ITIS 6200/8200 Principles of Information Security and Privacy

Final Exam

If the questions ask for explanations, please **briefly** explain the **key reasons**.

**Notation**: here are some of techniques and notations that may be useful.

* AES: Enc(K, M) and Dec(K, C) where K, M, and C denote the key, plaintext, and ciphertext
* HMAC: HMAC(K, V) where K, V are the inputs to the HMAC function
* Hash: Hash(M) where M is the input to the hash function
* RSA: Ekey(M), and Dkey(C). The key here could be pub\_[name], pri\_[name] with the corresponding names. For example, Epub\_alice(M).
* c1 || c2: String concatenation of c1 and c2.

**Question 1**. **Cryptography Fundamentals (10 points)**

Q 1.1: Fill the following table. The first row shows security properties, and first column shows cryptographic techniques. Put a mark in a cell if its corresponding technique is designed for the corresponding security property.

|  |  |  |
| --- | --- | --- |
|  | Confidentiality | Integrity |
| AES-CBC |  |  |
| MAC |  |  |
| RSA encryption |  |  |
| RSA signature |  |  |

Q 1.2: Fill the following table. The first row shows cryptographic techniques, and the first column shows a few mechanisms/building blocks for these techniques. Put a mark in a cell if its corresponding technique uses the corresponding mechanism in the table. (Note that multiple mechanisms can be used one mechanism, i.e., one column may have more than one )

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | AES-CBC | MAC | RSA Encryption | RSA Signature |
| Hash |  |  |  |  |
| PRNG |  |  |  |  |
| Modulo operation |  |  |  |  |
| XOR |  |  |  |  |

**Question 2**. **Encryption (15 points)**

Alice wants to send a large size message M to Bob. Their public keys are known to each other.

Q 2.1: What is the main advantage of using symmetric key encryption like AES? What is the main advantage of using Asymmetric key encryption like RSA? (be brief in your answer)

Ans: AES: performance. RSA: no need to share a secret.

Q 2.2: Alice wants to use AES to encrypt M. To do that, Alice first needs to share a symmetric key K with Bob. How can Alice share K with Bob with both confidentiality and integrity? Write down the formula for sending K.

Ans: Epri-Alice(Epub-Bob(K))

Q 2.3: What can go wrong (with the scheme from Q 2.2) if a powerful attacker Mallory can intercept messages passing between Alice and Bob?

Ans: Mallory can drop the symmetric key sent from Alice to Bob, and replace it with another symmetric key generated by Mallory: Epri-Mallory(Epub-Mallory(KMallory)). Mallory can then send modified messages to Bob using KMallory.

**Question 3**. **Confidentiality and Integrity (15 points)**

Alice is sending message M to Bob in the following way:

Ciphertext c = c1 || c2 where c1 = Enc(K1, M) and c2 = MAC(K2, M)

Where K1 and K2 are two secret keys known only to Alice and Bob.

Q 3.1: Does the scheme provide both confidentiality and integrity. Briefly explain why.

Ans: No. No confidentiality for c2.

Q 3.2: Design a scheme that provides both confidentiality and integrity.

Ans: No. c1 || c2 where c1 = Enc(K1, M) and c2=MAC(K2, c1).

Q 3.3: Assume AEC-CBC for Enc(K1, M), but the same IV is reused for encryption. Does the scheme you design provide IND-CPA? If not. Design the inputs that can break INC-CPA.

Ans: No. Eve can send Bob two inputs in1 and in2. Bob randomly chooses one of them to encrypt and send ciphertext c back to Alice. Then Eve sends in1 Bob to encrypt. If the result is the same as c, then we can determine which the message encrypted was in1, otherwise in2.

**Question 4**. **Hash (10 points)**

The ecommerce website *Zmazon.com* decides to store all its users’ username and password as Hash(**username**) **||** Hash(**password**): for every pair of username and password, the website calculates the hash value of **password**, concatenates it with the **username**, and stores the result in a database. Every time when a user logs in, the website performs the same operations with the login information, and finds if the value entry in the database.

Assume Hash() is the cryptographic hash function SHA-256. MAC is a secure MAC. And *Zmazon.com*’s secret key k is confidential. And usernames are all alphabetic (no numbers or special characters) and password can have numbers, special characters, and letters.

Q 4.1: Assume an attacker is able to modify entries in the database. Can the attacker modify a record without being detected? If yes, how?

Ans: Yes. There are many ways to do this:

1. Attacker can generate valid hashes for any username and password of choice.
2. Length extension attack on Hash(**username**) to generate another valid hash
3. Change it toHash(**password**) || Hash(**username**)

Q 4.2: Assume we store MAC(k, **username**) || MAC(k, **password**), can the attacker modify a record without being detected? If yes, how? If no, why?

Ans: Yes. Change it toMAC(k, **password**) || MAC(k, **username**)**.** Note thatthe attackers **cannot** generate MAC, since they don’t have the key k.

Q 4.3: Design an approach that we can prevent / detect changes by the attacker.

Ans: there are different ways, for example:

1. MAC(k, username || password)
2. Username || MAC(k, username || password)

**Question 5. Access Control (10 points)**

Assume that CCI has a file **F** that stores all students’ information. Only the SIS department chair Alice has read and write access to **F**. A powerful attack Mallory was able to infiltrate Alice’s laptop with a Trojan Horse program. The Trojan Horse works in the following steps:

1. creates a new file **G**
2. give Alice write access to **G**
3. give Mallory read access to **G**
4. copy the contents from **F** to **G**

Q 4.1: Write the access control lists for **F** and **G**.

Ans:

F -> [Alice: rw; Mallory: ]

G -> [Alice: w; Mallory: r]

Q 4.2: How can MAC (or the BLP model) help with the situation? Design the security levels for each subject and object.

Ans: The levels: Public Confidential

G: Confidential

F: Confidential

Mallory: public

Alice: Confidential

The information contained in F cannot leak to Mallory. Note that the copy operation should keep the same security level for the object being read.

Grading:

If the students don’t have (at least) two security levels. No marks.

If the students don’t say F is at confidential (or secret) level, deduce 3 points.

**Question 6. Web Security (20 points)**

A website uses a Go handler is processing HTTP requests to URL **https://vulnerable.com/hello** that takes an argument of name. The website is vulnerable to XSS attacks.

**func handleSayHello(w http.ResponseWriter, r \*http.Request) {**

**name := r.URL.Query()["name"][0]**

**fmt.Fprintf(w, "<html><body>Hello %s!</body></html>", name)**

**}**

Q 4.1: Design a GET request that can launch an attack to *Zmazon.com* to reset password to **abc**. The function for resetting password is: http://Zmazon.com/resetPassword?password=abc.

Ans: https://vulnerable.com/hello?name=

<script>fetch(“http://Zmazon.com/resetPassword?password=abc”)</script>

Q 4.2: Assume *Zmazon.com* does not check user inputs. What cookies would be attached to the request to *Zmazon.com*? Give their values for the fields of **Domain** and **Path**.Whatif the cookie’s **Secure** field is sent to True?

Ans: Domain: Zmazon.com

Path: /

If Secure field is True, No cookies will be sent, due to the **http** of *Zmazon.com*

Q 4.3: To log into *Zmazon.com*, you must go through authentication on *login.Zmazon.com*, and set a cookie to keep track of your authenticated status. The session token cookie should be secure against network attackers, and should get sent to as many pages on *Zmazon.com* as possible. What values should be set for the fields of **Domain**, **Path**, **Secure**, and **HttpOnly** in the cookies?

Ans: Domain: Zmazon.com

Path: /

Secure: True

HttpOnly: True

**Question 7. Network Security (15 points)**



***Standard TCP handshake***

The new ITIS6200 course decides to use a server to send messages to its students, but the student only responds with ACKs and never sends any messages after the initial handshake. They design an ITIS-TCP protocol which provides TCP’s properties for communications from the server to the student, but not for communications from the student to the server. This is accomplished using a modified version of the standard three step handshake.

Q 7.1: What information we **no longer need** in the 3-way handshake of the ITIS-TCP protocol?

Ans: No sequence number needed for packets sent from the student to the server.

No ACK number needed for packets sent from the server to the student.

Q 7.2: Consider an on-path attacker Eve who can observe the traffic but cannot modify it. Can Eve hijack the TCP connection between the students and the server? Are things easier for Eve compare with the normal TCP connections? Does Eve get extra power in attacking?

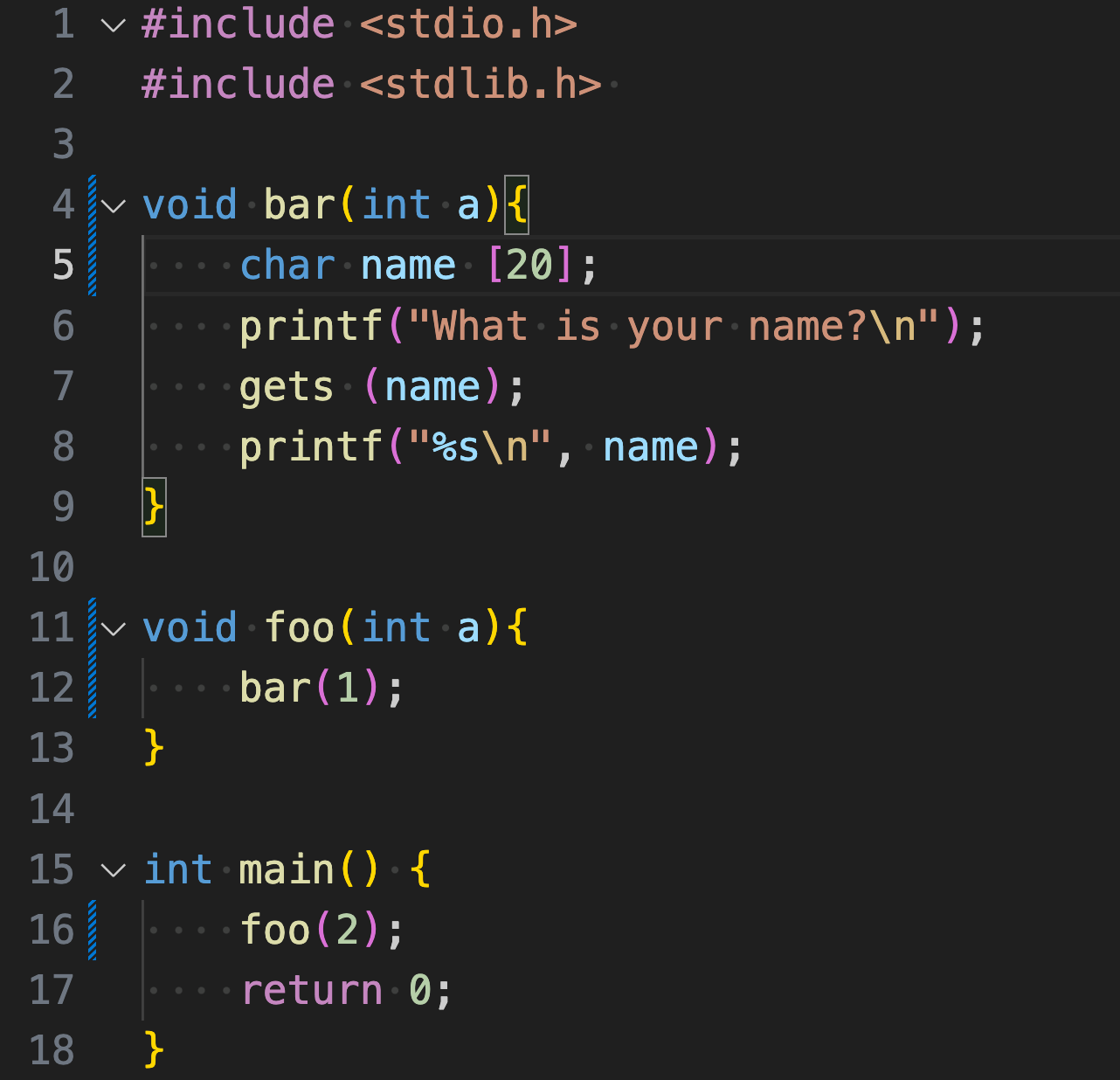
Yes. Eve can spoof packets in both directions, using the server’s IP and the client’s IP. The recipient cannot distinguish it from the legitimate messages.

Q 7.3: Consider an off-path attacker David who cannot observe and modify the traffic. Can David do anything malicious to the connection? Does the attack get easier for David compare with the normal TCP connections? Does Eve get extra power in attacking?

Yes. David can easily terminate the connection by sending RST packets to the server.

**Question 8. Memory Vulnerability (15 points)**

Consider the following vulnerable C code. Assume you are on a little-endian 32-bit x86 system and no memory safety defenses are enabled.

****

Q 5.1: Assume that execution has reached line 6. Fill in the following stack diagram. Assume that each row represents 4 bytes. Note that arrays are filled from lower addresses to higher addresses and are zero-indexed.

***Stack***

|  |
| --- |
| RIP of **foo** |
| SFP of **foo** |
| 1 |
| RIP of **bar** |
| SFP of **bar** |
| name[16]-name[19] |
| name[12]-name[15] |
| name[8]-name[11] |
| name[4]-name[7] |
| name[0]-name[3] |

Q 5.2: Assume that the address of the RIP of **foo** is 0x12345678.

Construct an input to gets that would cause the program to execute malicious shellcode. You may reference SHELLCODE as a 16-byte shellcode.