DEPARTMENT OF COMPUTER SCIENCE
ASSESSMENT DESCRIPTION 2019/20
(EXAM TESTS WORTH ≤15% AND COURSEWORK)

MODULE DETAILS:

Module Number:	700108	Semester:	1
Module Title:	Network Security		
Lecturer:	Brian Tompsett		

COURSEWORK DETAILS:

Assessment Number:	1	of	2		
Title of Assessment:	Network Protocol generation and decoding program				
Format:	Program	2nd format	Presentation		
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	50	and	80	hours on this assessment
Length of Submission:	This assessment should be **no** more than: *(over length submissions **will be** penalised as per University policy)*	N/A - coding exercise **words** *(excluding diagrams, appendices, references, code)*			

PUBLICATION:

Date of issue:	25 September 2019

SUBMISSION:

ONE copy of this assessment should be handed in via:	Canvas	If Other (state method)	Demonstration
Time and date for submission:	**Time** *(must be before 4pm)*	2pm	**Date**	7 November 2019
If **multiple hand–ins** please provide details*:*	A demonstration of the final program will be arranged		
Will submission be scanned via TurnitinUK?	No	Students are reminded they can **ONLY** submit **ONE** file and must ensure they upload the correct file.	

The assessment must be submitted **no later** than the time and date shown above, unless an extension has been authorised on a *Request for an Extension for an Assessment* form: search ‘student forms’ on https://share.hull.ac.uk.

Canvas allows multiple submissions: only the **last** assessment submitted will be marked and if submitted after the coursework deadline late penalties will be applied.

MARKING:

Marking will be by:	Student Name

ASSESSMENT:

The assessment is marked out of:	100	and is worth	30	% of the module marks
N.B If multiple hand-ins please indicate the marks and % apportioned to each stage above (i.e. Stage 1 – 50, Stage 2 – 50). It is these marks that will be presented to the exam board.				

ASSESSMENT STRATEGY AND LEARNING OUTCOMES:

The overall assessment strategy is designed to evaluate the student's achievement of the module learning outcomes, and is subdivided as follows:

LO	Learning Outcome	Method of Assessment *{e.g. report, demo}*
LO1	*Demonstrate a comprehensive understanding of the vulnerabilities of Networked Computers and their protocols*	code, demo
LO2	*Provide evidence of research, selection and assessment of a range of mitigation techniques against vulnerabilities*	code, demo
LO3	*Apply techniques for the Management and Configuration of Firewalls*	code, demo
LO4	*Select, justify and apply a range of Network Management Protocols*	code, demo

Assessment Criteria	Contributes to Learning Outcome	Mark
Packet Capture Tool:		
- Ability to capture packets	LO1	10
- Display of packets	LO2	10
- Decode of packets	LO3	10
- Overall Quality of product and code	LO4	10
- Submission and demonstration		10
SNMP agent tool		
- Ability to reply to SNMP queries		10
- Trap generation		10
- Abilty to interface with windows stats		10
- Overall Quality of product and code		10
- Submission and demonstration		10

FEEDBACK

Feedback will be given via:	Verbal (via demonstration)	Feedback will be given via:	Canvas
Exemption (staff to explain why)			
Feedback will be provided no later than 4 'teaching weeks' after the submission date.			

This assessment is set in the context of the learning outcomes for the module and does not by itself constitute a definitive specification of the assessment. If you are in any doubt as to the relationship between what you have been asked to do and the module content you should take this matter up with the member of staff who set the assessment as soon as possible.

You are advised to read the **NOTES** regarding late penalties, over-length assignments, unfair means and quality assurance in your student handbook, which is available on Canvas - https://canvas.hull.ac.uk/courses/17835/files/folder/Student-Handbooks-and-Guides.

In particular, please be aware that:

- Up to and including 24 hours after the deadline, a penalty of 10%
- More than 24 hours and up to and including 7 days after the deadline; either a penalty of 10% or the mark awarded is reduced to the pass mark, **whichever results in the lower mark**
- More than 7 days after the deadline, a mark of zero is awarded.

- The overlength penalty applies to your written report (which includes bullet points, and lists of text. It does not include contents page, graphs, data tables and appendices). 10-20% over the word count incurs a penalty of 10%. Your mark will be awarded zero if you exceed the word count by more than 20%.

Please be reminded that you are responsible for reading the University Code of Practice on Academic Misconduct through the Assessment section of the Quality Handbook (via the SharePoint site). This govern all forms of illegitimate academic conduct which may be described as cheating, including plagiarism. The term ‘academic misconduct’ is used in the regulations to indicate that a very wide range of behaviour is punishable.

In case of any subsequent dispute, query, or appeal regarding your coursework, you are reminded that it is your responsibility, not the Department’s, to produce the assignment in question.

Network Datagram Decoder and SNMP agent

INTRODUCTION

This assessment is for you to learn about the format of internet datagrams by capturing, decoding and displaying network packets. It also explores the network management protocol SNMP and how such packets are encoded and communicated to network management software. You should therefore create two application programs in C#:

1. A (client) application to capture, decode and display internet packets with a particular focus on SNMP. Your program may capture all kinds of packets and decode and display them which will earn further marks. The user interface to this application and the clarity with which it decodes and displays packets will also be rewarded.
2. An application to act as an SNMP agent (which can act as a server or service) which will return SNMP information about the device which runs it.

METHODOLOGY

The application desired could be quite large and students are required to use open source C# libraries (with the restrictions detailed below) as the basis of building their applications. This allows the resultant applications to be more capable than if the project was coded from scratch by the student, but also follows standard industry practice for tools of this nature. Students need to experience obtaining and learning how to use larger code libraries.

The following open source projects should be consulted. No other sources of external code should be included in the student's work unless previously agreed with the course lecturer. The division of code between that imported from the open source library and that amended or created by the student should also be clearly indicated in the code comments.

1. The CsharpPcap – Packet Capture Library in C Sharp

   - https://sourceforge.net/projects/sharppcap/

2. The Simple Network Sniffer in C Sharp

   - https://www.codeproject.com/Articles/17031/A-Network-Sniffer-in-C

3. The SNMPSharpNet – An SNMP Library for C#

   - http://www.snmpsharpnet.com/

If you discover you need Administrator rights to develop, test or run your applications you should consult your course lecturer as information on this will be provided during class.

SUGGESTED SCHEDULE

The module will not have timetabled laboratory sessions but there will be time during the taught classes to discuss aspects of the assignment and answer questions. Also, staff will, if required, run laboratory workshops but you should request this during timetabled classes.

In order to create a large software artifact an incremental, or staged, approach is suggested. You may wish to consider the following stages of work:

Step 1 - Prepare the necessary materials and information for the work. Create an ordered set of directories to store the work in your University assigned home directories (G:). Select your desired workstation in the laboratory in preparation for need Administrator rights in windows. Ensure you have access to the university resources such as SVN, Box and Canvas. Download the sources for the suggested tools and place the source under SVN, check that the tools compile and run individually before starting to use their code. Prepare for working off-site, if you have your own computer, as this will enable you to work more conveniently outside laboratory hours. You should download the University VPN client and check that it works, and map the university resources at home. Ensure you have the necessary tools (such as Visual Studio) using the university Microsoft Imagine store if required.

Step 2 - Use the packet capture tool (from the open source toolset) to capture packets from your own machine (localhost) and send packets to it using the telnet client. Create a simple SNMP agent using the open source tools (unchanged) and bind this to localhost. Use the simple SNMP query tool from the open source tools to check that you can fetch and decode SNMP datagrams properly from locahost. Use the packet capture tool to examine the packets from the SNMP transaction.

Step 3 - Now enable the tools to operate between different hosts and IP addresses, and also packet capture in promiscuous mode. This activity may need Administrator rights to operate.

Step 4 - Complete the design of the user interface for your desired tools and finish the creation of an SNMP agent and a packet capture and decoding tool.

Step 5 - When implementation is complete run a series of tests to ensure the tools will operate satisfactorily when assessed.

Step 6 - Create the necessary zip file for submission, with a README file and upload to the Canvas VLE for submission before the deadline.

If you feel you do not understand the assignment or what is required do not hesitate to ask during class as time has been allocated for discussions of the work in the timetable.