Д-р. техн. наук. В.Я. Геча, канд. техн. наук С.Г. Семенцов, инж. С.А. Козлов, инж. А.М. Коноплев

КОНЦЕПЦИЯ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ LINUX-ПОДОБНЫХ ОПЕРАЦИОННЫХ СИСТЕМ В СИСТЕМАХ УПРАВЛЕНИЯ И ЗАЩИТЫ РЕАКТОРОВ АЭС

В последние несколько лет в сфере системного программного обеспечения систем управления различного назначения явно прослеживается тенденция к переходу от Windows-платформ и специализированных операционных систем (ОС) к Linux-подобным операционным системам. В частности Linux-подобные ОС все шире используются в системах управления и защиты реакторов на АЭС нового поколения. Причины этого очевидны: Linux-подобные системы сочетают в себе всю мощь 32-разрядной Unix-архитектуры и модульную структуру с открытостью исходных текстов, что позволяет компилировать различные дистрибутивы ОС с широким набором прикладных модулей под конкретную задачу. Модульная структура Linux-подобных ОС значительно упрощает процесс верификации и тестирования системы в соответствии с требованиями международных стандартов МЭК 60880, МЭК 60880-2.

Быстродействие Linux-подобных систем является одним из основных ее преимуществ. Следствием открытости исходных текстов явилось то, что тысячи инженеров-программистов получили доступ к корректировке, исправлению ошибок, отладке механизмов работы ОС, а также прикладного ПО. Сетевые компоненты развивались одновременно с разработкой новых протоколов передачи данных [1].

Существенным преимуществом Linux-подобных систем по сравнению с другими ОС является также поддержка динамического кэширования дисковой памяти, в то время как, например, в OS/2 реализован традиционный подход, состоящий в выделении фиксированного объема памяти (от 512 Кбайт до 2 Мбайт). В результате производительность Linux-подобных ОС оказывается существенно выше, поскольку необходимые для обработки данные в большинстве случаев оказываются уже в кэш-памяти. Для многих практических ситуаций эта возможность ставит Linux-подобные системы вне конкуренции при работе с широким спектром прикладного программного обеспечения.

Очевидно, что каждая дополнительная функция, реализованная

в системе, приводит к увеличению объема системы, что сказывается на требованиях к оперативной памяти и жестким дискам. Кроме того, чем больше объем операционной системы, тем медленнее она обычно работает. Однако Linux является примером ОС, которая максимально эффективно использует ресурсы аппаратного обеспечения. Для примера существуют версии Linux, использующие место на жестком диске лишь для системного свопа и прикладных программ и запускающиеся с дискеты.

Собственно говоря, сам Linux занимает исключительно малый объем, но система X-Window является довольно большим приложением. Впрочем, в большинстве случаев, наличие графического интерфейса обусловлено техническими требованиями и его присутствие в системе оправдано. При этом, для решения большинства задач достаточно 4 Мбайт оперативной памяти. В результате Linux с успехом может эксплуатироваться на платформах, оснащенных процессором 80386, в то время как для других операционных систем требования к аппаратному обеспечению значительно жестче.

Необходимо отметить, что минимальный размер системы является одной из наиболее базовых характеристик Linux-подобных ОС. Система изначально проектировалась максимально компактной и производительной, в то время как для других систем основной упор в проектировании делался на переносимость или другие факторы.

Спектр задач, ставящихся перед ОС различного назначения достаточно широк, однако можно выделить несколько базовых требований, предъявляемых к безопасным ОС [2]. Одним из них является необходимость наличия в системе элементов мониторинга и управления, причем как самих системных функций, так и прикладных программ и сетевых компонентов. Системы мониторинга позволяют контролировать работоспособность отдельных компонентов системы, производить перезапуск модулей при их зависании и вести журнал состояния, в котором логируется вся служебная информация о состоянии памяти, свободном месте на дисках и т.д. Служебные модули могут быть интегрированы в систему как при установке и первичной настройке, так и на этапе инсталляции прикладных программных модулей.

Еще одним важным критерием безопасности ОС является степень разграничения прав пользователей [3]. Эта концепция, безусловно, должна быть положена в основу при разработке систем управления на базе Linux-подобных систем. При этом под разграничением прав пользователей подразумевается не только ограничение прав на выполнение тех или иных операций, но и прав доступа к файлам. Безопасность работы ОС на пользовательском

уровне была и остается важнейшим критерием при администрировании ОС, особенно при отсутствии постоянного мониторинга системы.

В дистрибутиве Linux-подобных безопасных ОС должна быть предусмотрена возможность установки специальных флагов на всех файлах и директориях системы, определяющих права записи и/или чтения для каждого пользователя или группы пользователей. При добавлении нового пользователя для работы в системе обязательным параметром его работы является уровень его доступа к объектам системы.

При регистрации новых пользователей, наряду с их именами, необходимо указать идентификационный номер пользователя (UID) и идентификационный номер группы (GID), членом которой он является. UID 0 – это специальный привилегированный пользователь (root), в большинстве Linux-подобных систем режим привилегированного пользователя позволяет обойти большинство проверок безопасности и используется для администрирования системы. На некоторых системах Unix GID 0 также является специальным и разрешает неограниченный доступ к ресурсам на уровне группы; этот режим не является обязательным на других системах, но даже в таких системах группа 0, по существу, является всесильной, потому что много специальных файлов системы принадлежат группе 0. В соответствии с группой, к которой принадлежит пользователь, система и предоставляет необходимые привилегии при работе в ОС, как с консольного терминала, так и при активации файлов через сетевые эмуляторы терминалов.

Что касается устойчивости к внешним воздействиям, то, благодаря Unix-архитектуре, Linux-подобные ОС функционируют в защищенном режиме и единственной потенциальной уязвимостью здесь являются возможные DOS-атаки (отказ от обслуживания) и утечка информации о пароле привилегированных пользователей, так как в системе не существует возможности запуска исполняемых двоичных файлов. Существует также потенциальная опасность атаки на уровне сетевых протоколов и служб, однако в системе службы, не являющиеся необходимыми, могут быть либо деактивированы и, соответственно, не запускаются при загрузке, либо закрыты неиспользуемые порты, что существенно усложняет процесс взлома. Вирусная атака в Linux-подобных ОС может привести лишь к переполнению буфера пакетов на сетевом интерфейсе, что не может повлиять на работу самой системы и ее ядра. Linux-подобные ОС обеспечивают надежную защиту от сетевых атак различного характера. Встроенный анализатор протокола IP позволяет осуществлять интеллектуальный анализ поступающих на сетевой интер-

фейс пакетов (причем, как по адресу отправителя, так и по адресу получателя). Операционная система может быть установлена с минимальным анализом пакетов по отдельным службам (например авторизация и фильтрация при работе с протоколом FTP), а также с расширенным анализом (установка пакетов ПО программной фильтрации – программ межсетевых экранов). Как уже было показано выше, контроль за доступом может быть реализован как на уровне пользователя, так и на уровне протокола. В сегментах Ethernet должны передаваться только пакеты с заранее известными MAC-адресами. Появление в сегменте постороннего пакета означает подключение к сегменту дополнительного устройства, на что система безопасности должна выдать предупредительный сигнал. Для отслеживания посторонних пакетов могут использоваться специальные программы, осуществляющие контроль всего трафика в выделенном сегменте. Для предотвращения распространения посторонних пакетов в другие сегменты можно использовать возможности коммутаторов, настраивая соответствующим образом их внутренние таблицы переадресации и возможности маршрутизаторов, задавая им жесткие таблицы маршрутизации.

В качестве примера реализации безопасной Linux-подобной операционной системы [4] можно привести дистрибутив LiNEM, разработанный в НПП ВНИИЭМ и предназначенный для работы в составе программно-технического комплекса информационно-диагностической сети системы управления и защиты реакторов ВВЭР-1000. При разработке этого дистрибутива были учтены все сформулированные выше рекомендации. В частности в системе была реализована возможность мониторинга статуса прикладных и системных процессов. ОС отслеживает состояние флагов активности прикладных модулей, и в случае зависания одного (или нескольких) из них производится перезапуск прикладного модуля. Если после трех перезапусков модуль по-прежнему не отвечает на системные вызовы, то управление передается резервному серверу и происходит перезагрузка ОС на основном сервере. Также посредством сетевых интерфейсов контролируется работоспособность самой операционной системы. Если после трех перезагрузок ОС не восстанавливается работоспособность прикладного модуля или системных процессов, то система запускается с диска аварийной загрузки/восстановления.

В рамках разработки программно-технического комплекса информационно-диагностической сети для обеспечения корректного функционирования комплекса была разработана система управления - программное обеспечение живучести (ПОЖ) [5]. В системе ПТК ИДС ПОЖ выполняет следующие основные функции:

- диагностику программных средств комплекса;
- диагностику аппаратных средств комплекса;
- обеспечение работы комплекса в режиме горячего резервирования;
- оперативную переустановку программного обеспечения в случае отказов и сбоев аппаратных и программных средств.

ПОЖ представляет собой систему управления, построенную по схеме «менеджер-агент». Схема взаимодействия «менеджер-агент» представлена на рис.1. По этой схеме взаимодействия построены практически все современные системы управления сетью. Используя эту схему можно построить систему управления сетью практически любой сложности.

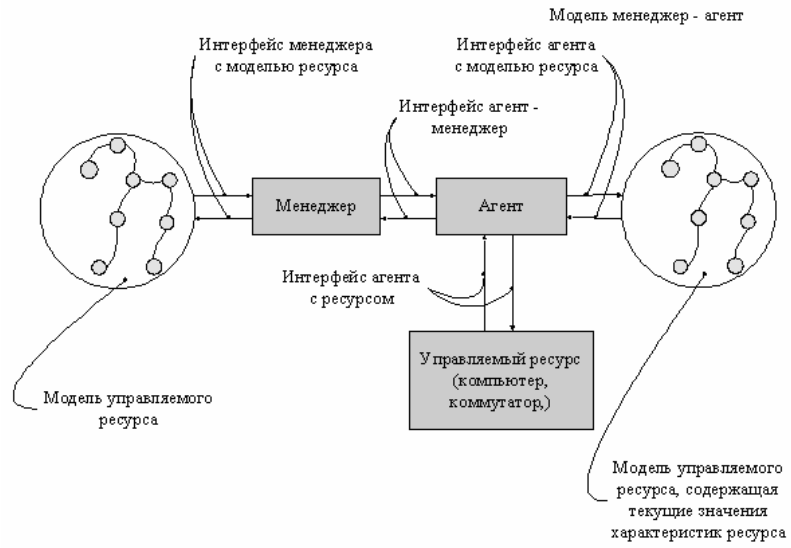

Рис. 1. Схема взаимодействия «менеджер-агент»

Агент является посредником между управляемым ресурсом и основной управляющей программой-менеджером. Чтобы один и тот же менеджер мог управлять различными реальными ресурсами, создается некоторая модель управляемого ресурса, которая отражает только те характеристики ресурса, которые нужны для его контроля и управления.

Менеджер получает от агента только те данные, которые описываются моделью ресурса. Агент же является некоторым экраном, освобождающим менеджера от ненужной информации о деталях реализации ресурса. Агент поставляет менеджеру обработанную и

представленную в нормализованном виде информацию. На основе этой информации менеджер принимает решение по управлению.

Для получения требуемых данных от объекта, а также для выдачи на него управляющих воздействий агент взаимодействует с реальным ресурсом некоторым нестандартным способом. Когда агенты встраиваются в коммуникационное оборудование, то разработчик оборудования предусматривает точки и способы взаимодействия внутренних узлов устройства с агентом. При разработке агента для операционной системы разработчик агента пользуется теми интерфейсами, которые существуют в этой ОС, например интерфейсами ядра, драйверов и приложений.

Как менеджер, так и агент должны располагать одной и той же моделью управляемого ресурса, иначе они не смогут понять друг друга. Однако в использовании этой модели агентом и менеджером имеется существенное различие. Агент наполняет модель управляемого ресурса текущими значениями характеристик данного ресурса, и в связи с этим модель агента называют базой данных управляющей информации — Management Information Base (MIB). Менеджер использует модель, чтобы знать о том, чем характеризуется ресурс, какие характеристики он может запросить у агента и какими параметрами можно управлять.

Вкратце алгоритм работы ПОЖ можно описать следующим образом. В такие элементы оборудования ПТК ИДС, как компьютеры, коммутаторы, системы контроля микроклимата, источники бесперебойного питания встроены программные агенты, имеющие сетевой интерфейс. Специальные программы-менеджеры собирают данные от агентов и передают их в программы-серверы, которые, после обработки данных от агентов, позволяют получить картину состояния оборудования и программного обеспечения комплекса и осуществляют контроль и управление.

Таким образом, собираются данные, описывающие состояние элементов ПТК ИДС, начиная с информации о состоянии аппаратуры компьютеров, последовательных портов, сетевых интерфейсов, загрузки процессора, оперативной памяти, режимах работы прикладного и системного ПО и заканчивая такими параметрами, как состояние портов коммутатора, трафик в сегментах сети, температура и влажность в помещении, уровень зарядки батарей в ИБП.

Собранная информация по протоколу IPX передается главному менеджеру, расположенному на сервере сети. Для обеспечения отказоустойчивости этот менеджер установлен также на дублирующем сервере, а данные передаются по всем возможным каналам связи, определенным при первоначальном тестирование канального уровня сети.

Путем совместной обработки и анализа полученных в процессе тестирования количественных и качественных характеристик оборудования и программного обеспечения составляется, так называемая, «карта неисправностей» ПТК ИДС, на основании которой автоматически принимаются решения по управлению работой комплекса, назначаются основной и дополнительный серверы, текущее состояние выводится на мониторы и передается в системы верхнего уровня, осуществляется управление процессами, функционирующими на компьютерах комплекса и т.д. Управление осуществляется путем взаимодействия главного менеджера и локальных программ управления по инициативе менеджера, а также частично осуществляется локальными менеджерами самостоятельно. Локальные менеджеры взаимодействуют с оборудованием и ПО и уведомляют главного менеджера о своих действиях, что обеспечивает корректность полученной им информации.

Для синхронизации данных между основным и резервным серверами в дистрибутиве реализована возможность зеркалирования информации по специальным надежным протоколам транспортного уровня IP. Программный модуль, работающий в системе резидентно, запускается с момента старта системы. Этот модуль может осуществлять зеркалирование как отдельных файлов, так и целых директорий с любой степенью вложенности. В системе также существует возможность задания следующих критериев зеркалирования: время создания файла, название файла, признак изменения файла или директории. Программа зеркалирования имеет удобный интерфейс настройки с возможностью указания серверов, с которыми устанавливается режим синхронизации файловой структуры, интервал времени через который будет осуществляться копированием, а так же специальные атрибуты работы самого ПО синхронизации.

Для предотвращения возможных сетевых атак в системе по умолчанию закрыты все порты, кроме портов IP, IPX, и программы зеркалирования. Разбиение всех пользователей на группы с предоставлением четко определенных прав для каждой группы позволяет организовать доступ к важным элементам файловой структуры и предоставить права на исполнение команд только для привилегированных пользователей. Для обеспечения устойчивости парольной защиты к программам взлома пароли привилегированных режимов должны содержать не менее 7 знаков, причем из разных регистров и с использованием специальных символов. Если при входе в систему несколько раз вводится неправильный пароль, то работа пользователя, под именем которого осуществляется попытка входа, запрещается. Такой подход гарантирует целостность системы и сохранность служебной информации.

Для проверки аппаратно-независимых функций СПО в НПП ВНИИЭМ был разработан стенд оценки надежности ПО, позволяющий реализовать основные режимы работы ПТК ИДС. В частности на стенде организован информационный обмен между основным и резервным сервером по протоколам TCP/IP, IPX и интерфейсу модуля зеркалирования данных. Структура стенда оценки надежности представлена на рис. 2.

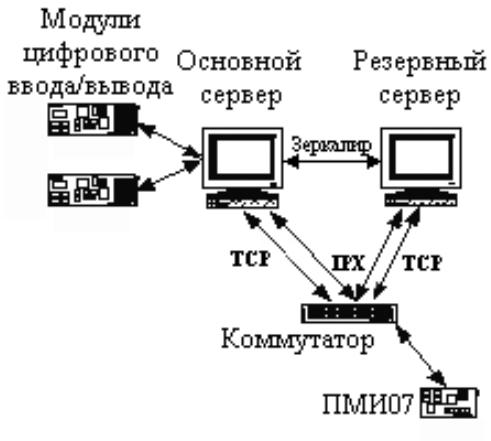

Рис. 2. Структура стенда оценки надежности СПО LiNEM

Информационный поток от органов регулирования в виде IPX-пакетов формируется с помощью аппаратного модуля ПМИ07 и поступает на коммутатор. Сигналы неисправностей заводятся через плату цифрового ввода или эмулируются с помощью программного имитатора диагностических данных (ИДД) непосредственно на основной сервер. Затем эти данные обрабатываются на основном сервере и копируются на резервный сервер с помощью программы зеркалирования. Обмен между основным и резервным сервером по протоколу TCP/IP необходим для работы программы ПОЖ, осуществляющей контроль функционирования СПО на аппаратном уровне.

Разработанный дистрибутив подвергался тестированию в течение 20000 ч на стенде оценки надежности СПО LiNEM. Контролировалась устойчивость работы системных служб, стабильность передачи данных по протоколам IP и IPX, надежность службы зеркалирования и парольной защиты, контроллеров цифрового ввода/вывода. Для проведения испытаний была разработана методика, позволяющая объективно оценить соответствие ОС требованиям технического задания. В частности для контроля службы зеркали-

рования проверялись контрольные суммы последовательностей тестовых файлов. Тестирование сетевых протоколов проводилось методом контроля среднего времени прохождения тестовых пакетов TCP (рис. 3) за все время тестирования.

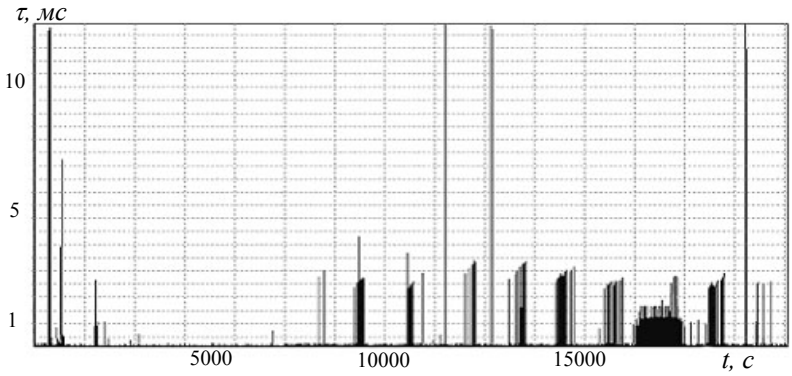

Рис. 3. Время прохождения тестовых пакетов TCP в процессе тестирования

Контролировалось также число потерянных пакетов на 1000 посланных. При тестировании контроллеров цифрового ввода/вывода генерировалась и передавалась псевдослучайная последовательность тестовых сигналов. После приема данных осуществлялась сверка контрольных сумм посланной и принятой информации.

Результаты испытаний подтвердили высокий уровень безопасности разработанного дистрибутива и показали, что Linux-подобные операционные системы удовлетворяют требованиям, предъявляемым к важным для безопасности системам управления.

ЛИТЕРАТУРА

1. Саранцев П.В./Программы и протоколы, осуществляющие межсетевое взаимодействие в системах управления и защиты АЭС // См. наст. том.

2. David Wheeler. Secure Programming for Linux and Unix. Linux.com. 2002.

3. Козлов С.А./Проблемы безопасности прикладного программного обеспечения для Linux-подобных операционных систем в важных для безопасности АЭС системах // См. наст. том.

4. Липаев В.В. Надежность программного обеспечения. М.: Энергоиздат. 1999.

5. Герман Н.Р., Коноплев А.М./Опыт реализации подсистемы контроля и диагностики в составе операционной системы LiNem //См. наст. том.