COMP8050 – Security for Software Systems

Assignment 2 (50%)

For this assignment, we will use the C program provided below, **assignment1.c** . Compile it with the usual command:

```
gcc -g -O0 -mpreferred-stack-boundary=2    -m32      -fno-
stack-protector -z execstack  -D_FORTIFY_SOURCE=0
assignment1.c –o assignment1.o
```

Also ensure that you have disabled ASLR:

```
sudo sysctl -w kernel.randomize_va_space=0
```

If you have not already done so in the labs, you may need to install the following packages first too:

```
sudo apt-get install libc6-dev libc6-dev-i386 gcc-multilib
```

Question 1 – [25 Marks]

question1.c:

```
1.  #include <stdlib.h>
2.  #include <unistd.h>
3.  #include <string.h>
4.  #include <stdio.h>
5.  #include <sys/types.h>
6.
7.  struct politician {
8.    int id;
9.    int votes;
10.   char *name;
11. };
12.
13. void cheater()
14. {
15.   printf("Election Rigged!!\n" );
16. }
17.
18. int main(int argc, char **argv)
19. {
20.   struct politician *p1, *p2;
21.
22.   p1 = malloc(sizeof(struct politician));
```

```
23.   p1->id = 0;
24.   p1->votes = 16000;
25.   p1->name = malloc(36);
26.
27.   p2 = malloc(sizeof(struct politician));
28.   p2->id = 1;
29.   p2->votes = 28000;
30.   p2->name = malloc(36);
31.
32.   strcpy(p1->name, argv[1]);
33.   strcpy(p2->name, argv[2]);
34.
35.   printf("Election results calculated!\n");
36. }
```

What you must do:

You must perform an attack on the program and cause it to run line 15 and output "Election Rigged!!". You must document your approach clearly in the following way:

1) Provide a large paragraph, or two, which gives a high-level description of the approach you intend to take in order to achieve your attack. It should be clear and concise. If you started with one approach, but swapped midway to another after realising something, describe both the initial idea and your final one here too.

2) Show step-by-step how you performed your attack. You should include screenshots of your input/output (e.g. using the Windows "snipping tool") and provide short comments explaining **why** you did each action. For example:

   Sample Command (should be screenshotted with the output included): x\24x $esp
   Sample explanations for why:

   I used it to show the stack. **(BAD – this is what you did, not why you did it)**

   I used it to show the contents of 24 addresses on the stack, in order to identify the exact location of the buffer and calculate how much overflow was necessary to write over the saved base pointer. **(GOOD! – here the purpose of using a command to show the stack is explained!)**

3) You must show how you would change the code to **fix all the vulnerabilities** in the program. Provide a brief description of why your changes fix the issues.

Question 2 – [30 Marks]

A marketing company has developed a social platform to allow fans to interact directly with Virtual Youtubers (https://en.wikipedia.org/wiki/Virtual_YouTuber) through a smartphone app, "Hitogata". The "Hitogata" app serves as a portal to a centralised online system located in the marketing company's HQ building. It allows fans to send messages to their favourite Virtual Youtubers, all of which are permanently recorded in a database, and allows them to send financial support. All such payments are also recorded. Fans may also interact with other fans, and a system is provided to allow fans to organise trades of vtuber merchandise.

There are 3 types of uses of the system: fans, vtubers and admins.

The app allows customers to send messages to vtubers and receive responses, as well as to view their message history. Similarly payments can be made and reviewed. Credit card details, legal name and proof of identification (e.g. passport id), and bank account details (to receive payments for merchandise trades) are required to register a fan account.

Vtubers can view and respond to messages from fans, provide refunds for payments received from fans, as well as monitor any trades relating to their merchandise. They can also apply to the admins to ban any fan account which is behaving too stalker-ish.

Admins have complete control over other accounts, being able to create, delete or modify any of their details.

All interaction is via the "Hitogata" smartphone app front end.

There are at least 2 databases: users-dbms and banking-dbms
Users-dbms contains all fan and vtubers' data as well as their message history. Banking-dbms contains credit card and bank information, and transaction histories.

Task:
Create a threat model of the system.

Clearly, the system is not tightly specified, so take liberty to add functionality or detail as you see fit, and **document all of your assumptions/details about the system** which were not specified in the above description.

The outcome of the task should be a level 1 DFD, and pdf file containing a threat analysis.

The DFD should describe the system above and include your own assumptions for the parts of the system not specified:

- To develop your DFD, make use of the MS Threat Modelling Tool (https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool). Use the "Create Full Report" option to create an html output summary of the model. Copy the relevant portions/images from this html output into your final pdf file for submission.

- Don't forget to include all your assumptions into the "Assumptions" section of the report, under "File->Threat Model Information", don't leave it blank.

The PDF should contain an analysis of 5 ***distinct*** threats which exist in your threat model (i.e. which were identified/generated by the MS Threat Modelling tool).

For each of your 5 threats you must:

- Provide a DREAD analysis of their risk rating (use the min(D, DREAD) metric). Provide a brief justification for each of the individual D, R, E, A, and D values you assign (for each of the 5 threats).
- ***In your own words***, describe the threat that the betting company developers face and what mitigation(s) they should make to solve it. (for each of the 5 threats)

Question 3 – [25 Marks]

1) Fully explain the 3 different types of *XSS*, the risks they pose, and how a website developer can mitigate them, in your own words.

2) Explain, giving a short example, what is meant by a *SQL Injection attack*, the risks they pose, and how a website developer can mitigate them, in your own words.

   Your answers for 1+2 should be understandable by a novice in the area of computer security, and clearly define the differences between XXS and SQLi.

Question 4 - [20 Marks]

1) Using your own example (i.e. not the exact one in the lecture notes), explain how a *Use-After-Free vulnerability* can manifest itself and the risks it poses.

2) Explain what the *Emsi Vulnerability* is, clearly illustrate which types of canaries this vulnerability renders ineffective, and explain how certain types of canaries are still effective in mitigating it.

Include all of the above in a single .pdf document. The name of the document **must be** your name followed by your student ID. e.g. "David Stynes R100000924.pdf". Penalties will be applied for incorrectly named submissions. Submit your pdf on Canvas in the submission facility located in "Assignments -> Assignment 2 Submission".

Due Date: Friday 18th December 23:59 (Week 12)

There will be a 1 week no-penalty late submission for the assignment: i.e. there will be no penalty for submissions before Friday 25th December 23:59. Late submissions after that date will be penalised -20%, or receive a score of 0 if later than Friday 1st January 2021 23:59.

Students may be interviewed to verify that their submissions were their own.