Zadanie 4 – BEZPEČNÉ PROGRAMOVANIE

Vypracovali: Samuel Michalčík, Marek Štrba

1. Inštalácia

Na statickú analýzu sme použili nástroj cppcheck a na dynamickú analýzu kódu sme využili AFL.

- Inštálácia cez cmdline:
  brew install cppcheck

2. Analýza zraniteľností cez cppcheck a manuálne testovanie kódu

V adresári s main.c súborom projektu sme spustili príkaz:
cppcheck --enable=all main.c

Nájdené zraniteľnosti:

- [main.c:10]: (warning) Recursive call to 'stack_operation' without termination condition leads to stack overflow.
- [main.c:27]: (warning) Buffer 'img.header' may not be null-terminated when used with '%s'.
- [main.c:27]: (warning) Buffer 'img.data' may not be null-terminated when used with '%s'.
- [main.c:29]: (warning) Variable 'size1' may be negative; passing negative value to 'malloc' as size parameter.
- [main.c:32]: (error) Buffer overflow: 'buff1' may not be large enough to hold 'img.data'.
- [main.c:34]: (error) Possible double free of 'buff1'.
- [main.c:36]: (error) Use after free: 'buff1' is accessed after it was freed.
- [main.c:39]: (warning) Variable 'size2' may be negative; passing negative value to 'malloc'.
- [main.c:42]: (error) Buffer overflow: 'buff2' may not be large enough to hold 'img.data'.
- [main.c:51]: (warning) Possible division by zero in 'size3 = img.width / img.height'.
- [main.c:56]: (error) Out of bounds access: 'buff3' index 'size3' may be out of range.
- [main.c:57]: (error) Out of bounds access: 'buff4' index 'size3' may be out of range.
- [main.c:55]: (warning) Uninitialized variable 'OOBR_heap' used.
- [main.c:65]: (warning) Variable 'size4' may cause integer overflow in multiplication.

- [main.c:68]: (error) Infinite recursion detected in 'stack_operation()'.
- [main.c:70]: (error) Potential infinite loop due to memory allocation in 'do-while' loop without proper exit condition.
- [main.c:24]: (warning) Return value of 'fread' is not checked; potential incomplete read.
- [main.c:28]: (warning) Format string vulnerability: unchecked user input in 'printf'.
- [main.c:77]: (warning) Possible null pointer dereference: 'argv[1]' may be null.
- [main.c:19]: (warning) Resource leak: 'fp' is not closed on all error paths.

3. Popis a riešenie 5-tich vybraných zraniteľností

- **[main.c:10]:**
  - **Riadok:** 10
  - **Typ problému:** Varovanie
  - **Problém:** Funkcia stack_operation() sa volá rekurzívne bez akejkoľvek ukončovacej podmienky, čo vedie k nekonečnej rekurzii a následnému pretečeniu zásobníka.
  - **Opravený kód:**
    - void stack_operation(int count)
    - {
    - if (count <= 0) return;
    - char buff[0x1000];
    - (void)buff;
    - stack_operation(count - 1);
    - }

- **[main.c:27]:**
  - **Riadok:** 27
  - **Typ problému:** Varovanie
  - **Problém:** Pole img.header má pevne stanovenú veľkosť 4 bajty a nemusí byť ukončené nulovým znakom. Použitie %s v printf očakáva nulou ukončený reťazec, čo môže viesť k čítaniu mimo hraníc poľa.
  - **Opravený kód:**
    - struct ImageData {
    - char header[5]; // Zvýšená veľkosť na 5
    - int width;
    - int height;
    - char data[10];
    - };
    -

        - // Po načítaní img.header
        - img.header[4] = '\0';
- **[main.c:27]:**
    - **Riadok:** 27
    - **Typ problému:** Varovanie
    - **Problém:** Pole img.data nemusí byť ukončené nulovým znakom. Output s %s môže čítať mimo hraníc poľa, čo vedie k nedefinovanému správaniu.
    - **Opravený kód:**
        - struct ImageData {
        - char header[5];
        - int width;
        - int height;
        - char data[11]; // Zvýšená veľkosť na 11
        - };
        -
        - // Po načítaní img.data
        - img.data[10] = '\0';
- **[main.c:29]:**
    - **Riadok:** 29
    - **Typ problému:** Varovanie
    - **Problém:** Premenná size1 je vypočítaná ako img.width + img.height. Ak sú tieto hodnoty záporné alebo dôjde k pretečeniu, size1 môže byť záporné, čo vedie k neočakávanému správaniu pri predaní funkcii malloc.
    - **Opravený kód:**
        - int size1 = img.width + img.height;
        - if (size1 <= 0 || size1 > MAX_ALLOWED_SIZE) {
        - // Ošetrenie chyby
        - return -1;
        - }
        - char *buff1 = (char *)malloc(size1);
- **[main.c:32]:**
    - **Riadok:** 32
    - **Typ problému:** Chyba
    - **Problém:** Kód kopíruje sizeof(img.data) bajtov do buff1 bez kontroly, či je buff1 dostatočne veľký. Ak je size1 menšie ako sizeof(img.data), dôjde k pretečeniu bufferu.
    - **Opravený kód:**
        - if (size1 >= sizeof(img.data)) {
        - memcpy(buff1, img.data, sizeof(img.data));
        - } else {
        - // Ošetrenie chyby

        - free(buff1);
        - return -1;
        - }
- **[main.c:34]:**
    - **Riadok:** 34
    - **Typ problému:** Chyba
    - **Problém:** Ukazovateľ buff1 je uvoľnený a potom potenciálne uvoľnený znova, ak size1 % 2 == 0, čo vedie k zraniteľnosti dvojitého uvoľnenia pamäte.
    - **Opravený kód:**
        - if (buff1)
        - {
        - if (size1 % 3 == 0)
        - {
        - buff1[0] = 'a';
        - }
        - free(buff1);
        - }
- **[main.c:36]:**
    - **Riadok:** 36
    - **Typ problému:** Chyba
    - **Problém:** Po uvoľnení buff1 kód pristupuje k buff1[0], čo vedie k chybe použitia pamäte po jej uvoľnení.
    - **Opravený kód:**
        - if (buff1)
        - {
        - if (size1 % 3 == 0)
        - {
        - buff1[0] = 'a';
        - }
        - free(buff1);
        - }
- **[main.c:39]:**
    - **Riadok:** 39
    - **Typ problému:** Varovanie
    - **Problém:** Premenná size2 je vypočítaná ako img.width - img.height + 100. Môže byť záporná, čo spôsobí neočakávané správanie pri predaní funkcii malloc.
    - **Opravený kód:**
        - int size2 = img.width - img.height + 100;
        - if (size2 <= 0 || size2 > MAX_ALLOWED_SIZE) {

        - // Ošetrenie chyby
        - return -1;
        - }
        - char *buff2 = (char *)malloc(size2);
- **[main.c:42]:**
    - **Riadok:** 42
    - **Typ problému:** Chyba
    - **Problém:** Podobne ako pri buff1, buff2 nemusí byť dostatočne veľký na uloženie sizeof(img.data) bajtov, čo vedie k pretečeniu bufferu počas memcpy.
    - **Opravený kód:**
        - if (size2 >= sizeof(img.data)) {
        - memcpy(buff2, img.data, sizeof(img.data));
        - } else {
        - // Ošetrenie chyby
        - free(buff2);
        - return -1;
        - }
- **[main.c:56]:**
    - **Riadok:** 56
    - **Typ problému:** Chyba
    - **Problém:** Index size3 môže prekročiť hranice poľa buff3[10]. Prístup k buff3[size3] môže viesť k čítaniu/zápisu mimo hraníc, ak size3 nie je v rozsahu [0, 9].
    - **Opravený kód:**
        - if (size3 >= 0 && size3 < sizeof(buff3)) {
        - char OOBR = buff3[size3];
        - buff3[size3] = 'c';
        - } else {
        - // Ošetrenie chyby
        - }