COMP8053 – Embedded Software Security

Assignment 1 - 50% of final grade.

Answer **all** questions. Submit your answers in a single pdf file through the submission facility on Canvas in "Assignments -> Assignment 1 Submission". The assignment is due at 23:59 on November 17th (end of week 9).

What you must do (for each of Q1, Q2, Q3 and Q4):

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 description of the process taken to perform 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, along with any relevant output included too): 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.*

For Questions 2,3 and 4 you must also do the following:

3) You must show how you would change the code to **fix all the vulnerabilities** in the programs provided for Q2, 3 and 4. Provide a brief description of why your changes fix the issues.

For questions 2, 3 and 4, you will need to download the relevant C program file and copy it into the Protostar virtual machine. This can be done on Linux machines by using the "scp" (secure copy) command. For the Windows machines PuTTY, https://www.putty.org/, includes its own version of "scp" in the file "pscp.exe" located in the PuTTY installation directory. To use it, open a command prompt (start menu -> type in "cmd.exe") and navigate to the PuTTY installation directory. From there, type the following command:

```
pscp <filename> user@<ip address>:/home/user/
```

Where <filename> is the name of the C program file you are copying over, and <ip address> is the IP Address of the Protostar VM you are running (get it by typing "ip addr" inside the Protostar VM). This will create a copy of the file inside the /home/user/ directory on Protostar (the default starting directory).

The ip address "eth0" should start as something with 192.168... if the VM network connections have been set up correctly. They should be set in VMWare by default. In Virtualbox, you will need to manually set the network type to "Bridged Adapter" if it does not have the correct form of ip address for eth0.

Next you must compile the C program, with the following command:

```
gcc <filename> -o <outfile name>
```

Where <filename> is the name of the C program file, and <outfile name> is your name for the compiled binary. You can then run the compiled binary by typing:

```
./<outfile name>
```

Contact me if you have any difficulties in copying over and compiling the C programs! It is not intended that getting the files onto the VM are part of the challenge!

Question 1 – [10 Marks]

Download the 3 Firmware binaries, named "Firmware1.bin", "Firmware2.bin" and "Firmware3.bin" in "Units -> Marked Assignment 1" on Canvas. For these 3 binaries, you must do the following:

1. Locate a private cryptographic key stored as plaintext in 2 of the firmware binaries.
2. Locate login credentials for a remote connection (e.g. ssh, telnet, https, etc..) in one of the firmwares. Note that you must make sure you find the full login name and password combination (not merely a variable for either of them, if it is part of a script).

You may use any of the tools we used in the labs to assist in your search. You may also use any of linux's built-in commands to assist you (e.g. 'grep' to search files). Remember you can type "man <linux command>" to get instructions on how to use linux commands if you are unsure. You may not use any other types of tools than these.

Question 2 – [20 Marks]

For this question, you will use the program "Question2.c" available in "Units -> Marked Assignment 1". You must make the program run the "insecurefunction" function and **output "I haven't been called in so long, do the users hate me?"** and then cause the program to **exit without crashing**.

Hint: You can find the address of any libc function inside of gdb with the command: p <libc function name> (replacing <libc function name> with the name of a function, e.g. *p system*)

Question 3 – [25 Marks]

For this question, you will use the program "Question3.c" available in "Units -> Marked Assignment 1". Before you compile the code, change the code so the "studentid" variable is being assigned your real numeric student id. Note: if the rng gives globalcanary a value that results in you needing to input null bytes for your attack, you may increment your studentid until the result of rand() does not give you any null bytes. You must cause the program to **output "Nobody ever calls me, I'm so bored."** from the function "lonelyfunction". **It is fine if the program crashes** after doing this.

Question 4 – [30 Marks]

For this question, you will use the program "Question4.c" available in "Units -> Marked Assignment 1". You must cause the program to **output "Student received a grade of 100."** from the function "evaluate", **it is fine if the program crashes or behaves abnormally** after doing this.

Question 5 – [15 Marks]

1) Explain what is meant by a Canary, and then evaluate the effectiveness and limitations of Canaries as a defence versus attacks on format string vulnerabilities.

2) Explain what is meant by ASLR, and then evaluate the effectiveness and limitations of ASLR as a defence versus attacks on format string vulnerabilities.

3) Explain what is meant by a Executable Space Protection (also known as NX-bits/DEP/W^X), and then evaluate the effectiveness and limitations of Executable Space Protection as a defence versus all types of attacks on stack buffer overflow vulnerabilities.

Academic Integrity

This is an individual assignment. The work you submit must be your own. In no way, shape or form should you submit work as if it were your own when some or all of it is not. Any online source that is used must be cited. If you are unsure on whether something should be cited, general rule of thumb is to err on the side of caution and include the citation. You can also ask me via email. You must not directly copy text even from a cited source. All answers must be in your own words.

Collusion: Given how much freedom there is in the assignment, everybody's work will be different. It will be obvious if there is collusion. All parties to collusion will be penalized.

Deliberate plagiarism: You must not plagiarise the programs, results, writings or other efforts of another student or any other third-party (including AI tools like ChatGPT). Plagiarism will meet with severe penalties, which can include exclusion from the Institute.

Your report will be checked for signs of collusion, plagiarism, falsification and fabrication. You may be called to discuss your submission and implementation with me and this will inform the grading, any penalties and any disciplinary actions.