CS2107 Assignment 1

Last Updated: 20 September 2024

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. For password reset, it may take up to 5 working days.

Acknowledgements

This assignment is a collective work of present and past teaching assistants, including Verity (AY24/25), River (AY24/25), Gaanesh (AY24/25), Yitian (AY23/24, AY24/25), 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).

Grading Scheme and Due Date

This is an individual assignment. You are allowed to post questions on the Canvas Discussions forum but ensure that the questions do not ask for the solution. Additionally, do not post the answers to the challenges.

Assignment 1 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 total).
3. **Hard (15 points each):** Of the 2 challenges, solve at least 1 (15 points total).

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.** Solving challenges more than the intended maximum for medium/hard will not give you additional marks.

To illustrate how the point calculation is done, you can consider the following 2 examples. Suppose Bob correctly answers all easy challenges, 4 medium challenges, and 0 hard challenges. Bob obtains 45+40+0=85. Alice, meanwhile, correctly answers all easy challenges, 2 medium challenges, and 2 hard challenges. Alice obtains 45+40+15=100.

The assignment is due **6th October (23:59)**.

Penalties

Late submission of challenges

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 9th October (23:59)**

Other Penalties

Full marks for this assignment is **100**.

1. Submission of past flags (per flag): -10 pts
2. Late submission of Writeup (see more about its submission below): -10 pts
3. No source code (if necessary for completeness): -10 pts
4. Unclear Writeup:
    - Interview to explain solve
    - Unable to explain = -30% (of the relevant challenge)
5. Blank Writeups: -40% (of the relevant challenge)
    - Will also be asked for interview
    - Unclear writeup deduction will also apply (total -70% of relevant challenge)

Note that submitting a late flag beyond the due date will make your whole submission be considered as a late submission, and the mentioned score penalty scheme applies to your total score obtained.

Contact

Please direct any inquiries about the assignment to

1. yitian@u.nus.edu (Cao Yitian)
2. river@u.nus.edu (River Koh)
3. gaanesh.t@u.nus.edu (Gaanesh Theivasigamani)
4. verity_lim@u.nus.edu (Verity Lim)

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.

Rules and Guidelines

PLEASE READ THE FOLLOWING BEFORE BEGINNING

1. You are required to log in to CTFd to submit flags.
2. Do not attack any infrastructure not **explicitly authorised** in this document.
3. Multiple flag submission is permitted on the scoring platform without any penalty, but **no bruteforcing of flag submission on the server** will be tolerated.
4. Work **individually**. Discussion of concepts on the forum is allowed but refrain from posting solutions. The university takes plagiarism very seriously. Any sharing of answers detected will be reported and disciplinary actions will be taken.
5. Students may be randomly selected to satisfactorily explain how they obtain their flags, or else a zero mark will be given on their unexplainable challenges.
6. 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.
7. All challenges have a solution. They are guaranteed to be solvable with assistance of the internet and some research.
8. Ask the TAs for assistance only after you have exhausted every other avenue of self-help.
9. 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.

One of the most important skills in the information security field is the skill of seeking an answer independently. It is expected that the participant be able to utilise resources discovered through Google or any other search engine to achieve the tasks.

While the challenges might not be covered in entirety in class, the topics in the assignment are very applicable to security problems in real life. In the long run, the practical skills gained would benefit participants immensely.

Writeup and Submission Guidelines

A **writeup** documenting the approach you took in solving every problem is required as part of your assignment submissions.

This must be in PDF format with the following filename format:
StudentID_Name_WU.pdf (e.g. A01234567_Alice Tan_WU.pdf)
Submit this as a separate file. Please do not submit this PDF as part of a zip file.

Your submission should include all source codes and scripts that you used while solving the problem, if any.

Submit each script as a **separate file** named after the challenge it applies to (e.g. e1_challengename.py).

Note: Grades are not directly determined by this writeup. However, your writeup should **sufficiently share the approach** that you took in solving every problem. Include your thought process, or even any method you might have tried. Screenshots may be helpful in showing your steps too. Your writeup may be analysed and you may need to be interviewed by the teaching team to explain your steps at the TAs' discretion. This writeup also serves as a proof of your work in case the submission server malfunctions.

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 (but not limited to)
   M.1: AES-ECB
   H1: Broadcasting
   E.2 - xor_secure
2. We will not accept any writeups with just one-line explanations. Please explain your thought process in solving the challenges, as well as any concepts covered by the challenges.
   - For example (1 bad example followed by 2 good examples):
     - Decrypt the encryption algorithm... *(Don't do this! - Explain how to decrypt the encryption algorithm. Yes, we have received submissions like these.)*
     - In this reverse engineering challenge, the bytes are xor'd to produce... *(This is ok)*
     - This is a SQL injection challenge... *(This is ok)*

3. You may link to any other scripts, writeups, guides etc. that you have referred to inside your writeup. However, we expect a rough explanation of what the linked article is about.
4. **The concepts have to be explained clearly (this doesn't necessarily mean verbose!), however trivial they may be.**

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.

Linux Environment

A Linux system is crucial for solving some of the challenges, the challenges in this section will prepare you for the more advanced sections by presenting some elementary tasks to solve. 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.

However, more knowledge might be needed, and it is expected that the participant do some self-exploration.

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.

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, you can use pwntools

```
from pwn import * # Import pwntools

r = remote("123.123.123.123", 15000) # Connect to 123.123.123.123 at port 1

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

r.interactive() # Change to interative mode
```

Note that all the received message are in bytes. So you might to some conversion if necessary.

You can also change to debug mode with

`r = remote("123.123.123.123", 15000, level='debug')`

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

Easy Challenges (45 marks total)

Answer **all** challenges.

E.0 Sanity Check (0 marks)

A flag, written in our flag format, is placed somewhere in the assignment instruction file.

Try to find and submit it!

Flag format: `CS2107{...}`

E.1 Caesar Salad (15 marks)

I accidentally mixed a secret into the salad I made... Can you recover it for me?

IY8763{iG9y0X_7y_sE_l0b6aXoZ9_y0RgJ}

Flag format: `CS2107{...}`

Author: Verity (verity_lim@u.nus.edu)

E.2 Birds of a Feather (15 marks)

I heard birds are good messengers...?

The answer is in all CAPS and wrap it in the flag format (CS2107{...}) when submitting e.g. CS2107{MESSAGEHERE}

Flag format: `CS2107{...}`

Author: Verity (verity_lim@u.nus.edu)

E.3 Xoring gone wrong (15 marks)

You thought XOR-ing your message could keep it safe?! Think again.

Flag format: `CS2107{...}`

Author: Gaanesh (gaanesh.t@u.nus.edu)

Medium Challenges (40 marks total)

Answer **at least 2** challenges.

M.1 The Chain and The Weakest Link (20 marks)

I heard that AES-256 is said to be quantum resistant, so I have employed it in my code. With such a such a secure encryption algorithm in place, there's no way for people to crack my flag!

Flag format: `CS2107{...}`

Author: Yitian (yitian@u.nus.edu)

M.2 baby rsa (20 marks)

I wonder if the secret message can be decrypted?

Flag format: `CS2107{...}`

Author: Verity (verity_lim@u.nus.edu)

M.3 flag frequency (20 marks)

DO YOU KNOW ABOUT FREQUENCY ANALYSIS?

RN PBYNXARLMRB RGQA, TNXAQPBL RGB SNWWNDQXI SWMI: TA2107{E0V_S0VXP_7G3_SL3UVBXTE_NS_RG3_S14I}. HE ARVPEQXI RGB SLBUVBXTE NS WBRRBLA QX MX BXTLECRBP YBAAMIB MXP TNYCMLQXI RGBY RN RGB JXNDX SLBUVBXTE PQARLQHVRQNX NS WBRRBLA QX BXIWQAG, NXB TMX NSRBX PBRBLYQXB RGB AVHARQRVRQNX CMRRBLX VABP HE RGB TQCGBL.

Flag format: `CS2107{...}`

Author: River (river@u.nus.edu)

M.4 Diffie-Hellman Decipher (20 marks)

We've provided you with a simplified Diffie-Hellman Key exchange. Compute the private key using the provided public key and base. Then, use this private key to decipher the flag!

- The provided parameters are as follows:
  - Public Mod p: 991
  - Public Base g: 6
  - Public Key A: 299
  - Public Key B: 925
- A helper script has been provided for you as well.

Flag format: `CS2107{...}`

Author: Gaanesh (gaanesh.t@u.nus.edu)

Hard Challenges (15 marks total)

Answer **at least 1** challenge.

H.1 God Of RNG (15 marks)

Welcome to the ultimate game of chance! Feeling lucky today? Can you prove the cosmos wrong and hit the 0.0000001% chance for jackpot?

Challenge Instances:

```
nc cs2107-challs.nusgreyhats.org 8051
```

```
nc cs2107-challs.nusgreyhats.org 8052
```

```
nc cs2107-challs.nusgreyhats.org 8053
```

Flag format: `CS2107{...}`

Author: Yitian (yitian@u.nus.edu)

H.2 Re-al or Fa-ke (15 marks)

Is python's random number generator re-al or fa-ke? If you can guess the next number, I'll give you the flag.

Challenge Instances:

```
nc cs2107-challs.nusgreyhats.org 5051
nc cs2107-challs.nusgreyhats.org 5052
```

```
nc cs2107-challs.nusgreyhats.org 5053
```

Flag format: `CS2107{...}`

Author: River (river@u.nus.edu)

CS2107{good_job_finding_the_sanity_check!}