COMP8050 – Security for Software Systems

Assignment 1 (20%)

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 did not already do so in the labs, you may need to install the following packages too first:

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

assignment1.c:

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int grade;

void halfway()
{
  printf("Impressive performance!\n");
}

void thefinale()
{
  printf("I rate it as %d/100!\n", grade);
}

void vuln()
{
  char buffer[48];

  grade = 20;
  gets(buffer);
24.   printf("Buffer contents:");
25.   printf( buffer );
26. }
27.
28. int main(int argc, char **argv)
29. {
30.   vuln();
31. }
```

What you must do:

You must perform an attack on the program and cause it to run lines 10 and 15 in response to the user's input, outputting "Impressive performance!" and "I rate it as 100/100!". Any attack you may choose is valid if it outputs both lines (even if it does other things between them) and the program state is only modified by your user input to the program (i.e. GDB has commands to directly modify the value of global variable "grade". This is not permitted – GDB is to be used only to help you view the program state, not modify it directly, the same as it was demonstrated in the labs). You must document your approach clearly in the following way:

1) Provide a large summary 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!)**

   *The goal of your step-by-step description is to **show that you understand what you were doing and why you were doing it.** The best way to achieve this clearly is to ensure that your description provides enough detail to make sense to a computer scientist who had no prior security experience.*

Finally, you must also do the following:

3) Rewrite the necessary lines of the program to make it secure, fixing all of the vulnerabilities. Include a short description of the kind of attack you are attempting to block for any changes you make. (you do not need to rewrite the entire program, just show clearly the lines you do change).

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 1 Submission”.

Due Date: 27th October 23:59 (End of Week 7)

Students may be interviewed to verify that their submissions were their own. These interviews will occur during your assigned lab times.