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 ПОЛЬЗОВАТЕЛЕЙ. ПРОТОКОЛ KERBEROS

Выполнил: студент гр. 253503
Тимошевич К.С.

Проверил: ассистент кафедры информатики Герчик А.В.

Минск 2025

СОДЕРЖАНИЕ

1 ПОСТАНОВКА ЗАДАЧИ

Постановка задачи для данной лабораторной работы заключается в разработке и программной реализации упрощённой модели протокола аутентификации *Kerberos* на языке *Python*. *Kerberos* – это сетевой протокол, который обеспечивает безопасную идентификацию пользователей и сервисов, используя билеты и симметричное шифрование [1]. Цель работы заключается в изучении принципов работы протокола, его основных компонентов и этапов, а также в практической реализации этих этапов с использованием криптографических методов.

В рамках работы необходимо разработать три ключевых компонента системы: сервер аутентификации (*Authentication Server*), сервер выдачи билетов (*Ticket Granting Server*) и сервер приложений (*Application Server*). Сервер аутентификации отвечает за проверку личности пользователя и выдачу *TGT* (*Ticket Granting Ticket*), который затем используется для получения доступа к сервисам. Сервер выдачи билетов отвечает за выдачу сервисных билетов на основе *TGT*. Сервер приложений проверяет полученный билет и предоставляет клиенту доступ к запрашиваемому ресурсу.

Для реализации системы необходимо использовать симметричное шифрование на основе библиотеки *cryptography*, а также реализовать механизмы управления ключами, проверки временных меток и сериализации данных в формате *JSON*. В результате выполнения работы должно быть создано программное средство, демонстрирующее весь процесс аутентификации: от запроса билета до доступа к сервису.

2 ХОД ВЫПОЛНЕНИЯ РАБОТЫ

Разработка программного средства включает реализацию трёх основных компонентов: сервера аутентификации, сервера выдачи билетов и сервера приложений. Для обеспечения безопасности взаимодействия между этими модулями используется симметричное шифрование с динамической генерацией сессионных ключей.

Сервер аутентификации отвечает за начальную аутентификацию клиента и выдачу билета на получение билетов (*TGT*). При регистрации клиента в базе данных сервера создается запись с идентификатором клиента и ключом, генерируемым на основе пароля. Когда клиент отправляет запрос на аутентификацию, сервер проверяет наличие клиента в базе данных и генерирует сессионный ключ. Сессионный ключ и *TGT* шифруются с использованием ключа клиента и ключа *TGS* соответственно. Ответ сервера содержит зашифрованный *TGT*, который клиент может использовать для дальнейшего взаимодействия с системой.

Сервер выдачи билетов предоставляет клиенту билет для доступа к сервисам. Для получения билета клиент отправляет *TGT* и аутентификатор, содержащий идентификатор клиента и временную метку, зашифрованные сессионным ключом. *TGS* расшифровывает *TGT* с использованием своего ключа, проверяет подлинность клиента и срок действия *TGT*. Если проверка прошла успешно, *TGS* генерирует новый сессионный ключ для взаимодействия клиента с сервером приложений и создает сервисный билет, зашифрованный с использованием ключа сервера приложений. Ответ сервера включает сервисный билет и новый сессионный ключ.

Сервер приложений проверяет подлинность клиента на основе полученного сервисного билета и аутентификатора. Клиент отправляет сервисный билет и аутентификатор, зашифрованные новым сессионным ключом. Сервер приложений расшифровывает сервисный билет с использованием своего ключа, извлекает сессионный ключ и проверяет аутентификатор. Если проверка прошла успешно, сервер предоставляет доступ к запрашиваемому сервису.

Для выполнения криптографических операций использована библиотека cryptography. Ключи генерируются с помощью функции *Fernet.generate_key()*, а данные шифруются и дешифруются через объекты *Fernet* [2]. Все данные между клиентом и серверами передаются в формате *JSON*, при этом бинарные объекты кодируются с использованием *Base64* для безопасной передачи.

В результате был разработан функциональный прототип, моделирующий ключевые этапы протокола *Kerberos*, что обеспечивает безопасную аутентификацию и авторизацию клиента.

ЗАКЛЮЧЕНИЕ

В процессе разработки программного средства для моделирования протокола *Kerberos* была реализована функциональность его ключевых компонентов системы: сервера аутентификации, сервера выдачи билетов и сервера приложений. Каждый из этих компонентов выполняет свою роль в обеспечении безопасной аутентификации и авторизации пользователей. Сервер аутентификации отвечает за начальную проверку идентификации клиента и выдачу билета на получение билетов (*TGT*), который позволяет клиенту без повторной аутентификации взаимодействовать с другими сервисами в системе.

Сервер выдачи билетов генерирует сервисные билеты (*Service Tickets*), которые предоставляют доступ клиенту к конкретным сервисам. А сервер приложений осуществляет проверку подлинности этих билетов и предоставляет клиенту запрашиваемый сервис. Все операции между клиентом и серверами проводятся с использованием криптографического шифрования сессионных ключей и билетов. Для реализации криптографических операций была использована библиотека *cryptography*, что позволило эффективно управлять ключами и шифровать данные [3].

В ходе тестирования были проверены все этапы протокола *Kerberos*, начиная от аутентификации клиента, получения *TGT*, генерации сервисных билетов и заканчивая успешным доступом к сервисам. Все этапы протокола функционировали корректно, что подтверждает правильность реализации всех компонентов системы.

Реализованное программное средство продемонстрировало соответствие основным принципам безопасности протокола *Kerberos*. Использование временных меток и динамическая генерация сессионных ключей для каждой сессии позволяют защититься от атак повторного воспроизведения, гарантируя, что даже в случае компрометации одного из компонентов системы, другие остаются защищенными. Это обеспечит защиту данных и конфиденциальность, делая систему безопасной и устойчива к потенциальным угрозам.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] What is Kerberos and How Does it Work? [Электронный ресурс]. – Режим доступа: https://www.techtarget.com/searchsecurity/definition/Kerberos.. – Дата доступа: 10.02.2025

[2] Fernet (symmetric encryption) [Электронный ресурс]. – Режим доступа: https://cryptography.io/en/latest/fernet/. – Дата доступа: 10.02.2025

[3] Cryptography documentation. [Электронный ресурс]. – Режим доступа: https://cryptography.io/.. – Дата доступа: 10.02.2025