Михаил Фленов

WEB-СЕРВЕР глазами ХАКЕРА

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2009

УДК 681.3.06
ББК 32.973.26-018.2
Ф69

Фленов М. Е.

Ф69 Web-сервер глазами хакера: 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2009. — 320 с.: ил. + CD-ROM

ISBN 978-5-9775-0471-3

Рассмотрена система безопасности Web-серверов и типичные ошибки, совершаемые Web-разработчиками при написании сценариев на языках PHP, ASP и Perl. Приведены примеры взлома реальных Web-сайтов, имеющих уязвимости, в том числе и популярных. В теории и на практике рассмотрены распространенные хакерские атаки: DoS, Include, SQL-инъекции, межсайтовый скриптинг, обход аутентификации и др. Описаны основные приемы защиты от атак и рекомендации по написанию безопасного программного кода. Во втором издании добавлены новые примеры реальных ошибок, а также описание каптча. Компакт-диск содержит листинги из книги и программы автора.

Для Web-программистов и администраторов

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Игорь Шишигин*
Зав. редакцией	*Григорий Добин*
Редактор	*Анна Кузьмина*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Виктория Пиотровская*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.07.09.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 25,8.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12

ISBN 978-5-9775-0471-3

Оглавление

Введение

Интернет захватывает все новые и новые области, и с его помощью мы уже можем управлять даже бытовой техникой. Например, доступ к Интернету встраивают в холодильник, чтобы он сам мог заказывать заканчивающиеся продукты. Но все это пока только на словах, на выставках и, возможно, у избранных людей. Если посмотреть на нашу реальность, то в быту Интернет пока не прижился. Возможно, что это из-за высокой цены устройств, а может, вероятных проблем с безопасностью.

Лично я на данный момент не готов предоставить управление через Интернет даже самыми безобидными бытовыми приборами. Простейший пример: если хакер получит управление холодильником, то он сможет отключить его, и все продукты пропадут. Если под его управление попадет микроволновая печь, то он сможет ее сжечь. Дело в том, что печь нельзя включать пустой (так написано в паспорте). И после этого мы будем рады, если сгорит не вся квартира, а только прибор.

Внутри одной квартиры я готов управлять бытовыми устройствами через сеть или WiFi, потому что свою домашнюю сеть я могу защитить сетевыми экранами, которые предотвратят доступ извне. А вот на счет открытия доступа через общедоступную сеть — я пока еще не готов к принятию такого кардинального решения.

Но действительно ли хакеры так страшны? Может быть, страх навеян благодаря журналистам, которые пишут на тему интернет-взломов и любят приукрасить? Да, действия хакеров содержат угрозу, но более опасны программисты и администраторы, не уделяющие проблемам безопасности достаточно внимания. Ошибаются все, я и сам не без греха. Но иногда встречается откровенный непрофессионализм, когда нет даже простейших попыток обеспечить Web-серверу достойную защиту. Чаще всего этим грешат люди, не имеющие достаточно навыков работы с компьютером, которые только недавно подключились к Интернету и решили создать свой Web-сайт.

Но непрофессионализм или невнимательность составляют не такую уж и большую долю в нашем мире. Если бы сайты так легко было взломать, то они бы и не существовали. В Интернете много сайтов, которые оперируют не только безобидной информацией, но и деньгами в виде электронной наличности и даже реальной наличностью.

Из этой книги вы узнаете, каким образом действуют хакеры: как находят уязвимости и используют их для получения конфиденциальной информации или прав доступа администратора или хотя бы просто повышенных привилегий, позволяющих выполнять привилегированные операции, например, модерирование. Мы будем рассматривать методы взломов на практике и на примерах реальных Web-сайтов. Да, именно реальных, чтобы вы могли увидеть всю опасность невнимательности программистов и администраторов. Все примеры мы будем подробно разбирать, и я постараюсь предложить возможные варианты решения рассмотренных проблем и исправления ошибок.

Зачем мы будем рассматривать взлом, да еще и на практике? С одной стороны, этот материал можно воспринимать как инструкции по взлому, но с другой стороны, вы не сможете защититься, если не будете знать, откуда может прийти угроза. Допустим, что вы полководец и хотите защитить свою территорию от вторжения. Вы можете выкопать вокруг своих земель ров, заминировать дороги и растянуть колючую проволоку, но все эти действия будут бессмысленными, если враг готовит воздушный удар или авиацию с бомбами. Поэтому сначала следует выяснить, как может действовать неприятель, а потом уже искать достойный ответ. Именно так мы и поступим: будем рассматривать возможную угрозу, а потом искать защиту от нее.

Несмотря на то, что я описываю взлом и знаю, как взламываются сайты, я еще ни разу в жизни не взламывал ради выгоды или в корыстных целях. То, что я делал, даже нельзя называть взломом. Да, я находил уязвимости на сайте и обнаруживал двери проникновения на Web-сайты, но я никогда не брал чужой информации и всегда сообщал о найденных уязвимостях владельцам сайтов.

Что подразумевается под взломом Web-сервера? Это взлом Web-сайта или службы, которая обрабатывает Web-страницы? Мы будем рассматривать проблему комплексно, включая защиту аппаратной части и операционной системы (ОС), а также Web-сервера, баз данных и самих сценариев, которые выполняются на Web-сервере. Аппаратную часть и ОС мы будем рассматривать поверхностно, по мере того, как понадобится нам та или иная информация. Просто я не думаю, что стоит лишний раз говорить о том, как защищать BIOS компьютера или загрузчик: этот вопрос уж слишком отдален от тематики книги.

Все владельцы взломанных мною во время написания этой книги Web-сайтов были извещены о найденных уязвимостях, поэтому если у вас не получилось повторить описанные действия, значит, ошибку уже исправили.

Что не вошло в книгу

Что не будет рассмотрено в данной книге подробно, так это социальная инженерия. Эту тему мы затронем лишь поверхностно, хотя именно данный метод позволяет осуществить взлом достаточно быстро и эффективно. Тут можно писать отдельную книгу, и если вас интересует более подробная информация, то советую обратиться либо к моей книге "Компьютер глазами хакера" [5], либо к книге гуру социальной инженерии Кевина Митника — "Искусство обмана: контролирование человеческого фактора в безопасности" [1].

Немного отвлекусь и скажу, что когда Кевин Митник отбывал наказание в местах не столь отдаленных, то большинство считало его величайшим хакером всех времен и народов, потому что он получил большой срок. Помню, как в Интернете легко было встретить лозунги типа "Free Kevin Mitnick" ("Освободите Кевина Митника"). Но стоило человеку выйти на свободу, как его тут же начали считать чуть ли не ламером и зазнайкой, потому что он начал писать книги и специализироваться на консалтинге в сфере безопасности. Я считаю этого человека очень умным и весьма опытным в сфере безопасности и особенно в сфере социальной инженерии и настоятельно рекомендую к прочтению его труды.

В некоторых случаях для понимания представленного материала могут понадобиться знания программирования. Конечно же, я постараюсь все описывать доступно и понятно каждому, вне зависимости от уровня подготовки, и все же опыт программирования и знание команд ОС Linux желательны. О безопасности ОС Linux и ее командах можно почитать в книге "Linux глазами хакера" [2], а о программировании для Интернета на языке PHP можно узнать из книги "PHP глазами хакера" [3].

Интернет

Из интернет-ресурсов я могу порекомендовать:

- **www.flenov.info** — блог очень умного парня. Сам себя не похвалишь, так никто не похвалит;
- **www.profwebdev.com** — на этом сайте я веду блог на английском языке по безопасности и программированию для Web;

- **www.securitylab.ru** — отличный сайт по безопасности, где можно почитать много интересных статей и пообщаться на форуме с очень опытными людьми в сфере безопасности;
- **www.void.ru** — один из старейших сайтов по безопасности;
- **www.xakep.ru** — сайт знаменитого журнала "Хакер", в котором работал и ваш покорный слуга.

Сайтов по безопасности в Рунете очень много, но для начала этого будет достаточно. Не буду выделять какие-то, как лучшие, я рекомендую читать разные сайты, чтобы увидеть различные точки зрения.

Благодарности

В каждой своей книге я благодарю тех, кто помогает мне в работе. Не устану благодарить своих родных и близких (жену, детей, родителей), которые ежедневно окружают меня и терпят мои исчезновения в виртуальной реальности. Я вас всех люблю и рад, что вы у меня есть.

Отдельная благодарность Александру Лозовскому, с которым мы дружим уже долгие годы, именно его рецензии можно наблюдать на задней обложке большинства моих книг.

Отдельная и особая благодарность издательству "БХВ-Петербург" и всем его сотрудникам, которые помогали мне в создании этой книги.

Хочу поблагодарить всех моих читателей и Вас, за то, что купили эту книгу, а не скачали из Интернета нелегальную копию, и надеюсь, что эта работа вам понравится. Мы постарались сделать все необходимое, чтобы книга была интересной и никто не пожалел потраченных на нее денег.

Если возникнут вопросы или пожелания по улучшению данной книги, то вы всегда можете связаться со мной через мой сайт **www.flenov.info**.

Глава 1

Основы безопасности

В этой главе мы познакомимся с исходными положениями, которые позволят нам узнать, как взламываются Web-сайты и каким образом идет поиск уязвимостей. В начале главы поверхностно будет затронута тема социальной инженерии, в дальнейшем мы не будем использовать ее (ну, может, совсем чуть-чуть) для достижения необходимого результата. После мы познакомимся с основами взлома и рассмотрим возможные варианты защиты от него.

Содержимое этой главы пересекается с некоторыми отрывками из других моих книг, потому что мне уже много приходилось писать о безопасности. В этой главе я собрал самое интересное из своих предыдущих работ (но не все, а только то, что касается веба), дополнил и обновил информацию с учетом текущих реалий. Даже если вы читали мои предыдущие книги, эта глава не должна стать для вас скучным чтивом.

1.1. Социальная инженерия

Социальная инженерия — это очень мощное оружие, которое может срабатывать даже там, где программы на сервере написаны идеально, потому что она использует самое слабое звено — человека. Наверное, каждая уязвимость связана с человеческим фактором, ведь серверные программы, в которых мы будем искать уязвимости, написаны человеком и именно он делает ошибку, которая приводит к взлому. В данном случае, социальная инженерия ищет слабое место (можно сказать, уязвимость, если проводить аналогию с программами) в человеке.

С помощью социальной инженерии происходило большинство наиболее громких взломов и создавались самые известные вирусы. Вспомните вирус Анны Курниковой, когда пользователям приходило письмо с вложением и предложением посмотреть фотографию обнаженной Анны. Это тоже социальная инженерия, которая играет на слабостях человека. Я думаю, что любопытство мужчин, которые запускали прикрепленный файл и таким образом

заражали свой компьютер, помогло распространению этого вируса. А ведь на тот момент мужская половина была бóльшей частью пользователей Интернета. В данном случае использовалась слабость (можно снова назвать уязвимостью) человека — любопытство и похоть.

Социальная инженерия основана на психологии человека и использует его слабые стороны. С ее помощью хакеры заставляют жертву делать то, что им нужно: заражают компьютеры, получают пароли. Сколько раз я слышал про украденные номера кредитных карт с помощью простых почтовых сообщений. Пользователь получает письмо с просьбой сообщить свой пароль, потому что база данных банка порушилась из-за погодных условий, проказ хакера или неисправности оборудования. Ничего не подозревающие пользователи всегда сообщают данные, потому что боятся потерять информацию.

Да, в последнее время доверчивость у пользователей Интернета уменьшается благодаря СМИ. Теперь уже все сложнее найти человека, который откроет свой пароль в ответ на поддельное письмо от службы поддержки. Сейчас наоборот, пользователи боятся использовать некоторые защищенные и очень полезные сервисы. Но и хакеры не дремлют и ищут все новые и новые методы.

Есть и такие хакеры, которые редко придумывают что-либо новое, а используют старые и проверенные способы. И, несмотря на это, всегда находятся жертвы, которые попадаются на удочку. Я пользуюсь Интернетом уже очень долгое время и ежедневно получаю десятки писем с просьбой запустить файл для обновления защиты или для того, чтобы увидеть что-то интересное. А ведь большинство отправителей — пользователи зараженных компьютеров. Значит, кто-то открывает такие вложения.

Несмотря на широкое использование защитных программных комплексов и антивирусных программ, количество вирусов не уменьшается, а если и уменьшается, то не сильно. Каждый день в Интернете появляются новые пользователи, которые еще ничего не знают о мерах предосторожности. Именно они чаще всего попадаются на различные уловки.

Итак, давайте рассмотрим некоторые способы, которыми пользуются хакеры. Это поможет вам распознавать их и выделять попытки психологического воздействия от простого общения с людьми. Помните, что социальная инженерия максимально сильна в Интернете, когда вы не можете воочию оценить намерения своего собеседника.

В последнее время снова начинает набирать ход метод взлома через смену пароля. Я стал больше получать писем с просьбой обновить свои реквизиты на Web-странице банка, и при этом ссылка из письма указывает совершенно на другой Web-сайт, где введенные пользователем данные попадают в руки хакеру.

Недавно мне пришло письмо, в котором использовался очень старый и давно забытый способ социальной инженерии. Письмо имело примерно следующее содержание: "Здравствуйте. Я администратор хостинговой компании XXXXX. Наша база была подвержена атаке со стороны хакера, и мы боимся, что некоторые данные были изменены. Просьба просмотреть следующую информацию, и если что-то неверно, то сообщите мне, я восстановлю данные в базе".

После этого следовало перечисление данных обо мне, которые легко получить с помощью службы Whois. На любом Web-сайте регистрации доменов есть такая служба, позволяющая определять имя владельца домена. Хакер воспользовался этим и указал в письме всю найденную информацию. Помимо этого он указал еще два параметра: имя пользователя и пароль. Конечно же, эти данные хакер не мог знать, поэтому здесь были неверные значения. Кое-кто из пользователей при получении таких писем теряется и, волнуясь за свой Web-сайт, пишет ответное письмо, в котором сообщает лжеадминистратору, а точнее — хакеру, свои правильные параметры доступа.

Данный метод использует хороший психологический прием: сначала приводится достоверная информация, и только в параметрах доступа заложена ошибка. Таким образом завоевывается расположение и доверие жертвы, и вероятность получить пароль достаточно высока, если пользователь не знаком с таким принципом социальной инженерии. Это подтверждает множество знаменитых взломов в 80-х гг. прошлого столетия.

Сейчас количество взломов этим методом сократилось, но это может быть затишьем перед бурей. Пользователи могут расслабиться, и атака снова станет популярной. Ведь все хорошее — это хорошо забытое старое. Если немного модифицировать подход, чтобы пользователи сразу не заметили подвоха, то атака может стать очень эффективной.

Задача хакера — войти в доверие к защищающейся стороне и узнать пароли доступа. Для этого используются психологические приемы воздействия на личность. Человеку свойственны любопытство, доверчивость и чувство страха. Любое из этих чувств может стать причиной утери информации.

Благодаря излишнему любопытству мы верим призывам открыть прикрепленный к письму файл и самостоятельно запускаем на своем компьютере вирусы. В силу нашей доверчивости хакерам удается узнать секретную информацию. Но самые сильные эмоции вызывает страх. Именно на страхе и боязни потери паролей основана большая часть атак, с помощью которых пользователя вынуждают сказать необходимые сведения.

Еще две слабости, которые очень часто приводят к положительному результату, — жадность и алчность. Деньги портят людей, а хакеры умеют этим пользоваться. Наилучший результат достигается тогда, когда хакер использу-

ет сразу несколько слабостей, например, и страх, и любопытство одновременно.

Еще один пример из личной жизни. Однажды мне насолил один хакер, мне просто лень было искать какие-то ошибки в сценариях его Web-сайта (да и сайт был слишком прост, чтобы что-то найти), а хотелось как-то проучить. С помощью Whois я узнал, на каком Web-сервере находится Web-сайт моего обидчика, и нашел адрес электронной почты службы поддержки его провайдера. Далее действия были простыми: я завел временный почтовый ящик, чтобы не выдать своего реального адреса, и попросил своего друга, чтобы он отправил письмо примерно следующего содержания: "Здравствуйте. Вас беспокоит капитан милиции Василий Пупкин из Управления К. Ленинского района города Москвы. На вашем сервере находится Web-сайт хакеров с информацией, нарушающей УК РФ. Просим вас самостоятельно принять меры, дабы не усложнять ситуацию. С уважением, Василий Пупкин".

Вот так вот скромненько, но со вкусом. Конечно, вместо "Василий Пупкин" было имя более похожее на реальное, поэтому администратор поверил — Web-сайт тут же был закрыт и оставался недоступным в течение двух часов. После его работа была возобновлена, когда в хостинговой компании поняли, что перед ними обман.

Хакеры пользуются социальной инженерией незаметно, но эффективно. Вы даже не почувствуете подвоха, когда у вас попросят пароль или секретную информацию, и послушно все отдадите. Чтобы не попасться на крючок, вы должны иметь представление о том, какие методы социальной инженерии могут использоваться для достижения необходимой цели. С основными методами можно познакомиться в книге самого знаменитого хакера — Кевина Митника [1].

1.2. Природа взлома

Универсального способа взлома Интернета (а точнее Web-сайтов) не существует. Если бы такое средство существовало, Интернет бы уже охватили анархия и беспредел, а все сайты были бы взломаны. Вместо этого каждый раз приходится искать свое решение, которое откроет необходимую дверь для определенного сайта.

Да, есть атаки, которые могут уничтожить любую защиту: DDoS (Distributed Denial of Service, распределенная атака на отказ в обслуживании) или подбор паролей, но затраты на проведение этих атак могут оказаться слишком большими, хотя они не требуют много ума и доступны для реализации даже новичку. За взлом сервера с помощью перебора паролей или за DDoS-атаку ха-

керы никогда не будут признаны общественностью как профессионалы, поэтому к подобным методам прибегают только в крайних случаях и, в основном, начинающие взломщики.

Почему количество атак с каждым годом только увеличивается? Я не говорю сейчас о свершившихся или удачных атаках, я говорю о попытках. Раньше вся информация об уязвимостях хранилась на закрытых BBS (Bulletin Board System, электронная доска объявлений) и была доступна только избранным. К этой категории относились и хакеры, совершавшие безнаказанные атаки, потому что уровень их знаний и опытности был достаточно высок. Проникнуть на такую BBS непосвященному или новичку было очень сложно, а чаще всего просто невозможно. Информация об уязвимостях и программы для реализации атак были доступны ограниченному количеству людей.

В настоящее время сведения об уязвимостях стали практически общедоступными. Существует множество сайтов, где можно узнать не только подробные инструкции о взломе, но и найти программу, которая вообще без специализированных знаний по нажатию кнопки будет производить атаку. В некоторых случаях достаточно только указать адрес Web-сайта, который вы хотите взломать, нажать на "волшебную" кнопку, и компьютер сделает все необходимое сам без вашего вмешательства, при этом вы абсолютно не будете знать, как это произошло.

С одной стороны, информация действительно должна быть открытой. Администраторы, зная методы взлома, могут построить соответствующую защиту. Другое дело, что далеко не каждый следит за тенденциями в безопасности, и далеко не все администраторы отрабатывают свою зарплату, строя безопасные сети.

С другой стороны, если закрыть Web-сайты, на которых содержится информация об уязвимостях, то количество атак резко сократится. Большинство взломов совершается именно непрофессионалами, которые нашли где-то программу для совершения злодеяний и выполнили ее.

Лично я разрываюсь между двух огней и не могу проголосовать за открытость или закрытость информации. С одной точки зрения, информация должна быть открыта, а администраторы и программисты должны быть внимательнее и быстрее реагировать на найденные ошибки, а с другой — ее лучше закрыть, чтобы у злоумышленников не было соблазна использовать готовые программы.

Наверное, я все же проголосую за открытость, но при этом правоохранительные органы должны лучше реагировать на действия вандалов, а администраторы должны лучше контролировать безопасность. Я даже за регулирование Интернета, иначе возникает анархия. Каждое общество требует разумного

управления. Это не значит, что за каждым щелчком нужно следить, это значит, что взломы должны наказываться по определенным законам. Мы должны вести себя цивилизованно, иначе, как раковая опухоль уничтожает живой организм, мы уничтожим сами себя.

Безусловно, интересно наблюдать за тем, как две команды хакеров воюют между собой, взламывая Web-сайты друг друга, но все должно иметь свой предел. Каков этот предел, я судить не могу. Никто не может вынести решение, каким быть Интернету, потому что на данный момент он свободен и в каждой стране подчиняется своим законам. Но Интернет — это единое общество, а закон должен быть для всех единым. Пока не будет закона, его соблюдения и контроля, анархия будет продолжаться.

1.3. Исследование

Допустим, что у вас на примете есть сервер или компьютер, который нужно взломать или протестировать на защищенность от взлома. С чего нужно начинать? Что сделать в первую очередь?

Четкой последовательности действий нет. Взлом — это творческий процесс, а значит, и подходить к нему надо с этой точки зрения. Нет определенных правил, и нельзя все подвести под один шаблон.

Самое первое, с чего начинается взлом или тест ОС на уязвимость, — сканирование портов. Для чего? А для того, чтобы узнать, какие службы (в Linux это демоны) установлены в системе. Каждый открытый порт — это программа, установленная на сервере, к которой можно подключиться и выполнить определенные действия. Например, на 21-м порту работает служба FTP. Если вы сможете к ней подключиться, то вам станет доступной возможность скачивания и загрузки файлов. Но это только, если вы будете обладать соответствующими правами на удаленном сервере.

Сначала следует просканировать первые 1024 порта. Среди них очень много стандартных служб типа FTP, HTTP, Telnet и т. д. Открытый порт — это дверь с замочком для входа на сервер. Чем больше таких дверей, тем больше вероятность, что какой-то засов не выдержит натиска и откроется, поэтому на сервере должно быть запущено только то, что необходимо.

Будет лучше, если вы установите на сервер только те программы, которые реально будут использоваться. Все остальное лучше не запускать, запрещать, а лучше вообще не устанавливать, чтобы хакер не смог самостоятельно их запустить и использовать.

У хорошего администратора открыты только самые необходимые порты. Например, если это Web-сервер, не предоставляющий доступ к электронной

почте, то нет смысла держать почтовые службы. Должен быть открыт только 80-й порт, на котором как раз и работает Web-сервер. Все остальные порты должны быть не просто закрыты сетевым экраном, а лучше, если соответствующие службы совсем не будут установлены.

Распространенная ошибка администраторов: установлю на всякий случай все, просто не буду запускать или прикрою сетевым экраном. Это очень серьезная ошибка. Если хакер проникнет на вашу систему, то он запустит остановленную службу или приоткроет сетевой экран, чтобы воспользоваться уже запущенной. Я об этом говорю уже очень давно, и то, что Microsoft движется в этом направлении, говорит о том, что я верно мыслю. Уверен, это не я подсказал сотрудникам Microsoft, что не нужно устанавливать лишнее, и об этом говорю не только я. Сначала IIS (Internet Information Server) стал устанавливаться на компьютеры в минимальной конфигурации. Теперь и MS Windows Server устанавливается в минимальной конфигурации, а вы можете добавлять роли и только нужные вам.

Хороший сканер портов определяет не только номер открытого порта работающего на удаленной системе сервиса, но и показывает название работающей на нем службы (жаль, что не настоящее, а только имя возможного сервера). Так, для 80-го порта будет показано "http". Если сканер не показывает имен служб, то в ОС Windows их можно посмотреть в файлах protocol и services из каталога C:\WINDOWS\system32\drivers\etc. Просто откройте их в Блокноте или любой другой программе просмотра текстовых файлов. В результате вы увидите что-то похожее на следующий список:

```
echo        7/tcp
echo        7/udp
discard     9/tcp     sink null
discard     9/udp     sink null
systat      11/tcp     users          #Active users
systat      11/tcp     users          #Active users
daytime     13/tcp
daytime     13/udp
qotd        17/tcp     quote          #Quote of the day
qotd        17/udp     quote          #Quote of the day
```

Файл имеет следующую структуру:

<служба> <номер порта>/<протокол> [псевдонимы...] [#<комментарий>]

Но не забывайте, что это только описание стандарта, который легко нарушить. Администратор без проблем может запустить Web-сервер не на 80-м порту, а на 21-м, и сканер портов напишет нам, что это FTP-сервер.

Остановитесь и посмотрите сейчас файл services. Здесь описаны наиболее распространенные на данный момент службы и порты, на которых они работают. Если вы еще не знакомы с этими номерами, то следует хотя бы что-то из этого оставить в памяти, чтобы знать потом, что искать на атакуемой системе. Я рекомендую обратить внимание и запомнить, что на портах 1433 и 1434 протоколов TCP и UDP работает Microsoft SQL Server и Microsoft SQL Monitor. На порту 1512 работают WINS (Microsoft Windows Internet Name Service, служба имен Интернета для сетей Windows), которая далеко не без изъяна. Я весь файл приводить не буду, потому что он большой, а вы и без меня сможете в любой момент его посмотреть.

Но существуют программы, которые не доверяют стандарту и проверяют полученную информацию самостоятельно. Как это определить? Да очень легко:

- по строке приветствия, которая возвращается при подключении к порту. Большая часть служб при подключении возвращает сообщение, которое содержит название и версию службы. Позже мы еще будем говорить о том, что это сообщение может быть подделано;
- по ответу на подключение или по ответу на команду. Например, подключившись к 80-му порту, можно попробовать отправить серверу HTTP команду, и если сервер ответит корректно, то перед нами именно Web-сервер. Если мы увидим ошибку, то нас пытаются обмануть;
- службы в ответ на подключения присылают разные пакеты.

Желательно, чтобы сканер умел сохранять результат своей работы в какомнибудь файле сам или позволял получить данные в виде текста и даже распечатывать. Если этой возможности нет, то придется переписать все вручную и положить на видное место, чтобы не забивать мозг лишней информацией. В дальнейшем вам пригодится каждая строчка этих записей.

После этого можно начинать сканировать порты свыше 1024. Здесь стандартные службы встречаются редко. Зачем же тогда сканировать? А вдруг кто-то до вас уже побывал на этом месте и оставил открытую дверку или установил на сервер троянскую программу. Большинство троянских программ держит открытыми порты свыше 1024, поэтому если вы администратор и нашли открытый порт в этом диапазоне, необходимо сразу насторожиться. Ну, а если вы взломщик, то нужно узнать имя троянской программы и найти для нее клиентскую часть, чтобы воспользоваться ею для управления чужой машиной.

Среди служб, использующих порты выше 1024, встречаются и некоторые коммерческие, например, СУБД (система управления базами данных). Дело в том, что номера из первой тысячи уже давно распределены, и использовать

какой-то из этого диапазона достаточно проблематично, поэтому современные службы эксплуатируют весь диапазон номеров до 65 535.

Первые 1024 порта в ОС Linux обладают еще одним очень важным свойством: запустить службу, работающую на таком порту, может только пользователь с правами администратора (для UNIX-систем это пользователь с правами root). Таким образом, система гарантирует, что службы, работающие на портах ниже 1024, запущены администратором. Они являются наиболее критичными с точки зрения безопасности сервера, поэтому рядовые пользователи не должны иметь права их запускать.

Если после сканирования вы нашли программу, через которую можно получить полный доступ к серверу, то на этом взлом может закончиться. Жаль, что такое происходит очень и очень редко, и чаще всего нужно затратить намного больше усилий.

Хорошо было во времена появления троянской программы Back Orifice, когда один хакер заражал компьютер пользователя, а другой без проблем мог воспользоваться уже готовым взломом. В настоящее время по Интернету гуляет слишком большое количество троянских программ, которые используют разные порты и чаще всего даже позволяют настраивать номер порта, на котором они будут работать. А если серверная часть программы защищена паролем, то воспользоваться чужим трудом будет проблематично.

С другой стороны, в современном мире большинство компьютеров оснащены сетевым экраном, который может не позволить воспользоваться даже работающей троянской программой. Сетевые экраны — мощное средство в мире обороны и серьезное препятствие для взломщика.

1.3.1. Определение типа операционной системы

Сканирование — это всего лишь начальный этап, который дал вам информацию для размышления, особенно для нас. Ведь мы рассматриваем безопасность Web-серверов, а значит, нас больше всего интересует 80-й порт. Так зачем нам нужно было сканировать остальные? Это поможет нам узнать, какая ОС установлена на сервере. Ведь если на сервере работает MS SQL Server, то там, скорее всего, стоит Windows.

Желательно иметь сведения о версии, но это удается выяснить не всегда, да и на первых порах изучения системы можно обойтись без конкретизации. Главное — иметь четкое представление об используемой платформе: Windows, Linux, BSD, Mac OS или др. От этого зависит очень многое:

- какие программы могут быть установлены на сервере;
- какие команды можно выполнять;

- где находится информация о пользователях и их паролях;
- где может быть установлена ОС (в какой папке или в каком разделе);
- какие существуют уязвимости для данной ОС.

Как определяется тип ОС? Для этого есть несколько способов. Помимо работающих на удаленной системе сервисах, о типе системы мы можем судить:

- **По реализации протокола TCP/IP.** Это низкоуровневый метод, который работает на уровне пакетов, передаваемых от клиента к серверу. В различных ОС по-разному организован стек протоколов. В основном этот вывод расплывчатый: Windows или Linux. Точную версию таким образом узнать невозможно, потому что в Windows 2000/XP/2003 реализация протокола практически не менялась, и отклики будут одинаковыми. Даже если программа определила, что на сервере установлен Linux, то какой именно дистрибутив, сказать будет сложно. И поэтому такая информация — это только часть необходимых данных для взлома.
- **По ответам служб.** Допустим, что на сервере жертвы есть анонимный доступ по FTP. Вам нужно всего лишь присоединиться к нему и посмотреть сообщение при входе в систему. По умолчанию в качестве приглашения используется надпись типа: "Добро пожаловать на сервер FreeBSD4.0, версия FTP-клиента X.XXX". Если вы такое увидели, то еще рано радоваться, т. к. не известно, правда это или нет. Хороший администратор изменяет строку приветствия, чтобы не упрощать взломщику жизнь. Могут не просто изменить, но и вводить в заблуждение, когда на Windows-сервере появится приглашение, например, Linux. В этом случае злоумышленник безуспешно потратит очень много времени в попытках взломать Windows, стараясь использовать ошибки Linux. Поэтому не очень доверяйте надписям и старайтесь их перепроверить другими способами.
- **По социальной инженерии.** Если вы хотите взломать сервер хостинговой компании, то можно обратиться с письменным запросом об установленных у нее серверах в службу поддержки. Как правило, такая информация не скрывается, но бывают случаи откровенной лжи. Возможно, что эти сведения будут лежать на главной Web-странице, но даже их следует проверить. Например, некоторые компании пишут на Web-сайте, что у них используется последняя и самая безопасная версия ОС FreeBSD, но при первом же осмотре оказывается, что на сервере стоит Linux, а это достаточно большая разница. Возможно, компании просто скрывают версию ОС для того, чтобы скрыть использование нелицензионного программного обеспечения. Ведь даже в мире Linux есть платные дистрибутивы. А может быть имеет место обман, ведь BSD-системы считаются более за-

щищенными и в openbsd.org на момент написания этих строк нашли всего две удаленные уязвимости.

Чтобы вас не обманули, обязательно обращайте внимание на используемые на сервере службы, например, в Linux, скорее всего, не будут работать Web-страницы, созданные по технологии ASP. Такие вещи подделывают редко, хотя это и возможно — достаточно использовать расширение asp для хранения PHP-сценариев и перенаправлять их интерпретатору PHP. Таким образом, хакер увидит, что на сервере работают файлы ASP, но реально это будут PHP-сценарии.

1.3.2. Определение имен работающих служб

Определение типа ОС — это только начальный этап. После этого необходимо переходить к определению имен служб, которые работают на сервере. Если открыт 80-й порт, то необходимо узнать, какой установлен Web-сервер: Apache или IIS. От используемой службы и ее версии зависит, как ее надо взламывать. Например, некоторые версии определенных служб могут содержать одну ошибку, а другие версии — другую ошибку. А бывают даже случаи, когда ошибок вообще нет. Хотя нет, таких случаев практически не бывает. Ошибки есть, просто их еще никто не нашел и нужно поискать, или подождать, когда найдут другие.

Иногда хакеры определяют версию ОС или установленных служб по наличию ошибок. Например, если Web-сервер IIS версии 5.0 содержал определенную ошибку при получении слишком большого пакета, то можно попытаться отправить такой пакет, и если в ответ мы получим ошибку, то перед нами именно IIS 5.0. Вроде бы все хорошо и метод определения достаточно точный, но не совсем. Дело в том, что наличие ошибки может зависеть и от конфигурации ОС или самой службы. Известная вам ошибка может проявляться только при определенных условиях, и если ошибка не проявилась, то, возможно, не соблюдены необходимые условия *(см. разд. 1.3.4)*.

Защищающая сторона должна как можно меньше дать информации противнику, чтобы у него меньше было возможности взломать сервер. Да, прятать нужно не только название ОС, но и названия служб. Например, если подделать Apache под IIS, то хакер будет пытаться взломать не тот Web-сервер, используя не те программы, что усложнит его задачу.

Задача хакера — четко определить, что же он взламывает. Без этого производить в дальнейшем какие-либо действия будет сложно, потому что он даже не будет знать, какие команды ему доступны после вторжения на чужую территорию, где искать системные файлы и пароли, а также какие исполняемые файлы нужно загружать на сервер.

1.3.3. Использование эксплоитов

Итак, теперь вы в курсе, какая на сервере установлена ОС, какие порты открыты и какие именно службы работают на этих портах. Всю добытую информацию нужно записать в удобном для восприятия виде: в файле или хотя бы на бумаге. Главное, чтобы было комфортно работать.

Не ленитесь записывать все собранные данные. Помните, что даже компьютеры иногда сбоят, а человеческий мозг делает это регулярно. Самое интересное, что чаще всего забывается наиболее необходимое. Ну, а если вы взломаете сервер, то записи смогут послужить доказательством содеянного в суде. Возможно, это сможет дисциплинировать вас и не позволит пойти дальше исследования, т. е. не позволит сотворить преступление.

Все поняли? Тогда на этом можно остановиться. У вас есть достаточно информации для простейшего взлома с помощью ошибок в ОС и службах, установленных на сервере. Просто посещайте регулярно **www.securityfocus.com**, а российскому пользователю могу посоветовать Web-сайт **www.securitylab.ru** (рис. 1.1) или **www.void.ru**. Я больше предпочитаю первый из них. Не знаю почему, просто предпочитаю. Именно здесь нужно искать информацию о новых уязвимостях. Уже давно известно, что на большей части серверов (по разным источникам от 70 до 90%) ошибки не устраняются или устраняются, но с большими задержками (обычно после взлома). Поэтому проверяйте все найденные ошибки на жертве, возможно, что-то и сработает.

После появления новой уязвимости на Web-сайте **www.securitylab.ru** тут же под этой новостью разгорается бурное обсуждение того, как ее использовать и где взять необходимый эксплоит (программу, позволяющую использовать уязвимость, иногда ее называют "сплоит"). С практической точки зрения, Web-сайт хороший и нужный. Когда я использовал на своем Web-сайте распространенные форумы типа phpBB, то благодаря **www.securitylab.ru** оперативно узнавал об уязвимостях и чаще всего успевал устранять найденные в ИТ-мире ошибки.

С другой стороны, на форуме гуляет очень много скрипткидди (людей, которые используют для взлома чужие программы, чаще всего это молодые ребята, которые только хотят стать хакерами), от которых часто слышатся высказывания типа: "Научите меня использовать эту уязвимость". А ведь если научат, то он пойдет крушить все Web-сайты подряд — надо же где-то попробовать полученные знания. Именно такие люди представляют наибольшую опасность.

На форуме по безопасности **www.securitylab.ru** встречаются и профессионалы, которые не особо "гнут пальцы", а показывают и объясняют, как защититься. Возможно, они и сами ломают что-то, но не показывают этого. Лично я в основном нахожусь там как наблюдатель (отслеживаю тенденции), а если и оставляю сообщения, то не под своим именем.

Рис. 1.1. Web-сайт **www.securitylab.ru**

Ошибки в программах проявляются достаточно часто, но если защита сервера, который вы решили взломать, в данный момент близка к совершенству (минимум служб, и установлены все последние обновления), то придется ждать появления новых ошибок и эксплоитов к установленным на сервере службам. Как только увидите что-нибудь интересное, сразу скачайте эксплоит (или напишите свой) и воспользуйтесь им, пока администратор не успел закрыть уязвимость. А хороший администратор закрывает ошибки быстро, сразу после их появления, а если есть возможность, то автоматизирует этот процесс.

Мы не будем рассматривать реальные примеры ошибок серверов и сервисов, потому что эти ошибки латаются очень быстро, и к моменту появления книги на полках магазинов информация станет не просто устаревшей, она станет подобной каменному веку, поэтому не имеет смысла тратить время на рассмотрение подобного класса ошибок. В данном случае нужна только актуальная информация, и ее можно получить в Интернете.

1.3.4. Автоматизация

Практически каждый день специалисты по безопасности находят в разных системах недочеты, дыры или даже пробоины в системе безопасности. Все эти материалы выкладываются в отчетах BugTraq (бюллетень безопасности). Я уже советовал посещать Web-сайт **www.securityfocus.com**, чтобы следить за новостями, и сейчас не отказываюсь от своих слов. Все новое, действительно, можно найти там, но ведь есть целый ворох старых уязвимостей, которые существовали и, возможно, еще не закрыты. Как же поступить с ними? Неужели придется качать все эксплоиты и проверять каждый на работоспособность? Ну, конечно же, нет. Для этого используются сканеры безопасности, наиболее распространенные из них: SATAN, Internet Scanner, NetSonar, CyberCop Scanner.

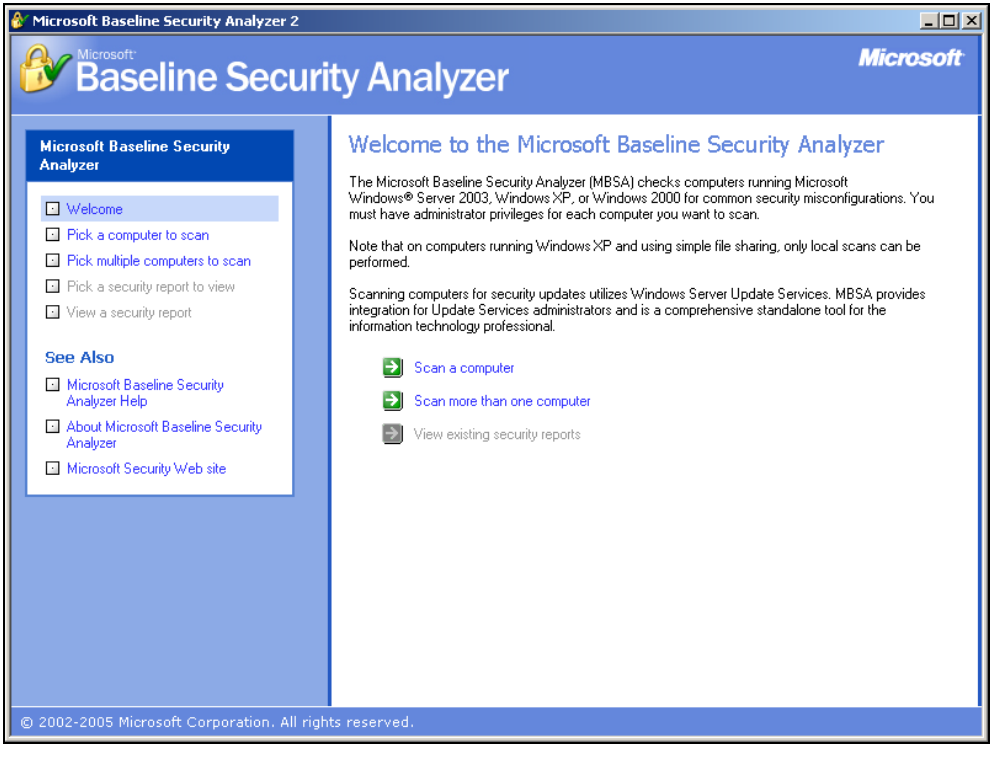

Рис. 1.2. Программа Microsoft Baseline Security Analyzer

Недавно появилась даже программа от Microsoft — Microsoft Baseline Security Analyzer (рис. 1.2), которая проверяет систему на наличие установленных

критических обновлений, говорят, она даже бесплатная, что не удивительно. После запуска программы на тестирование она подключается к серверу Microsoft и скачивает последние модули для тестирования безопасности, как это делают антивирусные программы. Само сканирование происходит достаточно быстро, но большая часть тестов банально проверяет наличие последних обновлений.

С точки зрения хакера, программа Microsoft Baseline Security Analyzer бесполезна, потому что основные тесты проводятся локально. А вот с точки зрения администратора, программа очень удобна и поможет быстро увидеть основные недостатки безопасности.

Я не стану рекомендовать для использования какой-то определенный продукт. Пока не существует такого сканера безопасности, в котором была бы база абсолютно всех потенциальных уязвимостей. Поэтому скачивайте все, что попадется под руку, и тестируйте систему всеми доступными программами. Возможно, что-то вам и пригодится. Но обязательно обратите внимание на продукты компании ISS (Internet Security Systems, **www.iss.net**), потому что сканеры этой фирмы (Internet Scanner, Security Manager, System Scanner и Database Scanner) используют три метода сканирования, о чем мы поговорим чуть позже. Сотрудники ISS работают в тесном контакте с Microsoft и постоянно обновляют базу данных уязвимостей. А с недавнего времени, эта компания принадлежит IBM. Но, несмотря на то, что продукты этой фирмы лучшие, я советую использовать хотя бы еще один сканер другого производителя.

Компания ISS разработала целый комплект утилит под общим названием SAFEsuite. В него входят не только компоненты проверки безопасности системы, но и модули выявления вторжения и оценки конфигурации основных серверных ОС.

Сканеры безопасности, как и антивирусы, защищают хорошо, но только от старых уязвимостей. Любой новый метод взлома не будет обнаружен, пока вы не обновите программу. Поэтому я не рекомендую целиком и полностью полагаться на отчеты автоматизированного сканирования, а после работы программы самостоятельно проверить наличие последних уязвимостей, описанных в каком-либо BugTraq.

С помощью автоматизированного контроля очень хорошо производить первоначальную проверку, чтобы убедиться в отсутствии старых ляпсусов. Если ошибки найдены, то нужно обновить уязвимую программу, ОС или службу, или поискать на том же Web-сайте **www.securityfocus.com** способ устранения ошибки. Почти всегда вместе с описанием уязвимости дается вакцина, позво-

ляющая залатать прореху. Вакцину может предложить и программа сканирования, если в базе данных есть решение проблемы для данного случая.

Почему даже после лучшего и самого полного сканирования нельзя быть уверенным, что уязвимостей нет? Помимо ошибок надо принимать во внимание еще и фактор конфигурации. На каждом сервере могут быть свои настройки, и при определенных условиях легко находимая вручную уязвимость может остаться незаметной для автоматического сканирования. На сканер надейся, а сам не плошай. Так что продолжайте тестировать систему на известные вам ошибки самостоятельно.

Каждый сканер безопасности использует свои способы и приемы поиска уязвимостей, и если один из них ничего не нашел, то другой может отыскать. Специалисты по безопасности любят приводить пример с квартирой. Допустим, что вы пришли к другу и позвонили в дверь, но никто не открыл. Это не значит, что дома никого нет, просто хозяин мог не услышать звонок, или он не работал. Но если позвонить по телефону, который лежит в этот момент возле хозяина, то он возьмет трубку. Может быть и обратная ситуация, когда вы названиваете по телефону, но его не слышно, а на звонок в дверь домочадцы отреагируют.

Точно так же происходит и при автоматическом сканировании: один сканер может позвонить по телефону, а другой — постучит в дверь. Они оба хороши, но в конкретных случаях при разных конфигурациях сканируемой машины могут быть получены различные результаты.

Существуют три метода автоматического определения уязвимости: сканирование, зондирование и имитация. В первом случае сканер собирает информацию о сервере, проверяет порты, чтобы узнать, какие на нем установлены службы, и на основе их выдает отчет о потенциальных ошибках. Например, сканер может проверить сервер и увидеть на 21-м порту работающий FTP-сервер. По строке приглашения (если она не была изменена), выдаваемой при попытке подключения, можно определить его версию, которая сравнивается с базой данных. И если в базе есть уязвимость для данного FTP-сервера, то пользователю выдается соответствующее сообщение.

Сканирование — далеко не самый точный процесс, потому что автоматическое определение легко обмануть, да и уязвимости может не быть. Некоторые погрешности в службах проявляются только при определенных настройках, т. е. при установленных вами параметрах ошибка не обнаружится.

При зондировании сканер не обследует порты, а проверяет программы на наличие в них уязвимого кода. Этот процесс похож на работу антивируса, который просматривает все программы на наличие соответствующего кода.

Ситуации похожие, но искомые объекты разные. Метод хорош, но одна и та же ошибка может встречаться в нескольких программах. И если код в них разный, то сканер ее не обнаружит.

Во время имитации сканер безопасности моделирует атаки из своей базы данных. Например, в FTP-сервере может возникнуть переполнение буфера при реализации определенной команды. Сканер не будет выявлять версию сервера, а попробует выполнить инструкцию. Конечно же, это приведет к зависанию, но вы точно будете знать о наличии или отсутствии ошибки на нем.

Имитация — самый долгий, но надежный способ, потому что если таким образом удалось взломать какую-либо службу, то и у хакера это получится. При установке нового FTP-сервера, который еще не известен сканерам безопасности, он будет опробован на уже известные ошибки других серверов. Очень часто программисты разных фирм допускают одни и те же ошибки. При использовании сканирования анализатор может без проблем найти уязвимость.

Когда проверяете систему, обязательно отключайте сетевые экраны. Если доступ заблокирован, то сканер безопасности не сможет протестировать нужную службу. В этом случае он сообщит, что ошибок нет, но реально они могут быть. Конечно же, это не критичные ошибки, потому что они закрыты сетевым экраном, но если хакер найдет потайной ход и обойдет сетевой экран, то уязвимость станет опасной.

Дайте сканеру безопасности все необходимые права и доступ к сканируемой системе. Например, некоторые считают, что наиболее эффективно удаленное сканирование выполняется, когда атака имитируется по сети, ведь хакер тоже будет находиться на удаленной машине. Это правильно, но сколько времени понадобится на проверку стойкости паролей для учетных записей? Очень много! А сканирование реестра и файловой системы станет невозможным. Поэтому локальный контроль может дать более быстрый результат. А что, если хакер уже получил доступ к системе и пытается поднять свои привилегии?

При дистанционном сканировании только производится попытка прорваться в сеть. Такой анализ может указать на стойкость защиты от нападения извне. Но по статистике большинство взломов происходит изнутри, когда зарегистрированный пользователь поднимает свои права и тем самым получает доступ к запрещенной информации. Хакер тоже может иметь какую-нибудь учетную запись с минимальным статусом и воспользоваться уязвимостями для повышения прав доступа. Поэтому сканирование должно происходить и дистанционно для обнаружения потайных дверей, и локально для выявления ошибок в конфигурации, с помощью которых можно изменить привилегии.

В случае с Web-серверами данное утверждение тоже имеет место. Дело в том, что сценарии сайта выполняются в системе с правами Web-сервера, которые должны быть минимальными, и в большинстве систем они являются таковыми по умолчанию. Найдя уязвимость в сценарии, хакер получает возможность выполнять в системе сценарии с этими правами, и следующая его задача — поднять свои права, чтобы увидеть защищенные данные.

Автоматические сканеры безопасности проверяют не только уязвимости ОС и ее служб, но и сложность пароля, и имена учетных записей. В их анализаторах есть база наиболее часто используемых имен и паролей, и программа перебором проверяет их. Если удалось проникнуть в систему, то выдается сообщение о слишком простом пароле. Обязательно замените его, потому что хакер может использовать тот же метод и легко узнает параметры учетной записи.

Сканеры безопасности могут использовать как хакеры, так и администраторы. Но задачи у них разные. Одним нужно автоматическое выявление ошибок для последующего применения, а вторые используют его с целью поиска уязвимости с последующим устранением ошибки, причем желательно это сделать раньше, чем найдет и будет использовать ее хакер.

Ошибки, которые вы можете найти и которые могут привести к взлому, можно разделить на следующие категории:

- ошибки в коде программ. Такие ошибки исправляются простым обновлением программ;
- ошибки конфигурации. В некоторых случаях вполне безобидные программы могут оказаться опасными при определенных настройках. Данные ошибки проявляются из-за непрофессионализма или невнимательности администраторов;
- ошибки в распределении прав доступа. Очень распространенная ошибка, когда пользователю даются излишние права на объекты. Сколько не говорят специалисты по безопасности, а еще очень много компаний, где сотрудники работают в программах под правами администратора и/или с простейшими паролями. На одной из последних работ все пароли администраторов были идентичны имени;
- принудительное понижение безопасности. Например, могла быть включена определенная опция, которая снижает безопасность или размер ключа шифрования для того, чтобы сохранить совместимость со старыми версиями программы. В данном случае необходимо обновить все программы и отказаться от использования старых систем.

Не все автоматические анализаторы разделяют ошибки по типам, но понимать источник ошибки желательно.

1.4. Взлом Web-сервера

При взломе Web-сервера есть свои особенности. Если на нем выполняются CGI, PHP или другие сценарии, то предварительные исследования, которые мы описывали ранее, можно даже опустить. Иногда можно обойтись и без сканирования портов, а вот ОС определить желательно, ведь когда вы получите возможность выполнять команды, то должны знать, какие именно команды может выполнять система.

Для начала нужно просканировать Web-сервер на наличие уязвимых CGI-сценариев. Да, именно просканировать, потому что есть программы, которые автоматизируют этот процесс. Вы не поверите, но опять же по исследованиям различных компаний, в Интернете работает большое количество "дырявых" сценариев. Это связано с тем, что при разработке Web-сайтов в них изначально вносятся ошибки. Нет, не специально. Начинающие программисты очень редко проверяют входящие параметры в надежде, что пользователь не будет изменять код Web-страницы или URL, где Web-серверу передаются необходимые данные для выполнения каких-либо действий. Это стало возможным потому, что программисты понадеялись на добросовестность посетителей. А зря. Посетители очень часто не добросовестные.

Ошибку с параметрами имела одна из знаменитых систем управления Web-сайтом — PHP-nuke. А если быть точнее, за все время существования программы, там нашли не одну ошибку. Хотя в этой программе такие ошибки встречались неоднократно, одна нашумела больше всего. PHP-nuke — это набор сценариев, позволяющих создать форум, чат или новостную ленту и управлять содержимым Web-сайта без знания программирования. Любой пользователь компьютера, даже с небольшим опытом, с помощью нее сможет легко создать свой Web-сайт.

Все параметры в сценариях передаются через адресную строку браузера, и просчет содержался в параметре `ID`. Разработчики предполагали, что в нем будет передаваться число, но не проверяли это. Хакер, знающий структуру базы данных (а это не сложно, потому что исходные коды PHP-nuke доступны), легко мог поместить SQL-запрос к базе данных сервера в параметр `ID` и получить пароли всех зарегистрированных на Web-сайте пользователей. Конечно, пароли будут зашифрованы, но для расшифровки не надо много усилий, это мы рассмотрим чуть позже. Дело в том, что если пользователей много, то велика вероятность, что у кого-то из них будет слабый пароль.

Проблема усложняется тем, что некоторые языки, например, Perl, изначально не были предназначены для использования в Интернете. Из-за этого в них существуют опасные функции для манипулирования системой, и если про-

граммист неосторожно применил их в своих модулях, то злоумышленник может воспользоваться такой неосмотрительностью. Но даже если язык разрабатывался специально для создания Web-сайтов, опасные функции также могут присутствовать и в нем.

Но самая большая уязвимость — неграмотный программист. Из-за нехватки специалистов в этой области программированием стали заниматься все, кому не лень. Многие самоучки даже не пытаются задуматься о безопасности, а взломщикам это только на руку.

1.4.1. Анализатор Web-уязвимостей

Итак, ваша первостепенная задача — запастись парочкой хороших CGI-сканеров. Какой лучше? Ответ однозначный — ВСЕ. Даже самый плохой сканер с минимальными возможностями может найти брешь, о которой неизвестно даже лучшему. А главное, что по закону подлости именно она окажется доступной на сервере. Помимо этого, нужно посещать все тот же Web-сайт **www.securityfocus.com**, где регулярно выкладываются описания уязвимостей различных пакетов программ для Web-сайтов.

На заре Интернета очень часто можно было встретить сканеры с базами данных уязвимостей, но поддержка такой базы данных в наше время — очень дорогое удовольствие, потому что различных сценариев в Интернете очень много, а уязвимостей еще больше (в одном сценарии иногда находят по 10 и более ошибок). Все это отслеживать очень дорого, а платить пользователи за подобную программу не хотят. Хакеры вообще мало за что платят, а администраторам и владельцам сайтов такая база просто не нужна. Их интересуют ошибки только сценариев, установленных у них, и только актуальные ошибки, а не устаревшие.

Из-за этого в мире тестирования Web-уязвимостей более популярны программы-имитаторы, которые имитируют ошибку. На мой взгляд, это наиболее эффективный в данной сфере способ, потому что позволяет искать ошибки не по базе, а на абсолютно любом сайте. Подобный анализатор я реализовал в своей программе *Network Utilities Сеть и безопасность* (**http://russia.cydsoft.com**). В программе есть модуль тестирования безопасности, который ищет на сайтах такие уязвимости, как SQL Injection, XSS и PHP Include.

Начиная с версии 2009-го года, в программе полностью был переработан алгоритм поиска уязвимостей. Раньше программа пыталась найти уязвимость на сайте, отправляя неправильные значения параметров в URL и проверяя результат. Проверка подразумевала поиск в ответе от сервера текста сообщения об ошибке. Я просто прописал в программе все знаменитые ошибки различных языков программирования и баз данных. Именно так и поступают