ФГБОУ ВПО «Воронежский государственный
технический университет»

Кафедра систем информационной безопасности

217-2015

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям по дисциплине
«Защита в операционных системах»
для студентов специальности
090301 «Компьютерная безопасность»
очной формы обучения

Воронеж 2015

Составители: д-р техн. наук А. Ю. Савинков, Н. А. Ленков, В. Н. Деревянко

УДК 004.056.5

Методические указания к практическим занятиям по дисциплине «Защита в операционных системах» для студентов специальности 090301 «Компьютерная безопасность» очной формы обучения / ФГБОУ ВПО «Воронежский государственный технический университет»; сост. А. Ю. Савинков, Н. А. Ленков, В. Н. Деревянко. Воронеж, 2015. 30 с.

Методические указания посвящены изучению задач политик безопасности операционных систем, а также закреплению студентами пройденного лекционного материала.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2013 и содержатся в файле Савинков_ПЗ_ЗОС.pdf.

Ил. 1. Библиогр.: 7 назв.

Рецензент д-р техн. наук, проф. А. Г. Остапенко

Ответственный за выпуск зав. кафедрой д-р техн. наук, проф. А. Г. Остапенко

Издается по решению редакционно-издательского совета Воронежского государственного технического университета

1. СРЕДСТВА РАЗГРАНИЧЕНИЯ ДОСТУПА К ОБЪЕКТАМ

Основу политики безопасности для компьютерной системы любой организации составляют правила разграничения доступа к объектам компьютерной системы. Разграничение доступа к компьютерным ресурсам базируется на различных моделях управления доступом. В данном докладе будут представлены результаты сравнительного анализа средств разграничения доступа к объектам операционных систем Microsoft Windows и Linux.

Дискреционная модель разграничения доступа предполагает назначение каждому объекту списка контроля доступа, элементы которого определяют права доступа к объекту конкретного субъекта. Правом редактирования дискреционного списка контроля доступа обычно обладают владелец объекта и администратор безопасности. Эта модель отличается простотой реализации, но возможна утечка конфиденциальной информации даже в результате санкционированных действий пользователей.

Мандатная модель разграничения доступа предполагает назначение объекту метки (грифа) секретности, а субъекту – уровня допуска. Доступ субъектов к объектам в мандатной модели определяется на основании правил «не читать выше» и «не записывать ниже». Использование мандатной модели, в отличие от дискреционного управления доступом, предотвращает утечку конфиденциальной информации, но снижает производительность компьютерной системы.

Ролевая модель разграничения доступа основана на конструировании ролей и назначении их пользователям на основании выполняемых ими конкретных должностных обязанностей. При назначении и использовании ролей возможно наложение динамических и статических ограничений на совмещение разных ролей одним субъектом, одновременное использование одной роли разными субъектами и т.п. Подобный подход к разграничению доступа к объектам

позволяет разделить обязанности между конструктором ролей и диспетчером ролей, а также более точно связать права доступа пользователей к объектам компьютерной системы с их обязанностями в организации, исключить избыточность полномочий.

В операционных системах Microsoft Windows и операционных системах клона Unix обычно применяется дискреционное управление доступом к объектам. Объекты разграничения доступа в Windows имеют дескриптор безопасности, содержащий информацию о владельце объекта (его идентификаторе безопасности SID, Security Identifier) и дискреционном списке управления доступом к объекту (Discretionary Access Control List, DACL), правом редактирования которого обладают владелец объекта и администратор. Владелец файла может лишить администратора права изменения разрешений на доступ к объекту. Администратор обладает специальной привилегией смены владельца на другого пользователя, обладающего такой же специальной привилегией (например, на самого себя).

Разграничение доступа к файлам и папкам возможно с помощью Проводника Windows (вкладки Безопасность функций Свойства контекстного меню выделенного объекта), принтеру – с помощью функции Принтеры и факсы Панели управления (вкладки Безопасность функции Свойства выделенного принтера), реестру Windows – с помощью Редактора реестра regedit.exe (функции Разрешения контекстного меню выделенного раздела).

Права доступа к объектам в операционной системе Windows делятся на специальные, стандартные (общие) и родовые (generic). Специальные права зависят от типа объекта разграничения доступа. Например, к файлам и папкам могут применяться следующие специальные права:

- обзор папок (выполнение файлов);
- содержание папки (чтение данных из файла);
- чтение атрибутов;
- чтение дополнительных атрибутов;

- создание файлов (запись данных в файл);
- создание папок (дозапись данных в файл);
- запись атрибутов;
- запись дополнительных атрибутов;
- удаление подпапок и файлов (только для папок).

Стандартные права доступа к объектам операционной системы Windows не зависят от типа объекта. Определены следующие стандартные права доступа;

- удаление;
- чтение разрешений;
- смена разрешений (для реестра это право названо Запись DAC);
- смена владельца;
- синхронизация (для реестра это право названо Уведомление).

Каждое из родовых разрешений представляет собой логическую группу специальных и стандартных разрешений. Например, для файлов и папок родовое право доступа «Изменение» включает все разрешения кроме «Удаление подпапок и файлов», «Смена разрешений» и «Смена владельца».

Существующий в Windows механизм наследования облегчает администраторам задачи назначения разрешений и управления ими. Благодаря этому механизму разрешения, установленные для контейнера, автоматически распространяются на все объекты этого контейнера. Например, файлы, создаваемые в папке, наследуют разрешения этой папки.

Если требуется предотвратить наследование разрешений, при настройке особых (отличающихся от родовых) разрешений на доступ к родительской папке (разделу реестра) можно выбрать режим «Применять эти разрешения к объектам и контейнерам только внутри этого контейнера». В случаях, когда необходимо отменить наследование разрешений только для некоторых файлов или подпапок (подразделов реестра), можно отменить режим «Наследовать от

родительского объекта применимые к дочерним объектам разрешения, добавляя их к явно заданным в этом окне».

Запрещение права доступа имеет более высокий приоритет, чем его разрешение, если только объект не наследует от различных папок противоречащие друг другу значения этих параметров. В таких случаях в силу вступает значение, унаследованное от родительского контейнера, ближайшего к объекту в иерархической структуре. Дочерние объекты наследуют только наследуемые разрешения.

К вновь созданным в сеансе пользователя объектам права доступа в Windows назначаются одним из следующих способов:

1. На основе явно заданного субъектом (управляемой пользователем программой) и корректного по форме дескриптора безопасности (например, при вызове системных функций CreateFile или CreateDirectory при создании файлов или папок);
2. На основе механизма наследования (если при создании объекта дескриптор безопасности не задается);
3. Из полученного при начале сеанса маркера доступа субъекта, создающего объект (если наследование невозможно).

Индекс файла в Linux содержит информацию о владельце файла (его идентификаторе, User Identifier, UID), его первичной группе (идентификаторе группы, Group Identifier, GID) и векторе доступа к файлу. В отличие от Windows вектор доступа в Linux состоит всегда из трех элементов, определяющих права доступа к объекту трех категорий субъектов (владельца, членов его группы и всех остальных). Суперпользователь в Linux имеет полные, никем не ограничиваемые права доступа к любому объекту.

В Linux существуют только три права доступа – чтение, запись и выполнение. Для каталогов право чтения означает разрешение на просмотр содержания каталога, право записи – разрешение создания, добавления и удаления файлов в

каталоге, право выполнения – разрешение на поиск файла в каталоге по его имени.

Ограниченность прав доступа к объектам в ОС Linux вынуждает использовать так называемые дополнительные биты доступа в каждой из его частей:

- SUID (дополнительный бит доступа в подвекторе прав владельца). Обеспечивает выполнение файла с правами не пользователя, а владельца файла (необходимо, например, для предоставления права записи в файл учетных записей /etc/passwdнепривилегированному пользователю при смене им своего пароля).

- SGID (дополнительный бит в подвекторе прав членов группы владельца файла). Обеспечивает выполнение файла с правами не пользователя, а членов группы владельца файла.

- Sticky (дополнительный бит в подвекторе прав всех остальных пользователей). Запрещает удаление и переименование в общем каталоге файлов, созданных другими пользователям.

Управлять значением вектора доступа к вновь созданным в сеансе пользователя файлам в ОС Linux администратор может с помощью системной переменной umask, значение которой может устанавливаться в файлах пользователей .login, .cshrc или .profile либо в системном файле / etc / profile. Значение umask определяет сбрасываемые биты в элементах вектора доступа к создаваемому объекту.

Сравнивая реализацию дискреционного разграничения доступа к объектам в операционных системах Microsoft Windows и Linux, можно отметить следующее:

- Использование привилегии администратора Windows «Овладение файлами или иными объектами» более безопасно, чем работа вLinux с правами суперпользователя, но менее удобна с точки зрения простоты администрирования.

- Большое количество разнообразных прав доступа к объектам в Windows увеличивает гибкость механизма управлении доступом, но повышает риск ошибочного

наделения пользователя или группы избыточными правами доступа.

- В Linux администратору проще управлять правами доступа к объектам, создаваемым пользователем в ходе своей работы в системе.

- В Windows возможно назначение индивидуальных прав доступа к объекту для любого отдельно взятого пользователя или группы.

В расширении подсистемы разграничения доступа к файлам для операционной системы Linux − Linux ACLs – реализована возможность настроить индивидуальные права доступа к файлам «с точностью» до отдельного пользователя.

В расширении базовой модели безопасности операционной системы Linux (Security Enhanced Linux − Linux с улучшенной безопасностью, SELinux) реализовано мандатное разграничение доступа к объектам в рамках модели домен-тип [1]. В этой модели каждый процесс запускается в определённом домене безопасности (с определенным уровнем допуска), а всем объектам ставится в соответствие определённый тип (метка секретности).

Список правил, ограничивающих возможности доступа доменов к типам, называется политикой и задаётся один раз в момент установки системы. Описание политики в SELinux − это набор текстовых файлов, которые могут быть скомпилированы и загружены в память ядра Linux при запуске системы.

Возможности SELinux по управлению доступом значительно превосходят возможности базовых прав Unix. Например, можно строго ограничить номер сетевого порта, с которым будет связан сетевой сервер или разрешить создание и запись в файл, но не его удаление. Это позволяет ограничить системные службы с помощью явно заданного набора существующих прав.

Поддержка ролевого разграничения доступа включена в серверные операционные системы Windows и в серверную

операционную систему ALT Linux Castle. Программисты и администраторы Windows-систем могут использовать преимущества ролевого разграничения доступа к объектам с помощью оснастки Диспетчер авторизации (Authorization Manager) [2].

В соответствии с реализованной в ОС ALT Linux Castle ролевой моделью управления доступом определяются роли и типы, а затем определяется, что может делать та или иная роль с тем или иным типом [3]. Таким образом, создается некоторая абстрактная модель, которая затем связывается к реальным пользователям, программам и файлам. Независимость модели от реальных субъектов и объектов позволяет производить мгновенную перенастройку политики безопасности быстрым изменением связей ролей и (или) типов. Кроме того, это очень удобно для создания готовых решений, например, распределения ролей и типов для защиты содержимого страниц Web-узла. Интересной особенностью является возможность запуска программ с ролью, отличной от роли пользователя, производящего запуск. В результате можно, например, произвести такие настройки, что прямой доступ к диску будут иметь только разрешенные программы, а все остальные пользователи системы (включая администратора) будут лишены такой возможности.

2. ЗАЩИТНЫЕ МЕХАНИЗМЫ ОПЕРАЦИОННЫХ СИСТЕМ

Перейдем к рассмотрению системы защиты операционных систем. Ее основными задачами являются идентификация, аутентификация, разграничение доступа пользователей к ресурсам, протоколирование и аудит самой системы [4, 5].

2.1. Идентификация и аутентификация

Обычно каждый пользователь в системе имеет уникальный идентификатор. Идентфикаторы пользователей применяются с теми же целями, что и идентификаторы любых других объектов, файлов, процессов [5]. *Идентификация* заключается в сообщении пользователем своего идентификатора. Для того чтобы установить, что пользователь именно тот, за кого себя выдает, то есть что именно ему принадлежит введенный идентификатор, в информационных системах предусмотрена процедура *аутентификации* (authentication, опознавание, в переводе с латинского означает установление подлинности), задача которой - предотвращение доступа к системе нежелательных лиц.

Обычно аутентификация базируется на одном или более из трех пунктов:

- то, чем пользователь владеет (ключ или магнитная карта),
- то, что пользователь знает (пароль),
- атрибуты пользователя (отпечатки пальцев, подпись, голос).

Пароли, уязвимость паролей

Наиболее простой подход к аутентификации - использование пользовательского пароля.

Когда пользователь идентифицирует себя при помощи уникального идентификатора или имени, у него запрашивается пароль. Если пароль, сообщенный пользователем, совпадает с паролем, хранимым в системе, система предполагает, что пользователь легитимен.

Пароли часто используются для защиты объектов в компьютерной системе в отсутствие более сложных схем защиты.

Проблемы паролей связаны с трудностью хранить пароль в секрете. Пароли могут быть скомпрометированы путем угадывания, случайно показаны или нелегально переданы авторизованным пользователем неавторизованному

Есть два общих способа угадать пароль. Один для нарушителя, который знает пользователя или информацию о пользователе. Люди обычно используют очевидную информацию (типа имен кошек) в качестве паролей. Для иллюстрации важности разумной политики назначения идентификаторов и паролей можно привести данные исследований, проведенных в AT&T, показывающие, что из 500 попыток несанкционированного доступа около 300 составляют попытки угадывания паролей или беспарольного входа по пользовательским именам guest, demo и т.д.

Другой способ - грубой силы - попытаться перебрать все возможные комбинации букв, чисел и пунктуации. Например, четыре десятичные цифры дают только 10000 вариантов, более длинные пароли, введенные с учетом регистра символов и пунктуации, менее уязвимы.

Хотя имеются проблемы с их использованием, пароли, тем не менее, распространены, так как они легки для понимания и использования.

Шифрование пароля

Для хранения секретного списка паролей на диске многие ОС используют криптографию. Система использует одностороннюю функцию, которую чрезвычайно трудно (дизайнеры надеются, что невозможно) инвертировать, но просто вычислить. Хранятся только кодированные пароли. В процессе аутентификации представленный пользователем пароль кодируется и сравнивается с хранящимися на диске. Таким образом, файл паролей нет необходимости держать в секрете.

При удаленном доступе к ОС нежелательно путешествие пароля по сети в открытом виде. Одним из типовых решений является использование криптографических протоколов. В качестве примера можно рассмотреть протокол опознавания с подтверждением установления связи путем вызова – CHAP (Challenge Handshake Authentication Protocol).

Опознавание достигается за счет проверки того, что у пользователя, осуществляющего доступ к серверу, имеется секретный пароль, который уже известен серверу.

Сервер посылает пользователю запрос (вызов), состоящий из идентифицирующего кода, случайного числа и имени узла сервера или имени пользователя. При этом пользовательское оборудование в результате затребования пароля пользователя отвечает следующим ответом, зашифрованным с помощью алгоритма одностороннего хэширования, наиболее распространенным видом которого является MD5. После получения ответа сервер при помощи той же функции с теми же аргументами шифрует собственную версию пароля пользователя. В случае совпадения результатов разрешается вход в систему. Существенно, что незашифрованный пароль при этом не посылается по каналу связи.

В микротелефонных трубках используется аналогичный метод.

2.2. Авторизация. Разграничение доступа к операционным системам

После того, как легальный пользователь вошел в систему необходимо осуществить *авторизацию* (authorization) – предоставление субъекту прав на доступ к объекту. Средства авторизации контролируют доступ легальных пользователей к ресурсам системы, предоставляя каждому из них именно те права, которые были определены администратором, а также осуществляют контроль возможности выполнения пользователем различных системных функций.

Как уже говорилось, компьютерная система может быть смоделирована как набор субъектов (процессы, пользователи) и объектов. Под объектами мы понимаем как ресурсы оборудования (процессор, сегменты памяти, принтер, диски и ленты), так и программные (файлы, программы, семафоры). Каждый объект имеет уникальное имя, отличающее его от других объектов в системе, и каждый из них может быть доступен через хорошо определенные и значимые операции. Объекты – абстрактные типы данных.

Операции зависят от объектов. Например, процессор может только выполнять команды. Сегменты памяти могут быть записаны и прочитаны, тогда как считыватель карт может только читать. Файлы данных могут быть записаны, прочитаны, переименованы и т.д.

Очевидно, что процессу может быть разрешен доступ только к тем ресурсам, к которым он имеет авторизованный доступ. Желательно добиться того, чтобы он имел доступ только к тем ресурсам, которые ему нужны для выполнения его задачи. Это требование имеет отношение только к принципу минимизации привилегий, полезному с точки зрения ограничения количества повреждений, которые процесс может нанести системе. Например, когда процесс P вызывает процедуру A, ей должен быть разрешен доступ только к переменным и формальным параметрам, переданным ей, она должна быть не в состоянии влиять на другие переменные

процесса. Аналогично компилятор не должен оказывать влияния на произвольные файлы, а только на их хорошо определенное подмножество (типа исходных файлов, листингов и др.), имеющих отношение к компиляции. С другой стороны, компилятор может иметь личные файлы, используемые для оптимизационных целей, к которым процесс P не имеет доступа.

Различают *дискреционный* (избирательный) способ управления доступом и *полномочный* (мандатный). При дискреционном доступе определенные операции над определенным ресурсом запрещаются или разрешаются субъектам или группам субъектов. С концептуальной точки зрения текущее состояние прав доступа при дискреционном управлении описывается матрицей, в строках которой перечислены субъекты, а в столбцах - объекты.

Полномочный подход заключается в том, что вся информация делится на уровни в зависимости от степени секретности, а все пользователи также делятся на группы, образующие иерархию в соответствии с уровнем допуска к этой информации.

Большинство операционных систем реализуют именно дискреционное управление доступом. Главное его достоинство - гибкость, главные недостатки - рассредоточенность управления и сложность централизованного контроля, а также оторванность прав доступа от данных, что позволяет копировать секретную информацию в общедоступные файлы.

2.2.1. Домены безопасности

Чтобы развить эту схему мы введем концепцию домена безопасности (protection domain). Процесс оперирует с доменом безопасности, который специфицирует ресурсы, к которым процесс может иметь доступ (рис.). Каждый домен определяет набор объектов и типов операций, которые могут быть осуществлены над каждым объектом. Возможность выполнять операции над объектом есть права доступа. Домен есть набор

прав доступа, каждое из которых есть упорядоченная пара <object-name, rights-set>. Например, если домен D имеет права доступа <file F, {read, write}>, это означает, что процесс, выполняемый в домене D, может читать или писать в файл F, но не может выполнять других операций над этим объектом.

Объект / Домен	F1	F2	F3	Printer
D1	read			
D2				print
D3		read	execute	
D4	read write		read write	

Спецификирование прав доступа к ресурсам

Связь процессов с доменами может быть статической и динамической. Организация динамической связи сложнее.

Заметим, что домен может быть реализован различными способами:

- Каждый пользователь может быть доменом. В этом случае набор объектов, к которым может быть организован доступ, зависит от идентификации пользователя. Переключение между доменами имеет место, когда меняется пользователь (один входит в систему, другой выходит из нее).
- Каждый процесс может быть доменом. В этом случае набор доступных объектов определяется идентификацией процесса. Переключение между доменами происходит, когда один из процессов посылает сообщение другому и ждет отклика.
- Каждая процедура может быть доменом. В этом случае набор доступных объектов соответствует локальным переменным, определенным внутри процедуры. Переключение между доменами происходит, когда процедура выполнена.

Рассмотрим стандартную двух режимную модель выполнения ОС. Когда процесс выполняется в режиме системы (kernel mode), он может выполнять привилегированные инструкции и иметь полный контроль над компьютерной системой. С другой стороны, если процесс выполняется в пользовательском режиме, он может вызывать только непривилегированные инструкции. Следовательно, он может выполняться только внутри предопределенного пространства памяти. Наличие этих двух режимов позволяет защитить ОС (monitor domain) от пользовательских процессов (выполняющихся в user domain). В мультипрограммных системах двух доменов недостаточно, так как появляется необходимость защиты пользователей друг от друга. Поэтому требуется лучше разработанная схема.

В ОС Unix домен связан с пользователем. Переключение доменов соответствует смене пользователя. Это изменение реализуется через файловую систему.

2.2.2. Матрица доступа

Модель безопасности, таким образом, выглядит как матрица, называемая *матрицей доступа*.

Какова может быть эффективная реализация матрицы доступа. В общем случае она будет разреженной, то есть большинство клеток будут пустыми. Хотя существуют структуры данных для представления разреженной матрицы, они не слишком полезны для приложений, использующих возможности защиты.

Список прав доступа. Access control list

Каждая колонка в матрице может быть реализована как список доступа для одного объекта. Очевидно, что пустые клетки могут не учитываться. В результате для каждого объекта имеем список упорядоченных пар <domain, rights-set>, который определяет все домены с непустыми наборами прав для данного объекта.

Список прав доступа может быть дополнен дефолтным набором прав.

Пример Unix. Все субъекты разделены на три группы, для членов каждой группы контролируются три операции (rwx), в итоге имеем ACL 9-битный код.

Capability list

Если матрицу доступа хранить по строкам, то есть каждый субъект хранит список объектов и для каждого объекта список допустимых операций, то такой способ хранения называется capability list.

Примерами систем такого рода являются Hydra, Cambridge CAP System.

Иногда применяется *комбинированный способ*. Например, в том же Unix на этапе открытия файла происходит анализ ACL. В случае благоприятного исхода (у процесса были соответствующие права) файл заносится в список открытых файлов, и при последующих операциях чтения и записи проверки прав доступа не происходит. Список открытых файлов можно рассматривать как capability list.

Существуют другие общие методы, используемые для смены домена в ОС, в которой идентификаторы пользователей используется для определения домена. Почти все системы нуждаются в таком механизме. Этот механизм используется, когда некая привилегированная возможность необходима большому количеству пользователей. Например, может быть желательно разрешить пользователям иметь доступ к сети без того, чтобы заставлять их писать собственные сетевые программы. Для этого случая в ОС Unix устанавливается бит setuid в сетевой программе, заставляя меняться домен на время ее выполнения. Таким образом, рядовой пользователь может получить нужные привилегии для доступа к сети.

Механизм Lock-Key

Схема lock-key - компромисс между access lists и capability lists. Каждый объект имеет список уникальных битовых шаблонов (patterns), называемых locks. Аналогично, каждый домен имеет список уникальных битовых шаблонов,

называемых ключами. Процесс, выполняющийся в домене, может иметь доступ к объекту, только если домен имеет ключ, который соответствует одному из locks объекта.

Как и в случае capability lists, список ключей для домена должен управляться ОС. Пользователям не разрешено проверять или модифицировать списки ключей (или locks) непосредственно.

2.2.3. Недопустимость повторного использование объектов

Контроль за повторным использованием объекта предназначен для предотвращения попыток незаконного получения конфиденциальной информации, остатки которой могли сохраниться в некоторых объектах, ранее использованных и освобожденных другим пользователем. Безопасность повторного использования должна гарантироваться для областей оперативной памяти (в частности, для буферов с образами экрана, расшифрованными паролями и т.п.), для дисковых блоков и магнитных носителей в целом. Очистка должна производиться путем записи маскирующей информации в объект при его освобождении (перераспределении). Например, для дисков на практике применяется способ двойной перезаписи удаленных файлов случайной битовой последовательностью.

2.3. Аудит, учет использования системы защиты

Аудит заключается в регистрации специальных данных о различных типах событий, происходящих в системе и так или иначе влияющих на состояние безопасности компьютерной системы. К числу таких событий относятся:

- вход или выход из системы;
- операции с файлами (открыть, закрыть, переименовать, удалить);
- обращение к удаленной системе;

- смена привилегий или иных атрибутов безопасности (режима доступа, уровня благонадежности пользователя и т.п.).

Если фиксировать все события, объем регистрационной информации, скорее всего, будет расти слишком быстро, а ее эффективный анализ станет невозможным. Следует предусматривать наличие средств выборочного протоколирования, как в отношении пользователей, когда слежение осуществляется только за подозрительными личностями, так и в отношении событий. Слежка важна в первую очередь как профилактическое средство. Можно надеяться, что многие воздержатся от нарушений безопасности, зная, что их действия фиксируются.

Помимо протоколирования можно сканировать систему периодически на наличие брешей в системе безопасности. Такое сканирование может проверить разнообразные аспекты системы:

- Короткие или легкие пароли
- Неавторизованные set-uid программы, если система поддерживает этот механизм
- Неавторизованные программы в системных директориях
- Долго выполняющиеся программы
- Нелогичная защита как пользовательских, так и системных директорий, системных файлов данных, таких как файлы паролей, драйверов, ядра
- Потенциально опасные списки поиска файлов, могущие привести к запуску троянского коня.
- Изменения в системных программах, обнаруженные при помощи контрольных сумм.

Любая проблема, обнаруженная сканером безопасности, может быть, как исправлена автоматически, так и доложена менеджеру системы.

2.4. Анализ некоторых популярных операционных систем с точки зрения их защищенности

Итак, ОС должна способствовать реализации мер безопасности или прямо поддерживать их. Примеры подобных решений в рамках аппаратуры и операционной системы - разделение команд по уровням привилегированности, защита различных процессов от взаимного влияния за счет выделения каждому своего виртуального пространства, особая защита ядра ОС, контроль за повторным использованием объекта.

Большое значение имеет структура файловой системы. Например, в ОС с дискреционным контролем доступа каждый файл должен храниться вместе с дискреционным списком прав доступа к нему, а, например, при копировании файла все атрибуты, в том числе и ACL, должны быть автоматически скопированы вместе с телом файла.

В принципе меры безопасности не обязательно должны быть заранее встроены в ОС - достаточно принципиальной возможности дополнительной установки защитных продуктов. Так, сугубо ненадежная система MS-DOS может быть улучшена за счет средств проверки паролей доступа к компьютеру и/или жесткому диску, за счет борьбы с вирусами путем отслеживания попыток записи в загрузочный сектор CMOS-средствами и т.п. Тем не менее, по-настоящему надежная система должна **изначально** проектироваться с акцентом на механизмы безопасности.

Среди архитектурных решений, с точки зрения информационной безопасности, целесообразны также следующие:

- деление аппаратных и системных функций по уровням привилегированности и контроль обмена информацией между уровнями;
- защита различных процессов от взаимного влияния за счет механизма виртуальной памяти;
- наличие средств управления доступом;

- структурированность системы, явное выделение надежной вычислительной базы, обеспечение компактности этой базы;
- следование принципу минимизации привилегий - каждому компоненту дается ровно столько привилегий, сколько необходимо для выполнения им своих функций;
- сегментация (в частности, сегментация адресного пространства процессов) как средство повышения надежности компонентов.
-

2.4.1. MS-DOS

ОС MS-DOS функционирует в реальном режиме (real-mode) процессора i80x86. В ней невозможно выполнение требования, касающегося изоляции программных модулей (отсутствует аппаратная защита памяти). Уязвимым местом для защиты является также файловая система FAT, не предполагающая в файлах наличие атрибутов, связанных с разграничением доступа к ним. Таким образом, MS-DOS, не будучи защищенной, находится на самом нижнем уровне в иерархии защищенных ОС

2.4.2. NetWare, IntranetWare

Замечания об отсутствии изоляции модулей друг от друга справедливо и относительно рабочей станции NetWare. Однако NetWare – сетевая ОС, поэтому к ней возможно применение и иных критериев. Это на данный момент единственная *сетевая* ОС, сертифицированная по классу С2 (следующей, по-видимому, будет Windows 2000). При этом важно изолировать наиболее уязвимый участок системы безопасности NetWare - консоль сервера и тогда следование определенной практике поможет увеличить степень защищенности этой сетевой операционной системы. Возможность создания безопасных систем обусловлена тем, что число работающих приложений *фиксировано* и

пользователь не имеет возможности запуска своих приложений.

2.4.3. OS/2

OS/2 работает в защищенном режиме (protected-mode) процессора i80x86. Изоляция программных модулей реализуется при помощи встроенных в этот процессор механизмов защиты памяти. Поэтому она свободна от вышеуказанного коренного недостатка систем типа MS-DOS. Но OS/2 была спроектирована и разработана без учета требований по защите от несанкционированного доступа. Это сказывается, прежде всего, на файловой системе. В файловых системах OS/2 HPFS (high performance file system) и FAT нет места ACL. Кроме того, пользовательские программы имеют возможность запрета прерываний. Следовательно, сертификация OS/2 на соответствие какому-то классу защиты не представляется возможной.

Считается, что такие операционные системы, как MS-DOS, MacOS, Windows, OS/2, имеют уровень защищенности D (по оранжевой книге). Но если быть совершенно точным, нельзя считать эти ОС даже системами уровня безопасности D, ведь они никогда не представлялись на тестирование.

2.4.4. Unix

Рост популярности Unix и все большая осведомленность о проблемах безопасности привели к осознанию необходимости достичь приемлемого уровня безопасности ОС, сохранив при этом мобильность, гибкость и открытость программных продуктов. В Unix есть несколько уязвимых с точки зрения безопасности мест, хорошо известным искушенным пользователям, вытекающими из самой природы Unix и открывающими двери для нападения. (см., например, раздел Типичные объекты атаки хакеров в книге [23]). Однако,

хорошее системное администрирование может ограничить эту уязвимость.

Существуют противоречивые сведения относительно защищенности Unix. В Unix изначально были заложены идентификация пользователей и разграничение доступа. Как оказалось, средства защиты данных в Unix могут быть доработаны, и сегодня можно утверждать, что многие клоны Unix по всем параметрам соответствуют классу безопасности C2.

Обычно, говоря о защищенности Unix, рассматривают защищенность автоматизированных систем, одним из компонентов которых является Unix сервер. Безопасность такой системы увязывается с защитой глобальных и локальных сетей, безопасностью удаленных сервисов типа telnet и rlogin/rsh и аутентификацией в сетевой конфигурации, безопасностью X Windows приложений. На системном уровне важно наличие средств идентификации и аудита.

В Unix существует список именованных пользователей, в соответствии с которым может быть построена система разграничения доступа.

Все пользователи, которым разрешена работа в системе, учитываются в файле пользователей /etc/passwd. Группы пользователей учитываются в файле /etc/group. Каждому пользователю назначается целочисленный идентификатор и пароль.

Когда пользователь входит в систему и предъявляет свое имя (процедура login), отыскивается запись в учетном файле /etc/passwd. В этой записи имеются такие поля как: имя пользователя, имя группы, к которой принадлежит данный пользователь, целочисленный идентификатор пользователя, целочисленный идентификатор группы, зашифрованный пароль пользователя.

В ОС Unix, считается, что информация, нуждающаяся в защите, находится главным образом в файлах.

По отношению к конкретному файлу все пользователи делятся на три категории:

- владелец файла;
- члены группы владельца;
- прочие пользователи.

Для каждой из этих категорий режим доступа определяет права на операции с файлом, а именно:

- право на чтение;
- право на запись;
- право на выполнение (для каталогов – право на поиск).

Стандартная команда ls -l выдает список файлов с правами доступа к ним, например:

rwxr-x--- ... filename

Здесь символы rwx означают наличие прав на чтение, запись и исполнение соответственно, а символ - - отсутствие такого права.

Указанных видов прав достаточно, чтобы определить допустимость любой операции с файлами. Например, для удаления файла необходимо иметь право на запись в соответствующий каталог.

Наличие всего трех видов субъектов доступа: владелец, группа, все остальные - затрудняет задание прав с точностью до пользователя, особенно в случае больших конфигураций. В популярной разновидности Unix - Solaris имеется возможность использовать списки управления доступом (ACL), позволяющие с помощью команды setfacl индивидуально устанавливать права доступа отдельных пользователей или групп.

Среди всех пользователей особое положение занимает пользователь root, обладающий максимальными привилегиями. Обычные правила разграничения доступа к нему не применяются - ему доступна вся информация на компьютере.

В Unix имеются инструменты системного аудита - хронологическая запись событий, имеющих отношение к безопасности. К таким событиям обычно относят: обращения программ к отдельным серверам; события, связанные с входом/выходом в систему и другие. Обычно регистрационные действия выполняются специализированным syslog-демоном, который проводит запись событий в регистрационный журнал в соответствии с текущей конфигурацией. Syslog-демон стартует в процессе загрузки системы.

Таким образом, безопасность ОС Unix может быть доведена до соответствия классу С2. Однако разработка на ее основе автоматизированных систем более высокого класса защищенности может быть сопряжена с большими трудозатратами.

2.4.5. Windows NT/2000/XP и новее

С момента выхода версии 3.1 осенью 1993 года в Windows NT гарантировалось соответствие уровню безопасности С2. В настоящее время сертифицирована версия NT с использованием файловой системы NTFS в автономной и сетевой конфигурации. Следует помнить, что этот уровень безопасности не подразумевает защиту информации, передаваемой по сети, и не гарантирует защищенности от физического доступа. Компоненты защиты NT частично встроены в ядро, а частично реализуются подсистемой защиты. Подсистема защиты регистрирует правила контроля доступа и контролирует учетную информацию.

В дальнейшем линейку продуктов Windows NT/2000/XP и новее, изготовленных по технологии NT, будем называть просто Windows NT.

Ключевая цель системы защиты Windows NT – следить за тем, кто и к каким объектам осуществляет доступ. Система защиты хранит информацию, относящуюся к безопасности для каждого пользователя, группы пользователей и объекта.

Единообразие контроля доступа к различным объектам (процессам, файлам, семафорам и др.) обеспечивается тем, что с каждым процессом связан маркер доступа, а с каждым объектом – дескриптор защиты. Маркер доступа в качестве параметра имеет идентификатор пользователя, а дескриптор защиты – списки прав доступа. ОС может контролировать попытки доступа, которые производятся процессами прямо или косвенно инициированными пользователем.

Windows NT отслеживает и контролирует доступ как к объектам, которые пользователь может видеть посредством интерфейса (такие, как файлы и принтеры), так и к объектам, которые пользователь не может видеть (например, процессы и именованные каналы). Любопытно, что, помимо разрешающих записей, списки прав доступа содержат и запрещающие записи, чтобы пользователь, которому доступ к какому-либо объекту запрещен, не смог получить его как член какой-либо группы, которой этот доступ предоставлен.

Microsoft Windows NT - относительно новая ОС, которая была спроектирована для поддержки разнообразных защитных механизмов от минимальных до С2. Дефолтный уровень называется минимальным, но он легко может быть доведен системным администратором до желаемого уровня. Утилита C2config.exe помогает администратору сделать нужные установки.

Система защиты ОС Windows NT состоит из следующих компонентов:

- Процедуры регистрации (Logon Processes), которые обрабатывают запросы пользователей на вход в систему. Они включают в себя начальную интерактивную процедуру, которая отображает начальный диалог с пользователем на экране и удаленные процедуры входа, которые позволяют удаленным пользователям получить доступ с рабочей станции сети к серверным процессам Windows NT.
- Подсистемы локальной авторизации (Local Security Authority, LSA), которая гарантирует, что пользователь имеет разрешение на доступ в систему.

Эта компонента – центральная для системы защиты Windows NT. Она порождает маркеры доступа, управляет локальной политикой безопасности и предоставляет интерактивным пользователям аутентификационные услуги. LSA также контролирует политику аудита и ведет журнал, в котором сохраняются аудитные сообщения, порождаемые диспетчером доступа.

- Менеджера учета (Security Account Manager, SAM), который управляет базой данных учета пользователей. Эта база данных содержит информацию обо всех пользователях и группах пользователей. SAM предоставляет услуги по легализации пользователей, которые используются в LSA.
- Диспетчера доступа (Security Reference Monitor, SRM), который проверяет, имеет ли пользователь право на доступ к объекту и на выполнение тех действий, которые он пытается совершить с объектом. Эта компонента проводит в жизнь легализацию доступа и политику аудита, определяемые LSA. Она предоставляет услуги для программ супервизорного и пользовательского режимов для того, чтобы гарантировать, что пользователи и процессы, осуществляющие попытки доступа к объекту, имеют необходимые права. Эта компонента также порождает аудитные сообщения, когда это необходимо.

Ключевая цель системы защиты Windows NT - мониторинг и контроль того, кто и к каким объектам осуществляет доступ. Система защиты хранит информацию, относящуюся к безопасности для каждого пользователя, группы пользователей и объекта. Она может идентифицировать попытки доступа, которые производятся прямо пользователем или непрямо программой или другим процессом, инициированным пользователем. Windows NT также отслеживает и контролирует доступ и к тем объектам, которые пользователь может видеть посредством пользовательского интерфейса (такие как файлы и принтеры), и к объектам, которые пользователь не может видеть (такие как процессы и именованные каналы).

3. ЗАДАНИЯ ДЛЯ ПРАКТИЧЕСКИХ РАБОТ

1. Создать учетную запись пользователя/группу пользователей с различными уровнями доступа.
2. Изменить пароли в различных группах пользователей.
3. Проработать различные варианты аутентификации для приложений с учетом PAM.
4. Сделать вход в систему и завершение сеанса.
5. Изучить базовые права доступа, используя команду ls с ключом вывода расширенной информации.
6. Рассмотреть режим обычного пользователя.
7. Сделать переход в режим суперпользователя.
8. Изучить БД пользователей в –nix системах.
9. Добавить нового пользователя. Создать для данной учетной записи пароль.
10. Удалить созданную раннее учетную запись.
11. Создайте текстовый файл и задайте права на него таким образом, чтобы он мог просматриваться только владельцем и никем не мог редактироваться.
12. Найдите все исполняемые файлы с установленным suid-битом.
13. Получите имена всех пользователей системы, у которых в качестве командной оболочки используется программа /bin/false.
14. Выясните, чем отличается реакция операционной системы (выводимое сообщение) на различные ошибки аутентификации (например, неправильный пользователь, неверный пароль и т.д.).

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение термину «разграничение доступа». Как реализуется разграничение доступа в различных операционных системах?
2. Как осуществляется хранение и управление паролями в операционных системах?
3. Как реализуются алгоритмы шифрования паролей в операционных системах?
4. Что представляет собой блок PAM?
5. Что представляет собой псевдопользователь? Дайте классификацию основным пользователям в Unix.
6. Для чего нужны атрибуты файла SUID (Set-UID) и SGID (Set-GID)?
7. Перечислить основные команды для Unix – подобных систем и дать характеристику.
8. Сравните права доступа к директориям /bin и /tmp. Какие операции сможет совершать в них простой пользователь?
9. Что смогут делать другие пользователями с файлами в домашней директории пользователя, если он задаст всем остальным пользователям право на запись в директорию, но удалит право исполнения на неё?
10. Чем отличаются номинальный и действительный субъект? Как они соотносятся с объектом безопасности? Что представляют собой субъект и объект безопасности в UNIX?
11. Что такое политика безопасности? Какие требования выдвигаются по отношению к ней?
12. Какие существуют наиболее распространённые схемы доступа? В чём заключаются основные отличия между ними? Какая схема доступа используется в UNIX?
13. Какие существуют права доступа в UNIX? Какие из них являются специфичными для простых файлов, а какие для директорий?

14. Что такое подмена идентификатора субъекта? Как такое право устанавливается и где применяется?

15. Опишите процесс аутентификации пользователя в UNIX.

16. Каким образом хранится информация обо всех пользователях системы?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. SELinux: теория и практика безопасности. - [Электронный ресурс] – Режим доступа: http://www.interface.ru/ home.asp?artId=2699.
2. Хорев, П. Б. Ролевое управление доступом к ресурсам в операционной системе Microsoft Windows Server 2003 [Текст] / П. Б. Хорев // Информационные средства и технологии: труды XVI международной научно-технической конференции. – 2008. – Т. 2. – С. 80-88.
3. ALTLinux Castle. Общие сведения. - [Электронный ресурс] – Режим доступа: http://www.linuxcenter.ru/ lib/articles/distrib/altlinux/castle.phtml.
4. Гордеев, А. В. Операционные системы [Текст]: учебник для вузов / А. В. Гордеев. – 2-е изд. – СПб.: Питер, 2007.
5. Таненбаум Э. Современные операционные системы [Текст] / Э. Таненбаум. – 2-е изд. – СПб. : Питер, 2006.
6. Denning, P. J. Before memory was virtual [Text] / P. J. Denning // Draft. – 1996. – June 6th.
7. Galvin, P. B. Operating System Concepts [Text]/ P. B. Galvin, A. Silberschatz. – 6th edition – John Willey & Sons, 2002.

СОДЕРЖАНИЕ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к практическим занятиям по дисциплине
«Защита в операционных системах»
для студентов специальности
090301 «Компьютерная безопасность»
очной формы обучения

Составители:
Савинков Андрей Юрьевич
Ленков Никита Александрович
Деревянко Владимир Николаевич

В авторской редакции

Подписано к изданию 27.04.2015.
Уч.-изд. л. 1,9.

ФГБОУ ВПО «Воронежский государственный
технический университет»
394026 Воронеж, Московский просп., 14