# TENTTI 12.1.2015 TURVALLINEN OHJELMOINTI, TIE-30600

Marko Helenius

**Tentissä ei saa käyttää oheismateriaalia eikä laskinta. Vastaa selkeästi.**

## 1. (6 pistettä)

Tarkastele oheista ohjelmaa, joka lukee käyttäjän syötettä ja tallentaa syötteen rivi kerrallaan tiedostoon.

Vasemmassa reunassa on ohjelmaan viittaamisen helpottamiseksi rivinumerot. Missä kohdassa ohjelmaa on haavoittuvuuksia, mitä nämä haavoittuvuudet ovat, mitä niistä voi seurata ja miksi? Mitä muuta parannettavaa ohjelmassa on turvallisuuden näkökulmasta, kun otetaan huomioon ohjelman tarkoitus?

```
 1  #include <string.h>
 2  #include <stdio.h>
 3
 4  int main(int argv, char **argc)
 5  {
 6        FILE *pFile;
 7        char *path = NULL;
 8        char *filename = NULL;
 9        char text[128];
10        unsigned int row=0;
11        if (argv > 1) //Jos annettiin vähintään yksi komentoriviparametri
12        {
13           char *environ = malloc(strlen(argc[1])+6);
14           filename = malloc(strlen(argc[1])+1); //Otetaan tiedoston nimi komentoriviparametrista.
15           strcpy(filename, argc[1]);
16           strcpy(environ, "TEXT="); //Laitetaan tiedoston nimi uudeksi oletukseksi.
17           strcat(environ, argc[1]);
18           putenv(environ);
19        }
20        else
21           filename = getenv("TEXT"); //TEXT-ympäristömuuttujalla voi asettaa tiedoston nimen.
22        if (filename == 0)
23           filename = "Text.txt";
24        printf(filename);
25        printf("\n");
26        if (argv == 3) //Jos annettiin kaksi komentoriviparametria.
27           pFile = fopen (filename,"w"); //Vanha tiedoston sisältö ylikirjoitetaan.
28        Else
29           pFile = fopen (filename,"a"); //Lisätään tiedostoon, jos se on olemassa.
30        do{
31          row++;
32          printf("%i:",row);  //Tulostetaan rivinumero
33          fgets (text,128,stdin); //Luetaan korkeintaan 128 merkkiä (sisältää NULL-merkin)
34          if (text[0]!='.')
35             fputs(text,pFile); //Kirjoitetaan syötetty tekstirivi tiedostoon
36        } while (text[0]!='.'); //Piste lopettaa syötön.
37        fputs(" ",pFile);
38        fclose(pFile);
39  }
```



Käännä!

**Vastaa seuraaviin kohtiin turvallisen ohjelmoinnin periaatteiden mukaisesti (6 pistettä/tehtävä):**

2. Oheisessa taulukossa on kuvattu tilapäistiedoston luomiseen liittyviä tietoturvanäkökulmia.
   a) Mitä tulee huomioida tilapäistiedostoja käytettäessä? Tuo esiin näkökohtia myös taulukon ulkopuolelta. (5 p)
   b) Mitä taulukon funktiota tai funktioita suosittelet mieluiten käytettäväksi ja miksi? (1 p)

| | `tmpnam` (C) | `tmpnam_s` (Annex K) | `tmpfile` (C) | `tmpfile_s` (Annex K) | `mktemp` (POSIX) | `mkstemp` (POSIX) |
|---|---|---|---|---|---|---|
| **Unpredictable Name** | not portably | yes | not portably | Yes | not portably | not portably |
| **Unique Name** | Yes | yes | Yes | Yes | Yes | yes |
| **Atomic open** | No | no | Yes | Yes | No | yes |
| **Exclusive Access** | Possible | possible | No | if supported by OS | possible | if supported by OS |
| **Appropriate Permissions** | Possible | possible | No | if supported by OS | possible | not portably |
| **File Removed** | No | no | yes* | yes* | No | no |

* If the program terminates abnormally, this behavior is implementation-defined.

3. Mitä tulee ottaa huomioon salasanaan perustuvaa autentikointia toteutettaessa?

4. Mitä erilaisia keinoja on estää CSRF-haavoittuvuudet (Cross-Site Request Forgery)? Laita vastauksen alkuun kymmenen ranskalaista viivaa ja kirjoita kymmenen kohdan perusteella essee-vastaus.