Instructions

Background:

Authorities were able to make an arrest against a known hacker group in the area. They have managed to recover a few files from the hard drives. One of the files they recovered was called `ctf.pyc` , but the authorities are not sure what is needed to view this file.

Rules of Engagement:

For this capture the flag challenge you will be exploring version control systems, Python3 syntax, Python3 package installers, basic ciphers, and various numbering systems.

All challenges can be solved using the command line and a web browser. While the tools used to solve the challenges are available on all operating systems, the process of installing and using each tool may differ slightly between OS's.

You may work with a team of other students in the class, but you must submit your own answers.

Scoring

Each challenge is worth 5 points for a total of 100 possible points. Keep in mind that some challenges build upon the correct answer from a previous challenge.

Tasks:

The authorities have asked you to attempt to figure out the following:

Command-line (interaction and open source intelligence gathering)

1. What is the utility used to interact with this repository?

2. Who created the utility you mentioned above?

3. What is the command using the utility above to clone this repo onto your local system?

4. If you were to make a change to this repo, what command would you use to add that change to your local system? (Hint: `commit`)

5. If you were to make a change to this repo, what command would use to push the local system changes to this repo? (Hint: `push`)

Enumeration and Exploitation

1. What programming language is the file written in?

2. What numbering system is the flag written in BEFORE being translated?

3. What is the numbering system that the flag is converted to?

4. What is the function `chr()` doing in this case?

5. What is the hidden flag value?

6. What is the value of the `use_me` variable?

7. What type of encoding was used for the `use_me` variable?

8. What cipher was used for the `use_this` variable?

9. How many shifts did you need to use to decode the variable?

10. The authorities realized through your help in viewing this code that information might be hidden in variables, and the attackers were using this methodology to share across platforms without the cleartext being shown. They think the `token_var` might contain useful information, but are unsure how to make

sense of it. They did find the value `egardoxxl` with this file. Using the information provided, what is the hidden flag?

Conversions (use the human-readable version of the flag unless otherwise specified)

1. Convert the first flag (in E&E: Q5) into hexadecimal.

2. Convert the first flag (in E&E: Q5) into binary representation.

Ciphers within Code

1. Within the file, there seems to be an odd `TODO` comment section. What is the flag contained within this section?

2. Relating to the above, what do the numbers represent that are concealing the flag? (Hint: think of the ascii code table)

3. If you were to convert the numbers in question 1 to characters in Python, what built-in method would you use? (e.g. `<name>()`)