UNIVERSITY OF BIRMINGHAM

School of Computer Science

Second Year Undergraduate

06-30432

30432 LI Security and Networks

Main Summer Examinations 2022

[Answer all questions]

Note

Answer ALL questions. The paper will be marked out of 60, which will be rescaled to a mark out of 100.

Question 1

(a) Consider ElGamal encryption with $g = 7$, $p = 11$.

- Bob chooses $x = 3$ as his private key. What is his public key? Show your workings. **[2 marks]**
- Alice sends Bob the ciphertext $(9, 8)$, encrypted with Bob's public key. What is the corresponding plaintext? Show your workings. **[3 marks]**

(b) Bob gives instructions to buy or sell shares to his broker over the internet. When Bob gives an instructions, he sends two messages. The first message consists of the RSA-encryption with the broker's public key of a 128-bit key which is shared between Bob and the broker. The second message consists of the RSA-encryption with the broker's public key of the instruction. If an attacker manages to obtain the encrypted messages, is it possible for the attacker to send an instruction to the broker which is different from all previous instructions? Justify your answer. **[5 marks]**

(c) Assume Alice and the bank share a symmetric key. Alice encrypts "Pay Tom 1000 pounds" in AES-counter mode using this key, and signs the encrypted message with El-Gamal using her private key. The bank accepts this message if it can decrypt it, and the signature matches. If the attacker has obtained the encrypted message and the signature, is it possible for the attacker to change the message so that message is the encryption of "Pay Bob 9999 pounds" and moreover create a matching signature which the bank will accept? If this is possible, describe how the attacker can do this. If this is not possible, explain why. **[5 marks]**

Question 2

You are reviewing a web application written in PHP for security issues.

(a) One script contains the following code to send an email as part of the registration process, where the `$_POST` parameters are generated by Javascript in the user's browser (hint: `system()` is used to invoke a shell command, such as `mail` in the below example). Assume that the script runs as `root`.

```
$addr = $_POST["addr"];
$subject = $_POST["subject"];
system("mail $email -s $subject < /var/www/message.txt");
```

Which vulnerability exists in this script? Give example values for `$_POST["addr"]` and `$_POST["subject"]` to exploit this issue.

[6 marks]

(b) A second script has a detection mechanism against Cross Site Scripting (XSS) that is implemented through matching for `<script>` tags in a function `check_xss()`. This function returns `TRUE` if it detects XSS and is called as follows on an externally supplied string `$message`:

```
if(check_xss($message) === TRUE) {
  // Forbidden tag detected, error out and abort
  echo "Error: XSS attempt in " . $message;
  exit();
}
else {
  // ... continue rendering a page based on $message
}
```

Assuming that `check_xss()` catches all relevant tags and XSS vectors, does this XSS detection work? If no, briefly explain a way to bypass it. **[6 marks]**

(c) The web application uses a global counter which initialised to a random value when the web server starts and then increased by one for each website served as a token to protect against Cross-Site Request Forgery (CSRF). Briefly explain why this is insecure.

[3 marks]

Question 3

(a) Describe the relationship between *key freshness* and *forward secrecy*. Does every protocol that provides *forward secrecy* automatically imply *key freshness*? Does every protocol satisfying *key freshness* also automatically provide *forward secrecy*? Justify your answer. **[5 marks]**

(b) Consider the following protocol for the following questions:

$$\begin{array}{rcl} A & \rightarrow & B : \{N_A\}_{pk(B)} \\ B & \rightarrow & A : \{N_A, N_B, B\}_{pk(A)} \\ A & \rightarrow & B : \{N_B, A\}_{pk(B)} \\ A & \rightarrow & B : \{M\}_{\#(N_A, N_B)} \end{array}$$

where N_A and N_B are nonces, and $\#(N_A, N_B)$ is a symmetric key based on the hash of N_A and N_B, and $pk(A)$ is the public key of A.

(i) Does this protocol satisfy the property of *key freshness*? Justify your answer. **[2 marks]**

(ii) Does this protocol satisfy the property of *forward secrecy*? Justify your answer. **[2 marks]**

(iii) Is it possible for an attacker to learn M without knowing any of the private keys of A and B? If so, give an attack in Alice-Bob notation. If not, explain why. **[6 marks]**

Question 4

You review a C program that performs a password check:

```
1  int check_authentication(char *password) {
2      int authenticated = 0; // 0: not authenticated, else authenticated
3      char password_buffer[16];
4
5      strcpy(password_buffer, password);
6      password_buffer[15] = '\0'; // prevent long strings!
7      if(strlen(password_buffer) > 15)
8          return 0;
9
10     if(strcmp(password_buffer, "mahgnimrib") == 0)
11         authenticated = 1;
12
13     return authenticated;
14 }
```

(a) Assume that the program is compiled for x86 in 32-bit mode.

(i) Sketch the state of the stack *before* line 5 is executed. Clearly indicate where top and bottom of the stack are located. Assume that all variables are aligned at 4-byte boundaries.

(ii) Explain which vulnerability is present in this code?

(iii) Indicate which part of the stack has been changed *after* the `strcpy` on line 5 has been executed when the input password is 20 characters long.

[7 marks]

(b) For each of the following exploits, explain how you would craft an input to the function to achieve it. If possible, give a concrete example.

(i) Circumvent the password check. Your input should make the function return `1` without knowing the correct password.

(ii) Achieve an arbitrary code execution?

[5 marks]

(c) The author of the code intended to prevent this type of vulnerability using the code in lines 6–8. Explain why these checks do not achieve the intended purpose and explain how you would need to change the code instead. **[3 marks]**