зависимость: в области малых значений этих параметров более надежен алгоритм Куттера, а в области больших значений этих параметров — алгоритм Брайндокса.

Таким образом, сравнение показало, что алгоритм Брайндокса превосходит алгоритм Куттера по таким характеристикам, как временная сложность (как внедрения, так и извлечения), устойчивость, скрытность, а также надежность в области больших значений параметров.

СПИСОК ЛИТЕРАТУРЫ:

1. *Kutter M., Jordan F., Bossen F.* Digital signature of color images using amplitude modulation // Proc. of the SPIE Storage and Retrieval for Image and Video Databases V. 1997.
2. *Darmstaedter V., Delaigle J.-F., Quisquater J., Macq B.* Low cost spatial watermarking // Computers and Graphics. 1998. Vol. 5. P. 417–423.

М. Ю. Ваганов

РАЗРАБОТКА ИСКУССТВЕННОЙ ИММУННОЙ СИСТЕМЫ, ПРЕДНАЗНАЧЕННОЙ ДЛЯ ОБНАРУЖЕНИЯ ЗАРАЖЕНИЙ КОМПЬЮТЕРНОЙ СИСТЕМЫ

Искусственная иммунная система

Под искусственной иммунной системой принято понимать программные комплексы, принципы функционирования которых аналогичны иммунным системам живых организмов [1].

В данной работе предпринимается попытка создания программного комплекса, использующего принципы иммунной системы [2] для детектирования вредоносного кода. В качестве целевой платформой выбраны операционные системы семейства Windows, а именно: XP, 2000, Server 2003, Vista, Server 2008 и 7. В качестве критериев состояния системы выбраны статистические параметры, на которые в большинстве случаев оказывают влияние вредоносные программы. Все параметры разбиты на четыре группы:

- связанные с работой файловой системы (создание/модификация исполняемых файлов, создание файлов с нестандартными именами, создание файлов с расширениями, не соответствующими заголовку, и т. п.);
- связанные с реестром операционной системы (модификация и удаление ключей, в том числе ключей, гарантированно не связанных с наблюдаемым процессом);
- связанные с сетью;
- связанные с запуском и работой других процессов (запуск процессов с уровнем доступа, допускающим его остановку или запись в его адресное пространство, открытие потоков других процессов, использование хуков, выгрузка системных процессов).

Каждому параметру приписывается вес, характеризующий степень его опасности. Значения весовых коэффициентов определялись в процессе компьютерного эксперимента. Для инициализации системы выбирается гарантированно не зараженная система в виде вектора параметров. В процессе работы системы подсистема защиты набирает статистику параметров и периодически вычисляет отклонения состояния системы от здорового. При обнаружении значительных отклонений идентифицируется подозрительное приложение и переводится в карантин. В дальнейшем приложения, попавшие в карантин, проверяются антивирусными средствами.

Сбор данных об активности приложений

Одним из важных компонентов системы является WDM-драйвер, отвечающий за сбор различных проявлений активности процессов, а именно: за работу с файловой системой и устройствами, сетевую и межпроцессную активность. Он также отвечает за создание и обновление таблицы процессов, которая требуется для выявления скрытых и замаскированных процессов. Ввиду достаточно большого количества вариантов сокрытия процесса в данной работе одновременно используются несколько различных способов получения списка процессов (в частности, функции ToolHelp API, Native API, ZwQuerySystemInformation, анализ структуры EPROCESS). В целях повышения производительности и стабильности работы системы, а также упрощения отладки на стадии разработки обработка собранной информации осуществляется отдельными модулями, функционирующими в пользовательском режиме.

Все компоненты, предназначенные для работы в пользовательском режиме, реализованы на языке C# с использованием .NET Framework версии 4.0. Для написания драйвера использовалась комбинация языков C++ и Assembler. Передача информации между драйвером и ядром осуществляется с помощью IRP-пакетов [3, 4].

Выработка антител

После того как вредоносный код обнаружен и обезврежен, система сохраняет в базе данных соответствующий профиль нейронной сети, а также характеристики исполняемого файла (размер, тип, данные о таблице экспорта и т. п.), что позволяет увеличить скорость реакции системы при повторном заражении.

Полученные результаты

Тестирование работоспособности системы проведено на серии известных вредоносных программ (в частности, Trojan-Downloader.Win32.Agent.bet). Для сравнения использовались две идентично сконфигурированные виртуальные машины: VirtualBox, Windows XP SP2 с включенным стандартным сетевым экраном. Обеим машинам был обеспечен доступ в сеть Интернет. В процессе тестирования одна из машин все время оставалась «здоровой», другая «заражалась» через некоторое время после начала эксперимента. Число успешных индикаций зараженного состояния составляет не менее 73 %, что говорит о высокой эффективности используемой модели.

СПИСОК ЛИТЕРАТУРЫ:

1. *Дасгупта В.* Искусственные иммунные системы. М.: Физматлит, 2006. – 344 с.
2. *Newsome J., Karp B., Song D.* Polygraph: Automatically Generating Signatures for Polymorphic Worms. URL: http://www.ece.cmu.edu/~dawnsong/papers/polygraph.pdf (дата обращения: 01.02.2010).
3. *Они У.* Использование Microsoft Windows Driver Model. СПб.: Питер, 2007. С. 213–293.
4. *Руссинович М., Соломон Д.* Внутреннее устройство Microsoft Windows: Windows Server 2003, Windows XP, Windows 2000. СПб.: Питер, Русская редакция, 2008. С. 566–648.