Министерство науки и высшего образования Российской Федерации
Байкальский государственный университет

А.А. Бутин, С.И. Носков, В.Д. Торопов

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Учебное пособие

Иркутск
Издательский дом БГУ
2023

УДК 004.056(075.8)
ББК 32.972.53я7
Б93

Издается по решению редакционно-издательского совета
Байкальского государственного университета

Авторы
канд. физ.-мат. наук, доц. А.А. Бутин (ИрГУПС)
д-р техн. наук, проф. С.И. Носков (ИрГУПС)
канд. техн. наук, доц. В.Д. Торопов (БГУ)

Рецензенты
д-р техн. наук, доц. Иркутского филиала
Московского государственного университета
гражданской авиации В.В. Ерохин
д-р физ.-мат. наук, проф. Института динамики систем
и теории управления СО РАН А.В. Лакеев

Бутин, А.А.
Б93 Безопасность операционных систем : учеб. пособие / А.А. Бутин, С.И. Носков, В.Д. Торопов. — Иркутск : Изд. дом БГУ, 2023. — 98 с.

ISBN 978-5-7253-3132-5.

В учебном пособии рассматриваются функциональные возможности встроенных (штатных) средств защиты информации в операционных системах общего назначения, достоинства и недостатки защитных механизмов. Пособие охватывает основные темы курса в части изучения базовых сервисов обеспечения информационной безопасности в этом сегменте.

Для обучающихся по специальности 10.05.03 «Информационная безопасность автоматизированных систем», по направлению 10.03.01 «Информационная безопасность», а также практикующих специалистов в области защиты информации.

УДК 004.056(075.8)
ББК 32.972.53я7

ISBN 978-5-7253-3132-5

ОГЛАВЛЕНИЕ

ПРЕДИСЛОВИЕ

В связи с бурным развитием информационных технологий в настоящее время жизнь многих людей все больше зависит от систем автоматизированного управления различными объектами. Для штатного функционирования подобные системы должны быть защищены от различного рода внешних и внутренних воздействий, которые могли бы привести к потере или искажению обрабатываемой и управляющей информации, а также к модификации самих систем. Одним из важнейших направлений в обеспечении надежной и безопасной работы является разработка методов и средств программно-аппаратного уровня защиты данных от различных угроз.

Предлагаемое учебное пособие предназначено для студентов, обучающихся по специальности «Информационная безопасность автоматизированных систем» и направлению «Информационная безопасность». В пособии рассматриваются вопросы, связанные с организацией защиты информации в операционных системах. При этом предполагается, что студенты уже имеют представление об основных понятиях и концепциях построения и функционирования операционных систем. Авторы старались не дублировать в настоящем пособии материал прежних изданий, исключение делается только для определений наиболее важных понятий информационной безопасности.

В качестве основной операционной системы для иллюстрации материала выбрана линейка операционных систем Microsoft Windows NT. Это было сделано из следующих соображений:

– Windows NT на сегодняшний день является наиболее распространенной защищенной операционной системой;

– Windows NT по пользовательскому интерфейсу совместима с другими версиями Windows и не требует от пользователя, ранее работавшего с Windows, приобретения специальных навыков;

– подсистема защиты Windows NT является одной из наиболее мощных и гибких, что позволяет с наибольшей эффективностью проиллюстрировать практическое приложение теоретического материала.

В настоящем пособии приводятся также краткие описания подсистем защиты операционной системы UNIX. Это сделано для того, чтобы студенты могли сравнить различные подходы к решению одних и тех же задач, возникающих при проектировании и обслуживании защищенной операционной системы. В качестве базовой операционной системы для практических занятий может быть взята любая из этих систем или какая-либо другая защищенная операционная система.

Вопросы организации защиты информации в конкретных операционных системах разнесены по отдельным главам. Это облегчает использование пособия в качестве справочного материала. Авторы практически не затрагивают вопросы защиты в вычислительных сетях, так как этой теме должно быть посвящено отдельное учебное пособие. В настоящем пособии подробно рассматриваются только те средства защиты информации, которые могут применяться как в сети, так и на автономном компьютере. Чисто сетевые средства защиты, встроенные в операционные системы, либо не упоминаются вовсе, либо описываются предельно кратко.

В пособии были частично использованы материалы работ, приведенных в библиографическом списке, авторам которых выражается глубокая признательность.

Глава 1
ОБЩИЕ ВОПРОСЫ ОБЕСПЕЧЕНИЯ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

§ 1. УГРОЗЫ БЕЗОПАСНОСТИ ОПЕРАЦИОННОЙ СИСТЕМЫ

Прежде чем начинать разговор об организации эффективной и надежной защиты информации в операционных системах (ОС), определим круг угроз, от которых необходимо защититься. Угрозы безопасности ОС существенно зависят от условий эксплуатации системы, от того, какая информация хранится и обрабатывается в системе, и т.д. Например, если ОС используется главным образом для организации электронного документооборота, наиболее опасны угрозы, связанные с несанкционированным доступом (НСД) к файлам. Если же ОС используется как платформа для провайдера интернет-услуг, очень опасны атаки на сетевое программное обеспечение операционной системы.

Классификация угроз безопасности операционной системы

Единой и общепринятой классификации угроз безопасности ОС пока не существует. Однако можно классифицировать эти угрозы по различным аспектам их реализации.

Классификация угроз по цели:

- несанкционированное чтение информации;
- несанкционированное изменение информации;
- несанкционированное уничтожение информации;
- полное или частичное разрушение ОС (под разрушением ОС понимается целый комплекс разрушающих воздействий от кратковременного вывода из строя («завешивания») отдельных

программных модулей системы до физического стирания с диска системных файлов).

Классификация угроз по принципу воздействия на ОС:

– использование известных (легальных) каналов получения информации (например, угроза несанкционированного чтения файла, доступ пользователей к которому определен некорректно, разрешен доступ пользователю, которому, согласно адекватной политике безопасности, доступ должен быть запрещен);

– использование скрытых каналов получения информации (например, угроза использования злоумышленником недокументированных возможностей ОС);

– создание новых каналов получения информации с помощью программных закладок.

Классификация угроз по характеру воздействия на ОС:

– активное воздействие или несанкционированные действия злоумышленника в системе;

– пассивное воздействие или несанкционированное наблюдение злоумышленника за процессами, происходящими в системе.

Классификация угроз по типу используемой злоумышленником слабости защиты:

– неадекватная политика безопасности, в том числе и ошибки администратора системы;

– ошибки и недокументированные возможности программного обеспечения ОС, в том числе и так называемые люки, случайно или преднамеренно встроенные в систему «служебные входы», позволяющие обходить систему защиты. Обычно люки создаются разработчиками программного обеспечения для тестирования и отладки, и иногда разработчики забывают их удалить или оставляют специально;

– ранее внедренная программная закладка.

Классификация угроз по способу воздействия на объект атаки:

– непосредственное воздействие;

– превышение пользователем своих полномочий;

– работа от имени другого пользователя;

– использование результатов работы другого пользователя (например, несанкционированный перехват информационных потоков, инициированных другим пользователем).

Классификация угроз по способу действий злоумышленника (нарушителя):

– в интерактивном режиме (вручную);

– в пакетном режиме (с помощью специально написанной программы, которая выполняет негативные воздействия на ОС без непосредственного участия нарушителя).

Классификация угроз по объекту атаки:

– ОС в целом;

– объекты ОС (файлы, устройства и т.д.);

– субъекты ОС (пользователи, процессы и т.д.);

– каналы передачи данных.

Классификация угроз по используемым средствам атаки:

– штатные средства ОС без использования дополнительного программного обеспечения;

– программное обеспечение третьих фирм (к этому классу программного обеспечения относятся как компьютерные вирусы и другие вредоносные программы (exploits), которые можно легко найти в сети Интернет, так и программное обеспечение, изначально разработанное для других целей: отладчики, сетевые мониторы и сканеры и т.д.);

– специально разработанное программное обеспечение.

Классификация угроз по состоянию атакуемого объекта ОС на момент атаки:

– хранение;
– передача;
– обработка.

Приведенная классификация не претендует ни на строгость, ни на полноту. Единственная цель включения ее в данное пособие — показать весь спектр возможных угроз безопасности ОС.

Типичные атаки на ОС

1. Сканирование файловой системы. Данная атака является одной из наиболее тривиальных, но в то же время одной из наиболее опасных. Суть атаки заключается в том, что злоумышленник просматривает файловую систему компьютера и пытается прочесть (или скопировать, или удалить) все файлы подряд. Если он не получает доступ к какому-то файлу или каталогу, то продолжает сканирование. Если объем файловой системы достаточно велик, рано или поздно обнаруживается хотя бы одна ошибка администратора. В результате злоумышленник получает доступ к информации, доступ к которой должен быть ему запрещен. Данная атака может осуществляться специальной программой, которая выполняет вышеописанные действия в автоматическом режиме.

Несмотря на кажущуюся примитивность описанной атаки, защититься от нее не так просто. Если политика безопасности допускает анонимный или гостевой вход в систему, администраторам остается только надеяться, что права доступа ко всем файлам системы, число которых может составлять сотни тысяч, определены абсолютно корректно. Если же анонимный и гостевой вход в систему запрещен, поддержание адекватной политики регистрации потенциально опасных событий (аудита) позволяет организовать эффективную защиту от этой угрозы. Впрочем, следует отметить, что поддержание адекватной политики аудита требует от администраторов системы определенного искусства. Кроме того,

если данная атака осуществляется злоумышленником от имени другого пользователя, аудит совершенно неэффективен.

2. Кража ключевой информации. В простейшем случае эта атака заключается в том, что злоумышленник подсматривает пароль, набираемый пользователем. То, что все современные ОС не высвечивают на экране вводимый пользователем пароль, несколько затрудняет эту задачу, но не делает ее невыполнимой. Известно, что для того, чтобы восстановить набираемый пользователем пароль только по движениям рук на клавиатуре, достаточно всего несколько недель тренировок. Кроме того, достаточно часто встречается ситуация, когда пользователь ошибочно набирает пароль вместо своего имени, которое, в отличие от пароля, на экране высвечивается.

Некоторые программы входа в ОС удаленного сервера допускают ввод пароля из командной строки. К таким командам относится, например, команда nwlogin операционной системы UNIX, предназначенная для входа на сервер NovellNetWare. При использовании с ключом p она позволяет вводить пароль в командной строке, например, nwloginserveruserppassword.

При вводе пароля в командной строке пароль, естественно, отображается на экране и может быть прочитан злоумышленником. Известны случаи, когда пользователи создавали командные файлы, состоящие из команд, подобных вышеприведенной, для автоматического входа на удаленные серверы. Если злоумышленник получает доступ к такому файлу, тем самым он получает доступ ко всем серверам, к которым имеет доступ данный пользователь, в пределах предоставленных ему полномочий.

Иногда пользователи, чтобы не забыть пароль, записывают его на бумагу, которую приклеивают к нижней части клавиатуры, к задней стенке системного блока или в какое-нибудь другое

якобы укромное место. В этом случае пароль рано или поздно становится добычей злоумышленника. Особенно часто такие ситуации имеют место в случаях, когда политика безопасности требует от пользователей использовать длинные, трудно запоминаемые пароли.

Известны случаи, когда для кражи пароля злоумышленники скрытно фиксировали отпечатки пальцев пользователя на клавиатуре непосредственно после набора пароля.

Наконец, если ключевая информация хранится пользователем на внешнем носителе (ключевая дискета, iButton и т.д.), этот носитель может быть потерян и затем найден злоумышленником, а может быть просто украден.

3. Подбор пароля. Здесь просто упомянем о существовании подобной угрозы. Подробно эта угроза описана в ниже.

4. Сборка «мусора». Во многих ОС информация, уничтоженная пользователем, не удаляется физически, а помечается как уничтоженная. С помощью специальных программных средств (утилит) эта информация (так называемый мусор) может быть в дальнейшем восстановлена. Суть данной атаки заключается в том, что злоумышленник восстанавливает эту информацию, просматривает ее и копирует интересующие его фрагменты. По окончании просмотра и копирования вся эта информация вновь «уничтожается».

Сборка «мусора» может осуществляться не только на дисках компьютера, но и в оперативной памяти. В этом случае специальная программа, запущенная злоумышленником, выделяет себе всю или почти всю доступную оперативную память, просматривает ее содержимое и копирует фрагменты, содержащие заранее определенные ключевые слова. Если ОС не предусматривает очистку памяти при выделении, злоумышленник может получить таким об-

разом много интересной для него информации, например содержание области памяти, только что освобожденной текстовым редактором, в котором редактировался конфиденциальный документ.

5. Превышение полномочий. При реализации данной угрозы злоумышленник, используя ошибки в программном обеспечении ОС и/или политике безопасности, получает полномочия, превышающие те, которые ему предоставлены в соответствии с политикой безопасности. Обычно это достигается путем запуска программы от имени другого пользователя или подмены динамически подгружаемой библиотеки.

Эта угроза представляет наибольшую опасность для ОС, в которых допускается временное повышение полномочий пользователя (например, в ОС UNIX до полномочий суперпользователя root).

6. Программные закладки. Программные закладки, внедряемые в ОС, не имеют существенных отличий от других классов программных закладок.

7. Жадные программы. Жадными называются программы, преднамеренно захватывающие значительную часть ресурсов компьютера, в результате чего другие программы не могут выполняться или выполняются крайне медленно и неэффективно. Часто запуск жадной программы приводит к краху ОС.

§ 2. ПОНЯТИЕ ЗАЩИЩЕННОЙ ОПЕРАЦИОННОЙ СИСТЕМЫ

Мы будем называть ОС защищенной, если она предусматривает средства защиты от основных классов угроз, описанных выше. Такая система обязательно должна содержать средства разграничения доступа пользователей к своим ресурсам, а также

средства проверки подлинности пользователя, начинающего работу с ОС. Кроме того, защищенная ОС должна содержать средства противодействия случайному или преднамеренному выводу ее из строя.

Стандарты защищенности и адекватная политика безопасности

Как известно, в области обеспечения информационной безопасности в различных странах существует большой объем нормативно-методической литературы, в которой приводятся требования к защитным сервисам ОС. Приведем нужные определения и примеры таких требований.

Операционная система — взаимосвязанная совокупность программных модулей (модулей ОС) и объектов (или данных; далее по тексту — объекты ОС или данные ОС), обеспечивающая выполнение пользовательских и защитных свойств на всех этапах ее жизненного цикла.

Дополняемые прикладные приложения (ДПП) — последовательность команд ОС, составляющая функционально законченный фрагмент и локализованная в перезаписываемой памяти.

Модуль ОС (код модуля ОС) — последовательность символов заданного алфавита, отвечающая кодированию команд и использованию ими (командами) данных для конкретной аппаратной компоненты компьютера (или, иначе говоря, последовательность команд уровня аппаратного представления).

Команды ОС — однозначно специфицированные функции управления данными ОС, обеспечения пользовательских свойств компьютера, взаимодействия его с периферийными устройствами. Выделяются внешние команды (для общения с внешними устройствами) и команды ДПП.

Правила разграничения доступа (ПРД) — формально описанное однозначное представление отношения «субъект» (модуль ОС или ДПП) — «объект» (данные или объекты ОС).

Монитор обращений (МО) — программно-техническая реализация отношения «субъект — объект» для всех субъектов и объектов, локализованная в коде модуля (модулей) ОС.

Критическая ситуация — ситуация, связанная с нарушением пользовательских свойств и свойств безопасности информации, зафиксированная в состоянии программно-технических средств.

Пользовательские свойства компьютера — набор свойств, выполняемых в конкретной информационной системе и связанных с решаемой этой системой задачей по отношению к пользователю.

Этапы жизненного цикла — производство (запись кода ОС в постоянное запоминающее устройство (ПЗУ)), преперсонализация или персонализация изготовителя (запись ДПП и данных ОС в перезаписываемую память), персонализация (авторизация), эксплуатация (использование в конкретной информационной системе), разрешение критических ситуаций, уничтожение ОС (вывод из действия).

Монитор безопасности (или монитор контроля доступа) — модуль (модули) ОС, делающий однозначный вывод о возможности реализации отношения «субъект — объект» на основе ПРД и информации, полученной от МО.

Блокирование компьютера — состояние программно-технических средств, при котором для пользователя полностью или частично недоступна реализация пользовательских свойств. Состояние блокирования возникает, как правило, из-за критической ситуации.

Аудит — совокупность алгоритмов, способов их реализации и применения, ставящих целью подтверждение корректности

функционирования ОС и технических средств компьютера, используемых в период эксплуатации или при разрешении критических ситуаций.

Требования к архитектуре и совместимости

1. ОС должна иметь архитектуру, предусматривающую однозначное преобразование команд ОС на уровень их аппаратного представления.

2. ОС должна содержать следующий минимальный набор модулей:

- модуль инициализации и тестирования ОС;
- модуль сканирования команд ОС;
- модуль — интерпретатор команд;
- модуль инициирования работы ДПП;
- модуль фильтрации команд;
- модуль обеспечения связи с внешними устройствами (терминалом);
- модуль обработки обращения к данным;
- модуль криптографических функций;
- модуль реализации ПРД;
- модуль аудита состояния ОС.

Допускается реализация дополнительных модулей, не изменяющих и не блокирующих ни в каком случае выполнение функций вышеперечисленных модулей. Допускается декомпозиция перечисленных модулей на несколько взаимосвязанных модулей, выполняющих тождественные функции.

3. Команды ОС делятся на привилегированные и непривилегированные. Перед выполнением команды обязателен анализ кор-

ректности кода команды ОС (принадлежности множеству допустимых команд и проверка корректности аргументов (операндов) команды), который осуществляется модулем фильтрации команд.

4. Выполнение команд ОС подразумевает их преобразование к последовательности команд уровня аппаратного представления и реализуется только модулем — интерпретатором команд.

5. Не допускается семантическое тождество (по коду и формату команды) команд ОС и команд уровня аппаратного представления.

6. Любая привилегированная команда выполняется только после предъявления в качестве аргумента (операнда) команды секретного кода (пароля), заданного на одном из этапов жизненного цикла карты для конкретной команды, в случае удостоверения тождества предъявленного секретного кода ранее заданному.

Предъявление секретного кода для привилегированной команды может распространяться только на некоторые подмножества ее аргументов (например, для команды обращения к объектам ОС может требоваться секретный код только для избранных объектов).

7. В состав команд ОС может вводиться команда назначения и изменения секретного кода для выполнения привилегированной команды. Указанная команда также является привилегированной. Секретный код для нее задается на этапе изготовления или персонализации.

8. Длина секретного кода должна составлять не менее 6 байт. При его формировании (кроме кодов, вводимых владельцем карты) должен использоваться датчик случайных чисел (ДСЧ) с заданными статистическими характеристиками так, чтобы вектор формировался случайно и равновероятно по схеме с возвращением.

9. Привилегированные команды с использованием секретного кода должны выполняться таким образом, чтобы:

– трудоемкость восстановления секретного кода по информации в канале «терминал — карта» была не менее 1 020 элементарных операций;

– трудоемкость восстановления секретного кода по всему множеству хранимых в оперативной памяти массивов информации, зависящих от данного секретного кода, была не менее 1 020 элементарных операций.

10. Привилегированными также должны быть следующие команды:

– инициирование выполнения ДПП;

– обращение к модулю криптографических функций, связанных с управлением ключами.

Возможны дополнительные привилегированные команды.

11. Допускается выборочная реализация команд ОС из числа предусмотренных стандартом ISO 7816-4.

12. Допускается реализация дополнительных команд, не предусмотренных стандартом (проблемно ориентированные команды).

Требования к данным и их использованию

1. В ОС должны использоваться принципы структуризации данных (организации файловой структуры), предусмотренные стандартом ISO 7816-4.

2. Допускается неполная реализация принципов структуризации, определенных в стандарте.

3. Допускается реализация дополнительных принципов структуризации, не противоречащих стандарту (использование проблемно ориентированных типов).

4. Данные (объекты) ОС должны или уничтожаться привилегированной командой, или отсутствовать.

5. Хранение данных допускается в открытом, шифрованном видах, а также с фиксацией целостности. Должны быть предусмотрены дополнительные массивы информации, необходимые для реализации криптографических преобразований данных.

6. Все обращения к данным должны производиться только через модуль обработки обращения к данным. Программная реализация модуля обработки обращения к данным должна представлять собой монитор ссылок.

7. Должна быть предусмотрена возможность ограничения доступа к данным (в рамках ПРД) на основе секретного кода. Требования к механизмам применения секретного кода для данных совпадают с требованиями к механизмам реализации привилегированных команд.

Содержание следующего документа (автор — Гостехкомиссия при президенте России, 1972 г.), касающегося автоматизированных систем, вводит стандарты защищенности не отдельных программно-аппаратных модулей защиты, а всей защищаемой компьютерной системы в целом (в том числе ОС). Система стандартов этого документа более существенно отличается, например, от аналогичной системы стандартов «Оранжевой книги» (США). Все автоматизированные системы разделяются на три группы, в каждой из которых вводится своя иерархия классов защиты. Всего вводится девять классов защиты, требования которых в применении к ОС излагаются ниже (очень упрощенно).

ГРУППА 3. Однопользовательские системы.

Класс 3Б. Проверка подлинности пользователя при входе в систему. Регистрация входа и выхода пользователей из системы. Учет используемых внешних носителей информации.

Класс 3А. Выполняются все требования класса 3Б. Регистрация распечатки документов. Физическая очистка очищаемых областей оперативной и внешней памяти.

ГРУППА 2. Многопользовательские системы, в которых пользователи имеют одинаковые полномочия доступа ко всей информации.

Класс 2Б. Проверка подлинности пользователя при входе в систему. Регистрация входа и выхода пользователей из системы. Учет используемых внешних носителей информации.

Класс 2А. Выполняются все требования класса 2Б. Избирательное разграничение доступа. Регистрация событий, потенциально опасных для поддержания защищенности системы. Физическая очистка очищаемых областей оперативной и внешней памяти. Наличие подсистемы шифрования конфиденциальной информации, использующей сертифицированные алгоритмы.

ГРУППА 1. Многопользовательские системы, в которых пользователи имеют различные полномочия доступа к информации.

Класс 1Д. Проверка подлинности пользователя при входе в систему. Регистрация входа и выхода пользователей из системы. Учет используемых внешних носителей информации.

Класс 1Г. Выполняются все требования класса 1Д. Избирательное разграничение доступа. Регистрация событий, потенциально опасных для поддержания защищенности системы. Физическая очистка очищаемых областей оперативной и внешней памяти.

Класс 1В. Выполняются все требования класса 1Г. Полномочное разграничение доступа. Усилены требования к подсистеме регистрации событий, потенциально опасных для поддержания защищенности системы. Интерактивное оповещение администраторов системы о попытках несанкционированного доступа.

Класс 1Б. Выполняются все требования класса 1В. Наличие подсистемы шифрования конфиденциальной информации, использующей сертифицированные алгоритмы.

Класс 1А. Выполняются все требования класса 1Б. Различные субъекты доступа имеют различные ключи, используемые для шифрования конфиденциальной информации.

Нетрудно видеть, что требования к защищенности систем классов ЗБ, 2Б и 1Д (минимальная защита в каждой группе) совпадают и представляют собой несколько усиленную версию класса С1 «Оранжевой книги» (и соответствующего ему класса 6 для средств вычислительной техники). Требования класса 1Г в основном совпадают с требованиями класса С2 «Оранжевой книги», а класс 1В из рассматриваемого документа очень похож на класс В1 «Оранжевой книги». Таким образом, отличия требований Гостехкомиссии от требований «Оранжевой книги» в наиболее распространенном диапазоне защищенности операционных систем (С2 — В1 по «Оранжевой книге» и 1Г — 1В по документам Гостехкомиссии) незначительны.

Для примера рассмотрим некоторые (далеко не все) требования к конфигурации ОС Windows NT, которые должны выполняться для соответствия защищенности ОС классу С2 «Оранжевой книги»:

– загрузка компьютера невозможна без ввода пароля;

– на жестких дисках используется только файловая система NTFS;

– запрещено использование паролей короче шести символов;

– запрещена эмуляция OS/2 и POSIX;

– запрещен анонимный и гостевой доступ;

– запрещен запуск любых отладчиков;

– тумблер питания и кнопка RESET недоступны пользователям;

– запрещено завершение работы ОС (shut down) без входа пользователя в систему;

– политика безопасности в отношении аудита построена таким образом, что при переполнении журнала безопасности ОС прекращает работу (зависает). После этого восстановление работоспособности системы может быть произведено только администратором;

– запрещено разделение между пользователями ресурсов сменных носителей информации (флоппи-дисков, CD-ROM и т.д.);

– запись в системную директорию и файлы инициализации ОС разрешена только администраторам и системным процессам.

Определяя адекватную политику безопасности, администратор ОС должен в первую очередь ориентироваться на ее защиту от актуальных угроз ее безопасности. К сожалению, в настоящее время часто встречается ситуация, когда администратор формирует требования к политике безопасности исходя не из комплекса угроз, защиту от которых нужно реализовать, а из некоторых абстрактных рекомендаций по поддержанию безопасности той или иной ОС. В результате не исключена ситуация, когда ОС, сертифицированная по очень высокому классу защиты, оказывается уязвимой для некоторых угроз даже при соответствии политики безопасности требованиям соответствующего класса защиты. Например, ОС, защищенная по классу С2 «Оранжевой книги», уязвима для несанкционированных действий администратора.

Глава 2
АППАРАТНОЕ ОБЕСПЕЧЕНИЕ СРЕДСТВ ЗАЩИТЫ

§ 1. ЗАДАЧИ АППАРАТНОГО ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ИНФОРМАЦИИ

Под аппаратным обеспечением средств защиты ОС традиционно понимается совокупность средств и методов, используемых для решения следующих задач:

– управление оперативной и виртуальной памятью компьютера;

– распределение процессорного времени между задачами в многозадачной операционной системе;

– синхронизация выполнения параллельных задач в многозадачной ОС;

– обеспечение корректности совместного доступа задач к ресурсам ОС;

– исключение тупиковых ситуаций в процессе совместного доступа задач к ресурсам ОС.

§ 2. АППАРАТНАЯ ЗАЩИТА В ПРОЦЕССОРАХ СЕМЕЙСТВА X86/X64

Подавляющее большинство современных персональных компьютеров имеет процессор одной из моделей семейства Intel x86/x64. При этом функции аппаратной защиты, начиная с модели i386, изменились незначительно, и большинство современных операционных систем используют защитные функции только этой модели. Процессоры модели i386 и более поздних моделей могут

работать в одном из трех режимов: реальном режиме, защищенном режиме и режиме эмуляции виртуального 8086 (виртуальный режим). При старте процессора он начинает работу в реальном режиме, в котором защитные функции не поддерживаются. Виртуальный режим процессора предназначен для выполнения в защищенном режиме программ, предназначенных для работы в реальном режиме. В дальнейшем, если не оговорено противное, мы будем рассматривать защищенный режим процессора, когда поддерживается возможность реализации виртуальной памяти со страничной организацией.

Адресация оперативной памяти

Программа, выполняющаяся на процессоре, обращаясь к фрагменту оперативной памяти (байту, слову или двойному слову), должна указать адрес этого фрагмента. Этот адрес складывается из двух составляющих: селектора и смещения. Селектор представляет собой идентификатор сегмента, в котором располагается требуемый фрагмент памяти, а смещение определяет порядковый номер первого байта фрагмента в этом сегменте.

При обращении к оперативной памяти необходимый селектор должен быть заранее загружен в один из сегментных регистров процессора cs, ds, es, fs, gs или ss. Для адресации данных может использоваться любой сегментный регистр, для адресации кода — только регистр cs, для адресации стека — только ss. Каждый сегментный регистр имеет объем 16 бит.

С каждым сегментным регистром связан один дескрипторный регистр. Дескрипторные регистры не могут напрямую использоваться программами и поэтому не имеют имен. Длина каждого дескрипторного регистра 64 бита. При загрузке селектора в сегментный регистр в дескрипторный регистр, соответствующий

этому сегментному регистру, автоматически загружается дескриптор сегмента, соответствующего загружаемому селектору. Дескриптор сегмента содержит линейный адрес начала сегмента в оперативной памяти (при отключенной страничной адресации этот адрес совпадает с физическим), длину сегмента и ряд атрибутов сегмента. Таким образом, всегда, когда в сегментный регистр загружен некоторый селектор, в соответствующий дескрипторный регистр загружен дескриптор сегмента, идентифицируемый данным селектором.

Сегменты, которым соответствуют различные дескрипторы, могут пересекаться и даже совпадать.

Дескриптор сегмента может быть глобальным или локальным. В первом случае дескриптор доступен всем задачам (процессам или потокам), выполняющимся в системе, во втором случае — только одной задаче. Глобальные дескрипторы хранятся в специальном сегменте, называемом таблицей глобальных дескрипторов. Объем таблицы глобальных дескрипторов ограничен значением 64 Kb, следовательно, может одновременно существовать до 8 192 глобальных дескрипторов. Физический адрес таблицы глобальных дескрипторов хранится в 48-битовом регистре gdtr.

Каждая задача, выполняемая в системе, имеет таблицу локальных дескрипторов, содержащую дескрипторы, доступные только этой задаче. Таблица локальных дескрипторов задачи представляет собой сегмент специального типа. В процессе выполнения задачи регистр процессора Idtr объемом 16 бит содержит селектор этого сегмента. При переключении задач регистр idtr автоматически перегружается.

Уровни привилегированности

Защита оперативной памяти основана на понятии уровня привилегированности. Уровень привилегированности (privilegelevel,

PL) — это числовой идентификатор, принимающий значения от 0 до 3, который определяет возможности задачи выполнять команды процессора, модифицировать регистры и области оперативной памяти и т.д. Чем меньше числовое значение уровня привилегированности, тем выше этот уровень и тем более полный доступ имеет задача к аппаратным возможностям процессора. Множество задач, обладающих некоторым конкретным уровнем привилегированности, называют кольцом защиты. Например, если код программы имеет уровень привилегированности, равный трем, говорят, что программа выполняется в третьем кольце защиты (или просто в третьем кольце).

Обычно прикладные программы выполняются в третьем кольце защиты, а код ОС — в нулевом кольце. Первое и второе кольца защиты используются редко. Из известных авторам ОС эти кольца защиты используются только в OS/2. Видимо, это объясняется тем, что большинство RISC-процессоров поддерживают только два кольца защиты, и переносимые ОС вынуждены учитывать это ограничение.

Итак, каждая задача, выполняющаяся на процессоре, имеет свой уровень привилегированности, который называют текущим уровнем привилегированности (currentprivilegelevel, CPL).

Каждый дескриптор и каждый селектор также имеют свои уровни привилегированности, называемые соответственно уровнем привилегированности дескриптора (descriptorprivilegelevel, DPL) и запрашиваемым уровнем привилегированности (requestedprivilegedlevel, RPL). Фактически текущий уровень привилегированности задачи (CPL) представляет собой не что иное, как уровень привилегированности селектора (RPL), загруженного в данный момент в регистр cs.

Защита сегментов оперативной памяти

Защита сегментов оперативной памяти в процессоре основана на сравнении уровня привилегированности задачи (CPL), уровня привилегированности дескриптора (DPL) и уровня привилегированности селектора (RPL) в момент загрузки селектора в сегментный регистр. Фактически при загрузке селектора в сегментный регистр выполняются следующие три проверки:

1. Проверяется корректность селектора. Если индекс, содержащийся в селекторе, превышает объем таблицы дескрипторов (локальной или глобальной в зависимости от селектора), генерируется исключительная ситуация 11 (отсутствие сегмента) или 12 (ошибка стека) в зависимости от типа загружаемого сегмента.

2. Проверяется совместимость типа загружаемого селектора с типом сегментного регистра. В регистр cs может быть загружен только селектор сегмента кода, в регистр ss — только селектор сегмента стека, в любой другой сегментный регистр — либо селектор сегмента данных, либо селектор сегмента стека, либо селектор доступного для чтения сегмента кода. В случае несовпадения типов селектора и сегментного регистра генерируется исключительная ситуация 13 (общая ошибка защиты).

3. Проверяется достаточность текущего уровня привилегированности задачи для загрузки сегмента. Если загружаемый сегмент является сегментом стека, он может быть загружен только при точном совпадении CPL, RPL и DPL. Если же загружаемый сегмент не является сегментом стека, то для его успешной загрузки необходимо и достаточно, чтобы и CPL, и RPL были не ниже, чем DPL. В случае неудачной проверки генерируется исключительная ситуация 13.

Таким образом, задача может загрузить селектор сегмента стека в регистр ss тогда и только тогда, когда совпадают все три уровня привилегированности задачи, селектора и дескриптора сег-

мента. Отсюда следует, что при изменении уровня привилегированности задачи должен быть изменен сегмент стека задачи. Это происходит автоматически в процессе выполнения процессором машинных команд передачи управления из сегмента в сегмент.

Селектор сегмента кода или данных может быть загружен в сегментный регистр тогда и только тогда, когда задача имеет уровень привилегированности не ниже уровня привилегированности дескриптора сегмента и для доступа к сегменту используется селектор достаточно высокого уровня привилегированности. Низко привилегированная задача не может обращаться к высоко привилегированным сегментам ни при каких обстоятельствах. Высоко привилегированная задача может понизить свой уровень привилегированности в отношении конкретного сегмента, загрузив в сегментный регистр низко привилегированный селектор.

Если в сегментный регистр cs загружается селектор, указывающий на дескриптор с уровнем привилегированности, отличным от уровня привилегированности дескриптора, на который указывал селектор, ранее загруженный в регистр cs, изменяется текущий уровень привилегированности задачи. При этом уровень привилегированности задачи может только понижаться. Действительно, если производится попытка повышения уровня привилегированности задачи путем загрузки в cs селектора, указывающего на более привилегированный дескриптор, имеет место неравенство CPL>DPL, и третья проверка корректности загрузки селектора приводит к общей ошибке защиты.

Селекторы и дескрипторы сегментов

Выше в общих чертах было описано назначение селекторов и дескрипторов при адресации памяти в защищенном режиме процессора. Теперь более подробно рассмотрим форматы этих структур данных.

Формат дескриптора сегмента выглядит следующим образом: {младшие 16 бит границы сегмента; младшие 24 бита базового адреса сегмента; байт защиты сегмента; старшие 4 бита границы сегмента; флаг D: если он установлен, в сегменте по умолчанию применяется 32-разрядная адресация; флаг G: если он сброшен, границы сегмента измеряются в байтах, в противном случае в четырех килобайтовых страницах; старшие 8 бит базового адреса сегмента}.

При отключенной страничной адресации оперативной памяти базовый адрес сегмента представляет собой физический адрес начала сегмента в оперативной памяти. Граница сегмента для сегментов стека равна (размер сегмента), для других сегментов (размер_сегмента_). Дело в том, что в сегментах кода и данных смещение может принимать значения от 0 до (размер_сегмента_), а в сегментах стека, поскольку стек растет сверху вниз, от (размер сегмента) до 1. Содержание байта защиты сегмента различается в зависимости от типа сегмента.

Процессор поддерживает следующие типы сегментов:

- сегменты кода;
- сегменты данных;
- сегменты стека;
- локальные таблицы дескрипторов;
- сегменты состояния задач.

Сегменты двух последних типов называют системными сегментами, а их дескрипторы — системными дескрипторами (также к системным дескрипторам относят шлюзы, которые будут рассмотрены далее).

Бит 54-го дескриптора сегмента в зависимости от типа сегмента имеет следующий смысл:

– для сегментов кода и данных — разрядность сегмента; если этот бит установлен, в сегменте применяется по умолчанию 32-разрядная адресация, в противном случае 16-разрядная;

– для сегментов стека — разрядность стека; если этот бит установлен, вершина стека определяется 32-разрядным регистром esp и верхняя граница стека равна OxFFFFFFFF, в противном случае вершина стека определяется 16-разрядным регистром sp и верхняя граница стека равна OxFFFF;

– для других сегментов этот бит всегда равен нулю.

Содержание байта защиты для сегментов различных типов описывается ниже.

Байт защиты дескриптора сегмента кода. Смещение битов и содержание соответственно выглядят таким образом:

(0A) Если этот бит установлен, к сегменту осуществлялся доступ, в противном случае обращений к сегменту не было.

(1R) Если этот бит установлен, код, содержащийся в сегменте, может как выполняться, так и считываться, в противном случае только выполняться.

(2C) Если этот бит установлен, сегмент является согласованным, и код, содержащийся в нем, может параллельно выполняться несколькими задачами с различными уровнями привилегированности.

(3E)1.

(4S)1.

(56DPL) Уровень привилегированности дескриптора.

(7P) Если этот бит установлен, сегмент загружен в физическую память, в противном случае сегмент вытеснен на диск или отсутствует.

Формат байта защиты дескриптора сегмента стека отличается от формата, описанного ниже, только битом ED, который для

дескрипторов сегментов стека равен единице. Формат байта защиты дескриптора сегмента состояния задачи отличается от байта формата только значением поля TYPE, которое может принимать для дескрипторов сегментов состояния задач значения 1, 3, 8 или 11.

Итак, байт защиты дескриптора описывает тип сегмента, атрибуты сегмента и уровень привилегированности дескриптора сегмента.

Байт защиты дескриптора сегмента данных. Смещение битов и содержание соответственно выглядят таким образом:

(0A) Если этот бит установлен, к сегменту осуществлялся доступ, в противном случае обращений к сегменту не было.

(1W) Если этот бит установлен, данные, содержащиеся в сегменте, доступны как для чтения, так и для записи, в противном случае только для чтения.

(2ED)0.

(3E)0.

(4S)1.

(56DPL) Уровень привилегированности дескриптора.

(7P) Если этот бит установлен, сегмент загружен в физическую память, в противном случае сегмент вытеснен на диск или отсутствует.

Байт защиты дескриптора таблицы локальных дескрипторов. Смещение битов и содержание соответственно выглядят таким образом:

(03)TYPE 2.

(4S)0.

(56DPL) Уровень привилегированности дескриптора.

(7P) Если этот бит установлен, сегмент загружен в физическую память, в противном случае сегмент вытеснен на диск или отсутствует.

Тип сегмента определяется следующим образом:

- для сегментов кода S=1, E=1;
- для сегментов данных S=1, E=0, ED=0;
- для сегментов стека S=1, E=0, ED=1;
- для таблиц локальных дескрипторов S=0, TYPE=2;
- для сегментов состояния задач S=0, TYPE=1,3,8,11.

Пример обращения процессора к оперативной памяти без страничной адресации. Рассмотрим ситуацию, когда процессор обращается к фрагменту оперативной памяти, используя селектор, уже загруженный в сегментный регистр. Пусть процессор выполняет машинную команду

moves:[ebx+4],eax

При этом реализуются следующие действия:

1. Выполняется проверка бита E дескриптора d, загруженного в дескрипторный регистр, соответствующий сегментному регистру es. Если этот бит равен единице (производится попытка записи в сегмент кода), генерируется исключительная ситуация 13 (общая ошибка защиты). Выполнение текущей машинной команды прерывается.

2. Если предыдущая проверка прошла успешно (E=0), выполняется проверка бита W дескриптора d. Если этот бит равен нулю (сегмент доступен только для чтения), генерируется исключительная ситуация 13. Выполнение текущей машинной команды прерывается.

3. Если W=1, из дескриптора d извлекается граница сегмента seg_border. Если бит G дескриптора d равен единице, выполняется присваивание seg_border=4096.

4. Вычисляется смещение offset=ebx+4.

5. Если для дескриптора d ED=0 (сегмент является сегментом данных), производится сравнение offset+3 (поскольку в данном случае команда mov пытается записать в оперативную память

4 байта, при проверке границы сегмента к смещению прибавляется 41=3) и seg_border. Если offset+3>seg_border, фиксируется выход за пределы сегмента и генерируется исключительная ситуация 13. Выполнение текущей машинной команды прерывается.

6. Если для дескриптора d ED=1 (сегмент является сегментом стека), производится сравнение offset и seg_border. Если offset<seg_border, фиксируется выход за пределы сегмента и генерируется исключительная ситуация 13. Выполнение текущей машинной команды прерывается.

7. Если выход за пределы сегмента не зафиксирован, из дескриптора d извлекается физический адрес начала сегмента seg_start.

8. Вычисляется физический адрес в оперативной памяти запрашиваемого двойного слова, равный seg_start+offset.

9. Значение регистра eax копируется в двойное слово, расположенное по этому адресу.

Итак, при обращении задачи к оперативной памяти выполняются следующие проверки:

– допустимость операции записи для сегмента;

– наличие выхода за границы сегмента для указанного смещения.

Если обращение к оперативной памяти производится не для записи, как в приведенном примере, а для чтения, шаги 1 и 2 данного алгоритма пропускаются.

Глава 3
ТИПОВАЯ АРХИТЕКТУРА ПОДСИСТЕМЫ ЗАЩИТЫ ОПЕРАЦИОННОЙ СИСТЕМЫ

§ 1. РАЗГРАНИЧЕНИЕ ДОСТУПА К ОБЪЕКТАМ ОПЕРАЦИОННОЙ СИСТЕМЫ

Основные определения

Объектом доступа (или просто объектом) будем называть любой элемент ОС, доступ к которому пользователей и других субъектов доступа может быть произвольно ограничен. Ключевым словом в данном определении является слово «произвольно». Если правила, ограничивающие доступ субъектов к некоторому элементу ОС, определены жестко и не допускают изменения с течением времени, этот элемент ОС не будет считаться объектом. Другими словами, возможность доступа к объектам определяется не только архитектурой ОС, но и текущей политикой безопасности.

Методом доступа к объекту называется операция, определенная для некоторого объекта. Например, для файлов могут быть определены методы доступа «чтение», «запись» и «добавление» (дописывание информации в конец файла).

Субъектом доступа (или просто субъектом) будем называть любую сущность, способную инициировать выполнение операций над объектами (обращаться к объектам по некоторым методам доступа). Например, пользователи являются субъектами доступа.

В литературе, посвященной компьютерной безопасности, до сих пор не сформировалось единого представления о том, что такое субъект доступа. Часто множество субъектов доступа считают подмножеством множества объектов защищенной системы. Такой подход эффективен при описании формальных моделей политики безопасности, но при рассмотрении конкретных систем защиты

более удобно считать, что множество субъектов доступа и множество объектов доступа не пересекаются. Будем придерживаться второго подхода.

Рассмотрим далее типовые модели разграничения доступа.

Избирательное разграничение доступа. Система правил избирательного, или дискреционного, разграничения доступа (discretionaryaccesscontrol, DAC) формулируется следующим образом:

1. Для любого объекта ОС существует владелец.

2. Владелец объекта может произвольно ограничивать доступ других субъектов к данному объекту.

3. Для каждой тройки «субъект — объект — метод» возможность доступа определена однозначно.

4. Существует хотя бы один привилегированный пользователь (администратор), имеющий возможность обратиться к любому объекту по любому методу доступа. Это не означает, что данный пользователь может игнорировать разграничение доступа к объектам и поэтому является суперпользователем. Не всегда для реализации возможности доступа к объекту ОС администратору достаточно просто обратиться к объекту. Например, в Windows NT администратор для обращения к чужому (принадлежащему другому субъекту) объекту должен вначале объявить себя владельцем этого объекта, использовав привилегию администратора объявлять себя владельцем любого объекта, затем дать себе необходимые права, и только после этого администратор может обратиться к объекту. При этом использование администратором своей привилегии не остается незамеченным для прежнего владельца объекта.

Последнее требование введено для реализации механизма удаления потенциально недоступных объектов.

При создании объекта его владельцем назначается субъект, создавший данный объект. В дальнейшем субъект, обладающий необходимыми правами, может назначить объекту нового владельца. Обычно при изменении владельца объекта допускается назначать новым владельцем объекта только субъекта, изменяющего владельца объекта. Другими словами, субъект, изменяющий владельца объекта, может назначить новым владельцем объекта только себя. Такое ограничение вводится для того, чтобы владелец объекта не мог отдать «владение» объектом другому субъекту и тем самым снять с себя ответственность за некорректные действия с объектом.

Для определения прав доступа субъектов к объектам при избирательном разграничении доступа используется матрица доступа. Строки этой матрицы представляют собой объекты, столбцы — субъекты (или наоборот). В каждой ячейке матрицы хранится совокупность прав доступа, предоставленных данному субъекту на данный объект.

Поскольку матрица доступа достаточно велика, она никогда не хранится в системе в явном виде. Для сокращения объема матрицы доступа используется объединение субъектов доступа в группы. Права, предоставленные группе субъектов для доступа к данному объекту, предоставляются каждому субъекту группы.

Вместе с каждым объектом доступа хранятся его атрибуты защиты, описывающие, кто является владельцем объекта и каковы права доступа к данному объекту различных субъектов. Атрибуты защиты фактически представляют собой совокупность идентификатора владельца объекта и строку матрицы доступа в кодированном виде.

На практике используются два способа кодирования строки матрицы доступа:

1. Вектор доступа (UNIX) — вектор фиксированной длины, разбитый на несколько подвекторов. Каждый подвектор описывает права доступа к данному объекту некоторого субъекта. С помощью вектора доступа можно описать права доступа к объекту только фиксированного числа субъектов, что накладывает существенные ограничения на систему разграничения доступа.

2. Список доступа (VAX/VMS, Windows NT) — список переменной длины, элементами которого являются структуры, содержащие:

- идентификатор субъекта;
- права, предоставленные этому субъекту на данный объект;
- различные флаги и атрибуты.

Фактически, вектор доступа представляет собой список доступа фиксированной длины и является частным случаем списка доступа.

Кодирование матрицы доступа в виде совокупности списков доступа позволяет реализовать мощный и гибкий механизм разграничения доступа, однако требует гораздо больше оперативной и дисковой памяти для хранения атрибутов защиты объекта, усложняет техническую реализацию правил разграничения доступа и создает проблему, связанную с тем, что значения элементов списка доступа могут противоречить друг другу. Предположим, один элемент списка доступа разрешает некоторому пользователю доступ к объекту, а другой элемент списка запрещает доступ к объекту группе, в которую входит этот пользователь. При использовании списков доступа правила разграничения доступа должны включать в себя правила разрешения этих противоречий.

При создании нового объекта владелец объекта должен определить права доступа различных субъектов к этому объекту. Если владелец объекта не сделал этого, то либо новому объекту назначаются атрибуты защиты по умолчанию, либо новый объект

наследует атрибуты защиты от родительского объекта (каталога, контейнера и т.д.).

Изолированная программная среда

Изолированная, или замкнутая, программная среда представляет собой расширение модели избирательного разграничения доступа. Здесь правила разграничения доступа формулируются следующим образом:

1. Для любого объекта ОС существует владелец.
2. Владелец объекта может произвольно ограничивать доступ других субъектов к данному объекту.
3. Для каждой четверки «субъект — объект — метод — процесс» возможность доступа определена однозначно.
4. Существует хотя бы один привилегированный пользователь (администратор), имеющий возможность обратиться к любому объекту по любому методу.
5. Для каждого субъекта определен список программ, которые этот субъект может запускать.

При использовании изолированной программной среды права субъекта на доступ к объекту определяются не только правами и привилегиями субъекта, но и процессом, с помощью которого субъект обращается к объекту. Можно, например, разрешить обращаться к файлам с расширением .doc(x) только программам Word, Word Viewer и WPview.

Изолированная программная среда существенно повышает защищенность ОС от разрушающих программных воздействий, включая программные закладки и компьютерные вирусы. Кроме того, при использовании данной модели повышается защищенность целостности данных, хранящихся в системе. В то же время изолированная программная среда создает определенные слож-

ности в администрировании ОС. Например, при инсталляции нового программного продукта администратор должен модифицировать списки разрешенных программ для пользователей, которые должны иметь возможность работать с этим программным продуктом.

Изолированная программная среда не защищает от утечки конфиденциальной информации.

§ 2. ИДЕНТИФИКАЦИЯ, АУТЕНТИФИКАЦИЯ И АВТОРИЗАЦИЯ СУБЪЕКТОВ ДОСТУПА

Идентификация и аутентификация с помощью биометрических характеристик пользователей

Каждый человек обладает своим неповторимым набором биометрических характеристик, к которым относятся отпечатки пальцев, рисунок сетчатки, рукописный и клавиатурный почерк и т.д. Эти характеристики могут быть использованы для аутентификации пользователя.

Если аутентификация пользователя осуществляется на основе биометрических характеристик, угрозы кражи и подбора ключевой информации перестают быть актуальными: подделать биометрические характеристики человека, как правило, настолько дорого, что затраты злоумышленника на проникновение в защищенную ОС превысят выгоды от такого проникновения. Таким образом, механизм аутентификации пользователя на основе биометрических характеристик создает практически непреодолимую защиту на этапе аутентификации.

С другой стороны, техническая реализация данного механизма аутентификации неизбежно создает следующие проблемы:

– Поскольку псевдопользователи не являются людьми и, следовательно, не имеют биометрических характеристик, для их аутентификации должен поддерживаться альтернативный механизм. При этом ОС должна гарантировать, что этот альтернативный механизм не будет использоваться для аутентификации обычных пользователей.

– При двух последовательных входах в систему одного и того же человека его биометрические характеристики никогда в точности не совпадают. Поэтому в процессе аутентификации приходится использовать математический аппарат теории распознавания образов, при этом необходимо мириться с неизбежностью ошибок как первого рода (успешный вход от чужого имени), так и второго рода (отказ в доступе легальному пользователю).

– Большинство биометрических характеристик человека постепенно меняются со временем, что заставляет регулярно корректировать эталонный образ аутентифицирующей информации.

– Биометрические характеристики человека могут испытывать резкие кратковременные изменения. Например, если пользователь поцарапал палец, система аутентификации, основанная на отпечатках пальцев, не сможет его аутентифицировать до тех пор, пока царапина не заживет.

– Аутентификация пользователя на основе биометрических характеристик требует применения дорогостоящей аппаратуры для получения образа используемой характеристики и сложных вычислительных алгоритмов для сравнения этого образа с эталонным, что приводит к определенным финансовым затратам на создание системы аутентификации и затратам вычислительных ресурсов компьютера на ее поддержание.

§ 3. АУДИТ

Общие сведения

Процедура аудита применительно к ОС заключается в регистрации в специальном журнале, называемом журналом аудита (безопасности), событий, которые могут представлять опасность для ценных информационных ресурсов. Пользователи системы, обладающие правом чтения этого журнала, называются аудиторами.

Необходимость включения в защищенную ОС функций аудита диктуется следующими обстоятельствами:

1. Подсистема защиты ОС, не обладая интеллектом, не способна отличить случайные ошибки пользователей от злонамеренных действий. Например, то, что пользователь в процессе входа в систему ввел неправильный пароль, может означать как случайную ошибку при вводе пароля, так и попытку подбора пароля. Но, если сообщение о подобном событии записано в журнал аудита, администратор, просматривая этот журнал, легко сможет установить, что же имело место на самом деле: ошибка легального пользователя или атака злоумышленника. Если пользователь ввел неправильный пароль всего один раз, это явная ошибка. Если же пользователь пытался угадать собственный пароль 20–30 раз, это явная попытка подбора пароля.

2. Администраторы/аудиторы ОС должны иметь возможность получать информацию не только о текущем состоянии системы, но и о том, как она функционировала в недавнем прошлом. Журнал аудита дает такую возможность, накапливая информацию о важных событиях, связанных с безопасностью системы.

3. Если администратор/аудитор ОС обнаружил, что против системы проведена успешная атака, ему важно выяснить, когда была начата атака, кем и каким образом она осуществлялась. При

наличии в системе подсистемы аудита не исключено, что вся необходимая информация содержится в журнале.

Большинство экспертов по компьютерной безопасности сходятся во мнении, что привилегия работать с подсистемой аудита не должна предоставляться администраторам ОС. Другими словами, множество администраторов и множество аудиторов не должны пересекаться. При этом создается ситуация, когда администратор не может выполнять несанкционированные действия без того, чтобы это тут же не стало известно аудиторам, что существенно повышает защищенность системы от несанкционированных действий администраторов.

Требования к аудиту

Подсистема аудита операционной ОС должна удовлетворять следующим требованиям:

1. Только сама ОС может добавлять записи в журнал аудита. Если предоставить это право какому-то физическому пользователю, этот пользователь получит возможность компрометировать других пользователей, добавляя в журнал аудита соответствующие записи.

2. Ни один субъект доступа, в том числе и сама ОС, не имеет возможности редактировать или удалять отдельные записи в журнале аудита.

3. Только аудиторы, обладающие соответствующей привилегией, могут просматривать журнал аудита.

4. Только аудиторы могут очищать журнал аудита. После очистки журнала в него автоматически вносится запись о том, что журнал аудита был очищен, с указанием времени очистки журнала и имени пользователя, очистившего журнал. ОС должна поддерживать возможность сохранения журнала аудита перед очисткой в другом файле.

5. При переполнении журнала аудита ОС аварийно завершает работу («зависает»). После перезагрузки работать с системой могут только аудиторы. ОС переходит к обычному режиму работы только после очистки журнала аудита.

Для ограничения доступа пользователей к журналу аудита недостаточно использования обычных средств разграничения доступа. Дело в том, что в подавляющем большинстве ОС администраторы, используя свои привилегии, могут прочитать и изменить содержимое любого файла. Поэтому для ограничения доступа к журналу аудита должны применяться специальные средства защиты.

Политика аудита

Политика аудита — это совокупность правил, определяющих то, какие события должны регистрироваться в журнале аудита. Для обеспечения надежной защиты ОС в журнале аудита должны обязательно регистрироваться следующие события:

- попытки входа/выхода пользователей из системы;
- попытки изменения списка пользователей;
- попытки изменения политики безопасности, в том числе и политики аудита.

При определении политики аудита не следует ограничиваться регистрацией событий из перечисленных классов. Окончательный выбор того, какие события должны регистрироваться в журнале аудита, а какие не должны, возлагается на аудиторов. При этом политика аудита в значительной мере определяется спецификой информации, хранимой и обрабатываемой в операционной системе, и дать какие-либо рекомендации, не зная этой специфики, невозможно.

При выборе оптимальной политики аудита следует учитывать ожидаемую скорость заполнения журнала аудита. Если политика аудита предусматривает регистрацию слишком большого числа событий, это не только не повышает защищенность ОС, но, наоборот, снижает ее. Если новые записи добавляются в журнал аудита слишком часто, аудиторам будет трудно выделить в огромном объеме малоценной информации те события, которые на самом деле представляют угрозу безопасности системы. Кроме того, чем быстрее заполняется журнал аудита, тем чаще его нужно очищать и тем больше вероятность временного выхода из строя ОС из-за переполнения журнала аудита.

Политику аудита не следует рассматривать как нечто неизменное, заданное раз и навсегда. Политика аудита должна оперативно реагировать на изменения в конфигурации ОС, в характере хранимой и обрабатываемой информации и особенно на выявленные попытки атаки операционной системы. Если, например, с помощью аудита было обнаружено, что имела место попытка преодолеть защиту ОС, но основные принципы реализации этой атаки остались неясными, целесообразно изменить политику аудита таким образом, чтобы при дальнейших попытках осуществлять аналогичные атаки аудиторы получали более подробную информацию.

В целом политика аудита — это своего рода искусство, и выбор оптимальной политики в значительной мере определяется опытом и интуицией аудитора.

Глава 4
ЗАЩИТА В ОПЕРАЦИОННОЙ СИСТЕМЕ

§ 1. ОСНОВНЫЕ ПОЛОЖЕНИЯ

Операционная система UNIX (ОС UNIX) широко используется во всех областях, в том числе в организациях, где безопасности информации уделяется особое внимание. Следует отметить, что при начале разработки UNIX она не предназначалась для защиты информации. Это не означает, что современные реализации UNIX не обеспечивают полноценную реализацию механизмов безопасности. Первоначально заложенные в UNIX решения по защите информации, такие как идентификация пользователей и разграничение доступа, с некоторыми дополнениями успешно используются сегодня при построении защищенных систем. Более того, можно утверждать, что большинство систем UNIX соответствует классу безопасности C2.

Однако, изначально разрабатываясь как открытая система, она не могла не унаследовать некоторые элементы открытости. В силу этого при конфигурации системы UNIX возникает определенный набор выявленных ситуаций, которые могут приводить к нарушению механизмов безопасности. Поэтому для обеспечения безопасности ОС UNIX принципиально важны ее правильная конфигурация и настройка. Очевидно, чтобы правильно настроить систему, администратор должен четко представлять все требования, которые должны быть описаны в политике безопасности предприятия. Определив потенциальные угрозы, необходимо выяснить, какие средства существуют для обеспечения, развития и исследования защиты в ОС UNIX. При этом, учитывая вычислительные и материальные затраты на обеспечение безопасности,

надо строить систему защиты в соответствии с принципом разумной достаточности.

Следующей причиной нарушения безопасности ОС UNIX является наличие некорректно работающих программ, которые при некоторых нештатных ситуациях выполняют ряд дополнительных незапланированных действий. Нельзя сказать, что данная причина присутствует только в ОС UNIX, но и нельзя отрицать, что такие проблемы в UNIX существуют. Естественно, что мы можем описать только те ошибки, которые уже проявили себя, понимая, что всегда остается определенное количество невыявленных ошибок. Именно поэтому ни один производитель программного обеспечения не дает 100%-ную гарантию своего продукта. Ясно, что все это относится и к ОС UNIX. Справедливости ради надо добавить, что все производители систем UNIX весьма оперативно реагируют на сообщения об ошибках, и зарегистрированные пользователи могут их легко исправить.

Поскольку не существует компьютерной системы, в которой риск сведен к нулю, то употребляется термин «надежная» (trusted) вместо «безопасная» (secure). Под надежной системой будем понимать систему, в которой достигается некоторый уровень контроля над доступом к информации, обеспечивающий механизмы предотвращения или по крайней мере фиксирования несанкционированного доступа. Под предотвращением несанкционированного доступа будем понимать поддержку заданного уровня конфиденциальности (невозможности несанкционированного получения информации), целостности (невозможности несанкционированной ее модификации) и доступности (невозможности получения информации за разумное время).

Средствами защиты ОС UNIX могут быть как базовые средства, так и расширенные средства стандартных систем. При этом

расширенные средства защиты полностью совместимы с базовыми механизмами защиты ОС UNIX.

При дальнейшем изложении материала, где это возможно по смыслу, будет использоваться понятие «администратор» без уточнения его функций.

§ 2. КОНТРОЛЬ ЦЕЛОСТНОСТИ СИСТЕМЫ

Известно, что стоимость восстановления системы выше стоимости ее сопровождения. В задачи сопровождения ОС входит, в частности, контроль целостности системы. В ОС UNIX контроль целостности системы выполняется рядом команд. Например, для контроля и поддержки целостности SCO UNIX основной перечень команд системы следующий: integrity, authck, fixmog, cps, tcbck, smmck, authckrc, fsck. Практика показывает, что данный набор команд является достаточно полным.

Стандартная последовательность действий после возникновения сбоя в системе или каких-либо других отклонений следующая:

– проверка файловой системы;

– составление контрольного отчета;

– проверка базы данных аутентификации;

– проверка разрешений для системных файлов.

Следует отметить, что системные средства восстановления целостности системы работают до определенного предела. Авторами был проведен следующий эксперимент для SCO UNIX:

– всем файлам системы был назначен владелец root;

– у всех файлов системы были установлены права доступа со значением по умолчанию системной переменной umask.

В результате этих действий системными средствами не удалось восстановить нормальные права доступа и владельцев файлов. Такой эксперимент имеет практическую основу, например осуществление репликации (размножения) системы на другие компьютеры по сети. Поэтому единственным способом дублирования системы на другие компьютеры следует считать использование команды cpio. Также следует отметить, что в SCO UNIX права доступа и владельцы некоторых файлов по умолчанию не соответствуют базе данных контроля целостности системы.

§ 3. СРЕДСТВА АУДИТА

Будем считать, что действие контролируется, если можно определить реального пользователя, который его осуществил. Концептуально при построении ОС UNIX некоторые действия нельзя контролировать на уровне действий реального пользователя. Например, пользовательские бюджеты, такие как lp или uucp, используются анонимно, и их действия можно обнаружить только по изменениям в системной информации.

Система контроля регистрирует события в операционной системе, связанные с защитой информации, записывая их в контрольный журнал. В контрольных журналах возможна фиксация проникновения в систему и неправильного использования ресурсов. Контроль позволяет просматривать собранные данные для изучения видов доступа к объектам и наблюдения за действиями отдельных пользователей и их процессов. Попытки нарушения защиты и механизмов авторизации контролируются. Использование системы контроля дает высокую степень гарантии обнаружения попыток обойти механизмы обеспечения безопасности. Поскольку события, связанные с защитой информации, контролируются и

учитываются вплоть до выявления конкретного пользователя, система контроля служит сдерживающим средством для пользователей, пытающихся некорректно использовать систему.

В соответствии с требованиями по надежности операционная система должна создавать, поддерживать и защищать журнал регистрационной информации, относящейся к доступу к объектам, контролируемым системой. При этом должна быть возможность регистрации следующих событий:

– использования механизма идентификации и аутентификации;

– внесения объектов в адресное пространство пользователя (например, открытие файла);

– удаления объектов;

– действий администраторов;

– других событий, затрагивающих информационную безопасность.

Каждая регистрационная запись должна включать следующие поля:

– дата и время события;

– идентификатор пользователя;

– тип события;

– результат действия.

Для событий идентификации и аутентификации регистрируется также идентификатор устройства. Для действий с объектами регистрируются имена объектов.

Система контроля использует системные вызовы и утилиты для классификации действий пользователей, подразделяя их на события различного типа. Например, при возникновении события типа DAC Denials (отказ доступа при реализации механизма избирательного разграничения доступа) регистрируются попытки та-

кого использования объекта, которые не допускаются разрешениями для этого объекта. Иными словами, если пользовательский процесс пытается писать в файл с доступом «только для чтения», то возникает событие типа DAC Denials. Если просмотреть контрольный журнал, то легко можно увидеть повторяющиеся попытки доступа к файлам, на которые не получены разрешения.

Глава 5
ЗАЩИТА В ОПЕРАЦИОННОЙ СИСТЕМЕ

§ 1. ОБЪЕКТЫ И СУБЪЕКТЫ ДОСТУПА В WINDOWS NT

Объекты доступа Windows NT

Приведем примеры объектов доступа в ОС Windows NT и соответствующие им методы доступа.

Файл. Чтение. Запись. Добавление информации в конец. Выполнение. Получение атрибутов. Изменение атрибутов. Получение расширенных атрибутов. Изменение расширенных атрибутов.

Дисковая директория. Просмотр. Создание нового файла. Создание поддиректории. Проход (traverse). Удаление файла или поддиректории. Получение атрибутов. Изменение атрибутов. Получение расширенных атрибутов. Изменение расширенных атрибутов.

Ключ реестра. Чтение значений. Изменение значений. Создание подключа. Перечисление подключей. Требование оповещения при доступе к ключу другого потока. Создание символической связи.

Процесс. Завершение. Создание нового потока. Изменение атрибутов страниц адресного пространства. Чтение адресного пространства. Запись в адресное пространство. Дублирование дескриптора. Получение приоритета. Изменение приоритета.

Права доступа к объектам в Windows NT

Каждому специфичному методу доступа, поддерживаемому ОС Wndows NT, соответствует право на его осуществление. Эти

права доступа называются специфичными, поскольку они специфичны для каждого типа объектов. Для каждого типа объектов может поддерживаться до 16 специфичных прав доступа.

Каждому стандартному методу доступа, за исключением ACCESS_SYSTEM_SECURITY, также соответствует право доступа, дающее возможность реализации соответствующего метода доступа. Такие права доступа называются стандартными.

ОС Wndows NT поддерживает также общие (generic) или отображаемые (mapped) права доступа. Поддерживаются следующие отображаемые права доступа:

- чтение (GENERIC READ), запись (GENERIC WRITE);
- выполнение (GENERIC_EXECUTE);
- все действия (GENERIC_ALL).

Каждое из отображаемых прав доступа представляет собой некоторую комбинацию стандартных и специфичных прав доступа. Другими словами, отображаемое право дает возможность осуществить некоторый набор методов доступа к объекту. Отображаемые права доступа могут быть предоставлены для доступа к объекту любого типа, однако конкретное содержание отображаемого права доступа зависит от типа объекта. Например, отображаемое право на чтение объекта типа «файл» дает субъекту право выполнять доступ к объекту по следующим методам:

- чтение файла;
- получение DOS-атрибутов файла;
- получение расширенных атрибутов файла;
- получение атрибутов защиты файла;
- SYNCHRONIZE.

В то же время отображаемое право на чтение объекта типа «ключ реестра» дает субъекту права осуществлять доступ к объекту по следующим методам:

- чтение значений ключа;

– перечисление подключателей ключа;

– требование оповещения при доступе к ключу другого потока;

– получение атрибутов защиты ключа;

– SYNCHRONIZE.

Таким образом, отображаемое право на чтение объекта предоставляет субъекту право осуществлять доступ к объекту по методам, соответствующим (как правило) интуитивному представлению о понятии «чтение». При этом конкретный набор разрешенных субъекту методов доступа определяется типом объекта. Процесс преобразования отображаемого права доступа в набор прав на реализацию методов доступа к объекту называется отображением права доступа. Порядок отображения отображаемых методов доступа для объектов конкретного типа определяется при регистрации данного типа объектов.

Следует иметь в виду, что порядок отображения отображаемого права доступа в набор стандартных и специфичных прав доступа не всегда совпадает с интуитивным смыслом общего права доступа. Например, отображаемое право на все действия над объектом (GENERIC_ALL) не обязательно предоставляет субъекту все права на объект.

Отображаемые права доступа введены в систему разграничения доступа по следующей причине. Отображаемые права доступа позволяют пользователю устанавливать права доступа к объекту, ничего не зная о специфике объектов данного типа. Например, если пользователь желает, чтобы все пользователи могли читать некоторый файл, он просто предоставляет группе пользователей Everyone отображаемое право на чтение файла. При этом пользователь не обязан отдельно предоставлять группе Everyone права на получение различных атрибутов файла, поскольку все эти права

автоматически предоставляются группе Everyone при отображении отображаемого права доступа «чтение объекта». Пользователь может даже не знать, что чтение информации, содержащейся в файле, и чтение атрибутов файла реализуются разными методами доступа.

Последним классом прав доступа, поддерживаемых Windows NT, являются виртуальные права доступа. Виртуальные права доступа не могут быть предоставлены субъекту, но могут быть запрошены им. Поддерживаются два виртуальных права доступа:

- MAXIMUM_ALLOWED;
- ACCESS_SYSTEM_SECURITY.

Запрашивая виртуальное право MAXIMUM_ALLOWED на доступ к объекту, субъект тем самым требует открытия объекта с максимально доступными ему правами. Это виртуальное право позволяет субъекту открыть объект с максимально доступными правами, не производя детального анализа того, какие именно права доступны данному субъекту по отношению к данному объекту. ОС сама проводит такой анализ в процессе проверки прав доступа субъекта к объекту.

Виртуальное право ACCESS_SYSTEM_SECURITY — это право на осуществление одноименного стандартного метода доступа, т.е. право на получение и изменение параметров аудита по данному объекту. Возможность доступа к объектам по этому методу полностью регулируется соответствующей привилегией субъекта доступа. Субъект, обладающий этой привилегией, может обращаться по методу доступа ACCESS_SYSTEM_SECURITY к любому объекту ОС, а субъект, не обладающий этой привилегией, не может применять данный метод доступа ни к одному объекту. Таким образом, субъект, имеющий доступ к параметрам аудита некоторого объекта, имеет доступ к параметрам аудита любого объекта ОС. Разрешить или запретить доступ конкретного субъекта к

конкретному объекту по методу ACCESS_SYSTEM_SECURITY в Windows NT невозможно, и поэтому данное право доступа является виртуальным.

Привилегии субъектов в Windows NT

В Windows NT каждый субъект доступа обладает некоторым (возможно, пустым) набором привилегий. Привилегии представляют собой права на выполнение субъектом действий, касающихся системы в целом, а не отдельных ее объектов. Существуют следующие привилегии:

– привилегия завершать работу ОС и перезагружать компьютер;

– привилегия устанавливать системное время;

– привилегия анализировать производительность одного процесса;

– привилегия анализировать производительность всей ОС в целом;

– привилегия создавать постоянные объекты в оперативной памяти;

– привилегия создавать резервные копии информации, хранящейся на жестких дисках;

– привилегия восстанавливать информацию на жестких дисках с резервных копий;

– привилегия назначать процессам и потокам высокие приоритеты;

– привилегия изменять системные переменные среды;

– привилегия отлаживать программы;

– привилегия загружать и выгружать драйверы и сервисы (позволяет также приостанавливать и возобновлять выполнение сервисов, а также блокировать список сервисов);

– привилегия аудитора позволяет просматривать и очищать журнал аудита, получать информацию о политике аудита, изменять политику аудита и осуществлять доступ к любому объекту операционной системы по методу ACCESS_ SYSTEM_SECURITY;

– привилегия администратора позволяет объявлять себя владельцем любого объекта;

– привилегия добавлять записи в журнал аудита;

– привилегия создавать маркеры доступа;

– привилегия назначать процессам маркеры доступа;

– привилегия выступать как часть ОС;

– привилегия получать оповещения от файловых систем.

При входе в систему пользователь получает привилегии, предоставленные ему индивидуально, а также привилегии, предоставленные группам, в которые входит данный пользователь. Назначать привилегии субъектам доступа может только администратор. Если привилегии пользователя изменились за время его работы с системой, изменения начинают действовать только после того, как пользователь выйдет из системы и снова войдет в нее. Обычно все привилегии пользователя, кроме привилегии получать оповещения от файловых систем, выключены (disabled). Для того чтобы пользователь смог воспользоваться своей привилегией, он должен вначале ее «включить» (enable) с помощью системного вызова AdjustTokenPrivileges. После использования привилегию рекомендуется снова выключить.

Некоторые из перечисленных привилегий позволяют субъектам, обладающим ими, преодолевать те или иные элементы защиты операционной системы. Рассмотрим эти привилегии:

– Привилегия создавать в оперативной памяти постоянные объекты позволяет пользователю отключать механизм автоматического освобождения оперативной памяти при завершении процесса.

– Привилегия создавать резервные копии информации позволяет пользователю игнорировать разграничение доступа при чтении файлов, директорий, ключей и значений реестра.

– Привилегия восстанавливать информацию с резервных копий позволяет пользователю игнорировать разграничение доступа при создании файлов, директорий, ключей и значений реестра, а также при записи в файлы и значения реестра.

– Привилегия назначать процессам высокий приоритет позволяет пользователю «завесить» ОС, создав процесс с высоким приоритетом и введя его в вечный цикл.

– Пользователь, обладающий привилегией изменять системные переменные среды, может, изменив значения переменных path и windir, добиться того, чтобы поиск операционной системой исполняемых модулей для загрузки начинался с личной директории пользователя. Если затем пользователь поместит в эту директорию под именем одной из системных библиотек программную закладку, при первом же обращении ОС к данной библиотеке данная программная закладка будет запущена с полномочиями системного процесса.

– Привилегия отлаживать программы позволяет пользователю обращаться к любому процессу по любому методу доступа. В частности, программа, запущенная таким пользователем, может изменить произвольным образом содержимое адресного пространства любого процесса ОС, что предоставляет такому пользователю практически неограниченные полномочия.

– Привилегия загружать и выгружать драйверы и сервисы позволяет пользователю выполнять произвольный код от имени и

с правами ОС (псевдопользователя SYSTEM). Пользователь может внедрять в операционную систему программные закладки под видом драйверов и сервисов. Учитывая, что драйверы устройств Windows NT могут игнорировать большинство защитных функций ОС, эта привилегия дает обладающему ею субъекту практически неограниченные полномочия.

– Привилегия аудитора позволяет пользователю маскировать свои несанкционированные действия, изменяя политику аудита таким образом, чтобы эти действия не регистрировались.

– Привилегия администратора позволяет пользователю получать доступ к любому объекту по любому методу.

– Привилегия добавлять записи в журнал аудита позволяет пользователю записывать в журнал аудита произвольную информацию, в том числе и информацию, компрометирующую других пользователей.

– Привилегия назначать процессу маркер доступа позволяет пользователю запускать программы от имени любого работающего в системе пользователя. Если эта привилегия предоставлена в совокупности с привилегией создавать маркеры доступа, пользователь может запускать программы от имени любого, в том числе и не существующего реально пользователя.

– Привилегия выступать как часть ОС позволяет пользователю выполнять ряд потенциально опасных действий, в том числе посылать произвольные сообщения окнам программ, запущенных другими пользователями, а также брать на себя часть функций ОС, связанных с идентификацией, аутентификацией и авторизацией пользователей.

Администраторы ОС должны подходить к назначению субъектам доступа привилегий с максимальной ответственностью. Особое внимание следует уделять вышеперечисленным опасным привилегиям.

§ 2. МАРКЕР ДОСТУПА ПОЛЬЗОВАТЕЛЯ И ДЕСКРИПТОР ЗАЩИТЫ ОБЪЕКТА

Маркер доступа пользователя

В Windows NT каждый пользователь (в том числе и каждый псевдопользователь), работающий в системе, имеет свой маркер доступа (accesstoken) — объект специального вида, содержащий следующую информацию:

– идентификатор пользователя;

– идентификаторы групп и специальных групп, в которые входит пользователь;

– привилегии пользователя;

– идентификатор сеанса работы пользователя, к которому относится маркер доступа;

– атрибуты защиты, которые назначаются по умолчанию новым объектам ОС, созданным данным пользователем в текущем сеансе работы;

– имя и идентификатор подсистемы, выдавшей маркер доступа (Advapi, если пользователь вошел в систему локально, NetLogon, если пользователь вошел в систему по сети через SMB-сервер, и т.д.);

– некоторую служебную информацию.

Маркер доступа содержит всю информацию о пользователе, необходимую системе разграничения доступа для принятия решений о предоставлении пользователю доступа к тем или иным объектам ОС.

Каждому процессу Windows NT назначается так называемый первичный маркер доступа (primaryaccesstoken). Обычно первичный маркер доступа — это маркер доступа пользователя, запу-

стившего данный процесс. Субъект, обладающий соответствующей привилегией, может назначить процессу другой первичный маркер доступа. Отдельным потокам процесса могут назначаться свои маркеры доступа — маркеры олицетворения (impersonationaccesstokens). Механизм олицетворения обычно используется процессами-серверами. Когда процесс-сервер обслуживает запрос процесса-клиента, для выполнения запроса внутри процесса сервера создается поток, которому назначается маркер доступа пользователя, инициировавшего запрос. В дальнейшем этот поток работает с правами того пользователя, от имени которого выполняется процесс-клиент. Назначать маркер доступа процессу, а также создавать маркеры доступа могут только субъекты, обладающие соответствующими привилегиями.

Маркер доступа создается подсистемой аутентификации ОС в процессе авторизации пользователя, при этом псевдопользователь SYSTEM использует свою привилегию создавать маркеры доступа. После того как маркер доступа создан, информация о группах, в которые входит пользователь, и о привилегиях пользователя не может быть ни добавлена в маркер доступа, ни удалена из него. Однако группы и привилегии пользователя могут быть временно заблокированы (выключены). Если некоторая группа или привилегия заблокирована, подсистема разграничения доступа ОС при проверке прав доступа пользователя к объектам игнорирует наличие этой группы или привилегии в маркере доступа до тех пор, пока она не будет разблокирована.

Дескриптор защиты

Атрибуты защиты объекта Windows NT описываются специальной структурой данных, называемой дескриптором защиты (securitydescriptor) и содержащей следующую информацию:

– идентификатор владельца объекта;

– идентификатор первичной группы владельца объекта;

– список избирательного контроля доступа (discretionary access control list, DACL) — список, полностью описывающий права различных субъектов на объект;

– системный список контроля доступа (system access control list, SACL) — используется при генерации сообщений аудита.

Если объект не имеет дескриптора защиты, при обращениях субъектов к нему права доступа не проверяются. В этом случае любой субъект имеет абсолютные права на данный объект.

Если объект хранится на диске или ином внешнем устройстве, дескриптор защиты хранится вместе с ним. При этом формат хранения объекта должен предоставлять такую возможность.

Файловые системы FAT и HPFS не поддерживают хранение на диске дескрипторов защиты для файлов. Только файлы и директории, расположенные на логических дисках с файловой системой NTFS, могут иметь дескрипторы защиты. Ключи реестра могут иметь дескрипторы защиты независимо от файловой системы диска, на котором располагается реестр.

Элементами списка избирательного контроля доступа являются записи вида:

Разрешить/ запретить	Идентификатор субъекта	Права доступа	Флаги и атрибуты

Элементы списка избирательного контроля доступа называются элементами контроля доступа (accesscontrolentries, ACE). Каждый элемент контроля доступа разрешает или запрещает некоторому субъекту определенные права доступа к объекту.

Если список избирательного контроля доступа отсутствует в дескрипторе защиты, всем субъектам предоставляются все права

доступа к объекту. Если же список избирательного контроля доступа имеется, но пуст, доступ к объекту запрещен всем субъектам. Приведем пример списка избирательного контроля доступа некоторого исполняемого файла:

разрешить специальной группе INTERACTIVE — чтение;
разрешить группе Otdel5 — выполнение;
запретить пользователю Sidoroff — запись;
разрешить группе Programmers — запись;

Для данного файла определены следующие права доступа:

– чтение файла разрешено пользователям, работающим с системой интерактивно, и запрещено остальным пользователям;

– выполнение файла разрешено пользователям, входящим в группу Otdel5, и запрещено остальным пользователям;

– запись в файл разрешена пользователям, входящим в группу Programmers, и запрещена остальным пользователям, а также пользователю Sidoroff. При этом пользователю Sidoroff запись в файл запрещена даже в том случае, если он входит в группу Programmers, которой запись в файл разрешена.

Владелец объекта всегда имеет право изменять дескриптор защиты объекта, даже если это явно запрещено ему списком избирательного контроля доступа. Пользователь, обладающий привилегией объявлять себя владельцем любого объекта (привилегия администратора), может объявлять себя владельцем даже тех объектов, для которых это явно запрещено ему списком избирательного контроля доступа. В остальном список избирательного контроля доступа полностью описывает права различных субъектов на доступ к данному объекту.

Отображение отображаемых прав доступа происходит до их проверки. Таким образом, учитывая, что ACE, расположенные

ближе к началу DACL, имеют более высокий приоритет, в следующем примере ни один пользователь не может обращаться к файлу ни по одному методу доступа.

Следует отметить, что стандартные средства работы с файлами Explorer не поддерживают механизм разграничения доступа в полном объеме. Для того чтобы список избирательного контроля доступа файла можно было просматривать и редактировать, информация, содержащаяся в этом списке, должна удовлетворять следующим требованиям:

– все элементы списка, запрещающие доступ субъектов к объекту, должны находиться в начале списка;

– все элементы списка, запрещающие доступ субъектов к объекту, должны запрещать им все права доступа к объекту, а не только некоторые.

Порядок проверки прав доступа субъекта к объекту в Windows NT

Когда субъект Windows NT открывает объект, субъект должен сообщить ОС права доступа, которые необходимы ему для работы с данным объектом. Эти права доступа кодируются с помощью битовой маски, каждый бит которой соответствует некоторому праву доступа. Например, если пользователь открывает объект типа «файл», второй параметр системной функции CreateFile является битовой маской, описывающей запрашиваемые права доступа к объекту. Так, если субъект открывает файл для чтения и записи, эта маска доступа должна быть равна FILE_READ_DATA | FILE_WRITE_DATA | FILE_APPEND_DATA (или GENERIC_READ | GENERIC_WRITE).

В процессе открытия объекта операционная система получает маркер доступа субъекта и дескриптор защиты объекта и вызывает функцию ядра Windows NT SeAccessCheck. Эта функция

реализует проверку прав доступа субъекта к объекту по алгоритму, который будет описан ниже.

Перед тем как перейти к описанию алгоритма, введем следующие обозначения:

- — операция побитового отрицания;

& — операция побитовой конъюнкции (побитовое логическое И);

| — операция побитовой дизъюнкции (побитовое логическое ИЛИ);

m — маска доступа, описывающая права, запрошенные субъектом и пока не предоставленные ему; в начале алгоритма m содержит все права доступа, запрошенные субъектом;

a(i) — маска доступа i-го ACE;

n — количество ACE в дескрипторе защиты объекта;

x_i — i-й бит маски доступа x.

Алгоритм проверки прав доступа субъекта к объекту в Windows NT выглядит следующим образом:

1. В масках доступа m, a(1), ..., a(n) отображаются все отображаемые права доступа. Таким образом, в дальнейшем все маски доступа, используемые алгоритмом, включают только специфичные и стандартные права доступа (за исключением маски доступа m, которая может содержать виртуальные права доступа).

2. Создаются маски доступа g=0 и d=0. В дальнейшем маска доступа g содержит права доступа к объекту, разрешенные субъекту, а маска доступа d — права доступа, запрещенные субъекту.

3. Если субъект является владельцем объекта и m_{WRITE_DAC}=1, то выполняются присваивания g_{WRITE_DAC}=1. m_{WRITE_DAC}=0. Другими словами, владелец объекта всегда имеет право модифицировать атрибуты защиты объекта.

4. Если субъект обладает привилегией администратора, эта привилегия не заблокирована и m_{WRITE_OWNER}=1, то выполняются

присваивания $g_{WRITE\_OWNER}=1$, $m_{WRITE\_OWNER}=0$. Другими словами, администратор может объявить себя владельцем любого объекта.

5. Если субъект обладает привилегией аудитора, она не заблокирована и $m_{ACCESS\_SYSTEM\_SECURITY}=1$, то выполняются присваивания $g_{ACCESS\_SYSTEM\_SECURITY}=1$, $m_{ACCESS\_SYSTEM\_SECURITY}=0$. Это значит, что аудитор может обращаться по методу доступа ACCESS_SYSTEM_SECURITY к любому объекту.

6. Если $m_{MAXIMUM\_ALLOWED}=l$, происходит присваивание $m_{MAXIMUM\_ALLOWED}=0$, затем выполняется цикл по i от 1 до n.

На каждой итерации цикла выполняются следующие действия:

– если i-й ACE является разрешающим и i-й ACE относится к субъекту, обращающемуся к объекту, то выполняется присваивание g=g|(a(i)&-d). Другими словами, права доступа из a(i), которые не содержатся также и в d, добавляются в g;

– если i-й ACE является запрещающим и i-й ACE относится к субъекту, обращающемуся к объекту, то выполняется присваивание d=d|(a(i)&-g). Другими словами, права доступа из a(i), которые не содержатся также и в g, добавляются в d.

По окончании цикла маска доступа g содержит все предоставленные субъекту права на данный объект, а маска доступа d — все права доступа к данному объекту, явно запрещенные субъекту.

По окончании цикла, как нетрудно убедиться, маски доступа содержат следующую информацию:

– m — права доступа, запрошенные субъектом и не предоставленные ему;

– g — права доступа, предоставленные субъекту;

– d — права доступа, запрошенные субъектом и явно запрещенные ему.

7. Если m=0, субъект получает доступ к объекту. При этом маска g описывает права доступа, предоставленные субъекту.

Если m≠0, субъект получает отказ в доступе к объекту. При этом маска g описывает максимальное подмножество запрошенных прав доступа к объекту, которые могут быть предоставлены субъекту.

Примеры проверки прав доступа при обращении субъекта к объекту

В приведенных далее примерах для сокращения записи расчетов будем использовать следующие обозначения:

R — право на чтение файла (FILE_READ_DATA);

W — право на запись в файл (FILE_WTITE_DATA);

X — право на выполнение файла (FILE_EXECUTE);

P — право на изменение атрибутов защиты (WRITE_DAC);

O — право объявлять себя владельцем объекта (WRITE_OWNER);

M — виртуальное право доступа MAXIMUM_ALLOWED;

A — виртуальное право доступа ACCESS_SYSTEM_SECURITY.

Пусть пользователь Ivanoff входит в группы Users, Programmers, Otdel5 и NETWORK и не имеет никаких привилегий.

Пусть пользователь Petroff входит в группы Users, Otdell и NETWORK и обладает привилегией аудитора.

Пусть пользователь Sidoroff входит в группы Users, Programmers, Otdel5 и NETWORK и не имеет никаких привилегий.

Пусть пользователь Kuznetsoff входит в группы Users, Administrators, Programmers, Otdel5 и в специальную группу NETWORK и обладает привилегией администратора.

Дескриптор защиты файла имеет следующий вид:

Владелец: Ivanoff.

Первичная группа владельца: Users.

DACL:

разрешить Administrators — R;
разрешить Otdel5 — RX;
запретить Programmers — W;
разрешить Sidoroff — W;
разрешить Petroff — O.

Пример 1. Пользователь Sidoroff открывает файл для чтения.

1. m=R.
2. g=d=0.
3. Sidoroff не является владельцем объекта. Ничего не происходит.
4. Sidoroff не обладает привилегией администратора. Ничего не происходит.
5. Sidoroff не обладает привилегией аудитора. Ничего не происходит.
6. $m_{MAXIMUM\_ALLOVED}$= 0. Ничего не происходит.
7. m=0. Пользователь получил доступ.

Пример 2. Пользователь Sidoroff открывает файл для чтения и записи.

1. m=RW.
2. g=d=0.
3. Sidoroff не является владельцем объекта. Ничего не происходит.
4. Sidoroff не обладает привилегией администратора. Ничего не происходит.
5. Sidoroff не обладает привилегией аудитора. Ничего не происходит.
6. $m_{MAXIMUM\_ALLOVED}$=0. Ничего не происходит.
7. m≠0. Пользователь получил отказ в доступе. Максимальные разрешенные права доступа из запрошенных равны g=R.

Пример 3. Пользователь Sidoroff открывает файл с максимально доступными ему правами, при этом для успешного открытия файла ему обязательно должно быть предоставлено право записи.

1. m=MW.
2. g=d=0.
3. Sidoroff не является владельцем объекта. Ничего не происходит.
4. Sidoroff не обладает привилегией администратора. Ничего не происходит.
5. Sidoroff не обладает привилегией аудитора. Ничего не происходит.
6. $m_{MAXIMUM\_ALLOVED}$=1. Выполняется присваивание m=W. Начинается цикл по i.

i=1. Sidoroff не Administrators. Ничего не происходит.

i=2. Sidoroff в Otdel5. g=g|(a(2)&-d)=Q|(RX&-0)=RX, d=0.

i=3. Sidoroff в Programmers. g=RX, d=d|(a(3)&-g)=0|(W&-R)=W.

i=4. Sidorof=Sidoroff. g=g|(a(4)&-d)=RX|(W&-W)=RX, d=W.

i=5. Sidoroff не Petroff. Ничего не происходит. В конце цикла m=W, g=RX, d=W.

7. m≠0. Пользователь получил отказ в доступе. Максимальные разрешенные права из запрошенных: g=RX.

Назначение атрибутов защиты создаваемым объектам в Windows NT

При создании в Windows NT нового объекта ему назначаются атрибуты защиты согласно следующим правилам.

Если процесс, создающий объект, явно указывает корректный дескриптор защиты для создаваемого объекта, такому объекту назначается этот дескриптор защиты.

Если процесс, создающий объект, указывает, что атрибуты защиты должны быть установлены по умолчанию, или если указанный дескриптор защиты некорректен, дескриптор защиты объекта создается с помощью описанного ниже механизма наследования.

Если по каким-то причинам наследование дескриптора защиты невозможно, объекту присваивается дескриптор защиты, хранящийся в маркере доступа субъекта, создающего объект.

Процедура наследования атрибутов защиты в Windows NT заключается в том, что дескриптор защиты создаваемого объекта формируется на основе дескриптора защиты контейнера, в котором создается объект.

Владельцем объекта становится субъект, определенный в маркере доступа пользователя, создающего объект, как владелец создаваемых объектов по умолчанию. В стандартной конфигурации Windows NT для пользователей и псевдопользователей, входящих в группу Administrators, таким субъектом является эта группа, для остальных субъектов — личный идентификатор пользователя. Первичная группа владельца объекта также берется из маркера доступа пользователя, создающего объект. По умолчанию для администраторов в качестве первичной группы владельца объекта берется группа Administrators, для остальных пользователей — первичная группа этого пользователя.

DACL создаваемого дескриптора защиты формируется из ACE, входящих в DACL объекта-родителя. ACE объекта-родителя, которые будут включены в DACL создаваемого объекта, определяются следующими флагами ACE:

– CONTAINER_INHERIT_ACE (c): если этот флаг установлен и создаваемый объект является контейнером, данный ACE должен включаться в DACL создаваемого объекта;

– OBJECT_INHERIT_ACE (o): если этот флаг установлен и создаваемый объект не является контейнером, данный ACE должен включаться в DACL создаваемого объекта;

– NO_PROPAGATE_INHERIT_ACE (n): если этот флаг установлен, при наследовании ACE флаги c и o сбрасываются; другими словами, при наличии этого флага ACE наследуется только один раз;

– INHERIT_ONLY_ACE (i): если этот флаг установлен, данный ACE игнорируется при проверке прав доступа к объекту и используется только при наследовании.

Если создаваемый объект не является контейнером, в DACL создаваемого объекта включаются те и только те ACE объекта-родителя, в которых установлен флаг o. Все флаги унаследованных ACE сбрасываются. Если маска доступа ACE объекта-родителя содержит отображаемые права доступа, перед наследованием ACE производится их отображение. Если же создаваемый объект является контейнером, наследуются следующие ACE:

– ACE, в которых установлен флаг c, при этом, если в ACE установлен флаг n, после наследования флаги c и o сбрасываются;

– ACE, в которых не установлены ни флаг c, ни флаг n, но установлен флаг o, при этом в унаследованном ACE устанавливается флаг i.

Как правило, DACL контейнера содержит две группы ACE:

– ACE для разграничения доступа к объекту установлен флаг c;

– ACE для назначения объектам, создаваемым внутри контейнера, установлены флаги o и i.

Защита от несанкционированных действий администратора

В отличие от UNIX в Windows NT отсутствует суперпользователь. Все пользователи и псевдопользователи, включая администраторов и ОС, обладают ограниченными полномочиями. В Windows NT ни один субъект доступа не имеет возможности игнорировать разграничение доступа к объектам. Поэтому, если, например, запретить псевдопользователю SYSTEM обращаться к системной директории Windows NT, ОС не сможет загрузиться.

Однако субъект, обладающий привилегией администратора, может получить доступ к любому объекту ОС по любому методу доступа, за исключением метода ACCESS_SYSTEM_SECURITY, для которого требуется привилегия аудитора. Для этого субъект должен выполнить следующие действия:

1. Используя привилегию администратора, объявить себя владельцем объекта.

2. Используя полномочия владельца, предоставить себе необходимые права доступа к объекту.

3. Обратиться к объекту, используя только что полученные права.

Вышеописанные действия необратимы. Дело в том, что администратор, вступив во владение объектом, не может отдать его обратно. Как уже говорилось выше, в Windows NT при изменении владельца объекта новым владельцем может быть назначен только субъект, выполняющий эту операцию. Кроме того, полномочия владельца объекта не дают право получить атрибуты защиты объекта. Если это право не предоставлено владельцу объекта явно, владелец может изменить атрибуты защиты объекта, но он не может узнать, какие атрибуты защиты объект имел до этого изменения.

Таким образом, в Windows NT администратор может обратиться к любому объекту ОС по любому методу доступа, но не может сделать это незаметно для других пользователей.

§ 3. ИДЕНТИФИКАЦИЯ, АУТЕНТИФИКАЦИЯ И АВТОРИЗАЦИЯ ПОЛЬЗОВАТЕЛЕЙ В WINDOWS NT

Архитектура подсистемы аутентификации Windows NT

1. Общие сведения. Задачи идентификации, аутентификации и авторизации пользователей в Windows NT решаются специальной подсистемой, обычно называемой в документации подсистемой защиты (security subsystem). Поскольку это название представляется не вполне корректным (в эту подсистему не входит, например, монитор ссылок, являющийся неотъемлемой частью подсистемы защиты), будем использовать термин «подсистема аутентификации».

Подсистема аутентификации состоит из нескольких программных модулей, связанных между собой. Эти модули распределены на три уровня. Средний уровень подсистемы аутентификации пользуется услугами нижнего уровня и предоставляет услуги верхнему.

Рассмотрим функции каждого из этих уровней. Если не оговорено противное, речь будет идти только о локальной аутентификации, когда пользователь входит в систему с локальной консоли.

На верхнем уровне находятся так называемые библиотеки-провайдеры или просто провайдеры, заменяемые библиотеки, реализующие большую часть высокоуровневых функций процесса аутентификации.

WinLogon представляет собой обычный процесс Win32 API, выполняющийся от имени псевдопользователя SYSTEM. WinLogon автоматически запускается при старте ОС и остается активным до выключения питания или перезагрузки. При аварийном завершении этого процесса происходит аварийное завершение работы всей ОС («синий экран»). Таким образом, подменить WinLogon в процессе функционирования ОС практически невозможно.

При входе пользователя в систему с локального или удаленного терминала провайдер, обслуживающий данный терминал, получает от пользователя его имя и пароль. При входе пользователя в систему с локальной консоли в качестве провайдера по умолчанию выступает библиотека msgina.dll, которая осуществляет все взаимодействия между локальным пользователем и процессом аутентификации. Если в системе установлено программное обеспечение, поддерживающее вход пользователя с удаленного терминала (например, сервер Telnet), это программное обеспечение должно зарегистрировать своего провайдера, который будет получать из сети имя и пароль пользователя и передавать их на нижние уровни подсистемы аутентификации.

Вход пользователя в систему в Windows NT производится следующим образом:

1. Провайдер получает от пользователя идентифицирующую и аутентифицирующую информацию. В стандартной конфигурации ОС в качестве идентифицирующей информации используется имя, а в качестве аутентифицирующей — пароль. Однако при использовании нестандартных провайдеров можно реализовать схему аутентификации, предусматривающую применение внешних носителей ключевой информации или биометрических характеристик пользователя. Здесь рассматривается только стандартная конфигурация Windows NT.

2. Провайдер осуществляет аутентификацию, передавая имя и пароль на средний уровень подсистемы аутентификации с помощью системного вызова LogonUser. При этом, если аутентификация прошла успешно, создается маркер доступа пользователя.

3. Если маркер доступа пользователя создан успешно, провайдер осуществляет авторизацию пользователя, запуская процесс Userlnit.exe от имени аутентифицированного пользователя. Для этого используется системный вызов CreateProcessAsllser, который отличается от вызова CreateProcess только тем, что запускаемому процессу назначается маркер доступа, отличный от маркера доступа процесса-родителя. В данном случае процессу Userlnit назначается только что созданный маркер доступа авторизуемого пользователя.

4. Процесс Userlnit загружает индивидуальные настройки пользователя из его профиля (profile), монтирует ключ реестра, соответствующий данному пользователю, и загружает программную среду пользователя (по умолчанию в Windows NT 4.0 Explorer, в более ранних версиях — Program Manager). После этого Userlnit завершает работу.

При выполнении второго этапа данной процедуры WinLogon использует привилегии псевдопользователя SYSTEM создавать маркеры доступа и выступать от имени ОС, а при выполнении третьего этапа — привилегию назначать процессам маркеры доступа. Таким образом, если эти привилегии не будут предоставлены псевдопользователю SYSTEM, вход пользователей в систему станет невозможен.

В средний уровень подсистемы аутентификации Windows NT входят локальный распорядитель безопасности (local security authority, LSA) и так называемые пакеты аутентификации — заменяемые библиотеки, реализующие большую часть низкоуровневых функций аутентификации.

Так же как и WinLogon, LSA представляет собой обычный процесс (по имени lsass.exe), выполняющийся от имени псевдопользователя SYSTEM. Аварийное завершение LSA приводит к аварийному завершению работы всей ОС. Как и WinLogon, LSA передоверяет большинство своих функций заменяемым библиотекам. Стандартная схема аутентификации реализуется пакетом MSV 1.0 (msv1_0.dll), могут быть установлены и другие пакеты аутентификации. Пакет аутентификации осуществляет аутентификацию пользователя в процессе обработки системного вызова LogonUser. Аутентификация производится следующим образом:

1. Пакет аутентификации получает от верхнего уровня подсистемы аутентификации имя и пароль пользователя и генерирует образ пароля.

2. Используя услуги нижнего уровня подсистемы аутентификации, пакет аутентификации получает эталонный образ пароля и сравнивает его с образом пароля из п. 1.

3. При совпадении образов паролей LSA получает от нижнего уровня подсистемы аутентификации информацию о том, может ли данный пользователь начинать в данный момент работу с данной рабочей станцией (не устарел ли пароль, не заблокирован ли бюджет пользователя и т.д.).

4. При положительном результате проверки LSA формирует маркер доступа пользователя, получая необходимую информацию от нижнего уровня подсистемы аутентификации.

5. LSA передает сформированный маркер доступа верхнему уровню подсистемы аутентификации.

Для генерации образа пароля стандартный пакет аутентификации MSV 1.0 применяет хеш-функцию MD4. Для совместимости с более ранними версиями Windows MSV 1.0 поддерживает другой формат образа пароля. В этом случае пароль преобразуется в формат ANSI, все латинские буквы приводятся к верхнему регистру,

пароль разбивается на две строки по 7 байт (если пароль короче, он дополняется нулями). Затем каждая половина пароля используется в качестве ключа при шифровании «магической» строки «KGS!@#$%» по алгоритму DES. Полученные два шифртекста по 8 байт каждый и представляют собой образ пароля.

Недостатком обоих алгоритмов генерации образа пароля является то, что ни один из них не использует маркант, и поэтому одинаковым паролям различных пользователей соответствуют одинаковые образы паролей.

Нижний уровень подсистемы аутентификации Windows NT отвечает за хранение в системе учетной информации о пользователях, в том числе и эталонных образов паролей. При аутентификации пользователя нижний уровень подсистемы аутентификации передает среднему уровню эталонный образ пароля пользователя, а при авторизации список групп и привилегий пользователя.

При стандартной конфигурации ОС нижний уровень подсистемы аутентификации включает в себя систему управления списком пользователей (Security Account Manager, SAM) и сервис NetLogon. SAM используется для извлечения информации из реестра локального компьютера, а NetLogon — информации из реестра контроллера домена. Администраторы системы могут устанавливать и другие сервисы аналогичного назначения.

Учетная информация о пользователях хранится в ключах реестра \Registry\Machine\SAM и \Registry\Machine\SECURITY. Эталонные образы паролей хранятся зашифрованными на идентификаторе пользователя по алгоритму DES.

Параметры аутентификации в Windows NT

Подсистема аутентификации Windows NT обладает достаточно большой гибкостью и позволяет администраторам ОС

настраивать различные параметры аутентификации как для отдельных пользователей системы, так и для всех пользователей в совокупности.

Администраторы Windows NT могут вводить следующие ограничения на пароли пользователей:

– минимальный и максимальный срок действия пароля;

– минимальную допустимую длину пароля;

– минимальное допустимое количество смен пароля до первого повторения;

– максимально допустимое количество неудачных попыток входа в систему;

– срок, по истечении которого счетчик неудачных попыток входа в систему обнуляется;

– срок, на который пользователю запрещается вход в систему в случае превышения максимально допустимого количества неудачных попыток входа в систему (может быть неограниченным, в этом случае запрет на вход пользователя в систему снимается только администратором).

Администратор определяет, могут ли пользователи самостоятельно менять пароль в случае истечения максимального срока его действия или они должны уведомлять его об этом.

Механизм автоматической блокировки (lockout) пользователя при превышении максимально допустимого количества неудачных попыток входа в систему не распространяется на пользователя Administrator.

Для каждого конкретного пользователя могут быть установлены следующие флаги:

– пользователь обязан сменить пароль при ближайшем входе в систему — обычно применяется для только что зарегистрированных пользователей;

– пользователь не может менять свой пароль — обычно применяется для «групповых» пользователей (Guest, Anonymous и т.д.);

– на пользователя не распространяется ограничение максимального срока действия пароля — обычно применяется в совокупности с предыдущим требованием;

– пользователь не может работать в системе — применяется для временного блокирования учетной записи пользователя (например, на период отпуска или болезни пользователя).

Кроме того, могут быть введены следующие требования к процедуре авторизации пользователя:

– может быть явно указан путь к профилю (profile) пользователя. В этом случае индивидуальные настройки пользователя будут загружаться не из системной директории локального компьютера, а из той директории того компьютера, которая указана в пути к профилю. В результате индивидуальные настройки пользователя могут быть сделаны одинаковыми для нескольких компьютеров;

– пользователю может быть назначен скрипт (программа или командный файл), который будет автоматически выполняться при каждом входе пользователя в систему, неважно, локальном или удаленном;

– пользователю может быть назначена домашняя директория, которая становится текущей директорией по умолчанию для всех программ пользователя. Домашняя директория может располагаться на любом компьютере сети, установка соединения при необходимости выполняется автоматически.

Для пользователей домена Windows NT могут быть введены следующие дополнительные требования к процедурам идентификации, аутентификации и авторизации:

– время работы пользователя с операционной системой может быть ограничено; при этом вход пользователя в систему разрешается только в отведенные для этого часы;

– количество компьютеров, с которых пользователь может входить в домен, может быть ограничено; администратор может указать до восьми компьютеров, с которых разрешен вход пользователя в домен;

– может быть установлена автоматическая блокировка учетной записи пользователя по истечении определенного времени;

– пользователю может быть запрещен интерактивный вход на любой компьютер домена; в этом случае пользователь может работать с доменом только извне.

Помимо вышеперечисленных требований и ограничений при идентификации и аутентификации пользователя также осуществляется проверка одной из следующих трех «привилегий»:

– входить в систему интерактивно (с локального или удаленного терминала);

– входить в систему через SMB-сервер;

– запускать сервис от своего имени.

«Привилегия», которая должна проверяться, определяется провайдером при вызове функции LogonUser. Например, если четвертый параметр этой функции равен LOGON32_ LOGON_SERVICE, это означает, что пользователь входит в систему в качестве сервиса, т.е. запускает сервис от своего имени, и должна быть проверена «привилегия» запускать сервисы от своего имени.

В следующих версиях Windows NT ожидается появление еще одной «привилегии» запускать от своего имени пакетное задание (batchjob).

Несмотря на впечатляющий список настраиваемых параметров процедур идентификации, аутентификации и авторизации, эти процедуры в Windows NT имеют ряд недостатков, в том числе:

– не поддерживается возможность принудительного расширения алфавита паролей (например, требование обязательно включать в каждый пароль как строчные, так и заглавные буквы);

– отсутствует проверка на совпадение имени и пароля пользователя;

– отсутствует поддержка списка «плохих» паролей;

– при превышении пользователем максимально допустимого количества неудачных попыток входа в систему блокируется не терминал, с которого осуществлялись эти попытки, а учетная запись пользователя; таким образом, злоумышленник может заблокировать любого пользователя ОС, кроме пользователя Administrator, многократно пытаясь войти в систему от его имени с неверным паролем.

Альтернативные схемы идентификации и аутентификации в Windows NT

Выше была изложена стандартная схема идентификации и аутентификации пользователя в Windows NT, которая применяется при использовании стандартных провайдеров и пакетов аутентификации. Однако, поскольку и провайдеры, и пакеты аутентификации являются заменяемыми компонентами подсистемы аутентификации, администратор ОС может, установив нестандартный провайдер и/или пакет аутентификации, реализовать в Windows NT любую другую схему аутентификации. Для этого необходимо всего лишь разместить в системной директории Windows NT необходимые библиотеки и внести изменения в соответствующие ключи реестра.

При использовании нестандартных провайдеров и пакетов аутентификации в качестве аутентифицирующей информации может использоваться произвольная строка Unicode длиной до 128 символов. Эта строка не обязательно должна вводиться с клавиатуры, она может быть получена с любого внешнего устройства, к которому имеет доступ провайдер. На рынке программных продуктов встречаются расширения подсистемы аутентификации Windows NT, поддерживающие ввод аутентифицирующей информации с электронных ключей Touch Memory. Microsoft планирует реализовать в одной из следующих версий Windows NT встроенную поддержку интеллектуальных карт в качестве внешних носителей ключевой информации.

§ 4. АУДИТ В WINDOWS NT

Журнал аудита

В Windows NT журнал аудита представляет собой файл с именем SecEvent.evt, расположенный в поддиректории system32\config системной директории. Формат этого файла недокументирован. Информация хранится в журнале аудита в открытом виде, защита журнала аудита организуется исключительно средствами подсистемы разграничения доступа. Поэтому администраторы Windows NT обязательно должны убедиться, что никто, кроме аудиторов, не имеет доступа к файлу журнала аудита.

Для просмотра журнала аудита используется стандартная утилита EventViewer, которую можно применять и для просмотра других системных журналов. Эта утилита разрешает читать журнал аудита только членам группы Administrators, а также пользователям, обладающим привилегией аудитора. Эти ограничения доступа действуют и в том случае, когда системный раздел жесткого

диска отформатирован под FAT или HTFS. Все пользователи, которые могут читать журнал аудита, могут и очищать его. Факт очистки журнала регистрируется сразу после очистки.

Пользователи, не имеющие возможности читать журнал аудита с помощью утилиты EventViewer, но обладающие правом чтения файла SecEvent.evt, могут читать этот файл с помощью других программных средств. Поэтому права доступа субъектов к этому файлу должны быть ограничены, например, так:

разрешить — Auditors — все права доступа;

разрешить — SYSTEM — все права доступа.

Здесь Auditors — это группа аудиторов, являющаяся подмножеством группы администраторов. Сделать группы аудиторов и администраторов непересекающимися в Windows NT практически невозможно.

Размер журнала аудита по умолчанию ограничен значением 512К, однако администратор ОС может установить любое другое значение, кратное 64К. Администратор может также определить поведение ОС при переполнении журнала аудита. Возможны три варианта реакции на эту ситуацию:

1. Старые события стираются по мере необходимости (по умолчанию).

2. Если самое старое событие в журнале аудита зафиксировано более N дней назад (число N выбирает администратор), одно или несколько самых старых событий стирается, в противном случае новые события не регистрируются до тех пор, пока не пройдет N дней с момента регистрации самого старого события.

3. Новые события не регистрируются до тех пор, пока журнал не будет очищен.

Если значение CrashOnAuditFail ключа реестра \Registry\Machine\SYSTEM\CurrentControlSet\Control\Lsa равно

единице, при переполнении журнала аудита это значение становится равным двум и происходит крах ОС («синий экран»). При следующей загрузке ОС в систему может войти только администратор. Он должен очистить журнал аудита, вернуть данное значение реестра в исходное состояние и перезагрузить компьютер. До тех пор, пока все эти действия не будут выполнены, подсистема аудита не будет регистрировать события.

Для добавления записей в журнал аудита используются специальные системные вызовы программного интерфейса, пять из которых (ObjectOpenAuditAlarm, ObjectCloseAuditAlarm, ObjectPrivilegeAuditAlarm, PrivilegedServiceAuditAlarm и AccessCheckAndAuditAlarm) документированы.

Добавлять записи в журнал аудита может лишь субъект доступа, обладающий соответствующей привилегией. По умолчанию эта привилегия предоставляется только псевдопользователю SYSTEM, эту установку не следует изменять ни в коем случае. Если эта привилегия предоставляется какому-то физическому пользователю, этот пользователь тем самым получает возможность записывать в журнал аудита произвольную информацию, в том числе и информацию, компрометирующую других пользователей. Обычно новые записи в журнал аудита добавляют ядро, подсистема Win32 и подсистема аутентификации Windows NT.

Политика аудита

Множество событий, информация о которых записывается в журнал аудита, зависит от политики аудита, которую определяют пользователи/аудиторы. Windows NT позволяет регистрировать в журнале аудита события следующих категорий:

- вход/выход пользователя из системы;
- доступ субъектов к объектам;
- использование субъектами доступа опасных привилегий;

– изменения в списке пользователей;
– изменения в политике безопасности;
– системные события;
– запуск и завершение процессов.

Для каждого класса событий могут регистрироваться либо только успешные события (соответствующая операция выполнена успешно), либо только неуспешные (при выполнении операции произошла ошибка), либо и те, и другие, либо никакие.

В Windows NT считаются опасными следующие привилегии субъектов доступа:

– получать оповещения от файловой системы;
– добавлять записи в журнал аудита;
– создавать маркеры доступа;
– назначать маркеры доступа процессам;
– создавать резервные копии информации, хранящейся на жестких дисках;
– восстанавливать информацию на жестких дисках с резервных копий;
– отлаживать программы.

Далеко не все объективно опасные привилегии субъектов считаются опасными с точки зрения подсистемы защиты Windows NT. Например, не считается опасной привилегия загружать и выгружать драйверы и сервисы.

С другой стороны, в журнале аудита Windows NT регистрируется использование некоторых других привилегий, которые, согласно документации, не считаются опасными.

Порядок регистрации событий при доступе субъектов к объектам определяется не только политикой аудита, но и атрибутами защиты объекта. Как уже говорилось, в состав дескриптора защиты может входить системный список контроля доступа (SACL), определяющий порядок регистрации событий аудита при доступе

субъектов к данному объекту. Так же как и DACL, SACL представляет собой список переменной длины, элементами которого являются ACE, имеющие следующий формат:

| регистрировать | идентификатор субъекта права доступа | флаги и атрибуты |

В отличие от ACE из DACL ACE в SACL всегда имеют тип «регистрирующий ACE» (systemaudit ACE). Разработчики Windows NT зарезервировали еще один тип ACE — «тревожный» ACE (systemalarm ACE), предназначенный для интерактивного оповещения администраторов операционной системы, однако этот механизм до сих пор не реализован.

ACE, входящий в SACL, имеет все флаги, которые имеет ACE, входящий в DACL. Кроме того, ACE из SACL имеют еще два флага:

– SUCCESSFUL_ACCESS_ACE_FLAG (s) — если этот флаг установлен, будут регистрироваться в журнале аудита все успешные обращения к объекту субъекта, идентификатор которого записан в ACE, по любому из методов доступа, перечисленных в маске доступа ACE;

– FAILED_ACCESS_ACE_FLAG (f) — если этот флаг установлен, будут регистрироваться в журнале аудита все неуспешные обращения к объекту субъекта, идентификатор которого записан в ACE, по любому из методов доступа, перечисленных в маске доступа ACE.

Если в ACE установлены оба флага, регистрируются любые обращения субъекта к объекту по перечисленным методам доступа, как успешные, так и неуспешные. Если в ACE установлен флаг i, при доступе субъектов к объекту ACE игнорируется.

Поскольку все ACE в SACL однотипны, порядок их взаимного расположения не имеет значения.

Если в дескрипторе защиты объекта SACL отсутствует, обращения субъектов к этому объекту не регистрируются.

При создании нового объекта SACL назначается объекту по тем же правилам, что и DACL. При наследовании ACE флаги s и f остаются неизменными.

Для того чтобы событие, связанное с доступом субъекта к объекту, было зафиксировано в журнале аудита, необходимо одновременное выполнение следующих двух условий:

1. Политика аудита ОС допускает регистрацию в журнале аудита событий, связанных с успешным (или неуспешным) доступом субъектов к объектам.

2. SACL объекта содержит хотя бы один ACE, в котором:

– идентификатор субъекта относится к субъекту, открывающему объект;

– установлен флаг s (или соответственно f) и не установлен флаг i;

– после отображения отображаемых прав доступа пересечение маски доступа ACE и маски доступа, содержащей права, запрашиваемые субъектом, непусто.

Таким образом, глобальные настройки политики аудита в отношении доступа субъектов к объектам выполняют роль фильтра, позволяя временно запретить регистрацию успешных/неуспешных попыток доступа всех субъектов ко всем объектам ОС.

Рекомендации для политики аудита

Здесь будет рассмотрен один из наиболее сложных вопросов, которые приходится решать администратору линейки Windows NT. Дело в том, что политика аудита настолько сильно связана с особенностями эксплуатации конкретного экземпляра ОС, что сформулировать эталонную политику аудита просто невозможно.

Более того, даже для конкретного экземпляра ОС нельзя сформулировать адекватную политику аудита «на все времена». Политика аудита должна постоянно меняться, реагируя на изменения в конфигурации ОС и на зарегистрированные опасные события.

В требованиях NCSC к конфигурации Windows NT, которые должны выполняться для соответствия защищенности ОС уровню С2, об аудите говорится очень мало. Фактически утверждается, что аудит должен быть и что при переполнении журнала аудита операционная система должна «показывать синий экран». Детальная проработка политики аудита возлагается на администраторов операционной системы (в этом документе неявно предполагается, что группа администраторов и группа аудиторов совпадают).

При определении политики аудита следует иметь в виду, что адекватность политики аудита заключается не в том, что регистрируется много событий, а в том, что регистрируется ровно столько событий, сколько необходимо. Если подсистема аудита регистрирует слишком много событий, то, с одной стороны, журнал аудита переполняется слишком быстро, а с другой — аудитору трудно выделить в огромном объеме малозначительной информации действительно важные события. Типичная ошибка при определении политики аудита заключается в том, что устанавливается регистрация всех обращений всех субъектов доступа ко всем файлам и поддиректориям системной директории. При такой политике аудита журнал аудита переполняется в считанные минуты.

В связи с вышесказанным не будем давать конкретные указания по поддержанию адекватной политики аудита, а ограничимся несколькими общими рекомендациями:

1. Вход и выход пользователей из системы должен регистрироваться в любом случае.

2. Доступ субъектов к объектам целесообразно регистрировать только в том случае, когда имеются обоснованные подозрения, что один или несколько пользователей злоупотребляют своими полномочиями. В этом случае следует установить регистрацию обращений этих пользователей к тем объектам, к которым они, возможно, несанкционированно обращаются. Если же таких подозрений нет, регистрацию доступа субъектов к объектам целесообразно отключить, поскольку некоторые объекты Windows NT, недоступные для стандартных утилит администрирования, по умолчанию имеют SACL. Если одновременно установлены регистрация успешных обращений субъектов к объектам и регистрация изменений в списке пользователей, то при каждом изменении в списке пользователей в журнале аудита будут регистрироваться два сообщения: одно в категории «доступ к объектам» и другое в категории «изменения в списке пользователей». Таким образом, если установлена регистрация доступа субъектов к объектам, в журнале аудита появляется много малоинтересных сообщений, что затрудняет его анализ.

3. Необходимость регистрации использования субъектами опасных привилегий зависит от версии ОС. Во многих версиях Windows NT регистрация событий данной категории происходит настолько некорректно, что использовать этот механизм нецелесообразно.

4. Успешные попытки внесения изменений в список пользователей должны регистрироваться в любом случае. Неуспешные попытки внесения таких изменений в Windows NT регистрироваться не могут.

5. Изменения в политике безопасности должны регистрироваться обязательно.

6. Регистрировать системные события в большинстве случаев нецелесообразно.

7. Регистрировать запуск и завершение процессов имеет смысл только в том случае, когда есть обоснованные подозрения, что операционная система атакована вирусом, получившим права администратора.

Эти рекомендации не следует рассматривать как догму. Адекватная политика аудита в конкретной реализации ОС может заметно отличаться от вышеописанной.

Администраторы и аудиторы

Архитектура подсистемы аудита Windows NT неявно предполагает совпадение групп администраторов и аудиторов, что заметно снижает защищенность ОС от несанкционированных действий администраторов. Если необходимо сделать группу аудиторов отличной от группы администраторов, следует выполнить следующие действия:

– создать группу по имени, например, Auditors и добавить в нее всех пользователей/аудиторов;

– сделать владельцем файла журнала аудита одного из аудиторов;

– присвоить файлу журнала аудита дескриптор защиты, содержащий DACL вида:

Разрешить — Auditors — все права доступа;

Разрешить — SYSTEM — все права доступа;

– предоставить группе Auditors привилегию аудитора и отнять эту привилегию у группы Administrators.

После выполнения вышеперечисленных действий просматривать и очищать журнал аудита, а также обращаться к SACL объектов ОС могут только пользователи, входящие в группу Auditors.

Поскольку утилита UserManager предоставляет доступ к политике аудита только членам группы Administrators, группа Auditors должна представлять собой подмножество группы

Administrators. При этом администраторы, не обладающие привилегией аудитора, также могут менять политику аудита. По всей видимости, это ошибка программного обеспечения. Есть подозрение, что можно устранить эту ошибку, манипулируя атрибутами защиты ключей реестра Windows NT, однако пока эксперименты не привели ни к каким определенным результатам.

Полное разделение группы аудиторов и группы администраторов в Windows NT с использованием документированных возможностей ОС невозможно. Это является существенным недостатком подсистемы аудита Windows NT.

Достоинства и недостатки подсистемы аудита Windows NT

Подсистема аудита в Windows NT является наиболее «сырой» частью подсистемы защиты и буквально «пропитана» ошибками в программном обеспечении. Единственное достоинство этой подсистемы заключается в том, что она существует и позволяет регистрировать в журнале аудита хоть какие-то события. Ниже перечисляются основные недостатки подсистемы аудита Windows NT:

1. Журнал аудита защищается от несанкционированного доступа только средствами разграничения доступа. Было бы логично дополнить защиту журнала аудита криптографическими средствами, а также средствами обеспечения корректности совместного доступа процессов к объектам. (Например, сервис EventLog при старте открывает файл SecEvent.evt в режиме монопольного доступа и в дальнейшем предоставляет другим процессам все услуги, связанные с чтением, пополнением и очисткой журнала аудита. В этом случае никакой другой процесс независимо от своих полномочий не сможет обратиться к файлу SecEvent.evt без посредничества сервиса EventLog.)

2. Отсутствуют удобные и эффективные средства, позволяющие отделить пользователей/аудиторов от администраторов ОС.

3. Недостаточное количество категорий событий аудита затрудняет управление политикой аудита.

4. Многие важные с точки зрения безопасности системы события (например, старт драйвера) не могут быть зарегистрированы в журнале аудита.

5. Многие объекты ОС, недоступные для стандартных утилит администрирования, по умолчанию имеют SACL, что приводит к регистрации в журнале аудита множества малоинтересных событий.

6. Из-за ошибок в программном обеспечении многие события регистрируются не всегда или регистрируются некорректно.

7. Реальный порядок регистрации многих событий противоречит описанному в документации.

Из вышеперечисленных недостатков следует, что построить и претворить в жизнь адекватную политику аудита с помощью встроенных средств Windows NT крайне трудно. Для организации надежной защиты информации в Windows NT необходимы дополнительные (наложенные) средства регистрации событий.

§ 5. ПРОЦЕССЫ-СЕРВЕРЫ В WINDOWS NT

Общие сведения

При функционировании любой ОС время от времени возникают ситуации, когда пользователь для выполнения некоторых действий должен получить полномочия, недоступные ему в другое время. Например, пользователь должен иметь возможность изменять свой пароль. Но для этого он должен иметь доступ на запись к базе паролей, что недопустимо, так как в этом случае он сможет менять и чужие пароли.

В ОС UNIX для решения этой проблемы предусмотрен механизм SUID/SGiD, позволяющий пользователю запустить программу от имени другого пользователя. В Windows NT этот механизм не поддерживается даже в эмуляторе POSIX.

Если пользователь Windows NT должен осуществить операцию, для самостоятельного выполнения которой у него недостаточно полномочий, пользователь должен запросить услуги одного из процессов-серверов, выполняющихся от имени пользователя или псевдопользователя, обладающего необходимыми полномочиями (как правило, процессы-серверы работают от имени SYSTEM). В стандартной конфигурации Windows NT имеются следующие процессы-серверы:

- csrss.exe — сервер Win32;
- services.exe — диспетчер сервисов;
- lsass.exe — сервер LSA;
- RpcSs.exe — сервер удаленного вызова процедур (RPC);
- spoolss.exe — сервер печати;
- smss.exe — диспетчер сеансов (sessionmanager).

Все процессы-серверы запускаются при старте ОС. Аварийное завершение процесса-сервера приводит к аварийному завершению работы всей операционной системы.

Процессы-серверы могут пользоваться услугами друг друга. Подавляющее большинство системных вызовов, инициируемых прикладными программами Windows NT, выполняются посредством процессов-серверов. Только системные вызовы, не связанные с изменением глобальных данных ОС, могут быть выполнены прикладной программой самостоятельно. Даже при создании окна на экране или при открытии файла программа должна пользоваться услугами процесса-сервера (в данном случае сервера Win32).

Этот механизм заметно повышает надежность ОС. Прикладные программы Windows NT, не имея прямого доступа к глобальным данным ОС, не дают возможности несанкционированно изменить их и тем самым вызвать крах ОС. В то же время организация функционирования ОС по архитектуре клиент-сервер заметно ухудшает ее производительность. Для компенсации этого эффекта в Windows NT применяются кеширование и отложенное выполнение запросов.

Таким образом, временное повышение полномочий пользователя в Windows NT осуществляется не путем запуска новой программы, а путем передачи сообщения процессу, обладающему необходимыми полномочиями. Это практически устраняет угрозу превышения полномочий пользователя подменой программных модулей. Процессы-серверы стартуют до того, как какой-либо пользователь может войти в систему, и к тому времени, когда пользователь работает в системе, инициализация всех процессов серверов завершена и все используемые ими библиотеки уже загружены. Поэтому данная угроза реализуется подменой программного модуля в Windows NT только с помощью закладки.

Олицетворение пользователя

При любом обращении прикладной программы к процессу-серверу запрос выполняется процессом-сервером с правами пользователя или псевдопользователя, от имени которого запущен процесс-сервер. Однако во многих случаях (например, при создании окна на экране) запрос должен быть выполнен процессом-сервером не со своими полномочиями, а с полномочиями клиента.

Для этого в Windows NT реализована концепция олицетворения или воплощения (impersonation) пользователя. Олицетворение пользователя заключается в том, что в контексте процесса, выпол-

няющегося от имени одного пользователя, создается поток, выполняющийся от имени другого пользователя (олицетворяющий этого пользователя).

Подсистема разграничения доступа Windows NT при определении прав доступа пользователя к некоторому объекту, а также при проверке привилегий для выполнения некоторой информации берет необходимую информацию из маркера доступа пользователя. Обычно маркер доступа назначается процессу, и все потоки процесса выполняются с полномочиями, определенными этим маркером доступа.

Однако маркеры доступа могут назначаться не только процессам, но и отдельным потокам процессов. При олицетворении пользователя одному из потоков процесса-сервера назначается личный маркер доступа. При этом данный поток выполняется не с правами пользователя, запустившего процесс, а с правами пользователя, маркер доступа которого назначен данному потоку. Назначение маркера доступа конкретному потоку процесса и есть олицетворение пользователя.

Таким образом, процесс-сервер имеет потоки двух типов: потоки, имеющие свои маркеры доступа, и потоки, не имеющие своих маркеров доступа. Полномочия потоков второго типа определяются маркером доступа, назначенным процессу.

Олицетворение пользователя не требует от пользователя или псевдопользователя, от имени которого запущен процесс-сервер, никаких специальных привилегий. Все, что нужно этому пользователю или псевдопользователю, — это право «назначение маркера доступа» на доступ к потоку и право «назначение потоку» на доступ к маркеру доступа (если маркер доступа при олицетворении дублируется, вместо права «назначение потоку» на доступ к маркеру доступа необходимо иметь права «чтение» и «дублирование»).

Олицетворение пользователя осуществляет системная функция SetThreadToken, находящаяся в библиотеке advapi32.dll и позволяющая назначить любому потоку любой маркер доступа (естественно, если имеются необходимые права доступа к потоку и к маркеру доступа). Процессы-серверы чаще используют более высокоуровневые функции ImpersonateLoggedOnUser, ImpersonateNamedPipeClient и DdeImpersonateClient, назначающие маркер доступа текущему потоку. Для завершения олицетворения пользователя текущим потоком используется функция RevertToSelf. Для начала и завершения олицетворения клиента сервером RPC используются соответственно функции RpcImpersonateClient и RpcRevertToSelf.

Механизм олицетворения может также использоваться обычными прикладными программами, если необходимо включить некоторую привилегию в одном потоке и оставить ее выключенной для остальных потоков процесса. Для этого следует назначить потоку копию маркера доступа, назначенного процессу, и в дальнейшем включить нужную привилегию в этой копии. Другими словами, пользователь олицетворяет самого себя в одном из потоков процесса и в дальнейшем модифицирует это свое олицетворение. Эта операция реализуется системной функцией ImpersonateSelf.

Если поток процесса-сервера, выполняющийся в режиме олицетворения пользователя, порождает новый процесс, этому процессу назначается маркер доступа процесса-родителя. Другими словами, олицетворение пользователя не распространяется на порождаемые процессы.

Когда поток, выполняющийся в режиме олицетворения пользователя, выполняет привилегированную операцию, наличие соответствующей привилегии проверяется с помощью маркера доступа потока. Некоторые привилегии, например привиле-

гия добавлять записи в журнал аудита, проверяются относительно маркера доступа процесса, что позволяет процессам-серверам генерировать сообщения аудита, не прерывая олицетворение пользователя.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

Аутентификация. Теория и практика обеспечения безопасного доступа к информационным ресурсам : учеб. пособие / А.А. Афанасьев, Л.Т. Веденьев, А.А. Воронцов, Э.Р. Газизова ; под ред. А.А. Шелупанова [и др.]. — 2-е изд., стер. — Москва : Горячая линия — Телеком, 2012. — 550 с. — Текст: электрон. // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/5114 (дата обращения: 04.04.2023). — Режим доступа: для авториз. пользователей.

Бирюков, А.А. Информационная безопасность: защита и нападение : учебник / А.А. Бирюков. — Москва : ДМК Пресс, 2012. — 474 с. — Текст: электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/39990 (дата обращения: 04.04.2023). — Режим доступа: для авториз. пользователей.

Ададуров, С.Е. Информационная безопасность и защита информации на железнодорожном транспорте : учеб. пособие / С.Е. Ададуров, А.П. Глухов, А.А. Корниенко, С.В. Диасамидзе. — Москва : УМЦ по образованию на ж/д трансп., 2014. — 352 с.

Проскурин, В.Г. Защита в операционных системах : учеб. пособие / В.Г. Проскурин. — Москва : Горячая линия — Телеком, 2016. — 192 с. — Текст: электрон. // Лань : электронно-библиотечная система. — URL: http://e.lanbook.com/books/element.php?pl1_id=63241 (дата обращения: 04.04.2023). — Режим доступа: для авториз. пользователей.

Гордеев, А.В. Операционные системы : учеб. для вузов / А.В. Гордеев. — Санкт-Петербург : Питер, 2007. — 2-е изд. — 416 с.

Столлингс, В. Операционные системы: внутреннее устройство и принципы проектирования / В. Столлингс. — Москва : Вильямс, 2004.

Мартемьянов, Ю.Ф. Операционные системы. Концепции построения и обеспечения безопасности : учеб. пособие / Ю.Ф. Мартемьянов, Ал.В. Яковлев, Ан.В. Яковлев. — Москва : Горячая линия-Телеком, 2011. — 332 с. — ISBN 978-5-9912-0128-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/5176 (дата обращения: 04.04.2023). — Режим доступа: для авториз. пользователей.

Мельников, В.П. Информационная безопасность и защита информации / В.П. Мельников, С.А. Клейменов, А.М. Петраков. — Москва : Академия, 2007.

Учебное издание

Бутин Александр Алексеевич
Носков Сергей Иванович
Торопов Виктор Дмитриевич

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Учебное пособие

Подготовлено к печати Т.В. Мари
Дизайн обложки А.А. Мартыновой

ИД № 06318 от 26.11.01.
Подписано в печать 24.04.23. Формат 60×90 1/16. Бумага офсетная.
Печать цифровая. Усл. печ. л. 6,1. Тираж 300 экз. (1-й з-д 1–30).
Заказ .

Издательский дом ФГБОУ ВО «БГУ».
Отпечатано в ИПО ФГБОУ ВО «БГУ».
664003, г. Иркутск, ул. Ленина, 11.
http://bgu.ru.