A22652

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

Postgraduate Affiliate Computer Science
MSc Advanced Computer Science
Undergraduate Affiliate Computer Science/Software Engineering
Fourth Year – MSci Computer Science
Fourth Year – MEng Computer Science/Software Engineering
MSc Cyber Security
MSc Computer Security

06 20010

Secure Programming

Summer May/June Examinations 2016

Time allowed: 1 hour 30 minutes

[Answer ALL Questions]

1. **[Total: 10%]** In this task, we look at secure programming in general. At the beginning of the lecture, general patterns of secure programming and ways how to write secure code such as *default to deny* or *defense in depth* have been introduced. List at least 4 more patterns of secure programming and how to write secure code. **[10%]**

2. **[Total: 25%]** In this task we look at SQL injections.

(a) Describe what an SQL injection vulnerability is and give an example for it in any programming language of your choice. **[7%]**

(b) Name and explain two different countermeasures against SQL injection. **[8%]**

(c) To counter SQL injection, a programmer decides to remove all whitespace characters from untrusted input. Is it sufficient to counter SQL injection, or is SQL injection still possible when all whitespaces have been removed from untrusted input? **[5%]**

(d) What could you do (coding, configuration of your application and environment, architecture of your software application) to limit the impact of an SQL injection should there still be a SQL injection in your code? **[5%]**

3. [Total: 25%] In this task we look at timing based information leakage.

(a) Explain why code that accesses memory addresses based on a confidential value might leak information about this value by timing. **[8%]**

(b) What kind of general pattern can be used to replace small lookup-tables in code (which may leak information about the accessed values) with something else that executes in constant time? How do you (in general) apply the pattern? **[7%]**

(c) We would like to implement a function in C that returns 0 when the argument is positive or zero, 1 otherwise. Consider the following code and explain why it might not run in constant time. **[5%]**

```
int sign(int a) {
  if (a < 0) {
    return 1;
  } else {
    return 0;
  }
}
```

(d) Implement that function in C (or in another language of your choice) in constant execution time. When you chose to implement it in C, you may assume that int is 64 bit long. **[5%]**

4. [Total: 26%]

In some companies, a lot of small computations have to performed. For example in IT companies designing new chips, a lot of different configurations of the final chip layout are compiled and optimized and the best configuration is determined. Assume that you work for such a company and you have to design an application that runs on every office PC in the company. Engineers who design such chips upload work packages (an input file with the chip configuration that needs to be optimized) to a central server. Your app then downloads such a work package in the evening and runs the optimization program with the associated input over night. The final results are then uploaded to the server. Your platform is Linux.

(a) What would you suggest for the design of the application to make it more secure? Sketch a design for this application. **[11%]**

(b) How may you use chroot to improve the security of your application? **[5%]**

(c) How may an App Armor profile be used to provide additional security for this application. **[5%]**

(d) How may seccomp be used to enhance the security of this application. **[5%]**

5. **[Total: 14%]** You are working for a company that modifies cars. About 20 years ago, a previous employee of the company wrote a C program that takes a plan for a car and the description of a planned modification as input and calculates whether the car frame would still be safe enough from a safety point of view, or reinforcements are required. Now your company would like to include that program in your webshop so that customers may upload a sketch of their modifications and the webshop will determine whether your company can do the modification cheaply or only with expensive reinforcement of the car frame.

The code was probably never designed to handle untrusted inputs. You are concerned that someone might upload a bogus description of a proposed modification and exploit something in your webshop. There are several ways how you can find weaknesses in the code.

(a) How might a tool like CBMC help you to spot security problems in the code? **[7%]**

(b) How may a static source code scanner or compiler help you to find weaknesses in the code. **[7%]**