Петров Д.М. КІТ -М120Б

Мета

Дослідити можливі вразливості переповнення буферу.

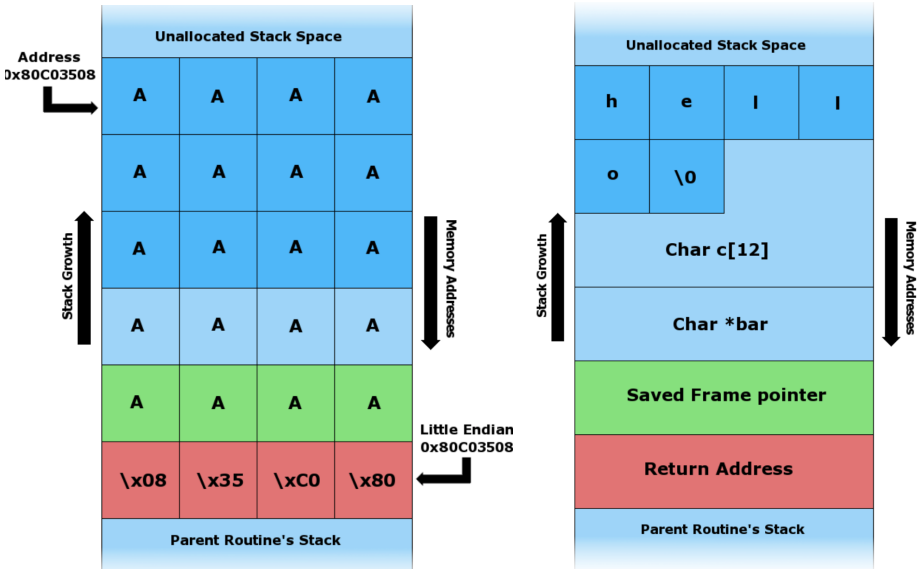
Завдання:

Взяти за основу наведений текст програми.

ХІД РОБОТИ

В галузі комп'ютерної безпеки і програмування, переповнення буфера (англ. buffer overflow або англ. buffer overrun), це явище, при якому програма, під час запису даних в буфер, перезаписує дані за межами буфера. Це може викликати несподівану поведінку, включно з помилками доступу до даних, невірними результатами, збоєм програми або дірою в системі безпеки. Переповнення буфера може бути викликане недостатньою перевіркою вхідних даних. Воно є базою для багатьох уразливостей в програмних продуктах і може бути злонамірено використане. Додаткова перевірка може запобігти переповненню буфера, хоча така перевірка відіб'ється на швидкодії програми. Мови програмування, зазвичай згадувані у зв'язку з переповненням буфера, це здебільшого C та C++. Вони не мають вбудованого механізму проти доступу або перезаписування даних у будь-якій частині пам'яті і не провадять автоматичної перевірки даних, які записують в масив, на вихід за межі масиву.

В програмах, переповнення буфера стеку трапляється, коли програма пише за адресами в програмному стеку виклику поза призначеною структурою даних; це зазвичай буфер фіксованої довжини. Баг переповнення буфера стеку трапляється, коли програма пише більше даних в буфер розміщений в стеку, ніж було фактично виділено місця для буфера. Це майже завжди призводить до псування прилеглих даних в стеку, і в разі якщо переповнення було зроблено помилково, часто призводить до краху програми або некоректної роботи. Цей тип переповнення є одним з випадків загальнішого класу багів програмування, знаних як переповнення буфера.



Якщо атакована програма виконується з спеціальними привілеями, або приймає дані з недовірених хостів мережі (напр. вебсерверів), тоді баг є потенційною вразливістю безпеки. Якщо стековий буфер залитий даними, які надійшли від недовіреного користувача, тоді цей користувач може пошкодити стек в такий спосіб, що в стеку опиняється виконуваний код, інжектований ним, відтак він отримує управління процесом. Це один з найстаріших і найнадійніших методів для зловмисників отримати неавторизований доступ до комп'ютера.

Беремо код вихідний програми

```
bool isPasswordOk() {
    char password[12];
    gets(password);
    return 0 == strcmp(password, "goodpass");
}

int main(int argc, char* argv[]) {
    bool status;
    puts("Enter password: ");
    status = isPasswordOk();
    if (!status) {
        puts("Access denied");
        exit(-1);
    }
    puts("Passed");
    return 0;
}
```

Далі відкриваємо дизасембльований код

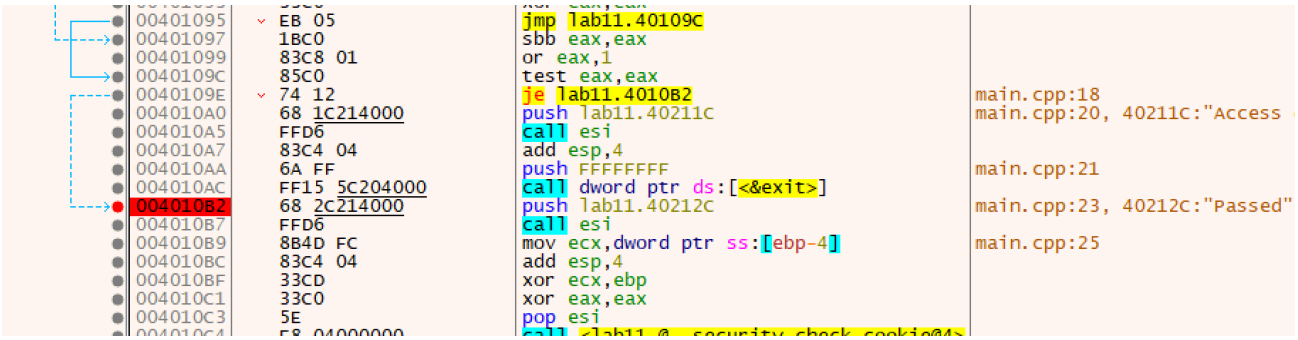


Рисунок 1 - Дизасембльований код

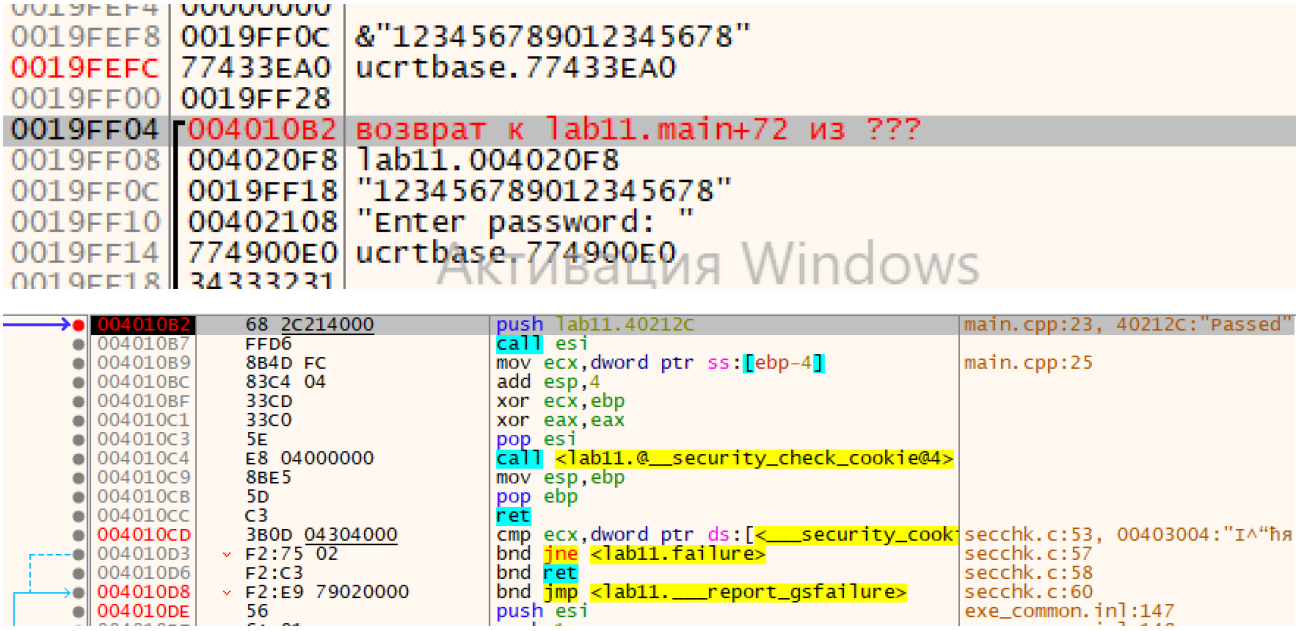


Рисунок 2 – Адреса повернення

Після заміни адреси повернення: програма попадає в місце, яке виконується тільки при вірному значенні пароля.

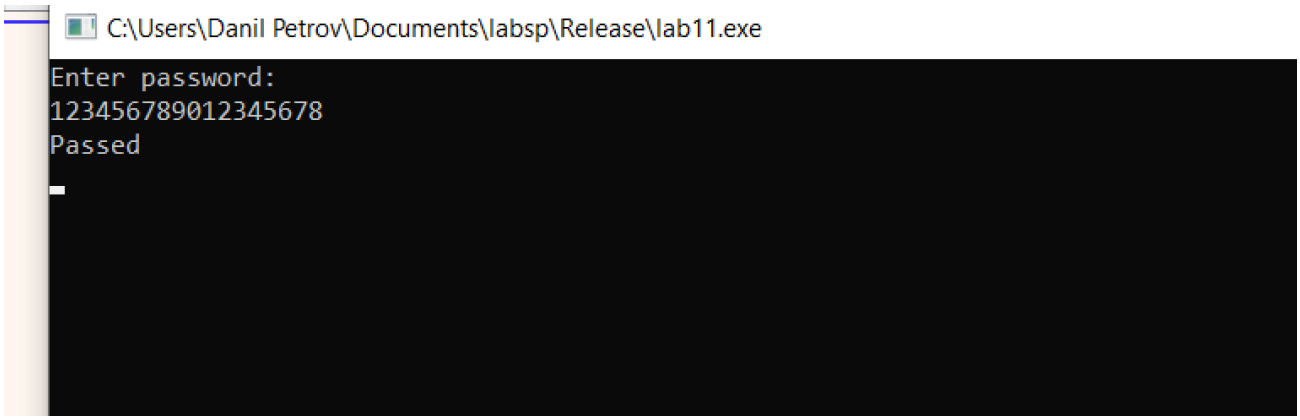


Рисунок – Ввід паролю

Висновок: Дослідили можливі вразливості переповнення буферу.