Студент: Савостей В.В.
Курс: 4, **Группа:** 9

Лабораторная работа №1

Эксплуатация переполнения буфера

Основное задание

Задача: Необходимо заставить программу вызвать функцию `unreachable`, которая не вызывается в стандартном потоке выполнения, используя уязвимость переполнения буфера.

Решение: Уязвимость находится в использовании функции `strcpy(buf, argv[1])` без проверки длины входной строки. Буфер `buf` имеет размер 80 байт. Чтобы переопределить адрес возврата функции `main`, необходимо заполнить буфер и служебные данные стека.

Теория (схема стека):

- 80 байт — сам буфер (`buf`).
- 4 байта — сохраненный EBP (указатель базы стека).
- 4 байта — адрес возврата (EIP), который мы перезаписываем.

Итоговое смещение (offset) составляет $80 + 4 = 84$ байта. Целевой адрес функции `unreachable`: `0x40100f`. Архитектура x86 использует Little Endian (обратный порядок байт), поэтому адрес `0x40100f` в памяти должен быть записан как последовательность байт: `0f 10 40 00`. Функция `strcpy` автоматически добавляет нулевой байт в конец строки, поэтому явно передавать последний `00` не требуется.

Команда для запуска (One-liner):

```
python -c "import subprocess; payload = b'A'*84 + b'\x0f\x10\x40'; subprocess.run(['
    level12.exe', payload])"
```

Дополнительное задание 1

Задача: Реализовать обобщенное решение (скрипт), которое автоматически находит адрес функции `unreachable` (парсинг вывода) и эксплуатирует уязвимость, даже если адрес меняется при перекомпиляции.

Решение: Ниже представлен скриншот работы скрипта `solver.py`:

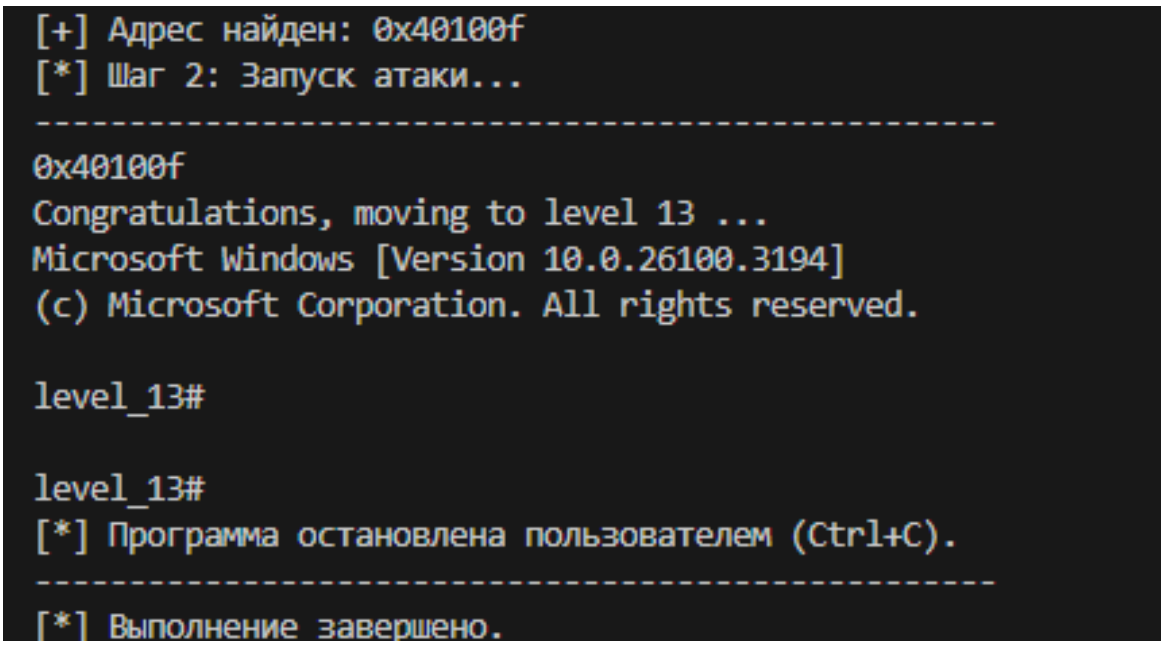

Рис. 1: Демонстрация работы автоматического эксплоита