Отчет по лабораторной работе №1 по предмету "Защита программ и данных"

Дородный Дмитрий СКБ172

12 декабря 2021 г.

1 Постановка задачи

Реализовать программу на языке C или C++, которая должна включать следующую последовательность действий:

1. Запрос какой-либо информации от пользователя
2. Произведение манипуляций по разработанному алгоритму
3. Запрос ключа от пользователя и сравнение его со сгенерированным
4. Сравнить введенный ключ с посчитанным

При реализации программы должны быть соблюдены условия:

1. Внутри программы должен быть какой-либо алгоритм преобразования данных для генерации ключевой информации.
2. Должны быть реализованы меры защиты от отладки.

2 Ход работы

Исходный код программы до применения защитных мер.
Исходный код программы после применения защитных мер.
Сборка при помощи g++: g++ -O3 -static -funroll-loops -o auth

2.1 Описание алгоритма

Алгоритм аутентификации следующий:

1. Запросить логин от пользователя
2. При необходимости дополнить/урезать до 16 символов
3. Первая половина ключа - все символы логина, сдвинутые по таблице ASCII вперед на количество символов, равное количеству единиц в коде символа (т.е. результат от %10)
4. Вторая половина ключа - все символы логина, сдвинутые назад на количество десятков в коде символа.

2.2 Защитные меры

Были применены следующие меры защиты от отладки:

1. Хеширование всех имен переменных / функций по MD5 для уменьшения читаемости
2. Замена числовых значений на эквивалентные им выражения с использованием hex чисел.

3. Замена всех строковых значений на эквивалентные в escaped-hex представлении
4. Проверка, что программа запускается без дебаггера, при помощи проверки /proc/self/status на наличие Tracer Pid
5. Использование усложняющих конструкций в коде:
    - Control flow flattening (создание нескольких переплетающихся case выражений, что затрудняет анализ кода, в том числе если используются инструменты для визуализации, например граф в IDA
    - Бесполезные циклы и переменные, всегда ложные if выражения
6. Опции компилятора:
    - ¨O3¨ усложнение анализа из-за оптимизаций
    - ¨static¨ Статическая линковка библиотек
    - ¨funrollloops¨ Циклы не сворачиваются

2.3 Другие методы

Иные методы защиты, которые могут усложнить анализ защищенной программы

- Сторонние утилиты, например CXX-OBFUS, выполняющие автоматическую обфускацию, например эквивалентные замены имен и значений, удаление пробелов и переносов строк и т.д.
- Шифрование функций или бинарных файлов
- Вычисление хеша, чтобы предотвратить изменение ("патч")