– вычисления хэш-функции по ГОСТ Р 34.11-94 и СТБ 34.101.31 2011 (функция хэширования);

– вычисления электронной цифровой подписи по ГОСТ Р 34.10-2001, а также вычисления цифровой подписи и обеспечения функции транспорта ключей по СТБ П 34.101.45-2011;

- встроенное фирменное программное обеспечение StellarisWare с возможностью его перепрограммирования со стороны хост-компьютера через USB интерфейс;

– драйвер и библиотека StellarisWare, обеспечивающие программный интерфейс взаимодействия с устройством со стороны хост-компьютера;

– прикладная программа icar_mod.exe, обеспечивающая тестирование и основные функции взаимодействия и обработки открытых и конфиденциальных данных.

Разрабатываемый модуль обеспечит:

– скорость шифрования данных по ГОСТ 28147-89 в режиме гаммирования без обмена с хост-компьютером – не менее 800 кбайт/с;

– скорость шифрования данных по ГОСТ 28147-89 в режиме гаммирования при обмене данными с хост-компьютером – не менее 600 кбайт/с;

– время формирования цифровой подписи при объеме блока данных 1 Кбайт – не более 2 сек;

– время вычисления хэш-функции при объеме данных 400 Кбайт – не более 4 сек.

Разрабатываемый модуль обеспечит защиту информации в распределенных проводных и/или беспроводных информационно-коммуникационных системах.

Литература:

1. Stellaris LM3S9997 Microcontroller. Datasheet. – Texas Instruments Incorporated, 2011. – 1328 p.

2. Stellaris LM3S9D96 Microcontroller. Datasheet. – Texas Instruments Incorporated, 2012. – 1408 p.

Abstract

The description of autonomous unit, designed in Institute of Applied Physical Problems of name A.N.Sevchenko BSU, is given in this work. The unit allows storing cryptographic keys and critical data in internal smart card such as simblock, to transmit data on the Ethernet network and wireless local area networks (IEEE 802.11 b/g-compliant), to interact with host-computer through the interface USB 2.0.

ПОВЫШЕНИЕ СЕТЕВОЙ БЕЗОПАСНОСТИ В КОМПЬЮТЕРНОМ ПАРКЕ ВУЗА ЗА СЧЕТ БУФЕРИЗАЦИИ И ИЗОЛЯЦИИ РЕСУРСОВ

П.С. Пойта, Д.А. Костюк, С.С. Дереченник, П.Н. Луцюк,
БрГТУ, г. Брест

Одной из проблем, сопутствующих развитию современного учебного заведения, как и любой иной крупной организации, является обеспечение работоспособности инфраструктуры внутреннего информационного пространства, объединяющего вычислительные ресурсы подразделений. Решения, обеспечивающие функционирование локальной вычислительной сети (ЛВС), должны выполнять защиту данных и программного обеспечения (ПО) от внешних и внутренних угроз. Централизованное администрирование компьютерного парка и общий набор накладываемых на пользователей ограничений, однако, часто не распространяется на подсети кафедр компьютерного профиля, например, взамен дополнительных мер контроля сетевого трафика на шлюзах, соединяющих их с основной ЛВС. Это обусловлено требованиями учебного процесса, т. к. в рамках ряда дисциплин, изучаемых специалистами компьютерного профиля, требуется предоставить студентам более полный доступ к аппаратной подсистеме рабочих станций [1, 4]. Данные кафедры обычно используют свой набор средств защиты – как существующих для нужд учебного процесса (например, практическое изучение средств «Аккорд NT/2000», «Аккорд – РАУ» и «Шипка–1.6», переданных Брестскому государственному техническому университету (БрГТУ) ОКБ «САПР» и Всероссийским НИИ проблем вычислительной техники и информатизации в рамках Соглашения о научно-техническом сотрудничестве [1]), так и разработанных сотрудниками кафедр в рамках исследовательских проектов в сфере информационной безопасности. Последний аспект обеспечивает площадку для тестирования инновационных подходов перед их возможным внедрением в ЛВС вуза.

Ниже, нами рассмотрен один из таких подходов, направленный на прозрачное усиление безопасности данных и снижение возможного ущерба от активности вредоносного кода, разработанный и примененный кафедрой электронных вычислительных машин и систем БрГТУ.

Сервисы (службы), предоставляемые пользователям в рамках подсети ЛВС БрГТУ, принадлежащей данной кафедре, включают:

– аутентификацию пользователей для работы в сети;

– функции файл-сервера (хранение личных файлов пользователей на персональных сетевых дисках, а также предоставление централизованного сетевого доступа к методическим материалам и ряду программных продуктов);

– обмен трафиком TCP/IP с внешней частью ЛВС (включая опосредованный доступ к ресурсам сети Интернет).

Основные роли пользователей сегмента сети – это студенты, преподаватели и учебно-вспомогательный персонал. Соответственно, защита сегмента ЛВС в рамках обеспечения учебного процесса и предоставления пользователям перечисленных сервисов, в свою очередь, подразделяется на следующие категории:

– физическая защита аппаратных средств;

– защита от вредоносной активности со стороны пользователей;

– резервное копирование системных и прикладных данных.

Физическая защита обеспечивается использованием отдельных помещений с ограниченным доступом для размещения коммутационных узлов сети и серверов. Авторизация пользователей в сегменте ЛВС кафедры выполняется сервером под управлением GNU/Linux, предоставляющим LDAP-базу пользовательских учетных записей (этот же сервер предоставляет доступ к сетевым дискам по протоколу CIFS), что является типичным для белорусских вузов [2]. Защита от возможной вредоносной активности пользователей выполняется заданием прав доступа, как на рабочих станциях, так и на сетевом разделе с методическими и демонстрационными материалами. Однако, в ряде случаев, пользователям необходимы повышенные привилегии. Помимо собственно администрирования компьютерного парка, в рамках сегмента сети кафедры таких задач две:

1. Предоставление преподавательскому составу возможности простой модификации раздела файл-сервера с методическими материалами и программным обеспечением;

2. Предоставление студентам более полного набора прав доступа к рабочей станции при изучении отдельных дисциплин [3].

В первом случае от файл-сервера требуется обеспечить права на запись и модификацию файлов без риска их повреждения. Во втором случае необходимо ослабить защиту рабочей станции, частично изолированной от окружающей среды, с последующей быстрой отменой сделанных изменений и возвращением системы к исходному состоянию.

Особенность первой задачи в том, что классическая защита с помощью антивирусного ПО не может полностью решить проблему. Преподавательскому составу периодически требуются права для модификации файлов на разделе файл-сервера с методическими материалами и ПО. Однако эти модификации могут иметь негативные последствия в случае доступа к серверу с зараженного компьютера (например, персонального ноутбука), на котором антивирусный контроль либо отсутствует, либо не справляется со своей работой, из-за чего вредоносное ПО, получив доступ к сетевым дискам, может подменять на них исходные файлы. Установленный на файл-сервере антивирусный монитор заблокирует доступ к зараженному файлу, предотвратив его дальнейшее распространение. Однако файл может оказаться недоступным до вмешательства системного администратора: во-первых, возможность «лечения» зараженных файлов имеют не все антивирусные системы и не для всех разновидностей заражения, а во-вторых, специфика механизма «лечения» не может гарантировать идентичность восстановленного файла исходному.

Для решения этой проблемы нами использована схема, показанная на рисунке 1-а. Помимо основного сетевого ресурса, содержащего методические материалы и ПО, в ЛВС доступен дополнительный ресурс, связанный с буферной файловой системой, имеющей специфический набор прав доступа: в ней запрещена модификация файлов, а доступны только их создание и удаление (причем удаление по умолчанию не уничтожает файл или каталог, а переносит объекты в «корзину»). Оба сетевых ресурса (основной и буферный) предоставляются рабочим станциям с помощью пакета Samba версии 3.x. Содержимое буфера анализируется антивирусным монитором (в качестве которого, например, может выступать ClamAV) и в случае обнаружения вредоносных свойств блокируется. Затем незаблокированные файлы средствами монитора Lsyncd автоматически переносятся на основной раздел файл-сервера, сетевой доступ к которому предоставляется только на чтение, независимо от привилегий пользователя. Помимо очевидной защиты данных от заражения, буферный ресурс одновременно играет роль хранилища резервных копий.

Вторая задача решалась нами с помощью средств виртуализации. ОС от Microsoft, которые помещались на рабочих станциях в изолированные окружения (рисунок 1-б). Особенности типичного офисного оборудования (в первую очередь, отсутствие аппаратной поддержки виртуализации в дешевых процессорах) оставляет систему виртуализации Oracle VirtualBox в качестве единственного решения приемлемой производительности, не требующего оплаты лицензий и обладающего к тому же открытым исходным кодом. По сходной причине в качестве хост-системы выбрана ОС GNU/Linux [1].

Сеть гостевой системы настроена в режиме трансляции адресов. Результатом является недоступность гостевой ОС извне при сохранении комфортного доступа из нее к сетевым сервисам – так же, как это происходит с рабочими станциями

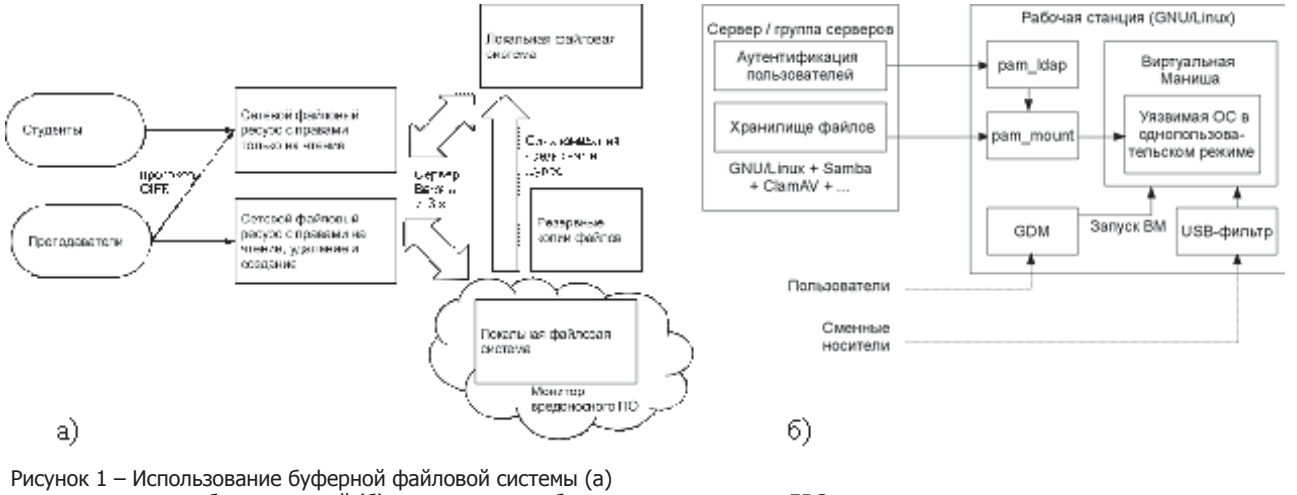

Рисунок 1 – Использование буферной файловой системы (а) и виртуализации рабочих станций (б) для повышения безопасности сегмента ЛВС

локальной сети, имеющими только локальные сетевые адреса и соединяющимися с внешней сетью proxy-сервером.

Хост-системы рабочих станций, работающие под управлением GNU/Linux, получают доступ к учетным записям и персональным файлам пользователя на файл-серверах сети с помощью стандартных модулей системы аутентификации Linux LDAP. При этом рабочие станции различаются только IP-адресом и имеют идентичный дисковый образ, что выгодно отличается от решения с ОС от Microsoft, работающей под управлением контроллера домена [3, 4]. Для обеспечения прозрачного доступа к виртуализованному окружению, полноэкранный запуск нужной виртуальной машины включен в список доступных графических оболочек на хост-системе и выбирается пользователем при вводе логина и пароля в менеджере сеансов GDM. Каталоги файл-сервера однотипно монтируются в файловую систему рабочей станции как при входе пользователя в гостевое окружение, так и при входе в графическую оболочку на хост-системе: гостевая система работает в однопользовательском режиме, автоматически (через механизм разделяемых папок VirtualBox) монтируя при запуске каталоги хост-системы, уже отображающие соответствующие конкретному пользователю сетевые ресурсы. Так выполняется изоляция гостевой ОС от внешней сетевой активности и делегирование аутентификации хост-системе.

Дисковый образ гостевой ОС при завершении сеанса работы автоматически откатывается к исходному состоянию (его фиксация выполняется функцией «immutable disk image» VirtualBox). Гостевое окружение благодаря этому становится «неповреждаемым», что позволяет безопасно работать в нем с правами администратора.

При необходимости автоматизировать передачу USB-накопителей в гостевое окружение, проброс может автоматически выполняться VirtualBox, для чего необходимо создание соответствующего фильтра устройств в настройках. Фильтрация USB-устройств позволяет в ряде случаев обеспечить дополнительную безопасность гостевой системы: например, может быть разрешен только проброс аппаратных ключей защиты, с игнорированием накопителей и других устройств, подсоединяемых к USB-разъемам.

Литература:

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

2. S.S. Derechennik, D.A. Kostiuk, D.A Pynkin. Free/libre software usage in the Belarusian system of higher education institutions // Друга міжнародна науково-практична конференція FOSS Lviv 2012: Збірник науковых праць, Львів, 26–28 квітня 2012 р. Львів, 2012. – С. 62–65.

3. П.С. Пойта, В.И. Драган, А.П. Дунец, Д.А. Костюк, В.И. Хведчук, С.С. Дереченник. Подход к модернизации гетерогенной сетевой инфраструктуры на примере информационно-вычислительной сети университета // Вестник БрГТУ. – 2010. – №5: Физика, математика, информатика. – С. 75–78.

4. Д.А. Костюк, Р.В. Сченснович. Использование в образовательном процессе вычислительных сетей на базе Windows Server 2008 // Информационные технологии в образовании: материалы Международной научно-практической конференции (Минск, 21–22 мая 2009). – Минск: БНТУ, 2009. – С. 109–113.

ИСПОЛЬЗОВАНИЕ НЕЙРОННЫХ СЕТЕЙ В ТЕСТИРОВАНИИ СЛОЖНЫХ ПРОГРАММНЫХ СИСТЕМ

Д.В. Бабин,
ОКБ САПР, г. Москва

Разработка программного обеспечения – сложный и трудоемкий процесс, в ходе которого непременно возникают ошибки. Часть из них так и не будут выявлены, пока программный продукт не попадет к конечному пользователю. Для того, чтобы сократить число скрытых проблем, применяются различные методы тестирования ПО. Проанализируем их.

1. Модульное тестирование. Автоматический метод тестирования, позволяющий проверять работоспособность небольших модулей, за счет написания unit-тестов. На сегодняшний день имеется огромное количество библиотек, помогающих быстро и удобно писать такие тесты.

2. Непрерывное интегрирование. Метод тестирования связей между модулями, за счет частых сборок разрабатываемого проекта и выполнения небольших проверок на совместимость модулей.

3. Поведенческие скрипты. Этот метод применяется как для частей ПО, так и для всей программы в целом. Суть его довольно проста. Пишутся небольшие «жесткие» скрипты, симулирующие работу человека. Но этот метод ограничен, так как разработка таких скриптов трудоемка, в них отсутствует случайность и реакция на исключительные ситуации.

4. Ручное тестирование. Пожалуй, лучший метод, так как ни одна программа не может заменить человека. Однако он имеет существенный недостаток, человеческие ресурсы ограничены.

Перечисленные методы автоматического тестирования показывают довольно хорошие результаты в своих узких областях. Но когда необходимо проверить работоспособность всего ПО в целом, ни один из автоматических методов не может даже близко приблизиться к ручному тестированию. Это обусловлено следующими проблемами:

– сложность написания тестов не линейно зависит от сложности ПО;

– отсутствует случайность в поведении тестов.

Таким образом, возникают две задачи:

1. Необходимо упростить создание или автоматически создавать тестовое покрытие.