Содержание

Введение

В открытой сетевой вычислительной среде рабочая станция не может быть доверенной в части корректной идентификации своих пользователей сетевым службам. Протокол Kerberos реализует альтернативный подход с использованием доверенной сторонней службы аутентификации для подтверждения идентификационных данных пользователей.

Эта статья содержит обзор модели аутентификации Kerberos, реализованной в проекте Athena Массачусетского технологического института. Описываются протоколы, используемые клиентами, серверами, и службой Kerberos при выполнении аутентификации. Также описываются необходимые задачи управления и репликация базы данных. Рассматривается представление Kerberos с точки зрения пользователя, программиста и администратора. Наконец, описывается роль Kerberos в общем контексте проекта Athena наряду со списком приложений, которые теперь используют Kerberos для аутентификации пользователей. Мы приводим добавление аутентификации Kerberos в сетевую файловую систему SUN как пример практического применения для интеграции Kerberos с существующим приложением.

Создатели Kerberos

- Jennifer G. Steiner, Проект Athena, Массачусетский технологический институт, Кембридж, MA 02139, steiner@ATHENA.MIT.EDU
- Clifford Neuman, Department of Computer Science, FR-35, университет Вашингтона, Сиэтла, WA 98195, bcn@CS.WASHINGTON.EDU. Clifford Neuman был участником Участников проекта Athena во время дизайна и фазы начального внедрения Kerberos.
- Jeffrey I. Schiller, Проект Athena, Массачусетский технологический институт, Кембридж, MA 02139, jis@ATHENA.MIT.EDU

Введение в Kerberos

Эта бумага дает обзор Kerberos, система аутентификации, разработанная Миллером и Неуменом. для сред открытых сетевых вычислений, и описывает наш опыт с помощью него в Проекте Athena MIT. В разделе по Мотивации мы объясняем, почему новая модель проверки подлинности необходима для открытых сетей, и каковы ее требования. Что такое Kerberos? разделите перечисляет компоненты Программного обеспечения Kerberos и описывает, как они взаимодействуют в обеспечении сервиса проверки подлинности. В разделе Названий Kerberos мы описываем схему наименования Kerberos.

То, как Kerberos Работает, представляет составляющий компоненты проверки подлинности Kerberos - билет и средство проверки подлинности. Это приводит к обсуждению этих двух протоколов аутентификации: начальная аутентификация пользователя к Kerberos (аналогичный регистрации), и протокол для обоюдной проверки подлинности потенциального клиента и потенциального изготовителя сетевого сервиса.

Kerberos требует базы данных, содержащая информацию о своих клиентах; раздел базы данных Kerberos описывает базу данных, ее управление и протокол для ее модификации. Kerberos От Внешнего Взгляда В разделе описывает интерфейс Kerberos своим пользователям, программистам приложений и администраторам. В разделе Общих сведений мы описываем, как Проекты Athena Kerbero вписываются в остаток Среды Athena. Мы также описываем взаимодействие других доменов аутентификации Kerberos или области; в нашем случае, отношении между Проектами Athena Kerbero и Kerberos, работающим в Лаборатории MIT для Компьютерных наук.

В разделе Неполадок и открытых проблем мы упоминаем нерешенные вопросы и проблемы, пока еще нерешенные. Последний раздел дает текущий статус Kerberos в Проекте Athena. В Приложении мы описываем подробно, как Kerberos применен к сетевой файловой службе для аутентификации пользователей, которые хотят получить доступ к

удаленным файловым системам.

Общие сведения о Kerberos

Всюду по этой бумаге мы используем термины, которые могут быть неоднозначны, в новинку для читателя или используемые по-другому в другом месте. Ниже мы сообщаем наше использование тех сроков.

Пользователь, Клиент, Сервер? Пользователем мы имеем в виду человека, который использует программу или сервис. Клиент также использует что-то, но является не обязательно человеком; это может быть программа. Часто сетевые приложения состоят из двух частей; одна программа, которая работает на одной машине и запрашивает удаленный сервис и другую программу, которая работает на удаленной машине и выполняет тот сервис. Мы называем тех клиентской стороной и стороной сервера приложения, соответственно. Часто, клиент свяжется с сервером от имени пользователя.

Каждый объект, который использует систему Kerberos, быть им пользователь или сервер сети, находится в одном смысле клиент, так как это использует Сервис Kerberos. Таким образом для различения клиентов Kerberos от клиентов других сервисов мы используем термин принципал для указания на такой объект. Обратите внимание на то, что главная часть Kerberos может быть или пользователем или сервером. (Мы описываем именование главных частей Kerberos в последующем разделе.)

Сервис по сравнению с Сервер? Мы используем сервис в качестве общего определения некоторых действий, которые будут выполнены. Процесс, который выполняет те действия, называют сервером. В установленный срок может быть несколько серверов (обычно работающий на других машинах) выполнение указанного сервиса. Например, в Афине существует один сервер rlogin BSD UNIX, работающий на каждой из наших машин разделения по времени.

Ключ, Секретный ключ, Пароль? Kerberos использует шифрование с помощью закрытого ключа. Каждой главной части Kerberos назначают большое число, его секретный ключ, известный только тому принципалу и Kerberos. В случае пользователя секретный ключ является результатом односторонней функции, применился к паролю пользователя. Мы используем ключ в качестве сокращения для секретного ключа.

Учетные данные? К сожалению, это слово имеет особое значение и для Сетевой файловой системы SUN и для системы Kerberos. Мы явно сообщаем, имеем ли мы в виду учетные данные NFS или Учетные данные Kerberos, иначе термин использован в обычном смысле Английского языка.

Ведущее устройство и Ведомое устройство? Возможно выполнить программное обеспечение проверки подлинности Kerberos на нескольких машинах. Однако всегда существует только одна категорическая копия базы данных Kerberos. Машину, которая помещает эту базу данных, называют основным компьютером, или просто ведущим устройством. Другие машины могут обладать копиями только для чтения базы данных Kerberos, и их называют ведомыми устройствами.

Мотивация использования Kerberos

В несетевой среде вычислений на ПК ресурсы и информация могут быть защищены путем

физического обеспечения персонального компьютера. В вычислительной среде разделения по времени операционная система защищает пользователей от друг друга и управляет ресурсами. Для определения то, что каждый пользователь в состоянии считать или модифицировать, необходимо для системы разделения времени определить каждого пользователя. Это выполнено когда входы пользователя в систему в.

В сети пользователей, требующей сервисов от многих отдельных компьютеров, существует три подхода, которые можно проявить к управлению доступом: ничего нельзя сделать, полагаясь на машину, к которой в пользователя входят для предотвращения неавторизованный доступа; можно потребовать, чтобы хост удостоверил свою личность, но доверять слову хоста относительно того, кто пользователь; или можно потребовать, чтобы пользователь доказал подлинность конкретного лица для каждого требуемого сервиса.

В закрытой среде, где все машины находятся под строгим контролем, можно использовать первый подход. Когда контроль организации все хосты, связывающиеся по сети, это - разумный подход.

В большей открытой среде можно было бы выборочно доверять только тем хостам под управлением организации. В этом случае каждый хост должен потребоваться, чтобы удостоверять свою личность. Rlogin и программы rsh используют этот подход. В тех протоколах аутентификация сделана путем проверки интернет-адреса, от которого было установлено соединение.

В Среде Athena нам необходимо соблюдать запросы от хостов, которые не находятся под управлением организации. Пользователи имеют полный контроль над своими рабочими станциями: они могут перезагрузить их, перевести их в рабочее состояние автономный, или даже загрузиться от их собственных лент. Также, третий подход должен быть проявлен; пользователь должен доказать подлинность конкретного лица для каждого заданного сервиса. Сервер должен также удостоверить свою личность. Это не достаточно к физически безопасному хост, выполняющий сервер сети; кто-то в другом месте в сети может подменять данным сервером.

Наша среда размещает несколька требований в механизм идентификации. Во-первых, это должно быть безопасный. Хитрость это должно быть достаточно трудный, что потенциальный нарушитель не находит, что механизм аутентификации слабая ссылка. Кто-то наблюдающий сеть should not быть в состоянии получить информацию, необходимую для исполнения роли другого пользователя. Во-вторых, это должно быть надежный. Доступ ко многим сервисам будет зависеть от сервиса проверки подлинности. Если это не будет надежно, то система сервисов в целом не будет. В-третьих, это должно быть прозрачно. Идеально, пользователь не должен знать опознавательный имеющий место. Наконец, это должно быть масштабируемым. Много систем могут связаться с хостами Афины. Не все они поддержат наш механизм, но разрыв программного обеспечения should not, если они сделали.

Kerberos является результатом нашей работы удовлетворить вышеупомянутые требования. Когда пользователь приближается к рабочей станции, они входят. Насколько пользователь может сказать, эта начальная идентификация достаточна для удостоверения их личности ко всем серверам нужной сети на время сеанса регистрации. Безопасность Kerberos полагается на безопасность нескольких серверов проверки подлинности, но не в системе, от которой пользователи входят, ни на безопасности конечных серверов, которые будут использоваться. Сервер проверки подлинности предоставляет должным образом проверенный пользователь со способом доказать подлинность конкретного лица серверам, рассеянным по сети.

Аутентификация является основополагающим элементом структуры для безопасного сетевого окружения. Если, например, сервер знает наверняка личность клиента, это может решить, предоставить ли сервис, нужно ли пользователю дать особые привилегии, кто должен получить счет для сервиса и т.д. Другими словами, авторизация и бухгалтерские схемы могут быть созданы поверх аутентификации, которую Kerberos предоставляет, приводя к эквивалентному уровню безопасности к одинокому персональному компьютеру или системе разделения времени.

Что такое Kerberos?

Kerberos является надежным сервисом аутентификации третьей стороны на основе модели, представленной Нидхэмом и Шредером. Этому доверяют в том смысле, что каждый из его клиентов полагает, что решение Kerberos относительно идентичности каждого из его других клиентов точно. Метки времени (большие числа, представляющие текущую дату и время), были добавлены к исходной модели для способствования обнаружению воспроизведения. Когда сообщение ускользается сеть и повторно передано позже, воспроизведение происходит. Для большего количества полного описания воспроизведения и других проблем аутентификации, посмотрите Voydock и Кент.

Каковы функции Kerberos?

Kerberos поддерживает базу данных своих клиентов и их секретных ключей. Секретный ключ является большим числом, известным только Kerberos и клиенту, которому это принадлежит. В случае, что клиент является пользователем, это - зашифрованный пароль. Сетевые сервисы, требующие опознавательного регистра с Kerberos, также, как и клиенты, желающие использовать те сервисы. О секретных ключах выполняют согласование при регистрации.

Поскольку Kerberos знает эти секретные ключи, он может создать сообщения, которые убеждают одного клиента, что другой действительно, кем он утверждает, что был. Kerberos также генерирует временные секретные ключи, названные ключами сеанса, которые даны двум клиентам и никому больше. Ключ сеанса может использоваться для шифрования сообщений между двумя сторонами.

Kerberos предоставляет три отдельных уровня защиты. Разработчик приложения определяет, который является соответствующим, согласно требованиям приложения. Например, некоторые приложения требуют только, чтобы подлинность была установлена при инициировании сетевого подключения и могла предположить, что последующие сообщения от заданного сетевого адреса происходят из аутентифицируемой стороны. Наша аутентифицируемая файловая система сети использует этот уровень безопасности.

Другие приложения требуют аутентификации каждого сообщения, но не заботятся, раскрыто ли содержание сообщения или нет. Для них Kerberos предоставляет безопасные сообщения. Все же более высокий уровень безопасности предоставлен личными сообщениями, где каждое сообщение не только аутентифицируется, но также и шифруется. Личные сообщения используются, например, самим сервером Kerberos для передачи паролей по сети.

Компоненты программного обеспечения Kerberos

Внедрение Athena включает несколько модулей:

- Библиотека приложений Kerberos
- библиотека шифрования
- библиотека базы данных
- программы администрирования базы данных
- административный сервер
- сервер проверки подлинности
- программное обеспечение распространения db
- программы пользователя
- приложения

Библиотека Приложений Kerberos предоставляет интерфейс для агентов приложения и серверов приложений. Это содержит, среди других, подпрограмм для создания или чтения запросов аутентификации и подпрограмм для создания сейфа или личных сообщений.

Шифрование в Kerberos основывается на DES, Стандарте шифрования данных. Библиотека шифрования внедряет те подпрограммы. Несколько методов шифрования предоставлены с компромиссами между скоростью и безопасностью. Расширение к DES Cypher Block Chaining (CBC) режим, названный Распространяющимся режимом CBC, также предоставлено. В CBC ошибка распространяется только через текущую блокировку шифра, тогда как в PCBC, ошибка распространяется всюду по сообщению. Это представляет полное сообщение, бесполезное, если ошибка происходит, а не просто часть его. Библиотека шифрования является независимым модулем и может быть заменена другими внедрениями DES или другой библиотекой шифрования.

Другой заменяемый модуль является системой управления базами данных. Текущая конфигурация Athena библиотеки базы данных использует ndbm, невзирая на то, что первоначально использовался Энгр. Другими библиотеками управления базой данных можно было пользоваться также.

Потребности базы данных Kerberos являются прямыми; запись проводится для каждого принципала, содержа название, секретный ключ, и дата окончания действия принципала, наряду с небольшим количеством административной информации. (Дата окончания действия является датой, после которой запись больше не действительна. Это обычно устанавливается в несколько лет в будущее при регистрации.)

Другие сведения о пользователе, такие как настоящее имя, номер телефона, и т.д, сохранены другим сервером, сервером имен Гесиода. Таким образом, уязвимые данные, а именно, пароли, могут быть обработаны Kerberos, с помощью мер по довольно высокому уровню безопасности; в то время как с нечувствительной информацией, хранившей Гесиодом, имеют дело по-другому; это может, например, быть передано дешифрованное по сети.

Серверы Kerberos используют библиотеку базы данных, также, как и программные средства для администрирования базы данных.

Административный сервер (или сервер KDBM) предоставляет сетевой интерфейс чтения-записи базе данных. Клиентская сторона программы может быть выполнена на любой машине в сети. Сторона сервера, однако, должна работать на машине, помещающей базу данных Kerberos для внесения изменений в базу данных.

Сервер проверки подлинности (или сервер Kerberos), с другой стороны, выполняет операции чтения на базе данных Kerberos, а именно, проверку подлинности принципалов и генерацию ключей сеанса. Так как этот сервер не модифицирует базу данных Kerberos, он

может работать на машине, помещающей копию только для чтения главной базы данных Kerberos.

Программное обеспечение распространения базы данных управляет репликацией базы данных Kerberos. Возможно иметь копии базы данных по нескольким другим машинам с копией сервера проверки подлинности, работающего на каждой машине. Каждая из этих ведомых машин получает обновление базы данных Kerberos от основного компьютера в данных интервалах.

Наконец, существуют программы конечного пользователя для регистрации к Kerberos, изменение Пароля Kerberos, и отображение или уничтожение билетов Kerberos (билеты объяснены позже).

Имена Kerberos

Часть аутентификации объекта называет его. Процесс аутентификации является проверкой, что клиент является тем, названным в запросе. Из чего состоит название? В Kerberos называют и пользователей и серверы. Насколько сервер проверки подлинности затронут, они эквивалентны. Название состоит из исходного имени, экземпляра и области, выраженной как название instance@realm.

Исходное имя является именем пользователя или сервиса. Экземпляр используется для различения изменения на исходном имени. Для пользователей экземпляр может повлечь за собой особые привилегии, такие как экземпляры "admin" или "root". Для сервисов в Среде Athena экземпляр обычно является названием машины, на которой выполняется сервер. Например, сервис rlogin имеет другие экземпляры на других хостах: rlogin.priam является сервером rlogin на хосте с именем priam. Билет Kerberos только хорош для одиночного именованного сервера. Также, отдельный билет требуется, чтобы получать доступ к другим экземплярам того же сервиса. Область является названием административной записи, которая поддерживает данные проверки подлинности. Например, другие учреждения могут каждый иметь свою собственную машину Kerberos, помещая другую базу данных. У них есть другие Области "Kerberos". (Именованные области (Realm) обсуждены далее во Взаимодействии с другими Kerberos.)

Как работает Kerberos

В этом разделе описываются протоколы проверки подлинности Kerberos. Как упомянуто выше, Модель проверки подлинности Kerberos основывается на протоколе распределения ключей Нидхэма и Шредера. Когда запросы пользователя сервис, должна быть установлена подлинность конкретного лица. Чтобы сделать это, билет представлен серверу, наряду с доказательством, что билет был первоначально выполнен пользователю, не украденному. Существует три фазы к аутентификации через Kerberos. В первой фазе пользователь получает учетные данные, которые будут использоваться, чтобы запросить доступ к другим сервисам. Во второй фазе, аутентификации запросов пользователя для определенного сервиса. В конечной фазе пользователь представляет те учетные данные до конца сервер.

Учетные данные Kerberos

Существует два типа учетных данных, используемых в Модели проверки подлинности Kerberos: билеты и средства проверки подлинности. Оба основываются на шифровании с

помощью закрытого ключа, но они зашифрованы с помощью других ключей. Билет используется для безопасной передачи личности человека, которому билет был выполнен между сервером проверки подлинности и конечным сервером. Билет также передает информацию, которая может использоваться, чтобы удостовериться, что человек, использующий билет, является тем же человеком, к которому это было выполнено. Средство проверки подлинности содержит дополнительные сведения, которые, когда сравнено с этим в билете доказывают, что клиент, представляющий билет, является тем же самым, к которому был выполнен билет.

Билет хорош для одиночного сервера и одиночного клиента. Это содержит название сервера, имя клиента, интернет-адрес клиента, метки времени, срока действия и случайного ключа сеанса. Эта информация зашифрована с помощью ключа сервера, для которого будет использоваться билет. Как только билет был выполнен, он может использоваться многократно именованным клиентом для получения доступа к именованному серверу, пока не истекает билет. Обратите внимание на то, что, потому что билет зашифрован в ключе сервера, безопасно позволить пользователю передавать билет на сервер, не имея необходимость волноваться о пользователе, модифицирующем билет.

В отличие от билета, средство проверки подлинности может только использоваться однажды. Новый должен генерироваться каждый раз, когда клиент хочет использовать сервис. Это не представляет проблему, потому что клиент в состоянии создать само средство проверки подлинности. Средство проверки подлинности содержит имя клиента, IP-адреса рабочей станции, и текущее время рабочей станции. Средство проверки подлинности зашифровано в ключе сеанса, который является частью билета.

Получите начальный мандат Kerberos

Когда пользователь приближается к рабочей станции, только одна часть информации может доказать подлинность конкретного лица: пароль пользователя. Начальный обмен с сервером проверки подлинности разработан для сведения к минимуму вероятности, которую пароль поставится под угрозу, в то же время не позволяя пользователю должным образом аутентифицировать себя без ведома того пароля. Процесс входа в систему, кажется пользователю совпадает с регистрацией к системе разделения времени. Негласно, тем не менее, это очень отличается.

Пользователю предлагают для ее/его имени пользователя. Как только это было введено, запрос отправлен к серверу проверки подлинности, содержащему название пользователя и название специального сервиса, известного как сервис мандатов.

Сервер проверки подлинности проверяет, что знает о клиенте. Если так, это генерирует случайный ключ сеанса, который будет позже использоваться между клиентом и сервером выдачи разрешений. Это тогда создает тикет для сервера выдачи разрешений, который содержит название клиента, название сервера выдачи разрешений, текущее время, срок действия для билета, IP-адреса клиента и случайного ключа сеанса, просто созданного. Это все зашифровано в ключе, известном только серверу выдачи разрешений и серверу проверки подлинности.

Сервер проверки подлинности тогда передает билет, наряду с копией случайного ключа сеанса и некоторых дополнительных сведений, назад клиенту. Этот ответ зашифрован в секретном ключе клиента, известном только Kerberos и клиенту, который получен из пароля пользователя.

Как только ответ был получен клиентом, пользователя просят относительно ее/его пароля. Пароль преобразовывается в ключ DES и используется для дешифрования ответа от сервера проверки подлинности. Билет и ключ сеанса, наряду с частью другой информации, сохранены для дальнейшего использования, и пароль пользователя и ключ DES стерты из памяти.

Как только обмен был завершен, рабочая станция обладает информацией, которую это может использовать для удостоверения личности ее пользователя для срока действия разрешения на получение разрешения. Пока в программное обеспечение на рабочей станции ранее не вмешались, никакая информация не существует, который позволит кому-то еще исполнять роль пользователя за пределами жизнь билета.

Запросите сервис Kerberos

В настоящий момент давайте притворимся, что у пользователя уже есть билет для необходимого сервера. Для получения доступа к серверу приложение создает средство проверки подлинности, содержащее название и IP-адрес клиента, и текущее время. Средство проверки подлинности тогда зашифровано в ключе сеанса, который был получен с билетом для сервера. Клиент тогда передает средство проверки подлинности наряду с билетом к серверу способом, определенным отдельным приложением.

Как только средство проверки подлинности и билет были получены сервером, сервер дешифрует билет, использует ключ сеанса, включенный в билет для дешифрования средства проверки подлинности, сравнивает информацию в билете с этим в средстве проверки подлинности, IP-адресе, от которого запрос был получен, и настоящее время. Если все совпадает, это позволяет запросу продолжиться.

Предполагается, что часы синхронизируются с в течение нескольких минут. Если время в запросе слишком далеко в будущем или прошлом, сервер рассматривает запрос как попытку воспроизвести предыдущий запрос. Серверу также позволяют отслеживать все прошлые запросы с метками времени, которые все еще допустимы. Для дальнейшей помехи атакам с повторением пакетов, запросу, полученному с тем же билетом и меткой времени как один уже полученный может быть сброшен.

Наконец, если клиент указывает, что это хочет, чтобы сервер удостоверил свою личность также, сервер добавляет тот к метке времени клиент, передаваемый в средстве проверки подлинности, шифрует результат в ключе сеанса и передает результат обратно клиенту.

В конце этого обмена сервер уверен, что, согласно Kerberos, клиент - то, кто это говорит, что это. Если обоюдная проверка подлинности происходит, клиент также убежден, что сервер подлинен. Кроме того, клиент и сервер совместно используют ключ, который никто больше не знает и может безопасно предположить, что обоснованно последнее сообщение зашифровало в том ключе, инициируемом с другой стороной.

Получите мандаты сервера Kerberos

Вспомните, что билет только хорош для одиночного сервера. Также, необходимо получить отдельный билет для каждого сервиса, который клиент хочет использовать. Билеты для индивидуальных серверов могут быть получены из сервиса мандатов. Так как сервис мандатов является самостоятельно сервисом, он использует сервисный протокол доступа, описанный в предыдущем разделе.

Когда программа требует билета, который уже не запросили, она отправляет запрос к серверу выдачи разрешений. Запрос содержит название сервера, на который билет запрашивают, наряду с разрешением на получение разрешения и средством проверки подлинности, созданным, как описано в предыдущем разделе.

Сервер выдачи разрешений тогда проверяет средство проверки подлинности и разрешение на получение разрешения, как описано выше. Если допустимый, сервер выдачи разрешений генерирует новый случайный ключ сеанса, который будет использоваться между клиентом и новым сервером. Это тогда создает билет для нового сервера, содержащего название клиента, имя сервера, текущее время, IP-адрес клиента и новый ключ сеанса, который это просто генерировало. Срок действия нового билета является минимумом остаточного срока службы для разрешения на получение разрешения и по умолчанию для сервиса.

Сервер выдачи разрешений тогда передает билет, наряду с ключом сеанса и другой информацией, назад клиенту. На этот раз, однако, ответ зашифрован в ключе сеанса, который был частью разрешения на получение разрешения. Таким образом, нет никакой потребности в пользователе ввести ее/его пароль снова.

База данных Kerberos

До этой точки мы обсудили операции, требующие доступа только на чтение к базе данных Kerberos. Эти операции выполнены сервисом проверки подлинности, который может работать и на основных и ведомых машинах.

В этом разделе мы обсуждаем операции, которые требуют доступа с правом записи к базе данных. Эти операции выполнены административной службой, названной Kerberos Database Management Service (KDBM). Текущая реализация предусматривает, что изменения могут только быть внесены в главную базу данных Kerberos; ведомые копии только для чтения. Поэтому сервер KDBM может только работать на главном компьютере Kerberos.

Обратите внимание на то, что, в то время как аутентификация может все еще произойти (на ведомых устройствах), запросы администрирования не могут быть обслужены, если основной компьютер не работает. В нашем опыте это не представило проблему, поскольку запросы администрирования являются нечастыми.

KDBM обрабатывает запросы от пользователей для изменения их паролей. Клиентская сторона этой программы, которая отправляет запросы к KDBM по сети, является программой kpasswd. KDBM также принимает запросы от Администраторов Kerberos, которые могут добавить принципалы к базе данных, а также изменить пароли для существующих принципов. Клиентская сторона программы администрирования, которая также отправляет запросы к KDBM по сети, является программой Kadmin.

Сервер KDBM

Сервер KDBM принимает запросы добавить принципалы к базе данных или изменить пароли для существующих принципов. Этот сервис уникален в этом, сервис мандатов не выполнит билеты для него. Вместо этого сам сервис проверки подлинности должен использоваться (тот же сервис, который используется для получения разрешения на получение разрешения). Цель этого состоит в том, чтобы потребовать, чтобы пользователь ввел пароль. Если бы это не было так, то, если пользователь оставил ее/его рабочую

станцию необслуживаемой, прохожий мог бы приблизиться и изменить ее/его пароль для них, что-то, что должно быть предотвращено. Аналогично, если бы администратор оставил ее/его рабочую станцию неосторожной, то прохожий мог бы изменить любой пароль в системе.

Когда сервер KDBM получает запрос, он авторизует его путем сравнения аутентифицируемого главного имени запрашивающей стороны изменения к главному имени цели запроса. Если они - то же, запрос разрешен. Если они не то же, сервер KDBM консультируется со списком контроля доступа (сохраненный в файле на основной системе Kerberos). Если главное имя запрашивающей стороны найдено в этом файле, запрос разрешен, иначе это запрещено.

Условно, названия с экземпляром NULL (экземпляр по умолчанию) не появляются в файле списка контроля доступа; вместо этого, администраторская запись используется. Поэтому для пользователя для становления администратором Kerberos администраторская запись для того имени пользователя должна быть создана и добавлена к списку контроля доступа. Это соглашение позволяет администратору использовать другой администраторский пароль Kerberos тогда, она/он использовала бы для обычного входа в систему.

Все запросы к программе KDBM, или разрешенный или запрещенный, зарегистрированы.

Программы kadmin и kpasswd

Администраторы Kerberos используют программу Kadmin, чтобы добавить принципалы к базе данных или изменить пароли существующих принципов. Администратор обязан вводить пароль для их названия администраторской записи, когда они вызывают программу Kadmin. Этот пароль используется для выборки билета для сервера KDBM.

Пользователи могут изменить свои Пароли Kerberos с помощью программы kpasswd. Они обязаны вводить свой старый пароль, когда они вызывают программу. Этот пароль используется для выборки билета для сервера KDBM.

Репликация базы данных Kerberos

Каждая Область "Kerberos" имеет главный компьютер Kerberos, который помещает основную копию базы данных проверки подлинности. Возможно (невзирая на то, что не необходимый) иметь дополнительный, копии только для чтения базы данных по ведомым машинам в другом месте в системе. Преимущества наличия множественных копий базы данных являются обычно цитируемыми за репликацию: более высокая доступность и лучшая производительность. Если основной компьютер не работает, аутентификация может все еще быть достигнута на одной из ведомых машин. Способность выполнить аутентификацию на любой из нескольких машин уменьшает вероятность узкого места в основном компьютере.

Хранение множественных копий базы данных представляет проблему целостности данных. Мы нашли, что очень простые методы достаточны для контакта с несоответствием. Главная база данных разгружается каждый час. База данных передается, полностью, к ведомым машинам, которые тогда обновляют их собственные базы данных. Программа на основном узле, названном kprop, передает обновление одноранговой программы, названной kpropd, работая на каждой из ведомых машин. Первый kprop передает контрольную сумму новой базы данных, которую он собирается передать. Контрольная сумма зашифрована в ключе

главной базы данных Kerberos, которым обладают и основные и ведомые машины Kerberos. Данные тогда переданы по сети kpropd на ведомой машине. Ведомый сервер распространения вычисляет контрольную сумму данных, которые он получил, и если он совпадает с контрольной суммой, передаваемой ведущим устройством, новая информация используется для обновления базы данных ведомого устройства.

Все пароли в базе данных Kerberos зашифрованы в ключе главной базы данных Поэтому, информация, которую передают от ведущего устройства к ведомому устройству по сети, не полезна для eavesdropper. Однако важно, что только информация от основного узла принята ведомыми устройствами, и что вмешательство данных, которые будут обнаружены, таким образом контрольная сумма.

Kerberos с разных точек зрения

В этом разделе описываются Kerberos с практической точки зрения, сначала, как замечено пользователем, затем с точки зрения разработчика приложения, и наконец, через задачи Администратора Kerberos.

Kerberos с точки зрения пользователя

Если все будет подходить, то пользователь едва заметит, что присутствует Kerberos. В нашей реализации Unix разрешение на получение разрешения получено из Kerberos как часть процесса регистрации в системе. Изменение Пароля Kerberos пользователя является частью программы passwd. И билеты Kerberos автоматически уничтожены когда входы пользователя в систему.

Если сеанс регистрации пользователя продлится дольше, чем срок действия разрешения на получение разрешения (в настоящее время 8 часов), то пользователь заметит присутствие Kerberos, потому что в следующий раз аутентифицируемое на Kerberos приложение выполняется, это откажет. Билет Kerberos для него истечет. В той точке пользователь может выполнить kinit программу для получения нового билета для сервера выдачи разрешений. Входя, пароль должен быть предоставлен для получения его. Пользователь, выполняющий klist команду из любопытства, может быть удивлен всеми билетами, которые были тихо получены от ее/его имени для сервисов, которые требуют проверки подлинности Kerberos.

Протокол Kerberos с точки зрения программиста

Программист, пишущий Приложение Kerberos уже, будет часто добавлять аутентификацию к приложение существующей сети, состоящее из стороны клиента и сервера. Мы называем этот процесс "Kerberizing" программой. Kerberizing обычно включает звонка к Библиотеке Kerberos для выполнения аутентификации в исходном запросе для сервиса. Это может также включить вызовы к библиотеке DES для шифрования сообщений и данных, которые впоследствии передаются между агентом приложения и сервером приложений.

Обычно используемые функции библиотеки являются krb_mk_req на клиентской стороне и krb_rd_req на стороне сервера. krb_mk_req подпрограмма берет в качестве параметров название, экземпляр и область конечного сервера, который запросят, и возможно контрольная сумма данных, которые будут передаваться. Клиент тогда передает сообщение, возвращенное krb_mk_req перекличкой сеть к стороне сервера приложения. Когда сервер получает это сообщение, он звонит к krb_rd_req программы библиотеки.

Подпрограмма возвращает суждение о подлинности предполагаемой идентичности отправителя.

Если приложение требует, чтобы сообщения, передаваемые между клиентом и сервером, были секретными, то вызовы библиотеки могут быть выполнены к krb_mk_priv (krb_rd_priv) для шифрования (дешифруют) сообщения в ключе сеанса, который теперь совместно используют обе стороны.

Работа администратора Kerberos

Задание Администратора Kerberos начинается с выполнения программы для инициализации базы данных. Другая программа должна быть выполнена для регистрации основных принципов в базе данных, таких как название Администратора Kerberos с администраторской записью. Сервер проверки подлинности Kerberos и административный сервер должны быть запущены. Если существуют ведомые базы данных, администратор должен расположить, что программы для распространения обновлений базы данных от ведущего устройства к ведомым устройствам периодически начинаются.

После того, как эти первые шаги были взяты, администратор манипулирует базой данных по сети, с помощью программы Kadmin. Через ту программу могут быть добавлены новые принципалы, и пароли могут быть изменены.

В частности когда новое Приложение Kerberos добавлено к системе, Администратор Kerberos должен сделать несколько шагов для получения его работа. Сервер должен быть зарегистрирован в базе данных и назначил секретный ключ (обычно, это - автоматически генерируемый случайный ключ). Затем некоторые данные (включая ключ сервера) должны быть извлечены из базы данных и установлены в файле на машине сервера. Файл по умолчанию является/etc/srvtab. Программа библиотеки krb_rd_req, вызванная сервером (см. предыдущий раздел), использует информацию в том файле для дешифрования сообщений, передаваемых зашифрованными в секретном ключе сервера./etc/srvtab файл аутентифицирует сервер, как пароль, введенный в терминале, аутентифицирует пользователя.

Администратор Kerberos должен также гарантировать, что машины Kerberos физически безопасны, и также были бы мудры для поддержания резервных копий Главной базы данных.

Больший рисунок Kerberos

В этом разделе мы описываем, как Kerberos вписывается в Среду Athena, включая ее использование другими сетевыми сервисами и приложениями, и как это взаимодействует с удаленными Областями "Kerberos". Для большего количества полного описания Среды Athena см. G.W. Treese.

Другое использование Kerberos сетевыми службами

Несколько сетевых приложений модифицировались для использования Kerberos. Команды rlogin и rsh сначала пробуют к Kerberos используемой аутентификации. Пользователь с допустимыми билетами Kerberos может rlogin к другой машине Афины, не имея необходимость устанавливать файлы .rhosts. Если проверка подлинности Kerberos отказывает, программы возвращаются к своим обычным методам авторизации, в этом

случае, файлам .rhosts.

Мы модифицировали Почтовый протокол для использования Kerberos для аутентификации пользователей, которые хотят получить их электронную почту из "почты". Программа доставки сообщения, названная Zephyr, была недавно разработана в Афине, и это использует Kerberos для аутентификации также.

Программа для новых пользователей регистрации, названных регистром, использует и Service Management System (SMS) и Kerberos. От CM это определяет, допустима ли информация, введенная потенциальным новым пользователем Athena, таким как название и идентификационный номер MIT. Это тогда сверяется с Kerberos, чтобы видеть, уникально ли запрошенное имя пользователя. Если все подходит, новая запись сделана к базе данных Kerberos, содержа имя пользователя и пароль.

Для подробного обсуждения использования Kerberos для обеспечения Файловой системы сети Sun, см. приложение.

Взаимодействие с другими Kerberi

Ожидается, что другие административные организации захотят использовать Kerberos для проверки подлинности пользователя. Также ожидается, что во многих случаях, пользователи в одной организации захотят использовать сервисы в другом. Административные домены Kerberos поддерживает несколько административных доменов. Спецификация названий в Kerberos включает поле, названное областью. Это поле содержит название административного домена, в котором должен аутентифицироваться пользователь.

Сервисы обычно регистрируются в одиночной области и только примут учетные данные, выполненные сервером проверки подлинности для той области. Пользователь обычно регистрируется в одиночной области (локальная сфера), но для нее/его возможно получить учетные данные, выполненные другой областью (удаленная область), на основании аутентификации, предоставленной локальной сферой. Учетные данные, допустимые в удаленной области, указывают на область, в которой первоначально аутентифицировался пользователь. Сервисы в удаленной области могут выбрать, соблюдать ли те учетные данные, в зависимости от требуемой степени безопасности и уровень доверия в области, которая первоначально аутентифицировала пользователя.

Для выполнения аутентификации перекрестной области необходимо, чтобы администраторы каждой пары областей выбрали ключ, который будет разделен между их областями. Пользователь в локальной сфере может тогда запросить разрешение на получение разрешения от локального сервера проверки подлинности для сервера выдачи разрешений в удаленной области. Когда тот билет используется, удаленный сервер выдачи разрешений распознает, что запрос не от его собственной области, и это использует ранее обмененный ключ для дешифрования разрешения на получение разрешения. Это тогда выполняет билет, как это обычно было бы, за исключением того, что поле области для клиента содержит название области, в которой первоначально аутентифицировался клиент.

Этот подход мог быть расширен, чтобы позволить тому аутентифицировать себя через последовательность областей до достижения области с заданным сервисом. Чтобы сделать это, тем не менее, было бы необходимо сделать запись всего пути, который был взят, и не только название начальной области, в которой аутентифицировался пользователь. В такой ситуации все, что известно сервером, - то, что SAID, что B говорит, что C говорит, что

пользователь является таким-сяким. Этому оператору можно только доверять, если всем вдоль пути также доверяют.

Вопросы и открытые проблемы Kerberos

Существует много неполадок и открытых проблем, привязанных к механизму проверки подлинности Kerberos. Среди проблем то, как решить корректный срок действия для билета, как позволить прокси, и как гарантировать целостность рабочей станции.

Проблемой срока действия билета является вопрос выбора подходящего компромисса между безопасностью и удобством. Если жизнь билета длинна, то, если билет и его связанный ключ сеанса украдены или неуместны, они могут использоваться для более длинного периода времени. Если пользователь забывает выходить из общедоступной рабочей станции, такая информация может быть украдена. Также, если пользователь аутентифицировался в системе, которая позволяет несколько пользователей, другой пользователь с доступом для укоренения мог бы быть в состоянии найти, что информация должна была использовать украденные билеты. Проблема с предоставлением билета, который короткий срок действия, однако, то, что, когда это истекает, пользователь должен будет получить новый, который требует, чтобы пользователь ввел пароль снова.

Открытая проблема является проблемой с прокси. Как проверенный пользователь может позволить серверу получать другие сетевые сервисы от ее/его имени? Примером, где это было бы важно, является использование сервиса, который получит доступ к защищенным файлам непосредственно от файлового сервера. Другой пример этой проблемы что мы передача аутентификации вызова. Если бы пользователь зарегистрирован в рабочую станцию и входит к удаленному хосту, было бы хорошо, если бы у пользователя был доступ к тем же сервисам, доступным локально при выполнении программы на удаленном хосте. Что делает, это трудное - то, что пользователь не мог бы доверять удаленному хосту, таким образом пересылка проверки подлинности не выбираема во всех случаях. У нас в настоящее время нет решения этой проблемы.

Другая проблема и та, которая важна в Среде Athena, состоят в том, как гарантировать целостность программного обеспечения, работающего на рабочей станции. Это не большая часть проблемы на закрытых рабочих станциях, так как пользователь, который будет использовать ее, управляет ею. На общедоступных рабочих станциях, однако, кто-то, возможно, приехал и модифицировал программу входа в систему для сохранения пароля пользователя. Единственное решение, в настоящее время доступное в нашей среде, состоит в том, чтобы мешать людям модифицировать программное обеспечение, работающее на общедоступных рабочих станциях. Лучшее решение потребовало бы, чтобы ключ пользователя никогда не оставлял систему, которую знает пользователь, может доверяться. Одним путем это могло быть сделано, был бы то, если бы пользователь обладал смарт-картой, способной к выполнению шифрования, требуемого в протоколе аутентификации.

Статус Kerberos

Экспериментальная версия Kerberos вошла в производство в сентябре 1986. С января 1987 Kerberos был единственными средствами Проекта Athena аутентификации его 5,000 пользователей, 650 рабочих станций и 65 серверов. Кроме того, Kerberos теперь используется вместо файлов .rhosts для управления доступом в нескольких из систем разделения времени Афины.

Подтверждение Kerberos

Kerberos был первоначально разработан Steve Miller и Clifford Neuman с предложениями от Джефа Шиллера и Jerry Saltzer. С этого времени многочисленные другие люди были связаны с проектом. Среди них Jim Aspnes, Bob Baldwin, Джон Барба, Ричард Бэш, Jim Bloom, Bill Bryant, Mark Colan, Rob French, Dan Geer, Джон Коль, Джон Кубиэтович, Боб Макки, Brian Murphy, Ken Raeburn Джона Остланда, Chris Reed, Jon Rochlis, Mike Shanzer, Bill Sommerfeld, Тед T'co, Win Treese и Stan Zanarotti.

Мы благодарны Dan Geer, Кэти Либен, Джошу Лубарру, Ken Raeburn, Jerry Saltzer, Ed Steiner, Robbert van Ренессе и Win Treese, предложения которого очень улучшили более ранние проекты этой бумаги.

Jedlinsky, J.T. Коль и W.E. Зоммерфельд, "система оповещений Zephyr", в проведениях конференции Usenix (зима, 1988).

M.A. Розенштейн, D.E. Geer и P.J. Левин, в проведениях конференции Usenix (зима, 1988).

R. Sandberg, D. Голдберг, S. Клейман, D. Уолш и B. Лион, "Разработка и реализация сетевой файловой системы Sun", в проведениях конференции Usenix (лето, 1985).

Приложение: Приложение Kerberos для сетевой файловой системы (NFS) SUN

Основной компонент системы рабочей станции Проекта Athena является вставкой сети между рабочей станцией пользователя и ее/его хранилищем личного файла (главный каталог). Вся частная система хранения данных находится на ряде компьютеров (в настоящее время VAX 11/750s), которые выделены этой цели. Это позволяет нам предлагать услуги на общедоступных рабочих станциях UNIX. Когда входы пользователя в систему в к одной из этих общедоступных рабочих станций, скорее тогда проверьте ее/его имя и пароль против локально резидентского файла с паролями, мы используем Kerberos для определения ее/его подлинности. Программа входа в систему вызывает для имени пользователя (как на любой системе UNIX). Это имя пользователя используется для выборки разрешения на получение разрешения Kerberos. Программа входа в систему использует пароль для генерации ключа DES для дешифрования билета. Если расшифровка успешна, главный каталог пользователя расположен путем консультации с сервисом назначения имен Гесиода и установлен через NFS. Программа входа в систему тогда передает контроль в оболочку пользователя, которая тогда может выполнить традиционные файлы настройки по каждому пользователю, потому что главный каталог теперь "подключен" к рабочей станции. Сервис Гесиода также используется для построения записи в файле локального пароля. (Это в пользу программ что информация о поиске в/etc/passwd.)

От нескольких вариантов поставки удаленного file service мы выбрали Sun's Network File System. Однако, эта система не в состоянии сцепляться с нашими потребностями в ключевом пути. NFS предполагает, что все рабочие станции попадают в две категории (как просматривается с точки зрения файлового сервера): доверяемый и недоверяемый. Системы не имеющая доверия не могут обратиться ни к каким файлам вообще, доверяемый может. Надежным системам полностью доверяют. Предполагается, что надежной системой управляет дружественный менеджмент. В частности возможно от

доверенной рабочей станции подменить любым допустимым пользователем системы file service и таким образом получить доступ к примерно каждому файлу в системе. (Только файлы, принадлежавшие "root", освобождены.)

В нашей среде управление рабочей станции (в традиционном смысле управления системы UNIX) находится в руках пользователя, в настоящее время использующего его. Мы не делаем тайны пароля при загрузке на наших рабочих станциях, поскольку мы понимаем, что действительно недружественный пользователь может ворваться самым фактом, что он или она находится в том же физическом размещении как машина и имеет доступ ко всем функциям консоли. Поэтому мы не можем действительно трастовый наши рабочие станции в интерпретации NFS доверия. Позволять соответствующий доступ управляет в нашей среде, мы должны были сделать некоторые модификации к основному программному обеспечению NFS и интегрировать Kerberos в схему.

NFS, не измененная Kerberos

В реализации NFS, который мы запустили с (из университета Висконсина), аутентификация была предоставлена в форме части данных, включенных в каждый запрос NFS (названный "учетными данными" в терминологии NFS). Эти учетные данные содержат информацию об уникальном идентификаторе пользователя (UID) запрашивающей стороны и список идентификаторов группы (GIDS) членства запрашивающей стороны. Эта информация тогда используется сервером NFS для проверки доступа. Различие между доверяемым и рабочей станцией недоверенного - приняты ли ее учетные данные сервером NFS.

Серверы NFS с поддержкой Kerberos

В нашей среде серверы NFS должны принять учетные данные от рабочей станции, если и только если учетные данные указывают на UID пользователя рабочей станции и никого другого.

Одно очевидное решение должно было бы изменить характер учетных записей от простых индикаций UID и GIDS к полноценным данным, прошедшим проверку подлинности Kerberos. Если бы это решение было принято, однако, значительное снижение производительности было бы заплачено. Учетными данными обмениваются на каждой операции NFS включая все чтение с диска и действия записи. Включая проверку подлинности Kerberos на каждой дисковой транзакции добавил бы справедливое количество полноценного шифрования (сделанный в программном обеспечении) на транзакцию и, согласно нашим расчетам конверта, отправит неприемлемую производительность. (Это также потребовало бы размещения подпрограмм Библиотеки Kerberos в адресном пространстве ядра.)

Нам был нужен гибридный подход, описанный ниже. Основной идее нужно было получить учетные данные карты сервера NFS от клиентских рабочих станций к допустимому (и возможно другой) учетные данные на серверной системе. Это сопоставление выполнено в ядре сервера на каждом NFS - транзакции и является настройкой во время "установки" процессом пользовательского уровня, который участвует в Kerberos сдерживал проверку подлинности до установления сопоставления учетных данных допустимого сопоставления.

Для реализации этого, мы добавили новый системный вызов к ядру (требуемый только на серверных системах, не на системах клиента), который обеспечивает контроль функции сопоставления, которая сопоставляет входящие учетные данные от клиентских рабочих станций до учетных данных, допустимых для использования на сервере (если таковые

имеются). Основная функция сопоставления сопоставляет кортеж:

к допустимым учетным данным NFS на серверной системе. CLIENT-IP-ADDRESS извлечен из пакета запроса NFS, предоставленного системой клиента. Примечание: от всей информации в генерируемых клиентами учетных данных кроме UID-ON-CLIENT сбрасывают.

Если никакое сопоставление не существует, сервер реагирует одним из двух способов, завися это настроено. В нашей дружественной конфигурации мы принимаем значение по умолчанию неотображаемые запросы в учетные данные для пользователя "никто", кто не имеет никакого привилегированного адреса и имеет уникальный UID. Когда никакое действительное сопоставление не может быть найдено для входящих учетных данных NFS, недружелюбные серверы возвращают ошибку доступа NFS.

Наш новый системный вызов используется, чтобы добавить и удалить записи из отображения резидентов Kernel. Это также предоставляет способность сбросить все записи, которые сопоставляют с определенным UID на серверной системе или сбрасывают все записи от данного CLIENT-IP-ADDRESS.

Мы модифицировали демона установки (который обрабатывает запросы установки NFS на серверных системах) принять новый тип транзакции, запрос сопоставления проверки подлинности Kerberos. В основном, как часть процесса установки, система клиента предоставляет Аутентификатор Kerberos наряду с индикацией относительно ее/его UID-ON-CLIENT (зашифрованный в Аутентификаторе Kerberos) на рабочей станции. Демон установки сервера преобразовывает название главной части Kerberos в локальное имя пользователя. Это имя пользователя тогда ищется в особом файле для получения UID пользователя и списка GIDS. Для эффективности этот файл является ndbm файлом базы данных с именем пользователя как ключ. От этой информации учетные данные NFS созданы и вручены ядру как действительное сопоставление <CLIENT-IP-ADDRESS, кортежа CLIENT-UID> для этого запроса.

Во время unmount запрос отправлен демону установки для удаления ранее добавленного сопоставления из ядра. Также возможно отправить запрос во время выхода из системы для лишения законной силы всего сопоставления для текущего пользователя на рассматриваемом сервере, таким образом очищая любые остающиеся сопоставления, которые существуют (хотя они не были должны), прежде чем рабочая станция будет сделана доступной для следующего пользователя.

Значение измененной NFS для безопасности Kerberos

Эта реализация не абсолютно безопасна. Для начинающих пользовательские данные все еще передаются по сети в незашифрованном, и поэтому interceptable, форма. Низкий уровень, аутентификация на транзакцию основывается на <CLIENT-IP-ADDRESS, пара CLIENT-UID> предоставила дешифрованный в пакете запроса. Эта информация могла быть создана и таким образом поставившая под угрозу безопасность. Однако нужно обратить внимание, что только, в то время как пользователь активно использует ее/его файлы (т.е. в то время как вошли) существуют действительные сопоставления, и поэтому эта форма атаки ограничена тем, когда входят в рассматриваемого пользователя. Когда в пользователя не войдут, никакая сумма подделки IP-адреса не разрешит неавторизованный доступ к ее/его файлам.

Справочник по Kerberos

1. S.P. Миллер, B.C. Неумен, J.I. Шиллер и J.H. Saltzer, раздел E.2.1: проверка подлинности Kerberos и система авторизации, проект Athena M.I.T., Кембридж, Массачусетс (21 декабря 1987).
2. E. Балькович, S.R. Лермен и R.P. Parmelee, "Вычисляющий в Высшем образовании: Опыт Афины", Связь ACM, Издания 28 (11), стр 1214-1224, ACM (ноябрь 1985).
3. R.M. Нидхэм и доктор медицины Шредер, "Использование Шифрования для аутентификации в Больших сетях Компьютеров", Связь ACM, Издания 21 (12), стр 993-999 (декабрь 1978).
4. V.L. Voydock и Кент S.T., "Механизмы обеспечения безопасности в сетевых протоколах высокого уровня", компьютерные обозрения, издание 15 (2), ACM (июнь 1983).
5. Национальное бюро стандартов, "стандарт шифрования данных", публикация 46 федеральных стандартов обработки информации (FIPS), правительствнное издательство, Вашингтон, округ Колумбия (1977).
6. Красильщик SP, "Гесиод", в проведениях конференции Usenix (зима, 1988).
7. W.J. Брайант, Учебное руководство программиста Kerberos, Проект Athena Массачусетского технологического института (MIT) (В подготовке).
8. W.J. Брайант, Руководство Администратора Kerberos, Проект Athena Массачусетского технологического института (MIT) (В подготовке).
9. G.W. Treese, "Berkeley Unix на 1000 рабочих станций: Афина изменяется на 4.3BSD", в проведениях конференции Usenix (зима, 1988).
10. C . O. DellaFera, M.W. Eichin, R.S. Французский, D.C. Jedlinsky, J.T. Коль и W.E. Зоммерфельд, "система оповещений Zephyr", в проведениях конференции Usenix (зима, 1988).
11. M.A. Розенштейн, D.E. Geer и P.J. Левин, в проведениях конференции Usenix (зима, 1988).
12. R. Sandberg, D. Голдберг, S. Клейман, D. Уолш и B. Лион, "Разработка и реализация сетевой файловой системы Sun", в проведениях конференции Usenix (лето, 1985).

Дополнительные сведения

- Страница поддержки Kerberos
- Cisco Systems – техническая поддержка и документация