D-тип: по требованиям к используемому предположению о трудно решаемой задаче (decisional Diffie-Hellman, etc);

E-тип: по требованиям к модели безопасности (standard model, random oracle model) и другие.

В скобках указаны в качестве примера возможные значения параметра классификации.

Термины из работ по схемам прокси-перешифрования могли бы пополнить список терминов следующей редакции Словаря криптографических терминов [4].

СПИСОК ЛИТЕРАТУРЫ:

1. *Blaze M., Bleumer G., Strauss M.* Divertible protocols and atomic proxy cryptography // EUROCRYPT. Lecture Notes in Computer Science. Vol. 1403. Berlin: Springer-Verlag, 1998. P. 127–144.
2. *Libert B., Vergnaud D.* Unidirectional chosen-ciphertext secure proxy re-encryption // Public Key Cryptography. Lecture Notes in Computer Science. Vol. 4939. Berlin: Springer-Verlag, 2008. P. 360–379.
3. *Matsuda* T., *Nishimaki R., Tanaka K.* CCA Proxy Re-Encryption without Bilinear Maps in the Standard Model // PKC 2010. Lecture Notes in Computer Science. 2010.Vol. 6056/2010. P. 261–278.
4. Словарь криптографических терминов / Под ред. Б. А. Погорелова и В. Н. Сачкова. М.: МЦНМО, 2006. – 94 с.

Е. И. Гончаров

ПРОБЛЕМА БЕЗОПАСНОСТИ БАЗ ДАННЫХ В ИНФОРМАЦИОННЫХ СИСТЕМАХ С ТРЕХЗВЕННОЙ АРХИТЕКТУРОЙ

В последнее время для построения информационных систем, в том числе благодаря бурному развитию Интернета, все чаще используют трехзвенную архитектуру. Она удобна как для публичных сервисов (информационные порталы, платежные сервисы и т. д.), так и для применения в закрытых ИС. В связи с распространением данной архитектуры проблемы ее безопасности становятся еще острее.

Трехзвенная архитектура состоит из:

1. базы данных (БД), в которой хранятся обрабатываемые данные и которая недоступна для пользователей системы напрямую,

2. сервера приложений — некоторого ПО, обрабатывающего запросы пользователей и взаимодействующего с базой данных,

3. клиентского приложения — некоторого ПО для взаимодействия с сервером приложений.

Безопасность подобных систем, как правило, обеспечивается следующим образом:

1. ограничение сетевого доступа к БД и анализ запросов с помощью межсетевых экранов и систем обнаружения вторжений таким образом, что только сервер приложений имеет прямой доступ к БД;

2. анализ запросов к серверу приложений посредством систем обнаружений вторжений;

3. разграничение доступа сервера приложений к базе данных путем использования учетных записей с минимальными правами;

4. ограничение на уровне СУБД прямого доступа к таблицам с помощью хранимых процедур и других технологий [1].

Таким образом, исходя из стандартных методов защиты первым рубежом защиты от несанкционированного доступа к БД является сервер приложений, вторым — СУБД. Однако

если учесть, что при обработке клиентских запросов сервер приложений использует одну и ту же (или несколько) учетную запись БД, то получается, что остается только единственный рубеж защиты — сервер приложений. Это легко увидеть на примере описанного ниже сценария атаки. Для определенности возьмем распространенную трехзвенную архитектуру, где клиентское приложение — браузер, сервер приложений — веб-сервер, а в БД для доступа к данным используются, например, хранимые процедуры.

Возможный сценарий атаки состоит из следующих шагов:

1. Злоумышленник отправляет некоторый некорректный запрос на сервер приложений или использует иную его уязвимость, в результате чего реализуется угроза выполнения произвольного программного кода в контексте сервера приложений.

2. В зависимости от использованной уязвимости и среды выполнения сервера приложений для доступа к БД злоумышленник может воспользоваться подключениями самого сервера приложений к БД или, прочитав учетные данные для подключения к БД, подключиться с сервера приложений самостоятельно.

3. Получив тем или иным способом подключение к базе данных, злоумышленник изучает доступные ему хранимые процедуры, находит те, что отвечают за получение интересующей информации, и посредством их последовательного вызова получает всю информацию из защищаемых таблиц. Стоит учесть вариант, когда исполняемый код сервера приложений был написан на интерпретируемом языке программирования (например, PHP). В этом случае анализ кода может значительно упростить задачу на этом шаге.

Таким образом, в случае взлома сервера приложений злоумышленник получает доступ ко всем данным под учетной записью сервера приложений — как правило, это вся база. Ограничения доступа и защитные механизмы на уровне БД в этом случае выполняют больше функции «затягивания» времени и минимизации последствий атаки, нежели представляют собой препятствие для злоумышленника. Кроме того, из-за наличия небольшого количества применяемых СУБД (самые популярные — Oracle Database, Microsoft SQL Server, MySQL) большинство их уязвимостей заранее известны злоумышленнику. Потому второй и третий шаги из приведенного сценария атаки могут быть сведены к использованию известной уязвимости. Следовательно, при использовании стандартных механизмов защиты и разграничения доступа после взлома сервера приложений между злоумышленником и БД уже не остается серьезных преград.

В качестве простейшего решения данной проблемы безопасности можно было бы предложить использование хранимых процедур совместно с сессионными ключами, но, несмотря на кажущуюся простоту, данный подход требует серьезной модификации существующей БД, сложен для реализации в крупных ИС, не защищает от уязвимостей СУБД и, более того, почти бесполезен в ИС с равноправными пользователями.

В связи с этим логично предположить необходимость некоторого средства защиты, размещенного между сервером приложений и БД, способного анализировать запросы к БД и результат их выполнения. Средство должно удовлетворять следующим базовым требованиям:

1. быть простым и компактным как на уровне исходного кода, так и на уровне настроек для облегчения доказательства его корректного функционирования,

2. иметь унифицированные интерфейсы для подключения как к базе данных, так и к серверу приложений для упрощения интеграции в существующие ИС.

Для повышения уровня защищенности базы данных средство должно выполнять следующие функции:

1. анализировать запросы к БД по таким параметрам, как запрашиваемые поля и таблицы, операции над данными и т. д.,

2. проверять корректность переходов от одного запроса к другому согласно определенным правилам,

3. проверять соответствие результата выполнения запросов определенным шаблонам,

4. вести статистику и аудит запросов, выявлять аномалии.

Программное средство с описанным функционалом, отвечающее указанным требованиям, разрабатывается для защиты базы персональных данных веб-портала одного из поставщиков государственных услуг г. Москвы.

СПИСОК ЛИТЕРАТУРЫ:

1. *Смирнов С. Н.* Безопасность систем баз данных. Гелиос АРВ, 2007.

Н. Е. Гунько

ИСПОЛЬЗОВАНИЕ ПРИЗНАКОВ ПОЧЕРКА ДЛЯ СИСТЕМ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

Системы доступа и защиты информации, основанные на биометрических технологиях, являются не только достаточно надежными (биометрические данные невозможно передать другому лицу), но и очень удобными для пользователей на сегодняшний день, поэтому они приобретают все большую популярность как среди юридических лиц (фирмы, компании, организации), так и среди частных пользователей [1].

Исследованием почерка и его признаков активно занимаются в таких областях, как криминалистика, медицина, психология, археология, а также информационная безопасность.

На данный момент такой объект, как почерк, применяется в системах защиты информации для идентификации пользователя. Используется его роспись (иногда написание кодового слова).

Задача нашего исследования пойти дальше, чем известная идентификация [2, 3] пользователя по его почерку, и разработать (как и в [2, 3]) *правило принятия решения* (ППР) о психологических характеристиках человека, в нашем случае потенциального злоумышленника, для систем защиты информации.

Связь между признаками почерка и психологическими особенностями личности изучается в графологии и отчасти в психологии. Однако следует отметить, что в криминалистике почерк изучен достаточно хорошо и уже получен ряд результатов, связанных с исследованием почерка и психологических особенностей личности.

Детальное изучение почерка основывается на целой системе информативных признаков [4], например следующих: признаки формы; признаки размера; признак положения букв; признак направления, наклон почерка; промежутки между элементами, буквами и словами; величина оставляемых полей; вычисление повторяемости выделяющихся признаков; подпись человека как самостоятельный признак.

Исследование особенностей наиболее информативных признаков почерка невозможно без изучения индивидуально-психологических особенностей личности, поскольку каждый графологический признак содержит ценную информацию о тех или иных свойствах характера пишущего, особенностях темперамента, способности к креативному мышлению и т. д.