UNIVERSITY OF Hull

COMPUTER SCIENCE
ASSESSMENT DESCRIPTION 2017/18
(EXAM TESTS WORTH ≤15% AND COURSEWORK)

MODULE DETAILS:

Module Number:	700100	Semester:	2
Module Title:	Trustworthy Computing		
Lecturer:	BCT & CK		

COURSEWORK DETAILS:

Assessment Number:	1	of	3		
Title of Assessment:	Programming of a Windows Sandbox				
Format:	Program	Demonstration	3rd format		
Method of Working:	Individual				
Workload Guidance:	Typically, you should expect to spend between	40	and	60	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			

PUBLICATION:

Date of issue:	28th January 2020

SUBMISSION:

ONE copy of this assessment should be handed in via:	Canvas		If Other (state method)	
Time and date for submission:	**Time**	14:00	**Date**	27th Feb 20201
If **multiple hand–ins** please provide details*:*	Demo – dates to be arranged			
Will submission be scanned via TurnitinUK?	No	If submission is via TurnitinUK within E-Bridge students MUST only submit Word, RTF or PDF files. Students MUST NOT submit ZIP or other archive formats. 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:* see the Canvas site: Help&Support > Student Forms. 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 Number

ASSESSMENT:

The assessment is marked out of:	100	and is worth	20%	% 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 an understanding of the need for security and fundamental security concepts in a professional, legal and ethical context.*	Program, Demo
LO2	*Provide evidence of research, selection and assessment of a range of security algorithms and techniques.*	Program, Demo
LO4	*Use a range of security algorithms to secure network-based applications*	Program, Demo
LO5	*Use source-based security policies to define privileges and implement security in distributed applications.*	Program, Demo

Assessment Criteria	Contributes to Learning Outcome	Mark
Ability to execute a variety of programs	1,2,4,5	20%
Ability to change access policies	1,2,4,5	20%
User Interface Capability	1,2,4,5	20%
Code Review	1,2,4,5	20%
Packaging, Testing, Delivery	1,2,4,5	20%

FEEDBACK

Feedback will be given via:	Verbal (via demonstration)	Feedback will be given via:	Feedback Sheet
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.

Windows Sandbox Tool

INTRODUCTION

This assignment requires you to use the knowledge and skills learned in the first semester to build a tool that is useful in exploring the concepts of trustworthy computing. This assignment also introduces security software and attributes that will be explored further in the second assignment, and acts as a foundation for software development for the third course assignment.

When running an executable programs from various random authors, such as student delivered code or untested applications from the internet, we are potentially risking our computer to being compromised. The code may inadvertantly contain a malware vector or could even be deliberately contructed to perform unwanted actions on our machine, such as damage files or access otherise hidden data.

One way of safely running such code would be to use a virtual machine or a software sandbox. These may permit the code to be run with affeccting other things, but there still may be no clear indication of any hidden unwanted actions they may have initiated. What is required is a tool that can be configured to say what actions are permitted and what are prohibited by any untrusted code and when the code is run some form of alert or exception should be raised if the prohibition is triggered. There are some tools that can perform this task but they are often geared towards the execution of a single application in an interactive environment.

When assessing student programming assignments there can be many hundreds of untrustworthy programs that need to be executed and thus an automated way of running a sandbox would be required.

THE TASK

The task is to create a sandbox tool running in windows that can be used in the testing of student code submissions, such as might be submitted as first, second or third year assignments.

In a Windows 10 system using .NET managed code there are two possible technologies that might be useful in developing a sandbox tool for untrusted code. Windows 10 introduced a built-in sandbox which launched a protected copy of Windows using the processors Hyper-V virtualization. This was upgraded to include controls that could, in a limited way, restrict the facilities available to the sandbox, however this is mainly designed for interactive use.

Prior to the Windows 10 virtualisation, the .NET managed code system had another method for achieving a sandbox capability in Windows 7. This capability is still available through .NET in windows 10.

The Microsoft online documentation has an example illustrating how to program a .NET tool in the article “How To: Run a Partially Trusted Code in a Sandbox” (http://msdn.microsoft.com/en-us/library/bb763046(v=vs.110).aspx) .

The Windows 10 Sandbox capability is described in various online articles such as:
https://techcommunity.microsoft.com/t5/Windows-Kernel-Internals/Windows-Sandbox/ba-p/301849

Originally the Windows 10 Sandbox was not configurable, but this feature was added later:
https://techcommunity.microsoft.com/t5/Windows-Kernel-Internals/Windows-Sandbox-Config-Files/ba-p/354902

Your task is to start with these examples and initially try out both a Windows 10 Sandbox and a .NET sandbox. Once you have evaluated their capabilities you should build a tool with an appropriate user interface that permits an executable program to be run with specified reduced access permissions. The tool might be useful in running student course work submissions during assessment. The user interface should allow the user to select a program and choose the relevant permissions. As a minimum the tools should be able to restrict an executing program's ability to access the file system, to access the network so any read or write to files or read or writes from the network can be detected.

It might be necessary to execute this tool from a script so the ability to execute from the command prompt may be useful. A windowed interface will be useful for interactive use of the tool for less experienced users. The example code supplied by Microsoft only illustrates how to execute managed code in a sandbox, which is satisfactory for this assigment. However, you may wish to explore how un-managed code could be executed in a sandbox environment, but this is not a requirement. The tool will be tested by executing itself in its sandbox as well as executing other samples of coursework. In particular you need to allow the code to read command line arguments and read from the keyboard and write to the console as in normal execution and permit that programs input and output to be handled as normal.

It is recommended that you get the example programs working first, and then expand this to add the additional features needed by your tool.

THE PACKAGING

Your submission of your code will be made electronically through Canvas. You should submit a single zip file containing your source code and a release executable for the tool you have created. You should not submit multiple versions of the tool; there should only be one copy of the source code and one single executable submitted. You should create a small user manual for your tool to explain its installation and use. You should submit the PDF of that documentation. The zip file should have a README file at the top level indicating the contents of the submission to enable the marker to find the necessary materials. Your tool should be able to run without the use of Visual Studio being installed.

You will also be required to demonstrate your tool operating in a demonstration slot which willl be announced nearer the time.

If the requirements are not clear there can be discussions of the assignment in class where you should ask for any further clarification of the requirements.