CS2107 Assignment 2

Last Updated: 23 October 2025

Introduction

This assignment takes the form of an information security capture-the-flag (CTF) style competition. In a CTF, participants solve problems involving security weaknesses to bypass defences to obtain a sensitive piece of information called the `"flag"`.

In this assignment, participants are exposed to some of the common skills required to play in these competitions. When using the assignment platform, do not change your username.

CTF Rules and Guidelines

PLEASE READ THE FOLLOWING BEFORE BEGINNING

1. You are required to log in to CTFd (https://cs2107-ctfd-i.comp.nus.edu.sg) to submit flags. *If you are unable to access the site, use incognito mode.* **Make sure you are connected to SoC VPN!**
   - Connection is SSL-based, and login is with your NUS student account (Microsoft account).
   - If you need help setting up the VPN, do refer to the official docs here. The staff at `helpdesk@comp.nus.edu.sg` or the IT helpdesk in COM1, Level 1, will be able to help with you setting up if needed.
2. Do not attack any infrastructure not **explicitly authorised** in this document. This includes brute-forcing flag submissions.
3. Work **individually**. Discussion of concepts on the forum is allowed but refrain from posting solutions. Any sharing of answers detected will be reported and disciplinary actions will be taken.
4. Students may be randomly selected to explain how they obtain their flags, or else a zero mark will be given on their unexplainable challenges.
5. The skills taught in this assignment are not to be used on any system you do not own or have express permission to test. This is a **criminal offence** under the Singapore Computer Misuse and Cybersecurity Act.
6. Every challenge will contain a flag and will provide the accepted flag format. Please ensure your submissions meet the flag format stated **exactly**. This means include the `CS2107{}` portion unless otherwise stated.

Grading Scheme and Due Date

The assignment is due **16th November (23:59)**.

This is an individual assignment. You are encouraged to post questions on Piazza, provided they do not directly ask for the solution. Additionally, do not post the answers to the challenges.

Assignment 2 is divided into the following sections:

1. **Easy (15 points each):** Answer all challenges (45 points total)
2. **Medium (20 points each):** Of the 4 challenges, solve at least 2 (40 points maximum)
3. **Hard (10 points each + 5 points for writeup):** Of the 2 challenges, solve at least 1 (10 points maximum for solving, up to 5 additional points for the writeup)

The maximum number of points that can be obtained in this assignment is **100** (worth 10% of the total score for the entire course). **There are no partial marks for easy and medium challenges.** Solving challenges more than the intended maximum for medium/hard will not give you additional marks, but is still encouraged.

Writeups for hard challenges will be graded based on:

1. **Solution Accuracy**
2. **Demonstrated understanding of solution**
3. **Explanation of steps taken to solve**

You can be awarded partial marks for the writeup even if you did not solve the hard challenge fully.

A writeup is still necessary for every easy and medium challenge you solved or your challenge solve will be rendered invalid.

Scoring Examples

To illustrate how the point calculation is done, you can consider the following examples.

- Suppose Bob correctly answers all easy challenges, all 4 medium challenges, and 0 hard challenges.
    - Score: 45 + 40 + 0 = **85**
- Alice, meanwhile, correctly answers all easy challenges, 2 medium challenges, and 2 hard challenges (and gets full marks for writeup).
    - Score: 45 + 40 + 10 + 5 = **100**.
- Charlie also correctly answers all easy challenges, 3 medium challenges and 1 hard challenge. However, he did not submit any writeup for the hard challenge, and only wrote for the easy and medium challenges.
    - Score: 45 + 40 + 10 = **95**

If you do not submit any writeup at all, you will score 0!

Writeup Guidelines

Your writeups should **sufficiently share the approach** that you took in solving every problem. It should document your thought process, and failed attempts that led you towards the solution. Screenshots may be helpful in showing your steps too.

The following are some general guidelines to creating your writeup:

1. Please append challenge difficulty and number to the challenge name in the writeup, for example

> M.1: Rolling Thunder! <insert writeup>

2. You may link to any other scripts, writeups, guides etc. that you have referred to inside your writeup. However, a short explanation of the linked article is expected.
3. Concepts have to be explained clearly (this doesn't necessarily mean verbose!), however trivial they may be.

Writeup Examples:

- Decrypt the encryption algorithm... *(Don't do this! - Explain how to decrypt the encryption algorithm.)*
- This is a SQL injection challenge... *(Acceptable for Easy/Medium Challenges)*
- Since packet structures typically involve a fixed header and footer, packets with the targeted strings should have a larger payload and thus larger packet size. So let’s scan through the larger packets first by sorting the packets by size in Wireshark.... *(Ideal — thought process is explained and linked to steps taken to solve)*

As assignment 2 includes much harder topics than assignment 1, here are some example writeups that you may refer to for formatting, concept explanation or otherwise:

Pwn - https://blog.uhg.sg/article/14.html (not accessible with NUS wifi)

Web - https://nusgreyhats.org/posts/writeups/lakectf-finals-2022-carboncredit-suisse/

Reverse Engineering - https://hackmd.io/@tahaafarooq/picoctf-2024-reverse-engineering

Forensics - https://sunshinefactoryyy.notion.site/Forensics-1fdc708f7d9c80ef86d2fa3dff66e7a6

Submission Guidelines

Compress any additional files or solution scripts that you used while solving the problem, if any, into a zip file.

Within the zip file, submit each solution script as a **separate file** named after the challenge it applies to (e.g. e1_challengename.py). That is to say, you should not have 1 mega solution script for multiple challenges.

You are also expected to submit your writeups in PDF format with the following filename format: *StudentID_Name_WU.pdf* (e.g. A01234567_Alice Tan_WU.pdf)

Submit this writeup PDF as a separate file. Please do not submit it as part of a zip file.

Late submissions

Score penalties will apply for late submissions:

- Late up to 12 hours beyond due date: **10% penalty** to total score obtained
- Later than 12 hours but up to 36 hours beyond due date: **20% penalty** to total score obtained
- Later than 36 hours but up to 72 hours beyond due date: **30% penalty** to total score obtained

- 72 hours beyond the due date: **Submissions will not be entertained after 19th November (23:59)**

Contact

Please direct any inquiries about the assignment to

1. yitian@u.nus.edu (Cao Yitian)
2. kaixuan.lee@u.nus.edu (Lee Kai Xuan)
3. vincent.yeo@u.nus.edu (Yeo Beng Jun Vincent)
4. yuewei@u.nus.edu (Wu Yuewei)

Note that the TAs will **not** be debugging your code, but will only be around to discuss high level ideas. Do allow 3 working days for replies. Discussion on forums are highly encouraged.

Academic Honesty

NUS students are expected to maintain and uphold the highest standards of integrity and honesty at all times. As this is an **individual assignment**, please refrain from any forms of academic dishonesty.

If any form of plagiarism or cheating is found, you will be penalized and be subject to disciplinary action by the University. You may read more about NUS Student Code of Conduct here.

You are free to use or not use GenAI chatbots in this course in any way. There is no penalty for using GenAI chatbots. However, you are encouraged to first search for the answers yourself before resorting to these chatbots to maximise your learning. In addition, keep in mind that GenAI chatbots are not all-knowing - they are not always right!

Resources you may find helpful

Linux Environment

A Linux system is crucial for solving some of the challenges. It is expected that the participant has rudimentary proficiency in using a Linux system that can be gleaned by reading the tutorial at this link: https://www.digitalocean.com/community/tutorials/an-introduction-to-the-linux-terminal.

The nc Command

Throughout the assignments, if you see challenge with `nc aaa.bbb.ccc.ddd xxxx`, then it means that the challenge is hosted on the `aaa.bbb.ccc.ddd` server on `xxxx` port.

You can connect to the server by using the `nc` command in your terminal. In short, you can just copy & paste `nc aaa.bbb.ccc.ddd xxxx` and run it directly.

Make sure you are connected to SoC VPN!

Python3 Cheatsheet

Some challenges in the assignment might require some scripting to solve. Although you can use any programming languages you prefer, we recommend Python3.

To dynamically with interact with TCP server, we recommend the usage of pwntools

```
from pwn import * # Import pwntools

# Start a local process on local binary
r = process("./binary_name")
## OR ##
# Connect to cs2107-ctfd-i.comp.nus.edu.sg at port 15000
r = remote("cs2107-ctfd-i.comp.nus.edu.sg", 15000)

s = b'abcde'

r.sendline(s)                 # Send bytes s to the server
r.sendafter(b'message:', s)  # Send bytes s after received bytes
'message:'
r.recvline()                  # Receive a line from the server
r.recvuntil(b'Nonce: ')       # Receive until the bytes 'Nonce: ' from
the server
r.recvall()                   # Receive all bytes until EOF

####
# recv(), recvline(), recvuntil(), recvall() will not print
# any data by default. You have to pass them to print() to
# print their output.
# e.g. print(r.recvall())
####
```

Note that all the received message are in bytes. So you might have to the revelant conversions if necessary.

You can also change to debug mode by appending `level='debug'` as follows: `r = remote("123.123.123.123", 15000, level='debug')`

Here's a link to a cheatsheet:
https://gist.github.com/DavidTan0527/43edbf49fc550100a5a88d23627480ff

GNU Debugger (GDB)

You may need to use GDB to debug the app or your exploit code. We highly recommend installing either GEF OR pwndbg on your VM/Linux machine as the added features can greatly help make the debugging process easier (install only one, they will clash with each other).

You may find a GDB/GEF cheatsheet here (by trebledj) or here (by enigmatrix).

Pwndbg also has an official cheatsheet here, but commands should be very similar to GEF (and by extension GDB).

Acknowledgements

This assignment is a collective work of present and past teaching assistants, including Yuewei (AY25/26), Vincent Yeo (AY25/26), Kai Xuan (AY24/25, AY25/26), Jonathan Loh (AY24/25), Verity (AY24/25), River (AY24/25), Gaanesh (AY24/25), Yitian (AY23/24, AY24/25, AY25/26), Yong Liang (AY23/24), Ariana (AY23/24), Quang Vinh (AY23/24), Ashok (AY23/24), Arnav Aggarwal (AY23/24), Devesh Logendran (AY22/23, AY23/24), Akash (AY23/24, AY22/23), Sean Tay (AY22/23), Kel Zin (AY22/23, AY21/22), Weiu Cheng (AY22/23, AY21/22), Wen Junhua (AY22/23, AY20/21), Shawn Chew (AY 21/22), Chan Jian Hao (AY21/22), Ye Guoquan (AY21/22), Debbie Tan (AY20/21), Jaryl Loh (AY20/21, AY21/22), Daniel Lim (AY20/21), Chenglong (AY19/20), Shi Rong (AY17/18, AY19/20), Glenice Tan (AY19/20, AY18/19), Ngo Wei Lin (AY19/20, AY18/19), Lee Yu Choy (AY20/21, AY19/20, AY18/19, AY17/18), Nikolas Tay (AY 16/17) and Jeremy Heng (AY 16/17).