



**Mittuniversitetet**  
MID SWEDEN UNIVERSITY

Final exam  
DT145G Computer Security

Daniel Bosk  
`daniel.bosk@miun.se`  
Phone: 060-148709

2014-10-28

## Instructions

Carefully read the questions before you start answering them. Note the time limit of the exam and plan your answers accordingly. Only answer the question, do not write about subjects remotely related to the question.

Write your answers on separate sheets, not on the exam paper. Only write on one side of the sheets. Start each question on a new sheet. Do not forget to *motivate your answers*.

Make sure you write your answers clearly, if I cannot read an answer the answer will be awarded no points – even if the answer is correct. The questions are *not* sorted by difficulty.

**Time** 5 hours.

**Aids** Dictionary.

**Maximum points** 29

**Questions** 5

## Preliminary grades

The following grading criteria applies: E  $\geq$  50%, D  $\geq$  60%, C  $\geq$  70%, B  $\geq$  80%, A  $\geq$  90%.

## Questions

The questions are given below. They are not given in any particular order.

1. Explain the following terms:
  - (1p) (a) Confidentiality
  - (1p) (b) Integrity
  - (1p) (c) Availability
  - (1p) (d) Accountability
  - (1p) (e) Non-Repudiation
2. A user wishes to provide confidentiality to a file.
  - (3p) (a) She can accomplish this through mechanisms provided in the operating system. Explain how this works and what are the limits.
  - (3p) (b) She can also accomplish this through purely cryptographic mechanisms. Explain how this works and what are the limits.
3. The University password composition policy is as follows<sup>1</sup>: The password must be at least eight (8) characters long, further, the first eight characters must contain two uppercase and two lowercase letters as well as two numbers.
  - (3p) (a) Compute the information gained by an attacker from knowing this policy.
  - (3p) (b) Compare this to the information gained if the policy would have been “the password must be at least 16 characters long”.
4. The apps in the Google Play store or Apple’s AppStore are provided with some DRM, e.g. you cannot copy a paid-for app from one phone to another etc. There are also other things you are not allowed to do with these smartphones, e.g. installing whatever software you like, modifying the software in any way you like, etc. (However, many know of the terms “jail-breaking” or “rooting” their phones, which bypasses these mechanisms on the phones.)
  - (3p) (a) Explain how this type of protection system works, including the fundamental assumptions needed for it to fulfill its purpose.
  - (3p) (b) Explain how the above system can be broken and whether it could be fully solved or not.
5. Look at the C code in Listing 1 on the next page.
  - (3p) (a) Identify all vulnerabilities in that code and motivate by stating how they can be exploited.
  - (3p) (b) Suggest improvements to remedy these vulnerabilities, you must motivate why they work.

---

<sup>1</sup>It’s slightly more advanced than this, but it’s simplified here for reasons of convenience. Also, the effect might actually be the same anyway, depending on if users actually use the extension or not.

```

1 #include <stdio.h>
2
3 int
4 get_some_input( void )
5 {
6     char buffer[128];
7
8     printf( "Please enter the key: " );
9     scanf( "%s", buffer );
10
11     /* process input */
12
13     return 0;
14 }
15
16 void
17 make_full_name( char *dst, int dstlen,
18                const char *src, int srclen,
19                int maxsize )
20 {
21     if ( dstlen + srclen + 1 >= maxsize )
22         return -1;
23
24     strncat( dst, " ", 1 );
25     return strncat( dst, src, srclen );
26 }
27
28 int
29 main( int argc, char **argv )
30 {
31     char first[256];
32     char last[256];
33
34     printf( "Please enter your first name: " );
35     scanf( "%s", first );
36     printf( "Please enter you last name: " );
37     scanf( "%s", last );
38
39     make_full_name( first, strlen( first ),
40                   last, strlen( last ), 256 );
41
42     if ( get_some_input() < 0 )
43         return -1;
44
45     return 0;
46 }

```

Listing 1: Some vulnerable C code.