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

ПРОТОКОЛ KERBEROS

Выполнил: студент гр.253501
Станишевский А.Д.

Проверил: ассистент кафедры информатики
Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ЦЕЛЬ РАБОТЫ

Целью данной работы является исследование принципов аутентификации и управления доступом на примере реализации протокола Kerberos. В рамках исследования была поставлена задача создания программы, демонстрирующей основные этапы работы протокола Kerberos: аутентификацию пользователя, получение билетов для доступа к сервисам (Ticket Granting Ticket — TGT и Service Ticket) и предоставление доступа к защищенным ресурсам.

Программа поддерживает работу с несколькими пользователями, что позволяет исследовать многопользовательскую среду и взаимодействие между клиентами, сервером аутентификации (Authentication Server, AS), сервером выдачи билетов (Ticket Granting Service, TGS) и сервисами. Особое внимание уделено моделированию процессов шифрования данных и проверки подлинности билетов, что отражает ключевые принципы безопасности Kerberos.

Особенностью программы является удобство взаимодействия с ней через консольный интерфейс, который позволяет пользователю выполнять такие действия, как аутентификация, запрос билетов и доступ к защищенным ресурсам. Результаты выполнения каждого этапа отображаются в консоли, что делает процесс прозрачным и наглядным. Кроме того, программа предоставляет возможность просмотра состояния системы, включая информацию о текущих пользователях, их статусах и доступах, что помогает лучше понять принципы работы протокола.

В процессе разработки были изучены основные принципы функционирования Kerberos, включая использование сессионных ключей, временных меток и защищенных билетов для предотвращения атак повторного использования данных (replay attacks). Реализация этих механизмов позволила глубже понять особенности протокола, его преимущества, такие как централизованное управление доступом и защита от несанкционированного доступа, а также ограничения, например, зависимость от надежности центрального сервера аутентификации.

Таким образом, в ходе выполнения работы были получены практические навыки реализации и анализа протокола Kerberos, а также создана программа, демонстрирующая базовые подходы к аутентификации и управлению доступом в распределенных системах. Разработанное решение может служить основой для дальнейшего изучения более сложных методов защиты информации и современных протоколов аутентификации.

2 ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

Kerberos — это сетевой протокол аутентификации, разработанный для обеспечения безопасного взаимодействия между клиентами и серверами в распределенных системах. Протокол основан на использовании симметричной криптографии и позволяет пользователям и сервисам аутентифицироваться друг перед другом без передачи паролей по сети. Kerberos был создан в Массачусетском технологическом институте (MIT) и получил широкое распространение благодаря своей надежности и эффективности.

Основная цель Kerberos — предоставить механизм аутентификации, который защищает пользователей и сервисы от атак, таких как перехват данных (man-in-the-middle), повторные атаки (replay attacks) и несанкционированный доступ.

Протокол Kerberos состоит из трех основных этапов:

1 Аутентификация клиента у сервера аутентификации (Authentication Server, AS).

2 Получение билета для доступа к сервису (Ticket Granting Service, TGS).

3 Доступ к сервису.

Аутентификация — это процесс проверки подлинности субъекта (пользователя, устройства или системы), который пытается получить доступ к ресурсу. Цель аутентификации — убедиться, что субъект действительно является тем, за кого себя выдает. В контексте Kerberos аутентификация выполняется с использованием зашифрованных билетов и временных меток, что исключает необходимость передачи паролей по сети.

Таким образом, изучение Kerberos позволяет глубже понять принципы аутентификации и управления доступом.

3 ХОД РАБОТЫ

Для удобства взаимодействия программа предоставляет консольный интерфейс, позволяющий пользователю выполнять следующие действия:

1 Аутентификация пользователя.

2 Получение доступа к защищенным сервисам.

3 Просмотр активных пользователей.

4 Выход из системы.

На рисунках 3.1 изображен результат работы программного продукта.

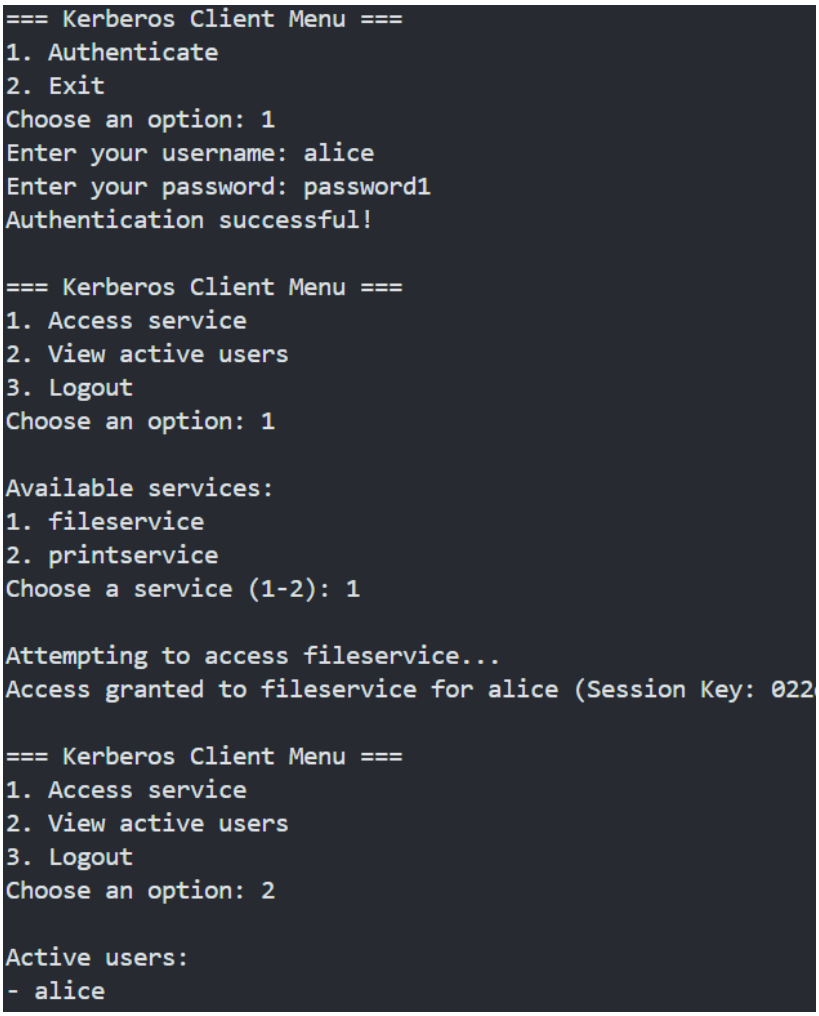

Рисунок 3.1 – Результат работы программного продукта

Таким образом, программа успешно справляется с поставленными задачами.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была достигнута поставленная цель — изучение принципов аутентификации и управления доступом на примере реализации протокола Kerberos. Разработанная программа успешно демонстрирует основные этапы работы протокола, которые соответствуют его ключевым принципам.

Принцип работы программы согласно протоколу Kerberos:

1 Аутентификация пользователя: пользователь вводит свои учетные данные (логин и пароль). Программа проверяет их корректность и, если данные верны, генерирует Ticket Granting Ticket (TGT) и сессионный ключ для взаимодействия с сервером TGS. TGT зашифровывается секретным ключом сервера TGS, что обеспечивает безопасность передачи данных.

2 Получение билета для доступа к сервису: после успешной аутентификации пользователь запрашивает билет для доступа к конкретному сервису (например, файловому серверу). Программа проверяет TGT, расшифровывает его секретным ключом сервера TGS и выдает Service Ticket, который содержит информацию о пользователе и сессионный ключ для взаимодействия с сервисом. Service Ticket зашифровывается секретным ключом сервиса, что гарантирует его подлинность.

3 Предоставление доступа к защищенному ресурсу: пользователь отправляет Service Ticket на сервер сервиса. Программа проверяет подлинность билета, расшифровывает его секретным ключом сервиса и предоставляет доступ к защищенному ресурсу, например, к содержимому файла. Если билет действителен, пользователь получает доступ к ресурсу.

Программа поддерживает работу с несколькими пользователями, что позволяет исследовать многопользовательскую среду и взаимодействие между клиентами, сервером аутентификации (Authentication Server, AS), сервером выдачи билетов (Ticket Granting Service, TGS) и сервисами. Особое внимание уделено моделированию процессов шифрования данных и проверки подлинности билетов, что отражает ключевые принципы безопасности Kerberos.

Разработанная программа предоставляет удобный консольный интерфейс, позволяющий выполнять следующие действия: аутентификация пользователей, получение доступа к защищенным сервисам, просмотр активных пользователей.

Таким образом, в результате проделанной работы были получены практические навыки реализации и анализа протокола Kerberos, а также создана программа, которая может быть использована для демонстрации принципов аутентификации и управления доступом в распределенных системах.

ПРИЛОЖЕНИЕ А
(обязательное)
Листинг программного кода

```
def authenticate(username, password):
          global authenticated_user, tgt_id, tgt_hash, session_key
          tgt_id, tgt_hash, session_key = authenticate_user(username, password)
          if tgt_id:
              authenticated_user = username
              if username not in active_users:
                  active_users.append(username)
              return True
          return False

def generate_session_key():
          return hashlib.sha256(str(uuid.uuid4()).encode()).hexdigest()

def generate_tgt(username):
          tgt_id = str(uuid.uuid4())
          user = get_user(username)
          secret_key = user["secret_key"]
          session_key = generate_session_key()
          tgt_data = f"{username}:{tgt_id}:{session_key}"
          tgt_hash           =           hashlib.sha256((tgt_data           +
secret_key).encode()).hexdigest()
          return tgt_id, tgt_hash, session_key

      def authenticate_user(username, password):
          if verify_user(username, password):
              return generate_tgt(username)
          return None, None, None
      def verify_tgt(tgt_id, tgt_hash, username, session_key):
          user = get_user(username)
          if user:
              secret_key = user["secret_key"]
              tgt_data = f"{username}:{tgt_id}:{session_key}"
              expected_hash        =        hashlib.sha256((tgt_data        +
secret_key).encode()).hexdigest()
              return tgt_hash == expected_hash
          return False

def generate_service_ticket(username, service_name, session_key):
          service = get_service(service_name)
          if service:
              secret_key = service["secret_key"]
              ticket_id = str(uuid.uuid4())
              service_session_key = generate_session_key()
              ticket_data                                                    =
f"{username}:{service_name}:{ticket_id}:{service_session_key}"
              ticket_hash        =        hashlib.sha256((ticket_data        +
secret_key).encode()).hexdigest()
              return ticket_id, ticket_hash, service_session_key
          return None, None, None

def   request_service_ticket(tgt_id,   tgt_hash,   username,   service_name,
session_key):
          if verify_tgt(tgt_id, tgt_hash, username, session_key):
              return      generate_service_ticket(username,      service_name,
session_key)
          return None, None, None
```