НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО”

КАФЕДРА АВТОМАТИЗОВАНИХ СИСТЕМ ОБРОБКИ ІНФОРМАЦІЇ І УПРАВЛІННЯ

Практична робота № 5

з дисципліни

“Основи захисту інформації”

Варіант 4

Виконала:

студентка групи ІС-71

Вознюк О. В.

Перевірив:

аспірант

Ільїн К. І.

Київ-2020

**Хід виконання роботи**

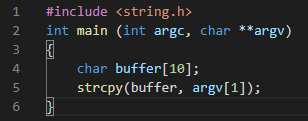


Рисунок 1 – перша програма для аналізу

У цьому прикладі використовуємо код містить помилку переповнення буфера.

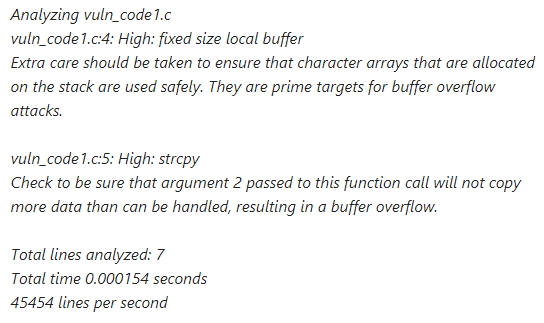


Рисунок 2 – вивід RATS

RATS повідомляє нам про двох помилках, причому обидва він привласнює високий (high) рівень небезпеки.

Перша «fixed size local buffer» - використання фіксованого розміру буфера в рядку 4 - char buffer [10].

Друга «buffer overflow» - переповнення буфера в рядку 5 - при використанні функції strcpy (). Якщо замість функції strcpy () використовувати gets (), то повідомлення RATS буде виглядати ось так:

vuln\_code1.c: 5: High: gets

Gets is unsafe !!! No bounds checking is performed, buffer is easily overflowable by user. Use fgets (buf, size, stdin) instead

Перевіримо як RATS реагує на код містить уразливість рядка формату, для цього пишемо код з дефектом, функція printf отримує вхідні рядок, відповідно до якої функція очікує, що перед її викликом в стек були занесені два аргументи якщо цю програму запустити з специфікаторами "% x % x "в якості параметра, можна побачити вміст 4 байт стека.

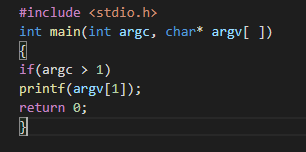


Рисунок 3 – друга прграма для аналізу

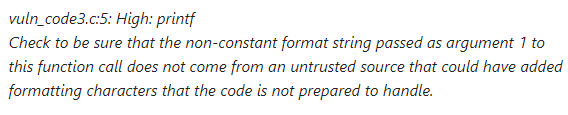


Рисунок 4 – вивід RATS

У наступному прикладі використовуємо код містить помилку переповнення буфера при взаємодії зі змінними оточення:

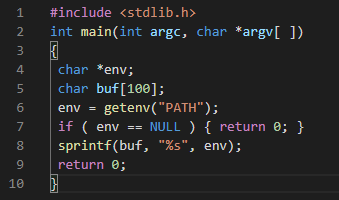


Рисунок 5 – третя программа для аналізу

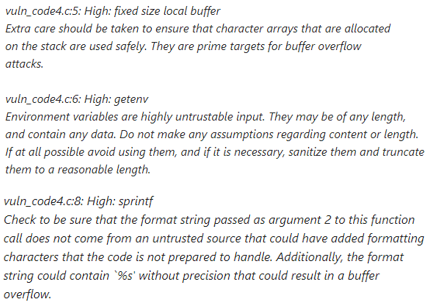


Рисунок 6 – вивід RATS

**Відповіді на теоретичні питання**

1. Якими є принципи роботи засобів автоматизованого аналізу за кодом застосунку?

Вони полягають в спільному уважному читанні вихідного коду і висловленні рекомендацій щодо його поліпшення. У процесі читання коду виявляються помилки або ділянки коду, які можуть стати помилковими в майбутньому. Програмне забезпечення часто містить різноманітні уразливості через помилки в коді програм. Помилки, допущені при розробці програм, в деяких ситуаціях призводять до збою програми, а отже, порушується нормальна

робота програми: при цьому часто виникає зміна і псування даних, зупинка програми або навіть системи. Більшість вразливостей пов'язано з неправильною обробкою даних, одержуваних ззовні, або недостатньо суворої їх перевіркою. Аналізатори вказують на ті місця в програмі, в яких можливо знаходиться помилка.

2. Наведіть відмінності між статичним та динамічним аналізом.

Динамічні відладчики дозволяють виробляти налагодження програми в процесі її виконання. Статичні аналізатори (статичні відладчики) використовують інформацію, накопичену в ході статичного аналізу програми.

3. Які типи попереджень генеруються засобами автоматизованого аналізу?

Класифікація вразливостей захисту в залежності від програмних помилок:

* Переповнення буфера (buffer overflow). Ця вразливість виникає через

відсутність контролю за виходом за межі масиву в пам'яті під час

виконання програми. Коли занадто великий пакет даних переповнює

буфер обмеженого розміру, вміст сторонніх елементів пам'яті перезаписується, і відбувається збій і аварійний вихід з програми. За

місцем розташування буфера в пам'яті процесу розрізняють переповнення буфера в стеку (stack buffer overflow), купі (heap buffer overflow) і області статичних даних (bss buffer overflow).

* Уразливості (tainted input vulnerability). Уразливості можуть виникати у

випадках, коли вводяться користувачем дані без достатнього контролю передаються інтерпретатору деякого зовнішнього мови (зазвичай це мова Unix shell або SQL). У цьому випадку користувач може таким чином поставити вхідні дані, що запущений інтерпретатор виконає зовсім не ту команду, яка передбачалася авторами вразливою програми.

* Помилки форматних рядків (format string vulnerability). Даний тип вразливостей захисту є підкласом уразливості. Він виникає через недостатній контроль параметрів при використанні функцій форматного введення-виведення printf, fprintf, scanf, і т. Д. Стандартної бібліотеки мови Сі. Ці функції приймають в якості одного з параметрів символьний рядок, що задає формат вводу або виводу наступних аргументів функції. Якщо користувач сам може задати вид форматування, то ця уразливість може виникнути в результаті невдалого застосування функцій форматування рядків.
* Уразливості як наслідок помилок синхронізації (race conditions). Проблеми, пов'язані з багатозадачністю, призводять до ситуації, під назвою "стан гонки": програма, не розрахована на виконання в багатозадачному середовищі, може вважати, що, наприклад, використовувані нею при роботі файли не може змінити інша програма. Як наслідок, зловмисник, вчасно підмінюючий вміст цих робочих файлів, може нав'язати програмі виконання певних дій

**Висновок**

В ході виконання лабораторної роботи відбувся пошук потенційних

вразливостей і помилок програмування в вихідних текстах на мові

програмування C / C++ з використанням автоматизованого засобу виявлення поширених помилок програмування RATS. Також була досліджена виявлена проблема: визначення типу та категорії помилки, локалізація, розроблення пропозицій щодо усунення і було виконане ранжування виявлених проблемних місць за ступенем серйозності.