Страж у моста или установка и настройка Heimdal Kerberos в гетерогенной сети.

mkondrin из hppi.troitsk.ru

Аннотация

Установка и конфигурирование сервера Heimdal Kerberos и установка клиентского программного обеспечения на компьютерах с операционной системой Linux и Windows XP.

1

Любой системный администратор по мере расширения и развития подконтрольной ему локальной сети сталкивается с проблемой управления регистрационной информацией локальных пользователей. Задачи, возникающие перед администратором, связаны как с синхронизацией системных учётных записей большого числа пользователей на все увеличивающемся числе рабочих станций, но и с управлением доступа пользователей к сетевым сервисам внутри локальной сети. Причём, мало того, что сервисы могут работать на разных компьютерах, но определённые службы предпочитают не прибегать к системной регистрационной информации, а держать собственные учётные записи для авторизованных пользователей. Решением этих проблем является введение единой системы регистрации в локальной сети с помощью протокола Kerberos.

Чтобы подробнее разобраться с возникающими задачами рассмотрим ситуацию с организацией распределённых вычислительной системы. Разумеется, чем больше компьютеров входят в кластер, тем его вычислительная мощность выше. Наиболее распространенные сегодня системы кластерных вычислений - это Parallel Virtual Machine и Message Passing Interface. Обе позволяют пользователю, в данном случае разработчику программ, пересылать куски данных нуждающихся в обработке между узлами кластера и синхронизировать получение результатов с разных узлов. Это, так сказать, фасад системы. За кулисами тем не менее происходит обращение к удалённому командному интерпретатору и вызов определённых программ в нем. Т.е. администратору такой системы необходимо добиться, чтобы пользователи кластера имели доступ к командному интерпретатору на узлах кластера и

более того этот доступ должен быть беспарольным для каждой пары компьютеров в кластере. Не очень-то удобна система, где запуск нескольких параллельных копий программы, требует от пользователя регистрации на каждом из узлов кластера. Таким образом, во-первых, информация о пользователях должна совпадать на всех этих компьютерах. Одним из вариантов решения - это иметь одинаковые копии `/etc/passwd` и `/etc/shadow` на каждом из узлов с их последующем обновления с помощью скриптов при добавлении нового пользователя. Во-вторых, если в качестве удалённого командного интерпретатора используется rsh, то добиться беспарольного входа с помощью внесения всех компьютеров входящих в этот кластер в файл /etc/hosts.equiv (этот файл также должен совпадать на всех узлах кластера). Однако, использование rsh гарантирует вам проблемы с безопасностью, если предположить возможность доступа к кластеру извне локальной сети, который, как вы помните, должен быть беспарольным. Можно сконфигурировать доступ по адресу компьютера (с помощью tcp-wrappers) и бороться с ip-spoofing внешними средствами или настраивать openssh в качестве удалённого командного интерпретатора. В последнем случае вам придётся мирится с тем, что процессорные циклы будут расходоваться не на расчёт, а на кодирование/раскодирование блоков данных. Тем не менее ни одно из этих решений нельзя считать удачным.

Далеко не каждому системному администратору приходится сталкиваться с настройками вычислительных кластеров. Но именно эта проблема построения распределённой вычислительной системы Athena заставила в начале 80-х годов программистов из Массачусетского Технологического Института разработать и внедрить протокол удалённой аутентификации пользователей. Комбинирование специальных криптографических средств, позволяла с одной стороны свести на нет вероятность перехвата паролей и иметь шифрованный канал для передачи данных между компьютерами (эта возможность могла отключаться по желанию пользователя), а с другой - иметь систему с единой регистрацией (single sign-on), что даёт возможность пользователю регистрироваться один раз при входе в систему, и в дальнейшем иметь свободный доступ к сетевым ресурсам на основе этой регистрации.

Понятно, что число пользователей, которым такая функциональность была бы удобна, значительно превышает число нуждающихся в распределенных системах, и в дальнейшем этот протокол, получивший название Kerberos (по имени трёхголового пса стерегущего вход в царство мёртвых из древнегреческих мифов), стал широко применяться независимо от Athena в крупных финансовых и академических учреждениях. Вкратце, история и создания Kerberos выглядит таким образом - в 1983 году была начата работа по созданию системы Athena. Работу финансировали компании IBM и DEC и в 1987 году была выпущена первая версия протокола Kerberos (Kerberos 4). Дальнейшая эксплуатация выявила недостатки протокола и обновлённый вариант Kerberos5 вышел в свет в 1993 году. В настоящее время

протокол 4 практически не используется, но в реализациях протокола от МТИ совместимость с предыдущей версией продолжает сохраняться. К 1993 году протокол Kerberos уже завоевал популярность и многие компании, разработчики программного обеспечения стремились использовать его в своих программных продуктах. Но тут имел место юридический казус - дело в том, что то время в США ещё действовали законы, введённые ещё во время холодной войны, запрещающие экспорт военных технологий. Поскольку криптографическая защита использованная в Kerberos классифицировалась как военная технология, это создавало препятствия для использования его в программных продуктах сторонних фирм (и распространению их за пределами США). Для решения этой проблемы была выпущена версия протокола Kerberos 4 из которого была изъята вся сильная криптография. Эта реализация Kerberos получила название Bones (Кости) и ограничения на её экспорт уже не действовали. В 1997 году группа программистов из Стокгольмского Королевского Университета, взяв за основу Bones , проделала обратную работу и вставила недостающую криптографическую функциональность. Вот так экспортные ограничения Соединённых Штатов способствовали развитию европейского hi-tech. Впоследствии ими же была выпущена реализация протокола Kerberos 5 получившая название Heimdal, о которой и пойдёт в дальнейшем речь в этой статье. Heimdal (Хеймдалль) - это божество из скандинавского пантеона, чьи функции состояли в охране стратегических коммуникаций, а именно - моста разделяющего Асгард и Мидгард. Так же как и в случае с древнегреческим прототипом Kerberos здесь также эксплуатируется образ неподкупного стража. Интересно отметить, что мотивом для программистов из Стокгольмского университета, также как и для их коллег из МТИ, являлась задача обеспечения публичного доступа к вычислительному кластеру. Последним эпизодом из истории Kerberos стало объявление компанией Microsoft в 1999 году о поддержке Kerberos в своей будущей операционной системе NT5.0 (впоследствии названной Windows2000), что действительно было реализовано в качестве компонента Active Directory. В настоящее время Kerberos является промышленным стандартом удалённой аутентификации пользователей.

Как с точки зрения пользователя выглядит работа с Kerberos? На пользовательском уровне все реализации Kerberos выглядят однотипно, поэтому рассмотрим как это происходит в моем случае. Регистрация в Kerberos осуществляется командой kinit, которая автоматически вызывается при моем входе на рабочую станцию под управлением WindowsXP. В результате мне выдаётся основной документ, удостоверяющий мою личность в Kerberos - т.н. начальный билет (TGT - ticket granting ticket). При этом у меня всегда есть возможность зарегистрироваться в Kerberos под другим именем с помощью все той-же команды kinit, что приводит к обновлению начального билета. Теперь предположим, что я хочу посмотреть логи на своём роутере/файрволле под управлением Линукс. Я открываю терминал cygwin. Первое

что можно сделать - просмотреть имеющиеся билетики:

```
mike@alex ~
\$ klist
Credentials cache: FILE:/tmp/krb5cc_1017
        Principal: mike@HPPI.TROITSK.RU

  Issued           Expires          Principal
May 29 22:18:22  May 30 22:18:22  krbtgt/HPPI.TROITSK.RU@HPPI.TROITSK.RU
```

В списке как вы видите имеется только один билет - тот самый TGT. Credential Cache - это файл, в котором хранятся полученные мной сертификаты. По-умолчанию его название - это комбинация /tmp/krb5cc_ и id пользователя. Срок действия любого билетика ограничен по времени - это снижает интерес к его перехвату со стороны злоумышленника. Теперь я запускаю командный интерпретатор на удаленном компьютере:

```
\$ telnet -F relay
Trying 192.168.1.254...
Connected to relay.hppi.troitsk.ru.
Escape character is '^]'.
Waiting for encryption to be negotiated...
[ Trying mutual KERBEROS5 (host/relay.hppi.troitsk.ru@HPPI.TROITSK.RU)...
[ Kerberos V5 accepts you as ``mike@HPPI.TROITSK.RU'' ]
[ Kerberos V5 accepted forwarded credentials ]
Encryption negotiated.
\$klist
Credentials cache: FILE:/tmp/krb5cc_1000
        Principal: mike@HPPI.TROITSK.RU

  Issued           Expires          Principal
May 29 22:21:16  May 30 22:18:22  krbtgt/HPPI.TROITSK.RU@HPPI.TROITSK.RU
```

Вот пример магии Kerberos в действии - воспользовавшись моим первоначальным билетиком, Kerberos прозрачно для пользователя организует доступ к сетевому ресурсу (в данном случае он фигурирует под именем host/relay.hppi.troitsk.ru@HPPI и более того - Kerberos автоматически шифрует весь сетевой трафик между клиентом и сервером. Список билетов при этом не меняется - ключ -F позволяет перемещать имеющиеся у меня билетики с компьютера на компьютер. Закрытие telnet-сессии автоматически очищает кэш с помощью вызова команды kdestroy, так что

вы можете не опасаться, что ваш кэш может быть использован кем-то ещё. Так как начальный билетик по-прежнему со мной - то это позволяет мне получить доступ к другому компьютеру, серверу Kerberos.

```
\$telnet -F kenga
Trying 192.168.1.253...
Connected to kenga.hppi.troitsk.ru.
Escape character is '^]'.
Waiting for encryption to be negotiated...
[ Trying mutual KERBEROS5 (host/kenga.hppi.troitsk.ru@HPPI.TROITSK.RU)...
[ Kerberos V5 accepts you as ``mike@HPPI.TROITSK.RU'' ]
[ Kerberos V5 accepted forwarded credentials ]
Encryption negotiated.
mike@kenga:~\$
```

Точно так же я могу получить доступ к любому сетевому сервису - например, к локальному ftp-серверу.

```
mike@kenga:~\$ ftp kenga
Connected to kenga.hppi.troitsk.ru.
220 kenga FTP server (Version 6.00+Heimdal 0.6.3) ready.
Trying GSSAPI...
Authenticated to <host/kenga.hppi.troitsk.ru@HPPI.TROITSK.RU>
Authentication successful.

Name (kenga:mike):
S:232-Linux 2.4.25.
S:232 User mike logged in.
S:230 Password not necessary
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
S:221 Goodbye.
```

Заметьте, что во всех случаях мне не потребовался ввод пароля. Доступ ко всем компьютерам мне предоставлялся на основании моего TGT. Все операции с Kerberos (выдача билетов, добавление/удаление пользователей и т.д.) фиксируются в его логах, и можете убедиться, что вся моя активность с перемещениями от компьютера к компьютеру зафиксирована в файле `/var/log/krb5kdc.log`. Для удобства я разделил кусок log-файла на 4 части, соответствующие процессам получения доступа к моей рабочей станции, удалённого доступа к двум серверам и серверу ftp.

```
2005-05-29T22:18:22 AS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.45 fo
2005-05-29T22:18:22 Using des-cbc-crc/des-cbc-md5
2005-05-29T22:18:22 Requested flags: renewable_ok, renewable, forwardable
2005-05-29T22:18:23 sending 574 bytes to IPv4:192.168.1.45
2005-05-29T22:18:23 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.45 f
2005-05-29T22:18:23 sending 593 bytes to IPv4:192.168.1.45
......
2005-05-29T22:21:15 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.45 f
2005-05-29T22:21:16 sending 546 bytes to IPv4:192.168.1.45
2005-05-29T22:21:16 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.45 f
2005-05-29T22:21:16 sending 589 bytes to IPv4:192.168.1.45
......
2005-05-29T22:21:43 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.254
2005-05-29T22:21:44 sending 550 bytes to IPv4:192.168.1.254
2005-05-29T22:21:44 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.254
2005-05-29T22:21:44 sending 593 bytes to IPv4:192.168.1.254
......
2005-05-29T22:22:27 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253
2005-05-29T22:22:27 Server not found in database: ftp/kenga.hppi.troitsk.
2005-05-29T22:22:27 sending 145 bytes to IPv4:192.168.1.253
2005-05-29T22:22:27 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253
2005-05-29T22:22:27 sending 550 bytes to IPv4:192.168.1.253
2005-05-29T22:22:28 TGS-REQ mike@HPPI.TROITSK.RU from IPv4:192.168.1.253
2005-05-29T22:22:28 sending 546 bytes to IPv4:192.168.1.253
```

Опытный системный администратор может сказать: “Как же так - протоколы telnet и ftp не безопасны и правильнее использовать openssh. И ничего удивительного в беспарольной аутентификации нет - ту же самую функциональность обеспечивает ssh, предоставляя возможность регистрироваться по парам публичных/секретных ключей”. Все правильно, только для того чтобы в сети из N компьютеров, обеспечить доступ по ssh с любого из этих компьютеров на любой другой, вам придётся в общем случае проделать N^2*число пользователей перемещений публичных ключей (каждая пара компьютеров в сети обменивается публичными ключами пользователей) и в случае больших сетей трудноосуществимо. В то же время с помощью Kerberos вам нужно только зарегистрировать каждый из компьютеров на сервере Kerberos и иметь один ключ на каждом из хостов (в отличии от ssh в Kerberos используется симметричное шифрование)- итого 2N операций. Что же касается первой части вопроса, то я использую специальный керберизованный вариант telnet, что защищает соединения между хостами не хуже, чем ssh.

Главный недостаток стандартного telnet (и не только его) состоит в том, что при аутентификации на удалённом компьютере, telnet пересылает пароль пользователя по сети в виде открытого текста, что позволяет злоумышленнику перехватить его. Ssh обходит эту опасность с помощью несимметричного шифрования пароля. А каким же образом Kerberos-у удаётся избегать брешей в защите связанных с удалённой аутентификацией?

Делается это с помощью уже упоминавшихся ранее билетиков/сертификатов (tickets/credentials - оба слова используются как синонимы) - специальным образом изготовленных и упакованных шифровальных ключей. В Kerberos как пользователь так и сетевая служба не различаются между собой и именуются principal (принципал). Принципал - юридический термин, означающий лицо, поручающее агенту совершить сделку от его имени, что весьма точно описывает функции принципалов в Kerberos. Принципалы определяются своим именем и паролем, причём в случае сетевой службы в качестве этого пароля выступает ключ хранящийся на том же компьютере, где работает защищаемый сервис. База данных принципалов хранится на сервере Kerberos и при необходимости проверить аутентичность пользователя или сервиса компьютеры, объединённые в сектора (realms), соединяются с этим сервером. По-поводу терминологии: точный перевод “realm”(“царство”) применительно к Kerberos не прижился, а иногда используемый термин домен не кажется мне удачным, поскольку слово и так перегружено. Так же как и в случае с DNS главный контроллер сектора Kerberos (Key Distribution Center, KDC - точный) может иметь как дополнительные, slave, контроллеры (что позволяет обеспечить бесперебойную работу при выходе из строя основного контроллера), так и в одиночку держать несколько секторов. Типичное имя принципала, например сервиса удаленного доступа к командному интерпретатору компьютера, выглядит таким вот образом - `host/kdc.myrealm.ru@MYREALM.RU`, что обозначает сервис с основным именем (primary name) host и характеристикой (instance) kdc.myrealm.ru, принадлежащий сектору MYREALM.RU. Разделение имен принципалов на несколько частей позволяет различать, с одной стороны, различные службы, работающие на одном хосте (с помощью primary name, host в нашем случае), а с другой, среди нескольких однотипных служб, работающих в сети, выбирать конкретную, запущенную на определённом сервере (с помощью поля instance, которая в нашем случае совпадает с именем компьютера). Название сектора не обязано повторять доменное имя сети, но именно такое правило наименований считается устоявшейся практикой.

Теперь посмотрим, что происходит если пользователь `joeuser` (а точнее принципал `joeuser@MYREALM.RU` - поле instance для “живых” пользователей обычно не используется) пытается получить доступ к этому серверу. Предполагается, что как клиентская программа telnet, так и telnetd демон собраны с поддержкой kerberos

(обе этих программы входят в дистрибутив Heimdal). Разумеется как сам сервер так и пользователь должны быть зарегистрированы в одном и том же секторе kerberos. Как обычно, сеанс подключения к серверу начинается с того, что пользователь запускает telnet со своим регистрационным именем и именем удалённого компьютера `telnet -l joeuser kdc.myrealm.ru`. Клиентская программа после этого обращается к KDC с просьбой предоставить для joeuser доступ к хосту kdc.myrealm.ru. Kerberos генерирует по определённым правилам шифровальный ключ (session key) и разыскивает по своей базе данных принципалов joeuser и `host/kdc.myrealm.ru`, а также их пароли (ключи). Если пароли найдены (в противном случае сеанс заканчивается аварийно - т.к. кто-то из этих двух принципалов не зарегистрирован в секторе Kerberos), kerberos приступает к генерированию сессионного ключа (session key). С помощью ключа принципала `host/kdc.myrealm.ru` он шифрует сгенерированный session key вместе с именем пользователя joeuser, потом прилагает к зашифрованному блоку вторую копию того же session key и снова шифрует получившийся билетик с помощью пароля пользователя. После этого пакет отсылается клиентской программе. Если к этому времени пользователь набрал свой пароль и он совпадает с паролем использованным kerberos-ом при шифровке, то клиентская программа сумеет расшифровать билетик. Далее, половина билетика (с session key) остаётся пользователю, а вторая, которая не может быть расшифрована клиентом, поскольку ключ сервера ей неизвестен, пересылается на сервер. Сервер расшифровывает её с помощью своего ключа и таким образом узнает как session key так и имя пользователя, что позволяет серверу авторизовать его своими собственными средствами. Заметим, что помимо регистрации в этом случае имеется возможность использовать полученный session key, поскольку в итоге он известен как клиенту так и серверу, для шифрования сетевого траффика в рамках данной сессии, чем многие приложения и пользуются. Схематически процесс обмена сертификами показан на рисунке.

Что ещё интересного в используемом Kerberos-ом механизме аутентификации? Дело в том, что Kerberos оказывается ещё более параноидальным, чем мы предполагали вначале, т.е. считает ненадёжными не только сетевые соединения (не один из паролей, как вы могли заметить, не передаётся в открытую по сети), но и клиентский и даже серверный компьютер. Для каждого из них ключ/пароль партнёра так и остаётся неизвестным - только session key, который для потенциального взломщика интереса не представляет, поскольку используется только в данной сессии. Как известно безопасность и удобство пользователей вещи взаимоисключающие. Но учёным из МТИ удалось найти удачный компромисс, придумав ещё одну замечательную вещь - ticket granting ticket (TGT). Если пытаться переводить этот термин на русский - “билет для выдачи других билетов”, что-то вроде единого проездного. Смысл его в том, что пользователь сети проходит полностью процедуру

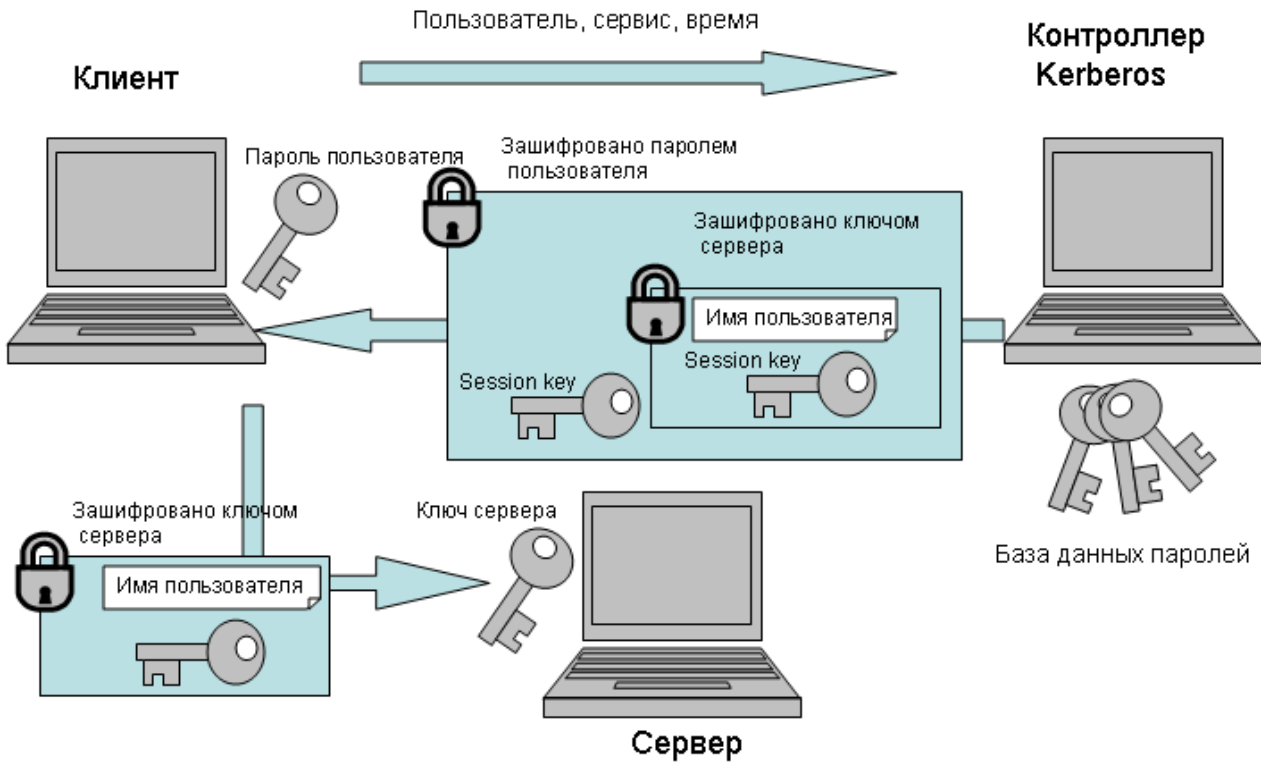

Рис. 1: Схема взаимодействия клиентского компьютера, контроллера Kerberos и сервера.

регистрации (с вводом имени и пароля) в своём секторе только один раз, а сертификат, полученный в результате, затем используется клиентскими приложениями как эквивалент пользовательского пароля для получения доступа к сетевым сервисам. Процесс выдачи TGT ничем не отличается от описанного выше, только в качестве службы, к которой пользователь получает доступ, выступает сам Kerberos - его Ticket Granting Service. После получения TGT записывается на диск клиентского компьютера и затем по мере надобности извлекается оттуда. В схеме описанной выше, session-key, закодированный в TGT, используется вместо пароля пользователя при шифровании сертификата, предоставляемого для доступа к сетевому ресурсу. И хотя TGT хранится в кэше на диске рабочей станции пользователя, угроза связанная с его перехватом не столь уж серьёзна, поскольку его действенность ограничена по времени. TGT позволяет организовать в корпоративной сети single sign-on (система единой регистрации) систему. Например, с помощью замены системного login-а на керберизованный аналог (программа с тем же именем входит в состав Heimdal Kerberos), при входе на рабочую станцию пользователь получает TGT, который затем используется для генерирования билетиков, специфичных для какой-нибудь из сетевых служб.

Таким образом следует помнить, что при любой аутентификации в Kerberos всегда участвуют два принципала - один соответствует пользователю, пытающемуся получить доступ к сервису, а второй - самому этому сервису. Мы не будем рассматривать более сложный механизм аутентификации, так называемый user-to-user, когда сетевой сервис не имеет собственной записи в базе данных Kerberos, а использует сертификаты пользователя, запустившего сервис. Аутентификация user-to-user могла бы быть полезна, например, для X-серверов, но и для них эта методика не получила большого распространения.

Так в общих чертах выглядит принцип работы Kerberos. Многие детали реализации протокола Kerberos при этом пришлось опустить. В частности, формат сертификата в действительности более сложен, чем описано здесь. В него входит время выдачи и срок годности сертификата, адрес компьютера, на который отсылается сертификат, и флажки, позволяющие контролировать использование сертификатов и тем самым настраивать политику безопасности внутри сектора Kerberos. Скажем, с помощью флажка forwardable можно “привязать” сертификат к одному компьютеру и запретить его перемещение на другие хосты. Пример рассмотренный в начале статьи был бы невозможен в этом случае - после получения удалённого доступа мне каждый раз потребовалось бы вводить пароль для запроса нового TGT.

Более детальное описание протокола обмена между клиентскими программами и сервером Kerberos можно найти в статье [1]. Ещё более фундаментальное обсуждение протокола Kerberos можно найти в работах ”основоположников”.

2

В первой части статьи были рассмотрены принципы работы Kerberos и взгляд на него с пользовательской точки зрения. Теперь можно перейти к практической части и начать развёртывание инфраструктуры Kerberos в локальной сети (будем считать, что её доменное имя myrealm.ru).

Начнём с установки сервера Heimdal на компьютере с ОС Linux. Если Heimdal не включён в состав вашего дистрибутива, то вы можете получить исходный код Heimdal с ftp-сервера Стокгольмского университета [2], сконфигурировать и скомпилировать его.

```
tar xzvf heimdal-0.6.3.tar.gz
cd heimdal-0.6.3
./configure --prefix=/usr --enable-shared
make
make install
```

Опции конфигурации позволяют вам собрать heimdal в виде разделяемых библиотек, что в дальнейшем упрощает сборку программ с поддержкой Kerberos, и установить Heimdal в каталог /usr. При этом пользовательские программы (kinit, klist, telnet и т.д.) записываются в каталог /usr/bin, программы для удаленного администрирования контроллера Kerberos - в /usr/sbin, а серверная часть Kerberos - в /usr/libexec.

Если предполагаемое число принципалов невелико (например, порядка сотни), то Heimdal не требует особенно большого количества вычислительных ресурсов. В моем случае в качестве kdc используется 486 компьютер. Желательно, тем не менее, держать базу данных Kerberos на специально выделенном для этой цели компьютере, т.к. захват злоумышленником этого сервера, полностью компрометирует безопасность всей системы.

После того как компьютер выбран, на нем нужно создать каталог для хранения баз данных Kerberos.

```
mkdir /var/heimdal
chmod 655 /var/heimdal
```

После этого нужно проделать две вещи: создать конфигурационный файл kdc и инициализировать (заселить несколькими основными принципалами) базу данных Kerberos. Конфигурационный файл (/etc/krb5.conf) используется как сервером Kerberos, так и приложениями собранными с поддержкой Kerberos. Поэтому этот файл (практически без изменений) можно перенести на все компьютеры, входящие в ваш сектор.

```
[libdefaults]
        default_realm=MYREALM.RU
[realms]
        MYREALM.RU={
                kdc=kdc.myrealm.ru
                admin_server=kdc.myrealm.ru
                kpasswd_server = kdc.myrealm.ru
        }
[logging]
        kdc=FILE:/var/log/krb5kdc.log
        admin_server=FILE:/var/log/kadmin.log
        default=FILE:/var/log/krb5.log
```

В этом файле содержится необходимый минимум настроек Kerberos - название сектора по умолчанию (строка, которая будет добавляться к именам принципалов без доменной части), расположение серверов Kerberos : kdc, сервера удалённого администрирования и их лог-файлов. В качестве альтернативы для передачи настроек Kerberos можно использовать службу DNS, для чего в файл зоны для домена myrealm.ru нужно добавить записи (предполагается, что адрес для kdc уже записан в файле зоны):

```
_kerberos            TXT       "MYREALM.RU"
kerberos             CNAME     kdc
_kerberos._udp       SRV       0 0 88 kdc
_kerberos._tcp       SRV       0 0 88 kdc
_kerberos-adm._tcp   SRV       0 0 749 kdc
_kpasswd._udp        SRV       0 0 464 kdc
```

Передача настроек Kerberos по DNS полезна при большом числе клиентских компьютеров, когда синхронизировать файлы /etc/krb5.conf вручную сложно. Правда, некоторые версии Heimdal по умолчанию не используют записей DNS для получение настроек сектора Kerberos, так что не лишним будет всё же держать на клиентах Kerberos файл /etc/krb5.conf такого вот вида:

```
[libdefaults]
        dns_lookup_kdc = yes
         dns_lookup_realm=yes
        default_realm=MYREALM.RU
```

Теперь можно приступать к пополнению баз данных Kerberos:

```
/usr/sbin/kadmin -l
```

```
kadmin>init MYREALM.RU
kadmin>add admin/admin@MYREALM.RU
kadmin>add mike@MYREALM.RU
kadmin>add --random-key host/kdc.myrealm.ru@MYREALM.RU
kadmin>ext */kdc.myrealm.ru@MYREALM.RU
```

Ключ -l позволяет запускать kadmin в локальном режиме, без аутентификации в Kerberos. Команда init создаёт базу данных Kerberos, и заселяет её несколькими принципалами со случайным образом сгенерированными ключами для сервисов предоставляемых самим Kerberos. Это - krbtgt/MYREALM.RU для Ticket Granting Service, kadmin/admin для сервиса kadmind и kadmin/changepw для сервиса kpasswd, позволяющего пользователям менять свой пароли в Kerberos. Принципал kadmin/hprop, также создаваемый при инициализации Kerberos, используется при синхронизации баз данных между дополнительными и основным контроллерами сектора Kerberos, и в нашем случае он не актуален. С помощью команда add в эту вновь созданную базу записываются пользователи и сервисы. В данном случае добавляются пользователи - администратор Kerberos (admin/admin), ещё один рядовой пользователь (mike) и принципал для компьютера, на котором работает контроллер Kerberos (host/kdc.myrealm.ru). В отличии от первых двух случаев, для сервиса пароль выбирается случайным образом (ключ --random-key), поскольку знать его ни администратору Kerberos, ни тем более рядовым пользователям необязательно. Затем этот ключ извлекается из базы данных Kerberos (команда ext) и дописывается в файл /etc/krb5.keytab (что является значением по умолчанию). Керберизованные серверные программы знают, что их "пароли"записаны в этом файле и при подключении клиента (см. схему работы Kerberos изложенную выше) могут извлечь свой пароль оттуда.

Теперь все готово для запуска сервера kdc. Если после команды:

```
/usr/libexec/kdc -c /etc/krb5.conf --detach
```

в log-файле появились сообщения вида:

```
2005-03-15T23:43:46 listening on IPv4:127.0.0.1 port 88/udp
2005-03-15T23:43:46 listening on IPv4:127.0.0.1 port 88/tcp
```

значит сервер успешно стартовал и можно переходить к запуску сервера удалённого администрирования kadmind (749/tcp порт) и службы смены паролей kpasswdd (464/udp порт).

```
/usr/libexec/kadmind
/usr/libexec/kpasswdd
```

Хотя для смены паролей можно использовать программу kadmin (/usr/sbin/kadmin passwd mike), но для пользователей более удобно проделать тоже самое с помощью утилиты kpasswd, которая подключается к службе kpasswdd.

Поскольку сервис kadmind нужен не очень часто (при добавлении/удалении принципалов), то имеет смысл использовать его вызов в сервере inetd (xinetd). Для этого вам, во-первых, нужно убедиться, что ссылки на керберизованные сервисы (и kadmind в том числе) присутствуют в файле /etc/services. Необходимые для этого данные содержатся в файле heimdal-0.6.3/etc/services.append в дистрибутиве heimdal. Во-вторых, нужно добавить запись, касающуюся сервера удаленного администрирования в файл /etc/inetd.conf и перезапустить inetd.

```
kerberos-adm stream tcp nowait root /usr/libexec/kadmind kadmind
```

В тоже время kpasswdd может работать только как самостоятельная служба и организовать его вызов с помощью inetd невозможно.

Контроль доступа к kadmind и управлению пользователями Kerberos обеспечивается списками хранящимся в файле /var/heimdal/kadmin.acl. Разумная политика - предоставить администратору (admin/admin) полный контроль по управлению записями Kerberos, а всем остальным - позволить лишь менять свои пароли. Для такой цели достаточно иметь такой acl-файл:

```
admin/admin@MYREALM.RU     all
*@MYREALM.RU                cpw
```

Для проверки работоспособности сервера Kerberos попробуйте аутентифицироваться на нём (как администратор) и посмотреть содержимое его баз данных:

```
kinit -p admin/admin
kadmin
kadmin> list *
```

На этом установку сервера Kerberos можно считать законченной. Настройка же клиентов для использования этого сервера различается в случае рабочей станции и серверов приложений. Разберём два чистых случая - настройка только сервера (компьютера предлагающего сетевые сервисы) и только рабочей станции (которая не предлагает никаких сервисов и предназначена только для локальной работы).

Настройка серверов приложений достаточно проста (при условии, что на сервере работает юниксоподобная система). Во-первых, устанавливаете дистрибутив heimdal, как описано выше. Этого уже достаточно, чтобы компьютер (предположим, что его сетевое имя server.myrealm.ru) работал как клиент Kerberos, нужно только скопировать файл krb5.conf с сервера Kerberos на настраиваемый компьютер. Конфигурационный файл можно не редактировать - приведённый выше файл

годится как для серверов, так и рабочих станций. Кроме того, как уже говорилось, на хосте с сетевыми сервисами должен присутствовать файл /etc/krb5.keytab с набором ключей для этих сетевых служб (точнее, для принципалов Kerberos, соответствующим этим сетевым службам). Также сервису должно быть известно имя принципала, который представляет его в Kerberos. Как правило, этот параметр настраивается для каждого сервиса индивидуально (с помощью конфигурационных файлов) или используются какие-то фиксированные значения. В любом случае вам придётся проконсультироваться в документации, предлагаемой разработчиками программы. Как уже говорилось ранее, для сервисов используются трёхчленные имена - поле instance содержит сетевое имя компьютера, а основное имя обозначает тип предлагаемого сервиса. Предположим, вы хотите установить все сервисы входящие в дистрибутив heimdal (telnet, ftp, несколько r* служб). Для этого регистрируетесь в Kerberos как администратор, добавляете принципалов `host/server.myrealm.ru@MYREALM.RU`, `ftp/server.myrealm.ru@MYREALM.RU` и извлекаете их ключи в локальный /etc/krb5.keytab файл.

```
kinit admin/admin
kadmin
kadmin>add --random-key  host/server.myrealm.ru@MYREALM.RU
kadmin>add --random-key  ftp/server.myrealm.ru@MYREALM.RU
kadmin>ext */server.myrealm.ru@MYREALM.RU
```

Если вы ошиблись - добавили в /etc/krb5.keytab ключ службы, которая не используется на данном компьютере, то для управления ключами в файле /etc/krb5.keytab предназначена утилита ktutil. Удалить лишнюю запись можно командой /usr/sbin/ktutil remove -p ftp/server.myrealm.ru, точно так же как и просмотреть все записи в keytab-файле (ktutil list) или добавить новый ключ (ktutil get ftp/server.myrealm.ru)

Теперь настроим запуск сервисов через демон inetd, для чего нужно добавить следующие строки в файл inetd.conf (не забудьте также добавить записи из файла heimdal-0.6.3/etc/services.append к системному файлу /etc/services):

```
telnet  stream  tcp nowait  root /usr/libexec/telnetd telnetd -L /usr/bin
ftp     stream  tcp nowait  root /usr/libexec/ftpd ftpd
shell   stream  tcp nowait  root /usr/libexec/rshd rshd -v
kshell  stream  tcp nowait  root /usr/libexec/rshd rshd -k
ekshell stream  tcp nowait  root /usr/libexec/rshd rshd -kx
kx      stream  tcp nowait  root /usr/libexec/kxd kxd
```

Почти все сервисы стандартные, shell/kshell/ekshell - это вариации на тему rshd. Первый из них работает просто как ”заглушка выдаёт сообщение об ошибке, если

пользователь подключается по-старинке, без TGT; второй и третий - керберизованные сервисы, в последнем случае с шифрованием траффика. Нестандартным является сервис kx (использует 2111/tcp порт), который предназначен для форвардинга X соединений через шифрованный канал. Если вы, например, подключитесь к серверу server.myrealm.ru с помощью команды rxtelnet server.myrealm.ru, то при успешном соединении на вашем X-сервере откроется окно xterm с командным интерпретатором, запущенном на удаленном компьютере (server.myrealm.ru). Вся графика при этом будет прозрачно переадресовываться на вашу рабочую станцию. В некотором смысле мы имеем аналог ssh -X server.myrealm.ru.

При добавлении ключей в keytab файл есть один тонкий момент, на котором стоит остановиться. Это вопрос прав доступа к keytab файлу. Поскольку он содержит хэши паролей принципалов Kerberos, то имеет смысл максимально ограничить доступ к этому файлу, например, так же как к /etc/shadow - доступ на чтение и запись только администратору. Однако, в случае с keytab файлом этого делать не следует. Пока что рассмотренные нами службы запускаются с правами root и проблем с доступом к таким образом защищённому /etc/krb5.keytab у них не будет. Но как правило сетевые службы стараются запускать от непривилегированного пользователя (что вызывает меньше проблем с безопасностью), и тем не менее эта служба должна иметь возможность читать свой собственный ключ из keytab файла. Для этого можно, во-первых, создать индивидуальный keytab для этой службы, при этом не забывая сообщить службе, что ее ключ находится в нестандартном файле (у разных сервисов имена опции для этой цели различны). С другой стороны, можно ослабить права доступа и заменить владельца /etc/krb5.keytab, разрешив его чтение специальной группе (пусть она будет называться kerberos), в которую нужно будет включить root и всех пользователей, от имени которых запущены интересующие нас сетевые службы.

Поскольку время действия любого сертификата Kerberos ограничено, то одной из задач при настройке сектора Kerberos является синхронизация времени между компьютерами в локальной сети. Локальное время компьютера используется клиентами Kerberos при запросе на выдачу билета, и если это время значительно (более чем на 5 минут) отличается от времени контроллера Kerberos, то такие запросы отвергаются. Так что бесполезно, например, предлагать контроллеру Kerberos просроченный TGT, просто переведя часы на локальном компьютере. Но с другой стороны, удалённый доступ к какому-то компьютеру также может быть невозможным, только потому, что его часы опережают часы контроллера Kerberos, скажем, на 10 минут. Поэтому синхронизация времени внутри сектора необходима для корректного функционирования Kerberos. Если у вас в локальной сети уже работает сервер ntp, то можно воспользоваться им. Об установке и настройке сервера ntpd написано в статье [3]. Однако для целей Kerberos достаточно, если вы выберете

какой-то из компьютеров в вашей сети (допустим тот же самый server.myrealm.ru) в качестве эталонного и будете сверять по нему часы остальных компьютеров (и контроллера Kerberos в том числе) по протоколу time. Служба time уже встроена в сервер inetd, нужно только убедиться, что в файле `/etc/inetd.conf` не закомментирована строка:

```
time    dgram   udp     wait    root    internal
```

Синхронизация осуществляется с помощью базовой Unix-утилиты netdate server.myrealm.r
Для рабочих станций её достаточно запускать при загрузке, для серверов, которые перегружаются не так часто, имеет смысл вызывать ее периодически с помощью демона cron.

Перейдём теперь к настройке рабочих станций. Для этого достаточно иметь на них упоминавшийся выше конфигурационный файл `/etc/krb5.conf` и собственно сам дистрибутив heimdal. Но бывает также удобно настроить системный login на этой рабочей станции таким образом, чтобы при входе на компьютер, пользователь автоматически получал TGT с контроллера Kerberos. Для рабочих станций под управлением Linux, такой функциональности можно добиться заменив системный login (/bin/login) на керберизованный аналог из состава Heimdal (/usr/bin/login). Проще всего это сделать отредактировав файл /etc/inittab:

```
c1:123:respawn:/sbin/agetty 38400 tty1 linux
c2:123:respawn:/sbin/agetty 38400 tty2 linux
k1:5:respawn:/sbin/agetty -l /usr/bin/login 38400 tty1 linux
k2:5:respawn:/sbin/agetty -l /usr/bin/login 38400 tty2 linux
```

и заменив default run-level на 5. Таким образом по умолчанию загрузка компьютера будет осуществляться в керберизованном режиме, а для аварийных ситуаций у администратора компьютера сохраняется возможность заходить на компьютер в стандартном режиме (в run-level 1,2,3). Следует только иметь ввиду, что керберизованный login запрашивает, помимо собственно первоначального сертификата, ещё и session ticket для доступа к рабочей станции. Как вы понимаете, это требует наличия принципала соответствующего данной рабочей станции в базе данных Kerberos и её ключа в файле /etc/krb5.keytab. Заметим также, что если по каким-то причинам аутентификация пользователя на сервере Kerberos закончилась неудачей, то login переключается в стандартный режим и аутентифицирует пользователя по локальному файлу /etc/shadow.

Для рабочих станций под управлением Windows2000/XP получить такую же функциональность несколько сложнее. Во-первых, авторы Heimdal не предлагают свой продукт в виде стандартного Windows приложения и Heimdal может работать только в Cygwin-окружении (www.cygwin.com) - эмуляторе POSIX системы для

Windows. Вы можете установить Cygwin целиком с помощью установщика, который можно скачать с сайта cygwin, и затем скомпилировать исходный код Heimdal. Гораздо проще, однако, взять уже готовые бинарные пакеты Heimdal и урезанную версию Cygwin с ftp-сервера Стокгольмского университета [4] . Поскольку Cygwin для приложений, запущенных в нем, выглядит как полное Unix окружение, то дальнейшая конфигурация ничем не отличается от установки под Linux. После установки Cygwin и Heimdal, пользователь по-крайней мере может аутентифицироваться на сервере Kerberos, запустив терминал Cygwin и введя команду kinit. Полученный сертификат в дальнейшем может быть использован клиентскими программами из дистрибутива Heimdal (ftp, telnet и т.д.). При этом остаются две проблемы - заставить Heimdal использовать ваше имя и пароль введённый при входе в Windows для аутентификации на сервере Kerberos и обеспечить доступ к Kerberos не только для Cygwin, но и Windows приложений. К сожалению, только средствами heimdal эти проблемы не решаются.

Разберёмся со второй проблемой. Хотя библиотеки Cygwin и библиотеки Heimdal в том числе являются стандартными динамически линкуемыми библиотеками Windows, но использование их для приложений Windows невозможно (подробнее об этом написано на сайте cygwin - [5]). Поэтому для Windows приложений есть две возможности - или использовать API предлагаемой самой Windows или использовать библиотеки MIT-Kerberos, которые доступны в Windows-версии, но при этом бесполезны для Cygwin приложений. Возникает вопрос - какие из библиотек лучше установить на свой компьютер? Правильный ответ - ставить все! Пакет Support Tools от Microsoft [6] включает в себя две утилиты ksetup и kpasswd, с помощью которых можно настроить получение начальных сертификатов Kerberos при входе пользователя на рабочую станцию, а пакет Kerberos-For-Windows от МТИ [7] позволяет обеспечить доступ к этим сертификатам как для библиотек MIT-Kerberos, так и Heimdal.

Так что, начнём с настройки керберизованного входа в Windows. Я использую чистую WindowsXP-Pro без service-pack'ов, но тот же метод работает и для более новых версий Windows. Установка Support Tools не должна вызвать затруднений. Затем вам нужно добавить принципала для этой машины (для определённости, назовём её xp.myrealm.ru) в базу данных Kerberos. Сделать это можно, например, из терминала Cygwin (я полагаю, что вы уже настроили Heimdal в нем), но в отличии от Линукс рабочих станций, вам нужно будет придумать пароль для этого компьютера:

```
kinit admin/admin
kadmin
kadmin>add -pw  passwordforWinXP host/xp.myrealm.ru@MYREALM.RU
```

Затем с помощью утилиты ksetup.exe, зайдя на машину под администратор-

ским логином, из Командной строки Windows вы должны указать название сектора Kerberos, адрес контроллера Kerberos и ввести пароль для машины (тот самый, что вы только что придумали)

```
C:> ksetup /setdomain MYREALM.RU
C:> ksetup /addkdc MYREALM.RU kdc.myrealm.ru
C:> ksetup /setcomputerpassword passwordforWinXP
```

После чего перегрузить компьютер.

При загрузке машины обратите внимание, что форма с приглашением входа в Windows изменилась. На ней должно появиться ещё одно поле выбора с двумя вариантами - зарегистрироваться на компьютере xp.myrealm.ru или в секторе Kerberos MYREALM.RU. Но второй вариант пока не работоспособен, т.к. вы ещё не настроили соответствия между локальными именами пользователей и именами принципалов Kerberos. Так что вам нужно снова войти на компьютер как Администратор и настроить эти соответствия. Тут есть два варианта:

```
C:> ksetup /mapuser * *
C:> ksetup /mapuser mike@MYREALM.RU mikeXP
C:> ksetup /mapuser * DefaultUser
```

В первом случае каждый принципал из базы данных Kerberos отображается на пользователя Windows с тем-же именем (только без доменной части). Во втором случае (если вас такая политика не устраивает), вы разрешаете только определённому пользователю (mikeXP) аутентифицироваться в Kerberos под именем принципала mike@MYREALM.RU. Последний случай, может быть интересен тем, что нет необходимости заводить учётные записи на компьютере (DefaultUser присутствует в WinXP по умолчанию). Правда, при такой настройке компьютер превращается практически в однопользовательскую систему, поскольку любой пользователь после ввода своего имени и пароля, под которыми он зарегистрирован в Kerberos, получает доступ к одному и тому же пользовательскому профилю и файлам DefaultUser. В какой-то мере индивидуальную настройку можно получить, если хранить файлы и профиль на сетевом ресурсе (например, каталоге Samba/CIFS, сконфигурированных с поддержкой Kerberos), но тут я ничего не могу посоветовать.

Так или иначе, но теперь вы можете при входе на рабочую станцию одновременно получать сертификаты Kerberos. Но этот сертификат храниться в памяти, а не на дисковом файле, как у Heimdal, и доступ к нему осуществляется посредством API Windows. Чтобы сделать его доступным для Heimdal нужно воспользоваться утилитой ms2mit из пакета MIT-Kerberos, разработчики которого работают в тесном контакте с Microsoft и поэтому представляют себе особенности её реализации Kerberos.

Установка MIT-Kerberos очень проста. Помимо инсталлятора [7] они предлагают свой пакет в виде обычного ZIP-архива [8], который можно распаковать в удобный для вас каталог. Не стоит только устанавливать его в директорию Program Files, т.к. она входит в переменную PATH Cygwin, что может привести к конфликтам между одноимёнными утилитами Heimdal и MIT-Kerberos. Так что лучше распаковать его в какой-нибудь из каталогов на диске C: (у меня это `C:\kfw-2.6.5`). Помимо утилиты ms2mit в него входят уже знакомый вам набор стандартных утилит Kerberos (kinit, klist, kdestroy) и также графический интерфейс к ним (Leash32). После установки MIT-Kerberos вам нужно настроить Heimdal и MIT-Kerberos таким образом, чтобы они использовали один и тот-же кэш сертификатов, с тем чтобы начальный сертификат, полученный с помощью одной из этих библиотек, был бы автоматически доступен и другой. У Heimdal кэш - это (как вы помните) файл с именем `krb5cc_ + id` пользователя во временном каталоге (временный каталог Cygwin - это `C:\Cygwin\tmp`), но это название можно поменять с помощью переменной окружения Cygwin `KRB5CCNAME`. У MIT-Kerberos в дефолтной установке используется кэш в памяти, но его можно настроить на использование дискового файла. Сделать это можно с помощью установки переменной окружения Windows (KRB5CCNAME) и записей в реестре Windows `HKCU\Software\MIT\Kerberos5\ccname`, `HKLM\Software\MI`
Системой используется первое не пустое значение в этой последовательности или значение по-умолчанию - "API:". Я использую bat-файл запускаемый при входе в Windows, который устанавливает переменную окружения `KRB5CCNAME` на название файла используемого Heimdal и затем перемещает сертификат Windows в этот файл утилитой `ms2mit`. Для каждого из пользователей Windows этот файл свой, поскольку значение `KRB5CCNAME` зависит от идентификатора этого пользователя в Cygwin окружении.

```
set KRB5CCNAME = FILE:C:\Cygwin\tmp\krb5cc_1017
ms2mit
```

Тоже самое вы можете проделать с помощью программы Leash32 (меню Action -> Import Ticket(s)/Token(s)), также как просматривать/удалять имеющиеся сертификаты и настраивать кэш сертификатов (меню Options -> Kerberos v5 Properties Dialog).

Здесь имеется еще одна ловушка, заботливо расставленная компанией Microsoft. Для большей безопасности в установке по-умолчанию Support Tools экспорт начального сертификата из памяти Windows запрещён, что делает его использование бесполезным для MIT-Kerberos и Heimdal. К счастью (точнее благодаря давлению со стороны разработчиков из MIT), Microsoft оставила ключи в реестре, с помощью которых можно снять этот запрет. Так что, для того чтобы утилита `ms2mit` работала, в реестре должны быть установлены значения:

```
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters
   AllowTGTSessionKey = 0x01 (DWORD)
HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos
   AllowTGTSessionKey = 0x01 (DWORD)
```

В заключении еще несколько слов о совместимости различных версий Kerberos. Как вы могли убедиться выше, билетики получаемые от сервера Heimdal работают во всех версиях клиентов Kerberos (Microsoft, MIT, Heimdal). То же самое верно, если бы в качестве сервера использовался бы KDC от MIT. Пример же использования библиотеки MIT-Kerberos в домене Windows2000 с Active Directory, где реализован сервер Kerberos от Microsoft, разобран в статье [1]. Так что, на уровне протокола совместимость между реализациями удовлетворительная. В то же время на уровне API совместимость даже между Heimdal и MIT Kerberos оставляет желать лучшего.

Для того, чтобы приложение (клиент или сервер) могло использовать Kerberos, соответствующая поддержка должна быть заложена в код программы. Реализации протокола Kerberos с нуля сложна, поэтому как правило приложение использует динамические библиотеки Kerberos. Реализации от MIT пользуется большей популярностью, поэтому скорее всего интересующее вас приложение будет использовать именно ее библиотеки. При этом поддержка Heimdal тоже может быть реализована авторами программы. Хотя заголовки функций Heimdal практически повторяют функции MIT-Kerberos, но структуры данных существенно различны. Поэтому перелинковка приложения разработанного под MIT Kerberos с библиотеками Heimdal, даже если у вас получится это проделать, скорее всего приведёт к некорректной работе приложения. Таким образом модификация кода приложения, пусть и небольшая, неизбежна в любом случае.

Это замечание особенно актуально для Windows. Многие керберизованные приложения, разработанные под Unix-системы, можно использовать в Windows в среде Cygwin, поскольку для авторов портирование под Cygwin даётся легче, чем перелицовывание программы под полноценное Windows приложение. Единственная реализация Kerberos, доступная для такого рода программ - это Heimdal и вам следует убедиться, что интересующее вас приложение поддерживает Heimdal. Удивительно, но хотя MIT-Kerberos и разрабатывается под Unix-системы, но в Cygwin окружении он неработоспособен.

Исключением является ситуация, когда приложение использует Kerberos не напрямую, а через оболочку GSSAPI. Примером такого приложения является ftp из дистрибутива Heimdal. GSSAPI (Generic Security Service Application Program Interface) - это интерфейс обеспечивающий стандартизованный доступ к функциям аутентификации Kerberos. Как MIT-Kerberos, так и Heimdal предлагают свои реализации GSSAPI, поэтому в данном случае обе библиотеки оказываются полно-

стью взаимозаменяемыми.

На этом установку Kerberos в гетерогенной сети можно считать завершённой. В итоге вы можете управлять регистрационной информацией пользователей из одного места (сервера kdc). Ваши пользователи получают регистрацию в секторе Kerberos при входе на свои рабочие станции, прозрачный доступ к компьютерам внутри сети с помощью команд telnet и rsh, удалённый доступ к файлам при помощи ftp и rcp, шифрованный канал доступа к X-серверам. Разумеется, количество керберизованных сервисов не ограничивается только службами из состава Heimdal, но установка и конфигурирование других сервисов требует отдельного обсуждения.

3

Будет полезно обсудить ещё пару вопросов связанных с ограничением доступа пользователей к сетевым ресурсам средствами Kerberos и управления учётными записями пользователей администратором Kerberos.

Ограничение доступа можно организовать на уровне отдельных групп пользователей. Более тонкий контроль доступа, на уровне отдельного пользователя, вряд ли осуществим средствами Kerberos и в этом случае проще использовать встроенные ACL интересующего вас сервиса.

Типичная ситуация - организация состоящая из нескольких отделов. Каждый из отделов располагает своими сетевыми ресурсами, которые доступны сотрудникам этого отдела и невидимы для остальных. Кроме того имеются в распоряжении организации есть сервисы, которые должны быть доступны всем сотрудникам организации. Такая задача легко решается с помощью нескольких подсекторов Kerberos и однонаправленных трастов между ними. Правда, задача становится неразрешимой, если вы попытаетесь немного усложнить её, скажем, захотите предоставить сотруднику из одного отдела доступ к ресурсам другого отдела [1]

Как на практике выглядит это решение. Схема подсекторов представлена на рисунке 2. Вам нужно обеспечить доступ пользователя `alice@A.MYREALM.RU` к компьютерам `host/venus.myrealm.ru@A.MYREALM.RU` и `ftp/server.myrealm.ru@MYREALM` при том чтобы у уже созданного пользователя `mike@MYREALM.RU` доступа к `host/venus.my` не было. Сектор `MYREALM.RU` в данном случае соответствует всей организации, а подсектор `A.MYREALM.RU` - одному из отделов. Добавить ещё один подсектор (отдел) не составляет особого труда, причём, если следовать описанному ниже рецепту, то службы из двух подсекторов будут доступны только принципалам из того же самого подсектора.

Для реализации этой программы, во-первых, нам нужно настроить контроллер

[1] Разве что выделить для этого сотрудника отдельный подсектор

Kerberos на поддержку двух секторов, а, во-вторых, с помощью конфигурационных файлов /etc/krb5.conf сообщить двум используемым хостам, кто из них к какому сектору принадлежит. Можно было бы настроить доменную систему имён, так чтобы каждый из подсекторов Kerberos совпадал с подзоной DNS, тогда можно было бы использовать опции dns_lookup_kdc = yes,dns_lookup_realm=yes для настройки клиентов. Тем не менее пока у нас оба подсектора Kerberos будут находиться в одной зоне DNS.

Предполагается, что сектор MYREALM.RU уже настроен как было описано ранее. Уже имеющийся файл следует подредактировать следующим образом:

```
[libdefaults]
        default_realm=MYREALM.RU
[realms]
        MYREALM.RU={
                kdc=kdc.myrealm.ru
                admin_server=kdc.myrealm.ru
                kpasswd_server = kdc.myrealm.ru
        }
        A.MYREALM.RU={
                kdc=kdc.myrealm.ru
                admin_server=kdc.myrealm.ru:1749
                kpasswd_server = kdc.myrealm.ru:1464
        }
[logging]
        kdc=FILE:/var/log/krb5kdc.log
        admin_server=FILE:/var/log/kadmin.log
        default=FILE:/var/log/krb5.log
```

Так выглядит файл на контроллере Kerberos и на компьютере host/earth.myrealm.ru. На компьютере host/venus.myrealm.ru нужно только поменять default_realm=A.MYRE. После чего перезапустить kdc, чтобы заставить его перечитать конфигурационный файл, и проинициализировать сектор A.MYREALM.RU с помощью команды:

```
/usr/sbin/kadmin -l
kadmin>init A.MYREALM.RU
```

Это не должно приводить к катастрофическим последствиям, т.е. уже имеющиеся в базе данные для сектора MYREALM.RU пострадать при этом не должны [2]

Прежде чем заполнять вновь созданный сектор принципалами, разберёмся со службами kadmind и kpasswd. В отличии от сервера kdc каждая из этих служб

[2] Но бэкап, о котором речь впереди, сделать всё же не помешает!

может обслуживать только один сектор Kerberos, и в том случае если этот сектор отличается от сектора по умолчанию, он должен быть указан при запуске в виде опции, также как и нестандартный порт (мы не можем привязать к одному порту два сервера). Т.е. скрипт запуска этих служб должен выглядеть примерно так:

```
#Для сектора MYREALM.RU (сектор по умолчанию)
/usr/libexec/kadmind
/usr/libexec/kpasswdd
#Для сектора A.MYREALM.RU
/usr/libexec/kadmind --realm=A.MYREALM.RU --port=1749
/usr/libexec/kpasswdd --realm=A.MYREALM.RU --port=1764
```

Acl файл /var/heimdal/kadmin.acl используется всеми этими сервисами, и поэтому нуждается в небольшой доработке:

```
admin/admin@MYREALM.RU     all
*@MYREALM.RU               cpw
admin/admin@A.MYREALM.RU   all
*@A.MYREALM.RU             cpw
```

В принципе можно пока обойтись без этих сервисов, используя локальную версию `kadmin -l`. Специальной аутентификации она не требует, так что её можно использовать для добавления принципалов в оба подсектора. Стоит только помнить, что необходимо указывать имя принципалов полностью, вместе с доменной частью. Таким образом нам предстоит добавить принципалов администратора подсектора `admin/admin@A.MYREALM.RU`, пользователя `alice@A.MYREALM.RU` и компьютера `host/venus.myrealm.ru@A.MYREALM.RU`. Для последнего требуется ещё извлечь ключ в keytab-файл, например, с помощью команды `kadmin -l ext_keytab -k venus.ke` а затем каким-то образом перенести созданный файл venus.keytab по назначению. Кроме того важно создать в каждом из секторов одного и того же принципала, который позволяет пользователям из одного подсектора использовать ресурсы из другого, так называемый однонаправленный траст[3], в нашем случае `krbtgt/MYREALM.RU@A.MYRE` Ключ/пароль для этого принципала должен быть один и тот же в каждом из секторов.

Последнее усилие, для того чтобы траст заработал, это указать какой из сетевых ресурсов к какому сектору принадлежит. В принципе удобнее было бы это сделать с помощью DNS, но поскольку в нашем распоряжении всего два ресурса, то можно просто добавить записи с именами используемых ресурсов в файлы `/etc/krb5.conf`:

[3]Для двунаправленного траста нужно было бы создать ещё одну пару принципалов krbtgt/A.MYREALM.RU@MYREALM.RU

```
[domain_realm]
venus.myrealm.ru=A.MYREALM.RU
earth.myrealm.ru=MYREALM.RU
```

Таким образом можно осуществлять грубый контроль доступа пользователей к ресурсам.

Последняя тема, которую я хотел бы затронуть - это управление базой данных пользователей, для чего используются команды `merge,load,dump` из локальной версии `kadmin -l`. Само собой команда `dump filename` снимает дамп базы данных в указанный файл, а команды `merge filename` и `load filename` соответственно добавляют и полностью заменяют базу данных, используя указанный файл. Однако, эти команды можно использовать, например, для переноса пользователей из Active Directory в Heimdal. Для этого потребуется ещё утилита pwdump2 [9] [4], которая выгружает хэши паролей из ActiveDirectory (из командной строки на сервере, где запуоен сервер ActiveDirectory, командой `pwdump2.exe > ADHashDump`), и awk-скрипт `pwd.awk`, который преобразует этот дамп к виду пригодному для загрузки в Heimdal.

```
BEGIN {
  realm="MYREALM.RU"                # modify this
  time=strftime ("%Y%m%d%k%M%S"); FS=":" }
{if ($1 !~ /\$$/)                 # avoid machine accounts
  printf "%s@%s 1::23:%s:- %s:kadmin/admin@%s - - - - - - 126 -\n",
    $1, realm, $4, time, realm }
```

Нужно будет отредактировать вторую строчку, вставив туда название вашего сектора. Запускается скрипт командой:

```
awk -f pwd.awk ADHashDump > HeimdalDump
```

В принципе на этом все мои трюки по управлению и администрированию сервера Heimdal исчерпаны. В заключение небольшая таблица 1 со службами, которые поддаются интегрированию в single-sign-on систему средствами Kerberos.

Список литературы

[1] Р.Гребенников. Танцуем Самба. *Системный Администратор*, 11, 2004.

[4] Многие антивирусные программы эту утилиту не любят (вполне допускаю, что у них на это есть основания) и реагируют на неё как на вирус. Вместе с тем исходный код программы доступен, если есть сомнения в безопасности этой программы, то лучше собрать её из исходников, предварительно их изучив.

Сервис	Сервер	Клиент	принципал
Удаленный доступ	OpenSSH	ssh	host
FTP	ProFTPd + gssapi patch	?	ftp, host
HTTP	Apache + mod_krb_auth	Firefox	http
SMTP	Postfix + SASL	Thunderbird, Evolution, Kmail	smtp
IMAP,POP4	Cyrus-Imap + SASL	Thunderbird, Evolution, Kmail	imap
Sieve (скрипт фильтрации почты)	Cyrus-IMAP+SASL	sieve shell	sieve, imap
Instant Messenger	Jabberd2 + SASL	Tkabber, Gaim	xmpp
LDAP	OpenLDAPv3 + SASL	ldapsearch	ldap
Сетевая файловая система NFS	Линукс+NFS4	?	nfs
Сетевая файловая система Samba	Smaba3.x, Windows CIFS	?	cifs
Распределённая файловая система Coda	Coda		coda
Распределённая файловая система AFS	OpenAFS		?
Системы контроля версий	Subversion		

Таблица 1: Керберизованные сервисы.

[2] Heimdal - исходный код.
ftp://ftp.pdc.kth.se/pub/heimdal/src/.

[3] М.Платов. Атомные часы на вашем столе. *Системный Администратор*, 2004.

[4] Heimdal и Cygwin - бинарная версия.
ftp://ftp.pdc.kth.se/pub/heimdal/binaries/i386-pc-cygwin/latest.

[5] Cygwin - FAQ.
http://cygwin.com/faq.html.

[6] Microsoft Support Tools.
http://www.microsoft.com/downloads/details.aspx?familyid=
49AE8576-9BB9-4126-9761-BA8011FABF38\&displaylang=en.

[7] Kerberos for Windows (самораспаковывающийся архив).
http://web.mit.edu/kerberos/www/dist/kfw/2.6/kfw-2.6.5/
MITKerberosForWindows-2.6.5.exe.

[8] Kerberos for Windows (ZIP - архив).
http://web.mit.edu/kerberos/www/dist/kfw/2.6/kfw-2.6.5/
kfw-2-6-5.zip.

[9] Pwdump-download.
http://www.doubleupsoftware.com/HowToGetPwdump2.asp?AfId=
&affiliateid=.