canary	(4) `clone`
○ (1) `rebel`	(2) canary	(3) `clone`	(4) canary
○ (1) `clone`	(2) canary	(3) canary	(4) `rebel`

Q4.2 (12 points) Provide inputs to execute shellcode.

Hint: 64 *is represented in hexadecimal as* `\x40`. 16 *is represented in hexadecimal as* `\x10`.

Fill in Box 1, or Box 2, but not both. (If you fill in both, we'll grade the worse of your two answers.)

Box 1 (our solution uses this structure):

`droid` (input to `gets` on Line 6):

```
__________  + '\n'
```

`force` (input to `gets` on Line 7):

```
('\x_____'   *   _____   )   +   '\x_____\x_____\x_____\x_____'
+   '\x_____\x_____\x_____\x_____'  + '\n'
```

`clone` (input to `gets` on Line 8):

```
('A' *   _____   )   +   ('\x_____'   *   _____   )
+ ('B' *   _____   )   + '\x_____\x_____\x_____\x_____' + '\n'
```

Box 2 (use this if you have a solution that doesn't fit our template):

`droid` (input to `gets` on Line 6):

```
_____________________________________________________________________
_____________________________________________________________________
```

`force` (input to `gets` on Line 7):

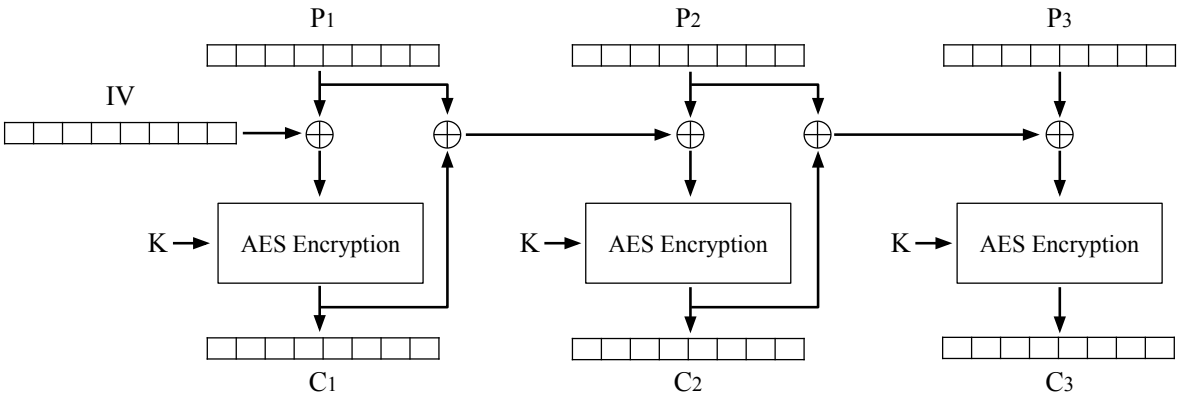
```
_____________________________________________________________________
_____________________________________________________________________
```

`clone` (input to `gets` on Line 8):

```
_____________________________________________________________________
_____________________________________________________________________
```

Q5 *Cryptography: EvanBlock Cipher* **(24 points)**

EvanBot invents a new block cipher chaining mode called the EBC (EvanBlock Cipher). The encryption diagram is shown below:



Q5.1 (2 points) Write the encryption formula for C_i, where $i > 1$. You can use E_K and D_K to denote AES encryption and decryption respectively.

Q5.2 (2 points) Write the decryption formula for P_i, where $i > 1$. You can use E_K and D_K to denote AES encryption and decryption respectively.

Q5.3 (4 points) Select all true statements about this scheme.

- ☐ It is IND-CPA secure if we use a random IV for every encryption.
- ☐ It is IND-CPA secure if we use a hard-coded, constant IV for every encryption.
- ☐ Encryption can be parallelized.
- ☐ Decryption can be parallelized.
- ☐ None of the above

Q5.4 (4 points) Alice has a 4-block message (P_1, P_2, P_3, P_4). She encrypts this message with the scheme and obtains the ciphertext $C = (IV, C_1, C_2, C_3, C_4)$.

Mallory tampers with this ciphertext by changing the IV to 0. Bob receives the modified ciphertext $C' = (0, C_1, C_2, C_3, C_4)$.

What message will Bob compute when he decrypts the modified ciphertext C'?

X represents some unpredictable "garbage" output of the AES block cipher.

- ○ (P_1, P_2, P_3, P_4)
- ○ (X, X, P_3, P_4)
- ○ (X, X, X, X)
- ○ (X, P_2, X, P_4)
- ○ (X, P_2, P_3, P_4)
- ○ None of the above

Alice has a 3-block message (P_1, P_2, P_3). She encrypts this message with the scheme and obtains the ciphertext $C = (IV, C_1, C_2, C_3)$.

Mallory tampers with this ciphertext by swapping two blocks of ciphertext. Bob receives the modified ciphertext $C' = (IV, C_2, C_1, C_3)$.

When Bob decrypts the modified ciphertext C', he obtains some modified plaintext $P' = (P'_1, P'_2, P'_3)$. In the next three subparts, write expressions for P'_1, P'_2, and P'_3.

Q5.5 (4 points) P'_1 is equal to these values, XORed together. Select as many options as you need.

For example, if you think $P'_1 = P_1 \oplus C_2$, then bubble in P_1 and C_2.

- ☐ P_1 ☐ P_2 ☐ P_3 ☐ IV ☐ C_1 ☐ C_2 ☐ C_3

Q5.6 (4 points) P'_2 is equal to these values, XORed together. Select as many options as you need.

- ☐ P_1 ☐ P_2 ☐ P_3 ☐ IV ☐ C_1 ☐ C_2 ☐ C_3

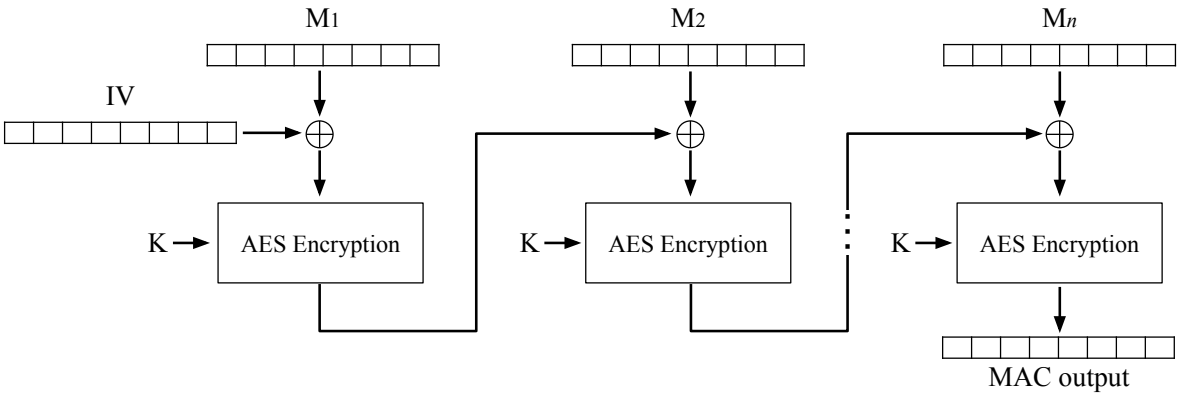
Q5.7 (4 points) P'_3 is equal to these values, XORed together. Select as many options as you need.

- ☐ P_1 ☐ P_2 ☐ P_3 ☐ IV ☐ C_1 ☐ C_2 ☐ C_3

Q6 *Cryptography: Lights, Camera, MACtion* **(19 points)**

Alice and Bob design a new MAC algorithm using the AES block cipher:

$$\mathsf{MAC}(K, IV, M) = E_K(M_n \oplus E_K(\ldots M_2 \oplus E_K(M_1 \oplus IV)))$$



Q6.1 (3 points) Alice has a one-block message M_1. She chooses a random IV and computes: $t = \mathsf{MAC}(K, IV, M_1)$.

Mallory wants to change the message to M_1', without changing the MAC value. To do this, Mallory needs to choose an IV' such that $\mathsf{MAC}(K, IV', M_1') = t$.

What value of IV' should Mallory choose?

- ○ $IV' = IV \oplus M_1$
- ○ $IV' = IV \oplus M_1'$
- ○ $IV' = IV \oplus M_1 \oplus M_1'$
- ○ $IV' = 0$

Q6.2 (3 points) Alice now has a 6-block message $M = (M_1, M_2, \ldots, M_6)$. Alice chooses a random IV and computes a MAC on this message.

Mallory again wants to change the message without changing the MAC value. Which block(s) of the message can Mallory change? Select all that apply.

- ☐ M_1
- ☐ M_2
- ☐ M_3
- ☐ M_4
- ☐ M_5
- ☐ M_6
- ☐ None of the above

For the rest of the question, Alice **always** sets $IV = 0$ when computing MACs.

Alice has a 2-block message $M = (M_1, M_2)$ and a 3-block message $M' = (M'_1, M'_2, M'_3)$. She computes a MAC t on M, and a MAC t' on M'.

Q6.3 (5 points) Mallory sees both messages and their MACs. Construct a valid message/tag pair such that the message is not exactly equal to either M or M'.

Your expressions can include $M_1, M_2, M'_1, M'_2, M'_3, t, t'$.

Message:

Tag:

To fix the vulnerability from the previous subpart, Alice appends a block to the end of the message equal to the length of the message (in blocks). For example, for the message (M_1, M_2), Alice would compute the MAC of $(M_1, M_2, 2)$.

Q6.4 (8 points) Is this scheme secure?

○ Yes ○ No

If you selected "Yes", explain how this prevents the attack from the previous subpart.

If you selected "No", demonstrate another attack, like in the previous part, by providing a valid message/tag combination for some new message. In your attack, you can query for the MACs of messages (that are not the final message you forge a tag for).

Q7 *Web Security: Botgram* **(30 points)**

The website `www.botgram.com` lets users post and view doodles of their Bot friends. Unless otherwise specified, Botgram does not sanitize any inputs.

Botgram stores submitted doodles in their `doodles` database, which has the following schema:

```
1 CREATE TABLE doodles (
2     doodle_url TEXT,
3     submission_timestamp INTEGER
4     -- Additional fields not shown.
5 );
```

When a user submits an image URL, Botgram stores the URL with this SQL query (replacing `%s` with the user-provided URL):

```
INSERT INTO doodles (doodle_url, submission_timestamp)
VALUES '%s', CURRENT_TIMESTAMP;
```

Users can visit `www.botgram.com/latest` to view the 100 doodles with the greatest timestamps.

To display the doodles, each URL is inserted into the HTML of the webpage as follows (replacing `%s` with the URL from the database):

```
<img src='%s'>
```

Q7.1 (4 points) Eve is an attacker who wants to post a doodle with the URL `evil.com/a.jpg` to Botgram. Eve wants to make this doodle stay on `www.botgram.com/latest` for a long time by setting its timestamp to 999.

Provide an input for `doodle_url` that posts Eve's doodle with timestamp 999.

For the rest of the question, assume that Eve's doodles always show up on `www.botgram.com/latest`.

`botgram.com` uses session tokens for authentication. Session tokens are stored as cookies with `Secure = False, HttpOnly = False`.

Eve wants any user who views her doodles to send their session token to `evil.com`.

Q7.2 (4 points) Eve uploads a doodle with the URL `evil.com`. She reasons that the `img` tag will send a GET request to `evil.com` originating from `botgram.com`, which will then attach the session token from `botgram.com` to the request.

Briefly explain why this attack does not work.

Q7.3 (4 points) Provide an input for `doodle_url` that sends the session token of any user that views the doodle to `evil.com`.

You may use the JavaScript function `post(URL, data)` which sends a POST request to the given `URL` with the given `data`.

Q7.4 (3 points) Which of the following cookie attributes would stop the attack from the previous subpart? Select all that apply.

- ☐ `Secure=True, HttpOnly=False`
- ☐ `Secure=True, HttpOnly=True`
- ☐ `Secure=False, HttpOnly=True`
- ☐ None of the above

For the rest of the question, Botgram implements an update that **prevents all JavaScript from executing** on Botgram webpages.

Q7.5 (4 points) Alice is a user on Botgram. Alice performs bank transfers by making a GET request to

```
https://www.bank.com/transfer?amount={AMOUNT}&to={RECEIVER}
```

where `{AMOUNT}` and `{RECEIVER}` are values chosen by Alice.

Provide an input to `doodle_url` that sends $100 to the username "Eve" when Alice loads Botgram. Assume Alice is currently logged into `www.bank.com`.

Q7.6 (3 points) What type of attack did Eve execute in the previous subpart?

○ Stored XSS ○ Reflected XSS ○ CSRF ○ Clickjacking

Q7.7 (5 points) Eve wants to force anyone who loads `www.botgram.com/latest` to make 500 GET requests. What `doodle_url` should Eve submit to Botgram? You can describe the input in words or provide the actual input.

Remember that `www.botgram.com/latest` only loads 100 images, and all JavaScript is disabled.

Q7.8 (3 points) Using the strategy from the previous subpart, give the name of one attack from class that Eve could execute. (There may be multiple correct answers.)

Q8 *Network Security: Life of a Packet* **(17 points)**

Alice, Bob, EvanBot, and CodaBot need to load the CS 161 website to work on Project 3. Mallory wants to cause them to load Mallory's website instead.

Mallory is an on-path attacker who can guarantee that her spoofed packets arrive before any legitimate packets, 100% of the time.

Q8.1 (3 points) Alice connects to the network for the first time and uses DHCP to request a configuration.

How many spoofed packets does Mallory need to send in order to trick Alice into accepting Mallory's malicious DHCP configuration?

○ 0 ○ 1 ○ 2 ○ More than 2

Q8.2 (3 points) Suppose Alice has accepted Mallory's malicious DHCP configuration.

Alice makes a UDP request for the CS 161 website. How many spoofed packets does Mallory need to send in order to cause Alice to load Mallory's website?

Assume that both websites fit in one UDP packet.

○ 0 ○ 1 ○ 2 ○ More than 2

The following subparts are all independent.

Q8.3 (3 points) Mallory notices that a packet is sent through 4 autonomous systems to get from Bob to the CS 161 web server.

Mallory hacks into Nikhil's computer and takes control of AS400555. Can Mallory exploit BGP to cause Bob to load Mallory's website?

○ Yes, but only if AS400555 is one of the 4 ASes between Bob and CS 161

○ Yes, even if AS400555 is not one of the 4 ASes between Bob and CS 161

○ No, because Mallory would need to control all 4 ASes between Bob and CS 161

○ No, because Mallory would need to control at least 2 ASes between Bob and CS 161

Q8.4 (5 points) EvanBot wants to load `http://box.cs161.org` using HTTP (over TCP). Mallory wants to cause EvanBot to load `http://www.mallory.com` instead. Mallory doesn't care if EvanBot loads other pages, as long as EvanBot loads `http://www.mallory.com` at some point. Assume that each website fits in 10 TCP packets.

What is the minimum number of spoofed packets Mallory needs to send to force EvanBot to load `http://www.mallory.com`?

○ Less than 10 ○ 10 ○ More than 10

Briefly describe, in words or pseudocode, the contents of the packet(s) that Mallory spoofs in the previous part.

Q8.5 (3 points) CodaBot wants to load `https://box.cs161.org` over TLS. Mallory wants to cause CodaBot to fail to load the CS 161 website.

Which of the following attacks could potentially make CodaBot unable to load the CS 161 website? Select all that apply.

☐ SYN flooding on the CS 161 web server

☐ TCP RST injection

☐ Spoofing TCP packets with the FIN flag set

☐ None of the above

Q9 *Networking: TLS Times Two* **(14 points)**

A client and server form a secure connection with Diffie-Hellman TLS. The client uses Diffie-Hellman secret c_1, and the server uses secret s_1. After the first connection ends, Mallory, a MITM attacker, compromises s_1.

Next, the same client and server form a second connection with Diffie-Hellman TLS. For this connection, the client uses Diffie-Hellman secret c_2, and the server uses secret s_2.

Mallory wants to impersonate the server in the second connection (i.e. Mallory wants to be able to send her own messages to the client in the second connection).

Q9.1 (3 points) During the second handshake, the server sends $g^{s_2} \bmod p$ to the client, along with a signature on this value.

Mallory intercepts this message and replaces it, sending the replaced message to the client. What should the replaced message be?

Your answer can contain any values that Mallory knows.

Q9.2 (3 points) What is the shared premaster secret that the client derives?

Q9.3 (3 points) After executing this attack, what can Mallory do in the second TLS connection? Select all that apply.

- ☐ Read messages sent by the client
- ☐ Send messages to the server
- ☐ Read messages sent by the server
- ☐ None of the above

Q9.4 (5 points) Suppose the server acts as a certificate authority for EvanBot. (In other words, the server can use their secret key to sign EvanBot's public keys.)

The client wants to form a TLS connection with EvanBot. Can Mallory use this attack to cause EvanBot to derive a shared secret that Mallory knows?

- ○ Yes
- ○ No

Briefly justify your answer.

Q10 *Networking: Don't Need Security* (22 points)

EvanBot's DNS resolver has the following records cached:

```
Record 1:   toon.cs161.org      A         192.5.6.7
Record 2:   org.                NS        a.org-servers.net
Record 3:   a.org-servers.net   A         192.5.6.10
Record 4:   org.                DNSKEY    {PK of a.org-servers.net}
```

Each subpart is independent.

Q10.1 (3 points) How many DNS requests are needed to learn the IP address of `toon.cs161.org`?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

Q10.2 (3 points) How many DNS requests are needed to learn the IP address of `evanbot.cs161.org`?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

For the next four subparts, assume DNSSEC is enabled.

Q10.3 (3 points) How many DNS requests are needed to validate the public key in Record 4?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

Q10.4 (3 points) How many additional DNS records are needed to validate the public key in Record 4?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

Q10.5 (3 points) How many DNS requests are needed to validate the answer in Record 1?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

Q10.6 (3 points) How many additional DNS records are needed to validate the answer in Record 1?

○ 0 ○ 1 ○ 2 ○ 3 ○ More than 3

For the rest of this question, **assume DNSSEC is disabled**.

Q10.7 (4 points) Eve is an on-path attacker. Eve tricks EvanBot into loading `assets.cs161.org`.

Which of the following domains could Eve poison in EvanBot's DNS cache? Select all that apply.

☐ `assets.cs161.org`
☐ `a.org-servers.net`
☐ `www.wikipedia.org`
☐ None of the above
☐ `www.google.com`

Q11 *(OPTIONAL) A+ Question: Claw-Free Constructions* (0 points)

This question is not worth points. It can only affect your course grade if you have a high A and might receive an A+. We strongly recommend completing the rest of the exam before attempting this question due to the relatively high difficulty. Ask your proctor for additional paper if you need more space to write.

Ryan gets bored of SHA256 and decides to use the cryptographic primitive known as **claw-free permutations** to build a new hash function. A claw-free permutation is a pair of functions f_0 and f_1 such that finding any pair of inputs x, y such that $f_0(x) = f_1(y)$ is computationally difficult. Such a pair is called a **claw**. This is similar to the idea of collision-resistant hash functions – a claw-free permutation is *pair* of functions that are collision-resistant with each other, whereas a hash function is a single function that is collision resistant with itself.

Note that f_0 and f_1 take in exactly n bits and output exactly n bits.

Q11.1 (0 points) A common way of constructing these claw-free permutations is by using the malleable (homomorphic) properties of RSA.

Construct a claw-free permutation (f_0, f_1) and prove its security based on RSA.

HINT: Define $f_0(x) = x^e \mod N$ and $f_1(x) = yx^e \mod N$ for some fixed e, N, y and show that finding a claw in f_0, f_1 implies the ability to reverse RSA encryption if y is the RSA ciphertext. Since y is arbitrarily chosen, this implies finding a claw in f_0, f_1 breaks RSA.

Q11.2 (0 points) Assuming the existence of a claw-free permutation, design a collision-resistant, one-way, deterministic hash function H that takes in **arbitrary-length** inputs and outputs **exactly** n bits.

Q11.3 (0 points) Show that any an adversary that can efficiently find a collision in H can efficiently find a claw in f_0, f_1, i.e., prove the security of your construction assuming the security of the claw-free permutation.

Post-Exam Activity: Botgram

(ungraded, just for fun) Help EvanBot craft the perfect Botgram post!