Д.Н. СТАРОДУБОВ

Муромский институт (филиал) ФГБОУ ВПО «Владимирский государственный университет имени А.Г. и Н.Г. Столетовых», г.Муром

LDAP-аутентификация в сервере баз данных

УДК 004.658.2

В статье рассмотрен метод аутентификации пользователей в сервере баз данных на основе информации из службы каталогов LDAP.

Article describes method of users authentication in database server when account information is stored in LDAP.

Базы данных различного характера используются в настоящее время практически повсеместно. Они хранят банковские данные, содержимое веб-сайтов, цифровые изображения, сведения о пользователях и клиентах, экспериментальные данные и многое другое. Чаще всего доступ к этим данным ограничен и предоставляется только пользователям, зарегистрированным на сервере БД.

При подключении к базе данных такие пользователи проходят процедуру аутентификации – проверки их подлинности. Простейший и наиболее распространённый вариант такой проверки – сравнение пароля, предъявленного пользователем, с паролем, хранящимся в его учётной записи на сервере. Могут быть использованы и другие механизмы аутентификации – например, с использованием сертификата пользователя или его биометрических данных – отпечатков пальцев, рисунка сетчатки глаза и т.д.

Традиционно учётные данные пользователей содержатся в каком-либо локальном для сервера БД хранилище – обычно в самой базе данных, к которой пользователь будет иметь доступ, или в специальной базе данных безопасности.

У такого способа есть существенный недостаток – сложность или невозможность синхронизации учётных данных пользователя на сервере баз данных с его учётными данными в других приложениях.

Ведь обычно люди, работающие с базами данных, используют и множество других приложений с парольной защитой: сама операционная система, электронная почта, системы контроля версий, прокси-сервера, общие сетевые ресурсы и т.д. Создавать сложные пароли для каждого такого приложения обычный пользователь, естественно, не будет. В результате он старается использовать один пароль для всех приложений, что резко увеличивает вероятность его компрометации из-за возможных уязвимостей в организации хранения и передачи пароля этими приложениями.

Выход из этого положения известен уже довольно давно – единое хранилище учётных данных пользователя. Обычно для этого используется каталог LDAP, обеспечивающий хранение и изменение данных произвольной структуры. Наиболее известная реализация службы каталогов – открытый сервер OpenLDAP. В Windows-системах широко используется служба Active Directory, также поддерживающая протокол LDAP.

Многие современные СУБД, такие, как Oracle, Microsoft SQL Server, PostgreSQL, уже поддерживают возможность хранения учётных данных в LDAP. В других СУБД – например, в Firebird, – такая возможность отсутствует, что несколько ограничивает возможности их использования в крупных предприятиях и организациях с большим количеством пользователей и жёсткими требованиями к хранению их учётных данных.

Поэтому был разработан метод аутентификации пользователей сервера баз данных с использованием службы каталогов через протокол LDAP. Этот метод был реализован в сервере с открытым исходным кодом «Ред База Данных 2.5» [1,2], который основан на ядре популярной реляционной СУБД Firebird 2.5.

Традиционно как Firebird, так и Ред База Данных хранят учётные данные пользователей в БД безопасности – security2.fdb. Разработанный метод позволяет добавить дополнительный источник учётной информации – службу каталогов на основе сервера OpenLDAP. При этом LDAP используется именно как дополнение к традиционной схеме безопасности сервера. При проверке пользовательских учётных данных на сервере Ред Базы сначала выполняется попытка аутентификации классическим способом – в БД безопасности. Если она не удаётся, а в настройках сервера заданы па-

раметры подключения к каталогу LDAP, поиск выполняется и в нём тоже. С точки зрения конечного пользователя всё это работает совершенно прозрачно и ему не нужно выполнять никаких дополнительных действий.

Для аутентификации по протоколу LDAP в настройках сервера – файле firebird.conf должен быть задан ряд параметров.

Адрес сервера LDAP (IP-адрес или символьное имя) указывается в параметре «LDAPServer».

Тип шифрования, используемый при подключении к LDAP, задаётся параметром «LDAPEncryption». Он может принимать три значения: None – шифрование отсутствует; SSL – подключение по протоколу LDAPS; TLS – подключение с командой START_TLS. Последние два параметра обеспечивают шифрование передаваемых данных и поэтому рекомендуются к использованию, но при этом сервер LDAP, к которому выполняется подключение, должен быть настроен соответствующим образом.

Если в конфигурации задан параметр «VerifyCertChain» (по умолчанию), то при SSL/TLS-соединениях будет выполняться верификация сертификата LDAP-сервера со стороны сервера Ред Базы. Если сертификат не проходит проверку, соединение разрывается. Если параметр «VerifyCertChain» отключен, проверка сертификата LDAP-сервера не выполняется.

Имя пользователя, которое будет использоваться для подключения к серверу LDAP, указывается в параметре «LDAPUserDN» в виде DN, например:

```
LDAPUserDN = uid=rdb,ou=people,dc=example,dc=com
```

Пароль этого пользователя указывается в параметре «LDAPPassword».

Ветвь в каталоге, относительно которой будут искаться пользователи, указывается в параметре «LDAPUserBase» в виде DN. Поиск пользователей по данной базе выполняется рекурсивно по всем вложенным веткам, т.е. внутри этой ветви можно создавать другие ветви с учётной информацией пользователей – они будут найдены.

Ветвь в каталоге, относительно которой будут искаться группы, указывается в параметре «LDAPGroupBase» в виде DN.

Для определения списка групп, к которым принадлежит пользователь, в LDAP могут использоваться различные схемы. Указать конкретную схему можно в параметре «LDAPMembershipFilter». В

нём в качестве имени предполагаемого пользователя указывается шаблон %u. Две основные схемы определения групп выглядят следующим образом:

```
LDAPMembershipFilter = memberUid=%u
LDAPMembershipFilter = member=uid=%u,ou=people,dc=example,dc=com
```

Группы пользователя на сервере LDAP определяются для того, чтобы назначить ему соответствующие роли при подключении к базе данных. Если пользователь на сервере LDAP включён в определённую группу, и в базе данных, к которой он подключается, есть одноимённая роль, то эта роль ему назначается при подключении.

Также предусмотрена аутентификация пользователя через LDAP по сертификату.

Пользователь при подключении к серверу Ред Базы вместо логина и пароля может передать свой сертификат. Сервер выполняет процедуру верификации цепочки сертификации полученного сертификата (может быть отключена параметром конфигурации «VerifyCertChain»). Проверка сертификата считается проваленной в следующих случаях:

– не удалось построить цепочку сертификации;

– любой сертификат из цепочки отозван;

– любой сертификат из цепочки просрочен;

– любой сертификат из цепочки не прошёл проверку подписи;

– любой сертификат из цепочки используется не по назначению;

– цепочка основана на недоверенном корневом центре сертификации;

– цепочка сертификации содержит цикл;

– цепочка сертификации построена не полностью;

– не удалось проверить статус отзыва для любого сертификата из цепочки.

Если сертификат прошёл проверку, из него извлекается имя владельца, которое считается именем пользователя на сервере Ред Базы. Далее на основе открытого ключа, извлечённого из сертификата, создаётся сессионный ключ шифрования, обеспечивая, таким образом, безопасное сетевое соединение между клиентом и сервером.

В данной процедуре аутентификации по сертификату служба каталогов LDAP может использоваться двумя способами.

Во-первых, в LDAP может быть создан пользователь с именем владельца сертификата. Ранее если пользователь-владелец сертификата не был найден в БД безопасности, аутентификация считалась проваленной, даже если сертификат проходил верификацию. Теперь такой пользователь также будет искаться в службе каталогов.

Во-вторых, можно задать параметр конфигурации «LDAPUserCertificate», содержащий название атрибута, в котором сертификат пользователя будет храниться на сервере LDAP. При этом сервер сверяет сертификат, предъявленный пользователем с его сертификатом, сохранённым в LDAP. Если они не совпадают, проверка сертификата считается проваленной. Эта возможность может быть использована, например, для отключения процедуры верификации сертификата, если с ней возникают сложности программного или организационного характера. Чтобы пользователи всё же могли использовать сертификаты, а сервер мог проверять их достоверность, сертификаты можно сохранить в учётных записях пользователей в каталоге LDAP, а в конфигурации сервера включить параметр «LDAPUserCertificate» и отключить опцию «VerifyCertChain».

Пароль пользователя в LDAP может меняться / задаваться с использованием утилиты Ред Базы GSEC или её механизма сервисов. Для этого используется параметр конфигурации «LDAPPasswordSync». В нём через «;» указываются пароли, которые необходимо сменить (по умолчанию – все возможные). При изменении пароля указанный пользователь сначала ищется в security2.fdb и, если он там найден, его пароль меняется. Затем пользователь ищется в LDAP (если задан адрес LDAP-сервера). В LDAP могут меняться следующие атрибуты:

– userPassword – пароль пользователя в Linux, используется также большинством приложений, поддерживающих LDAP-аутентификацию. Сюда записывается хэш SHA1 в кодировке BASE64.

– sambaLMPassword – пароль для Samba-протокола (общие сетевые папки).

– sambaNTPassword – усовершенствованная версия пароля для Samba (использован более совершенный алгоритм хеширования).

– rdbPassword – пароль пользователя на сервере Ред Базы, зашифрованный алгоритмом NATIVE, используемым при традиционной аутентификации.

– rdbSecurePassword – более защищённый пароль пользователя на сервере Ред Базы, зашифрованный каким-либо алгоритмом из криптоплагина (средство использования сторонних криптографических провайдеров).

Алгоритм шифрования пароля NATIVE, традиционный для Ред Базы состоит из следующих действий:

– Пароль пользователя шифруется UnixCrypt, где шифруемой фразой является «9z», а ключом шифрования – пароль пользователя. Из полученного результата удаляются 2 первых символа «9z».

– Генерируется «соль» – 12 случайных символов в кодировке BASE64.

– К строке соли присоединяются имя пользователя и пароль, зашифрованный UnixCrypt.

– Считается хеш SHA1 полученной строки.

– К соли присоединяется захешированная строка в виде BASE64. Полученная в результате строка – зашифрованный пароль пользователя.

Соль здесь, как и вообще в криптографии, используется для того, чтобы усложнить анализ хеша с использованием радужных таблиц (наборы готовых хешей и соответствующих им паролей) и скрыть одинаковые пароли пользователей. Соль, как видно из алгоритма, хранится вместе с зашифрованным паролем и, как и сам этот пароль, является открытой информацией.

Кроме описанного алгоритма NATIVE был также разработан более криптостойкий алгоритм шифрования паролей пользователей:

– Генерируется соль – 12 случайных символов в кодировке BASE64.

– К строке соли присоединяются имя пользователя и пароль.

– Считается хеш полученной строки, используя один из алгоритмов из криптоплагина Ред Базы.

– Вычисленный хеш повторно хешируется 200000 раз. Это значение выбрано исходя из времени работы алгоритма. Сейчас на типичной пользовательской ЭВМ алгоритм работает около секунды, что не доставляет пользователю неудобства.

– К строке соли присоединяется полученная хешированная строка в виде BASE64. Полученный результат – зашифрованный пароль пользователя.

Преимущества данного алгоритма – произвольный алгоритм хеширования и множество итераций его применения. Рекомендуется использовать алгоритм ГОСТ Р 34.11-94, который имеет отличную криптостойкость и является обязательным для применения в качестве алгоритма хеширования в государственных организациях РФ и ряде коммерческих организаций. Многократное хеширование применяется для того, чтобы усложнить атаку пароля методом прямого перебора. Даже при однократном хешировании используя известную уязвимость алгоритма ГОСТ Р 34.11-94 количество операций для подбора пароля составляет 2^{105}, что на данный момент практически не реализуемо. Множество операций хэширования ещё более увеличивает вычислительную сложность подбора пароля.

Таким образом, новый алгоритм является более криптостойким, но с другой стороны алгоритм NATIVE обеспечивает совместимость со старыми клиентскими библиотеками Firebird / Ред Базы, т.к. он же используется при хранении паролей в БД безопасности. Кроме того, алгоритм NATIVE не требует установки и настройки криптопровайдера для использования криптоплагина Ред Базы.

Разработанный и реализованный метод позволяет серверу баз данных использовать для хранения учётных данных пользователя общую службу каталогов LDAP. При этом допускается использование как парольной аутентификации, так и аутентификации по сертификату пользователя. Группы пользователя из LDAP трактуются как его роли в базе данных, что позволяет гибко настраивать доступ к данным в базе на основе данных из LDAP. Использование традиционного для сервера БД алгоритма шифрования пароля позволяет пользователю работать «прозрачно», а применение нового алгоритма – создавать более криптостойкие хеши на основе алгоритма хеширования из ГОСТ.

Литература

1. *Симаков, Р.А.* Использование возможностей СУБД «Ред База Данных» 2.1 для защиты информации в банковской сфере // Р.А. Симаков, Д.Н. Стародубов // Информационная безопасность регионов. – 2010. – № 1. С. 71-75.

2. *Симаков, Р.А.* Аудит изменений в СУБД «Ред База Данных» / Р.А. Симаков, Д.Н. Стародубов // Информационная безопасность регионов. – 2008. – № 2(3). – С. 73 – 79.