СИСТЕМЫ И СРЕДСТВА ЗАЩИТЫ ИНФОРМАЦИИ

УДК 681.3.06

А. В. ПОТИЙ, канд. техн. наук, С. С. ЛАВРИНЕНКО

ЗАЩИЩЕННЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

По данным годового отчета «2001 Computer Crime and Security Survey» [1] Института компьютерной безопасности в Сан-Франциско и ФБР, финансовые потери от компьютерных преступлений в США за минувший год выросли на 43% с 265,6 млн. долл. до 377,8 млн. При этом 85% респондентов из 538, в основном из промышленных и государственных структур, заявили о фактах нарушения компьютерной безопасности, причем не только из-за атак злоумышленников. Почти 64% были озабочены понесенными убытками, но лишь 35% смогли оценить их в денежном выражении. Около 70% респондентов заявили, что чаще всего атакам подвергались Internet-каналы, а 31% показали, что атакам подвергались внутрикорпоративные системы. Случаи вторжения извне подтверждали 40% респондентов (в 2000 г. – 25%), а 38% фиксировали отказ в обслуживании (27% в 2000 г.). На нарушение привилегий из-за злоупотребления сотрудниками работой в Сети жаловались 91% респондентов, а 94% обнаружили в своих системах вирусы (в 2000 г. это отмечали 85%).

Таким образом, одними из ведущих проблем в области информационной безопасности в минувшем году стали атаки на платежные системы, дискредитация компаний (отказ в обслуживании), производственный саботаж, вскрытие корпоративных секретов, нарушение прав интеллектуальной собственности. По оценкам отдела по науке и технологиям при Президенте США, ежегодный урон, наносимый американскому бизнесу компьютерными злоумышленниками в последние годы, достигал 100 млрд. долл. Потери от несанкционированного доступа к информации, связанной с деятельностью финансовых институтов США, составляли не менее 1 млрд. долл. в год. Таким образом, американский бизнес вплотную подошел к тому рубежу, когда своевременное и адекватное решение вопросов безопасности для него становится экономически целесообразным.

1 Критерии и ориентиры в области безопасности

Работы над критериями безопасности систем начались еще в 1967 г., и в 1970 г. появился первый отчет под названием «Security Controls for Computer Systems». В 1983 г. Министерство обороны США выпустило «Orange Book» – книгу в оранжевой обложке под названием «Критерии оценки достоверных компьютерных систем» (Trusted Computer Systems Evaluation Criteria). В «Оранжевой книге» достоверная система определяется как «система, использующая достаточные аппаратные и программные средства для обеспечения одновременной обработки информации разной степени секретности группой пользователей без нарушения прав доступа».

Выделяются два основных критерия оценивания достоверных систем:

- политика безопасности (набор правил и норм, определяющих дисциплину обработки, защиты и распространения информации, а также выбор конкретных механизмов обеспечения безопасности; это активный компонент защиты);
- гарантированность (степень доверия, которая может быть оказана конкретной реализации ОС; отражает уровень корректности механизмов безопасности; является пассивным компонентом защиты).

В соответствии с «Оранжевой книгой» выделяются три роли: системный администратор, системный оператор и администратор безопасности.

За пределами США также появились аналоги «Оранжевой книги»: это руководящие документы Гостехкомиссии (1992 г.), а также «Критерий оценки безопасности информационных технологий» (ITSEC – Information Technology Security Evaluation Criteria, 1991), действующий в Великобритании, Германии, Франции и Нидерландах.

Конечно же, в силу необходимости унификации подходов к информационной безопасности в конце концов возникла потребность снять двойственность регулирования, которая отдельно велась в США (TCSEC) и Европе (ITSEC). Поэтому был принят новый международный стандарт, получивший название «Единые критерии для оценки безопасности в области информационных технологий» [2], являющийся международным стандартом ISO/IEC 15408.

Описание Common Criteria V2.1 содержится в трех книгах:

1. Введение и общая модель (CCIMB-99-031).
2. Функциональные требования к безопасности (CCIMB-99-032).
3. Требования к гарантиям безопасности (CCIMB-99-033).

В «Единых критериях» выделяются 11 функциональных классов:

- аудит;
- криптографическая поддержка;
- передача данных;
- защита данных пользователя;
- идентификация и аутентификация;
- управление безопасностью;
- конфиденциальность;
- защита функций безопасности целевой системы;
- утилизация ресурсов;
- доступ к целевой системе;
- достоверные пути/каналы.

Внутри каждого их этих классов содержится несколько семейств, а в каждом семействе – от одного до нескольких компонентов.

Критерии, сформулированные в TCSEC, ITSEC и CCITSE, определяют разбиение компьютерных систем на 4 уровня безопасности (A, B, C, D) в зависимости от степени достоверности. Уровень A самый высокий. Далее идет уровень B (в порядке понижения безопасности здесь идут классы B3, B2, B1). Затем наиболее распространенный уровень C (классы C2 и C1). Самый нижний уровень – D (системы, которые не смогли получить аттестацию по заявленным выше классам).

2 Защищенность операционных систем

Любую стандартную операционную систему можно сделать более защищенной или укрепить ее с помощью простых процедур – например, отказаться от очевидных значений пароля администратора или отключить соединения с Web, если они не используются. Но эти отвечающие здравому смыслу шаги могут требовать больших затрат времени и, увы, далеко не всегда способны защитить критически важный сервер от целеустремленного хакера.

Истинная защищенная операционная система изначально строится с учетом требований безопасности. Защищенную ОС отличает наличие трех следующих «рычагов»:

- Политика принудительного контроля доступа (mandatory access control). MAC обеспечивает принудительное управление доступом на основе классической модели Белла-Лападулы, цель которой – не допустить перетекания информации из более секретных объектов в менее секретные.
- Администрирование привилегий, которые можно использовать для управления и ограничений возможностей пользователя или приложения, управляющих системой или ее частью. В защищенной ОС можно устанавливать программу, которая никогда не сможет изменить назначение привилегий, даже в том случае, если каким-то образом эта программа попадет в полную власть хакера. Такое решение не позволит хакеру проникнуть в систему через какое-то приложение, если, скажем, ему удастся отключить пароль, защищающий другие приложения.
- Независимая экспертиза, проводимая, например, в Национальном институте стандартов и технологии или Агентством национальной безопасности США.

Если исходить из этих критериев, то самые распространенные ОС – Microsoft Windows NT и Windows 2000, а также различные версии Unix – не являются защищенными системами, хотя Windows 2000 – это заметный шаг вперед благодаря предусмотренной в ней «защите системных файлов», позволяющей обеспечить безопасность некоторых критически важных компонентов.

Защищенные операционные системы крупнейших производителей Unix-систем, таких как Sun Microsystems и Hewlett-Packard, существуют уже давно, но они не пользовались особой популярностью из-за сложности управления и отсутствия ряда важных функций, которые есть у их коммерческих аналогов. Кроме того, они были не полностью совместимы с приложениями, работавшими с их менее защищенными вариантами.

Как правило, эти защищенные ОС использовались только в условиях с высоким уровнем риска – в банках и госучреждениях, которые могли позволить себе уделять достаточно времени и средств управлению подобными системами.

3 Использование Unix-систем

Сегодня многие индивидуумы и организации остановили свой выбор на операционной системе Linux, взяв ее на вооружение в качестве основной платформы. Причем, если раньше эта ОС использовалась преимущественно в качестве основы для построения недорогих маршрутизаторов, Web-серверов и почтовых серверов, то сегодня имеется целый ряд более серьезных задач, которые удобно решать на ее базе: серверы баз данных и доступа, серверы приложений и полнофункциональные межсетевые экраны. Благодаря высокой устойчивости ОС Linux ей можно смело поручать длительные вычислительные задачи.

Как известно, Linux – это клон Unix, семейства операционных систем, объединенных общей идеологией. Первые Unix-системы создавались исключительно как средство совместного ведения проектов научными коллективами университетов, и ее разработчики не придавали особого значения вопросам безопасности и защиты информации. Во всяком случае, эти вопросы точно стояли не на первом месте. Это привело к тому, что ни одну из классических Unix-систем нельзя сегодня назвать по-настоящему безопасной.

Относительно защищенный вариант Unix-системы можно построить на базе ее штатных средств, и для большинства применений это будет вполне адекватный уровень безопасности. Однако в самой идеологии построения Unix имеется ряд фундаментальных изъянов, которые невозможно преодолеть только грамотным администрированием.

Фактически, в Unix предполагается всего два варианта статуса пользователя: обычный и суперпользователь (root), что сразу порождает две проблемы. Первая проблема заключается в том, что root никому не подконтролен: он может изменить любую настройку, имеет доступ абсолютно ко всем объектам в файловой системе, может стереть или модифицировать любые

данные и записи в журналах регистрации. Такая «концентрация власти», конечно же, неприемлема для защищенной системы. Вторая проблема связана с необходимостью изменения текущего уровня привилегий пользователя для выполнения некоторых действий (например, для смены пароля пользователю требуется временное разрешение на запись в файл /etc/shadow или аналогичный). Традиционно это достигалось путем установки флага SUID (или SGID) на файлы исполняемых модулей программ, в результате чего порождаемый при запуске такого файла процесс приобретает не права запустившего его пользователя, а права владельца файла. Нет необходимости говорить, к чему может привести даже небольшая ошибка в программе с установленным флагом SUID и владельцем root.

Реализованная в Unix модель разграничения доступа дискретна – права доступа субъектов (пользователей, процессов) к объектам (файлам, каталогам и т.п.) задаются явно, в матричной форме. На практике же часто возникает необходимость в реализации принудительной модели, при которой права доступа субъекта к объекту определяются уровнем субъекта и классом объекта, либо комбинированной модели, включающей в себя принудительную и доступ на основе списков управления доступом. Причем, если в других ОС, использующих дискретную модель (например, Novell NetWare), необходимого разграничения достичь можно, то в Unix реализация некоторых требований по разграничению может оказаться попросту невозможной. Это связано с тем, что в Unix права доступа задаются всего для трех категорий субъектов: владельца файла, ассоциированной с файлом группы и всех прочих.

Ниже приводятся две системы, которые могут быть применены в качестве решения приведенных проблем с организацией защиты: RSBAC (Rule Set Based Access Control) и Security-Enhanced Linux, позволяющие построить защищенную систему на базе ядра Linux.

3.1 RSBAC

RSBAC – это надстройка над ядром Linux и комплект утилит управления, позволяющие создать на базе любого дистрибутива Linux защищенную систему. Одной из целей создания RSBAC, по заявлению авторов, является выполнение всех требований B1 по классификации так называемой «Оранжевой Книги» (хотя набор этих требований уже несколько устарел).

Реализация механизмов обеспечения безопасности выполнена на уровне ядра системы и позволяет эффективно контролировать все процессы. Системные вызовы, затрагивающие безопасность, дополняются специальным кодом, выполняющим обращение к центральному компоненту RSBAC. Этот компонент принимает решение о допустимости данного системного вызова на основе многих параметров:

- типа запрашиваемого доступа (чтение, запись, исполнение);
- субъекта доступа;
- атрибутов субъекта доступа;
- объекта доступа;
- атрибутов объекта доступа.

Функционально RSBAC [3] состоит из нескольких модулей, а центральный компонент принимает комплексное решение, основываясь на результатах, возвращаемых каждым из активных в данный момент модулей (какие модули задействовать и в каком объеме определяется на этапе настройки системы). Начиная с версии 1.1.0, в RSBAC включены следующие модули, реализующие различные функции и модели управления доступом:

- **MAC (Mandatory Access Control).** Модуль MAC обеспечивает принудительное управление доступом на основе классической модели Белла-Лападулы [4], цель которой – не допустить перетекания информации из более секретных объектов в менее секретные.
- **FC (Functional Control).** Данный модуль реализует простую ролевую модель, в которой доступ к системной информации разрешен только администраторам системы, а

доступ к информации, связанной с безопасностью, разрешен только офицерам безопасности.

- **SIM (Security Information Modification).** Модуль SIM обеспечивает возможность модификации данных, помеченных как «security information», только администраторами безопасности.
- **PM (Privacy Model).** Данный модуль реализует модель безопасности, направленную на обеспечение приватности личных данных. Основная идея [5] состоит в том, чтобы пользователь мог получить доступ к персональным данным только, если они ему необходимы для выполнения текущей задачи и если он авторизован на ее выполнение. Кроме того, цели выполнения текущей задачи должны совпадать с целями, для которых эти данные собраны, либо должно быть получено согласие субъекта этих данных.
- **MS (Malware Scan).** Этот модуль обеспечивает сканирование всех файлов на наличие вредоносного кода. Дополнительно данный модуль может контролировать все запросы на чтение файлов и соединений TCP/UDP. В текущей версии модуль умеет обнаруживать вирусы Bliss.A, Bliss.B, VHP-648, Israeli, Eddie2, Dark Avenger и 1704C. Конечно, это немного и вирусы не самые новые и опасные, но это только начало.
- **FF (File Flags).** Модуль FF предоставляет механизм установки и проверки флагов на файлы и каталоги. Причем модифицировать флаги разрешено только офицерам безопасности системы. Пока поддерживаются флаги execute_only (для файлов), read_only (для файлов и каталогов), search_only (для каталогов), secure_delete (для файлов), no_execute (для файлов) и add_inherited (для файлов и каталогов). Смысл флагов ясен из их названия и, конечно, знаком тем, кто пользовался механизмами атрибутов и прав в файловой системе Novell NetWare.
- **RC (Role Compatibility).** Данный модуль определяет 64 роли и 64 типа для каждого вида объекта. Виды объектов могут быть следующими: file, dir, dev, ipc (interprocess communication), scd (system control data), process. Для каждой роли отношение к различным типам и другим ролям настраивается индивидуально, в зависимости от вида запроса. Используя данный модуль, можно настроить разделение обязанностей между администраторами, избежав при этом назначения избыточных прав.
- **AUTH (Authorization Enforcement).** У этого модуля задача очень конкретная – контролировать запросы процессов на смену текущего идентификатора пользователя. Под контролем данного модуля программе недостаточно просто иметь установленный бит suid, ей необходим специальный атрибут, разрешающий такое действие. Причем он может быть как глобальным (uid может быть сменен на любой), так и списочным (процесс может сменить свой uid только на определенные).
- **ACL (Access Control List).** ACL – самый «прикладной» модуль из данного списка. Он определяет, какие субъекты могут получать доступ к данному объекту и какие типы запросов им разрешены. Субъектом доступа может быть как простой пользователь, так и роль RC и/или группа ACL. Объекты группируются по видам, но каждый имеет собственный список ACL. Если права доступа к объекту не заданы явно, они наследуются от родительского объекта с учетом маски наследования прав. Эффективные права доступа субъекта к объекту складываются из прав, полученных непосредственно, и прав, полученных через назначение на роль или членство в группе ACL. Такой подход к назначению прав покажется очень знакомым специалистам, работавшим с NetWare. Более того, утилита управления acl_grant имеет режим совместимости по названиям прав с аналогичной утилитой из NetWare.

Вся дополнительная информация, используемая RSBAC, хранится в дополнительном каталоге, который доступен только ядру системы. В зависимости от описанного уровня абстракции исполняемых задач и необходимой степени защиты выбираются различные сочетания модулей.

RSBAC дает возможность избавить систему от общих недостатков Unix. Появление должности «офицер безопасности» (security officer, security administrator) решает проблему неподконтрольности основного администратора системы, а категории «офицер по защите данных» (data protection officer) децентрализует администрирование, позволяя практически реализовать принцип «четырех глаз», в соответствии с которым все критичные операции не должны производиться в одиночку. Действительно, возможно построение системы, в которой нового пользователя по-прежнему заводит root, однако его уровень безопасности и права доступа к ресурсам задаются офицером безопасности и офицером по защите данных.

Стоит, однако, помнить об одной простой вещи: любая защита накладывает ограничения, а степень защищенности всегда обратно пропорциональна удобству работы с системой. Погоня за секретностью и безопасностью может привести к временной блокировке доступа к данным, а в отдельных случаях – и к их потере.

RSBAC можно интегрировать с любым дистрибутивом Linux, необходимо лишь внести соответствующие исправления в исходные тексты ядра системы. Эта процедура осуществляется при помощи стандартной утилиты patch и прилагаемого файла-»заплатки», который предлагается для разных версий ядра. После этого происходит его обычная настройка (make config, make menuconfig и т.п.), при этом в настройке появляется несколько новых пунктов. После загрузки ядра настройка системы производится при помощи прилагаемых утилит администрирования. Система снабжена документацией и подборкой теоретических материалов, где описаны все используемые модели защиты.

3.2 Security-Enhanced Linux

Вторая система – Security-Enhanced Linux имеет такое же назначение, как RSBAC и также представляет собой дополнения к ядру и набор утилит. Обе системы доступны под лицензией GPL, однако разработка Security Enhanced Linux продвигается Агентством национальной безопасности США. Security Enhanced Linux «моложе» RSBAC – ее релиз впервые был представлен в декабре 2000 года [6]. Security-Enhanced Linux обеспечивает гибкую архитектуру принудительного контроля доступа (flexible mandatory access control architecture – FLASK) [7], использующую развитый язык описания конфигураций политики безопасности. С использованием этого языка описаний разработана конфигурация системы, реализующая идеологию Type Enforcement [8].

В терминологии данной системы политика безопасности определяет набор доменов и типов. Каждый субъект (процесс) в каждый момент времени ассоциирован с определенный доменом, а каждый объект – с определенным типом. Как и в классической матрице, в политике определяются возможные виды доступа доменов к типам и допустимые способы взаимодействия между доменами. В частности, в зависимости от используемых типов может происходить автоматическое переключение домена.

В политике безопасности также определен набор ролей. Для пользователей первичная установка роли происходит в процедуре регистрации (login), а переключение роли – при помощи команды newrole (в некотором роде аналог su). Системные процессы работают с ролью system_r, обычные пользователи – с ролью usr_r, а для системных администраторов зарезервирована роль sysadm_r.

Для каждой роли в политике безопасности задается набор доменов, в которых допускается работа с этой ролью. Всем пользовательским ролям назначается стартовый домен: user_t для роли user_r и sysadm_t для роли sysadm_r. По мере выполнения программ, запускаемых из стартовой оболочки shell, может происходить автоматическое перемещение в другие домены для обеспечения изменения привилегий. Выбор домена, в который должно быть произведено перемещение, осуществляется не только исходя из типа запускаемой программы, но и с учетом текущего домена. В частности, при запуске браузера Netscape обычным пользовате-

лем (текущий домен user_t) произойдет перемещение в домен user_netscape_t, а при запуске этой же программы администратором (текущий домен sysadm_t) перемещение произойдет в другой домен – sysadm_netscape_t. Такой подход не позволит программе выполнить потенциально опасные действия – соответствующий домен серьезно ограничит права администратора. В обычных дистрибутивах Linux в случае с браузером Netscape проблема решалась проще – в настройке по умолчанию программа просто не запускалась с правами root.

Для администраторов в Security-Enhanced Linux заданы достаточно жесткие ограничения. В частности, нельзя зайти в систему удаленно – при необходимости такой процедуры необходимо сначала произвести вход обычным пользователем, а потом переключиться на административную роль при помощи команды newrole, производящей дополнительную аутентификацию. Впрочем, и в большинстве современных Unix-подобных систем администратору также запрещен удаленный вход и для этой цели используется команда su.

Пример с браузером не случаен – ему уделено особое внимание в перечне целей политики безопасности. Во избежание исполнения браузером вредоносного динамического кода (Java-апплеты, сценарии JavaScript) для него определен специальный домен (точнее, два домена – для пользователей и администраторов), ограничивающий полномочия. Причем определяются два подтипа: один ограничивает доступ браузера к локальным файлам только чтением, другой допускает запись в них.

Политика безопасности должна контролировать различные формы прямого доступа к данным, поэтому в ней определяются различные типы для устройств памяти ядра (kernel memory device), дисковых устройств, специальных файлов в каталоге /proc, а также различные домены для процессов, которым необходим доступ к этим ресурсам. Обеспечение целостности ядра достигается определением различных типов для загрузочных файлов, объектных файлов подключаемых модулей, утилит для работы с модулями и файлов конфигурации модулей. Соответствующим процессам, которым требуется право записи в эти файлы, назначаются специальные домены.

Системное ПО, файлы конфигурации и журналы также нуждаются в защите. Для этой цели также определяются отдельные типы для системных библиотек, исполняемых файлов, файлов конфигурации и журналов, а работающим с ними программам и ролям назначаются специальные домены. В результате запись журналов может вести только система регистрации (syslog), а модификация системного ПО производится только администраторами. Есть решение и для уже упомянутой проблемы, присущей программам с повышенными привилегиями (с установленными флагами suid или sgid). Для таких программ также назначаются отдельные домены, не позволяющие им превысить необходимые полномочия.

Политика безопасности уделяет особое внимание тщательному разделению процессов по привилегиям и защиту привилегированных процессов от выполнения ошибочного или опасного кода. Путем установки атрибута executable только на необходимые для исполнения привилегированным процессом программы политика безопасности может достичь того, что при входе в привилегированный домен процессу будет позволено выполнение только стартовой программы для данного домена, динамического компоновщика и системных разделяемых библиотек. Администратор ограничен выполнением системных программ и программ, созданных им самим – выполнение программ, созданных другими пользователями запрещено. Более того, система минимизирует взаимодействие между обычными пользовательскими процессами и системными. Только системным процессам и администраторам разрешен доступ к записям в procfs, относящимся к другим доменам; ограничено использование вызова ptrace по отношению к другим процессам и доставка сигналов между разными доменами. При использовании файловой системы также приняты дополнительные меры предосторожности: домашние каталоги администраторов и обычных пользователей отнесены к разным типам; создаваемым в каталогах общего пользования (/tmp и др.) файлам также присваиваются разные типы, в зависимости от создавшего их домена.

По сравнению с RSBAC система Security-Enhanced Linux менее гибкая, зато имеет очень хорошую предопределенную политику безопасности. Ее сложно применить для «небольшого усиления» стандартного дистрибутива Linux – настройка достаточно сложна и невозможна без изучения специального языка конфигурации. Но это не недостатки, а скорее следствия целей создания системы. В [8] отмечается, что не стоит пытаться создать на базе идеологии Type Enforcement операционную систему общего назначения, которая автоматически обеспечит надежность, безопасность и другие свойства любым запускаемым в ней приложениям. Type Enforcement – это механизм, позволяющий произвести интеграцию приложений и менеджера ресурсов, сведя при этом к минимуму присущие им недостатки. Это средство построения специализированных защищенных систем, в которых все лишние функции убраны, а поведение оставленных жестко определено матрицей Type Enforcement.

4 Новый стандарт

NIST-ом (Национальный Институт Сертификации и Технологий) ведутся разработки защищенного профиля (ЗП) для операционных систем, который называется «CSPP-OS – COTS Security Protection Profile – Operating Systems» [9]. На данное время существует черновая версия v.0.4 от 5 февраля 2001г. Этот ЗП разработан на основе руководства CSPP [10]. Ниже приводится краткий обзор данного документа.

4.1 Назначение

Назначение CSPP-OS состоит в том, чтобы определить и объяснить требования, необходимые для решения проблемы защиты операционных систем COTS (коммерческого назначения).

4.2 Возможности

Тип системы. CSPP-OS предусматривает требования, необходимые для определения потребностей операционных систем и в автономных, и в распределенных, и в многопользовательских информационных системах.

Тип доступа. CSPP-OS признает две формы легитимного доступа, а именно: общественный (публичный) и «аутентифицированные пользователи». При общественном доступе пользователь не имеет уникальный идентификатор и не аутентифицируется до получения доступа. Пример – доступ к информации относительно публично доступной web-страницы. Такие пользователи имеют законный доступ, но отличаются от «аутентифицированных пользователей», которые:

1) уникально распознаются системой;
2) имеют законный доступ к информации, отличной от общедоступной;
3) аутентифицируются до предоставления такого доступа (2).

Характер использования. CSPP-OS-операционные системы применяются для защиты информации в реальных мировых средах: и коммерческих, и правительственных.

- В контексте правительственной среды CSPP-OS рассматриваются как подходящие для определения базовых требований защиты для «чувствительной-но-несекретной» или одноуровневой секретной информации в среде, где все заверенные пользователи имеют доступ к уровню секретности обрабатываемой информации. Для секретной среды общественный доступ в CSPP-OS запрещен. Для «чувствительной-но-несекретной» среды общественный доступ может использоваться с дополнительным средством управления, вне механизмов ОС, предоставленных операционной средой.

- В контексте коммерческой среды CSPP-OS рассматриваются как подходящие для определения базовых требований защиты для информации в средах, где все аутентифицированные пользователи:
    1. Доверенные, т.е. злонамеренно не пытаются ни войти в систему, ни обходить средство управления доступа.
    2. Не проявляют наклонностей или способностей к сложным попыткам проникновения.

Общественный доступ позволяется со средством управления окружающей средой и находящимися вне ОС механизмами защиты.

Ключевые предположения. Ключевые предположения, которые применяются к CSPP-OS:

- Цель оценивания (TOE – ОС, для которой определяются требования) представляет собой информационную технологию (ИТ) COTS (коммерческого назначения).
- Заверенные пользователи признают потребность в защищенной ИТ.
- Заверенным пользователям можно разумно доверять, чтобы правильно применять политику безопасности организации в их контролируемых действиях.
- Администрирование защиты выполнено компетентно.
- Автоматизация процесса бизнеса (цели) осуществлена с должным отношением к тому, что не может быть предусмотрено в CSPP-OS.

4.3 Обзор требований CSPP-OS

Системы, входящие в число операционных систем COTS, достигают таких преимуществ, как высокий спрос на продукт – например, сочетанием высоких функциональных возможностей с низкой стоимостью. Однако эти преимущества не могут быть достигнуты без некоторых потерь; например – способность защиты. CSPP-OS отождествляют собой рентабельность, базовый уровень безопасности для систем, построенных на основе COTS ОС-м, гарантируя, что был достигнут разумный уровень защиты.

CSPP-OS также затрагивают те области, где нереально ожидать, что типичная операционная система COTS обеспечит достаточную защиту. Эти области – прямой результат того факта, что управляющие факторы COTS (функциональные возможности, стоимость, и время нахождения на рынке) имеют тенденцию работать против увеличения способностей защиты, оговариваемые в CSPP-OS.

Гарантия. CSPP-OS гарантии были выбраны, чтобы обеспечить уровень доверия, следующий из:

1) существующих лучших методик разработки COTS;
2) не всеобщей (а следовательно дорогостоящей) оценки третьей стороной.

Это приводит, в результате, к техническим контрмерам ОС-м, которые:

- Являются достаточными, чтобы управлять сообществом доверенных (то есть непреднамеренно злоумышленных) заверенных пользователей.
- Могут предусматривать защиту против несложных (простых) технических нападений.
- Не предусматривают достаточную защиту против сложных (искушенных), технических нападений (включая отказ в обслуживании).

Функциональные возможности. CSPP-OS-операционная система содержит следующие потребности пользователей:

- Принудительная политика управления доступом между активными сущностями (субъектами) и пассивными объектами, основанная на идентификации субъекта и позволенных для него действий.

- Обеспечение поддержки для управления доступом, основанном на свойствах среды типа времени-дня и точки входа.
- Сопротивление истощению ресурса путем внедрения контроля выделения ресурсов.
- Обеспечение механизмов обнаружения некоторых ненадежностей.
- Обеспечение механизмов для доверенного восстановления в случае некоторых системных отказов или обнаруженной ненадежности.
- Поддержка этих способностей в распределенной системе, связанной через неавторизованную сеть

CSPP-OS не предусматривают следующего:

- Обеспечения средств управления на основе меток, свойственных защите управляемой информации (правительственная секретная информация, собственность компаний или экспортируемые ограниченные данные) в средах, содержащих заверенных пользователей, которым не позволяют доступ к такой информации.
- Защиты против злоупотребления полномочиями.
- Адекватной защиты против сложных нападений (включая отказ в обслуживании).
- Обеспечения достаточной защиты против инсталляции, эксплуатации или ошибок администрирования.

4.4 Требуемые функциональные возможности защиты

CSPP-OS определяет требования для операционной системы к функциональным возможностям защиты, перечисленные ниже:

- Выполнение политики управления доступом, продиктованные политикой безопасности ИТ.
- Назначение уникального идентификатора каждому заверенному пользователю.
- Назначение уникального идентификатора каждому системному процессу, включая те, которые выполнены не от имени пользователя (например, процессы, запущенные в системе bootup подобно Unix «inetd»).
- Подтверждение требуемых полномочий перед разрешением любому пользователю действий, отличных от общеизвестного набора операций (например, чтение с общественного сайта).
- Аудит для поддержки индивидуальной ответственности и обнаружения причин ненадежностей и неполадок.
- Предоставление управления разрешениями доступа – то есть инициализация, назначение и модификация прав доступа (например чтение, запись, выполнение) к объектам данных относительно:
    1) названия (имени) объекта или принадлежности к группе;
    2) свойств среды, как-то системного времени и точки входа.
- Особенности распределения ресурсов, обеспечивающие меру сопротивления к их истощению.
- Механизмы для обнаружения некоторой ненадежности.
- Системные особенности восстановления, обеспечивающие меру живучести в контексте системных отказов и ненадежности.
- Автоматизированная поддержка для помощи в проверке безопасной доставки, инсталляции, эксплуатации и администрирования.

5 Заключение

Исследование всего вышесказанного позволило сделать следующие выводы о недостатках Secure Patches (SP):

- **Новизна** – при создании SP, и RSBAC, и Security-Enhanced Linux разработчики руководствовались уже устаревшими на сегодняшний день требованиями к защищенным ОС (например, RSBAC основывается на «Оранжевой книге»). Таким образом, эти SP не удовлетворяют всем требованиям к защищенности, которые существуют сегодня.
- **Сложность** – SP сильно усложнили архитектуру ОС, а также работу с ней. Например, защищенная ОС может заключить приложения в непроницаемые «скафандры», и в этом случае у системного администратора может возникнуть впечатление, что в приложении возник сбой, когда всего-навсего ему не предоставлено право данное приложение контролировать.
- **Совместимость версий** – неизвестно, будет ли работать SP при выходе следующей версии ядра ОС. Таким образом, для каждой версии ядра необходимо будет и обновлять SP.
- **Стоимость** – SP должны использовать лишь тогда, когда даваемые ими преимущества оправдывают затраты на обучение и время, которое приходится тратить на их сопровождение. Например, SP семейства PitBull, которые улучшают защиту Sun Solaris, IBM AIX и Linux, – стоят от 5 тыс. долл. за ОС, работающую на однопроцессорном Web-сервере, до 50 тыс. долл. за развертывание в рамках всей корпоративной информационной системы. При обновлении каждой версии ядра ОС, которые выходят примерно раз в полгода – это выльется в огромную сумму.
- **Проверка временем** – SP сложны в реализации и должно пройти довольно много времени, чтобы оно послужило доказательством работоспособности данного ПО.

Вышеуказанные недостатки SP позволяют сформулировать следующее: необходимо отказаться от практики «латания дыр» и начать строить новую ОС, изначально удовлетворяющую требованиям безопасности. Предлагается использовать руководящий документ, рассмотренный в (4) для создания изначально защищенной ОС. В качестве фундамента можно использовать ядро Linux – linux-2.4.18. Это перекликается с наметившимся сегодня интересом к достоверным (trusted) и защищенным (secure) операционным системам. Требования к безопасности должны быть определяющими в проектировании ОС, а не вводиться как вспомогательные службы.

Список литературы: 1. *2001* Computer Crime and Security Survey // Computer Security Institute, San Francisco, March 12, 2001; www.gocsi.com/prelea_000321.htm. 2. *Common* Criteria for Information Technology Security Evaluation (CCITSE) V2.1 // 1998. 3. *Станислав Иевлев.* Начала RSBAC. http://linux.ru.net/~inger/RSBAC-DOC-ru.html 4. Rule Set Based Access Control (RSBAC) для Linux – Модели // http://www.asmodeus.com.ua/library/os/linux/rsbac/models.html 5. *Simone Fischer-Hubner, Amon Ott.* From a Formal Privacy Model to its Implementation, http://www.rsbac.org/niss98.htm 6. *Security-Enhanced* Linux. http://www.nsa.gov/selinux/index.html 7. *Peter Loscocco, Stephen Smalley.* Integrating Flexible Support for Security Policies into the Linux Operating System. http://www.nsa.gov/selinux/slinux-abs.html 8. *Earl Boebert,* Some thoughts on the occasion of the NSA Linux release. http://www2.linuxjournal.com/articles/buzz/0043.html 9. *CSPP-OS – COTS* Security Protection Profile – Operating Systems // National Institutes of Standards and Technology, 2001. 10. *CSPP – Guidance* for COTS Security Protection Profiles // National Institutes of Standards and Technology, 1999.

Харьковский национальный университет радиоэлектроники

Поступила в редколлегию 10.05.2003