This document is for Coventry University students for their own use in completing their assessed work for this module and should not be passed to third parties or posted on any website. Any infringements of this rule should be reported to facultyregistry.eec@coventry.ac.uk.

Faculty of Engineering, Environment and Computing

4061CEM Programming and Algorithms 1

Assignment Brief

Module Title	**Assignment Type**	**Cohort**	**Module Code**
Programming and Algorithms 1	Group	September	4061CEM
Coursework Title Local Enumeration and Privilege Escalation (Attempt 1)			**Hand Out Date** 24/10/2022
Lecturer Dr Ian Cornelius			**Due Date and Time** 21/11/2022 18:00
Estimated Time 50 Hours **Word Limit** Not Applicable		**Coursework Type** Group Programming Assignment	**Credit Value Assessed** 20

Submission Arrangement: Aula

File Types Accepted: `docx pdf`

Marks and Feedback Date: 05/12/2022

Feedback Method: TurnItIn

Module Learning Outcomes Assessed:

- Working with version control systems
- Use appropriate testing methodologies and tools
- Develop software based on a design

Preliminary Instructions

Tasks are to be undertaken in the Python programming language.

You will be expected to include comments in your code to explain the behaviour of your code and provide a justification of your algorithm selection. You are also strongly advised to test your code for compilation on a system other than your own, prior to submission. Non-compiling code will not pass, see the marking rubric for further information.

You will create a Coventry University GitHub **private** repository to store your source-code and manage version control of your work. Evidence of version control must be included in your regular commits to the repository over the period between the hand out date and due date.

Your eventual submission via TurnItIn will link to that repository which must include all of your source-code.

Task and Mark Distribution

The purpose of this task is for students to work in groups (a maximum of **five** people per group, but no less than **three**) to build a local enumeration and privilege escalation tool.

Local enumeration is concerned with analysing the target host for details such as:

- Usernames and group names
- Hostnames
- Network shares and services

This list is not exhaustive and can consist of more features. You can read this article for more information on what this tool may contain.

Privilege escalation is concerned with the increase of level access, either through a bug or design flaw. You can read this article for more information on what this tool will contain.

Assignment Tasks

To successfully complete this assignment you are required to complete the following tasks. Each task has a weight that is attributed towards a portion of the overall grade.

Remember, that each of these tasks will feed into a project (or piece of code) that will run cohesively.

Task 1: Basic Tool (15%)

The purpose of this task involves creating a basic tool that consists of a menu system that can perform **two methods of enumeration** and **one method of privilege escalation**.

As this is a basic task, it is not required to work on multiple operating systems, and it can be assumed that you will be developing it to run on the system that it is being developed within. The enumeration method will be expected to have a form of output displayed to the screen, whilst the privilege escalation should drop the user into a shell with an increased level of privilege.

Note: It will be expected that your privilege escalation method will not work on the markers machine, as such you will **need** to provide evidence (in the form of a video) of this working on your machine.

Task 2: Advanced Tool (30%)

The purpose of this task is to build upon the work you have done in task one and adding **six more methods of enumeration** and another **two methods of privilege escalation**.

Additionally, the tool should also determine which operating system it is running on and only suggest appropriate options of your implemented methods to the user. The methods you have implemented for enumeration and privilege escalation, at least one of these should work on a different operating system to the others.

Note: It will be expected that your privilege escalation method will not work on the markers machine, as such you will **need** to provide evidence (in the form of a video) of this working on your machine.

Task 3: Outputting (10%)

The purpose of this task is to build upon task one and two, and provide a more advanced logging method. Instead of outputting to the console, it is expected for your groups tool to output to a file. An option is to be added to the menu to change from outputting to the terminal, and to a file instead.

Note: it is best to ensure that the user has the ability to create a log file of a given name for a givne location. For example:

```
/home/ian/Desktop/log_enum.txt
```

Additionally, it is also expected for your tool to run with a parameter used in terminal. For example, instead of running `python ./LEAP.py`, it is expected that the user can use `python ./LEAP.py enumerate` to run the enumeration methods for the appropriate operating system the script is running on.

For example, if the user calls `python ./LEAP.py enumerate` on a Windows machine, it will run the scripts for Windows only. Whereas, if the same script was called upon Linux, it will only run the enumeration methods for Linux.

Task 4: Testing (15%)

The purpose of this task will build upon the work you and your group have done in tasks one to three. For each method implemented, you are expected to write a test case to ensure it is working as expected.

Note: It will be expected that the unit testing will be provided in a separate class file, and following the convention as shown in the lectures and labs.

Task 5: Version Control (5%)

The purpose of this task is to ensure that you are developing the tool iteratively and collaboratively using relevant version control features, such as forking, cloning and merging.

As this is a group project, you may want to consider using further functionality of the Git web-interface; i.e. the issue tracker and Wiki functionality.

For your submission, you are to include a URL to the **Coventry University GitHub** service repository where the groups solution resides, along with the **commit** tag you want marking.

Task 6: Documentation (10%)

To aid in the marking of the assignment, documentation is relevant for the tool you are implementing. For each function that a member has worked upon, it is recommended you included a description of the methodology followed along with the name and student identification number of who implemented the function. It may have been a collaborative effort on a function, therefore explain who worked upon which part.

Additionally, documentation will be required to explain how the tool works. Therefore, it is important to include a `README` file in the repository outlining how the tool works, a quick synopsis of functions that exist, and any test-cases that have been run/performed along with their outcomes.

You are also expected to provide relevant documentation with the source-code provided. The `README.md` file on the groups repository should contain information relating to the following:

1. An introduction to your software written for a user
2. A set of instructions on how to use your software
3. Provide a table on the unit tests you have performed
    - this will contribute towards the marks awarded for testing
4. A description of the algorithm(s) you have employed to solve the problem. This should include details for interesting or unusual choice or the problems solved.
    - you may want to include this part as comments within the actual source-code
5. The relevant videos showcasing the privilege escalation methods working on your machines
    - these should be provided as unlisted videos uploaded to YouTube

Task 7: Team Software Development (5%)

As this is a group assignment, you will be expected to work within a group of maximum **five** team members, but no less than **three** members. A portion of the marks have been allocated towards working as a group; and it will be expected to see that each group member has provided something towards the project.

The purpose of this task is to see how you can work as a team and how you can interact with code written by other group members. It will also demonstrate your ability to work with other code repositories and using advanced functionality of the version control system.

Task 8: Submission Guidelines (10%)

You will be expected to follow the submission guidelines, as outlined in the document below. Essentially, you are required to follow these rules:

1. Page One: Consists of a GitHub URL to the repository of your source-code
2. For each source-code file you have (`filename.py`), you need to provide:
    1. A single page, with the name of the file
    2. A single (or multiple) page(s) with the source-code in that file The Python code file(s) must be submitted, the screenshots will not be accepted.

It is easier to visualise what is meant by these rules by looking at the example document provided below.

> Submission Example

You will either be awarded zero marks, or full marks, depending on how you follow the guidelines.

Your source-code will be submitted to a plagiarism checker - so please ensure that any source-code acquired online is appropriately referenced. You are not to push or commit code to the GitHub repository after you have submitted your coursework. Timestamps will be checked, and if any changes made after the submission timestamp will not be marked.

Extensions and Deferrals

If you require an extension or deferral, you can find out more information at the following link:

> Information on Extensions/Deferral

Remember: You must supply evidence with your application. Failure to do so may result in the extension/deferral not being approved.

Plagiarism and Academic Misconduct Policy

Ensure that you are familiar with the university guidance on Plagiarism and Academic Misconduct. Any work found to be in violation of the rules will be submitted for an academic misconduct case.

Marking Allocation

0 - 39	40 - 49	50 - 59	60 - 69	70+	80+
Work mainly incomplete and/or weaknesses in most areas.	Most elements completed; weaknesses outweigh the strengths.	Most elements are strong; minor weaknesses.	Strength in all elements.	Most work exceeds the standard expected.	All work substantially exceeds the standard expected.

Marking Rubric

Task	Fail	Third	Lower Second	Upper Second	First
1	No attempt made.	There was an attempt to develop the required functionality, but it	There is some attempt at implementing a local enumeration	All functions have been implemented for local enumeration	All necessary components of the task have been implemented beyond

		has not been achieved.	function; however, there was no attempt at implementing a privilege escalation function.	alongside a function for privilege escalation.	expectations; with an output being provided on the local enumeration and a terminal window provided with an increased level of access.
2	No attempt made.	There was an attempt to develop the required functionality, but it has not been achieved.	There is some attempt at implementing a local enumeration function; however, there was no attempt at implementing a privilege escalation function.	All functions have been implemented for local enumeration alongside additional functions for privilege escalation.	All necessary components of the task have been implemented beyond expectations; the functions work across multiple operating systems (i.e. Windows, Linux and macOS).
3	No attempt made.	There is an attempt made towards outputting to a file; although the function does not behave as expected.	The provided function is able to output to a file, but it is done automatically without a user-prompt.	Functionality has been added to select whether the user would like the output to be displayed to the screen or a file. However, you are unable to run the tool with parameters from the command-line.	The logging function of the project provides the user with an option to destination of their choice, alongside running the tool with parameters from the command-line.
4	No attempt made for testing. There is little or no test-cases present.	There was an attempt at some testing, but it does not cover the obvious areas of the project.	The test cases that have been presented are for some functions; but do not cover all areas of the project.	There are a good range of test cases for each function in the project.	There is a full test-suite, with extensive coverage. In some cases, there is evidence of other types of testing.
5	No attempt to maintain an organised repository, with no history of iterative development. No comments have been provided in the source-code explaining the methodology.	Some evidence of version control; however, there is no evidence of iterative development. Some commentary have been provided, but It's little or not clear.	The repository is organised; however, it does not utilise the version control features well. There are clear comments, albeit not that in-depth.	The repository is organised and there is evidence of an iterative development process. The comments are clear with a good explanation of the methodology followed.	All contents of the version control are organised well, with repository features used correctly, i.e. branches and evidence of merging. Comments are in-depth and clearly explains the project undertaken.
6	No attempt made at providing documentation, apart from building upon the `README` file.	There was an attempt of providing some documentation, but it is incomplete and not following the correct formats.	The documentation has been provided in the correct format; but it may not cover all the areas or does not contain as much detail as expected.	The provided documentation is clear, concise and complete. It covers all areas of the project.	The documentation has been provided in a clear and readable format. It exceeds what is expected by providing examples and links to resources that are appropriate.
7	No evidence of contribution to another repository or having contributors to this repository.	There is some evidence of group work, but it is not clear in the organisation.	There is contributions towards another repository and contributions made from another team member.	There is clear documentation on how the plugins have been developed; there is also evidence of good practice with collaboration; i.e. the use of forks and pull requests.	Evidence of extensive collaboration can be seen, whilst maintaining clear evidence on who has contributed towards the project. There are advanced features of version control being used such as submodules to include contributed plugins.
8	Did not follow the guide-lines.	Did not follow the guide-lines.	Did not follow the guide-lines.	Did not follow the guide-lines.	Followed the guidelines correctly.

Notes

1. You are expected to use the Coventry University APA Referencing Style. For support and advice on this students can contact Centre for Academic Writing (CAW).
2. Please notify your registry course support team and module leader for disability support.
3. Any student requiring an extension or deferral should follow the university process as outlined here.
4. The University cannot take responsibility for any coursework lost or corrupted on disks, laptops or personal computer. Students should therefore regularly back-up any work and are advised to save it on the University system.
5. If there are technical issues that prevent students submitting coursework through the online coursework submission system on the day of a coursework deadline, an appropriate extension to the coursework submission deadline will be agreed. This extension will normally be 24 hours and will be communicated via your Module Leader.
6. Collusion between students (where sections of your work are similar to the work submitted by other students in this or previous module cohorts) is taken extremely seriously and will be reported to the academic conduct panel. This applies to both coursework and exam answers.
7. A marked difference between your writing style, knowledge and skill level demonstrated in class discussion, any test conditions and that demonstrated in a coursework assignment may result in you having to undertake a Viva Voce in order to prove the coursework assignment is entirely your own work.
8. If you make use of the services of a proofreader in your work you must keep your original version and make it available as a demonstration of your written efforts.
9. You must not submit work for assessment that you have already submitted (partially or in full), either for your current course or for another qualification of this university, except resits, where for the coursework, you maybe asked to rework and improve a previous attempt. This requirement will be specifically detailed in your assignment brief or specific course or module information. Where earlier work by you is citable, i.e. it has already been published/submitted, you must reference it clearly. Identical pieces of work submitted concurrently may also be considered to be self-plagiarism.