МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ
И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

Методические указания для проведения практических занятий по дисциплине

Безопасность информационных процессов в компьютерных системах и сетях

(направление подготовки 09.03.01 - Информатика и вычислительная техника)

Ростов-на-Дону
2022

Методические указания для проведения практических занятий по дисциплине

Безопасность информационных процессов в компьютерных системах и сетях

Составил: И.А. Сосновский, доцент кафедры ИТСС

Рассмотрено и одобрено
на заседании кафедры
Протокол от «19» декабря 2022 г. № 5

Практическое занятие №1. Нормативно-правовая база защиты компьютерных сетей от несанкционированного доступа. Компьютерные преступления и особенности их расследования.

***Цель работы*:** ознакомиться с нормативно-правовой базой защиты компьютерных сетей от несанкционированного доступа.

Несанкционированный доступ – чтение, обновление или разрушение информации при отсутствии на это соответствующих полномочий.

Несанкционированный доступ осуществляется, как правило, с использованием чужого имени, изменением физических адресов устройств, использованием информации, оставшейся после решения задач, модификацией программного и информационного обеспечения, хищением носителя информации, установкой аппаратуры записи.

Для успешной защиты своей информации пользователь должен иметь абсолютно ясное представление о возможных путях несанкционированного доступа. Основные типовые пути несанкционированного получения информации:

· хищение носителей информации и производственных отходов;

· копирование носителей информации с преодолением мер защиты;

· маскировка под зарегистрированного пользователя;

· мистификация (маскировка под запросы системы);

· использование недостатков операционных систем и языков программирования;

· использование программных закладок и программных блоков типа "троянский конь";

· перехват электронных излучений;

· перехват акустических излучений;

· дистанционное фотографирование;

· применение подслушивающих устройств;

· злоумышленный вывод из строя механизмов защиты и т.д..

Для защиты информации от несанкционированного доступа применяются:

1) организационные мероприятия;

2) технические средства;

3) программные средства;

4) шифрование.

Организационные мероприятия включают в себя:

· пропускной режим;

· хранение носителей и устройств в сейфе (дискеты, монитор, клавиатура и т.д.);

· ограничение доступа лиц в компьютерные помещения и т.д..

Технические средства включают в себя:

· фильтры, экраны на аппаратуру;

· ключ для блокировки клавиатуры;

· устройства аутентификации – для чтения отпечатков пальцев, формы руки, радужной оболочки глаза, скорости и приемов печати и т.д.;

· электронные ключи на микросхемах и т.д.

Программные средства включают в себя:

· парольный доступ – задание полномочий пользователя;

· блокировка экрана и клавиатуры с помощью комбинации клавиш в утилите Diskreet из пакета Norton Utilites;

· использование средств парольной защиты BIOS – на сам BIOS и на ПК в целом и т.д.

Шифрование–это преобразование (кодирование) открытой информации в зашифрованную, не доступную для понимания посторонних. Методы шифрования и расшифровывания сообщения изучает наука криптология, история которой насчитывает около четырех тысяч лет.

Защита информации в беспроводных сетях

Невероятно быстрые темпы внедрения в современных сетях беспроводных решений заставляют задуматься о надежности защиты данных[17].

Сам принцип беспроводной передачи данных заключает в себе возможность несанкционированных подключений к точкам доступа.

Не менее опасная угроза - вероятность хищения оборудования. Если политика безопасности беспроводной сети построена на MAC-адресах, то сетевая карта или точка доступа, украденная злоумышленником, может открыть доступ к сети.

Часто несанкционированное подключение точек доступа к ЛВС выполняется самими работниками предприятия, которые не задумываются о защите.

Решением подобных проблем нужно заниматься комплексно. Организационные мероприятия выбираются исходя из условий работы каждой конкретной сети. Что касается мероприятий технического характера, то весьма хорошей результат достигается при использовании обязательной взаимной аутентификации устройств и внедрении активных средств контроля.

В 2001 году появились первые реализации драйверов и программ, позволяющих справиться с шифрованием WEP. Самый удачный - PreShared Key. Но и он хорош только при надежной шифрации и регулярной замене качественных паролей (рис.1).

Рисунок 1 - Алгоритм анализа зашифрованных данных

Современные требования к защите

Аутентификация

В настоящее время в различном сетевом оборудовании, в том числе в беспроводных устройствах, широко применяется более современный способ аутентификации, который определен в стандарте 802.1x - пока не будет проведена взаимная проверка, пользователь не может ни принимать, ни передавать никаких данных.

Ряд разработчиков используют для аутентификации в своих устройствах протоколы EAP-TLS и PEAP, Cisco Systems, предлагает для своих беспроводных сетей, помимо упомянутых, следующие протоколы: EAP-TLS, PEAP, LEAP, EAP-FAST.

Все современные способы аутентификации подразумевают поддержку динамических ключей.

Главный недостаток LEAP и EAP-FAST - эти протоколы поддерживаются в основном в оборудовании Cisco Systems (рис. 2).

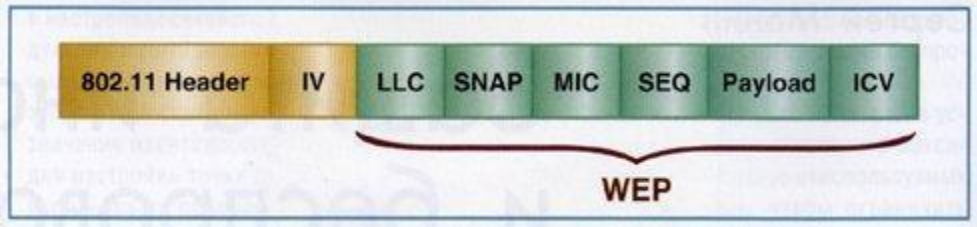

Рисунок 2 - Структура пакета 802.11x при использовании TKIP-PPK, MIC и шифрации по WEP.

Шифрование и целостность

На основании рекомендаций 802.11i Cisco Systems реализован протокол TKIP (Temporal Integrity Protocol), обеспечивающий смену ключа шифрования PPK (Per Packet Keying) в каждом пакете и контроль целостности сообщений MIC (Message Integrity Check).

Другой перспективный протокол шифрования и обеспечения целостности - AES (Advanced Encryption Standart). Он обладает лучшей криптостойкостью по сравнению DES и ГОСТ 28147-89. Он обеспечивает и шифрацию, и целостность.

Заметим, что используемый в нем алгоритм (Rijndael) не требует больших ресурсов ни при реализации, ни при работе, что очень важно для уменьшения времени задержки данных и нагрузки на процессор.

Стандарт обеспечения безопасности в беспроводных локальных сетях - 802,11i.

Стандарт Wi-Fi Protected Access (WPA) - это набор правил, обеспечивающих реализацию защиты данных в сетях 802.11x. Начиная с августа 2003 года соответствие стандартам WPA является обязательным требованием к оборудования, сертифицируемому на звание Wi-Fi Certified.

В спецификацию WPA входит измененный протокол TKOP-PPK. Шифрование производится на сочетании нескольких ключей - текущего и последующего. При этом длина IV увеличена до 48 бит. Это дает возможность реализовать дополнительные меры по защите информации, к примеру ужесточить требования к реассоциациям, реаутентификациям.

Спецификации предусматривают и поддержку 802.1x/EAP, и аутентификацию с разделяемым ключом, и, несомненно, управление ключами.

Рекомендуется распределять пользователей с разной степенью защищенности по разным виртуальным ЛС, и, в соответствии с этим, реализовывать политику безопасности (табл.3).

Таблица 3 - Способы реализации политики безопасности

Показатель	Способ			
	LEAP	EAP-FAST	PEAP	EAP-TLS
Поддержка современных ОС	Да	Да	Не все	Не все
Сложность ПО и ресурсоёмкость аутентификации	Низкая	Низкая	Средняя	Высокая

Сложность управления	Низкая *	Низкая	Средняя	Средняя
Single Sign on (единый логин в Windows)	Да	Да	Нет	Да
Динамические ключи	Да	Да	Да	Да
Одноразовые пароли	Нет	Да	Да	Нет
Поддержка баз пользователей не в формате MS Windows	Нет	Да	Да	Да
Fast Secure Роуминг	Да	Да	Нет	Нет
Возможность локальной аутентификации	Да	Да	Нет	Нет
* Сложность управления низкая, но необходима продуманная политика генерации паролей, что усложняет управление.				

При условии использования современного оборудования и ПО в настоящее время вполне возможно построить на базе стандартов серии 802.11x защищенную и устойчивую к атакам беспроводную сеть.

Почти всегда беспроводная сеть связана с проводной, а это, помимо необходимости защищать беспроводные каналы, необходимо обеспечивать защиты в проводных сетях. В противном случае сеть будет иметь фрагментарную защиту, что, по сути, является угрозой безопасности. Желательно использовать оборудование, имеющее сертификат Wi-Fi Certified, то есть подтверждающий соответствие WPA.

Нужно внедрять 802.11x/EAP/TKIP/MIC и динамическое управление ключами. В случае смешанной сети следует использовать виртуальные локальные сети; при наличии внешних антенн применяется технология виртуальных частных сетей VPN.

Необходимо сочетать как протокольные и программные способы защиты, так и административные.

Практическое занятие №2. Основные виды формальных моделей безопасности. Применение иерархического метода для построения защищенной операционной системы; исследование корректности систем защиты; методология обследования и проектирования защиты; модель политики контроля целостности.

***Цель работы*:** ознакомиться с основными видами и формальными моделями безопасности.

Формальные модели безопасности

Наибольшее развитие получили два подхода, каждый из которых основан на своем видении проблемы безопасности и нацелен на решение определенных задач, — это формальное моделирование политики безопасности и криптография. Причем эти различные по происхождению и решаемым задачам подходы дополняют друг друга: криптография может предложить конкретные методы защиты информации в виде алгоритмов идентификации, аутентификации, шифрования и контроля целостности, а формальные модели безопасности предоставляют разработчикам основополагающие принципы, лежащие в основе архитектуры защищенной системы и определяющие концепцию ее построения.

Модель политики безопасности — формальное выражение политики безопасности. Формальные модели используются достаточно широко, потому что только с их помощью можно доказать безопасность системы, опираясь при этом на объективные и неопровержимые постулаты математической теории.

Основная цель создания политики безопасности ИС и описания ее в виде формальной модели — это определение условий, которым должно подчиняться поведение системы, выработка критерия безопасности и проведение формального доказательства соответствия системы этому критерию при соблюдении установленных правил и ограничений. На практике это означает, что только соответствующим образом уполномоченные пользователя получат

доступ к информации и смогут осуществлять с ней только санкционированные действия.

Среди моделей политик безопасности можно выделить два основных класса: дискреционные (произвольные) и мандатные (нормативные). В данном подразделе в качестве примера изложены основные положения наиболее распространенных политик безопасности, основанных на контроле доступа субъектов к объектам.

Дискреционная модель Харрисона—Руззо — Ульмана.

Модель безопасности Харрисона—Руззо —Ульмана, являющаяся классической дискреционной моделью, реализует произвольное управление доступом субъектов к объектам и контроль за распространением прав доступа.

В рамках этой модели система обработки информации представляется в виде совокупности активных сущностей — субъектов (множество *S),*которые осуществляют доступ к информации, пассивных сущностей — объектов (множество *O),* содержащих защищаемую информацию, и конечного множества прав доступа $R = \{r_1, \ldots, r_n\}$, означающих полномочия на выполнение соответствующих действий (например, чтение, запись, выполнение).

Причем для того чтобы включить в область действия модели и отношения между субъектами, принято считать, что все субъекты одновременно являются и объектами. Поведение системы моделируется с помощью понятия «состояние». Пространство состояний системы образуется декартовым произведением множеств составляющих ее объектов, субъектов и прав — *OSR.* Текущее состояние системы Q в этом пространстве определяется тройкой, состоящей из множества субъектов, множества объектов и матрицы прав доступа *M,* описывающей текущие права доступа субъектов к объектам, — $Q = (S, O, M)$. Строки матрицы соответствуют субъектам, а столбцы — объектам, поскольку множество объектов включает в себя множество субъектов, матрица имеет вид прямоугольника. Любая ячейка матрицы $M[s, o]$ содержит набор прав субъекта s к объекту o, принадлежащих множеству

прав доступа *R*. Поведение системы во времени моделируется переходами между различными состояниями. Переход осуществляется путем внесения изменений в матрицу *M* с помощью команд следующего вида:

command a (x_1, ..., x_k)
If r_1 in $M[x_{S_1}, x_{O_1}]$ and (*условия выполнения команды*)
r_1 in $M[x_{S_2}, x_{O_2}]$ and
.
.
r_m in $M[x_{S_m}, x_{O_m}]$
then
op_1, op_2, ..., op_n (*операции, составляющие команду*)

Здесь *a* — имя команды; x_i — параметры команды, являющиеся идентификаторами субъектов и объектов; s_i и o_i — индексы субъектов и объектов в диапазоне от 1 до *k*; op_i — элементарные операции.

Элементарные операции, составляющие команду, выполняются только в том случае, если все условия, означающие присутствие указанных прав доступа в ячейках матрицы *M,* являются истинными. В классической модели допустимы только следующие элементарные операции:

enter *r* into *M[s, o] (добавление субъекту s права r для объекта o)*

delete *r* from *M[s, o] (удаление у субъекта s права r для объекта o)*

create subject *s (создание нового субъекта s)*

create object *o (создание нового объекта o)*

destroy subject *s (удаление существующего субъекта s)*

destroy object *o (удаление существующего объекта o)*

Критерий безопасности модели Харрисона — Руззо — Ульмана формулируется следующим образом.

Для заданной системы начальное состояние $Q_0 = (S_0, O_0, M_0)$ является безопасным относительно права *r*, если не существует применимой к Q_0 последовательности команд, в результате которой право *r* будет занесено в ячейку матрицы *M,* в которой оно отсутствовало в состоянии Q_0.

Нужно отметить, что все дискреционные модели уязвимы по отношению к атаке с помощью «троянского коня», поскольку в них контролируются только

операции доступа субъектов к объектам, а не потоки информации между ними. Поэтому, когда «троянская» программа, которую нарушитель подсунул некоторому пользователю, переносит информацию из доступного этому объекта в объект, доступный нарушителю, то формально никакое правило дискреционной политики безопасности не нарушается, но утечка информации происходит.

Таким образом, дискреционная модель Харрисона—Руззо — Ульмана в своей общей постановке не дает гарантий безопасности системы, однако именно она послужила основой для целого класса моделей политик безопасности, которые используются для управления доступом и контроля за распределением прав во всех современных системах.

Типизованная матрица доступа. Другая дискреционная модель, получившая название «Типизованная матрица доступа» (*Type Access Matrix*— далее *TAM),* представляет собой развитие модели Харрисона—Руззо—Ульмана, дополненной концепцией типов, что позволяет несколько смягчить те условия, для которых возможно доказательство безопасности системы.

Состояние системы описывается четверкой $Q = (S, O, t, M)$, где S, O и M обозначают соответственно множество субъектов, объектов и матрицу доступа, а $t: O \rightarrow T$ — функция, ставящая в соответствие каждому объекту некоторый тип.

Состояние системы изменяется с помощью команд из множества C. Команды *TAM* имеют тот же формат, что и в модели Харрисона—Руззо—Ульмана, но всем параметрам приписывается определенный тип:

```
command a(x1 : t1, xk : tk)
    If r1 in M[xS1, xO1] and
       r1 in M[xS2, xO2] and
    .
    .
    rm in M[xSm, xOm]
    then
    op1, op2, ..., opn
```

(*условия выполнения команды*)

(*операции, составляющие команду*)

Перед выполнением команды происходит проверка типов фактических параметров, и, если они не совпадают с указанными в определении, команда не выполняется. Фактически введение контроля типов для параметров команд приводит к неявному введению дополнительных условий, так как команды могут быть выполнены только при совпадении типов параметров. В модели используются следующие шесть элементарных операций, отличающихся от аналогичных операций модели Харрисона—Руззо —Ульмана только использованием типизованных параметров:

enter r into $M[s, o]$
create subject s of type t
create object o of type t
} (*монотонные операции*)

delete r from $M[s, o]$
destroy subject s
destroy object o
} (*немонотонные операции*)

Смысл элементарных операций совпадает со смыслом аналогичных операций их классической модели Харрисона — Руззо — Ульмана с точностью до использования типов.

Таким образом, *TAM* является обобщением модели Харрисона—Руззо—Ульмана, которую можно рассматривать как частный случай *TAM* содним-единственным типом, к которому относятся все объекты и субъекты. Появление в каждой команде дополнительных неявных условий, ограничивающих область применения команды только сущностями соответствующих типов, позволяет несколько смягчить жесткие условия классической модели, при которых критерий безопасности является разрешимым.

Несмотря на различающиеся подходы, суть всех моделей безопасности одинакова, поскольку они предназначены для решения одних и тех же задач. Целью построения модели является получение формального доказательства безопасности системы, а также определения достаточного критерия безопасности.

Безопасность системы определяется в равной степени тремя факторами: свойствами самой модели, ее адекватностью угрозам, воздействующим на

систему, и тем, насколько корректно она реализована. Поскольку при существующем разнообразии теоретических наработок в области теории информационной безопасности выбор модели, адекватной заданным угрозам, не является проблемой, последнее, решающее слово остается за ее реализацией в защищенной системе.

Иерархический метод построения защиты.

Создание гарантированно защищенных баз данных связано с некоторыми общими проблемами синтеза систем защиты. Если политика безопасности в базе данных не включает вопросов, связанных с взаимным выводом информации и каналов утечки, основанных на этом выводе, вопросов восстановления зашумленной информации путем повторных запросов в базу данных, то такая политика не является адекватной для безопасности информации. Однако, все эти вопросы нельзя включать в политику безопасности, поддерживаемую самой вычислительной системой, что приведет к симбиозу операционной системы и системы управления базой данных. Вместе с тем, сложилась такая практика, что производителями операционных систем и систем управления базами данных являются разные фирмы или организации, что делает такой симбиоз невозможным. Поэтому политика безопасности частично должна поддерживаться самой базой данных. Аналогичные проблемы возникают при модернизации защищенных систем и могут быть сформулированы как противоречие между единой оценкой защищенности всей системы и многопрофильностью подсистем, создаваемых различными производителями. С аналогичной проблемой мы сталкивались в тема 8. Однако теперь существенным образом возникает зависимость частей защиты друг от друга (ТСВ вычислительной системы управляет субъектами системы защиты (ТСВ), поддерживающей политику безопасности базы данных). Такая структура не вкладывается в систему независимых защищенных компонент распределенной сети. В параграфе 7.1 приводится описание модели ТСВ-подмножеств, иерархически связанных друг с другом, которые при выполнении некоторых дополнительных условий могут удовлетворять

условиям ТСВ системы. В п. 9.2 приведены наиболее распространенные архитектуры защищенных баз данных, в которых теория п.9.1 может быть реализована.

Иерархический метод построения защиты.

В данном п. рассматривается еще один пример иерархической декомпозиции сложных систем. Если в п. 2.2 мы рассматривали примеры, не связанные непосредственно с задачей защиты, то сейчас основное внимание будет посвящено иерархическому построению ТСВ в электронных системах обработки данных. Как и раньше, основная задача ТСВ - поддержка монитора обращений. Однако, в отличие от теории предыдущего параграфа, где мониторы обращения были независимыми, сейчас мы покажем, что одни ТСВ-подмножества могут использовать ресурсы в других ТСВ-подмножествах. А именно, рассмотрим следующую модель.

Определение. ТСВ-подмножество M есть совокупность программно-аппаратных ресурсов системы, которые управляют доступами множества S субъектов к множеству O объектов на основе четко определенной политики P и удовлетворяет свойствам:

1) M определяет и контролирует каждый доступ к объектам из O со стороны субъектов из S;

2) M гарантировано защищено;

3)M достаточно просто устроено, чтобы существовала возможность проанализировать его системой тестов, полнота которых доказана.

Зависимость ТСВ-подмножеств состоит в том, что M использует ресурсы точно определенного множества более примитивных ТСВ-подмножеств (то есть предполагается заданным некоторый частичный порядок на ТСВ-подмножествах), для того, чтобы создать объекты из O, создать и поддерживать структуры данных и поддерживать политику P.

Если более примитивных ТСВ-подмножеств нет, то такое ТСВ-подмножество опирается в решении тех же задач на аппаратную часть.

Отметим, что кроме монитора обращений необходимо существование механизма поддержки этого монитора. Рассмотрим условия, при которых из ТСВ-подмножеств удается собрать ТСВ системы (то есть доказать, что выполняются все требования к ТСВ).

Пусть даны ТСВ-подмножества M(i), которые управляют доступом субъектов S(i) к объектам 0(i) в соответствии с политикой безопасности P(i). Предположим, что нет объектов в системе, которые контролируются более, чем одним ТСВ-подмножеством. Здесь принимается та же точка зрения, как в главе VI, состоящая в том, что ТСВ-подмножества могут экспортировать объекты, подлежащие управлению другим ТСВ-подмножеством. Однако принятый и переданный объекты - это разные объекты, контролируемые разными ТСВ-подмножествами. При этом политика безопасности P(i) ТСВ-подмножества M(i) может отличаться от политики безопасности P(j) ТСВ-подмножества M(j). Однако все вместе должны составлять единую политику P системы, причем каждое правило из P должно поддерживаться определенной системой ТСВ-подмножеств.

Необходимо также предполагать, что каждый доверенный субъект, то есть субъект, который может нарушать правила P(i) является частью ТСВ-подмножества M(i).

Рассмотрим ограничения на зависимости ТСВ-подмножеств.

Определение. ТСВ-подмножество A прямо зависит (в своей правильности) от ТСВ-подмножества B тогда и только тогда, когда доводы о подтверждении правильности A (верификация установки A обозначается vA) частично или полностью основаны на предположении, что B установлена верно в соответствии соспецификацией B (обозначается sB).

Определение. ТСВ-подмножество A менее примитивно, чем ТСВ-подмножество B, если:

а) A прямо зависит от B;

б) существует цепочка ТСВ-подмножеств от A к B такая, что каждый элемент цепи прямо зависит от предыдущего элемента цепи.

Рассмотрим примеры, поясняющие понятие зависимости, взятые из "Розовой книги".

Пример 1. Пусть TCB-подмножество B предоставляет услугу в виде "файла", которым B управляет в соответствии с политикой P(B), а TCB-подмножествоA использует B-файл для хранения информации. Если vA использует факт, что различные B-файлы действительно различаются и доступ к ним определяется политикой P(B), то vA полагается на факт, что sB и B соответствуют друг другу. Тогда A прямо зависит от B.

Пример 2. Пусть A и B взаимно не доверяющие друг другу системы бронирования авиабилетов, расположенные отдельно и принадлежащие различным организациям. Предположим sA и sB дают возможность получить заказ на бронирование по сети от других систем, используя взаимно согласованный протокол. Пусть этот протокол полностью определен и соответствует sA и sB. Пусть также vA и vB с заданной степенью уверенности подтверждают, что A и B соответствуют своим спецификациям sA и sB. A не зависит в правильности своей установки от правильности установки B, так как sA - полная, то есть какая бы последовательность бит не пришла от B, sA определяет, что A должна делать, а vA демонстрирует, что именно это и делается. Аналогично, B не зависит от правильности A. Поэтому A и B не зависят одна отдругой.

Пример 3. Пусть A является сервером электронной почты, а B управляет запросами в базу данных. Спецификация sA может совсем не упоминать систему управления базой данных, она просто определяет интерфейс почтовой системы. Однако в vA мы находим, что программа обеспечения A использует таблицы, предоставляемые B, для хранения сообщений A и B на различных, связанных машинах. Ни sB, ни vB не упоминают о почтовой системе. Как и в предыдущем примере, sB полностью определяет поведение B для всех получаемых последовательностей бит. Здесь A прямо зависит от B, но B не зависит от A. Отметим, что информационные потоки в обоих направлениях

являются законными и никоим образом не компрометируют целостность B. Зависимость находится в другой плоскости от потока данных.

Этот пример замечателен тем, что анализ структуры элементов не позволяет выявить существование зависимости, при этом архитектура системы внешне совпадает с приведенной в примере 2 архитектурой взаимно независимых систем. Кроме того, этот пример показывает, что описания интерфейса не достаточно для выявления зависимостей.

Пример 4. Пусть A и B взаимно зависимые системы. A зависит от B и B зависит от A. Это значит, что правильность установки vA доказывается из предположения, что B установлена правильно в соответствии с sB. Также правильность установки vB доказывается из предположения, что A установлена правильно в соответствии с sA. Пусть vA и vB подтверждают правильность установки A и B. Однако отсюда не следует, что A и B функционируют правильно.

В самом деле, если A и B функционируют правильно относительно sA и sB, то vA и vB поддерживают правильность установки. Если A установлено неправильно по отношению к sA и B установлено неправильно по отношению к sB, то никто не мешает возникновению ситуации, когда vA подтверждает правильность исходя из того, что B не соответствует sB. Аналогично, vB подтверждает правильность установки sB, хотя A не соответствует sA. Тогда можно доказать, что vA и vB подтверждают правильность A и B, хотя A и B установлены неверно.

Для того, чтобы понять возникающую здесь коллизию, рассмотрим две системы бронирования билетов на самолеты A и B, как в примере 2. Предположим, что A содержит информацию об исходных пунктах и времени вылета всех полетов в США, а B - в Европе. Пусть sA включает утверждение, что A обеспечивает пассажирам услугу в подборе вылета по транзиту, минимизирующем время ожидания следующего полета. Пусть sB предлагает аналогичные услуги. Из утверждения A соответствует sA и B соответствует sB можно вывести истинность утверждения, что A соответствует спецификации sA

в предоставлении услуги подбора транзитного рейса в Европе и Америке. Аналогичное утверждение можно вывести для В. Однако, если А хранит информацию в местном времени, а В - во времени по Гринвичу, то транзитный маршрут, составленный А и В на основе информации, полученной друг от друга, будет неправильным из-за различия во времени. То есть каждая система функционирует правильно, но результат неверен. Это происходит из-за того, что А и В зависимы.

Гарантии защищенности ТСВ-подмножеств имеют большое значение в проблеме построения единой ТСВ системы из ТСВ-подмножеств. В случае зависимых ТСВ-подмножеств менее примитивное ТСВ-подмножество может использовать объекты и субъекты, предоставленные более примитивным ТСВ-подмножеством. Потому первая проблема состоит в доказательстве того факта, что невозможны никакие изменения данных, критических для политики безопасности, или кодов ТСВ-подмножества. То есть никакая внешняя по отношению к ТСВ-подмножеству система (кроме, может быть, более примитивных ТСВ-подмножеств) не может инициировать произвольное изменение кодов ТСВ-подмножества или его структур данных. Вторая проблема состоит в доказательстве того, что данные, хранящиеся в ТСВ-подмножестве и критичные для политики безопасности, не могут изменяться иначе, чем в соответствии с логикой ТСВ-подмножества. Разумеется, эти доказательства возможны при условии правильности той информации, которая вносится в ТСВ-подмножество более примитивным ТСВ-подмножеством.

Ясно, что в доказательстве возможности построения единой ТСВ системы из ТСВ-подмножеств присутствуют условия не только на локальные свойства ТСВ-подмножеств, но также интегральные требования.

Суммируем перечень всех условий, которые необходимо выполнить, чтобы из семейства ТСВ-подмножеств можно было бы синтезировать ТСВ системы. Если, кроме нижеприведенных требований, выполняются требования какого-либо класса в стандарте "Оранжевая книга", то построенная таким образом система может быть аттестована по соответствующему классу.

Условия:

1. ТСВ-подмножества четко определены.

2. Политика безопасности системы распределена по ТСВ-подмножествам.

3. Каждое ТСВ-подмножество M(i) включает доверенные процессы по отношению к своей политике безопасности P(i).

4. Архитектура ТСВ-подмножеств четко определена .

5. Все ТСВ-подмножества занимают различные домены.

6.Во всех случаях примитивные ТСВ-подмножества поддерживают правильное функционирование монитора обращений в менее примитивном ТСВ-подмножестве.

Исследование корректности реализации и верификация ас

Понятие корректности или правильности подразумевает соответствие проверяемого объекта некоторому эталонному объекту или совокупности формализованных этапонных характеристик и правил. Корректность ПО при разработке наиболее полно определяется степенью соответствия предъявляемым к ней формализованным требованиям программной спецификации. В спецификациях отражается совокупность этапонных характеристик, свойств и условий, которым должна соответствовать программа. Основную часть спецификации составляют функциональные критерии и характеристики. Исходной программной спецификацией, которой должна соответствовать программа, является ТЗ.

При отсутствии полностью формализованной спецификации требований в качестве ТЗ, которому должна соответствовать АС и результаты ее функционирования, иногда испопьзуются неформализованные представления разработчика, пользователя или заказчика программ. Однако понятие корректности программ по отношению к запросам пользователя или заказчика сопряжено с неопределённостью самого эталона, которому должна соответствовать АС. Для сложных программ всегда существует риск обнаружить их некорректность (по мнению пользователя или заказчика) при

формальной корректности относительно спецификаций вследствие неточности самих спецификаций.

Традиционный взгляд на спецификацию требований заключается в том, что она представляет собой документ на естественном языке, который является интерфейсом между заказчиком и изготовителем. Хотя подготовке документа может предшествовать некоторое взаимодействие, именно этот документ в значительной степени выступает как "отправная точка" для изготовителя программ.

Таким образом, можно сделать вывод о том, что создание совокупности взаимоувязанных непротиворечивых спецификаций является необходимой базой для обеспечения корректности проектируемой программы. При этом спецификации должны:

- быть формальными;
- позволять проверять непротиворечивость и полноту требований заказчика;
- служить основой для дальнейшего формализованного проектирования ОС.
- Существует несколько подходов к определению спецификаций требований.

Спецификация как описание. Заказчик выдает спецификацию, чтобы изготовители могли снабдить его тем изделием, которое он желает, поэтому заказчик видит этот документ главным образом как описание системы, которую он желал бы иметь. В принципе, в описании должно быть указано, что должна и что не должна делать система. На практике обычно по умолчанию предполагается, что система должна делать то, что уточняется в спецификации, и не должна делать ничего более. В этом состоит главная проблема с описательной стороной спецификации. Предполагается, что заказчик всегда точно знает всё, что система должна и не должна делать. Более того, в дальнейшем предполагается, что заказчик полностью перенёс это знание в специфицированный документ.

Спецификация как предписание. Изготовитель смотрит на специфицированный документ как на набор составных частей, подлежащих сборке, чтобы разрешить проблему заказчика. Такой предписывающий взгляд обуславливается не только трудностями создания описательного документа (как указывалось выше), но и сведениями, которые умышленно или неумышленно расширяют или ограничивают свободу изготовителя.

Договорная методология. В рамках "описание заказчика - предписание изготовителю" спецификация рассматривается как формальное разделение между сторонами. Что касается заказчика, то он оговаривает минимально приемлемое, тогда как изготовитель - максимально требуемое. Договор предлагается и принимается при зарождении системы и заканчивается после завершения системы, когда заказчик принимает систему как отвечающую его минимальным требованиям. Во время изготовления системы в принципе не предполагается никаких взаимодействий, даже если изготовитель подозревает, что предписываемое не совсем соответствует тому, что заказчик желает видеть в действительности.

Спецификация как модель. Современные более строгие представления о спецификации трактуют ее как модель системы. При условии, что лежащая в основе модели семантика в достаточной мере обоснована, такая спецификация обеспечивает чёткую формулировку требований.

Соответствующие модели подходят также для автоматизированного контроля целостности и другого прогнозного анализа, который, в частности, обеспечит прекращение разработки системы, в принципе не способной удовлетворить требованиям.

Модели как описание системы имеют следующие отличительные черты по сравнению с другими способами формального описания:

- хорошее сочетание нисходящего и восходящего подходов к их разработке с возможностью выбора абстрактного описания;
- возможность описания параллельной, распределенной и циклической работы;

- возможность выбора различных формализованных аппаратов для описания систем.

Основное преимущество использования формальной модели заключается в возможности исследования с ее помощью особенностей моделируемой системы. Основывая формальный метод разработки на математической модели и затем исследуя модель, можно выявить такие грани поведения системы, которые в противном случае не были бы очевидны до более поздних стадий.

Так как целевым объектом проектирования является АС, то модель может описывать либо саму АС, либо ее пбведение, те. внешние проявления функционирования АС. Модель, описывающая поведение АС по сравнению с моделью АС, обладает одним важным преимуществом-она может быть проверена и оценена как исполнителями, так и заказчиками, поскольку заказчики не знают, как должна работать АС, но зато они представляют, что она должна делать. В результате такого моделировайия может быть проверена корректность спецификаций относительно исходной постановки задачи, т.е. ТЗ. Кроме того, критерии правильности считаются достаточными при условии, что спецификация представляет собой исчерпывающее описание "внешнего" поведения объекта при всех возможных (или запланированных) ситуациях его использования.

Как было отмечено выше, при разработке АС, особенно ее компонентов, представляющих систему защиты информации, для обеспечения высоких гарантий отсутствия неисправностей и последующего доказательства того, что система функционирует согласно требованиям ТЗ, используются формальные подходы к ее проектированию.

Формальное проектирование алгоритмов базируется, в основном, на языках алгоритмических логик, которые включают высказывание вида

Q{S}R.

читающееся следующим образом: "если до исполнения оператора S было выполнено условие Q, то после него будет R". Здесь Q называется преду-словием, а R-постусловием. Эти языки были изобретены практически

одновременно Р. У. Флойдом (1967 г.), С. А. Р. Хоаром (1969 г.) и учеными польской логической школы (А. Сальвицкий и др.. 1970 г.). Как предусловие, так и постусловие являются предикатами.

Рассмотрение программ в качестве некоего "преобразователя предикатов" позволяет прямо определить связь между начальными и конечными состояниями без каких-либо ссылок на промежуточные состояния, которые могут возникнуть во время выполнения программы.

Преимущество представления алгоритма в виде преобразователя предикатов состоит в том, что оно дает возможность:

- анализировать алгоритмы как математические объекты;
- дать формальное описание алгоритма, позволяющее интеллектуально охватить алгоритм;
- синтезировать алгоритмы по представленным спецификациям;
- провести формальное верифицирование алгоритма, т.е. доказать корректность его реализации.

Методология формальной разработки и доказательства корректности алгоритмов в настоящее время хорошо разработана и изложена в целом ряде работ. Вкратце суть этих методов сводится к следующему:

- разработка алгоритма проводится методом последовательной декомпозиции, с разбивкой общей задачи, решаемой алгоритмом, на ряд более мелких подзадач;
- критерием детализации подзадач является возможность их реализации с помощью одной конструкции ветвления или цикла;
- разбиение общей задачи на подзадачи предусматривает формулирование пред- и постусловий для каждой подзадачи с целью их корректного проектирования и дальнейшей верификации.

Для доказательства корректности алгоритма (верификация) формулируется математическая теорема Q{S}R, которая затем доказывается. Доказательство теоремы о корректности принято разбивать на две части. Одна часть служит для доказательства того, что рассматриваемый алгоритм вообще

может завершить работу (проводится анализ всех циклов). В другой части доказывается корректность постусловия в предположении, что алгоритм завершает работу.

Методология обследования и проектирования систем защиты при организации информационной безопасности

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Требования нормативных документов к средствам защиты информации, проведение обследования и испытаний средств защиты информации.

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников

Управление процессами функционирования систем защиты при организации информационной безопасности

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Определение процесса, функции системы защиты, администрирование системы защиты.

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников

Перечень примерной тематики практических работ по теоретическому материалу

1. Виды моделей безопасности. Модель дискреционного разграничения доступа
2. Виды моделей безопасности. Модель мандатного разграничения доступа.
3. Виды моделей безопасности. Модель безопасности информационных потоков.
4. Виды моделей безопасности. Модель ролевого разграничения доступа
5. Политика безопасности. Модель Белла-ЛаПадула и Low-Water-Mark
6. Политика безопасности. Модель БИБА
7. Примеры практической реализации. Программно-аппаратная реализация системы безопасности компьютерной информации. SecretNet, SecretDisk.
8. Примеры практической реализации. Аппаратная реализация системы безопасности компьютерной информации. Ключи HASP.
9. Примеры практической реализации. Программная реализация системы безопасности компьютерной информации. Антивирусная защита, установка и настройка пакета Shadow Defender. Ревизор XP.
10. Управление пользователями и правами в ОС Windows 2000, Windows XP
11. Построение парольных систем. Генераторы паролей. Оценка стойкости парольной защиты.
12. Построение парольных систем. Защита документов в Microsoft Office. Защита архивов.
13. Методы криптографической защиты. Принципы криптографической защиты информации. Традиционные симметричные криптосистемы.
14. Методы криптографической защиты. Элементы криптоанализа.

15. Методы криптографической защиты. Особенности реализации систем с несимметричными ключами, несимметричные криптосистемы шифрования, несимметричные алгоритмы.

16. Модель политики контроля целостности. Функция хеширования, ЭЦП.

Код 5 (опд.Ф.11 рп2009 Организационное обеспечение информационной безопасности)

Анализ и оценка угроз информационной безопасности объекта

1 Теоретическая информация (тезисное изложение материала примерно на 20 страниц)

Оценка возможностей технических разведок и других источников угроз безопасности конфиденциальной информации. Риск-Менеджмент. Методы оценки уязвимости информации. Методы оценки достоверности информационной базы моделей прогнозирования значений показателей уязвимости информации. Методы оценки ущерба от реализации угроз ИБ. Эмпирический подход к оценке уязвимости информации. Практическая реализация модели «Угроза-Защита». Методы определения требований к ЗИ

2 Практический пример использования теоретического материала (примерно на 5 страниц)

3 Электронные слайды в MicrosoftOfficePowerPoint2003 по теоретическому материалу в виде рисунков

4 Глоссарий по теоретическому материалу (примерно на 5 страниц)

5 Список используемых информационных источников

Модели контроля целостности

Рассмотрим модели безопасности, контролирующие целостность информации. В частности, модели Биба, использующиеся для синтеза механизмов контроля целостности информации в системе, а также модель Кларка – Вилсона (КВМ), которая является примером неформального

выражения политики безопасности. Последняя модель сформулирована в виде набора неформальных правил, и хотя в литературе она названа моделью безопасности, ее скорее можно назвать политикой контроля целостности.

Модель Биба

Кен Биба в середине семидесятых годов прошлого века сделал два наблюдения. Они были последовательно внесены в модель безопасности, которая с тех пор называется моделью целостности Биба (или просто моделью Биба). В контексте разговора о моделях контроля целостности запись наверх может представлять угрозу в том случае, если субъект с низким уровнем безопасности искажает или уничтожает данные в объекте, лежащем на более высоком уровне. Поэтому, исходя из задач целостности, можно потребовать, чтобы такая запись была запрещена. Кроме того, можно рассматривать чтение снизу как поток информации, идущий из объекта нижнего уровня и нарушающий целостность субъекта высокого уровня. Поэтому весьма вероятно, что и такое чтение необходимо запретить.

Биба выразил свою модель таким же способом, каким была выражена БЛМ, за тем исключением, что правила его модели являются полной противоположностью правилам БЛМ. Возможны три вариации модели Биба: мандатная модель целостности, модель понижения уровня субъекта и модель понижения уровня объекта. Фактически, общий термин «модель Биба» используется для обозначения любой или сразу всех трех моделей.

Мандатная модель целостности Биба

Ее часто называют инверсией БЛМ. Это довольно точное название, поскольку основные правила этой модели просто переворачивают правила БЛМ. Мы будем ссылаться на эти правила как «нет чтения снизу» (NRD) и «нет записи наверх» (NWU) и определим их в терминах субъектов, объектов и нового типа уровней безопасности – уровней целостности, над которыми может быть введено отношение преобладания.

Правило NRD мандатной модели целостности Биба определяется как запрет субъектам на чтение информации из объекта с более низким уровнем

целостности. Правило NWU мандатной модели целостности Биба определяется как запрет субъектам на запись информации в объект с более высоким уровнем целостности.

Одним из преимуществ этой модели является то, что она унаследовала многие важные характеристики БЛМ, включая ее простоту и интуитивность. Это значит, что проектировщики реальных систем могут легко понять суть этих правил и использовать их для принятия решений при проектировании. Кроме того, поскольку мандатная модель целостности Биба, подобно БЛМ, основана на простой иерархии, ее легко объяснить и изобразить пользователям системы.

С другой стороны, модель представляет собой очевидное противоречие с правилами NRU и NWD. Это значит, что если необходимо построить систему, которая предотвращает угрозы как секретности, так и целостности, то одновременное использование правил моделей БЛМ и Биба может привести к ситуации, в которой уровни безопасности и целостности будут использоваться противоположными способами.

Рассмотрим формальное описание модели Биба. Для этого опишем простые математические конструкции, которые помогут описать различные правила, составляющие мандатную модель целостности Биба.

Начнем с представления множества субъектов и объектов. Уровни целостности субъекта или объекта x обозначаются как уровень (x), и для них введено отношение преобладания. Используя эти определения, сформулируем правила NRD и NWU мандатной модели целостности Биба в терминах булевой функции разрешить:

NRD: s S, o O: разрешить (s, o, чтение) уровень (o) > уровень (s).

Данный тип определения предусматривает условия, при которых функция разрешить принимает значение истинно. Определение утверждает, что для всех определенных субъектов и объектов операция чтения разрешена только в том случае, если выполняется условие преобладания. Правило NWU просто переворачивает использование отношения преобладания, как показано в следующем определении:

Практическое занятие №3. Основные этапы построения системы комплексной защиты вычислительных систем; анализ моделей нарушителя; угрозы информационно-программному обеспечению вычислительных систем и их классификация.

***Цель работы*:** ознакомиться с основными этапами построения системы комплексной защиты вычислительных систем.

Система защиты информации должна создаваться совместно с создаваемой компьютерной системой. При построении системы защиты могут использоваться существующие средства защиты, или же они разрабатываются специально для конкретной АС. В зависимости от особенностей компьютерной системы, условий ее эксплуатации и требований к защите информации процесс создания КСЗИ может не содержать отдельных этапов, или содержание их может несколько отличаться от общепринятых норм при разработке сложных аппаратно-программных систем. Но обычно разработка таких систем включает следующие этапы:

- разработка технического задания;
- эскизное проектирование;
- техническое проектирование;
- рабочее проектирование;
- производство опытного образца.

Одним из **основных** этапов разработки КСЗИ является этап разработки технического задания. Именно на этом этапе решаются практически все специфические задачи, характерные именно для разработки КСЗИ.

Процесс разработки систем, заканчивающийся выработкой **технического задания**, называют научно-исследовательской разработкой, а остальную часть работы по созданию сложной системы называют **опытно-конструкторской разработкой**. Опытно-конструкторская разработка аппаратно-программных средств ведётся с применением систем автоматизации проектирования,

алгоритмы проектирования хорошо изучены и отработаны. Поэтому особый интерес представляет рассмотрение процесса научно-исследовательского проектирования.

Научно-исследовательская разработка ксзи

Целью этого этапа является разработка **технического задания** на проектирование КСЗИ. Техническое задание содержит основные технические требования к разрабатываемой КСЗИ, а также согласованные взаимные обязательства заказчика и исполнителя разработки. Технические требования определяют значения основных технических характеристик, выполняемые функции, режимы работы, взаимодействие с внешними системами и т. д.

Аппаратные средства оцениваются следующими характеристиками: быстродействие, производительность, ёмкость запоминающих устройств, разрядность, стоимость, характеристики надёжности и др. Программные средства характеризуются требуемым объёмом оперативной и внешней памяти, системой программирования, в которой разработаны эти средства, совместимостью с ОС и другими программными средствами, временем выполнения, стоимостью и т. д.

Получение значений этих характеристик, а также состава выполняемых функций и режимов работы средств защиты, порядка их использования и взаимодействия с внешними системами составляют основное содержание этапа научно-исследовательской разработки. Для проведения исследований на этом этапе заказчик может привлекать исполнителя или научно-исследовательское учреждение, либо организует совместную их работу.

Научно-исследовательская разработка начинается с анализа угроз безопасности информации, анализа защищаемой АС и анализа конфиденциальности и важности информации в АС.

Сначала производится анализ конфиденциальности и важности информации, которая должна обрабатываться, храниться и передаваться в АС. На основе анализа делается вывод о целесообразности создания КСЗИ. Если информация не является конфиденциальной и легко может быть восстановлена,

то создавать СЗИ нет необходимости. Не имеет смысла также создавать КСЗИ в АС, если потеря целостности и конфиденциальности информации связана с незначительными потерями.

В этих случаях достаточно использовать штатные средства АС и, возможно, страхование от утраты информации.

При анализе информации определяются потоки конфиденциальной информации, элементы АС, в которых она обрабатывается и хранится. На этом этапе рассматриваются также вопросы разграничения доступа к информации отдельных пользователей и целых сегментов АС. На основе анализа информации определяются требования к её защищённости. Требования задаются путём присвоения определённого грифа конфиденциальности, установления правил разграничения доступа.

Очень важная исходная информация для построения КСЗИ получается в результате анализа защищаемой АС. Так как КСЗИ является подсистемой АС, то взаимодействие системы защиты с АС можно определить как внутреннее, а взаимодействие с внешней средой - как внешнее (рис. 1).

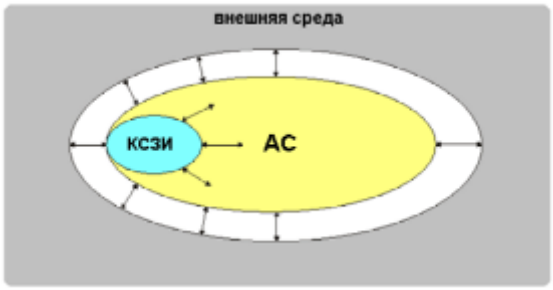

Рис. 1. Схема взаимодействия КСЗИ

Внутренние условия взаимодействия определяются архитектурой АС. При построении КСЗИ учитываются:

- географическое положение АС;
- тип АС (распределённая или сосредоточенная);
- структуры АС (техническая, программная, информационная и т. д.);
- производительность и надёжность элементов АС;
- типы используемых аппаратных и программных средств и режимы их работы;

- угрозы безопасности информации, которые порождаются внутри АС (отказы аппаратных и программных средств, алгоритмические ошибки и т. п.).

Учитываются следующие внешние условия:

- взаимодействие с внешними системами;
- случайные и преднамеренные угрозы.

Анализ угроз безопасности является одним из обязательных условий построения КСЗИ. По результатам проведённого анализа строится модель угроз безопасности информации в АС.

Модель угроз безопасности информации в АС содержит систематизированные данные о случайных и преднамеренных угрозах безопасности информации в конкретной АС. Систематизация данных модели предполагает наличие сведений обо всех возможных угрозах, их опасности, временных рамках действия, вероятности реализации. Часто модель угроз рассматривается как композиция модели злоумышленника и модели случайных угроз.

Модели представляются в виде таблиц, графов или на вербальном уровне. При построении модели злоумышленника используются два подхода:

1. модель ориентируется только на высококвалифицированного злоумышленника-профессионала, оснащённого всем необходимым и имеющего легальный доступ на всех рубежах защиты;
2. модель учитывает квалификацию злоумышленника, его оснащённость (возможности) и официальный статус в АС.

Первый подход проще реализуется и позволяет определить верхнюю границу преднамеренных угроз безопасности информации.

Второй подход отличается гибкостью и позволяет учитывать особенности АС в полной мере. Градация злоумышленников по их квалификации может быть различной. Например, может быть выделено три класса злоумышленников:

1. высококвалифицированный злоумышленник-профессионал;

2. квалифицированный злоумышленник-непрофессионал;

3. неквалифицированный злоумышленник-непрофессионал.

Класс злоумышленника, его оснащённость и статус на объекте определяют возможности злоумышленника по несанкционированному доступу к ресурсам АС.

Угрозы, связанные с непреднамеренными действиями, хорошо изучены, и большая часть их может быть формализована. Сюда следует отнести угрозы безопасности, которые связаны с конечной надёжностью технических систем. Угрозы, порождаемые стихией или человеком, формализовать сложнее. Но с другой стороны, по ним накоплен большой объем статистических данных. На основании этих данных можно прогнозировать проявление угроз этого класса.

Модель злоумышленника и модель случайных угроз позволяют получить полный спектр угроз и их характеристик. В совокупности с исходными данными, полученными в результате анализа информации, особенностей архитектуры проектируемой АС, модели угроз безопасности информации позволяют получить исходные данные для построения модели КСЗИ.

Классификация угроз информационной безопасности

Угрозы можно классифицировать по нескольким критериям:

- по аспекту информационной безопасности (доступность, целостность, конфиденциальность), против которого угрозы направлены в первую очередь;
- по компонентам информационных систем, на которые угрозы нацелены (данные, программы, аппаратура, поддерживающая инфраструктура);
- по способу осуществления (случайные/преднамеренные действия природного/техногенного характера);
- по расположению источника угроз (внутри/вне рассматриваемой ИС).

Необходимость классификации угроз ИБ АС обусловлена тем, что архитектура современных средств автоматизированной обработки информации, организационное, структурное и функциональное построение информационно-

вычислительных систем и сетей, технологии и условия автоматизированной обработки информации такие, что накапливаемая, хранимая и обрабатываемая информация подвержена случайным влияниям чрезвычайно большого числа факторов, в силу чего становится невозможным формализовать задачу описания полного множества угроз. Как следствие, для защищаемой системы определяют не полный перечень угроз, а перечень классов угроз.

Классификация всех возможных **угроз информационной безопасности** АС может быть проведена по ряду базовых признаков.

1. По природе возникновения.

1.1. Естественные угрозы-угрозы, вызванные воздействиями на АС и ее компоненты объективных физических процессов или стихийных природных явлений, независящих от человека.

1.2. Искусственные угрозы- угрозы информационной безопасности АС, вызванные деятельностью человека.

2. **По степени преднамеренности проявления.**

2.1. Угрозы случайного действия и/или угрозы, вызванные ошибками или халатностью персонала. Например:

- проявление ошибок программно-аппаратных средств АС;
- некомпетентное использование, настройка или неправомерное отключение средств защиты персоналом службы безопасности;
- неумышленные действия, приводящие к частичному или полному отказу системы или разрушению аппаратных, программных, информационных ресурсов системы (неумышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т. п.);
- неправомерное включение оборудования или изменение режимов работы устройств и программ;
- неумышленная порча носителей информации;
- пересылка данных по ошибочному адресу абонента (устройства);
- ввод ошибочных данных;
- неумышленное повреждение каналов связи.

2.2. Угрозы преднамеренного действия(например, угрозы действий злоумышленника для хищения информации).

3. **По непосредственному источнику угроз**.

3.1. Угрозы, непосредственным источником которых является природная среда(стихийные бедствия, магнитные бури, радиоактивное излучение и т.п.).

3.2. Угрозы, источником которых является человек:

- внедрение агентов в число персонала системы (в том числе, возможно, и в административную группу, отвечающую за безопасность);
- вербовка (путем подкупа, шантажа и т.п.) персонала или отдельных пользователей, имеющих определенные полномочия;
- угроза несанкционированного копирования секретных данных пользователем АС;
- разглашение, передача или утрата атрибутов разграничения доступа (паролей, ключей шифрования, идентификационных карточек, пропусков и т.п.).

3.3. Угрозы, непосредственным источником которых являются санкционированные программно-аппаратные средства:

- запуск технологических программ, способных при некомпетентном пользовании вызывать потерю работоспособности системы (зависания) или зацикливания) или необратимые изменения в системе (форматирование или реструктуризацию носителей информации, удаление данных и т. п.);
- возникновение отказа в работе операционной системы.

3.4. Угрозы, непосредственным источником которых являются несанкционированные программно-аппаратные средства:

- нелегальное внедрение и использование неучтенных программ (игровых, обучающих, технологических и др., не являющихся необходимыми для выполнения нарушителем своих служебных обязанностей) с последующим необоснованным расходованием ресурсов (загрузка процессора, захват оперативной памяти и памяти на внешних носителях);
- заражение компьютера вирусами с деструктивными функциями.

4. **По положению источника угроз**.

4.1. Угрозы, источник которых расположен вне контролируемой зоны территории (помещения), на которой находится АС:

- перехват побочных электромагнитных, акустических и других излучений устройств и линий связи, а также наводок активных излучений на вспомогательные технические средства, непосредственно не участвующие в обработке информации (телефонные линии, сети питания, отопления и т. п.);
- перехват данных, передаваемых по каналам связи, и их анализ с целью выяснения протоколов обмена, правил вхождения в связь и авторизации пользователя и последующих попыток их имитации для проникновения в систему;
- дистанционная фото- и видеосъемка.

4.2. Угрозы, источник которых расположен в пределах контролируемой зоны территории (помещения), на которой находится АС:

- хищение производственных отходов (распечаток, записей, списанных носителей информации и т.п.);
- отключение или вывод из строя подсистем обеспечения функционирования вычислительных систем (электропитания, охлаждения и вентиляции, линий связи и т.д.);
- применение подслушивающих устройств.

4.3. Угрозы, источник которых имеет доступ к периферийным устройства АС(терминалам).

4.4. Угрозы, источник которых расположен в АС:

- проектирование архитектуры системы и технологии обработки данных, разработка прикладных программ, которые представляют опасность для работоспособности системы и безопасности информации;
- некорректное использование ресурсов АС.

5. **По степени зависимости от активности АС**.

5.1. Угрозы, которые могут проявляться независимо от активности АС:

- вскрытие шифров криптозащиты информации;

- хищение носителей информации (магнитных дисков, лент, микросхем памяти, запоминающих устройств и компьютерных систем).

5.2. Угрозы, которые могут проявляться только в процессе автоматизированной обработки данных(например, угрозы выполнения и распространения программных вирусов).

6. **По степени воздействия на АС.**

6.1. Пассивные угрозы, которые при реализации ничего не меняют в структуре и содержании АС (угроза копирования секретных данных).

6.2. Активные угрозы, которые при воздействии вносят изменения в структуру и содержание АС:

- внедрение аппаратных спецвложений, программных "закладок" и "вирусов" ("троянских коней" и "жучков"), т.е. таких участков программ, которые не нужны для выполнения заявленных функций, но позволяют преодолеть систему защиты, скрытно и незаконно осуществить доступ к системным ресурсам с целью регистрации и передачи критической информации или дезорганизации функционирования системы;
- действия по дезорганизации функционирования системы (изменение режимов работы устройств или программ, забастовка, саботаж персонала, постановка мощных активных радиопомех на частотах работы устройств системы и т.п.);
- угроза умышленной модификации информации.

7. **По этапам доступа пользователей или программ к ресурсам АС.**

7.1. Угрозы, которые могут проявляться на этапе доступа к ресурсам АС(например, угрозы несанкционированного доступа в АС).

7.2. Угрозы, которые могут проявляться после разрешения доступа к ресурсам АС(например, угрозы несанкционированного или некорректного использования ресурсов АС).

8. **По способу доступа к ресурсам АС**.

8.1. Угрозы, направленные на использование прямого стандартного пути доступа к ресурсам АС:

- незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, подбором, имитацией интерфейса системы и т.д.) с последующей маскировкой под зарегистрированного пользователя ("маскарад");
- несанкционированное использование терминалов пользователей, имеющих уникальные физические характеристики, такие как номер рабочей станции в сети, физический адрес, адрес в системе связи, аппаратный блок кодирования и т.п.

8.2. Угрозы, направленные на использование скрытого нестандартного пути доступа к ресурсам АС:

- вход в систему в обход средств защиты (загрузка посторонней опера ционной системы со сменных магнитных носителей и т.п.);
- угроза несанкционированного доступа к ресурсам АС путем использования недокументированных возможностей ОС.

9. **По текущему месту расположения информации, хранимой и обрабатываемой в АС**.

9.1. Угрозы доступа к информации на внешних запоминающих устройства(например, угроза несанкционированного копирования секретной информации с жесткого диска).

9.2. Угрозы доступа к информации в оперативной памяти:

- чтение остаточной информации из оперативной памяти;
- чтение информации из областей оперативной памяти, используемых операционной системой (в том числе подсистемой защиты) или другими пользователями, в асинхронном режиме, используя недостатки мультизадачных АС и систем программирования;
- угроза доступа к системной области оперативной памяти со сторон прикладных программ.

9.3. Угрозы доступа к информации, циркулирующей в линиях связи:

- незаконное подключение к линиям связи с целью работы во время пауз в действиях законного пользователя от его имени с вводом ложных сообщений или модификацией передаваемых сообщений;
- незаконное подключение к линиям связи с целью прямой подмены законного пользователя путем его физического отключения после входа в систему и успешной аутентификации с последующим вводом дезинформации и навязыванием ложных сообщений;
- перехват всего потока данных с целью дальнейшего анализа не в реальном масштабе времени.

9.4. Угрозы доступа к информации, отображаемой на терминале или печатаемой на принтере(например, угроза записи отображаемой информации на скрытую видеокамеру). Вне зависимости от конкретных видов угроз или их проблемно-ориентированной классификации АС удовлетворяет потребности эксплуатирующих ее лиц, если обеспечиваются следующие свойства информации систем ее обработки.

В качестве основного критерия будем использовать первый (по аспекту ИБ), привлекая при необходимости остальные.

Угроза доступности (отказа служб) возникает всякий раз, когда в результате преднамеренных действий, предпринимаемых другим пользователем или злоумышленником, блокируется доступ к некоторому ресурсу вычислительной системы. Реально блокирование может быть постоянным - запрашиваемый ресурс никогда не будет получен, или оно может вызывать только задержку запрашиваемого ресурса, достаточно долгую для того, чтобы он стал бесполезным. В этих случаях говорят, что ресурс исчерпан.

Доступность информации – свойство системы (среды, средств и технологии обработки), в которой циркулирует информация, характеризующееся способностью обеспечивать своевременный беспрепятственный доступ субъектов к интересующей их информации и готовность соответствующих автоматизированных служб к обслуживанию

поступающих от субъектов запросов всегда, когда возникает в этом необходимость.

Самыми частыми и самыми опасными (с точки зрения размера ущерба) являются непреднамеренные ошибки штатных пользователей, операторов, системных администраторов и других лиц, обслуживающих ИС.

Иногда такие ошибки и являются собственно угрозами (неправильно введенные данные или ошибка в программе, вызвавшая крах системы), иногда они создают уязвимые места, которыми могут воспользоваться злоумышленники. По некоторым данным, до 65% потерь – следствие непреднамеренных ошибок.

Пожары и наводнения не приносят столько бед, сколько безграмотность и небрежность в работе. Самый радикальный способ борьбы с непреднамеренными ошибками – максимальная автоматизация и строгий контроль.

Другие угрозы доступности классифицируем по компонентам ИС, на которые нацелены угрозы:

- отказ пользователей;
- внутренний отказ информационной системы;
- отказ поддерживающей инфраструктуры.

Обычно применительно к пользователям рассматриваются следующие угрозы:

- нежелание работать с информационной системой;
- невозможность работать с системой в силу отсутствия соответствующей подготовки (недостаток компьютерной грамотности, неумение интерпретировать диагностические сообщения, неумение работать с документацией);
- невозможность работать с системой в силу отсутствия технической поддержки (неполнота документации, недостаток справочной информации и т.п.).

Основными источниками внутренних отказов являются:

- отступление от установленных правил эксплуатации;
- выход системы из штатного режима эксплуатации в силу случайных или преднамеренных действий пользователей или обслуживающего персонала (превышение расчетного числа запросов, чрезмерный объем обрабатываемой информации и т.п.);
- ошибки при (пере)конфигурировании системы;
- отказы программного и аппаратного обеспечения;
- разрушение данных;
- разрушение или повреждение аппаратуры.

По отношению к поддерживающей инфраструктуре рекомендуется рассматривать следующие угрозы:

- нарушение работы (случайное или умышленное) систем связи, электропитания, водо- и/или теплоснабжения, кондиционирования;
- разрушение или повреждение помещений;
- невозможность или нежелание обслуживающего персонала и/или пользователей выполнять свои обязанности (гражданские беспорядки, аварии на транспорте, террористический акт или его угроза, забастовка и т.п.).

Весьма опасны так называемые "обиженные" сотрудники – нынешние и бывшие (они стремятся нанести вред организации-"обидчику", например: испортить оборудование; встроить логическую бомбу, которая со временем разрушит программы и/или данные; удалить данные). Необходимо следить за тем, чтобы при увольнении сотрудника его права доступа (логического и физического) к информационным ресурсам аннулировались.

Опасны, разумеется, стихийные бедствия и события, воспринимаемые как стихийные бедствия,– пожары, наводнения, землетрясения, ураганы, по статистике, (среди которых самый опасный – перебой электропитания) приходится 13% потерь, нанесенных ИС.

Некоторые примеры угроз доступности.

Угрозы доступности могут выглядеть грубо – как повреждение или даже разрушение оборудования (в том числе носителей данных) и может вызываться

естественными причинами (чаще всего – грозами). Опасны протечки водопровода и отопительной системы, в сильную жару, ломаются кондиционеры, установленные в серверных залах, набитых дорогостоящим оборудованием.

Общеизвестно, что периодически необходимо производить резервное копирование данных. Однако даже если это предложение выполняется, резервные носители обычно хранят небрежно, не обеспечивая их защиту от вредного воздействия окружающей среды.

Перейдем теперь к программным атакам на доступность.

В качестве средства вывода системы из штатного режима эксплуатации может использоваться агрессивное потребление ресурсов (обычно – полосы пропускания сетей, вычислительных возможностей процессоров или ОЗУ). По расположению источника угрозы такое потребление подразделяется на локальное и удаленное. При просчетах в конфигурации системы локальная программа способна практически монополизировать процессор и/или физическую память, сведя скорость выполнения других программ к нулю.

Простейший пример удаленного потребления ресурсов – атака, получившая наименование "SYN-наводнение". Она представляет собой попытку переполнить таблицу "полуоткрытых" TCP-соединений сервера (установление соединений начинается, но не заканчивается), что приводит к затруднению установление новых соединений пользователей, то есть сервер блокируется.

По отношению к атаке "Papa Smurf" уязвимы сети, воспринимающие ping-пакеты с широковещательными адресами. Ответы на такие пакеты "съедают" полосу пропускания.

Удаленное потребление ресурсов в последнее время проявляется в особенно опасной форме – как скоординированные распределенные атаки, когда на сервер с множества разных адресов с максимальной скоростью направляются вполне легальные запросы на соединение и/или обслуживание.

Для выведения систем из штатного режима эксплуатации могут использоваться уязвимые места в виде программных и аппаратных ошибок. Например, известная ошибка в процессоре Pentium I дает возможность локальному пользователю путем выполнения определенной команды "подвесить" компьютер, так что помогает только аппаратный RESET.

Программа "Teardrop" удаленно "подвешивает" компьютеры, эксплуатируя ошибку в сборке фрагментированных IP-пакетов.

Угроза нарушения целостности включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую, в том числе и несанкционированное изменение информации при случайных ошибках программного или аппаратного обеспечения. Санкционированными изменениями являются те, которые сделаны уполномоченными лицами с обоснованной целью (например, периодическая запланированная коррекция некоторой базы данных).

Целостность информации - существование информации в неискаженном виде (неизменном по отношению к некоторому фиксированному ее состоянию). Обычно субъектов интересует обеспечение более широкого свойства – достоверности информации, которое складывается из адекватности (полноты и точности) отображения состояния предметной области и непосредственно целостности информации, т.е. ее неискаженности.

На втором месте по размерам ущерба стоят кражи и подлоги. По данным газеты USA Today, еще в 1992 году в результате подобных противоправных действий с использованием персональных компьютеров американским организациям был нанесен общий ущерб в размере 882 миллионов долларов. В наши дни ущерб от такого рода действий вырос многократно.

В большинстве случаев виновниками оказывались штатные сотрудники организаций, знакомые с режимом работы и мерами защиты, что подтверждает опасность внутренних угроз, хотя им уделяют меньшее внимание, чем внешним.

Существует различие между статической и динамической целостностью. С целью нарушения статической целостности злоумышленник может: ввести неверные данные; изменить данные.

Иногда изменяются содержательные данные, иногда – служебная информация. Показательный случай нарушения целостности имел место в 1996 году. Служащая Oracle (личный секретарь вице-президента) предъявила судебный иск, обвиняя президента корпорации в незаконном увольнении после того, как она отвергла его ухаживания. В доказательство своей правоты женщина привела электронное письмо, якобы отправленное ее начальником президенту. Содержание письма для нас сейчас не важно; важно время отправки. Дело в том, что вице-президент предъявил, в свою очередь, файл с регистрационной информацией компании сотовой связи, из которого явствовало, что в указанное время он разговаривал по мобильному телефону, находясь вдалеке от своего рабочего места. Таким образом, в суде состоялось противостояние "файл против файла". Очевидно, один из них был фальсифицирован или изменен, то есть была нарушена его целостность. Суд решил, что подделали электронное письмо (секретарша знала пароль вице-президента, поскольку ей было поручено его менять), и иск был отвергнут...

Угрозой целостности является не только фальсификация или изменение данных, но и отказ от совершенных действий. Если нет средств обеспечить "неотказуемость", компьютерные данные не могут рассматриваться в качестве доказательства.

Потенциально уязвимы с точки зрения нарушения целостностине толькоданные, но ипрограммы. Внедрение рассмотренного выше вредоносного ПО – пример подобного нарушения.

Угрозами динамической целостности являются нарушение атомарности транзакций, переупорядочение, кража, дублирование данных или внесение дополнительных сообщений (сетевых пакетов и т.п.). Соответствующие действия в сетевой среде называются активным прослушиванием.

Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. Иногда, в связи с угрозой нарушения конфиденциальности, используется термин "утечка".

Конфиденциальность информации– субъективно определяемая (приписываемая) характеристика (свойство) информации, указывающая н необходимость введения ограничений на круг субъектов, имеющих доступ к данной информации, и обеспечиваемая способностью системы (среды) сохранять указанную информацию в тайне от субъектов, не имеющих полномочий доступа к ней. Объективные предпосылки подобного ограничения доступности информации для одних субъектов заключены в необходимости защиты их законных интересов от других субъектов информационных отношений.

Конфиденциальную информацию можно разделить на предметную и служебную. Служебная информация (например, пароли пользователей) не относится к определенной предметной области, в информационной системе она играет техническую роль, но ее раскрытие особенно опасно, поскольку оно чревато получением несанкционированного доступа ко всей информации, в том числе предметной.

Многим людям приходится выступать в качестве пользователей не одной, а целого ряда систем (информационных сервисов). Если для доступа к таким системам используются многоразовые пароли или иная конфиденциальная информация, то наверняка эти данные будут храниться не только в голове, но и в записной книжке или на листках бумаги, которые пользователь часто оставляет на рабочем столе, а то и попросту теряет. И дело здесь не в неорганизованности людей, а в изначальной непригодности парольной схемы. Невозможно помнить много разных паролей; рекомендации по их регулярной смене только усугубляют положение, заставляя применять несложные схемы чередования или стараться свести дело к двум-трем легко запоминаемым паролям.

Описанный класс уязвимых мест можно назвать размещением конфиденциальных данных в среде, где им не обеспечена необходимая защита. Угроза же состоит в том, что кто-то не откажется узнать секреты, которые сами просятся в руки. Помимо паролей, хранящихся в записных книжках пользователей, в этот класс попадает передача конфиденциальных данных в открытом виде (в разговоре, в письме, по сети), которая делает возможным перехват данных. Для атаки могут использоваться разные технические средства (подслушивание или прослушивание разговоров, пассивное прослушивание сети и т.п.), но идея одна – получить доступ к данным в тот момент, когда они наименее защищены.

Угрозу перехвата данных следует принимать во внимание не только при начальном конфигурировании ИС, но и, что очень важно, при всех изменениях. Еще один пример изменения, о котором часто забывают, – хранение данных на резервных носителях. Для защиты данных на основных носителях применяются развитые системы управления доступом; копии же нередко просто лежат в шкафах и получить доступ к ним могут многие.

Перехват данных – очень серьезная угроза, и если конфиденциальность действительно является критичной, а данные передаются по многим каналам, их защита может оказаться весьма сложной и дорогостоящей. Технические средства перехвата хорошо проработаны, доступны, просты в эксплуатации, а установить их, например на кабельную сеть, может кто угодно, так что эту угрозу нужно принимать во внимание по отношению не только к внешним, но и к внутренним коммуникациям.

Кражи оборудования являются угрозой не только для резервных носителей, но и для компьютеров, особенно портативных. Часто ноутбуки оставляют без присмотра на работе или в автомобиле, иногда просто теряют.

Опасной нетехнической угрозой конфиденциальности являются методы морально-психологического воздействия, такие какмаскарад– выполнение действий под видом лица, обладающего полномочиями для доступа к данным

К неприятным угрозам, от которых трудно защищаться, можно отнести злоупотребление полномочиями. На многих типах систем привилегированный пользователь (например системный администратор) способен прочитать любой (незашифрованный) файл, получить доступ к почте любого пользователя и т.д. Другой пример – нанесение ущерба при сервисном обслуживании. Обычно сервисный инженер получает неограниченный доступ к оборудованию и имеет возможность действовать в обход программных защитных механизмов.

Таковы основные угрозы, которые наносят наибольший ущерб субъектам информационных отношений.

На современном этапе развития информационных технологий под системы или функции защиты являются неотъемлемой частью комплекса по обработке информации. Информация не представляется "в чистом виде", на пути к ней имеется хотя бы какая-нибудь система защиты, и поэтому чтобы угрожать, атакующая сторона должна преодолеть эту систему. Однако не существует абсолютно стойкой системы защиты, вопрос лишь во времени и средствах, требующихся на ее преодоление. Исходя из данных условий, примем следующую модель: **защита информационной системы считается преодоленной, если в ходе ее исследования определены все уязвимости системы**. Поскольку преодоление защиты также представляет собой угрозу, для защищенных систем будем рассматривать ее четвертый вид - **угрозу раскрытия параметров АС**, включающей в себя систему защиты. С точки зрения практики любое проводимое мероприятие предваряется этапом разведки, в ходе которого определяются основные параметры системы, её характеристики, в результате чего уточняется поставленная задача и выбираются оптимальные технические средства.

Угрозу раскрытия можно рассматривать как опосредованную. Последствия ее реализации не причиняют какой-либо ущерб обрабатываемой информации, но дают возможность реализоваться первичным или непосредственным угрозам, перечисленным выше.

Практическое занятие №4. Этапы доступа к ресурсам вычислительной системы; использование простого пароля; использование динамически изменяющегося пароля; взаимная проверка подлинности и другие случаи опознания; способы разграничения доступа к компьютерным ресурсам; разграничение доступа по спискам.

***Цель работы*:** ознакомиться с этапами доступа к ресурсам вычислительной системы.

ВЫЧИСЛИТЕЛЬНАЯ СИСТЕМА И ЕЕ РЕСУРСЫ

Вычислительная система (ВС) - это взаимосвязанная совокупность аппаратных средств вычислительной техники и программного обеспечения, предназначенная для обработки информации.

Иногда под ВС понимают совокупность технических средств ЭВМ, в которую входит не менее двух процессоров, связанных общностью управления и использования общесистемных ресурсов (память, периферийные устройства, программное обеспечение и т.п.).

Ресурсы вычислительной системы

К ресурсам вычислительной системы относят такие средства вычислительной системы, которые могут быть выделены процессу обработки данных на определенный квант времени. Основными ресурсами ВС являются процессоры, области оперативной памяти, наборы данных, периферийные устройства, программы.

В зависимости от ряда признаков различают следующие вычислительные системы (ВС):

- **однопрограммные и многопрограммные** (в зависимости от количества программ, одновременно находящихся в оперативной памяти);
- **индивидуального и коллективного пользования** (в зависимости от числа пользователей, которые одновременно могут использовать ресурсы ВС);

- **с пакетной обработкой и разделением времени** (в зависимости от организации и обработки заданий);
- **однопроцессорные, многопроцессорные и многомашинные** (в зависимости от числа процессоров);
- **сосредоточенные, распределенные** (вычислительные сети) и ВС с теледоступом (в зависимости от территориального расположения и взаимодействия технических средств);
- **работающие или не работающие в режиме реального времени** (в зависимости от соотношения скоростей поступления задач в ВС и их решения);
- **универсальные, специализированные и проблемно-ориентированные** (в зависимости от назначения).

РЕЖИМЫ РАБОТЫ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ

1. Мультипрограммирование

2. Режим реального времени

3. Однопрограммный режим работы вычислительной системы (ВС)

4. Мультипрограммный режим работы вычислительной системы (ВС)

5. Режим пакетной обработки

6. Режим коллективного доступа

Мультипрограммирование - это режим обработки данных, при котором ресурсы вычислительной системы предоставляются каждому процессу из группы процессов обработки данных, находящихся в ВС, на интервалы времени, длительность и очередность предоставления которых определяется управляющей программой этой системы с целью обеспечения одновременной работы в интерактивном режиме.

Режим реального времени - режим обработки данных, при котором обеспечивается взаимодействие вычислительной системы с внешними по отношению к ней процессами в темпе, соизмеримом со скоростью протекания этих процессов.

Этот режим обработки данных широко используется в системах управления и информационно-поисковых системах.

Аппаратные средства ЭВМ совместно с программным обеспечением образуют ВС. В зависимости от класса ЭВМ и вида операционной системы ВС могут работать в режимах **однопрограммном и мультипрограммном**.

В **однопрограммном режиме** работы в памяти ЭВМ находится и выполняется только одна программ. Такой режим обычно характерен для микро-ЭВМ и персональных ЭВМ, то есть для ЭВМ индивидуального пользования.

В **мультипрограммном (многопрограммном) режиме** работы в памяти ЭВМ находится несколько программ, которые выполняются частично или полностью между переходами процессора от одной задачи к другой в зависимости от ситуации, складывающейся в системе.

В мультипрограммном режиме более эффективно используются машинное время и оперативная память, так как при возникновении каких-либо ситуаций в выполняемой задаче, требующих перехода процессора в режим ожидания, процессор переключается на другую задачу и выполняет ее до тех пор, пока в ней не возникает подобная ситуация, и т.д.

При реализации мультипрограммного режима требуется определять очередность переключения задач и выбирать моменты переключения, чтобы эффективность использования машинного времени и памяти была максимальной.

Мультипрограммный режим обеспечивается аппаратными средствами ЭВМ и средствами операционной системы. Он характерен для сложных ЭВМ, где стоимость машинного времени значительно выше, чем у микро-ЭВМ. Разработаны также мультипрограммные ОС, позволяющие одновременно следить за решением нескольких задач и повышать эффективность работы пользователя.

Режим пакетной обработки

В зависимости от того, в каком порядке при мультипрограммном режиме выполняются программы пользователей, различают режимы пакетной обработки задач и коллективного доступа.

В режиме пакетной обработки задачи выстраиваются в одну или несколько очередей и последовательно выбираются для их выполнения.

Режим коллективного доступа

В режиме коллективного доступа каждый пользователь ставит свою задачу на выполнение в любой момент времени, то есть для каждого пользователя в такой ВС реализуется режим индивидуального пользования. Это осуществляется обычно с помощью квантования машинного времени, когда каждой задаче, находящейся в оперативной памяти ЭВМ, выделяется квант времени. После окончания кванта времени процессор переключается на другую задачу или продолжает выполнение прерванной в зависимости от ситуации в ВС. Вычислительные системы, обеспечивающие коллективный доступ пользователей с квантованием машинного времени, называют ВС с разделением времени.

Объектно-ориентированное программирование

Объектно-ориентированное программирование представляет собой метод программирования, который весьма близко напоминает наше поведение. Оно является естественной эволюцией более ранних нововведений в разработке языков программирования. Объектно-ориентированное программирование является более структурным, чем все предыдущие разработки, касающиеся структурного программирования. Оно также является более модульным и более абстрактным, чем предыдущие попытки абстрагирования данных и переноса деталей программирования на внутренний уровень. Объектно-ориентированный язык программирования характеризуется тремя основными свойствами:

1. **Инкапсуляция**. Комбинирование записей с процедурами и функциями, манипулирующими полями этих записей, формирует новый тип данных - объект.

2. **Наследование.** Определение объекта и его дальнейшее использование для построения иерархии порожденных объектов с возможностью для каждого порожденного объекта, относящегося к иерархии, доступа к коду и данным всех порождающих объектов.

3. **Полиморфизм**. Присваивание действию одного имени, которое затем совместно используется вниз и вверх по иерархии объектов, причем каждый объект иерархии выполняет это действие способом, именно ему подходящим.

В разработке программного обеспечения, **стадии разработки программного обеспечения** используются для описания степени готовности программного продукта. Также стадия разработки может отражать количество реализованных функций, запланированных для определённой версии программы. Стадии либо могут быть официально объявлены и регламентируются разработчиками, либо иногда этот термин используется неофициально для описания состояния продукта.

Пре-альфа - Начальная стадия разработки — Период времени со старта разработки до выхода стадии Альфа (или до любой другой, если стадии Альфа нет). Также так называются программы, не вышедшие еще в стадию альфа или бета, но прошедшие стадию разработки, для первичной оценки функциональных возможностей в действии. В отличие от альфа и бета версий, пре-альфа может включать в себя не весь спектр функциональных возможностей программы. В этом случае, подразумеваются все действия выполняемые во время проектирования и разработки программы вплоть до тестирования. К таким действиям относятся — разработка дизайна, анализ требований, собственно разработка приложения, а также отладка отдельных модулей.

Альфа-тестирование - Внутреннее тестирование — Стадия начала тестирования программы в целом специалистами-тестерами, обычно не разработчиками программного продукта, но, как правило, внутри организации

или сообществе разрабатывающих продукт. Также это может быть стадия добавления новых функциональных возможностей. Программы на данной стадии могут применяться только для ознакомления с будущими возможностями.

Бета-тестирование - Публичное тестирование — Стадия активного бета-тестирования и отладки, прошедшей альфа-тестирование (если таковое было). Программы этого уровня могут быть использованы другими разработчиками программного обеспечения для испытания совместимости. Тем не менее программы этого этапа могут содержать достаточно большое количество ошибок.

Релиз-кандидат или RC (англ. release candidate) — стадия-кандидат на то, чтобы стать стабильной. Программы этой стадии прошли комплексное тестирование, благодаря чему были исправлены все найденные критические ошибки. Но в то же время, существует вероятность выявления ещё некоторого числа ошибок, не замеченных при тестировании.

Релиз или RTM (англ. release to manufacturing) — стабильная версия программы, прошедшая все предыдущие стадии, в которых исправлены основные ошибки, и готовая к применению.

Парольная защита компьютера

Для обеспечения безопасности ПК необходимо обеспечить защиту отдельных файлов и папок и предпринять действия по физической защите компьютера. Если на компьютере имеются конфиденциальные сведения, они должны храниться в безопасном месте.

Простейшими способами защиты компьютера являются его блокировка на время отсутствия на рабочем месте и настройка заставки, защищенной паролем.

Блокировка пк

Нажмите комбинацию клавиш **CTRL+ALT+DELETE**.

1. В появившемся окне нажмите кнопку **Блокировка**. В результате отобразится диалоговое окно**Блокировка компьютера**. Теперь компьютер

заблокирован, что не позволит никому, кроме администратора и пользователя, заблокировавшего компьютер, войти в систему и открывать файлы и программы.

2. Для разблокирования ПК необходимо вновь нажать комбинацию клавиш **CTRL+ALT+DEL**, ввести пароль, а затем нажать кнопку**OK**.

Защита файлов с помощью пароля экранной заставки

1. Щелчком правой кнопки мыши по свободному месту Рабочего стола откройте контекстное меню и выберите команду **Свойства**.

2. В открывшемся диалоговом окне **Свойства: Экран**перейдите на вкладку**Заставка**.

3. В раскрывающемся списке **Заставка**выберите какую-либо заставку.

4. Включите флажок **Защита паролем**и нажмите кнопку**Применить**.

При установке флажка **Защита паролем**работа ПК блокируется при активизации заставки. Для разблокирования ПК при возобновлении работы необходимо ввести пароль, который совпадает с паролем текущего пользователя.

Использование паролей

Эффективным средством защиты, используемым для управления входом в систему по учетным записям пользователей, а также организации доступа к компьютерам и ресурсам является пароль.

Пароль– это уникальный набор разрешенных символов, который должен быть введен пользователем для проверки его учетного имени и получения доступа к ресурсам ПК.

Для организации надежной защиты ПК пароль должен выбираться, исходя из следующих требований:

1) иметь длину не менее семи символов (наиболее надежные пароли состоят из 7 - 14 символов;

2) содержать символы каждой из трех следующих групп:

- буквы (прописные и строчные) A, B, C,...; a, b, c,...; А, Б, В,…; а, б, в,…

- цифры 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
- специальные символы ` ~ ! @ # $ % ^ & * ( ) _ + - = { } | [ ] \ : " ; ' < > ? , . /

3) содержать хотя бы один специальный символ на позициях со второй по шестую;

4) существенно отличаться от ранее использованных паролей;

5) не содержать фамилии или имени пользователя;

6) не являться распространенным словом или именем.

Следует помнить, что программное обеспечение для взлома паролей использует один из трех подходов: *угадывание*, *подбор вариантов* и *автоматический перебор всех возможных комбинаций символов*. Имея достаточное количество времени можно взломать любой пароль методом автоматического перебора.

В Windows XP пароль учетной записи пользователя может содержать до 127 символов, однако если компьютер используется в локальной сети, где есть ПК с ОС Windows 95/98, поддерживающих пароли длиной до 14 знаков, использовать пароли большей длины нельзя.

Для обеспечения безопасности необходимо быть аккуратным при использовании паролей. Следует руководствоваться следующими рекомендациями:

- никогда не записывайте свой пароль в каком-либо обозримом месте;
- не сообщайте пароль никому;
- не используйте сетевой пароль для других целей;
- используйте разные пароли для входа в сеть и учетной записи администратора на компьютере;
- изменяйте свой пароль каждые 60-90 дней;
- измените пароль немедленно, если возникнут подозрения, что он был раскрыт;

- будьте осторожны при сохранении пароля на компьютере. В некоторых диалоговых окнах имеется возможность сохранить пароль. Никогда не устанавливайте этот параметр.

Программные средства и способы защиты компьютерной информации

Защита от несанкционированного доступа к ресурсам компьютера: пароли, разграничение доступа.

Данные меры защиты предусматривают защиту доступа к дисковым накопителям информации, к клавиатуре и дисплею компьютера.

Защита жёстких и гибких магнитных дисков предусматривает:

- защиту от любого несанкционированного доступа к диску;
- разграничение доступа пользователей к файлам дисков;
- контроль обращения к диску и проверка целостности защиты информации диска;
- периодические проверки наличия вирусов на диске;
- стирание в файлах остатков закрытой информации.

Защита дисков от несанкционированного доступа к ним осуществляется с помощью паролей. Пароль – это набор символов, который соответствует определенному объекту идентификации. Пароли для защиты дисков делятся:

- по типу объектов идентификации: пароли пользователей, ресурсов и файлов;
- по типу символов: цифровые, буквенные, смешанные;
- по способу ввода в компьютер: с клавиатуры, с помощью мыши, со специальной ключевой дискеты;
- по срокам применения: с неограниченным сроком, периодически сменяемые, разовые;
- по длине: фиксированной и переменной длины.

Чем больше длина пароля и меньше срок его использования, тем выше уровень защиты соответствующего диска.

Парольная защита дисков осуществляется, специальными компьютерными программами, а также утилитой «Пароли» в Панели управления операционной системы Windows98.

Более детальная процедура доступа к файлам организуется на основе таблиц разграничения доступа пользователей к файлам по назначению файлов, а также по характеру работы пользователей (создание, чтение, редактирование, удаление файлов и другие). Разработаны специальные программы, предоставляющие пользователям только те файлы и те возможности работы с ними, которые указаны в таблице разграничения доступа. Разработан также ряд специальных программ, которые фиксируют даты и время обращения к диску, а также случаи «взлома» защиты диска, несанкционированной работы с ним.

Необходимость стирания в файлах остатков закрытой информации вызвана тем, что, во-первых, при удалении файла командами MS DOS, Norton Commander или Windowsстирается только имя файла, а не сама информация на диске; во-вторых объём данных любого файла обычно меньше отведенного для него пространство на диске, поэтому в конце (в «хвосте») нового файла могут сохраниться остатки закрытой информации от предыдущего файла. Операцию стирания «хвостов» файлов осуществляет, например, утилита Wipe Info из пакета Norton Utilities.

Защита клавиатуры и дисплея применяется, когда пользователь отлучается с рабочего места на короткое время. Например, утилита Diskreet из пакета Norton Utilities блокирует клавиатуру и гасит экран при ее вводе. Ограничения снимаются введением пароля, т.е. когда пользователь возвращается к работе в предположении, что никто кроме него этот пароль не знает.

Защита дисков от копирования

Основное направление действий по защите от копирования дисков относится к дискетам, т.е. гибким дискам, так как такие диски легче похитить и скопировать. Применяются следующие способы защиты дискет от копирования:

- парольная защита дискеты, при которой без ввода пароля дискета не копируется;
- привязка информации к определенной дискете: фирме и типу дискеты, нестандартному способу форматирования дискеты, нанесению уникальных признаков на дискету, связанных с введением заранее помеченных «сбойных» участков;
- привязка информации к определенному компьютеру, например, к его тактовой частоте, параметрам накопителей, дисплея, принтера и других устройств.

В соответствии с задачами защиты от копирования имеется много программ защиты, в том числе, обеспечивающих многоуровневую систему защиты дискет от копирования.

Жесткие диски компьютеров чаще всего защищают от копирования с помощью использования паролей.

Разграничение доступа

После выполнения идентификации и аутентификации необходимо установить полномочия (совокупность прав) субъекта для последующего контроля санкционированного использования вычислительных ресурсов, доступных в АС. Такой процесс называется разграничением (логическим управлением) доступа.

Обычно полномочия субъекта представляются: списком ресурсов, доступных пользователю, и правами по доступу к каждому ресурсу из списка. В качестве вычислительных ресурсов могут быть программы, информация, логические устройства, объем памяти, время процессора, приоритет и т. д.

Обычно выделяют следующие методы разграничения доступа:

- разграничение доступа по спискам;
- использование матрицы установления полномочий;
- по уровням секретности и категориям;
- парольное разграничение доступа.

При разграничении доступа по спискам задаются соответствия:

- каждому пользователю – список ресурсов и прав доступа к ним или
- каждому ресурсу – список пользователей и их прав доступа к данному ресурсу.

Списки позволяют установить права с точностью до пользователя. Здесь нетрудно добавить права или явным образом запретить доступ. Списки используются в большинстве ОС и СУБД.

Использование матрицы установления полномочий подразумевает применение матрицы доступа (таблицы полномочий). В указанной матрице (см. таблицу 2.7) строками являются идентификаторы субъектов, имеющих доступ в АС, а столбцами – объекты (информационные ресурсы) АС. Каждый элемент матрицы может содержать имя и размер предоставляемого ресурса, право доступа (чтение, запись и др.), ссылку на другую информационную структуру, уточняющую права доступа, ссылку на программу, управляющую правами доступа и др.

Таблица 2.7 - Фрагмент матрицы установления полномочий

Субъект	Каталог d:\Heap	Программа prty	Принтер
Пользователь 1	cdrw	e	w
Пользователь 2	r		w с 9:00 до 17:00

c – создание, d – удаление, r – чтение, w – запись, e – выполнение.

Данный метод предоставляет более унифицированный и удобный подход, т. к. вся информация о полномочиях хранится в виде единой таблицы, а не в виде разнотипных списков. Недостатками матрицы являются ее возможная громоздкость и не совсем оптимальное использование ресурсов (большинство клеток – пустые).

Разграничения доступа по уровням секретности и категориям состоят в том, что ресурсы АС разделяются в соответствии с уровнями секретности или категорий.

При разграничении по уровню секретности выделяют несколько уровней, например,: общий доступ, конфиденциально, секретно, совершенно секретно. Полномочия каждого пользователя задаются в соответствии с максимальным уровнем секретности, к которому он допущен. Пользователь имеет доступ ко всем данным, имеющим уровень (гриф) секретности не выше, чем он имеет.

При разграничении по категориям задается и контролируется ранг категории, соответствующей пользователю. Соответственно, все ресурсы АС декомпозируют по уровню важности, причем определенному уровню соответствует некоторый ранг персонала (типа: руководитель, администратор, пользователь).

Парольное разграничение, очевидно, представляет использование методов доступа субъектов к объектам по паролю. При этом используются все методы парольной защиты. Очевидно, что постоянное использование паролей создает неудобства пользователям и временные задержки. Поэтому указанные методы используют в исключительных ситуациях.

На практике обычно сочетают различные методы разграничения доступа. Например, первые три метода усиливают парольной защитой.

В завершении подраздела заметим, что руководящие документы могут регламентировать два вида (принципа) разграничения доступа:

- дискретное управление доступом;
- мандатное управление доступом.

Дискретное управление доступом представляет собой разграничение доступа между поименованными субъектами и поименованными объектами. Субъект с определенным правом доступа может передать это право любому другому субъекту. Данный вид организуется на базе методов разграничения по спискам или с помощью матрицы.

Мандатное управление доступом регламентирует разграничение доступа субъектов к объектам, основанное на характеризуемой меткой конфиденциальности информации, содержащейся в объектах, и официальном разрешении (допуске) субъектов обращаться к информации

такого уровня конфиденциальности. Иначе, для реализации мандатного управления доступом каждому субъекту и каждому объекту присваивают классификационные метки, отражающие их место в соответствующей иерархии. С помощью этих меток субъектам и объектам должны быть назначены классификационные уровни, являющиеся комбинациями уровня иерархической классификации и иерархических категорий. Данные метки должны служить основой мандатного принципа разграничения доступа. Ясно, что методы разграничения доступа по уровням секретности и категориям являются примерами мандатного управления доступом.