TU/e EINDHOVEN UNIVERSITY OF TECHNOLOGY

Surname and initials:

Student number:

Examination cover sheet

(to be completed by the examiner)

Course name: **Software Security** Course code: **2DMI20**

Date: **01-02-2023**

Start time: **9.00** End time: **12.00**

Number of pages: **17 + 3 Extra Pages**

Number of questions: **15**

Maximum number of points/distribution of points over questions:
128 + 5% * 128 * proportional performance in quizzes

Method of determining final grade: **Number of total points obtained * 10/128 (rounded)**

Answering style: **Open questions**

Exam inspection: **With the lecturer**

Other remarks: GOOD LUCK!

INSTRUCTIONS FOR STUDENTS AND INVIGILATORS
(to be indicated by examinor)

Write in black or blue. Pencil only allowed for drawings.

Permitted examination aids (to be supplied by students):

☐ Notebook
☑ Calculator
☑ Graphic calculator
☐ Lecture notes/book
☐ One A4 sheet of annotations
☐ Dictionaries:

☐ Other:

Important:

- examinees are only permitted to visit the toilets under supervision
- it is not permitted to leave the examination room within 15 minutes of the start and within the final 15 minutes of the examination, unless stated otherwise
- examination scripts (fully completed examination paper, stating name, student number, etc.) must always be handed in
- the house rules must be observed during the examination
- the instructions of subject experts and invigilators must be followed
- keep your work place as clean as possible: put pencil case and breadbox away, limit snacks and drinks
- examinees are not permitted to share examination aids or lend them to each other

During written examinations, the following actions will in any case be deemed to constitute fraud or attempted fraud:

- using another person's proof of identity/campus card (student identity card)
- having a mobile telephone or any other type of media-carrying device on your desk or in your clothes
- using, or attempting to use, unauthorized resources and aids, such as the internet, a mobile telephone, smartwatch, smart glasses etc.
- having any paper at hand other than that provided by TU/e, unless stated otherwise
- copying (in any form)
- visiting the toilet (or going outside) without permission or supervision

The final grade will be announced no later than fifteen working days after this examination took place. Final grades of first-year bachelor study components in Q4 will be announced within 5 working days. Final test grades of bachelor study components in the interim period will be announced no later than 5 working days before the 1st of September.

EINDHOVEN UNIVERSITY OF TECHNOLOGY
Department of Mathematics and Computer Science

Exam Software Security (2DMI20)
on Wednesday, February 1st, 2023, 09.00 – 12.00.

1. Consider the 7+1 Kingdoms of Software Vulnerabilities as given by McGraw.
   (a) [7 pt.] Name all kingdoms together with a brief explanation for each of them.
   (b) [4 pt.] For four of the kingdoms, name concrete vulnerabilities/attacks that were covered in the lecture.

2. In the lecture, we have seen several mechanisms for finding programming errors in software.
   (a) [5 pt.] Is it possible that one day we will have an automatic way to find all software bugs – and only those – in any given source code automatically? Provide reasons for your answer.

3. C/C++ programs are very susceptible to software errors related to memory corruption, which in turn may often lead to security vulnerabilities.

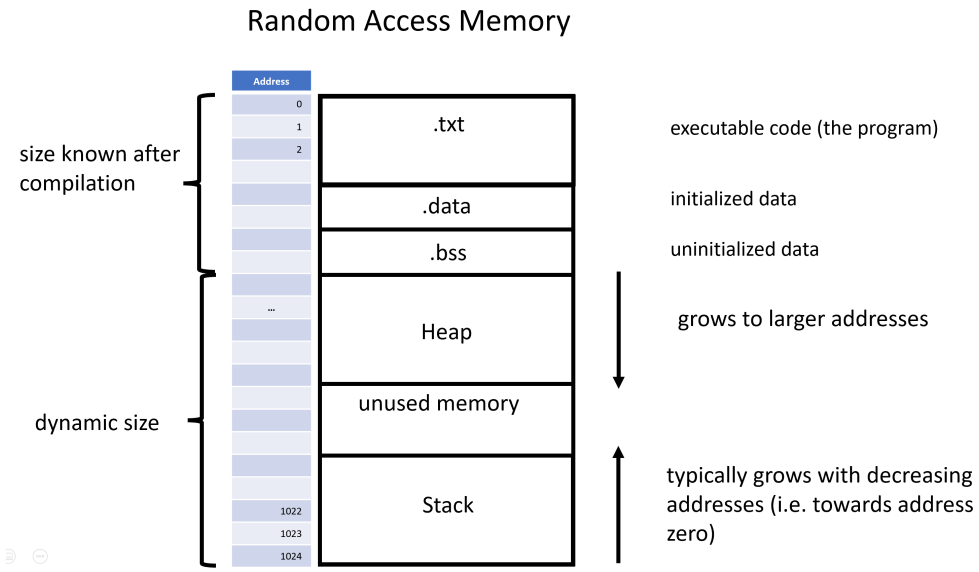
   (a) [4 pt.] Explain why many developers still use these programming languages. What are concrete benefits that C/C++ offer?

4. Von-Neumann Cycle

   (a) [4 pt.] What are the phases of the Von-Neumann Cycle?

5. Buffer overflow (in the narrower sense)

   (a) [8 pt.] Explain in detail and in a step-by-step fashion how a basic buffer overflow attack on the stack works. Assume we use a standard stack that grows towards lower memory addresses.



Extra Space for Question 5.

6. Memory Corruption Vulnerabilities

(a) [4 pt.] Identify the memory corruption vulnerability/weakness in the following C source code and explain it. What is the name of the vulnerability?

```
…
char input[5];
char input[1]="1";
char input[2]="2";
char input[3]="3";
char input[4]="4";
char input[5]="5";
…
```

(b) [4 pt.] Identify the memory corruption vulnerabilities/weakness in the following C source code and explain it. What is the name of the vulnerability?

```
…
int printCharacterInString(int maxLength, int position, char *string)
{

    //check that max Length not exceeded
    if( position >= maxLength)
    {
        printf("Error!");
    }
    else
    {
        printf("The character is %c", char[position]);
    }
}
…
```

(c) [4 pt.] Identify two memory corruption vulnerabilities/weaknesses in the following C source code and explain them. What are the names of the vulnerabilities?

```
...
//Variables
int *result = (int*) malloc(100*sizeof(int));              //holds result
int ConditionOne, ConditionTwo, GlobalCondition;    //indicate if important conditions are fulfilled
...
//Logic1
if (ConditionOne) { ConditionTwo=true; free(result); }
else { ConditionTwo=false; }
//Logic2
if (ConditionTwo && GlobalCondition) { print(result); free(result); }
...
```

Extra Space for Question 6.

7. Protection Mechanisms

   (a) [4 pt.] Name and describe four countermeasures to buffer overflow attacks?

   (a) [4 pt.] Briefly describe techniques to circumvent these countermeasures?

8. F and Fx Measure

   (a) [3 pt.] Assume we have two code analyzers A1 and A2 that both check if a given program is free of double-free-bugs: A1 has precision of 65% and recall of 85% whereas A2 has a higher precision of 70% but lower recall of 80%. Which of the analyzers has a better F measure? Compute both F measures and compare!

   (b) [3 pt.] Now, assume we value recall twice as high as precision. Which of the analyzers has a better Fx measure? Compute both Fx measures and compare!

9. Fuzzing

   (a) [5 pt.] Name the five algorithms that can found in fuzzers and briefly describe what they do.

   (b) [2+1 pt.] What is the difference between a model-based fuzzer and a mutation-based fuzzer? Which of the five algorithms present in most fuzzers can be either model-based or mutation-based?

   (c) [2 pt.] Explain the Fuzz Configuration Problem.

   (d) [2 pt.] Explain the Fuzzer Taming Problem.

10. Safe Programming Languages

   (a) [2 pt.] Explain Memory-Safety.

   (b) [2 pt.] Explain Type-Safety.

   (c) [2 pt.] What is the difference between spatial and temporal safety? For each of the two, name a memory corruption vulnerability that exemplifies it.

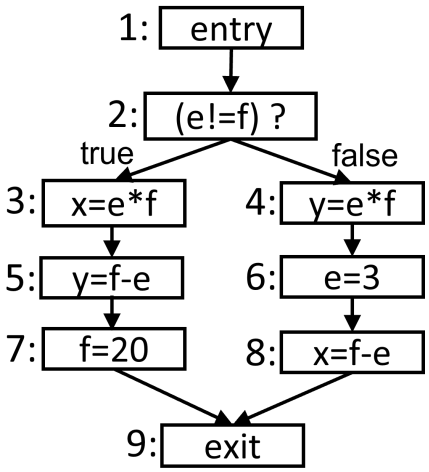
11. Static Code Analysis

(a) [2 pt] Explain why data flow analysis, as covered in the lecture, always terminates.

(b) [4+6 pt.] Generate the Control Flow Graph from the following source code and perform a Reaching Definitions Analysis.

```
x = 2;
y = 3;
z = 5;
while (z != 4)
{
        if (x == 3) { y = 0;} else { y = 1; };
        while (y*z != 0)
        {
                y=y-1;
        }
}
```

(c) [6 pt.] Perform a Very Busy Analysis on the following Control Flow Graph.



Extra Space for Question 11.

Extra Space for Question 11.

12. Web Security

(a) [2 pt.] What is the purpose of an SQL Injection attack?

(b) [4 pt.] Assume we have a website that loads contents dynamically from a database table Accounts. To login, the user has to provide username and password. Launch an SQL Injection attack that destroys the entire table!

```
<?php
$servername = "localhost";
$username = "username";
$password = "password";
// Create connection
$conn = new mysqli($servername, $username, $password);
...
$valid=mysql_query("SELECT * from Accounts WHERE
        (UserID='$usernameField' AND Password='$passwordField');");
...
$conn->close();
?>
<!DOCTYPE html>
...
<form action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>" method="post">
<input type="text" name="username" class="form-control" value="<?php echo $usernameField; ?>">
...
</html>
```

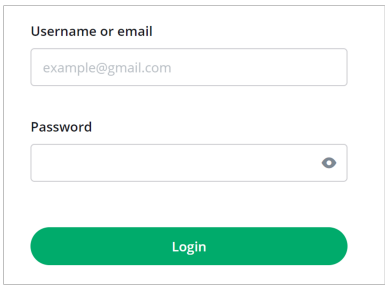


Table Accounts in Database

UserID	Password	Balance
Alice	...	...
...	...	...

(c) [4 pt.] Explain what prepared SQL statements are and how they can be applied to avoid SQL injection attacks.

13. Web Security

    (a) [1+4 pt.] What is the purpose of a Cross-Site Request Forgery attack? Explain how it works.

14. Web Security

    (a) [1+4 pt.] What is the purpose of a Cross-Site Scripting attack? Explain how it works.

15. RSA Fault Injection Attack on RSA-FDH

(a) [9 pt.] Assume Alice draws two primes $p = 11$, $q = 17$ and computes the public key $pk = (e, n)$ where $n = 11 \cdot 17$ and $e = 3$. Suppose Alice wants to sign message $m = 7$ and assume that $h(m) = 2$. Compute a valid signature $s = (h(m))^d \bmod n$ using the Chinese Remainder Theorem (CRT). Explicitly use the Extended Euclidean Algorithm (EEA) and the Square and Multiply (SQMUL) algorithm whenever possible and show all steps in the computations.

(b) [1 pt.] Verify that the signature is correct. It is not necessary to explicitly use the EEA and SQMUL algorithm for your computations.

(c) [4 pt] Assume an attacker stresses the computing device such that in the computation of sp there will be an error resulting in a distinct value $sp' = 9 \neq sp$. Show how to factor n using this information: Compute s' by applying the CRT to sq and sp'. Compute $h' = (s')^e \bmod n$. Finally, compute the factors of n via an appropriate application of the gcd algorithm. It is not necessary to explicitly use the EEA and SQMUL algorithm for your computations.

Extra Page 1

Extra Page 2

Extra Page 3