Протокол аутентификации и обмена ключами Kerberos 5

Kerberos 5 – authentification and keys exchange protocol.

Эссе по курсу "Защита информации" [*], кафедра радиотехники,

Московский физико-технический институт (ГУ).

Сергей Уланов, гр.113

21 мая 2005 г.

Содержание

1 Введение

Одно из свойств сети Internet состоит в том, что в используемом семействе протоколов TCP/IP не предусмотрено никаких средств защиты соединения. В частности, получив доступ к шлюзу можно производить (1) перехват пакетов проходящих через шлюз, (2) изменения содержимого пакетов, (3) отправка пакетов имеющие чужой IP адрес отправителя.

Протокол Kerberos является одним из возможных решений этих проблем. Он (1) обеспечивает аутентификацию сторон участвующих в соединении, и (2) предоставляет способ обмена секретным ключом, который может использоваться для защиты передаваемых данных. Этот протокол широко применяющийся в настоящее время. Так например в последних версиях операционной системы Microsoft Windows® для аутентификации по умолчанию используется протокол Kerberos (В Windows NT использовался протокол NTLM).

Kerberos 5 – это последняя на текущий момент версия протокола Kerberos. Она была разработана в Массачусетском Институте Технологий (Massachusetts Institute of Technology, MIT) в 1993 году и стандартизована в RFC 1510.

[*] http://www.re.mipt.ru/infsec

2 Основные принципы

2.1 Сервер аутентификации

В протоколе Kerberos в процессе установления соединения, кроме двух соединяющихся сторон (один из которых является сервером, предоставляющим некоторый сервис, и второй – клиентом, которых хочет соединиться с этим сервером и получить доступ к этому сервису) участвует так же так называемый *сервер аутентификации* (Athentification Server, AS). Иногда сервер аутентификации называют *центром распределения ключей* (Key Distributing Center). Считается, что как клиенты, так и серверы доверяют серверу аутентификации. Для каждого сервера на сервере аутентификации хранится секретный ключ известный только самому серверу и серверу аутентификации. Для установления соединения между сервером аутентификации и клиентом используется разделяемый секретный ключ, соответствующие каждому конкретному клиенту. Как правило, секретный ключ клиента является значением некоторой односторонней хеш-функции от пароля этого клиента (т.е. пользователя).

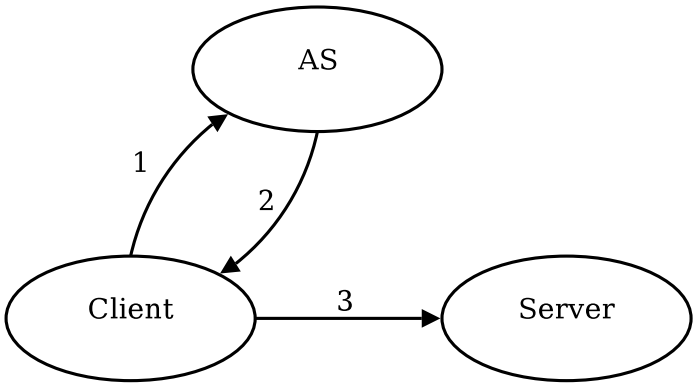
2.2 Мандат и билет

Клиент, желающий получить доступ к определенному сервису, посылает запрос к серверу аутентификации на получение *мандата* (credentials), после чего сервер отправляет назад мандат зашифрованный секретным ключом клиента. Мандат состоит из так называемого *билета* (ticket) и временного ключа. Билет содержит тот же временный ключ и имя клиента зашифрованные секретным ключом сервера, для которого этот билет предназначен. Каждый билет содержит в себе так же поле называемое *временем жизни* (lifetime), которое ограничивает временной интервал, когда этот билет может быть использован. После того, как время жизни истекло, клиент должен получить новый билет.

Кроме билетов предназначенных для получения доступа к сервису, существуют “билеты для получения билетов” называемые *TGT* (ticket-granting ticket). Билет TGT может используется для аутентификации на *TGS* сервере (ticket-granting service) и получения от него билетов для других серверов. В частности TGT используется для аутентификации клиента из одной области на сервере аутентификации, который относится к другой области.

2.3 Аутентификатор

После того как билет получен клиентом, он может использовать его для подключения к серверу более одного раза. Так как при подключении к серверу билет передается в не зашифрованном виде, то сторона перехватившая сообщение содержащее билет могла бы использовать этот билет для аутентификации на сервере. Для предотвращения подобной ситуации, на сервер кроме билета передается дополнительная информация называемая *аутентификатором* (authenticator). Аутентификатор шифруется сессионным ключом и содержит метку времени, для предотвращения повторного использования билета злоумышленником. Таким образом производится доказательство того, что клиент знает секретный ключ, и поэтому действительно является тем, за кого себя выдает.



1. Client → AS: C, S, n
2. AS → Client: $\{K_{C,S}, n\}_{K_C}, T_{C,S}$, где $T_{C,S} = \{K_{C,S}, C, \tau_s, \tau_e\}_{K_S}$
3. Client → Server: $\{A_C\}_{K_{C,S}}, T_{C,S}$

Рис. 1: Процесс соединения

2.4 Области

Протокол Kerberos может быть использован не только в пределах одной организации, но так-же для взаимодействия систем относящихся к разным организациям, т.е. клиент из одной организации может аутентифицироваться на сервере другой организации и использовать сервисы предоставляемые этим сервером. *Областью* (realm) называют организацию имеющую свой сервер аутентификации.

3 Процесс соединения

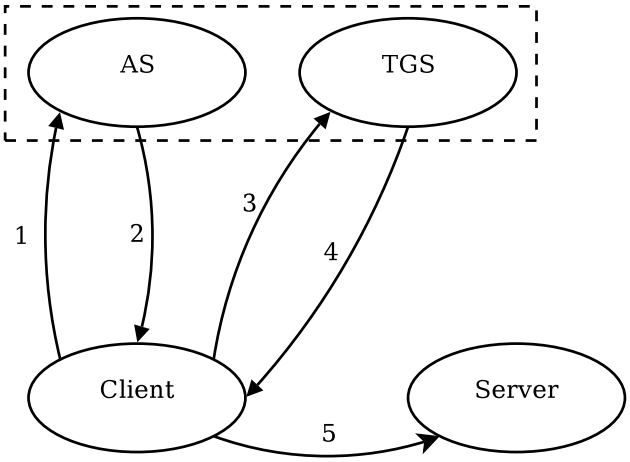
На рисунке 1 показана упрощенная схема процесса установки соединения. На первом шаге клиент (Client) соединяется с сервером аутентификации (AS) и отправляет запрос на получение мандата, в котором указывает свое имя , имя сервиса S и некоторое уникальное число n (например текущее время).

На втором шаге, сервер аутентификации генерирует случайных ключ $K_{C,S}$ называемый *сессионным ключом*, после чего создает билет $T_{C,S} = \{K_{C,S}, C, \tau_s, \tau_e\}_{K_S}$, где τ_s и τ_e задают интервал времени, когда этот билет может быть использован. Т.к. билет зашифрован секретным ключом K_S который известен только серверу аутентификации и серверу S, то никакая третья сторона, включая клиента C, не может прочитать, либо изменить имя клиента C содержащееся в билете, так же как и время действия этого билета. После того, как билет получен, клиенту отправляется мандат[1] $\{\{K_{C,S}, n\}_{K_C}, T_{C,S}\}$. Мандат содержит сессионный ключ в зашифрованном виде, поэтому третья сторона, перехватившая это сообщение, не сможет использовать содержащийся в нем билет для соединения с сервером S.

Клиент получив мандат расшифровывает его, и проверяет число n, после чего сохраняет сессионный ключ и билет у себя в кэше, для дальнейшего использования.

На третьем шаге, клиент желая установить соединение с сервером S отправляет

[1] В четвертой версии протокола мандат имел вид $\{K_{C,S}, n, T_{C,S}\}_{K_C}$. Однако очевидно, что в шифровании билета нет необходимости, поэтому формат мандата был изменен.



1. Client → AS: C, tgs, n
2. AS → Client: $\{K_{C,tgs}, n\}_{K_C}, T_{C,tgs}$, где $T_{C,tgs} = \{K_{C,tgs}, C, \tau_s, \tau_e\}_{K_tgs}$
3. Client → TGS: $\{A_C\}_{K_{C,tgs}}, T_{C,tgs}, S, n$
4. TGS → Client: $\{K_{C,S}, n\}_{K_{C,tgs}}, T_{C,S}$, где $T_{C,S} = \{K_{C,S}, C, \tau_s, \tau_e\}_{K_S}$
5. Client → Server: $\{A_C\}_{K_{C,S}}, T_{C,S}$

Рис. 2: Процесс соединения с использованием сервера TGS

ему предварительно сгенерированный аутентификатор A_C зашифрованный сессионным ключом и билет $T_{C,S}$. Сервер, получив билет, получает из него сессионный ключ и расшифровывает аутентификатор, после чего проверяет метку времени которая содержится в этом аутентификаторе. Таким образом сервер убеждается, что клиенту пославшему запрос так же известен сессионный ключ, и следовательно можно считать, что он является тем, за кого себя выдает.

В случае если клиент требует аутентификации сервера, сервер посылает сообщение зашифрованное сессионным ключом, чем доказывает, что он ему этот ключ известен, и следовательно известен ключ K_S.

В дальнейшем сессионный ключ $K_{C,S}$ используется для шифрования сообщений между сервером и клиентом во время установленного соединения.

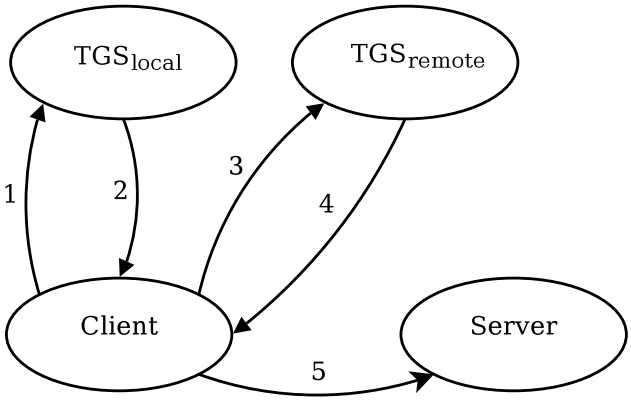
3.1 Использованием сервера TGS для получения билета

Приведенная в предыдущем разделе схема установления соединения как правило используется только для получения билета для сервера TGS, который потом выдает билеты для доступа к остальным сервисам. Это делается для того, чтобы уменьшить риск того, что секретный ключ клиента будет похищен, т.к. при использовании сервера TGS хранение секретного ключа (полученного из пароля) клиента не требуется, и этот ключ используется то при получении билета TGT.

Сервер TGS и сервер аутентификации могут быть физически разделены, и работать на различных физических серверах, однако в большинстве случаев они совмещены.

Упрощенная схема аутентификации при использовании TGS показана на рисунке 2.

Сначала (шаги 1, 2) сервер аутентификации выдает клиенту мандат для доступа



1. Client $\rightarrow$ TGS$_{local}$: $\{A_C\}_{K_{C,tgs_{local}}}, T_{C,tgs_{local}}, tgs_{remote}$
2. TGS$_{local}$ $\rightarrow$ Client: $\{K_{C,tgs_{remote}}\}_{K_{tgs_{local}}}, T_{C,tgs_{remote}}$, где $T_{C,tgs} = \{K_{C,tgs_{remote}}, C, \tau_s, \tau_e\}_{K_t gs_{remote}}$
3. Client $\rightarrow$ TGS$_{remote}$: $\{A_C\}_{K_{C,tgs_{remote}}}, T_{C,tgs_{remote}}, S, n$
4. TGS$_{remote}$ $\rightarrow$ Client: $\{K_{C,S}, n\}_{K_{C,tgs_{remote}}}, T_{C,S}$, где $T_{C,S} = \{K_{C,S}, C, \tau_s, \tau_e\}_{K_S}$
5. Client $\rightarrow$ Server: $\{A_C\}_{K_{C,S}}, T_{C,S}$

Рис. 3: Процесс межобластного соединения

к серверу TGS. Это происходит так же как и в описанном ранее процессе получения билета для сервера, когда сервер TGS не используется.

На третьем шаге клиент подключается к серверу TGS и отправляет ему свой аутентификатор и билет полученный от AS, имя сервиса S, для которого он желает получить мандат, и случайное число n. После этого сервер TGS генерирует случайных ключ $K_{C,S}$ и билет $T_{C,S}$, и отправляет назад мандат $\{\{K_{C,S}, n\}_{K_{C,tgs}}, T_{C,S}\}$, в котором сессионный ключ зашифрован ключом $K_{C,tgs}$, а не K_C, т.е. клиенту уже не нужно использовать K_C для получения билета, и поэтому от пользователя требуется ввести пароль только один раз, при получении билета TGT, и при получении билетов для различных сервисов ввод пароля не требуется.

После получения мандата, клиент использует его для подключения к сервису S так же, как и в случае когда TGS сервер не используется.

3.2 Межобластная аутентификация

Как было сказано ранее протокол Kerberos имеет возможность аутентифицировать клиентов из одной области для использования сервисов другой области.

Если создан “межобластной” ключ (inter-realm key) для двух областей то клиент находящийся в одной из этих областей может аутентифицироваться для использования сервиса предоставляемого сервером из другой области. Клиент из некоторой области A для того чтобы аутентифицироваться на сервере относящемуся к области B запрашивает у сервера аутентификации из области A билет TGT для доступа к серверу

аутентификации области B. После этого билет TGT используется для получения у сервера аутентификации области B билета для доступа к сервису из области B. Кроме этого полученный билет TGT может быть использован для получения билета TGT для связи с сервером аутентификации из третьей области C, если серверы аутентификации областей B и C имеют общий межобластной ключ.

Две области считаются *связанными* либо если они имеют общий межобластной ключ, либо если локальный сервер аутентификации имеет общий межобластной ключ с областью, которая связана с удаленной областью. Обычно области образуют иерархическую структуру, и тогда, в случае если две области не имеют общего межобластного ключа, иерархическая организация позволяет легко установить путь аутентификации между двумя областями[2].

Для того, чтобы сервер мог знать, какие серверы аутентификации участвовали в процессе аутентификации клиента из другой области, каждый билет содержит список серверов использованных при получении этого билета.

Упрощенная схема установки соединения между системами находящимися в различных областях для которых имеется межобластной ключ, изображена на рисунке 3. Процесс соединения между системами находящимися в различных связанных областях которые не имеют межобластного ключа аналогичен этой схеме, но в нем последовательно участвуют несколько TGS серверов.

Список литературы

[1] RFC 1510, September 1993.

[2] *B. Clifford Neuman and Theodore T'so*, Kerberos: An Authentication Service for Computer Networks, IEEE Communications, 32(9) pp33-38. September 1994.

[3] *John T. Kohl, B. Clifford Neuman, and Theodore Y. T'so*, The Evolution of the Kerberos Authentication System. Distributed Open Systems, pp78-94. IEEE Computer Society Press, 1994.

[4] Kerberos page at MIT, http://web.mit.edu/kerberos/www .

[2] В четвертой версии протокола получение билета по цепочке было невозможно, и всегда для соединения систем из разных областей требовалось существования межобластного ключа для этих областей.