Secure Programming Coursework (Part 2, not-for-credit)

Arthur Chan, Margus Lind and David Aspinall

School of Informatics, University of Edinburgh

This is an **individual** assessed practical exercise. Your work will be marked and feedback returned to you, but the assessment will **not** count towards your final mark for the course.

The practical will be awarded a mark out of 25. The deadline for submission is **4pm, Fri 13th April 2018**. The final page summarises the submission instructions.

Virtual machinery

We provide a virtual machine for you to use. The VM has one user, `user`. The password is the same as the usernames.

To install the VM, you should use a virtual disk file stored on a local disk on your machine. For example, 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 from the file:

```
/afs/inf.ed.ac.uk/group/teaching/module-sp/SecureProgramming-Coursework-Part2.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`. Unlike the other lab and part 1 of the coursework, you are not given the root access for this virtual machine. But you are given sudo right to execute some commands (restarting of services).

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`.

Additionally the VM runs a web server on **port 80**. This is forwarded to **port 8080** on the host machine.

We've supplied some tools to make things easier but feel free to install additional software in the VM. In your **answers3.pdf**, please describe **all tools you have used**, including Linux packages, browser plugins used in your host machine, etc.

3. Secure Server Management

In this section, you are only given the access to the VM with an unprivleged user `user` with sudo right to execute some privileged commands. You can always type the command

```
sudo -l
```

to check your allowed privilege commands list.

The VM is set to use NAT. We have already set three port forwarding rules in the VM. If you think it is necessary, you can add more port forwarding rules. If you do so, please include all the new / modified port forwarding rules in your **answers3.pdf**. The existsing port forwarding rules are as follows:

> **For SSH** Virtual Machine port 22 is forward to host machine port 2222. (TCP)
>
> **For Openssl** Virtual Machine port 12345 is forward to host machine port 54321. (TCP)

All the description and answers should be written in **answers3.pdf**, while there are additional files required for each question below which you should submit separately. The full list of submission files for this part of the coursework is summarized at the final page.

3.1 OpenSSH Configuration (5 Marks)

The server has ssh service turned on. It allows users connect to the server through port 22 (which is forwarded to port 2222 on your host machine). Each students are given the same VM with the same set of login information. Using default password is not a secure way to manage your own web server. Others can possibly break into your VM and ruins your work. So the first thing you need to do is to secure the ssh connection with some configurations. This can prevent others using default credential to crack into your Virtual Machine. You will need to complete the following requirements.

a) Only allow `user` to login through ssh. Deny all other user to login through ssh service. (2 Marks)

b) Disable password authentication and only allow `user` to login with a private key. (Hints: You may need to generate key pairs with *ssh-keygen* command which exists in almost all linux environment) (3 Marks)

Remarks: You may need to restart the ssh server to apply some of the changes. You have been given the sudo privilege to restart the sshd service. You may want to read the man page of sshd by **man sshd**.

For this question, submit a tar ball **sshconfig.tar.gz** which includes all the config files you have added or modified. Then provide a brief description in **answers3.pdf** to describe the use of each of your modification and how it links to the requirements. If you think the config files need to be stored in a specific path to make something work, please also mention that in your description. For clarification, please use absolute paths in your description (Starting from root /).

3.2 Fun with Heartbleed (10 Marks)

Heartbleed is a serious bug in OpenSSL which was discovered a few years ago. Now you will have the chance to fix heartbleed bugs which exists in this virtual machine. There are three folders in `/home/user/` which is related to this question.

/home/user/key This folder contains the key and certificate that you need to start up the openssl service.

/home/user/openssl This folder contains the openssl package which is compiled from the source version 1.0.1f.

/home/user/openssl-1.0.1f-source This folder contains the source code for openssl version 1.0.1f.

The server has been configured with the correct PATH. Whenever you execute the openssl command, it will first choose to execute /home/user/openssl/bin/openssl by default.

If you need to regenerate the key, you may need to use this command.

```
openssl req -x509 -newkey rsa:2048 -keyout key.pem -out cert.pem -days 365 -nodes
```

If you need to recompile the source (after you fix something), please go into the source code folder and execute this command.

```
./config --prefix=/home/user/openssl enable-tlsext no-shared && make && make install_sw
```

The above command will compile the source code and overwrite the package in /home/user/openssl automatically.

If you want to start the openssl service, please execute this command.

```
openssl s_server -key /home/user/key/key.pem -cert /home/user/key/cert.pem -accept 12345 -www
```

The above command will start a web openssl service on port 12345, which will forward to port 54321 on your host machine.

a) In **answers3.pdf**, briefly explain how an attacker can exploit a heartbleed bug and what is the possible consequence of such an attack? (2 Marks)

b) Try to review the code and fix the heartbleed bug. Briefly describe how you fix it in **answers3.pdf**, then submit a separate patch file **question3.diff** that show your change in the code. Your patch file should only fix the heartbleed bugs. Zero marks will be awarded for this question if your patch 1) alter the functionality, remove or stop the hearbeat features, or 2) affect or fix other part of the code which is not related to the heartbleed bug. (8 Marks)

Hints: Version 1.0.1f is the last openssl version affected by Heartbleed bug, the patch update for later version may help you to discover how to fix the heartbleed bug.

Hints 2: There are loads of heartbleed testing tools around the Internet for testing your patch. (Don't execute those tools on a real server unless you have the owner's permission!)

Hints 3: Heartbleed is a bug happen in the stage of processing heartbeat (thats why it is named as heartbleed), try look into the **ssl** folder in the source code to find the problem and fix it.

Remark: Even if you cannot fix the bug, if your description is partially correct, you will still get part of the marks.

For this question, submit a patch file **question3.diff** to demonstrate your fix to the heartbleed bug. We will apply and test if your fix really prevent the heartbleed problem. If it is not, we will look for your description in **answers3.pdf** and see if your approach and description is acceptable to be awarded with partial marks. Your **answers3.pdf** should also contains your answer to part (a) of this question.

3.3 Side Channel Timing Attack (10 Marks)

Side Channel Attack is an interesting topic in Secure Programming. We always know that when we need the computer to perform some actions, it may always consume different amount of time and computational power. If these physical implementation of the program execution follows some patterns, it may become another sort of vulnerability.

In this question, you will find an executable **vulnerable** which take in an input password as the first argument and check if it is correct. In this program we purposely add in some timing patterns and allows you to easily bruteforce the password. Your target is to brute force for a correct password. When this correct password pass to the program, it should print "Your password is correct!!". The target program is stored in `/home/user/timing/vulnerable`. No source code is given for this question.

a) Create a script file **bruteforce** which will brute force for the correct password with consideration of the simulated timing patterns. You are allowed to use any scripting language, provided it can execute as **./bruteforce**. If you use high level language, please also include your code in **answers3.pdf**. (5 Marks)

   Remark: You can assume the correct password only contains A-Z / a-z / 0-9 and the maximum length of the password is 20 characters.

   Remark 2: If your answer only contains a general brute force without considering the side channel attack, you will only get 1 marks for this question.

b) In **answers3.pdf**, briefly descibe the timing patterns that we added to the program. Also, provide a brief description of your approach. (5 Marks)

For this question, submit the script file **bruteforce** and add in your description and answer for part (b) in **answers3.pdf**.

Hints: Try the following wrong password and experience the difference.

WrongPassword

WrongPassword000

WrongPassword0000

ArthurChanWithDA

Submission instructions (Part 2)

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:

answers3.pdf A PDF document containing your description to question 3.

sshconfig.tar.gz A tar gz file containing all your modified config for *Question 3.1*.

question3.diff The patch file generated for *Question 3.2*.

bruteforce The attack script required to exploit the program for *Question 3.3*.

You can create the tar gz file by

tar -cxf code.tar.gz <The directory of all your code>

You can create the diff patch file by

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

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.

To receive feedback on your work, please submit by the **deadline 4pm, Fri 13th April 2018**.