ROBERT GORDON UNIVERSITY ABERDEEN

School of Computing Science and Digital Media

Faculty of Design and Technology

Coursework Assignment

Student Name:	
Matriculation Number:	
Contact phone number (if we require to contact you urgently):	
Course**: ***	
Stage: *	
Time taken to complete: *	1-4 5-9 10-14 15-19 20+ hours
Lecturer/Examiner:	Carlos Moreno Garcia
Module Name:	System Programming and Security
Module Number:	CMM530
Coursework Title:	Securing and Breaking into Linux Virtual Environments
Due Date:	30th November, 2018

Declaration ** *This **must** be affirmed by signing below*

I confirm

- **that the work undertaken for this assignment is entirely my own and that I have not made use of any unauthorised assistance**
- **that the sources of all reference material has been properly acknowledged.**

Student Signature	
Date Submitted	

For Office use

Marker's Comments	
Marker	**Grade**

****** An extract from the University Regulations

6. Academic Misconduct

Refer also to Schedule 3.3 of this Regulation for guidance on this procedure.

6.1 **Academic Misconduct** is defined as any attempt by students to gain an unfair advantage in assessments and examinations. Examples of academic misconduct include plagiarism, cheating, falsifying data, collusion, bribery or attempted bribery, personation or any other activity intended to provide an unfair advantage.

(i) **Plagiarism** is the practice of presenting the thoughts or writings of another or others as original, without acknowledgement of their source(s). All material used to support a piece of work should be carefully referenced and should not normally be copied directly unless as an acknowledged quote. Text translated into the words of the individual student should in all cases acknowledge the source.

(ii) **Cheating** includes:

- the taking of any unauthorised material into an examination;
- obtaining copy of "unseen" papers in advance of an examination;
- communicating or attempting to communicate in any way with another student during an examination;
- copying or attempting to copy from another student during an examination or in the production of coursework;
- wilful deception in any element of an examination or assessment.

(iii) **Falsification of data** consists of the misrepresentation of the results of experimental work or the presentation of results from fictitious work.

(iv) **Collusion** is the representation of unauthorised group work as that of an individual student.

(v) **Bribery** is the paying, offering or attempted exchange of an inducement for information or material intended to advantage the recipient in an examination or assessment.

(vi) **Personation** consists of a substitute taking the place of a student in an examination.

A student who aids and abets a fellow student to commit academic misconduct shall be deemed to have committed academic misconduct and will be dealt with accordingly.

System Programming and Security Coursework 2018

INTRODUCTION

The main aim of this coursework is twofold:

1. To make the default Linux system more secure by implementing a variety of operating system security measures.
2. To exploit the vulnerabilities of a non-secured Linux system and retrieve confidential information.

These activities will be evaluated by using virtual machines (VMs).

COMPONENT 1 – SECURING A LINUX OS

This component is worth 50% of your final grade for this coursework.

Requirements:

1. Using VMWare Workstation, create an Ubuntu VM called `**Student_ID**_Client`. Then, perform the following tasks **describing their outcomes in a report with corresponding screenshots**.

   1.1. Using `CLI` commands:

      i. In the client VM, create two user accounts called `User_1` and `User_2` belonging to the *LOCAL* user group, and a third user account called `User_3` belonging to the *GLOBAL* user group.

   1.2. In the `User_1` home directory, write a shell script called `cw1` that enables through a menu the following options:

      i. Display a *long* list of files and subdirectories from the home directory.

      ii. Display the first 5 lines of the log file messages (`/var/log/syslog`).

      iii. Display all lines of the script `cw1`.

      iv. Exit the script.

   NOTE: The script should be able to identify and indicate if the user selected an invalid option, and to loop the script until a correct option is selected.

   1.3. Store the script file in the `/scripts` subdirectory, owned by `User_1`.

   1.4. Using the `chmod` command and permission flags/access control lists:

      i. Set the `read`, `write` and `execute` permission of `cw1` for `User_2`. Check that `User_2` can exercise these permissions by modifying the script so that when executed, it can display its own information (i.e. the long list of files and subdirectories of the home directory of `User_2`).

      ii. Set only the `execute` permission of `cw1` for `User_3`. Check that `User_3` can execute the script modified by `User_2`, but that `User_3` cannot modify the script.

2. On the client VM create a nested VM called `**Student_ID**_Server` to be used as a server. Perform the following tasks, **describing their outcomes in a report with the corresponding screenshots**.

   2.1. Configure a firewall on the server so that *secure shell* is enabled.

   2.2. Ensure that the client VM (using any user) can connect to the server VM using the `ping` command.

   2.3. Using the `gnome-system-tools` package:

i. In the client VM, create a user account called `User_4` that belongs to the *REMOTE* group.

ii. With `User_4`, remotely login to the Server VM via SSH (TIP: you might need to start a SSH daemon `/etc/init.d/sshd` on the server VM to make this work).

iii. Remotely run a shell script of your choice on the server, making sure that `User_4` can continue running the chosen script until he/she wants to exit.

COMPONENT 2 – EXPLOITING A CROSS SITE SCRIPTING (XSS) VULNERABILITY

This component is worth 50% of your final grade for this coursework.

Cross-site scripting (XSS) is a type of security vulnerability typically found in web applications. XSS usually occurs due to poor programming practices of programmers, which enables attackers to inject client side scripts into web pages viewed by other users. XSS attacks can be used to bypass access controls like firewalls rules. In this section of the coursework, you will exploit a XSS vulnerability to gain access to a target machine (victim) in order to steal sensitive information.

1. Using VMWare Workstation and the three virtual machines provided, setup a network topology as shown in the diagram below:

1.1. Use the Linux *ping* command to verify that each machine can communicate with the others. (Hint: you may need to use *ifconfig* command to find the IP address of each machine).

1.2. Harden the victim machine to only allow www and email traffic both ways (Hint: You may configure a host firewall to achieve this).

1.3. Use the Browser Exploitation Framework (BeEF) on the attacker's machine to create a malicious payload.

1.4. Using a cross site scripting (XSS) attack, inject above payload to the "vulnerable web server", so that the injected code is executed every time when the vulnerable page is loaded.

i. To locate a vulnerable page, type www.vul-`**Student_ID**`.com on the attacker's browser (e.g. www.vul-0001.com). Then, click on "DVWA". Enter username "admin" and password "password" on the login page. Once you

log into the page, use the "DVWA security" tab in the left pane to lower the security of the website. Then, click on "XSS Stored" tab to open the vulnerable page. You will use this page to execute your XSS attack.

1.5. Load the vulnerable page on the victim's browser, and make sure victim is hooked into the BeEF. **Describe the following outcomes in a report with the corresponding screenshots.**

i. What is the victim's browser name and version?

ii. List all the plugins installed on victim's browser.

iii. Take a screenshot of the victim's browser window. This should be done remotely and in a stealthy manner from the attacker's machine.

iv. Use "Google Phishing" to steal the victim's user credentials from the victim's Google account.

1.6. Though we blocked all incoming connections to the victim (in 1.2 above), the attacker could gain access to the victim's machine. In this scenario, how would the attacker bypass the access controls on the firewall? (max. 100 words).

1.7. As a programmer, what coding practices will you adhere to in order to avoid the XSS vulnerability in your future applications? (max. 100 words).

SUBMISSION

You need to submit a single report for both components to the dropbox submission folder on CampusMoodle (with a signed coursework front sheet), which records screenshots and the commands/steps used to perform each task.

Submission Feedback:

Written feedback (comments written in the individual coursework) and summary feedback via CampusMoodle will be provided to students within 20 working days after submission.