CS2107 Assignment 1

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 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), 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 LumiNUS forum but ensure that the questions do not ask for the solution. Additionally, do not post the answers to the challenges.

This assignment is worth 15% of the grade for the entire module. Assignment 1 is divided into the following sections:

1. **Easy (75 points):** Answer all challenges.
2. **Medium (60 points):** Answer at least 3 challenges from the given 6 challenges to get 60 points; the points from answering medium-level challenges are capped at 60.
3. **Hard (15 points):** Answer at least one challenge; Solving the other challenge earns you 15 bonus points.

The maximum number of points that can be obtained in this assignment is 150. You only need to answer **one question** from Section C to be able to obtain full marks, but solving the other one 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 following CA components: 2 CTF assignments (30%) and 1 Group Presentation (5%).

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: 75+60+0=135. Alice, meanwhile, correctly answers all easy challenges, 2 medium challenges, and 2 hard challenges. Alice obtains: 75+40+30=145. Alice actually earns her 15 bonus points, which are then used to directly top up her A1 points.

The assignment is due **18 September 2022 (Sunday), 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 21 September 2022, 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. kelzin@u.nus.edu (Tan Kel Zin)
2. weiucheng.tan@u.nus.edu (Tan Weiu Cheng)
3. wen_junhua@u.nus.edu (Wen Junhua)
4. c.akash@u.nus.edu (Akash Chandrasekaran)
5. dcssu@nus.edu.sg (Prof. Sufatrio*)

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.

*: Please cc me if you email your queries about the given challenges; For issues with access to the CTFd server, please email your TAs.

Rules and Guidelines

PLEASE READ THE FOLLOWING BEFORE BEGINNING

1. You are required to log in to https://cs2107-ctfd-i.comp.nus.edu.sg/ (accessible only within NUS Network) to submit flags.
2. You are **required** to upload a zip file to the "Assignment > Assignment 1 > A1-supporting-files" folder on LumiNUS 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 (75 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 Shift Cipher (15 marks)

Julius Caesar was a smart man, yet not smart enough since he shifts each letter in his message by the same amount. For me, I shift my message a little differently depending on each letter's position in the message.

This is what I do: I first generate a random number 0 <= k <=26 ; then the letter at the i-th index of the flag is shifted $k+i$ times to the right (mod 26). Can you decrypt my ciphertext and tell me its plaintext?

Author: Weiu Cheng

E.3 AES Refresher (15 marks)

My `.jpg` image is encrypted in `AES-CBC`, can you recover the image if I tell you the following? - Key: `cs2107isveryfun!` - IV: `ThisIsJustAnIV!!` Please tell me the secret flag as shown in the image!

Author: Junhua

E.4 MAC Refresher (15 marks)

Find the MAC of the given file using HMAC with MD5, and put the MAC as the flag. The key used for the MAC is: `cs2107isTheBestModEver`

Flag format: `CS2107{mac_here}`

Author: Junhua

E.5 RSA Refresher (15 marks)

RSA is one of the most important public-key encryption algorithms. Do you know how to use it? Let me give you my *c*, *p*, *q* and *e* values, and please tell me the message *m*!

Author: Weiu Cheng

Medium Challenges (60 marks)

You may choose to answer **3 out of the 6** challenges from this section. Doing extra **will *not*** earn bonus points. However, you are welcome to answer more than 3 challenges, with the score capped at 60 marks.

M.1 One-Time Pad using My Keys (20 marks)

One-Time Pad is a secure cipher. But I don't like it if the key contains a '\x00' byte. This is since any byte XOR-ed with '\x00' is just itself. Hence, in my usage of One-Time Pad, I make sure that my keys have no '\x00', so you won't get to see ANY BYTE of the flag at all!!! I'm smart, right??

You can repeatedly connect to the server to get different ciphertexts of the same flag encrypted using One-Time Pad with a different key each time. For each encryption, I have used a strong random key (with the same length as the flag) from urandom, but with no '\x00' bytes in the key. Show me that you're smarter than me by finding out the flag!

Server: `nc cs2107-ctfd-i.comp.nus.edu.sg 5053`

Author: Weiu Cheng

M.2 Small Key Space(20 marks)

The following ciphertext is from an encryption with `AES-CTR` mode:
`7d294f3c7a71e2808e8ce5afd9eb99c343460ca9f5f89ff062fed96b` (in hex).

But, I only used 8-digit number as the key, and the following IV: `0000000000000000`.

Now, I'm not sure if it is secure enough. Can you decrypt the ciphertext, and get the plaintext?

Author: Junhua

M.3 Birthday Hash (20 marks)

I dare to challenge you to find a collision for my hash algorithm!

Server: `nc cs2107-ctfd-i.comp.nus.edu.sg 5052`

Author: Kel Zin

M.4 Phishing for Free Request (20 marks)

I want to create a program for my tool. However, I do not want to do it myself. Luckily for me, I've found this organization that can do it for me FOR FREE, yay! :D.

Now, how would I go about making that free request.... Their company website is here.

I heard there is a higher chance of free software if we mention one of their employees.

To achieve my goal, maybe I should send them a phishing message on Telegram based on the information I can find online?

Author: Junhua

M.5 AES ECB (20 marks)

I'm wondering why nobody is using AES-ECB. Since AES is secure, my intuition is that it should be pretty secure.

The server provides an encryption oracle for AES ECB. Hence, you can query plaintexts of your choice to the server to get the corresponding ciphertexts. Tell me if you can see the FLAG appended to your chosen plaintext.

Server: `nc cs2107-ctfd-i.comp.nus.edu.sg 5051`

Author: Kel Zin

M.6 RSAgain? (20 marks)

RSA again??? Since you already know how it works, there shouldn't be a problem solving this... right? Now, you can have my c, e, n and $7p-3q$ values. Please tell me the message m this time again!

Author: Weiu Cheng

Hard Challenges (15 marks + 15 marks)

You may choose **1 out of 2** challenges to solve. Solving both challenges earns you 15 points as bonus marks for CAs.

H.1 Securely-Encrypted Emails (15 marks)

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

Author: Kel Zin

H.2 Criminal (15 marks)

This criminal organisation doing sus... Help us gain access to their server.

Server: `nc cs2107-ctfd-i.comp.nus.edu.sg 5054`

Author: Kel Zin

CS2107{let_the_games_begin}