г) провести оценку качества обучения нейронной сети при использовании векторов показателей $\{Index_{i_1}, \ldots, Index_{i_k}\}_k$ для каждого класса атак $AttackClass_k$ на отдельной нейронной сети;
д) если качество тестов ухудшилось, то вернуть множество классов атак, полученное на предыдущем шаге (при $K-1$);
е) если качество тестов улучшилось и ошибки лежат в установленных пределах, то вернуть текущее множество классов атак;
ж) перейти к шагу «в».

В результате использования методики получено множество классов атак и соответствующее каждой атаке множество показателей для работы нейронной сети. Каждый класс атак обрабатывается отдельной нейронной сетью.

После формирования классов атак необходимо с помощью топологических тестов нейронной сети убедиться, что для различных атакующих воздействий внутри одного класса атак оптимальными являются схожие параметры архитектуры нейронной сети. В противном случае атакующие воздействия с отличными оптимальными параметрами архитектуры нейронной сети выделяются в отдельный класс атак.

ЛИТЕРАТУРА

1. *Жульков Е. В.* Построение нейронных сетей для обнаружения классов сетевых атак: дис. ... канд. техн. наук. СПб., 2007. 155 с.
2. *Александров И. С.* Разработка системы защиты web-приложений от автоматизированного копирования информации: дис. ... канд. техн. наук. М., 2003. 127 с.
3. *Хафизов А. Ф.* Нейросетевая система обнаружения атак на www-сервер: дис. ... канд. техн. наук. Уфа, 2004. 172 с.
4. *Сорокин С. Н.* Метод обнаружения атак типа «отказ в обслуживании» на web-приложения // Прикладная дискретная математика. 2014. № 1(23). С. 55–64.
5. *Menasce D. A. and Almeida V. A. F.* Capacity Planning for Web Services. Metrics, Models, and Methods. New Jersey: Prentice Hall PTR, 2001. 608 p.
6. *Хайкин С.* Нейронные сети: полный курс. 2-е изд., испр. М.: ООО «И.Д. Вильямс», 2006. 1104 с.

УДК 004.94

РЕАЛИЗАЦИЯ МОНИТОРА БЕЗОПАСНОСТИ СУБД MySQL В DBF/DAM-СИСТЕМАХ

Н. О. Ткаченко

Предлагается прототип механизма, реализующего политику мандатного управления доступом типа multilevel security (MLS) и type enforcement (TE) на основе разработанной ранее формальной ДП-модели, а также механизм сокрытия структуры базы данных на основе метода переписывания запросов. Прототип реализован в виде DBF/DAM-модуля — для MySQL-proxy, функционирующей между клиентом и сервером системы управления базами данных (СУБД) MySQL как прокси-сервер. При реализации модели безопасности предложен и использован подход, при котором функции в коде соответствуют де-юре правилам формальной ДП-модели.

Ключевые слова: *компьютерная безопасность, управление доступом, реализация моделей безопасности, DBF/DAM-системы, СУБД MySQL, MySQL-proxy.*

В настоящее время активно развивается подход на основе DBF/DAM-технологий, заключающийся в реализации специализированного прокси-сервера, обепечивающего базовое управление доступом, защиту от основных атак и мониторинг СУБД. Такие системы получили название *Database Firewall* (DBF) или *Database Activity Monitoring* (DAM) [1].

Все основные DBF/DAM-системы, к которым можно отнести, например, Oracle Database Firewall, GreenSQL, McAfee DAM, Imperva SecureSphere, ориентированы, как правило, на обнаружение подозрительной активности пользователя и предотвращение возможных атак. При этом механизмам управления доступом уделяется недостаточно внимания, так как предполагается, что они уже реализованы на уровне самой СУБД. В связи с этим реализация современных научно обоснованных механизмов управления доступом на урове DBF/DAM-систем, несомненно, является перспективным направлением и позволяет решить следующие проблемы реализации политик управления доступом в изначально дискреционных СУБД:

— необходимость изменять исходный код защищаемой СУБД;
— реализацию механизма управления доступом для всех СУБД, поддерживающих язык SQL;
— необходимость изменять существующую инфраструктуру СУБД;
— уменьшение «поверхности атак» на защищаемую СУБД.

Предлагается прототип механизма управления доступом, реализующий политики мандатного управления доступом типа MLS и TE на основе разработанной ранее формальной ДП-модели [2] в виде DBF/DAM-модуля — для MySQL-proxy, а также механизм сокрытия структуры БД на основе метода переписывания запросов. При реализации ДП-модели в коде используется подход, заключающийся в разделении кода на две части: функции управления доступом, соответствующие де-юре правилам преобразования ДП-модели и реализующие логику политик безопасности, и функции адаптации, необходимые для взаимодействия элементов СУБД с элементами механизмов управления доступом.

Основой прототипа является система MySQL-proxy [3]. Данная система функционирует между клиентом и сервером СУБД MySQL, предназначена для балансировки нагрузки, обработки запросов, проходящих как от клиента к серверу, так и от сервера к клиенту, реализует механизм аварийного переключения. Для обработки запросов MySQL-proxy использует встроенный язык Lua [4].

Реализован модуль на языке Lua для MySQL-proxy, выполняющий следующие функции:

1) присвоение сущностям (базам данных, таблицам и столбцам) меток безопасности, а также их хранение;
2) синтаксический анализ запроса с целью идентификации всех сущностей;
3) принятие решения о продолжении обработки запроса или его прекращении на основе меток безопасности и мандатной политики управления доступом;
4) сокрытие внутренней структуры БД.

ЛИТЕРАТУРА

1. Database Activity Monitoring / Database Firewall. `http://www.provision.ro/threat-management/database-security/database-activity-monitoring-database-firewall#pagei-1| pagep-1|`

2. *Колегов Д. Н., Ткаченко Н. О., Чернов Д. В.* Разработка и реализация мандатных механизмов управления доступом в СУБД MySQL //Прикладная дискретная математика. Приложение. 2013. № 6. С. 62–67.
3. *Hinz S., DuBois P., and Stephens J.* MySQL 5.7 Reference Manual. `http://dev.mysql.com/doc/refman/5.7/en/mysql-proxy.html`
4. *Ierusalimschy R., Henrique de Figueiredo L., and Celes W.* The Programming Language Lua. Lua 5.2 Reference Manual. `http://lua.org/manual/5.2/`

УДК 004.056.57

МЕТОД ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПОЛЬЗОВАТЕЛЕЙ ИНТЕРНЕТ-МАГАЗИНОВ МОБИЛЬНЫХ ПРИЛОЖЕНИЙ

Е. А. Толюпа

Предложен метод противодействия распространению вредоносного ПО через интернет-магазины Android-приложений. Метод основан на применении доверенных цифровых подписей (ДЦП) и алгоритма (n, t)-пороговой ДЦП с арбитром, который позволяет разработчику антивирусного ПО реализовать механизм распределения доверенности и права проверки приложений на наличие вредоносного кода между n центрами проверки таким образом, что ДЦП будет вычислена только с участием арбитра и только в том случае, если t $(t < n)$ центров проверки подтвердят отсутствие вирусов. Роль арбитра можно возложить на удостоверяющие центры. Таким образом, проверяющий доверяет разработчику антивирусного ПО и удостоверяющему центру (арбитру) и может не опасаться, что центр проверки окажется злоумышленником и подпишет небезопасное приложение. Предложенный метод могут использовать интернет-магазины для повышения уровня доверия к себе среди потенциальных клиентов.

Ключевые слова: *антивирусная защита, доверенные цифровые подписи, пороговые доверенные цифровые подписи, Android, магазин приложений Android.*

Бурное развитие платформы Android повлекло создание большого числа интернет-магазинов Android-приложений. На сегодняшний день популярностью пользуются Google Play, Яндекс.Store, Amazon mobile app distribution, Samsung Apps, SlideMe, GetJar. Пётр Меркулов, директор по развитию продуктов и услуг Лаборатории Касперского, утверждает: «Магазины приложений для Android — это лакомый кусочек для вирусописателей. ... 99 % всех обнаруженных в 2012 г. мобильных зловредов были нацелены на Android-устройства». В 2012 г. Google запустил сервис Bouncer, который представляет собой виртуальную машину, эмулирующую окружение Android и осуществляющую динамический анализ приложений. Магазин Яндекс.Store проверяет антивирусом своё приложение. Таким образом, пользователи вынуждены доверять процедуре проверки файла самого магазина. Пользователь может доверять крупным корпорациям, имеющим положительную репутацию.

Платформа Android является открытой, и любой может разрабатывать свои приложения и публиковать их в сети или создать свой магазин приложений. Пользователь вынужден доверять тому, кто публикует приложения в сети. Интернет-магазин может оказаться злоумышленником и публиковать приложения, утверждая, что они проверены антивирусом. Возможен вариант, когда интернет-магазин может отправить файл на проверку надёжному разработчику антивирусного ПО (АПО), чтобы тот, проверив файл, сформировал электронную подпись (ЭП). Наличие корректной ЭП для файла является гарантией отсутствия вредоносного ПО. В этом случае разработчик АПО