Учреждение образования
«Минский государственный колледж электроники»

УТВЕРЖДЕНО
Заместитель директора
по учебной работе
_____Е.В.Филипцова
«____»_____20__ г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС
ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

Защита компьютерной информации

(название учебной дисциплины)

ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ

для специальности (направления специальности) 2-40 01 01 «Программное обеспечение информационных технологий»

(код и наименование специальности (направления специальности, специализации)

Составители: Артемьева Е.А.

Рассмотрено на заседании цикловой комиссии по специальности 2-40 01 01 «Программное обеспечение информационных технологий»

Протокол № _____
«___»_____20__г.

Председатель цикловой комиссии Шавейко А.А./_____

Содержание

1. Введение

Информация имеет первостепенное значение. Современное общество называют информационным.

В 1972 году американский специалист в области связи и вычислительной техники Роберт Фано говорил: «Стремление сохранить тайну предприятий и отдельных лиц – не единственный повод для поиска надежных средств обеспечения неприкосновенности информации, хранимой в вычислительных системах, а также для поиска средств контроля над ее применением. Такие средства требуются также для выполнения договорных обязательств, заключаемых между создателями программного обеспечения и банков данных, с одной стороны, и потребителями этой продукции, с другой. Следует иметь в виду и то, что бесконтрольный сбор, хранение и распределение информации неизбежно сопровождается «загрязнением» информационной среды – явлением, которое уже начало приводить если не к серьезным, то, во всяком случае, к тревожным последствиям. Наконец, соображения общественной безопасности диктуют необходимость надежного контроля над информацией, способной оказаться источником угрожающего положения, скажем вследствие распространения среди населения панических настроений либо вследствие потворствования незаконным действиям»

Проблема безопасности информационных технологий (ИТ) возникла на пересечении двух активно развивающихся направлений – безопасности технологий и информатизации.

Обеспечение собственной безопасности – задача первостепенной важности для любой системы независимо от ее сложности и назначения, будь то биологический организм или система обработки информации. Однако когда средства нападения имеют форму информационных воздействий, необходимо разрабатывать и применять совершенно новые технологии, методы зашиты.

Научные и технические предпосылки кризисной ситуации.

Современные компьютеры приобрели гигантскую вычислительную мощь, но одновременно с этим стали и проще в эксплуатации.

Все большее количество новых (и неквалифицированных) людей получает доступ к компьютерам, что приводит к снижению средней квалификации пользователей. Большинство пользователей имеют личные компьютеры и осуществляют их администрирование самостоятельно. Они не в состоянии постоянно поддерживать безопасность своих систем на должном уровне, т.к. это требует соответствующих знаний, времени и средств. Распространение сетевых технологий объединило отдельные машины в локальные сети, совместные использующие общие ресурсы, а применение технологий клиент-сервер и кластеризации преобразовало такие сети в распределенные вычислительные среды.

Безопасность сети определяется защищенностью всех входящих в нее компьютеров и сетевого оборудования и достаточно нарушить работу только одного компьютера, чтобы скомпрометировать всю сеть.

Если компьютер, который является объектом атаки, подключен к глобальной вычислительной сети (Internet), то независимо от характера обрабатываемой в нем информации то не имеет значения, где он находится – в соседней комнате или на другом континенте.

Бурное развитие программного обеспечения.

В настоящее время большинство операционных систем не отвечает требованиям безопасности, хотя в последнее время и осуществляют определенные усилия в этом направлении. Существует огромное количество различных недокументированных возможностей, обеспечивающих реализацию намеренных злоумышленных действий.

Развитие гибких и мобильных технологий привело к тому, что практически исчезает грань между обрабатываемыми данными и исполняемыми программами за счет появления и широкого распространения виртуальных машин и интерпретаторов. Теперь любое развитое приложение не просто обрабатывает данные, а интерпретирует интегрированные в них инструкции специальных языков программирования, т.е. по сути дела является отдельной машиной с привычной фон-неймановской архитектурой, для которых можно создавать средства нападения. Это увеличивает возможности злоумышленников и затрудняет задачу зашиты таких систем, т.к. наличие «вложенных» систем требует и реализации защиты для каждого уровня.

Несоответствие бурного развития средств обработки информации и медленного процесса разработки теории информационной безопасности привело к разрыву между теоретическими

моделями, оперирующими абстрактными понятиями и реальными категориями современных информационных технологий. Кроме того, многие средства защиты (например, средства борьбы с компьютерными вирусами) и системы защиты корпоративных систем на данный момент вообще не имеют системной научной базы. Такое положение является следствием отсутствия общей теории защиты информации, комплексных моделей безопасности обработки информации, отсутствие средств, позволяющих эффективно промоделировать адекватность тех или иных решений в области безопасности. Сегодня нет даже общепринятой терминологии, адекватно воспринимаемой всеми специалистами в области безопасности.

Необходимость создания глобального информационного пространства и обеспечение безопасности протекающих в нем процессов потребовала разработки международных стандартов, следование которым может обеспечить необходимый уровень гарантий обеспечения ИБ. Причем в современных условиях важным является не только стандартизация требований безопасности, но и обоснование их применения, а также методов подтверждения адекватности реализованных средств защиты и корректности самой реализации.

Перед разработчиками современных ИС стоят следующие задачи:

–Обеспечение безопасности новых типов информационных ресурсов. Это означает, что системы защиты должны обеспечивать безопасность не отдельных документов, файлов или сообщений, а решать задачи ИБ на уровне информационных ресурсов (Например, гипертекст, мультимедиа).

Гипертекст – информационный массив, на котором заданы и автоматически поддерживаются ассоциативные и смысловые связи между выделенными элементами, понятиями, терминами или разделами.

Мультимедиа – комплексное представление информации – вывод данных в текстовом, графическом, видео-, аудио-, мультипликационном видах.

–Организация доверенного взаимодействия сторон.

–Защита от автоматических средств нападения – разрушающих программных средств (РПС) т.е. компьютерных вирусов, «троянских коней» программных закладок. Средства разграничения доступа не решают в полной мере этой проблемы.

–Интеграция защиты информации в процессе автоматизации ее обработки в качестве обязательного элемента. Это означает, что средства безопасности не должны вступать в конфликт с существующими приложениями и сложившимися технологиями обработки информации, а напротив, должны стать неотъемлемой частью этих средств и технологий.

Понятие «защищенная система».

Защищенная система обработки информации для определенных условий эксплуатации обеспечивает безопасность (конфиденциальность и целостность) обрабатываемой информации и поддерживает свою работоспособность в условиях воздействия на нее заданного множества угроз.

Защищенная система должна обладать следующими свойствами.

1. Она должна автоматизировать процесс обработки конфиденциальной информации, включая все аспекты этого процесса, связанные с обеспечением безопасности.
2. Успешно и эффективно противостоять угрозам безопасности.
3. Соответствовать требованиям и критериям стандартов информационной безопасности. Наличие общепринятых стандартов позволяет согласовать подходы различных участников процесса создания защищенных систем (требования потребителей, технологии и методы производителей, критерии независимой экспертизы).

Информационная безопасность, актуальность ее обеспечения

Общее содержание проблемы информационной безопасности

Безопасность – это такое состояние рассматриваемой системы, при котором она с одной стороны, способна противостоять дестабилизирующему воздействию внешних и внутренних угроз, а с другой – ее наличие и функционирование не создает угроз для элементов самой системы и внешней среды.

Меры безопасности системы:

– с точки зрения способности противостоять дестабилизирующему воздействию внешних и внутренних угроз. Степень (уровень) сохранения системой своей структуры, технологии и эффективности функционирования под воздействием дестабилизирующих факторов;

– с точки зрения отсутствия угроз для элементов системы и внешней среды. Степень (уровень) возможности (или отсутствия возможности) появления таких дестабилизирующих факторов, которые могут представить угрозу элементам самой системы или внешней среде.

Информация, как непременный компонент любой организованной системы, с одной стороны, легко уязвима (т.е. весьма доступна для дестабилизирующего воздействия большого числа разноплановых угроз), а с другой сама может быть источником большого числа разноплановых угроз как для элементов самой системы, так и для внешней среды.

Обеспечение информационной безопасности может быть достигнуто лишь при взаимоувязанном решении трех составляющих проблем:

– защиты находящейся в системе информации от дестабилизирующего воздействия внешних и внутренних угроз;

– защиты элементов системы от дестабилизирующего воздействия внешних и внутренних информационных угроз;

– защиты внешней среды от информационных угроз со стороны рассматриваемой системы.

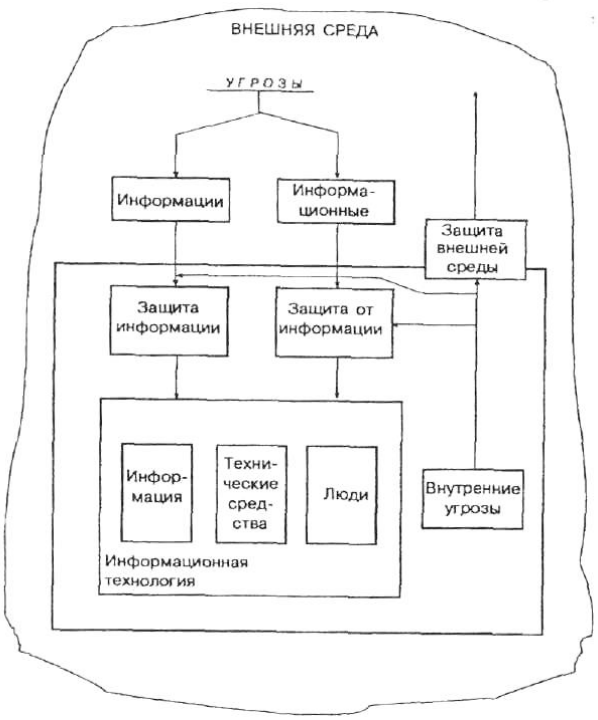


Рис. 1.1. Общая схема обеспечения информационной безопасности

Защита от информации заключается в использовании специальных методов и средств в целях предупреждения или нейтрализации негативного воздействия на элементы рассматриваемой системы (людей и технических комплексов) информации как имеющейся (генерируемой, хранимой, обрабатываемой и используемой) внутри системы, так и поступающей из внешней среды (защита системы от информации), а также предупреждение негативного воздействия выходной информации системы на элементы внешней среды (информационная экология).

Информация и информационные отношения. Субъекты информационных отношений

Информация – это сведения о фактах, событиях, процессах и явлениях, о состоянии объектов (их свойствах, характеристиках) в некоторой предметной области, используемые (необходимые) для оптимизации принимаемых решений в процессе управления данными объектами.

Отношения между субъектами будем называть информационными отношениями, а самих участвующих в них субъектов - субъектами информационных отношений.

Автоматизированная система обработки информации (АС) – организационно-техническая система, представляющая собой совокупность следующих взаимосвязанных компонентов:

- технических средств обработки и передачи данных (средств вычислительной техники и связи);
- методов и алгоритмов обработки в виде соответствующего программною обеспечения;
- информации (массивов, наборов, баз данных) на различных носителях;
- персонала и пользователей системы, объединенных по организационно-структурному, тематическому, технологическому или другим признакам для выполнения автоматизированной обработки информации (данных) с целью удовлетворения информационных потребностей субъектов информационных отношений.

Обработка информации в АС –любая совокупность операций (прием, сбор, накопление, хранение, преобразование, отображение, выдача и т.п.), осуществляемых над информацией с использованием средств АС.

Субъекты по отношению к определенной информации могут выступать в качестве:

- источников (поставщиков) информации;
- пользователей (потребителей) информации;
- собственников (владельцев, распорядителей) информации;
- физических и юридических лиц, о которых собирается информация;
- владельцев систем сбора и обработки информации и участников процессов обработки и передачи информации и т.д.

Для успешного осуществления своей деятельности по управлению объектами некоторой предметной области субъекты информационных отношений могут быть заинтересованы в обеспечении:

- своевременного доступа к необходимой им информации;
- конфиденциальности определенной части информации;
- достоверности информации;
- защиты от навязывания им ложной информации;
- защиты части информации от незаконного ее тиражирования;
- разграничения ответственности за нарушения законных прав других субъектов информационных отношений и установленных правил обращения с информацией;
- возможности осуществления непрерывного контроля и управления процессами обработки и передачи информации.

Поэтому под безопасностью автоматизированной системы обработки информации (компьютерной системы) будем понимать защищенность всех ее компонентов (технических средств, программного обеспечения, данных и персонала) от подобного рода нежелательных для соответствующих субъектов информационных отношений воздействий.

Безопасность любого компонента (ресурса) АС складывается из обеспечения трех его характеристик: конфиденциальности, целостности и доступности.

Конфиденциальность компонента системы заключается в том, что он доступен только тем субъектам доступа (пользователям, программам, процессам), которым предоставлены на то соответствующие полномочия.

Целостность компонента системы предполагает, что он может быть модифицирован только субъектом, имеющим для этого соответствующие права. Целостность является гарантией корректности (неизменности, работоспособности) компонента в любой момент времени.

Доступность компонента означает, что имеющий соответствующие полномочия субъект может в любое время без особых проблем получить доступ к необходимому компоненту системы (ресурсу).

Ценность информации

Под ценностью информации понимается ее свойство, характеризующее потери собственника данной информации при реализации определенной угрозы, выраженные в стоимостном, временном либо ином эквиваленте.

Среди подходов к построению моделей защиты ИС, основанных на понятии ценности информации наиболее известными являются: оценка, анализ и управление рисками, порядковые шкалы ценностей, модели решетки ценностей.

Пример

При оценке ценности информации в государственных структурах используется линейная порядковая шкала ценностей. Всю информацию сравнивают экспертным путем и относят к различным уровням ценности. В этом случае документам, отнесенным к некоторому уровню по шкале, присваиваются соответствующие грифы секретности. Сами грифы секретности образуют порядковую шкалу, например (принятую почти всеми государствами): НЕСЕКРЕТНО < КОНФИДЕНЦИАЛЬНО < СЕКРЕТНО < СОВЕРШЕННО СЕКРЕТНО. Более высокий класс имеет более высокую ценность и поэтому требования по его защите от несанкционированного доступа более высокие.

Рассматриваемая шкала хронологически была самой ранней и перестала удовлетворять требованиям ИТ, более детальной классификации. Разработка формализованных моделей информационных систем привело к разработке ценностной модели в виде решетки ценностей, которая является обобщением порядковой шкалы. Ее элементы представляют дискретную модель на базе введенной алгебры: с требованиями рефлексивности, транзитивности, антисимметричности, а также верхней и нижней грани.

Модель решетки ценностей

Пусть дано SC - конечное частично упорядоченное множество относительно бинарного отношения <, т.е. для каждых A, B, C выполняется

1) рефлексивность: $A<A$,

2) транзитивность: $A<B$, $B<C \Rightarrow A<C$,

3) антисимметричность: $A<B$, $B<A \Rightarrow A=B$.

Определение 1.8. Для $A, B \in SC$ элемент $C=A \oplus B \in SC$ называется наименьшей верхней границей (верхней гранью), если

1) $A<C$, $B<C$;

2) $A<D$, $B<D \Rightarrow C<D$ для всех $D \in SC$.

Элемент $A \oplus B$, вообще говоря, может не существовать. Если наименьшая верхняя граница существует, то из антисимметричности следует единственность.

Определение 1.9. Для $A, B \in C$ элемент $E=A \otimes B \in SC$ называется наибольшей нижней границей (нижней гранью), если

1) $E<A$, $E<B$;

2) $D<A$, $D<B \Rightarrow D<E$.

Эта граница также может не существовать. Если она существует, то из антисимметричности следует единственность.

Определение 1.10. (SC, <) называется решеткой, если для любых $A, B \in SC$ существует $A \oplus B \in SC$ и $A \otimes B \in SC$.

Лемма. Для любого набора $S=\{A_1,...,A_n\}$ элементов из решетки SC существуют единственные элементы,:

$\oplus S=A_1 \oplus ... \oplus A_n$ - наименьшая верхняя граница S;

$\otimes S=A_1 \otimes ... \otimes A_n$ - наибольшая нижняя граница S.

Для всех элементов SC в конечных решетках существует верхний элемент $High = \oplus SC$, аналогично существует нижний элемент $Low = \otimes SC$.

Определение 1.11. Конечная линейная решетка - это линейно упорядоченное множество, можно всегда считать $\{0, 1, ..., n\}=SC$.

Для большинства встречающихся в теории защиты информации решеток существует представление решетки в виде графа. Рассмотрим корневое дерево на вершинах из конечного

множества $X=\{X_1, X_2...X_n\}$ с корнем в X_i. Пусть на единственном пути, соединяющем вершину X_1 с корнем, есть вершина X_j. Положим по определению, что $X_i<X_j$. Очевидно, что таким образом на дереве определен частичный порядок. Кроме того, для любой пары вершин X_i и X_j существует элемент $X_i \oplus X_j$, который определяется точкой слияния путей из X_i и X_j в корень. Однако такая структура не является решеткой, т.к. здесь нет нижней грани. Оказывается, что от условия единственности пути в корень можно отказаться, сохраняя при этом свойства частичного порядка и существование верхней грани. Например, добавим к построенному дереву вершину L, соединив с ней все концевые вершины. Положим i=l,...,n, $L<X_j$. Для остальных вершин порядок определяется как раньше. Построенная структура является решеткой.

Приведенный пример не исчерпывает множество решеток, представимых в виде графов, однако поясняет как связаны графы и решетки. Не всякий граф определяет решетку.

MLS решетка

Название происходит от аббревиатуры Multilevel Security и лежит в основе государственных стандартов оценки информации. Решетка строится как прямое произведение линейной решетки L и решетки SC подмножеств множества X, т.е. (α,β), (α',β') -элементы произведения, $\beta,\beta' \in L$ - линейная решетка, $\alpha,\alpha' \in SC$ - решетка подмножеств некоторого множества X. Тогда

$(\alpha,\beta)<(\alpha',\beta') \Leftrightarrow \alpha \subseteq \alpha', \beta<\beta'$

Верхняя и нижняя границы определяются следующим образом:

$(\alpha,\beta) \oplus (\alpha',\beta') \Leftrightarrow (\alpha \cup \alpha', \max\{\beta,\beta'\})$,

$(\alpha,\beta) \otimes (\alpha',\beta') \Leftrightarrow (\alpha \cap \alpha', \min\{\beta,\beta'\})$.

Вся информация {объекты системы} отображается в точки решетки $\{(a,\beta)\}$. Линейный порядок, как правило, указывает гриф секретности. Точки множества X обычно называются категориями.

Свойства решетки в оценке информации существенно используются при классификации новых объектов, полученных в результате вычислений. Пусть дана решетка ценностей SC, множество текущих объектов O, отображение C: $0 \rightarrow S$, программа использует информацию объектов $0_1,..,0_n$, которые классифицированы точками решетки $C(0_1),...,C(0_n)$. В результате работы программы появился объект O, который необходимо классифицировать. Это можно сделать, положив $C(0)= C(0_1) \oplus ... \oplus C(0_n)$. Такой подход к классификации наиболее распространен в государственных структурах. Например, если в сборник включаются две статьи с грифом секретно и совершенно секретно соответственно, и по тематикам: первая - кадры, вторая - криптография, то сборник приобретает гриф совершенно секретно, а его тематика определяется совокупностью тематик статей (кадры, криптография).

Определение требований к защищенности информации

Исторически сложившийся подход к классификации государственной информации (данных) по уровням требований к ее защищенности основан на рассмотрении и обеспечении только одного свойства информации - ее конфиденциальности (секретности). Требования же к обеспечению целостности и доступности информации, как правило, лишь косвенно фигурируют среди общих требований к системам обработки этих данных. Считается, что раз к информации имеет доступ только узкий круг доверенных лиц, то вероятность ее искажения (несанкционированного уничтожения) незначительна.

Если такой подход в какой-то степени оправдан в силу существующей приоритетности свойств безопасности важной государственной информации, то это вовсе не означает, что его механический перенос в другую предметную область (с другими субъектами и их интересами) будет иметь успех.

Во многих областях деятельности (предметных областях) доля конфиденциальной информации сравнительно мала. Для коммерческой и персональной информации, равно как и для государственной информации, не подлежащей засекречиванию, приоритетность свойств безопасности информации может быть иной. Для открытой информации, ущерб от разглашения которой несущественен, важнейшими могут быть такие качества, как доступность, целостность или защищенность от неправомерного тиражирования. К примеру, для платежных (финансовых) документов самым важным является свойство их целостности (достоверности, не искаженности). Затем, по степени важности, следует свойство доступности (потеря платежного документа или задержка платежей может обходиться очень дорого). Требований к обеспечению конфиденциальности отдельных платежных документов может не предъявляется вообще.

Попытки подойти к решению вопросов защиты такой информации с позиций традиционного обеспечения только конфиденциальности, терпят провал. Основными причинами этого, на наш взгляд, являются узость существующего подхода к защите информации, отсутствие опыта и соответствующих проработок в плане обеспечения целостности и доступности информации, не являющейся конфиденциальной.

Развитие системы классификации информации по уровням требований к ее защищенности предполагает введение ряда степеней (градаций) требований по обеспечению каждого из свойств безопасности информации: доступности, целостности, конфиденциальности и защищенности от тиражирования. Пример градаций требований к защищенности:

- нет требований;
- низкие;
- средние;
- высокие;
- очень высокие.

Количество дискретных градаций и вкладываемый в них смысл могут различаться. Главное, чтобы требования к защищенности различных свойств информации указывались отдельно и достаточно конкретно (исходя из серьезности возможного наносимого субъектам информационных отношений ущерба от нарушения каждого из свойств безопасности информации).

В дальнейшем любой отдельный функционально законченный документ (некоторую совокупность знаков), содержащий определенные сведения, вне зависимости от вида носителя, на котором он находится, называется информационным пакетом.

К одному типу информационных пакетов будем относить пакеты (типовые документы), имеющие сходство по некоторым признакам (по структуре, технологии обработки, типу сведений и т.п.).

Задача состоит в определении реальных уровней заинтересованности (высокая, средняя, низкая, отсутствует) субъектов в обеспечении требований к защищенности каждого из свойств различных типов информационных пакетов, циркулирующих в АС.

Требования же к системе защиты АС в целом (методам и средствам защиты) должны определяться, исходя из требований к защищенности различных типов информационных пакетов, обрабатываемых в АС, и с учетом особенностей конкретных технологий их обработки и передачи (уязвимости).

В одну категорию объединяются типы информационных пакетов с равными приоритетами и уровнями требований к защищенности (степенью важности обеспечения их свойств безопасности: доступности, целостности и конфиденциальности).

Предлагаемый порядок определения требований к защищенности циркулирующей в системе информации представлен ниже:

1. Составляется общий перечень типов информационных пакетов, циркулирующих в системе (документов, таблиц). Для этого с учетом предметной области системы пакеты информации разделяются по ее тематике, функциональному назначению, сходности технологии обработки и т.п. признакам.

2. На последующих этапах первоначальное разбиение информации (данных) на типы пакетов может уточняться с учетом требований к их защищенности.

3. Затем для каждого типа пакетов, выделенного в первом пункте, и каждого критического свойства информации (доступности, целостности, конфиденциальности) определяются (например, методом экспертных оценок):

- перечень и важность (значимость по отдельной шкале) субъектов, интересы которых затрагиваются при нарушении данного свойства информации;
- уровень наносимого им при этом ущерба (незначительный, малый, средний, большой, очень большой и т.п.) и соответствующий уровень требований к защищенности.
- при определении уровня наносимого ущерба необходимо учитывать:
- стоимость возможных потерь при получении информации конкурентом;
- стоимость восстановления информации при ее утрате;

 затраты на восстановление нормального процесса функционирования АС и т.д.

Если возникают трудности из-за большого разброса оценок для различных частей информации одного типа пакетов, то следует пересмотреть деление информации на типы пакетов, вернувшись к предыдущему пункту методики.

4. Для каждого типа информационных пакетов с учетом значимости субъектов и уровней наносимого им ущерба устанавливается степень необходимой защищенности по каждому из свойств информации (при равенстве значимости субъектов выбирается максимальное значение уровня).

Пример оценки требований к защищенности некоторого типа информационных пакетов приведен в таблице 1.1.

Таблица 1.1. Пример оценки требований к защищенности

Субъекты	Уровень ущерба по свойствам информации			
	конфиденциальность	целостность	доступность	защита от тиражирования
N1	Нет	Средняя	Средняя	Нет
N2	Высокая	Средняя	Средняя	Нет
Nm	Низкая	Низкая	Низкая	Нет
В итоге	Высокая	Средняя	Средняя	Нет

Критерии, условия и принципы отнесения информации к защищаемой.

Виды конфиденциальной информации.

Информация составляет служебную или коммерческую тайну в случае, если

- информация имеет действительную или потенциальную коммерческую ценность в силу неизвестности ее третьим лицам;
- к ней нет свободного доступа на законном основании;
- обладатель информации принимает меры к охране ее конфиденциальности.

Под служебной тайной (по аналогии с коммерческой тайной в негосударственных структурах) следует понимать служебную информацию в государственных структурах, имеющую коммерческую ценность. В отличие от коммерческой тайны (в коммерческих структурах) защищаемая государством конфиденциальная информация не ограничивается только коммерческой ценностью, поэтому служебная тайна является составной частью конфиденциальной информации. В государственных структурах еще может быть информация, имеющая политическую или иную ценность. Поскольку к служебной тайне она не относится, ей необходимо присваивать гриф “конфиденциально” или иной гриф.

2. Угрозы информационной безопасности, их классификация. Основные методы реализации угроз, этапы осуществления атаки на информационную систему

ПОНЯТИЕ УГРОЗЫ БЕЗОПАСНОСТИ

С позиции обеспечения безопасности информации в ИВС целесообразно рассматривать в виде трех связных взаимовлияющих друг на друга компонент:

1) информация;
2) технические и программные средства;
3) обслуживающий персонал и пользователи.

Целью создания любой ИВС является удовлетворение потребностей пользователей в своевременном получении достоверной информации и сохранении ее конфиденциальности. При этом задача обеспечения информации должна решаться путем защиты от внешних и внутренних неразрешенных(несанкционированных) воздействий.

Под угрозой обычно понимают потенциально возможно событие, действие(воздействие), процесс или явление, которое может привести к нанесению ущерба чьим-либо интересам. В дальнейшем изложении угрозой информационной безопасности АС будем называть возможность реализации воздействия на информацию, обрабатываемую АС, приводящего к искажению, уничтожению, копированию, блокированию, доступа к ин-формации, а также возможность воздействия на компоненты АС, приводящего к утрате, уничтожению или сбою функционирования носителя информации, средства взаимодействия с носителем или средства его управления.

Утечка информации рассматривается как бесконтрольный и неправомерный выход конфиденциальной информации за пределы организации или круга лиц, которым эта ин-формация была доверена.

Существует три разновидности угроз.

1. Угроза нарушения конфиденциальности заключается в том, что информация становится известной тому, кто не располагает полномочиями доступа к ней. Она имеет место всякий раз, когда получен доступ к некоторой секретной информации, хранящейся в вычислительной системе или передаваемой от одной системы к другой. Иногда, в связи с угрозойнарушения конфиденциальности, используется термин «утечка».

2. Угроза нарушения целостности включает в себя любое умышленное изменение информации, хранящейся в вычислительной системе или передаваемой из одной системы в другую. Когда злоумышленники преднамеренно изменяют информацию, говорится, что целостность информации нарушена. Целостность также будет нарушена, если к несанкционированному изменению приводит случайная ошибка программного или аппаратного обеспечения. Санкционированными изменениями являются те, которые сделаны уполномоченными лицами с обоснованной целью(например, санкционированным изменением является периодическая запланированная коррекция некоторой базы данных).

Целостность информации – существование информации в неискаженном виде(неизменном по отношению к некоторому фиксированному ее состоянию). Чаще 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.

3. Угроза 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 в обращении к ним возникает необходимость.

КЛАССИФИКАЦИЯ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

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

1. По природе возникновения.

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

Искусственные угрозы– угрозы информационной безопасности АС, вызванные деятельностью человека.

2. По степени преднамеренности проявления.

Угрозы случайного действия и/или угрозы, вызванные ошибками или халатностью персонала. Угрозы, не связанные с преднамеренными действиями злоумышленников и реализуемые в случайные моменты времени, называют случайными или непреднамеренными.

Реализация угроз этого класса приводит к наибольшим потерям информации (до80 % ущерба). При этом может происходить уничтожение, нарушение целостности, доступности и конфиденциальности информации, 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(не-умышленная порча оборудования, удаление, искажение файлов с важной информацией или программ, в том числе системных и т.п.);
- неправомерное включение оборудования или изменение режимов работы уст-ройств и программ;
- неумышленная порча носителей информации;
- пересылка данных по ошибочному адресу абонента(устройства);
- ввод ошибочных данных;
- неумышленное повреждение каналов связи.

Угрозы преднамеренного действия, например:

- традиционный или универсальный шпионаж и диверсии(подслушивание, визуальное наблюдение; хищение документов и машинных носителей, хищение программ и атрибутов системы защиты, подкуп и шантаж сотрудников, сбор и анализ отходов машинных носителей, поджоги, взрывы);
- несанкционированный доступ к информации(реализуется посредством отсутст-вия системы разграничения доступа(СРД), сбоями или отказами технических средств), ошибками в СРД, фальсификацией полномочий);
- побочные электромагнитные излучения и наводки(ПЭМИН);
- несанкционированная модификация структур(алгоритмической, программной, технической);
- информационные инфекции(вредительские программы).

3. По непосредственному источнику угроз.

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

Угрозы, источником которых является человек, например:

- 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.).

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

− 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.

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

− нелегальное внедрение и использование неучтенных программ(игровых, обу-чающих, технологических и других, не являющихся необходимыми для выполнения наруши-телем своих служебных обязанностей) с последующим необоснованным расходованием ре-сурсов(загрузка процессора, захват оперативной памяти и памяти на внешних носителях);

− заражение компьютера вирусами с деструктивными функциями.

4. По положению источника угроз.

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

− перехват побочных электромагнитных, акустических и других излучений уст-ройств и линий связи, а также наводок активных излучений на вспомогательные техниче-ские 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 выясне-ния протоколов обмена, правил вхождения в связь и авторизации пользователя и после-дующих попыток их имитации для проникновения в систему;

− дистанционная фото- и видеосъемка.

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

− хищение производственных отходов(распечаток, записей, списанных носителей информации и т.п.);

− отключение или вывод из строя подсистем обеспечения функционирования вы-числительных систем(электропитания, охлаждения и вентиляции, линий связи и т.д.);

− применение подслушивающих устройств.

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

Угрозы, источник которых расположен в АС, например:

− 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.

Угрозы, которые могут проявляться независимо от активности АС, например:

− вскрытие шифров криптозащиты информации;

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

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

6. По степени воздействия на АС.

Пассивные угрозы, которые при реализации ничего не меняют в структуре и содер-жании АС, например: угроза копирования секретных данных.

Активные угрозы, которые при воздействии вносят изменения в структуру и содер-жание АС, например:

− 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.

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

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

8. По способу доступа к ресурсам АС.

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

− незаконное получение паролей и других реквизитов разграничения доступа (агентурным путем, используя халатность пользователей, подбором, имитацией интер-фейса системы и т.д.) с последующей маскировкой под зарегистрированного пользовате-ля(«маскарад»);

− несанкционированное использование терминалов пользователей, имеющих уни-кальные физические характеристики, такие как номер рабочей станции в сети, физиче-ский адрес, адрес в системе связи, аппаратный блок кодирования и т.п.

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

− вход в систему в обход средств защиты(загрузка посторонней операционной системы со сменных магнитных носителей и т.п.);

− угроза несанкционированного доступа к ресурсам АС путем использования не-документированных возможностей ОС.

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

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

Угрозы доступа к информации в оперативной памяти, например:

− чтение остаточной информации из оперативной памяти;

− чтение информации из областей оперативной памяти, используемых операцион-ной системой(в том числе подсистемой защиты) или другими пользователями, в асин-хронном режиме, используя недостатки мультизадачных АС и систем программирова-ния;

− угроза доступа к системной области оперативной памяти со сторон прикладных программ.

Угрозы доступа к информации, циркулирующей в линиях связи, например:

− незаконное подключение к линиям связи с целью работы«между строк» с ис-пользованием пауз в действиях законного пользователя от его имени с последующим 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 ау-тентификации с последующим вводом дезинформации и навязыванием ложных сообще-ний;

− перехват всего потока данных с целью дальнейшего анализа не в реальном мас-штабе времени.

Угрозы доступа к информации, отображаемой на терминале или печатаемой на принтере, например, угроза записи отображаемой информации на скрытую видеокамеру.

ОСНОВНЫЕ МЕТОДЫ РЕАЛИЗАЦИИ УГРОЗ ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ

К основным направлениям реализации злоумышленником информационных угроз относятся:

− непосредственное обращение к объектам доступа;

− создание программных и технических средств, выполняющих обращение к объектам доступа в обход средств защиты;

− модификация средств защиты, позволяющая реализовать угрозы информационной безопасности;

− внедрение в технические средства АС программных или технических механиз-ов, нарушающих предполагаемую структуру и функции АС.

К числу основных методов реализации угроз информационной безопасности АС относятся:

− определение злоумышленником типа и параметров носителей информации;

− получение злоумышленником информации о программно-аппаратной среде, типе и параметрах средств вычислительной техники, типе и версии операционной системы, составе прикладного программного обеспечения;

− получение злоумышленником детальной информации о функциях, выполняемых АС;

− получение злоумышленником данных о системах защиты;

− определение способа представления информации;

− определение злоумышленником содержания данных, обрабатываемых в АС, на качественном уровне(мониторинг дешифрования сообщений);

− хищение(копирование) машинных носителей информации, имеющих конфиденциальные данные;

− хищение(копирование) носителей информации;

− использование специальных технических средств для перехвата побочных электромагнитных излучений и наводок(ПЭМИН) – конфиденциальные данные перехватываются злоумышленником путем изменения информативных сигналов из электромагнитного излучения и наводок по цепям питания средств вычислительной техники, входящей в АС;

− уничтожение средств ВТ и носителей информации;

− несанкционированный доступ пользователя к ресурсам АС путем преодоления систем защиты с использованием спецсредств, приемов, методов;

− несанкционированное превышение пользователем своих полномочий;

− несанкционированное копирование программного обеспечения;

− перехват данных, передаваемых по каналам связи;

− визуальное наблюдение– конфиденциальные данные считываются с экранов терминалов, распечаток в процессе их печати и т.п.;

− раскрытие представления информации(дешифрование данных);

− раскрытие содержания информации на семантическом уровне к смысловой составляющей информации, хранящейся в АС;

− уничтожение машинных носителей информации;

− внесение пользователем несанкционированных изменений программно-аппаратные компоненты АС и обрабатываемых данных;

− установка и использование нештатного аппаратного и/или программного обеспечения;

− заражение программными вирусами;

− внесение искажений в представление данных, уничтожение на уровне представ-ления, искажение информации при передаче по линиям связи;

− внедрение дезинформации;

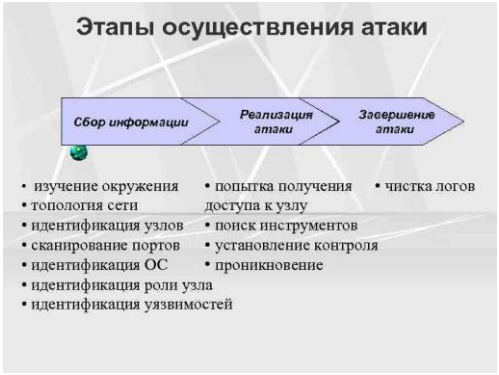
− выведение из строя машинных носителей информации без уничтожения информации(выведение из строя электронных блоков жестких дисков и т.п.);

− проявление ошибок проектирования и разработки аппаратных программных компонентов АС;

− обход(отключение) механизмов защиты– загрузка злоумышленником нештатной операционной системы с дискеты, использование режимов программно-аппаратных компонент АС

− искажение соответствия синтаксических и семантических конструкций языка– установление новых значений слов, выражений и т.п.;

− запрет на использование информации– имеющаяся информация каким-либо причинам не может быть использована.



Подготовительный этап заключается в поиске злоумышленником предпосылок для осуществления той или иной атаки (поиск уязвимостей в системе). На этапе реализации атаки осуществляется использование найденных уязвимостей. На третьем, заключительном, этапе злоумышленник завершает атаку и старается скрыть следы вторжения.

3. Средства и методы обеспечения целостности информации. Средства и методы обеспечения конфиденциальности информации

Процесс осуществления атаки на АС включает три этапа. Первый этап, подготовительный, заключается в поиске предпосылок для осуществления той или иной атаки. На этом этапе ищутся уязвимости, использование которых приводит к реализации атаки, т. е. ко второму этапу. На третьем этапе атака завершается, «заметаются» следы и т. д. При этом первый и третий этапы сами по себе могут являться атаками.

Обнаруживать, блокировать и предотвращать атаки можно несколькими путями. Первый способ, и самый распространённый, – это обнаружение уже реализуемых атак. Данный способ функционирует на втором этапе осуществления атаки. Этот способ применяется в «классических» системах обнаружения атак:

- серверах аутентификации;
- системах разграничения доступа;
- межсетевых экранах и т. п.

Основным недостатком средств данного класса является то, что атаки могут быть реализованы повторно. Они также повторно обнаруживаются и блокируются. И так далее, до бесконечности.

Второй путь – предотвратить атаки ещё до их реализации. Осуществляется это путём поиска уязвимостей, которые могут быть использованы для реализации атаки.

И наконец, третий путь – обнаружение уже совершённых атак и предотвращение их повторного осуществления.

Таким образом, системы обнаружения атак могут быть классифицированы по этапам осуществления атаки.

1. Системы, функционирующие на первом этапе осуществления атаки и позволяющие обнаружить уязвимости информационной системы, используемые нарушителем для реализации атаки. Средства этой категории называются системами анализа защищенности (security assessment systems) или сканерами безопасности (security scanners).

Системы анализа защищённости проводят всесторонние исследования систем с целью обнаружения уязвимостей. Результаты, полученные от средств анализа защищённости, представляют «мгновенный снимок» состояния защиты системы в данный момент времени. Несмотря на то что эти системы не могут обнаруживать атаку в процессе её развития, они могут определить возможность реализации атак.

Эти системы реализуют две стратегии. Первая стратегия – пассивная, реализуемая на уровне операционной системы, СУБД и приложений, при которой осуществляется анализ конфигурационных файлов и системного реестра на наличие неправильных параметров, файлов паролей на наличие легко угадываемых паролей, а также других системных объектов на нарушения политики безопасности. Вторая стратегия – активная, осуществляется в большинстве случаев на сетевом уровне. Она заключается в воспроизведении наиболее распространенных сценариев атак и анализе реакции системы на эти сценарии.

2. Системы, функционирующие на втором этапе осуществления атаки и позволяющие обнаружить атаки в процессе их реализации, т.е. в режиме реального (или близкого к реальному) времени. Именно эти средства и принято считать системами обнаружения атак в классическом понимании. Помимо этого в последнее время выделяется новый класс средств обнаружения атак – обманные системы.

Обнаружение атак реализуется посредством анализа или журналов регистрации операционной системы и прикладного программного обеспечения, или сетевого трафика в реальном времени. Компоненты обнаружения атак, размещенные на узлах или сегментах сети, оценивают различные действия, в том числе и использующие известные уязвимости, сравнивая контролируемое пространство (сетевой трафик или журналы регистрации) с известными шаблонами (сигнатурами) несанкционированных действий.

Обманные системы могут использовать следующие методы: сокрытие, камуфляж и дезинформацию. Ярким примером использования первого метода является сокрытие сетевой топологии при помощи межсетевого экрана. Примером камуфляжа можно назвать использование Unix-подобного графического интерфейса в системе, функционирующей под управлением

операционной системы Windows NT. Если злоумышленник случайно увидел такой интерфейс, то он будет пытаться реализовать атаки, характерные для ОС Unix, а не для ОС Windows NT . Это существенно увеличит время, необходимое для «успешной» реализации атаки. И наконец, в качестве примера дезинформации можно назвать использование заголовков, которые бы давали понять злоумышленнику, что атакуемая им система уязвима.

Системы, реализующие камуфляж и дезинформацию, эмулируют те или иные известные уязвимости, которых в реальности не существует. Использование таких систем приводит к следующему.

• Увеличение числа выполняемых нарушителем операций и действий. Так как невозможно заранее определить, является ли обнаруженная нарушителем уязвимость истинной или нет, злоумышленнику приходится выполнять много дополнительных действий, чтобы выяснить это. И даже дополнительные действия не всегда помогают. Например, попытка запустить программу подбора паролей на сфальсифицированный и несуществующий в реальности файл приведёт к бесполезной трате времени без какого-либо видимого результата. Нападающий будет думать, что он не смог подобрать пароли, в то время как на самом деле программа «взлома» была просто обманута.

• Получение возможности отследить нападающих. За тот период времени, когда нападающие пытаются проверить все обнаруженные уязвимости, в том числе и фиктивные, администраторы безопасности могут проследить весь путь до нарушителя или нарушителей и предпринять соответствующие меры.

3. Системы, функционирующие на третьем этапе осуществления атаки и позволяющие обнаружить уже совершённые атаки. Эти системы делятся на два класса – системы контроля целостности, обнаруживающие изменения контролируемых ресурсов, и системы анализа журналов регистрации.

Системы контроля целостности работают по замкнутому циклу, обрабатывая файлы, системные объекты и атрибуты системных объектов с целью получения контрольных сумм; затем они сравнивают их с предыдущими контрольными суммами, отыскивая изменения. Когда изменение обнаружено, система посылает сообщение администратору, фиксируя вероятное время изменения.

Существует ещё одна распространённая классификация систем обнаружения нарушения политики безопасности – по принципу реализации: host-based, т.е. обнаруживающие атаки, направленные на конкретный узел сети, и network-based, направленные на всю сеть или сегмент сети. Существуют три основных вида систем обнаружения атак на уровне узла.

4. Системы, обнаруживающие атаки на конкретные приложения.

5. Системы, обнаруживающие атаки на операционные системы.

6. Системы, обнаруживающие атаки на системы управления базами данных (СУБД).

В сетевых операционных системах вопросы защиты целостности данных решаются средствами разграничения доступа (блокировка возможности доступа, запрет на изменение, мониторинг использования файлов). К сожалению, средствами операционной системы Windows XP невозможно защитить файлы от удаления. Windows XP позволяет устанавливать атрибут файла «только для чтения», «скрытый» и делать файлы невидимыми в окне Проводника. Это может служить определённой защитой от ошибочных действий пользователя. Однако и эти файлы могут быть удалены (например, при удалении содержащей их папки).

Как сохранность данных, так и надёжная работа программного обеспечения невозможны без решения задачи их защиты от разрушающих воздействий компьютерных вирусов. Немаловажную роль при этом играет знание путей попадания вируса в компьютерную систему и соблюдение мер предосторожности при выполнении потенциально опасных действий (или отказ от их выполнения). Вместе с тем надёжная защита от вирусов может быть обеспечена только с использованием специальных антивирусных программных средств.

Средства и методы обеспечения конфиденциальности обеспечения информации

В рамках направления защиты конфиденциальной информации необходимы: организация контроля доступа к информации, защита информации от действий нелегальных пользователей и от

несанкционированных действии легальных пользователей. Если речь идёт об авторских программных системах, важным вопросом является защита данных от копирования.

Наиболее распространёнными мероприятиями защиты конфиденциальной информации являются:

- разграничение доступа к данным;
- парольная защита;
- шифрование;
- скрытие данных;
- уничтожение остаточных данных;
- защита от копирования программных систем.

Большинство сетевых операционных систем располагают развитыми средствами разграничения доступа и защиты от несанкционированного доступа (НСД). Для скрытия и шифрования данных могут использоваться специальные утилиты.

Проблема остаточных данных вызвана типичной схемой удаления файлов: запись о файле удаляется из специальной базы данных ОС – таблицы размещения файлов (FAT), а занимаемое им место на диске помечается как свободное. Таким образом, производится логическое, но не физическое удаление файла. Незащищённая конфиденциальная информация может быть прочитана из остаточных данных с помощью специальных утилит.

Уничтожение остаточных данных подразумевает возможность полного удаления файлов на физическом уровне, очистку свободного дискового пространства, включая данные из хвостовых частей последних кластеров файлов. Эти возможности должны обеспечиваться средствами защищённой операционной системы, Windows XP не выполняет подобных функций.

MS Office располагает собственными средствами защиты документов. Для документов MS Office имеются возможности: ограничить доступ к документу (парольная защита открытия документа, шифрование), установить запрет на изменение документа или его частей, скрыть часть документа (MS Excel). Кроме того, приложение MS Access позволяет установить защиту на уровне пользователя. Этот способ защиты реализует контроль доступа к объектам базы данных и подобен методам разграничения доступа, используемым в большинстве сетевых систем.

Идентификация и аутентификация – взаимосвязанные методы защиты от НСД, при их реализации часто используется криптографическое преобразование информации (шифрование).

Идентификация – это присвоение пользователям идентификатора и проверка предъявляемых идентификаторов по списку присвоенных.

Аутентификация – это проверка принадлежности пользователю предъявленного им идентификатора (подтверждение или проверка подлинности).

Под безопасностью (стойкостью) системы идентификации и аутентификации понимается степень обеспечиваемых ею гарантий того, что злоумышленник не способен пройти аутентификацию от имени другого пользователя.

Система идентификации и аутентификации является одним из ключевых элементов инфраструктуры защиты от НСД любой информационной системы.

Различают три группы методов аутентификации, основанных на наличии у каждого пользователя:

- индивидуального объекта заданного типа;
- знаний некоторой известной только ему и проверяющей стороне информации;
- индивидуальных биометрических характеристик.

К первой группе относятся методы аутентификации, использующие удостоверения, пропуска, магнитные карты и другие носимые устройства, которые широко применяются для контроля доступа в помещения, а также входят в состав программно-аппаратных комплексов защиты от НСД к средствам вычислительной техники.

Во вторую группу входят методы аутентификации, использующие пароли. По экономическим причинам они включаются в качестве базовых средств защиты во многие программно-аппаратные комплексы защиты информации. Все современные операционные системы и многие приложения имеют встроенные механизмы парольной защиты.

Применение методов аутентификации, основанных на измерении и сравнении с эталоном заданных индивидуальных характеристик пользователя: тембра голоса, отпечатков пальцев, структуры радужной оболочки глаза, сдерживается высокой стоимостью необходимого оборудования.

Если в процедуре аутентификации участвуют только две стороны, устанавливающие подлинность друг друга, то такая процедура называется непосредственной аутентификацией. Если же в процессе аутентификации участвуют не только эти стороны, но и другие, вспомогательные, говорят об аутентификации с участием доверенной стороны. При этом третью сторону называют сервером аутентификации или арбитром

Уровень конфиденциальности информации является одной из самых важных категорий, принимаемых в рассмотрение при создании определённой политики безопасности.

Классификация по степени конфиденциальности – одна из основных и наиболее старых классификаций данных. Она применялась ещё задолго до появления вычислительной техники и с тех пор изменилась незначительно.

Различные классы конфиденциальной информации необходимо снабжать различными по уровню безопасности системами техниче-ских и административных мер.

При работе с информацией 1-го класса конфиденциальности ре-комендуется выполнение следующих требований:

- осведомление сотрудников о закрытости данной информации;
- общее ознакомление сотрудников с основными возможными методами атак на информацию;
- ограничение физического доступа;
- полный набор документации по правилам выполнения опера-ций с данной информацией

При работе с информацией 2-го класса конфиденциальности к перечисленным выше требованиям добавляются следующие:

- расчёт рисков атак на информацию;
- поддержание списка лиц, имеющих доступ к данной информации;
- по возможности выдача подобной информации под расписку (в том числе электронную);
- автоматическая система проверки целостности системы и её средств безопасности;
- надёжные схемы физической транспортировки;
- обязательное шифрование при передаче по линиям связи;
- схема бесперебойного питания ЭВМ.

При работе с информацией 3-го класса конфиденциальности ко всем перечисленным выше требованиям добавляются следующие:

- детальный план спасения либо надежного уничтожения информации в аварийных ситуациях (пожар, наводнение, взрыв);
- защита ЭВМ либо носителей информации от повреждения водой и высокой температурой;
- криптографическая проверка целостности информации.

4. Законодательный уровень информационной безопасности. Подсистема организационно-правовой защиты. Защита программного обеспечения авторским правом

ПРАВОВЫЕ И ОРГАНИЗАЦИОННЫЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ В ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМАХ

Комплексная защита информации создается на объектах для блокирования(парирования) всех возможных или наиболее вероятных угроз безопасности информации. Для парирования той или иной угрозы используется определенная совокупность средств и методов защиты, некоторые из них защищают от нескольких угроз одновременно.

Среди методов защиты имеются и универсальные методы, являющиеся базовыми при построении любой системы защиты.

Правовые методы защиты информации служат основой легитимного построения и использования системы защиты любого назначения.

Организационные методы защиты информации используются для парирования нескольких угроз, кроме того, их использование в любой системе защиты обязательно.

ПРАВОВОЕ РЕГУЛИРОВАНИЕ В ОБЛАСТИ БЕЗОПАСНОСТИ ИНФОРМАЦИИ

Государство должно обеспечить в стране защиту информации как в масштабах всего государства, так и на уровне организаций и своих граждан. Для этого государство обязано:

- выработать государственную политику безопасности в области информационных технологий;
- законодательно определить правовой статус ИВС, информации, систем защиты информации, владельцев и пользователей информации и т.д.;
- создать иерархическую структуру государственных органов, вырабатывающих и проводящих в жизнь политику безопасности информационных технологий;
- создать систему стандартизации, лицензирования и сертификации в области защиты информации;
- обеспечить приоритетное развитие отечественных защищенных информационных технологий;
- повышать уровень образования граждан в области информационных технологий, воспитывать у них патриотизм и бдительность;
- установить ответственность граждан за нарушения законодательства в области информационных технологий.

Защищаемая информация – информация, являющаяся предметом собственности и подлежащая защите в соответствии с требованиями правовых документов или требованиями, устанавливаемыми собственниками информации. Собственниками информации могут быть: государство, юридическое лицо, группа физических лиц, от-дельное физическое лицо.

Угроза (безопасности информации) – совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения безопасности информации.

Источник угрозы безопасности информации – субъект (физическое лицо, материальный объект или физическое явление), являющийся непосредственной причиной возникновения угрозы безопасности информации. По типу источника угрозы делят на связанные и не-связанные с деятельностью человека. Примерами могут служить удаление пользователем файла с важной информацией и пожар в здании, соответственно. Угрозы, связанные с деятельностью человека, разделяют на угрозы случайного и преднамеренного характера. В послед-нем случае источник угрозы называют нарушителем или злоумышленником.

Уязвимость (информационной системы) – свойство информационной системы, обуславливающее возможность реализации угроз безопасности обрабатываемой в ней информации. Например, угроза

потери информации из-за сбоя в сети электропитания реализуется, если в АС не применяются источники бесперебойного питания или средства резервного электроснабжения (это является уязвимостью).

Если говорить об информационных ресурсах, то реализация угрозы может привести к таким последствиям как получение информации людьми, которым она не предназначена, уничтожение или изменение информации, недоступность ресурсов для пользователей. Та-ким образом, мы подошли к определению трех основных угроз безопасности.

Угроза конфиденциальности (угроза раскрытия) – это угроза, в результате реализации которой, конфиденциальная или секретная ин-формация становится доступной лицу, группе лиц или какой-либо организации, которой она не предназначалась. Здесь надо пояснить разницу между секретной и конфиденциальной информацией. В отечественной литературе «секретной» обычно называют информацию, относящуюся к разряду государственной тайны, а «конфиденциальной» –персональные данные, коммерческую тайну и т. п.

Угроза целостности – угроза, в результате реализации которой информация становится измененной или уничтоженной. Необходимо отметить, что и в нормальном режиме работы АС данные могут изменяться и удаляться. Являются ли эти действия легальными или нет, должно определяться политикой безопасности. Политика безопасности – совокупность документированных правил, процедур, практических приемов или руководящих принципов в области безопасности информации, которыми руководствуется организация в своей деятельности.

Угроза отказа в обслуживании (угроза доступности) – угроза, реализация которой приведет к отказу в обслуживании клиентов АС, несанкционированному использованию ресурсов злоумышленниками по своему усмотрению.

Таким образом, безопасность информации – это состояние за-щищенности информации, при котором обеспечены ее конфиденциальность, доступность и целостность. А защита информации может быть определена как деятельность, направленная на предотвращение утечки защищаемой информации, несанкционированных и непреднамеренных воздействий на защищаемую информацию. Выделяются следующие направления защиты информации:

- правовая защита информации – защита информации правовы-ми методами, включающая в себя разработку законодательных и нор-мативных правовых документов (актов), регулирующих отношения субъектов по защите информации, применение этих документов (ак-тов), а также надзор и контроль за их исполнением;
- техническая защита информации – защита информации, заключающаяся в обеспечении не криптографическими методами безопасности информации (данных), подлежащей (подлежащих) защите в соответствии с действующим законодательством, с применением технических, программных и программно-технических средств;
- криптографическая защита информации – защита информации с помощью ее криптографического преобразования;
- физическая защита информации – защита информации путем применения организационных мероприятий и совокупности средств, создающих препятствия для проникновения или доступа неуполномоченных физических лиц к объекту защиты.

Защита информации осуществляется с использованием способов и средств защиты. Способ защиты информации – порядок и правила применения определенных принципов и средств защиты информации.

Средство защиты информации – техническое, программное, программно-техническое средство, вещество и (или) материал, предназначенные или используемые для защиты информации. Отдельно выделяют:

- средства контроля эффективности защиты информации;
- средства физической защиты информации;
- криптографические средства защиты информации.

ОБЩАЯ СХЕМА ПРОЦЕССА ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ

Рассмотрим теперь взаимосвязь основных субъектов и объектов обеспечения безопасности, как это предлагается в международном стандарте ISO/IEC-15408 (в России он принят как ГОСТ Р ИСО/МЭК 15408-2002 [4]).

Безопасность связана с защитой активов от угроз. Разработчики стандарта отмечают, что следует рассматривать все разновидности угроз, но в сфере безопасности наибольшее внимание уделяется тем из них, которые связаны с действиями человека. Рисунок 1.1 иллюстрирует взаимосвязь между высокоуровневыми понятиями безопасности.

За сохранность активов отвечают их владельцы, для которых они имеют ценность. Существующие или предполагаемые нарушители также могут придавать значение этим активам и стремиться использовать их вопреки интересам их владельца. Действия нарушите-лей приводят к появлению угроз. Как уже отмечалось выше, угрозы реализуются через имеющиеся в системе уязвимости. Владельцы активов анализируют возможные угрозы, чтобы определить, какие из них могут быть реализованы в отношении рассматриваемой системы. В результате анализа определяются риски (т. е. события или ситуации, которые предполагают возможность ущерба) и

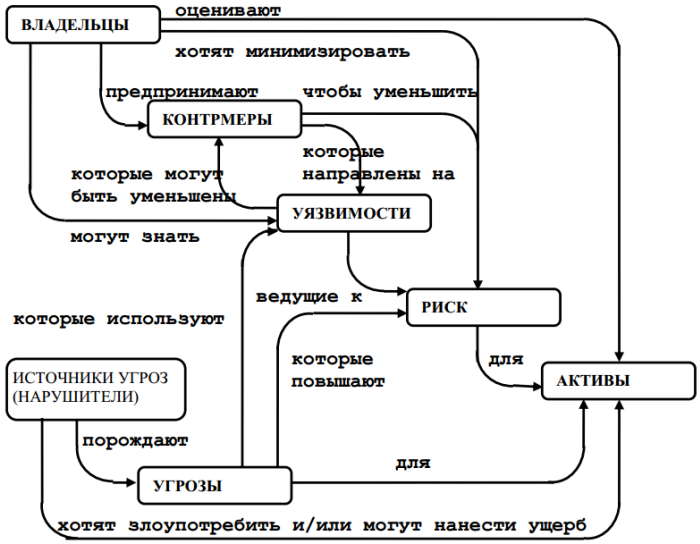


Рис. 1.1. Понятия безопасности и их взаимосвязь

проводится их анализ.

Владельцы актива предпринимают контрмеры для уменьшения уязвимостей и выполнения политики безопасности. Но и после введения этих контрмер могут сохраняться остаточные уязвимости и соответственно – остаточный риск.

ОБЩАЯ ХАРАКТЕРИСТИКА ОРГАНИЗАЦИОННЫХ МЕТОДОВ ЗАЩИТЫ

Законы и нормативные акты исполняются только в том случае, если они подкрепляются организаторской деятельностью соответствующих структур, создаваемых в государстве, в ведомствах, учреждениях и организациях. При рассмотрении вопросов безопасности информации такая деятельность относится к организационным методам защиты информации.

Организационные методы защиты информации включают меры, мероприятия и действия, которые должны осуществлять должностные лица в процессе создания и эксплуатации ИВС для обеспечения заданного уровня безопасности информации.

Организационные методы защиты информации тесно связаны с правовым регулированием в области безопасности информации. В соответствии с законами и нормативными актами в министерствах, ведомствах, на предприятиях(независимо от форм собственно-сти) для защиты информации создаются специальные службы безопасности. Эти службы подчиняются, руководству учреждения. Руководители служб организуют создание и функционирование систем защиты информации. На организационном уровне решаются следующие задачи обеспечения безопасности информации в ИВС:

- организация работ по разработке системы защиты информации;
- ограничение доступа на объект и к ресурсам КС;
- разграничение доступа к ресурсам КС;

- планирование мероприятий;
- разработка документации;
- воспитание и обучение обслуживающего персонала и пользователей;
- сертификация средств защиты информации;
- лицензирование деятельности по защите информации;
- аттестация объектов защиты;
- совершенствование системы защиты информации;
- оценка эффективности функционирования системы защиты информации;
- контроль выполнения установленных правил работы в КС.

Организационные методы являются стержнем комплексной системы защиты ин-формации в КС. Только с помощью этих методов возможно объединение на правовой основе технических, программных и криптографических средств защиты информации в единую комплексную систему. Конкретные организационные методы защиты информации будут приводиться при рассмотрении парирования угроз безопасности информации.

Наибольшее внимание организационным мероприятиям уделяется при изложении вопросов построения и организации функционирования комплексной системы защиты информации.

5.4 Защита программного обеспечения авторским правом

Прежде всего, рассмотрим вопрос о защите программного обеспечения авторским правом.

Авторское право восходит к британскому законодательству начала XVIII века, когда Парламентом был принят так называемый “Статус Анны” (1710), в котором говорилось о “поощрении ученых мужей составлять и писать полезные книги”. Летом 1787 г. на Конституциональном конвенте в Филадельфии была принята Конституция Соединенных Штатов (ратифицирована в июне 1788 г.). В ней было заложено будущее патентное и авторское право. Согласно Конституции, Конгресс имеет право “поощрять развитие наук и полезных искусств, обеспечивая на определенный срок авторам и изобретателям исключительное право на их произведения и открытия” (Конституция США).

Основные положения авторского права устанавливают баланс между общественным интересом и защитой прав автора. С одной стороны, общество нуждается в работах “ученых мужей” во имя процветания, с другой — права автора должны быть защищены для того, чтобы поощрить его к дальнейшей работе. Такую балансировку может обеспечить только очень хорошо продуманное, взвешенное законодательство.

Задолго до принятия Акта об авторском праве 1976 г. были установлены следующие два требования к “произведению”, необходимые для защиты его авторским правом: оригинальность и реализация в материальной форме. Степень “художественности” произведения не играет роли, важно, чтобы оно было собственным произведением автора.

Здесь, однако, возникает вопрос о единственности представления идеи, точнее о запасе возможных представлений идеи. Если идея представляется единственным выражением, то защита выражения равносильна запрету использования идеи. Простая идея имеет небольшой запас выражений, ее представляющих, и они, как таковые, не могут защищаться авторским правом. Поэтому должна быть установлена некая граница, начиная с которой “произведение” защищаемо авторским правом. Это особенно актуально применительно к программам. Ассемблерная программа перемножения двух чисел с фиксированной точкой вряд ли может быть защищена авторским правом. Однако правовое определение границы, начиная с которой программы защищаемы авторским правом, представляет собой непреодолимую трудность.

Провести четкую демаркационную линию между выражением и идеей нельзя. Известно следующее рассуждение судьи Л.Хэнда, так называемый “Абстракционный тест”:

“Любое произведение, особенно пьесу, можно хорошо уложить в последовательность схем, общность которых будет возрастать по мере того, как все больше эпизодов опускается. Последняя из них может, пожалуй, оказаться не более чем общим утверждением, о чем эта пьеса, а иногда может попросту состоять из ее названия; однако в этой серии абстракций имеется пункт, начиная с которого они уже не защищаемы, ибо в противном случае драматург мог бы воспрепятствовать использованию его “идей”, на которые, в отличие от их выражения, его собственность никогда не распространялась”. Ни законодательно, ни прецедентно указать эту демаркационную линию не удалось.

Авторское право обеспечивает автоматическую защиту. Защита авторским правом возникает вместе с созданием произведения независимо от того, предоставил ли автор копию произведения в

Бюро по авторскому праву для регистрации. Однако без регистрации держатель авторского права не может реализовать свои права. Например, он не может возбудить иск о нарушении его права и не может получить возмещение.

Закон подробно оговаривает, в каком виде должны представляться “ копии” программ или баз данных для их регистрации. В случае опубликованной или неопубликованной программы требуется представить один экземпляр “идентифицирующей порции” программы, воспроизведенной в форме, визуально воспринимаемой без помощи машины или какого-либо устройства, на бумаге или на микроформе. Оговаривается, какова эта “порция”. После установления, что представленное произведение защищаемо авторским правом, и просмотра сопровождающих (несложных!) документов Регистр Авторского Права (Register of Copyright) регистрирует требование и выдает автору свидетельство о регистрации.

Авторское право защищает произведение от копирования, но не запрещает независимого создания эквивалентов. Таким образом, риск монополизации знания при использовании авторского права существенно меньше, чем при использовании патентного права и, как следствие, стандарты защиты авторским правом не столь строги, как стандарты защиты патентным правом.

Авторское право США предоставляет автору следующие пять прав:

- воспроизведение;
- подготовка производных произведений;
- распространение копий или звукозаписей;
- публичное исполнение;
- выставка (display).

Авторское право, как уже говорилось, защищает не идею, а ее выражение, конкретную форму представления. Поэтому в основу защиты программ авторским правом кладутся следующие соображения.

Последовательность команд. Программа — это последовательность команд, поэтому она может рассматриваться как “выражение” идеи автора, т.е. как его произведение.

Копирование. Это понятие, используемое в авторском праве, может быть распространено на перенос программ с одного носителя на другой, в том числе — на носитель другого типа (с ленты на диск, в ПЗУ (ROM) и т.п.). Математически это понятие формализуется следующим образом. Пусть имеются виды носителей A и B и процессы “перехода” с одного носителя на другой:

A———>B и B————>A.

Если объект a при переходе с A на B преобразуется в объект b, который при переходе с B на A переходит в прежний объект a, то такой “переход” считается копированием.

Судить об “идентичности” программ на носителях A и B можно по многим признакам, например по их одинаковым функциональным свойствам; однако совпадение функциональных свойств не защищается авторским правом; одинаковость функциональных свойств, как таковая, еще не свидетельствует о воспроизведении “формы”, т.е. о копировании.

Творческая активность. Подобно другим формам фиксации, защищаемым авторским правом, компьютерная программа есть результат творчества. Хотя эта форма выражения или фиксации все еще не является общеизвестной, уровень творческой активности, искусности и изобретательности, необходимый для создания программы, позволяет утверждать, что программы подлежат защите авторским правом не менее, чем любые другие произведения, им защищаемые. Тот факт, что компьютерные программы имеют утилитарное назначение, этого не меняет.

Стиль. Творчество, искусность и изобретательность автора проявляются в том, как создается программа. Необходимо поставить задачи, подлежащие решению. Затем проанализировать, как достичь решения, выбрать цепочки шагов, ведущих к решению; все это должно быть зафиксировано написанием текста программы. Способ, которым все это проделывается, придает программе ее характерные особенности и даже стиль.

Алгоритм. Собственно шаги представляют собой элементы, с помощью которых строится программа, т.е. алгоритмы, не могут защищаться от неавторизованного воспроизведения. Это — аналоги слов в литературе или — мазков кистью в живописи.

Отбор и сопряжение элементов. Как и в случае других произведений, в частности литературных, защита компьютерных программ рассматривается с точки зрения отбора и сопряжения автором этих базовых элементов, в чем и проявляется его творчество и искусность, и что отличает его

произведение от произведений других авторов. Случай, когда два автора независимо друг от друга написали бы для одной и той же цели две идентичные программы, практически исключен. Однако субрутины, которыми пользуются программисты, в основном общеизвестны (их берут в одной и той же операционной среде из единой библиотеки).

Оригинальность программ — первое основное требование авторского права — часто основана на отборе и сопряжении этих общеизвестных элементов.

Удачность. Успех в решении задачи в значительной степени определяется тем отбором элементов, который автор произвел на каждом шаге построения. Поэтому программа может работать быстрее; она проще и надежнее в обращении, легче воспринимается и в целом более производительна, чем ее предшественница или конкуренты.

Эти и другие соображения были положены в основу защиты программ авторским правом. Здесь необходимо было обсудить ряд специфических положений:

- кто является автором произведения;
- что именно защищается (замысел, программа, документация);
- какие именно права гарантируются авторским правом;
- каким должен быть срок действия авторского права применительно к программе;
- в чем должна состоять процедура “регистрации” произведения;
- какие процедуры следует применять в случае нарушения авторского права и др.

Мы не останавливаемся на этих вопросах, а также на вопросах сравнения законодательства по защите программ авторским правом в разных странах и на сравнении этого законодательства с основными международными конвенциями (UCC — Universal Copyright Convention — Всеобщая конвенция по авторскому праву; Буэнос-Айресская и Бернская конвенции).

Детальное рассмотрение их проблем требует отдельной работы. Имеются и более специфические вопросы, хуже осмысленные на сегодня в правовом отношении. К ним относятся: вопрос о форматах данных, используемых при вводе/выводе информации и вопрос об интерфейсе пользователь/программа; а также о структуре и организации программы.

Первая проблема состоит в следующем. Является ли нарушением авторского права использование форматов данных, особенно графических форматов — “экранов”, примененных в программах другого автора. Графические форматы сегодня широко распространены, например в связи со вводом оперативной экономической информации (системы key-to-disk в банковском деле и т.п.).

Вопрос об интерфейсе пользователь/программа получил название “look and feel” — “облик и ощущение”. В какой мере пользовательский интерфейс новой программы выглядит как интерфейс более ранней программы, в какой степени он создает такое же ощущение? Эти вопросы важны, поскольку “удачность” программы связана в первую очередь с “приятным ощущением пользователя”.

Программное обеспечение (software) состоит из трех компонент:

- замысла (основания, подосновы);
- собственно программ;
- сопровождающей документации.

Замысел (подоснова) — это идеи, концепции, алгоритмы, соображения по реализации и т.п.

Программа может выступать в одной из трех форм: исходный, объектный или исполняемый коды.

К документации относятся: руководство по использованию, блок диаграммы, книги по обучению; иногда сложное программное изделие, такое, как операционная система, сопровождается специальным аудиовизуальным курсом обучения.

Мы не рассматриваем здесь аппаратных и программных средств защиты программных изделий (аппаратные ключи, вставляемые в параллельный порт, ключевые дискеты, прожигание отверстий лазером, привязка к аппаратному идентификатору машины и пр.). Мы касаемся только правовой защиты.

Имеется два основных подхода к правовой защите программного обеспечения:

- защита на основе уже существующей правовой системы;
- использование нового законодательства, независимо от существующего.

Правовая защита программного обеспечения по своей проблематике во многом совпадает с более широкой задачей — правовой защитой интеллектуальной собственности.

В настоящее время имеется пять основных правовых механизмов защиты программного обеспечения:

- авторское право;
- патентное право;
- право промышленных тайн;
- право, относящееся к недобросовестным методам конкуренции;
- контрактное право.

Два основных игрока на этой арене — авторское и патентное право. Три последних механизма защиты часто объединяют в одну группу.

Сменяемость компьютерных систем составляет характерную для рынка аппаратных средств величину: 40 мес. При сдаче компьютерной системы в аренду помесячная оплата составляет 1/40 от стоимости системы; эта цифра приводится, например в таких справочниках, как GML Corporation booklet. Через 40 мес. система устаревает и должна быть заменена новой моделью. Никто, по-видимому, не проводил анализа, который позволил бы выяснить, какова “постоянная времени” для сменяемости программных изделий. За 14 лет существования (1976–1990гг.) операционной системы VAX/VMS (корпорация ДЭК) она прошла путь от первой версии до версии 5.3 через многие промежуточные версии (4.5, 4.7 и т.д.). Во всяком случае она претерпела за это время четыре крупных перехода и около 20 мелких. По-видимому, правильной “постоянной времени” для сменяемости программных изделий является 24–30 мес. Эта оценка важна потому, что срок патентования составляет несколько лет (до 5 и более). Так что даже если бы не было никаких правовых трудностей с патентованием программного обеспечения, механизм патентной защиты плохо подходил бы к программному обеспечению.

5. Стандарт ISO/IEC 15408 «Критерии оценки безопасности информационных технологий»

В 1990 году в Международной организацией стандартов (ISO) была начата работа по созданию международных критериев оценки безопасности компьютерных систем. Результатом явился стандарт «Общие критерии безопасности информационных технологий» (ОК), который на данный момент признается одним из наиболее функциональных стандартов в сфере информационной безопасности (ИБ). Его разработка велась совместными усилиями США, Канады, Франции, Германии, Нидерландов и Великобритании. В последствии к проекту присоединился ряд других стран. Версия 2.1 ОК в 1999 году была утверждена в качестве международного стандарта ISO/IEC 15408. В России в настоящее время внедряется адаптированная 3 версия стандарта ISO/IEC 15408.

ОК разработаны таким образом, чтобы удовлетворить потребности трех групп специалистов: разработчиков, экспертов по сертификации и пользователей продуктов информационных технологий (ИТ-продуктов). Под ИТ-продуктом понимается программный (или аппаратно-программный) продукт или информационная система. В процессе оценки ИТ-продукт именуется объектом оценки (ОО). К таким объектам относятся, например, операционные системы, вычислительные сети, распределенные системы, прикладные программы.

Стандарт ISO 15408 состоит из трех частей:

- Часть 1. Введение и общая модель.
- Часть 2. Функциональные требования безопасности.
- Часть 3. Гарантийные требования безопасности (*вариант перевода - "требования гарантированности"*).

Как видно из приведенного перечня, "Общие критерии" предусматривают наличие двух типов требований безопасности - функциональных и гарантированности. Функциональные требования относятся к сервисам безопасности, таким как идентификация, аутентификация, управление доступом, аудит и т.д. Требования гарантированности относятся к технологии разработки, тестированию, анализу уязвимостей, поставке, сопровождению, эксплуатационной документации и т.д.

Описание обоих типов требований выполнено в едином стиле: они организованы в иерархию "класс - семейство - компонент - элемент". Термин "класс" используется для наиболее общей группировки требований безопасности, а элемент - самый нижний, неделимый уровень требований безопасности.

Между компонентами могут существовать зависимости. Они возникают, когда компонент недостаточен для выполнения цели безопасности и необходимо наличие другого компонента.

Промежуточная комбинация компонентов названа пакетом. Пакет включает набор требований, которые обеспечивают выполнение многократно используемого поднабора целей безопасности.

Основные структуры "Общих критериев" - это Профиль защиты и Проект защиты.

По определению стандарта ISO/IEC 15408 Профиль Защиты есть независимое от реализации множество требований безопасности для некоторой категории объектов оценки, которые отвечают определенным нуждам потребителей. Профиль состоит из компонентов или пакетов

функциональных требований и одного из уровней гарантированности. Структура профиля защиты представлена на рис.2.4.

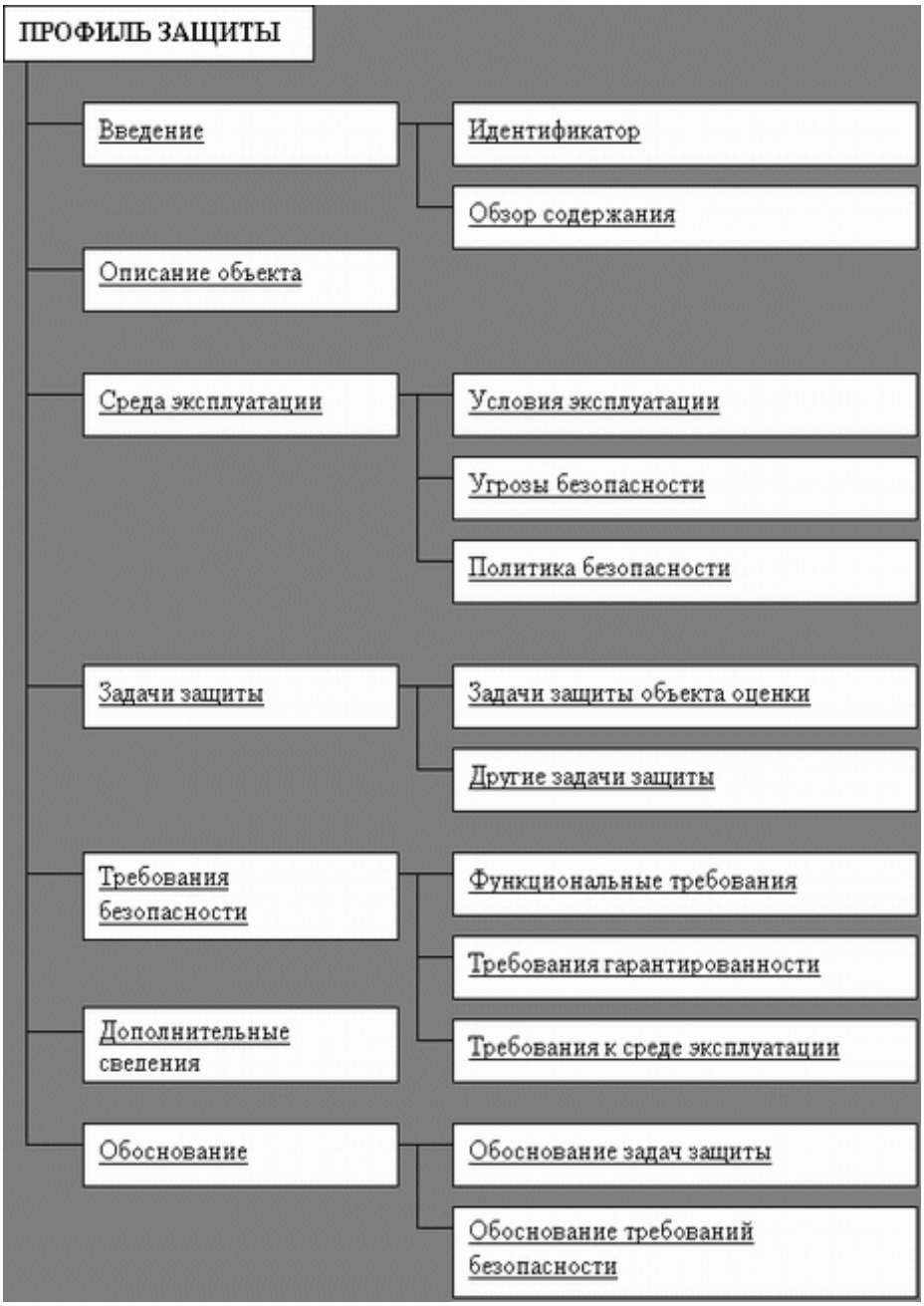


Рис.2.4. Структура профиля защиты.

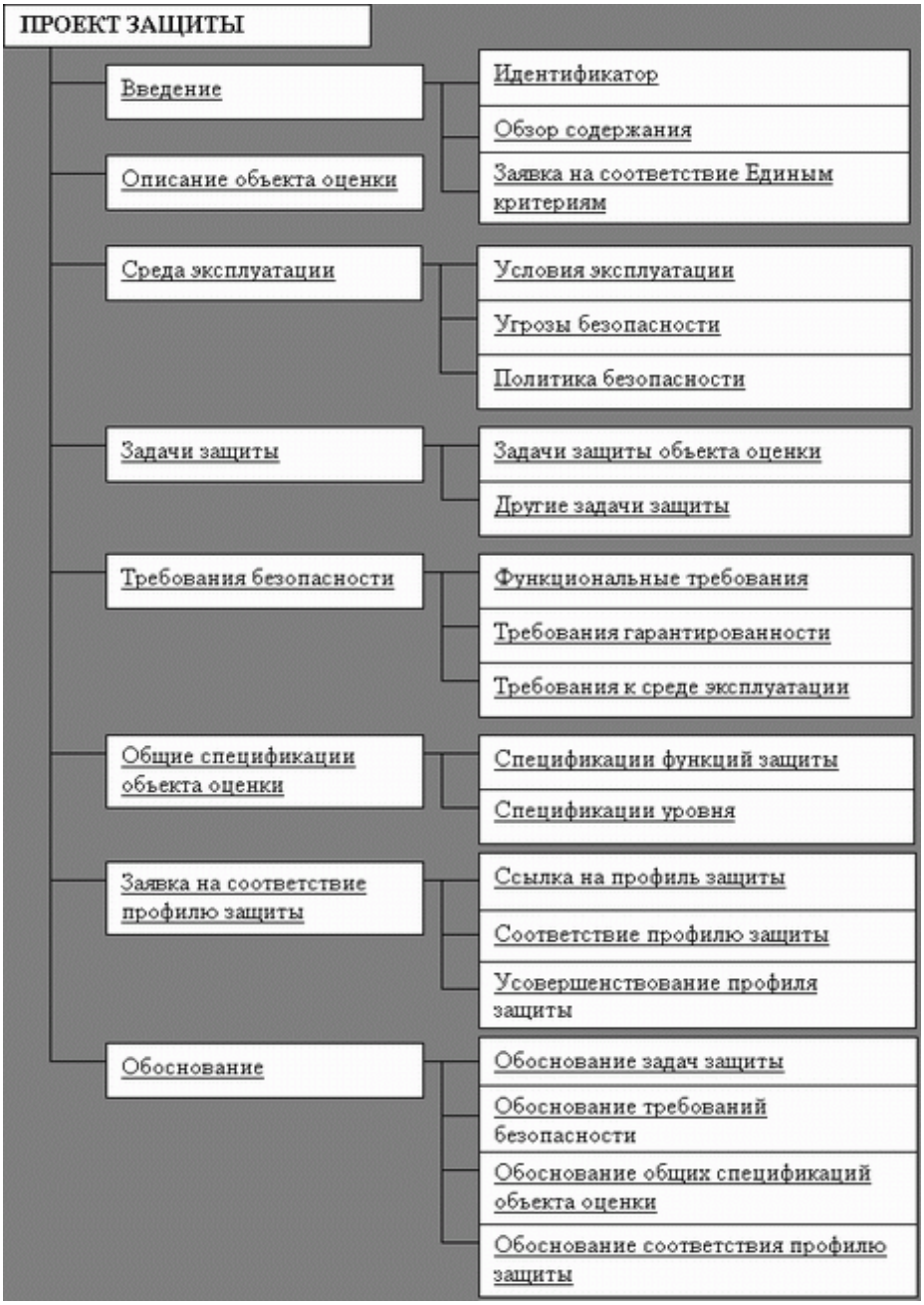


Рис.2.5. Структура Проекта защиты.

Как отмечается в литературе, Профиль защиты "Общих критериев", по сути, является аналогом классов "Оранжевой книги" и классов защищенности РД ГТК РФ, но базируется на значительно более полной и систематизированной совокупности компонентов требований безопасности. Количество стандартизованных профилей общих критериев потенциально не ограничено.

Профиль защиты служит основой для создания Проекта защиты, который является техническим проектом для разработки объекта оценки. Структура Проекта защиты представлена на рис.2.5. В отличие от Профиля, Проект защиты описывает также уровень функциональных возможностей средств и механизмов защиты, реализованных в объекте оценки, и приводит обоснование степени их адекватности.

Для того, чтобы профиль безопасности мог быть эффективно разработан и применен, в процессе его разработки производится выявление всех угроз безопасности осуществимых в

отношении ИТ-продуктов, для которых разрабатывается профиль. В процессе исследования строятся модели угроз.

Модель угрозы - это формальное, полуформальное или неформальное описание:

- жизненного цикла угрозы;
- направленности угрозы;
- источника угрозы;
- системы ИТ, подверженной угрозе;
- ИТ и не-ИТ среды изделия ИТ;
- активов, требующих защиты;
- методов, способов и алгоритмов реализации угрозы;
- нежелательных событий;
    - анализа рисков и ряда других аспектов.У процесса описании модели угроз много общего с проведением анализа рисков. Так при описании модели угроз, источником которых является преднамеренная деятельность человека, оценивается тип источника по уровню практических навыков реализации угрозы (шкала - "низкий", "средний", "высокий", "неопределенный"), и шансы реализации угрозы (шкала - "маловероятно", "вероятно", "большая вероятность", "не определено"). Также оценивается вероятность компрометации активов в виде pc(y,z), где y - идентифицированный метод нападения, z - идентифицированный актив.

В рассмотрение вводится понятие "потенциал нападения". Под этим термином понимается прогнозируемый потенциал для успешного, в случае реализации, нападения, выраженный в показателях компетентности, ресурсов и мотивации нарушителя. Существует три уровня потенциала нападения: низкий, умеренный и высокий.

Стандарт определяет функцию безопасности, как часть или части ОО, на которые возлагается реализация тесно связанного подмножества правил из политики безопасности. Функции безопасности характеризуются стойкостью. Стойкость функции безопасности ОО - это ее характеристика, выражающая минимально необходимое воздействие на ее механизмы безопасности, в результате которого нарушается политика безопасности в части этой функции Выделяется базовая, средняя и высокая стойкость.

Базовая стойкость означает, что функция обеспечивает адекватную защиту от случайного нарушения безопасности ОО нарушителем с низким потенциалом нападения.

Средняя стойкость - функция обеспечивает защиту от целенаправленного нарушения безопасности ОО нарушителем с умеренным потенциалом нападения.

Высокая стойкость - такой уровень стойкости функции безопасности ОО, на котором она обеспечивает защиту от тщательно спланированного и организованного нарушения безопасности ОО нарушителем с высоким потенциалом нападения.

В литературе описана следующая схема вычисления потенциала нападения, в которой учитываются следующие факторы.

1. При идентификации уязвимости:
    - время, затрачиваемое на идентификацию уязвимости (x1) ("за минуты", "за часы", "за дни", "за месяцы");
    - уровень специальной подготовки (x2) ("эксперт", "специалист", "неспециалист");
    - знание проекта и функционирования ОО (x3) ("отсутствие информации об ОО", "общедоступная информация об ОО", "закрытая информация об ОО");

- доступ к ОО (x4) (требуемое время на доступ к ОО, как в случае x1);
- аппаратные средства, программное обеспечение или другое оборудование (x5) ("стандартное оборудование", "специализированное оборудование", "уникальное оборудование").

2. При использовании:

- время, затраченное на использование уязвимости (y1);
- уровень специальной подготовки (y2);
- знание проекта функционирования ОО (y3);
- доступ к ОО (y4);
- аппаратные средства, программное обеспечение или другое оборудование, необходимое для использования уязвимости (y5).

Далее десяти факторам x1-x5 и y1-y5 назначаются веса и они суммируются. Сумма используется для оценки уязвимости.

Описанный подход раскрывает еще одну "грань" задачи анализа рисков, на которой ранее мы не акцентировали внимание - в ходе анализа необходимо не только оценить возможные потери от реализации угрозы, но и описать модель нарушителя, дать оценку его возможностей по реализации угрозы, что в конечном итоге позволит оценить вероятность атаки на систему.

, эксплуатационной документации и т.д.

Описание обоих типов требований выполнено в едином стиле: они организованы в иерархию "клас.4.

6. Задачи идентификации, аутентификации, авторизации, методы их реализации. Методы биометрической аутентификации пользователей

Идентификация призвана каждому пользователю (группе пользователей) сопоставить соответствующую ему разграничительную политику доступа на защищаемом объекте.

Для этого пользователь должен себя идентифицировать – указать своё «имя» (идентификатор). Таким образом ,проверяется, относится ли регистрирующийся пользователь к пользователям, идентифицируемым системой. И в соответствии с введённым идентификатором пользователю будут сопоставлены соответствующие права доступа.

Аутентификация предназначена для контроля процедуры идентификации. Для этого пользователь должен ввести пароль. Правильность вводимого пароля подтверждает однозначное соответствие между регистрирующимся пользователем и идентифицированным пользователем.

В общем случае, идентифицируются и аутентифицируются не только пользователи, но и другие субъекты доступа к ресурсам.

Совокупность выполнения процедур идентификации и аутентификации принято называть процедурой авторизации. Иногда не требуется идентифицировать пользователя, а достаточно только выполнения процедуры аутентификации. Например, это происходит когда требуется подтвердить текущего (уже зарегистрированного) пользователя при выполнении каких-либо действий, требующих дополнительной защиты. В свою очередь, не всегда требуется осуществлять контроль идентификации, то есть в некоторых случаях аутентификация может не производиться.

Процедура авторизации имеет ключевое значение при защите компьютерной информации, т.к. вся разграничительная политика доступа к ресурсам реализуется относительно идентификаторов пользователей. То есть, войдя в систему с чужим идентификатором, злоумышленник получает права доступа к ресурсу того пользователя, идентификатор которого был им предъявлен при входе в систему.

Чтобы исключить работу с системой незаконных пользователей, необходима процедура распознавания системой каждого законного пользователя (или групп пользователей). Для этого в защищенном месте система обязана хранить информацию, по которой можно опознать пользователя, а пользователь при входе в систему, при выполнении определенных действий, при доступе к ресурсам обязан себя идентифицировать, т. е. указать идентификатор, присвоенный ему в данной системе. Получив идентификатор, система проводит его аутентификацию, т. е. проверяет его содержательность (подлинность) - принадлежность к множеству идентификаторов. Если бы идентификация не дополнялась аутентификацией, то сама идентификация теряла бы всякий смысл. Обычно устанавливается ограничение на число попыток предъявления некорректного идентификатора. Аутентификация пользователя может быть основана на следующих принципах:

- на предъявлении пользователем пароля;
- на предъявлении пользователем доказательств, что он обладает секретной ключевой информацией;
- на ответах на некоторые тестовые вопросы;
- на предъявлении пользователем некоторых неизменных признаков, неразрывно связанных с ним;
- на предоставлении доказательств того, что он находится в определенном месте в определенное время;
- на установлении подлинности пользователя некоторой третьей, доверенной стороной.

Процедуры аутентификации должны быть устойчивы к подлогу, подбору и подделке. После распознавания пользователя система должна выяснить, какие права предоставлены этому пользователю, какую информацию он может использовать и каким образом (читать, записывать, модифицировать или удалять), какие программы может выполнять, какие ресурсы ему доступны, а также другие вопросы подобного рода. Этот процесс называется авторизацией. Таким образом, вход пользователя в систему состоит из идентификации, аутентификации и авторизации. В процессе дальнейшей работы иногда может появиться необходимость дополнительной авторизации в отношении каких-либо действий.

Существуют различные механизмы реализации разграничения доступа. Например, каждому ресурсу (или компоненту) системы может быть поставлен в соответствие список управления

доступом, в котором указаны идентификаторы всех пользователей, которым разрешен доступ к данному ресурсу, а также определено, какой именно доступ разрешен. При обращении пользователя к конкретному ресурсу система проверяет наличие у данного ресурса списка управления доступом и, если он существует, проверяет, разрешено ли этому пользователю работать с данным ресурсом в запрошенном режиме. Другим примером реализации механизма авторизации пользователя является профиль пользователя - список, ставящий в соответствие всем идентификаторам пользователей перечень объектов, к которым разрешен доступ данному пользователю, с указанием типа доступа. Может быть организована системная структура данных, так называемая матрица доступа, которая представляет собой таблицу, столбцы которой соответствуют идентификаторам всех системных ресурсов, а строки - идентификаторам всех зарегистрированных пользователей. На пересечении i-го столбца j-й строки таблицы администратор системы указывает разрешенный тип доступа владельца i-го идентификатора j-му ресурсу. Доступ к механизмам авторизации должны иметь только специальные системные программы, обеспечивающие безопасность системы, а также строго ограниченный круг пользователей, отвечающих за безопасность системы. Рассматриваемые механизмы должны быть тщательно защищены от случайного или преднамеренного доступа неавторизованных пользователей. Многие атаки на информационные системы нацелены именно на вывод из строя или обход средств разграничения доступа. Аналогичные действия осуществляются в системе и при аутентификации других субъектов взаимодействия (претендентов), например прикладных процессов или программ, с системой (верификатором). В отличие от аутентификации субъекта взаимодействия, процедура аутентификации объекта, устанавливая подлинность электронной почты, банковского счета и т. п., проверяет факт принадлежности данного объекта владельцу указанного идентификатора.

Требования к идентификации и аутентификации

Формализованные требования к данным механизмам защиты состоят в следующем:

• Должны осуществляться идентификация и проверка подлинности субъектов доступа при входе в систему по идентификатору (коду) и паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов

• Система защиты должна требовать от пользователей идентифицировать себя при запросах на доступ.

• Система защиты должна подвергать проверке подлинность идентификации — осуществлять аутентификацию. Для этого она должна располагать необходимыми данными для идентификации и аутентификации.

• Система защиты должна препятствовать доступу к защищаемым ресурсам неидентифицированных пользователей и пользователей, подлинность идентификации которых при аутентификации не подтвердилась (для 5 класса защищенности по классификации СВТ).

Кроме ограничения «...паролю условно-постоянного действия длиной не менее шести буквенно-цифровых символов... » данные требования никак не формализуют подходы к реализации механизмов парольной защиты. Кроме того, данные требования не определяют, каким образом должны быть реализованы механизмы парольной защиты, а также не накладывают дополнительных ограничений, связанных с повышением стойкости пароля к подбору. В частности, они не регламентируют использование внешних носителей парольной информации — дискет, смарт-карт и т.д.

Дополнительные требования:

Существует целая группа угроз, связанная с некорректностью реализации процедуры авторизации в современных ОС, а также с наличием ошибок в реализации соответствующих механизмов защиты. Это обусловливает целесообразность рассмотрения механизмов авторизации с целью их добавочной защиты. Кроме того, механизмы идентификации и аутентификации являются важнейшими для противодействия НСД к информации, а значит, следует рассматривать возможные варианты их резервирования.

Кроме того, в рамках декларируемого системного подхода к проектированию системы защиты, при разработке механизмов авторизации следует рассматривать как явные, так и скрытые угрозы преодоления защиты.

Авторизация в контексте количества и вида зарегистрированных пользователей

Кого следует воспринимать в качестве потенциального злоумышленника/

1. В системе зарегистрирован один пользователь

Данный пользователь является и прикладным пользователем, и администратором безопасности. Здесь источником потенциальной угрозы является только сторонний сотрудник предприятия, а вся задача защиты сводится к контролю доступа в компьютер (либо в систему), т.е. к парольной защите.

2. В системе зарегистрированы администратор безопасности и один прикладной пользователь

Общий случай функционирования системы с одним прикладным пользователем — это наличие в системе администратора безопасности и только одного прикладного пользователя. В задачи администратора безопасности здесь входит ограничение прав прикладного пользователя по доступу к системным (администратора безопасности) и иным ресурсам компьютера. В частности, может ограничиваться набор задач, разрешенных для решения на компьютере, набор устройств, которые могут быть подключены к компьютеру (например, внешний модем, принтер и т.д.), способ сохранения обрабатываемых данных (например, на дискетах только в шифрованном виде) и т.д.

В данном случае потенциальным злоумышленником в части несанкционированного использования ресурсов защищаемого объекта может являться как сторонний сотрудник предприятия, так и собственно прикладной пользователь. Заметим, что прикладной пользователь здесь может выступать в роли сознательного нарушителя, либо стать «инструментом» в роли стороннего нарушителя, например, запустив по чьей-либо просьбе какую-нибудь программу).

3. В системе зарегистрированы администратор безопасности и несколько прикладных пользователей

Кроме администратора безопасности, в системе может быть заведено несколько прикладных пользователей. При этом ресурсами защищаемого компьютера могут пользоваться несколько сотрудников, решая различные задачи. Ввиду этого информационные и иные ресурсы защищаемого объекта должны между ними разграничиваться.

В данном случае к потенциальным нарушителям добавляется санкционированный прикладной пользователь, целью которого может служить НСД к информации, хранимой на защищаемом объекте другим пользователем.

Рекомендации по построению авторизации, исходя из вида и количества зарегистрированных пользователей

Наиболее простой в реализации защитой является защита от стороннего сотрудника. В этом случае все мероприятия по защите возлагаются на использование механизма парольного входа.

Простота состоит в том, что, как увидим далее, в этом случае следует оказывать противодействие только явным угрозам преодоления парольной защиты, от которых защититься не представляет большого труда.

Однако основной угрозой служат преднамеренные или неумышленные действия санкционированного пользователя, который обладает возможностью осуществления скрытой атаки на защищаемый ресурс (например, запустив какую-либо программу собственной разработки).

Механизмы идентификации и аутентификации должны предусматривать противодействие всем потенциальным злоумышленникам, т.е. как сторонним по отношению к защищаемому объекту, так и санкционированным пользователям, зарегистрированным на компьютере. При этом речь идет о прикладных пользователях, т.к. осуществить какую-либо защиту от НСД к информации от администратора безопасности невозможно, даже включая применение механизмов криптографической защиты (он сумеет снять информацию до момента ее поступления в драйвер шифрования).

С учетом сказанного можем сделать следующие выводы:

1. На защищаемом объекте, как правило, зарегистрированы, по крайней мере, два пользователя — прикладной пользователь и администратор безопасности. Поэтому в качестве потенциального злоумышленника при реализации механизмов парольной защиты в общем случае следует рассматривать не только стороннее по отношению к защищаемому объекту лицо, но и санкционированного пользователя, который преднамеренно либо неумышленно может осуществить атаку на механизм парольной защиты.

2. Рассматривая атаки на парольную защиту следует учитывать, что по сравнению со сторонним лицом, которое может характеризоваться явными угрозами парольной защите, защита от

атак санкционированного пользователя качественно сложнее, т.к. им могут быть реализованы скрытые угрозы.

МЕТОДЫ БИОМЕТРИЧЕСКОЙ АУТЕНТИФИКАЦИИ

В настоящее время существует множество методов биометрической аутентификации, которые делятся на две группы, рассмотренные ниже.

Статические методы

Статические методы биометрической аутентификации основываются на физиологической (статической) характеристике человека, то есть уникальной характеристике, данной ему от рождения и неотъемлимой от него. Рассмотрим их ниже:

- Аутентификация по отпечатку пальца. Все существующие на сегодняшний день сканеры отпечатков пальцев по используемым ими физическим принципам можно выделить в три группы:

- - оптические;
- - кремниевые (или полупроводниковые);
- - ультразвуковые.

В основе работы оптических сканеров лежит оптический метод получения изображения. По видам используемых технологий можно выделить следующие группы оптических сканеров.

FTIR-сканеры - устройства, в которых используется эффект нарушенного полного внутреннего отражения (Frustrated Total Internal Reflection, FTIR).

Существуют модификации описанного сканера, в которых каждый полупроводниковый элемент в матрице сканера выступает в роли одной пластины конденсатора, а палец - в роли другой. При приложении пальца к сенсору между каждым чувствительным элементом и выступом-впадиной папиллярного узора образуется некая емкость, величина которой определяется расстоянием между поверхностью пальца и элементом. Матрица этих емкостей преобразуется в изображение отпечатка пальца.

Чувствительные к давлению сканеры (pressure scanners) - в этих устройствах используются сенсоры, состоящие из матрицы пьезоэлементов.

Данные типы сканеров являются самыми распространенными. Во всех приведенных полупроводниковых сканерах используются матрица чувствительных микроэлементов (тип которых определяется способом реализации) и преобразователь их сигналов в цифровую форму. Таким образом, обобщенно схему работы приведенных полупроводниковых сканеров можно продемонстрировать следующим образом.

Радиочастотные сканеры (RF-Field scanners) -- в таких сканерах используется матрица элементов, каждый из которых работает как маленькая антенна.

Из достоинств можно выделить следующие.

Пользователю не нужно запоминать логин-пароль. В некоторых случаях это позволяет избавиться от шпаргалок на мониторе или под клавиатурой. Если же сканер встроен в мышь, то можно проводить незаметную идентификацию довольно часто.

Малая вероятность подделки (соотношение цена/надёжность очень высоко).

Малые размеры сканеров (можно сделать размером со щель 1х10мм и даже меньше) позволяют размещать их в мобильных устройствах. Сейчас люди используют смартфоны и флешки для хранения конфиденциальной информации - сканер отпечатков оказывается неплохой защитой (если использовать разумно). В случае со смартфонами можно обеспечить защиту от несанкционированного использования в случае "гоп-стопа".

Из недостатков возможно выделить следующие.

Пользователи считают, что их отпечатки пальцев могут использоваться в криминалистике (впрочем, иногда это так и есть).

В случае сильного ожога или множественных порезов, идентификация пользователя становится невозможной.

Зависимость от чистоты пальца.

Для сухой кожи качество распознавания ниже.

- Аутентификация по радужной оболочке глаза. Считается, что технология аутентификации по радужной оболочке глаза произошла от еще одной очень известной технологии - аутентификации по сетчатке глаза. Ученые провели ряд исследований, которые показали, что сетчатка глаза человека может меняться со временем, в то время как радужная оболочка глаза остается неизменной.

Невозможно найти два абсолютно идентичных рисунка радужной оболочки глаза, даже у близнецов. Очки и контактные линзы, даже цветные, никак не повлияют на процесс получения изображения. Также нужно отметить, что произведенные операции на глазах, удаление катаракты или вживление имплантатов роговицы не изменяют характеристики радужной оболочки, ее невозможно изменить или модифицировать. Слепой человек также может быть идентифицирован при помощи радужной оболочки глаза. Пока у глаза есть радужная оболочка, ее хозяина можно идентифицировать, что проиллюстрировано на рисунке 2.

Камера может быть установлена на расстоянии от 10 см до 1 метра, в зависимости от сканирующего оборудования. Термин «сканирование» может быть обманчивым, так как в процессе получения изображения проходит не сканирование, а простое фотографирование.

Радужная оболочка по текстуре напоминает сеть с большим количеством окружающих кругов и рисунков, которые могут быть измерены компьютером. Программа сканирования радужной оболочки глаза использует около 260 точек привязки для создания образца. Для сравнения, лучшие системы идентификации по отпечаткам пальцев используют 60-70 точек.

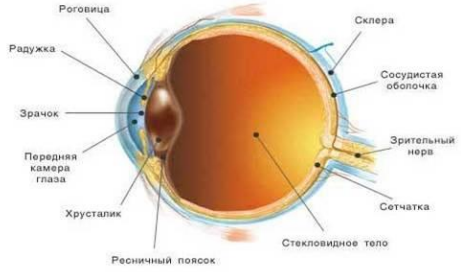


Рисунок 2 - Строение глаза, анализ участков

Стоимость всегда была самым большим сдерживающим моментом перед внедрением технологии, но сейчас системы идентификации по радужной оболочке становятся более доступными для различных компаний. Сторонники технологии заявляют о том, что распознавание радужной оболочки глаза очень скоро станет общепринятой технологией идентификации в различных областях.

- Аутентификация по геометрии руки. В биометрике в целях идентификации человека большое распространение получил метод аутентификации по геометрии руки. Ключевыми признаками здесь являются размер, форма руки, а также определенные информационные знаки на тыльной стороне руки.

Существует два основных подхода к использованию геометрических характеристик кисти руки. Первый из этих подходов основан чисто на геометрических характеристиках руки. Второй же вводит еще и образцовые характеристики руки (образы на сгибах между фалангами пальцев и узоры кровеносных сосудов проиллюстрированы на рисунке 3).

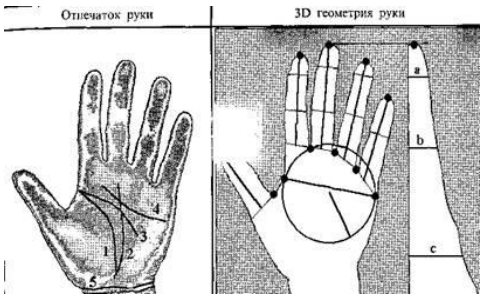


Рисунок 3 - Образы сгибов между фалангами пальцев и узоры кровеносных сосудов

Основными геометрическими признаками являются: ширина ладони, радиус вписанной в ладонь окружности, длины пальцев, ширина пальцев, высота кисти руки в трёх местах.

Все эти признаки объединяются в так называемый вектор значений. Метод идентификации по вектору значений достаточно прост. В начале с пользователя снимают несколько силуэтов его руки. Для каждого из этих силуэтов формируется свой вектор значений. На основе нескольких векторов значений создается специальный класс. Далее все признаки в классе усредняются, и получаются признаки эталонного образа (или, говорят, находится центр класса). В процессе работы исходные образы могут модифицироваться. При сравнении нового образа с эталоном, в случае успеха он может быть помещен в класс исходных признаков. Сравнивать же между собой два образа можно по нескольким критериям. Наиболее очевидный из них - наименьшее расстояние от исследуемого образа до эталона. Более сложный метод - снимать четыре характеристики, три из которых - характерные размеры, а четвертая - полутоновое изображение складок кожи на сгибе между фалангами. Такой метод сильно затрудняет обман прибора. Стоит отметить, что в принципе более подробной

информации по используемым характеристикам и алгоритмах сравнения найти не удастся, потому что компании, занимающиеся распознаванием по руке, не разглашают эту информации из соображений защиты от обмана их устройств.

В заключении стоит отметить, что метод идентификации по геометрии руки, построенный с использованием полутонового изображения обладает высокой надежностью. Кроме того, сканеры геометрии рук не выдвигают никаких требований к характеристикам рук (чистоте, температуре рук) и не наводят пользователей на мысли о криминалистике, как в случае сканеров отпечатков пальцев.

Достоинства метода.

- «Ключ» всегда с пользователем.
- Не предъявляются требования к чистоте, влажности, температуре рук.
- Пользователь не стесняется "криминалистического" уклона технологии.

Недостатки метода.

- Громоздкость устройств (за некоторым исключением).
- Невысокая сложность изготовления муляжа для устройств первого типа (использующих только геометрические характеристики).
- Аутентификация по геометрии лица. Система распознавания по лицу - наиболее древний и распространенный способ идентификации. Именно такой процедуре подвергается каждый, кто пересекает границу. При этом пограничник сверяет фото на паспорте с лицом владельца паспорта и принимает решение, его это паспорт или нет. Примерно такую же процедуру выполняет компьютер, но с той лишь разницей, что фото уже находится в его памяти. Привлекательность данного метода основана на том, что он наиболее близок к тому, как мы идентифицируем друг друга. Развитие данного направления обусловлено быстрым ростом мультимедийных видеотехнологий, благодаря которым можно увидеть все больше видеокамер, установленных дома и на рабочих местах.

Существенный импульс это направление получило с повсеместным распространением технологии видеоконференций Internet/intranet. Ориентация на стандартные видеокамеры персональных компьютеров делает этот класс биометрических систем сравнительно дешевым. Тем не менее, идентификация человека по геометрии лица представляет собой достаточно сложную (с математической точки зрения) задачу. Хотя лицо человека - уникальный параметр, но достаточно динамичный; человек может улыбаться, отпускать бороду и усы, надевать очки - все это добавляет трудности в процедуре идентификации и требует достаточно мощной и дорогой аппаратуры, что соответственно влияет на степень распространенности данного метода.

Алгоритм функционирования системы опознавания достаточно прост. Изображение лица считывается обычной видеокамерой и анализируется. Программное обеспечение сравнивает введенный портрет с хранящимся в памяти эталоном. Некоторые системы дополнительно архивируют вводимые изображения для возможного в будущем разбора конфликтных ситуаций. Весьма важно также то, что биометрические системы этого класса потенциально способны выполнять непрерывную идентификацию (аутентификацию) пользователя компьютера в течение всего сеанса его работы. Большинство алгоритмов позволяет компенсировать наличие очков, шляпы и бороды у исследуемого индивида. Было бы наивно предполагать, что с помощью подобных систем можно получить очень точный результат. Несмотря на это, в некоторых странах они довольно успешно используются для верификации кассиров и пользователей депозитных сейфов.

Основными проблемами, с которыми сталкиваются разработчики данного класса биометрических систем, являются изменение освещенности, вариации положения головы пользователя, выделение информативной части портрета (гашение фона). С этими проблемами удается справиться, автоматически выделяя на лице особые точки и затем измеряя расстояния между ними. На лице выделяют контуры глаз, бровей, носа, подбородка. Расстояния между характерными точками этих контуров образуют весьма компактный эталон конкретного лица, легко поддающийся масштабированию. Задача оконтуривания характерных деталей лица легко может быть решена для плоских двухмерных изображений с фронтальной подсветкой, но такие биометрические системы можно обмануть плоскими изображениями лица-оригинала. Для двухмерных систем изготовление муляжа-фотографии - это не сложная техническая задача.

Существенные технические трудности при изготовлении муляжа возникают при использовании трехмерных биометрических систем, способных по перепадам яркости отраженного света восстанавливать трехмерное изображение лица. Такие системы способны компенсировать

неопределенность расположения источника освещенности по отношению к идентифицируемому лицу, а также неопределенность положения лица по отношению к видеокамере. Обмануть системы этого класса можно только объемной маской, точно воспроизводящей оригинал.

Данный метод обладает существенным преимуществом: для хранения данных об одном образце идентификационного кода (одном лице) требуется совсем немного памяти. А все потому, что, как выяснилось, человеческое лицо можно поделить на относительно небольшое количество «блоков», неизменных у всех людей. Этих блоков больше, чем известных нам частей лица, но современная техника научилась выделять их и строить на их основе модели, руководствуясь взаимным расположением блоков.

Технология идентификации геометрии лица может использоваться, в частности, для такой экзотической цели, как слежение. Алгоритм позволяет выделять изображение лица на некотором расстоянии и на любом фоне, даже состоящем из других лиц, чтобы затем сравнить его с хранящимся в памяти эталонным кодом. Система была испытана для выявления преступников на чемпионате США по американскому футболу. Факт применения этой системы скрывали до конца чемпионата, и зрители пришли в негодование от такого посягательства на демократические свободы. Технология состояла в преобразовании фотографии лица в математическое выражение, описывающее геометрию его черт. Система переводила изображение в 84-разрядный файл, называемый face print. Затем файлы, полученные при помощи видеокамер во время матчей, сравнивались с face print известных преступников. Хотя несанкционированное применение такой технологии, равно как и сама технология, подверглись осуждению со стороны общественности, правоохранительные органы ряда городов уже выделили средства для ее развертывания.

Программа One-on-One, используя камеру, распознает лица и обеспечивает «ненавязчивый» контроль над пользователем. При инсталляции системы пользователь должен зарегистрировать свое лицо в базе данных. В результате этой процедуры One-onOne создаст цифровой шаблон (подпись), связанный с изображением лица. При дальнейшем использовании системы она будет проверять, совпадает ли изображение лица (вернее -- шаблон) пользователя с хранящимся в базе.

Наличие косметики не влияет на работу системы распознавания, которая распознает людей даже в тех случаях, когда они решили отказаться от очков.

One-on-One не сохраняет изображение лица. Поэтому компьютерный взломщик не может реконструировать изображение по учетной записи в базе данных.

NVisage - это наиболее продвинутая разработка Cambridge Neurodynamics. Уникальность продукта заключается в том, что он ориентирован на распознавание трехмерных объектов, в то время как в большинстве современных устройств используется только двухмерная техника.

Более надежной разновидностью описываемого метода является идентификация по «тепловому портрету» лица или тела человека в инфракрасном диапазоне. Этот метод, в отличие от обычного, оптического, не зависит от изменений лица человека (например, появления бороды), так как тепловая картина лица меняется крайне редко. Недавно появилось сообщение об устройствах Technology Recognition Systems (США), в которых происходит распознавание лица в инфракрасном свете. Данная технология основана на том, что термограмма лица человека (тепловая картинка, созданная излучением тепла кровеносными сосудами лица) уникальна для каждого человека и, следовательно, может быть использована в качестве биокода для систем контроля допуска. Данная термограмма является более стабильным кодом, чем геометрия лица, поскольку не зависит от времени и изменений внешности человека.

7. Общие подходы к построению парольных систем и основные угрозы их безопасности

Наиболее распространенные методы аутентификации основаны на применении многоразовых или одноразовых паролей. Из-за своего широкого распространения и простоты реализации парольные схемы часто в первую очередь становятся мишенью атак злоумышленников. Эти методы включают следующие разновидности способов аутентификации:

- *по хранимой копии пароля или его свёртке;*
- *по некоторому проверочному значению;*
- *без непосредственной передачи информации о пароле проверяющей стороне;*
- *с использованием пароля для получения криптографического ключа.*

В первую разновидность способов входят системы аутентификации, предполагающие наличие у обеих сторон копии пароля или его свертки. Для организации таких систем требуется создать и поддерживать базу данных, содержащую пароли или сверки паролей всех пользователей. Их слабой стороной является то, что получение злоумышленником этой базы данных позволяет ему проходить аутентификацию от имени любого пользователя.

Способы, составляющие вторую разновидность, обеспечивают более высокую степень безопасности парольной системы, так как проверочные значения, хотя они и зависят от паролей, не могут быть непосредственно использованы злоумышленником для аутентификации.

Наконец, аутентификация без предоставления проверяющей стороне какой бы то ни было информации о пароле обеспечивает наибольшую степень защиты. Этот способ гарантирует безопасность даже в том случае, если нарушена работа проверяющей стороны (например, в программу регистрации в системе внедрен "троянский конь"). Пример системы парольной защиты ("доказательство с нулевым разглашением"), построенной по данному принципу, будет рассмотрен ниже.

Особым подходом в технологии проверки подлинности являются криптографические протоколы аутентификации. Такие протоколы описывают последовательность действий, которую должны совершить стороны для взаимной аутентификации, кроме того, эти действия, как правило, сочетаются с генерацией и распределением криптографических ключей для шифрования последующего информационного обмена. Корректность протоколов аутентификации вытекает из свойств задействованных в них математических и криптографических преобразований и может быть строго доказана.

Обычные парольные системы проще и дешевле для реализации, но менее безопасны, чем системы с криптографическими протоколами. Последние обеспечивают более надежную защиту и дополнительно решают задачу распределения ключей. Однако используемые в них технологии могут быть объектом законодательных ограничений.

Для более детального рассмотрения принципов построения парольных систем сформулируем несколько основных определений.

- *Пароль пользователя* - некоторое секретное количество информации, известное только пользователю и парольной системе, которое может быть запомнено пользователем и предъявлено для прохождения процедуры аутентификации. Одноразовый пароль дает возможность пользователю однократно пройти аутентификацию. Многоразовый пароль может быть использован для проверки подлинности повторно.
- *Учетная запись пользователя* - совокупность его идентификатора и его пароля.
- *База данных пользователей* парольной системы содержит учетные записи всех пользователей данной парольной системы.
- Под *парольной системой* будем понимать программно-аппаратный комплекс, реализующий системы идентификации и аутентификации пользователей КС на основе одноразовых или многоразовых паролей. Как правило, такой комплекс функционирует совместно с подсистемами разграничения доступа и регистрации событий. В отдельных случаях парольная система может выполнять ряд дополнительных функций, в частности генерацию и распределение кратковременных (сеансовых) криптографических ключей.

Основными компонентами парольной системы являются:

- интерфейс пользователя;
- интерфейс администратора;
- модуль сопряжения с другими подсистемами безопасности;
- база данных учетных записей.

Парольная система представляет собой "передний край обороны" всей системы безопасности. Некоторые ее элементы (в частности, реализующие интерфейс пользователя) могут быть расположены в местах, открытых для доступа потенциальному злоумышленнику. Поэтому парольная система становится одним из первых объектов атаки при вторжении злоумышленника в защищенную систему. Ниже перечислены *типы угроз безопасности парольных систем*.

1. Разглашение параметров учетной записи через:

- подбор в интерактивном режиме;
- подсматривание;
- преднамеренную передачу пароля его владельцем другому лицу;
- захват базы данных парольной системы (если пароли не хранятся в базе в открытом виде, для их восстановления может потребоваться подбор или дешифрование);
- перехват переданной по сети информации о пароле;
- хранение пароля в доступном месте.

2. Вмешательство в функционирование компонентов парольной системы через:

- внедрение программных закладок;
- обнаружение и использование ошибок, допущенных на стадии разработки;
- выведение из строя парольной системы.

Некоторые из перечисленных типов угроз связаны с наличием так называемого человеческого фактора, проявляющегося в том, что пользователь может:

- выбрать пароль, который легко запомнить и также легко подобрать;
- записать пароль, который сложно запомнить, и положить запись в доступном месте;
- ввести пароль так, что его смогут увидеть посторонние;
- передать пароль другому лицу намеренно или под влиянием заблуждения.

В дополнение к выше сказанному необходимо отметить существование *"парадокса человеческого фактора"*. Заключается он в том, что пользователь нередко стремится выступать скорее противником парольной системы, как, впрочем, и любой системы безопасности, функционирование которой влияет на его рабочие условия, нежели союзником системы защиты, тем самым ослабляя ее.

В большинстве систем пользователи имеют возможность самостоятельно выбирать пароли или получают их от системных администраторов. При этом для уменьшения деструктивного (разрушительного) влияния описанного выше человеческого фактора необходимо реализовать ряд требований к выбору и использованию паролей (табл.1).

Требование к выбору пароля	Получаемый эффект
Установление минимальной длины пароля	Усложняет задачу злоумышленника при попытке подсмотреть пароль или подобрать пароль методом "тотального опробования" ***Метод полного (тотального) опробования ключей-*** *Метод анализа криптографического, состоящий в переборе всех возможных ключей криптосистемы с отбраковкой ложных вариантов по некоторому критерию.*
Использование в пароле различных групп символов	Усложняет задачу злоумышленника при попытке подобрать пароль методом "тотального опробования"
Проверка и отбраковка пароля по словарю	Усложняет задачу злоумышленника при попытке подобрать пароль по словарю
Установление максимального срока действия пароля	Усложняет задачу злоумышленника по подбору паролей методом тотального опробования, в том числе без непосредственного обращения к системе защиты (режим off-line)
Установление минимального срока действия пароля	Препятствует попыткам пользователя заменить пароль на старый после его смены по предыдущему требованию

Ведение журнала истории паролей	Обеспечивает дополнительную степень защиты по предыдущему требованию "
Применение эвристического алгоритма, бракующего пароли на основании данных журнала истории	Усложняет задачу злоумышленника при попытке подобрать пароль по словарю или с использованием эвристического алгоритма
Ограничение числа попыток ввода пароля	Препятствует интерактивному подбору паролей злоумышленником
Поддержка режима принудительной смены пароля пользователя	Обеспечивает эффективность требования, ограничивающего максимальный срок действия пароля
Использование задержки при вводе неправильного пароля	Препятствует интерактивному подбору паролей злоумышленником
Запрет на выбор парс-ля самим пользователем и автоматическая генерация паролей	Исключает возможность подобрать пароль по словарю. Если алгоритм генерации паролей не известен злоумышленнику, последний может подбирать пароли только методом "тотального опробования"
Принудительная смена пароля при первой регистрации пользователя в системе	Защищает от неправомерных действии системного администратора, имеющего доступ к паролю в момент создания учетной записи

Параметр	Способ определения
Мощность алфавита паролей *A*	Могут варьироваться для обеспечения заданного значения $S(S=A^L)$
Длина пароля L	
Мощность пространства паролей S ′	Вычисляется на основе заданных значений P, T или V
Скорость подбора паролей V: • Для интерактивного режима определяется как скорость обработки одной попытки регистрации проверяющей стороной. • Для режима off-line (на основе свертки пароля) определяется как скорость вычисления значения свертки для одного пробного пароля	• Может быть искусственно увеличена для защиты от данной угрозы. • Задается используемым алгоритмом вычисления свертки. Алгоритм, имеющий медленные реализации, повышает стойкость по отношению к данной угрозе
Срок действия пароля (задает промежуток времени, по истечении которого пароль должен быть обязательно сменен) T	Определяется исходя из заданной вероятности P, или полагается заданным для дальнейшего определения S
Вероятность подбора пароля в течение его срока действия (подбор продолжается непрерывно в течение всего срока действия пароля) P	Выбирается заранее для дальнейшего определения S или T

В качестве иллюстрации рассмотрим задачу определения минимальной мощности пространства паролей (зависящей от параметров A и L) в соответствии с заданной вероятностью подбора пароля в течение его срока действия.

Задано $P=10^{-6}$. Необходимо найти минимальную длину пароля, которая обеспечит его стойкость в течение одной недели непрерывных попыток подобрать пароль. Пусть скорость интерактивного подбора паролей V=10 паролей/мин. Тогда в течение недели можно перебрать

$10 \cdot 60 \cdot 24 \cdot 7 = 100800$ паролей.

Далее, учитывая, что параметры S, V, T и P связаны соотношением P $P=V \cdot T/S$, получаем

$S=100 \cdot 800/10^{-6}=1{,}008 \cdot 10^{11} \approx 10^{11}$

Полученному значению S соответствуют пары: A=26, L=8 и A=36, L=6.

Другим важным аспектом стойкости парольной системы, является способ хранения паролей в базе данных учетных записей. Возможны следующие варианты хранения паролей:

- в открытом виде;
- в виде свёрток (хеширование);
- зашифрованными на некотором ключе.

Наибольший интерес представляют второй и третий способы, которые имеют ряд особенностей.

Хеширование не обеспечивает защиту от подбора паролей по словарю в случае получения базы данных злоумышленником. При выборе алгоритма хеширования, который будет использован для вычисления сверток паролей, необходимо гарантировать несовпадение значений сверток, полученных на основе различных паролей пользователей. Кроме того, следует предусмотреть механизм, обеспечивающий уникальность сверток в том случае, если два пользователя выбирают одинаковые пароли. Для этого при вычислении каждой свертки обычно используют некоторое количество "случайной" информации, например, выдаваемой генератором псевдослучайных чисел.

При шифровании паролей особое значение имеет способ *генерации и хранения ключа шифрования* базы данных учетных записей. Перечислим некоторые возможные варианты:

- ключ генерируется программно и хранится в системе, обеспечивая возможность ее автоматической перезагрузки;
- ключ генерируется программно и хранится на внешнем носителе, с которого считывается при каждом запуске;
- ключ генерируется на основе выбранного администратором пароля, который вводится в систему при каждом запуске.

Процедура опознавания с использованием простого пароля может быть представлена в виде следующей последовательности действий:

- пользователь посылает запрос на доступ к компьютерной системе и вводит свой идентификатор;
- система запрашивает пароль;
- пользователь вводит пароль;
- система сравнивает полученный пароль с паролем пользователя, хранящимся в базе эталонных данных системы защиты, и разрешает доступ, если пароли совпадают; в противном случае пользователь к ресурсам компьютерной системы не допускается.

Поскольку пользователь может допустить ошибку при вводе пароля, то системой должно быть предусмотрено допустимое количество повторений для ввода пароля.

В базе эталонных данных пароли, как и другую информацию, никогда не следует хранить в явной форме, а только зашифрованными. При этом можно использовать метод как обратимого, так и необратимого шифрования.

Согласно методу *обратимого шифрования* эталонный пароль при занесении в базу эталонных данных зашифровывается по ключу, совпадающему с этим эталонным паролем, а введенный после идентификации пароль пользователя для сравнения с эталонным также зашифровывается по ключу, совпадающему с этим введенным паролем. Таким образом, при сравнении эталонный и введенный пароли находятся в зашифрованном виде и будут совпадать только в том случае, если исходный введенный пароль совпадет с исходным эталонным. При несовпадении исходного введенного пароля с исходным эталонным исходный введенный пароль будет зашифрован по-другому, так как ключ шифрования отличается от ключа, которым зашифрован эталонный пароль, и после зашифровки не совпадет с зашифрованным эталонным паролем.

Для обеспечения возможности контроля правильности ввода пароля при использовании необратимого шифрования на винчестер записывается таблица преобразованных паролей. Для их преобразования используется односторонняя криптографическая функция $y= F(x)$, обладающая следующим свойством: для данного аргумента x значение $F(x)$ вычисляется легко, а по данному y вычислительно сложно найти значение аргумента x, соответствующего данному y. В таблице паролей хранятся значения односторонних функций, для которых пароли берутся в качестве аргументов. При вводе пароля система зашиты легко вычисляет значение функции от пароля текущего пользователя и сравнивает со значением, приведенным в таблице для пользователя с выбранным идентификатором. Нарушитель, захвативший компьютер, может прочитать таблицу значений функций паролей, однако вычисление пароля практически не реализуемо.

При работе с паролями должна предусматриваться и такая мера, как недопустимость их распечатки или вывода на экраны мониторов. Поэтому система защиты должна обеспечивать ввод

пользователями запрошенных у них паролей без отображения этих паролей на мониторах.

Можно выделить следующие основные способы повышения стойкости системы защиты на этапе аутентификации:

- повышение степени нетривиальности пароля;
- увеличение длины последовательности символов пароля;
- увеличение времени задержки между разрешенными попытками повторного ввода неправильно введенного пароля;
- повышение ограничений на минимальное и максимальное время действительности пароля.

Чем *нетривиальнее пароль,* тем сложнее его запомнить. Плохо запоминаемый пароль может быть записан на листе бумаги, что повышает риск его раскрытия. Выходом здесь является использование определенного числа незаписываемых на бумаге пробелов или других символов в начале, внутри, а также в конце последовательности основных символов пароля. Кроме того, отдельные символы пароля могут набираться на другом регистре (например, вместо строчных быть прописными или наоборот), что также не должно отражаться на листе бумаги. В этом случае незаконно полученный лист бумаги с основными символами пароля не будет достаточным условием раскрытия пароля целиком.

Для исключения необходимости запоминания пользователями длинных и нетривиальных паролей в системе защиты может быть предусмотрена возможность записи паролей в зашифрованном виде на информационные носители, например флешки, магнитные карты, носители данных в микросхемах и т. д., а также считывания паролей с этих информационных носителей. Такая возможность позволяет повысить безопасность за счет значительного увеличения длины паролей, записываемых на носители информации. Однако при этом администрации службы безопасности следует приложить максимум усилий для разъяснения пользователям ВС о необходимости тщательной сохранности носителей информации с их паролями.

На степень информационной безопасности при использовании простого парольного метода проверки подлинности пользователей большое влияние оказывают *ограничения на минимальное и максимальное время действительности каждого пароля.* Чем чаше меняется пароль, тем обеспечивается большая безопасность.

Минимальное время действительности пароля задает время, в течение которого пароль менять нельзя, а *максимальное* — время, по истечении которого пароль будет недействительным. Соответственно, пароль должен быть заменен в промежутке между минимальным и максимальным временем его существования. Поэтому понятно, что более частая смена пароля обеспечивается при уменьшении минимального и максимального времени его действительности.

Минимальное и максимальное времена действительности пароля задаются для каждого пользователя администратором службы безопасности, который должен постоянно контролировать своевременность смены паролей пользователей.

При выборе пароля руководствуйтесь следующими инструкциями:

- Не указывайте свой ИД пользователя, а также всевозможные его модификации (в обратном порядке, удвоенный) в качестве пароля.
- Не используйте пароли повторно. Повторное использование паролей может быть запрещено конфигурацией системы.
- Не указывайте в качестве паролей личные имена.
- Не указывайте в качестве пароля слова, хранящиеся в электронных орфографических словарях.
- Длина пароля должна составлять не менее шести символов.
- Не указывайте в качестве паролей ругательства; при угадывании паролей их пробуют прежде всего.
- Выбирайте легко запоминающиеся пароли, чтобы вам не пришлось их записывать.
- Выбирайте пароли, содержащие цифры, а также строчные и прописные буквы.
- Рекомендуется задавать пароли, состоящие из двух слов, разделенных цифрами.
- Выбирайте легко произносимые пароли. Их легче запомнить.
- Не записывайте пароль. Если все же возникает необходимость записать его, поместите запись в надежное место, например, в сейф.

8. Основные модели криптосистем. Требования к криптосистемам. назначение и основные функции криптосистем

О важности сохранения информации в тайне знали уже в древние времена, когда с появлением письменности появилась и опасность прочтения ее нежелательными лицами. Более того, первоначально письменность сама по себе была криптографической системой, так как в древних обществах ею владели только избранные. С широким распространением письменности криптография стала формироваться как самостоятельная наука.

Точное время возникновения этих способов обмена тайной информацией теряется в глубине веков, и установить его невозможно. Историки полагают, что первые протокриптографические приемы появились в Древнем Египте около 4 тыс. лет назад. Писцы, составлявшие жизнеописания правителей, стремились придать стандартным иероглифам необычный вид на монументах и гробницах, чтобы сообщить надписям менее обыденный и более почтительный стиль. Жрецы пользовались этим же приемом при переписывании религиозных текстов, чтобы те выглядели для мирян загадочнее и внушительнее. Такие «переводы» становились все менее понятными простому люду, который в результате оказывался во все большей зависимости от жрецов.

По мере развития египетской цивилизации ширилось использование иероглифов. С увеличением количества надписей, высекавшихся на стенах храмов, люди теряли к ним интерес. Египтологи считают, что писцы тогда стали еще больше видоизменять некоторые знаки в стремлении пробудить любопытство и привлечь внимание населения. Эти модификации никоим образом не были кодами или шифрами, но они заключали в себе два основных принципа криптологии, а именно: изменение письма и сокрытие его смысла.

Примерно с 500 г. до н. э. в Индии также широко применялись секретные записи, в частности в донесениях шпионов и текстах, предположительно использовавшихся Буддой. Методы засекречивания включали в себя фонетическую замену, когда согласные и гласные менялись местами, использование перевернутых букв и запись текста под случайными углами.

Существуют проблемы тайной передачи информации и ее сокрытия от злоумышленника на расстоянии. Путей ее решения существует множество, среди которых можно выделить три основных направления.

1. Создать абсолютно надежный, недоступный для других канал связи между абонентами.

2. Использовать общедоступный канал связи, но скрыть сам факт передачи информации.

3. Использовать общедоступный канал связи, но передавать по нему нужную информацию в таком преобразованном виде, чтобы восстановить ее мог только адресат.

Проанализируем эти возможности.

1. С древних времен практиковалась охрана документа (носителя информации) физическими лицами, передача его специальным курьером (человеком (дипломатом) или животным (голубиная почта)) и т.д. Но, документ можно выкрасть, курьера можно перехватить, подкупить, в конце концов, убить. В настоящий момент для реализации данного механизма защиты используются современные телекоммуникационные каналы связи. Однако следует заметить, что данный подход требует значительных капитальных вложений. При современном уровне развития науки и техники сделать такой канал связи между удаленными абонентами для многократной передачи больших объемов информации практически нереально.

2. Разработкой средств и методов скрытия факта передачи сообщения занимается стеганография. Первые следы стеганографических методов теряются в глубокой древности. Так, в трудах древнегреческого историка Геродота встречается описание двух методов сокрытия информации: на обритую голову раба записывалось необходимое сообщение, а когда его волосы отрастали, он отправлялся к адресату, который вновь брил его голову и считывал доставленное сообщение. Второй способ заключался в следующем: сообщение наносилось на деревянную дощечку, а потом она покрывалась воском, и, тем самым, не вызывала никаких подозрений. Потом воск соскабливался, и сообщение становилось видимым. В настоящий момент стеганографические методы в совокупности с криптографическими нашли широкое применение в целях сокрытия и передачи конфиденциальной информации.

3. Разработкой методов преобразования информации с целью ее защиты от несанкционированного прочтения занимается криптография.

В истории развития криптографии можно выделить три этапа:

- наивная криптография;
- формальная криптография;
- математическая криптография.

Наивная криптография

Для наивной криптографии (до начала XVI в.) характерно использование любых, обычно примитивных, способов запутывания противника относительно содержания передаваемых сообщений. На начальном этапе для защиты информации использовались методы кодирования и стеганографии, которые родственны, но не тождественны криптографии. Шифровальные системы сводились к использованию перестановки или замены букв на различные символы (другие буквы, знаки, рисунки, числа и т.п.). Одни и те же способы шифрования использовались повторно, ключи были короткими, использовались примитивные способы преобразования исходной информации в зашифрованное сообщение. Это позволяло, однажды установив алгоритм шифрования, быстро расшифровывать сообщения.

Одним из первых зафиксированных примеров является шифр Цезаря. Другой шифр, полибианский квадрат, авторство которого приписывается греческому писателю Полибию, является шифром простой однозначной замены.

С VIII века н. э. развитие криптографии происходит в основном в арабских странах. Считается, что арабский филолог Халиль аль-Фарахиди первым обратил внимание на возможность использования стандартных фраз открытого текста для дешифрования. Он предположил, что первыми словами в письме на греческом языке византийскому императору будут «Во имя Аллаха», что позволило ему прочитать оставшуюся часть сообщения. Позже он написал книгу с описанием данного метода — «Китаб аль-Муамма» («Книга тайного языка»). В 855 г. выходит «Книга о большом стремлении человека разгадать загадки древней письменности» арабского учёного Абу Бакр Ахмед ибн Али Ибн Вахшия ан-Набати, одна из первых книг о криптографии с описаниями нескольких шифров, в том числе с применением нескольких алфавитов.

В древние времена широкое применение нашли различные простейшие криптографические устройства.

Греческим поэтом Архилохом, жившим в VII веке до н. э. упоминается устройство под названием сцитАла (греч. - жезл). Оно представляет собой цилиндр (иногда жезл командующего) и узкую полоску пергамента, обматывавшуюся вокруг него по спирали, на которой в свою очередь писалось сообщение.

Шифруемый текст писался на пергаментной ленте по длине палочки, после того как длина палочки оказывалась исчерпанной, она поворачивалась и текст писался далее, пока либо не заканчивался текст, либо не исписывалась вся пергаментная лента. В последнем случае использовался очередной кусок пергаментной ленты. Для расшифровки адресат использовал палочку такого же диаметра, на которую он наматывал пергамент, чтобы прочитать сообщение. Античные греки и спартанцы в частности, использовали этот шифр для связи во время военных кампаний. Однако такой шифр может быть легко взломан. Например, метод взлома сциталы был предложен ещё Аристотелем. Он состоит в том, что не зная точного диаметра палочки, можно использовать конус, имеющий переменный диаметр и перемещать пергамент с сообщением по его длине до тех пор, пока текст не начнёт читаться - таким образом дешифруется диаметр сциталы.

Другим широко известным криптографическим устройством защиты информации был «диск Энея» - инструмент для защиты информации, придуманный Энеем Тактиком в IV веке до н. э. Устройство представляло собой диск диаметром 13-15 см и толщиной 1-2 см с проделанными в нём отверстиями, количество которых равнялось числу букв в алфавите. Каждому отверстию ставилась в соответствие конкретная буква. В центре диска находилась катушка с намотанной на неё ниткой [17].

Механизм шифрования был очень прост. Для того, чтобы зашифровать послание, необходимо было поочерёдно протягивать свободный конец нити через отверстия обозначающие буквы исходного не зашифрованного сообщения. В итоге, сам диск, с продетой в его отверстия ниткой, и являлся зашифрованным посланием. Получатель сообщения последовательно вытягивал нить из каждого отверстия, тем самым получал последовательность букв. Но эта последовательность являлась обратной по отношению к исходному сообщению, то есть он читал сообщение наоборот. Зашифрованное сообщение было доступно к прочтению любому, кто смог завладеть диском. Так как

сообщение предавали обычные гонцы, а не воины, Эней предусмотрел возможность быстрого уничтожения передаваемой информации. Для этого было достаточно вытянуть всю нить за один из её концов, либо сломать диск, просто наступив на него. На самом деле «диск Энея» нельзя назвать настоящим криптографическим инструментом, поскольку прочитать сообщение мог любой желающий. Но это устройство стало прародителем первого по истине криптографического инструмента, изобретение которого также принадлежит Энею.

Формальная криптография

Этап формальной криптографии (конец XV – начало XX вв.) связан с появлением формализованных и относительно стойких к ручному криптоанализу шифров.

Отцом западной криптографии называют учёного эпохи Возрождения Леона Баттисту Альберти. Изучив методы вскрытия использовавшихся в Европе моноалфавитных шифров (шифров однозначной замены), он попытался создать шифр, который был бы устойчив к частотному криптоанализу. Он предложил вместо единственного секретного алфавита, как в моноалфавитных шифрах, использовать два или более, переключаясь между ними по какому-либо правилу. Однако флорентийский учёный так и не смог оформить своё открытие в полную работающую систему, что было сделано уже его последователями (Блез Вижинер).

В 1550 г. итальянский математик Джероламо Кардано, состоящий на службе у папы римского, предложил новую технику шифрования - решётку Кардано.

Значительный толчок криптографии дало изобретение телеграфа. Сама передача данных перестала быть секретной, и сообщение, в теории, мог перехватить кто угодно. Интерес к криптографии возрос, в том числе, и среди простого населения, в результате чего многие попытались создать индивидуальные системы шифрования. Преимущество телеграфа было явным и на поле боя, где командующий должен был отдавать немедленные приказы на поле сражения, а также получать информацию с мест событий. Это послужило толчком к развитию полевых шифров.

В 1883 г. голландец Огюст Керкгоффс2 опубликовал труд под названием «Военная криптография» (фр. «La Cryptographie Militaire»). В нём он описал шесть требований, которым должна удовлетворять защищённая система. Хотя к некоторым из них стоит относиться с подозрением, стоит отметить труд за саму попытку:

1. шифр должен быть физически, если не математически, невскрываемым;
2. система не должна требовать секретности, на случай, если она попадёт в руки врага;
3. ключ должен быть простым, храниться в памяти без записи на бумаге, а также легко изменяемым по желанию корреспондентов;
4. зашифрованный текст должен (без проблем) передаваться по телеграфу;
5. аппарат для шифрования должен быть легко переносимым, работа с ним не должна требовать помощи нескольких лиц;
6. аппарат для шифрования должен быть относительно прост в использовании, не требовать значительных умственных усилий или соблюдения большого количества правил.

Им же был сформулирован известный «принцип Керкгоффса» - правило разработки криптографических систем, согласно которому в засекреченном виде держится только определённый набор параметров алгоритма, называемый ключом, а сам алгоритм шифрования должен быть открытым. Другими словами, при оценке надёжности шифрования необходимо предполагать, что противник знает об используемой системе шифрования всё, кроме применяемых ключей.

Математическая криптография

После Первой мировой войны правительства стран засекретили все работы в области криптографии. К началу 1930-х годов окончательно сформировались разделы математики, являющиеся основой для будущей науки: общая алгебра, теория чисел, теория вероятностей и математическая статистика. К концу 1940-х годов построены первые программируемые счётные машины, заложены основы теории алгоритмов, кибернетики. Тем не менее, в период после Первой мировой войны и до конца 1940-х годов в открытой печати было опубликовано совсем немного работ и монографий, но и те отражали далеко не самое актуальное состояние дел. Наибольший прогресс в криптографии достигается в военных ведомствах.

В 1960-х годах начали появляться различные блочные шифры, которые обладали большей криптостойкостью по сравнению с результатом работы роторных машин. Однако они предполагали

обязательное использование цифровых электронных устройств - ручные или полумеханические способы шифрования уже не использовались.

Примерно в это же время Хорст Фейстель переходит из Военно-воздушных сил США на работу в лабораторию корпорации IBM. Там он занимается разработкой новых методов в криптографии и разрабатывает ячейку Фейстеля, являющуюся основой многих современных шифров, в том числе шифра Lucifer, ставшего прообразом шифра DES – бывшего стандарта шифрования США, первого в мире открытого государственного стандарта на шифрование данных. На основе ячейки Фейстеля были созданы и другие блочные шифры, в том числе TEA (1994 г.), Twofish (1998 г.), IDEA (2000 г.), а также ГОСТ 28147-89, являющийся стандартом шифрования в России.

В 1976 г. публикуется работа Уитфилда Диффи и Мартина Хеллмана «Новые направления в криптографии» (англ. «New Directions in Cryptography»). Данная работа открыла новую область в криптографии, теперь известную как криптография с открытым ключом. Также в работе содержалось описание алгоритма Диффи - Хеллмана - Меркла, позволявшего сторонам сгенерировать общий секретный ключ, используя открытый канал связи.

Чарльз Беннет (Charles Bennet) и Жиль Брассард (Gilles Brassard), опираясь на работу Стивена Уиснера (Stephen Wiesner), разработали теорию квантовой криптографии, которая базируется скорее на квантовой физике, нежели на математике.

Применение криптографии в решении вопросов аутентификации, целостности данных, передачи конфиденциальной информации по каналам связи и т.п. стало неотъемлемым атрибутом информационных систем. В современном мире криптография находит множество различных применений - она используется в сотовой связи, платном цифровом телевидении, при подключении к Wi-Fi, для защиты билетов от подделок на транспорте, в банковских операциях, в системах электронных платежей и т.д.

Современные методы использования криптографии

Появление доступного интернета перевело криптографию на новый уровень. Криптографические методы стали широко использоваться частными лицами в электронных коммерческих операциях, телекоммуникациях и многих других средах. Первая получила особенную популярность и привела к появлению новой, не контролируемой государством валюты — биткойна.

Многие энтузиасты быстро смекнули, что банковский перевод — штука, конечно, удобная, однако, для покупки таких приятных в быту вещей, он не подходит. Не подходит он и при запущенных случаях паранойи, ибо требует от получателя и отправителя обязательной аутентификации.

в 2009 году Сатоши Накамото разработал платежную систему нового типа — BitCoin. Так родилась криптовалюта. Ее транзакции не требуют посредника в виде банка или другой финансовой организации, отследить их невозможно. Сеть полностью децентрализована, биткойны не могут быть заморожены или изъяты, они полностью защищены от государственного контроля. В то же время биткойн может использоваться для оплаты любых товаров — при условии согласия продавца.

Новые электронные деньги производят сами пользователи, предоставляющие вычислительные мощности своих машин для работы всей системы BitCoin. Такой род деятельности называется майнинг (mining — добыча полезных ископаемых). Заниматься майнингом в одиночку не очень выгодно, гораздо проще воспользоваться специальными серверами — пулами. Они объединяют ресурсы нескольких участников в одну сеть, а затем распределяют полученную прибыль.

Крупнейшей площадкой купли-продажи биткойнов является японская Mt. Gox, через которую проводятся 67% транзакций в мире. Заядлые анонимы предпочитают ей российскую BTC-E: регистрация здесь не требует идентификации пользователя. Курс криптовалюты довольно-таки нестабилен и определяется только балансом спроса и предложения в мире. Предостережением новичкам может служить известная история о том, как 10 тысяч единиц, потраченых одним из пользователей на пиццу, превратились через некоторое время в 2,5 миллиона долларов.

Криптология разделяется на два направления – криптографию и криптоанализ.

Криптография– наука, изучающая методы преобразования информации, обеспечивающие ее конфиденциальность и аутентичность.

Современная криптография включает в себя четыре крупных раздела:

- симметричные криптосистемы;
- криптосистемы с открытым ключом;
- системы электронной подписи;

- управление ключами.

Основные направления использования криптографических методов – передача конфиденциальной информации по каналам связи (например, электронная почта), установление подлинности передаваемых сообщений, хранение информации (документов, баз данных) на носителях в зашифрованном виде.

Виды шифров

Рассмотрим классификации шифров по разным признакам. По типу преобразований шифры можно разделить на следующие группы:

- шифры замены (подстановки);
- шифры перестановки;
- шифры гаммирования;
- шифры на основе аналитических преобразований.

При этом надо учитывать, что некоторые современные шифры совместно используют преобразования различных типов.

Шифры замены (подстановки): преобразование заключается в том, что символы шифруемого текста заменяются символами того или иного алфавита (алфавита криптограммы) в соответствии с заранее обусловленной схемой замены.

Подстановки разделяются на одноалфавитные и многоалфавитные. В первом случае, определенному символу алфавита исходного сообщения всегда ставится в соответствие один и тот же символ алфавита криптограммы. Один из наиболее известных шифров данного класса – шифр Цезаря. К достоинству таких шифров относится простота преобразования. Но они легко взламываются путем сравнения частоты появления различных символов в естественном языке и криптограмме.

При использовании многоалфавитных подстановок, учитываются дополнительные параметры (например, положение преобразуемого символа в тексте) и в зависимости от них символ исходного алфавита может заменяться на один из нескольких символов алфавита шифртекста. Например, нечетные символы сообщения заменяются по од-ному правилу, четные – по другому.

Шифры перестановок: шифрование заключается в том, что символы исходного текста переставляются по определенному правилу в пределах блока этого текста. При достаточной длине блока и сложном, неповторяющемся порядке перестановки, можно достичь приемлемой стойкости шифра.

Шифрование гаммированием заключается в том, что символы шифруемого текста складываются с символами некоторой случайной последовательности, называемой гаммой шифра или ключевой гаммой. Стойкость шифрования определяется длиной (периодом) неповторяющейся части гаммы шифра, а также сложностью предугадывания следующих элементов гаммы по предыдущим.

Шифрование аналитическими преобразованиями подразумевает использование аналитического правила (формулы) по которому пре-образуется текст.

По типу использования ключей шифры делятся на:

- симметричные, использующие для шифрования и расшифровывания информации один и тот же ключ;
- асимметричные, использующие для шифрования и расшифровывания два различных ключа.

По размеру преобразуемого блока шифры делятся на блочные и потоковые.

Блочные шифры осуществляют преобразование информации блоками фиксированной длины. Если длина шифруемого сообщения некратна размеру блока, то его добавляют до нужной длины последовательностью специального вида. Например, это может быть последовательность 100…0. После расшифровки, последний блок просматривают справа налево и отбрасывают «хвост» до первой единицы включительно. Чтобы подобное дополнение было применимо во всех случаях, если сообщение кратно длине блока, в его конец надо добавить целый блок указанного вида.

Потоковые шифры предназначены для преобразования сообщения поэлементно (элементом может быть бит, символ и т. п.). Примером такого вида шифров являются шифры гаммирования.

В качестве информации, подлежащей шифрованию и расшифрованию, а также электронной подписи будут рассматриваться тексты, построенные на некотором алфавите.

Алфавит – конечное множество используемых для кодирования информации знаков.

Текст – упорядоченный набор из элементов алфавита.

В качестве примеров алфавитов, используемых в современных информационных системах, можно привести следующие:

- алфавит Z33 – 32 буквы русского алфавита (исключая «ё») и пробел;
- алфавит Z256 – символы, входящие в стандартные коды ASCII и КОИ-8;
- двоичный алфавит – Z2 = {0,1};
- восьмеричный или шестнадцатеричный алфавиты.

Зашифрование – процесс преобразования открытых данных в зашифрованные при помощи шифра. Вместо термина «открытые данные» часто употребляются термины «открытый текст» и «исходный текст», а вместо термина «зашифрованные данные» – «шифрованный текст».

Расшифрование – процесс, обратный зашифрованию, т.е. процесс преобразования зашифрованных данных в открытые при помощи ключа. В некоторых отечественных источниках отдельно выделяют термин дешифрование, подразумевая под этим восстановление исходного текста на основе шифрованного без знания ключа, т.е. методами криптоанализа. В дальнейшем будем считать расшифрование и дешифрование синонимами.

Криптографическая система, или шифр, представляет собой семейство T обратимых преобразований открытого текста в шифрованный. Участники этого семейства индексируются или обозначаются символом k; параметр k обычно называется ключом. Преобразование Tk определяется соответствующим алгоритмом и значением ключа k.

Ключ – конкретное значение некоторых параметров алгоритма криптографического преобразования, обеспечивающее выбор одного преобразования из семейства. Секретность ключа должна обеспечивать невозможность восстановления исходного текста по шифрованному.

Пространство ключей K – набор возможных значений ключа. Обычно ключ представляет собой последовательный ряд букв алфавита. Следует отличать понятия «ключ» и «пароль». Пароль также является секретной последовательностью букв алфавита, однако используется не для шифрования (как ключ), а для идентификации субъектов.

Электронной (цифровой) подписью называется присоединяемое к тексту его криптографическое преобразование, которое позволяет при получении текста другим пользователем проверить авторство и подлинность сообщения.

Требования к криптографическим системам

Процесс криптографического закрытия данных может осуществляться как программно, так и аппаратно. Аппаратная реализация отличается существенно большей стоимостью, однако ей присущи и преимущества: высокая производительность, простота, защищенность и т.д. Программная реализация более практична, допускает известную гибкость в использовании. Для современных криптографических систем защиты информации сформулированы следующие общепринятые требования.

1. Знание алгоритма шифрования не должно снижать криптостойкости шифра.
2. Зашифрованное сообщение должно поддаваться чтению только при наличии ключа.
3. Шифр должен быть стойким даже в случае, если нарушителю известно достаточно большое количество исходных данных и соответствующих им зашифрованных данных.
4. Число операций, необходимых для расшифровывания информации путем перебора всевозможных ключей, должно иметь строгую нижнюю оценку и должно либо выходить за пределы возможностей современных компьютеров, либо требовать создания использования дорогих вычислительных систем.
5. Незначительное изменение ключа или исходного текста должно приводить к существенному изменению вида зашифрованного текста.
6. Структурные элементы алгоритма шифрования должны быть неизменными.
7. Длина шифрованного текста должна быть равной длине исходного текста.
8. Дополнительные биты, вводимые в сообщение в процессе шифрования, должны быть полностью и надежно скрыты в шифрованном тексте.
9. Не должно быть простых и легко устанавливаемых зависимостей между ключами, последовательно используемыми в процессе шифрования.

10. Любой ключ из множества возможных должен обеспечивать надежную защиту информации.

Главным действующим лицом в криптоанализе выступает нарушитель (или криптоаналитик) – лицо или группа лиц, целью которых является прочтение или подделка защищенных криптографическими методами сообщений.

Криптоанализ — наука о методах получения исходного значения зашифрованной информации, не имея доступа к секретной информации (ключу), необходимой для этого.

В отношении нарушителя принимается ряд допущений, которые, как правило, лежат в основе математических или иных моделей.

- Нарушитель знает алгоритм шифрования (или электронную цифровую подпись (ЭЦП)) и особенности его реализации в конкретном случае, но не знает ключа.
- Нарушителю доступны все зашифрованные тексты. Нарушитель может иметь доступ к некоторым исходным текстам, для которых известен соответствующий им зашифрованный текст.
- Нарушитель имеет в своем распоряжении вычислительные, людские, временные и иные ресурсы, объем которых оправдывает потенциальную ценность информации, которая будет добыта в результате криптоанализа.

При анализе криптостойкости шифра необходимо учитывать и человеческий фактор, например, подкуп конкретного человека, в руках которого сосредоточена необходимая информация, может стоить на несколько порядков дешевле, чем создание суперкомпьютера для взлома шифра.

Попытка прочтения или подделки зашифрованного сообщения, вычисления ключа методами криптоанализа называется криптоатакой, или атакой на шифр. Удачную криптоатаку называют взломом.

Принято различать несколько уровней криптоатаки в зависимости от объема информации, доступной криптоаналитику. По нарастанию сложности можно выделить три уровня криптоатаки.

- Атака на шифрованный текст (уровень КА1) – нарушителю доступны все или некоторые зашифрованные сообщения.
- Атака на пару «исходный текст – шифрованный текст» (уровень КА2) – нарушителю доступны все или некоторые зашифрованные сообщения и соответствующие им исходные сообщения.
- Атака на выбранную пару «исходный текст – шифрованный текст» (уровень КА3) – нарушитель имеет возможность выбирать исходный текст, получать для него шифрованный текст и на основе анализа зависимостей между ними вычислять ключ.

Все современные криптосистемы обладают достаточной стойкостью даже к атакам уровня КА3, т.е. когда нарушителю доступно, по сути, шифрующее устройство.

Криптостойкостью называется характеристика шифра, определяющая его стойкость к расшифрованию без знания ключа (т.е. криптоатаке). Показатель криптостойкости – главный параметр любой криптосистемы. В качестве показателя криптостойкости можно выбрать:

- количество всех возможных ключей или вероятность подбора ключа за заданное время с заданными ресурсами;
- количество операций или время (с заданными ресурсами), необходимое для взлома шифра с заданной вероятностью;
- стоимость вычисления ключевой информации или исходного текста.

Все эти показатели должны учитывать также уровень возможной криптоатаки. Однако следует понимать, что эффективность защиты информации криптографическими методами зависит не только от криптостойкости шифра, но и от множества других факторов, включая вопросы реализации криптосистем в виде устройств или программ.

9. Классические методы шифрования. Шифрование методами перестановки: простая перестановка, одиночная перестановка по ключу, двойная перестановка, магический квадрат, шифр Кардано, шифр Ришелье

Простая перестановка

Одним из шифров, основанных на перестановке строк и столбцов в таблице с открытым текстом, является шифр простой перестановки. Создание криптограммы при использовании данного шифра следует начать с составления таблицы, в ячейки которой необходимо вписать по строкам буквы открытого текста. При этом количество строк и столбцов в такой шифровальной таблице выбирается произвольно. После заполнения таблицы буквы в криптограмму выписываются по столбцам, сначала из первого столбца, затем из второго и так далее.

В качестве примера зашифруем с помощью этого шифра открытый текст МЕСТО ВСТРЕЧИ ИЗМЕНИТЬ НЕВОЗМОЖНО. При выборе таблицы, состоящей из пяти строк и шести столбцов, ее ячейки будут заполнены следующим образом:

М	Е	С	Т	О	В
С	Т	Р	Е	Ч	И
И	З	М	Е	Н	И
Т	Ь	Н	Е	В	О
Э	М	О	Ж	Н	О

Теперь для создания криптограммы достаточно последовательно выписать буквы из ячеек первого столбца, затем из ячеек второго столбца и так далее.

В окончательном виде криптограмма для открытого текста МЕСТО ВСТРЕЧИ ИЗМЕНИТЬ НЕВОЗМОЖНО будет выглядеть так:

МСИТЗ ЕТЗЬМ СРМНО ТЕЕЕЖ ОЧНВН ВИИОО

Если записать эту криптограмму без пробелов, то она примет следующий вид:

МСИТЗЕТЗЬИСРМНОТЕЕЕЖОЧНВНВИИОО

Для расшифровки такого шифрованного сообщения достаточно в таблицу аналогичных размеров по столбцам вписать буквы криптограммы, а затем по строкам прочитать открытый текст. Естественно, для этого получатель сообщения должен знать размер таблицы.

УСМАНОВА ИРИНА ОЛЕГОВНА

У	А	В	Р	А	Е	В
С	Н	А	И	О	Г	Н
М	О	И	Н	Л	О	А

УАВРАЕВСНАИОГНМОИНЛОА ТАБИЦА 3Х7

Метод шифрующих таблиц с одиночной перестановкой по ключу

Отличается от метода шифрующих таблиц, перестановкой столбцов таблицы по ключевому слову после заполнения таблицы исходным текстом. Длина ключевого слова, фразы или числа, задающего способ перестановки, должна быть равна числу столбцов таблицы. Столбцы переставляются в порядке следования в алфавите символов ключевого слова.

Пример 3. Зашифруем фразу «СИСТЕМНЫЙ ПАРОЛЬ ИЗМЕНЕН» с помощью таблицы размером 4х6 и ключевого слова «СКАНЕР».

Для записи заданной фразы достаточно одной таблицы. Сначала ее заполняют по столбцам исходной фразой (рис. 4а), затем переставляют столбцы по ключевому слову «СКАНЕР» и считывают символы по строкам (рис. 4б). В результате получаем шифрованное сообщение: «Й_ЕРЕС_ИМОНИПЗНЛЕСАМЫЬНТ».

а) исходная таблица б) после перестановки

Рис. 4. Реализация шифрующих таблиц с одиночной перестановкой по ключу
БОРЬБА ЕСТЬ УСЛОВИЕ ЖИЗНИ Ключ Слово Слово

С	Л	О	В	О
5	2	3	1	4
Б	А	Ь	О	Ж
О	-	-	В	И
Р	Е	У	И	З
Ь	С	С	Е	Н
Б	Т	Л	-	И

В	Л	О	О	С
1	2	3	4	5
О	А	Ь	Ж	Б
В	-	-	И	О
И	Е	У	З	Р
Е	С	С	Н	Ь
-	Т	Л	И	Б

ОАЬЖБВ—ИОИЕУЗРЕССНЬ-ТЛИБ
БОЯТЬСЯ НАДО НЕ СМЕРТИ, А ПУСТОЙ ЖИЗНИ.

И	Г	Р	А
З	2	4	1
Б	О	И	Й
О	-	,	-
Я	Н	-	Ж
Т	Е	А	И
Ь	-	-	З
Я	С	П	Н
-	М	У	И
Н	Е	С	.
А	Р	Т	1
Д	Т	О	1

А	Г	И	Р
1	2	3	4
Й	О	Б	И
-	-	О	,
Ж	Н	Я	-
И	Е	Т	А
З	-	Ь	-
Н	С	Я	П
И	М	-	У
.	Е	Н	С
1	Р	А	Т

1	Т	Д	О

ЙОБИ—О,ЖНЯ-ИЕТАЗ-Ь-НСЯПИМ-У.ЕНС1РАТ1ТДО

А	Г	И	Р
1	2	З	4
Й	О	Б	И
-	-	О	,
Ж	Н	Я	-
И	Е	Т	А
З	-	Ь	-
Н	С	Я	П
И	М	-	У
.	Е	Н	С
1	Р	А	Т
1	Т	Д	О

Шифрующие таблицы с двойной перестановкой по ключу

Используют для повышения скрытности шифра. В данном методе используются два ключевых слова. Первое слово определяет перестановку столбцов, второе – перестановку строк таблицы. Перестановки производятся согласно порядку следования в алфавите символов ключевых слов.

На первом этапе исходный текст (или его фрагмент) построчно записывается в таблицу. Далее перестанавливаются столбцы исходной таблицы по первому ключевому слову. Затем переставляются строки полученной таблицы по второму ключевому слову. На последнем этапе из итоговой таблицы считывается шифртекст по столбцам.

Пример 4. Зашифруем фразу из третьего примера с помощью таблицы размером 4х6 и ключевых слов «СКАНЕР» и «4123».

После заполнения исходной таблицы по строкам (рис. 5а) переставляем столбцы по порядку следования в алфавите букв слова «СКАНЕР» (рис. 5б). Затем переставляем строки. Порядковый номер строки определяет цифра второго ключевого слова «4123» (рис. 5в). На этом перестановки в таблице заканчиваются. Шифртекст считываем по столбцам и получаем: «ЙЛЕСП_ЕЕЫОМИ_ЬНТАИНМНРЗС»

	С	К	А	Н	Е	Р
	6	3	1	4	2	5
4	С	И	С	Т	Е	М
1	Н	Ы	Й		П	А
2	Р	О	Л	Ь		И
3	З	М	Е	Н	Е	Н

а) исходная таблица

	А	Е	К	Н	Р	С
	1	2	3	4	5	6
4	С	Е	И	Т	М	С
1	Й	П	Ы		А	Н
2	Л		О	Ь	И	Р
3	Е	Е	М	Н	Н	З

б) перестановка столбцов

	А	Е	К	Н	Р	С
	1	2	3	4	5	6
1	Й	П	Ы		А	Н
2	Л		О	Ь	И	Р
3	Е	Е	М	Н	Н	З
4	С	Е	И	Т	М	С

в) перестановка строк

Рис. 5. Пример шифрования методом двойной перестановки

Шифрование по методу магических квадратов.

Магическими квадратами называют квадратные таблицы с вписанными в их клетки последовательными натуральными числами, начиная от 1, которые дают в сумме по каждому столбцу, строке и диагонали одно и то же число.

При шифровании буквы открытого текста необходимо вписать в магический квадрат в соответствии с нумерацией его клеток. Для получения шифртекста считывают содержимое заполненной таблицы по строкам.

Пример 5. Зашифруем фразу «МАГИЧЕСКАЯ СИЛА» с помощью магического квадрата размером 4х4. Для этого выберем один из 880 вариантов магических квадратов заданного размера (рис. 6а). Затем вписываем каждую букву сообщения в отдельную ячейку таблицы с номером, соответствующим порядковому номеру буквы в исходной фразе (рис. 6б). При считывании заполненной таблицы по строкам получаем шифртекст: «_ГАИАЕССЧЯ_КИАЛМ».

16	3	2	13
9	6	7	12
5	10	11	8
4	15	14	1

а) магический квадрат

	Г	А	И
А	Е	С	С
Ч	Я		К
И	А	Л	М

б) квадрат с сообщением

Рис. 6. Пример шифрования с помощью магических квадратов

10. Шифрование методами замены: полибианский квадрат, шифр Цезаря, шифр Цезаря с ключевым словом, аффинная система подстановок Цезаря, диск Альберти, шифр Гронсфельда, шифр Виженера, одноразовый блокнот

Шифр Цезаря

Шифр Цезаря — один из древнейших шифров. При шифровании каждый символ заменяется другим, отстоящим от него в алфавите на фиксированное число позиций. Шифр Цезаря можно классифицировать как шифр подстановки, при более узкой классификации — шифр простой замены.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки. Естественным развитием шифра Цезаря стал шифр Виженера. С точки зрения современного криптоанализа, шифр Цезаря не имеет приемлемой стойкости.

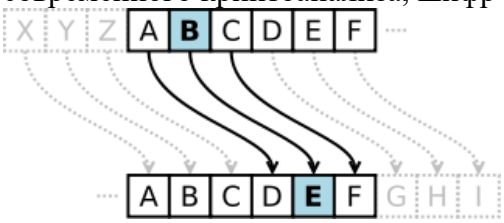


Рисунок 1 Шифр Цезаря

Математическая модель

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами:

$$y = x + k \ (mod\ n)$$
$$x = y - k \ (mod\ n),$$

где x — символ открытого текста

y — символ шифрованного текста

n — мощность алфавита (кол-во символов)

k — ключ.

Можно заметить, что суперпозиция двух шифрований на ключах k1 и k2 — есть просто шифрование на ключе k1+k2. Более общее, множество шифрующих преобразований шифра Цезаря образует группу Z.

Алфавит:

Буква	**А**	**Б**	**В**	**Г**	**Д**	**Е**	**Ё**	**Ж**	**З**	**И**	**Й**
Номер	1	2	3	4	5	6	7	8	9	10	11
Буква	**К**	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**	**Ф**
Номер	12	13	14	15	16	17	18	19	20	21	22
Буква	**Х**	**Ц**	**Ч**	**Ш**	**Щ**	**Ь**	**Ы**	**Ъ**	**Э**	**Ю**	**Я**
Номер	23	24	25	26	27	28	29	30	31	32	33

Пример:

Сообщение	К	Р	И	П	Т	О	Г	Р	А	Ф	И	Я
Номер 1	12	18	10	17	20	16	4	18	1	22	10	33
Номер 1 +5	17	23	15	22	25	21	9	23	6	27	15	5
Шифр	П	Х	Н	Ф	Ч	У	З	Х	Е	Щ	Н	Д

Ответ: «Пхнфчузхещнд»

Система Цезаря с ключевым словом.

В этой системе шифрования наряду с числовым ключом K, 0 ? K ? (M-1), задающим смещение, используется ключевое слово для изменения порядка символов в заменяющем алфавите.

В качестве ключевого слова необходимо выбирать слово или короткую фразу (не более длины алфавита). Все буквы ключевого слова должны быть различными.

Для создания таблицы замены ключевое слово записываем под буквами алфавита, начиная с буквы, числовой код которой совпадает с выбранным числовым ключом K. Оставшиеся буквы алфавита замены записываем в алфавитном порядке (избегая повтора букв) после ключевого слова. При достижении конца таблицы циклически переходим на ее начало и дописываем последние буквы алфавита не встречавшиеся ранее.

Пример 9. Пусть задан ключ K=3, ключевое слово «ШИФРОВКА» и русский алфавит из 32 букв. Необходимо создать таблицу замен для системы шифрования Цезаря с ключевым словом и с ее помощью зашифровать слово «НЕПТУН».

Первую букву ключевого слова («Ш») записываем под символом «Г» открытого текста с числовым кодом, определенным ключом K=3. Остальные буквы слова «ШИФРОВКА» записываем подряд. Оставшиеся ячейки заполняем теми буквами алфавита, которые не вошли в ключевое слово: «Б», «Г», «Д», «Е» и т.д. до буквы «Ь». Оставшиеся буквы «Э», «Ю», «Я» вписываем в начало таблицы под буквами «А», «Б» и «В», соответственно (табл. 4).

код	0	1	2	**3**	4	5	6	7	8	9	10	11	12	13	14	15
исх. текст	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
шифртекст	Э	Ю	Я	**Ш**	**И**	**Ф**	**Р**	**О**	**В**	**К**	**А**	Б	Г	Д	Е	Ж
код	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
исх. текст	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
шифртекст	З	Й	Л	М	Н	П	С	Т	У	Х	Ц	Ч	Щ	Ъ	Ы	Ь

Таблица 4. Таблица замен символов для системы шифрования Цезаря при K=3, M=32 и ключевом слове «ШИФРОВКА»

Далее с помощью табл. 4 шифруем побуквенно слово «НЕПТУН». В результате получаем шифртекст: «ДФЖЛМД».

Аффинная система подстановок Цезаря. В данном методе используется ключ шифрования в виде пары целых чисел (A, K). Число A задает переход при шифровании вперед на A?J букв, а число K – дополнительное смещение по алфавиту на K букв. Следовательно, аффинную систему подстановок Цезаря можно описать следующей формулой:

I = (A?J+K) mod M. (3)

Формула (3) может быть использована только при выполнении следующих условий: 0 ? (A, J)? (M-1), 0 ? K ? (M-1), НОД (A, M)=1.

Наибольший общий делитель чисел A и M должен быть равен единице, чтобы избежать ситуации повтора, когда разным символам открытого текста соответствует один и тот же символ шифртекста.

Пример 8. Создадим таблицу замен для аффинной системы подстановок Цезаря с ключом (5, 4) на примере русского алфавита. Возьмем алфавит из 32 букв (все кроме буквы «Ё»). Таким образом, A = 5, K = 3, M = 32 и все условия (в том числе и НОД (5, 32) = 1) необходимые для использования (3) выполняются. Код буквы шифртекста находим из соотношения I = (5?J+4) mod 32.

Сведем числовые коды букв открытого и зашифрованного текстов в таблицу (табл. 2).

J	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
I	3	8	13	18	23	28	1	6	11	16	21	26	31	4	9	14
J	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
I	19	24	29	2	7	12	17	22	27	0	5	10	15	20	25	30

Таблица 2. Таблица кодов для аффинных подстановок при A=5, K=3, M=32

Преобразуем числовые коды в соответствующие буквы русского алфавита и получим соответствие для символов открытого текста и шифртекста (табл. 3).

J	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Исх. текст	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Шифртекст	Г	И	Н	Т	Ч	Ь	Б	Ж	Л	Р	Х	Ъ	Я	Д	Й	О
J	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Исх. текст	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
Шифртекст	У	Ш	Э	В	З	М	С	Ц	Ы	А	Е	К	П	Ф	Щ	Ю

Таблица 3. Таблица символов для аффинных подстановок при A=5, K=3, M=32

С помощью табл. 3 или формулы (3) слово «МИР» преобразуется в шифртекст «ЯЛУ».

Система Вижинера

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет вид:

ATTACKATDOWN

Человек, посылающий сообщение, записывает ключевое слово («LEMON») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

LEMONLEMONLE

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Первый символ исходного текста A зашифрован последовательностью L, которая является первым символом ключа. Первый символ L шифрованного текста находится на пересечении строки L и столбца A в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа; то есть второй символ шифрованного текста X получается на пересечении строки E и столбца T. Остальная часть исходного текста шифруется подобным способом.

Исходный текст: ATTACKATDAWN

Ключ: LEMONLEMONLE

Зашифрованный текст: LXFOPVEFRNHR

Расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова; в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом.

Если буквы A—Z соответствуют числам 0—25, то шифрование Виженера можно записать в виде формулы:

$$C_i = (P_i + K_i) \bmod 26$$

Расшифровка:

$$P_i = (C_i - K_i + 26) \bmod 26$$

11. Понятие о генераторах псевдослучайной последовательности. Алгоритмы генерации

Случайное число – число, представляющее собой реализацию случайной величины.

Детерминированный алгоритм – алгоритм, который возвращает те же выходные значения при тех же входных значениях.

Псевдослучайное число – число, полученное детерминированным алгоритмом, используемое в качестве случайного числа.

Физическое случайное число (истинно случайное) – случайное число, полученное на основе некоторого физического явления.

Генератор псевдослучайных чисел — алгоритм, порождающий последовательность чисел, элементы которой почти независимы друг от друга и подчиняются заданному распределению (обычно равномерному).

Линейный конгруэнтный генератор псевдослучайных чисел

Генераторы псевдослучайных чисел могут работать по разным алгоритмам. Одним из простейших генераторов является так называемый линейный конгруэнтный генератор, который для вычисления очередного числа k_i использует формулу

$k_i=(a*k_{i-1}+b) \bmod c$,

где a, b, c — некоторые константы, а k_{i-1} — предыдущее псевдослучайное число.

Для получения k_1 задается начальное значение k_0. Возьмем в качестве примера a=5,b=3,c=11 и пусть k_0= 1. В этом случае мы сможем по приведенной выше формуле получать значения от 0 до 10 (так как c = 11). Вычислим несколько элементов последовательности:

$k_1 = (5 * 1 + 3) \bmod 11 = 8$;

$k_2 = (5 * 8 + 3) \bmod 11 = 10$;

$k_3 = (5 * 10 + 3) \bmod 11 = 9$;

$k_4 = (5 * 9 + 3) \bmod 11 = 4$;

$k_5 = (5 * 4 + 3) \bmod 11 = 1$.

Полученные значения (8, 10, 9, 4, 1) выглядят похожими на случайные числа. Однако следующее значение k_6 будет снова равно 8:

$k_6 = (5 * 1 + 3) \bmod 11 = 8$,

а значения k_7 и k_8 будут равны 10 и 9 соответственно:

$k_7 = (5 * 8 + 3) \bmod 11 = 10$;

$k_8= (5 * 10 + 3) \bmod 11 = 9$.

Выходит, наш генератор псевдослучайных чисел повторяется, порождая периодически числа 8, 10, 9, 4, 1. К сожалению, это свойство характерно для всех линейных конгруэнтных генераторов. Изменяя значения основных параметров a, b и c, можно влиять на длину периода и на сами порождаемые значения k_i. Так, например, увеличение числа c в общем случае ведет к увеличению периода. Если параметры a, b и c выбраны правильно, то генератор будет порождать случайные числа с максимальным периодом, равным c. При программной реализации значение c обычно устанавливается равным 2^{b-1} или 2^b, где b — длина слова ЭВМ в битах.

Достоинством линейных конгруэнтных генераторов псевдослучайных чисел является их простота и высокая скорость получения псевдослучайных значений. Линейные конгруэнтные генераторы находят применение при решении задач моделирования и математической статистики, однако в криптографических целях их нельзя рекомендовать к использованию, так как специалисты по криптоанализу научились восстанавливать всю последовательность ПСЧ по нескольким значениям. Например, предположим, что противник может определить значения k_0, k_1, k_2, k_3. Тогда:

$k_1=(a*k_0+b) \bmod c$

$k_2=(a*k_1+b) \bmod c$

$k_3=(a*k_2+b) \bmod c$

Решив систему из этих трех уравнений, можно найти a, b и c.

Для получения псевдослучайных чисел предлагалось использовать также квадратичные и кубические генераторы:

$k_i=(a_1^2*k_{i-1}+a_2*k_{i-1}+b) \bmod c$

$k_i=(a_1^3*k_{i-1}+a_2^2*k_{i-1}+a_3*k_{i-1}+b) \bmod c$

Однако такие генераторы тоже оказались непригодными для целей криптографии по той же самой причине "предсказуемости".

Метод Фибоначчи с запаздыванием

Известны и другие схемы получения псевдослучайных чисел.

Метод Фибоначчи с запаздываниями (Lagged Fibonacci Generator) — один из методов генерации псевдослучайных чисел. Он позволяет получить более высокое "качество" псевдослучайных чисел.

Наибольшую популярность фибоначчиевы датчики получили в связи с тем, что скорость выполнения арифметических операций с вещественными числами сравнялась со скоростью целочисленной арифметики, а фибоначчиевы датчики естественно реализуются в вещественной арифметике.

Известны разные схемы использования метода Фибоначчи с запаздыванием. Один из широко распространённых фибоначчиевых датчиков основан на следующей рекуррентной формуле:

$$k_i = \begin{cases} k_{i-a} - k_{i-b}, & если\ k_{i-a} \geq k_{i-b} \\ k_{i-a} - k_{i-b} + 1, & если\ k_{i-a} < k_{i-b} \end{cases}$$

где k_i — вещественные числа из диапазона [0,1], a, b — целые положительные числа, параметры генератора. Для работы фибоначчиеву датчику требуется знать max{a,b} предыдущих сгенерированных случайных чисел. При программной реализации для хранения сгенерированных случайных чисел необходим некоторый объем памяти, зависящих от параметров a и b.

Пример. Вычислим последовательность из первых десяти чисел, генерируемую методом Фибоначчи с запаздыванием начиная с k_5 при следующих исходных данных: a = 4, b = 1, k_0=0.1; k_1=0.7; k_2=0.3; k_3=0.9; k_4=0.5:

$k_5 = k_1 - k_4 = 0.7 - 0.5 = 0.2$;

$k_6 = k_2 - k_5 = 0.3 - 0.2 = 0.1$;

$k_7 = k_3 - k_6 = 0.9 - 0.1 = 0.8$;

$k_8 = k_4 - k_7 + 1 = 0.5 - 0.8 + 1 = 0.7$;

$k_9 = k_5 - k_8 + 1 = 0.2 - 0.7 + 1 = 0.5$;

$k_{10} = k_6 - k_9 + 1 = 0.1 - 0.5 + 1 = 0.6$;

$k_{11} = k_7 - k_{10} = 0.8 - 0.6 = 0.2$;

$k_{12} = k_8 - k_{11} = 0.7 - 0.2 = 0.5$;

$k_{13} = k_9 - k_{12} + 1 = 0.5 - 0.5 + 1 = 1$;

$k_{14} = k_{10} - k_{13} + 1 = 0.6 - 1 + 1 = 0.6$.

Видим, что генерируемая последовательность чисел внешне похожа на случайную. И действительно, исследования подтверждают, что получаемые случайные числа обладают хорошими статистическими свойствами.

Для генераторов, построенных по методу Фибоначчи с запаздыванием, существуют рекомендуемые параметры a и b, так сказать, протестированные на качество. Например, исследователи предлагают следующие значения: (a,b) = (55, 24), (17, 5) или (97,33). Качество получаемых случайных чисел зависит от значения константы a: чем оно больше, тем выше размерность пространства, в котором сохраняется равномерность случайных векторов, образованных из полученных случайных чисел. В то же время с увеличением величины константы a увеличивается объём используемой алгоритмом памяти.

В результате значения (a,b) = (17,5) рекомендуются для простых приложений. Значения (a,b) = (55,24) позволяют получать числа, удовлетворительные для большинства криптографических алгоритмов, требовательных к качеству случайных чисел. Значения (a,b) = (97,33) позволяют получать очень качественные случайные числа и используются в алгоритмах, работающих со случайными векторами высокой размерности.

Генераторы ПСЧ, основанные на методе Фибоначчи с запаздыванием, использовались для целей криптографии. Кроме того, они применяются в математических и статистических расчетах, а также при моделировании случайных процессов. Генератор ПСЧ, построенный на основе метода Фибоначчи с запаздыванием, использовался в широко известной системе Matlab.

12. **Шифрование методом гаммирования. Потоковые шифры**

В аддитивных шифрах используется сложение по модулю (mod) исходного сообщения с гаммой, представленных в числовом виде. Напомним, что результатом сложения двух целых чисел по модулю является остаток от деления (например, 5+10 mod 4 = 15 mod 4 = 3).

В литературе шифры этого класса часто называют потоковыми, хотя к потоковым относятся и другие разновидности шифров. Стойкость закрытия этими шифрами определяется, главным образом, качеством гаммы, которое зависит от длины периода и случайности распределения по периоду [8]. При этом символы в пределах периода гаммы являются ключом шифра.

Длиною периода гаммы называется минимальное количество символов, после которого последовательность цифр в гамме начинает повторяться. Случайность распределения символов по периоду означает отсутствие закономерностей между появлением различных символов в пределах периода.

По длине периода различаются гаммы с конечным и бесконечным периодом. Если длина периода гаммы превышает длину шифруемого текста, гамма является истинно случайной и не используется для шифрования других сообщений, то такое преобразование является абсолютно стойким (совершенный шифр).

Сложение по модулю N. В 1888 г. француз маркиз де Виари в одной из своих научных статей, посвященных криптографии, доказал, что при замене букв исходного сообщения и ключа на числа справедливы формулы

Ci = (Pi + Ki) mod N, (6.1)

Pi = (Ci + N - Ki) mod N, (6.2)

где Pi, Ci - i-ый символ открытого и шифрованного сообщения;

N - количество символов в алфавите;

Ki - i-ый символ гаммы (ключа). Если длина гаммы меньше, чем длина сообщения, то она используется повторно.

Данные формулы позволяют выполнить зашифрование / расшифрование по Виженеру при замене букв алфавита числами согласно следующей таблице (применительно к русскому алфавиту):

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

Рис.6.1. Таблица кодирования символов

Например, для шифрования используется русский алфавит (N = 33), открытое сообщение – «АБРАМОВ», гамма – «ЖУРИХИН». При замене символов на числа буква А будет представлена как 0, Б – 1, …, Я – 32. Результат шифрования показан в следующей таблице.

Таблица 6.1. Пример аддитивного шифрования по модулю N = 33

Символ								
	открытого сообщения, P_i	А	Б	Р	А	М	О	В
		0	1	17	0	13	15	2
	гаммы, K_i	Ж	У	Р	И	Х	И	Н
		7	20	17	9	22	9	14
	шифрограммы, C_i	Ж	Ф	Б	И	В	Ч	П
		7	21	1	9	2	24	16

Сложение по модулю 2. Значительный успех в криптографии связан с именем американца Гильберто Вернама [15]. В 1917 г. он, будучи сотрудником телеграфной компании AT&T, совместно с Мейджором Джозефом Моборном предложил идею автоматического шифрования телеграфных сообщений. Речь шла о своеобразном наложении гаммы на знаки алфавита, представленные в соответствии с телетайпным кодом Бодо пятизначными «импульсными комбинациями». Например, буква А представлялась комбинацией («– – – + +»), а комбинация («+ + – + +») представляла символ перехода от букв к цифрам. На бумажной ленте, используемой при работе телетайпа, знаку «+» отвечало наличие отверстия, а знаку «–» - его отсутствие. При считывании с ленты металлические

щупы проходили через отверстия, замыкали электрическую цепь и, тем самым, посылали в линию импульс тока.

Вернам предложил электромеханически покоординатно складывать «импульсы» знаков открытого текста с «импульсами» гаммы, предварительно нанесенными на ленту. Сложение проводилось «по модулю 2». Имеется в виду, что если «+» отождествить с 1, а «–» с 0, то сложение определяется двоичной арифметикой:

⊕	0	1
0	0	1
1	1	0

Т.о., при данном способе шифрования символы текста и гаммы представляются в двоичных кодах, а затем каждая пара двоичных разрядов складывается по модулю 2 (⊕, для булевых величин аналог этой операции – XOR, «Исключающее ИЛИ»). Процедуры шифрования и дешифрования выполняются по следующим формулам

$Ci = Pi \oplus Ki$, (6.3)

$Pi = Ci \oplus Ki$. (6.4)

Вернам сконструировал и устройство для такого сложения. Замечательно то, что процесс шифрования оказывался полностью автоматизированным, в предложенной схеме исключался шифровальщик. Кроме того, оказывались слитыми воедино процессы зашифрования / расшифрования и передачи по каналу связи.

В 1918 г. два комплекта соответствующей аппаратуры были изготовлены и испытаны. Испытания дали положительные результаты. Единственное неудовлетворение специалистов - криптографов было связано с гаммой. Дело в том, что первоначально гамма была нанесена на ленту, склеенную в кольцо. Несмотря на то, что знаки гаммы на ленте выбирались случайно, при зашифровании длинных сообщений гамма регулярно повторялась. Этот недостаток так же отчетливо осознавался, как и для шифра Виженера. Уже тогда хорошо понимали, что повторное использование гаммы недопустимо даже в пределах одного сообщения. Попытки удлинить гамму приводили к неудобствам в работе с длинным кольцом. Тогда был предложен вариант с двумя лентами, одна из которых шифровала другую, в результате чего получалась гамма, имеющая длину периода, равную произведению длин исходных периодов.

Шифры гаммирования стали использоваться немцами в своих дипломатических представительствах в начале 20-х гг., англичанами и американцами – во время Второй мировой войны. Разведчики-нелегалы ряда государств использовали шифрблокноты1. Шифр Вернама (сложение по модулю 2) применялся на правительственной «горячей линии» между Вашингтоном и Москвой, где ключевые материалы представляли собой перфорированные бумажные ленты, производившиеся в двух экземплярах [23].

Перед иллюстрацией использования шифра приведем таблицу кодов символов Windows 1251 и их двоичное представление.

Таблица 6.2. Коды символов Windows 1251 и их двоичное представление

Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код	Буква	Dec-код	Bin-код
А	192	1100 0000	Л	203	1100 1011	Ц	214	1101 0110
Б	193	1100 0001	М	204	1100 1100	Ч	215	1101 0111
В	194	1100 0010	Н	205	1100 1101	Ш	216	1101 1000
Г	195	1100 0011	О	206	1100 1110	Щ	217	1101 1001
Д	196	1100 0100	П	207	1100 1111	Ъ	218	1101 1010
Е	197	1100 0101	Р	208	1101 0000	Ы	219	1101 1011
Ж	198	1100 0110	С	209	1101 0001	Ь	220	1101 1100
З	199	1100 0111	Т	210	1101 0010	Э	221	1101 1101
И	200	1100 1000	У	211	1101 0011	Ю	222	1101 1110
Й	201	1100 1001	Ф	212	1101 0100	Я	223	1101 1111

К	202	1100 1010	Х	213	1101 0101			

Примечание. Dec-код – десятичный код символа, Bin-код – двоичный код символа.

Пример шифрования сообщения «ВОВА» с помощью гаммы «ЮЛЯ» показан в следующей таблице.

Таблица 6.3. Пример аддитивного шифрования по модулю 2

Открытое сообщение, P_i	Буква	В	О	В	А
	Dec-код	194	206	194	192
	Bin-код	1100 0010	1100 1110	1100 0010	1100 0000
Гамма, K_i	Буква	Ю	Л	Я	Ю
	Dec-код	222	203	223	222
	Bin-код	1101 1110	1100 1011	1101 1111	1101 1110
Шифрограмма, C_i	Dec-код	28	5	29	30
	Bin-код	0001 1100	0000 0101	0001 1101	0001 1110

Шифрование по модулю 2 обладает замечательным свойством, вместо истинной гаммы противнику можно сообщить ложную гамму, которая при наложении на шифрограмму даст осмысленное выражение.

Таблица 6.4. Пример использования ложной гаммы

Шифрограмма, C_i	Dec-код	28	5	29	30
	Bin-код	0001 1100	0000 0101	0001 1101	0001 1110
Ложная гамма, K'_i	Буква	Ю	Е	М	Б
	Dec-код	222	197	204	193
	Bin-код	1101 1110	1100 0101	1100 1100	1100 0001
Ложное открытое сообщение, P'_i	Dec-код	194	192	209	223
	Bin-код	1100 0010	1100 0000	1101 0001	1101 1111
	Буква	В	А	С	Я

В 2013 г. ученые Корнельского университета предложили использовать для генерации ключей (шифроблокнотов) куски полупрозрачного стекла [42]. Кратко принцип нового метода шифрования заключается в том, что отправитель и получатель (Алиса и Боб) во время встречи формируют общий ключ, облучая кусочки стекла изображением-паттерном с IDi (внешне он напоминает QR-код). В результате отражения и преломления, характер которого индивидуален для каждого куска стекла, у Алисы и Боба получаются собственные случайные изображения, которые затем оцифровываются (получаются ключи keyAi и keyBi). Из этих изображений и составляется общий ключ (keyABi = keyAi $\oplus$ keyBi).

Потоковые шифры

Потоковый шифр - симметричный тип шифра, где каждый элемент открытого текста переводится в зашифрованный вид, в зависимости от применяемого ключа и его позиции в текстовом потоке. Особенность поточного шифра - иной подход к процессу шифрования в сравнении с блочным типом шифра.

Потоковый шифр - вид шифра, в котором каждый бит данных зашифровывается посредством гаммирования. Процесс гаммирования подразумевает наложение на информацию гамм кода по строго определенным правилам. Чтобы расшифровать данные, требуется наложение той же гаммы на зашифрованный текст.

Первые потоковые шифры были использованы еще во времена Второй мировой войны (до появления электроники). Более чем через два десятка лет (в 1965 году) норвежский криптограф Эранст Селмер разработал свою теорию последовательности регистровых сдвигов. Еще через время Соломон

Голомб написал книгу о последовательности сдвиговых регистров. При этом популярность потоковым шифрам пришла раньше - в 1949 году, когда миру была представлена работа Клода Шеннона о стойкости шифра Вернама.

Принцип потокового шифрования. Генератор случайных чисел выдает определенную гамму (числовую последовательность). Последняя накладывается на шифруемую информацию с применением операции XOR. На выходе получаются зашифрованные данные. Наиболее популярный потоковый шифр - RC4, признанный органами стандартизации. Надежность потокового шифрования зависит от числовой последовательности, выдаваемой генератором.

Сфера применения потоковых шифров - военные, сетевые, телефонные и другие системы, где необходимо преобразование речевой информации в цифровую форму и надежное шифрование данных. Причина популярности - простота реализации и конструирования генераторов, надежность шифрования, отсутствие ошибок в потоковом шифре.

В проектировании поточных шифров применяется один из следующих подходов:

- системно-теоретический - основан на создании проблемы, которая еще не известна и не исследована криптоаналитиком;
- информационно-технический - базируется на попытке спрятать разгадку от криптоаналитика;
- сложно-теоретический - подход, базирующийся на известной, хоть и сложной проблеме;
- рандомизированный - подход, где создается объемная задача, решение которой выглядит невозможным.

13.Общие принципы построения современных симметричных криптосистем. Общая характеристика блочных шифров. Криптоалгоритм DES. Криптоалгоритм ГОСТ 28147-89

Симметричные криптосистемы (с секретным ключом – secret key systems) –построены на основе сохранения в тайне ключа шифрования. Процессы зашифрования и расшифрования используют один и тот же ключ. Секретность ключа является постулатом. Основная проблема при применении симметричных криптосистем для связи заключается в сложности передачи обоим сторонам секретного ключа. Однако данные системы обладают высоким быстродействием. Раскрытие ключа злоумышленником грозит раскрытием только той информации, что была зашифрована на этом ключе. Американский и Российский стандарты шифрования DES и ГОСТ 28.147-89, а также новый стандарт AES Rijndael – все эти алгоритмы являются представителями симметричных криптосистем.

Симметричные криптосистемы в настоящее время принято подразделять на блочные и поточные.

Блочные криптосистемы разбивают текст сообщения на отдельные блоки и затем осуществляют преобразование этих блоков с использованием ключа.

Поточные криптосистемы работают несколько иначе. На основе ключа системы вырабатывается некая последовательность – так называемая гамма, которая затем накладывается на текст сообщения. Таким образом, преобразование текста осуществляется как бы потоком по мере выработки гаммы.

Общая структура использования симметричной криптосистемы представлена на рис.13.

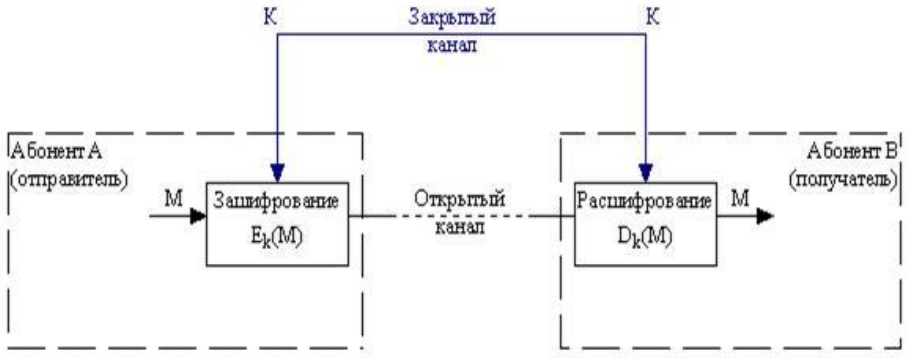


Рис. 13. Структура симметричной криптосистемы

Здесь M – открытый текст, K – секретный ключ, передаваемый по закрытому каналу, Ек(M) – операция зашифрования, а Dk(M) – операция расшифрования.

Все многообразие существующих симметричных криптографических методов можно свести к следующим классам преобразований:

1. Моно- и многоалфавитные подстановки (замены). Наиболее простой вид преобразований, заключающийся в замене символов исходного текста на другие (обычно того же алфавита) по более или менее сложному правилу. Для обеспечения высокой криптостойкости требуется использование больших ключей.

2. Перестановки. Символы исходного текста переставляются по некоторому правилу. Используется, как правило, в сочетании с другими методами.

3. Гаммирование. Этот метод заключается в наложении на исходный текст некоторой псевдослучайной последовательности, генерируемой на основе ключа.

4. Блочные шифры. Криптосистемы с секретным ключом подразделяются на два вида: блочные (block) и поточные (stream). Поточные криптосистемы работают с сообщением как с единым потоком, блочные криптосистемы представляют собой блочные (групповые) шифропреобразования. Блочная криптосистема разбивает открытый текст на последовательные блоки и зашифровывает каждый блок с помощью одного и того же обратимого преобразования, выбранного с помощью ключа. Любое из них можно рассматривать как последовательность операций, проводимых с элементами ключа и открытого текста, а так же производными от них величинами. Произвол в выборе элементов алгоритма шифрования достаточно велик, однако "элементарные" операции должны обладать хорошим

криптографическими свойствами и допускать удобную техническую или программную реализацию. Обычно используются операции:

- побитового сложения по модулю 2 двоичных векторов (XOR)
- сложение или умножение целых чисел по некоторому модулю
- перестановка битов двоичных векторов;
- табличная замена элементов двоичных векторов.

Блочные шифры

Блочные шифры оперируют с блоками открытого текста и используют простую замену блоков. Основные процедуры, используемые при получении таких шифров сводятся к следующему:

−рассеивание (diffusion) – изменение любого знака открытого текста или ключа влияет на большое число знаков шифротекста, что скрывает статистические свойства открытого текста;

−перемешивание (confusion) – использование преобразований, затрудняющих получение статистических зависимостей между шифротектстом и открытым текстом.

В блочных шифрах, когда длина блока достаточно велика, таблица замены становится необозримой и саму замену приходится задавать не таблицей, а неким алгоритмом преобразования.

Практически все современные блочные шифры являются композиционными – то есть состоят из композиции простых преобразований. Само по себе преобразование может и не обеспечивать нужных свойств, но их цепочка позволяет получить необходимый результат.

Американский стандарт шифрования данных DES. Стандарт шифрования данных DES (Data Encryption Standard) опубликован в 1977 г. Национальным бюро стандартов США. Он предназначен для защиты от несанкционированного доступа к важной, но не секретной информации в государственных и коммерческих организациях США.

Основные достоинства алгоритма DES:

- используется только один ключ длиной 56 бит;
- относительная простота алгоритма обеспечивает высокую скорость обработки;
- достаточно высокая стойкость алгоритма.

Алгоритм DES основан на комбинировании методов подстановки и перестановки и состоит из чередующейся последовательности блоков перестановки и подстановки. DES осуществляет шифрование 64-битовых блоков данных с помощью 64-битового ключа, в котором значащими являются 56 бит (остальные 8 бит – проверочные биты для контроля на четность). Расшифрование в DES является операцией, обратной шифрованию, и выполняется путем повторения операций шифрования в обратной последовательности. Обобщенная схема процесса шифрования в алгоритме DES показана на рисунке 6.5.

Рисунок 6.5 – Обобщенная схема шифрования в алгоритме DES

Процесс шифрования заключается в начальной перестановке битов 64-битового блока, шестнадцати циклах шифрования и, наконец, в конечной перестановке битов.

Стандарт шифрования данных (ГОСТ 28147-89). Алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ был разработан в СССР и опубликован в виде государственного стандарта ГОСТ 28147-89 в 1989 году. Алгоритм криптографического преобразования данных предназначен для аппаратной и программной реализации, удовлетворяет криптографическим требованиям и не

накладывает ограничений на степень секретности защищаемой информации. Алгоритм шифрования данных представляет собой 64-битовый блочный алгоритм с 256-битовым ключом.

Стандарт шифрования гост 28147-89

Краткое описание шифра

ГОСТ 28147-89 — советский и российский стандарт симметричного шифрования, введённый в 1990 году, также является стандартом СНГ. Полное название — «ГОСТ 28147-89 Системы обработки информации. Защита криптографическая. Алгоритм криптографического преобразования». Блочный шифроалгоритм. При использовании метода шифрования с гаммированием, может выполнять функции поточного шифроалгоритма.

ГОСТ 28147-89 — блочный шифр с 256-битным ключом и 32 циклами преобразования, оперирующий 64-битными блоками. Основа алгоритма шифра — Сеть Фейстеля. Базовым режимом шифрования по ГОСТ 28147-89 является режим простой замены (определены также более сложные режимы гаммирование, гаммирование с обратной связью и режим имитовставки).

Принцип работы алгоритма

Алгоритм принципиально не отличается от DES. В нем также происходят циклы шифрования (их 32) по схеме Фейстеля (Рис. 2.9.).

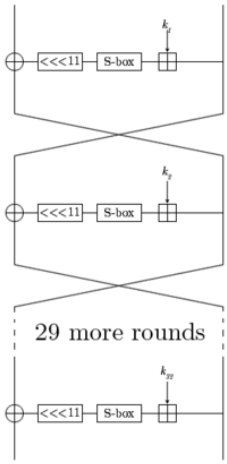


Рис. 2.9. Раунды шифрования алгоритма ГОСТ 28147-89.

Для генерации подключей исходный 256-битный ключ разбивается на восемь 32-битных блоков: $k_1 \ldots k_8$. Ключи $k_9 \ldots k_{24}$ являются циклическим повторением ключей $k_1 \ldots k_8$ (нумеруются от младших битов к старшим). Ключи $k_{25} \ldots k_{32}$ являются ключами $k_1 \ldots k_8$, идущими в обратном порядке.

После выполнения всех 32 раундов алгоритма, блоки A_{33} и B_{33} склеиваются (следует обратить внимание на то, что старшим битом становится A_{33}, а младшим - B_{33}) – результат есть результат работы алгоритма.

Функция $f(A_i,K_i)$ вычисляется следующим образом: A_i и K_i складываются по модулю 2^{32}, затем результат разбивается на восемь 4-битовых подпоследовательностей, каждая из которых поступает на вход своего узла таблицы замен (в порядке возрастания старшинства битов), называемого ниже S-блоком. Общее количество S-блоков ГОСТа — восемь, т. е. столько же, сколько и подпоследовательностей. Каждый S-блок представляет собой перестановку чисел от 0 до 15. Первая 4-битная подпоследовательность попадает на вход первого S-блока, вторая — на вход второго и т. д. Выходы всех восьми S-блоков объединяются в 32-битное слово, затем всё слово циклически сдвигается влево (к старшим разрядам) на 11 битов. Все восемь S-блоков могут быть различными. Фактически, они могут являться дополнительным ключевым материалом, но чаще являются параметром схемы, общим для определенной группы пользователей. В тексте стандарта указывается, что поставка заполнения узлов замены (S-блоков) производится в установленном порядке, т.е. разработчиком алгоритма. Сообщество российских разработчиков СКЗИ согласовала используемые в Интернет узлы замены.

Расшифрование выполняется так же, как и зашифрование, но инвертируется порядок подключей k_i.

Режимы работы алгоритма ГОСТ 28147-89

Алгоритм ГОСТ 28147-89 имеет четыре режима работы.

1.Режим простой замены принимает на вход данные, размер которых кратен 64-м битам. Результатом шифрования является входной текст, преобразованный блоками по 64 бита в случае зашифрования циклом «32-З», а в случае расшифрования — циклом «32-Р».

2.Режим гаммирования принимает на вход данные любого размера, а также дополнительный 64-битовый параметр — синхропосылку. В ходе работы синхропосылка преобразуется в цикле «32-З», результат делится на две части. Первая часть складывается по модулю 2^{32} с постоянным значением 1010101_{16}. Если вторая часть равна 2^{32}-1, то её значение не меняется, иначе она складывается по модулю 2^{32}-1 с постоянным значением 1010104_{16}. Полученное объединением обеих преобразованных частей значение, называемое гаммой шифра, поступает в цикл «32-З», его результат порязрядно складывается по модулю 2 с 64-разрядным блоком входных данных. Если последний меньше 64-х разрядов, то лишние разряды полученного значения отбрасываются. Полученное значение подаётся на выход. Если ещё имеются входящие данные, то действие повторяется: составленный из 32-разрядных частей блок преобразуется по частям и так далее.

3.Режим гаммирования с обратной связью также принимает на вход данные любого размера и синхропосылку. Блок входных данных поразрядно складывается по модулю 2 с результатом преобразования в цикле «32-З» синхропосылки. Полученное значение подаётся на выход. Значение синхропосылки заменяется в случае зашифрования выходным блоком, а в случае расшифрования — входным, то есть зашифрованным. Если последний блок входящих данных меньше 64 разрядов, то лишние разряды гаммы (выхода цикла «32-З») отбрасываются. Если ещё имеются входящие данные, то действие повторяется: из результата зашифрования заменённого значения образуется гамма шифра и т.д.

4.Режим выработки имитовставки принимает на вход данные, размер которых составляет не меньше двух полных 64-разрядных блоков, а возвращает 64-разрядный блок данных, называемый имитовставкой. Временное 64-битовое значение устанавливается в 0, далее, пока имеются входные данные, оно поразрядно складывается по модулю 2 с результатом выполнения цикла «16-З», на вход которого подаётся блок входных данных. После окончания входных данных временное значение возвращается как результат.

Криптоанализ шифра

В шифре ГОСТ 28147-89 используется 256-битовый ключ и объем ключевого пространства составляет 2^{256}. Ни на одном из существующих в настоящее время компьютере общего применения нельзя подобрать ключ за время, меньшее многих сотен лет. Российский стандарт ГОСТ 28147-89 проектировался с большим запасом и по стойкости на много порядков превосходит американский стандарт DES с его реальным размером ключа в 56 бит и объемом ключевого пространства всего 2^{56}.

Существуют атаки и на полнораундовый ГОСТ 28147—89 без каких-либо модификаций. Одна из первых открытых работ, в которых был проведен анализ алгоритма, использует слабости процедуры расширения ключа ряда известных алгоритмов шифрования. В частности, полнораундовый алгоритм ГОСТ 28147—89 может быть вскрыт с помощью дифференциального криптоанализа на связанных ключах, но только в случае использования слабых таблиц замен. 24-раундовый вариант алгоритма (в котором отсутствуют первые 8 раундов) вскрывается аналогичным образом при любых таблицах замен, однако, сильные таблицы замен делают такую атаку абсолютно непрактичной.

Отечественные ученые А.Г. Ростовцев и Е.Б. Маховенко в 2001 г. предложили принципиально новый метод криптоанализа путем формирования целевой функции от известного открытого текста, соответствующего ему шифртекста и искомого значения ключа и нахождения ее экстремума, соответствующего истинному значению ключа. Они же нашли большой класс слабых ключей алгоритма ГОСТ 28147—89, которые позволяют вскрыть алгоритм с помощью всего 4-х выбранных открытых текстов и соответствующих им шифротекстов с достаточно низкой сложностью.

В 2004 году группа специалистов из Кореи предложила атаку, с помощью которой, используя дифференциальный криптоанализ на связанных ключах, можно получить с вероятностью 91,7% 12 бит секретного ключа. Для атаки требуется 2^{35} выбранных открытых текстов и 2^{36} операций шифрования. Как видно, данная атака практически бесполезна для реального вскрытия алгоритма.

Таблица замен является долговременным ключевым элементом, то есть действует в течение гораздо более длительного срока, чем отдельный ключ. Предполагается, что она является общей для всех узлов шифрования в рамках одной системы криптографической защиты. От качества этой таблицы зависит качество шифра. При "сильной" таблице замен стойкость шифра не опускается ниже

некоторого допустимого предела даже в случае ее разглашения. И наоборот, использование "слабой" таблицы может уменьшить стойкость шифра до недопустимо низкого предела. Никакой информации по качеству таблицы замен в открытой печати России не публиковалось, однако существование "слабых" таблиц не вызывает сомнения - примером может служить "тривиальная" таблица замен, по которой каждое значение заменяется на него самого. В ряде работ ошибочно делается вывод о том, что секретные таблицы замен алгоритма ГОСТ 28147-89 могут являться частью ключа и увеличивать его эффективную длину (что несущественно, поскольку алгоритм обладает весьма большим 256-битным ключом).

14.Общие принципы построения современных асимметричных криптосистем. Асимметричные криптоалгоритмы RSA и Рабина

Асимметричные криптосистемы шифрования

Асимметричные криптографические системы были разработаны в 1970-х гг. Принципиальное отличие асимметричной криптосистемы от криптосистемы симметричного шифрования состоит в том, что для шифрования информации и ее последующего расшифровывания используются различные ключи:

• открытый ключ К используется для шифрования информации, вычисляется из секретного ключа к;

• секретный ключ к используется для расшифровывания информации, зашифрованной с помощью парного ему открытого ключа К.

Эти ключи различаются таким образом, что с помощью вычислений нельзя вывести секретный ключ к из открытого ключа К. Поэтому открытый ключ К может свободно передаваться по каналам связи.

Асимметричные системы называют также двухключевыми криптографическими системами, или криптосистемами с открытым ключом.

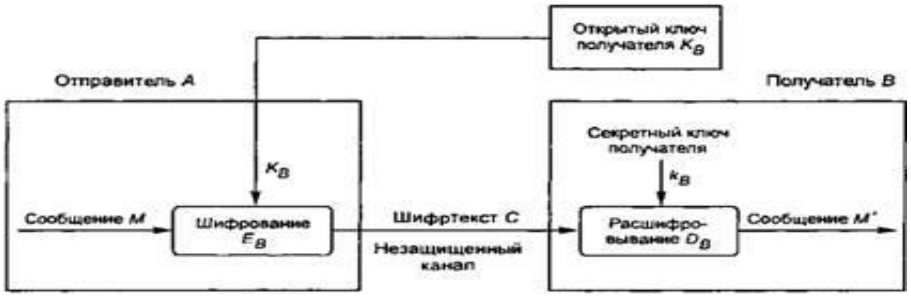


Рис. 5.3. Обобщенная схема асимметричной криптосистемы шифрования

Обобщенная схема асимметричной криптосистемы шифрования с открытым ключом показана на рис. 5.3.Для криптографического закрытия и последующего расшифровывания передаваемой информации используются открытый и секретный ключи получателя В сообщения.

В качестве ключа зашифровывания должен использоваться открытый ключ получателя, а в качестве ключа расшифровывания - его секретный ключ.

Секретный и открытый ключи генерируются попарно. Секретный ключ должен оставаться у его владельца и быть надежно защищен от НСД (аналогично ключу шифрования в симметричных алгоритмах). Копия открытого ключа должна находиться у каждого абонента криптографической сети, с которым обменивается информацией владелец секретного ключа.

Процесс передачи зашифрованной информации в асимметричной криптосистеме осуществляется следующим образом.

Подготовительный этап:

• абонент В генерирует пару ключей: секретный ключ $к_в$ и открытый ключ $К_в$;

• открытый ключ $К_в$ посылается абоненту А и остальным абонентам (или делается доступным, например на разделяемом ресурсе).

Использование — обмен информацией между абонентами А и В:

• абонент А зашифровывает сообщение с помощью открытого ключа $К_в$ абонента В и отправляет шифртекст абоненту В;

• абонент В расшифровывает сообщение с помощью своего секретного ключа $к_в$. Никто другой (в том числе абонент А) не может расшифровать данное сообщение, так как не имеет секретного ключа абонента В. Защита информации в асимметричной криптосистеме основана на секретности ключа $к_в$ получателя сообщения.

Характерные особенности асимметричных криптосистем:

• открытый ключ $К_в$ и криптограмма С могут быть отправлены по незащищенным каналам, т. е. противнику известны $К_в$ и С;

• алгоритмы шифрования и расшифровывания: $E_в : M \rightarrow C$; $D_B : C \rightarrow M$ являются открытыми.

У. Диффи и М. Хеллман сформулировали требования, выполнение которых обеспечивает безопасность асимметричной криптосистемы.

1. Вычисление пары ключей ($К_в$, $к_в$) получателем В должно быть простым.

2. Отправитель А, зная открытый ключ $К_в$ и сообщение М, может легко вычислить криптограмму $С=E_{Кв}(М)$.

3. Получатель В, используя секретный ключ $к_в$ и криптограмму С, может легко восстановить исходное сообщение $М=О_{кя}(С)$.

4. Противник, зная открытый ключ $К_в$, при попытке вычислить секретный ключ $к_в$ наталкивается на непреодолимую вычислительную проблему.

5. Противник, зная пару ($К_в$, С), при попытке вычислить исходное сообщение М наталкивается на непреодолимую вычислительную проблему.

Концепция асимметричных криптографических систем с открытым ключом основана на применении однонаправленных функций. Однонаправленной функцией называется функция F(X), обладающая двумя свойствами:

• существует алгоритм вычисления значений функции Y= F(X);

• не существует эффективного алгоритма обращения (инвертирования) функции F (т. е. не существует решения уравнения F(X) = Y относительно X).

В качестве примера однонаправленной функции можно указать целочисленное умножение. Прямая задача — вычисление произведения двух очень больших целых чисел P и Q, т. е. нахождение значения N = P × Q — относительно несложная задача для компьютера.

Обратная задача — факторизация, или разложение на множители большого целого числа, т. е. нахождение делителей P и Q большого целого числа N = P × Q, — является практически неразрешимой при достаточно больших значениях N.

Другой характерный пример однонаправленной функции — это модульная экспонента с фиксированными основанием и модулем.

Как и в случае симметричных криптографических систем, с помощью асимметричных криптосистем обеспечивается не только конфиденциальность, но также подлинность и целостность передаваемой информации. Подлинность и целостность любого сообщения обеспечивается формированием цифровой подписи этого сообщения и отправкой в зашифрованном виде сообщения вместе с цифровой подписью. Проверка соответствия подписи полученному сообщению после его предварительного расшифровывания представляет собой проверку целостности и подлинности принятого сообщения. Процедуры формирования и проверки электронной цифровой подписи рассмотрены в разделе «Электронная цифровая подпись и функция хэширования».

Преимущества асимметричных криптографических систем перед симметричными криптосистемами:

• в асимметричных криптосистемах решена сложная проблема распределения ключей между пользователями, так как каждый пользователь может сгенерировать свою пару ключей сам, а открытые ключи пользователей могут свободно публиковаться и распространяться по сетевым коммуникациям;

• исчезает квадратичная зависимость числа ключей от числа пользователей; в асимметричной криптосистеме число используемых ключей связано с числом абонентов линейной зависимостью (в системе из N пользователей используются 2Nключей), а не квадратичной, как в симметричных системах;

• асимметричные криптосистемы позволяют реализовать протоколы взаимодействия сторон, которые не доверяют друг другу, поскольку при использовании асимметричных криптосистем закрытый ключ должен быть известен только его владельцу.

Недостатки асимметричных криптосистем:

• на настоящий момент нет математического доказательства необратимости используемых в асимметричных алгоритмах функций;

• асимметричное шифрование существенно медленнее симметричного, поскольку при шифровании и расшифровке используются весьма ресурсоемкие операции. По этой же причине реализовать аппаратный шифратор с асимметричным алгоритмом существенно сложнее, чем реализовать аппаратно симметричный алгоритм;

• необходимость защиты открытых ключей от подмены.

15. Функции хеширования и целостность данных. Криптографические функции хеширования. Хеш-функции на основе симметричных блочных алгоритмов

Что такое хеш и хэширование простыми словами

Слово хеш происходит от английского «hash», одно из значений которого трактуется как путаница или мешанина. Собственно, это довольно полно описывает реальное значение этого термина. Часто еще про такой процесс говорят «хеширование», что опять же является производным от английского hashing (рубить, крошить, спутывать...).

Появился этот термин в середине прошлого века среди людей занимающихся обработках массивов данных. Хеш-функция позволяла привести любой массив данных к числу заданной длины. Например, если любое число (любой длинны) начать делить много раз подряд на одно и то же простое число, то полученный в результате остаток от деления можно будет называть хешем. Для разных исходных чисел остаток от деления (цифры после запятой) будет отличаться.

Для обычного человека это кажется белибердой, но как ни странно в наше время без хеширования практически невозможна работа в интернете. Так что же это такая за функция? На самом деле она может быть любой (приведенный выше пример это не есть реальная функция — он придуман мною чисто для вашего лучшего понимания принципа). Главное, чтобы результаты ее работы удовлетворяли приведенным ниже условиям.

Зачем нужен хэш

Смотрите, еще пример. Есть у вас текст в файле. Но на самом деле это ведь не текст, а массив цифровых символов (по сути число). Как вы знаете, в компьютерной логике используются двоичные числа (ноль и единица). Они запросто могут быть преобразованы в шестнадцатиричные цифры, над которыми можно проводить математические операции. Применив к ним хеш-функцию мы получим на выходе (после ряда итераций) число заданной длины (хеш-сумму).

Если мы потом в исходном текстовом файле поменяем хотя бы одну букву или добавим лишний пробел, то повторно рассчитанный для него хэш уже будет отличаться от изначального (вообще другое число будет), зачем все это нужно? Ну, конечно же, для того, чтобы понять, что файл именно тот, что и должен быть. Это можно использовать в целом ряде аспектов работы в интернете и без этого вообще сложно представить себе работу сети.

Где и как используют хеширование

Например, простые хэш-функции (не надежные, но быстро рассчитываемые) применяются при проверке целостности передачи пакетов по протоколу TCP/IP (и ряду других протоколов и алгоритмов, для выявления аппаратных ошибок и сбоев — так называемое избыточное кодирование). Если рассчитанное значение хеша совпадает с отправленным вместе с пакетом (так называемой контрольной суммой), то значит потерь по пути не было (можно переходить к следующему пакету).

А это, ведь на минутку, основной протокол передачи данных в сети интернет. Без него никуда. Да, есть вероятность, что произойдет накладка — их называют коллизиями. Ведь для разных изначальных данных может получиться один и тот же хеш. Чем проще используется функция, тем выше такая вероятность. Но тут нужно просто выбирать между тем, что важнее в данный момент — надежность идентификации или скорость работы. В случае TCP/IP важна именно скорость. Но есть и другие области, где важнее именно надежность.

Похожая схема используется и в технологии блокчейн, где хеш выступает гарантией целостности цепочки транзакций (платежей) и защищает ее от несанкционированных изменений. Благодаря ему и распределенным вычислениям взломать блокчен очень сложно и на его основе благополучно существует множество криптовалют, включая самую популярную из них — это биткоин. Последний существует уже с 2009 год и до сих пор не был взломан.

Более сложные хеш-функции используются в криптографии. Главное условие для них — невозможность по конечному результату (хэшу) вычислить начальный (массив данных, который обработали данной хеш-функцией). Второе главное условие — стойкость к коллизиями, т.е. низкая вероятность получения двух одинаковых хеш-сумм из двух разных массивов данных при обработке их этой функцией. Расчеты по таким алгоритмам более сложные, но тут уже главное не скорость, а надежность.

Так же хеширование используется в технологии электронной цифровой подписи. С помощью хэша тут опять же удостоверяются, что подписывают именно тот документ, что требуется. Именно он (хеш) передается в токен, который и формирует электронную цифровую подпись.

Для доступа к сайтам и серверам по логину и паролю тоже часто используют хеширование. Согласитесь, что хранить пароли в открытом виде (для их сверки с вводимыми пользователями) довольно ненадежно (могут их похитить). Поэтому хранят хеши всех паролей. Пользователь вводит символы своего пароля, мгновенно рассчитывается его хеш-сумма и сверяется с тем, что есть в базе. Надежно и очень просто. Обычно для такого типа хеширования используют сложные функции с очень высокой криптостойкостью, чтобы по хэшу нельзя было бы восстановить пароль.

Коллизии хэш-функций

В теории хэш-функций предусмотрено такое явление, как коллизия. В чем его сущность? Коллизия хэш-функции - ситуация, при которой два разных файла имеют одинаковый хэш-код. Это возможно, если длина целевой последовательности символов будет небольшой. В этом случае вероятность совпадения хэша будет выше. Для того чтобы избежать коллизии, рекомендуется, в частности, задействовать двойной алгоритм под названием "хеширование хеш-функции". Он предполагает формирование открытого и закрытого кода. Многие программисты при решении ответственных задач рекомендуют не применять хэш-функции в тех случаях, когда это необязательно и всегда тестировать соответствующие алгоритмы на предмет наилучшей совместимости с теми или иными ключами.

Какими свойствами должна обладать хеш-функция

Хочу систематизировать кое-что из уже сказанного и добавить новое.

1. Как уже было сказано, функция эта должна уметь приводить любой объем данных (а все они цифровые, т.е. двоичные, как вы понимаете) к числу заданной длины (по сути это сжатие до битовой последовательности заданной длины хитрым способом).
2. При этом малейшее изменение (хоть на один бит) входных данных должно приводить к полному изменению хеша.
3. Она должна быть стойкой в обратной операции, т.е. вероятность восстановления исходных данных по хешу должна быть весьма низкой (хотя последнее сильно зависит от задействованных мощностей)
4. В идеале она должна иметь как можно более низкую вероятность возникновения коллизий. Согласитесь, что не айс будет, если из разных массивов данных будут часто получаться одни и те же значения хэша.
5. Хорошая хеш-функция не должна сильно нагружать железо при своем исполнении. От этого сильно зависит скорость работы системы на ней построенной. Как я уже говорил выше, всегда имеется компромисс между скорость работы и качеством получаемого результата.
6. Алгоритм работы функции должен быть открытым, чтобы любой желающий мог бы оценить ее криптостойкость, т.е. вероятность восстановления начальных данных по выдаваемому хешу.

Хеш — это маркер целостности скачанных в сети файлов

Где еще можно встретить применение этой технологии? Наверняка при скачивании файлов из интернета вы сталкивались с тем, что там приводят некоторые числа (которые называют либо хешем, либо контрольными суммами) типа:

```
CRC32: 7438E546
MD5: DE3BAC46D80E77ADCE8E379F682332EB
SHA-1: 332B317FB97126B0F79F7AF5786EBC51E5CC82CF
```

Что это такое? И что вам с этим всем делать? Ну, как правило, на тех же сайтах можно найти пояснения по этому поводу, но я не буду вас утруждать и расскажу в двух словах. Это как раз и есть результаты работы различных хеш-функций (их названия приведены перед числами: CRC32, MD5 и SHA-1).

Зачем они вам нужны? Ну, если вам важно знать, что при скачивании все прошло нормально и ваша копия полностью соответствует оригиналу, то нужно будет поставить на свой компьютер программку, которая умеет вычислять хэш по этим алгоритмам (или хотя бы по некоторым их них).

После чего прогнать скачанные файлы через эту программку и сравнить полученные числа с приведенными на сайте. Если совпадают, то сбоев при скачивании не было, а если нет, то значит были сбои и есть смысл повторить закачку заново.

Популярные хэш-алгоритмы сжатия

1. CRC32 — используется именно для создания контрольных сумм (так называемое избыточное кодирование). Данная функция не является криптографической. Есть много вариаций этого алгоритма (число после CRC означает длину получаемого хеша в битах), в зависимости от нужной длины получаемого хеша. Функция очень простая и нересурсоемкая. В связи с этим используется для проверки целостности пакетов в различных протоколах передачи данных.

2. MD5 — старая, но до сих пор очень популярная версия уже криптографического алгоритма, которая создает хеш длиной в 128 бит. Хотя стойкость этой версии на сегодняшний день и не очень высока, она все равно часто используется как еще один вариант контрольной суммы, например, при скачивании файлов из сети.

3. SHA-1 — криптографическая функция формирующая хеш-суммы длиной в 160 байт. Сейчас идет активная миграция в сторону SHA-2, которая обладает более высокой устойчивостью, но SHA-1 по-прежнему активно используется хотя бы в качестве контрольных сумм. Но она так же по-прежнему используется и для хранения хешей паролей в базе данных сайта (об этом читайте выше).

4. ГОСТ Р 34.11-2012 — текущий российский криптографический (стойкий к взлому) алгоритм введенный в работу в 2013 году (ранее использовался ГОСТ Р 34.11-94). Длина выходного хеша может быть 256 или 512 бит. Обладает высокой криптостойкостью и довольно хорошей скоростью работы. Используется для электронных цифровых подписей в системе государственного и другого документооборота.

Требования к хэш-функциям

Существует ряд требований к хэш-функциям, предназначенным для практического задействования в той или иной области. Во-первых, соответствующий алгоритм должен характеризоваться чувствительностью к изменениям во внутренней структуре хешируемых документов. То есть в хэш-функции должны распознаваться, если речь идет о текстовом файле, перестановки абзацев, переносы. С одной стороны, содержимое документа не меняется, с другой — корректируется его структура, и этот процесс должен распознаваться в ходе хеширования. Во-вторых, рассматриваемый алгоритм должен преобразовывать данные так, чтобы обратная операция (превращение хэша в изначальный документ) была на практике невозможна. В-третьих, хэш-функция должна предполагать задействование таких алгоритмов, которые практически исключают вероятность формирования одинаковой последовательности символов в виде хэш, иными словами — появления так называемых коллизий. Их сущность мы рассмотрим чуть позже. Отмеченные требования, которым должен соответствовать алгоритм хэш-функции, могут быть обеспечены главным образом за счет задействования сложных математических подходов.

Структура

Изучим то, какой может быть структура рассматриваемых функций. Как мы отметили выше, в числе главных требований к рассматриваемым алгоритмам — обеспечение однонаправленности шифрования. Человек, имеющий в распоряжении только хэш, практически не должен иметь возможности получить на его основе исходный документ.

В какой структуре может быть представлена используемая в подобных целях хеш-функция? Пример ее составления может быть таким: H (hash, то есть, хэш) = f (T (текст), H1), где H1 — алгоритм обработки текста T. Данная функция хеширует T таким образом, что без знания H1 открыть его как полноценный файл будет практически невозможно.

Использование хэш-функций на практике: скачивание файлов

Изучим теперь подробнее варианты использования хэш-функций на практике. Задействование соответствующих алгоритмов может применяться при написании скриптов скачивания файлов с интернет-серверов. В большинстве случаев для каждого файла определяется некая контрольная сумма — это и есть хэш. Она должна быть одинаковой для объекта, располагающегося на сервере и скачанного на компьютер пользователя. Если это не так, то файл может не открыться либо запуститься не вполне корректно.

Хэш-функция и ЭЦП

Использование хэш-функций распространено при организации обмена документами, содержащими электронно-цифровую подпись. Хэшируется в данном случае подписываемый файл, для того чтобы его получатель мог удостовериться в том, что он подлинный. Хотя формально хэш-функция не входит в структуру электронного ключа, она может фиксироваться во флеш-памяти аппаратных средств, с помощью которых подписываются документы, таких как, например, eToken.

Электронная подпись представляет собой шифрование файла при задействовании открытого и закрытого ключей. То есть к исходному файлу прикрепляется зашифрованное с помощью закрытого ключа сообщение, а проверка ЭЦП осуществляется посредством открытого ключа. Если хэш-функция обоих документов совпадает — файл, находящийся у получателя, признается подлинным, а подпись отправителя распознается как верная.

Хеширование, как мы отметили выше, не является непосредственно компонентом ЭЦП, однако позволяет весьма эффективно оптимизировать алгоритмы задействования электронной подписи. Так, шифроваться может, собственно, только хэш, а не сам документ. В итоге скорость обработки файлов значительно возрастает, одновременно становится возможным обеспечивать более эффективные механизмы защиты ЭЦП, так как акцент в вычислительных операциях в этом случае будет ставиться не на обработке исходных данных, а на обеспечении криптографической стойкости подписи. Хэш-функция к тому же делает возможным подписывать самые разные типы данных, а не только текстовые.

Проверка паролей

Еще одна возможная область применения хеширования — организация алгоритмов проверки паролей, установленных для разграничения доступа к тем или иным файловым ресурсам. Каким образом при решении подобных задач могут быть задействованы те или иные виды хеш-функций? Очень просто.

Дело в том, что на большинстве серверов, доступ к которым подлежит разграничению, пароли хранятся в виде хэшированных значений. Это вполне логично — если бы пароли были представлены в исходном текстовом виде, хакеры, получившие доступ к ним, могли бы запросто читать секретные данные. В свою очередь, на основе хэш вычислить пароль непросто.

Каким образом осуществляется проверка доступа пользователя при задействовании рассматриваемых алгоритмов? Пароль, вводимый пользователем, сверяется с тем, что зафиксирован в хэш-функции, что хранится на сервере. Если значения текстовых блоков совпадают — пользователь получает необходимый доступ к ресурсам. В качестве инструмента проверки паролей может быть задействована самая простая хэш-функция. Но на практике IT-специалисты чаще всего используют комплексные многоступенчатые криптографические алгоритмы. Как правило, они дополняются применением стандартов передачи данных по защищенному каналу — так, чтобы хакеры не смогли обнаружить либо вычислить пароль, передаваемый с компьютера пользователя на сервера — до того, как он будет сверяться с хешированными текстовыми блоками.

Однонаправленные хэш-функции на основе симметричных блочных алгоритмов

Однонаправленную хэш-функцию можно построить, используя симметричный блочный алгоритм. Наиболее очевидный подход состоит в том, чтобы шифровать сообщение M посредством блочного алгоритма в режиме CBC или CFB с помощью фиксированного ключа и некоторого вектора инициализации IV. Последний блок шифртекста можно рассматривать в качестве хэш-значения сообщения M. При таком подходе не всегда возможно построить безопасную однонаправленную хэш-функцию, но всегда можно получить код аутентификации сообщения MAC (Message Authentication Code).

Примечание: Режимы работы алгоритма DES: CBC (Cipher Block Chaining)- сцепление блоков шифра; CFB (Cipher Feed Back) обратная связь по шифртексту)

Более безопасный вариант хэш-функции можно получить, используя:

- блок сообщения в качестве ключа,
- предыдущее хэш-значение - в качестве входа,
- а текущее хэш-значение - в качестве выхода.

Реальные хэш-функции проектируются еще более сложными:

1) длина блока обычно определяется длиной ключа

2) длина хэш-значения совпадает с длиной блока.

Поскольку большинство блочных алгоритмов являются 64-битовыми, некоторые схемы хэшированияпроектируют так, чтобы хэш-значение имело длину, равную двойной длине блока.

Если принять, что получаемая хэш-функция корректна, безопасность схемы хэширования базируется на безопасности лежащего в ее основе блочного алгоритма.

Схема хэширования, у которой длина хэш-значения равна длине блока, показана на рис.3.

Ее работа описывается выражениями:

H0 = Iн,

Hi = EA(B) ⊕ C,

где ⊕ - сложение по модулю 2 (исключающее ИЛИ); Iн - некоторое случайное начальное значение; A, B, C могут принимать значения Mi, Hi-1, (Mi ⊕ Hi-1) или быть константами.

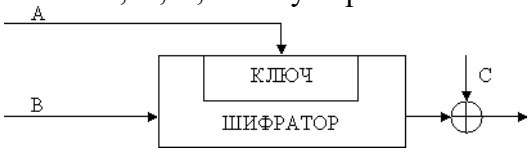


Рис.3. Обобщенная схема формирования хэш-функции

Сообщение M разбивается на блоки Mi принятой длины, которые обрабатываются поочередно.

Три различные переменные A, B, C могут принимать одно из четырех возможных значений, поэтому в принципе можно получить 64 варианта общей схемы этого типа. Из них 52 варианта являются либо тривиально слабыми, либо небезопасными. Остальные 12 схем безопасного хэширования, у которых длина хэш-значения равна длине блока перечислены в табл.1.

Таблица 1

Номер схемы	Функция хэширования
1	$H_i = E_{Hi-1}(M_i) \oplus M_i$
2	$H_i = E_{Hi-1}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
3	$H_i = E_{Hi-1}(M_i) \oplus M_i \oplus H_{i-1}$
4	$H_i = E_{Hi-1}(M_i \oplus H_{i-1}) \oplus M_i$
5	$H_i = E_{Mi}(H_{i-1}) \oplus H_{i-1}$
6	$H_i = E_{Mi}(M_i \oplus H_{i-1}) \oplus M_i \oplus H_{i-1}$
7	$H_i = E_{Mi}(H_{i-1}) \oplus M_i \oplus H_{i-1}$
8	$H_i = E_{Mi}(M_i \oplus H_{i-1}) \oplus H_{i-1}$
9	$H_i = E_{Mi \oplus Hi-1}(M_i) \oplus M_i$
10	$H_i = E_{Mi \oplus Hi-1}(H_{i-1}) \oplus H_{i-1}$
11	$H_i = E_{Mi \oplus Hi-1}(M_i) \oplus H_{i-1}$
12	$H_i = E_{Mi \oplus Hi-1}(H_{i-1}) \oplus M_i$

Первые четыре схемы хэширования, являющиеся безопасными при всех атаках, приведены на рис.4.

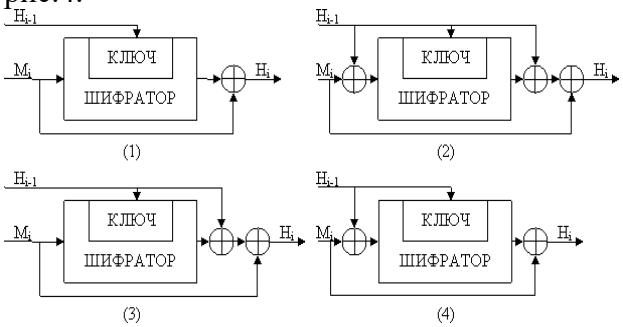


Рис.4. Четыре схемы безопасного хэширования

Недостатком хэш-функций, спроектированных на основе блочных алгоритмов, является несколько заниженная скорость работы.

Дело в том, что ту же самую стойкость относительно двух основных требований к хэш-функции можно обеспечить за гораздо меньшее количество операций над входными данными. Но для этого алгоритм необходимо изначально проектировать специально, исходя из тандема требований (стойкость, скорость).

16. Обобщенная модель электронной цифровой подписи

Схему, изложенную в разделе 5 для асимметричных систем с открытым ключом, можно также использовать для цифровой подписи сообщений, которую невозможно подделать за полиномиальное время.

Пусть пользователю А необходимо подписать сообщение x. Он, зная секрет K, находит такое y, что F_K (y) = x, и вместе с сообщением x посылает y пользователю B в качестве своей цифровой подписи. Пользователь B хранит y в качестве доказательства того, что A подписал сообщение x.

Сообщение, подписанное цифровой подписью, можно представлять себе как пару (x, y), где x —сообщение, y — решение уравнения F_K (y) = x, F_K: X → Y — функция с секретом, известная всем взаимодействующим абонентам. Из определения функции F_K очевидны следующие достоинства цифровой подписи:

a) подписать сообщение x, т.е. решить уравнение F_K (y) = =x, может только абонент — обладатель данного секрета K; другими словами, подделать подпись невозможно;

b) проверить подлинность подписи может любой абонент, знающий открытый ключ, т.е. саму функцию F_K ;

c) при возникновении споров отказаться от подписи невозможно в силу ее неподделываемости;

d) подписанные сообщения (x, y) можно, не опасаясь ущерба, пересылать по любым каналам связи.

Важным преимуществом асимметричных методов является возможность идентификации отправителя путем использования его электронной подписи. Идея технологии электронной подписи состоит в следующем. Отправитель передает два экземпляра одного сообщения: открытое и расшифрованное его закрытым ключом (т.е. обратно шифрованное). Получатель шифрует с помощью открытого ключа отправителя расшифрованный экземпляр. Если он совпадет с открытым вариантом, то личность и подпись отправителя считается установленной.

Формально выражаясь, асимметричный метод обеспечивает реализацию электронной подписи при выполнении следующего тождества:

E(D(T)) = D(E(T)) = T.

При практической реализации электронной подписи также шифруется не все сообщение, а лишь специальная контрольная сумма — хэш (hash total), защищающая послание от нелегального изменения. Важно, что электронная подпись здесь как гарантирует целостность сообщения, так и удостоверяет личность отправителя.

Вопросы реализации электронной подписи и вычисления ее хэша определены в отечественных стандартах “Информационная технология. Криптографическая защита информации”, а именно: ГОСТ 34.10-94 “Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма” и ГОСТ 34.11-94 “Функция хэширования”.

Контроль целостности

Криптографические методы позволяют надежно контролировать целостность как отдельных порций данных, так и их наборов (таких, как поток сообщений); определять подлинность источника данных; гарантировать невозможность отказаться от совершенных действий ("неотказуемость").

В основе криптографического контроля целостности лежат два понятия:

• хэш-функция;

• электронная цифровая подпись (ЭЦП).

Хэш-функция — это труднообратимое преобразование данных (односторонняя функция), реализуемое, как правило, средствами симметричного шифрования со связыванием блоков. Результат шифрования последнего блока (зависящий от всех предыдущих) и служит результатом хэш-функции.

Пусть имеются данные, целостность которых нужно проверить, хэш-функция и ранее вычисленный результат ее применения к исходным данным (так называемый дайджест). Обозначим хэш-функцию через h, исходные данные — через T, проверяемые данные — через T. Контроль целостности данных сводится к проверке равенства h(T') = h(T). Если оно выполнено, считается, что T' = T. Совпадение дайджестов для различных данных называется коллизией. В принципе, коллизии, конечно, возможны, поскольку мощность множества дайджестов меньше, чем мощность множества

хэшируемых данных, однако то, что h есть функция односторонняя, означает, что за приемлемое время специально организовать коллизию невозможно.

Рассмотрим теперь применение асимметричного шифрования для выработки и проверки электронной цифровой подписи. Пусть E(T) обозначает результат зашифрования текста T с помощью открытого ключа, а D(T) — результат расшифрования текста T (как правило, шифрованного) с помощью секретного ключа. Чтобы асимметричный метод мог применяться для реализации ЭЦП, необходимо выполнение тождества «(/ ; »

$$E(D(T)) = D(E(T)) = T \setminus „ T .$$

На рис. 11.1 показана процедура выработки электронной цифровой подписи, состоящая в шифровании преобразованием D дайджеста h(T).

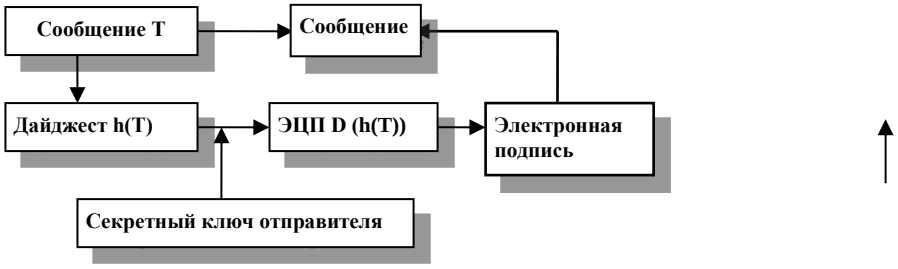


Рис.11.1. Схема процедуры выработки электронной цифровой подписи

Проверка ЭЦП может быть реализована так, как показано на рис.11.2.

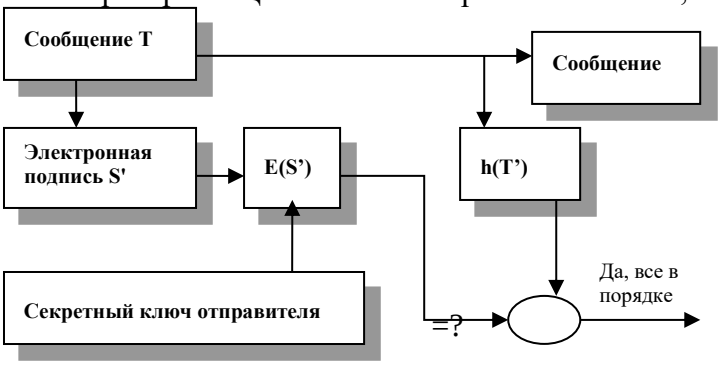


Рис. 11.2. Схема процедуры проверки электронной цифровой подписи

Из равенства E(S') = h(T') I следует, что S' = D(h(T) (для доказательства достаточно применить к обеим частям преобразование D и вычеркнуть в левой части тождественное преобразование D(E())). Таким образом, электронная цифровая подпись защищает целостность сообщения и удостоверяет личность отправителя, то есть защищает целостность источника данных и служит основой неотказуемости.

Два российских стандарта: ГОСТ Р 34.10-94 "Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма" и ГОСТ Р 34.11-94 "Функция хэширования", объединенные общим заголовком "Информационная технология. Криптографическая защита информации", регламентируют вычисление дайджеста и реализацию ЭЦП. В сентябре 2001 года был утвержден, а 1 июля 2002 года вступил в силу новый стандарт ЭЦП — ГОСТ Р 34.10-2001, разработанный специалистами ФАПСИ.

Для контроля целостности последовательности сообщений (то есть для защиты от кражи, дублирования и переупорядочения сообщений) применяют временные штампы и нумерацию элементов последовательности, при этом штампы и номера включают в подписываемый текст.

Цифровые сертификаты

При использовании асимметричных методов шифрования (и, в частности, электронной цифровой подписи) необходимо иметь гарантию подлинности пары (имя пользователя, открытый ключ пользователя). Для решения этой задачи в спецификациях X.509 вводятся понятия цифрового сертификата и удостоверяющего центра.

Удостоверяющий центр — это компонент глобальной службы каталогов, отвечающий за управление криптографическими ключами пользователей. Открытые ключи и другая информация о пользователях хранятся удостоверяющими центрами в виде цифровых сертификатов, имеющих следующую структуру:

- порядковый номер сертификата;
- идентификатор алгоритма электронной подписи;
- имя удостоверяющего центра;
- срок годности;
- имя владельца сертификата (имя пользователя, которому принадлежит сертификат);
- открытые ключи владельца сертификата (ключей может быть несколько);
- идентификаторы алгоритмов, ассоциированных с открытыми ключами владельца сертификата;
- электронную подпись, сгенерированную с использованием секретного ключа удостоверяющего центра (подписывается результат хэширования всей информации, хранящейся в сертификате).

Цифровые сертификаты обладают следующими свойствами:

- любой пользователь, знающий открытый ключ удостоверяющего центра, может узнать открытые ключи других клиентов центра и проверить целостность сертификата;
- никто, кроме удостоверяющего центра, не может модифицировать информацию о пользователе без нарушения целостности сертификата.

В спецификациях X.509 не описывается конкретная процедура генерации криптографических ключей и управления ими, однако даются некоторые общие рекомендации. В частности, оговаривается, что пары ключей могут порождаться любым из следующих способов:

- ключи может генерировать сам пользователь; в таком случае секретный ключ не попадает в руки третьих лиц, однако нужно решать задачу безопасной связи с удостоверяющим центром;
- ключи генерирует доверенное лицо; здесь приходится решать задачи безопасной доставки секретного ключа владельцу и предоставления доверенных данных для создания сертификата;
- ключи генерируются удостоверяющим центром, в данном случае остается только задача безопасной передачи ключей владельцу.

Цифровые сертификаты в формате X.509 версии 3 стали не только формальным, но и фактическим стандартом, поддерживаемым многочисленными удостоверяющими центрами.

Следует отметить, что криптографические методы используются также для контроля целостности информации и программ. Для этого применяется шифрованная контрольная сумма исходного текста (имитоприставка), вычисленная с применением секретного ключа. В отличие от традиционной контрольной суммы (используемой для защиты от программно-аппаратных сбоев и ошибок) имитоприставка обеспечивает практически абсолютную защиту как от непреднамеренной, так и преднамеренной модификации данных или программы.

Основные типы криптоаналитических атак

Нет невзламываемых шифров. Все системы шифрования просто делают взламывание шифровок или заведомо дороже содержащейся в сообщении информации, или затягивают время расшифрования на неприемлемо большой срок.

В.Жельников

Фундаментальное правило криптоанализа, впервые сформулированное голландцем А.Керкхоффом еще в XIX в., заключается в том, что стойкость шифра (криптосистемы) должна определяться только секретностью ключа. Иными словами, правило Керкхоффа состоит в том, что весь алгоритм шифрования, кроме значения секретного ключа, известен криптоаналитику противника. Это обусловлено тем, что криптосистема, реализующая семейство криптографических преобразований, обычно рассматривается как открытая система. Такой подход отражает очень важный принцип технологии защиты информации: защищенность системы не должна зависеть от секретности чего-либо такого, что невозможно быстро изменить в случае утечки секретной информации.

Обычно криптосистема представляет собой совокупность аппаратных и программных средств, которую можно изменить только при значительных затратах времени и средств, тогда как ключ является легко изменяемым объектом. Именно поэтому стойкость криптосистемы определяется только секретностью ключа.

Другое почти общепринятое допущение состоит в том, что криптоаналитик имеет в своем распоряжении шифр тексты сообщений.

Существует четыре основных типа криптоаналитических атак [2.4]. Конечно, все они формулируются в предположении, что криптоаналитику известны применяемый алгоритм шифрования и шифртексты сообщений.

1. Криптоаналитическая атака при наличии только известного шифртекста. Криптоаналитик имеет только шифр тексты C_1, C_2, …, C_i нескольких сообщений, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_k. Работа криптоаналитика заключается в том, чтобы раскрыть исходные тексты M_1, M_2, …, M_i по возможности большинства сообщений или, еще лучше, вычислить ключ K, использованный для шифрования этих сообщений, с тем чтобы расшифровать и другие сообщения, зашифрованные этим шифром.

Этот вариант соответствует модели внешнего нарушителя, который имеет физический доступ к линии связи, но не имеет доступа к аппаратуре шифрования и дешифрования.

2. Криптоаналитическая атака при наличии известного открытого текста. Криптоаналитик имеет доступ не только к шифртекстам C_1, C_2, …, C_i и нескольких сообщений, но также к открытым текстам M_1, M_2, …, M_i этих сообщений. Его работа заключается в нахождении ключа K, используемого при шифровании этих сообщений, или алгоритма расшифрования D_k любых новых сообщений, зашифрованных тем же ключом, причем все они зашифрованы с использованием одного и того же алгоритма шифрования E_k.

Возможность проведения такой атаки складывается при шифровании стандартных документов, подготавливаемых по стандартным формам, когда определенные болки данных повторяются и известны. Он также применим при использовании режима глобального шифрования, когда вся информация на встроенном магнитном носителе записывается в виде шифртекста, включая главную корневую запись, загрузочный сектор, системные программы и пр. При хищении этого носителя (или компьютера) легко установить, какая часть криптограммы соответствует системной информации, и получить большой объем известного исходного текста для выполнения криптоанализа.

3. Криптоаналитическая атака при возможности выбора открытого текста. Криптоаналитик не только имеет доступ к шифртекстам C_1, C_2, …, C_i и связанным с ними открытым текстам M_1, M_2, …, M_i этих сообщений, но и может по желанию выбирать открытые тексты, которые затем получает в зашифрованном виде. Такой криптоанализ получается более мощным по сравнению с криптоанализом с известным открытым текстом, потому что криптоаналитик может выбрать для шифрования такие блоки открытого текста, которые дадут больше информации о ключе. Работа криптоаналитика состоит в поиске ключа K, использованного для шифрования сообщений, или алгоритма рашифрования D_k новых сообщений, зашифрованных тем же ключом.

Этот вариант атаки соответствует модели внутреннего нарушителя. На практике такая ситуация может возникнуть при вовлечении в криптоатаку лиц, которые не знают секретного ключа, но в силу своих служебных полномочий имеют возможность использовать шифрование для передачи своих сообщений.

4. Криптоаналитическая атака с адаптивным выбором открытого текста. Это особый вариант атаки с выбором открытого текста. Криптоаналитик может не только выбирать открытый текст, который затем шифруется, но и изменять свой выбор в зависимости от результатов предыдущего шифрования. При криптоанализе с простым выбором открытого текста криптоаналитик обычно может выбирать несколько крупных блоков открытого текста для их шифрования; при криптоанализе с адаптивным выбором открытого текста он имеет возможность выбрать сначала более мелкий пробный блок открытого текста, затем выбрать следующий блок в зависимости от результатов первого выбора и т.д. Эта атака предоставляет криптоаналитику еще больше возможностей, чем предыдущие типы атак.

5. Криптоаналитическая атака с использованием выбранного шифртекста. Криптоаналитик может выбирать для расшифрования различные шифртексты и имеет доступ к расшифрованным открытым

текстам . Например, криптоаналитик получил доступ к защищенному от несанкционированного вскрытия блоку, который выполняет автоматическое расшифрование. Работа криптоаналитика заключается в нахождении ключа. Этот тип криптоанализа представляет особый интерес для раскрытия алгоритмов с открытым ключом.

6. Криптоаналитическая атака методом полного перебора всех возможных ключей. Эта атака предполагает использование криптоаналитиком известного шифр текста и осуществляется посредством полного перебора всех возможных ключей с проверкой, является ли осмысленным получающийся открытый текст. Такой подход требует привлечения предельных вычислительных ресурсов и иногда называется силовой атакой.

Существуют и другие, менее распространенные виды криптоаналитических атак.

Итак, разумеется, отразить в нескольких лекциях все вопросы и проблемы современной криптологии задача невыполнимая. Объем знаний в этой области чрезвычайно велик и продолжает интенсивно увеличиваться. Кроме того, для полноценного освоения всех вопросов криптологии требуется весьма солидная университетская математическая подготовка.

Важно подчеркнуть, что шифрование информации, с одной стороны, требует определенных затрат на его выполнение, а с другой — не гарантирует 100—процентной надежности защиты от злоумышленника. Поэтому всегда надо четко оценивать необходимость применения это способа защиты информации в конкретных ситуациях.

17.Угрозы безопасности ПО. Программные закладки. Троянские программы. Клавишные шпионы

Обеспечение безопасности автоматизированных информационных систем зависит от безопасности используемого в них программного обеспечения и, в частности, следующих видов программ:

- обычных программ пользователей;
- специальных программ, рассчитанных на нарушение безопасности системы;
- разнообразных системных утилит и коммерческих прикладных программ, которые отличаются высоким профессиональным уровнем разработки и тем не менее могут содержать отдельные недоработки, позволяющие захватчикам атаковать системы.

Программы могут порождать проблемы двух типов: во-первых, могут перехватывать и модифицировать данные в результате действий пользователя, который к этим данным не имеет доступа, и, во-вторых, используя упущения в защите компьютерных систем, могут или обеспечивать доступ к системе пользователям, не имеющим на это права, или блокировать доступ к системе законных пользователей.

Чем выше уровень подготовки программиста, тем более неявными (даже для него) становятся допускаемые им ошибки и тем более тщательно и надежно он способен скрыть умышленные механизмы, разработанные для нарушения безопасности системы.

Целью атаки могут быть и сами программы по следующим причинам:

- В современном мире программы могут быть товаром, приносящим немалую прибыль, особенно тому, кто первым начнет тиражировать программу в коммерческих целях и оформит авторские права на нее.
- Программы могут становиться также объектом атаки, имеющей целью модифицировать эти программы некоторым образом, что позволило бы в будущем провести атаку на другие объекты системы. Особенно часто объектом атак такого рода становятся программы, реализующие функции защиты системы.

Рассмотрим несколько типов программ и приемы, которые наиболее часто используются для атак программ и данных. Эти приемы обозначаются единым термином — «программные ловушки». К ним относятся «программные люки», «троянские кони», «логические бомбы», атаки «салями», скрытые каналы, отказы в обслуживании и компьютерные вирусы.

Люки в программах.Использование люков для проникновения в программу — один из простых и часто используемых способов нарушения безопасности автоматизированных информационных систем. Люкомназывается не описанная в документации на программный продукт возможность работы с этим программным продуктом. Сущность использования люков состоит в том, что при выполнении пользователем некоторых не описанных в документации действий он получает доступ к возможностям и данным, которые в обычных условиях для него закрыты (в частности, выход в привилегированный режим).

Люки чаще всего являются результатом забывчивости разработчиков. В качестве люка может быть использован временный механизм прямого доступа к частям продукта, созданный для облегчения процесса отладки и не удаленный по ее окончании. Люки могут образовываться также в результате часто практикуемой технологии разработки программных продуктов «сверху вниз»: в их роли будут выступать оставленные по каким-либо причинам в готовом продукте «заглушки» — группы команд, имитирующие или просто обозначающие место подсоединения будущих подпрограмм. Наконец, еще одним распространенным источником люков является так называемый «неопределенный ввод» — ввод «бессмысленной» информации, абракадабры в ответ на запросы системы. Реакция недостаточно хорошо написанной программы на неопределенный ввод может быть, в лучшем случае, непредсказуемой (когда при повторном вводе той же неверной команды программа реагирует каждый раз по-разному); гораздо хуже, если программа в результате одинакового «неопределенного» ввода выполняет некоторые повторяющиеся действия, — это дает возможность потенциальному захватчику планировать свои действия по нарушению безопасности.

Неопределенный ввод — частная реализация прерывания. То есть в общем случае захватчик может умышленно пойти на создание в системе некоторой нестандартной ситуации, которая бы позволила ему выполнить необходимые действия. Например, он может искусственно вызвать

аварийное завершение программы, работающей в привилегированном режиме, с тем, чтобы перехватить управление, оставшись в этом привилегированном режиме.

Борьба с возможностью прерывания, в конечном счете, выливается в необходимость предусмотреть при разработке программ комплекса механизмов, образующих так называемую «защиту от дурака». Смысл этой защиты состоит в том, чтобы гарантированно отсекать всякую вероятность обработки неопределенного ввода и разного рода нестандартных ситуаций (в частности, ошибок) и тем самым не допускать нарушения безопасности компьютерной системы даже в случае некорректной работы с программой.

Таким образом, люк (или люки) может присутствовать в программе ввиду того, что программист:

- забыл удалить его;
- умышленно оставил его в программе для обеспечения тестирования или выполнения оставшейся части отладки;
- умышленно оставил его в программе в интересах облегчения окончательной сборки конечного программного продукта;
- умышленно оставил его в программе с тем, чтобы иметь скрытое средство доступа к программе уже после того, как она вошла в состав конечного продукта.

Люк — первый шаг к атаке системы, возможность проникнуть в компьютерную систему в обход механизмов защиты.

«Троянские кони». Существуют программы, реализующие, помимо функций, описанных в документации, и некоторые другие функции, в документации не описанные. Такие программы называются «троянскими конями». Вероятность обнаружения «троянского коня» тем выше, чем очевиднее результаты его действий (например, удаление файлов или изменение их защиты). Более сложные «троянские кони» могут маскировать следы своей деятельности (например, возвращать защиту файлов в исходное состояние).

«Логические бомбы». «Логической бомбой» обычно называют программу или даже участок кода в программе, реализующий некоторую функцию при выполнении определенного условия или в определенное время.. Этим условием может быть, например, наступление определенной даты или обнаружение файла с определенным именем.«Взрываясь», «логическая бомба» реализует функцию, неожиданную и, как правило, нежелательную для пользователя (например, удаляет некоторые данные или разрушает некоторые системные структуры). «Логическая бомба» является одним из излюбленных способов мести программистов компаниям, которые их уволили или чем-либо обидели.

Атака «салями». Атака «салями» превратилась в настоящий бич банковских компьютерных систем. В банковских системах ежедневно производятся тысячи операций, связанных с безналичными расчетами, переводами сумм, отчислениями и т. д.

При обработке счетов используются целые единицы (рубли, центы), а при исчислении процентов нередко получаются дробные суммы. Обычно величины, превышающие половину рубля (цента), округляются до целого рубля (цента), а величины менее половины рубля (цента) просто отбрасываются. При атаке «салями» эти несущественные величины не удаляются, а постепенно накапливаются на некоем специальном счете.

Как свидетельствует практика, сумма, составленная буквально из ничего, за пару лет эксплуатации «хитрой» программы в среднем по размеру банке может исчисляться тысячами долларов. Атаки «салями» достаточно трудно распознаются, если злоумышленник не начинает накапливать на одном счете большие суммы.

Скрытые каналы. Под скрытыми каналами подразумеваются программы, передающие информацию лицам, которые в обычных условиях эту информацию получать не должны. В тех системах, где ведется обработка критичной информации, программист не должен иметь доступа к обрабатываемым программой данным после начала эксплуатации этой программы.

Если захватчик имеет возможность доступа к компьютеру во время работы интересующей его программы, скрытым каналом может стать пересылка критичной информации в специально созданный в оперативной памяти компьютера массив данных.

Скрытые каналы наиболее применимы в ситуациях, когда захватчика интересует даже не содержание информации, а, допустим, факт ее наличия (например, наличие в банке расчетного счета с определенным номером).

Отказ в обслуживании. Большинство методов нарушения безопасности направлено на то, чтобы получить доступ к данным, не допускаемый системой в нормальных условиях. Однако не менее интересным для захватчиков является доступ к управлению самой компьютерной системой или изменение ее качественных характеристик, например, получить некоторый ресурс (процессор, устройство ввода-вывода) в монопольное использование или спровоцировать ситуацию клинча для нескольких процессов.]

Различают пассивные и активные угрозы.

Пассивные угрозы (нарушение конфиденциальности данных, циркулирующих в сети) — это просмотр и/или запись данных, передаваемых по линиям связи. К ним относятся:

- просмотр сообщения;
- анализ графика — злоумышленник может просматривать заголовки пакетов, циркулирующих в сети, и на основе содержащейся в них служебной информации делать заключения об отправителях и получателях пакета и условиях передачи (время отправления, класс сообщения, категория безопасности, длина сообщения, объем трафика и т. д.).

Активные угрозы (нарушение целостности или доступности ресурсов и компонентов сети) — несанкционированное использование устройств, имеющих доступ к сети для изменения отдельных сообщений или потока сообщений. К ним относятся:

- отказ служб передачи сообщений — злоумышленник может уничтожать или задерживать отдельные сообщения или весь поток сообщений;
- «маскарад» — злоумышленник может присвоить своему узлу или ретранслятору чужой идентификатор и получать или отправлять сообщения от чужого имени;
- внедрение сетевых вирусов — передача по сети тела вируса с его последующей активизацией пользователем удаленного или локального узла;
- модификация потока сообщений — злоумышленник может выборочно уничтожать, модифицировать, задерживать, переупорядочивать и дублировать сообщения, а также вставлять поддельные сообщения.

МОДЕЛЬ УГРОЗ И ПРИНЦИПЫ ОБЕСПЕЧЕНИЯ БЕЗОПАСНОСТИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Использование при создании программного обеспечения КС сложных операционных систем, инструментальных средств разработки ПО импортного производства увеличивают потенциальную возможность внедрения в программы преднамеренных дефектов диверсионного типа. Помимо этого, при создании целевого программного обеспечения всегда необходимо исходить из возможности наличия в коллективе разработчиков программистов - злоумышленников, которые в силу тех или иных причин могут внести в разрабатываемые программы РПС.

Характерным свойством РПС в данном случае является возможность внезапного и незаметного нарушения или полного вывода из строя КС. Функционирование РПС реализуется в рамках модели угроз безопасности ПО, основные элементы которой рассматриваются в следующем разделе.

Подход к созданию модели угроз технологической безопасности ПО

Один из возможных подходов к созданию модели технологической безопасности ПО АСУ может основываться на обобщенной концепции технологической безопасности компьютерной инфосферы, которая определяет методологический базис, направленный на решение, в том числе, следующих основных задач:

- создания теоретических основ для практического решения проблемы технологической безопасности ПО;
- создания безопасных информационных технологий;
- развертывания системы контроля технологической безопасности компьютерной инфосферы.

Модель угроз технологической безопасности ПО должна представлять собой официально принятый нормативный документ, которым должен руководствоваться заказчики и разработчики программных комплексов.

Модель угроз должна включать:

- полный реестр типов возможных программных закладок;
- описание наиболее технологически уязвимых мест компьютерных систем (с точки зрения важности и наличия условий для скрытого внедрения программных закладок);

- описание мест и технологические карты разработки программных средств, а также критических этапов, при которых наиболее вероятно скрытое внедрение программных закладок;
- реконструкцию замысла структур, имеющих своей целью внедрение в ПО заданного типа (класса, вида) программных закладок диверсионного типа;
- психологический портрет потенциального диверсанта в компьютерных системах

В указанной Концепции также оговариваются необходимость содержания в качестве приложения банка данных о выявленных программных закладках и описания связанных с их обнаружением обстоятельств, а также необходимость периодического уточнения и совершенствования модели на основе анализа статистических данных и результатов теоретических исследований.

На базе утвержденной модели угроз технологической безопасности компьютерной инфосферы, как обобщенного, типового документа должна разрабатываться прикладная модель угроз безопасности для каждого конкретного компонента защищаемого комплекса средств автоматизации КС. В основе этой разработки должна лежать схема угроз, типовой вид которой применительно к ПО КС представлен на рис.1.2.

Наполнение модели технологической безопасности ПО должно включать в себя следующие элементы: матрицу чувствительности КС к "вариациям" ПО (то есть к появлению искажений), энтропийный портрет ПО (то есть описание "темных" запутанных участков ПО), реестр камуфлирующих условий для конкретного ПО, справочные данные о разработчиках и реальный (либо реконструированный) замысел злоумышленников по поражению этого ПО. На рис.1.3 приведен пример указанной типовой модели для сложных программных комплексов.

1.5.2. Элементы модели угроз эксплуатационной безопасности ПО

Анализ угроз эксплуатационной безопасности ПО КС позволяет, разделить их на два типа: случайные и преднамеренные, причем последние подразделяются на активные и пассивные. Активные угрозы направлены на изменение технологически обусловленных алгоритмов, программ функциональных преобразований или информации, над которой эти преобразования осуществляются. Пассивные угрозы ориентированы на нарушение безопасности информационных технологий без реализации таких модификаций.

Вариант общей структуры набора потенциальных угроз безопасности информации и ПО на этапе эксплуатации КС приведен в табл.1.2.

Таблица 1.2

Рассмотрим основное содержание данной таблицы. Угрозы, носящие случайный характер и связанные с отказами, сбоями аппаратуры, ошибками операторов и т.п. предполагают нарушение заданного собственником информации алгоритма, программы ее обработки или искажение содержания этой информации. Субъективный фактор появления таких угроз обусловлен ограниченной надежностью работы человека и проявляется в виде ошибок (дефектов) в выполнении операций формализации алгоритма функциональных преобразований или описания алгоритма на некотором языке, понятном вычислительной системе.

Угрозы, носящие злоумышленный характер вызваны, как правило, преднамеренным желанием субъекта осуществить несанкционированные изменения с целью нарушения корректного выполнения преобразований, достоверности и целостности данных, которое проявляется в искажениях их содержания или структуры, а также с целью нарушения функционирования технических средств в процессе реализации функциональных преобразований и изменения конструктива вычислительных систем и систем телекоммуникаций.

На основании анализа уязвимых мест и после составления полного перечня угроз для данного конкретного объекта информационной защиты, например, в виде указанной таблицы, необходимо осуществить переход к неформализованному или формализованному описанию модели угроз эксплуатационной безопасности ПО КС. Такая модель, в свою очередь, должна соотноситься (или даже являться составной частью) обобщенной модели обеспечения безопасности информации и ПО объекта защиты.

К неформализованному описанию модели угроз приходится прибегать в том случае, когда структура, состав и функциональная наполненность компьютерных системы управления носят многоуровневый, сложный, распределенный характер, а действия потенциального нарушителя информационных и функциональных ресурсов трудно поддаются формализации. После

окончательного синтеза модели угроз разрабатываются практические рекомендации и методики по ее использованию для конкретного объекта информационной защиты, а также механизмы оценки адекватности модели и реальной информационной ситуации и оценки эффективности ее применения при эксплуатации КС.

Таким образом, разработка моделей угроз безопасности программного обеспечения КС, являясь одним из важных этапов комплексного решения проблемы обеспечения безопасности информационных технологий, на этапе создания КС отличается от разработки таких моделей для этапа их эксплуатации.

Принципиальное различие подходов к синтезу моделей угроз технологической и эксплуатационной безопасности ПО заключается в различных мотивах поведения потенциального нарушителя информационных ресурсов, принципах, методах и средствах воздействия на ПО на различных этапах его жизненного цикла.

Основные принципы обеспечения безопасности ПО

В качестве объекта обеспечения технологической и эксплуатационной безопасности ПО рассматривается вся совокупность его компонентов в рамках конкретной КС. В качестве доминирующей должна использоваться стратегия сквозного тотального контроля технологического и эксплуатационного этапов жизненного цикла компонентов ПО. Совокупность мероприятий по обеспечению технологической и эксплуатационной безопасности компонентов ПО должна носить конфиденциальный характер. Необходимо обеспечить постоянный, комплексный и действенный контроль за деятельностью разработчиков и пользователей компонентов. Кроме общих принципов, обычно необходимо конкретизировать принципы обеспечения безопасности ПО на каждом этапе его жизненного цикла. Далее приводятся один из вариантов разработки таких принципов.

Принципы обеспечения технологической безопасности при обосновании, планировании работ и проектном анализе ПО

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Комплексности обеспечения безопасности ПО, предполагающей рассмотрение проблемы безопасности информационно - вычислительных процессов с учетом всех структур КС, возможных каналов утечки информации и несанкционированного доступа к ней, времени и условий их возникновения, комплексного применения организационных и технических мероприятий.

Планируемости применения средств безопасности программ, предполагающей перенос акцента на совместное системное проектирование ПО и средств его безопасности, планирование их использования в предполагаемых условиях эксплуатации.

Обоснованности средств обеспечения безопасности ПО, заключающейся в глубоком научно-обоснованном подходе к принятию проектных решений по оценке степени безопасности, прогнозированию угроз безопасности, всесторонней априорной оценке показателей средств защиты.

Достаточности безопасности программ, отражающей необходимость поиска наиболее эффективных и надежных мер безопасности при одновременной минимизации их стоимости.

Гибкости управления защитой программ, требующей от системы контроля и управления обеспечением информационной безопасности ПО способности к диагностированию, опережающей нейтрализации, оперативному и эффективному устранению возникающих угроз в условиях резких изменений обстановки информационной борьбы.

Заблаговременности разработки средств обеспечения безопасности и контроля производства ПО, заключающейся в предупредительном характере мер обеспечения технологической безопасности работ в интересах недопущения снижения эффективности системы безопасности процесса создания ПО.

Документируемости технологии создания программ, подразумевающей разработку пакета нормативно-технических документов по контролю программных средств на наличие преднамеренных дефектов.

Принципы достижения технологической безопасности ПО в процессе его разработки

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Регламентации технологических этапов разработки ПО, включающей упорядоченные фазы промежуточного контроля, спецификацию программных модулей и стандартизацию функций и формата представления данных.

Автоматизации средств контроля управляющих и вычислительных программ на наличие дефектов, создания типовой общей информационной базы алгоритмов, исходных текстов и программных средств, позволяющих выявлять преднамеренные программные дефекты.

Последовательной многоуровневой фильтрации программных модулей в процессе их создания с применением функционального дублирования разработок и поэтапного контроля.

Типизации алгоритмов, программ и средств информационной безопасности, обеспечивающей информационную, технологическую и программную совместимость, на основе максимальной их унификации по всем компонентам и интерфейсам.

Централизованного управления базами данных проектов ПО и администрирование технологии их разработки с жестким разграничением функций, ограничением доступа в соответствии со средствами диагностики, контроля и защиты.

Блокирования несанкционированного доступа соисполнителей и абонентов государственных сетей связи, подключенных к стендам для разработки программ.

Статистического учета и ведения системных журналов о всех процессах разработки ПО с целью контроля технологической безопасности.

Использования только сертифицированных и выбранных в качестве единых инструментальных средств разработки программ для новых технологий обработки информации и перспективных архитектур вычислительных систем.

Принципы обеспечения технологической безопасности на этапах стендовых и приемо-сдаточных испытаний

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Тестирования ПО на основе разработки комплексов тестов, параметризуемых на конкретные классы программ с возможностью функционального и статистического контроля в широком диапазоне изменения входных и выходных данных.

Проведения натурных испытаний программ при экстремальных нагрузках с имитацией воздействия активных дефектов.

Осуществления "фильтрации" программных комплексов с целью выявления возможных преднамеренных дефектов определенного назначения на базе создания моделей угроз и соответствующих сканирующих программных средств.

Разработки и экспериментальной отработки средств верификации программных изделий.

Проведения стендовых испытаний ПО для определения непреднамеренных программных ошибок проектирования и ошибок разработчика, приводящих к невыполнению целевых функций программ, а также выявление потенциально "узких" мест в программных средствах для разрушительного воздействия.

Отработки средств защиты от несанкционированного воздействия нарушителей на ПО.

Сертификации программных изделий АСУ по требованиям безопасности с выпуском сертификата соответствия этого изделия требованиям технического задания.

Принципы обеспечения безопасности при эксплуатации программного обеспечения

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Сохранения и ограничения доступа к эталонам программных средств, недопущение внесения изменений в них.

Профилактического выборочного тестирования и полного сканирования программных средств на наличие преднамеренных дефектов.

Идентификации ПО на момент ввода его в эксплуатацию в соответствии с предполагаемыми угрозами безопасности ПО и его контроль.

Обеспечения модификации программных изделий во время их эксплуатации путем замены отдельных модулей без изменения общей структуры и связей с другими модулями.

Строгого учета и каталогизации всех сопровождаемых программных средств, а также собираемой, обрабатываемой и хранимой информации.

Статистического анализа информации обо всех процессах, рабочих операциях, отступлениях от режимов штатного функционирования ПО.

Гибкого применения дополнительных средств защиты ПО в случае выявления новых, непрогнозируемых угроз информационной безопасности.

18.Классификация компьютерных вирусов. Диагностика заражения компьютерным вирусом. Основы функционирования антивирусного ПО

Свойства компьютерных вирусов

Вирус - это программа.

Такое простое утверждение само по себе способно развеять множество легенд о необыкновенных возможностях компьютерных вирусов. Вирус может перевернуть изображение на вашем мониторе, но не может перевернуть сам монитор. К легендам о вирусах-убийцах, «уничтожающих операторов посредством вывода на экран смертельной цветовой гаммы 25-м кадром» также не стоит относиться серьезно.

Вирус - программа, обладающая способностью к самовоспроизведению. Такая способность является единственным средством, присущим всем типам вирусов. Но не только вирусы способны к самовоспроизведению. Любая операционная система и еще множество программ способны создавать собственные копии. Копии же вируса не только не обязаны полностью совпадать с оригиналом, но и могут вообще с ним не совпадать!

Вирус не может существовать в «полной изоляции»: сегодня нельзя представить себе вирус, который не использует код других программ, информацию о файловой структуре или даже просто имена других программ. Причина понятна: вирус должен каким-нибудь способом обеспечить передачу себе управления.

Классификация вирусов:

- по среде обитания
- по способу заражения среды обитания
- по степени воздействия
- по особенностям алгоритма

В зависимости от среды обитания вирусы можно разделить на сетевые, файловые, загрузочные и файлово-загрузочные.

Сетевые вирусы распространяются по различным компьютерным сетям.

Файловые вирусы внедряются главным образом в исполняемые модули, т. е. В файлы, имеющие расширения COM и EXE. Файловые вирусы могут внедряться и в другие типы файлов, но, как правило, записанные в таких файлах, они никогда не получают управление и, следовательно, теряют способность к размножению.

Загрузочные вирусы внедряются в загрузочный сектор диска (Boot-сектор) или в сектор, содержащий программу загрузки системного диска (Master Boot Re-cord).

Файлово-загрузочные вирусы заражают как файлы, так и загрузочные сектора дисков.

По способу заражения вирусы делятся на резидентные и нерезидентные.

Резидентный вирус при заражении (инфицировании) компьютера оставляет в оперативной памяти свою резидентную часть, которая потом перехватывает обращение операционной системы к объектам заражения (файлам, загрузочным секторам дисков и т. п.) и внедряется в них. Резидентные вирусы находятся в памяти и являются активными вплоть до выключения или перезагрузки компьютера.

Нерезидентные вирусы не заражают память компьютера и являются активными ограниченное время.

По степени воздействия вирусы можно разделить на следующие виды:

- неопасные, не мешающие работе компьютера, но уменьшающие объем свободной оперативной памяти и памяти на дисках, действия таких вирусов проявляются в каких-либо графических или звуковых эффектах
- опасные вирусы, которые могут привести к различным нарушениям в работе компьютера
- очень опасные, воздействие которых может привести к потере программ, уничтожению данных, стиранию информации в системных областях диска.

По особенностям алгоритма вирусы трудно классифицировать из-за большого разнообразия.

Простейшие вирусы - паразитические, они изменяют содержимое файлов и секторов диска и могут быть достаточно легко обнаружены и уничтожены.

Вирусы-репликаторы, называемые червями, которые распространяются по компьютерным сетям, вычисляют адреса сетевых компьютеров и записывают по этим адресам свои копии.

Известны вирусы-невидимки, называемые стелс-вирусами, которые очень трудно обнаружить и обезвредить, так как они перехватывают обращения операционной системы к пораженным файлам и секторам дисков и подставляют вместо своего тела незараженные участки диска.

Наиболее трудно обнаружить вирусы-мутанты, содержащие алгоритмы шифровки-расшифровки, благодаря которым копии одного и того же вируса не имеют ни одной повторяющейся цепочки байтов. Имеются и так называемые квазивирусные или «троянские» программы, которые хотя и не способны к самораспространению, но очень опасны, так как, маскируясь под полезную программу, разрушают загрузочный сектор и файловую систему дисков.

Загрузочные вирусы

Рассмотрим схему функционирования очень простого загрузочного вируса, заражающего дискеты. (boot-sector).

Пусть у вас имеются чистая дискета и зараженный компьютер, под которым мы понимаем компьютер с активным резидентным вирусом. Как только этот вирус обнаружит, что в дисководе появилась подходящая жертва - в нашем случае не защищенная от записи и еще не зараженная дискета, он приступает к заражению. Заражая дискету, вирус производит следующие действия:

- выделяет некоторую область диска и помечает ее как недоступную операционной системе, это можно сделать по-разному, в простейшем и традиционном случае занятые вирусом секторы помечаются как сбойные (bad)
- копирует в выделенную область диска свой хвост и оригинальный (здоровый) загрузочный сектор
- замещает программу начальной загрузки в загрузочном секторе (настоящем) своей головой
- организует цепочку передачи управления согласно схеме.

Таким образом, голова вируса теперь первой получает управление, вирус устанавливается в память и передает управление оригинальному загрузочному сектору.

Файловые вирусы

Рассмотрим теперь схему работы простого файлового вируса.

В отличие от загрузочных вирусов, которые практически всегда резидентны, файловые вирусы совсем не обязательно резидентны. Рассмотрим схему функционирования нерезидентного файлового вируса. Пусть у нас имеется инфицированный исполняемый файл. При запуске такого файла вирус получает управление, производит некоторые действия и передает управление «хозяину»

Какие же действия выполняет вирус? Он ищет новый объект для заражения - подходящий по типу файл, который еще не заражен. Заражая файл, вирус внедряется в его код, чтобы получить управление при запуске этого файла. Кроме своей основной функции - размножения, вирус вполне может сделать что-нибудь замысловатое (сказать, спросить, сыграть) - это уже зависит от фантазии автора вируса. Если файловый вирус резидентный, то он установится в память и получит возможность заражать файлы и проявлять прочие способности не только во время работы зараженного файла. Заражая исполняемый файл, вирус всегда изменяет его код - следовательно, заражение исполняемого файла всегда можно обнаружить.

Но, изменяя код файла, вирус не обязательно вносит другие изменения:

- ◊ он не обязан менять длину файла
- ◊ неиспользуемые участки кода
- ◊ не обязан менять начало файла

Таким образом, при запуске любого файла вирус получает управление (операционная система запускает его сама), резидентно устанавливается в память и передает управление вызванному файлу.

Загрузочно-файловые вирусы

Основное разрушительное действие - шифрование секторов винчестера. При каждом запуске вирус шифрует очередную порцию секторов, а, зашифровав половину жесткого диска, радостно сообщает об этом. Основная проблема при лечении данного вируса состоит в том, что недостаточно просто удалить вирус из файлов, надо расшифровать зашифрованную им информацию.

Полиморфные вирусы

Этот вид компьютерных вирусов представляется на сегодняшний день наиболее опасным. Объясним же, что это такое.

Полиморфные вирусы - вирусы, модифицирующие свой код в зараженных программах таким образом, что два экземпляра одного и того же вируса могут не совпадать ни в одном бите.

Такие вирусы не только шифруют свой код, используя различные пути шифрования, но и содержат код генерации шифровщика и расшифровщика, что отличает их от обычных шифровальных вирусов, которые также могут шифровать участки своего кода, но имеют при этом постоянный код шифровальщика и расшифровщика.

Полиморфные вирусы - это вирусы с самомодифицирующимися расшифровщиками. Цель такого шифрования: имея зараженный и оригинальный файлы, вы все равно не сможете проанализировать его код с помощью обычного дизассемблирования. Этот код зашифрован и представляет собой бессмысленный набор команд. Расшифровка производится самим вирусом уже непосредственно во время выполнения. При этом возможны варианты: он может расшифровать себя всего сразу, а может выполнить такую расшифровку «по ходу дела», может вновь шифровать уже отработавшие участки. Все это делается ради затруднения анализа кода вируса.

Стелс-вирусы

Стелс-вирусы обманывают антивирусные программы и в результате остаются незамеченными. Тем не менее, существует простой способ отключить механизм маскировки стелс-вирусов. Достаточно загрузить компьютер с не зараженной системной дискеты и сразу, не запуская других программ с диска компьютера (которые также могут оказаться зараженными), проверить компьютер антивирусной программой.

Троянские кони, программные закладки и сетевые черви

Троянский конь – это программа, содержащая в себе некоторую разрушающую функцию, которая активизируется при наступлении некоторого условия срабатывания. Обычно такие программы маскируются под какие-нибудь полезные утилиты. Вирусы могут нести в себе троянских коней или "троянизировать" другие программы – вносить в них разрушающие функции.

«Троянские кони» представляют собой программы, реализующие помимо функций, описанных в документации, и некоторые другие функции, связанные с нарушением безопасности и деструктивными действиями. Отмечены случаи создания таких программ с целью облегчения распространения вирусов. Списки таких программ широко публикуются в зарубежной печати. Обычно они маскируются под игровые или развлекательные программы и наносят вред под красивые картинки или музыку.

Если вирусы и «троянские кони» наносят ущерб посредством лавинообразного саморазмножения или явного разрушения, то основная функция вирусов типа «червь», действующих в компьютерных сетях, – взлом атакуемой системы, т.е. преодоление защиты с целью нарушения безопасности и целостности.

В более 80% компьютерных преступлений, расследуемых ФБР, "взломщики" проникают в атакуемую систему через глобальную сеть Internet. Когда такая попытка удается, будущее компании, на создание которой ушли годы, может быть поставлено под угрозу за какие-то секунды.

Этот процесс может быть автоматизирован с помощью вируса, называемого сетевой червь.

Червями называют вирусы, которые распространяются по глобальным сетям, поражая целые системы, а не отдельные программы. Это самый опасный вид вирусов, так как объектами нападения в этом случае становятся информационные системы государственного масштаба. С появлением глобальной сети Internet этот вид нарушения безопасности представляет наибольшую угрозу, т. к. ему в любой момент может подвергнуться любой из 40 миллионов компьютеров, подключенных к этой сети.

Признаки появления вирусов

При заражении компьютера вирусом важно его обнаружить. Для этого следует знать об основных признаках проявления вирусов. К ним можно отнести следующие:

- прекращение работы или неправильная работа ранее успешно функционировавших программ
- медленная работа компьютера
- невозможность загрузки операционной системы
- исчезновение файлов и каталогов или искажение их содержимого
- изменение даты и времени модификации файлов
- изменение размеров файлов
- неожиданное значительное увеличение количества файлов на диске
- существенное уменьшение размера свободной оперативной памяти
- вывод на экран непредусмотренных сообщений или изображений
- подача непредусмотренных звуковых сигналов
- частые зависания и сбои в работе компьютера

Методы защиты от компьютерных вирусов

Каким бы не был вирус, пользователю необходимо знать основные методы защиты от компьютерных вирусов.

Для защиты от вирусов можно использовать:

- общиесредства защиты информации, которые полезны также и как страховка от физической порчи дисков, неправильно работающих программ или ошибочных действий пользователя;
- профилактические меры, позволяющие уменьшить вероятность заражения вирусом;
- специализированные программы для защиты от вирусов.

Общие средства защиты информации полезны не только для защиты от вирусов. Имеются две основные разновидности этих средств:

- копирование информации - создание копий файлов и системных областей дисков;
- разграничение доступа предотвращает несанкционированное использование информации, в частности, защиту от изменений программ и данных вирусами, неправильно работающими программами и ошибочными действиями пользователей.

Антивирусные программы

Несмотря на то, что общие средства защиты информации очень важны для защиты от вирусов, все же их недостаточно. Необходимо и применение специализированных программ для защиты от вирусов. Эти программы можно разделить на несколько видов: детекторы, доктора (фаги), ревизоры, доктора-ревизоры, фильтры и вакцины (иммунизаторы).

ПРОГРАММЫ-ДЕТЕКТОРЫ позволяют обнаруживать файлы, зараженные одним из нескольких известных вирусов. Эти программы проверяют, имеется ли в файлах на указанном пользователем диске специфическая для данного вируса комбинация байтов. При ее обнаружении в каком-либо файле на экран выводится соответствующее сообщение.

Многие детекторы имеют режимы лечения или уничтожения зараженных файлов.

Следует подчеркнуть, что программы-детекторы могут обнаруживать только те вирусы, которые ей "известны". Программа Scan фирмы McAfee Associates и Aidstest Д.Н. Лозинского позволяют обнаруживать около 1000 вирусов, но всего их более пяти тысяч! Некоторые программы-детекторы, например Norton AntiVirus или AVSP фирмы "Диалог-МГУ", могут настраивать на новые типы

вирусов, им необходимо лишь указать комбинации байтов, присущие этим вирусам. Тем не менее невозможно разработать такую программу, которая могла бы обнаруживать любой заранее неизвестный вирус.
AIDSTEST
DOCTOR WEB
Microsoft Antivirus
ADINF
(Advanced Diskinfoscope)

ДЕЙСТВИЯ ПРИ ЗАРАЖЕНИИ ВИРУСОМ

При заражении компьютера вирусом (или при подозрении на это) важно соблюдать 4-е правила:

1) Прежде всего не надо торопиться и принимать опрометчивых решений.

Непродуманные действия могут привести не только к потери части файлов, но к повторному заражению компьютера.

2) Надо немедленно выключить компьютер, чтобы вирус не продолжал своих разрушительных действий.

3) Все действия по обнаружению вида заражения и лечению компьютера следует выполнять при загрузке компьютера с защищенной от записи дискеты с ОС (обязательное правило).

4) Если Вы не обладаете достаточными знаниями и опытом для лечения компьютера, попросите помочь более опытных коллег.

ПРОГРАММЫ-РЕВИЗОРЫ имеют две стадии работы. Сначала они запоминают сведения о состоянии программ и системных областей дисков (загрузочного сектора и сектора с таблицей разбиения жесткого диска). Предполагается, что в этот момент программы и системные области дисков не заражены. После этого с помощью программы-ревизора можно в любой момент сравнить состояние программ и системных областей дисков с исходным. О выявленных несоответствиях сообщается пользователю.

ДОКТОРА-РЕВИЗОРЫ, - программы, которые не только обнаруживают изменения в файлах и системных областях дисков, но и могут в случае изменений автоматически вернуть их в исходное состояние. Такие программы могут быть гораздо более универсальными, чем программы-доктора, поскольку при лечении они используют заранее сохраненную информацию о состоянии файлов и областей дисков. Это позволяет им вылечивать файлы даже от тех вирусов, которые не были созданы на момент написания программы.

Но они могут лечить не от всех вирусов, а только от тех, которые используют "стандартные", известные на момент написания программы, механизмы заражения файлов.

Существуют также ПРОГРАММЫ-ФИЛЬТРЫ, которые располагаются резидентно в оперативной памяти компьютера и перехватывают те обращения к операционной системе, которые используются вирусами для размножения и нанесения вреда, и сообщают о них пользователя. Пользователь может разрешить или запретить выполнение соответствующей операции.

Некоторые программы-фильтры не "ловят" подозрительные действия, а проверяют вызываемые на выполнение программы на наличие вирусов. Это вызывает замедление работы компьютера.

Однако преимущества использования программ-фильтров весьма значительны – они позволяют обнаружить многие вирусы на самой ранней стадии, когда вирус еще не успел размножиться и что-либо испортить. Тем самым можно свести убытки от вируса к минимуму.

ПРОГРАММЫ-ВАКЦИНЫ, или ИММУНИЗАТОРЫ, модифицируют программы и диски таким образом, что это не отражается на работе программ, но тот вирус, от которого производится вакцинация, считает эти программы или диски уже зараженными. Эти программы крайне неэффективны.

Ни один тип антивирусных программ по отдельности не дает полной защиты от вирусов. Лучшей стратегией защиты от вирусов является многоуровневая, "эшелонированная" оборона. Опишу структуру этой обороны.

19. Технологическая и эксплуатационная безопасность ПО. Классификация систем защиты ПО. Системы защиты от несанкционированного копирования и изменения

Разработка терминологии в области обеспечения безопасности ПО является базисом для формирования нормативно-правового обеспечения и концептуальных основ по рассматриваемой проблеме. Единая терминологическая база является ключом к единству взглядов в области, информационной безопасности, стимулирует скорейшее развитие методов и средств защиты ПО. Термины освещают основные понятия, используемые в рассматриваемой области на данный период времени. Определения освещают толкование конкретных форм, методов и средств обеспечения информационной безопасности.

Термины и определения

Непреднамеренный дефект - объективно и (или) субъективно образованный дефект, приводящий к получению неверных решений (результатов) или нарушению функционирования КС.

Преднамеренный дефект - криминальный дефект, внесенный субъектом для целенаправленного нарушения и (или) разрушения информационного ресурса.

Разрушающее программное средство (РПС) - совокупность программных и/или технических средств, предназначенных для нарушения (изменения) заданной технологии обработки информации и/или целенаправленного разрушения извне внутреннего состояния информационно-вычислительного процесса в КС.

Средства активного противодействия - средства защиты информационного ресурса КС, позволяющие блокировать канал утечки информации, разрушающие действия противника, минимизировать нанесенный ущерб и предотвращать дальнейшие деструктивные действия противника посредством ответного воздействия на его информационный ресурс.

Несанкционированный доступ - действия, приводящие к нарушению целостности, конфиденциальности и доступности информационных ресурсов.

Нарушитель (злоумышленник, противник) - субъект (субъекты), совершающие несанкционированный доступ к информационному ресурсу.

Модель угроз - вербальная, математическая, имитационная или натурная модель, формализующая параметры внутренних и внешних угроз безопасности ПО.

Оценка безопасности ПО - процесс получения количественных и/или качественных показателей информационной безопасности при учете преднамеренных и непреднамеренных дефектов в системе.

Система обеспечения информационной безопасности - объединенная совокупность мероприятий, методов и средств, создаваемых и поддерживаемых для обеспечения требуемого уровня безопасности информационного ресурса.

Информационная технология - упорядоченная совокупность организационных, технических и технологических процессов создания ПО и обработки, хранения и передачи информации.

Технологическая безопасность - свойство программного обеспечения и информации не быть преднамеренно искаженными и (или) начиненными избыточными модулями (структурами) диверсионного назначения на этапе создания КС.

Эксплуатационная безопасность - свойство программного обеспечения и информации не быть несанкционированно искаженными (измененными) на этапе их эксплуатации.

Жизненный цикл программного обеспечения компьютерных систем. Технологическая и эксплуатационная безопасность программ

Необходимость определения этапов жизненного цикла (ЖЦ) ПО обусловлена стремлением разработчиков к повышению качества ПО за счет оптимального управления разработкой и использования разнообразных механизмов контроля качества на каждом этапе, начиная от постановки задачи и заканчивая авторским сопровождением ПО. Наиболее общим представлением жизненного цикла ПО является модель в виде базовых этапов – процессов [Лип1], к которым относятся:

· системный анализ и обоснование требований к ПО;
· предварительное (эскизное) и детальное (техническое) проектирование ПО;
· разработка программных компонент, их комплексирование и отладка ПО в целом;
· испытания, опытная эксплуатация и тиражирование ПО;
· регулярная эксплуатация ПО, поддержка эксплуатации и анализ результатов;

· сопровождение ПО, его модификация и совершенствование, создание новых версий.

Данная модель является общепринятой и соответствует как отечественным нормативным документам в области разработки программного обеспечения, так и зарубежным. С точки зрения обеспечения технологической безопасности целесообразно рассмотреть более подробно особенности представления этапов ЖЦ.

Графическое представление моделей ЖЦ позволяет наглядно выделить их особенности и некоторые свойства процессов. Первоначально была создана каскадная модель ЖЦ [Лип1], в которой крупные этапы начинались друг за другом с использованием результатов предыдущих работ. Наиболее специфической является спиралевидная модель ЖЦ [Лип1]. В этой модели внимание концентрируется на итерационном процессе начальных этапов проектирования. На этих этапах последовательно создаются концепции, спецификации требований, предварительный и детальный проект. На каждом витке уточняется содержание работ и концентрируется облик создаваемого ПО.

Для проектирования ПО сложной системы, особенно системы реального времени, целесообразно использовать общесистемную модель ЖЦ, основанную на объединении всех известных работ в рамках рассмотренных базовых процессов. Эта модель предназначена для использования при планировании, составлении рабочих графиков, управлении различными программными проектами.

Совокупность этапов данной модели ЖЦ целесообразно делить на две части, существенно различающихся особенностями процессов, технико-экономическими характеристиками и влияющими на них факторами.

В первой части ЖЦ производится системный анализ, проектирование, разработка, тестирование и испытания ПО. Номенклатура работ, их трудоемкость, длительность и другие характеристики на этих этапах существенно зависят от объекта и среды разработки. Изучение подобных зависимостей для различных классов ПО позволяет прогнозировать состав и основные характеристики графиков работ для новых версий ПО.

Этой совокупности этапов ЖЦ ПО соответствует процесс внесения в разрабатываемые программы определенных защитных функций. Этот процесс называется обеспечением технологической безопасности и характеризуется необходимостью предотвращения модификации ПО за счет внедрения РПС априорного типа (алгоритмических и программных закладок).

Вторая часть ЖЦ, отражающая поддержку эксплуатации и сопровождения ПО, относительно слабо связана с характеристиками объекта и среды разработки. Номенклатура работ на этих этапах более стабильна, а их трудоемкость и длительность могут существенно изменяться, и зависят от массовости применения ПО. Для любой модели ЖЦ обеспечение высокого качества программных комплексов возможно лишь при использовании регламентированного технологического процесса на каждом из этих этапов. Такой процесс поддерживается CASE-средствами (средствами автоматизации разработки ПО), которые целесообразно выбирать из имеющихся или создавать с учетом объекта разработки и адекватного ему перечня работ.

Этапам эксплуатации и сопровождения ПО соответствует процесс обеспечения эксплуатационной безопасности ПО. Этот процесс характеризуется необходимостью защиты программ от компьютерных вирусов и программных закладок апостериорного типа. Последние могут внедряться за счет злонамеренного использования методов исследования программ и их спецификаций. Кроме того, на этапе обеспечения эксплуатационной безопасности ПО применяются методы защиты программ от несанкционированного копирования, распространения и использования.

Использование при создании программного обеспечения КС сложных операционных систем, инструментальных средств разработки ПО увеличивают потенциальную возможность внедрения в программы преднамеренных дефектов диверсионного типа. Помимо этого, при создании прикладного программного обеспечения всегда необходимо исходить из возможности наличия в коллективе разработчиков программистов - злоумышленников, которые в силу тех или иных причин могут внести в разрабатываемые программы РПС.

Характерным свойством РПС в данном случае является возможность внезапного и незаметного нарушения или полного вывода из строя КС. Функционирование РПС реализуется в рамках модели угроз безопасности ПО, основные элементы которой рассматриваются в следующем разделе.

Принципы классификации систем защиты программного обеспечения (по)

Средства защиты(СЗ) ПО можно классифицировать по ряду признаков:

- По используемому механизму защиты

- По методу установки
- По принципу функционирования

Классификация систем защиты по по методу установки

Для производителей ПО удобней всего использовать защиту, устанавливаемую на скомпилированные модули. Но такая защита наименее стойка к атакам. Системы с внедрением СЗ в исходный код неудобны для производителей, так как им приходится обучать персонал работе с СЗ и ряд других неудобств. Лучшим решением является использование СЗ комбинированного типа.

Системы защиты ПО по методу установки можно подразделить:

- на системы, устанавливаемые на скомпилированные модули ПО;
- системы, встраиваемые в исходный код ПО до компиляции;
- комбинированные.

Системы первого типа наиболее удобны для производителя ПО, так как легко можно защитить уже полностью готовое и оттестированное ПО (обычно процесс установки защиты максимально автоматизирован и сводится к указанию имени защищаемого файла и нажатию "Enter"), а потому и наиболее популярны. В то же время стойкость этих систем достаточно низка, так как для обхода защиты достаточно определить точку завершения работы "конверта" защиты и передачи управления защищенной программе, а затем принудительно ее сохранить в незащищенном виде.

Системы второго типа неудобны для производителя ПО, так как возникает необходимость обучать персонал работе с программным интерфейсом (API) системы защиты с вытекающими отсюда денежными и временными затратами. Но такие системы являются более стойкими к атакам, потому что здесь исчезает четкая граница между системой защиты и как таковым ПО.

Наиболее живучими являются комбинированные системы защиты. Сохраняя достоинства и недостатки систем второго типа, они максимально затрудняют анализ и дезактивацию своих алгоритмов.

Классификация систем защиты ПО по используемым механизмам защиты

Возможно два подхода решения проблемы защиты ПО. Один – это решение проблемы с помощью технических средств защиты. Включать СЗ в состав ПО. Другой – использование юридической защиты. ПО не содержит технические СЗ, а сопровождается информацией об управлении правами.

По используемым механизмам защиты СЗ можно классифицировать на:

системы, использующие сложные логические механизмы;

системы, использующие шифрование защищаемого ПО - комбинированные системы.

Системы первого типа используют различные методы и приёмы, ориентированные на затруднение дизассемблирования, отладки и анализа алгоритма СЗ и защищаемого ПО. Этот тип СЗ наименее стоек к атакам, так как для преодоления защиты достаточно проанализировать логику процедур проверки и должным образом их модифицировать.

Более стойкими являются системы второго типа. Для дезактивации таких защит необходимо определение ключа дешифрации ПО. Самыми стойкими к атакам являются комбинированные системы.

Классификация систем защиты по по принципу функционирования

По принципу функционирования СЗ можно подразделить на: - упаковщики/шифраторы;- СЗ от несанкционированного копирования- СЗ от несанкционированного доступа (НСД).

Назначение упаковщиков-шифраторов

Упаковщики/шифраторы. Их цель защита ПО от анализа его алгоритмов и несанкционированной модификации. Для достижения этого используются алгоритмы компрессии данных; шифрование данных, алгоритмы мутации, запутывание логики программы, приведение ОС в нестабильное состояние на время работы ПО и др.

Системы защиты от несанкционированного копирования

СЗ от несанкционированного копирования осуществляют "привязку" ПО к дистрибутивному носителю (гибкий диск, CD ...). Данный тип защит основывается на изучении работы контроллеров накопителей, их физических показателей, нестандартных режимах разбивки, чтения/записи и т.п.

Системы защиты от копирования можно разделить на следующие группы: - привязка к дискете; - привязка к компьютеру; - привязка к ключу; - опрос справочников - ограничение использования ПО.

7. Системы защиты от несанкционированного доступа

В защите информации ПК от НСД можно выделить три основных направления:

- специальные технических средства опознавания пользователя; -специальное программное обеспечение по защите информации; -специальные средства защиты информации ПК от несанкционированного доступа.

Системы защиты информации от НСД обеспечивают выполнение следующих функций:

1. идентификация, т.е. присвоение уникальных признаков - идентификаторов, по которым в дальнейшем система производит аутентификацию

2. аутентификация, т.е. установление подлинности на основе сравнения с эталонными идентификаторами;

3. разграничение доступа пользователей к ПЭВМ;

4. разграничение доступа пользователей по операциям над ресурсами (программы, данные и т.д.);

5. администрирование: a. определение прав доступа к защищаемым ресурсам, b. обработка регистрационных журналов, c. установка системы защиты на ПЭВМ, d. снятие системы защиты с ПЭВМ;

6. регистрация событий: a. входа пользователя в систему, b. выхода пользователя из системы, c. нарушения прав доступа;

7. реакция на попытки НСД;

8. контроль целостности и работоспособности систем защиты;

9. обеспечение информационной безопасности при проведении ремонтно-профилактических работ;

10. обеспечение информационной безопасности в аварийных ситуациях. Права пользователей по доступу к программам и данным описывают таблицы, на основе которых и производится контроль и разграничение доступа к ресурсам. Доступ должен контролироваться программными средствами защиты. Если запрашиваемый доступ не соответствует имеющемуся в таблице прав доступа, то системы защиты регистрирует факт НСД и инициализирует соответствующую реакцию.

Системы привязки по к компьютеру

Системы "привязки" ПО при установке на ПК пользователя осуществляют поиск уникальных признаков компьютерной системы, либо они устанавливаются самой системой защиты. После этого модуль защиты в самом ПО настраивается на поиск и идентификацию данных признаков, по которым в дальнейшем определяется авторизованное или неавторизованное использование ПО. При этом возможно применение методик оценки скоростных и иных показателей процессора, материнской платы, дополнительных устройств, ОС, чтение/запись в микросхемы энергонезависимой памяти, запись скрытых файлов, настройка на наиболее часто встречаемую карту использования ОЗУ и т.п.

4. 9. Выделение объективных характеристик программы

Идентификация программы или отдельного модуля представляет интерес в том случае, когда другие методы защиты не приносят успеха. Широко обсуждаются проблемы авторского права для отдельной процедуры программы и взаимосвязь между идеей и способом ее реализации. Выделение объективных характеристик программы - довольно сложная процедура, тем не менее, признаки подобия двух программ или модулей, содержащихся в больших программах, указать можно. Проблема заключается в том, чтобы уметь идентифицировать программы, которые изменены хакером, погружены в другую программу или откомпилированы в машинный код.

"Родимые пятна"

Понятие "родимые пятна" используется для описания характеристик, появляющихся в результате естественного процесса разработки программы и относящихся к особенностям стиля программирования, ошибкам и избыточностям, которые не должны иметь места в независимо написанной программе. Каждое из них может служить убедительной уликой нарушения авторского права.

Водяные знаки как метод пассивной защиты.

Использование водяных знаков как метода выявления подделки занимает особое место, поскольку препятствует созданию точной копии, которую пользователь не мог бы отличить от оригинала.

Психологические методы защиты.

Психологические методы основаны на том, чтобы создать у нарушителя чувство неуверенности и психологического напряжения, заставляя его все время помнить, что в похищенном программном продукте могут сохраняться средства защиты. Поэтому полезно было бы дать объявление, что в программное обеспечение встроены механизмы защиты (независимо от того, так ли это на самом деле). Существует огромное число хитроумных способов расстановки отличительных меток в программе и никакой хакер не может быть уверен, что ему удалось уничтожить все ключи и механизмы защиты.

Существующие системы защиты ПС (СЗ ПС) можно классифицировать по ряду признаков: методы установки, используемые механизмы защиты и принципы функционирования.

Методы установки

Системы, устанавливаемые на скомпилированные модули ПС, удобны для производителя ПС, так как позволяют легко защитить полностью готовое и оттестированное ПС. Стойкость этих систем достаточно низка, так как для обхода защиты достаточно определить точку завершения работы «конверта» защиты и передачи управления защищенной программе, а затем принудительно ее сохранить в незащищенном виде.

Системы, встраиваемые в исходный код ПС до компиляции, неудобны для производителя ПС, так как возникает необходимость обучать персонал работе с программным интерфейсом (API) системы защиты с вытекающими отсюда денежными и временными затратами. Кроме того, усложняется процесс тестирования ПС и снижается его надежность, поскольку, кроме самого ПС, ошибки могут быть как в API системе защиты, так и в процедурах, его использующих. Но такие системы являются стойкими к атакам, потому что в них исчезает четкая граница между системой защиты и ПС как таковым.

Комбинированные системы защитыявляются наиболее живучими. Сохраняя достоинства и недостатки систем второго типа, они максимально затрудняют анализ и дезактивацию своих алгоритмов.

Методы защиты

Рассмотрим основные методы:

· алгоритмы запутывания: используются хаотичные переходы в разные части кода, внедрение ложных процедур-«пустышек», холостые циклы, искажение количества реальных параметров процедур ПС, разброс участков кода по разным областям ОЗУ;

· алгоритмы мутации: создаются таблицы соответствия операндов-синонимов и замена их друг на друга при каждом запуске программы по определенной схеме или случайным образом, случайные изменения структуры программы;

· алгоритмы компрессии данных: программа упаковывается, а затем распаковывается по мере выполнения;

· алгоритмы шифрования данных: программа шифруется, а затем расшифровывается по мере выполнения;

· вычисление сложных математических выражений в процессе отработки механизма защиты: элементы логики защиты зависят от результата вычисления значения какой-либо формулы или группы формул;

· методы затруднения дизассемблирования: используются различные приемы, направленные на предотвращение дизассемблирования в пакетном режиме;

· методы затруднения отладки: используются различные приемы, направленные на усложнение отладки программы;

· эмуляция процессоров и операционных систем: создаются виртуальный процессор и/или операционная система (не обязательно реально существующие) и программа-переводчик из системы команд IBM в систему команд созданного процессора или ОС; после такого перевода ПС может выполняться только при помощи эмулятора, что резко затрудняет исследование алгоритма ПС;

· нестандартные методы работы с аппаратным обеспечением: модули системы защиты обращаются к аппаратуре ЭВМ, минуя процедуры операционной системы, и используют мало известные или недокументированные ее возможности;

· шифрование защищаемого ПС:для дезактивации защиты необходимо определение ключа дешифрации ПС;

· комбинированные методы.

Принципы функционирования

Упаковщики/шифраторызащищают ПС от анализа его алгоритмов и несанкционированной модификации. Для достижения этого используются: алгоритмы компрессии данных; приемы, связанные с использованием недокументированных особенностей операционных систем (ОС) и процессоров; шифрование данных, алгоритмы мутации, запутывание логики программы и др. Этот вид защиты замедляет выполнение ПС, затрудняет обновление и исправление ошибок в ПС.

Защита от несанкционированного копирования обеспечивает «привязку» ПС к дистрибутивному носителю (гибкий диск, CD...). Данный тип защиты основан на глубоком изучении работы контроллеров-накопителей, их физических показателей, нестандартных режимах разбивки, чтения/записи и т.п. При этом на физическом уровне создается дистрибутивный носитель, обладающий (предположительно) неповторимыми свойствами (обычно это достигается при помощи нестандартной разметки носителя информации или/и записи на него дополнительной информации – пароля или метки), а на программном – создается модуль, настроенный на идентификацию и аутентификацию носителя по его уникальным свойствам. При этом возможно применение приемов, используемых упаковщиками/шифраторами.

20. Основные принципы обеспечения безопасности ПО на различных стадиях его жизненного цикла

В качестве объекта обеспечения технологической и эксплуатационной безопасности ПО рассматривается вся совокупность его компонентов в рамках конкретной КС. В качестве доминирующей должна использоваться стратегия сквозного тотального контроля технологического и эксплуатационного этапов жизненного цикла компонентов ПО. Совокупность мероприятий по обеспечению технологической и эксплуатационной безопасности компонентов ПО должна носить конфиденциальный характер. Необходимо обеспечить постоянный, комплексный и действенный контроль за деятельностью разработчиков и пользователей компонентов. Кроме общих принципов, обычно необходимо конкретизировать принципы обеспечения безопасности ПО на каждом этапе его жизненного цикла. Далее приводятся один из вариантов разработки таких принципов.

Принципы обеспечения технологической безопасности при обосновании, планировании работ и проектном анализе ПО

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Комплексности обеспечения безопасности ПО, предполагающей рассмотрение проблемы безопасности информационно - вычислительных процессов с учетом всех структур КС, возможных каналов утечки информации и несанкционированного доступа к ней, времени и условий их возникновения, комплексного применения организационных и технических мероприятий.

Планируемости применения средств безопасности программ, предполагающей перенос акцента на совместное системное проектирование ПО и средств его безопасности, планирование их использования в предполагаемых условиях эксплуатации.

Обоснованности средств обеспечения безопасности ПО, заключающейся в глубоком научно-обоснованном подходе к принятию проектных решений по оценке степени безопасности, прогнозированию угроз безопасности, всесторонней априорной оценке показателей средств защиты.

Достаточности безопасности программ, отражающей необходимость поиска наиболее эффективных и надежных мер безопасности при одновременной минимизации их стоимости.

Гибкости управления защитой программ, требующей от системы контроля и управления обеспечением информационной безопасности ПО способности к диагностированию, опережающей нейтрализации, оперативному и эффективному устранению возникающих угроз в условиях резких изменений обстановки информационной борьбы.

Заблаговременности разработки средств обеспечения безопасности и контроля производства ПО, заключающейся в предупредительном характере мер обеспечения технологической безопасности работ в интересах недопущения снижения эффективности системы безопасности процесса создания ПО.

Документируемости технологии создания программ, подразумевающей разработку пакета нормативно-технических документов по контролю программных средств на наличие преднамеренных дефектов.

Принципы достижения технологической безопасности ПО в процессе его разработки

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Регламентации технологических этапов разработки ПО, включающей упорядоченные фазы промежуточного контроля, спецификацию программных модулей и стандартизацию функций и формата представления данных.

Автоматизации средств контроля управляющих и вычислительных программ на наличие дефектов, создания типовой общей информационной базы алгоритмов, исходных текстов и программных средств, позволяющих выявлять преднамеренные программные дефекты.

Последовательной многоуровневой фильтрации программных модулей в процессе их создания с применением функционального дублирования разработок и поэтапного контроля.

Типизации алгоритмов, программ и средств информационной безопасности, обеспечивающей информационную, технологическую и программную совместимость, на основе максимальной их унификации по всем компонентам и интерфейсам.

Централизованного управления базами данных проектов ПО и администрирование технологии их разработки с жестким разграничением функций, ограничением доступа в соответствии со средствами диагностики, контроля и защиты.

Блокирования несанкционированного доступа соисполнителей и абонентов государственных сетей связи, подключенных к стендам для разработки программ.

Статистического учета и ведения системных журналов о всех процессах разработки ПО с целью контроля технологической безопасности.

Использования только сертифицированных и выбранных в качестве единых инструментальных средств разработки программ для новых технологий обработки информации и перспективных архитектур вычислительных систем.

Принципы обеспечения технологической безопасности на этапах стендовых и приемо-сдаточных испытаний

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Тестирования ПО на основе разработки комплексов тестов, параметризуемых на конкретные классы программ с возможностью функционального и статистического контроля в широком диапазоне изменения входных и выходных данных.

Проведения натурных испытаний программ при экстремальных нагрузках с имитацией воздействия активных дефектов.

Осуществления "фильтрации" программных комплексов с целью выявления возможных преднамеренных дефектов определенного назначения на базе создания моделей угроз и соответствующих сканирующих программных средств.

Разработки и экспериментальной отработки средств верификации программных изделий.

Проведения стендовых испытаний ПО для определения непреднамеренных программных ошибок проектирования и ошибок разработчика, приводящих к невыполнению целевых функций программ, а также выявление потенциально "узких" мест в программных средствах для разрушительного воздействия.

Отработки средств защиты от несанкционированного воздействия нарушителей на ПО.

Сертификации программных изделий АСУ по требованиям безопасности с выпуском сертификата соответствия этого изделия требованиям технического задания.

Принципы обеспечения безопасности при эксплуатации программного обеспечения

Принципы обеспечения безопасности ПО на данном этапе включают принципы:

Сохранения и ограничения доступа к эталонам программных средств, недопущение внесения изменений в них.

Профилактического выборочного тестирования и полного сканирования программных средств на наличие преднамеренных дефектов.

Идентификации ПО на момент ввода его в эксплуатацию в соответствии с предполагаемыми угрозами безопасности ПО и его контроль.

Обеспечения модификации программных изделий во время их эксплуатации путем замены отдельных модулей без изменения общей структуры и связей с другими модулями.

Строгого учета и каталогизации всех сопровождаемых программных средств, а также собираемой, обрабатываемой и хранимой информации.

Статистического анализа информации обо всех процессах, рабочих операциях, отступлениях от режимов штатного функционирования ПО.

Гибкого применения дополнительных средств защиты ПО в случае выявления новых, непрогнозируемых угроз информационной безопасности.

Хакерская атака в узком смысле слова — в настоящее время под словосочетанием понимается «Покушение на систему безопасности», и склоняется скорее к смыслу следующего термина Крэкерская атака. Это произошло из-за искажения смысла самого слова «хакер».

Хакерская атака в широком смысле слова (изначальный смысл) — мозговой штурм, направленный на нахождение пути решения сложных задач. В хакерской атаке могут принимать участие один или несколько высококлассных специалистов (хакеров). В результате мозгового штурма могут быть придуманы нетрадиционные методы решения проблемы, или внесены оптимизирующие корректировки в уже существующие методы.

Крэкерская атака — действие, целью которого является захват контроля (повышение прав) над удалённой/локальной вычислительной системой, либо её дестабилизация, либо отказ в обслуживании.

Считается самым старым методом атак, хотя суть его проста и примитивна: большое количество почтовых сообщений делают невозможными работу с почтовыми ящиками, а иногда и с целыми почтовыми серверами. Для этой цели было разработано множество программ, и даже неопытный пользователь мог совершить атаку, указав всего лишь e-mail жертвы, текст сообщения, и количество необходимых сообщений. Многие такие программы позволяли прятать реальный IP-адресотправителя, используя для рассылки анонимный почтовый сервер. Эту атаку сложно предотвратить, так как даже почтовые фильтры провайдеров не могут определить реального отправителя спама. Провайдер может ограничить количество писем от одного отправителя, но адрес отправителя и тема зачастую генерируются случайным образом.

Переполнение буфера

Пожалуй, один из самых распространенных типов атак в Интернете. Принцип данной атаки построен на использовании программных ошибок, позволяющих вызвать нарушение границ памяти и аварийно завершить приложение или выполнить произвольный бинарный код от имени пользователя, под которым работала уязвимая программа. Если программа работает под учётной записью администратора системы, то данная атака позволит получить полный контроль над компьютером жертвы, поэтому рекомендуется работать под учётной записью рядового пользователя, имеющего ограниченные права на системе, а под учётной записью администратора системы выполнять только операции, требующие административные права.

21. Понятие о политике безопасности: анализ риска; угрозы/видимость; уязвимость/последствия; учет информационных ценностей

Под *политикой безопасности* организации понимают совокупность документированных управленческих решений, направленных на защиту информации и ассоциированных с ней ресурсов. Политика безопасности является тем средством, с помощью которой реализуется деятельность в компьютерной информационной системе организации. Вообще политики безопасности определяются используемой компьютерной средой и отражают специфические потребности организации.

Обычно корпоративная информационная система представляет собой сложный комплекс разнородного, иногда плохо согласующегося между собой аппаратного и программного обеспечения: компьютеров, операционных систем, сетевых средств, СУБД, разнообразных приложений. Все эти компоненты обычно обладают собственными средствами защиты, которые нужно согласовать между собой. Поэтому очень важна эффективная политика безопасности в качестве согласованной платформы по обеспечению безопасности корпоративной системы. По мере роста компьютерной системы и интеграции ее в глобальную сеть необходимо обеспечить отсутствие в системе слабых мест, поскольку все усилия по защите информации могут быть обесценены лишь одной оплошностью.

Можно построить такую политику безопасности, которая будет устанавливать, кто имеет доступ к конкретным активам и приложениям, какие роли и обязанности будут иметь конкретные лица, а также предусмотреть процедуры безопасности, которые четко предписывают, как должны выполняться конкретные задачи безопасности. Индивидуальные особенности работы сотрудника могут потребовать доступа к информации, которая не должна быть доступна другим работникам. *Например, менеджер по персоналу может иметь доступ к частной информации любого сотрудника, в то время как специалист по отчетности может иметь доступ только к финансовым данным этих сотрудников. А рядовой сотрудник будет иметь доступ только к своей собственной персональной информации.*

Политика безопасности определяет позицию организации по рациональному использованию компьютеров и сети, а также процедуры по предотвращению и реагированию на инциденты безопасности. В большой корпоративной системе может применяться широкий диапазон разных политик от бизнес-политик до специфичных правил доступа к наборам данных. Эти политики полностью определяются конкретными потребностями организации.

- обеспечение сохранности, целостности информационных ресурсов и предоставление доступа к ним в строгом соответствии с установленными приоритетами и правилами разграничения доступа;
- обеспечение защиты подсистем, задач и технологических процессов от угроз информационной безопасности;
- обеспечение защиты управляющей информации от угроз информационной безопасности;
- обеспечение защиты каналов связи.
- Политика безопасности *определяет стратегию управления в области информационной безопасности*, а также ту меру внимания и количество ресурсов, которые считает целесообразным выделить руководство.
- Политика безопасности строится на основе *анализа рисков*, которые признаются реальными для информационной системы организации. Когда проведен анализ рисков и определена стратегия защиты, составляется программа, реализация которой должна обеспечить информационную безопасность. Под эту программу выделяются ресурсы, назначаются ответственные, определяется порядок контроля выполнения программы и т.п.
- *Для того чтобы ознакомиться с основными понятиями политик безопасности рассмотрим в качестве конкретного примера гипотетическую локальную сеть, принадлежащую некоей организации, и связанную с ней политику безопасности.*
- Политика безопасности организации должна иметь структуру краткого, легко понимаемого документа высокоуровневой политики, поддерживаемого рядом более конкретных документов специализированных политик и процедур безопасности.
- Высокоуровневая политика безопасности должна периодически пересматриваться, чтобы гарантировать, что она учитывает текущие потребности организации. Этот документ

составляют таким образом, чтобы политика была относительно независимой от конкретных технологий. В таком случае этот документ политики не потребуется изменять слишком часто.

- Политика безопасности обычно оформляется в виде документа, включающего такие разделы, как: *описание проблемы, область применения, позиция организации, распределение ролей и обязанностей, санкции и др.*
- *Описание проблемы.* Информация, циркулирующая в рамках локальной сети, является критически важной. Локальная сеть позволяет пользователям совместно использовать программы и данные, что увеличивает угрозу безопасности. Поэтому каждый из компьютеров, входящих в сеть, нуждается в более сильной защите. Эти повышенные меры безопасности и являются темой данного документа. Документ преследует следующие цели: продемонстрировать сотрудникам организации важность защиты сетевой среды, описать их роль в обеспечении безопасности, а также распределить конкретные обязанности по защите информации, циркулирующей в сети.
- *Область применения.* В сферу действия данной политики попадают все аппаратные, программные и информационные ресурсы, входящие в локальную сеть предприятия. Политика ориентирована также на людей, работающих с сетью, в том числе на пользователей, субподрядчиков и поставщиков.

Позиция организации. Целью организации является обеспечение целостности, доступности и конфиденциальности данных, а также их полноты и актуальности. Более частными целями являются:

- обеспечение уровня безопасности, соответствующего нормативным документам;
- следование экономической целесообразности в выборе защитных мер (расходы на защиту не должны превосходить предполагаемый ущерб от нарушения информационной безопасности);
- обеспечение безопасности в каждой функциональной области локальной сети;
- обеспечение подотчетности всех действий пользователей с информацией и ресурсами;
- обеспечение анализа регистрационной информации;
- предоставление пользователям достаточной информации для сознательного поддержания режима безопасности;
- выработка планов восстановления после аварий и иных критических ситуаций для всех функциональных областей с целью обеспечения непрерывности работы сети;
- обеспечение соответствия с имеющимися законами и общеорганизационной политикой безопасности.

Распределение ролей и обязанностей. За реализацию сформулированных выше целей отвечают соответствующие должностные лица и пользователи сети.

- *Руководители подразделений* отвечают за доведение положений политики безопасности до пользователей и за контакты с ними.
- *Администраторы локальной сети* обеспечивают непрерывное функционирование сети и отвечают за реализацию технических мер, необходимых для проведения в жизнь политики безопасности.
- *Администраторы сервисов* отвечают за конкретные сервисы и, в частности, за то, чтобы защита была построена в соответствии с общей политикой безопасности.
- *Пользователи* обязаны работать с локальной сетью в соответствии с политикой безопасности, подчиняться распоряжениям лиц, отвечающих за отдельные аспекты безопасности, ставить в известность руководство обо всех подозрительных ситуациях.
- *Санкции.* Нарушение политики безопасности может подвергнуть локальную сеть и циркулирующую в ней информацию недопустимому риску. Случаи нарушения безопасности со стороны персонала должны оперативно рассматриваться руководством для принятия дисциплинарных мер вплоть до увольнения.
- *Дополнительная информация.* Конкретным группам исполнителей могут потребоваться для ознакомления какие-то дополнительные документы, в частности документы специализированных политик и процедур безопасности, а также другие руководящие указания. Необходимость в дополнительных документах политик безопасности в значительной степени зависит от размеров и сложности организации. Для достаточно большой организации могут потребоваться в дополнение к базовой политике специализированные политики безопасности.

Организации меньшего размера нуждаются только в некотором подмножестве специализированных политик. Многие из этих документов поддержки могут быть довольно краткими - объемом в одну - две страницы.

- С практической точки зрения политики безопасности можно разделить на три уровня: *верхний, средний и нижний.*

Верхний уровень политики безопасности определяет решения, затрагивающие организацию в целом. Эти решения носят весьма общий характер и исходят, как правило, от руководства организации.

Такие решения могут включать в себя следующие элементы:

- формулировка целей, которые преследует организация в области информационной безопасности, определение общих направлений в достижении этих целей;
- формирование или пересмотр комплексной программы обеспечения информационной безопасности, определение ответственных лиц за продвижение программы;
- обеспечение материальной базы для соблюдения законов и правил;
- формулировка управленческих решений по вопросам реализации программы безопасности, которые должны рассматриваться на уровне организации в целом.
- Политика безопасности *верхнего уровня* формулирует цели организации в области информационной безопасности в терминах целостности, доступности и конфиденциальности. Если организация отвечает за поддержание критически важных баз данных, на первом плане должна стоять целостность данных. Для организации, занимающейся продажами, важна актуальность информации о предоставляемых услугах и ценах, а также ее доступность максимальному числу потенциальных покупателей. Режимная организация в первую очередь будет заботиться о конфиденциальности информации, то есть о ее защите от несанкционированного доступа.
- На верхний уровень выносится управление ресурсами безопасности и координация использования этих ресурсов, выделение специального персонала для защиты критически важных систем, поддержание контактов с другими организациями, обеспечивающими или контролирующими режим безопасности.
- Политика верхнего уровня должна четко определять сферу своего влияния. Это могут быть все компьютерные системы организации или даже больше, если политика регламентирует некоторые аспекты использования сотрудниками своих домашних компьютеров. Возможна и такая ситуация, когда в сферу влияния включаются лишь наиболее важные системы.
- В политике должны быть определены обязанности должностных лиц по выработке программы безопасности и по проведению ее в жизнь, то есть политика может служить основой подотчетности персонала.
- Политика верхнего уровня имеет дело с тремя аспектами законопослушности и исполнительской дисциплины. Во-первых, организация должна *соблюдать существующие законы*. Во-вторых, следует *контролировать действия лиц, ответственных за выработку программы безопасности*. В-третьих, *необходимо обеспечить исполнительскую дисциплину персонала* с помощью системы поощрений и наказаний.

Средний уровень политики безопасности определяет решение вопросов, касающихся отдельных аспектов информационной безопасности, но важных для различных систем, эксплуатируемых организацией.

Примеры таких вопросов - отношение к доступу в Интернет (проблема сочетания свободы получения информации с защитой от внешних угроз), использование домашних компьютеров и т.д.

Политика безопасности среднего уровня должна определять для каждого аспекта информационной безопасности следующие моменты:

- описание аспекта - позиция организации может быть сформулирована в достаточно общем виде как набор целей, которые преследует организация в данном аспекте;
- область применения - следует специфицировать, где, когда, как, по отношению к кому и чему применяется данная политика безопасности;
- роли и обязанности - документ должен содержать информацию о должностных лицах, отвечающих за проведение политики безопасности в жизнь;

- санкции - политика должна содержать общее описание запрещенных действий и наказаний за них;
- точки контакта - должно быть известно, куда следует обращаться за разъяснениями, помощью и дополнительной информацией. Обычно «точкой контакта» служит должностное лицо.

Нижний уровень политики безопасности относится к конкретным сервисам. Эта политика включает в себя два аспекта: цели и правила их достижения, — поэтому ее порой трудно отделить от вопросов реализации. В отличие от двух верхних уровней, рассматриваемая политика должна быть более детальной.

Приведем несколько примеров вопросов, на которые следует дать ответ при следовании политике безопасности нижнего уровня:

кто имеет право доступа к объектам, поддерживаемым сервисом;

при каких условиях можно читать и модифицировать данные;

как организован удаленный доступ к сервису.

Политика безопасности нижнего уровня может исходить из соображений целостности, доступности и конфиденциальности, но она не должна на них останавливаться. В общем случае цели должны связывать между собой объекты сервиса и осмысленные действия с ними.

Из целей выводятся правила безопасности, описывающие, кто, что и при каких условиях может делать. Чем детальнее правила, чем более четко и формально они изложены, тем проще поддержать их выполнение программно-техническими мерами. Обычно наиболее формально задаются права доступа к объектам.

Приведем более детальное описание обязанностей каждой категории персонала.

Руководители подразделений отвечают за доведение положений политики безопасности до пользователей. Они обязаны:

постоянно держать в поле зрения вопросы безопасности. Следить за тем, чтобы то же самое делали их подчиненные;

проводить анализ рисков, выявляя активы, требующие защиты, и уязвимые места систем, оценивая размер возможного ущерба от нарушения режима безопасности и выбирая эффективные средства защиты;

организовать обучение персонала мерам безопасности. Обратить особое внимание на вопросы, связанные с антивирусным контролем;

информировать администраторов локальной сети и администраторов сервисов об изменении статуса каждого из подчиненных (переход на другую работу, увольнение и т.п.);

обеспечить, чтобы каждый компьютер в их подразделениях имел хозяина или системного администратора, отвечающего за безопасность и обладающего достаточной квалификацией для выполнения этой роли.

Администраторы локальной сети обеспечивают непрерывное функционирование сети и отвечают за реализацию технических мер, необходимых для проведения в жизнь политики безопасности. Они обязаны:

обеспечить защиту оборудования локальной сети, в том числе интерфейсов с другими сетями;

оперативно и эффективно реагировать на события, таящие угрозу. Информировать администраторов сервисов о попытках нарушения защиты;

использовать проверенные средства аудита и обнаружения подозрительных ситуаций. Ежедневно анализировать регистрационную информацию, относящуюся к сети в целом и к файловым серверам в особенности;

не злоупотреблять своими большими полномочиями. Пользователи имеют право на тайну;

разработать процедуры и подготовить инструкции для защиты локальной сети от вредоносного программного обеспечения. Оказывать помощь в обнаружении и ликвидации вредоносного кода;

регулярно выполнять резервное копирование информации, хранящейся на файловых серверах;

выполнять все изменения сетевой аппаратно-программной конфигурации;

гарантировать обязательность процедуры идентификации и аутентификации для доступа к сетевым ресурсам. Выделять пользователям входные имена и начальные пароли только после заполнения регистрационных форм;

периодически производить проверку надежности защиты локальной сети. Не допускать получения привилегий неавторизованными пользователями.

- Администраторы сервисов отвечают за конкретные сервисы и, в частности, за то, чтобы защита была построена в соответствии с общей политикой безопасности. Они обязаны:

управлять правами доступа пользователей к обслуживаемым объектам;

оперативно и эффективно реагировать на события, таящие угрозу. Оказывать помощь в отражении угрозы, выявлении нарушителей и предоставлении информации для их наказания;

- регулярно выполнять резервное копирование информации, обрабатываемой сервисом;

выделять пользователям входные имена и начальные пароли только после заполнения регистрационных форм;

ежедневно анализировать регистрационную информацию, относящуюся к сервису. Регулярно контролировать сервис на предмет вредоносного программного обеспечения;

- периодически производить проверку надежности защиты сервиса. Не допускать получения привилегий неавторизованными пользователями.

  Пользователи обязаны работать с локальной сетью в соответствии с политикой безопасности, подчиняться распоряжениям лиц, отвечающих за отдельные аспекты безопасности, ставить в известность руководство обо всех подозрительных ситуациях. Они обязаны:

- знать и соблюдать законы, правила, принятые в данной организации, политику безопасности, процедуры безопасности. Использовать доступные защитные механизмы для обеспечения конфиденциальности и целостности своей информации;
- использовать механизм защиты файлов и должным образом задавать права доступа;
- выбирать качественные пароли, регулярно менять их. Не записывать пароли на бумаге, не сообщать их другим лицам;
- информировать администраторов или руководство о нарушениях безопасности и иных подозрительных ситуациях;
- не использовать слабости в защите сервисов и локальной сети в целом. Не совершать неавторизованной работы с данными, не создавать помех другим пользователям;
- всегда сообщать корректную идентификационную и аутентификационную информацию, не пытаться работать от имени других пользователей;
- обеспечивать резервное копирование информации с жесткого диска своего компьютера;
- знать принципы работы вредоносного программного обеспечения, пути его проникновения и распространения. Знать и соблюдать процедуры для предупреждения проникновения вредоносного кода, его обнаружения и уничтожения;
- знать и соблюдать правила поведения в экстренных ситуациях, последовательность действий при ликвидации последствий аварий.
- *Управленческие меры обеспечения информационной безопасности.* Главной целью мер, предпринимаемых на управленческом уровне, является формирование программы работ в области информационной безопасности и обеспечение ее выполнения путем выделения необходимых ресурсов и осуществления регулярного контроля состояния дел. Основой этой программы является многоуровневая политика безопасности, отражающая комплексный подход организации к защите своих ресурсов и информационных активов

22.Модель матрицы доступов Харрисона-Рузо-Ульмана. Модель системы безопасности Белла-ЛаПуды

Рассматривая вопросы защиты информации в КС, мы уже использовали ранее понятие политики безопасности. Напомним, что под политикой безопасности понимается совокупность норм и правил, регламентирующих процесс обработки информации, выполнение которых обеспечивает защиту от определенного множества угроз и составляет необходимое (а иногда и достаточное) условие безопасности системы.

Для строгого и однозначного толкования норм и правил политики безопасности обычно дается ее формализованное описание в виде соответствующей модели. Основная цель такого описания – 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 пользователи получат доступ к информации и смогут осуществить с ней только санкционированные действия.

Все существующие в настоящее время модели безопасности основаны на следующих базовых представлениях.

1. Компьютерная система является совокупностью взаимодействующих сущностей – субъектов и объектов. Объекты можно интуитивно представлять в виде контейнеров, содержащих информацию, а субъектами считать выполняющиеся программы, которые воздействуют на объекты различными способами. При таком представлении безопасность обработки информации обеспечивается путем решения задачи управления доступом субъектов к объектам в соответствии с тем набором правил и ограничений, которые образуют политику безопасности. Считается, что система безопасна, если субъекты не имеют возможности нарушать правила политики безопасности. Таким образом, общим кодом для всех моделей является именно разделение множества сущностей, образующих систему, на множества субъектов и объектов.

2. Все взаимодействия в системе моделируются установлением отношений определенного типа между субъектами и объектами. Множество типов таких отношений определяется в виде набора операций, которые субъекты могут производить над объектами.

3. Все операции между субъектами и объектами, контролируемые монитором взаимодействий, либо запрещаются, либо разрешаются в соответствии с правилами политики безопасности.

4.Политика безопасности задается в виде правил, определяющих все взаимодействия между субъектами и объектами. Взаимодействия, приводящие к нарушению этих правил, пресекаются средствами контроля доступа и не могут быть осуществлены.

5. Совокупность множеств субъектов, объектов и отношений между ними (установившихся взаимодействий) определяет состояние системы. В этом пространстве состояний каждое состояние системы является либо безопасным, либо небезопасным в соответствии с принятым в модели критерием безопасности.

6. Основной элемент модели безопасности – это доказательство того, что система, находящаяся в безопасном состоянии, не может перейти в небезопасное состояние при соблюдении всех установленных правил и ограничений.

Среди моделей политики безопасности можно выделить три основных типа: дискреционные (произвольные), мандатные (нормативные) и ролевые. В основе этих моделей лежат, соответственно, дискреционное управление доступом (Discretionary Access Control – DAC), мандатное управление доступом (Mandatory Access Control – MAC) и ролевое управление доступом (Role-Based Access Control – RAC) .

4.1. Управление доступом

Под управлением доступом (УД) будем понимать ограничение возможностей использования ресурсов системы пользователями (процессами) в соответствии с правилами разграничения доступа.

Существует четыре основных способа разграничения доступа субъектов к совместно используемым объектам :

- физический – субъекты обращаются к физически различным объектам (однотипным устройствам, наборам данных на разных носителях и т. д.);
- временной – субъекты получают доступ к объекту в различные промежутки времени;

- логический – субъекты получают доступ к объектам в рамках единой операционной среды, но под контролем средств разграничения доступа;
- криптографический – все объекты хранятся в зашифрованном виде, права доступа определяются знания ключа для расшифрования объекта.

На практике основными способами разграничения доступа являются логический и криптографический. Рассмотрим логическое УД, которое может быть реализовано в соответствии с одной из трех формальных моделей УД (безопасности). В табл. 4.1 приведены модели и профили защиты УД.

Таблица 4.1

Модели и профили защиты УД

Модель УД	Профиль защиты	Класс безопасности по «Оранжевой книге»	Класс защищенности по классификации Гостехкомиссии РФ
Дискреционная	Контролируемый доступ	C2	5
Мандатная	Меточная защита	B1	4
Ролевая	Универсальная надстройка, применяемая с дискреционным и мандатным УД		

Модели УД

Дискреционное (произвольное, матричное, разграничительное) УД

Дискреционное УД (Discretionary Access Control – DAC) – разграничение доступа между поименованными субъектами и поименованными объектами. Оно основанное на задании владельцем объекта или другим полномочным лицом прав доступа других субъектов (пользователей) к этому объекту.

В рамках этой модели система обработки информации представляется в виде совокупности активных сущностей – субъектов (множество S – пользователи, процессы и т. д.), которые осуществляют доступ к информации, пассивных сущностей – объектов (множество O – файлы, каталоги, процессы и т. д.), содержащих защищаемую информацию, и конечного множества прав доступа $R = \{r_1, ..., r_n\}$, означающих полномочия на выполнение соответствующих действий (например, чтение (Read – R), запись (Write – W), выполнение (Execute – E), удаление (Delete – D), владение (Ownership – O) и т. д.).

Каждый объект объявляется собственностью соответствующего субъекта (владельца). При чем в конкретный момент времени у объекта может быть только один владелец, но с течением времени они могут меняться. Владелец имеет все права доступа к объекту и он определяет права доступа других субъектов к этому объекту.

Поведение системы моделируется с помощью понятия состояния. Пространство состояний системы образуется декартовым произведением множеств составляющих ее объектов, субъектов и прав – OxSxR. 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). Матрица доступ представлена табл. 4.2.

Таблица 4.2

Матрица доступа

		O_1	O_2	O_3	O_4	O_5
M =	S_1		R			
	S_2		RW	R		
	S_3					RW
	S_4	Own	Own	Own	Own	Own

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 $a(x_1, \ldots, x_k)$, состоящих из элементарных операций ($x_1, \ldots, x_k$ – параметры команды).

В классической модели допустимы следующие элементарные операции: создание нового субъекта или объекта, удаление существующего субъекта или объекта, добавление субъекту или удаление у субъекта права для объекта.

Применение любой элементарной операции к системе, находящейся в состоянии Q=(S,O,M) влечет за собой переход в другое состояние Q'=(S',O',M'), которое отличается от предыдущего состояния Q по крайней мере одним компонентом.

«Произвольность» этой модели УД состоит в том, что права субъекта по отношению к объекту могут быть изменены в любой момент времени произвольным образом.

Достоинства дискреционного УД:

- гибкость – позволяет независимо управлять правами для любой пары «субъект»–«объект»;
- не требует никаких сложных алгоритмов реализации.

Недостатки искреционного УД:

- 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 пользователю объекта в объект, доступный нарушителю, то формально никакое правило дискреционной политики безопасности не нарушается, но утечка информации происходит;
- сложный контроль за распространением прав доступа. Например, владелец файла передает все права или их часть на объект другому пользователю. Тот, в свою очередь, передает их третьему и т. д. В результате передачи прав доступ к объекту может получить некоторый субъект даже в том случае, если исходный владелец был против этого;
- в реальных системах процедуры по обслуживанию и поддержанию в адекватном состоянии матриц доступа оказываются весьма трудоемкими. Работа администратора защиты становится узким местом в работе систем, обладающих динамикой состава пользователей, программ, данных и т. п.

В зависимости от способа представления матрицы прав доступа в ИС различают несколько способов реализации произвольного УД.

Наиболее распространенными для операционных систем являются:

- «парольная» защита;
- списки прав доступа;
- списки полномочий субъектов;
- биты доступа.

«Парольная» защита осуществляется следующим образом: пользователь использует отдельный пароль для доступа к каждому объекту в системе. В большинстве реализации парольных систем существуют различные пароли для каждого объекта и для каждого типа доступа. Этот механизм был реализован в операционных системах IBM MVS и NOS.

Использование данного метода доставляет пользователю массу неудобств, т. к. запомнить пароли для каждого объекта и типа доступа невозможно, а хранить их в программах и файлах – ненадежно. Существенную проблему для применения парольной защиты представляет собой и необходимость периодической смены паролей.

Списки управления доступом (Access – ACT Control List). При реализации произвольного управление доступом с помощью ACL с каждым объектом ассоциируется список пользователей или групп пользователей с указанием их прав доступа к объекту (рис. 4.1). При принятии решения о доступе, соответствующий объекту доступа список проверяется на наличие прав, ассоциированных с индентификатором пользователя, запрашивающего доступ, или его группы.

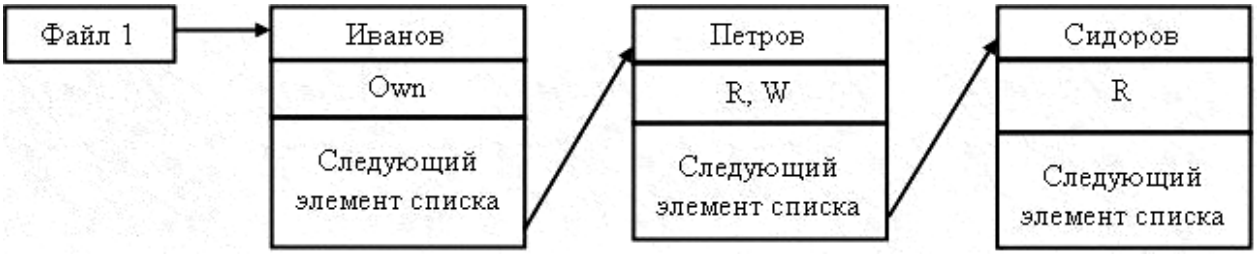


Рис. 4.1. Пример списка контроля доступа к файлам

Данный подход реализован ОС Novell Netware и Windows (точнее в NTFS – New Tchnology File System).

В NTFS 5.0 существует набор из 6 стандартных прав (в терминологии Windows – разрешений):

- полный доступ (включает все права);
- изменение;
- чтение и выполнение;
- список содержимого папки (только для папок);
- чтение;
- запись.

Разрешения перечислены по мере убывания возможностей по работе с объектом (рис. 4.2).

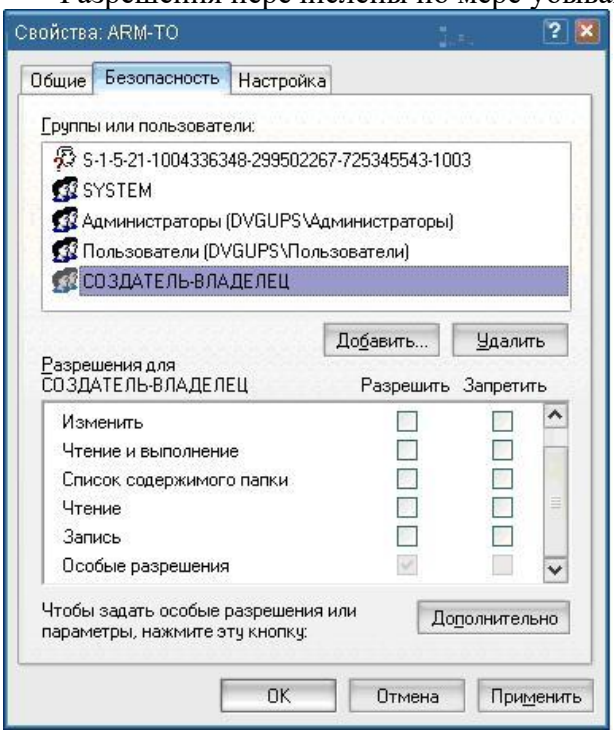


Рис. 4.2. Задание стандартных разрешений

При указании прав для объекта можно сразу выбрать несколько стандартных разрешений.

Каждое из стандартных разрешений в свою очередь представляется комбинацией более мелких, специальных прав, общее число которых равно 13 (рис. 4.3).

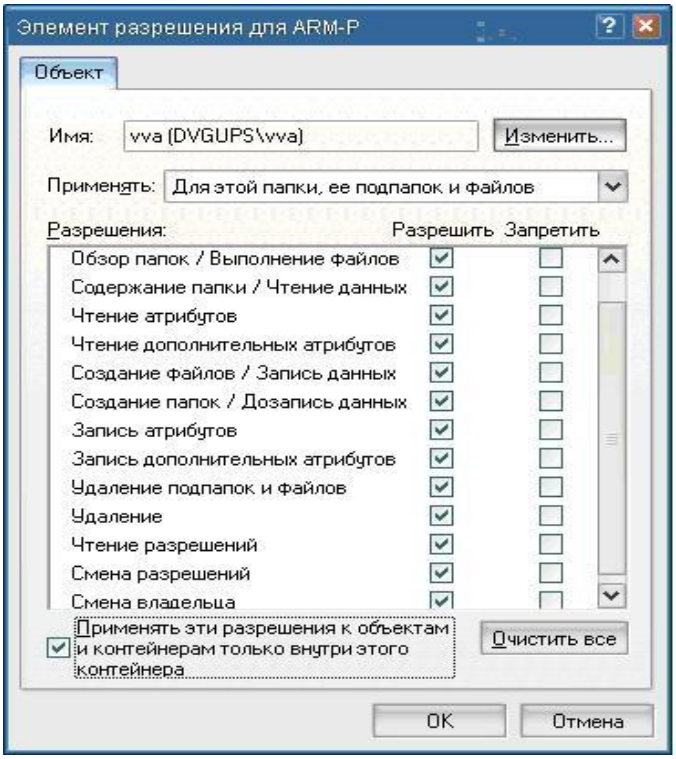


Рис. 4.3. Задание специальных разрешений

Кроме прав чтения, записи и т. п. объекта, они включают в себя также разрешения для работы с его атрибутами и с назначением прав доступа других субъектов по отношению к объекту. Менять стандартные или специальные разрешения может только владелец и субъекты, обладающий специальным правом «Смена разрешений». Следует отметить, что владельцем объекта является субъект его создавший, и в один и тот же момент времени у объекта может быть только один владелец. В тоже время, группе (учетной записи) «Администраторы» по умолчанию назначается право «Смена владельца». Хотя администратор может вступить в права владения, он не может передавать это право другим пользователям. Такое ограничение необходимо для того, чтобы администратор нес ответственность за свои действия.

Если стандартные комбинации (права) специальных прав, не удовлетворяет потребностям защиты, то можно создать произвольный набор специальных прав. В этом случае, такой набор в списке стандартных разрешений будет называться «Особые разрешения».

Кроме применения ACL для разрешения доступа к файлам и папкам, в Windows они также используются для разрешения доступа к разделам реестра, службам, принтерам, объектам Active Directory и т. п.

Следует отметить, что сама по себе NTFS не шифрует данные на диске, поэтому они могут читаться в обход ОС и, следовательно, в обход установленных прав доступа. Существует драйвера NTFS для MS-DOS и Linux. Таким образом при наличии физического доступа к компьютеру и возможности его перезагрузить в другой ОС, злоумышленник получит доступ ко всем данным на диске безо всякой аутентификации и проверки прав. Избежать такой угрозы безопасности можно за счет применения EFS (Encrypting File System, шифрующей файловой системы), работающей только поверх NTFS 5.0, или за счет применения других криптографических средств.

Основным недостатком применения ACL являются большие временные затраты на обработку списков.

Списки полномочий субъектов (списки возможностей). В данной модели с каждым субъектом ассоциируется список прав доступа для всех объектов, к которым он имеет доступ. Пример списка полномочий субъектов приведен на рис. 4.4.

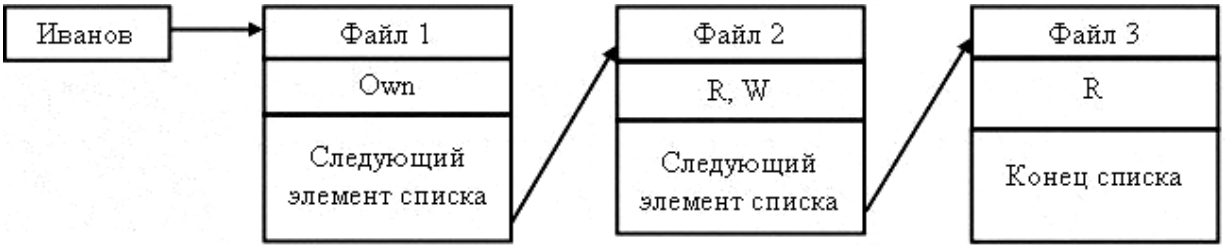


Рис. 4.4. Пример списка полномочий субъектов

Биты защиты. Данный подход реализован в большинстве защищенных ОС, ведущих свое происхождение от UNIX. Вместо списка пользователей, которым разрешен доступ к объекту, с объектом связываются биты защиты.

Владелец			Группа			Все пользователи		
Чтение	Запись	Выполнение	Чтение	Запись	Выполнение	Чтение	Запись	Выполнение
1	2	3	4	5	6	7	8	9

Рис. 4.5. Биты защиты UNIX

В ОС UNIX биты защиты указывают, права доступа для трех категорий пользователей: все пользователи (world), члены группы владельца (group) и владелец объекта (owner). При этом биты защиты может изменять только владелец объекта и администратор.

Недостатком использования механизма битов защиты является неполная реализация произвольного УД, т.к. доступ к объекту нельзя разрешить или запретить для отдельных пользователей. Сложные комбинации прав доступа могут быть установлены путем создания индивидуальных групп для каждого файла и копированием файлов, но это очень сложно организовать для большого количества пользователей.

Мандатное (принудительное) УД

Мандатное УД (Mandatory Access Control – MAC) – 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. Оно основано на сопоставлении атрибутов безопасности субъекта (уровня допуска пользователя) и объекта (грифа секретности информации).

Мандатная модель управления доступом основана на правилах секретного документооборота, принятых в государственных и правительственных учреждениях многих стран. Основным положением данной ПБ, взятым из реальной жизни, является назначение всем участникам процесса обработки защищаемой информации и документам, в которых она содержится, специальной метки, получившей название уровень безопасности (метка безопасности). Метка субъекта описывает его благонадежность, а метка объекта – степень закрытости содержащейся в нем информации. Уровни секретности, поддерживаемые системой, образуют множество, упорядоченное с помощью отношения доминирования. Такое множество может выглядеть следующим образом: сов. секретно, секретно, конфиденциально, несекретно и т. д.

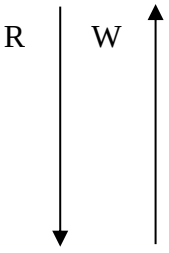
Система в мандатной модели представляется в виде множеств субъектов S, объектов O, решетки уровней безопасности L и матрицы доступа M.

С помощью решетки уровней безопасности (табл. 4.3) задается соотношение между уровнями безопасности, субъектами и объектами.

Таблица 4.3

Решетка уровней безопасности

Уровень безопасности	Субъекты	Объекты
Совершенно секретно	S_1, S_2	O_5
Секретно	S_3	O_1, O_2
Конфиденциально	S_4	O_4
Несекретно	S_5, S_6	O_3, O_6



В данной модели набор прав ограничен двумя: read (чтение) и write (запись). При этом контроль доступа осуществляется в зависимости от уровней безопасности взаимодействующих сторон на основании двух простых правил.

1. Уполномоченное лицо (субъект) имеет право читать только те документы, уровень безопасности которых не превышает его собственный уровень безопасности. Данное правило обеспечивает защиту информации, обрабатываемой более доверенными (высокоуровневыми) лицами, от доступа со стороны менее доверенных (низкоуровневых).

2. Уполномоченное лицо (субъект) имеет право заносить информацию только в те документы, уровень безопасности которых не ниже его собственного уровня безопасности. Это правило предотвращает утечку информации (сознательную или несознательную) со стороны высокоуровневых участников процесса обработки информации к низкоуровневым.

Матрица доступа для приведенного в табл. 4.4 примера выглядит следующим образом.

Таблица 4.4

Матрица доступа

	O_1	O_2	O_3	O_4	O_5	O_6
S_1	R	R	R	R	RW	R
S_2	R	R	R	R	RW	R
S_3	RW	RW	R	R	W	R
S_4	W	W	R	RW	W	R
S_5	W	W	RW	W	W	RW
S_6	W	W	RW	W	W	RW

Таким образом можно констатировать следующий факт – мандатное УД является частным случаем дискреционного УД, с более строгими правилами разграничения доступа. Использование столь жесткого подхода, не позволяющего осуществлять гибкое управление доступом, объясняется тем, что в мандатной модели контролируются не операции, осуществляемые субъектом над объектом, а потоки информации, которые могут быть только двух видов: либо от субъекта к объекту (запись), либо от объекта к субъекту (чтение). Остальные права моделируются через эти две базовые операции.

Достоинства мандатного УД:

- экономия памяти, так как элементы матрицы доступа не хранятся, а динамически вычисляются при попытке доступа для конкретной пары субъект-объект на основе их меток;
- удобство корректировки базы данных защиты, то есть модификации меток;
- принудительное УД хорошо согласуется с работой государственных, правительственных и военных организаций, так как переносит общепринятые и хорошо отработанные принципы обращения с бумажными секретами на современную основу работы с документами.

Недостатки мандатного УД:

- затруднено задание прав доступа конкретного субъекта к конкретному объекту;
- каждый субъект и объект должен быть помечен и при любых операциях с данными метки должны оставаться правильными. В особенности это относится к экспорту и импорту данных. Например, печатный документ должен открываться заголовком, содержащим текстовое и/или графическое представление метки безопасности. Аналогично, при передаче файла по каналу связи должна передаваться и ассоциированная с ним метка, причем в таком виде, чтобы удаленная система могла ее правильно трактовать, несмотря на возможные различия в уровнях секретности и наборе категорий.

Наиболее известными мандатными моделями являются ММ Белла-Лападула (Дэвид Белл и Леонардо ЛаПадула), ММ Биба, решетчатая модель Д. Деннинга, ММ совместного доступа с уполномоченными субъектами и т. д.

Ролевое УД

Ролевое УД (Role-Based Access Control – RAC) – универсальная надстройка (каркас), применяемая с дискреционным и мандатным УД и предназначенная для упрощения функций администрирования систем с большим количеством субъектов и объектов.

Суть ролевого УД состоит в том, что между пользователями и их правами доступа к объектам появляются промежуточные сущности – роли. Для каждого пользователя одновременно могут быть активными несколько ролей, каждая из которых дает ему определенные права доступа к объекту и, наоборот, несколько пользователей может выступать в одной роли по отношению к одному объекту. Между ролями могут быть установлены связи, аналогичные отношению наследования в ООП. Таким образом может быть построена иерархия ролей, используя которую можно существенно сократить количество контролируемых (администрируемых) связей. Пример применения ролевого УД приведен на рис. 4.5.

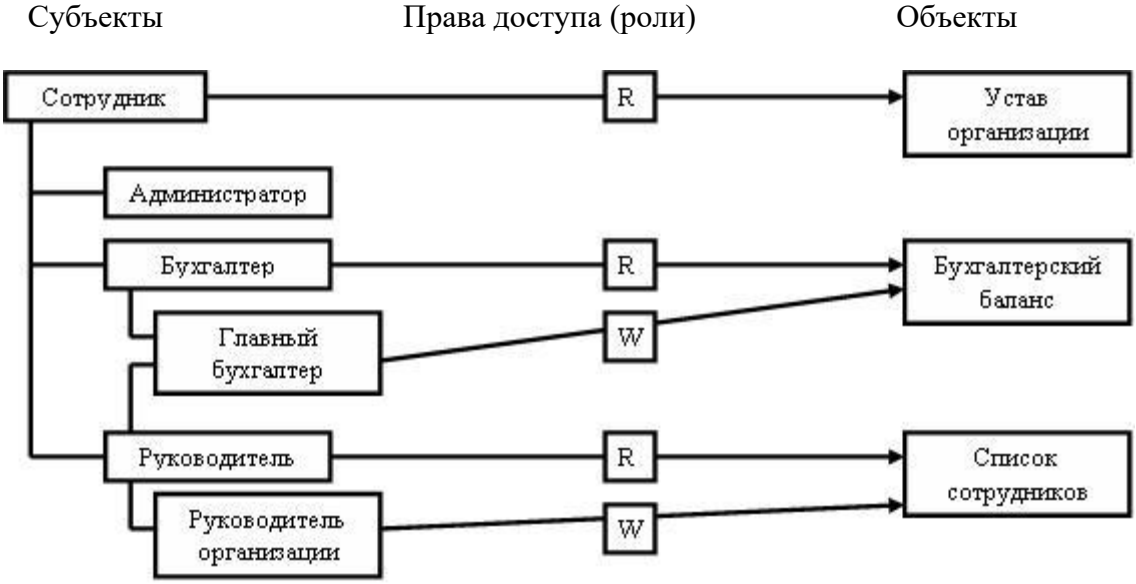


Рис. 4.5. Пример применения ролевого УД

Основополагающими принципами организации ролевого УД являются принципы минимизации привилегий и разделения обязанностей.

Принцип минимизации привилегий гласит о том, что для роли целесообразно назначить только такие права, которые необходимы для выполнения ее служебных обязанностей. В связи с этим, дерево иерархии строят, наращивая (расширяя) права ролей. При построении этой иерархии допускается множественное наследование ролей (например, роль «Главный бухгалтер» наследует права от непосредственных родителей «Бухгалтер» и «Руководитель»).

Разделения обязанностей делится на два вида:

- статическое;
- динамическое.

Статическое разделение обязанностей налагает ограничения на приписывание ролей пользователям. В простейшем случае членство в некоторой роли запрещает приписывание пользователя определенному множеству других ролей. Например, может существовать пять бухгалтерских ролей, но политика УД допускает членство не более чем в двух таких ролях. При наличии наследования ролей ограничение приобретает более сложный вид, но суть остается простой: при проверке членства в ролях нужно учитывать приписывание пользователей ролям-наследницам.

Динамическое разделение обязанностей отличается от статического только тем, что рассматриваются роли, одновременно активные (быть может, в разных сеансах) для данного пользователя (а не те, которым пользователь статически приписан). Например, один пользователь может играть роль и кассира, и контролера, но не одновременно. Чтобы стать контролером, он должен сначала закрыть кассу.

Существуют модель политики безопасности АДЕПТ – 50, формальные модели монитора безопасности объектов: дискреционные модели Харрисона-Руззо-Ульмана и Типизированная матрица

доступа, мандатная модель Белла-Лападулы, модель ролевой политики безопасности, модель политики безопасности распределенной компьютерной системы.

В качестве классических примеров моделей этих типов можно назвать дискреционную модель Харрисона-Руззо- Ульмана (модель HRU) и мандатную модель Белла-Лападула (модель БЛ).

В рамках модели Белла-Лападула доказывается важное утверждение, указывающее на принципиальное отличие систем, реализующих мандатную защиту, от систем с дискреционной защитой: если начальное состояние системы безопасно, и все переходы системы из состояния в состояние не нарушают ограничений, сформулированных политикой безопасности, то любое состояние системы безопасно.

4.2. Модель политики безопасности адепт – 50

Модель АДЕПТ относится к одной из первых дискреционных моделей политики безопасности МБО.

В модели рассматриваются:

- множество объектов компьютерной системы $O=[o_j]$, $j = 1,\ldots n$
- множество субъектов компьютерной системы $S=[s_j]$, $i=1,\ldots m$
- множество прав доступа субъектов к объектам $R=[r_g]$, $k = 1, \ldots k$
- множество, содержащее перечень предметных областей, к которым относятся выполняемые в компьютерной системе процессы $E=[e_l]$, $l = 1,\ldots q$.
- множество, содержащее категории безопасности субъектов и объектов $D=[d_x]$, $x = 1,\ldots v$.

Субъекты в компьютерной системе характеризуются:

- предметной областью, к которой они принадлежат
- правами, которыми они наделяются
- категорией безопасности, которая им назначается.

Объекты компьютерной системы характеризуются:

- предметной областью, к которой они принадлежат
- категорией безопасности, которая им назначается.

Критерий безопасности

Субъект наследует категорию безопасности объекта, с которым он связан (ассоциирован).

Описание политики безопасности

Если в компьютерной системе инициализируется поток Stream $(s_j, o_j) \rightarrow o_i$, то

1) субъект s_j должен принадлежать множеству S

$s_j \in S;$

2) объект o_i должен принадлежать множеству O

$o_i \in O;$

3) субъект s_j получает право доступа r_{sj} к объекту o_i, если право доступа субъекта s_j (r_{sj})принадлежит множеству его прав доступа к объекту o_i

$r_{sj} \in R_{oi};$

4) предметная область субъекта sj принадлежит предметной области объекта oi

$e_{sj} \in e_{oi};$

5) категория безопасности субъекта sj больше или равна категории безопасности объекта o_i

$d_{oi} \in d_{sj}$.

4.3. Дискреционная модель харрисона-руззо-ульмана

Формальная модель Харрисона- Руззо-Ульмана – это классическая дискреционная модель, которая реализует произвольное управление доступом субъектов к объектам и осуществляет контроль за распределением прав доступа.

В модели рассматриваются:

- конечное множество объектов компьютерной системы $O=[o_j]$, $1,\ldots n$;
- конечное множество субъектов компьютерной системы $S=[s_j]$, $1,\ldots m$.

Считается, что все субъекты системы одновременно являются и ее объектами.

Конечное множество прав доступа $R=[r_g]$, $1, \ldots k$.

Матрица прав доступа, содержащая права доступа субъектов к объектам $A=[a_{ij}]$, $i=1, \ldots m$, $j=1,\ldots n+m$, причем элемент матрицы a_{ij} рассматривается как подмножество множества R.

Каждый элемент матрицы a_{ij} содержит права доступа субъекта s_i к объекту o_j.

Конечное множество команд C=[c_z (аргументы)], z=1, l, аргументами команд служат идентификаторы объектов и субъектов.

Каждая команда включает условия выполнения команды и элементарные операции, которые могут быть выполнены над субъектами и объектами компьютерной системы и имеют следующую структуру:

Command c_z (аргументы)

Условия выполнения команды

Элементарные операции

End.

Поведение системы моделируется с помощью понятия состояние. Состояние системы определяется:

- конечным множеством субъектов (S);
- конечным множеством объектов (O), считается, что все субъекты системы одновременно являются и ее объектами ($S \subset O$);
- матрицей прав доступа (A).

Если система находится в состоянии Q, то выполнение элементарной операции переводит ее в некоторое другое состояние Q' ,которое отличается от предыдущего состояния хотя бы одним компонентом.

В модели Харрисона-Руззо-Ульмана определены следующие элементарные операции, при выполнении которых система может перейти из одного состояния в другое.

Добавление субъекту s_i права r_g для объекта o_j.

Enter r_g into a_{ij}

При выполнении этой операции множество субъектов и множество объектов не изменяются. Подмножеству прав доступа добавляется новое право, остальные подмножества не изменяются. Если право уже содержится в подмножестве, то это подмножество также не изменяется.

Операция добавления права называется монотонной, поскольку она только добавляет права в матрицу доступа, но ничего не изменяет.

Удаление у субъекта s_i права r_g для объекта o_j.

Delete r_g from a_{ij}

При выполнении этой операции множество субъектов и множество объектов не изменяются. Из подмножества прав доступа удаляется право r_g, остальные подмножества не изменяются. Если удаляемое право не содержится в подмножестве, то это подмножество также не изменяется.

Создание нового субъекта s_i.

Create subject s_i

При выполнении этой операции изменяются следующие состояния системы:

- к множеству объектов системы добавляется новый объект;
- к множеству субъектов системы добавляется новый субъект;
- множество прав доступа субъекта s_i к объектам системы становится пустым;
- множество прав доступа всех субъектов системы к объекту o_j становится пустым.

Остальные подмножества матрицы доступа не изменяются.

Удаление существующего субъекта s_i

Destroy subject s_i

При выполнении этой операции изменяются следующие состояния системы

- из множества объектов системы удаляется объект o_j ;
- из множества субъектов системы удаляется субъект s_i ;
- множество прав доступа субъекта s_i к объектам системы становится пустым;
- права доступа всех субъектов системы к объекту o_j становится пустым;

Остальные подмножества матрицы доступа не изменяются.

Создание в системе нового объекта o_j

Create object o_j

При выполнении этой операции изменяются следующие состояния системы:

- к множеству объектов системы добавляется новый объект o_j ;

- множество субъектов системы не изменяется ;
- множество прав доступа всех субъектов к новому объекту системы становится пустым .

Остальные подмножества матрицы доступа не изменяются.

Удаление существующего объекта o_j из системы

Destroy object o_j

При выполнении этой операции изменяются следующие состояния системы:

- из множества объектов системы удаляется объект o_j ;
- множество субъектов системы не изменяется ;
- права доступа всех субъектов системы к объекту o_j становится пустым.

Остальные подмножества матрицы доступа не изменяются.

Описание модели

Поведение системы во времени моделируется последовательностью состояний Q_1, Q_2,…Q_n, в которой каждое последующее состояние является результатом применения команды из множества C к предыдущему состоянию.

Критерий безопасности

Начальное состояние системы считается безопасным относительно права доступа r_g, если не существует применимой к Q_0 последовательности команд, в результате выполнения которых право доступа r_g будет приобретено субъектом s_i для объекта o_j, если это право не принадлежит подмножеству a_{ij} в состоянии Q_0.

Харрисон-Руззо-Ульман доказали, что в общем случае не существует алгоритма, позволяющего решить является ли конфигурация системы, соответствующая ее начальному состоянию $Q_0 = (S_0, O_0, A_0)$ безопасной.

Указанная задача может быть разрешена в одном из следующих случаев:

- команды cz (аргументы), z = 1, 2,… l являются монооперационными, т.е. состоят только из одной операции;
- команды cz (аргументы), z = 1, 2,… l являются одноусловными и монотонными, т. е. содержат не более одного условия и не содержат операций Destroy и Delete;
- команды cz (аргументы), z = 1, 2,… l не содержат команды Create.

Вывод: 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 контролем за распространением прав доступа во всех современных системах.

4.4. Типизированная матрица доступа

Дискреционная модель Type Access Matrix (TAM) – типизированная матрица доступа – является развитием модели Харрисона- Руззо-Ульмана. Модель TAM дополняет модель Харрисона- Руззо-Ульмана концепцией типов, что позволяет смягчить те условия, для которых возможно доказательство безопасности системы.

В модели TAM рассматриваются:

- конечное множество объектов компьютерной системы O = [o_j], 1,…n;
- конечное множество субъектов компьютерной системы S = [s_i], 1, …m.

Считается, что все субъекты системы одновременно являются и ее объектами.

- конечное множество прав доступа R = [rg], 1, …k.

Матрица прав доступа, содержащая права доступа субъектов к объектам A = [a_{ij}], I = 1, …m, j = 1,…n+m, причем элемент матрицы a_{ij} рассматривается как подмножество множества R.

Каждый элемент матрицы a_{ij} содержит права доступа субъекта s_i к объекту o_j.

Множество типов, которые могут быть поставлены в соответствие объектам и субъектам системы T = [t_b], b = 1,…w.

Конечное множество команд C=[c_z (аргументы с указанием типов)], z=1,...I, включающих условия выполнения команд и их интерпретацию в терминах элементарных операций. Элементарные операции TAM отличаются от элементарных операций дискреционной модели Харрисона- Руззо-Ульмана использованием типизированных аргументов.

Структура команды

Command c_z (аргументы и их типы)

Условия выполнения команды
Элементарные операции
End .

Смысл элементарных операций совпадает со смыслом аналогичных операций, используемых в модели Харрисона- Руззо-Ульмана с точностью до использования типов.

Поведение системы моделируется с помощью понятия состояние. Состояние системы определяется:

- конечным множеством субъектов (S);
- конечным множеством объектов (O), считается, что все субъекты системы одновременно являются и ее объектами (S□O);
- матрицей прав доступа (A)

и описывается тройкой Q (S, O, A).

Выполнение элементарных операций переводит систему, находящуюся в состоянии Q в другое состояние Q'.

В модели TAM определены следующие элементарные операции:

Добавление субъекту s_i права r_g для объекта o_j.

Enter r_g into a_{ij}

Удаление у субъекта s_i права r_g для объекта o_j.

Delete r_g from a_{ij}

Выполнение этих элементарных операций приводит к тем же изменениям в состоянии системы, как и их выполнение в модели Харрисона-Руззо-Ульмана.

Типы объектов и субъектов при выполнении этих операций остаются без изменения.

Создание нового субъекта s_i с типом t_{sx}

Create subject s_i of type t_{sx}

Выполнение этой элементарной операции приводит к тем же изменениям в состоянии системы, как и ее выполнение в модели Харрисона-Руззо-Ульмана.

Субъекту s_x и объекту o_x ставится в соответствие тип t_{sx} из множества T, типы остальных субъектов и объектов остаются без изменения.

Удаление существующего субъекта s_i с типом t_{sx}

Destroy subject s_x of type t_{sx}

Выполнение этой элементарной операций приводит к тем же изменениям в состоянии системы, как и ее выполнение в модели Харрисона-Руззо-Ульмана.

Типы субъекта s_x и соответствующего ему объекта o_x становятся неопределенными, типы остальных объектов и субъектов остаются без изменения.

Создание в системе нового объекта o_j типом t_{oy}

Create object o_y of type t_{oy}

Выполнение этой элементарной операции приводит к тем же изменениям в состоянии системы, как и их выполнение в модели Харрисона- Руззо-Ульмана.

Объекту o_y ставится в соответствие тип t_{oy} из множества T, типы остальных объектов и субъектов при выполнении этой операций остаются без изменения.

Удаление существующего объекта o_y с типом t_{oy}

Destroy object o_y of type t_{oy}

Выполнение этой элементарной операции приводит к тем же изменениям в состоянии системы, как и их выполнение в модели Харрисона- Руззо-Ульмана.

Тип объекта o_y становится неопределенным, типы остальных объектов и субъектов при выполнении этой операций остаются без изменения.

Таким образом, TAM является обобщением модели Харрисона-Руззо-Ульмана, которую можно рассматривать как частный случай TAM с одним единственным типом, к которому относятся все объекты и субъекты.

Строгий контроль соответствия типов в модели TAM позволяет смягчить требование одноусловности, заменив его ограничением на типы аргументов команд, при выполнении которых происходит создание новых объектов и субъектов.

Для регулирования этого ограничения вводятся понятия родительского и дочернего типов. Тип аргументов команды

C_z (s_j : t_{sj}, o_i : t_{oi}, s_x :t_{sx}, o_y : t_{oy})
считается дочерним, если в этой команде используются элементарные операции вида:

Create Subject s_j of type t_{sj} ;

Create Object o_i of type t_{oi} ,

т. е. элементарные операции создания субъекта или объекта типа, который указан для этих субъекта и объекта в аргументах команды. В противном случае тип аргументов в команде считается родительским. При этом предполагается, что объект или субъект не могут быть созданы, если для них отсутствует родительский тип.

В рассматриваемой команде:

t_{sj} и t_{oi} считаются дочерними типами;

t_{sx} и t_{oy} считаются родительскими типами.

Другими словами, для того, чтобы создать объект o_i типа t_{oi} или субъект s_j типа t_{sj} в системе должен быть объект o_y типа t_{oy} или субъект s_x типа t_{sx}.

Следует отметить, что в одной команде тип может быть одновременно и родительски и дочерним. Например,

Command_1 (s_1 : t_1, s_2 : t_1)

Create Subject s_2 of type t_1

End.

В этой команде тип t_1 считается родительским относительно субъекта s_1 и дочерним относительно субъекта s_2.

Связи между родительскими и дочерними типами описываются с помощью графа создания, определяющего отношение наследственности.

Граф создания представляет собой направленный граф с множеством вершин T, в котором ребро от t_i к t_j существует тогда и только тогда, когда в системе имеется команда создания субъекта или объекта, в которой t_i является родительским типом, а t_j– дочерним.

Этот граф для каждого типа позволяет определить:

- объекты и субъекты, каких типов должны существовать в системе, чтобы в ней мог появиться объект или субъект заданного типа;
- объекты и субъекты, каких типов могут быть порождены при участии объектов и субъектов заданного типа.

В теории компьютерной безопасности рассматривается модифицированная ТАМ – МТАМ (монотонная типизированная матрица доступа), к которой отсутствуют немонотонные элементарные операции: Delete, Destroy Subject, Destroy Object.

Реализация МТАМ называется ациклической, если ее граф создания не содержит циклов, в противном случае реализация МТАМ называется циклической.

Доказано, что критерий безопасности, предложенный Харрисоном-Руззо-Ульманом, разрешим для ациклических реализаций ТАМ, и что требование одноусловности команд можно заменить требованием ацикличности графа создания. Смысл этой замены состоит в том, что последовательность состояний системы должна определяться некоторым маршрутом соответственно графу создания, т. к. не возможно появление субъектов и объектов дочерних типов, если в системе отсутствуют родительские типы, которые должны участвовать в их создании. Отсутствие циклов на графе создания позволяет избежать зацикливания при доказательстве критерия безопасности, т. к. количество путей на графе без циклов является ограниченным. Это означает, что поведение системы становится предсказуемым, поскольку в любом состоянии можно определить субъекты и объекты каких типов могут появиться в системе, а каких – нет.

Однако доказано, что сложность проверки критерия безопасности для МТАМ с ростом в системе количества субъектов и объектов значительно увеличивается. Время на решение этой задачи растет в степенной зависимости от количества субъектов и объектов. Этот недостаток может быть преодолен с помощью тернарной ТАМ, в которой команды могут иметь не более трех аргументов.

Тернарная МТАМ является монотонной версией тернарная ТАМ. Для тернарной МТАМ доказательство безопасности системы значительно упрощается, поскольку запись условий в команде ограничивается небольшим фрагментом матрицы доступа.

Доказано, что при ациклической реализации МТАМ время, затрачиваемое на проверку критерия безопасности, растет полиноминально в зависимости от размерности начальной матрицы доступа.

Вывод: введение строгого контроля типов в дискреционную модель Харрисона-Руззо-Ульмана позволило доказать критерий безопасности компьютерных систем для более приемлемых ограничений, что расширило область ее применения.

4.5. Мандатная модель Белла-Лападулы

Модель Белла-Лападулы была предложена в 1975 году. Возможность ее использования в качестве формальной модели отмечена в критерии TCSES ("Оранжевая книга").

Мандатная модель Белла – Лападулы основана на правилах секретного документооборота, которые приняты в государственных и правительственных учреждениях большинства стран. Согласно этим правилам всем участникам процесса обработки критичной информации и документам, в которых она содержится, присваивается специальная метка, которая называется уровнем безопасности.

Мандатное управление доступом подразумевает, что:

- задан линейно упорядоченный набор меток секретности (например, секретно, совершенно секретно и т. д.);
- каждому объекту системы присвоена метка секретности, определяющая ценность содержащейся в нем информации, т. е. его уровень секретности в КС;
- каждому субъекту системы присвоена метка секретности, определяющая уровень доверия к нему в КС или, иначе, его уровень доступа.

Все уровни безопасности упорядочиваются с помощью установленного отношения доминирования. Контроль доступа к документам осуществляется в зависимости от уровней безопасности на основании двух простых правил:

1. Уполномоченное лицо имеет право читать только те документы, уровень безопасности которых не превышает его собственный уровень безопасности. Уровень безопасности уполномоченного лица должен доминировать над уровнем безопасности читаемого им документа.

Это правило обеспечивает защиту информации, обрабатываемую высокоуровневым лицом, от доступа со стороны низкоуровневых лиц.

2. Уполномоченное лицо может записывать информацию только в документы, уровень безопасности которых не ниже его собственного уровня безопасности. Уровень безопасности документа должен доминировать над уровнем безопасности уполномоченного лица.

Это правило предотвращает утечку информации со стороны высокоуровневых участников процесса обработки документов к низкоуровневым.

В формальной модели Белла – Лападулы рассматриваются:

- конечное множество объектов компьютерной системы $O=[o_j]$, 1,…n;
- конечное множество субъектов компьютерной системы $S=[s_i]$, 1, …m, считается, что все субъекты системы одновременно являются и ее объектами ($S \subset O$).;
- права доступа read и write.

Матрица прав доступа, содержащая права доступа субъектов к объектам $A=[a_{ij}]$, i=1, …m, j=1,…n+m.

Множество запросов на выполнение потоков типа read или write $R=[r_g]$, 1, …k.

Функция уровня безопасности F, которая ставит в соответствие каждому объекту и субъекту системы определенный уровень безопасности, принадлежащий множеству уровней безопасности D, на котором определена решетка.

Функция перехода T, которая при выполнении запроса на запись или чтение, переводит систему из состояния Q в состояние Q'.

Состояние системы характеризуется состоянием матрицы A, содержащей права доступа, и функцией уровня безопасности F. Множество состояний системы представляется в виде упорядоченных пар (F, A).

Начальное состояние системы обозначается как Q_0. Выполнение запроса r_0 из множества R переводит систему в состояние Q_1 с помощью функции перехода T. Система, находящаяся в состоянии Q_1, при получении следующего запроса r_1 из множества R переходит в следующее состояние Q_2 и т. д.

Состояние Q_k достижимо тогда и только тогда, когда существует последовательность

$(Q_0, r_0), (Q_1, r_1), \dots (Q_{k-1}, r_{k-1})$

такая, что $T (Q_{k-1}, r_{k-1}) = Q_k$.

Считается, что для любой системы состояние Q_0 тривиально достижимо.

Следует отметить, что в мандатных моделях контролируются не операции, которые выполняются субъектами над объектами, а потоки информации. Потоки типа:

$Stream(s_i, o_i) \rightarrow o_j$ (поток типа чтение),

$Stream(s_i, o_j) \rightarrow o_i$ (поток типа запись).

Решетка уровней безопасности

Решетка уровней безопасности – это формальная алгебра, использование которой позволяет упорядочить потоки информации в компьютерной системе.

Решетка уровней безопасности представлена

- множеством уровней безопасности D;
- оператором отношения $\leq$;
- оператором, позволяющим определить наименьшую верхнюю границу для пары уровней безопасности;
- оператором, позволяющим определить наибольшую нижнюю границу для пары уровней безопасности.

Смысл этих операций заключается в том, что для каждой пары элементов множества D всегда можно указать единственный элемент, ограничивающий ее сверху или снизу таким образом, что между ними и этим элементом не будет других элементов.

Множество уровней безопасности может быть представлено как целыми числам, так и более сложными составными элементами. Например, в государственных организациях в качестве уровней безопасности используются комбинации, состоящие из уровня безопасности, представленного целым числом, и набора категорий из некоторого множества (1.секретно, 1.совершенно секретно и т.п.)

Пусть имеется множество субъектов S и множество объектов O. В модели Белла-Лападулы каждому субъекту и объекту системы из множества D функцией уровня безопасности F назначается соответствующий уровень безопасности.

Естественно, что некоторые субъекты и объекты могут иметь один и тот же уровень безопасности. Подмножества, в которых субъекты и объекты имеют одинаковый уровень безопасности, называются классами. Пусть имеются два класса X и Y. Рассмотрим применение основных положений теории решеток применительно к этим классам.

В теории решеток рассматривается отношение порядка $\leq$, использование этого отношения в теории компьютерной безопасности позволяет установить направление потоков информации.

Отношение порядка обладает следующими свойствами:

- рефлексивности;
- антисимметричности;
- транзитивности.

Вывод: мандатная модель управляет доступом неявным образом – с помощью назначения всем сущностям системы уровней безопасности, которые определяют все допустимые взаимодействия между ними.

4.5.1. Классическая мандатная модель Белла-Лападулы

В классической мандатной модели Белла – Лападулы состояния системы делятся на:

- безопасные, в которых информационные потоки не противоречат установленным в модели правилам;
- небезопасные, в которых эти правила нарушаются, и происходит утечка информации.

Белл и Лападула предложили следующее определение безопасного состояния:

1. Состояние (F, A) называется безопасным по чтению (или просто безопасным) тогда и только тогда, когда для каждого субъекта si, который реализует в этом состоянии поток типа read к объекту oj, уровень безопасности субъекта si доминирует над уровнем безопасности объекта oj

$\forall s_i \in S, \forall o_j \in O, read \in a_{ij} \rightarrow F(s_i) \geq F(o_j), \quad i = 1,\ldots m, j = 1,\ldots,n$

2. Состояние (F, A) называется безопасным по записи (или *- безопасным) тогда и только тогда, когда для каждого субъекта si, который реализует в этом состоянии поток типа write к объекту oj, уровень безопасности объекта oj доминирует над уровнем безопасности субъекта si

$\forall s_i \in S, \forall o_j \in O, write \in a_{ij} \rightarrow F(o_j) \geq F(s_i), \quad i = 1,\ldots m, j = 1,\ldots,n$

3. Состояние системы безопасно тогда и только тогда, когда оно безопасно и по чтению и по записи.

В соответствии с определением безопасного состояния критерий безопасности системы формулируется следующим образом.

Система (Q_0, R, T) безопасна тогда и только тогда, когда ее начальное состояние Q_0 безопасно и все состояния, достижимые из Q_0 в результате применения конечной последовательности запросов из R безопасны.

Белл и Лападула доказали теорему, формально определяющую безопасность системы при соблюдении необходимых условий. Эта теорема называется Основной теоремой безопасности.

Основная теорема безопасности

Система (Q_0, R, T) безопасна тогда и только тогда, когда

а) начальное состояние системы Q_0 безопасно

б) для любого состояния $Q\not\in$, достижимого из Q_0 в результате выполнения конечной последовательности запросов из R таких, что

$T(Q, r) = Q'$; $Q = (F, A)$; $Q' = (F', A')$

Для каждого $s_i \in S$ и $O_j \in O$ ($i = 1, \ldots, m, j = 1,\ldots, n$) выполняются следующие условия:

1) если $read \in a_{ij}' \wedge read \notin a_{ij}$, то $F' (s_i) \geq F' (o_j)$
2) если $read \in a_{ij} \wedge F' (s_i) < F' (o_j)$, то $read \notin a_{ij}'$
3) если $write \in a_{ij}' \wedge write \notin a_{ij}$, то $F' (o_j) \geq F' (s_i)$
4) если $write \in a_{ij} \wedge F' (o_j) < F' (s_i)$, то $write \in a_{ij}'$ □

Доказательство

Теорема утверждает, что система с безопасным начальным состоянием Q_0 является безопасной тогда и только тогда, когда при любом переходе системы из одного состояния в другое не возникает никаких новых и не сохраняется никаких старых отношений доступа (потоков), которые будут небезопасны по отношению к функции уровня безопасности нового состояния.

Формально эта теорема определяет все необходимые и достаточные условия, которые должны быть выполнены для того, чтобы система, начиная свою работу в безопасном состоянии, никогда не достигла небезопасного состояния.

Безопасная функция перехода. Теорема безопасности Мак – Лина

Недостаток основной теоремы безопасности Белла–Лападулы состоит в том, что ограничения, накладываемые теоремой на функцию перехода, совпадают с критериями безопасности состояния системы. Поэтому данная теорема является избыточной по отношению к определению безопасного состояния.

Из теоремы так же следует, что все состояния, в которые может перейти система из безопасного состояния, при определенных условиях безопасны. Но при этом ничего не говорится о том, что могут ли в процессе перехода изменятся уровни безопасности субъектов и объектов.

Если в процессе перехода системы из одного состояния в другое уровни безопасности субъектов и объектов могут изменяться, то это может привести к потере свойств безопасности системы.

Действительно можно представить гипотетическую систему (в литературе оно получила название Z – системы), в которой при попытке субъекта с низким уровнем безопасности прочитать информацию из объекта, имеющего более высокий уровень безопасности, будет происходить понижение уровня безопасности объекта до уровня безопасности субъекта и осуществляться чтение. В этом случае функция перехода Z – системы удовлетворяет ограничениям (условиям) основной теоремы безопасности, и все состояния такой системы также являются безопасными в смысле критерия Белла–Лападулы, но вместе с тем в этой системе любой субъект может реализовать поток типа read к любому объекту, что, очевидно, не совместимо с безопасностью в обычном понимании.

Следовательно, необходимо сформулировать теорему, в которой не только бы констатировалась безопасность всех достижимых состояний в системе при выполнении определенных условий, но и гарантировалась бы безопасность в процессе осуществления переходов между состояниями. Для этого необходимо регламентировать изменения уровней безопасности при переходе системы из одного состояния в другое с помощью дополнительных правил.

Такую интерпретацию мандатной модели осуществил Мак-Лин. Он предложил свою формулировку основной теоремы безопасности, основанную не на понятии безопасного состояния, а на понятии безопасного перехода.

При таком подходе функция уровня безопасности представляется в виде двух функций:

F_s – которая ставит каждому субъекту системы в соответствие определенный уровень безопасности из множества D;

F_o – которая ставит каждому объекту системы в соответствие определенный уровень безопасности из множества D.

Функция перехода T считается безопасной по чтению, если для любого перехода

$T(Q, r) = T'$

выполняются следующие условия:

1) если $read \in a_{ij}' \wedge read \notin a_{ij}$, (возникает новое отношение доступа), то

$F_s'(s_i) \geq F_o'(o_j)$; $F = F'$

2) если $F_s \neq F_s'$ (изменяются уровни безопасности субъекта), то

$A = A'$, $F_o = F_o'$ и

для $\forall s_i$ и o_j, для которых $F_s'(s_i) < F_o'(o_j)$,

$read \notin a_{ij}'$, $i = 1 \ldots m$, $j=1 \ldots n$.

3) если $F_o \neq F_o'$ (изменяется уровень безопасности объекта), то

$A = A'$, $F_s = F_s'$ и

для $\forall s_i$ и o_j, для которых $F_s'(s_i) < F_o'(o_j)$,

$read \notin a_{ij}'$, $i = 1 \ldots m$, $j=1 \ldots n$.

Функция перехода T считается безопасной по записи, если для любого перехода

$T(Q,r) = T'$

выполняются следующие три условия:

1) если $write \in a_{ij}' \wedge read \notin a_{ij}$, (возникает новое отношение доступа), то

$F_o'(o_j) \geq F_s'(s_i)$; $F = F'$

2) если $F_s \neq F_s'$ (изменяются уровни безопасности субъекта), то

$A = A'$, $F_o = F_o'$ и

для $\forall s_i$ и o_j, для которых $F_s'(s_i) > F_o'(o_j)$,

$write \notin a_{ij}'$, $i = 1 \ldots m$, $j=1 \ldots n$.

3) если $F_o \neq F_o'$ (изменяется уровень безопасности объекта), то

$A = A'$, $F_s = F_s'$ и

для $\forall s_i$ и o_j, для которых $F_s'(s_i) > F_o'(o_j)$,

$write \notin a_{ij}'$, $i = 1 \ldots m$, $j=1 \ldots n$.

Функция перехода является безопасной тогда и только тогда, когда она одновременно безопасна и по чтению и по записи.

Смысл введения перечисленных ограничений и их принципиальное отличие от условий теоремы Белла–Лападулы состоит в том, что нельзя изменить одновременно более одного компонента состояния системы.

В процессе перехода

- либо возникает новое отношение доступа;
- либо изменяется уровень безопасности субъекта;
- либо изменяется уровень безопасности объекта.

Следовательно, функция перехода является безопасной тогда и только тогда, когда она изменяет только один из компонентов состояния, и изменения не приводят к нарушению безопасности системы.

Поскольку безопасный переход из состояния в состояние позволяет изменяться только одному элементу и так как этот элемент может быть изменен только способами, сохраняющими безопасность состояния, была доказана следующая теорема о свойствах безопасной системы.

Теорема безопасности Мак-Лина

Система безопасна в любом состоянии и в процессе переходов между ними, если ее начальное состояние является безопасным, а ее функция перехода удовлетворяет критерию Мак-Лина.

Однако система может быть безопасной по определению Белла-Лападулы, но не иметь безопасной функции перехода.

Такая формулировка основной теоремы безопасности предоставляет в распоряжение разработчиков защищенных систем базовый принцип их построения, в соответствии с которым для обеспечения безопасности системы, как в любом состоянии, так и в процессе перехода между ними,

необходимо реализовать для нее такую функцию перехода, которая соответствует указанным условиям.

Выводы: классическая модель Белла-Лападулы построена для анализа систем защиты, реализующих мандатное (полномочное) разграничение доступа.

4.6. Ролевая политика безопасности

Ролевая политика безопасности представляет собой существенно усовершенствованную модель Харрисона–Руззо–Ульмана, однако ее нельзя отнести ни к дискреционным, ни к мандатным моделям.

В ролевой политике безопасности классическое понятие субъекта заменяется понятиями пользователь и роль.

Пользователь – это человек, работающий с системой и выполняющий определенные служебные обязанности, а с понятием роли связывается набор полномочий, необходимых для выполнения этих служебных обязанностей.

При использовании ролевой политики безопасности управление доступом осуществляется в две стадии:

- каждому пользователю назначается список доступных эму ролей;
- для каждой роли указывается набор полномочий, представляющий набор прав доступа к объектам.

Причем полномочия назначаются ролям в соответствии с принципом наименьших привилегий, из которого следует, что каждый пользователь должен обладать только минимально необходимым для выполнения своей роли набором полномочий.

В модели ролевой политики безопасности используются следующие множества:

U – множество пользователей;

R – множество ролей;

P – множество полномочий на доступ к объектам компьютерной системы, представленное, например, в виде матрицы доступа;

S – множество сеансов работы пользователей с системой.

Для перечисленных множеств определены следующие отношения:

$UA \subseteq U \times R$ – отображает множество пользователей на множество ролей, определяя для каждого пользователя набор доступных ему ролей,

$RA \subseteq P \times R$ – отображает множество полномочий на множество ролей, устанавливает для каждой роли набор присвоенных ей полномочий.

Правила управления доступом ролевой политики безопасности определяются следующими функциями:

User: $S \rightarrow U$ – эта функция для каждого сеанса S определяет пользователя, который осуществляет этот сеанс работы с системой: $User(s) = u \in U$,

Roles: $S \rightarrow P(R)$ – эта функция для каждого сеанса s определяет набор ролей из множества R, которые могут быть одновременно доступны пользователю в этом сеансе:

$Roles(s) = \{ r_i \mid (user(s), r_i) \in UA \}$

Permission: $S \rightarrow P$ – эта функция для каждого сеанса задает набор доступных в нем полномочий, который определяется как совокупность полномочий всех ролей, задействованных в этом сеансе:

$Permission(s) = U_{r \in roles(s)} \{ p_i \mid (p_i, r) \in RA \}$

Критерий безопасности ролевой модели определяется следующим правилом:

Система считается безопасной, если любой пользователь системы, работающий в сеансе s, может выполнять действия, требующие полномочия p, только в том случае, если $p \in Permission(s)$.

Из формулировки критерия безопасности ролевой модели следует, что управление доступом осуществляется главным образом не с помощью назначения полномочий ролям, а определением отношения UA, с помощью которого назначаются роли пользователям, и функции Roles (s), которая определяет доступный в сеансе набор ролей.

Существуют различные интерпретации ролевой модели, различающиеся видом функций User( ), Roles ( ), Permission ( ), а так же ограничениями, накладываемыми на отношения RA и UA.

Иерархическая организация ролей

Иерархическая организация ролей представляет собой наиболее распространенный тип ролевой модели, поскольку она очень точно отражает установившиеся реальном мире отношения подчиненности между участниками процесса обработки информации и разделение между ними сфер

ответственности. В реальной жизни, как правило, пользователи жестко упорядочены по степени ответственности, соответствующей уровню полномочий, которыми они обладают. Причем, более доверенные пользователи, стоящие на служебной лестнице выше, всегда обладают всеми полномочиями, менее доверенных, подчиненных им.

При иерархической организации ролей роли упорядочиваются по уровню предоставляемых полномочий. Чем выше роль пользователя в иерархии, тем больше с ней связано полномочий, поскольку считается, что если пользователю присвоена некоторая роль, то ему автоматически назначаются и все подчиненные ей по иерархии роли. Иерархия ролей допускает множественное наследование.

Иерархическая модель отличается от классической модели следующим:

- вводится дополнительное отношение $RH \subseteq R \times R$, которое отображает частичное отношение порядка на множестве R. Это отношение определяет иерархию ролей и задает на множестве ролей оператор доминирования $\leq$, такой что, если

$r_2 \leq, r_1$,

то r_1 находится в иерархии выше, чем r_2;

- отношение UA записывается в виде

$UA^h \subseteq U \times R$

С помощью этого отношения каждому пользователю назначается набор ролей, причем вместе с каждой ролью в него включаются и все роли, подчиненные ей по иерархии, т. е.

для $\forall r_1, r_2 \in R, u \in U: r_2 \leq r_1 \wedge (u, r_1) \in UA^h \Rightarrow (u, r_2) \in UA^h$

- функция Roles: $S \rightarrow P(R)$ принимает вид

$Roles^h : S \rightarrow P(R)$

Эта функция назначает каждому сеансу s набор ролей из иерархии ролей пользователя, работающих в этом сеансе:

$Roles^h(s) \subseteq \{ r_i \mid \exists r_i \leq r_1 (user(s), r_1) \in UA^h \}$

- функция Permission: S ® P заменяется функцией

$Permission^h: S \rightarrow P$

Эта функция определяет полномочия сеанса как совокупность полномочий всех задействованных в нем ролей и полномочий всех ролей подчиненных им

$Permission^h(s) = U_{r1 \in roles^h(s)} \{ p_i \mid \exists r_2 \leq r_1 (p_i, r_2) \in RA^h \}$

Таким образом, каждому пользователю назначается некоторое подмножество иерархии ролей, а в каждом сеансе является доступной совокупность полномочий ролей, которая составляет фрагмент этой иерархии.

Такой подход позволяет существенно упростить управление доступом за счет неявного назначения полномочий.

Взаимоисключающие роли

В этом случае множество ролей пользователей разбивается на подмножества, объединяющее роли, которые не могут быть назначены пользователю одновременно и эти подмножества считаются несовместимыми. Таким образом, пользователю может быть назначено только по одной роли из каждого подмножества несовместимых ролей.

Для определения отношения несовместимости на множестве ролей R задается функция

Exclusive: $R \rightarrow P(R)$,

которая для каждой роли определяет множество несовместимых с ней ролей. В этом случае на отношение UA, отображающее множество ролей на множество пользователей, накладывается следующее ограничение:

если $(u, r_1) \in UA \wedge r_2 \in Exclusive(r_1) \Rightarrow (u, r_2) \notin UA$.

Взаимоисключающие роли реализуют так называемое статическое разделение обязанностей, когда конфликт несовместимости полномочий решается на стадии назначения ролей.

Ограничение на одновременное использование ролей в рамках одного сеанса

В этом случае множество ролей также разбивается на подмножество несовместимых ролей, но отношение UA может назначить пользователю любую комбинацию ролей. Однако в ходе сеанса работы с системой пользователь может одновременно активизировать не более одной роли из каждого подмножества несовместимых ролей.

В этом случае на функцию Roles: $S \to P(R)$, которая назначает каждому сеансу s некоторый набор ролей, накладывается следующее ограничение:

если $\forall r_1, r_2 \in R: r_1 \in Roles(s) \wedge r_2 \in Exclusive(r_1) \Rightarrow r_2 \notin Roles(s)$.

Поскольку в процессе сеанса пользователь может переключаться между различными ролями, то он должен избегать конфликтов несовместимости между ними. Эта политика получила название динамического разделения обязанностей.

Количественные ограничения при назначении ролей и полномочий

Эта модель предназначена для тех случаев, когда роль может быть назначена только ограниченному числу пользователей, и/или представление некоторых полномочий допускается только для ограниченного числа ролей.

Модель использует функцию $Cardinality^r: R \to N$, которая определяет какому числу пользователей может быть назначена каждая роль, и функцию $Cardinality^p: P \to N$, которая для каждого полномочия определяет скольким ролям оно может быть присвоено.

На отношения UA и RA накладываются следующие ограничения

для $\forall r_i \{ | u_j | (u_j, r_i) \in UA | \} \le Cardinality^r(r_i)$ и

для $\forall p_i \{ | r_j | (p_i, r_j) \in RA | \} \le Cardinality^p(p_i)$

Смысл данных ограничений состоит в том, что благодаря ограничению количества пользователей, осуществляющих те или иные операции, сужается круг лиц, на которых лежит ответственность за совершение соответствующих действий.

Группировка ролей по полномочиям

Роли и полномочия, которые дополняют друг друга, и назначение которых по отдельности не имеет смысла, объединяются в группы, которые могут быть назначены только целиком. Для этого вводятся дополнительные правила, в соответствии с которыми

- любая роль может быть назначена пользователю только в том случае, если ему уже присвоен определенный набор ролей, для этого на множестве ролей R задается функция

$Prerequisite^r: R \to P(R)$,

определяющая для каждой роли подмножество ролей, которые должны быть назначены пользователю прежде, чем он получить эту роль, а для отношения UA, назначающего пользователям роли, вводится ограничение, требующее наличия у пользователя всех ролей, принадлежащих одной группе:

для $\forall (u, r) \in UA \wedge r' \in Prerequisite^r(r) \Rightarrow (u, r') \in UA$

- роль может быть наделена полномочиями только тогда, когда с ней уже связан определенный набор полномочий, для этого на множестве P задается функция

$Prerequisite^p: P \to P(P)$,

определяющая для каждого полномочия подмножества полномочий, которые должны быть присвоены роли перед назначением полномочия, а на отношение RA, отображающее множество полномочий на множество ролей, накладывается ограничение, в соответствии с которым роли сразу присваиваются все полномочия из группы

для $\forall (p, r) \in PA \wedge p' \in Prerequisite^p(p) \Rightarrow (p', r) \in RA$.

Введение подобных ограничений упрощает администрирование системы в тех случаях, когда полномочия должны предоставляться определенным набором, или когда назначение ролей должно производиться в определенной последовательности.

Вывод: Ролевая политика представляет широкий выбор для разработчиков систем управления доступом – с одной стороны, использование матрицы прав доступа может превратить ее в разновидность дискреционной модели, но, с другой стороны, применение жестких правил распределения ролей между сеансами и пользователями, а также полномочий между ролями, позволяет построить на ее основе полноценную нормативную политику.

Модель политики безопасности распределенной компьютерной системы

Распределенная компьютерная система состоит из двух сегментов:

- локального, который включает в себя один компьютер или сегмент локальной вычислительной сети;
- внешнего, который представляет собой компьютеры либо сегменты локальной вычислительной сети, не включенные в локальный сегмент.

Удаленным субъектом называется субъект, принадлежащий множеству субъектов внешнего сегмента компьютерной сети.

Удаленным объектом называется объект, принадлежащий множеству объектов внешнего сегмента компьютерной сети.

В распределенной компьютерной системе существует четыре вида потоков:

- потоки между локальными субъектами и локальными объектами;
- потоки между локальными субъектами и удалёнными субъектами;
- потоки между удаленными субъектами и локальными объектами;
- потоки между удаленными субъектами и удаленными объектами.

Примечание: Поток между субъектом и объектом означает поток между ассоциированным объектом субъекта и объектом.

В дальнейшем будут рассматриваться только потоки между локальными субъектами и локальными объектами при участии в потоке локального субъекта.

Передача данных на расстояние требует использования телекоммуникационного программного обеспечения (ТПО), с помощью которого осуществляется совместная работа прикладных программных средств и аппаратуры передачи данных. В объектно-субъектной модели телекоммуникационное программное обеспечение рассматривается как субъект $s_{ком}$, относящийся к подмножеству субъектов локального сегмента компьютерной системы.

Обозначим поток от ассоциированного объекта $o_{уд}$ удаленного субъекта $s_{уд}$ к ассоциированному объекту $o_{ком}$ субъекта $s_{ком}$ как

Stream ($s_{уд}$, $o_{уд}$) $\rightarrow$ $o_{ком}$.

Предположим, что свойства субъекта $s_{ком}$ таковы, что возможно существование потока

Stream ($s_{ком}$, $o_{ком}$) $\rightarrow$ $o_{лок}$,

тогда по свойству транзитивности потоков имеет место доступ субъекта $s_{уд}$ к объекту $o_{лок}$ через субъект $s_{ком}$.

Метод межсетевого экранирования

Недостатки классической модели политики безопасности с полным проецированием прав пользователя на всё множество субъектов привели к появлению методов защиты, связанных с "экранированием" локального сегмента компьютерной системы от внешнего сегмента компьютерной системы.

Суть экранирования состоит в том, что поток между удаленным субъектом $s_{уд}$ и телекоммуникационным субъектом $s_{ком}$ осуществляется через дополнительный объект o_f, ассоциированный с субъектом-анализатором s_f.

В этой системе существуют следующие потоки:

Stream($s_{уд}$, $o_{уд}$) $\rightarrow$ o_f

Stream(s_f, o_f) $\rightarrow$ $o_{ком}$

Stream($s_{ком}$, $o_{ком}$) $\rightarrow$ $o_{лок}$

и потоки обратного направления

Stream($s_{ком}$, $o_{лок}$) $\rightarrow$ $o_{ком}$

Stream($s_{ком}$, $o_{ком}$) $\rightarrow$ o_f

Stream(s_f, o_f) $\rightarrow$ $o_{уд}$.

В этой модели субъект s_f рассматривается как некоторый фильтр, который может определить факт доступа субъекта $s_{уд}$ к объекту $o_{лок}$, или зафиксировать поток между $o_{уд}$ и $o_{ком}$. Допускается, что поток может рассматриваться на различном уровне относительно объекта $o_{лок}$.

Рассмотрим случай передачи объекта o_j удаленному субъекту $s_{уд}$. При использовании режима пакетной передачи данных объект o_j разбивается на подобъекты:

$o_j = \{ o_j^{t1}, o_j^{t2}, \ldots o_j^{ti}, \ldots o_j^{tk} \}$

(осуществляется декомпозиция объект на последовательность подобъектов)

Тогда телекоммуникационный субъект $s_{ком}$ должен инициализировать поток:

Stream($s_{ком}$, o_j^{ti}) $\rightarrow$ o_f^{ti},

где $t_i \in T = \{ t_1, t_2, \ldots, t_i, \ldots t_k \}$, t_i – промежуток времени.

При чем объекты o_j^{ti} и o_j^{ti} должны быть тождественными, а за период времени T через субъект $s_{ком}$ должен пройти весь объект o_j.

При пакетной передаче данных объект o_j^{ti} представляется в виде:

$o_j^{ti} = o_j^{ti}(адр) \,||\, o_j^{ti}(инф)$,

где $o_j^{ti}(адр)$ – называется адресной частью подобъекта o_j^{ti} ; $o_j^{ti}(инф)$ – информационной частью подобъекта o_j^{ti}

Тогда объект o_j можно представить в вид:

$o_j = (o_j^{t1}(адр), o_j^{t2}(адр),\ldots, o_j^{ti}(адр),\ldots, o_j^{t1}(инф), o_j^{t2}(инф),\ldots, o_j^{ti},\ldots)$

(как последовательность слов, определяющих адресные и информационные составляющие подобъектов).

Очевидно, что только последовательность подобъектов

$o_j^{t1}(инф), t_{i \in T}$

представляет объект o_j, а дополнительные подобъекты

$o_j^{ti}(адр), t_{i \in T}$

необходимы для передачи информационной части подобъекта по соответствующему адресу.

Утверждение (о существовании декомпозиции на подобъекты).

Если существует поток

$Stream(s_{уд,} o_j^{ti}) \rightarrow o_{уд}$,

где $o_{уд}$ – ассоциированный объект удаленного субъекта $s_{уд}$ и объекты o_j^{ti} и $o_{уд}$ тождественны, то для любого субъекта $s_{уд}$ существует декомпозиция каждого объекта

$o_j^{ti} = o_j^{ti}(адр) \,||\, o_j^{ti}(инф)$

при которой $||\, o_j^{ti}(инф)$ (конкатенация всех информационных подобъектов) составляет целый объект o_j.

Доказательство

Из утверждения также следует, что на произвольном уровне r (одном из семи) поток подобъектов, проходящих через субъект-фильтр, содержит полную информацию о всем объекте o_j. Однако структура объекта o_j^{ti} зависит от конкретного субъекта.

При рассмотрении любого множества подобъектов, составляющих o_j, получение полной информации о том, к какому именно объекту локального сегмента компьютерной сети происходит доступ, не представляется возможным. В связи с этим как минимально необходимую задачу реализации политики безопасности в субъекте-фильтре необходимо рассмотреть сборку полного пакета (объекта) из подпакетов (подобъектов).

Отметим, что политика безопасности реализуется на уровне целого объекта, а не составляющих его подобъектов.

Сформулируем задачу корректного экранирования на некотором уровне r.

Субъект s_f называется корректно экранирующим (или корректно фильтрующим) на вход относительно субъекта $s_{ком}$, если для любого объекта o_j при потоке

$Stream(s_{ком}, o_f) \rightarrow o_j$

По последовательности

$o_f^{t1}, o_f^{t2},\ldots o_f^{ti},\ldots o_f^{tk}$

можно однозначно восстановить объект o_j.

Субъект s_f называется корректно экранирующим (или корректно фильтрующим) на выход относительно субъекта $s_{ком}$, если для любого объекта o_j при потоке

$Stream(s_{ком}, o_{j)} \rightarrow o_f$

по последовательности

$o_f^{t1}, o_f^{t2},\ldots o_f^{ti},\ldots o_f^{tk}$

можно однозначно восстановить o_j.

Субъект s_f называется корректным фильтром, если он является корректно фильтрующим кок на вход, так и на выход.

Утверждение (основная теорема о корректном экранировании)

Экранирующий субъект s_f, участвующий в потоке подобъектов уровня r, будет корректным на вход и на выход тогда и только тогда, когда для любого $s_{ком}$ и для любого o_j по последовательности

$o_j^{t1}, o_j^{t2},\ldots o_j^{ti},\ldots o_j^{tk}$ уровня r

однозначно определяется объект o_j.

Доказательство

Основная теорема о корректном экранировании хотя и является критерием, но тем не менее недостаточна конструктивна. Кроме того, субъект-фильтр не производит разделения потоков на легальные и нелегальные.

Отметим два существенных момента:

- субъект-фильтр должен иметь информацию о самих объектах o_j для осуществления сравнения;
- субъект-фильтр должен иметь информацию о разрешенных и запрещенных потоках между объектами $o_{уд}$ и $o_{ком}$.

Гарантированно изолирующим фильтром называется корректный фильтр, который разрешает порождение потоков

$Stream(s_{уд}, o_j) \rightarrow o_{уд}$ и $Stream(s_{уд}, o_{уд}) \rightarrow o_j$

только для потоков, принадлежащих множеству легальных потоков.

На практике субъект-фильтр не имеет доступа к множеству объектов локального сегмента компьютерной системы. В этом случае задача восстановления объекта o_j по последовательности подобъектов o_j^{ti} не может быть решена в явном виде.

Утверждение (необходимое условие гарантированной изоляции для субъекта-фильтра)

Для того чтобы фильтр был гарантировано изолирующим, необходимо обеспечить существование потоков

$Stream(s_f, o_j) \rightarrow o_f$

и выполнить условие тождественности объектов o_j и o_f, где o_f – ассоциированный объект субъекта s_f, который служит для сравнения объекта o_j с объектом, восстановленным по последовательности подобъектов объекта o_j.

Доказательство

В зарубежных разработках вводится понятие сервиса – субъекта, в котором реализованы конкретные алгоритмы декомпозиции. При чем эти алгоритмы порождают последовательности подобъектов, свойственные только данному субъекту.

В этом случае для описания доступа из внешней среды выделяется множество доступных сервисов. Эти сервисы описывают множество субъектов, для которых разрешается поток к произвольному объекту локального сегмента компьютерной системы.

При этом сформулированные выше замечания относительно политики безопасности, реализованной в локальном сегменте компьютерной системы, остаются действительными. Любая политика с полным проецированием прав также будет некорректна относительно сервиса, допускающего доступ удаленного субъекта к объекту локального сегмента компьютерной системы.

Однако, телекоммуникационный объект может обладать такими свойствами, которые позволят исключить опасные для защищенности локального сегмента компьютерной системы потоки между $o_{уд}$ и $o_{ком}$. Следовательно, ограничение доступных сервисов имеет смысл для построения защиты.

С другой стороны фильтрация сервисов ограничивает множество локальных субъектов, которые могут иметь доступ к объектам локального сегмента компьютерной системы (т.к. для каждого из этих субъектов должен быть свой алгоритм декомпозиции объекта над подобъекты.

Утверждение (о тождестве фильтра сервисов и изолированной программной среды в рамках локального сегмента компьютерной системы).

Возможности внешнего злоумышленника по отношению к объектам локального сегмента компьютерной системы одинаковы как в случае существования фильтрующего субъекта s_f, так и генерации изолированной программной среды с включением субъектов

$s_1 \ldots s_m$ из s_n,

где $s_1 \ldots s_m$ – допустимые сервисы, s_f – фильтр сервисов, который допускает существование только сервисов $s_1 \ldots s_m$ из s_n.

Доказательство

Из утверждения следует, что методы защиты, связанные с разрешенными сервисами, в принципе эквивалентны методу генерации изолированной программной среды для локального сегмента компьютерной системы, в которую включены локальные субъекты, обеспечивающие телекоммуникационное взаимодействие.

В сущности, уменьшение множества субъектов, как методом генерации изолированной программной среды, так и методом фильтрации сервисов является только гарантией выполнения

политики безопасности, реализованной в субъектах локального сегмента компьютерной системы, либо субъекте-фильтре.

Далее следует отметить, что свойства произвольного субъекта $s_{уд}$ внешнего сегмента относительно s_f могут быть произвольными.

Аксиома. При произвольном составе субъектов внешнего сегмента компьютерной системы возможно формирование подобъектов на уровне ассоциированных объектов o_f субъекта-фильтра s_f для потока

$Stream(s_{уд}, o_{уд)} \rightarrow o_f$

с произвольной адресной и информационной частью.

Из аксиомы следует, что фильтрация подобъектов изолированно от содержания объектов локального сегмента компьютерной системы в общем случае потенциально ненадежна относительно любых критериев фильтрации при возможности управления телекоммуникационным субъектом локального сегмента компьютерной системы со стороны злоумышленника.

Пусть в субъекте-фильтре однозначно выделяются информационные подобъекты и реализация потока

$Stream(s_{ком}, o_{j)} \rightarrow o_f$

является тождественным отображением (технически это означает безошибочную передачу в тракте "фильтр-компютер").

Для всех объектов локального сегмента компьютерной системы вычисляется хэш-функция

$H(o_{j,} s_{удi}) = h_{ji}$

(хэш-функция может зависеть от индивидуальной информации пользователя $s_{удi}$) и гарантируется их доступность для субъекта-фильтра.

Процедура фильтрации на выход (относительно существующих объектов) формулируется следующим образом:

По последовательности подобъектов

$o_j^{t1}, o_j^{t2},\ldots o_j^{ti},\ldots o_j^{tk}$

восстанавливается объект o_z.

Вычисляется хэш-функция

$H(o_{z,} s_{удi}) = h_{zi}$

Вычисленное значение h_{zi} сравнивается с h_{ji}.

В случае совпадения проверяются права доступа к объекту o_j.

В случае доступности объекта для передачи во внешний сегмент компьютерной системы разрешается передача подобъектов, соответствующих декомпозиции объекта, во внешнюю сеть.

В случае несовпадения передача запрещается.

Указанный метод может быть дополнен фильтрацией сервисов для обеспечения достоверного восстановления объекта по последовательности подобъектов.

23. Назначение состав и архитектура информационно-справочных систем. Структура и состав подсистемы защиты информации. Методы и средства защиты информации в СУБД

Информационная система представляет собой среду, составляющими элементами которой являются компьютеры, компьютерные сети, программные продукты, базы данных, люди, различного рода технологические и программные средства. А информационная технология есть совокупность операций и действий над данными. Все процессы преобразования информации в информационной системе осуществляются с помощью информационных технологий. В результате информационная технология является более емким понятием, чем информационная система. Реализация функций информационной системы невозможна без знаний ориентированной на нее информационной технологии. Информационная технология может существовать и вне сферы информационной системы.

Информационная система (ИС) представляет собой совокупность информационных, технических, программных, математических, организационных, правовых, эргономических, лингвистических, технологических и других средств, а также персонала, предназначенных для сбора, обработки, хранения и выдачи информации и принятия управленческих решений. Функционирование ИС во времени заключается в сборе, хранении, обработке и распространении информации о деятельности какого-то экономического объекта реального мира.

Набор этих функций определяет процессы в информационной системе:
- ввод информации из внешних и внутренних источников;
- обработка входящей информации;
- хранение информации для последующего ее использования;
- вывод информации в удобном для пользователя виде;
- обратная связь, то есть использование переработанной информации для сопоставления с исходной, с целью корректировки входящей информации.

Если ранее информационные системы воспринимались как средство автоматизации вспомогательной деятельности предприятия, то теперь информационные системы стали средством получения конкурентного преимущества.

Структура каждой информационной системы состоит из функциональных и обеспечивающих подсистем, что представлено на рисунке.

Функциональные подсистемы ИС информационно обслуживают определенные виды деятельности предприятия, характерные для структурных подразделений предприятия и функций управления. Функциональная подсистема представляет собой комплекс экономических задач с высокой степенью информационных обменов (связей) между ними. При этом под задачей будем понимать некоторый процесс обработки информации с четко определенным множеством входной и выходной информации (например, начисление заработной платы, учет заказов, оформление брони и т.д.). Состав функциональных подсистем во многом определяется особенностями экономической системы, ее отраслевой принадлежностью, формой собственности, размером, характером деятельности предприятия.

Функциональная подсистема – это подсистема, реализующая одну или несколько взаимосвязанных функций. Назначение подсистемы, ее основные задачи, цели и функции определяются видами деятельности производственных и хозяйственных объектов: производственная, кадровая, финансовая, маркетинговая. Указанные направления деятельности и определяют типовой набор функциональных подсистем ИС.

Обеспечивающая подсистема – это среда, в которой используются средства для преобразования информации независимо от сферы применения. Интеграция функциональных подсистем в единую систему достигается за счет создания и функционирования обеспечивающих подсистем, таких как программная, техническая, организационная, правовая, информационная, эргономическая, лингвистическая и математическая подсистемы.

1. Подсистема «Программное обеспечение» – это совокупность программ, реализующих функции ИС; инструктивно-методические материалы по применению средств программного обеспечения; а также персонал, занимающийся разработкой и сопровождением программ на весь период жизненного цикла ИС.

Программное обеспечение делится на два комплекса: общесистемное (операционные системы, операционные оболочки, компиляторы, интерпретаторы, программные среды для разработки прикладных программ, СУБД, сетевые программы, антивирусные программы, тестовые и диагностические программы) и прикладное программное обеспечение (совокупность прикладных программ, разработанных для конкретных задач в рамках функциональных подсистем, и контрольные примеры).

2. Подсистема «Техническое обеспечение» – это комплекс технических средств, предназначенных для обработки данных в ИС; методические и руководящие материалы, техническая документация; обслуживающий эти технические средства персонал. В состав комплекса входят компьютеры, средства сбора и регистрации информации, средства передачи данных по каналам связи, средства накопления и хранения данных и выдачи результатной информации, вспомогательное оборудование и организационная техника, что представлено на рисунке.

Средства вычислительной техники предназначены в основном для реализации комплексных технологий обработки и хранения информации и являются базой интеграции всех современных технических средств обеспечения управления информационными ресурсами:

- персональные компьютеры, все ресурсы которых полностью направлены на обеспечение деятельности одного работника;

- корпоративные компьютеры (main frame), обеспечивающиесовместную деятельность многих работников в рамках одной организации, одного проекта, одной сферы информационной деятельности при использовании одних и тех же информационно-вычислительных ресурсов;

- суперкомпьютеры– это вычислительные системы с предельными характеристиками вычислительной мощности и информационных ресурсов (военная, космическая области деятельности, фундаментальные научные исследования, глобальный прогноз погоды).

Средства коммуникационной техники обеспечивают одну из основных функций управленческой деятельности – передачу информации в рамках системы управления и обмен данными с внешней средой, предполагают использование разнообразных методов и технологий.

К средствам коммуникационной техники относятся:

- средства и системы стационарной и мобильной телефонной связи;
- средства и системы телеграфной связи;
- средства и системы факсимильной передачи информации и модемной связи;
- средства и системы кабельной и радиосвязи, включая оптико-волоконную и спутниковую связи (вычислительные сети).

Средства оргтехники предназначены для автоматизации и механизации управленческой деятельности. Реализуются технологии хранения, представления и использования информации, а также для выполнения различных вспомогательных операций в рамках тех или иных технологий информационной поддержки управленческой деятельности.

Всю совокупность оргтехники можно представить в виде следующих групп:

- носители информации;
- средства изготовления текстовых и табличных документов;
- средства репрографии и оперативной полиграфии;
- средства обработки документов;
- средства хранения, поиска и транспортировки документов;
- банковская оргтехника;
- малая оргтехника;
- офисная мебель и оборудование;
- прочая оргтехника.

Подсистема «Организационное обеспечение» является одной из важнейших подсистем ИС, от которой зависит успешная реализация целей и функций системы. В составе организационного обеспечения можно выделить четыре группы компонентов.

Первая группа включает важнейшие методические материалы, регламентирующие процесс создания и функционирования системы:

- общеотраслевые руководящие методические материалы по созданию ИС;
- типовые проектные решения;
- методические материалы по организации и проведению предпроектного обследования на предприятии;
- методические материалы по вопросам создания и внедрения проектной документации.

Вторым компонентом в структуре организационного обеспечения ИС является совокупность средств, необходимых для эффективного проектирования и функционирования ИС (типовые пакеты прикладных программ, типовые структуры управления предприятием, унифицированные системы документов, общесистемные и отраслевые классификаторы и т.п.).

Третьим компонентом подсистемы организационного обеспечения является техническая документация, получаемая в процессе обследования, проектирования и внедрения системы: технико-экономическое обоснование, техническое задание, технический и рабочий проекты и документы, оформляющие поэтапную сдачу системы в эксплуатацию.

Четвертым компонентом подсистемы организационного обеспечения является персонал, где представлена организационно-штатная структура проекта, определяющая, в частности, состав главных конструкторов системы и специалистов по функциональным подсистемам управления.

4. Подсистема «Правовое обеспечение» предназначена для регламентации процесса создания и эксплуатации ИС, которая включает совокупность юридических документов с констатацией регламентных отношений по формированию, хранению, обработке промежуточной и результатной информации системы.

К правовым документам, действующим на этапе создания системы, относятся: договор между разработчиком и заказчиком; документы, регламентирующие отношения между участниками процесса создания системы.

К правовым документам, создаваемым на этапе внедрения, относятся: характеристика статуса создаваемой системы; правовые полномочия подразделений ИС; правовые полномочия отдельных видов процессов обработки информации; правовые отношения пользователей в применении технических средств.

5. Подсистема «Информационное обеспечение» представляет собой совокупность проектных решений по объемам, размещению, формам организации информации, циркулирующей в ИС (информационные потоки). Она включает в себя совокупность показателей, справочных данных, классификаторов и кодификаторов информации, унифицированные системы документации, специально организованные для обслуживания, массивы информации на соответствующих носителях.

В состав подсистемы включаются два комплекса. Это компоненты внемашинного информационного обеспечения (классификаторы технико-экономической информации, кодификаторов информации, справочные данные, унифицированные системы документации) и компоненты внутримашинного информационного обеспечения (макеты/экранные формы для ввода/вывода информации, структура информационной базы). В нее также входит персонал, обеспечивающий надежность хранения, своевременность и качество технологии обработки информации.

Центральным компонентом информационного обеспечения является база данных, через которую осуществляется обмен данными различных задач. База данных обеспечивает интегрированное использование различных информационных объектов в функциональных подсистемах.

Методы и средства защиты информации с СУБД

В настоящее время объём информации в мире настолько велик, что самым оптимальным методом работы с ней является база данных (БД). База данных – это представленная в объективной форме совокупность материалов, систематизированных так, чтобы эти материалы могли быть найдены и обработаны с помощью компьютера. Её защита является одной из самых сложных задач на сегодняшний день.

Угрозы потери конфиденциальной информации стали обычным явлением, и если в системе защиты есть недостатки, то ценные данные могут оказаться в руках третьих лиц. Каждый сбой работы

БД может парализовать работу целых корпораций, фирм, что приведет к весомым материальным потерям.

Методы защиты баз данных в различных СУБД условно делятся на две группы (анализ современных фирм Borland и Microsoft): основные и дополнительные.

К основным средствам защиты относится:

- защита паролем;
- шифрование;
- разделение прав доступа к объектам БД;
- защита полей и записей таблиц БД.

Защита паролем – это самый простой способ защиты БД от несанкционированного доступа.

Пароли устанавливаются пользователями или администраторами. Их учет и хранение выполняется системой управления базой данных (СУБД). Пароли хранятся в специальных файлах СУБД в шифрованном виде. После ввода пароля пользователю предоставляется доступ к требуемой информации.

Несмотря на простоту парольной защиты, у неё имеется ряд недостатков. Во-первых, пароль уязвим, особенно если он не шифруется при хранении в СУБД. Во-вторых, пользователю надо запоминать или записать пароль, а при небрежном отношении к записям пароль может стать достоянием других.

Более мощным средством защиты данных является шифрование. Шифрование – это процесс перевода информации по определенному алгоритму в вид непригодный для чтения, в целях защиты от несанкционированного просмотра или использования. Важной особенностью любого алгоритма шифрования является использование ключа, который утверждает выбор конкретного метода кодирования из всех возможных. В основном применяется для защиты уязвимых данных.

Шифрование обеспечивает три состояния безопасности информации:
- конфиденциальность.
- целостность.
- идентифицируемость.

В целях контроля использования основных ресурсов СУБД во многих системах имеются средства установления прав доступа к объектам БД. Права доступа определяют возможные действия над объектами. Владелец объекта, а также администратор БД имеют все права. Остальные пользователи имеют те права и уровни доступа к объектам, которыми их наделили.

Разрешение на доступ к конкретным объектам базы данных сохраняется в файле рабочей группы.

Файл рабочей группы содержит данные о пользователях группы и считывается во время запуска. Файл содержит следующую информацию: имена учетных записей пользователей, пароли пользователей, имена групп, в которые входят пользователи.

К дополнительным средствам защиты БД можно отнести следующие средства:
- встроенные средства контроля значений данных в соответствии с типами:
- повышение достоверности вводимых данных:
- обеспечения целостности связей таблиц:
- организации совместного использования объектов БД в сети.

Описанные выше методы и способы являются основополагающими, однако их использование не гарантирует полной сохранности данных. Для повышения уровня безопасности информации в БД рекомендуется использование комплексных мер.

24.Назначение, состав и архитектура сложных корпоративных информационных систем. Угрозы информации, которые характерны им

В составе информационных систем можно выделить две относительно независимых составляющие. Первая представляет собой собственно *компьютерную инфраструктуру* организации в широком смысле этого слова (сетевая, телекоммуникационная, программная, информационная, организационная инфраструктура - то есть то, что носит в статье обобщенное название *Корпоративная Сеть*). Вторая составляющая суть взаимосвязанные функциональные подсистемы, обеспечивающие решение задач организации и достижение ее целей. Если первая отражает системно-техническую, структурную сторону любой информационной системы, то вторая целиком относится к прикладной области и сильно зависит от специфики задач организации и ее целей.

Первая составляющая представляет собой базис, основу для интеграции функциональных подсистем и целиком определяет свойства информационной системы, важные для ее успешной эксплуатации. Требования к ней едины и стандартизованы, а методы ее построения хорошо известны и многократно проверены на практике.

Вторая составляющая строится целиком на базе первой и привносит в информационную систему прикладную функциональность. Требования к ней сложны и зачастую противоречивы, так как выдвигаются специалистами из различных прикладных областей. Однако эта составляющая в конечном счете более важна для функционирования организации, так как ради нее, собственно, и строится вся инфраструктура.

2.2. Соотношение

Между двумя составляющими информационной системы можно проследить следующие взаимосвязи.

Составляющие независимы в определенном смысле. Организация будет эксплуатировать высокоскоростную сеть 100 MB Ethernet вне зависимости от того, какие методы и программы для организации бухгалтерского учета планируется принять на вооружение. Сеть организации будет построена на базе протокола TCP/IP независимо от того, какой текстовый процессор будет принять в качестве стандартного. Иными словами, в современных условиях базовая инфраструктура становится все более универсальной.

Составляющие зависимы в определенном смысле. Вторая невозможна без первой, первая без второй ограничена, поскольку лишена необходимой функциональности. Невозможно эксплуатировать прикладную систему с архитектурой клиент-сервер, когда отсутствует или некачественно построена сетевая инфраструктура. Однако, имея развитую инфраструктуру, можно предоставить сотрудникам организации ряд полезных общесистемных сервисов, (например, электронную почту), упрощающих работу и делающих ее эффективной (в нашем примере - за счет электронных коммуникаций). Если выбран этот эволюционный путь развития информационной системы, то в процессе своего развития Корпоративная Сеть постепенно приобретает ряд прикладных сервисов, направленных на решение универсальных задач организации - задач управления и координации.

Вторая составляющая более изменчива. Действительно, инфраструктура организации зависит только от территориального расположения ее подразделений, да и то скорее в отношении инфраструктуры, никак не влияя на используемые для ее построения технологии. Вторая составляющая сильно зависит от организационно-управленческой структуры организации, ее функциональности, распределения функций, принятых в организации финансовых технологий и схем, существующей технологии документооборота и множества других факторов.

Первая составляющая имеет долговременный характер. Инфраструктура создается на многие годы вперед - так как капитальные затраты на ее создание настолько велики, что практически исключают возможность полной или частичной переделки уже построенного. Напротив, вторая составляющая изменчива по своей природе, так как в предметной части деятельности организации постоянно происходят более или менее существенные подвижки, которые должны быть отражены и в функциональных подсистемах. Этот тезис особенно актуален в контексте постоянно происходящих изменений в административных структурах многих отечественных организаций.

Степень определенности в выборе технологических решений для первой составляющей несколько выше, чем для второй. Действительно, современные компьютерные технологии предлагают такие промышленные решения для построения инфраструктуры организации, которые гарантировано обеспечат непрерывное развитие и совершенствование системно-технической базы информационной системы с перспективой на многие годы вперед. Первая составляющая имеет более отношение к

технике, чем к экономике и управлению, и в этом смысле более стабильна, а ее развитие является более прогнозируемым и управляемым.

До недавнего времени в технологии создания информационных систем доминировал традиционный подход, когда вся архитектура информационной системы строилась "сверху-вниз" - от прикладной функциональности к системно-техническим решениям и первая составляющая информационной системы целиком выводилась из второй.

Практика многих больших российских проектов показала, что начинать построение КС только с анализа бизнес-процессов (не уделяя должного внимания инфраструктуре), весьма и весьма проблематично. Автоматизация деятельности корпорации на основе концепции "сверху-вниз" и принципов BPR (Business Process Reengineering) предполагает такую реорганизацию КС, которая наилучшим образом служит решению управленческих задач. Проблема заключается в том, что в современных российских условиях - условиях сверхдинамичного бизнеса, постоянно возникающих форс-мажорных обстоятельств и исключительно быстро меняющихся правил игры (социальных, политических, экономических), в рамках которой строится вся прикладная функциональность (как раз и обеспечивающая решение управленческих задач) - систематизация управленческой деятельности представляет собой весьма сложную задачу ввиду высокой степени неопределенности.

В то же время бессмысленно строить инфраструктуру, не обращая внимания на прикладную функциональность. Если в процессе создания системно-технической инфраструктуры не проводить анализ и автоматизацию управленческих задач, то инвестированные в нее средства не дадут впоследствии реальной отдачи. Аппаратное и программное обеспечение инфраструктуры будет "висеть мертвым грузом" на плечах организации, требуя ежегодных затрат на сопровождение и модернизацию. Подход к построению КС "снизу-вверх" (с акцентом на системно-техническую инфраструктуру) вряд ли можно рассматривать в качестве магистрального.

В настоящее время развивается комбинированный подход, который можно характеризовать как "встречное движение": компьютерная инфраструктура и системная функциональность строятся так, чтобы в максимальной степени обеспечить изменчивость на уровне прикладной функциональности. Параллельно проводится анализ и структуризация бизнесс-процессов, сопровождающиеся внедрением соответствующих программных решений, привносящих в КС прикладную функциональность.

Опираясь на сказанное выше, рискнем сделать следующий вывод. Разработку информационной системы целесообразно начинать с построения компьютерной инфраструктуры (Корпоративной Сети) как наиболее важной (фундаментальной) системообразующей составляющей, опирающейся на апробированные промышленные технологии и гарантировано реализуемой в разумные сроки в силу высокой степени определенности как в постановке задачи, так и в предлагаемых решениях. Одновременно, в контексте архитектуры Корпоративной Сети, как единого обобщенного взгляда на фундамент информационной системы, на наиболее важных и ответственных участках целесообразно выполнять разработки, насыщающие систему прикладной функциональностью (то есть внедрять системы финансового учета, управления кадрами и т.д.). Далее, прикладные программные системы будут распространены и на другие, первоначально менее значимые области управленческой деятельности.

В этом контексте особенно важными становятся:

- Широкий спектр готовых к применению промышленных прикладных систем для различных областей управленческой деятельности (как правило, поставляемых одной компанией);
- Высокая степень гранулярности таких решений (не обязательно внедрять сразу всю систему целиком - можно начать с отдельных участков);
- Построение на основе единого системного фундамента (как правило, в качестве фундамента выступает современная реляционная СУБД).

Подобный эволюционный подход, опирающийся на корпоративные стандарты, в конечном счете позволить построить реальную КС.

Предлагаемая вниманию читателя концепция опирается на обобщенное понятие Корпоративной Сети как *базовой несущей конструкции современной организации.* Концепция ориентирована на крупномасштабные организации, имеющие распределенную инфраструктуру, вне зависимости от того, является ли данная организация коммерческой (торговой, промышленной, многопрофильной) или относится к государственному сектору.

Для определенности рассмотрим крупную организацию (которую далее будем называть Корпорацией), нуждающуюся в построении информационной системы в целях эффективного управления. Предположим, что Корпорация представляет собой стабильную многопрофильную территориально распределенную структуру, обладающую всеми необходимыми системами жизнеобеспечения и функционирующую на принципах децентрализованного управления (последнее означает, что принятие решений оперативного и тактического характера делегировано на места и находится в компетенции подразделений, входящих в составКорпорации).

3.2. Характеристики

Попытаемся выделить основные характеристики Корпорации. В целом они типичны для представителя семейства больших организаций и представляют для нас интерес именно в этом качестве.

Масштабы и распределенная структура. Корпорация включает множество предприятий и организаций, расположенных по всей территории Российской Федерации, а также за ее пределами.

Широкий спектр подотраслей и направлений деятельности, подлежащих автоматизации. В рамках создания информационной системы Корпорации планируется автоматизировать целые направления ее деятельности, и в том числе, бухгалтерский учет, управление финансами, капитальное строительство и управление проектами, материально-техническое снабжение, управление производством и персоналом, внешнеэкономические связи и ряд других направлений.

Организационно-управленческая структура Корпорации. Предприятия и организации в составе Корпорации обладают определенной самостоятельностью в выработке и проведении технической политики собственной автоматизации.

Разнообразие парка вычислительных средств, сетевого оборудования и, в особенности, базового программного обеспечения.

Большое количество приложений специального назначения. В Корпорации эксплуатируется большое количество разнообразных приложений специального назначения, созданных на базе различного базового программного обеспечения.

Существует множество других, менее значимых характеристик, которые мы в данной статье рассматривать не будем.

3.3. Принципы построения КС

Что является основным при определении подходов к построению КС? Видимо, это два принципа:

- КС как стратегическая система жизнеобеспечения Корпорации;
- Основа КС - эффективная система централизованных коммуникаций

Суть первого принципа предельно проста. Не привлекая сложные экономические выкладки в целях технико-экономического обоснования необходимости построения информационной системы Корпорации, будем придерживаться следующей формулы. Предлагается рассматривать информационную систему Корпорации как одну из стратегических систем жизнеобеспечения, имеющую ключевое значение для ее эффективной деятельности. Такое определение делает ненужным многочисленные экономические расчеты по ожидаемой эффективности внедрения средств вычислительной техники. Опять-таки, будем реалистами и признаем, что такое внедрение не будет иметь моментального прямого эффекта - ни в денежном выражении, ни в сокращении персонала, ни в чем другом. Просто примем на веру, что информационная система - это в каком-то смысле аналог сети электропитания, телефонной системы, системы пожарной безопасности и т.п. Информационная система просто должна быть - и все.

Второй принцип нуждается в некоторых пояснениях. Известный американский специалист в области Intranet Стивен Теллин в работе [1] предлагает простую классификацию систем, исходя из двух их аспектов - коммуникаций и управления. Стивен Теллин отмечает, что до последнего времени для большинства крупных организаций, связанных с бизнесом, некоммерческих или правительственных, была характерна структура с централизованным управлением и централизованными коммуникациями (так называемая "пирамидальная" структура). Однако ряд сверхбольших организаций в силу своих размеров и масштабов деятельности было бы правильным рассматривать как структуры с распределенным управлением и централизованными коммуникациями. В этот ряд попадает и рассматриваемая организация.

По Теллину, для структур такого класса ключевым фактором эффективного контроля, координации и стратегического управления является эффективная система централизованных коммуникаций, которой и является Корпоративная Сеть.

Корпоративная Сеть

В терминах теории систем информационная система Корпорации - это *сложная система, ориентированная на цели*. Следуя теории систем и учитывая существенно *распределенный характер* данной системы, мы делаем вывод о том, что в ее основу должен быть положен принцип *централизованных коммуникаций и координации*, в сжатом виде изложенный в работе [1].

Действительно, как уже указывалось выше, Корпорация состоит из множества предприятий и организаций, обладающих весьма высокой степенью самостоятельности. В то же время в своей деятельности она ориентируется на вполне конкретные цели. Чтобы обеспечить их достижение, в своем развитии Корпорация нуждается в исключительно четко организованной *координации* деятельности входящих в ее состав предприятий и организаций. Такая координация, в свою очередь, возможна только на основе эффективной *системы централизованных коммуникаций (Корпоративная Сеть)*.

4.2. Техническая политика и стандарты

Ключевым фактором построения системы централизованных коммуникаций и координации является единая техническая политика. Именно она предопределяет возможность сопряжения различных подсистем информационной системы. Именно она позволяет сформировать единый взгляд на систему и ее архитектуру и разработать общий язык для ее определения и описания. С практической точки зрения единая техническая политика выражается, прежде всего, в корпоративных стандартах и принимает силу технического закона, действующего для всех без исключения подразделений Корпорации. Единая техническая политика предотвращает "волюнтаризм" в выборе программно-аппаратного обеспечения и сводит на нет попытки несанкционированной рационализации, периодически предпринимаемые техническими специалистами на местах.

4.3. Принципы построения

Существует несколько базовых принципов построения Сети.

Всеобъемлющий характер. Область действия Сети распространяется на Корпорацию в целом. Нет такого подразделения Корпорации, которое не было бы подключено к ней.

Интеграция. Корпоративная Сеть предоставляет возможность доступа ее пользователей к любым данным и приложениям (разумеется, в рамках политики информационной безопасности). Нет такого информационного ресурса, доступ к которому нельзя было бы получить по Сети.

Глобальный характер. Корпоративная Сеть - это глобальный взгляд на Корпорацию вне физических или политических границ. Сеть позволяет получить практически любую информацию о жизнедеятельности организации. Ее объем существенно выше, а спектр - неизмеримо шире, чем, например, информации в рамках локальной сети одного из подразделений Корпорации.

Адекватные эксплуатационные характеристики. Сеть обладает свойством управляемости и имеет высокий уровень RAS (reliability, availability, serviceability) - безотказность, живучесть, обслуживаемость при поддержке критически важных для деятельности Корпорации приложений.

Архитектура Корпоративной Сети

Общее представление

Корпоративная Сеть - это инфраструктура организации, поддерживающая решение актуальных задач и обеспечивающая достижение ее целей (то есть выполнение *миссии* организации). Она объединяет в единое пространство информационные системы всех объектов Корпорации. Корпоративная Сеть создается в качестве системно-технической основы информационной системы, как ее главный системообразующий компонент, на базе которого конструируются другие подсистемы.

Корпоративную Сеть необходимо рассматривать в различных аспектах. Общее представление о Сети складывается из проекций, получаемых в результате ее рассмотрения с различных точек зрения.

Корпоративная Сеть задумана и проектируется в единой системе координат, основу которой составляет понятия *системно-технической инфраструктуры* (структурный аспект), *системной функциональности* (сервисы и приложения) и *эксплуатационных характеристик* (свойства и службы). Каждое понятие находит свое отражение в том или ином компоненте Сети и реализуется в конкретных технических решениях.

С функциональной точки зрения Сеть - это эффективная среда передачи актуальной информации, необходимой для решения задач Корпорации. С системно-технической точки зрения Сеть представляет собой целостную структуру, состоящую из нескольких взаимосвязанных и взаимодействующих уровней:

- интеллектуальное здание;
- компьютерная сеть;
- телекоммуникации;
- компьютерные платформы;
- программное обеспечение промежуточного слоя (middleware);
- приложения.

С точки зрения системной функциональности Корпоративная Сеть выглядит как единое целое, предоставляющее пользователям и программам набор полезных в работе услуг (*сервисов*), общесистемных и специализированных *приложений*, обладающее набором полезных качеств (*свойств*) и содержащее в себе *службы*, гарантирующее нормальное функционирование Сети. Ниже будет дана краткая характеристика сервисов, приложений, свойств и служб.

5.2. Сервисы

Одним из принципов, положенных в основу создания Сети, является максимальное использование *типовых решений*, стандартных *унифицированных компонентов*. Конкретизируя этот принцип применительно к прикладному ПО, можно выделить ряд универсальных сервисов, которые целесообразно сделать базовыми компонентами приложений. Такими сервисами являются сервис СУБД, файловый сервис, информационный сервис (Web-сервис), электронная почта, сетевая печать и другие.

Особо отметим, что основным средством для построения прикладных и системных сервисов является ПО промежуточного слоя. В данной статье ПО промежуточного слоя принято в трактовке Филиппа Бернстайна, то есть так, как это изложено в работе [2]. Напомним, что в этой трактовке в ПО промежуточного слоя включено все, что находится между платформой (компьютер плюс операционная система) и приложениями. То есть Бернстайн включает в ПО промежуточного слоя, например, и СУБД.

Понятие сервисов ПО промежуточного слоя исключительно полезно при проработке архитектуры КС. Фактически, программная инфраструктура КС представляется многослойной, где каждый слой суть совокупность сервисов ПО промежуточного слоя. Нижние слои составляют низкоуровневые сервисы, такие как сервис имен, сервис регистрации, сетевой сервис и т.д. Вышележащие слои включают сервисы управления документами, сервисы управления сообщениями, сервисы событий и так далее. Верхний слой представляет собой сервисы, к которым опосредованно (через приложения) обращаются пользователи.

Здесь уместна аналогия с телефонной службой. Если пользователь нуждается в получении определенной услуги от информационной системы, то он должен программно подключиться к соответствующему сервису. Для этого он должен установить на свой компьютер приложение, которое такое подключение обеспечивает, и запросить от системного администратора выполнения административных действий. Например, если пользователь подключается к электронной почте, он должен установить приложение-клиент электронной почты, и системный администратор должен зарегистрировать нового пользователя. Точно так же сотрудник организации, желающий подключиться к телефонной сети, попросту должен подключить телефонный аппарат к розетке (предварительно затребовав от системного администратора выполнения соответствующих действий).

Проект КС исключительно удобно описывть в терминах сервисов. Так, например, политику информационной безопасности целесообразно строить, исходя их потребности в защите существующих и вводимых в действие сервисов. Подробнее об этом можно прочесть в работе [3].

5.3. Приложения

К *общесистемным приложениям* относят средства автоматизации индивидуального труда, используемые разнообразными категориями пользователей и ориентированные на решение типичных офисных задач. Это - текстовые процессоры, электронные таблицы, графические редакторы, календари, записные книжки и т.д. Как правило, общесистемные приложения представляют собой тиражируемые локализованные программные продукты, несложные в освоении и простые в использовании, ориентированные на конечных пользователей.

Специализированные приложения направлены на решение задач, которые невозможно или технически сложно автоматизировать с помощью общесистемных приложений. Как правило, специализированные приложения либо приобретаются у компаний-разработчиков, специализирующихся в своей деятельности на конкретную сферу, либо создаются компаниями-разработчиками по заказу организации, либо разрабатываются силами самой организации. В большинстве случаев специализированные приложения обращаются в процессе работы к общесистемным сервисам, таким, например, как файловый сервис, СУБД, электронная почта и т.д. Собственно, специализированные приложения, рассматриваемые в совокупности в масштабах Корпорации, как раз и определяют весь спектр прикладной функциональности.

5.4. Свойства и службы

Как уже говорилось выше, срок службы системно-технической инфраструктуры в несколько раз больше, чем у приложений. Корпоративная Сеть обеспечивает возможность развертывания новых приложений и их эффективное функционирование при сохранении инвестиций в нее, и в этом смысле должна обладать свойствами открытости (следование перспективным стандартам), производительности и сбалансированности, масштабируемости, высокой готовности, безопасности, управляемости.

Перечисленные выше свойства, по сути, представляют собой *эксплуатационные характеристики* создаваемой информационной системы и определяются в совокупности качеством продуктов и решений, положенных в ее основу.

Профессионально выполненная интеграция компонентов информационной системы (*системное конструирование*) гарантирует, что она будет обладать заранее заданными свойствами. Эти свойства вытекают также из высоких эксплуатационных характеристик (свойств) сервисов ПО промежуточного слоя. Бернстайн называет их *диффузионными* свойствами, имея в виду, что они "проникают" или "распространяются" снизу-вверх по слоям ПО промежуточного слоя и гарантируют высокое качество сервисов верхнего уровня. Здесь уместна аналогия со зданием, высокие эксплуатационные характеристики которого определяются в том числе и качеством его фундамента.

Разумеется, хорошие показатели по конкретным свойствам будут достигаться за счет грамотных технических решений системного конструирования.

Так, система будет обладать свойствами *безопасности, высокой готовности и управляемости* за счет реализации в проекте Корпоративной Сети соответствующих служб.

Масштабируемость в контексте компьютерных платформ (например, для серверной платформы) означает возможность адекватного наращивания мощностей компьютера (производительности, объема хранимой информации и т.д.) и достигается такими качествами линии серверов, как плавное наращивание мощности от модели к модели, единая операционная система для всех моделей, удобная и продуманная политика модификации младших моделей в направлении старших (upgrade) и т.д.

Общесистемные службы - это совокупность средств, не направленных напрямую на решение прикладных задач, но необходимых для обеспечения нормального функционирования информационной системы Корпорации. В качестве обязательных в Корпоративную Сеть должны быть включены службы информационной безопасности, высокой готовности, централизованного мониторинга и администрирования.

В составе информационных систем можно выделить две относительно независимых составляющие. Первая представляет собой собственно *компьютерную инфраструктуру* организации в широком смысле этого слова (сетевая, телекоммуникационная, программная, информационная, организационная инфраструктура - то есть то, что носит в статье обобщенное название *Корпоративная Сеть*). Вторая составляющая суть взаимосвязанные функциональные подсистемы, обеспечивающие решение задач организации и достижение ее целей. Если первая отражает системно-техническую, структурную сторону любой информационной системы, то вторая целиком относится к прикладной области и сильно зависит от специфики задач организации и ее целей.

Первая составляющая представляет собой базис, основу для интеграции функциональных подсистем и целиком определяет свойства информационной системы, важные для ее успешной эксплуатации. Требования к ней едины и стандартизованы, а методы ее построения хорошо известны и многократно проверены на практике.

Вторая составляющая строится целиком на базе первой и привносит в информационную систему прикладную функциональность. Требования к ней сложны и зачастую противоречивы, так как

выдвигаются специалистами из различных прикладных областей. Однако эта составляющая в конечном счете более важна для функционирования организации, так как ради нее, собственно, и строится вся инфраструктура.

2.2. Соотношение

Между двумя составляющими информационной системы можно проследить следующие взаимосвязи.

Составляющие независимы в определенном смысле. Организация будет эксплуатировать высокоскоростную сеть 100 MB Ethernet вне зависимости от того, какие методы и программы для организации бухгалтерского учета планируется принять на вооружение. Сеть организации будет построена на базе протокола TCP/IP независимо от того, какой текстовый процессор будет принять в качестве стандартного. Иными словами, в современных условиях базовая инфраструктура становится все более универсальной.

Составляющие зависимы в определенном смысле. Вторая невозможна без первой, первая без второй ограничена, поскольку лишена необходимой функциональности. Невозможно эксплуатировать прикладную систему с архитектурой клиент-сервер, когда отсутствует или некачественно построена сетевая инфраструктура. Однако, имея развитую инфраструктуру, можно предоставить сотрудникам организации ряд полезных общесистемных сервисов, (например, электронную почту), упрощающих работу и делающих ее эффективной (в нашем примере - за счет электронных коммуникаций). Если выбран этот эволюционный путь развития информационной системы, то в процессе своего развития Корпоративная Сеть постепенно приобретает ряд прикладных сервисов, направленных на решение универсальных задач организации - задач управления и координации.

2.3. Изменчивость

Вторая составляющая более изменчива. Действительно, инфраструктура организации зависит только от территориального расположения ее подразделений, да и то скорее в отношении инфраструктуры, никак не влияя на используемые для ее построения технологии. Вторая составляющая сильно зависит от организационно-управленческой структуры организации, ее функциональности, распределения функций, принятых в организации финансовых технологий и схем, существующей технологии документооборота и множества других факторов.

Первая составляющая имеет долговременный характер. Инфраструктура создается на многие годы вперед - так как капитальные затраты на ее создание настолько велики, что практически исключают возможность полной или частичной переделки уже построенного. Напротив, вторая составляющая изменчива по своей природе, так как в предметной части деятельности организации постоянно происходят более или менее существенные подвижки, которые должны быть отражены и в функциональных подсистемах. Этот тезис особенно актуален в контексте постоянно происходящих изменений в административных структурах многих отечественных организаций.

Степень определенности в выборе технологических решений для первой составляющей несколько выше, чем для второй. Действительно, современные компьютерные технологии предлагают такие промышленные решения для построения инфраструктуры организации, которые гарантировано обеспечат непрерывное развитие и совершенствование системно-технической базы информационной системы с перспективой на многие годы вперед. Первая составляющая имеет более отношение к технике, чем к экономике и управлению, и в этом смысле более стабильна, а ее развитие является более прогнозируемым и управляемым.

2.4. Что первично?

До недавнего времени в технологии создания информационных систем доминировал традиционный подход, когда вся архитектура информационной системы строилась "сверху-вниз" - от прикладной функциональности к системно-техническим решениям и первая составляющая информационной системы целиком выводилась из второй.

Практика многих больших российских проектов показала, что начинать построение КС только с анализа бизнес-процессов (не уделяя должного внимания инфраструктуре), весьма и весьма проблематично. Автоматизация деятельности корпорации на основе концепции "сверху-вниз" и принципов BPR (Business Process Reengineering) предполагает такую реорганизацию КС, которая наилучшим образом служит решению управленческих задач. Проблема заключается в том, что в современных российских условиях - условиях сверхдинамичного бизнеса, постоянно возникающих форс-мажорных обстоятельств и исключительно быстро меняющихся правил игры (социальных, политических, экономических), в рамках которой строится вся прикладная функциональность (как раз

и обеспечивающая решение управленческих задач) - систематизация управленческой деятельности представляет собой весьма сложную задачу ввиду высокой степени неопределенности.
В то же время бессмысленно строить инфраструктуру, не обращая внимания на прикладную функциональность. Если в процессе создания системно-технической инфраструктуры не проводить анализ и автоматизацию управленческих задач, то инвестированные в нее средства не дадут впоследствии реальной отдачи. Аппаратное и программное обеспечение инфраструктуры будет "висеть мертвым грузом" на плечах организации, требуя ежегодных затрат на сопровождение и модернизацию. Подход к построению КС "снизу-вверх" (с акцентом на системно-техническую инфраструктуру) вряд ли можно рассматривать в качестве магистрального.
В настоящее время развивается комбинированный подход, который можно характеризовать как "встречное движение": компьютерная инфраструктура и системная функциональность строятся так, чтобы в максимальной степени обеспечить изменчивость на уровне прикладной функциональности. Параллельно проводится анализ и структуризация бизнесс-процессов, сопровождающиеся внедрением соответствующих программных решений, привносящих в КС прикладную функциональность.

2.5. Выводы

Опираясь на сказанное выше, рискнем сделать следующий вывод. Разработку информационной системы целесообразно начинать с построения компьютерной инфраструктуры (Корпоративной Сети) как наиболее важной (фундаментальной) системообразующей составляющей, опирающейся на апробированные промышленные технологии и гарантировано реализуемой в разумные сроки в силу высокой степени определенности как в постановке задачи, так и в предлагаемых решениях. Одновременно, в контексте архитектуры Корпоративной Сети, как единого обобщенного взгляда на фундамент информационной системы, на наиболее важных и ответственных участках целесообразно выполнять разработки, насыщающие систему прикладной функциональностью (то есть внедрять системы финансового учета, управления кадрами и т.д.). Далее, прикладные программные системы будут распространены и на другие, первоначально менее значимые области управленческой деятельности.
В этом контексте особенно важными становятся:

- Широкий спектр готовых к применению промышленных прикладных систем для различных областей управленческой деятельности (как правило, поставляемых одной компанией);
- Высокая степень гранулярности таких решений (не обязательно внедрять сразу всю систему целиком - можно начать с отдельных участков);
- Построение на основе единого системного фундамента (как правило, в качестве фундамента выступает современная реляционная СУБД).

Подобный эволюционный подход, опирающийся на корпоративные стандарты, в конечном счете позволить построить реальную КС.

3. Корпорация

3.1. Определение

Предлагаемая вниманию читателя концепция опирается на обобщенное понятие Корпоративной Сети как *базовой несущей конструкции современной организации.* Концепция ориентирована на крупномасштабные организации, имеющие распределенную инфраструктуру, вне зависимости от того, является ли данная организация коммерческой (торговой, промышленной, многопрофильной) или относится к государственному сектору.
Для определенности рассмотрим крупную организацию (которую далее будем называть Корпорацией), нуждающуюся в построении информационной системы в целях эффективного управления. Предположим, что Корпорация представляет собой стабильную многопрофильную территориально распределенную структуру, обладающую всеми необходимыми системами жизнеобеспечения и функционирующую на принципах децентрализованного управления (последнее означает, что принятие решений оперативного и тактического характера делегировано на места и находится в компетенции подразделений, входящих в составКорпорации).

3.2. Характеристики

Попытаемся выделить основные характеристики Корпорации. В целом они типичны для представителя семейства больших организаций и представляют для нас интерес именно в этом качестве.

Масштабы и распределенная структура. Корпорация включает множество предприятий и организаций, расположенных по всей территории Российской Федерации, а также за ее пределами.
Широкий спектр подотраслей и направлений деятельности, подлежащих автоматизации. В рамках создания информационной системы Корпорации планируется автоматизировать целые направления ее деятельности, и в том числе, бухгалтерский учет, управление финансами, капитальное строительство и управление проектами, материально-техническое снабжение, управление производством и персоналом, внешнеэкономические связи и ряд других направлений.
Организационно-управленческая структура Корпорации. Предприятия и организации в составе Корпорации обладают определенной самостоятельностью в выработке и проведении технической политики собственной автоматизации.
Разнообразие парка вычислительных средств, сетевого оборудования и, в особенности, базового программного обеспечения.
Большое количество приложений специального назначения. В Корпорации эксплуатируется большое количество разнообразных приложений специального назначения, созданных на базе различного базового программного обеспечения.
Существует множество других, менее значимых характеристик, которые мы в данной статье рассматривать не будем.
3.3. Принципы построения КС
Что является основным при определении подходов к построению КС? Видимо, это два принципа:

- КС как стратегическая система жизнеобеспечения Корпорации;
- Основа КС - эффективная система централизованных коммуникаций

Суть первого принципа предельно проста. Не привлекая сложные экономические выкладки в целях технико-экономического обоснования необходимости построения информационной системы Корпорации, будем придерживаться следующей формулы. Предлагается рассматривать информационную систему Корпорации как одну из стратегических систем жизнеобеспечения, имеющую ключевое значение для ее эффективной деятельности. Такое определение делает ненужным многочисленные экономические расчеты по ожидаемой эффективности внедрения средств вычислительной техники. Опять-таки, будем реалистами и признаем, что такое внедрение не будет иметь моментального прямого эффекта - ни в денежном выражении, ни в сокращении персонала, ни в чем другом. Просто примем на веру, что информационная система - это в каком-то смысле аналог сети электропитания, телефонной системы, системы пожарной безопасности и т.п. Информационная система просто должна быть - и все.
Второй принцип нуждается в некоторых пояснениях. Известный американский специалист в области Intranet Стивен Теллин в работе [1] предлагает простую классификацию систем, исходя из двух их аспектов - коммуникаций и управления. Стивен Теллин отмечает, что до последнего времени для большинства крупных организаций, связанных с бизнесом, некоммерческих или правительственных, была характерна структура с централизованным управлением и централизованными коммуникациями (так называемая "пирамидальная" структура). Однако ряд сверхбольших организаций в силу своих размеров и масштабов деятельности было бы правильным рассматривать как структуры с распределенным управлением и централизованными коммуникациями. В этот ряд попадает и рассматриваемая организация.
По Теллину, для структур такого класса ключевым фактором эффективного контроля, координации и стратегического управления является эффективная система централизованных коммуникаций, которой и является Корпоративная Сеть.
4. Корпоративная Сеть
4.1. Определение
В терминах теории систем информационная система Корпорации - это *сложная система, ориентированная на цели*. Следуя теории систем и учитывая существенно *распределенный характер* данной системы, мы делаем вывод о том, что в ее основу должен быть положен принцип *централизованных коммуникаций и координации*, в сжатом виде изложенный в работе [1].
Действительно, как уже указывалось выше, Корпорация состоит из множества предприятий и организаций, обладающих весьма высокой степенью самостоятельности. В то же время в своей деятельности она ориентируется на вполне конкретные цели. Чтобы обеспечить их достижение, в своем развитии Корпорация нуждается в исключительно четко организованной *координации*

деятельности входящих в ее состав предприятий и организаций. Такая координация, в свою очередь, возможна только на основе эффективной *системы централизованных коммуникаций (Корпоративная Сеть)*.

4.2. Техническая политика и стандарты

Ключевым фактором построения системы централизованных коммуникаций и координации является единая техническая политика. Именно она предопределяет возможность сопряжения различных подсистем информационной системы. Именно она позволяет сформировать единый взгляд на систему и ее архитектуру и разработать общий язык для ее определения и описания. С практической точки зрения единая техническая политика выражается, прежде всего, в корпоративных стандартах и принимает силу технического закона, действующего для всех без исключения подразделений Корпорации. Единая техническая политика предотвращает "волюнтаризм" в выборе программно-аппаратного обеспечения и сводит на нет попытки несанкционированной рационализации, периодически предпринимаемые техническими специалистами на местах.

4.3. Принципы построения

Существует несколько базовых принципов построения Сети.

Всеобъемлющий характер. Область действия Сети распространяется на Корпорацию в целом. Нет такого подразделения Корпорации, которое не было бы подключено к ней.

Интеграция. Корпоративная Сеть предоставляет возможность доступа ее пользователей к любым данным и приложениям (разумеется, в рамках политики информационной безопасности). Нет такого информационного ресурса, доступ к которому нельзя было бы получить по Сети.

Глобальный характер. Корпоративная Сеть - это глобальный взгляд на Корпорацию вне физических или политических границ. Сеть позволяет получить практически любую информацию о жизнедеятельности организации. Ее объем существенно выше, а спектр - неизмеримо шире, чем, например, информации в рамках локальной сети одного из подразделений Корпорации.

Адекватные эксплуатационные характеристики. Сеть обладает свойством управляемости и имеет высокий уровень RAS (reliability, availability, serviceability) - безотказность, живучесть, обслуживаемость при поддержке критически важных для деятельности Корпорации приложений.

5. Архитектура Корпоративной Сети

5.1. Общее представление

Корпоративная Сеть - это инфраструктура организации, поддерживающая решение актуальных задач и обеспечивающая достижение ее целей (то есть выполнение *миссии* организации). Она объединяет в единое пространство информационные системы всех объектов Корпорации. Корпоративная Сеть создается в качестве системно-технической основы информационной системы, как ее главный системообразующий компонент, на базе которого конструируются другие подсистемы.

Корпоративную Сеть необходимо рассматривать в различных аспектах. Общее представление о Сети складывается из проекций, получаемых в результате ее рассмотрения с различных точек зрения.

Корпоративная Сеть задумана и проектируется в единой системе координат, основу которой составляет понятия *системно-технической инфраструктуры* (структурный аспект), *системной функциональности* (сервисы и приложения) и *эксплуатационных характеристик* (свойства и службы). Каждое понятие находит свое отражение в том или ином компоненте Сети и реализуется в конкретных технических решениях.

С функциональной точки зрения Сеть - это эффективная среда передачи актуальной информации, необходимой для решения задач Корпорации. С системно-технической точки зрения Сеть представляет собой целостную структуру, состоящую из нескольких взаимосвязанных и взаимодействующих уровней:

- интеллектуальное здание;
- компьютерная сеть;
- телекоммуникации;
- компьютерные платформы;
- программное обеспечение промежуточного слоя (middleware);
- приложения.

С точки зрения системной функциональности Корпоративная Сеть выглядит как единое целое, предоставляющее пользователям и программам набор полезных в работе услуг (*сервисов*), общесистемных и специализированных *приложений*, обладающее набором полезных качеств

(*свойств*) и содержащее в себе *службы*, гарантирующее нормальное функционирование Сети. Ниже будет дана краткая характеристика сервисов, приложений, свойств и служб.

5.2. Сервисы

Одним из принципов, положенных в основу создания Сети, является максимальное использование *типовых решений*, стандартных *унифицированных компонентов*. Конкретизируя этот принцип применительно к прикладному ПО, можно выделить ряд универсальных сервисов, которые целесообразно сделать базовыми компонентами приложений. Такими сервисами являются сервис СУБД, файловый сервис, информационный сервис (Web-сервис), электронная почта, сетевая печать и другие.

Особо отметим, что основным средством для построения прикладных и системных сервисов является ПО промежуточного слоя. В данной статье ПО промежуточного слоя принято в трактовке Филиппа Бернстайна, то есть так, как это изложено в работе [2]. Напомним, что в этой трактовке в ПО промежуточного слоя включено все, что находится между платформой (компьютер плюс операционная система) и приложениями. То есть Бернстайн включает в ПО промежуточного слоя, например, и СУБД.

Понятие сервисов ПО промежуточного слоя исключительно полезно при проработке архитектуры КС. Фактически, программная инфраструктура КС представляется многослойной, где каждый слой суть совокупность сервисов ПО промежуточного слоя. Нижние слои составляют низкоуровневые сервисы, такие как сервис имен, сервис регистрации, сетевой сервис и т.д. Вышележащие слои включают сервисы управления документами, сервисы управления сообщениями, сервисы событий и так далее. Верхний слой представляет собой сервисы, к которым опосредованно (через приложения) обращаются пользователи.

Здесь уместна аналогия с телефонной службой. Если пользователь нуждается в получении определенной услуги от информационной системы, то он должен программно подключиться к соответствующему сервису. Для этого он должен установить на свой компьютер приложение, которое такое подключение обеспечивает, и запросить от системного администратора выполнения административных действий. Например, если пользователь подключается к электронной почте, он должен установить приложение-клиент электронной почты, и системный администратор должен зарегистрировать нового пользователя. Точно так же сотрудник организации, желающий подключиться к телефонной сети, попросту должен подключить телефонный аппарат к розетке (предварительно затребовав от системного администратора выполнения соответствующих действий).

Проект КС исключительно удобно описывть в терминах сервисов. Так, например, политику информационной безопасности целесообразно строить, исходя их потребности в защите существующих и вводимых в действие сервисов. Подробнее об этом можно прочесть в работе [3].

5.3. Приложения

К *общесистемным приложениям* относят средства автоматизации индивидуального труда, используемые разнообразными категориями пользователей и ориентированные на решение типичных офисных задач. Это - текстовые процессоры, электронные таблицы, графические редакторы, календари, записные книжки и т.д. Как правило, общесистемные приложения представляют собой тиражируемые локализованные программные продукты, несложные в освоении и простые в использовании, ориентированные на конечных пользователей.

Специализированные приложения направлены на решение задач, которые невозможно или технически сложно автоматизировать с помощью общесистемных приложений. Как правило, специализированные приложения либо приобретаются у компаний-разработчиков, специализирующихся в своей деятельности на конкретную сферу, либо создаются компаниями-разработчиками по заказу организации, либо разрабатываются силами самой организации. В большинстве случаев специализированные приложения обращаются в процессе работы к общесистемным сервисам, таким, например, как файловый сервис, СУБД, электронная почта и т.д. Собственно, специализированные приложения, рассматриваемые в совокупности в масштабах Корпорации, как раз и определяют весь спектр прикладной функциональности.

5.4. Свойства и службы

Как уже говорилось выше, срок службы системно-технической инфраструктуры в несколько раз больше, чем у приложений. Корпоративная Сеть обеспечивает возможность развертывания новых приложений и их эффективное функционирование при сохранении инвестиций в нее, и в этом смысле

должна обладать свойствами открытости (следование перспективным стандартам), производительности и сбалансированности, масштабируемости, высокой готовности, безопасности, управляемости.

Перечисленные выше свойства, по сути, представляют собой *эксплуатационные характеристики* создаваемой информационной системы и определяются в совокупности качеством продуктов и решений, положенных в ее основу.

Профессионально выполненная интеграция компонентов информационной системы (*системное конструирование*) гарантирует, что она будет обладать заранее заданными свойствами. Эти свойства вытекают также из высоких эксплуатационных характеристик (свойств) сервисов ПО промежуточного слоя. Бернстайн называет их *диффузионными* свойствами, имея в виду, что они "проникают" или "распространяются" снизу-вверх по слоям ПО промежуточного слоя и гарантируют высокое качество сервисов верхнего уровня. Здесь уместна аналогия со зданием, высокие эксплуатационные характеристики которого определяются в том числе и качеством его фундамента.

Разумеется, хорошие показатели по конкретным свойствам будут достигаться за счет грамотных технических решений системного конструирования.

Так, система будет обладать свойствами *безопасности, высокой готовности и управляемости* за счет реализации в проекте Корпоративной Сети соответствующих служб.

Масштабируемость в контексте компьютерных платформ (например, для серверной платформы) означает возможность адекватного наращивания мощностей компьютера (производительности, объема хранимой информации и т.д.) и достигается такими качествами линии серверов, как плавное наращивание мощности от модели к модели, единая операционная система для всех моделей, удобная и продуманная политика модификации младших моделей в направлении старших (upgrade) и т.д.

Общесистемные службы - это совокупность средств, не направленных напрямую на решение прикладных задач, но необходимых для обеспечения нормального функционирования информационной системы Корпорации. В качестве обязательных в Корпоративную Сеть должны быть включены службы информационной безопасности, высокой готовности, централизованного мониторинга и администрирования.

25. Типовые удалённые атаки в Интернет и механизмы их реализации. Типовые уязвимости, позволяющие организовать удаленные атаки

Компьютерные сети проектируются (и создаются) на основе одних и тех же принципов, правил (шаблонов) и, следовательно, имеют практически одинаковые проблемы безопасности в сетевых информационных системах и можно ввести понятие типовой удаленной атаки.

Типовая удаленная атака – это удаленное информационное воздействие, программно осуществляемое по каналам связи и характерное для любой распределенной системы.

Распределённая система — система, для которой отношения местоположений элементов (или групп элементов) играют существенную роль с точки зрения функционирования системы, а, следовательно, и с точки зрения анализа и синтеза системы.

Рассмотрим типовые удаленные атаки и механизмы их реализации.

Удаленные угрозы можно классифицировать по следующим признакам.

- По характеру воздействия:
  - ☐ пассивные (класс 1.1);
  - ☐ активные (класс 1.2).

Пассивным воздействием на распределенную вычислительную систему называется воздействие, которое не оказывает непосредственного влияния на работу системы, но может нарушать ее политику безопасности. Именно отсутствие непосредственного влияния на работу сети приводит к тому, что пассивное удаленное воздействие практически невозможно обнаружить. Примером пассивного типового удаленного воздействия в вычислительных сетях является прослушивание канала связи в сети.

Под активным воздействием на вычислительную сеть понимается воздействие, оказывающее непосредственное влияние на работу сети (изменение конфигурации, нарушение работоспособности и т. д.) и нарушающее принятую в ней политику безопасности. Практически все типы удаленных угроз являются активными воздействиями. Это связано с тем, что в самой природе разрушающего воздействия содержится активное начало. Очевидной особенностью активного воздействия по сравнению с пассивным является принципиальная возможность его обнаружения, так как в результате его осуществления в системе происходят определенные изменения. В отличие от активного, при пассивном воздействии не остается никаких следов (просмотр чужого сообщения ничего не меняет).

- По цели воздействия:
  - ☐ нарушение конфиденциальности информации (класс 2.1);
  - ☐ нарушение целостности информации (класс 2.2);
  - ☐ нарушение доступности информации (работоспособности системы) (класс 2.3).

Этот классификационный признак является прямой проекцией трех основных типов угроз — раскрытия, целостности и отказа в обслуживании.

Одна из основных целей злоумышленников — получение несанкционированного доступа к информации. Существуют две принципиальные возможности доступа к информации: перехват и искажение. Возможность перехвата информации означает получение к ней доступа, но невозможность ее модификации. Следовательно, перехват информации ведет к нарушению ее конфиденциальности. Примером перехвата информации может служить прослушивание канала в сети. В этом случае имеется несанкционированный доступ к информации без возможности ее искажения. Очевидно также, что нарушение конфиденциальности информации является пассивным воздействием.

Возможность искажения информации означает либо полный контроль над информационным потоком между объектами системы, либо возможность передачи сообщений от имени другого объекта. Таким образом, очевидно, что искажение информации ведет к нарушению ее целостности. Данное информационное разрушающее воздействие представляет собой яркий пример активного воздействия. Примером удаленной угрозы, цель которой нарушение целостности информации, может служить типовая удаленная атака «Ложный объект распределенной вычислительной сети».

Принципиально другая цель преследуется злоумышленником при реализации угрозы для нарушения работоспособности сети. В этом случае не предполагается получение несанкционированного доступа к информации. Его основная цель — добиться, чтобы узел сети или какой то из сервисов поддерживаемый им вышел из строя и для всех остальных объектов сети доступ к ресурсам атакованного объекта был бы невозможен. Примером удаленной атаки, целью которой

является нарушение работоспособности системы, может служить типовая удаленная атака «Отказ в обслуживании».

- По условию начала осуществления воздействия

Удаленное воздействие, также как и любое другое, может начать осуществляться только при определенных условиях. В вычислительных сетях можно выделить три вида условий начала осуществления удаленной атаки:

- ☐ атака по запросу от атакуемого объекта (класс 3.1);
- ☐ атака по наступлению ожидаемого события на атакуемом объекте (класс 3.2);
- ☐ безусловная атака (класс 3.3).

В первом случае, злоумышленик ожидает передачи от потенциальной цели атаки запроса определенного типа, который и будет условием начала осуществления воздействия. Примером подобных запросов в сети Internet служат DNS — запросы. Отметим, что данный тип удаленных атак наиболее характерен для распределенных вычислительных сетей.

Во втором случае, злоумышленник осуществляет постоянное наблюдение за состоянием операционной системы удаленной цели атаки и при возникновении определенного события в этой системе начинает воздействие. Как и в предыдущем случае, инициатором осуществления начала атаки выступает сам атакуемый объект.

Реализация третьего вида атаки не связана ни с какими событиями и реализуется безусловно по отношению к цели атаки, то есть атака осуществляется немедленно.

- По наличию обратной связи с атакуемым объектом:
  - ☐ с обратной связью (класс 4.1);
  - ☐ без обратной связи (однонаправленная атака) (класс 4.2).

Удаленная атака, осуществляемая при наличии обратной связи с атакуемым объектом, характеризуется тем, что на некоторые запросы, переданные на атакуемый объект, атакующему требуется получить ответ, а, следовательно, между атакующим и целью атаки существует обратная связь, которая позволяет атакующему адекватно реагировать на все изменения, происходящие на атакуемом объекте.

В отличие от атак с обратной связью удаленным атакам без обратной связи не требуется реагировать на какие-либо изменения, происходящие на атакуемом объекте. Атаки данного вида обычно осуществляются передачей на атакуемый объект одиночных запросов, ответы на которые атакующему не нужны. Подобную удаленную атаку можно называть однонаправленной удаленной атакой. Примером однонаправленных атак является типовая удаленная атака «отказ в обслуживании».

- По расположению субъекта атаки относительно атакуемого объекта:
  - ☐ внутрисегментное (класс 5.1);
  - ☐ межсегментное (класс 5.2).

Рассмотрим ряд определений:

Субъект атаки (или источник атаки) — это атакующая программа или злоумышленник, непосредственно осуществляющие воздействие.

Маршрутизатор (router) — устройство, обеспечивающее маршрутизацию пакетов обмена в глобальной сети.

Подсеть (subnetwork) (в терминологии Internet) — совокупность хостов, являющихся частью глобальной сети, для которых маршрутизатором выделен одинаковый номер подсети. Хосты внутри одной подсети могут взаимодействовать между собой непосредственно, минуя маршрутизатор.

Сегмент сети — физическое объединение хостов. Например, сегмент сети образуют совокупность хостов, подключенных к серверу по схеме «общая шина». При такой схеме подключения каждый хост имеет возможность подвергать анализу любой пакет в своем сегменте.

С точки зрения удаленной атаки чрезвычайно важно, как по отношению друг к другу располагаются субъект и объект атаки, то есть в одном или в разных сегментах они находятся. В случае внутрисегментной атаки, как следует из названия, субъект и объект атаки находятся в одном сегменте. При межсегментной атаке субъект и объект атаки находятся в разных сегментах.

Данный классификационный признак позволяет судить о так называемой «степени удаленности» атаки.

Важно отметить, что межсегментная удаленная атака представляет гораздо большую опасность, чем внутрисегментная. Это связано с тем, что в случае межсегментной атаки объект её и

непосредственно атакующий могут находиться на расстоянии многих тысяч километров друг от друга, что может существенно воспрепятствовать мерам по локализации субъекта атаки.

- По уровню модели ISO/OSI, на котором осуществляется воздействие:
  - ☐ физический (класс 6.1);
  - ☐ канальный (класс 6.2);
  - ☐ сетевой (класс 6.3);
  - ☐ транспортный (класс 6.4);
  - ☐ сеансовый (класс 6.5);
  - ☐ представительный (класс 6.6);
  - ☐ прикладной (класс 6.7).

Сетевая модель OSI — сетевая модель стека (магазина) сетевых протоколов OSI/ISO (ГОСТ Р ИСО/МЭК 7498-1-99). Посредством данной модели различные сетевые устройства могут взаимодействовать друг с другом. Модель определяет различные уровни взаимодействия систем. Каждый уровень выполняет определённые функции при таком взаимодействии.

Модель OSI				
Уровень (layer)		Тип данных (PDU[1])	Функции	Примеры
Host layers	7. Прикладной (application)	Данные	Доступ к сетевым службам	HTTP, FTP, POP3
	6. Представительский (представления) (presentation)		Представление ишифрование данных	ASCII, EBCDIC
	5. Сеансовый (session)		Управление сеансом связи	RPC, PAP
	4. Транспортный (transport)	Сегменты (segment) /Дейтаграммы(datagram)	Прямая связь между конечными пунктами и надёжность	TCP, UDP, SCTP, PORTS
Media layers	3. Сетевой (network)	Пакеты(packet)	Определение маршрута и логическая адресация	IPv4, IPv6, IPsec, AppleTalk
	2. Канальный (data link)	Биты (bit)/ Кадры (frame)	Физическая адресация	PPP, IEEE 802.22, Ethernet, DSL, ARP, L2TP, сетевая карта.
	1. Физический (physical)	Биты (bit)	Работа со средой передачи, сигналами и двоичными данными	USB, кабель ("витая пара", коаксиальный, оптоволоконный), радиоканал

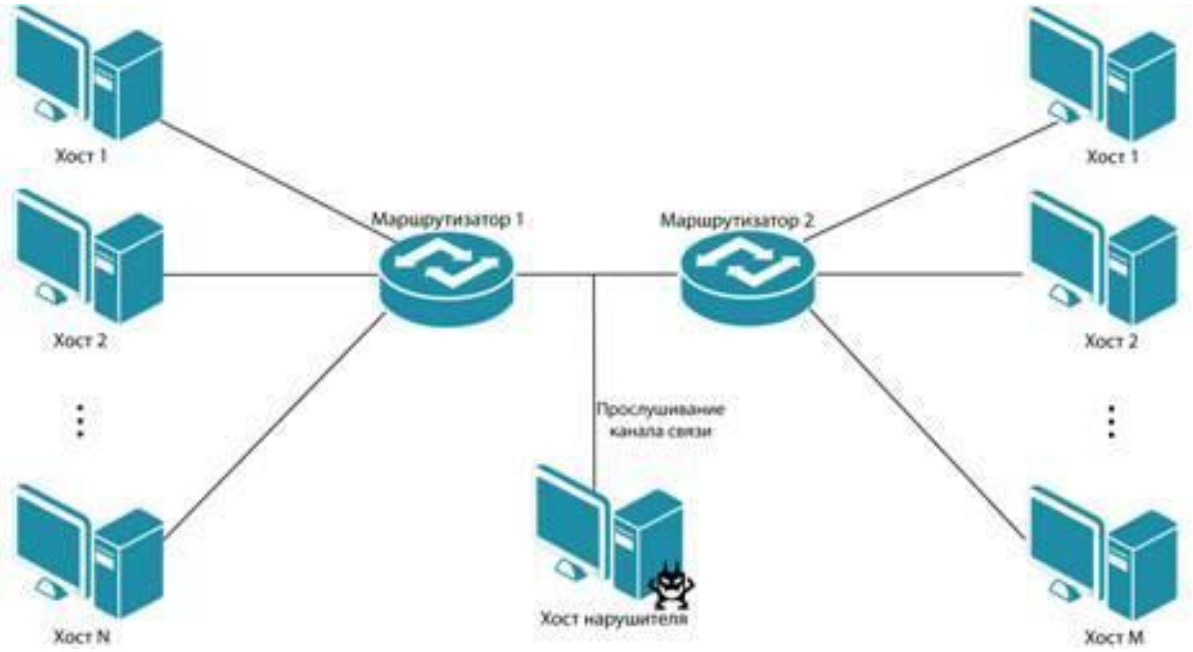
Удаленная атака "анализ сетевого трафика"

Основной особенностью распределенной вычислительной сети является распределенность ее объектов в пространстве и связь между ними по физическим линиям связи. При этом все управляющие сообщения и данные, пересылаемые между объектами вычислительной сети, передаются по сетевым соединениям в виде пакетов обмена. Эта особенность привела к появлению специфичного для

распределенных вычислительных сетей типового удаленного воздействия, заключающегося в прослушивании канала связи, называемого анализом сетевого трафика.

Анализ сетевого трафика позволяет:

- изучить логику работы распределенной вычислительной сети, это достигается путем перехвата и анализа пакетов обмена на канальном уровне (знание логики работы сети позволяет на практике моделировать и осуществлять другие типовые удаленные атаки);
- перехватить поток данных, которыми обмениваются объекты сети, т. е. удаленная атака данного типа заключается в получении несанкционированного доступа к информации, которой обмениваются пользователи (примером перехваченной при помощи данной типовой удаленной атаки информации могут служить имя и пароль пользователя, пересылаемые в незашифрованном виде по сети).



26.Обеспечение безопасности систем, входящих в состав глобальных сетей: межсетевые экраны, виртуальные частные сети

Под доверенным объектом понимается элемент сети (компьютер, межсетевой экран, маршрутизатор и т.п.), имеющий легальное подключение, и которому присвоены права для доступа к сетевым ресурсам информационной системы.

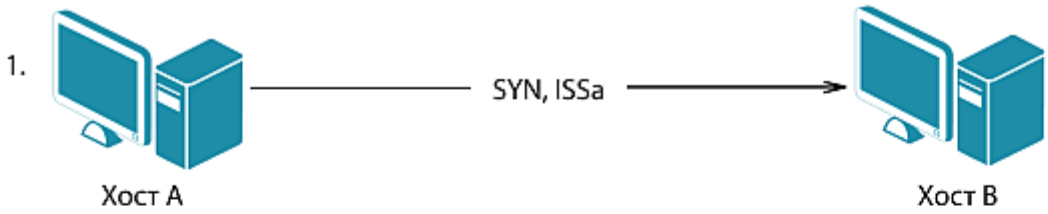
Осуществление атаки "подмена доверенного объекта сети" и передача по каналам связи сообщений от его имени с присвоением его прав доступа возможна в системах, где используются нестойкие алгоритмы идентификации и аутентификации хостов. Типичным примером является перехват TCP-сессии.

Протокол TCP является одним из базовых протоколов транспортного уровня сети Интернет. Он позволяет исправлять ошибки, которые могут возникнуть в процессе передачи пакетов, устанавливая логическое соединение – виртуальный канал. По этому каналу передаются и принимаются пакеты с регистрацией их последовательности, осуществляется управление информационным потоком, организовывается повторная передача искаженных пакетов, а в конце сеанса канал разрывается. Для идентификации TCP-пакета в TCP-заголовке существуют два 32-разрядных идентификатора – Sequence Number (номер последовательности) и Acknowledgment Number (номер подтверждения), которые также играют роль счетчиков пакетов

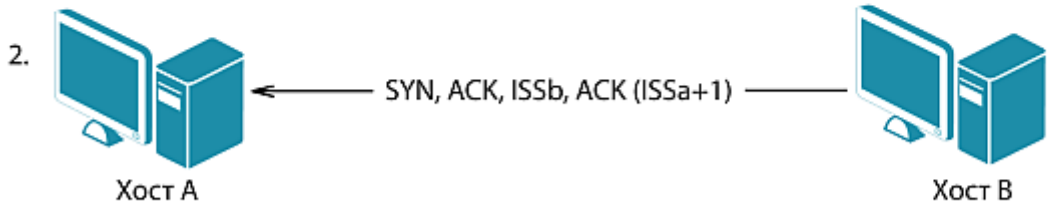
Существуют две разновидности процесса осуществления удаленной атаки типа "подмена доверенного объекта сети":

- атака с установлением виртуального канала;
- атака без установления виртуального канала.

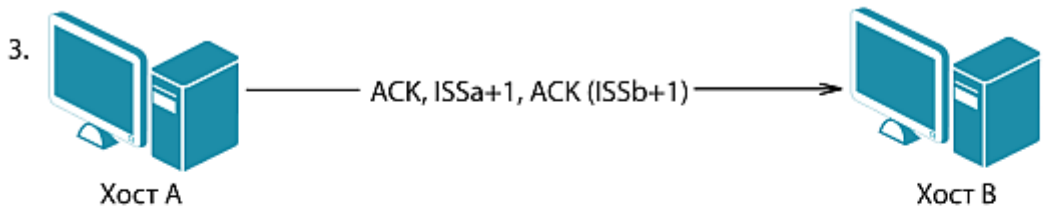
Процесс осуществления атаки с установлением виртуального канала состоит в присвоении прав доверенного пользователя, что позволяет злоумышленнику вести сеанс работы с объектами системы от имени доверенного пользователя. Для формирования ложного TCP-пакета атакующему достаточно подобрать соответствующие текущие значения идентификаторов TCP-пакета (ISSa и ISSb, см. рисунок 1.4) для данного TCP-соединения (например, FTP- или TELNET-подключение).



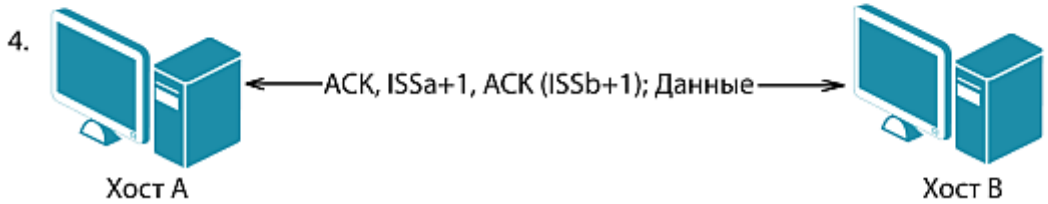
В сообщении, передаваемом хостом A установлен бит SYN (Synchronize Sequence Number), а в поле Sequence Number установлено начальное 32-битное значение ISSa (Initial Sequence Number).



Хост B посылает сообщение, в котором установлены бит SYN и бит ACK; в поле Sequence Number хостом B задается начальное значение счетчика ISSb; поле Acknowledgment Number содержит значение ISSa, увеличенное на 1.



В этом сообщении установлен бит ACK;
поле Sequence Number содержит ISSa+1;
поле Acknowledgment Number - ISSb+1.
TCP-соединение между хостами A и B считается установленным.



Хост A может посылать пакеты с данными на хост B.

Так как для служебных сообщений в распределенных сетях часто используется передача одиночных сообщений, не требующих подтверждения, виртуальное соединение не создается. Атака без установления виртуального канала заключается в передаче служебных сообщений от имени сетевых управляющих устройств (например, от имени маршрутизаторов) о ложном изменении маршрутно-адресных данных. Идентификация передаваемых сообщений осуществляется только по сетевому адресу отправителя, который легко подделать. Типовая удаленная атака, использующая навязывание ложного маршрута, основана на описанной идее.

Подмена доверенного объекта сети является активным воздействием, совершаемым с целью нарушения конфиденциальности и целостности информации. Данная удаленная атака может являться как внутрисегментной, так и межсегментной, как с обратной связью, так и без обратной связи с атакуемым объектом и осуществляется на сетевом и транспортном уровнях модели OSI.

Удаленная атака "ложный объект"

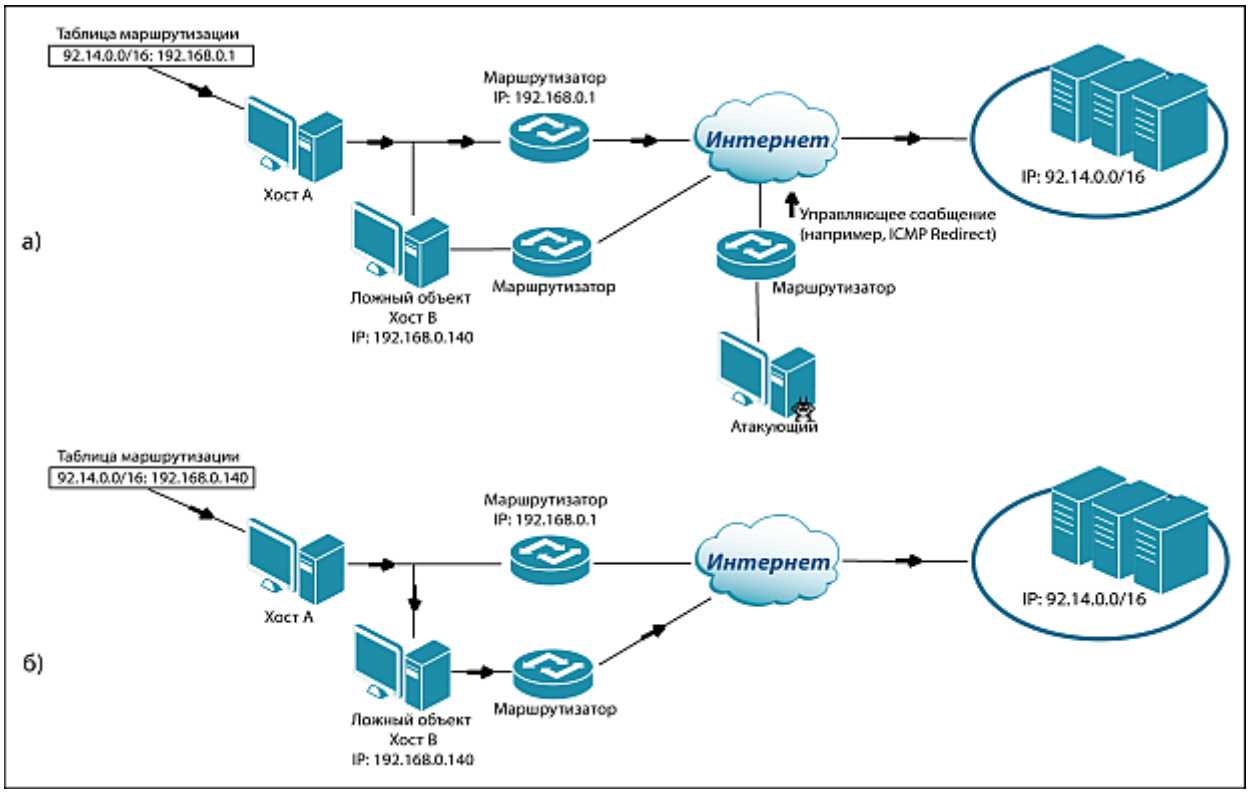
Архитектура Интернета создавалась в условиях, когда внутри сети существовало доверие к действиям отдельных участников. В распределенных сетях механизмы идентификации сетевых управляющих устройств (маршрутизаторов) не обеспечивают безопасное использование протоколов управления сетью. Если участник сети (маршрутизатор) заявляет, что он владеет блоком адресного пространства, остальная часть IP-сети верит ему на слово и адресует ему весь соответствующий трафик. Значит, можно создать любой сетевой блок и запустить его в IP-сети, придав анонимность любой атаке, связанной с изменением маршрутизации и внедрением в систему ложного объекта. Такой тип воздействия на сетевую информационную систему ещё называют атакой типа MITM (man in the middle, "человек посередине").

Для перехвата трафика злоумышленники используют уязвимости, присущие протоколам различных уровней стека TCP/IP: сетевому, транспортному и прикладному. На сегодняшний день в подавляющем большинстве применяются стандартные протоколы семейства TCP/IP, среди которых к наиболее уязвимым относятся следующие: протокол управления передачей TCP, межсетевого взаимодействия IP, эмуляции терминала Telnet, передачи файлов FTP, разрешения адресов ARP, службы доменных имен DNS, управляющих сообщений сети Интернет ICMP и сетевого управления SNMP. Кроме того, для обеспечения эффективной и оптимальной маршрутизации в сетях применяются динамические протоколы RIP и OSPF, позволяющие маршрутизаторам обмениваться информацией друг с другом и обновлять таблицы маршрутизации.

Атакующий ставит целью внедрение ложного объекта в сеть путем изменения таблиц маршрутизации и навязывания ложного маршрута. Основная задача злоумышленника – не только прервать сообщение между сетями, а в первую очередь перевести трафик через свой хост, чтобы извлечь полезную информацию.

Реализация атаки основывается на уязвимостях или ошибках настройки протоколов маршрутизации (RIP, OSPF) и управления сетью (ICMP, SNMP). При этом злоумышленник посылает в сеть управляющее сообщение от имени сетевого управляющего устройства (например, маршрутизатора). Рисунок 1.5 иллюстрирует реализацию удаленной атаки "навязывание ложного маршрута" с использованием протокола ICMP. Пакеты с запросами в сеть 92.14.0.0/16 с хоста A проходят через маршрутизирующее устройство с IP-адресом 192.168.0.1 (рисунок 1.5а). Атакующий посылает управляющее сообщение ICMP Redirect о наилучшем маршруте в сеть 92.14.0.0/16 и

получает возможность изменения таблиц маршрутизации хоста А. В результате весь трафик с хоста А, направляющийся в сеть 92.14.0.0/16, проходит через ложный объект хост В (рисунок 1.5б).



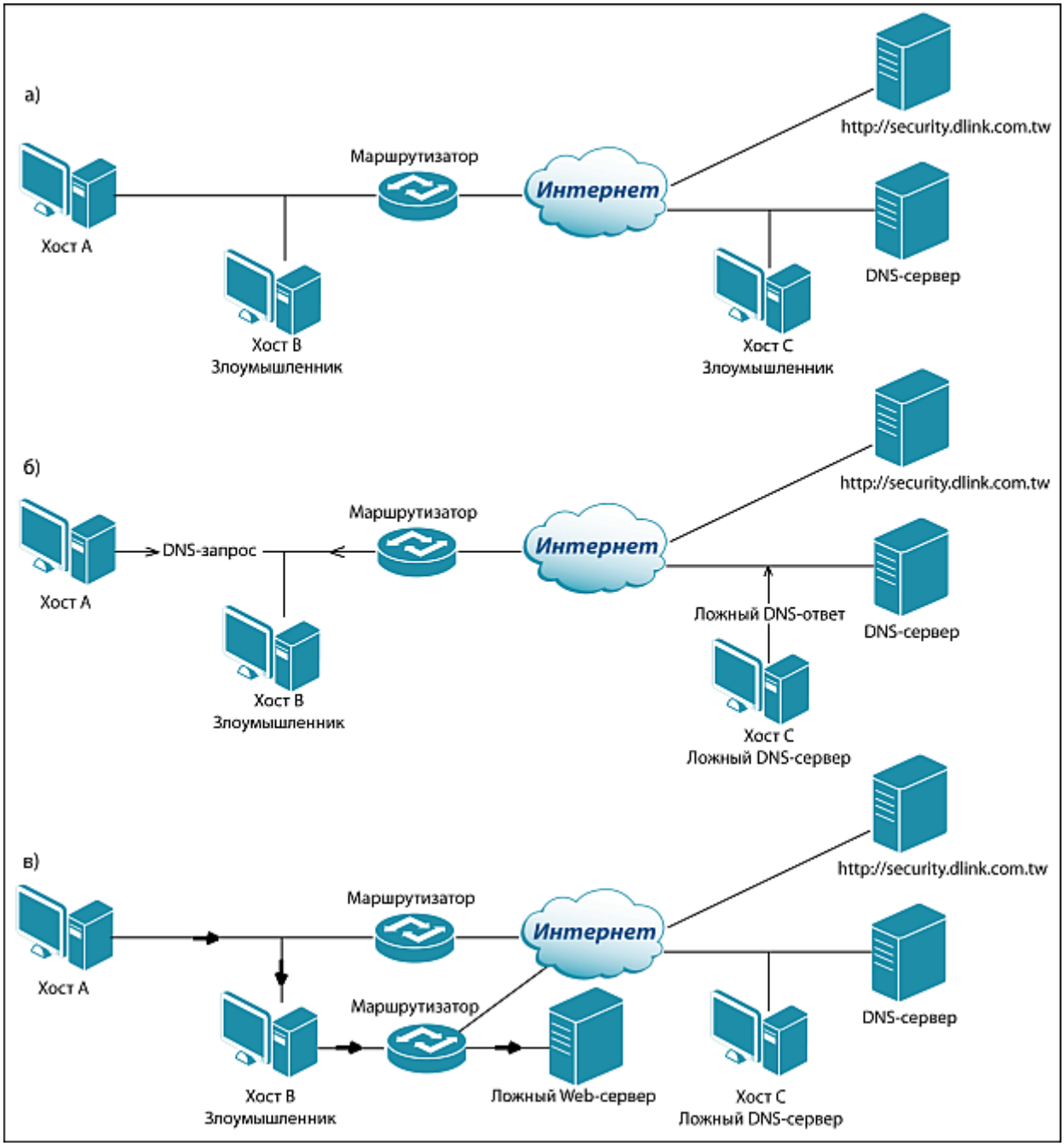
Относительно недавно разработчиками израильского центра Electronic Warfare Research and Simulation Center была обнаружена брешь в сетевом протоколе OSPF. Как утверждают исследователи, уязвимость существует из-за того, что сам протокол допускает прием поддельных запросов новых таблиц маршрутизации. Например, при помощи ноутбука, можно отправить периодический запрос Link State Advertisement (LSA) на обновление таблиц маршрутизации. После чего маршрутизатор опознает запрос как легальный, поскольку в подтверждение он проверяет лишь порядковые номера запросов, которые также можно подделать. В результате подобной манипуляции, у злоумышленника будет полный доступ к сети в течение примерно 15-ти минут, пока маршрутизатор опять не обновит таблицы.

Также успешной может оказаться удаленная атака, использующая уязвимости сервисов, установленных на хостах (серверах). Для преобразования адресов из одного формата в другой в распределенных сетях используются протоколы удаленного поиска, заключающиеся в передаче по сети специальных запросов и получения на них ответов с искомой информацией. Так, в сетях Ethernet протокол ARP решает вопрос отображения MAC-адреса (6 байтов) в пространство сетевых IP-адресов (4 байта) и наоборот; протокол DNS используется при преобразовании текстового доменного имени в IP-адрес. При этом существует возможность перехвата злоумышленником поискового запроса и выдачи на него ложного ответа, использование которого приведет к изменению маршрутно-адресных данных. В результате весь сетевой трафик жертвы будет проходить через ложный объект.

На рисунке 1.6 представлена схема реализации атаки "внедрение ложного DNS-сервера" путем перехвата DNS-запроса. Атакующий (может находится либо на хосте В, либо на хосте С) ожидает DNS-запрос от хоста А (рисунок 1.6а). После перехвата поискового запроса от хоста А, атакующий посылает ему ложный DNS-ответ (рисунок 1.6б). Особенно стоит отметить возможность преднамеренного искажения информации: вместо ресурса http://security.dlink.com.tw хост А в

результате запроса может получить ресурс с таким же Web-интерфейсом, как и у запрашиваемого, только с искаженной информацией (рисунок 1.6в).

Перехват пользовательского сетевого трафика через ложный объект дает злоумышленнику возможность проведения анализа данных, передаваемых по сети, модификации информации, а также



полной ее подмены.

Ниже приведены примеры некоторых наиболее распространенных атак, связанных с внедрением ложного объекта.

- Одним из способов внедрения ложного объекта может быть SQL-инъекция – это один из распространённых способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода.

Атакующий использует индексы поисковых систем для идентификации уязвимых сайтов. Злоумышленники ищут сайты, использующие распространенные системы управления контентом и другое ПО, содержащее уязвимости процессов обработки входных данных, применяемых в SQL-запросах. Результатом одной из последних атак такого рода стало то, что пользователи, посещающие зараженные страницы переводятся на другие сайты и на сервер Lilupophilupop.com, где им

предлагается загрузить вредоносное ПО под видом Adobe Flash Player или несуществующего антивируса.

С использованием SQL-инъекций злоумышленник может не только получить закрытую информацию из базы данных, но и, при определенных условиях, внести туда изменения.

В целом, атаки, связанные с различного рода инъекциями, возможны ввиду недостаточной проверки входных данных и подразумевают внедрение сторонних команд или данных в работающую систему (чаще всего это связано с Web-сайтами) с целью изменения хода её работы, а в результате – получение доступа к закрытым ресурсам и информации, либо дестабилизации работы системы в целом.

- Техника Clickjacking заключается в создании специального тега iFrame, который создает кнопку-подделку. При нажатии (или автоматически, без действия пользователя) на эту кнопку в невидимый iFrame загрузится специальная страница с вредоносным кодом. Спрятанная страница может быть подделкой текущей, где будет предложено вновь ввести идентификационные данные пользователя, которые при повторном вводе сохраняться на хосте злоумышленника.
- Как рассматривалось выше, существует множество вредоносных программ, которые инфицировав сетевой компьютер, обеспечивают злоумышленникам удаленный доступ и полное управление этим компьютером, а также возможность использовать его в качестве ложного объекта сети, выдавая себя за легального пользователя. Люки (Backdoors) – программы, обеспечивающие вход в систему или получение привилегированной функции (режима работы) в обход существующей системы полномочий. Часто используются для обхода системы безопасности. Люки не инфицируют файлы, но прописывают себя в реестр, модифицируя, таким образом, ключи реестра. BackDoor.Bitsex – троянская программа, представляющая собой полноценный сервер для удаленного управления инфицированным компьютером.
- Атака ARP-spoofing – применяется преимущественно в сетях Ethernet, но возможна и в других сетях, использующих протокол ARP. Данная атака основана на использовании такой уязвимости протокола ARP, как отсутствие системы аутентификации пользователей. Она состоит в том, что злоумышленник посылает ложные ARP-пакеты с целью убедить компьютер жертвы в том, что ложный объект и есть легальный конечный адресат. Далее пакеты пересылаются реальному получателю, MAC-адрес отправителя в них подменяется, чтобы ответные пакеты тоже шли через ложный объект. Злоумышленник получает возможность прослушивать трафик, например, общение по ICQ, почту жертвы и др. При этом в случае прохождения через ложный объект трафика многих пользователей может возникнуть переполнение ARP-таблиц и сетевой отказ в обслуживании.

Достаточно часто злоумышленник проводит атаку на систему с целью ее отказа в работе.

Получив контроль над проходящим потоком информации между объектами, ложный объект вычислительной сети может применять различные методы воздействия на перехваченную информацию, например:

- селекция потока информации и сохранение ее на ложном объекте (нарушение конфиденциальности);
- модификация информации:
  - модификация данных (нарушение целостности),
  - модификация исполняемого кода и внедрение разрушающих программных средств – программных вирусов (нарушение доступности, целостности);
- подмена информации (нарушение целостности).

Удаленная атака "отказ в обслуживании"

Одной из возможностей сетевой операционной системы (ОС), установленной на каждом объекте распределенной сети, является наличие сетевых служб, позволяющих удалённым пользователям использовать ресурсы данного объекта. Программа-сервер (например, FTP-сервер или Web-сервер), запущенная в сетевой ОС компьютера, обеспечивает удаленный доступ к FTP- или Web-ресурсам этого компьютера. Пользователь отправляет запросы на предоставление услуги, ОС обрабатывает приходящие извне запросы, пересылает их на соответствующий сервер (FTP или Web), а сервер отвечает на них по созданному виртуальному каналу.

Любая операционная система имеет ограничения по количеству открытых виртуальных соединений и существует предел ответов на поступающие запросы. Данные ограничения зависят от системных ресурсов, основными из которых являются вычислительные мощности, оперативная

память, дисковое пространство или пропускная способность каналов связи. Если какой-то из ресурсов достигнет максимальной загрузки, приложение будет недоступно.

Как правило, атаки типа DoS (Denial of service) направлены на исчерпание критичных системных ресурсов, что приводит к прекращению функционирования системы, т.е. к отказу в обслуживании и невозможности доступа к серверу удаленных пользователей.

Выделяется два типа отказа в обслуживании: первый, основанный на ошибке в приложении, и второй, основанный на плохой реализации или уязвимости протокола.

Отказ в обслуживании приложения становится возможен, если уязвимости приложения ведут к получению контроля над машиной (например, с помощью переполнения буфера обмена). Приложение станет недоступным либо из-за нехватки ресурсов, либо из-за аварийного завершения. Уязвимость приложения может быть использована и для нарушения работоспособности других компонентов системы, таких как сервер СУБД или сервер аутентификации.

Сетевой отказ в обслуживании основывается на особенностях стека протоколов TCP/IP.

Если атака выполняется одновременно с большого числа хостов, говорят о распределённой атаке типа "отказ в обслуживании" – DDoS-атаке (Distributed Denial of Service). В некоторых случаях к DDoS-атаке приводит легальное действие, например, на популярном Интернет-ресурсе указана ссылка на сайт, размещённый на не очень производительном сервере (так называемый слэшдот-эффект). Большой наплыв пользователей приводит к превышению допустимой нагрузки на сервер, и он очень быстро становится недоступным или доступ к нему затрудняется в результате перегруженности.

Ниже представлены некоторые типы подобных атак, однако, это всего лишь малая часть от существующих на сегодняшний день вариантов DoS-атак, информация о которых постоянно обновляется на специализированных Web-сайтах.

- SYN-flood. Выше был рассмотрен механизм установления TCP-соединения (рисунок 1.4). Атака типа SYN-flood использует именно этот механизм. TCP-соединение включает три состояния: отправка SYN-пакета, получение пакета SYN-ACK и посылка ACK-пакета.

Идея атаки состоит в создании большого количества не до конца установленных TCP-соединений. Для реализации этого злоумышленник отправляет на сервер-жертву множество запросов на установление соединения (пакеты, с выставленным флагом SYN), машина-жертва отвечает пакетами SYN-ACK. Злоумышленник же игнорирует эти пакеты, не высылая ответные, либо подделывает заголовок пакета таким образом, что ответный SYN-ACK отправляется на несуществующий адрес. Процесс установки соединения не завершается, а остается в полуоткрытом состоянии, ожидая подтверждения от клиента. А так как под каждый полученный SYN-пакет сервер резервирует место в своем буфере, то при огромном количестве запросов, буфер достаточно быстро переполняется. В результате, вновь поступающие SYN-запросы, в том числе от легальных пользователей, не обрабатываются, и новые соединения не устанавливаются.

- UDP-flood. Данный метод основан на применении UDP-протокола и обычно используется для того, чтобы максимально загрузить канал связи сервера-жертвы бесполезными данными.

Злоумышленник генерирует большое количество UDP-датаграмм (UDP-шторм), направленных на определенную машину. В результате происходит перегрузка сети и недоступность сервера-жертвы. В протоколе TCP есть механизмы предотвращения перегрузок: если подтверждения приема пакетов приходят со значительной задержкой, передающая сторона замедляет скорость передачи TCP-пакетов. Так как в протоколе UDP такой механизм отсутствует, то после начала атаки UDP-трафик "захватывает" практически всю доступную полосу пропускания.

Вредоносное ПО LOIC (Low Orbit Ion Cannon) выполняет распределённую атаку на отказ в обслуживании путём постоянной отправки TCP и UDP пакетов на целевой сайт или сервер. Это ПО создано для организации DDoS-атак на Web-сайты с участием тысяч анонимных пользователей, пользующихся программой. Атаки производятся на такие сайты, как Visa.com или Mastercard.com.

- ICMP-flood (ICPM-smurfing). Принцип работы такой DDoS-атаки довольно прост. Злоумышленник, изменяя адрес источника, посылает пакет ICMP Echo Request (больше известный как ping) к конкретным хостам.

Эти хосты отвечают пакетом ICMP Echo Reply, отправляя его на тот IP-адрес, который злоумышленник указал как источник. Часто для усиления атаки используются локальные сети (LAN)

с включенной опцией направленной широковещательной рассылки (directed broadcast) в ответ на команду "ping" с каждого хоста в составе сети. Например, на один запрос будет отправлено 100 ответов. В результате вся сеть подвергается отказу из-за перегрузки.

- Mailbombing. Суть атаки сводится к тому, чтобы генерировать большое количество сообщений с разных источников для почтового сервера (почтового ящика) с тем, чтобы реализовать ограничение доступа (или полный отказ) к этому почтовому серверу (ящику).
- Атаки, основанные на уязвимостях протоколов управления. Например, утилита THC-SSL-DOS, которую некоторые злоумышленники применяют в качестве инструмента для проведения DoS-атак на SSL-серверы, использует уязвимость в функции повторного подтверждения SSL (SSL renegotiation).

Функция, предназначенная для обеспечения большей безопасности SSL, на самом деле делает его более уязвимым перед атакой.

- Программы Backdoors способны производить DDoS атаку.

Например, троян Backdoor.IRCBot.ADEQ представляет собой вредоносное ПО, которое распространяется как регулярное обновление для Java платформы, и являет собой чрезвычайно опасный инструмент для инициации распределенной атаки "отказ в обслуживании". Данная программа имеет возможность установки ссылки целевого ресурса, назначения времени атаки, интервала и частоты запросов.

Удаленная атака типа "отказ в обслуживании" является активным воздействием, осуществляемым с целью нарушения работоспособности системы, безусловно относительно цели атаки. Данная удалённая атака является однонаправленным воздействием, как межсегментным, так и внутрисегментным, осуществляемым на транспортном и прикладном уровнях модели OSI.

27.Обеспечение безопасности электронной почты

Использование электронной почты

Электронная почта, или e-mail, — самый популярный вид использования Интернета. С помощью электронной почты в Интернете вы можете послать письмо миллионам людей по всей планете. Существуют шлюзы частных почтовых систем в интернетовский e-mail, что значительно расширяет ее возможности.

Помимо взаимодействия «один-один» e-mail может поддерживать списки электронных адресов для рассылки, поэтому человек или организация может послать e-mail всему этому списку адресов людей или организаций. Иногда списки рассылки e-mail имеют элементы, являющиеся указателями на другие списки рассылки, поэтому одно письмо может быть в конце концов доставлено тысячам людей.

Разновидностью списков рассылки являются дискуссионные группы на основе e-mail. Их участники посылают письмо центральному серверу списка рассылки, и сообщения рассылаются всем другим членам группы. Это позволяет людям, находящимся в разных временных зонах или на разных континентах, вести интересные дискуссии. При помощи специальных программ люди могут подписаться на список или отписаться от него без помощи человека. Серверы списков рассылки часто предоставляют другие сервисы, такие как получение архивов, дайджестов сообщений или связанных с сообщениями файлов. Группы новостей USENET являются усовершенствованием дискуссионных почтовых групп.

Электронная почта становится все более важным условием ведения повседневной деятельности. Организациям нужны политики для электронной почты, чтобы помочь сотрудникам правильно ее использовать, уменьшить риск умышленного или неумышленного неправильного ее использования, и чтобы гарантировать, что официальные документы, передаваемые с помощью электронной почты, правильно обрабатываются. Организациям нужно разработать политику для правильного использования электронной почты, такую же как для тлефона.

Политика должна давать общие рекомендации в следующих областях:

- использование электронной почты для ведения деловой деятельности;
- использование электронной почты для ведения личных дел;
- управление доступом и сохранение конфиденциальности сообщений;
- администрирование и хранение электронных писем.

Основы e-mail

Основными почтовыми протоколами в Интернете (не считая частных протоколов, шлюзуемых или туннелируемых через Интернет) являются SMTP (Simple Mail Transport Protocol), POP (Post Office Protocol) и IMAP (Internet Mail Access Protocol).

SMTP

SMTP — это почтовый протокол хост-хост. SMTP-сервер принимает письма от других систем и сохраняет их в почтовых ящиках пользователей. Сохраненные письма могут быть прочитаны несколькими способами. Пользователи с интерактивным доступом на почтовом сервере могут читать почту с помощью локальных почтовых приложений. Пользователи на других системах могут загрузить свои письма с помощью программ-почтовых клиентов по протоколам POP3 и IMAP.

UNIX-хосты сделали самым популярным SMTP. Широко используемыми SMTP-серверами являются Sendmail, Smail, MMDF и PP. Самым популярным SMTP-сервером в Unixe является Sendmail, написанный Брайаном Эллманом. Он поддерживает создание очередей сообщений, переписывание заголовков писем, алиасы, списки рассылки и т.д. Обычно он конфигурируется так, что должен работать как привилегированный процесс. Это означает, что если его защиту можно будет обойти каким-нибудь способом, атакующий сможет нанести вред, далеко превышающий удаление электронных писем.

POP

POP — это самый популярный протокол приема электронной почты. POP-сервер позволяет POP-клиенту загрузить письма, которые были получены им от другого почтового сервера. Клиенты могут загрузить все сообщения или только те, которые они еще не читали. Он не поддерживает

удаление сообщений перед загрузкой на основе атрибутов сообщения, таких как адрес отправителя или получателя. POP версии 2 поддерживает аутентификацию пользователя с помощью пароля, но пароль передается серверу в открытом (незашифрованном) виде.

POP версии 3 предоставляет дополнительный метод аутентификации, называемый APOP, который прячет пароль. Некоторые реализации POP могут использовать Kerberos для аутентификации.

IMAP

IMAP — это самый новый, и поэтому менее популярный протокол чтения электронной почты.

Как сказано в RFC:

IMAP4rev1 поддерживает операции создания, удаления, переименования почтовых ящиков, проверки поступления новых писем; оперативное удаление писем; установку и сброс флагов операций; разбор заголовков в формате RFC-1822 и MIME-IMB; поиск среди писем; выборочное чтение писем.

IMAP более удобен для чтения почты в путешествии, чем POP, так как сообщения могут быть оставлены на сервере, что избавляет от необходимости синхронизировать списки прочитанных писем на локальном хосте и на сервере.

MIME

MIME — это сокращение для Многоцелевых расширений интернетовской почты (Multipurpose Internet Mail Extensions). Как сказано в RFC 2045, он переопределяет формат сообщений электронной почты, чтобы позволить:

- передачу текстов в кодировке, отличной от US-ASCII,
- передачу в письме нетекстовой информации в различных форматах,
- сообщения из нескольких частей, и
- передачу в заголовке письма информации в кодировке, отличной от US-ASCII.

Он может использоваться для поддержки таких средств безопасности, как цифровые подписи и шифрованные сообщения. Он также позволяет посылать по почте выполняемые файлы, зараженные вирусами, или письма с РПС.

Как и веб-браузеры, программы чтения почты могут быть сконфигурированы автоматически запускать приложения-помощники для обработки определенных типов MIME-сообщений.

Потенциальные проблемы с электронной почтой

Случайные ошибки. Можно легко допустить ошибку при работе с электронной почтой. Письмо может быть послано случайно. Простое нажатие клавиши или щелчок мышкой могут послать письмо по неправильному адресу. Почтовые сообщения могут храниться годами, поэтому плохое выражение может аукнуться через много времени. Архивы писем могут возрасти до такой степени, что система будет аварийно завершаться. Неправильно настроенная программа чтения групп новостей может привести к посылке сообщения не в те группы. Ошибки в списках рассылки могут привести к долгому блужданию писем между почтовыми серверами, причем число писем может увеличиться до такой степени, что почтовые серверы аварийно завершатся.

Если почтовая система организации присоединена к Интернету, последствия ошибок могут привести к тяжелым последствиям. Вот некоторые из способов предотвратить ошибки:

- учить пользователей что делать, если они совершили ошибку, и как правильно работать с электронной почтой
- конфигурировать программы электронной почты так, чтобы стандартные действия пользователя, использующие установки по умолчанию, были бы самыми безопасными
- использовать программы, которые строго реализуют протоколы и соглашения Интернета. Каждый раз, когда онлайновый сервис шлюзует письмо из частной почтовой системы в интернетовскую электронную почту, слышатся вопли протеста из-за появления большого числа сообщений с ошибками, возникшими в результате неправильных настроек почтовых серверов этого сервиса.

Персональное использование. Так как письма обычно используются для обеспечения деятельности организации, как и телефон и факс, использование его в личных целях должно быть ограничено или запрещено (это зависит от организации).

Хотя проще всего определить, что электронная почта используется только для решения задач организации, все понимают, что эту политику тяжело претворить в жизнь. Мудрее будет установить четкие границы использования e-mail в личных целях.

Если вы используете служебный телефон для того, чтобы позвонить в химчистку, то маловероятно, что ваш звонок будет восприниматься как официальный запрос компании. Но посылка электронного письма с электронным почтовым адресом, содержащим адрес организации, будет похожа на посылку бумажного письма на фирменном бланке компании. Если отправитель использует свой логин в компании для посылки электронной почты в группу новостей, может показаться, что компания одобряет мнение, высказываемое им в письме.

Маркетинг. В прошлом, когда Интернет был исследовательской сетью, ее коммерческое использование было запрещено. Кроме того, слишком мало компаний и людей имели доступ к интернетовской почте, поэтому было нецелесообразно использовать ее для коммерческих целей. Сейчас Интернет расширился и разрешается использовать его в коммерческих целях, поэтому компании стали поддерживать списки рассылки для обмена информацией со своими клиентами. Как правило, клиенты должны послать запрос для того, чтобы попасть в список рассылки. Когда большие онлайновые сервисы стали шлюзовать письма в Интернет, неожиданно обнаружилось, что таким образом можно передать информацию гораздо большей аудитории. Так родился маркетинг в Интернете с помощью посылки отдельных почтовых сообщений.

Люди написали программы для автоматизации поддержания списков рассылки и образовали компании для сбора и продажи списков электронных почтовых адресов организациям, занимающимся маркетингом. Конгресс США принял билль, согласно которому прямой маркетинг с помощью электронной почты должен осуществляться в соответствии с теми же правилами, которыми ограничивается использование массовой посылки писем, чтобы лица, занимающиеся таким маркетингом, вели списки адресов, владельцы которых не желают получать рекламу в электронных письмах.

Угрозы, связанные с электронной почтой

Основные протоколы передачи почты (SMTP, POP3,IMAP4) обычно не осуществляют надежной аутентификации, что позволяет легко создать письма с фальшивыми адресами. Ни один из этих протоколов не использует криптографию, которая могла бы гарантировать конфиденциальность электронных писем. Хотя существуют расширения этих протоколов, решение использовать их должно быть явно принято как составная часть политики администрации почтового сервера. Некоторые такие расширения используют уже имеющиеся средства аутентификации, а другие позволяют клиенту и серверу согласовать тип аутентификации, который будет использоваться в данном соединении.

Фальшивые адреса отправителя. Адресу отправителя в электронной почте Интернета нельзя доверять, так как или отправитель может указать фальшивый обратный адрес, или заголовок может быть модифицирован в ходе передачи письма, или отправитель может сам соединиться с SMTP-портом на машине, от имени которой он хочет отправить письмо, и ввести текст письма.

Перехват письма. Заголовки и содержимое электронных писем передаются в чистом виде. В результате содержимое сообщения может быть прочитано или изменено в процессе передачи его по Интернету. Заголовок может быть модифицирован, чтобы скрыть или изменить отправителя или для того чтобы перенаправить сообщение.

Почтовые бомбы. Почтовая бомба - это атака с помощью электронной почты. Атакуемая система переполняется письмами до тех пор, пока она не выйдет из строя. Как это может случиться, зависит от типа почтового сервера и от того, как он сконфигурирован.

Угрожающие письма. Так как любой человек в мире может послать вам письмо, может оказаться трудным заставить его прекратить посылать их вам. Люди могут узнать ваш адрес из списка адресов организации, списка лиц, подписавшихся на список рассылки, или писем в Usenet. Если вы указали ваш почтовый адрес какому-нибудь веб-сайту, то он может продать ваш адрес "почтовым мусорщикам". Некоторые веб-браузеры сами указывают ваш почтовый адрес, когда вы посещаете веб-

сайт, поэтому вы можете даже не понять, что именно вы его дали. Много почтовых систем имеют возможности фильтрации почты, то есть поиска указанных слов или словосочетаний в заголовке письма или его теле и последующего помещения писем в определенный почтовый ящик или удаления. Но большинство пользователей не знает, как использовать механизм фильтрации. Кроме того, фильтрация у клиента происходит после того, как письмо уже получено или загружено, поэтому таким образом тяжело удалить большие объемы писем.

Для безопасной атаки можно использовать анонимный ремэйлер. Когда кто-то хочет послать оскорбительное или угрожающее письмо и при этом скрыть свою личность, он может воспользоваться анонимным ремэйлером. Если человек хочет послать электронное письмо, не раскрывая свой домашний адрес тем, кто может угрожать ему, он может тоже использовать анонимный ремэйлер. Если он начнет вдруг получать нежелательные письма по своему текущему адресу, он может отказаться от него и взять новый.

Одним часто используемым средством защиты, применяемым некоторыми пользователями Usenet, является конфигурирование своих клиентов для чтения новостей таким образом, что в поле Reply-To (обратный адрес) письма, посылаемого ими в группу новостей, помещается фальшивый адрес, а реальный адрес помещается в сигнатуре или в теле сообщения. Таким образом, программы сбора почтовых адресов, собирающие адреса из поля Reply-To, окажутся бесполезными.

В конгрессе США было подано несколько биллей об ограничениях на работу таких программ-мусорщиков. В одних предлагалось создать списки стоп-слов и помещать слово "реклама" в строку темы письма, в другом предлагалось считать их просто незаконными.

Защита электронной почты

Защита от фальшивых адресов. От этого можно защититься с помощью использования присоединения к письмам электронных подписей.

Защита от перехвата. От него можно защититься с помощью шифрования сообщения или канала, по которому он передается. Одним из самых популярных приложений является PGP. Коммерческая версия PGP включает в себя плагины для нескольких популярных почтовых программ, что делает ее особенно удобной для включения в письмо электронной подписи и шифрования письма клиентом. Последние версии PGP используют версию алгоритма шифрования RSA.

Корректное использование электронной почты

Все служащие должны использовать электронную почту так же, как и любое другое официальное средство организации. Из этого следует, что когда письмо посылается, то как отправитель, так и получатель должны гарантировать, что взаимодействие между ними осуществляется согласно принятым правилам взаимодействия. Взаимодействие с помощью почты не должны быть неэтичным, не должно восприниматься как конфликтная ситуация или содержать конфиденциальную информацию.

Защита электронных писем и почтовых систем

Защита писем, почтовых серверов и программ должна соответствовать важности информации, передаваемой по сетям. Как правило, должно осуществляться централизованное управление сервисами электронной почты и должна быть разработана политика, в которой указывался бы нужный уровень защиты.

Примеры политик безопасности для электронной почты

Низкий риск

Пользователь. Использование служб электронной почты для целей, явно противоречащих интересам организации или противоречащих политике безопасности организации, явно запрещено, так же как и чрезмерное использование их в личных целях.

Использование адресов организации в письмах-пирамидах запрещено.

Организация предоставляет своим сотрудникам электронную почту для выполнения ими своих обязанностей. Ограниченное использование ее в личных целях разрешается, если оно не угрожает организации.

Использование электронной почты для получения личной коммерческой выгоды запрещено.

Менеджер. Все сотрудники должны иметь адреса электронной почты.

Справочники электронных адресов должны быть доступны для общего доступа.

Если организация обеспечивает доступ к электронной почте внешних пользователей, таких как консультанты, контрактные служащие или партнеры, они должны прочитать правила доступа к электронной почте и расписаться за это.

Содержимое почтовых сообщений считается конфиденциальным, за исключением случая проведения расследований органами внутренних дел.

Сотрудник отдела автоматизации. POP-сервер должен быть сконфигурирован так, чтобы исключить использование незашифрованных паролей с локальных машин.

Средний риск

Пользователь. Электронная почта предоставляется сотрудникам организации только для выполнения ими своих служебных обязанностей. Использование ее в личных целях запрещено.

Конфиденциальная информация или информация, являющаяся собственностью организации, не может быть послана с помощью электронной почты.

Могут использоваться только утвержденные почтовые программы.

Нельзя устанавливать анонимные ремэйлеры.

Служащим запрещено использовать анонимные ремэйлеры

Менеджер. Конфиденциальная информация или информация, являющаяся собственностью организации, не может быть послана с помощью электронной почты.

Если будет установлено, что сотрудник в личных целях использует электронную почту, он будет наказан.

Сотрудник отдела автоматизации. Почтовая система должна обеспечивать только один внешний электронный адрес для каждого сотрудника. Этот адрес не должен содержать имени внутренней системы или должности.

Должен вестись локальный архив MIME-совместимых программ для просмотра специальных форматов, который был бы доступен для внутреннего использования.

Высокий риск

Пользователь. Электронная почта предоставляется сотрудникам организации только для выполнения своих служебных обязанностей. Использование ее в личных целях запрещено.

Все электронные письма, создаваемые и хранимые на компьютерах организации, являются собственностью организации и не считаются персональными.

Организация оставляет за собой право получить доступ к электронной почте сотрудников, если на то будут веские причины. Содержимое электронного письма не может быть раскрыто, кроме как в целях обеспечения безопасности или по требованию правоохранительных органов.

Пользователи не должны позволять кому-либо посылать письма, используя их идентификаторы. Это касается их начальников, секретарей, ассистентов или других сослуживцев.

Организация оставляет за собой право осуществлять наблюдение за почтовыми отправлениями сотрудников. Электронные письма могут быть прочитаны организацией, даже если они были удалены и отправителем и получателем. Такие сообщения могут использоваться для обоснования наказания.

Менеджер. Справочники электронных адресов сотрудников не могут быть сделаны доступными всем.

Если с помощью электронного письма должна быть послана конфиденциальная информация или информация, являющаяся собственностью организации, она должна быть зашифрована так, чтобы ее мог прочитать только тот, кому она предназначена, с использованием утвержденных в организации программ и алгоритмов.

Никто из посетителей, контрактников или временных служащих не имеет права использовать электронную почту организации.

Должно использоваться шифрование всей информации, классифицированной как критическая или коммерческая тайна, при передаче ее через открытые сети, такие как Интернет.

Выходящие сообщения могут быть выборочно проверены, чтобы гарантировать соблюдение политики.

Сотрудник отдела автоматизации. Входящие письма должны проверяться на вирусы или другие РПС.

Почтовые серверы должны быть сконфигурированы так, чтобы отвергать письма, адресованные не на компьютеры организации.

Журналы почтовых серверов должны проверяться на предмет выявления использования неутвержденных почтовых клиентов сотрудниками организации, и о таких случаях следует докладываться.

Почтовые клиенты должны быть сконфигурированы так, чтобы каждое сообщение подписывалось с помощью цифровой подписи отправителя.

Хранение электронных писем

Официальные документы организации, передаваемые с помощью электронной почты, должны быть идентифицированы и должны администрироваться, защищаться и сопровождаться настолько долго, насколько это нужно для деятельности организации, аудита, юристов или для других целей. Когда электронная почта - это единственный способ передачи официальных документов компании, то к ним применяются те же самые процедуры, как если бы они передавались на бумаге.

Для предотвращения случайного удаления писем сотрудники должны направлять копии таких сообщений в официальный файл или архив. Должны храниться как входящие, так и выходящие сообщения с приложениями. Любое письмо, содержащее формальное разрешение или выражающее соглашение организации с другой организацией, должно копироваться в соответствующий файл (или должна делаться его печатная копия) для протоколируемости и аудита.

Период хранения всех писем определяется юристами.

28.Назначение, состав и архитектура систем электронного документа оборота. Угрозы информации, характерные для них

Современный электронный документооборот нуждается не только в защите, но и в научном объяснении всех происходящих внутри него процессов и явлений. Изучение систем электронного документооборота является неотъемлемой частью науки документоведение. Системы электронного документооборота — это жизненный цикл электронных документов в организации, начиная от их получения: ввод, электронная почта, регистрация; прохождения в подразделениях с изменением состояния: доведение до сведения, согласование, подписание, работа с ним, закрытие и заканчивая списанием в архив. Иногда электронный документооборот обозначается термином workflow - управление потоками работ (Workflow Management Systems), который характеризует движение документов как поток работ, выполняемых в рамках бизнес-процесса. Либо ECM - управление корпоративным содержимым (Enterprise Content Management). Или EDMS — управление электронным документооборотом (Electronic Document Management Systems). В основном система электронного документооборота рассматривается как программное обеспечение, главными задачами которого являются организация и поддержка жизненного цикла электронных документов.

Под назначением электронного документооборота принято понимать организацию движения документов между подразделениями предприятия или организации, группами пользователей или отдельными пользователями. При этом, под движением документов подразумевается не их физическое перемещение, а их передача исполнителям с уведомлением конкретных пользователей и контролем за их исполнением.

Системы электронного документооборота обеспечивают процесс создания, управления доступом и распространения больших объемов документов в компьютерных сетях, а также обеспечивают контроль над потоками документов в организации. Часто эти документы хранятся в специальных хранилищах или в иерархии файловой системы. Типы файлов, которые поддерживают системы электронного документооборота, включают: текстовые документы, изображения, электронные таблицы, аудиоданные, видеоданные и Web-документы.

В настоящее время сложилось, что назначение систем электронного документооборота в организации — это хранение электронных документов, а также работы с ними (поиск как по атрибутам, так и по содержимому). В системе электронного документооборота автоматически отслеживаются изменения в документах, сроки их исполнения, движение, контроль.

Комплексные задачи СЭД предусматривают:

- охват всего цикла делопроизводства организации — от постановки задачи на создание документа до его списания в архив,
- обеспечение централизованного хранения документов в любых форматах.
- объединяют разрозненные потоки документов территориально удаленных предприятий в единую систему.
- обеспечивают гибкое управление документами как с помощью жесткого определения маршрутов движения, так и путем свободной маршрутизации документов.
- реализуют жесткое разграничение доступа пользователей к различным документам в зависимости от их компетенции, занимаемой должности и назначенных им полномочий.
- настраивают существующую организационно-штатную структуру и систему делопроизводства предприятия,
- интегрируются с существующими корпоративными системами.

Основными пользователями СЭД являются государственные организации, предприятия, банки, промышленные предприятия и все прочие структуры, чья деятельность сопровождается созданием, обработкой и хранением документов.

Основные свойства СЭД в документоведении состоят из открытости (построены по модульному принципу, а их интерфейсы являются открытыми, что позволяет добавлять к СЭД новые функции или совершенствовать уже имеющиеся. Возможность относительно простого добавления к СЭД множества модулей ввода документов со сканера, связи с электронной почтой, с программами пересылки факсов и др.); высокой степени интеграции с прикладным ПО (пользователи имеют дело только с обычными прикладными программами: в момент инсталляции клиентской части СЭД прикладные программы дополняются новыми функциями и элементами меню. Например, пользователь текстового процессора MS Word, открывая файл, сразу видит библиотеки и папки с

документами СЭД. При сохранении документ автоматически размещается в базе данных СЭД. То же относится и к другим офисным и специализированным программам. Следует также отметить, что в большинстве распространенных СЭД реализована интеграция с наиболее известными ERP-системами (SAP R/3, Oracle Applications и др.). Именно возможность интеграции с различными приложениями является одним из характерных свойств СЭД. Благодаря ему СЭД могут выступать в качестве связующего звена между различными корпоративными приложениями, создавая основу для организации делопроизводства на предприятии); особенностью хранения документов (в СЭД реализована иерархическая система хранения документов (по принципу "шкаф/полка/папка"). Каждый документ помещается в папку, которая, в свою очередь, находится на полке и т. д. Количество уровней вложения при хранении документов не ограничено. Один и тот же документ может входить в состав нескольких папок и полок за счет применения механизма ссылок. В ряде СЭД реализованы еще более мощные возможности хранения за счет организации связей между документами (эти связи можно устанавливать и редактировать в графическом виде. Любому документу в СЭД присущ определенный набор атрибутов (например, его название, автор документа, время его создания и др.). Набор атрибутов может меняться от одного типа документа к другому (в пределах одного типа документов он сохраняется неизменным). Для каждого типа документов с помощью визуальных средств создается шаблон карточки, где в понятном графическом виде представлены наименования атрибутов документа. При введении документа в СЭД берется необходимый шаблон и заносятся значения.

После заполнения карточка оказывается связанной с самим документом как на одном, так и на нескольких серверах.

Разграничение доступа в системе электронного документооборота реализовано в виде разграничения полномочий и контроля за доступом к документам. В большинстве случаев с их помощью определяются следующие виды доступа:

- полный контроль над документом;
- право редактировать, но не уничтожать документ;
- право создавать новые версии документа, но не редактировать его;
- право аннотировать документ, но не редактировать его и не создавать новые версии;
- право читать документ, но не редактировать его;
- право доступа к карточке, но не к содержимому документа;
- полное отсутствие прав доступа к документу.

Во время работы с СЭД каждое действие пользователя протоколируется, и, таким образом, вся история его работы с документами может быть легко отслежена.

Применяемые в СЭД модули управления изображениями и образами (imaging systems) осуществляют конвертацию отсканированной с бумажных носителей информации в электронную форму (обычно, в формате TIFF). Данная технология лежит в основе перевода в электронную форму информации со всех унаследованных бумажных документов. В число базовых функций стандартной системы обработки изображений входят: сканирование, хранение, ряд возможностей по поиску изображений.

Аннотирование документов над документами обычно весьма полезна. Так как в некоторых случаях пользователи лишены прав на внесение каких- либо изменений в документ в процессе его согласования, то они могут воспользоваться возможностью его аннотирования. В большинстве СЭД аннотирование реализуется за счет включения в карточку документа атрибута для аннотации и передачи пользователям прав на редактирование такого поля карточки. Но такое решение не всегда приемлемо (особенно при аннотировании графического документа). В связи с этим, в некоторых СЭД существует так называемая функция "красного карандаша", с помощью которой можно графически указать недостатки на самом изображении.

Иногда СЭД дополняют модулями управления корпоративными электронными записями. Корпоративные записи фиксированы во времени и неизменяем, они являются свидетельством бизнес-транзакций, различных прав и обязательств. Корпоративные пользователи должны сами определяют какое содержимое необходимо сделать корпоративной записью для оценки перспективных потребностей бизнеса. В число корпоративных решений, требующих сохранения содержимого могут входить основные бизнес-системы, бухгалтерские системы, почтовые системы, системы управления отчетами и выводом, системы электронной коммерции, программные средства коллективной работы (системы управления проектами, онлайновой конференц- связи.

В некоторых СЭД применяются модули управления выводом, основным предназначением которых является генерация выходных документов. В них дополнительно реализованы возможности архивации и долговременного хранения выходных отчетов и документов. В связи с этим они классифицируются, как интегрированные системы архивации и поиска документов. Однако главной причиной их популярности все же является занимаемая ими рыночная ниша — генерация документов и отчетов в информационных системах предприятий и организаций, построенных с использованием ERP- систем.

Современные условия позволяют выделить ряд основных принципов построения системы электронного документооборота: соответствие требованиям стандартов на формы и системы документации; распределенность; масштабируемость; модульность; открытость; переносимость на другие аппаратные платформы.

Основными функциями системы электронного документооборота помимо регистрации и контроля исполнения документов являются:

- создание справочников и работа с ними;
- контроль движения бумажного и электронного документа, ведение истории работы с документами;
- создание и редактирование реквизитов документов;
- формирование отчетов по документообороту предприятия;
- импорт документов из файловой системы и Интернета;
- создание документа прямо из системы на основе шаблона (прямая интеграция);
- работа с версиями документа, сложными многокомпонентными и многоформатными документами, вложениями;
- электронное распространение документов;
- работа с документами в папках;
- получение документов посредством сканирования и распознавания.
- уменьшением затрат на доступ к информации и обработку документов.

Безбумажный документооборот дает целый ряд преимуществ при обмене документами (указами, распоряжениями, письмами, постановлениями и т.д.). В этом случае временные затраты (распечатка, пересылка, ввод полученного документа с клавиатуры) существенно снижаются, убыстряется поиск документов, снижаются затраты на их хранение и т.д. Но при этом возникает проблема аутентификации автора документа и самого документа. Эти проблемы в обычной (бумажной) информатике решаются за счет того, что информация в документе жестко связана с физическим носителем (бумагой). На машинных носителях такой связи нет.

Для выявления возможных угроз в системе обмена электронными документами необходимо четко представлять жизненный цикл электронного документа в системе электронного документооборота. Исходя из анализа возможных видов атак на систему обмена и хранения электронных документов, можно сделать вывод о том, что основным понятием в системе обмена электронными документами является аутентификация.

Под аутентификацией информации понимается установление подлинности информации исключительно на основе внутренней структуры самой информации, установление того факта, что полученная законным получателем информация была передана подписавшим ее законным отправителем и при этом не была искажена.

Задачи аутентификации можно разделить на следующие типы: аутентификация абонента, аутентификация принадлежности абонента к заданной группе, аутентификация хранящихся на машинных носителях документов.

Основные характеристики системы аутентификации:

- время реакции на нарушение, требуемые для реализации вычислительные ресурсы;
- степень защищенности (стойкость) к возможным (известным на сегодня) атакам на средства защиты (например, криптостойкость).