Таблица 2 – Зависимость $\delta_п$ от $P_{лт}$ при скрытии (факта наличия) сигнала

Вероятность ложной тревоги $P_{лт}$	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Отношение сигнал/шум $\delta_п$	0,128	0,055	0,022	0,008	0,003	0,0012

Таблица 3 – Зависимость $\delta_п$ от $P_{лт}$ при скрытии передаваемой информации

Вероятность ложной тревоги $P_{лт}$	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-7}	10^{-8}
Отношение сигнал/шум $\delta_п$	0,235	0,088	0,032	0,012	0,004	0,0014

средств защиты и исключить из дальнейшего рассмотрения неопасные средства перехвата или воздействия на информацию, а также избежать излишнего засорения эфира при применении активных мер защиты. Аналогичные результаты получены также для скрытия ПЭМИ видеосистемы и клавиатуры компьютера.

Литература:

1. Хорев А.А. Технические каналы утечки информации, обрабатываемой средствами вычислительной техники // Специальная техника. – 2010. – № 2. – С. 39–56.

2. Нагорный С.И., Артамошин С.А. О применяемых технологиях предотвращения утечки информации по техническим каналам // Защита информации. Inside. – 2012. – № 2. – С. 82–86.

3. Акимов В.И., Данилов Н.С., Суворов П.А. Предложения по созданию адаптивных генераторов шума системы зашумления информативных сигналов средств электронно-вычислительной техники // Специальная техника. – 2012. – № 3. – С. 9–13.

4. Прасолова А.Е. Нейросетевые и статистические алгоритмы в задаче обнаружения сигналов // Телекоммуникации. – 2010. – № 2. – С. 2–6.

5. Хорев А.А. Оценка эффективности защиты информации от утечки по техническим каналам // Специальная техника. – 2006. – № 6. – С. 53–61.

ИЗОЛЯЦИЯ ПРИЛОЖЕНИЙ ДЛЯ ПОВЫШЕНИЯ НАДЕЖНОСТИ И БЕЗОПАСНОСТИ ФУНКЦИОНИРОВАНИЯ ПОРТАТИВНЫХ УСТРОЙСТВ

В.Ю. Коваленко, Д.А. Костюк,
БрГТУ, г. Брест

Одной из особенностей встраиваемых систем (ВС) является их длительное функционирование без перезагрузки и/или какого-либо специфического обслуживания. Поэтому для ВС часто характерны повышенные требования к надежности работы системного и прикладного программного обеспечения и устойчивости к сбоям, происходящим по вине как пользователя, так и из-за внутренних ошибок, а также достаточно актуальной становится устойчивость к попыткам проникновения и взлома. С другой стороны, для портативных ВС большая длительность работы сочетается с сильно ограниченным сроком работы без подзарядки. Для увеличения срока непрерывной работы от батареи подобные системы часто оснащают весьма ограниченными ресурсами (на текущий момент типичны частоты центрального процессора до 1ГГц, а чаще 200–400 МГц).

Высокой надежности функционирования многозадачной среды можно добиться несколькими путями: использованием только высококачественного и многократно оттестированного кода; изоляцией от внешних сетей, включая Интернет, использованием дополнительных средств защиты, таких как супервизоры, фаерволы, антивирусные программы. Однако ни одно из решений не является универсальным, а их совокупность оказывается слишком ресурсоемкой для работы на портативном устройстве.

Одним из способов добиться приемлемой производительности при высокой изоляции приложений друг от друга является виртуализация, подразумевающая запуск приложения в изолированной среде (англ. sandbox), когда его действия жестко контролируются виртуальной машиной (ВМ) [1]. Приложение изолировано от других программ и взаимодействует с ними через специальные интерфейсы, а доступ к ресурсам (памяти, портам и т.д.) инкапсулируется и также находится под контролем. В настоящее время это решение широко используется на серверах и высокопроизводительных системах, периодически применяется на рабочих станциях, однако, оно пока не характерно для маломощных портативных систем [2].

В настоящее время в сегменте ВС наблюдается интенсивный рост доли систем под управлением Unix-подобных операционных систем (ОС), в первую очередь, GNU/Linux. Для данного класса программных платформ, благодаря их длительному доминированию в сегменте серверов, сравнительно легко реализуем ряд подходов, обеспечивающих изоляцию отдельных программных компонент и, таким образом, повышающих общую надежность функционирования системы. В рамках данной работы нами выполнена адаптация одного из таких решений для нужд портативных ВС, работающих под управлением GNU/Linux.

Виды виртуализации зависят от того, насколько полно осуществляется эмуляция аппаратного обеспечения для виртуализованной ОС или приложения, и сильно различаются по степени потребления ресурсов хост-системы. К наименее ресурсоемким относятся следующие разновидности:

– частичная или нативная (native) виртуализация предполагает, что виртуальная машина имитирует лишь то количество аппаратного обеспечения, которое необходимо, чтобы она могла быть запущена изолированно; подход позволяет запускать гостевые ОС, разработанные только для той же архитектуры, что и у хост-системы;

– паравиртуализация избавляет от необходимости симулировать аппаратное обеспечение, однако, вместо этого используется специальный программный интерфейс для

взаимодействия с гостевой ОС, а сама гостевая ОС должна быть модифицирована, чтобы его использовать;

– виртуализация уровня приложений предполагает, что в ВМ запускаются не гостевые операционные системы, а отдельные приложения с необходимыми элементами для работы (файлами реестра, конфигурационными файлами, пользовательскими и системными объектами); этот подход применим только если виртуализованное приложение написано для той же ОС, которая установлена на хост-системе, но зато позволяет добиться высокой производительности при совпадении аппаратных архитектур гостевой и хост-систем.

Для сформулированной задачи наиболее разумно использовать именно метод виртуализации уровня приложений – изоляции (контейнеров, jail), который позволяет значительно повысить производительность и скорость отклика системы [3]. Контейнеры представляют собой специальные изолированные области, как в оперативной памяти, так и на системном диске без возможности доступа к внешним ресурсам и данным. При старте программа запускается внутри этого пространства и ее ошибки никак не влияют на работу других программ или самой системы. Также, реализуется изоляция от атак с помощью зараженных приложений (невозможность выйти за предел контейнера). Это справедливо и для такого критически важного параметра, как процессорное время, распределение которого между контейнерами тоже может контролироваться.

Существует несколько реализаций технологии контейнеров, но наиболее подходящей для портативных ВС под управлением ОС GNU/Linux, на наш взгляд, является технология lxc. В своей работе lxc не использует напрямую ни регистры процессора, ни какие-либо другие аппаратные особенности аппаратуры, на которой исполняется. Работа полностью основана на высокоуровневых особенностях ядра Linux, которые, в силу легкой портируемости этой операционной системы, делают продукт легко переносимым.

Предлагаемый подход был опробован в рамках разрабатываемого в Брестском государственном техническом университете информационно-вычислительной системы по наблюдению и прогнозированию наводнений [4], в состав которой входит портативный электронный терминал, позволяющий использовать беспроводные коммуникации и систему позиционирования GPS для отображения паводковой ситуации и ее прогноза применительно к точке нахождения оператора, либо для автоматизированного ввода результатов изменений с автоматическим определением их геолокации и включением точки измерения в банк данных сервера [5].

По результатам тестирования разработки можно сделать следующие выводы:

– выбранная технология изоляции приложений практически не влияет на производительность системы (потеря находится в пределах нескольких процентов);

– повышается устойчивость к сбоям внутри приложений и попыткам нарушения работоспособности системы в целом.

Литература:

1. Костюк, Д.А. Построение прозрачных виртуализованных окружений для изоляции уязвимых программных систем / Д.А. Костюк, С.С. Дереченник // Комплексная защита информации: матер. XVI научно-практич. конф., Гродно, 17–20 мая 2011 – г. Гродно, 2011. – С. 209–212.

2. Калиновский, Р.В. Технология виртуализации для смартфонов. Сборник конкурсных научных работ студентов и магистрантов / Р.В. Калиновский, Д.А. Костюк (научн. рук.). – Часть 1. – Брест: изд-во БрГТУ, 2007 – С. 99–102.

3. Коваленко, В.Ю. Обеспечение повышенной надежности и длительности работы программного обеспечения встраиваемых систем на базе ОС GNU/LINUX / В.Ю. Коваленко // Материалы 8-й Международной молодежной НТК «Современные проблемы радиотехники и телекоммуникаций РТ-2012». – Севастополь, 2012. – С. 369.

4. Волчек, А.А. Распределенная сетевая система отслеживания паводковой ситуации / А.А. Волчек, В.Ю. Коваленко, Д.А. Костюк, Д.О. Петров, Н.Н. Шешко. // Проблемы природопользования: итоги и перспективы: матер. Междунар. науч. конф., г. Минск, 21–23 ноября 2012 г. НАН Беларуси [и др.] – Минск: Минсктиппроект, 2012. – С. 120–123.

5. Коваленко, В.Ю. Разработка системного программного обеспечения для портативного терминала контроля паводковой ситуации / В.Ю. Коваленко // Современные информационные технологии в образовании и научных исследованиях (СИТОНИ-2012). Матер. III междунар. науч.-техн. конф. студентов и молодых ученых / Сб. науч. трудов. – Донецк: ДонНТУ, 2012. – С. 257–261.

ПРОБЛЕМЫ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ИНФОРМАЦИИ ОТ УТЕЧКИ ПО ТЕХНИЧЕСКИМ КАНАЛАМ В УСЛОВИЯХ СОВРЕМЕННОГО СОСТОЯНИЯ И ДАЛЬНЕЙШЕГО РАЗВИТИЯ СРЕДСТВ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

А.Ф. Мельник,
Государственное предприятие «НИИ ТЗИ», г. Минск

Защита информации от утечки по техническим каналам является одной из важнейших задач в области обеспечения безопасности информации, обрабатываемой средствами вычислительной техники (СВТ). Как известно, под техническим каналом утечки информации (ТКУИ) понимают совокупность источника информации, линии связи (физической среды), по которой распространяется информационный сигнал, шумов, препятствующих передаче сигнала в линии связи, и технических средств перехвата информации [1].

Среди всего разнообразия ТКУИ, обрабатываемой СВТ, в настоящее время особое место занимают электромагнитные и электрические каналы утечки информации, к которым