так как использование различных архитектур позволяет определить принадлежность множества классов без проведения оптимизации. В качестве первого критерия предлагается использовать время обработки потока данных соответствующего класса. При работе с набором классов время обработки суммарного потока данных будет не меньше максимального значения времени для отдельных классов. В наилучшем случае «влияние» классов данных из набора друг на друга будет минимальным. В процессе оптимизации распределения классов, соответствующего хранилищу, необходимо максимально уменьшить время обработки потока данных и разницу между существующим временем и «наилучшим». Далее, рассматривая систему в целом, необходимо распределить все классы данных по хранилищам с максимально возможной эффективностью каждого из них. Так, в качестве критерия J3 предлагается использовать наихудшее значение критериев J2, соответствующих хранилищам.

Таким образом, в работе предложена постановка задачи разбиения данных с использованием гибридной модели системы управления хранением данных. Разбиение осуществляется на основе предложенных свойств, характеристик данных. Для поиска оптимального разбиения введены критерии, позволяющие оценить эффективность работы отдельных хранилищ и всей системы в целом. Разделение данных также позволяет гибко управлять безопасностью хранилища, предоставляя раздельный доступ в зависимости от функциональности каждого модуля системы.

СПИСОК ЛИТЕРАТУРЫ:

1. *Дудаков Н. С., Пирогов Н. Е.* Разработка гибридной модели системы безопасного управления хранением данных // Безопасность информационных технологий. 2010. № 1. С. 68–69.
2. *Дудаков Н. С., Пирогов Н. Е., Шумилов Ю. Ю.* Кроссплатформенная система безопасного управления хранением динамических данных // Безопасность информационных технологий. 2009. № 3. С. 12–15.
3. Документация по базе данных PostgreSql. URL: http://www.postgresql.org/.

А. А. Дураковский, С. В. Дворянкин

ОБЕСПЕЧЕНИЕ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ В РАСПРЕДЕЛЕННЫХ СИСТЕМАХ ОБРАБОТКИ ДАННЫХ

Задачу анализа отказов аппаратно-программных средств (АПС) обеспечения информационной безопасности в распределенных системах обработки данных (РСОД), с учетом особенности применения распределенных систем защиты информации, можно решить в два этапа.

На первом этапе исследуются известные подходы к анализу отказов АПС, на основании которых разрабатывается математическая модель, позволяющая выяснить основные закономерности работы этой системы и определить основные требования к методике анализа отказов АПС. На втором этапе разрабатывается общий концептуальный подход к регистрации и анализу отказов, на основе которого и создается методика, направленная на своевременное выявление «слабых мест» территориально распределенных систем обработки данных большой протяженности и подготовку информации для принятия решений при планировании работ по восстановлению целостности информационной безопасности РСОД.

Алгоритм анализа отказов АПС обеспечения информационной безопасности РСОД с распределенной системой защиты информации есть не что иное, как последовательность правил,

рецептов, рекомендаций по выбору, анализу и разделению объектов, относящихся к АПС, на множества по единственному критерию «принадлежит — не принадлежит». Несмотря на кажущуюся простоту критерия, задача разработки алгоритма для таких систем достаточно трудоемкая и сложная. Для решения этой задачи разработана модель обеспечения безопасности эксплуатации АПС, с помощью которой алгоритм и разрабатывался эвристическими методами с выполнением рутинных работ, связанных с перебором вариантов, с использованием стандартного программного обеспечения.

С. Д. Жилкин

ВЫЯВЛЕНИЕ АНОМАЛИЙ РАБОТЫ ПО С ПОМОЩЬЮ МОДЕЛЕЙ ПОВЕДЕНИЯ

Предпосылкой к ведению работ в данном направлении является сложившаяся ситуация в сфере информационной безопасности. На данный момент существует множество программных средств, препятствующих выполнению вредоносного кода. Однако куда меньшее внимание уделено проблеме выявления и локализации недекларированных возможностей (НДВ) ПО, проявляющихся при различных предпосылках и вызывающих аномальное поведение. Так, некоторое ПО может начать вести себя необычным образом вследствие определенной последовательности действий. При этом программный код, провоцирующий такое поведение, сам по себе может не являться вредоносным. В данной работе решается задача выявления заранее неизвестных атак.

Одним из подходов выявления компьютерной атаки, не базирующихся на экспертных знаниях о том, как эта атака совершается, является мониторинг и анализ состояния узлов и серверов информационной системы с целью выявления отклонений от некоторого так называемого штатного, или эталонного, состояния [1, 2].

О поведении приложения можно судить по его взаимодействию с операционной системой: обращение к жесткому диску, сетевым ресурсам, вызовы функций драйверов, работа с реестром и прочее. Для того чтобы получать информацию такого рода о поведении ПО, требуется дополнительная система мониторинга. Она может основываться на журнальных файлах операционной системы (система мониторинга верхнего уровня) или, например, на более подробных данных, полученных на уровне драйверов ОС (системы мониторинга низкого уровня), как проиллюстрировано на рис. 1. При реализации методов моделирования ПО, приведенных в данной работе, для сбора информации о поведении ПО использовалась система «Инсайдер» компании «Праймтек». С помощью модулей конвертации можно осуществить поддержку других систем мониторинга. Таким образом, на основании данных от системы мониторинга возможно создание моделей поведения, описывающих штатную работу контролируемого ПО.

Так как при построении модели и анализе используется математический аппарат, необходимо описывать поведение ПО некоторым набором числовых значений — вектором координат. Предлагается описывать поведение процесса количественными, логическими и статистическими характеристиками. Количественные характеристики могут включать в себя такие параметры, как число обращений к системным файлам, библиотекам и прочим ресурсам системы. Логические характеристики описывают общее поведение ПО: создавались ли сетевые соединения, порождались ли дополнительные процессы и прочее. Статистические характеристики могут показывать частоту выделения дополнительной памяти, обращений к жесткому диску и другие частотные параметры. Чем более подробной будет информация о взаимодействии процесса с операционной системой,