Secure Programming Coursework Part 1

Arthur Chan and David Aspinall

School of Informatics, University of Edinburgh

This is an **individual** assessed practical exercise. It is the only assessed coursework for the Secure Programming course. It consists of two parts. Part 1 is issued first and covers the earlier portion of the course. Part 2 will be issued later and covers lectures and lab exercises yet to come. Provided you have attended the relevant lectures and lab sessions in the course, the work for both parts should take about 30 hours. The practical will be awarded a mark out of 100. The single deadline for submission (both parts) is **4pm, Fri 22nd March 2019**. The final page summarises the submission instructions.

Virtual machinery

We provide a virtual machine for you to use. The VM has two users, `user` and `root`. The **passwords are the same as their usernames**.

To install the VM, you should use a virtual disk file stored on a local disk on your machine. If you are working in the Appleton Tower Lab on the DICE machines, you can use the directory `/tmp/sNNNNNNN` if there is enough space. Configure VirtualBox to use the right disk area by setting `File → Preferences → General → Default Machine Folder`. Next, **import the appliance** (`File → Import Appliance`) from the file:

```
/afs/inf.ed.ac.uk/group/teaching/module-sp/SecureProgramming-Coursework.ova
```

If you are working remotely or on your own machine, we recommend taking a copy of the `.ova` file first rather than importing over directly from AFS. The file is about 1G. It may be more convenient to use a USB stick than download it over the Internet.

Important: make sure that your VM disk files are stored in a directory which is only readable by you. Beware that `/tmp` are disk areas which are not backed up. So if you are using a lab machine (and anyway, for safety), **back up your work** by saving any work that you do inside the virtual machine (edited source files, etc) in your home directory.

Using the machine

You should complete all questions as the unprivileged user called `user`.

The machine is set to use NAT. Once started you can either use the console window, or (recommended): SSH in via your local machine over port 2222, with: `ssh -p 2222 user@localhost`.

We've supplied some tools to make things easier but feel free to install additional software in the VM. In your answers, please describe **all tools you have used**, including Linux packages, browser plugins used in your host machine, etc.

1. Meltdown and Spectre (25 marks)

In 2017, two critical vulnerabilities has been discovered in modern processors, they are named as Meltdown and Spectre separately. The details of these vulnerabilities has been record in three CVE entries. In particular, we will look at CVE-2017-5715, CVE-2017-5753 and CVE-2017-5754 describing these two infamous vulnerabilities. You should put your answer of this question in **answers1.pdf**.

1. Study the three CVE entries and related documnetations for Meltdown and Spectre, in your own words briefly describe the similarity and difference of the two vulnerabilities. (4 marks)

2. There are three CVE entry describing Meldown and Spectre, please identify which of them is(are) describing Meltdown and which of them is(are) describing Spectre. In addition, please state the affecting system for each of them. (4 marks)

3. Identify the Common Weakness (CWE) of these two vulnerabilities. Also identify which scope in the CIA triad has been violated by this common weakness and why. (3 marks)

4. Identify the possible consequences of these vulnerabilities and how an attacker can make use of the consequences. (3 marks)

5. These two vulnerabilities are considered to be "hardware" vulnerabilities. Briefly discuss the difference between hardware vulnerabilities and software vulnerabilities and how you might draw a distinction. To help your explanation, use at least one other example vulnerability listed in the NVD (https://nvd.nist.gov) and occurring in the last two years. (4 marks)

"My computer has the newest version of anti-virus, firewall and internet security installed. So it is super safe against any vulnerabilities and attacks, including Meltdown and Spectre."

By Mr. Super Secure

6. State your proposition and discuss if you agree or disagree with Mr. Super Secure's word. Please provide some reasons to support your proposition and identify why those security software can help (or have no use) to stop attacks targeting Meltdown and Specture vulnerabilities. (5 marks)

7. Apart from installing the above mentioned security software, please briefly describe other ways to protect your computer from attacks targeting Meltdown and Sepctre vulnerabilities. (2 marks)

2. Secure Coding and an Exploit (25 marks)

Code for this question is in the `/home/user/exploit` folder on the VM. You are given a program called **vulnerable** which is compiled from **vulnerable.c** (to recompile the program, simply use the **make** command). The program is an authentication program for a message board, with an obvious vulnerability. There is a second similar program **vulnerable2.c** with other vulnerabilities.

Please put all your written text-based answers in **answers2.pdf** for this question.

1. Identify the vulnerability in the program **vulnerable** and give the name of a CWE which categorises the vulnerability type most closely. Briefly explain the vulnerability and identify how and why can an attacker abuse it. Lastly, identify possible consequences if the vulnerability is abused. (6 marks)

2. A successful attack should pass all the checking and cause the program to write the message `Hello World!` to the **messageboard.txt**, printing out `Message stored!!`, and *without* providing the correct user username and password as input. Create an exploit script called **exploit** that takes the path of the program as its first argument and launches a successful attack. We will run your program by executing:

   ```
   ./exploit ./vulnerable
   ```

   It must not output anything other than the output produced by the vulnerable program. You may use any scripting language to write your exploit, provided it runs as described. If you use a high level language, please also attach your original source code in **answers2.pdf**. If your exploit cannot be run as described (files missing or execution errors), no marks will be awarded for this part. We will execute your code in a fresh VM copy of the machine imported into Virtual Box. (3 marks)

3. Please briefly explain your **exploit** script created for the last question and describe it abuses the **vulnerable** program to force it to store your message. If you use some hard-coded values, please also explain how you get the value. (2 marks)

4. Provide a patch file that fixes the vulnerability of **vulnerable.c**. The patch file should be named as **question2a.diff**. (2 marks)

5. There is another program **vulnerable2.c** with multiple vulnerabilities. Perform a code review and report up to three *different* vulnerabilities in this second program.

   For each vulnerability, describe what the problem is, how it might be exploited, and what the possible consequences of an exploit might be. Finally, give a correction to the code to show how it may be fixed.

   You should provide your description and answers in **answers2.pdf** and provide a patch file that fixes your three reported vulnerabilities of **vulnerable2.c**. The patch file should be named as **question2b.diff**. (12 marks)

Note

Patch files can be created with the command

```
diff -c <oldfile> <newfile>  > question2x.diff
```

Keep a copy of the original file so you can make the patch file!

Reminder warning: *Never execute your exploits on a real machine* unless you have the express permission of the owners to do such testing.

Submission instructions (Part 1)

Remark: Submission instructions for part 2 will be released later.

You should submit your answers electronically with the command:

`submit sp cw` *filename*

or if you want to submit multiple files:

`submit sp cw` *filename filename* ...

Where *filename* is:

answers1.pdf A PDF document containing the answers to question 1.

answers2.pdf A PDF document containing the answers to question 2.

exploit The script required to exploit the program for *Question 2.2*.

question2a.diff The patch file generated for *Question 2.4*.

question2b.diff The patch file generated for *Question 2.5*.

Wrong *filename* arguments will not be accepted. The PDF documents should be well-formatted printable A4 PDFs, you may generate them with whatever program you want. Text answers should be brief and to-the-point.

Repeated submission of the same *filename* will overwrite the previous submission. Take care: the submission does not keep a history of submitted files, we will mark the most recent files and their submission timestamps must be before the deadline to avoid standard lateness penalties.

You must submit by the **final deadline 4pm, Fri 22nd March 2019**.

The coursework is separated into two parts and released in stages, but both parts have the same final deadline. However, you are **strongly encouraged** to submit your answer for the first part before the second part is released, to help you manage your time. So:

It is recommended to submit for this part by **Friday 1st March 2019**.

You're reminded that late coursework is not allowed without "good reason", see the fourth year honours course guide[1] for details, and the procedure to follow if you must submit late. In particular, if you have a good reason to submit late, use the ITO support form to make a request rather than asking us.

[1] http://www.inf.ed.ac.uk/teaching/years/ug4/courses.html#lateSubmission