Доклад на тему: Протокол Kerberos

Основы информационной безопасности

Ищенко Ирина Олеговна НПИбд-02-22

Содержание

Список иллюстраций

Список таблиц

1 Введение

Протокол Kerberos представляет собой криптографический протокол для аутентификации в компьютерных сетях. Его название происходит от имени мифологического существа Цербер, охраняющего вход в подземный мир. Этот протокол обеспечивает безопасную передачу данных между клиентами и серверами, используя механизм взаимной аутентификации. Протокол работает на основе тикетов, позволяя узлам обмениваться данными по незащищенной сети для подтверждения своей личности. Обе стороны взаимно аутентифицируют друг друга с помощью этого протокола. Kerberos предназначен для обеспечения безопасности и аутентификации.

Современные технологии позволяют нам хранить и обрабатывать огромные объемы информации, однако вместе с этим возрастает и риск ее утечки или компрометации. Именно поэтому протоколы аутентификации, такие как Kerberos, становятся все более важными инструментами для обеспечения безопасности данных.

В этом докладе мы рассмотрим основные принципы работы протокола Kerberos, его преимущества и недостатки, а также примеры его использования.

2 Протокол Kerberos

2.1 Идентификация и аутентификация

Идентификация - присвоение пользователям идентификаторов (уникальных имен или меток) под которыми система “знает” пользователя. Кроме идентификации пользователей, может проводиться идентификация групп пользователей, ресурсов ИС и т.д. Идентификация нужна и для других системных задач, например, для ведения журналов событий. В большинстве случаев идентификация сопровождается аутентификацией. Аутентификация - установление подлинности - проверка принадлежности пользователю предъявленного им идентификатора. Например, в начале сеанса работы в ИС пользователь вводит имя и пароль. На основании этих данных система проводит идентификацию (по имени пользователя) и аутентификацию (сопоставляя имя пользователя и введенный пароль).

Наиболее распространенными на данный момент являются парольные системы аутентификации. У пользователя есть идентификатор и пароль, т.е. секретная информация, известная только пользователю (и возможно - системе), которая используется для прохождения аутентификации.

В зависимости от реализации системы, пароль может быть одноразовым или многоразовым. Операционные системы, как правило, проводят аутентификацию с использованием многоразовых паролей. Совокупность идентификатора, пароля и, возможно, дополнительной информации, служащей для описания пользователя составляют учетную запись пользователя.

2.2 Протокол Kerberos

Протокол Kerberos был разработан в Массачусетском технологическом институте в середине 1980-х годов и сейчас является фактическим стандартом системы централизованной аутентификации и распределения ключей симметричного шифрования. Поддерживается операционными системами семейства Unix, Windows (начиная с Windows'2000), есть реализации для Mac OS.

В сетях Windows (начиная с Windows'2000 Serv.) аутентификация по протоколу Kerberos v.5 (RFC 1510) реализована на уровне доменов. Kerberos является основным протоколом аутентификации в домене, но в целях обеспечения совместимости с с предыдущими версиями, также поддерживается протокол NTLM.

Перед тем, как рассмотреть порядок работы Kerberos, разберем зачем он изначально разрабатывался. Централизованное распределение ключей симметричного шифрования подразумевает, что у каждого абонента сети есть только один основной ключ, который используется для взаимодействия с центром распределения ключей (сервером ключей). Чтобы получить ключ шифрования для защиты обмена данными с другим абонентом, пользователь обращается к серверу ключей, который назначает этому пользователю и соответствующему абоненту сеансовый симметричный ключ.

Протокол Kerberos обеспечивает распределение ключей симметричного шифрования и проверку подлинности пользователей, работающих в незащищенной сети. Реализация Kerberos - это программная система, построенная по архитектуре "клиент-сервер". Клиентская часть устанавливается на все компьютеры защищаемой сети, кроме тех, на которые устанавливаются компоненты сервера Kerberos. В роли клиентов Kerberos могут, в частности, выступать и сетевые серверы (файловые серверы, серверы печати и т.д.).

Серверная часть Kerberos называется центром распределения ключей (англ. Key Distribution Center, сокр. KDC) и состоит из двух компонент:

- сервер аутентификации (англ. Authentication Server, сокр. AS);

- сервер выдачи разрешений (англ. Ticket Granting Server, сокр. TGS).

Каждому субъекту сети сервер Kerberos назначает разделяемый с ним ключ симметричного шифрования и поддерживает базу данных субъектов и их секретных ключей [1]. Схема функционирования протокола Kerberos представлена на рис. 2.1 [2].

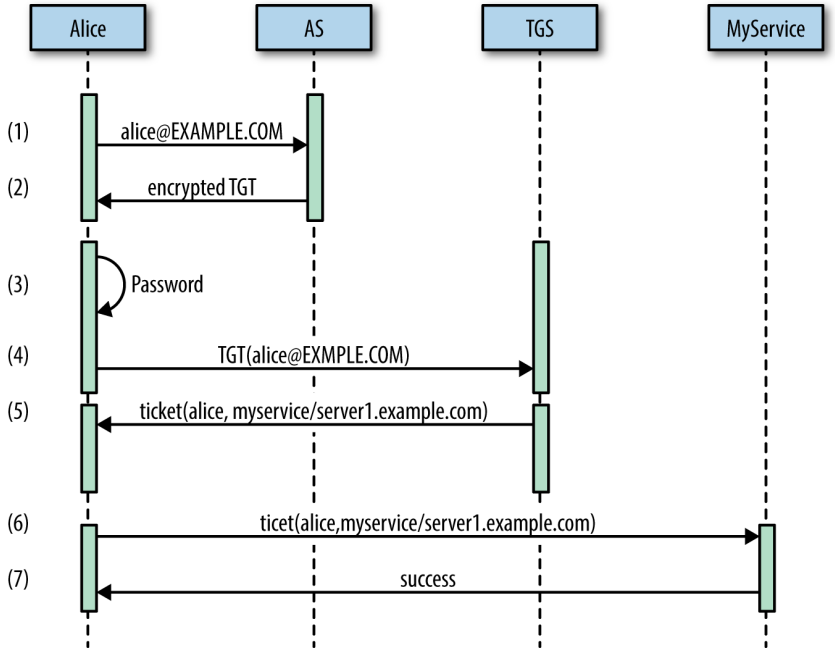

Рис. 2.1: Протокол Kerberos

2.3 Этапы проверки Kerberos

Проверка подлинности Kerberos состоит из 4 этапов, в зависимости от того, какие компоненты взаимодействуют между собой:

1. Вход пользователя / клиента: на этом этапе происходит взаимодействие между пользователем и клиентом. Пользователь вводит в клиент свое имя пользователя и пароль. Затем клиент преобразует этот пароль в ключ шифрования, хранящийся локально. Если это завершится правильно, клиент может начать аутентификацию с помощью AS.

2. Аутентификация клиента / AS: на этом этапе клиент и сервер аутентификации подключаются для аутентификации имени пользователя и проверки того, что они являются частью системы. Затем AS проверяет, что имя пользователя уже задокументировано в системе. В этом случае Клиент и AS обмениваются зашифрованными проверочными сообщениями для проверки друг друга. В конце оба аутентифицируются, соединение устанавливается, и клиент может перейти к аутентификации с помощью службы.

3. Аутентификация клиента / службы: на этом этапе клиент и сервер должны аутентифицировать друг друга в соответствии с практикой взаимной аутентификации. Клиент и сервер обмениваются зашифрованными проверочными сообщениями, как на предыдущем этапе. Если все это проходит, клиент и служба аутентифицируются, и клиент получает разрешение запросить их службу.

4. Клиент / запрос службы: наконец, клиент может запросить именованную службу у сервера службы. Затем сервисный сервер проверяет, доступна ли запрошенная услуга. Если да, сервер службы предоставляет услугу клиенту. Поскольку клиент прошел аутентификацию на всех этапах этого процесса, он может продолжать использовать службу до истечения срока действия разрешений.

2.4 Kerberos пошагово

1. Войти Пользователь вводит свое имя пользователя и пароль. Затем клиент с поддержкой Kerberos преобразует этот пароль в секретный ключ клиента.

2. Запросы клиентов на сервер выдачи билетов Затем клиент отправляет серверу аутентификации текстовое сообщение, содержащее:

- введенное имя пользователя

- название запрашиваемой услуги
- сетевой адрес пользователя
- как долго они запрашивают доступ

3. Сервер проверяет имя пользователя. Имя пользователя проверяется на соответствие проверенным именам пользователей, хранящимся в KDC. Если имя пользователя знакомо, программа продолжится.

4. Выдача билета. Билет возвращается клиенту. Сервер аутентификации отправляет клиенту два зашифрованных сообщения:

- Message A может быть расшифрован с помощью секретного ключа клиента, созданного на шаге 1. Он содержит имя TGS, временную метку, время жизни билета и вновь предоставленный сеансовый ключ сервера предоставления билетов.
- Message B является билетом на выдачу билета и может быть расшифрован только с помощью секретного ключа TGS. Он содержит ваше имя пользователя, имя TGS, метку времени, ваш сетевой адрес, время жизни билета и тот же ключ сеанса TGS.

5. Клиент получает сеансовый ключ TGS. Теперь клиент расшифровывает, message A, используя секретный ключ клиента, предоставляя клиенту доступ к ключу сеанса TGS. Message B хранится локально в зашифрованном состоянии.

6. Клиент запрашивает доступ к службе с сервера Теперь клиент отправляет обратно два сообщения:

- Message C представляет собой незашифрованное сообщение, содержащее имя запрошенной службы, время существования и все еще зашифрованное message B.
- Message D является аутентификатором, зашифрованным с помощью сеансового ключа TGS, и содержит ваше имя и временную метку.

7. Сервер проверяет службу. Затем TGS проверяет, существует ли служба запросов в KDC. Если это так, программа продолжается.

8. Сервер получает сеансовый ключ TGS. Теперь сервер получает все еще зашифрованные message Вотправленные message C. Message B(TGT) затем расшифровывается с использованием секретного ключа TGS сервера, давая серверу сеансовый ключ TGS. Теперь с помощью этого сеансового ключа TGS сервер может расшифровать message D. Теперь у сервера есть отметка времени и имя из message Ви message D(сообщения аутентификатора). Сервер следит за тем, чтобы имена и временные метки совпадали, чтобы предотвратить мошеннические сообщения. Он также проверяет метку времени на соответствие времени жизни билета, чтобы убедиться, что время ожидания не истекло.

9. Сервер генерирует служебный сеансовый ключ. Затем сервер генерирует случайный ключ сеанса службы и еще два сообщения.

- Message E зашифрован с помощью секретного ключа службы и содержит ваше имя, имя запрошенной службы, метку времени, ваш сетевой адрес, время жизни билета и ключ сеанса службы.
- Message F шифруется с помощью сеансового ключа TGS, хранимого как клиентом, так и сервером. Это сообщение содержит всю ту же информацию, message Екроме вашего имени пользователя и сетевого адреса.

10. Клиент получает ключ сеанса обслуживания. Используя ключ сеанса TGS, кэшированный на шаге 5, клиент расшифровывает message F, чтобы получить ключ сеанса службы.

11. Клиент связывается с Сервисом Теперь клиент отправляет еще два сообщения, на этот раз службе:

- Message G- еще одно сообщение аутентификатора, на этот раз зашифро-

ванное с помощью сеансового ключа службы. Он содержит ваше имя и метку времени.

- Message H является копией message E, которая все еще зашифрована служебным секретным ключом.

12. Расшифровка сервисом Message G Затем служба расшифровывает message Hсвой секретный ключ службы, чтобы получить ключ сеанса службы изнутри. С помощью этого ключа сервис расшифровывает message G.

13. Сервис проверяет запрос Затем служба проверяет запрос, сравнивая имена пользователей, временные метки и время жизни из messages G и H.

14. Сервис аутентифицируется для клиента. Затем служба отправляет message I, зашифрованные с помощью сеансового ключа службы, хранимого как службой, так и клиентом. Message I - аутентификатор, содержащий идентификатор службы и временную метку.

15. Клиент проверяет услугу. Затем клиент расшифровывает, message Iиспользуя ключ сеанса службы, кэшированный с шага 10. Затем клиент проверяет идентификатор и временные метки, содержащиеся в нем. Если оба соответствуют ожидаемым результатам, услуга считается безопасной.

16. Свободное общение между клиентом и службой Уверенный в том, что и клиент, и служба взаимно аутентифицированы, Kerberos позволяет клиенту связываться со службой [3].

2.5 Пакеты протокола

В данном блоке рассмотрим пример сессии Kerberos (файл krb-816.cap) с помощью программы Wireshark.

Пакет 1 (рис. 2.2): Клиент отправляет запрос AS-REQ в AS. Пакет содержит заголовки, приведенные ниже.

padata: Это данные для предварительной аутентификации. Клиент вставляет сюда временную метку и шифрует ее с помощью пароля клиента. Цель состоит в том, чтобы избежать повторных атак. cname: Оно содержит имя пользователя (клиента), которое необходимо аутентифицировать. realm: это доменное имя. sname: Это поле содержит название запрашиваемой услуги. till: Оно включает запрошенное время истечения срока действия запрашиваемого билета. etype: Типы шифрования, которые поддерживает клиент.

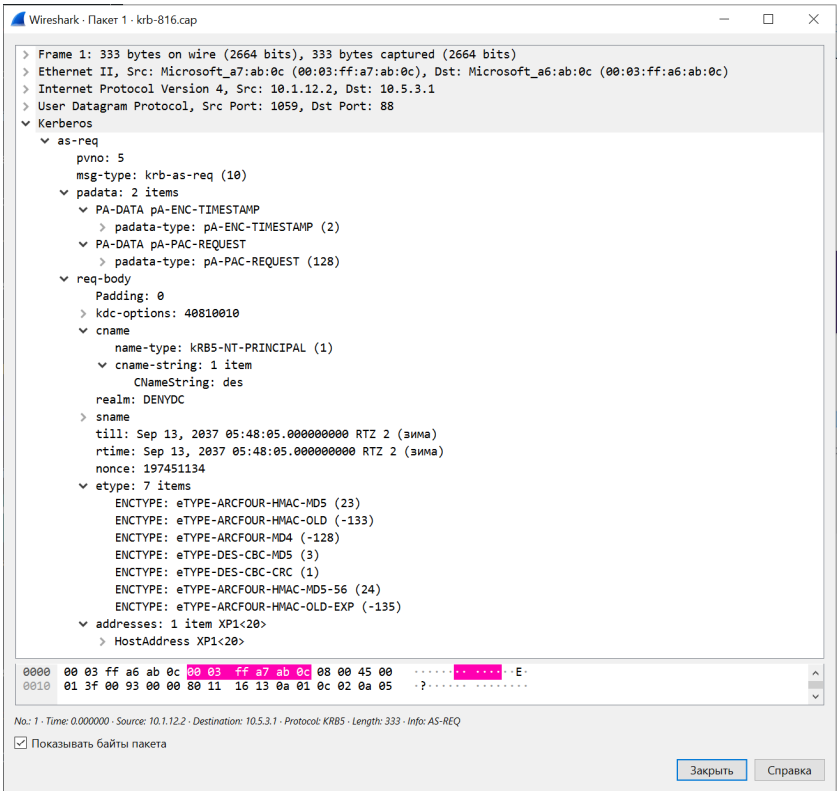

Рис. 2.2: Пакет 1

Пакет 2 (рис. 2.3): Поскольку клиент не указал данные предварительной аутентификации, AS выдает код ошибки, запрашивающий данные предварительной аутентификации.

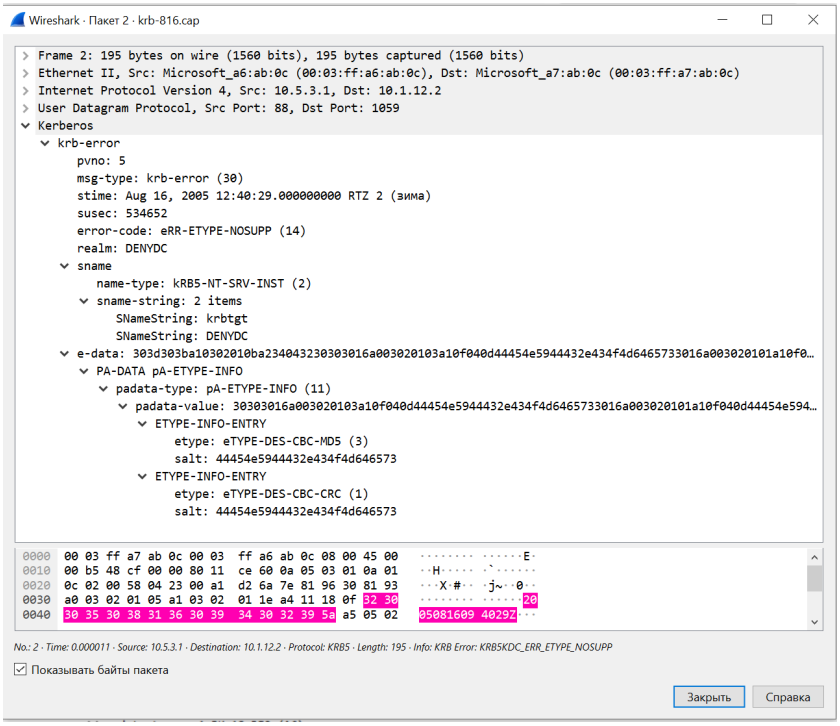

Рис. 2.3: Пакет 2

Пакет 3 (рис. 2.4): На этот раз клиент восстанавливает запрос AS-REQ с данными предварительной аутентификации и отправляет их в AS. Блок padata показывает расшифрованные данные предварительной аутентификации, которые содержат временную метку.

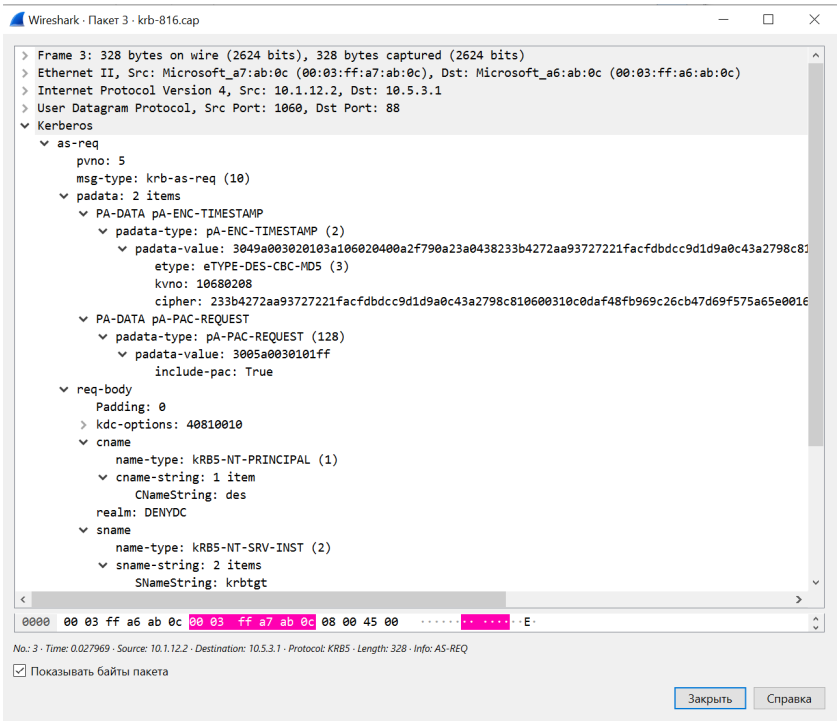

Рис. 2.4: Пакет 3

Пакет 4 (рис. 2.5): AS отвечает пакетом AS-REP. Пакет содержит зашифрованный TGT и сеансовый ключ. Как показано ниже, поскольку TGT зашифрован с помощью главного ключа AS, клиент не сможет его расшифровать. Расшифрованные части - enc-part. Клиент расшифровывает сеансовый ключ с помощью своего пароля.

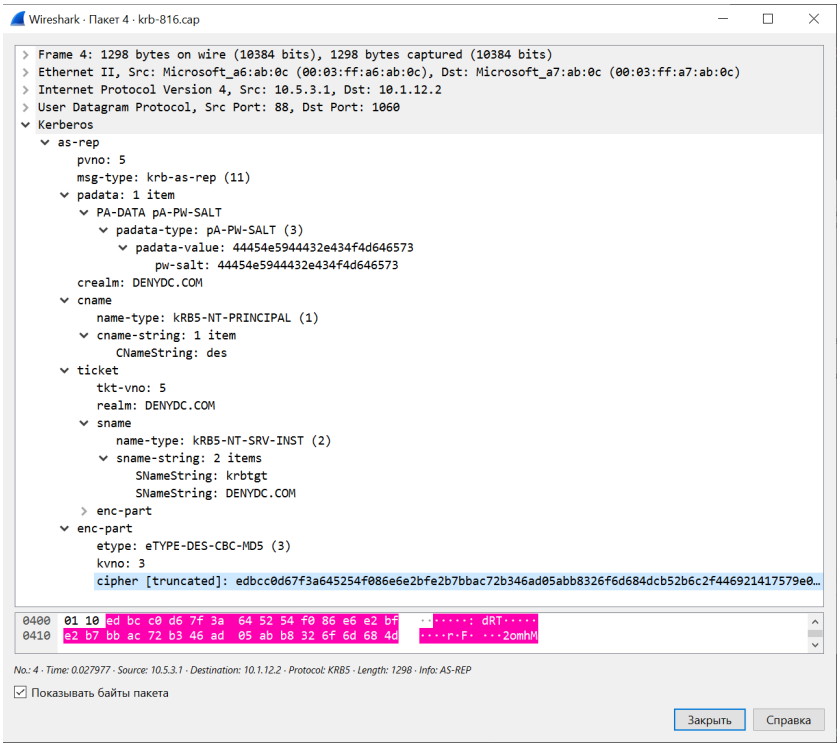

Рис. 2.5: Пакет 4

Пакет 5 (рис. 2.6): После получения TGT клиенты отправляют запрос (TGS-REQ) в TGS. Запрос содержит TGT и аутентификатор, который зашифрован с помощью сеансового ключа клиента. С помощью этого запроса клиент запрашивает у сервера разрешение на обслуживание. Когда TGS получает запрос, он расшифровывает TGT с помощью главного ключа, который используется совместно с AS.

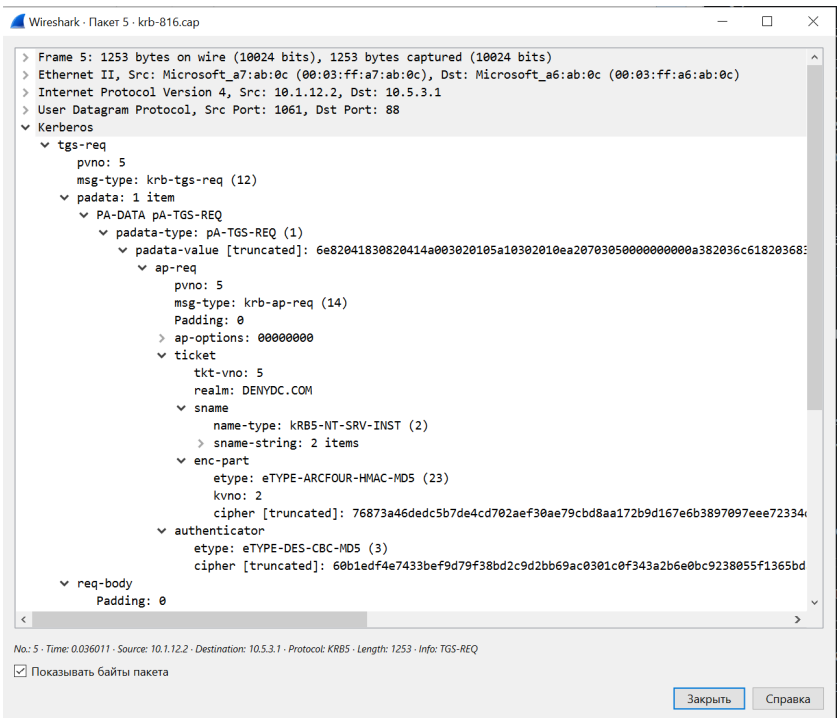

Рис. 2.6: Пакет 5

Пакет 6 (рис. 2.7): TGS создает запрос на обслуживание и шифрует его с помощью другого секретного ключа, который совместно используется TGS и файловым сервером. Запрос на обслуживание включает в себя такую информацию, как название службы, идентификатор клиента, дата истечения срока действия, новый сеансовый ключ, адрес клиента и т.д.(эти данные записаны в cipher, но так как он представлен в усеченном виде, увидеть мы их не можем). Одна копия нового сеансового ключа шифруется с помощью сеансового ключа клиента и вставляется в ответ [4].

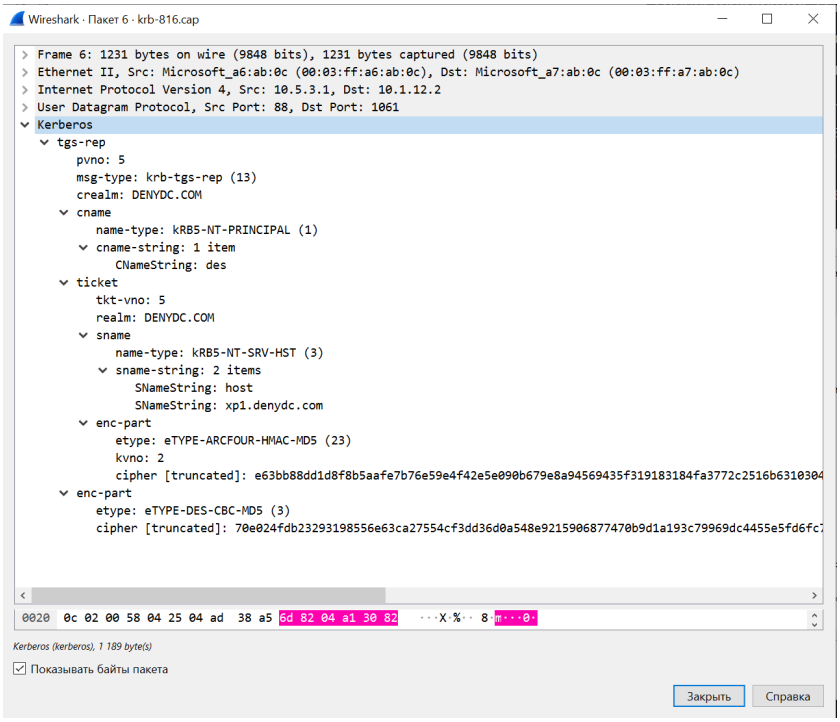

Рис. 2.7: Пакет 6

2.6 Реализация протокола

Что касается реализации протокола Kerberos в Windows, то надо отметить следующее.

Ключ пользователя генерируется на базе его пароля. Таким образом, при использовании слабых паролей эффект от надежной защиты процесса аутентификации будет сведен к нулю. В роли Kerberos-серверов выступают контроллеры домена, на каждом из которых должна работать служба Kerberos Key Distribution Center (KDC). Роль хранилища информации о пользователях и паролях берет на себя служба каталога Active Directory. Ключ, который разделяют между собой сервер аутентификации и сервер выдачи разрешений формируется на основе пароля служебной учетной записи krbtgt - эта запись автоматически создается при организации домена и всегда заблокирована. Microsoft в своих ОС использует расширение Kerberos для применения криптографии с открытым ключом.

Это позволяет осуществлять регистрацию в домене и с помощью смарт-карт, хранящих ключевую информацию и цифровой сертификат пользователя. Использование Kerberos требует синхронизации внутренних часов компьютеров, входящих в домен Windows. Для увеличения уровня защищенности процесса аутентификации пользователей, рекомендуется отключить использование менее надежного протокола NTLM (NT LAN Manager) и оставить только Kerberos (если использования NTLM не требуют устаревшие клиентские ОС).

Кроме того, рекомендуется запретить администраторским учетным записям получать билеты “с правом передачи” (это убережет от некоторых угроз, связанных автоматическим запуском приложений от имени таких записей) [1].

3 Заключение

В данном докладе были рассмотрены принципы работы протокола Kerberos и его реализация.

Основная идея работы протокола заключается в том, что каждый пользователь имеет свой уникальный ключ шифрования. Когда пользователь хочет получить доступ к какому-либо ресурсу (например, файлу), он отправляет запрос на сервер Kerberos. Сервер проверяет подлинность запроса и выдает пользователю временный билет (ticket) для доступа к этому ресурсу. Затем пользователь может использовать этот билет для получения доступа к нужному ресурсу.

Преимуществами протокола Kerberos являются высокая степень безопасности, возможность использования единого ключа шифрования для всех пользователей и серверов, а также простота реализации. Поддержка Kerberos встроена во все основные компьютерные операционные системы, включая Microsoft Windows, Apple macOS, FreeBSD и Linux. Срок действия билетов в Kerberos ограничен. Если билет будет украден, его будет сложно использовать повторно из-за необходимости строгой аутентификации. Пароли никогда не передаются по сети. Обе стороны (клиент и сервер) взаимно аутентифицируют друг друга с помощью протокола.

Однако у него есть и некоторые недостатки, такие как необходимость наличия централизованного сервера Kerberos и возможность атаки со стороны злоумышленников.

Список литературы

1. Лекция 5: Идентификация и аутентификация. Протокол Kerberos.
2. Ben Spivey J.E. Hadoop Security. O’Reilly Media, Inc., 2015. 340 с.
3. Kerberos: знакомство с сетевой аутентификацией.
4. Kerberos Authentication Packet Analysis with Wireshark.