CS2107 Assignment 1

Last Updated: 13 February 2023

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 so do use a secure yet memorable password.

Acknowledgements

This assignment is a collective work of present and past teaching assistants, including Devesh Logendran (AY22/23), Sean Tay (AY22/23), Akash (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), 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 fourm but ensure that the questions do not ask for the solution. Additionally, do not post the answers to the challenges.

This assignment is worth 10% of the grade for the entire module. Assignment 1 is divided into the following sections:

1. **Easy (45 points):** Answer all challenges.
2. **Medium (45 points):** Answer all challenges.
3. **Hard (30 points):** Answer all challenges.
4. **Bonus (Bonus points):** Answer all the challenges to get bonus marks.

The maximum number of points that can be obtained in this assignment is 120. Solving the other bonus challenges can help you earn additional bonus points. Note that any bonus points earned in this assignment can be used, if needed, to top up your the 2 CTF assignments (20%).

The assignment is due **02 March 2023 (Thursday), 2359 HRS**. 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 05 March 2023 (Sunday), 2359 HRS**

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. devesh.logendran@u.nus.edu (Devesh Logendran)
2. sean.tay@u.nus.edu (Sean Tay)
3. wen_junhua@u.nus.edu (Wen Junhua)
4. c.akash@u.nus.edu (Akash Chandrasekaran)

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 (accessible only within NUS SoC Network) to submit flags.
2. You are **required** to upload a zip file to the "Assignment 1" folder on Canvas before the given deadline. The zip file should be named in the form of StudentID_Name.zip (e.g. A01234567_Alice Tan.zip) containing

- A **write up** documenting the approach you took in solving every problem. This must be in PDF format with the following filename format: **StudentID_Name_WU.pdf** (e.g. A01234567_Alice Tan_WU.pdf) Note that grades are not determined by this writeup. However, your writeup should **sufficiently share the approach** that you took in solving every problem. Screenshots may be helpful in showing your steps too. If there are suspicion on plagiarism, your writeup may be analysed and you may need to be interviewed by the teaching team to explain your steps. This writeup also serve as proof of your work in case submission server malfunctions.
- All source codes and scripts, if any, in their respective folder based on the challenge name.

3. Do not attack any infrastructure not **explicitly authorised** in this document.
4. Multiple flag submission is permitted on the scoring platform without any penalty, but **no bruteforcing of flag submission on the server** will be tolerated.
5. 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.
6. 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.
7. 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.
8. All challenges have a solution. They are guaranteed to be solvable with assistance of the internet and some research.
9. Ask the TAs for assistance only after you have exhausted every other avenue of self-help.
10. 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.

11. The challenges are tested from the NUS WiFi within the School of Computing and outside of NUS. Connectivity cannot be guaranteed anywhere else. **SoC VPN is required** if you are outside of school network.

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.

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.

Do note that you should use a 32-bit / 64-bit Linux environment to aid you in completing some of the challenges. Please also take note that if you are running 64-bit Linux, you may need to run the following commands in Linux to run 32-bit binary executables:

```
sudo dpkg --add-architecture i386
sudo apt-get update
sudo apt-get install -y libc6:i386
```

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.

If you wish to host a TCP server locally, you can use ncat `ncat -lvk -p 15000 -e "python3 main.py"`

Then connect to it with `nc localhost 15000`

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. This is because Python3 has many useful libraries (PyCryptodome) that can deal with Cryptography scheme.

Python3 differentiates string and bytes with the `b''` syntax

```
s1 = b'abcd' # Bytes of 'abcd'
s2 = 'abcd' # String of 'abcd'
```

Here are some of the useful commands that is supported natively in Python3:

- `s.encode()` - Convert string `s` into bytes
- `b.decode()` - Convert bytes `b` into string
- `b.hex()` - Convert bytes `b` to hex string
- `bytes.fromhex("01abcd")` - Convert the hex string `01abcd` to bytes
- `bytes([1,2,3])` - Convert integer list to bytes
- `list(b) - Convert bytes` - Convert `b` into a list of integers
- `i = 0x1235` - Set the value of `i` to be the value `0x1235`
- `i = int("1234ab", 16)` - Convert hex string `"1234ab"` to integer
- `pow(c, e, m)` - Calculate c^e mod m

For most of Cryptography library in python3, they require the plaintext to be bytes and not a string. This is because a string in python3 might have different encoding, but the encoding for bytes is universally UTF-8

Here are some of the useful commands that is supported in PyCryptodome:

- `Crypto.Util.number.long_to_bytes(m)` - Convert integer `m` to bytes
- `Crypto.Util.number.bytes_to_long(b)` - Convert bytes `b` to integer
- `Crypto.Util.Padding.pad(b, x)` - Pad bytes `b` so that the length is multiple of `x`
- `Crypto.Cipher.AES.new(key, AES.MODE_ECB)` - A new AES instances in ECB mode

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 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

r.interative() # 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')
```

Easy Challenges (45 marks)

Answer **all** challenges.

E.1 Sanity Check (15 mark)

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

Try to find and submit it!

Flag format: `CS2107{...}`

Author: Akash

E.2 Caesar Salad (15 marks)

I found this weird message in my salad. Can you help me figure out what it says?
`PF2107{Gurer_V5_n_YbG_Bs_Uby3f_Va_Gu1f_Ba3}`

Author: Junhua

E.3 Needle in MD5 Haystack (15 marks)

None of these images in `images.zip` are like the other ... but find me the one that matches `8ac8fb3045e78a65a9df5685fe715dff` (MD5). Submit the flag with this format: `CS2107{<file_name>_<sha1 of file>}`.

For example, if you think the file is `y4_f0und_m3_am1g0s_00000.jpg`, the flag is `CS2107{y4_f0und_m3_am1g0s_00000_a1d4c89c427a050d97cb7cdc0804a98c70caedaf}`

Author: Akash

Medium Challenges (45 marks)

Answer **all** challenges.

M.1 Frequency Analysis (15 marks)

I think my keycaps got switched around when I typed this essay. Can you find out what it's supposed to say?

Author: Sean

M.2 AES ECB (15 marks)

I've been told AES has never been cracked.

I took the liberty of removing the file header before encrypting it, and I only remember that the image was **940 pixels wide**.

Perhaps you could try your hand at recovering the message?

Author: Sean

M.3 Collide Me (15 marks)

The range of random numbers is many, so are your chances of happiness. Or is it?

Server: `nc cs2107-ctfd-i.comp.nus.edu 10520`

Author: Akash

Hard Challenges (30 marks)

Answer **all** challenges.

H.1 Password Hacking (15 marks)

Greyson Caterine has been our target for a very long time. She holds key information to solving the missing piece of the puzzle.

Can you help us retrieve her password based on the information she provided?

Our sources indicates that her password is made up 1 or more of the following with _ between them. Also, replace spaces in the password with _.

An example will be:

- `John Cena`
- `@JohnCena`
- `block 12 #01-12`

The password might be: `John_Cena_JohnCena_block_12_#01-12`

- Address
- First name
- Last name
- Phone number
- Favorite movie
- Dob in YYYYMMDD
- Social Media handle

We have the admin page and her resume. Can you help us retrieve her password and log into the system?

Admin Page: `http://cs2107-ctfd-i.comp.nus.edu.sg:5100` Resume website: `http://cs2107-ctfd-i.comp.nus.edu.sg:5101`

Author: Junhua

H.2 FASTencrypt Version Alpha (15 marks)

In year 8192, when days were bright, Archaeologists took up their sight. To Kent Ridge, where legends once spun, Of an institution, now lost and undone.

They dug and they searched, with all their might, But no buildings did they come upon in sight. Yet, in the dirt, a box they did find, With a lock so magical and so confined.

The lock, it would change every 35 ticks, Making cracking the code quite the tricky tricks. But the archaeologists, they wouldn't give in, They called for help, a cry for a win.

And you, a time-travelling cryptographer, Were called forth to be their helping factor. Can you break the code, before time does depart, And unlock the secrets of this mystical art?

```
nc cs2107-ctfd-i.comp.nus.edu.sg 5001
```

Author: Sean

Bonus Challenges (Bonus Marks)

Answer **all** challenges to get bonus marks.

B.1 Email Encryption

RSA is so secure, My prof always encrypts his emails using RSA.

Note: names.txt and mail.txt will not be provided

Author: Kel Zin (kelzin@u.nus.edu)

B.2 The Oracle

My movie buff friend wrote this authentication system for... something, I'm not sure what. I looked at it once and now my browser history is messed up because the URLs are so long.

Server: `http://cs2107-ctfd-i.comp.nus.edu.sg:5000/`

Author: Devesh

CS2107{let_the_games_begin}