A35067

Calculators may be used in this examination provided they are not capable of being used to store alphabetical information other than hexadecimal numbers



UNIVERSITY OF BIRMINGHAM

School of Computer Science

Fourth Year Undergraduate/Postgraduate

06-34240

LM Secure Software and Hardware Systems

Main Summer Examinations 2025

Time allowed: 2 hours

[Answer all questions]

Note

Answer ALL questions.

Question 1

In your job at a software company you responsible for reviewing the source-code produced by your colleagues. Consider the following code snippet

```
 1. #include <stdio.h>
 2.
 3. void fun1(char *arg1, int len) {
 4.   char buffer[5];
 5.   sprintf(buffer, arg1);
 6.   if (len < sizeof(buffer)) {
 7.      printf("all good");
 8.   }
 9. }
10.
11. void main() {
12.    fun1("test",4);
13. }
```

(a) Draw a stack and populate it with the contents for `fun1(...)` before the execution of line 5. Make sure you indicate, the direction of growth, the registers ESP and EBP; and that you place all the local variables and other significant values on to the stack. **[10 marks]**

(b) What is the purpose of the registers ESP, EBP? Are these registers sufficient to enable correct execution flow between `main(...)` and `fun1(...)`? **[8 marks]**

(c) Explain how memory is allocated on the stack when local variables are loaded. You can use the given code above to explain your answer. **[5 marks]**

(d) What happens if line 12 is replaced with: `fun1("myweirdtest",4);`. Refer to the stack, and to the variables `len`, `buffer` and `arg1`. **[6 marks]**

(e) Assuming full control over how `fun1(...)` is called, and that the variable `buffer` is vulnerable to a stack overflow attack, describe a useful payload that an attacker would want to place in this buffer. How would this attack work? You do not need to write code here. **[6 marks]**

Question 2

(a) **Part 1:**

(i) Briefly describe how an operating system uses CPU operating modes and the Memory Management Unit (MMU) to enhance software execution security. **[8 marks]**

(ii) Describe what trap instructions are. Give an example of how these can be used by the operating system to facilitate concurrency on platforms with non-cooperative/preemptive multitasking support. **[7 marks]**

(b) **Part 2:**

(i) Embedded systems often only have a Memory Protection Unit (MPU), rather than an MMU. Briefly explain the main difference between the two and argue why an MPU is sufficient for many embedded systems. **[6 marks]**

(ii) Apart from the MPU, which other security features common on desktop systems are often missing in embedded systems? Name 3 examples and briefly explain why they are often not present on embedded systems. **[9 marks]**

Question 3

You are tasked to do a security review for the firmware on a newly developed embedded system: An "IoT-enabled" SSD, which allows you to retrieve files via bluetooth. As part of the review, you get access to the full source code, the tool chain to build the binary firmware, and the physical hardware running the binary firmware.

(a) As part of the specification for the security review, you read that this is a "Type-II" embedded system. What kind of operating system (if any) can you expect to be running on the device? How will this influence your security review and why? **[7 marks]**

(b) How would you implement a fuzzing campaign for this target? Describe the steps you would take to set up the campaign, and explain how you would confirm your findings. **[10 marks]**

(c) During manual code review, you find the following function:

```
1. /* Return 0 if password is correct; other values indicate a mismatch*/
2. int check_password(char *user_input, char *password, int len) {
3.   int i = 0;
4.   while (i++ < len) {
5.     if (user_input[i] != password[i]) {
6.       return -1;
7.     }
8.   }
9.   return 0;
10. }
```

For all invocations of this function, the only input controlled by a potential attacker is the content of `user_input`, while `len` is always guaranteed to match the string length of `password`.

However, this function still poses a security risk for the embedded system. What is the risk, and how would you exploit it as an attacker? **[10 marks]**

(d) Create a secure version of above function. Provide the source code for your fixed implementation, alongside of a brief explanation on how the risk is mitigated. **[8 marks]**.

This page intentionally left blank.

Do not complete the attendance slip, fill in the front of the answer book or turn over the question paper until you are told to do so

Important Reminders

- Coats/outwear should be placed in the designated area.
- Unauthorised materials (e.g. notes or Tippex) must be placed in the designated area.
- Check that you do not have any unauthorised materials with you (e.g. in your pockets, pencil case).
- Mobile phones and smart watches **must** be switched off and placed in the designated area or under your desk. They must not be left on your person or in your pockets.
- You are not permitted to use a mobile phone as a clock. If you have difficulty seeing a clock, please alert an Invigilator.
- You are not permitted to have writing on your hand, arm or other body part.
- Check that you do not have writing on your hand, arm or other body part – if you do, you must inform an Invigilator immediately
- Alert an Invigilator immediately if you find any unauthorised item upon you during the examination.

Any students found with non-permitted items upon their person during the examination, or who fail to comply with Examination rules may be subject to Student Conduct procedures.