УДК 004.451; 004.72:004.451

Управление доступом в ОС GNU/Linux

А. М. Каннер
Закрытое акционерное общество "ОКБ САПР", Москва, Россия

Л. М. Ухлинов, д-р техн. наук
Открытое акционерное общество «Концерн "Сириус"», Москва, Россия

Рассмотрены проблемы разграничения доступа в ОС GNU/Linux, связанные с особенностями семейства ОС.

Ключевые слова: управление доступом, ОС GNU/Linux, списки контроля доступа, изолированная программная среда.

Операционные системы семейства Linux (правильнее называть — семейство GNU/Linux) изначально получили широкое распространение как серверные ОС, однако сейчас ОС этого семейства постепенно все больше входят в жизнь каждого человека.

Ключевыми особенностями, из-за которых ОС семейства GNU/Linux становятся все популярнее, являются [1]:

- открытость исходных кодов большинства компонент системы (за исключением коммерческих, которые могут присутствовать в отдельных дистрибутивах). Лицензии по свободному использованию GNU/Linux позволяют беспрепятственно изменять исходные коды, что позволяет подстраивать ОС под выполнение конкретных задач. Кроме того, существует достаточно крупное сообщество GNU/Linux, оказывающее качественную и быструю поддержку ОС на базе ядра GNU/Linux, при этом процесс выпуска и утверждения обновлений строго систематизирован;
- наличие большого количества разнообразных дистрибутивов GNU/Linux и отсутствие единой принятой комплектации — т. е. ОС поставляется в виде ядра GNU/Linux (единое практически для всех дистрибутивов с точностью до релиза конкретного дистрибутива) и комбинацией многообразных прикладных программ и системных утилит. При этом пользователь может выбрать систему, удовлетворяющую исключительно его потребностям.

Крылатое выражение "*In UNIX everything is a File, unless it isn`t*" описывает одну из определяющих черт ОС Unix и ОС, производных от нее, в том числе и GNU/Linux (далее такое множество будем для краткости обозначать как ***nix**) — за небольшим исключением (например, таких сущностей как сокеты и т. п.), любая сущность (объект) ОС представляется на уровне ОС в виде файла в дереве каталогов файловой системы (ФС). Что такое файл в *nix? Вообще говоря, это просто набор бит, причем какого-либо признака формата у файла может и не быть — *чтобы понять (распознать) его содержимое нужно знать, что за данные в нем содержатся*. Таким образом в *nix-системах:

- физические диски – есть файлы, так называемые файлы блочных устройств (располагаются в каталоге /dev), причем для передачи данных какому-либо устройству нет необходимости передавать их через драйвер – достаточно записать данные в файл устройства с помощью стандартной функции write (при этом система сама отошлет эти данные драйверу устройства);
- различные порты ввода/вывода – также есть файлы;
- обычные файлы (имеются ввиду пользовательские или системные файлы, содержащие какие-либо данные) – также есть файлы в ФС, за тем отличием от описанных выше объектов, что эти файлы имеют нефиксированный размер (его можно изменять), а также такие файлы можно читать непоследовательно из произвольных мест и т. д.

Очевидный плюс такого подхода к построению системы в том, что операции с любой сущностью (объектом) ОС производятся одинаково — точно так же как с обычным файлом. А для использования в своих программах каких-либо устройств не нужно разрабатывать какой-либо дополнительный интерфейс взаимодействия (правда в данном случае речь идет скорее только о данных, которые не имеют какого-либо специфичного формата, так как, например, отправить на печать изображение формата JPEG простой его записью в файл принтера не получится, для этого необходимо преобразовать данные в понятный для принтера формат, например, Postscript, что производится с помощью специального прикладного ПО). Другими словами в ОС *nix имеется некоторый *механизм абстракции представления объектов в ОС от механизмов взаимодействия с этими объектами*.

Руководствуясь такой парадигмой становится очевидным, что для построения подсистемы разграничения доступа в *nix необходимо контролировать как минимум такие операции как открытие файловых объектов на чтение/запись/выполнение,

создание/удаление/переименование файловых объектов, переход в файловые объекты (в случае каталогов) и т. п. При этом по умолчанию контроль доступа к объектам ОС *nix может осуществляться штатными механизмами разграничения доступа, такими как:

- механизм дискреционного разграничения доступа (discretionary access control — DAC);
- механизм, реализующий списки контроля доступа (СКД или ACL — Access Control Lists);
- дополнительные средства разграничения доступа, позволяющие реализовать в том числе механизм мандатного разграничения доступа (mandatory access control — MAC), например, SELinux, Tomoyo, AppArmor и пр. (данные средства не входят в штатную комплектацию ОС *nix).

Механизм дискреционного разграничения доступа является основным механизмом контроля доступа в ОС *nix. Данный механизм реализуется в виде прав доступа для файлов (объектов) ОС, при этом права доступа разделяются на:

- права владельца файла;
- права группы, владеющей файлом;
- права для остальных субъектов.

Кроме прав доступа на объекты ОС (права на чтение/запись/исполнение для владельца/группы/прочих субъектов доступа) в ОС *nix существует механизм, реализующий списки контроля доступа (СКД), который по сути позволяет расширить традиционные права доступа и настраивать эти права доступа "индивидуально" на каждый объект ОС (т. е. указывать какой субъект доступа и что имеет/не имеет право делать с данным объектом). Существенным минусом СКД является невозможность их использования на всех ФС, поддерживаемых *nix, а также возможность отключения данного механизма на этапе загрузки ОС (например, через параметр ядра ОС acl_mode или с помощью опции no_acl в /etc/fstab, учитывающийся при монтировании определенного раздела).

Дополнительные средства защиты, позволяющие реализовать механизм мандатного разграничения доступа, страдают тем же изъяном, как правило, они требуют возможности в ФС создавать для файловых объектов определенные метаданные, при этом такие средства также существует возможность отключить на раннем этапе загрузки ОС.

Наверное, общим недостатком для всех указанных средств разграничения доступа является то, что при наличии физического доступа к носителю информации, на котором размещаются защищаемые данные, можно считать, что эти данные в общем случае никак не защищены.

Учитывая все вышеописанные недостатки штатных и внешних средств разграничения доступа в ОС *nix и в соответствии со следствиями из теоремы об использовании компонента безопасности (ИКБ) [2], в подсистему разграничения доступа должен быть включен аппаратный компонент. Основная идея здесь сводится к тому, что без использования внешнего по отношению к системе резидентного компонента безопасности (именно аппаратного) невозможно построить решение (только программное) по контролю доступа к информации. Данный аппаратный компонент призван в первую очередь осуществлять:

- идентификацию и аутентификацию (и/а) пользователя до начала загрузки ОС;
- контроль загрузки ОС с заранее определенного носителя информации;
- контроль неизменности оборудования ПК;
- контроль целостности загрузочных секторов носителя данных и отдельных разделов;
- контроль целостности определенных системных файлов ОС.

В случае успешного прохождения всех контрольных процедур, выполняемых самим резидентным компонентом (независимо от аппаратных и программных составляющих ПК) должна продолжаться штатная загрузка ОС и подсистемы разграничения доступа в ОС. При всем при этом сама подсистема разграничения доступа должна уметь взаимодействовать с данным аппаратным компонентом с целью интеграции его защитных механизмов с механизмами разграничения доступа в ОС.

В случае же выявления какого-либо несоответствия при проведении контрольных процедур – аппаратный компонент должен записывать соответствующее событие в свой журнал и блокировать дальнейшую загрузку ОС. Таким образом, становится возможным "обезопасить" подсистему разграничения доступа от отключения или изменения ее настроек на ранних этапах загрузки ОС.

Сама же подсистема разграничения доступа в свою очередь должна обеспечивать создание изолированной программной среды (ИПС), т. е. среды, в которой каждому пользователю доступен только определенный набор возможных для исполнения процессов (программ, для которых контролируется целостность), при этом для каждого процесса (как и для пользователя) ограничен круг доступных объектов ОС, а также в которой исключена возможность запуска процессов вне такой среды пользователя [6].

Рассмотрим основные принципы построения подсистемы разграничения доступа на примере программно-аппаратного комплекса средств защиты информации от несанкционированного доступа (ПАК СЗИ НСД) "Аккорд-X", разработанного компанией ОКБ САПР [4]. Функционально ПАК СЗИ НСД "Аккорд-X" состоит из нескольких модулей:

- монитора разграничения доступа (МРД), который должен запускаться в момент старта ОС до монтирования корневой файловой системы на запись (осуществляет основной функционал по раз-

граничению доступа субъектов доступа к объектам ОС);

• подсистемы и/а пользователей в ОС, реализованной в виде PAM-модуля (производит и/а пользователей в связке с МРД);

• подсистемы статического контроля целостности (производит статический контроль целостности при получении указания от МРД);

• подсистемы журналирования (производит запись событий безопасности, полученных от МРД в специальный журнал);

• подсистемы контроля печати (осуществляет контроль печати на уровне подсистемы печати ОС во взаимодействии с МРД);

• утилит администрирования комплекса (служат для редактирования БД пользователей МРД, а также для прочих настроек комплекса).

Ключевым элементом подсистемы разграничения доступа является МРД, который выполнен в виде модуля ядра ОС. Данный модуль фактически выполняет весь основной функционал подсистемы разграничения доступа, а именно:

• в момент загрузки регистрирует собственные перехватчики основных системных вызовов в ядре ОС, тем самым, позволяя организовывать дискреционный и/или мандатный механизмы доступа в ОС (при этом права доступа могут задаваться как для пользователей, так и для отдельных процессов);

• всем процессам и пользователям на уровне ядра ОС присваивает определенные метки конфиденциальности (начиная с момента своей загрузки);

• контролирует права при монтировании разделов в соответствии с правами разграничения доступа (ПРД) в своей БД, при этом после монтирования дополнительно может проводиться статический контроль целостности в соответствии с системным списком контроля целостности из БД пользователей;

• при аутентификации пользователя в ОС осуществляет взаимодействие с подсистемой и/а пользователей (PAM-модулем) и проверку введенных пользователем данных с данными в собственной БД пользователей;

• по окончании сессии пользователя сбрасывает признак аутентифицированности пользователя в собственных структурах с целью невозможности выполнения в ОС процессов от имени данного пользователя;

• производит статический и динамический контроль целостности данных:

статический контроль целостности осуществляется сразу после успешной процедуры и/а пользователя;

динамический контроль целостности осуществляется каждый раз при загрузке объекта доступа в память (производится самим МРД "на лету");

• при перехвате любого системного вызова на получение доступа к объекту ОС проверяет ПРД в соответствии с собственной БД пользователей;

• фиксирует все события безопасности и передает их в подсистему журналирования.

Загрузка МРД в ядро ОС должна осуществляться на раннем этапе загрузки операционной системы в связи с этим общий порядок загрузки ПК должен выглядеть следующим образом:

• при включении ПК управление передается штатному BIOS, который выполняет анализ и проверку подключенного оборудования;

• по окончании работы BIOS выполняется процедура RomScan, по окончании которой управление передается аппаратному компоненту ПАК СЗИ НСД "Аккорд-X" — аппаратному модулю доверенной загрузки "Аккорд-АМДЗ":

на данном этапе для корректной работы ПАК СЗИ НСД "Аккорд-X" необходимо дополнительно кроме прочих проверок контролировать целостность:

образа начальной загрузки системы initrd (МРД "Аккорд-X" запускается из скрипта инициализации образа initrd до выполнения /sbin/init);

ядра ОС vmlinux;

МРД в виде файла на диске;

БД "Аккорд-X" (в дальнейшем контроль БД осуществляется самим МРД, т. е. фактически БД пользователей после загрузки МРД защищает сама себя);

прочих модулей ПАК СЗИ НСД "Аккорд-X".

• после успешного прохождения контрольных процедур "Аккорд-АМДЗ" управление передается загрузчику ОС (который записан в boot-сектор загрузочного раздела /boot);

• загрузчик загружает ядро ОС и образ начальной загрузки в память, распаковывает образ начальной загрузки, загружает ядро ОС;

• в ядро ОС загружается МРД;

• ядро ОС запускает сценарий /sbin/init (первый пользовательский процесс);

• корневая файловая система монтируется на запись, запускаются необходимые службы ОС, монтируются дополнительные файловые системы в необходимые каталоги;

• после описанных выше шагов система остается в режиме ожидания дальнейших действий (запускается утилита login для и/а пользователя).

Как можно видеть, ключевым моментом в процессе загрузки является контроль целостности образа начальной загрузки, ядра ОС, файла с МРД, БД пользователей и прочих модулей ПАК СЗИ НСД "Аккорд-X" средствами аппаратного модуля доверенной загрузки "Аккорд-АМДЗ". Организовав именно таким образом загрузку подсистемы разграничения доступа, становится невозможным каким-либо образом отключить или изменить настройки "Аккорд-X", в том числе на ранних этапах загрузки системы.

Выводы

1. Встроенных (штатных) средств разграничения доступа в ОС семейства *nix недостаточно для полноценной защиты от НСД к информации.

2. При использовании только программных СЗИ практически всегда существуют способы их обхода, в связи с этим, наряду с программными компонентами, необходимо применять аппаратную часть, ликвидирующую недостатки только программных средств.

3. Использование аппаратного компонента в СЗИ НСД является первой и, по большому счету, ключевой фазой при создании ИПС работы пользователя – в данном случае обеспечивается целостность аппаратной составляющей рабочей станции, а также программной составляющей в лице ОС и критичных файлов подсистемы разграничения доступа (т. е. осуществляется доверенная загрузка ОС и ее компонент).

4. Загруженная после успешной отработки аппаратной части подсистема разграничения доступа в ОС реализует вторую фазу создания ИПС работы пользователя, а именно, внедрение механизмов дискреционного и мандатного разграничения доступа субъектов доступа (пользователей, процессов, псевдо-пользователей shadow, которые обеспечивают работу определенных служб от своего имени), статический и динамический контроль целостности данных, контроль печати, ведение журнала событий НСД и т. п. При этом работа указанных механизмов подсистемы разграничения доступа изначально доверенная, учитывая п. 3.

5. Целостный комплекс СЗИ в виде ПАК СЗИ НСД "Аккорд-X" производства ОКБ САПР, в котором реализован описанный выше подход в реализации подсистемы разграничения доступа в ОС *nix, соответствует требованиям, предъявляемым к СЗИ НСД в соответствии с руководящими документами (РД) ФСТЭК России для 3-го класса защищенности СВТ [3] и для класса защищенности АС 1Б [5].

Литература

1. http://en.wikipedia.org/wiki/Unix_architecture
2. *Конявский В. А.* Управление защитой информации на базе СЗИ НСД "Аккорд". – М.: Радио и связь, 1999. С. 50.
3. Руководящий документ "Средства вычислительной техники. Защита от несанкционированного доступа к информации. Показатели защищенности от несанкционированного доступа к информации"//Гостехкомиссия России, 1992.
4. *Бажитов И. А.* Возможности ПАК СЗИ НСД "Аккорд-X" для ОС Linux//Комплексная защита информации: Сб. матер. XIV Междунар. науч.-практ. конф. (19–22 мая 2009 г.). – Мн., 2009. С. 26, 27.
5. Руководящий документ "Автоматизированные системы. Защита от несанкционированного доступа к информации. Классификация автоматизированных систем и требования по защите информации"//Гостехкомиссия России, 1992.
6. *Бажитов И. А.* Обеспечение доверенной среды в ОС Linux с использованием ПАК СЗИ НСД "Аккорд-X"// Комплексная защита информации: Сб. матер. XV Междунар. науч.-практ. конф. (1–4 июня 2010 г., Иркутск). – М., 2010. С. 32.

Access control in GNU/Linux

A. M. Kanner

Closed Joint Stock Company "OKB SAPR", Moscow, Russia

L. M. Uhlinov

Open Joint Stock Company «Concern "Sirius"», Moscow, Russia

Highlights the problem of access control in OS GNU/Linux, associated with the features of such OS family.

Keywords: access control, OS GNU/ Linux, access control lists, sandbox.

Каннер Андрей Михайлович, программист группы программирования ядра СЗИ.
E-mail: kanner@okbsapr.ru
Ухлинов Леонид Михайлович, профессор, генеральный директор.
E-mail: info@con-sirius.ru