УДК 004.056.53

Окороков В. А.

ЗАЩИЩЕННЫЕ ОПЕРАЦИОННЫЕ СИСТЕМЫ

Рассмотрена роль операционных систем (ОС) в организации систем защиты информации. Перечислены основные требования и механизмы, которые необходимо реализовать в рамках операционной системы для обеспечения надежной защиты данных. Изучена возможность применения ОС Astra Linux в качестве базовой ОС, предназначенной для использования в учебном процессе по направлению «Информационная безопасность».

Ключевые слова: *информация, безопасность, операционные системы, защита данных, обучение.*

Okorokov V. A.

SECURE OPERATING SYSTEM

The role of the operating system (OS) in the organization of information security systems. The basic requirements and mechanisms to be implemented within the operating system to ensure reliable data protection. The possibility of using the OS Astra Linux as the base OS for use in the educational process in the direction of "Information Security".

Keywords: *information, security, operating systems, data protection, training.*

Проблема обеспечения информационной безопасности носит комплексный характер и для ее решения требуется разработка организационных и технических мер, которые должны поддерживаться в рамках любой организации, связанной с использованием информационных систем.

Базовым подходом к обеспечению информационной безопасности является ограничение доступа субъектов информационных отношений к данным. Меры по разграничению доступа могут быть эффективны только в том случае, если внутри предприятия существует и исполняется некоторая политика, определяющая порядок доступа пользователей к данным. В рамках политики безопасности необходимо для каждого субъекта определить разрешенные объекты данных и методы доступа к ним.

Реализация политики ограничения доступа в рамках информационной системы невозможна без применения программно-технических средств, позволяющих определить, имеет ли право определенный пользователь получать требуемый вид доступа к заданным объектам данных. Если соответствующие программные компоненты находятся в привилегированном положении по отношению к программам пользователя, то это существенно усложняет их модификацию или подмену. Единственной программой, которая работает в привилегированном режиме, является операционная система. Поэтому именно операционная система должна нести ответственность за поддержку основных механизмов обеспечивающих реализацию политики безопасности [6].

Базовые механизмы

Детали политики безопасности организации могут существенно изменяться с течением времени. Соответствующие изменения операционной системы не всегда возможны. Поэтому программно-технические средства поддержки безопасности разумно разделить на две группы [1].

К первой группе относятся базовые механизмы безопасности, которые требуют привилегированного доступа к системным ресурсам. Базовые механизмы обычно реализуются в виде модулей операционной системы, работающих в режиме ядра и использующих системные структуры данных, также находящиеся в области памяти ядра.

Ко второй группе относятся детальные механизмы ограничения доступа, реализующие текущую политику безопасности, принятую в организации. Поддержка таких механизмов выполняется с помощью программ пользовательского уровня, что позволяет относительно легко их модифицировать в соответствии с новыми требованиями. Программы пользовательского уровня, предназначенные для реализации политики безопасности, должны иметь доступ к базовым механизмам безопасности операционной системы. Такой интерфейс обычно реализуется с помощью набора соответствующих системных вызовов, которые принято называть системными вызовами безопасности.

Следует также иметь в виду, что сама операционная система может быть объектом атак злоумышленников. Основным методом противостояния таким атакам является использование специальной архитектуры операционной системы, подразумевающей локализацию всех базовых механизмов безопасности в рамках единой подсистемы, обладающей повышенной надежность работы [6].

Совокупность средств, обеспечивающих информационную безопасность вычислительной системы, обычно объединяется в рамках комплекса средств защиты (КСЗ), включающего как модули ядра ОС, так и программы, работающие на уровне пользователя.

Основные функции КСЗ включают [4]:

- идентификация и аутентификация пользователя;
- дискреционное разграничение доступа к ресурсам;
- мандатное разграничение доступа к ресурсам;
- контроль повторного использования объектов;
- протоколирование событий;
- надежное восстановление;
- контроль состояния системы безопасности.

Далее рассмотрим различные аспекты перечисленных функций и особенности их реализации в рамках специализированной ОС Astra Linux [4].

Идентификация и аутентификация

Функция идентификации и аутентификации пользователей в ОС Astra Linux основывается на использовании механизма PAM (Pluggable Authentication Modules). Данный механизм представляет собой набор разделяемых библиотек, с помощью которых системный администратор может организовать процедуру аутентификации пользователей прикладными программами. Каждый модуль реализует собственный механизм аутентификации. Изменяя набор и порядок следования модулей, можно построить произвольный сценарий аутентификации. Подобный подход позволяет изменять процедуру аутентификации без изменения исходного кода и повторного компилирования PAM.

Сценарии аутентификации описываются в конфигурационном файле /etc/pam.conf, а также в конфигурационных файлах, расположенных в каталоге /etc/pam.d/. Модули PAM располагаются в каталоге /lib/security в виде динамически загружаемых объектных файлов.

Предусмотрена возможность аутентификации как в рамках локального компьютера, так и в сети. В локальной системе, аутентификация осуществляется с помощью локальной БД пользователей (/etc/passwd и т.д.).

При работе в сети аутентификация пользователей осуществляется централизованно по протоколу Kerberos [3], а в качестве источника данных для идентификации и аутентификации пользователей применяются службы каталогов LDAP (Lightweight Directory Access Protocol). Вся служебная информация пользователей может располагаться на выделенном сервере в распределенной гетерогенной сетевой среде. Сетевые сервисы, поддерживающие возможность аутентификации пользователей (web, FTP, почта), могут вместо локальных учетных записей использовать тот же каталог LDAP.

Благодаря предоставлению информации LDAP в иерархической древовидной форме разграничение доступа в рамках службы каталогов LDAP может быть основано на введении доменов. Сервисы LDAP позволяют разграничивать доступ пользователей к разным поддеревьям каталога, хотя по умолчанию в ОС реализуется схема одного домена.

Дискреционное разграничение доступа

В ОС Astra Linux реализован механизм дискреционного разделения доступа [4], ко-

торый заключается в том, что на защищаемые именованные объекты устанавливаются базовые права доступа в виде идентификаторов номинальных субъектов (UID и GID), которые вправе распоряжаться доступом к данному объекту, и прав доступа к объекту. Определяются три вида доступа: чтение (r), запись (w) и исполнение (x). Права доступа включают список из девяти пунктов: по три вида доступа для трех групп – пользователя-владельца, группы-владельца и всех остальных. Сопоставляя определенные виды доступа для каждой пары субъект-объект, можно описать полные правила разграничения доступа в рамках информационной системы.

При обращении процесса к объекту система проверяет совпадение идентификаторов владельцев процесса и владельцев файла, и, в зависимости от результата, применяет ту или иную группу прав.

Права доступа файлового объекта могут быть изменены, если это разрешено текущими правилами.

Кроме общей схемы разграничения доступа, ОС поддерживает также списки ACL (Access Control List), с помощью которых можно для каждого объекта индивидуально задавать права доступа к нему всех субъектов.

Объектами доступа являются:

- файлы;
- соединения (сокеты);
- сетевые пакеты;
- механизмы IPC.

Механизм, реализующий дискреционное разграничение доступа, обеспечивает возможность санкционированного изменения списка пользователей и списка защищаемых файловых объектов.

Мандатное разграничение доступа

Механизм контроля мандатного разграничения доступа реализован, как и механизм дискреционного разграничения доступа, в ядре ОС Astra Linux4. При этом, принятие решения о запрете или разрешении доступа субъекта к объекту принимается на основе типа операции (r\w\x), мандатного контекста безопасности субъекта и мандатной метки объекта. Кроме того, при принятии решения могут учитываться полномочия субъекта.

Правила принятия решения могут быть записаны следующим образом. Пусть контекст безопасности субъекта содержит уровень L0 и категории C0, а мандатная метка объекта содержит уровень L1 и категории C1. Определим операции сравнения для уровней и категорий:

1) уровень L0 меньше уровня L1 (L0<L1), если численное значение L0 меньше численного значения L1;
2) уровень L0 равен уровню L1 (L0=L1), если численные значения L0 и L1 совпадают;
3) категории C0 меньше категорий C1 (C0<C1), если все биты набора C0 являются подмножеством набора бит C1;
4) категории C0 равны категориям C1 (C0=C1), если значения C0 и C1 совпадают;
5) операция записи разрешена, если L0=L1 и C0=C1;
6) операция чтения разрешена, если L0≥=L1 и C0≥C1;
7) операция исполнения разрешена, если L0≥L1 и C0≥C1.

В остальных случаях анализируются полномочия и тип мандатной метки. Тип метки может использоваться для того, чтобы изменять ее эффективное действие. Ненулевой тип метки может быть установлен только привилегированным процессом.

Механизм мандатного разграничения доступа затрагивает следующие подсистемы:

- механизмы IPC;
- стек TCP/IP (IPv4);
- файловые системы Ext2/Ext3/Ext4;
- сетевые файловые системы.

С каждым субъектом и объектом связаны: мандатный контекст безопасности и мандатная метка, соответственно.

При создании субъектом объект наследует метку на основе мандатного контекста безопасности процесса.

Контроль повторного использования объектов

Ядро ОС Astra Linux гарантирует, что обычный непривилегированный процесс не получит данные чужого процесса, если это явно не разрешено правилами разграничения доступа (ПРД). Это означает, что средства IPC контролируются с помощью ПРД и процесс не может получить неочищенную память (как оперативную, так и дисковую).

В ОС реализован механизм, который очищает неиспользуемые блоки ФС непосредственно при их освобождении. Работа названного механизма снижает скорость выполнения операций удаления и усечения размера файла. Механизм является настраивае-

мым и позволяет обеспечить работу ФС ОС (Ext2/Ext3/Ext4) в одном из следующих режимов:

- данные любых удаляемых/урезаемых файлов в пределах заданной ФС предварительно очищаются маскирующей последовательностью;
- данные ФС освобождаются обычным образом (без предварительного маскирования).

Ядро ОС обеспечивает для каждого процесса в системе собственное изолированное адресное пространство, на основе применения механизмов страничной организации памяти, а также трансляции виртуального адреса в физический. Одни и те же виртуальные адреса преобразуются в разные физические для разных адресных пространств процессов. Процесс не может несанкционированным образом получить доступ к адресному пространству другого процесса, т.к. он лишен возможности работать с физической памятью напрямую.

Механизм разделяемой памяти является санкционированным способом получить нескольким процессам доступ к одному и тому же участку памяти и находится под контролем дискреционных и мандатных ПРД.

Надежное восстановление

Основными причинами нарушения функционирования ОС являются сбои оборудования, приведшие к различным повреждениям файловой системы (ФС). К таковым относятся: сбои электропитания, повреждения носителей информации (жестких дисков), повреждения соединительных кабелей.

В процессе перезагрузки после сбоя автоматически выполняется программа проверки и восстановления ФС – fsck. Если повреждения ФС окажутся незначительными, то ее выполнения достаточно для обеспечения целостности ФС.

В случае обнаружения серьезных повреждений ФС данная программа может предложить перезагрузить компьютер в однопользовательский режим и произвести запуск программы fsck вручную. Администратор, контролирующий процесс загрузки ОС, после сбоя должен следовать инструкциям, выдаваемым программой fsck.

После завершения загрузки ОС следует проверить целостность файлов с помощью программы контроля целостности. Если в результате проверки найдутся поврежденные или измененные файлы, то следует восстановить поврежденные файлы с резервной копии.

Резервное копирование выполняется с целью получения копий данных, сохраняемых на случай их потери или разрушения. Подобные копии должны создаваться периодически, в соответствии с заранее установленным графиком.

Контроль целостности КСЗ

Для обеспечения контроля целостности (в т.ч. контроля целостности КСЗ) в ОС Astra Linux реализованы:

- средство подсчета контрольных сумм файлов и оптических дисков;
- средство контроля соответствия дистрибутиву;
- средства регламентного контроля целостности;
- средства создания замкнутой программной среды.

Для решения задач контроля целостности предназначена библиотека libgost, в которой для вычисления контрольных сумм реализована функция хэширования в соответствии с ГОСТ Р 34.11-20122. Названная библиотека используется в средствах подсчета контрольных сумм файлов и оптических дисков, контроля соответствия дистрибутиву и регламентного контроля целостности.

В ОС реализован механизм, обеспечивающий проверку неизменности и подлинности загружаемых исполняемых файлов. Проверка производится на основе контрольных сумм файлов.

Рассмотренные выше функции подсистемы безопасности ОС Astra Linux обеспечивают создание и функционирование защищенных информационных систем в соответствии с принятыми стандартами (см. например, приказ ФСТЭК от 18 февраля 2013 г. N 215). В рамках ОС поддерживаются все основные механизмы, предназначенные для реализации систем информационной безопасности. Данные механизмы имеют достаточно простой и интуитивно понятный интерфейс, обеспечивающий изучение их работы. Указанные обстоятельства позволяют рекомендовать к использованию ОС Astra Linux в качестве базовой системы в учебном процессе по направлению «Информационная безопасность».

Примечания

1. Безбогов А.А. Методы и средства защиты компьютерной информации: учебное пособие / А.А. Безбогов, А.В. Яковлев, В.Н. Шамкин. – Тамбов: Изд-во ТГТУ, 2006. – 120 с.

2. ГОСТ Р 34.11-2012 Информационная технология. Криптографическая защита информации. Функция хэширования. – М.: Стандартинформ, 2013. – 24 с.

3. Нестеров С. А. Информационная безопасность и защита информации: учебное пособие. / С. А. Нестеров СПб.: Изд-во Политехн. ун-та, 2009. - 126 с.

4. Операционная система специального назначения «Astra Linux special edition» Руководство по КСЗ. в 2 ч. Ч. 1: 2012. – 100 с.

5. Приказ ФСТЭК от 18 февраля 2013 г. N 21. Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных // Российская газета. – 2013. – 22 мая.

6. Таненбаум, Э. Современные операционные системы: монография / Э. Таненбаум. – СПб: Изд-во «Питер» 2011. – 1116 с.

Информация об авторах отсутствует на обоих языках