**Оглавление**

[Введение 3](#_Toc8069893)

[1 Анализ предметной области 3](#_Toc8069894)

[1.1 Описание предметной области 3](#_Toc8069895)

[1.2 Формулировка проблемы и актуальность 3](#_Toc8069896)

[1.3 Обзор аналогов 3](#_Toc8069897)

[2 Программная документация 3](#_Toc8069898)

[2.1 Техническое задание на программное обеспечение 3](#_Toc8069899)

[2.1.1 Общие сведения 3](#_Toc8069900)

[2.1.1.1 Наименование программы 3](#_Toc8069901)

[2.1.1.2 Плановые сроки начала и окончания работ 3](#_Toc8069902)

[2.1.1.3 Порядок оформления и предъявления заказчику результатов работ 3](#_Toc8069903)

[2.1.1.4 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ 3](#_Toc8069904)

[2.1.1.5 Определения, обозначения, сокращения 3](#_Toc8069905)

[2.1.2 Назначение и цели создания системы 3](#_Toc8069906)

[2.1.2.1 Назначение системы 3](#_Toc8069907)

[2.1.2.2 Цели создания системы 3](#_Toc8069908)

[2.1.3 Характеристика объектов автоматизации 3](#_Toc8069909)

[2.1.3.1 Краткие сведения об объекте автоматизации 3](#_Toc8069910)

[2.1.3.2 Существующие программное обеспечение 3](#_Toc8069911)

[2.1.4 Требования к системе 3](#_Toc8069912)

[2.1.4.1 Требование к структуре и функционированию системы 3](#_Toc8069913)

[2.1.4.2 Требования к надежности 3](#_Toc8069914)

[2.1.4.3 Требования к безопасности 4](#_Toc8069915)

[2.1.4.4 Требования к эргономике и технической эстетике 4](#_Toc8069916)

[2.1.4.5 Требования к интерфейсу 4](#_Toc8069917)

[2.1.4.6 Требования к защите информации от несанкционированного доступа 4](#_Toc8069918)

[2.1.4.7 Требования по сохранности информации при авариях 4](#_Toc8069919)

[2.1.4.8 Требования по стандартизации и унификации 4](#_Toc8069920)

[2.1.4.9 Дополнительные требования 4](#_Toc8069921)

[2.1.5 Требования к функциям (задачам), выполняемым системой 4](#_Toc8069922)

[2.1.6 Требования к видам обеспечения 4](#_Toc8069923)

[2.1.6.1 Требования к лингвистическому обеспечению системы 4](#_Toc8069924)

[2.1.6.2 Требования к программному обеспечению системы 4](#_Toc8069925)

[2.1.6.3 Требования к техническому обеспечению 4](#_Toc8069926)

[2.2 Пояснительная записка к программному обеспечению 4](#_Toc8069927)

[2.2.1 Назначение и область применения 4](#_Toc8069928)

[2.2.2 Технические решения 4](#_Toc8069929)

[2.2.2.1 Постановка задачи на разработку программы 4](#_Toc8069930)

[2.2.2.2 Описание структуры программной системы 4](#_Toc8069931)

[2.2.2.3 Описание технологии программирования 4](#_Toc8069932)

[2.2.2.4 Описание взаимодействия с другими программами 4](#_Toc8069933)

[2.2.2.5 Описание и обоснование выбора метода организации входных и выходных данных 4](#_Toc8069934)

[2.2.2.6 Описание алгоритмов и функционирования программы 4](#_Toc8069935)

[2.2.3 Ожидаемые технико-экономические показатели 4](#_Toc8069936)

[2.3 Описание программы 4](#_Toc8069937)

Введение

Йцуйцу

Йцу

1 Анализ предметной области

1.1 Описание предметной области

Ваыа

1.2 Формулировка проблемы и актуальность

Вввва

1.3 Обзор аналогов

йцвйцвйв

2 Программная документация

2.1 Техническое задание на программное обеспечение

2.1.1 Общие сведения

2.1.1.1 Наименование программы

2.1.1.2 Плановые сроки начала и окончания работ

2.1.1.3 Порядок оформления и предъявления заказчику результатов работ

2.1.1.4 Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

2.1.1.5 Определения, обозначения, сокращения

2.1.2 Назначение и цели создания системы

2.1.2.1 Назначение системы

2.1.2.2 Цели создания системы

2.1.3 Характеристика объектов автоматизации

2.1.3.1 Краткие сведения об объекте автоматизации

2.1.3.2 Существующие программное обеспечение

2.1.4 Требования к системе

2.1.4.1 Требование к структуре и функционированию системы

2.1.4.2 Требования к надежности

2.1.4.3 Требования к безопасности

2.1.4.4 Требования к эргономике и технической эстетике

2.1.4.5 Требования к интерфейсу

2.1.4.6 Требования к защите информации от несанкционированного доступа

2.1.4.7 Требования по сохранности информации при авариях

2.1.4.8 Требования по стандартизации и унификации

2.1.4.9 Дополнительные требования

2.1.5 Требования к функциям (задачам), выполняемым системой

2.1.6 Требования к видам обеспечения

2.1.6.1 Требования к лингвистическому обеспечению системы

2.1.6.2 Требования к программному обеспечению системы

2.1.6.3 Требования к техническому обеспечению

2.2 Пояснительная записка к программному обеспечению

2.2.1 Назначение и область применения

2.2.2 Технические решения

2.2.2.1 Постановка задачи на разработку программы

2.2.2.2 Описание структуры программной системы

2.2.2.3 Описание технологии программирования

2.2.2.4 Описание взаимодействия с другими программами

2.2.2.5 Описание и обоснование выбора метода организации входных и выходных данных

2.2.2.6 Описание алгоритмов и функционирования программы

2.2.3 Ожидаемые технико-экономические показатели

2.3 Описание программы

2.3.1

2.3.1.1

2.3.1.2

2.3.2

2.

LDAP – база данных для учётных данных (логины, права, и структура) (иерархическая (дерево)) + стандартные способы доступа.

Kerberos, OIDC… - протоколы (набор соглашений) по передаче данных между участниками(определёнными) сети.

НУЖНО – Сервер(KDC) реализующий(исполняющий) ПРОТОКОЛ, и хранящий данный в LDAP. (KDC выдаёт билеты и всё) (ИНФОРМАЦИЯ О РАСПОЛОЖЕНИИ KDC, алгоритмах шифрования в файле КОНФИГУРАЦИИ)

НУЖНО – Клиентская сторона способная создать нового USER’а и зарегистрироваться на KDC (ИНФОРМАЦИЯ О РАСПОЛОЖЕНИИ KDC, алгоритмах шифрования в файле КОНФИГУРАЦИИ)

НУЖНО – Пристройка(?!) к сервисам, которая может регистрироваться в KDC и принимать сформированные ПРОТОКОЛОМ билеты (ещё есть refresh ticket).

Какой сервер?? Как общаться с сервером?

(OIDC - общается json, Kerberos – не определенно (а как тогда? (GSSAPI ??)))

Критерии сравнения систем авторизации

1. Безопасность: количество участков системы, на которых возможен доступ третьих лиц, который может привести к потере безопасности (нужно описать все участки системы)
2. Удобство для пользователя (наличие SSO?) (кол-во действий для авторизации и необходимость повторной авторизации)
3. Удобство для ресурса (кол-во манипуляций, масштабируемость)
4. Стоимость?
5. Оборудование?
6. Поддержка работы? (выдача учёток)
7. Перспективность технологий (год? Популярность?)

Цель

Цель – оптимизация процессов авторизации и аутентификации в инфраструктуре сервисов компании ООО «БиАйВи»

Описание

Предметной областью является аутентификация/авторизация.

Это неотъемлемая часть всех современных программных продуктов. Необходима и для реализации конфиденциальности, и для безопасности, и для удобства пользователей.

Для успешного прохождения авторизации необходимы:

1. Идентификация – заявление о том, кем является пользователь
2. Аутентификация – проверка подлинности пользователя
3. Авторизация – выдача соответствующих прав доступа к запрашиваемому ресурсу

Одна из основных функций авторизации безопасность. Так как все процессы авторизации происходят удалённо, в рамках компьютерной сети, и защита таких сетей от несанкционированного доступа на сто процентов невозможна. Следовательно, необходимо обеспечить конфиденциальность данных с учётом этих уязвимостей. Первым этапом для решения этой задачи является выделение всех таких небезопасных участков системы.

* Программа, производящая авторизацию со стороны пользователя
* Каналы передачи данных
* Устройство (сервер) и программа производящие авторизацию
* ???

На каждом таком участке системы злоумышленник может производить:

1. Пассивное наблюдение - злоумышленник, перехватывая данные использует их для получения конфиденциальной информации.
2. Воздействие на обменную информацию - злоумышленник, перехватывая и манипулируя информацией, способен получить доступ к конфиденциальным данным/системам.
3. Пароли или иные удостоверяющие данные никогда не пересылаются по сети. Подразумевается, что сетевой трафик может быть прослушан, может произойти подмена сообщения или любая другая пакость.
4. Выдвигается обязательное требование, что информация о паролях/удостоверяющих данных хранится в единственном защищённом месте (Центре распределения ключей Kerberos). Поэтому удостоверяющие данные никогда не сохраняются на том хосте, который пользователь использует для входа/логина. После того, как произошёл первоначальный обмен в рамках аутентификации, этот хост должен забыть сведения о пароле.
5. Хосты/серверы приложений должны быть в состоянии подтвердить свою идентификационную сущность любому, кто запрашивает подобные доказательства.
6. Все коммуникации между аутентифицированными пользователями (клиентами) и сервисами приложений должны иметь возможность быть зашифрованными. С этой целью поддерживаются и могут применяться различные алгоритмы шифрования (все [симметричные](https://pro-ldap.ru/tr/zytrax/tech/encryption.html#symmetric)).

Для

Терминология Kerberos

Как и у всех серьёзных систем, у Kerberos есть своя уникальная терминология, а кое-какие общие термины в контексте Kerberos обретают новый смысл. Постараемся раскрыть их простыми словами (во возможности):

|  |  |
| --- | --- |
| Область действия (Realm) | Realm просто означает совокупность пользователей и серверов приложений, которые охватывает (или имеет о них информацию) Центр распределения ключей (KDC). Так, для того чтобы пользователь подсоединился (или вошёл) в Realm, у Сервера аутентификации (Authentication Server) этой области Realm должны быть сведения об удостоверяющих данных этого пользователя (и другая информация о нём), хранящиеся в защищённой базе данных безопасности того или иного вида (форма хранения не определяется в RFC). В терминологии Microsoft это будет называться Доменом (Domain). Области Realm могут доверять другим Realm, в этом случае доверяющие друг другу области должны быть взаимно аутентифицированными (Cross-Authenticated).  Форма имени Realm — ...@REALM (регистр символов имеет значение). Например, если Realm называется JOE, то его Realm-имя будет @JOE (что отличается от Realm-имени @joe), а если Realm называется EXAMPLE.COM, то его Realm-имя будет @EXAMPLE.COM. (Примечание: Согласно текущей рекомендации ([раздел 6.1 RFC 4120](http://tools.ietf.org/html/rfc4120#section-6.1)) в качестве имени REALM следует использовать доменное имя, которое часто преобразуется в верхний регистр.) Несмотря на то, что последняя форма может напоминать адрес электронной почты, никакого отношения к электронной почте она не имеет. Если буквы большие, это наверняка REALM, а не почта. |
| Принципал (Principal) | Принципал — это строка, полностью идентифицирующая пользователя службы Kerberos. Он имеет форму thing@REALM. Принципал может быть именем сервиса, выполняющегося на хосте (мы будем называть его принципалом сервиса (Service-Principal)), или именем пользователя (мы будем называть его принципалом пользователя (User-Principal)). Принципалы формируют индексное поле для информации об объекте, хранящейся в базе данных безопасности Kerberos (в Центре распределения ключей или KDC). Форматы принципалов для пользователей и сервисов различаются.  Имя принципала пользователя — это приблизительный эквивалент имени пользователя или имени учётной записи. Оно имеет форму principal-name[/instance-name]@REALM (где часть */instance-name* является опциональной). Например, если имя пользователя в принципале пользователя — alice, а Realm — joe, то полный принципал будет alice@joe. Расширение instance-name позволяет любому пользователю иметь более одного принципала. Так, если alice является администратором области Realm joe, имя её принципала будет alice/admin@joe, и у этого принципала будут другие права (и удостоверяющие данные).  Если речь идёт о принципале сервиса, то форма имени принципала становится service-name/QDN@REALM, где QDN — это доменное имя хоста (без точки в конце, как того требует FQDN), на котором работает сервис, а service-name — это специфичная для приложения строка, идентифицирующая сервис на этом хосте. Некоторые типы сервисов используют ключевое слово host. Так, для сервиса ftp, работающего на хосте с именем fileserver.example.com в области Realm @EXAMPLE.COM, имя принципала сервиса будет ftp/fileserver.example.com@EXAMPLE.COM. |
| Разрешение (Ticket) | Разрешение — это структура данных, содержимое которой известно только издателю этого разрешения, и какой-либо стороне или сторонам, к которым это разрешение имеет отношение. Промежуточные хосты, такие как клиентский хост, рассматривают эти разрешения как неразбираемый набор бит и просто передают их на конечный пункт назначения. В Kerberos разрешения могут быть либо Разрешениями на получение разрешения (Ticket Granting Tickets, TGT), — по сути представляют собой доказательства успешно пройденной аутентификации, — либо Сервисными разрешениями (Service Tickets, ST), — выдаются Службой выдачи разрешений (Ticket Granting Service, TGS) и позволяют пользователю получить доступ к требуемому Сервису приложений (Application Service). |

Полезные штуки

1. <https://pro-ldap.ru/tr/zytrax/tech/kerberos.html> - немного теории по Kerberos есть схема (С ссылками. Годно)