Министерство образования Республики Беларусь
Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Информационные сети. Основы безопасности

ОТЧЁТ
к лабораторной работе №7
на тему

ЗАЩИТА ПО ОТ НЕСАНКЦИОНИРОВАННОГО ИСПОЛЬЗОВАНИЯ

Выполнил: студент гр.253504
Фроленко К.Ю.
Проверил: ассистент кафедры информатики
Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Современные программные системы сталкиваются с множеством угроз, включая несанкционированный доступ и взлом исходного кода. Одним из методов защиты является обфускация — процесс изменения исходного кода таким образом, чтобы его было трудно понять и проанализировать. Это позволяет значительно усложнить задачу для злоумышленников, пытающихся извлечь и использовать код для атак.

Цель данной работы — продемонстрировать процесс обфускации исходного кода программы `traceroute`, написанной на языке C. Обфускация делает программу сложной для анализа и позволяет защитить ее от атак, направленных на извлечение логики работы приложения. В данном случае, основное внимание уделяется замене читаемых имен переменных и функций на трудночитаемые, а также изменению структуры кода для предотвращения его несанкционированного использования.

1 ФОРМУЛИРОВКА ЗАДАЧИ

Задача данной работы состоит в том, чтобы продемонстрировать обфускацию исходного кода программы `traceroute` с целью повышения безопасности кода и защиты от анализа злоумышленниками. Для этого были проведены следующие шаги:

1 Замена имен переменных и функций на трудночитаемые символы.

2 Использование макросов для выноса значений и ключевых элементов программы.

3 Усложнение структуры кода для предотвращения его быстрого анализа.

Обфускация не должна менять функциональность программы, а лишь усложнять ее восприятие. Программа должна выполнять ту же задачу трассировки маршрута до удаленного хоста, но ее исходный код должен стать трудным для понимания.

2 ПРИМЕР ОБФУСКАЦИИ КОДА

Изначально программа `traceroute` имеет понятные имена переменных и функций, такие как `first_ttl`, `max_ttl`, `parseCommandLine` и так далее. Однако после проведения обфускации:

1 Имена переменных были заменены на случайные символы, такие как `a1`, `a2`, `a3`.

2 Все строки и константы, такие как сообщения об ошибках и инструкция, были вынесены в макросы с трудными для восприятия названиями (`a7`, `a8`, `a9` и т. д.).

3 Функции также были переименованы в трудночитаемые идентификаторы, например, `isValidTTL` было заменено на `a21`, `parseCommandLine` — на `a28`.

Обфуцированный код на рисунке 1.

```
#include <bits/stdc++.h>
#include <winsock2.h>
#include <iphlpapi.h>
#include <ws2tcpip.h>
#include <icmpapi.h>
#pragma comment(lib,"Ws2_32.lib")
#pragma comment(lib,"Iphlpapi.lib")
#pragma comment(lib,"Icmp.lib")
#define a1 32
#define a2 30
#define a3 1000
#define a4 long long
#define a5 std
#define a6 chrono
#define a7 "Error resolving destination host."
#define a8 "Usage: "
#define a9 "First hop value must be in the range 1-255."
#define a10 "Max TTL value must be in the range 1-255."
#define a11 "Missing destination_host."
#define a12 "Max TTL value must be greater than or equal to first TTL value."
#define a13 "Unable to open ICMP handle: "
#define a14 "Reached destination"
#define a15 "Tracing route to "
#define a16 "ms"
#define a17 "* "
#define a18 "  "
#define a19 1
#define a20 255
using namespace a5; using namespace a6; bool a21(const string& a22) { try { size_t a23 = 0; a4 a
```

Рисунок 1 – Код после обусфакции

Такой код будет сложнее читать и анализировать, что затрудняет любые попытки извлечь его логику или найти уязвимости. Однако функциональность программы сохраняется.

3 РЕЗУЛЬТАТЫ И ОЦЕНКА

После обфускации код программы продолжает выполнять ту же задачу, что и до обфускации, но теперь его сложнее анализировать. Все переменные и функции имеют случайные имена, и вся логика программы скрыта за этими идентификаторами. Например:

1 Имена таких функций, как `traceRoute` и `resolveHostname`, были изменены на сложные идентификаторы (`a40`, `a35`), что усложняет их идентификацию.

2 Сообщения об ошибках и строковые значения были заменены на макросы с нелогичными именами, что делает код трудным для анализа.

3 Структура кода была изменена, чтобы она не имела очевидной связи между действиями и переменными.

Этот процесс значительно повышает безопасность программы, так как она становится трудной для обратного инжиниринга, несмотря на сохранение функциональности.

ЗАКЛЮЧЕНИЕ

В ходе работы была успешно выполнена обфускация программы `traceroute`, написанной на языке C. Результаты показали, что даже базовые методы обфускации, такие как замена имен переменных и функций, могут значительно усложнить анализ кода и повысить безопасность приложения.

Обфускация доказала свою эффективность как метод защиты программного обеспечения от несанкционированного доступа. Этот метод затрудняет задачу для злоумышленников, которые пытаются изучить исходный код с целью его эксплуатации. Важно подчеркнуть, что обфускация является важным элементом комплексной стратегии защиты программного обеспечения, позволяющим защитить код от обратного инжиниринга и атак.

Таким образом, проведенная работа демонстрирует значимость обфускации для повышения безопасности программного обеспечения в условиях современных угроз и подтверждает, что даже простые методы защиты, такие как изменение имен переменных и функций, могут существенно затруднить несанкционированное использование программы.