УЯЗВИМОСТИ В БАЗАХ ДАННЫХ

РАДОВ Родион

Технический Университет Молдовы

***Аннотация**: Статья посвящена анализу основных уязвимостей и слабых мест в БД. Проанализированы самые часто-допускаемые ошибки и возможные уязвимости. Показано как избежать и как не допустить проникновения злоумышленников к личным или к корпоративным данным, как лучше всего обезопасить систему баз данных от взлома и нападения.*

***Ключевые слова:** СУБД, SQL, аутентификация, SQL-инъекция, безопасность.*

1. Введение

Практически ни одна современная компания не может обойтись без использования баз данных. В самом простом случае для хранения небольших объемов данных в качестве системы управления базами данных (СУБД) может использоваться система Microsoft Access. Однако, если необходимо обеспечить доступ к большим объемам данных (сотни мегабайт) сразу нескольких пользователей с различными правами доступа, локальные базы данных (Microsoft Access, Paradox и т.д.) уже не могут помочь. В этом случае необходимы многопользовательские СУБД, спроектированные с учетом архитектуры «клиент/сервер». В таких СУБД обеспечение информационной безопасности, как серверной, так и клиентских частей, приобретает первостепенное значение. В данной статье я не стану описывать механизмы защиты, используемые в различных СУБД (Oracle, Microsoft SQL Server, Sybase и т.д.). Этому посвящено немало статей и книг. Рассмотрим подробнее характерные уязвимые места систем управления базами данных и механизмы поиска таких «уязвимостей».

Однако предварительно необходимо заметить, что защита СУБД не ограничивается только внутренними защитными механизмами самой СУБД. Необходимо также защитить и операционную систему, под управлением которой работает СУБД. Таким образом, защита информации в системах управления базами данных представляет собой непростой процесс, заключающийся не только в приобретении и настройке специализированных средств защиты, но и в периодической проверке этих настроек: неправильная их конфигурация может свести на нет всю эффективность принятых защитных мер.

2. Возможные точки нарушения безопасности

В дальнейшем узкие места СУБД будут показаны на примере Microsoft SQL Server ранних версий, но это не означает, что другие СУБД лишены недостатков в системе защиты. Все уязвимые точки в любом программном обеспечении, в зависимости от источника их появления, можно условно разделить на три класса: проблемы проектирования, реализации и конфигурации. Рассмотрим эти классы применительно к различным защитным механизмам, реализуемым в СУБД: подсистема контроля целостности, подсистема разграничения доступа, подсистема аутентификации и т.д.

3. «Лазейки» системы идентификации/аутентификации

Microsoft SQL Server не обеспечивает возможности блокировки учетной записи пользователя базы данных в случае серии неудачных попыток аутентификации. Это позволяет злоумышленнику осуществлять различные атаки на систему идентификации-аутентификации, например, пытаться подобрать имена пользователей, зарегистрированных в СУБД, и их пароли.

Второй пример — невозможность переименования учетной записи системного администратора базы данных (sa), что также позволяет злоумышленнику осуществлять попытки подбора пароля администратора СУБД. Эта точка потенциальной опасности присуща не только различным базам данных, но и операционным системам (ОС), и прикладному программному обеспечению.

Другим недостатком практически всех СУБД является отсутствие проверки эффективности выбранного пользователем пароля. Зачастую в этом качестве выступает имя пользователя (идентификатор), знаменательная дата, номер паспорта или телефона и иные легко угадываемые комбинации. Нередко у пользователя совсем отсутствует пароль. К чему это может привести, говорить не надо. Также пользователи могут месяцами не применять базу данных, но, будучи один раз в нее внесенными, они считаются ее полноправными пользователями. В СУБД и во многих ОС отсутствуют механизмы контроля учетных записей, не использованных в течение заданного промежутка времени.

Наличие описанных возможностей приводит к тому, что используемая в организации СУБД становится восприимчивой к атакам типа «подбор пароля» (brute force), которые в случае успеха могут привести к компрометации всей информации, хранимой в базе данных.

4. Потенциальные опасности, приводящие к компрометации всей сети

Существует ряд уязвимостей в СУБД, которые могут привести не только к компрометации информации в базах данных, но и к компрометации всей сети в целом. Эти уязвимости появляются вследствие расширения стандартных возможностей SQL-серверов. Например, использование расширенной хранимой процедуры (extended stored procedure) xp_cmdshell позволяет выполнять функции операционной системы из командной строки так, как будто удаленный пользователь СУБД работает за консолью сервера баз данных. При этом функции, вызываемые при помощи процедуры xp_cmdshell, выполняются с привилегиями той учетной записи, под управлением которой загружен SQL-сервер. По умолчанию это учетная запись System. С помощью следующих SQL-команд злоумышленник, получивший доступ к СУБД, сможет создать пользователя с заданным паролем и правами администратора:

```
xp_cmdshell 'NET USER Lexa password /ADD'
go
xp_cmdshell 'NET LOCALGROUP /ADD Администраторы Lexa'
go
```

С помощью первой команды злоумышленник создает на компьютере, на котором запущен SQL-сервер, пользователя с именем Lexa и паролем password. А с помощью третьей команды пользователь с именем Lexa заносится в группу Администраторы. Таким образом, пользователь Lexa становится обладателем максимальных прав на сервере баз данных, а так как сервер баз данных часто запускается на контроллере домена, пользователь под именем Lexa автоматически получает доступ ко всем компьютерам сети предприятия.

При помощи расширенных хранимых процедур злоумышленник может получить доступ к информации подсистемы защиты информации Windows, например, к паролям, которые хранятся в системном реестре. Осуществляется эта возможность при помощи таких процедур, как xp_regdeletevalue, xp_regwrite, xp_regread, и т.д. Например, следующей командой злоумышленник может получить доступ к преобразованным паролям пользователей для дальнейшего их изучения:

```
xp_regread 'HKEY_LOCAL_MACHINE',
   'SECURITY\SAM\DOMAINS\ACCOUNT\USERS\000001F4','F'
```

Таким образом, вся мощь систем управления базами данных встает на сторону пользователя, а он может применить ее не только во благо, но и во вред.

5. Программы типа «троянский конь»

Программы типа «троянский конь» могут быть легко созданы путем модификации системных хранимых процедур. Например, несанкционированный доступ к паролю пользователя может быть получен при его смене с помощью всего одной строчки кода.

```
/* процедура sp_password добавляет или изменяет пароль пользователя SQL-сервера */
create procedure sp_password
@oldvarchar(30) = NULL, /* старый (текущий) пароль */
@new varchar(30, /* новый пароль */
@loginame varchar(30) = NULL, /* флаг, разрешающий или запрещающий
пользователям менять пароль */
declare @suid int /* идентификатор пользователя, изменяющего пароль */
/* следующая строка является "троянским" включением, позволяющим
несанкционированно узнать пароль пользователя при его смене */
insert into spt_values values (@new,-1,'A',NULL,NULL,0)
```

Таким образом, при изменении пароля, который обычно хранится в зашифрованном виде в таблице master.dbo.syslogins, указанный «троянский конь» позволит увидеть пароль пользователя в открытом виде, сохраненный в таблице spt_values.

6. Источник опасности на этапе подключения к СУБД

Для подключения к Microsoft SQL Server можно использовать различные сетевые протоколы: TCP/IP, Named Pipes, Multi-Protocol. В первом случае, имя и пароль пользователя, проходящего аутентификацию на сервере базы данных, передаются в открытом виде и могут быть перехвачены при помощи простейшего анализатора протокола (sniffer). Кроме того, использование протоколов TCP/IP позволяет обойти процедуру аутентификации Windows NT. Аналогичным образом можно поступить и при использовании протокола Named Pipes, в котором пароль посылается в формате UUENCODE, что также не является преградой для злоумышленников. В случае использования Multi-Protocol передаваемая информация может шифроваться или не шифроваться.

В качестве другого примера можно назвать использование расширенных хранимых процедур (extended stored procedures), которые позволяют контролировать другие приложения Windows, например Web-сервер Internet Information Server. Осуществляется такой доступ при помощи хранимых процедур sp_OACreate, sp_OAGetProperty, sp_OAMethod и т.д. С их помощью из SQL-сервера возможно изменение конфигурации IIS. Это лишний раз подтверждает тот факт, что получение доступа к SQL-серверу и его компрометация позволяют очень серьезно нарушить линию обороны предприятия.

7. Анализ защищенности СУБД

Как было отмечено выше, для устранения описанных и других уязвимых мест разработано множество механизмов, которые реализуются или в самих системах управления базами данных или при помощи сторонних средств. Однако их разрабатывают люди, которые сами иногда делают ошибки. Кроме того, защищенность СУБД зависит не только от правильной реализации того или иного механизма. Немаловажное значение имеет правильная настройка параметров СУБД и ее окружения, влияющих на их защищенность. Однако во многих организациях (особенно в России) нередка ситуация, когда администратор базы данных одновременно является и администратором сети, и администратором безопасности и т.д., хотя это недопустимо. Администратор сбивается с ног, занимаясь администрированием различных систем. Он делает ошибки и не замечает этого, а если замечает, то откладывает их решение на потом и вскоре напрочь забывает об этом. В территориально распределенных организациях, в которых сеть построена по филиальной схеме, ситуация усугубляется еще и тем, что в филиалах, как правило, отсутствует квалифицированный персонал, осуществляющий администрирование баз данных.

Понимая сложившуюся ситуацию, разработчики предложили средства, автоматизирующие анализ конфигурации систем управления базами данных и уведомляющие администратора об этом в случае обнаружения отклонений от положений принятой политики безопасности. Такие средства называются сканерами безопасности (Security Scanner) или средствами анализа защищенности (Security Assessment System). Первоначально эти средства разрабатывались для сетевого и системного программного обеспечения. Известно более пятидесяти таких средств. Связано это в первую очередь с небольшим числом сетевых протоколов и операционных систем, что позволяет эффективно анализировать потенциально опасные точки. Необходимость в средствах анализа защищенности СУБД стала реальной, когда базы данных стали использоваться для получения информации через открытые сети передачи данных. Существует ряд средств, предназначенных для анализа защищенности СУБД:

- Database Scanner компании Internet Security Systems (ISS);
- SQL Secure Policy компании BrainTree Security Software.

Система Database Scanner была разработана компанией DB Secure и первоначально называлась SQL Auditor. Система была переименована в Database Scanner после приобретения компании DBSecure компанией Internet Security Systems. Указанное средство пополнило семейство продуктов адаптивного управления безопасностью SAFE suite компании ISS, которое состоит из трех систем анализа защищенности, функционирующих на всех четырех уровнях информационной инфраструктуры предприятия: уровне сети (Internet Scanner), уровне операционной системы (System Scanner), уровне СУБД (Database Scanner) и уровне прикладного программного обеспечения (Internet Scanner и System Scanner). Общее число уязвимостей, которые обнаруживаются средствами компании ISS, превышает 1600, что на порядок превышает аналогичное число у ближайших конкурентов. Система Database Scanner обеспечивает детальный анализ следующих областей: аутентификация, авторизация и целостность. Анализ осуществляется на основе громадного практического опыта, накопленного специалистами в области защиты баз данных и сконцентрированного в базе знаний Security Knowledge

Wizard. По результатам анализа создается отчет, содержащий подробное описание каждой обнаруженной уязвимости и пошаговое описание мер, позволяющих устранить риск использования этих возможностей злоумышленниками. Анализ настроек СУБД осуществляется в соответствии с шаблонами, учитывающими различные требования по безопасности баз данных (начиная от минимальных и заканчивая «параноидальными»). На основе имеющихся шаблонов администратор СУБД может создавать свои собственные шаблоны, учитывающие специфику применения СУБД в своей организации. Процесс анализа защищенности осуществляется дистанционно, что облегчает труд администраторов в территориально-распределенных сетях. Система Database Scanner поддерживает следующие СУБД Microsoft SQL Server, а также Sybase Adaptive Server. Планируется осуществить поддержку СУБД Oracle. Система Database Scanner обнаруживает около 200 «брешей» в защите названных СУБД, при этом все проводимые проверки можно условно разделить на 17 категорий, в том числе:

- проверки подсистемы аудита СУБД;
- проверки подсистемы резервного копирования (backup);
- проверки прав доступа пользователей к объектам СУБД;
- проверки возможности осуществления различных атак.

8. Заключение

В данной статье были рассмотрены основные уязвимости и “лазейки” в СУБД. Были рассмотрены часто допустимые ошибки системы аутентификации, ввода паролей и учетных записей. Так же были рассмотрены примеры и фрагменты кода при демонстрации примера взлома БД при помощи SQL - инъекций. Были приведены примеры перехвата учетных данных при помощи “снифера” и других уязвимостей протокола TCP/IP.

Литература

1. Методы и средства взлома баз данных MSSQL. [Электронный ресурс]. - Режим доступа: http://www.securitylab.ru/blog/personal/aguryanov/29918.php
2. Методы и средства взлома баз данных MySQL. [Электронный ресурс]. - Режим доступа: https://xakep.ru/2015/04/08/195-mssql-attacks/
3. Методы и средства взлома баз данных. [Электронный ресурс]. - Режим доступа: https://xakep.ru/2015/04/16/195-mysql/