

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 №3  
на тему

**ИДЕНТИФИКАЦИЯ И АУТЕНТИФИКАЦИЯ ПОЛЬЗОВАТЕЛЯ.  
ПРОТОКОЛ KERBEROS**

Выполнил: студент гр.253504 Лавренова А.С.  
Проверил: ассистент кафедры информатики  
Герчик А.В.

Минск 2025

## СОДЕРЖАНИЕ

Введение.....	3
1 Теоретические сведения .....	4
2 Пример выполнения программного средства .....	5
Заключение .....	7
Список использованных источников .....	8
Приложение А (обязательное) .....	9

## **ВВЕДЕНИЕ**

В современном мире, где информационные технологии играют ключевую роль в жизни организаций и пользователей, обеспечение безопасности данных становится одной из главных задач. Аутентификация и идентификация пользователей являются основными компонентами систем безопасности, обеспечивающими защиту от несанкционированного доступа и утечки информации.

В данной лабораторной работе рассматривается процесс идентификации и аутентификации пользователей, а также их реализация с использованием протокола Kerberos. Kerberos предоставляет надежный механизм для безопасной аутентификации в распределенных системах, позволяя пользователям и сервисам взаимодействовать без риска компрометации их учетных данных.

В данной лабораторной работе будут изучены принципы работы протокола Kerberos, его архитектуры и механизмов, а также будет реализовано подобное программное средство. В ходе работы будут рассмотрены ключевые аспекты, такие как выдача билетов, синхронизация времени и безопасность передачи данных. Результаты лабораторной работы помогут глубже понять механизмы аутентификации и их важность для защиты информации в современных информационных системах.

# 1 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Идентификация — это процесс, в ходе которого система определяет, кто является пользователь. Обычно это происходит через предоставление уникального идентификатора, такого как имя пользователя, номер учетной записи или адрес электронной почты. Идентификация сама по себе не предоставляет прав доступа; она лишь сообщает системе, с каким пользователем она имеет дело. [1]

Аутентификация — это процесс проверки подлинности идентифицированного пользователя. Она подтверждает, что пользователь действительно тот, за кого себя выдает. Аутентификация может осуществляться различными способами:

- пароли;
- биометрические данные;
- токены;
- многофакторная аутентификация (MFA).

Kerberos — это сетевой протокол аутентификации, разработанный для обеспечения безопасного обмена данными в небезопасных сетях. Он позволяет пользователям и сервисам аутентифицироваться друг перед другом с использованием симметричного шифрования. [2]

Ticket (билет) – зашифрованный пакет данных, который выдается доверенным центром аутентификации.

Key Distribution Center (KDC, центр распределения ключей), доверенный центр аутентификации — доверенная сторона.

Ticket Granting Ticket (TGT) – первичное удостоверение пользователя для доступа к сетевым ресурсам.

Ticket Granting Service (TGS) – удостоверение для доступа к конкретному сетевому ресурсу, выдается на основе TGT.

При любых взаимодействиях по сети не передаются ни пароли, ни значения хеша пароля в открытую. Вместо этого используется механизм билетов, что значительно повышает уровень безопасности.

Kerberos требует, чтобы системные часы всех участвующих во взаимодействии были синхронизированы. Это необходимо для предотвращения атак, основанных на повторном использовании старых билетов. Синхронизация времени помогает гарантировать, что билеты имеют срок действия и не могут быть использованы злоумышленниками после истечения этого срока.

Таким образом, Kerberos обеспечивает надежную аутентификацию и защиту данных, что делает его одним из наиболее популярных протоколов для безопасного доступа к сетевым ресурсам.

## 2 ПРИМЕР ВЫПОЛНЕНИЯ ПРОГРАММНОГО СРЕДСТВА

Запуск программного средства начинается с инициализации компонентов системы аутентификации Kerberos. В данном примере, изображённом на рисунке 2.1 реализованы три основных модуля: клиент (Client), центр распределения ключей (KDC) и сервис (Service). Также используется файл базы данных (database.txt) для хранения информации о зарегистрированных пользователях.

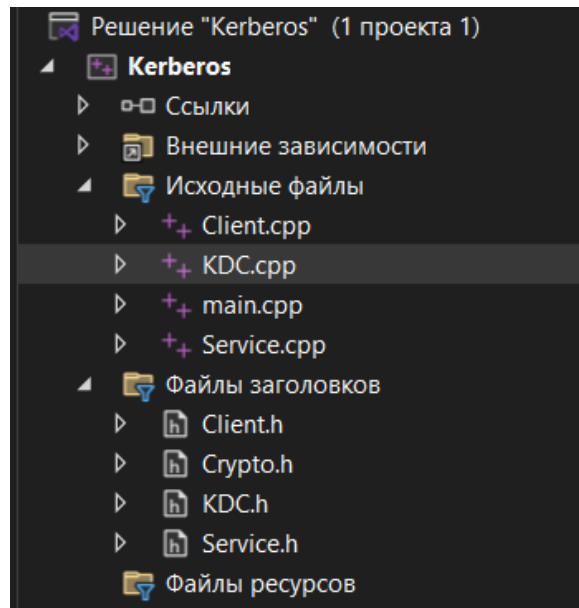


Рисунок 2.1 – Структура программы

При запуске программы KDC загружает базу данных пользователей из файла database.txt. Например, файл может содержать данный, изображённые на рисунке 2.2.

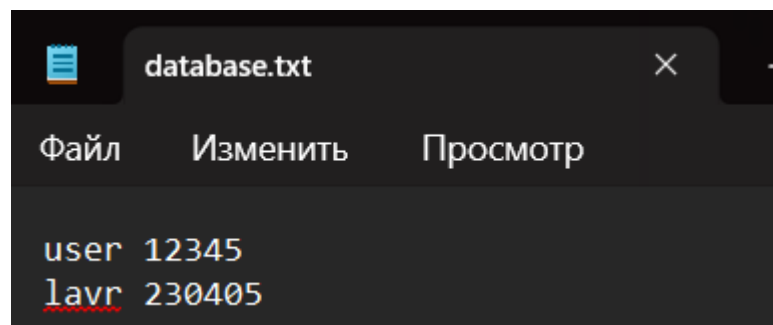
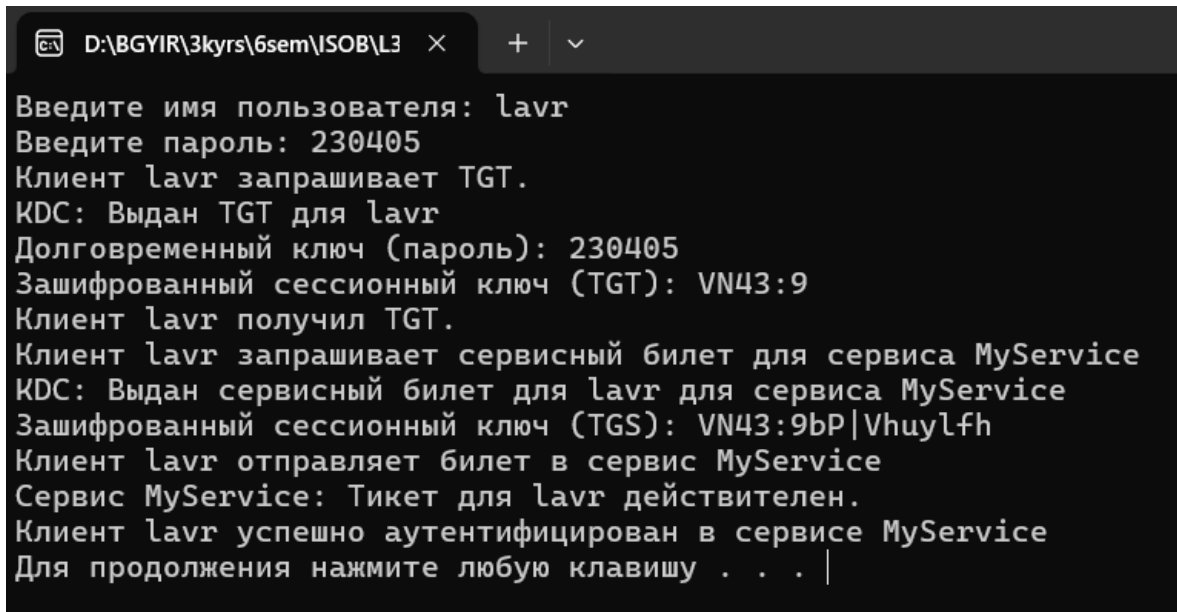


Рисунок 2.2 – Файл database.txt

После загрузки KDC хранит эту информацию в памяти для проверки учетных данных пользователей при их аутентификации.

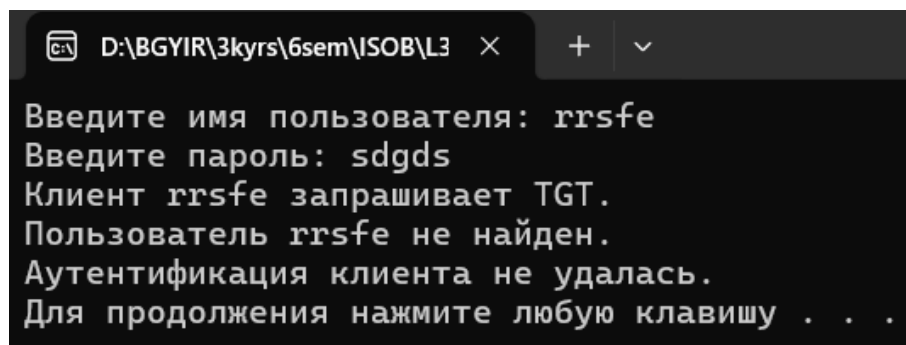
Далее клиент запрашивает аутентификацию, вводя свои учетные данные. KDC проверяет корректность введенных данных. Если проверка успешна, создается и шифруется сессионный ключ, который включается в Ticket Granting Ticket (TGT). Клиент запрашивает билет для доступа к сервису "MyService". KDC расшифровывает сессионный ключ из TGT, создаёт новый ключ, связанный с запрашиваемым сервисом, и снова шифрует его. Клиент получает сервисный билет. Клиент отправляет полученный сервисный билет в сервис "MyService". Сервис расшифровывает ключ и проверяет его корректность (рисунок 2.3).



```
D:\BGYIR\3kysr\6sem\ISOB\L3
Введите имя пользователя: lavr
Введите пароль: 230405
Клиент lavr запрашивает TGT.
KDC: Выдан TGT для lavr
Долговременный ключ (пароль): 230405
Зашифрованный сессионный ключ (TGT): VN43:9
Клиент lavr получил TGT.
Клиент lavr запрашивает сервисный билет для сервиса MyService
KDC: Выдан сервисный билет для lavr для сервиса MyService
Зашифрованный сессионный ключ (TGS): VN43:9bP|Vhuylfh
Клиент lavr отправляет билет в сервис MyService
Сервис MyService: Тикет для lavr действителен.
Клиент lavr успешно аутентифицирован в сервисе MyService
Для продолжения нажмите любую клавишу . . . |
```

Рисунок 2.3 – Консоль программного продукта

Если билет недействителен (например, срок действия истёк или данные были изменены), сервис отклоняет запрос (рисунок 2.4).



```
D:\BGYIR\3kysr\6sem\ISOB\L3
Введите имя пользователя: rrsfe
Введите пароль: sdgds
Клиент rrsfe запрашивает TGT.
Пользователь rrsfe не найден.
Аутентификация клиента не удалась.
Для продолжения нажмите любую клавишу . . .
```

Рисунок 2.4 – Консоль программного продукта при вводе неверных данных

Таким образом, программное средство демонстрирует процесс идентификации и аутентификации пользователей в соответствии с принципами протокола Kerberos.

## ЗАКЛЮЧЕНИЕ

В ходе выполнения данной лабораторной работы были изучены основные аспекты идентификации и аутентификации пользователей, а также реализация протокола Kerberos. Мы подробно рассмотрели архитектуру Kerberos, его составляющие, такие как Ticket Granting Ticket (TGT) и Ticket Granting Service (TGS), а также механизмы обеспечения безопасности при аутентификации.

Практическая часть работы заключалась в реализации кода на языке C++ в среде разработки Visual Studio. Это позволило глубже понять, как работает протокол Kerberos на уровне программирования и какие алгоритмы используются для обеспечения безопасного обмена данными.

По итогам работы можно сделать вывод, что Kerberos является мощным инструментом для аутентификации в распределенных системах, обеспечивая высокий уровень безопасности и защиты данных. Результаты лабораторной работы подтвердили, что правильная реализация протокола и соблюдение его принципов являются залогом надежной аутентификации пользователей в современных информационных системах. Полученные знания и навыки окажутся полезными для дальнейшего изучения вопросов информационной безопасности и разработки защищенных приложений.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] Защита данных. От авторизации до аудита. Андресс Д. – [Электронный ресурс]. – Режим доступа: <https://codelibs.ru/zashita-dannyh-ot-avtorizacii-do-audita/?ysclid=m7b2qkf497855134968>.

[2] Использование протокола Kerberos для авторизации пользователей прокси-сервера Squid на базе pfSense. М. К. Чернышов. – [Электронный ресурс]. – Режим доступа: <https://www.cs.vsu.ru/ipmt-conf/conf/2023/works/1.%20Методы%20и%20технологии%20разработки%20программных%20систем/2274.dokl.pdf>.



## ПРИЛОЖЕНИЕ А

### (обязательное)

### Исходный код

```
#include <iostream>
#include "KDC.h"
#include "Service.h"
#include "Client.h"
#include <cstdlib>
#include <ctime>

int main() {
    setlocale(LC_ALL, "Russian");
    // Инициализация генератора случайных чисел
    srand(static_cast<unsigned int>(time(nullptr)));

    std::string dbPath = "database.txt";
    KDC kdc(dbPath);
    if (!kdc.loadDatabase()) {
        return 1;
    }
    Service service("MyService");
    std::string username, password;
    std::cout << "Введите имя пользователя: ";
    std::cin >> username;
    std::cout << "Введите пароль: ";
    std::cin >> password;
    // Создаём клиента и иницилируем процедуру аутентификации
    Client client(username, password);
    client.requestService(service, kdc);

    system("pause");
    return 0;
}

#include "KDC.h"
#include "Crypto.h"
#include <fstream>
#include <sstream>
#include <iostream>
#include <cstdlib>

KDC::KDC(const std::string& dbPath) : databasePath(dbPath) {}

bool KDC::loadDatabase() {
    std::ifstream file(databasePath);
    if (!file) {
        std::cerr << "Не удалось открыть базу данных: " <<
databasePath << std::endl;
        return false;
    }
    std::string line;
    while (std::getline(file, line)) {
        std::istringstream iss(line);
        std::string username, password;
        if (iss >> username >> password) {
```

```

        credentials[username] = password;
    }
}
file.close();
return true;
}

std::string KDC::generateSessionKey() {
    // генерируем случайный сессионный ключ в виде строки
    int num = rand() % 1000000;
    return "SK" + std::to_string(num);
}

Ticket KDC::issueTGT(const std::string& clientName, const
std::string& password) {
    Ticket ticket;
    // Проверяем, существует ли пользователь в базе
    if (credentials.find(clientName) == credentials.end()) {
        std::cerr << "Пользователь " << clientName << " не найден."
<< std::endl;
        return ticket;
    }
    // Проверяем корректность пароля
    if (credentials[clientName] != password) {
        std::cerr << "Неверный пароль для " << clientName <<
std::endl;
        return ticket;
    }
    // Выдаем TGT
    ticket.clientName = clientName;
    std::string plainKey = generateSessionKey();
    // Шифруем сессионный ключ перед отправкой
    ticket.sessionKey = Crypto::simpleEncrypt(plainKey);
    ticket.timestamp = std::time(nullptr);
    std::cout << "KDC: Выдан TGT для " << clientName
        << ", зашифрованный сессионный ключ: " << ticket.sessionKey
<< std::endl;
    return ticket;
}

Ticket KDC::issueServiceTicket(const Ticket& tgt, const
std::string& serviceName) {
    Ticket serviceTicket;
    serviceTicket.clientName = tgt.clientName;
    // Дешифруем TGT, чтобы получить исходный сессионный ключ
    std::string decryptedKey =
Crypto::simpleDecrypt(tgt.sessionKey);
    // Модифицируем сессионный ключ для конкретного сервиса
    std::string serviceKey = decryptedKey + "_" + serviceName;
    // Шифруем полученный ключ для передачи клиенту
    serviceTicket.sessionKey = Crypto::simpleEncrypt(serviceKey);
    serviceTicket.timestamp = std::time(nullptr);
    std::cout << "KDC: Выдан TGS для " << tgt.clientName
        << " для сервиса " << serviceName << std::endl;
    return serviceTicket;
}

```