CSCI 191T – Computer Security

Assignment 4: XSS, SecretKeyEncryption and One-Way Hash Functions

Due: 12/4/2022 11:59 PM

Instructions:

There are two sections in this assignment. For Section 1, answer all questions completely. For Section 2, execute the code (using files from Assignment4_Files.zip) and answer all questions.

The final submission is a single report (PDF file). You need to submit a detailed report, with:

Section 1 - Detailed answers and observations.

Section 2 - Screenshots, to describe what you have done and what you have observed. You also need to provide explanation to the observations that are interesting or surprising. Also list the important code snippets followed by explanation. **Simply attaching code or screenshot without any explanation will not receive credits.**

Section 1

85points

1. (5 Points) To defeat XSS attacks, a developer decides to implement filtering on the browser side. Basically, the developer plans to add JavaScript code on each page, so before data is sent to the server, it filters out any JavaScript code contained inside the data. Let's assume that the filtering logic can be made perfect. Can this approach prevent XSS attacks?

2. (5 Points) Why is the CSP (Content Security Policy) effective in defeating the Cross-Site Scripting attack? What is the downside of this approach (on the developer's side)?

3. (5 Points) A message of 59 bytes is encrypted using AES with the CBC mode. The padding scheme is PKCS#7. Describe what the padding data will be. What if this message has 64 bytes?

4. (5 Points) Alice encrypts a message using AES, and sends the ciphertext to Bob. Unfortunately, during the transmission, the 2nd bit of the third block in the ciphertext is corrupted. How much of the plaintext can Bob still recover if the mode of encryption is one of the followings: OFB, or CTR?

5. (5 Points) Why is the hash function f(x) = x mod 10000 not a good one-way hash function?

6. (5 Points) Currently, the salt used in the shadow file is saved in the file in plaintext. Isn't it better not to save the salt in plaintext? Please explain.

7. (5 Points) A developer writes the following in a post: “I am writing a login for a forum, and I would like to hash the password at the client side in JavaScript before sending it to the server. If the hash matches with the one stored on the server, the user will be allowed to log in.” The developer believes that by sending the hash of the password, instead of sending the password directly, can improve the security. Do you agree or not, why?

8. (5 Points) In Linux, the password hash is produced by applying a hash function for many rounds (e.g., 5000 rounds for SHA-512). This seems to waste time, why does Linux do this?

9. (5 Points) Given h = Sha256(K || M), where K is a secret and "||" means concatenation (no padding is involved in calculating h). Please describe how one can calculate Sha256(K || X) for a different message X without knowing K.

10. (5 Points) In the length extension attack, do we need to know the length of the key?

11. (10 Points) The following message K:M is fed into SHA256. (1) What will be used as the padding? (2) Given hash(K:M), we need to calculate hash(K:M:N) without knowing the value K. The string N should contain the following message “extra content”. Please describe the actual content of N.
    K = abcd9313x
    M = 12345678901234567890123456789012345678 90
    K:M = abcd9313x:12345678901234567890123456789012345678 90

12. (10 Points) Charlie has arranged a blind date for Alice and Bob, who are both cryptographers, and they do not know each other before. Charlie also gave Alice and Bob a secret number K (nobody else knows K). Bob wants to make sure that the person he is dating is Alice, not somebody else. Please describe how Bob can ask Alice to securely prove that she is Alice (Alice will not reveal the secret number K to anybody). (HINT: HMAC)

13. (5 Points) Explain the Merkle-Damgard construction method for hash algorithms, with the help of a simple diagram.

14. (10 Points) If Bob happens to find two different messages M1 and M2 (each has 64 bytes), such that SHA256(M1) = SHA256(M2). Can you find another pair M3 and M4, such that SHA256(M3) = SHA256(M4)?

Section 2

15 Points

1. (10Points) The file pic_original.bmp (in Assignment4_Files.zip) needs to be encrypted, so people without the encryption keys cannot know what is in the picture.

Encrypt the file using the ECB (Electronic Code Book) and CBC (Cipher Block Chaining) modes, and then do the following:

a. Let us treat the encrypted picture as a picture and use a picture viewing software to display it. However, For the .bmp file, the first 54 bytes contain the header information about the picture, we have to set it correctly, so the encrypted file can be treated as a legitimate .bmp file. We will replace the header of the encrypted picture with that of the original picture. We can use the bless hex editor tool (already installed on our VM) to directly modify binary files. We can also use the following commands to get the header from p1.bmp, the data from p2.bmp (from offset 55 to the end of the file), and then combine the header and data together into a new file.

```
$ head -c 54 p1.bmp > header
$ tail -c +55 p2.bmp > body
$ cat header body > new.bmp
```

b. Display the encrypted picture using a picture viewing program (we have installed an image viewer program called **eog** on our VM). Can you derive any useful information about the original picture from the encrypted picture? Explain your observations.

2. (5 Points) Select a picture of your choice, repeat the experiment above, and report your observations.