CS2107 Assignment 2

Last Updated: 29 March 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 Yitian (AY23/24), 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 are not directly asking for a solution. Additionally, do not post the answers to the challenges anywhere.

Assignment 2 is divided into the following sections:

1. **Easy (10 points each):** Answer all challenges (40 points total).
2. **Medium (20 points each):** Of the 4 challenges, solve at least 2 (40 points total).
3. **Hard (20 points each):** Of the 4 challenges, solve at least 1 (20 points total).

The maximum number of points that can be obtained in this assignment is **100** (worth 15% of the total score for the entire course). 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 CTF assignment. **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 `40 (4x10 points) + 40 (2x20 points) + 0 (0x20 points) = 80 points`.
Alice, meanwhile, correctly answers all easy challenges, 2 medium challenges, and 2 hard challenges. Alice obtains `40 (4x10 points) + 40 (2x20 points) + 20 (1x20 points) =100 points`.

The assignment is due **21 April 2024 (21:07)**.

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 24 April 2024 (21:07)**

Other Penalties

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

1. Submission of past semesters' assignment flags (per flag): -10 pts
2. Late submission of assignment 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: Submitting a late flag beyond the due date will cause your whole submission to be considered as a late submission, and the mentioned score penalty scheme would apply to your total score obtained.

Contact

Please direct any inquiries about the assignment to **CS2107 Canvas discussion forums**.

In case of any queries **specific to you**, such as account recovery, special requests etc, feel free to email any of the assignment TAs.

1. ariana@u.nus.edu (Ariana Goh Zhi Hui)
2. yitian@u.nus.edu (Cao Yitian)
3. yongliangang@u.nus.edu (Ang Yong Liang)
4. nqvinh@u.nus.edu (Nguyen Quang Vinh)

Note that the TAs will **not** be debugging your code, but will only be around to discuss high level ideas. Do allow up to 3 working days for replies. Discussion of concepts on the forums is highly encouraged. There are no penalties for discussing concepts on the forums!

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 the required files to Canvas **as well as** submit the flags to the CTF website. Any part of this missed will void your submission for the affected challenge.
3. Do not attack the CTF infrastructure in any way 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. 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. **Work individually**.
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. **Once again, writeups should not just include 1 sentence for the explanation.**
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 some research and practical application of concepts. All challenges on the server have been tested by the TAs and verified working. However, do feel free to raise up any issues on Canvas forums.
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 to include the `CS2107{}` portion unless otherwise stated. *If a flag is not enclosed by* `CS2107{}`*, you may append it to the flag*.
10. 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 *knowing how to search for information and what to search for*. It is expected that the participant be able to utilise resources discovered through Google or any other search engine to achieve the tasks. In fact, we don't ban the use of ChatGPT or any other generative AI :)

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

This will be more thoroughly enforced for Assignment 2. Please read the following carefully when crafting your writeup.

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.**

Here are some example writeups that you may refer to for formatting, concept explanation or otherwise:
Pwn - https://ultimatehg.github.io/article/14.html
Web - https://nusgreyhats.org/posts/writeups/lakectf-finals-2022-carboncredit-suisse/
Reverse Engineering - https://blog.caprinux.com/posts/ctfs/2023/wgmy2023/
Forensics - https://blog.qz.sg/wireshark-ctfs-writeup-tryhackme-part-1-of-2/

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 (for pwn challenges, or otherwise), 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

GDB Cheatsheet

To aid in solving System Security (also known as binary exploitation / pwn) challenges, we've also prepared a GDB (GNU Debugger) cheatsheet here: https://gist.github.com/Enigmatrix/89b09b4c97d541df3dd9e0d8ace9ed1a

The purpose of GDB is to monitor the state of the program (directly view memory and register values, pause at certain conditions etc) so that you can see whether the input you have send is correctly influencing the execution of the program.

Any other cheatsheets or reference sheets are provided in the respective hint sections of each challenge. Do consult them if you find yourself stuck.

Easy Challenges (40 marks total)

Answer **all** challenges for full marks.

E.1 Can you GDB? (Reverse Engineering)

I heard dynamic analysis makes reverse engineering trivial...

Hint

- Refer to the GDB info section above if you find yourself stuck!

Author: Ariana Goh Zhi Hui (ariana@u.nus.edu)

E.2 keylogger (Forensics)

Someone has been spying on our corporate network, we suspect that they have installed a keylogger on our system.

Hint

- HID DATA
- Universal Serial Bus HID Usage Tables
- Maybe a Keyboard/Keypad map can help you decode easier

Author: Ang Yong Liang (yongliangang@u.nus.edu)

E.3 Babypwn (Application Security: Binary Exploitation/Pwn)

I read about buffer overflows, and I'm not too sure what that means. Can you check my application for me? It simply asks for your name and decides if you have permissions...

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one)

```
nc cs2107-ctfd-i.comp.nus.edu.sg 5051
nc cs2107-ctfd-i.comp.nus.edu.sg 5052
nc cs2107-ctfd-i.comp.nus.edu.sg 5053
```

Hint

- This is a buffer overflow challenge, recognise your input variable and visualise your stack layout! Here is a guide on buffer overflow that you may refer to. Do google it yourself for further clarification if required. Use `gdb` to debug the application and visualise the stack.
- Tip: `x/#x $rsp` will display # words (8-bytes) starting from the top of your stack. For example, `x/30x $rsp` displays 30 hexadecimal numbers worth of bytes from the top of your stack.

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

E.4 Cat Facts (Web Security)

I have just created an application to give you some cat facts. Give a lucky number and we will return a corresponding cat fact! There is a hidden table in my database with a reward waiting for the worthy cat lover.

The core logic source code is provided.

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one. You might need to use incognito mode to connect to the challenges)
http://cs2107-ctfd-i.comp.nus.edu.sg:8071
http://cs2107-ctfd-i.comp.nus.edu.sg:8072
http://cs2107-ctfd-i.comp.nus.edu.sg:8073

Hint

- SQL Injection
- SQLite Injection Cheat Sheet and The Schema Table might be useful
- When you try to discover the name of the secret table, your query should exclude the name of the table used in the source code.

Author: Nguyen Quang Vinh (nqvinh@u.nus.edu)

Medium Challenges (40 marks total)

Answer **at least 2** challenges for full marks.

M.1 exfiltrator64 (Forensics)

Blue teamers managed to capture the traffic of the compromised machine. Your task is to investigate the pcap file and determine what was exfiltrated out of the network. Oh wait, someone told me that a zip file containing our secret code and classified images has been exfiltrated... but weirdly, we don't see it in the pcap?

Hint

- https://www.trenchesofit.com/2020/08/01/data-exfiltration-with-base64/
- Maybe there is a faster way to extract what you need using `scapy` or a script.
- URL encoding converts characters into a format that can be transmitted over the Internet, you might want to url decode what you found before doing further actions.

Author: Ang Yong Liang (yongliangang@u.nus.edu)

M.2 Stacksweeper (Application Security: Binary Exploitation/Pwn)

We have intercepted a malicious package from some threat actors, but they were smart and redacted all their variable declarations! I've heard about certain techniques that allow you to overwrite variables, and a little birdie told me that you were the best around here. Could you help me cross this minefield of a stack?

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one)

```
nc cs2107-ctfd-i.comp.nus.edu.sg 5054
nc cs2107-ctfd-i.comp.nus.edu.sg 5055
nc cs2107-ctfd-i.comp.nus.edu.sg 5056
```

Hint

- This is a buffer overflow challenge. Interpret what the code is doing and return to the win function. This is a buffer overflow to ret2win challenge.
- You may refer to the following guide on ret2win for the completion of the exploit chain.

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

M.3 Cat Breeds (Web Security)

To accomodate cat lovers, I have created an application to check if a cat breed exists. Again, I have hidden the prize somewhere on the server. The core logic of the application is provided.

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one. You might need to use incognito mode to connect to the challenges)
http://cs2107-ctfd-i.comp.nus.edu.sg:8081
http://cs2107-ctfd-i.comp.nus.edu.sg:8082
http://cs2107-ctfd-i.comp.nus.edu.sg:8083

Hint

- Blind SQL Injection
- The flag on the server is in the "flags" table with two columns id, flag
- The flag follows the CS2107{...} format, and only include alphanumeric characters, and the character "_".

Author: Nguyen Quang Vinh (nqvinh@u.nus.edu)

M.4 Ghidra Dragon Breath (Reverse Engineering)

I have learnt my mistake from last time and now no more dynamic analysis for you!

Hint

- Decompile the binary, then perhaps rewrite the code in another language...

Author: Ariana Goh Zhi Hui (ariana@u.nus.edu)

Hard Challenges (20 marks total)

Answer **at least 1** challenge for full marks

H.1 Headlines (Web Security)

I have created an application for you to generate catchy headlines using Flask templates. A reward is waiting for whoever can discover the secret on the web server!

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one. You might need to use incognito mode to connect to the challenges)
http://cs2107-ctfd-i.comp.nus.edu.sg:8091
http://cs2107-ctfd-i.comp.nus.edu.sg:8092
http://cs2107-ctfd-i.comp.nus.edu.sg:8093

Hint

- For the novice writer who seek to have guidance getting started: Server Side Template Injection

Author: Nguyen Quang Vinh (nqvinh@u.nus.edu)

H.2 Life Of Pie (Application Security: Binary Exploitation/Pwn)

I love pies! I also love this philosophical story called Life of Pi, so I named my pie factory after this story! Feel free to order any pie you want, and I'm so generous I'm even providing the recipe for our signature pie flavour for free! However, I do have some secrets up my sleeves. Surely you can't do anything like finding my secret PIE with a leaked runtime memory address...

(Connect to any one of the 3. If one is laggy/unresponsive, try the next one)

```
nc cs2107-ctfd-i.comp.nus.edu.sg 5057
nc cs2107-ctfd-i.comp.nus.edu.sg 5058
nc cs2107-ctfd-i.comp.nus.edu.sg 5059
```

Hint

- This is a buffer overflow challenge, with PIE enabled (Position Independent Executable). PIE is a random offset that is generated on runtime and added to all relative offset memory addresses. The generated PIE value remains constant for the same instance of the application.
- The exploit chain should end similarly to ret2win. Here are some supplementary guides on PIE and ret2win to aid you in completing your exploit chain.

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

H.3 Lost Bitcoin Key (Reverse Engineering)

I lost my bitcoin key and my key generation program is taking too long, can you help me recover it?

Hint

- I heard you could patch binaries with some existing tools...

Author: Ariana Goh Zhi Hui (ariana@u.nus.edu)

H.4 Presentation (Forensics)

The PowerPoint file in our office serves multiple purposes, but what is it hiding?

Hint

- Maybe powerpoint slides is another type of file?
- PDF structure consists of the header, body, xref, and trailer: https://pypdf2.readthedocs.io/en/3.0.0/dev/pdf-format.html

Author: Ang Yong Liang (yongliangang@u.nus.edu)