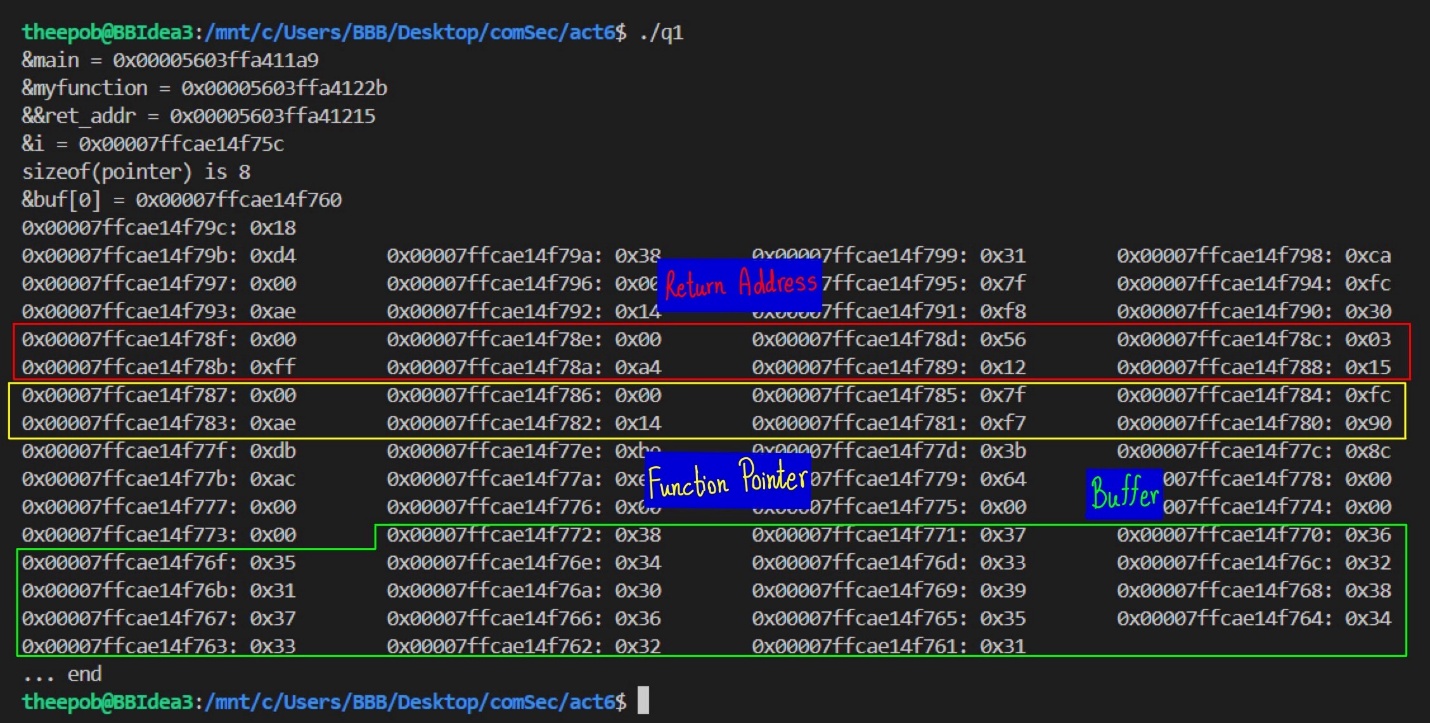
1.  


2.



3.



4. Bonus: From exercise 2 and 3, can you explode the buffer-overflow attack even when the canary-style protection is activated? Please explain your analysis.

ANS ได้ ถ้ายังคงสามารถรักษาค่าเดิมของ canary เอาไว้ได้ ซึ่งค่อนข้างยากเพราะอาจจะมีการเข้ารหัส

5. Question: Now you have mastered a type buffer-overflow attack. Please answer the following questions.

● Most viruses and worms use buffer overflow as a basis for its attack.

ANS YES.

● Do you think that exploiting buffer-overflow attacks is trivial? Please justify your answer. (i.e. Is it trivial to write a program to exploit buffer-overflow attacks in a server ?)

ANS ไม่ เพราะ บนระบบจริง/สมัยใหม่ ต้องเผชิญ ASLR, NX, canary, PIE, CFI — ต้องใช้เทคนิคขั้นสูง (info leak, ROP) และเวลา/ความเชี่ยวชาญ

● As a programmer, is it possible to avoid buffer overflow in your program (write secure code that is not vulnerable to such attack)? Explain your strategy

ANS ได้ แต่ต้องใช้หลายชั้นของการป้องกัน เช่น

* ใช้ภาษา memory-safe (Rust, Go) ถ้าเป็นไปได้
* หลีกเลี่ยงฟังก์ชันไม่ปลอดภัยใน C (gets, strcpy ฯลฯ) และใช้ฟังก์ชันแบบมี bounds checks
* ตรวจ input และเช็คขนาดก่อนเขียนลงบัฟเฟอร์
* เปิดการป้องกันของคอมไพเลอร์/OS (-fstack-protector, -D\_FORTIFY\_SOURCE=2, PIE, ASLR, NX)
* ทำ static analysis, fuzzing และใช้ sanitizers ในการพัฒนา
* รัน service ด้วยสิทธิ์จำกัด