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

2. OWASP TOP 10 –listalla on mm. seuraavat uhat. Mitä nämä tarkoittavat ja miten niiltä suojaudutaan

   2 A1 – Injection

   1 A3 – Sensitive Data Exposure

   0,5 A4 – XML External Entities (XXE)

3. Mitä keinoja on yhdistää tietoturvallisuus Scrum-menetelmään? Laita vastauksen alkuun kymmenen ranskalaista viivaa ja kirjoita kymmenen kohdan perusteella essee-vastaus.

   2

4. a) Piirrä kuva, josta selviää mahdollisimman hyvin, miten CSRF (Cross Site Request Forgery) toimii. 0,5

   b) Yksi keskeinen CSRF:n torjuntakeino on CSRF-token. Miten tämä toimii ja toteutetaan? 1

Käännä!

TENTTI 21.5.2019 TURVALLINEN OHJELMOINTI, TIE-30601

Marko Helenius, Tampereen yliopisto

Tentissä ei saa käyttää oheismateriaalia eikä laskinta. Vastaa selkeästi.

1. (6 pistettä)

Tarkastele oheista ohjelmaa. Vasemmassa reunassa on ohjelmaan viittaamisen helpottamiseksi rivinumerot. Missä kohdassa ohjelmaa on haavoittuvuuksia, mitä nämä haavoittuvuudet ovat, mitä niistä voi seurata ja miksi? Miten korjaisit haavoittuvuudet? Haavoittuvuudeksi tulee tehtävässä laskea myös sellaiset virheet, jotka johtavat viittaukseen virheelliselle muistialueelle tai jumittavat ohjelman.

```
 1: #include <stdio.h>
 2: #include <stdlib.h>
 3: #include <ctype.h>
 4:
 5: //Merkkijono muutetaan luvuksi
 6: int string_to_int(char *merkkijono)
 7: {
 8:   int tulos=0;
 9:
10:   //isdigit tarkistaa, että merkki on välillä '0'...'9'
11:   while (*merkkijono && isdigit(*merkkijono))
12:   {
13:      tulos *= 10; //Tulos kerrotaan kymmenellä
14:      tulos += *merkkijono-'0'; //Lisätään numero 0-9
15:      merkkijono++; //Siirrytään seuraavaan merkkiin
16:   }
17:   return tulos;
18: }
19:
20: char* lue_merkkijono(unsigned int koko)
21: {
22:   char merkki;
23:   int i;
24:   //Muistetaan varata tila myös NULL-merkille.
25:   char* palautus = malloc(sizeof(char)*(koko+1));
26:   for (i = 0; i<koko; i++)
27:      palautus[i] = getc(stdin); //Luetaan yksi merkki
28:   palautus[i] = NULL;
29:   return palautus;
30: }
31:
32: int main(int argc, char **argv)
33: {
34:   //Koko saadaan komentoriviparametrina
35:   unsigned int koko = string_to_int(argv[1]);
36:   printf("Anna %i merkkiä: ", koko);
37:   char* merkkijono = lue_merkkijono(koko);
38:   printf(merkkijono);
40: }
```


Käännä!