SEC204 Coursework

Computer Architecture and Low Level Programming

Aims: To create, interpret and manipulate IA32 assembly code via hardware debugging techniques. To apply reverse engineering techniques to identify main software flaws. To identify relevant countermeasures for main software flaws.

Task 1: Create a vulnerable IA32 Assembly program that receives a student name as input and calculates their score as the average of 2 randomly generated numbers between 1-100 each. Identify how one can cheat the program to receive the maximum score and discuss how the program can be strengthened accordingly.

Task 2: Reverse engineer the binary code to be provided on DLE, analyse what it does, identify and analyse any software vulnerabilities it might have and discuss how they can be fixed.

You are expected to work in pairs for this piece of coursework and perform all the tasks above. You will be expected to produce an IA32 Assembly file (.s) for task 1, and a written report to present your findings for tasks 1 and 2. The written report should not exceed 3,000 words and is expected to have an executive summary outlining your deliverables, main findings and recommendations. The assembly file should include basic running instructions for the end user as comments.

Submission information:

- You are asked to submit a single Zip file (.zip) containing the corresponding IA32 Assembly .s file for task 1, plus the written report for tasks 1 and 2. Your assembly file needs to be able to assemble and subsequently run on the Ubuntu-sec204 VM (linux ia32 environment).

- Your .s file is expected to contain basic running instructions for the end user. Comments explaining your code are optional, but desirable.

- This coursework is issued on the 28th October.

- The binary code for task 2 will be provided on DLE on the 12th November.

- Please email the module leader about your **group composition** by the **9th November 2018**. Groups composition to be confirmed by the 12th November.

- The Zip file containing the assembly code file and written report **must be submitted by the 10th January 2019, 4pm**. Coursework must be submitted by the specified deadline online via the DLE module website.

- Coursework submissions will be anonymous, please do not add any personally identifiable information in your submission.

- You should give due consideration to your personal time management to ensure that coursework is submitted in plenty of time prior to the deadline. The University cannot take any responsibility for late submission due to slow network speeds, etc.

- Coursework can be submitted at any time ahead of the deadline time. Please note that coursework, which is submitted after the deadline date and time will be capped at the minimum pass mark within the first 24 hours of the deadline and will be awarded a mark of zero if submitted more than 24 hours late.

- Extensions to deadlines for submission of coursework may not be granted by members of academic staff. A student who misses a deadline or believes that he or she will miss a deadline due to circumstances beyond her/his control should submit extenuating circumstances in accordance with these Regulations.

- You must correctly reference and cite all source materials. You are reminded of the University's rules on academic misconduct.

Assessment details and marking criteria:

It is worth 50% of the module mark. ***Relevant*** supporting information may be included as appendices if required. It will be expected to have an executive summary outlining your findings and recommendations. You are expected to support your claims by references.

Marking criteria				
Fail 0-40%	**3rd** 40-50%	**2:2** 50-60%	**2:1** 60-70%	**1st** 70%+
1F) Not all submission deliverables were met. Assembly code does not assemble	1P) All submission deliverables attempted. Assembly file assembles with limited functionality.	1M) All deliverables complete with good functionality.	1M) All deliverables complete with identifying fixes for security vulnerabilities.	1D) All deliverables complete with robust functionality.
2F) Applies general knowledge from course material with limited understanding	2P) Demonstrates basic understanding of assembly programming and reverse engineering	2M) Demonstrates good understanding of assembly programming and reverse engineering	2M) Demonstrates very good understanding of assembly programming, reverse engineering	2D) Demonstrates in-depth understanding of assembly programming and reverse engineering
3F) Little to no references to background literature	3P) Uses relevant background literature and material	3M) Occasional use of background literature to support writing	3M) Several uses of background literature to support writing	3D) Critical use of background literature to support writing
4F) Presentation is weak. The executive summary is missing. There are no user instructions and no code comments.	4P) Report presentation is basic, largely text-based. The executive summary is basic. Code comments provide user instructions	4M) Good presentation of report, with logical structure. Key points in the report are clearly highlighted in the executive summary. Code comments describe how the code works. Code comments provide user instructions	4M) Fulfil 4M) with emphasis on key points of report and discussion that flows well. Use of screenshots, figures, and captions. Clearly commented code and user instructions.	4D) Excellent presentation and well-documented report, which uses screenshots, figures, and captions to illustrate key points and justify findings. Clearly commented code and user instructions.
5F) Analysis of software vulnerabilities is flawed or unjustified.	5P) Analysis of software vulnerabilities is basic	5M) Fulfils 5P) with multiple solid concepts and methods.	5M) Analysis of software vulnerabilities with a methodical approach. Identification of appropriate software countermeasures.	5D) Extensive in-depth analysis of software vulnerabilities and identification of appropriate countermeasures.