

Инструменты безопасности с открытым исходным кодом

1. Лекция: Информационная безопасность и программное обеспечение с открытыми исходными текстами: версия для печати и PDA

Когда Том Пауэрс устраивался на работу в качестве системного администратора в энергетическую компанию среднего размера, он знал, что решающим фактором были его навыки в области компьютерной безопасности. За последний год компания несколько раз подвергалась атакам хакеров, а на ее домашнюю страницу помещали непристойные изображения. Руководство хотело, чтобы, помимо обеспечения повседневной работы компьютерной сети, он сделал информацию компании более защищенной от цифровых атак.

После первого же дня работы он понял, что перед ним стоит сложная проблема. В компании отсутствовала даже самая элементарная система безопасности. Соединение с Интернетом, защищенное только обычным маршрутизатором поставщика Интернет-услуг, было широко открыто миру. Общедоступные серверы плохо поддерживались и выглядели так, будто к ним не прикасались с момента установки. А бюджет для исправления ситуации был практически нулевым.

Однако в течение четырех месяцев Том обеспечил устойчивость сети, остановил все атаки, обезопасил точки общего доступа и очистил внутреннюю сеть, а также добавил сервисы, которых раньше не было. Как он смог все это сделать с такими ограниченными ресурсами? Он знал базовые принципы и концепции информационной безопасности (ИБ) и нашел подходящее программное обеспечение (ПО) для выполнения работы. Он разработал описанный далее план усовершенствования системы безопасности компании и методично выполнил его с помощью подходящих защитных инструментов.

Защита периметра

Прежде всего Том создал несколько базовых средств для защиты своей сети от внешнего мира, чтобы затем спокойно заняться безопасностью серверов и внутренней части сети. Он настроил межсетевой экран для соединений с Интернетом, используя программу Turtle Firewall (рассмотренную в [лекции 3](#)). С помощью этой программы и старого сервера, который больше ни для чего не использовался, он сконфигурировал машину так, чтобы разрешить соединения с внешним миром только изнутри сети; все входящие соединения, не запрошенные изнутри, блокировались. Правда, он сделал несколько исключений для общедоступных серверов.

Затыкание дыр

Том знал, что ему необходимо проверить свою сеть на наличие дыр в системе безопасности и определить места, через которые в нее проникают злоумышленники. Хотя теперь межсетевой экран защищал внутренние рабочие станции от случайных вторжений, общедоступные серверы, такие как web-серверы и серверы электронной почты, все еще были уязвимы. Межсетевой экран также стал теперь потенциальной целью нападений, поэтому требовался какой-то способ обеспечения его защиты. Том установил на этом сервере программу Bastille Linux, чтобы проверить, что он сконфигурирован безопасным образом ([лекция 2](#)). Затем он выполнил программу Nmap, как извне, так и изнутри сети ([лекция 4](#)). Она сообщила, какие прикладные порты были видимы извне по всем общедоступным IP-адресам. Внутреннее сканирование позволило узнать, имеются ли какие-то необычные или ненужные службы, выполняющиеся на внутренних машинах.

Затем он воспользовался программой Nessus для повторного сканирования сети извне и изнутри ([лекция 5](#)). Это программа "копает" значительно глубже, чем Nmap, она проверяет открытые порты для большого числа уязвимостей и позволяет выявлять неправильно сконфигурированные машины внутри сети. Программа Nessus создала отчеты, которые показали, где в системе безопасности на общедоступных серверах имеются слабые места, и предоставила подробные инструкции по их устранению. Он использовал эти отчеты для разрешения проблем, а затем еще раз выполнил программу Nessus, чтобы убедиться, что уязвимости ликвидированы.

Создание системы раннего предупреждения

Том ликвидировал все известные прорехи, но он также хотел знать, нет ли какой-то нетипичной активности в сети или на общедоступных серверах. Он воспользовался сетевым анализатором Ethereal для получения контрольных данных о различных типах активности в сети ([лекция 6](#)). Он также настроил

на сервере сетевую систему обнаружения вторжений, используя программный пакет Snort ([лекция 7](#)). Эта программа круглосуточно следила за сетью, выявляя подозрительную активность, которую Том определил специальным образом. Программа извещала его о новых атаках и нетипичном поведении пользователей внутри сети.

Создание системы управления данными безопасности

Вначале Том был перегружен всевозможными данными этих систем. Однако он настроил базу данных и использовал несколько программ для управления выводом программ системы безопасности. Одна из них, ACID (Analysis Console for Intrusion Database - Консоль анализа базы данных вторжений) помогла рассортировать и интерпретировать данные сетевой системы обнаружения вторжений ([лекция 8](#)). Программа "Командный центр Nessus" помещала результаты сканирования в базу данных и создавала на их основе отчеты ([лекция 8](#)). Том использовал также программу Swatch, которая отслеживала по файлам журналов различные аномалии ([лекция 8](#)). Эти программы позволили ему за полчаса просматривать отчеты на web-странице, в которой были объединены все его задания по мониторингу безопасности. Для Тома, который воплощал в одном лице техническую поддержку, программиста и, естественно, администратора системы безопасности, это было значительной экономией времени.

Реализация защищенного беспроводного решения

Еще одним заданием для Тома было построение для компании беспроводной сети. Том знал, что технология беспроводных сетей изобилует проблемами безопасности, поэтому он использовал две программы - NetStumbler и WEPCrack - для контроля защищенности и развернул беспроводную сеть с требуемыми параметрами безопасности ([лекция 10](#)).

Защита важных файлов и коммуникаций

Одной из проблем, беспокоивших руководство компании, было использование электронной почты для пересылки потенциально уязвимых документов. Том знал, что пересылка информации через обычную электронную почту аналогична отправке ее почтовой открыткой. Любой из посредников, обрабатывающих сообщение, мог его прочитать. Том заменил эту практику системой, использующей программное средство PGP, которое позволяет посылать файлы с конфиденциальной или уязвимой информацией в зашифрованном виде и защищать важные внутренние файлы от любопытных глаз неавторизованных пользователей ([лекция 9](#)).

Расследование вторжений

Наконец, когда сеть была защищена настолько, насколько это возможно, Том проверил каждый сервер на наличие каких-либо следов прошлых вторжений, чтобы убедиться, что не было оставлено ничего вредоносного, и, если было, попытаться выяснить, кто это сделал. Используя утилиты системного уровня, такие как wtmp и lsof, и программу The Coroner's Toolkit, Том смог идентифицировать возможных нарушителей, ответственных за прошлые вторжения ([лекция 11](#)). Хотя собранные доказательства были недостаточно надежными, чтобы возбудить уголовное преследование, он заблокировал IP-адреса злоумышленников в новом межсетевом экране, чтобы те не смогли помешать работе. Он использовал также эту информацию, чтобы пожаловаться на злоупотребления поставщику Интернет-услуг.

За несколько первых месяцев работы Том произвел впечатляющие преобразования. Но что самое удивительное, он смог сделать все это при почти полном отсутствии бюджета. Как ему это удалось? Его подготовка в области информационной безопасности помогла разработать план действий и реализовать его. Он смог воспользоваться этими знаниями для установки недорогих, но эффективных защитных решений, используя ПО с открытыми исходными текстами для создания всех своих систем. С помощью этих пакетов Том смог превратить плохо защищенную сеть в сеть, безопасность которой могла бы соперничать со значительно более дорогими аналогами. И он сделал это без дополнительного персонала и с минимальным количеством средств.

Вы также можете использовать открытое ПО для защиты своей организации. Эта книга познакомит вас с десятками программных пакетов, которые помогут это сделать, а также обучит правильным политикам и процедурам, обеспечивающим информационную безопасность. Как неоднократно подчеркивается в этой книге, программно-технические средства - прекрасное подспорье, но это лишь половина дела. Хорошо организованная программа информационной безопасности состоит также из политик и процедур, позволяющих в максимальной степени использовать возможности программного обеспечения. Поэтому, прежде чем переходить к установке ПО, давайте обсудим основы ИБ и происхождение ПО с открытыми исходными текстами.

Практика информационной безопасности

Наука информационной безопасности включает множество различных аспектов, однако имеются три области, которые являются основанием ИБ: конфиденциальность, целостность и доступность. Для их обозначения часто используется акроним КЦД. Эта триада представляет цели информационной безопасности ([рис. 1.1](#)). Каждая из них требует различных инструментов и методов и защищает определенный элемент или тип информации.



Рис. 1.1. Принципы информационной безопасности

Конфиденциальность

Элемент конфиденциальности информационной безопасности защищает данные от просмотра неавторизованными лицами. Это может быть информация, которая является внутренней для вашей организации, такая как инженерные планы, исходные тексты программ, секретные рецепты, финансовая информация или маркетинговые планы. Это может быть информация о заказчиках или сверхсекретные правительственные данные. Конфиденциальность относится также к необходимости сокрытия информации от любопытных глаз внутри организации. Разумеется, нежелательно, чтобы все служащие могли читать электронную почту высшего руководства или просматривать платежные ведомости.

Существует много способов защиты частных данных от просмотра. Один из них состоит в запрещении доступа к данным. Но иногда это невозможно, как в случае данных, передаваемых через Интернет. В подобных случаях необходимо использовать другие средства, такие как шифрование, чтобы скрыть и утаить данные во время их передачи.

Целостность

Элемент целостности помогает гарантировать, что неавторизованные лица не могут модифицировать данные. Он означает также, что авторизованные лица не вносят изменений без соответствующего разрешения. Следует различать два момента. Если кассир банка втайне дебетует чей-то счет и кредитует другой, то это проблема целостности. Он авторизован делать изменения счетов, но только после получения указания на внесение изменений. Целостность данных означает также, что данные соответствующим образом синхронизированы во всех системах.

Доступность

От защищенных данных нет никакой пользы, если к ним нельзя обратиться. По мере того, как атаки типа "отказ в обслуживании" становятся все более распространенными, одной из основных задач системы информационной безопасности становится не только защита данных от доступа злоумышленников, но и обеспечение доступности данных для тех, кому они предназначены. Многие компьютерные преступники будут в равной степени удовлетворены как

разрушением данных, так и прерыванием работы web-сайта. Элемент доступности включает также подготовку к аварийной ситуации и возможность безопасного восстановления после ее ликвидации.

Том понимал, что должен применить все эти принципы, чтобы полностью защитить сеть компании. Он нашел программные средства, которые поддерживали все элементы. Важно было использовать все доступные возможности. Из новостей и профессиональных статей он знал ужасающую статистику.

Состояние компьютерной преступности

Компьютерные преступления стали эпидемией, которая затрагивает всех пользователей - от руководителя компании из списка Fortune 500 до домохозяйки. Согласно ежегодному исследованию ФБР по компьютерным преступлениям, проведенному совместно с Институтом Компьютерной Безопасности (CSI), более 90% компаний в США стали жертвами какой-либо формы компьютерного преступления. Восемьдесят процентов опрошенных понесли в связи с этими атаками определенные финансовые потери. Если в 2000 г. потери составили 337 миллионов долларов, то в 2001 г. они составили уже 445. И можно с уверенностью утверждать, что большинство атак не фиксируется. Многие компании не хотят признавать, что их компьютерные системы были взломаны или скомпрометированы, и избегают обращаться к официальным лицам, так как боятся, что дурная слава может повлиять на цены акций или бизнес, особенно в таких отраслях как банковское дело, которые опираются на доверие общественности.

По прогнозам Центра Защиты Национальной Инфраструктуры (NIPC), компьютерные атаки в 2002 г. участятся и станут более сложными, часто с использованием нескольких методов нападения, как в случае с "червем" Code Red в 2001 г. Предполагается, что хакеры сконцентрируются на маршрутизаторах, межсетевых экранах и других некомпьютерных устройствах, так как они менее заметны и обеспечивают при незаконном использовании более полный доступ к корпоративной сети. Предсказывается также, что время между обнаружением новой уязвимости и появлением средств, которые ее используют, будет сокращаться, предоставляя меньше времени для ответа на потенциальную угрозу. Действительно, среднее время между сообщением об уязвимости и публикацией программы ее использования сократилось с месяцев до недель. Например, "червь" Blaster появился всего лишь через шесть недель после обнаружения в начале 2003 г. уязвимости в механизме удаленного вызова процедур от Microsoft.

Группа реагирования на нарушения информационной безопасности (CERT), которая действует совместно с Университетом Карнеги Меллон и федеральным правительством, отслеживает возникающие угрозы и старается предупредить компании о вновь обнаруженных уязвимостях и пробелах безопасности. Выяснилось, что количество отчетов об инцидентах, связанных с компьютерной безопасностью более чем удвоилось в 2001 г. по сравнению с предыдущим годом, с 21756 до 52658. Отмечен рост числа атак более чем на 100% каждый год, начиная с 1998 г. В 2003 г. число инцидентов выросло на 70%, хотя общее число новых уязвимостей, определяемых как слабые места в аппаратном или программном обеспечении, которые позволяют получить несанкционированный доступ, упало ([рис. 1.2](#)). Это связано с появлением "червей", которые быстро распространяются в Интернете, заражая множество систем.

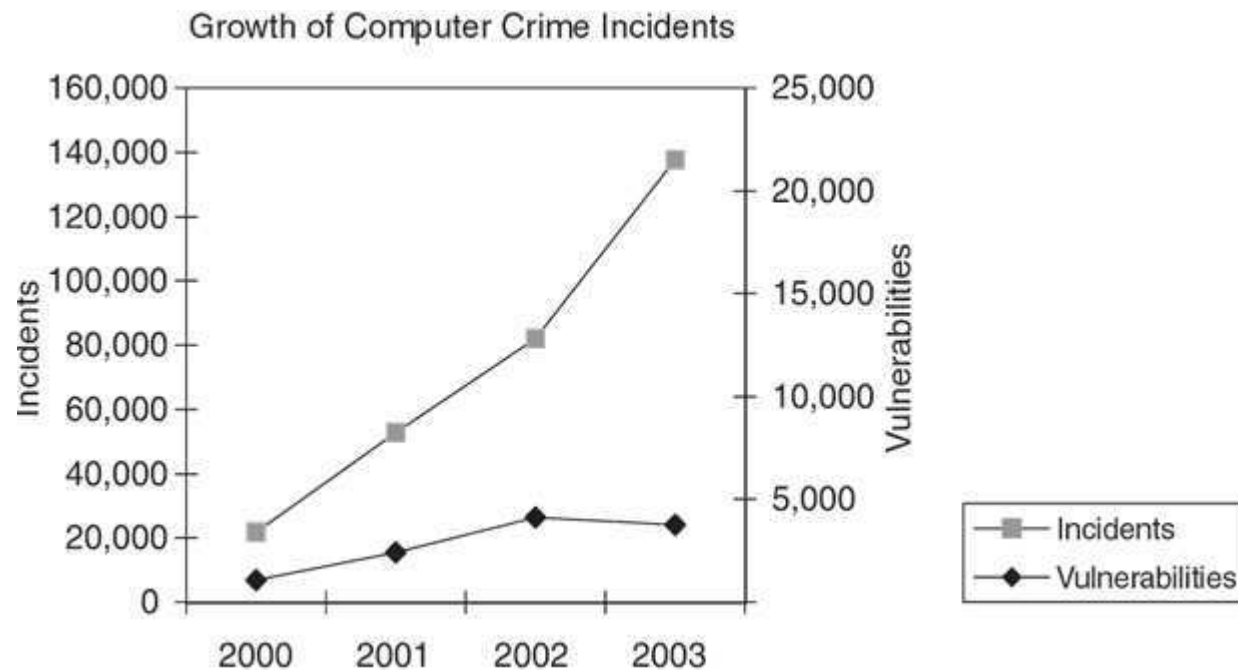


Рис. 1.2. График уязвимостей и инцидентов (источник - CERT).

Экспоненциальный рост как числа атак, так и методов их осуществления, является тревожной тенденцией, поскольку все больше организаций подключаются к Интернету. К сожалению, многие из них предпочитают оставаться в неведении и игнорируют информацию о проблемах безопасности. Обычное объяснение недостаточной защищенности компьютерной сети: "Зачем хакеру наша организация? У нас нет ничего такого, что им нужно". В прошлые годы такая позиция могла быть оправданной. Хакеры старой школы, как правило, охотились на крупные организации, которые имели ценные данные.

Однако радикальные изменения в информационных технологиях сделало потенциальной целью хакеров все организации, даже представителей малого бизнеса. Фактически организации малого и среднего размера теперь являются целями более 50% атак, о которых сообщает ФБР. Это изменение было вызвано несколькими факторами, которые описаны в следующих разделах.

Появление Интернет

Когда в Интернете были соединены всего несколько сетей, организациям в основном приходилось беспокоиться о риске того, что кто-то получит доступ к консоли компьютера, или о вирусе, который будет занесен через дискету. Защитой от физических угроз такого вида компании занимались годами. Дверные замки, системы сигнализации и даже вооруженная охрана могли защитить компьютеры и системы от физического доступа. Единственными программно-техническими средствами защиты до наступления эры Всемирной Паутины были антивирусное ПО и пароли.

С распространением сети Интернет хакеры получили возможность осуществлять нападения, находясь на удалении в тысячи миль, и похищать критически важные активы компании, обходя все физические барьеры. Им на руку анонимность, которую предоставляет Интернет. Они могут находиться в иностранных государствах, не имеющих соглашения об экстрадиции преступников. Они оставляют мало информации о том, кто они такие и что они сделали. Когда вы подключаетесь к Интернету, то оказываетесь буквально в нескольких нажатиях клавиш от любого хакера, взломщика и просто бездельника. Защиты паролем и антивирусным ПО уже недостаточно, чтобы удержать незваного гостя от проникновения в ваш виртуальный офис.

Повсеместно распространенная, недорогая широкополосная сеть

Относительно недавно подключение к Интернету по выделенным линиям было привилегией крупных организаций. Теперь можно получить доступ через цифровую абонентскую линию или кабельный модем для компании или домашнего использования менее чем за \$100 в месяц. Организации становятся

доступными в Интернете тысячами, и это в целом хорошо для бизнеса. Однако наличие выделенного соединения подвергает их большому риску, чем используемые ранее коммутируемые соединения через телефонную линию. Прежде всего, широкополосная сеть совершенно отлична от простого соединения через модем с точки зрения сети. Обычно при использовании коммутируемого доступа вы подключены, только пока вы работаете. При постоянно подключенной широкополосной сети хакеры могут продолжать атаки, повторяя попытки входа столько раз, сколько потребуется. Они особенно любят действовать в поздние ночные часы, когда системные администраторы, которые могут заметить что-то подозрительное, ушли домой.

Наличие доступа к производственной площадке с выделенным широкополосным доступом очень привлекательно для хакеров. Они могут использовать полосу пропускания для атак на другие организации. Если цель хакера состоит в блокировании таких популярных сайтов, как Yahoo или Amazon, с помощью простой грубой силы, ему потребуется широкая полоса пропускания. Большинство этих сайтов обладают полосой пропускания, которая измеряется гигабитами, а не мегабитами. Чтобы подавить эти сайты запросами, требуется канал с огромной пропускной способностью, которую средний хакер не может себе позволить. Однако если он проникает в другие машины в Интернете с широкополосными соединениями, он может использовать эти машины для атаки своей реальной цели. Если он смог овладеть достаточным количеством плацдармов, он как бы получает в свое распоряжение огромную пушку. Это называется распределенной атакой на доступность. Она обладает дополнительным преимуществом, сбивая со следа правоохранные органы, так как атаки идут с компьютеров ничего не подозревающих жертв, а не от самих атакующих. Эти машины-жертвы называются "зомби", и хакеры располагают специальным программным обеспечением, которое они могут загружать, чтобы заставить зомбированные компьютеры "проснуться" по специальной команде, которую могут послать только они. Эти программы часто очень трудно обнаружить и уничтожить, так как хост не проявляет никаких нездоровых признаков, пока ПО зомби неактивно. Единственное, что требуется хакерам, - ваша полоса пропускания; обычно им мало интересно, кто вы такой.

Другой причиной, по которой хакеры проникают в компьютеры, является хранение инструментов и полученной незаконным путем добычи. Используемые для этого машины хакеры называют "шкафчиками", туда они помещают противозаконные данные - порнографию, пиратское ПО, фильмы или хакерские инструменты. Вместо того чтобы хранить эти данные на своей машине, где они могут быть найдены и использованы против них в суде, хакеры предпочитают укрывать их на серверах своих жертв. Широкополосное соединение очень удобно, так как обладает большой пропускной способностью для пересылки файлов. Предпочтительнее использовать для этого небольшую организацию, которая вряд ли содержит обширный штат в отделе информационных технологий, чтобы следить за своим соединением с Интернетом, и, вероятно, использует не очень развитые средства защиты. Хакеры могут передать IP-адрес взломанного сервера своим приятелям и использовать его для неформальных встреч и обмена. Такие виды вторжения трудно обнаружить, так как компьютер при этом работает нормально, хотя можно заметить снижение производительности или скорости загрузки файлов.

Атаки "командных детишек"

Другим моментом, который изменил картину компьютерных правонарушений, является рост числа нарушителей, особенно с низким уровнем подготовки. Этих хакеров-новичков называют "командными детишками", так как они часто используют простейшие хакерские инструменты или командные файлы, которые находят в Интернете, а не свои собственные знания. Когда-то хакеры были частью элитарного сообщества высококвалифицированных (хотя и с моральными проблемами) индивидов, мастеров программирования, они понимали компьютеры на самом фундаментальном уровне. Они даже соблюдали неформальный кодекс хакера, который, хотя и отвергал идею приватности, гласил, что взламываемым компьютерам не следует наносить никакого вреда. Жизнь хакера была посвящена изучению и исследованию. Однако вскоре это сообщество разделилось, и в него влились новички. Теперь несложно найти сотни web-сайтов, которые могут научить, как проникнуть в чужую систему за пару минут. Многие так называемые хакеры являются подростками со слабым знанием программирования. Их целью является не поиск знания, а использование взломанных компьютеров, хвастовство и откровенный вандализм. Криминальные элементы, пополняющие сообщество хакеров, ищут наиболее легкие "способы взлома". Эти неопытные преступники атакуют системы небольших организаций, которые располагают меньшей сильной защитой и менее опытными администраторами, неспособными заметить ошибки этих неофитов. Большинство из них не осмелятся напасть на компьютеры Пентагона или ЦРУ, поскольку эти организации имеют развитую защиту и серьезные возможности судебного преследования. Немногие небольшие организации могут позволить себе расследование, еще меньшее количество - возбудить судебное преследование компьютерного злоумышленника, даже если они установят его личность. И, поскольку по большей части основной целью "командных детишек" является не изучение, а причинение вреда, они часто вызывают больше повреждений, чем опытный компьютерный преступник.

"Черви", автосуперы и другие вредоносные программные средства

Наконец, основная причина принципиального изменения картины компьютерной безопасности состоит в том, что большая часть взломов в настоящее время автоматизирована, а выбор жертвы осуществляется случайным образом. "Командные детишки" могут использовать инструменты, которые сканируют IP-адреса случайным образом в поисках имеющих уязвимости компьютеров. Эти программы часто работают целыми ночами, собирая урожай

потенциальных жертв. Существуют пакеты, называемые "автосуперами", которые получают на компьютере привилегии суперпользователя или администратора. Эти инструменты не только проводят разведку, но выполняют реальное проникновение в компьютер и размещают там троянские или другие вредоносные программы. В результате с помощью одного щелчка мышью кто-то с компьютерным образованием не выше чем у шестилетнего ребенка может взломать десятки машин за один вечер.

С появлением в Интернете таких "червей", как Nimda в 2001 г., даже человеческий элемент был исключен из этой картины. Эти автономные родственники компьютерных вирусов странствуют в Интернете в поисках компьютеров с определенным набором уязвимостей. Когда таковой находится, они помещают себя в этот компьютер, выполняют запрограммированные действия, а затем настраивают систему на поиск новых жертв. Эти автоматически действующие системы по взлому компьютеров заразили значительно больше сетей, чем сделали люди-нарушители порядка. Они также распространяются невероятно быстро. Согласно оценкам, "червь" Code Red распространился на более чем 300000 серверов в течение нескольких дней с момента своего появления.

Риски организаций, связанные с информационной безопасностью

Очевидно, что ситуация изменилась. Раньше небольшие организации, в основной массе, могли не слишком беспокоиться о безопасности своих данных; теперь организации любых размеров вынуждены тратить на эти проблемы время и деньги.

Каковы возможные риски? Немногие организации всерьез задумываются обо всех рисках, которым они подвергаются с точки зрения информационной безопасности. Следует осознавать все возможные риски, понимать, какие из них применимы к вашей организации и каков возможный ущерб от каждого из них. Это поможет выбрать разумные пути повышения компьютерной безопасности и обосновать необходимые затраты.

Потеря данных

Хотя из-за компьютерных вирусов эта угроза актуальна с 1980-х годов, немногие руководители задумываются об ущербе от потери части или всех данных. Без правильно организованного резервного копирования (отсутствующего во многих небольших организациях), потеря критически важных данных может стать катастрофической. Могут быть стерты накопленные за годы данные по бухгалтерии, платежным ведомостям или заказчикам. Могут быть потеряны заказы. Если данные принадлежат заказчикам, то организация может понести ответственность за их потерю. Представители определенных профессий, такие как адвокаты или бухгалтеры, могут быть подвергнуты штрафам или наказаны за потерю подобных данных. Добавьте к этому потери бизнеса и производительности, пока служащие восстанавливают данные или вынуждены оперировать с записями на бумаге. Даже если имеются резервные копии, время и усилия, необходимые для восстановления и запуска систем в работу, будут значительны. Немногие организации смогут долго просуществовать без своих компьютеризированных записей и систем. Имеет ли ваша организация документированный "План действий в аварийных ситуациях", который охватывает данные и системы? Если нет, то вы можете получить неприятный сюрприз в виде неожиданного перерыва в работе.

Отказ в обслуживании

Современные хакеры в большинстве своем вовсе не компьютерные гении, а всего лишь высокотехнологичные вандалы. Эти люди просто получают удовольствие от разрушения серверов или создания отказов в обслуживании.

Много ли организаций имеют оценки часового или дневного ущерба от потери доступа в Интернет? В некоторых отраслях или организациях, существенно опирающихся на информационные технологии, этот ущерб может быть очень высок. Немногие организации в наше время не зависят так или иначе от доступа в Интернет. Атака на доступность может быть либо незначительным раздражителем, либо существенным ударом для организации - в зависимости от того, насколько бизнес полагается на Интернет. Попробуйте оценить ущерб для вашей организации на основе числа сотрудников, которые не смогут работать, числа заказов, оперативно обрабатываемых в сети, и т.д.

Трудности/потеря заказчиков

Отсутствие доступа в Интернет может повредить имиджу организации. Невозможность общения по электронной почте или потеря важных сообщений в лучшем случае создают временные затруднения. Если web-сайт будет отключен, то заказчики немедленно начнут задавать вопросы. Для акционерной компании это может означать снижение котировок акций. Доказательством служит падение цен акций Yahoo и Amazon после сообщений об отказах в обслуживании. Миллионы или даже сотни миллионов долларов акционеров могут исчезнуть в одно мгновение. Для электронной коммерции, которая зависит от мнения клиентов о безопасности при размещении своей финансовой информации в Интернете, одна проблема с web-сайтом может свести на нет годы успешной работы. Компания CD Universe, сетевой розничный продавец компакт-дисков, у которого была украдена база данных о кредитных картах,

не смогла оправиться после этой атаки. Компания Cloud Nine Communications, поставщик Интернет-услуг в Англии, не работала неделю в связи с согласованной и длительной атакой на доступность, и в конечном счете вынуждена была закрыться. Существуют банды хакеров, которые выполняют массовые искажения web-сайтов, иногда поражая сотни сайтов за ночь. Прием в эти клубы хакеров подтверждается определенным числом обезображенных web-сайтов. Вы хотите, чтобы ваш web-сайт фигурировал в их отчетах?

Законодательная ответственность

В наш сутяжнический век небольшая ошибка может привести к судебному разбирательству, стоящему миллионы. Представьте последствия, если база данных заказчиков будет украдена, а затем помещена в Интернет. Результатом таких событий являются судебные дела с групповыми исками. Вследствие огромного роста краж персональных данных были созданы законы, требующие от организаций следования определенным стандартам хранения при работе с персональными или финансовыми данными клиентов. Одной из отраслей, которая была особенно сильно затронута законодательством, является здравоохранение. Закон США по обеспечению доступности и подотчетности в медицинском страховании 1996 г. требует, чтобы любая организация, имеющая дело с информацией о пациентах, должным образом защищала эти данные от неавторизованного использования. Требования этого закона по обеспечению приватности применительно к компьютерным сетям вступили в действие в 2003 г. Для нарушителей существуют административные и уголовные наказания, так что это больше не является чисто финансовым вопросом. Руководители организаций могут оказаться в тюрьме, если будут уличены в нарушениях.

Хакеры всегда ищут незащищенные компьютеры, чтобы проводить с них распределенные атаки на доступность. Если компьютеры вашей организации использовались в такой атаке, и жертвы не смогут найти настоящего злоумышленника, они могут обвинить вас в том, что вы пренебрегли защитой своей сети.

Еще один предмет заботы - ответственность за нарушение авторских прав. Копирование пиратских фильмов, музыки и программного обеспечения через Интернет достигло крайней степени. Медийные компании сыты этим по горло и начинают отслеживать нарушителей прямо по IP-адресам загружающих, направляя по этим адресам юристов. InternetMovies.com, web-сайт на Гавайях, лишился услуг своего Интернет-провайдера, когда тот получил повестку в суд на основании утверждений о пиратских файлах, находящихся в его сети. Пираты, которые хотят распространять свои изделия, прибегают к хранению их на компьютерах третьих сторон, компрометируя серверы корпоративных сетей. Если сервер вашей организации без ее ведома используется для подобных целей, или на нем будут обнаружены такие файлы, то вас могут отключить от Интернета, подвергнуть штрафу или вызвать в суд. Такого рода истории помогают убедить сопротивляющихся руководителей в необходимости установить более строгие правила для персонала, такие как выполнение повышенных требований к паролям или запрет на установку ПО разделения файлов.

Раскрытие корпоративных секретов и данных

Трудно дать денежную оценку этого риска, так как он варьируется от организации к организации. Например, ценность рецепта Coca Cola или жареных цыплят Colonel Sander может достигать миллиардов долларов. В меньшей организации подробная документация на запатентованные устройства или формулы может быть бесценна. В некоторых случаях большая часть ценностей организации может заключаться в подобных важных данных. Например, биотехнологическая компания может хранить в корпоративной сети результаты исследований, относящихся к новейшим патентам в области генетики.

Списки заказчиков всегда представляют ценность для конкурентов, особенно в сегменте рынка с высокой конкуренцией. Акционеры подали в суд на компанию Hewlett-Packard после того, как было опубликовано содержание закрытых переговоров между руководителями компании во время спорного слияния.

Однако этот риск существует даже в организациях, где нет секретных планов или рецептов. Например, представьте ущерб от возможного доступа к корпоративным файлам платежных ведомостей со стороны рядовых работников. Такие вещи происходят постоянно, обычно в связи с чрезмерным любопытством или мстительностью сотрудников. Разлад и ухудшение морального климата могут быть огромными, вплоть до ухода сотрудников, недовольных различиями в оплате. Часто всего этого можно избежать, если системный администратор правильно защитит систему.

Искажение записей

Иногда злоумышленник хочет не украсть или разрушать данные, а только изменить существующие записи, надеясь, что это не будет обнаружено. Компьютерные преступления такого типа обычно трудно обнаружить, так как системы продолжают функционировать как и прежде. Не происходит разрушения системы или падения производительности, служащих признаками вторжения. Отсутствуют вызывающие тревогу изменения web-сайта.

Очевидно, что для банков и правительственных агентств это может быть очень серьезной проблемой. Но и любой другой организации приходится заботиться о том, чтобы никто не мог проникнуть в систему заработной платы и изменить значения выплат. Школы и университеты могут иметь дело с учащимися, которые пытаются изменить оценки. Часто только бухгалтерские аудиторы находят свидетельства преступлений. Однако при правильно организованной системе безопасности перечисленных проблем можно избежать.

Потеря производительности

Это значительно менее заметный риск, которого часто очень трудно избежать. Он может состоять в использовании сотрудниками полосы пропускания для загрузки музыки или фильмов, что замедляет работу других служащих, или в посещении ненадлежащих web-сайтов. Хотя это относится к вопросам политики в отношении сотрудников, часто прибегают к услугам системных администраторов для решения проблемы техническими средствами, такими как фильтрация информационного наполнения и межсетевое экранирование. Многие из неавторизованных программ, например, Napster, Kazaa, программ мгновенного обмена сообщениями, помимо того, что снижают производительность, могут создавать уязвимости в сети организации.

При всех этих рисках можно предположить, что организации будут стараться установить необходимые средства защиты. Действительно, крупнейшие организации так и поступают, но большинство организаций малого и среднего размера практически ничего не делают в плане сетевой безопасности. В лучшем случае устанавливается межсетевая экран и антивирусное ПО; считается, что этого достаточно. Но, к сожалению, это не так.

Возникла целая отрасль, предлагающая решения этих проблем. Существуют коммерческие аппаратные и программные решения, такие как межсетевые экраны, системы обнаружения вторжений и сканеры уязвимостей. Однако большинство этих продуктов очень дороги, поэтому только крупные организации могут их себе позволить. Простой межсетевой экран стоит несколько тысяч долларов. Коммерческие системы обнаружения вторжений и решения для анализа защищенности могут стоить десятки тысяч долларов или больше. Кроме прямых расходов, существуют повседневные расходы на сопровождение ПО. Многие из этих программных решений требуют для своей работы очень мощных компьютеров. Для создания отчетов зачастую необходимы дорогие СУБД, такие как Oracle. С учетом этих расходов желательный уровень информационной безопасности оказывается недоступным для организаций малого и среднего размера. Но, как мы видели, риск для этих организаций так же высок, как и для компаний из Fortune 500 и, возможно, даже выше, так как они обладают значительно меньшими финансовыми ресурсами для противодействия атакам.

Что же делать вечно спешащему, перегруженному, имеющему недостаточно средств системному администратору? Существует решение, которое может обеспечить организации качественную защиту с очень небольшими расходами: программное обеспечение с открытыми исходными текстами.

История ПО с открытыми исходными текстами

Движение за ПО с открытыми исходными текстами ведет отсчет от рождения платформы UNIX, в связи с чем многие ассоциируют открытые исходные тексты с системами UNIX и Linux, хотя эта концепция распространилась почти на все операционные системы. ОС UNIX была изобретена в Bell Labs, в то время - исследовательском подразделении компании AT&T. AT&T впоследствии лицензировала это программное обеспечение университетам. Так как AT&T была регулируемой компанией, она не могла продавать UNIX, поэтому она передала университетам исходные тексты операционной системы, чего обычно не делают с коммерческими программными продуктами. AT&T не считала в то время, что ОС UNIX представляет большую коммерческую ценность.

Университеты, являясь питательной средой творческой мысли, немедленно начали создавать собственные дополнения и модификации оригинальных текстов от AT&T. Некоторые сделали только незначительные изменения. Другие, такие как университет в Беркли (Калифорния), внесли столько модификаций, что создали целую новую ветвь кода. Вскоре лагерь UNIX разделился на два: один - на основе текстов от AT&T продвигал ОС UNIX System V, используемую многими производителями мэйнфреймов и миникомпьютеров; другой - на основе кода BSD, который породил многие версии ОС UNIX с открытыми исходными текстами, которые мы имеем сегодня. ОС Linux первоначально основывалась на MINIX, ОС UNIX для ПК, которая имела корни в System V.

Ранние приверженцы открытого ПО также не избежали философского раскола. Программист Ричард Столмэн основал Фонд свободного программного обеспечения, который выступает за открытость исходных текстов любого ПО. Он разработал для этого специальную лицензию, называемую Генеральной публичной лицензией (GPL). Она обеспечивает авторам некоторую защиту их материала от коммерческого использования, но, тем не менее, предусматривает свободную передачу исходных текстов. Университет в Беркли ранее разработал свою собственную лицензию для открытых исходных текстов, лицензию BSD, которая является менее ограничительной, чем GPL, и используется во многих вариантах BSD UNIX.

Две эти лицензии позволяют программистам безбоязненно разрабатывать код для новых платформ UNIX, не беспокоясь о юридических последствиях, или о том, что их работа будет использоваться другими с коммерческой выгодой. Это стимулировало разработку многих приложений, которые в настоящее время используются в Интернете, а также базового инструментария, в первую очередь - семейства компиляторов Gcc, интерпретаторов Python, Awk, Sed, Exprect, и т.д.

Мощным толчком к разработке ПО с открытыми исходными текстами стало распространение Интернета в начале 1990-х годов. До этого разработчики использовали для общения и пересылки файлов коммутируемые сети и доски объявлений. Существовали сети, такие как USENET и DALnet, для облегчения работы множества специализированных форумов. Однако использовать эти сети было трудно и дорого, и часто они не выходили за национальные границы в связи с высокой стоимостью соединений.

Распространение Интернета все изменило. Недорогие глобальные коммуникации и облегчение доступа к данным через web-страницы вызвало взрыв инноваций и разработок в мире открытого ПО. Теперь программисты могли мгновенно взаимодействовать и создавать Web-сайты с описанием своей работы, которую любой в мире мог легко найти с помощью поисковых машин. Проекты, развивавшиеся параллельно, объединяли свои ресурсы. Одновременно от больших групп отделялись меньшие, уверенные, что теперь они смогут найти поддержку для своих проектов.

На сцену выходит Linux

Именно на этом плодородном поле выросло высшее современное достижение среди ПО с открытыми исходными текстами. Линус Торвалдс был старательным студентом колледжа в Финляндии и мастерски владел своим ПК. Он хотел использовать на нем привычную по учебе версию UNIX. Он купил MINIX - упрощенную версию ОС UNIX для ПК, но был разочарован ее ограниченностью, в частности, в области эмуляции терминала. Для выполнения своей работы ему нужно было соединиться с колледжем, так что наиболее быстро развивающаяся современная операционная система начиналась как проект для создания программы эмуляции терминала для его ПК.

Со временем он закончил свою программу и разместил ее в нескольких телеконференциях USENET; другие люди начали предлагать добавления и усовершенствования. В это время сформировалось ядро, превратившееся ныне в многонациональное объединение тысяч людей. В течение шести месяцев Торвалдс создал основу операционной системы. Она могла делать не очень много, но с участием десятков программистов, содействовавших ее развитию, потребовалось не так много времени, чтобы этот "научный проект" превратился в то, что мы знаем как операционную систему с открытыми исходными текстами, именуемую Linux.

Linux стал символом всего, что есть положительного в открытом ПО. Все начинается с того, что кто-то хочет улучшить уже существующие средства или создать новые. Если это представляет интерес, импульс улавливается, и вскоре появляется нечто, что в коммерческой среде потребовало бы для своего создания годы работы и миллионы долларов. Пока это не стоило ничего (если не считать тысяч затраченных часов). В связи с этим продукт может предлагаться бесплатно. В результате он может получить более широкое распространение и привлечь еще больше разработчиков. И цикл продолжается. Это истинная меритократия, где выживают только хорошие программы.

Однако это не означает, что ПО с открытыми исходными текстами не имеет коммерческого мотива или возможностей. Сам Торвалдс получил немало средств благодаря своим усилиям, хотя он первый скажет, что это никогда не было его целью. Вокруг Linux возникло много компаний - либо для его поддержки, либо для создания аппаратных и программных решений на его основе. RedHat и Turbo Linux являются примерами многочисленных компаний, которые получают значительные доходы и имеют высокую рыночную стоимость (хотя последняя и сократилась по сравнению с периодом их наивысшего расцвета в конце 1990-х). Даже компании, которые известны как производители патентованного программного обеспечения, такие как IBM, приняли Linux как дополнительный способ продажи своего оборудования и услуг.

Это не означает, что все программное обеспечение должно быть бесплатным или иметь открытые исходные тексты, хотя радикально настроенные представители мира открытого ПО будут с этим не согласны. Существует сейчас и будет существовать всегда место для патентованного ПО с закрытыми исходными текстами. Но движение за открытое ПО продолжает набирать обороты, его поддержка ширится. Возможно, со временем подобное ПО сможет составить большую часть установленной базы программного обеспечения. Оно представляет альтернативу коммерческим продуктам и заставляет их поставщиков непрерывно создавать и предлагать на рынке реальные ценности. В конце концов, если имеется бесплатная программа с открытыми исходными текстами, которая делает то же, что ее коммерческий аналог, производители последнего должны обеспечить такую поддержку, которая оправдывала бы запрашиваемые деньги.

Достоинства ПО с открытыми исходными текстами

Вы и ваша организация можете использовать ПО с открытыми исходными текстами как для сокращения расходов, так и для повышения своей безопасности. Следующие разделы затрагивают некоторые из множества причин, по которым защитные инструменты с открытыми исходными текстами могут иметь смысл для вас и вашей организации.

Стоимость

По этому показателю трудно превзойти бесплатную вещь! Хотя открытое ПО не обязательно бесплатное, по большей части оно является таковым. Наиболее распространенной лицензией ПО с открытыми исходными текстами служит лицензия GNU GPL, которая является лицензией бесплатного программного обеспечения. Другое открытое ПО может быть условно бесплатным или даже оплачиваемым авансом, как коммерческие серверы, доступные от RedHat. Но в любом случае ПО с открытыми исходными текстами доступно за часть стоимости коммерческих аналогов. Это существенно помогает при обосновании новых проектов по безопасности в вашей организации. Значительно легче получить одобрение для нового решения, если требуется лишь немного времени и, возможно, новый компьютер для работы программного обеспечения. Фактически, в зависимости от уровня ваших полномочий, можно реализовать его, не делая экономических обоснований.

Расширяемость

По определению, ПО с открытыми исходными текстами модифицируемо и расширяемо, при условии, что вы обладаете достаточной программистской подготовкой. Многие программы с открытыми исходными текстами поддерживают встроенные интерпретируемые языки, позволяющие создавать небольшие дополнительные модули, не обладая обширными познаниями в программировании. Nessus, сканер уязвимостей с открытыми исходными текстами, поддерживает язык сценариев NASL (далее в книге вы найдете описание этого языка и научитесь писать специальные тесты системы безопасности). Snort, упомянутая выше система обнаружения вторжений с открытыми исходными текстами, позволяет создавать свои собственные определения сигналов тревоги. Это означает, что для контроля специфических требований организации можно легко написать соответствующую командную процедуру. Например, если имеется особо важный для организации файл customer.mdb, который должны использовать только определенные подразделения, то можно написать правило Snort, которое следит за перемещением этого файла в сети и предупреждает вас при подозрении на нарушение безопасности.

И, конечно, если вы являетесь квалифицированным программистом, то можете включиться в развитие исходных текстов и получить как ценный опыт, так и признание в сообществе разработчиков открытого ПО. Это может оказаться полезным и в карьерном плане.

Безопасность

Некоторые (по большей части это люди, вовлеченные в разработку коммерческого ПО) заявляют, что программное обеспечение с закрытыми исходными текстами изначально более безопасно, так как хакеры не могут легко получить доступ к его внутренней структуре. Эта философская школа полагается на обеспечение безопасности путем засекречивания конструкции продукта. Однако подобная логика не выдерживает сопоставления с фактами. Windows является крупнейшим в мире патентованным программным продуктом, однако число известных уязвимостей на платформах Windows примерно такое же, как и в Linux и других платформах с открытыми исходными текстами. Правда состоит в том, что открытость или закрытость исходных текстов не заставляют программистов писать более защищенные программы.

Независимость

В программах с открытыми исходными текстами обнаружение и исправление проблем с безопасностью происходит значительно быстрее. Коммерческие компании зачастую имеют серьезные материальные стимулы не признавать уязвимости в своих продуктах. Множество уязвимостей, найденных в продукте, особенно защитном, может отпугнуть новых заказчиков. Если это акционерная компания открытого типа, то цена акций может упасть. Кроме того, разработка корректирующих заплат и передача их заказчикам - дорогое удовольствие, которое обычно не приносит прибыли. Поэтому заставить компанию признать проблему с безопасностью ее программного продукта может быть непросто. Это означает, что могут пройти дни или недели, в течение которых системы клиентов будут по-прежнему уязвимы. Разочарованные этим процессом, некоторые исследователи безопасности приняли политику открытой публикации данных о новых уязвимостях.

Когда уязвимость общеизвестна, компания выполняет сложный процесс разработки и тестирования исправлений, чтобы избежать проблем, связанных с обязательствами перед заказчиками, и выпустить исправление для всех платформ одновременно. А значит, пройдет еще больше времени, в течение которого хакеры смогут воспользоваться известной уязвимостью.

Проекты ПО с открытыми исходными текстами не имеют таких ограничений. Корректирующие заплатки обычно появляются в течение часов или дней, а не недель. И, конечно, не обязательно ждать официальной коррекции; если вы хорошо понимаете код, то можете написать собственную заплатку или создать временный обход.

Общее мнение сообщества открытого ПО состоит в том, что безопасность наилучшим образом обеспечивается в результате критического анализа большим числом людей, которые явно не заинтересованы в том, чтобы не выявить какие-либо уязвимости. Такие же меры качества применяют в своей работе специалисты по криптографии. Концепция открытости исходных текстов не гарантирует, что вы получите более защищенное программное обеспечение, но благодаря ей не приходится верить компании на слово, что ее продукт безопасен, а потом дожидаться решения очередных проблем с безопасностью.

Поддержка пользователей

Коммерческие программные продукты обычно имеют систему поддержки и формальный канал для обращения за помощью. Одной из основных причин, почему многие сторонятся решений с открытыми исходными текстами, является их убеждение в том, что необходимо заплатить за продукт, чтобы получить надлежащую поддержку. Однако поддержка, которую они получают за деньги, часто оказывается не так уж и хороша. Если программистская компания небольшая, вам, возможно, придется ждать часы или дни, пока они ответят. Если поставщик крупный, то вы, вероятно, будете помещены в очередь обращений. Когда вас в конце концов соединят, то это окажется технический специалист начального уровня, который может лишь ввести вашу проблему в базу данных, чтобы посмотреть, не встречалась ли такая проблема ранее, а затем предложить стандартное решение. Обычно приходится добираться до технического специалиста второго или третьего уровня, который действительно понимает продукт и может помочь в случае сложных проблем. Очевидно также, что компании не любят признавать наличие ошибок в своих продуктах; они будут стремиться переложить ответственность на чужие плечи (операционную систему, оборудование и т.д.).

Далее, многие компании теперь берут отдельную плату за поддержку. Деньги, которые вы платите за несколько лет поддержки программного обеспечения, могут превышать его первоначальную стоимость. Эти платежи создают постоянный поток прибыли для компании, даже если вы никогда не делаете обновлений. Большинство программистских компаний, если пока еще и не делают этого, то движутся в этом направлении. Бесплатные телефонные номера технической поддержки ПО становятся редкостью.

Продукты с открытыми исходными текстами часто имеют прекрасные сети поддержки, хотя и несколько нетрадиционные. Поддержка подобного ПО менее формализована, но часто более полезна и более надежна. Редко существуют телефонные номера, по которым можно позвонить, но обычно существует несколько возможностей, чтобы получить ответы по программному обеспечению. В небольшом проекте это может быть непосредственное общение с разработчиком по электронной почте. С более крупными пакетами обычно ассоциирован список почтовой рассылки, куда можно отправлять свои вопросы, а иногда и несколько списков, зависящих от вопроса (пользователь, разработчик, определенные модули или платформы). Многие используют чаты, где можно задавать вопросы, спрашивать о новых свойствах или просто общаться в реальном времени.

Важный момент состоит в том, что вы общаетесь с людьми, которые хорошо знакомы с программным обеспечением (возможно даже с реальными разработчиками). Можно спросить у них о новых свойствах или прокомментировать недавние добавления. Вы в конце концов будете общаться с самыми яркими и наиболее опытными людьми в отрасли. Я многое узнал, просто следя за общением в списках почтовой рассылки.

На большинство вопросов, которые я задавал в этих списках, ответы были даны в течение нескольких часов или еще быстрее. Ответы обычно были содержательными и нередко остроумными. Вы получите несколько различных мнений или решений вашей проблемы, которые все могут быть правильными! Помимо получения очень подробных ответов на свои вопросы, вы можете побеседовать о состоянии дел в определенной области или поучаствовать в философских дебатах о будущих версиях (если у вас много свободного времени). И конечно, если вы квалифицированный программист, вы можете предложить собственные ответы на вопросы.

Помните, что эти люди обычно не работают на компанию, производящую программное обеспечение, и могут иногда показаться немного резкими или грубыми. Простые вопросы, на которые имеются полные ответы на страницах INSTALL или в FAQ, могут привести к выговору. Но он будет обычно также

содержать ответ или, как минимум, указание, где его можно найти. Иногда борьба мнений в списках скрывает реальную информацию. Однако я в любом случае предпочту взволнованные дебаты небрежному ответу.

Наконец, если вы действительно считаете, что должны заплатить за поддержку, то существуют компании, которые делают именно это для ПО с открытыми исходными текстами. Многочисленные Linux-компании предлагают поддерживаемые версии этой операционной системы. Для многих из наиболее популярных приложений также существуют компании, которые оказывают для них поддержку. Вы можете купить упакованную коробку с системой обнаружения вторжений Snort у нескольких компаний, которые будут поддерживать вас и предоставлять регулярные обновления. Таким образом, вы сможете получить такую же поддержку, которую предлагают коммерческие продукты, но сохранить при этом все преимущества открытого ПО.

Продолжительность жизни продукта

При использовании коммерческого ПО вы полностью во власти корпорации, которая владеет выбранным вами продуктом. Если это большая компания, такая как Microsoft, то вы, вероятно, в хорошем положении. Однако даже Microsoft может попытаться войти в рыночный сегмент, а затем бросить его и соответствующую линию продуктов. Более мелкие компании могут уходить из бизнеса, их могут покупать и поглощать. В наше время это происходит все чаще. Если компания, которая купила производителя, имеет конкурирующие продукты, то, скорее всего, они избавятся от одной из линий. Если они решат отбросить ваш продукт, то вам не повезло с будущей поддержкой. При закрытых исходных текстах продукта не существует возможности задать какие-либо вопросы или сделать в нем какие-либо обновления, если компания-производитель выходит из игры.

Проекты с открытыми исходными текстами никогда до конца не умирают. Это не значит, что они не переходят в спящий режим. Проекты постоянно уходят на обочину по мере того, как участники заканчивают университет или переходят к новому этапу своей карьеры. Это особенно распространено для небольших программ и инструментов. Более крупные программы (которые составляют большинство упомянутых в этой книге продуктов) всегда кем-то поддерживаются. Фактически иногда происходит борьба за власть в иерархии, чтобы контролировать проект. Однако если кому-то не нравится направление развития проекта, ничто не мешает создать свое ответвление и перевести проект в желаемое русло. Даже в маленьких проектах, где имеется только один разработчик, который больше не развивает его активно, можно просто продолжить с того места, где он был остановлен. А если вам надо исправить что-то или добавить свойство, то исходные тексты доступны и позволяют это сделать. С ПО с открытыми исходными текстами вы никогда не зависите от прихотей рынка или финансовых интересов компаний.

Обучение

Если вы хотите узнать, как работает программное средство безопасности или усовершенствовать свои навыки программирования, то ПО с открытыми исходными текстами прекрасно подходит для этого. Стоимость небольшая, поэтому не приходится выкладывать пару тысяч долларов за обучение. Если вы делаете это самостоятельно, то потребуется только компьютер для работы и соединение с Интернетом для загрузки программного обеспечения (или компакт-диск, приложенный к этой книге). Если вы делаете это для организации, то это самый дешевый учебный курс, который когда-либо санкционировался. Кроме того, организация получит дополнительную выгоду, так как вы сможете использовать новые знания для улучшения ее информационной безопасности, не тратя на это лишние деньги.

Конечно, вдумчивые программисты любят ПО с открытыми исходными текстами, так как они могут попасть прямо в сердце программы, и увидеть, как она работает. Это - лучший способ изучения, когда можно увидеть все исходные тексты, которые обычно хорошо документированы. Можно вносить изменения, добавлять новые свойства и расширять базовый код, что невозможно для программ с закрытыми исходными текстами. Самое большее, чего вы можете достичь с последними, - стать опытным пользователем; с открытыми исходными текстами вы сможете быть новатором и создателем, если, конечно, захотите.

Списки почтовой рассылки и чаты для проектов с открытыми исходными текстами являются прекрасным местом, где можно задавать вопросы и подружиться с людьми, которые способны стать реальными наставниками в вашей карьере. Участие в подобном проекте является, вероятно, самым быстрым способом узнать, как разрабатывается программное обеспечение.

Репутация

Отточив свое мастерство и несколько раз поучаствовав в оживленных дискуссиях, став постоянным действующим членом проекта ПО с открытыми исходными текстами, вы обнаружите, что стали авторитетом для новичков. Репутация в мире открытого ПО прекрасно выглядит в резюме. Участие в разработке продукта с открытыми исходными текстами служит свидетельством вашей преданности и организаторских способностей, не говоря уже о

навыках программирования. Создание пакета с открытыми исходными текстами - хорошая тема дипломной работы. И конечно, когда вы будете уверены в себе, вы можете начать создавать собственное открытое ПО, заняться реализацией других проектов. Многие авторы ПО с открытыми исходными текстами работают в реальных компаниях, делающих реальные деньги. Однако независимо от того, будут ли ваши усилия в области создания подобного ПО просто увлечением, как бывает чаще всего, или станет вашей единственной целью в жизни, это может быть и полезно, и интересно.

Когда ПО с открытыми исходными текстами может не соответствовать требованиям

Выше много говорилось о достоинствах ПО с открытыми исходными текстами. Вы могли подумать, что с его помощью можно решить все проблемы. Однако существуют ситуации, когда оно просто не подходит. Их не так много, но они встречаются.

Компания по созданию программных средств защиты данных

Если вы работаете в организации, которая создает патентованное программное обеспечение защиты данных, то ПО с открытыми исходными текстами не очень подходит в качестве основы разрабатываемых продуктов. Это не означает, что вы не можете экспериментировать с подобным ПО, чтобы получить идеи и ознакомиться с методами, но надо быть очень осторожным с включением каких-либо фрагментов из проекта с открытыми исходными текстами, поскольку это может нарушать лицензии открытого ПО и сделать недействительной всю работу организации. Если ваша организация может работать с лицензиями, включенными в программы с открытым кодом, вы можете их использовать. Некоторые организации также начинают открывать исходные тексты некоторых частей своего ПО. Такие "гибридные" лицензии становятся все более распространенными. Если вы решите сделать это, то необходимо убедиться, что вы ясно понимаете лицензию открытого ПО, а юристы должны внимательно исследовать этот вопрос.

Это не означает, что в вашей организации нельзя использовать ПО с открытыми исходными текстами. Если вы работаете сетевым администратором, то можете применять, например, межсетевой экран с открытыми исходными текстами. Многие компании по разработке ПО с закрытыми исходными текстами делают это, как бы лицемерно это ни выглядело. Вы не можете использовать открытые тексты для создания продукта, который не будет распространяться как открытое ПО.

Полный аутсорсинг информационных технологий

ПО с открытыми исходными текстами может не подойти, если ваш отдел ИТ не способен выполнить установку, компиляцию программ и т.д. Хотя большая часть программ с открытыми исходными текстами весьма проста в использовании, требуется определенный уровень подготовки. Если администрированием вашей системы занимаются только в свободное время, или если вы передали функции отдела ИТ сторонней организации, то использовать открытое ПО, наверное, не имеет смысла, если только ваш подрядчик не имеет значительного опыта в этой области.

Ограничительные корпоративные стандарты в области ИТ

Наконец, вы можете столкнуться с ограничениями корпоративных стандартов, которые либо требуют ориентироваться на определенных поставщиков, либо полностью запрещают использование открытого исходного кода. Это становится все менее распространенным, так как организации понимают, что неразумно ориентироваться только на одного поставщика. Долгое время игнорируемое большими компаниями ПО с открытыми исходными текстами уверенно покоряет корпоративную Америку. Такие компании как IBM - когда-то крупнейший поставщик патентованных продуктов с закрытыми исходными текстами - берут на вооружение и даже пропагандируют открытое ПО. Старое изречение, что "никто никогда не будет уволен за покупку (вставьте поставщика на выбор из голубых фишек)" в большинстве организаций больше не действует. Обновленной версией данного высказывания может быть "никто никогда не будет уволен за экономию денег организации с помощью решения, которое работает". Однако, конечно, предложение новой концепции может оказаться более рискованным, чем сохранение сложившегося порядка вещей.

Windows и ПО с открытыми исходными текстами

Так случилось, что ПО с открытыми исходными текстами разрабатывалось прежде всего в операционных системах на основе UNIX. Многие разработчики считают операционную систему Windows и создавшую ее компанию антиподом того, за что выступает движение за открытое ПО. И сама компания этого не отрицает; фактически корпорация Microsoft заказывает исследования, которые представляют открытое ПО в невыгодном свете, и активно борется на рынке с операционной системой Linux, которая начинает вторгаться в ее рыночную долю серверов. Однако независимо от отношения Microsoft к самой концепции, пользователи Windows активно создают для нее программы и выпускают их с открытыми исходными текстами. Существуют версии основных инструментальных средств мира UNIX и Linux для Windows. Эти программы иногда являются не полными аналогами своих собратьев из UNIX, но

существуют также программы с открытыми исходными текстами, которые выпущены только для платформы Windows, такие как анализатор беспроводных сетей NetStumbler, который рассматривается в [лекции 10](#).

Технический персонал ограничен рамками операционной системы, которую они могут использовать в корпоративной сети. Даже если они могут выбирать ОС самостоятельно, у них просто может не быть времени для загрузки и изучения одной из операционных систем с открытыми исходными текстами, которые рекомендуются в следующей лекции. Поэтому для каждой прикладной области, рассмотренной в этой книге, делаются попытки представить варианты как для UNIX, так и для Windows (зачастую они совпадают). Нравится это или нет, но Windows является доминирующей операционной системой для настольных компьютеров, а игнорирование данного факта послужило бы дурную службу техническим специалистам, которые хотят воспользоваться возможностями открытого ПО.

Лицензии для ПО с открытыми исходными текстами

Многие считают, что открытость исходных текстов означает свободу программного обеспечения от всех ограничений. Действительно, во многих случаях за программу не нужно платить. Однако почти все ПО с открытыми исходными текстами охватывается лицензией, с которой вы должны согласиться при его использовании; так же это делается для коммерческих продуктов. Обычно эта лицензия значительно менее ограничительна, чем традиционная лицензия ПО с закрытыми исходными текстами; однако она устанавливает границы того, что можно делать с программным обеспечением. Без этих ограничений ни один программист не будет чувствовать себя в безопасности при предоставлении результатов своей работы в общее достояние. При использовании открытого ПО убедитесь, что вы действуете в соответствии с этой лицензией. Проверьте также, что все сделанные модификации ей также соответствуют. Это важный момент: если ваша организация тратит много времени, модифицируя программу с открытыми исходными текстами для собственного использования, необходимо понимать, что вы, согласно лицензии, берете на себя некоторые обязательства.

Существует два основных типа лицензий для ПО с открытыми исходными текстами: Генеральная публичная лицензия GNU GPL и лицензия BSD. При правильном их понимании вы сможете уверенно использовать большую часть открытого ПО, не боясь запутаться в каких-либо вопросах авторского права. Существует несколько необычных лицензий для ПО с открытыми исходными текстами, появляющихся для таких вещей, как иллюстрации в компьютерных играх и т.д. Эти "гибридные" лицензии несколько сложнее, а их использование требует определенной осторожности, так как с вас могут потребовать плату или вы невольно нарушите авторские права.

Цель обеих основных лицензий для ПО с открытыми исходными текстами в большей степени состоит не в защите существующего программного обеспечения, а в контроле использования производного кода, полученного на основе этого программного обеспечения. В конце концов, оно обычно бесплатно, и первоначальный разработчик не будет возражать, если вы сделаете миллион копий и распространите их среди своих друзей. Но когда вы начнете делать изменения в программном обеспечении и захотите его распространить, вы должны быть аккуратны. Две основные лицензии для ПО с открытыми исходными текстами, их сходства и различия описаны далее.

Генеральная публичная лицензия GNU GPL

Генеральная публичная лицензия GNU GPL используется, пожалуй, наиболее часто. Она поддерживается Фондом свободного программного обеспечения, который способствует созданию и распространению ПО с открытыми исходными текстами, используя эту лицензию. Сообщество GNU работает над определенными программными проектами и ставит на них свою печать одобрения. Эти проекты в большинстве своем являются базовыми инструментами и библиотеками, такими как семейство компиляторов Gcc. Любой может использовать лицензию GPL для программного обеспечения, пока вы применяете его в исходном виде, без изменений и добавлений. Ее используют многие разработчики, потому что она была проверена группой юристов и прошла испытание временем. Она настолько распространена, что когда говорят о "программе с GPL", то люди обычно понимают, что речь идет о программе, выпущенной с открытыми исходными текстами согласно лицензии GPL.

GPL сложнее, чем другая основная лицензия для ПО с открытыми исходными текстами, лицензия BSD. Она имеет больше ограничений на использование кода держателя лицензии, что делает ее более подходящей для организаций, которые создают коммерческий продукт. Обычно, если вы лицензируете что-то под GPL, то это понимается как свободное программное обеспечение. Поставщик, однако, может требовать плату за упаковку, распространение и поддержку. В этой области многие компании делают деньги на том, что, по общему мнению, является бесплатным пакетом. Свидетельством тому служат розничные пакеты разновидностей Linux, коммерческие версии web-серверов Apache и коммуникационного пакета Sendmail. Однако, если вы скачиваете или загружаете с компакт-диска ПО, распространяемое по лицензии GPL, и не указываете номер кредитной карты, то можно предположить, что вы не должны никому никаких денег.

Реальная ценность лицензии GPL с точки зрения разработчика состоит в том, что она позволяет автору программы поддерживать авторские права и некоторые привилегии, делая ее при этом бесплатной для максимального числа людей. Лицензия GPL также разрешает дальнейшее развитие, не беспокоясь, что исходный разработчик может не выдержать конкуренции с патентованной версией собственной программы.

В базовой форме лицензия GPL позволяет произвольным образом использовать и распространять программу со следующими ограничениями:

- Если вы распространяете свою работу, то должны включить авторские права исходного автора и GPL во всей полноте. Это делается для того, чтобы любой будущий пользователь вашего дистрибутива полностью понимал свои права и ответственность согласно GPL.
- Когда вы распространяете программу, вы всегда должны делать доступной версию исходных текстов. Можно также распространять бинарный код, но только вместе с исходными текстами. Это обеспечивает цель концепции открытого ПО. Если бы распространялся только бинарный код свободной программы и требовалось разыскивать его создателя, чтобы получить доступ к исходным текстам, то мощь свободного ПО оказалась бы существенно сниженной. Лицензия GPL гарантирует, что каждый получатель программы будет иметь полную возможность увидеть исходные тексты.
- Если вы делаете какие-либо изменения в программе и выпускаете или распространяете ее, вы должны сделать доступными исходные тексты изменений таким же образом, как и первоначальные, то есть общедоступными и с лицензией GPL. Ключевая фраза здесь "и выпускаете или распространяете ее". Если вы ее не выпускаете, то вы не обязаны выпускать исходные тексты. Если вы делаете индивидуальные изменения для своей организации, то она может не беспокоиться о распространении результатов ваших усилий. Пока вы не выпускаете программу и не намерены ее продавать, исходные тексты могут оставаться закрытыми.

Однако хорошим тоном считается выпуск модифицированных программ с лицензией GPL. Это не только создает большую доброжелательность со стороны сообщества открытого ПО, но также гарантирует, что ваши изменения окажутся совместимыми с будущими версиями программы и полностью протестированы. Можно использовать эту логику, чтобы убедить свою организацию, что в этом случае можно получить опыт и бесплатную рабочую силу всех остальных программистов проекта. Обычно выпуск таких программ не влияет на конкурентоспособность компании, если только это не является частью основного бизнеса, и в этом случае ПО с открытыми исходными текстами может быть вообще неуместным. И, наконец, это не повредит вашей репутации и работе вместе с другими разработчиками проекта и в любом сообществе разработчиков программного обеспечения.

Приложение А содержит полный текст лицензии GPL. Можно получить его в других текстовых форматах по адресу <http://www.gnu.org/licenses/gpl.html>.

Лицензия BSD

Лицензия BSD является лицензией для ПО с открытыми исходными текстами, с которой была выпущена исходная версия BSD UNIX. После того, как разработчики выиграли судебное дело против AT&T по поводу исходной лицензии, они выпустили программное обеспечение в общее пользование с разрешающей лицензией BSD. Основное отличие от GPL состоит в том, что лицензия BSD не включает требования выпуска модификаций под той же лицензией. На основе этого некоторые компании продолжили выпуск коммерческих версий UNIX на основе базового кода BSD. BSDI является одной из таких компаний. Некоторые считают, что это противоречит идее открытого ПО, когда компания может взять улучшенную версию и требовать за нее плату, в то время как другие полагают, что это стимулирует нововведения, создавая коммерческую заинтересованность. В любом случае, это породило целое семейство свободных версий UNIX, включая FreeBSD, NetBSD и OpenBSD, и ряд коммерческих, таких как BSDI. Приложение А содержит полный текст лицензии BSD. Можно также найти его по адресу <http://www.opensource.org/licenses/bsd-license.php>.

Теперь, когда вы имеете представление об основах информационной безопасности и ПО с открытыми исходными текстами, мы переходим к конкретным вопросам: установка, настройка и использование реальных пакетов программного обеспечения. В следующих лекциях рассматриваются программы, которые могут помочь в защите вашей сети и данных различными способами. Эти лекции организованы согласно различным аспектам информационной безопасности, в них рассмотрено большинство основных областей ИБ. Многие инструменты могут иметь различные применения. Например, хотя программа Snort рассмотрена в лекции о системах обнаружения вторжений, она может использоваться и в криминалистической работе. И, конечно, если вы интересуетесь инструментом для определенной области применения, то можно перейти непосредственно к интересующему вас разделу.

Инструменты безопасности с открытым исходным кодом

2. Лекция: Средства уровня операционной системы: версия для печати и PDA

Большинство средств, рассмотренных в этой книге, являются прикладными программами. Для их выполнения требуется поддержка операционной системы. Если рассматривать эти программы как инструментарий информационной безопасности, то операционную систему можно считать верстаком. Если ОС неустойчива, то работа по защите данных будет от этого страдать; вы никогда не сможете полностью доверять поступающим от нее данным. На самом деле, ваша ОС может даже понизить первоначальную безопасность сети. На жаргоне компьютерной безопасности окружение, в котором функционирует защищенная ОС, называется доверенной вычислительной базой (ДВБ). ДВБ включает в себя полный список элементов, обеспечивающих безопасность: операционную систему, программы, сетевое оборудование, средства физической защиты и даже организационные процедуры. Краеугольным камнем этой пирамиды служит операционная система. Без нее доверенная вычислительная база оказывается построенной на зыбучем песке.

Обзор лекции

Изучаемые концепции:

- Введение в доверенную вычислительную базу
- Рекомендации по настройке системы средств защиты
- Повышение безопасности операционной системы
- Основы использования средств, относящихся к уровню операционной системы

Используемые средства:

Bastille Linux, ping, traceroute, whois, dig, finger, ps, OpenSSH и Sam Spade for Windows.

Многие компьютерные атаки нацелены на операционную систему. Современные операционные системы "раздулись" до таких размеров, что одному человеку стало крайне сложно полностью понимать, что происходит "под капотом". XP, самая свежая версия Windows, содержит более 50 миллионов строк исходных текстов. Хотя она считается самой защищенной версией Windows (согласно Microsoft), почти ежедневно в ее системах защиты находят новые ошибки. Чем более сложным становится продукт, тем вероятнее, что непредвиденные исходные данные приведут к непредвиденному (но желательному для хакеров) результату.

Раньше программное обеспечение строилось контролируемым образом - прикладные программы предоставлялись или одобрялись поставщиком компьютеров. В наше время, с появлением Интернета и поддерживающих Java и ActiveX web-навигаторов, в компьютер могут поступать всевозможные виды трафика и кода, не предусмотренные при проектировании. Огромный объем программ в сочетании со всеми видами потоков данных, поступающих из Интернета, приводит к тому, что операционные системы со временем становятся не более, а менее защищенными, особенно если они используются без дополнительной настройки, в том виде, в котором поставляются изготовителями.

Добавьте к этому стремление поставщиков сделать компьютеры как можно более "готовыми к употреблению", чтобы пользователи могли просто включить их и приступить к работе. Может быть, для массового пользователя это и хорошо, но для безопасности, несомненно, плохо. Большинство защитных средств по умолчанию выключены, многие программы и службы загружаются автоматически, независимо от того, нужны они пользователю или нет, а к системе, в попытке превзойти конкурентов, приделано множество украшений. Хотя Microsoft Windows - наиболее злостный рецидивист в этой области, потребительские версии Linux немногим лучше, и даже операционные системы серверного уровня страдают этим грехом. Стандартная установка RedHat Linux по-прежнему загружает значительно больше служб и программ, чем требуется или хочется рядовому пользователю. В Windows Business Server 2000 по умолчанию устанавливается web-сервер. И хотя в Windows XP прежняя политика "открытых дверей" улучшена, в продукте при установке по умолчанию по-прежнему остаются дыры в безопасности.

Обеспечение доверия безопасности системы защитных средств важно по нескольким причинам. Прежде всего, если нарушается безопасность расположенного на "передовой" защитного устройства, такого как межсетевой экран, перестают действовать и предоставляемые им защитные функции. Если это устройство оповещения, например, система обнаружения вторжений, то потенциальные нарушители могут оккупировать компьютер и отключить систему раннего предупреждения. Или, что еще хуже, они могут изменить данные таким образом, что их действия не будут протоколироваться. Это может создать ложное чувство безопасности, в то время как нарушитель свободно орудует в вашей сети.

Существуют хакерские программы, разработанные именно для этой цели. Они изменяют некоторые системные файлы так, что любые данные, выходящие из компьютера, могут контролироваться хакером. Компьютеру, который был инфицирован одной из таких программ, никогда нельзя доверять. Практичнее всего переформатировать диск и начать все сначала.

Наконец, если неавторизованные пользователи получили контроль над вашей системой безопасности, они могут применить те же самые используемые вами защитные средства против вас и других сетей. Подключенный к Интернету компьютер с установленным инструментарием безопасности может стать очень ценной добычей для склонного к озорству злоумышленника.

Обеспечение доверия к безопасности базовой операционной системы на вашем защитном компьютере - первое, что вам следует сделать, прежде чем вы загрузите какие-либо средства или установите дополнительные программы. В идеале вы должны создать систему защитных средств с нуля, устанавливая гарантированно новую операционную систему. Таким образом вы можете быть уверены, что ни одна программа или процесс не будут мешать средствам безопасности. Это также гарантирует, что базовая операционная система свободна от каких-либо ранее измененных или вредоносных программ. Если по какой-то причине необходимо установить защитный инструментарий поверх ранее установленной операционной системы, обеспечьте выполнение приведенных ниже в данной лекции указаний по повышению безопасности ОС и защите системы. Далее в этой лекции рассматривается Bastille Linux, средство, используемое для достижения этой цели на платформе Linux. Существуют доступные бесплатные утилиты от Microsoft для повышения безопасности Windows. Можно также использовать описанные в [лекции 5](#) средства для сканирования уязвимостей существующей системы.

Выбор ОС для системы средств безопасности определяет, как вы собираетесь ее защищать. Я рекомендую операционную систему с открытыми исходными текстами, такую как Linux или BSD, но Windows также будет прекрасно работать, если вы ее сначала правильно обезопасите. Я использовал Mandrake Linux для установки и выполнения рекомендованных в этой книге средств на платформе Linux, которые, впрочем, можно применять на большинстве дистрибутивов Linux и вариантов операционной системы BSD или UNIX.

Как упоминалось в [лекции 1](#), доступно множество операционных систем с открытыми исходными текстами. Большинство из них основано на UNIX, хотя все они снабжены графическим интерфейсом X-Window и менеджерами окон, такими как KDE и GNOME. Эти интерфейсы привычны каждому, кто работал в Microsoft Windows, но есть и некоторые отличия.

Я не сторонник мнения, что какая-то операционная система в плане безопасности по своей природе лучше других. Все зависит от того, как она используется и как сконфигурирована, поэтому ниже следует длинный раздел о повышении безопасности установки ОС. Я использовал Linux, потому что имею наибольший опыт работы с этой операционной системой и убедился, что она совместима с большинством используемых систем. При наличии более 50 миллионов пользователей во всем мире и нескольких дюжин вариантов, Linux предоставляет широчайшее многообразие программ, и большинство защитных средств с открытыми исходными текстами, упоминаемых в этой книге, разработаны специально для этой ОС.

Первое обсуждаемое средство автоматизирует повышение безопасности Linux-системы, гарантируя, что вы работаете с рабочей станцией, безопасной настолько, насколько это изначально возможно. Приводятся также некоторые общие рекомендации, как правильно обезопасить операционную систему Windows для использования в качестве защитной рабочей станции. Наконец, вы освоите некоторые средства уровня операционной системы. Существует ряд функций системного уровня, которые вы будете регулярно использовать в защитных приложениях, и некоторые из них включены в раздел инструментария.

Данная лекция не претендует на роль исчерпывающего руководства по безопасности какой-либо из упоминаемых операционных систем. Скорее, она является обзором основ и некоторых используемых средств.

Повышение безопасности системы защитных средств

После установки операционной системы необходимо повысить ее безопасность для применения в качестве защитной системы. Этот процесс подразумевает отключение ненужных служб, ужесточение прав доступа и, как правило, минимизацию видимых извне областей компьютера. Детали выполняемых действий варьируются в зависимости от предполагаемого использования компьютера и от операционной системы.

Повышение безопасности обычно достигается напряженным ручным трудом, просмотром и модификацией всех возможных настроек. На эту тему для каждой конкретной операционной системы написано множество книг. Однако вам не придется целиком читать какую-либо другую книгу, если вы укрепляете операционную систему Linux - теперь для этого существуют автоматические средства. В результате не только экономится время, но и снижается вероятность упущений.

Bastille Linux: Программа повышения безопасности ОС для Linux

Bastille Linux

Автор/основной контакт: Jay Beale

Web-сайт: <http://www.bastille-linux.org/>

Платформы: Linux (RedHat, Mandrake, Debian), HP/UX

Лицензия: GPL

Рассмотренная версия 2.1.1

Важные адреса электронной почты:

Общие вопросы: jon@lasser.org

Технические вопросы: jay@bastille-Linux.org

Списки почтовой рассылки:

Извещения Bastille Linux:

<http://lists.sourceforge.net/mailman/listinfo/bastille-Linux-announce>

Разработка Bastille Linux:

<http://lists.sourceforge.net/mailman/listinfo/bastille-Linux-discuss>

Системные требования:

Perl 5.5_003 или выше

Perl TK Module 8.00.23 или выше

Perl Curses Module 1.06 или выше

Первым защитным средством является инструмент повышения безопасности операционной системы, именуемый Bastille Linux. Несмотря на название, это не самостоятельная операционная система, а, скорее, набор командных файлов, которые просматривают и устанавливают некоторые системные параметры, основываясь на ваших подсказках. Данное средство существенно упрощает процесс повышения безопасности, сводя его к ответам на некоторые вопросы. Возможна также установка межсетевого экрана (см. следующую лекцию). Bastille Linux можно запускать в Mandrake, RedHat, Debian и

HP/UX (последняя ОС даже не является вариантом Linux). Джей Бил, разработчик, продолжает выпускать версии, поддерживающие другие дистрибутивы Linux.

Установка Bastille Linux

Bastille написан с применением инструментального пакета Curses (вот уж действительно подходящее имечко для языка программирования!) ("curses" в переводе с английского означает "проклятие". - прим. ред.).

1. Сначала необходимо загрузить и установить Perl Curses и модули ТК, от которых зависит Bastille. Их можно получить со страницы на сайте Bastille:
<http://www.bastille-linux.org/perl-rpm-chart.html>.
2. Пользователи RedHat должны также установить пакет с именем Pwlib, который можно получить с той же страницы. Для установки пакета запустите в командной строке RPM с параметрами, заданными на этой странице.
3. После установки требуемых модулей загрузите RPM Bastille или возьмите его с прилагаемого к книге компакт-диска. Щелкните на нем мышью, и Bastille должен установиться автоматически.

Теперь можно запустить Bastille, чтобы повысить (укрепить) безопасность вашей операционной системы.



Флэми Тех советует:

Сначала запустите Bastille на непроизводственной системе!

В первый раз всегда запускайте все средства на непроизводственной или тестовой системе. Эти программы могут отключать службы, необходимые для функционирования web-сервера или сервера электронной почты, вызвав тем самым перебои в работе. Только после полного тестирования всех эффектов и проверки стабильности можно запускать их в производственной среде.

Запуск Bastille Linux

1. Если при установке ОС вы не задали запуск X-Window при загрузке, наберите `startx` в командной строке, и на экране появится графический интерфейс X-Window.
2. Запустите Bastille в интерактивном режиме, щелкнув мышью на значке Bastille, расположенном в каталоге `/usr/bin/bastille`. Можно также набрать `bastille` в терминальном окне, открытом в X.
3. Если вы не хотите или в силу каких-то причин не можете использовать Bastille в X-Window, можно запустить Bastille из командной строки, используя интерфейс на основе Curses.

Наберите

```
bastille c
```

в командной строке. Оба интерфейса дадут одинаковые результаты.

Можно запустить Bastille и в неинтерактивном режиме. В этом случае Bastille выполняется автоматически, не задавая никаких вопросов и действуя согласно предварительно созданному конфигурационному файлу. Конфигурационный файл создается при каждом запуске Bastille. После этого его можно использовать для выполнения Bastille на других компьютерах в неинтерактивном режиме. Этот метод полезен для быстрого повышения безопасности множества компьютеров. Если у вас есть конфигурационный файл, который делает то, что требуется, просто загрузите Bastille на другие машины и

скопируйте на них конфигурационный файл (или предоставьте им доступ к этому файлу по сети). Затем введите `bastille non-interactive config-file` (здесь `config-file` - это маршрутное имя нужного конфигурационного файла).

Чаще всего, однако, Bastille будет выполняться в интерактивном режиме. В этом режиме вы отвечаете на последовательность вопросов о том, как вы будете использовать компьютер. На основе ответов Bastille включает ненужные службы или ограничивает привилегии пользователей и служб. Он спрашивает что-нибудь вроде: "Вы собираетесь использовать этот компьютер для доступа к машинам с Windows?" При отрицательном ответе он отключает сервер Samba, который позволяет вашему компьютеру взаимодействовать с Windows-машинами. Потенциально Samba может создать некоторые уязвимости в вашей системе, поэтому, если он не нужен, его лучше отключить. Если требуется запускать некоторые серверы (например, SSH), то Bastille будет пытаться установить их с ограниченными привилегиями или использовать сдвиг корня файловой системы. Последнее означает, что если сервер должен выполняться с привилегиями root, его возможности по воздействию на другие части системы будут ограничены. Это смягчает последствия успешных атак на службу.

Каждый вопрос сопровождается пояснением, почему эта настройка важна, так что можно решить, подходит ли она для вашей установки. Имеется также кнопка "More detail" (Подробнее) для получения дополнительной информации. Bastille использует новейший подход, пытаясь обучать администратора в процессе повышения безопасности системы. Чем больше у вас информации, тем лучше вы будете вооружены для выполнения обязанностей по защите сети.

Можно пропустить вопрос, если вы не вполне уверены в ответе, и вернуться к нему позднее. Не беспокойтесь, в конце у вас будет возможность придать окончательный вид всем настройкам. Можно также запустить Bastille позже, когда ответ будет найден, и изменить настройку в это время. Еще одна приятная особенность данного средства - предоставление в конце сеанса списка "неделок" для всех элементов, оставшихся ненастроенными.

Теперь вы получили защищенный компьютер Linux для запуска средств безопасности. Если вы новичок в операционных системах на основе UNIX, то желательно ознакомиться с основными командами и навигацией. Если вы когда-то использовали DOS, то многие команды окажутся знакомыми, хотя их синтаксис несколько отличается. Одно из наиболее существенных различий между Windows и Linux и другими операционными системами на основе UNIX состоит в учете регистра символов в файловой системе. Приложение B содержит краткую таблицу наиболее часто используемых команд Linux и UNIX. Найдите время попрактиковаться в работе с операционной системой и убедитесь, что можете делать простые вещи, такие как смена текущего каталога, копирование файлов и т.д.

Существует несколько команд операционной системы, которые часто используются в защитной деятельности. Они не являются в полном смысле слова отдельными программами для защиты скорее - это утилиты операционной системы, которые можно применять для генерации данных безопасности. Они настолько часто используются в последующих лекциях и в целом в работе по обеспечению безопасности, что я хотел бы детально обсудить их.

ping: средство диагностики сети

ping

Автор: Mike Muus (покойный)

Web-сайт: <http://ftp.arl.mil/~mike/ping.html>

Платформы: Большинство платформ UNIX и Windows

Лицензии: Различные

Справочная информация в UNIX:

Наберите `man ping` в командной строке.

Если вы имели дело с системами в Интернете, то, вероятно, использовали `ping`, но в приложениях безопасности `ping` применяется специфическим образом и по-особому обрабатывается. `ping` расшифровывается как Packet Internet Groper (пакетный межсетевой щуп, звучит не вполне политкорректно) и является диагностическим средством, встроенным ныне в большинство стеков TCP/IP. Многие считают, что `ping` напоминает радар подводной лодки: испускается, отражается от цели и возвращается. Хотя это и хорошая общая аналогия, она не вполне точно отражает то, что происходит при эхо-

тестировании. `ping` использует сетевой протокол, называемый **ICMP** (Internet Control Message Protocol - межсетевой протокол управляющих сообщений). Эти сообщения применяются для передачи информации о сетях. `ping` использует ICMP-сообщения типов 8 и 0, которые также известны как Echo Request (Запрос отклика) и Echo Reply (Отклик) соответственно. Когда выдается команда `ping`, компьютер посылает запрос отклика другому компьютеру. Если машина на другом конце доступна и поддерживает совместимый стек TCP, то она ответит откликом. Ping-коммуникации в целом выглядят следующим образом:

Система А посылает `ping` системе В: Echo Request, "Есть кто-нибудь?"

Система В получает запрос отклика и отправляет назад отклик, "Да, есть."

В типичном сеансе `ping` это повторяется несколько раз, чтобы проверить, теряют ли пакеты целевая машина или сеть. `ping` применяется также для определения задержки, то есть времени, которое требуется пакету для перемещения между двумя точками.

При использовании `ping` можно получить от хоста и другие типы ICMP-сообщений. Каждый из них имеет свой смысл, объясняемый в последующих лекциях.

- Сеть недоступна.
- Хост недоступен.

С помощью `ping` о хосте можно узнать не только то, работает он или нет, но и многое другое. Как вы увидите далее, способ, которым компьютер отвечает на `ping`, часто показывает, какая операционная система на нем функционирует. Можно также использовать `ping` для генерации поискового DNS-запроса и получения имени целевого хоста (если таковое имеется). Иногда это позволяет определить, является ли компьютер сервером, маршрутизатором или, возможно, домашним компьютером с коммутируемым или широкополосным соединением. Можно эхо-тестировать IP-адрес или полностью заданное доменное имя. В [табл. 2.1](#) перечислены дополнительные ключи и опции команды `ping`, которые могут оказаться полезными.

Таблица 2.1. Опции ping

| Опция | Описание |
|------------|--|
| -c count | Посылает сообщение ping <code>count</code> раз. В системах Linux и UNIX по умолчанию сообщения посылаются непрерывно, в Windows - четыре раза |
| -f | Ping-наводнение. Посылается максимально возможное число пакетов в максимально быстром темпе. Это полезно для тестирования, чтобы увидеть, теряет ли хост пакеты, так как графически отображается, на сколько запросов поступили ответы. Будьте очень осторожны с этой командой, так как она может очень легко забить машину или сеть |
| -n | Не выполнять DNS на IP-адрес. Это может ускорить ответ и исключить проблемы с DNS при диагностике сетевых проблем |
| -s size | Посылает пакеты длины <code>size</code> . Это полезно при тестировании того, как машина или маршрутизатор обрабатывает большие пакеты. Ненормально большие пакеты часто используют в атаках на доступность, чтобы сбить или подавить систему |
| -p pattern | Посылает <code>pattern</code> в качестве полезной нагрузки пакета ICMP. Это также хорошая проверка того, как машина реагирует на необычные ICMP-воздействия |

tracert (UNIX) или tracert (Windows): средства диагностики сети

| tracert (UNIX) или tracert (Windows) |
|--|
| Автор/основной контакт: Неизвестен |
| Web-сайты: http://www.traceroute.org/ ; http://www.tracert.com/ |
| Платформы: Большинство платформ UNIX и все платформы Windows |
| Лицензии: Различные |

Справочная информация в UNIX:

Наберите `man traceroute` в командной строке

Эта команда аналогична `ping`, но предоставляет намного больше информации об удаленном хосте. По сути, `traceroute` эхо-тестирует хост, но при отсылке первого пакета устанавливает поле TTL (Time To Live - время жизни) пакета, равным единице. Этот параметр управляет количеством межсетевых переходов, которые может претерпеть пакет, прежде чем прекратить свое существование. Следовательно, первый пакет дойдет только до первого маршрутизатора или машины дальше вашей в Интернет, а затем вернется сообщение о том, что время жизни пакета истекло. Затем посылается следующий пакет с TTL, равным 2 и так далее, пока не будет достигнута цель. Это показывает виртуальный путь (маршрут), которым идут пакеты. Выясняется также имя каждого хоста на пути следования, поэтому можно видеть, как ваш трафик пересекает Интернет. Очень интересно видеть, как пакет, направленный из Хьюстона в Даллас, скачет от восточного до западного побережья, покрывая тысячи миль, чтобы за доли секунды достичь цели.

Это средство полезно, когда вы пытаетесь проследить источник или расположение злоумышленника, следы которого вы обнаружили в своих регистрационных файлах или тревожных сообщениях. Можно проследить маршрут до IP-адреса и кое-что узнать о нем. Результаты могут показать, имеете ли вы дело с домашним пользователем или сотрудником компании, кто является поставщиком Интернет-услуг (которому вы можете подать жалобу на ненадлежащее поведение), какой тип подключения используется и каковы его скоростные характеристики, где территориально он находится (иногда, в зависимости от содержательности промежуточных точек). [Листинги 2.1](#) и [2.2](#) содержат примеры использования `traceroute`.

```
Tracing route to www.example.com (Трассировка маршрута к www.example.com)
over a maximum of 30 hops: (не более чем за 30 переходов)
 1 <10 ms <10 ms <10 ms 192.168.200.1
 2 40 ms 60 ms 160 ms 10.200.40.1
 3 30ms 40ms 100ms gateway.smallisp.net
 4 100 ms 120ms 100ms iah-core-03.inet.genericisp.net [10.1.1.1]
 5 70 ms 100 ms 70 ms dal-core-03.inet.genericisp.net [10.1.1.2]
 6 61 ms 140 ms 70 ms dal-core-02.inet.genericisp.net [10.1.1.3]
 7 70 ms 71 ms 150 ms dal-brdr-02.inet.genericisp.net [10.1.1.4]
 8 60 ms 60 ms 91 ms 192.168.1.1
 9 70 ms 140 ms 100 ms sprintdslcust123.hou-pop.sprint.com [192.168.1.2]
10 101 ms 130 ms 200 ms core-cr7500.example.com [192.168.1.2]
11 180 ms 190 ms 70 ms acmefirewall-hou.example.com [216.32.132.149]
12 110 ms 110 ms 100 ms www.example.com [64.58.76.229]
Trace complete. (Трассировка завершена)
```

Листинг 2.1. `traceroute`, пример 1

На [листинге 2.1](#) имена реальных поставщиков Интернет-услуг изменены на вымышленные, но общая идея сохранена. Выполнив эту простую команду, можно понять, что исследуемый IP-адрес принадлежит компании с именем Асте, что это, вероятно, web-сервер, что он установлен в защищаемой межсетевым экраном сети или в демилитаризованной зоне, что в роли поставщика Интернет-услуг выступает Sprint и что он расположен в Хьюстоне. Многие сетевые администраторы и крупные поставщики Интернет-услуг используют географические сокращения или инициалы для именования своих маршрутизаторов, поэтому, глядя на DNS-имена и следуя по цепочке маршрутизаторов, можно прийти к выводу, что `hou-pop.sprint.com` является маршрутизатором компании Sprint в Хьюстоне.

```
Tracing route to resnet169-136.plymouth.edu [158.136.169.136]
(Трассировка маршрута к resnet169-136.plymouth.edu)
Over a maximum of 30 hops: (не более чем за 30 переходов)
 1 <1 ms <1 ms 192.168.200.1
 2 12 ms 7 ms 8 ms 10.200.40.1
 3 26 ms 28 ms 11 ms iah-edge-04.inet.qwest.net [63.237.97.81]
 4 37 ms 15 ms 12 ms iah-core-01.inet.qwest.net [205.171.31.21]
 5 51 ms 49 ms 47 ms dca-core-03.inet.qwest.net [205.171.5.185]
```

```
6 52 ms 55 ms 65 ms jfk-core-03.inet.qwest.net [205.171.8.217]
7 73 ms 63 ms 58 ms jfk-core-01.inet.qwest.net [205.171.230.5]
8 94 ms 67 ms 55 ms bos-core-02.inet.qwest.net [205.171.8.17]
9 56 ms 56 ms 60 ms bos-brdr-01.ip.qwest.net [205.171.28.14]
10 64 ms 63 ms 61 ms 63.239.32.230
10 67 ms 59 ms 55 ms so-7-0-0-0.core-rtr1.bos.verizon-gni.net [130.81.4.181]
11 56 ms 61 ms 62 ms so-0-0-1-0.core-rtr1.man.verizon-gni.net [130.81.4.198]
12 58 ms 59 ms 57 ms so-0-0-0-0.core-rtr2.man.verizon-gni.net [130.81.4.206]
13 59 ms 57 ms 64 ms a5-0-0-732.g-rtr1.man.verizon-gni.net [130.81.5.126]
15 74 ms 62 ms 61 ms 64.223.133.166
16 68 ms 67 ms 68 ms usnh-atm.inet.plymouth.edu [158.136.12.2]
17 80 ms 2968 ms 222 ms xhyd04-3.plymouth.edu [158.136.3.1]
18 75 ms 2337 ms 227 ms xspe04-2.plymouth.edu [158.136.2.2]
19 74 ms 65 ms 72 ms resnet169-136.plymouth.edu [158.136.169.136]
Trace complete. (Трассировка завершена)
```

Листинг 2.2. traceroute, пример 2

Из примера трассировки на [листинге 2.2](#) можно заключить, что рассматриваемый IP-адрес, вероятно, используется студентом Плимутского государственного университета (Плимут, Нью-Хемпшир). Как это можно узнать? Прежде всего, завершающее доменное имя обозначает шлюз. Если проследить маршрут, то он идет от bos (Boston) до man (Manchester), затем к plymouth.edu.; .edu означает, что это университет. Это была догадка на основе опыта, но ее можно проверить, посетив web-сайт plymouth.edu. Далее, выясненным именем хоста является resnet169-136. Имя подсказывает, что это сеть студенческого общежития (студенческих резиденций).

Можно видеть, что иногда толкование результатов трассировки напоминает расследование, являясь скорее искусством, чем наукой, но со временем вы узнаете больше и станете лучше понимать, что означает каждое сокращение.

Трассировка дает много информации для преследования лица, использовавшего данный IP-адрес как исходную точку вторжения или атаки. В примере на [листинге 2.1](#) можно пойти на web-сайт компании, чтобы найти основной контактный адрес. Можно подать жалобу поставщику Интернет-услуг. Крупные поставщики Интернет-услуг обычно предоставляют специальный адрес электронной почты или контактный телефон для жалоб и контролируют соблюдение клиентами условий договора на обслуживание. Альтернативный вариант - воспользоваться следующей командой, `whois`, чтобы выяснить технические контактные координаты организации.

whois: средство опроса DNS

whois

Автор/основной контакт: Неизвестен

Web-сайт: Неизвестен

Платформы: Большинство платформ UNIX

Лицензии: Различные

Справочная информация в UNIX: Наберите `man whois` в командной строке.

Команда `whois` полезна при выяснении контактной информации лица, создавшего проблемы в вашей сети. Эта команда опрашивает основные серверы доменной системы имен и возвращает всю информацию, которой располагает Internic (или другой регистратор имен). Internic - бывшее квазиправительственное агентство, ответственное за отслеживание всех доменных имен в Интернете. Затем Internic стала коммерческой компанией

Network Solutions и была приобретена компанией VeriSign. Теперь, когда регистрация имен открыта для конкуренции, существуют буквально дюжины официальных регистраторов имен. Однако с помощью команды `whois`, как правило, можно по-прежнему найти, кому принадлежит домен.

Команда полезна для прослеживания источников атак, идущих как изнутри компаний, так и из сетей поставщиков Интернет-услуг. В любом случае можно разыскать человека, ответственного за эту сеть, и сообщить ему о ваших проблемах. Это не всегда может оказаться полезным, но попробовать стоит в любом случае. Синтаксис команды таков:

```
whois domain-name.com
```

Аргумент `domain-name.com` - это имя интересующего вас домена. На [листинге 2.3](#) показана информация, которую можно получить.

```
Registrant:
Example Corp (EXAMPLE.DOM)
  123 Elm, Suite 123
  New York, NY 10000
  US
  212-123-4567
Domain Name: EXAMPLE.COM

Administrative Contact:
Jones, Jane (JJ189) jane.jones@example.com
  123 Elm, Ste 123
  New York, NY 10000
  212-123-4567

Technical Contact:
John Smith (JS189) john.smith@example.com
  123 Elm, Ste 123
  New York, NY 10000
  212-123-4567

Record expires on 06-Oct-2006.
Record created on 05-Sep-2002.
Database last updated on 30-Apr-2004 21:34:52 EDT.

Domain servers in listed order:

NS.EXAMPLE.COM 10.1.1.1
NS2.EXAMPLE.COM 10.1.1.2
```

Листинг 2.3. Результаты `whois`

Изучив [листинг 2.3](#), можно напрямую связаться с техническим специалистом, отвечающим за интересующий вас домен. Если это не поможет, всегда можно связаться с кем-то из административного персонала. Команда `whois` обычно выдает адрес электронной и обычной почты и иногда - номера телефонов. Видно, когда домен был создан, и вносились ли в последнее время изменения в регистрационные данные. Выдача также показывает серверы доменных имен, ответственные за это имя. Опросив эти данные с помощью команды `dig` (описанной далее), можно получить еще больше информации о конфигурации удаленной сети.

К сожалению, команда `whois` не встроена в платформы Windows, но существует множество `whois`-машин на web-платформе; см., например, web-сайт Network Solutions: <http://www.networksolutions.com/cgi-bin/whois/whois>



Флэми Тех советует:

Не вываливайте в `whois` содержимое корпоративных рабочих столов!

Если вы администрируете собственный домен, то проследите, чтобы выдача `whois` была актуальной и носила максимально общий характер. Размещение реальных фамилий и адресов электронной почты в полях контактной информации позволяет посторонним использовать эти данные либо в целях морально-психологического воздействия, либо для атаки путем подбора паролей. Кроме того, когда люди покидают организацию, такая запись становится неактуальной. Лучше использовать абстрактные адреса электронной почты, такие как `dnsmaster@example.com` или `admin@example.com`. Вы можете перенаправлять поступающие по этим адресам сообщения соответствующим должностным лицам, не раскрывая ценную информацию о технической структуре организации.

dig: средство опроса DNS

dig

Автор/основной контакт: Andrew Scherpbeir

Web-сайт: <http://www-search.ucl.ac.uk/htdig-docs/author.html>

Платформы: Большинство платформ UNIX

Лицензии: Различные

Справочная информация в UNIX: Наберите `man dig` в командной строке.

Команда `dig` запрашивает у сервера имен определенную информацию о домене. `Dig` является обновленной версией команды `nslookup`, постепенно выходящей из употребления. С помощью `dig` можно выяснить имена используемых в сети машин и связанные с ними IP-адреса, определить, которая из них является сервером электронной почты, и получить другую полезную информацию. Общий синтаксис команды `dig` таков:

```
dig @сервер домен тип
```

Здесь сервер - опрашиваемый сервер DNS, домен - интересующий вас домен, тип - тип запрашиваемой информации. Обычно опрашивают уполномоченный DNS-сервер домена, то есть сервер, указанный в данных `whois` как самый авторитетный источник информации о домене. Иногда этот сервер поддерживает компания; порой это делает поставщик Интернет-услуг. В [табл. 2.2](#) перечислены виды записей, которые можно запрашивать с помощью аргумента "тип".

На [листинге 2.4](#) приведен пример результатов работы команды `dig`. Можно видеть, что был загружен весь файл с данными о зонах домена. Это предоставляет ценную информацию, такую как имена хостов сервера электронной почты, сервера DNS и других важных машин в сети. Если вы поддерживаете сервер DNS, то должны иметь возможность сконфигурировать его для ответа на запросы этих видов только от авторизованных машин.

```
; <<>> DIG 9.2.1 <<>> @ns.example.com.com example.com ANY
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 4

;; QUESTION SECTION:
;example.com IN ANY

;; ANSWER SECTION:
```

```
example.com. 86400 IN MX 10 mail.example.com.
example.com. 2560 IN SOA ns.example.com
hostmaster.example.com. 1070057380 16384 2048 1048576 2560
example.com. 259200 IN NS ns.example.com.
example.com. 259200 IN NS ns2.example.com.
example.com. 86400 IN A 10.1.1.1

;; ADDITIONAL SECTION:
nat1.example.com. 86400 IN A 10.1.1.2
ns.example.com. 86400 IN 10.1.1.3
ns2.example.com. 86400 IN A 10.1.1.4
sql.example.com 86400 IN A 10.1.1.5
www.example.com 86400 IN A 10.1.1.6

;; Query time: 107 msec
;; SERVER: 64.115.0.245#53 (ns.example.com)
;; WHEN: Wed Dec 31 18:39:24 2003
;; MSG SIZE rcvd: 247
```

Листинг 2.4. Выдача команды dig @ns.example.com AXFR

Таблица 2.2. Типы записей dig

| Опция | Описание |
|-------|---|
| AXFR | Попытка получить весь файл для домена или файл с данными о зонах. Конфигурация некоторых современных серверов не разрешает передачу файлов с данными о зонах, поэтому, возможно, придется запрашивать конкретные записи |
| A | Возвращает все записи "A". Записи "A" являются именами отдельных хостов в сети, такими как webserver.example.com или firewall.example.com |
| MX | Возвращает зарегистрированное имя почтового хоста домена. Это полезно, если вы хотите вступить в контакт с администратором (попробуйте administrator@mailhost.example.com или root@mailhost.example.com) |
| CNAME | Возвращает все синонимы. Например, fido.example.com = www.example.com |
| ANY | Возвращает всю информацию, которую можно получить о домене. Иногда это работает, когда не проходит AXFR |

Finger: служба информации о пользователях

Finger

Автор/основной контакт: Неизвестен

Web-сайт: Различные, включая
<http://www.infonet.st-johns.nf.ca/adm/finger.html>
<http://www.developer.com/net/cplus/article.php/627661>

Платформы: Большинство платформ UNIX и Windows

Лицензии: Различные

Справочная информация в UNIX: Наберите `man finger` в командной строке.

Finger - старая команда UNIX, вышедшая из активного употребления, но все еще запускаемая на многих машинах как унаследованная служба. Она была разработана, когда Интернет был более дружелюбным местом, и пользователей не беспокоило, что люди на другом конце Земли узнают их расписание, номера кабинетов и другую информацию. Большинство знающих системных администраторов сейчас отключают демон `finger`, так как он ассоциируется с множеством дыр в безопасности. Однако вы будете удивлены, узнав, на скольких серверах он все еще запускается. Многие маршрутизаторы поставляются с ним (я не могу понять, почему; разве что поставщик реализовал включающий его стек TCP), и некоторые операционные системы семейства UNIX все еще по умолчанию включают его при установке, а люди забывают или не знают, как его отключить.

Команда `finger` позволяет запросить информацию о пользователях удаленной системы. Синтаксис команды таков:

```
finger имя_пользователя@имя_хоста
```

Замените аргумент `имя_пользователя` на имя пользователя, которым интересуетесь, а `имя_хоста` - на полностью заданное имя хоста, например, `host.example.com`. Можно также использовать IP-адрес. На [листинге 2.5](#) показаны результаты выполнения запроса `finger` для пользователя `bsmith` на хосте `server1.example.com`.

```
Login name: bsmith In real life: Bob Smith
Directory: /home/bsmith Shell: /bin/bash
Last Login: 7/03/04 0800:02
No unread mail
Project: Написание книги

Plan: Буду в отпуске в Европе с 1 по 15 сентября.
```

Листинг 2.5. Результаты запроса `finger`

Видно, что с помощью `finger` можно получить немало информации о Бобе, включая время последнего входа в систему, наличие непрочитанной почты и введенную им личную информацию. Он также любезно сообщил нам, когда его не будет в офисе. Все это может использоваться хакерами для составления портрета Боба как объекта морально-психологического воздействия, может помочь узнать обычное время входа в систему и расписание работы, чтобы попытаться взломать его системный счет, когда он в отъезде.

Еще одно хитрое применение `finger` состоит в выполнении команды без имени пользователя. При этом генерируется список всех пользователей, находящихся в данное время в системе. На [листинге 2.6](#) показаны результаты того, как могла бы выглядеть выдача подобной команды в вымышленном домене `example.com`. Можно видеть, кто находится в системе, какие у них реальные имена, бездействуют ли они (возможно, они забыли выйти из системы) и если да, то как долго. Наконец, указана рабочая станция, с которой они вошли (являются ли они локальными или удаленными пользователями), и, для удаленных пользователей, имя хоста или IP-адрес их рабочей станции. Видно, что один пользователь вошел в систему несколько раз, с одним неактивным сеансом, который злонамеренный наблюдатель этих данных может попытаться перехватить.

Изучив выдачу, можно выполнить полный запрос `finger` для любого из пользователей, подходящих для дальнейшей разработки. Команда `finger -l @hostname.example.com` генерирует полный запрос `finger` для всех пользователей, находящихся в данный момент в системе.

```
[hostname.example.com]
User      Real Name  What  Idle  TTY Host Console Location
bsmith    Bob Smith          2     lab1-30 (cs.example.edu)
ajohnson  Andrew Johnson    2     lab1-10 (dialup.generucisp.com)
bjones    Becky Jones       co     lab3-22
atanner   Allen H Tanner    0:50  co lab3-9
atanner   Allen H Tanner     co     lab3-1
atanner   Allen H Tanner    4:20  co lab3-8
cgarcia   Charles Garsia     3     lab1-10
```

ps: команда опроса процессов UNIX**ps**

Автор/основной контакт: Неизвестен

Web-сайты: Различные, включая

<http://www.nevis.columbia.edu/cgi-bin/man/sh?man=ps>

Платформы: Большинство платформ UNIX

Лицензии: Различные

Справочная информация в UNIX: Наберите `man ps` в командной строке.

Команда `ps` (сокращение от `process` (процесс)), показывает все процессы, выполняющиеся в системе. С ее помощью можно выяснить, нет ли среди выполняющихся демонов или процессов тех, которых не должно быть. Ее можно также использовать для отладки многих средств из этой книги. В [табл. 2.3](#) перечислены некоторые полезные ключи `ps`.

Таблица 2.3. Ключи `ps`

| Ключ | Описание |
|------|--|
| A | Показывает процессы всех пользователей |
| a | Показывает все процессы пользователей, имеющие терминалы |
| u | Показывает имя пользователя процесса |
| x | Выводит процессы без управляющих терминалов |

На [листинге 2.7](#) приведена выдача команды `ps` с ключами `-aux`.

```

USER      PID  %CPU  %MEM  VSZ   RSS TTY  STAT  START  TIME  COMMAND
root         1   0.1   0.7 1288   484  ?    S   18:00  0:04  init [3]
root         2   0.0   0.0    0     0  ?    SW   18:00  0:00  [keventd]
root         3   0.0   0.0    0     0  ?    SW   18:00  0:00  [kapmd]
root         5   0.0   0.0    0     0  ?    SW   18:00  0:00  [kswapd]
root         6   0.0   0.0    0     0  ?    SW   18:00  0:00  [bdf flush]
root         7   0.0   0.0    0     0  ?    SW   18:00  0:00  [kupdated]
root         8   0.0   0.0    0     0  ?    SW<  18:00  0:00  [mdrecoveryd]
root        12   0.0   0.0    0     0  ?    SW   18:00  0:00  [kjournald]
root       137   0.0   0.0    0     0  ?    SW   18:00  0:00  [khubd]
root       682   0.0   1.0 1412   660  ?    S   18:01  0:00  /sbin/cardmg
rpc        700   0.0   0.8 1416   532  ?    S   18:01  0:00  portmap
root       720   0.0   1.2 1640   788  ?    S   18:01  0:00  syslogd -m 0
root       757   0.0   1.8 1940  1148  ?    S   18:01  0:00  klogd -2
root       797   0.0   0.8 1336   500  ?    S   18:01  0:00  gpm -t ps/2 -m
xfs        869   0.0   5.8 5048  3608  ?    S   18:01  0:00  xfs -port -l
daemon    884   0.0   0.8 1312   504  ?    S   18:01  0:00  /usr/sbin/atd
root      928   0.0   2.0 2660  1244  ?    S   18:01  0:01  /usr/sbin/SSHd
root      949   0.0   1.5 2068   948  ?    S   18:01  0:00  xinetd -stayalive

```

```

root    951    0.0    0.7 1292   496    ?      S 18:01 0:00 /sbin/dhccpd -h m
root   1078    0.0    1.0 1492   628    ?      S 18:01 0:00 crond
root   1132    0.0    3.4 3808  2152    ?      S 18:01 0:02 nessusd: waiting
root   1134    0.0    1.9 2276  1224    ?      S 18:01 0:00 login -- tony
tony   1494    0.0    2.6 2732  1624  tty1    S 18:29 0:00 -bash
tony   1430    0.0    2.6 2744  1636  tty1    S 18:29 0:00 bash
tony   1805    0.0    1.2 2676   796  tty1    R 18:56 0:00 ps -aux

```

Листинг 2.7. Выдача `ps -aux`

На выдаче можно видеть каждый процесс, выполняющийся в системе, вместе с его идентификатором (PID). Это важно, если вы хотите терминировать службу или произвести какие-то иные действия. По ключу `-u` в левом столбце выводится имя пользователя процесса. На выдаче показаны различные системные процессы, принадлежащие пользователю `root`. Виден также пользователь, запустивший команду `ps`. Если вы обнаружите, что выполняется некая загадочная служба, то ситуацию необходимо исследовать дополнительно. На выдаче присутствует служба, которая может показаться подозрительной: демон `nessusd` - сканер уязвимостей, рассматриваемый в [лекции 5](#). Однако это ваша система защитных средств, поэтому его выполнение здесь вполне законно.

Можно построить конвейер из команд `ps` и `grep`, чтобы найти определенные работающие службы. Например, команда

```
ps -ax | grep snort
```

сообщит, запущен ли в вашей системе Snort, и какой у него идентификатор процесса (PID). Как и многие другие средства уровня операционной системы, команда `ps` полезна для всех видов деятельности по системному администрированию, а не только для обеспечения безопасности.

Клиент OpenSSH: служба защищенного терминала

Клиент OpenSSH

Автор/основной контакт: Tatu Ylonen

Web-сайт: <http://www.openssh.com/>

Платформы: Большинство платформ UNIX, Windows, OS/2

Лицензия: BSD

Другие web-сайты:

<http://www.uni-karlsruhe.de/~ig25/SSH-faq/>

<http://www.ssh.com/>

<http://kimmo.suominen.com/SSH/>

SSH - настолько полезное средство, что его серверной стороне посвящен целый раздел в [лекции 9](#). Однако я настоятельно рекомендую по возможности применять клиент SSH для интерактивного входа в систему вместо протокола Telnet или другого небезопасного метода. Вы будете использовать его так часто, что я хотел бы представить здесь некоторые основные моменты и синтаксис клиента. SSH (Secure Shell - защищенный командный интерпретатор) является средством удаленного доступа, который позволяет входить в удаленную систему защищенным образом. Ахиллесова пята большинства сетей состоит в том, что межсистемные коммуникации осуществляются в открытом виде. Можно как угодно повышать безопасность отдельных систем, но если удаленный вход в них производится с помощью небезопасной терминальной программы, злоумышленники все равно смогут перехватить ваши входные

атрибуты, используя сетевой анализатор, и с их помощью без всяких усилий входит в систему. Одно из наиболее популярных средств удаленного доступа - Telnet - страдает этим недостатком. SSH решает данную проблему, шифруя все коммуникации с первого нажатия клавиши.

SSH - программа с открытыми исходными текстами, доступная почти на любой платформе. Она поставляется по умолчанию с большинством операционных систем на основе Linux. На web-сайте <http://www.ssh.com/> имеется коммерческая версия, также с открытыми исходными текстами. Мы рассматриваем OpenSSH - свободную версию, присутствующую в большинстве дистрибутивов Linux и на прилагаемом к книге компакт-диске. Между версиями имеются небольшие различия, однако они взаимно совместимы, а большинство команд и синтаксических конструкций должны работать и там, и там.

Чтобы получить доступ к удаленной системе с помощью SSH, требуется клиент SSH на вашей стороне, а на удаленной должен быть запущен сервер SSH. Хотя SSH не так широко распространен, как Telnet, он становится все более популярным. Компания Cisco, наконец, стала устанавливать SSH на своих маршрутизаторах, хотя все еще оставляет по умолчанию включенным сервер Telnet, в то время как SSH остается необязательным.

SSH выпускается с лицензией открытых исходных текстов, которая по сути аналогична лицензии BSD. Убедитесь, что используете версию 3.6 или выше; некоторые более ранние версии имеют дефекты в реализации криптографических протоколов и поддаются взлому. На самом деле, стоит убедиться, что вы располагаете самой свежей из доступных версий, так как код постоянно улучшается, а алгоритмы совершенствуются.

SSH, помимо защищенного входа в удаленную систему, имеет ряд по-настоящему интересных применений. Его можно применять для туннелирования почти всех сервисов через зашифрованный канал между серверами (это приложение более детально обсуждается в последующих лекциях). Базовый синтаксис SSH для удаленного входа в систему таков:

```
ssh -l входное_имя имя_хоста
```

Замените входное_имя вашим входным именем на удаленной системе имя_хоста. Можно также использовать конструкцию

```
ssh входное_имя@имя_хоста
```

Поэтому, чтобы войти в систему web-сервера с именем web.example.com, используя мое имя tony, я набираю

```
ssh tony@web.example.com
```

Годится и `ssh -l tony web.example.com`. Если просто ввести `ssh web.example.com`, то сервер будет подразумевать то же имя пользователя, что и в вашей системе.

В [табл. 2.4](#) перечислены некоторые другие опции SSH.

Таблица 2.4. Дополнительные опции SSH

| Опция | Описание |
|----------------|--|
| -с протокол | Используйте конкретный криптографический протокол. Замените протокол на blowfish, 3des или des, в зависимости от криптографического алгоритма, который хотите применять. Отметим, что ваша версия SSH должна поддерживать эти алгоритмы |
| -р номер_порта | Соединиться с заданным номером порта, а не с используемым SSH по умолчанию портом 22 |
| -Р номер_порта | Использовать заданный порт, который не является частью стандартного списка собственных портов (обычно - порт с номером больше 1024). Это может быть полезно, если имеется межсетевой экран, который закрывает для коммуникаций порты с младшими номерами |
| -v | Подробный вывод. Полезно при отладке |
| -q | Молчаливый режим, противоположный |
| -С | Производить сжатие зашифрованного трафика. Может быть полезно для очень медленных соединений, таких как коммутируемые, но лучше иметь при этом мощный процессор для выполнения сжатия, иначе в итоге коммуникации замедлятся |
| -1 | Использовать только протокол SSH версии 1. Не рекомендуется по причинам, указанным в описании опции -С, но может потребоваться, если сервер, с которым вы соединяетесь, не модернизирован до версии 2 |

Особенности повышения безопасности Windows

Хотя это и не является основной темой книги, но при использовании систем Windows важно защитить их насколько возможно, чтобы можно было сформировать обсуждавшуюся выше надежную вычислительную базу. Известно, что в Windows обычно запускаются сетевые службы всех видов. Некоторые поставщики ПК с Windows даже загружают на них небольшие web-серверы, чтобы их персонал технической поддержки мог "войти" и интерактивно помочь вам, если вы к ним обратились. Излишне упоминать, что это в высшей степени небезопасно, и для множества таких "полезных мелочей" опубликованы способы вторжения. Многие и не подозревают обо всех этих программах, выполняющихся в фоновом режиме.

Если вы используете одну из современных версий Windows (NT, 2000 или XP), то одна вещь, которую вы можете сделать, - зайти в окно Services в разделе Administrative Tools меню Control Panel. Будет выведен список всех процессов, выполняющихся на компьютере (аналогично команде `ps` в UNIX). Можно просмотреть этот список и увидеть все небольшие программы, которые Windows любезно запускает для вас. Большинство из них - службы, требующиеся для нормальной работы Windows. Однако некоторые из них вам не нужны и просто отнимают процессорное время, замедляя компьютер и, возможно, создавая дыры в безопасности. Можно отключить их, щелкая на службе мышью и выбирая Stop. Не забудьте также задать тип запуска Manual или Disabled, иначе они будут снова запущены при перезагрузке системы.



Флэми Тех советует:

Убедитесь, что вы знаете, что отключаете!

Нужно быть очень осторожным при отключении подобных служб. Если вы не знаете точно, что делает служба, и не уверены, что она вам не нужна, то не трогайте ее. Многие процессы зависят от других, и если отключать их произвольным образом, это может нарушить нормальное функционирование системы.

Существует несколько прекрасных руководств, разработанных Агентством национальной безопасности США (<http://www.nsa.gov/>), для безопасного конфигурирования операционных систем Windows. Их можно найти по адресу <http://nsa1.www.conxion.com/index.html>.

Центр Безопасности Интернета (<http://www.cisecurity.org/>) также публикует средства измерения и оценки безопасности для Windows 2000 и NT. Они могут помочь безопасному конфигурированию машин Windows.

Во многих книгах и Интернет-ресурсах данная тема рассматривается более глубоко. Можно также использовать некоторые из описанных далее средств, например, сканеры портов и сканеры уязвимостей, для сканирования и повышения безопасности систем Windows. Как бы вы это ни сделали, убедитесь, что вы укрепили свою систему, прежде чем начинать на ней установку защитных средств.

Хотя в Windows присутствуют некоторые средства сетевой диагностики и опроса, аналогичные имеющимся в UNIX, такие как `ping` и `tracert`, эта система не предоставляет в готовом к употреблению виде ряд других служб, таких как `whois` и `dig`. Существует, однако, дополнительное защитное средство, Sam Spade for Windows, которое добавляет эти функции в систему Windows и улучшает имеющиеся.

Sam Spade for Windows: средство опроса сети для Windows

Sam Spade for Windows

Автор/основной контакт: Steve Atkins

Web-сайт: <http://www.samspade.org/>

Платформы: Windows 95, 98, ME, NT, 2000, XP

Рассмотренная версия: 1.14

Лицензия: GPL

Другие ресурсы:

См. справочный файл, включенный в комплект поставки

Этот прекрасный "швейцарский армейский нож" для машин Windows восполняет недостаток имеющихся в ОС Windows сетевых средств. Системные администраторы UNIX могут больше не злорадствовать над своими Windows-коллегами, у которых нет таких отточенных инструментов, как `dig`, `whois` и другие. На самом деле Sam Spade for Windows добавляет даже несколько таких средств, которых в UNIX нет. Это бесценный инструмент для получения информации о сети. Подобно одноименному персонажу-детективу, Sam Spade может узнать о сети почти все.

Установка и применение Sam Spade for Windows

Начните с посещения web-сайта Samspade.com и загрузки программы или возьмите ее с приложенного к книге компакт-диска. Затем просто дважды щелкните мышью на файле и позвольте программе установки позаботиться обо всем остальном. После установки Sam Spade запустите программу, и вы получите экран главной консоли ([рис. 2.1](#)).

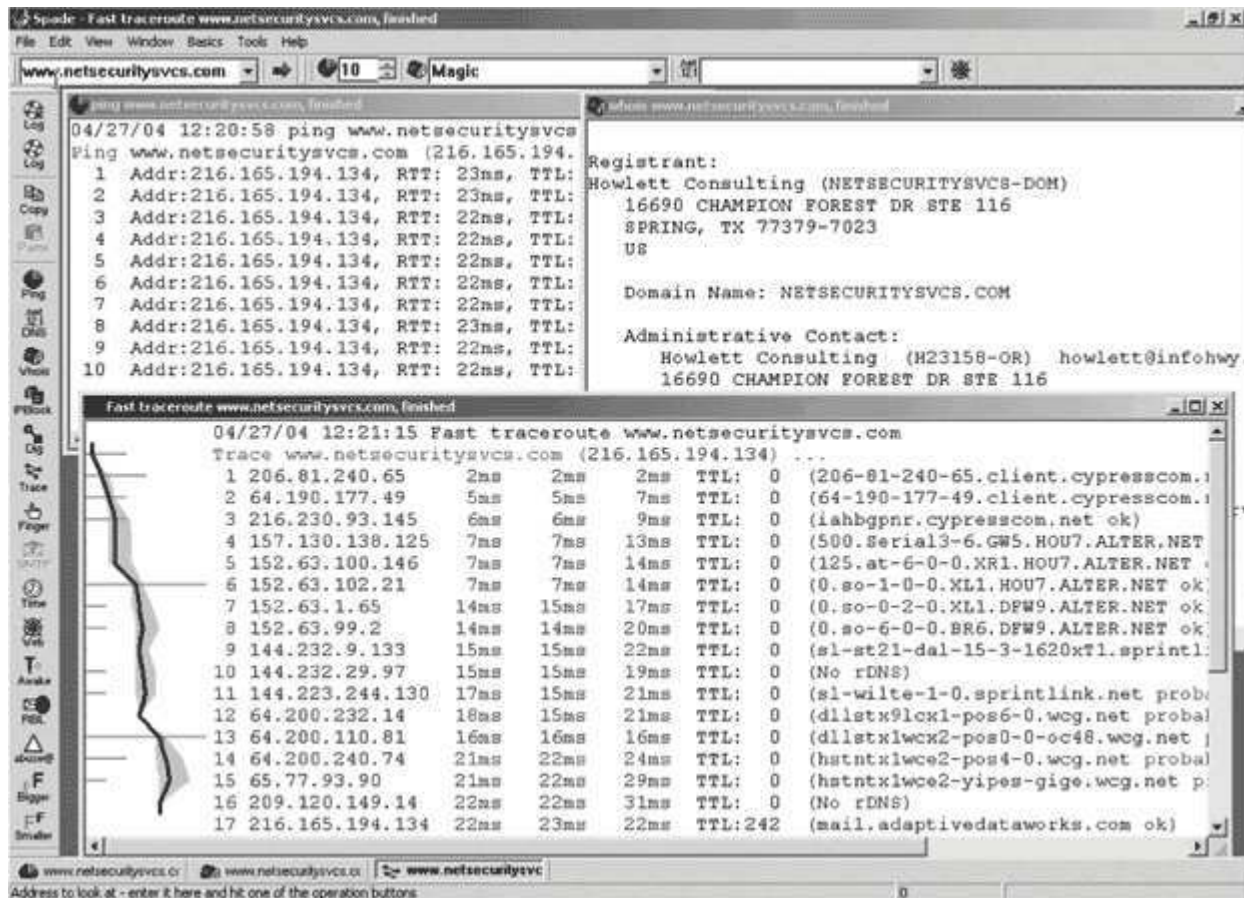


Рис. 2.1. Основной экран Sam Spade

Интерфейс Sam Spade несложен. В верхнем левом поле вы вводите IP-адрес или имя хоста, который хотите проверить, а затем щелкаете мышью на иконках, расположенных ниже, чтобы выполнить различные тесты для этой целевой системы. Каждый тест выполняется в собственном окне, а весь вывод запоминается в журнальном файле, который можно сохранить для дальнейшего использования и документирования. В меню Options следует задать подразумеваемый сервер имен, чтобы работали все тесты, использующие DNS. Можно также ввести этот адрес в панели меню с правого края.



Флэми Тех советует:

Будьте ответственным детективом!

Запуск Sam Spade в собственной сети, или в сети, за которую вы отвечаете, не вызывает никаких вопросов. Однако будьте очень осторожны при применении этих средств для сетей вне вашего контроля. Хотя большинство выполняемых тестов - мягкие, некоторые могут создать значительную нагрузку на сервер или пробудить мониторы вторжений. Поэтому не забудьте получить разрешение, прежде чем применять эти средства во внешних сетях. Это не только придаст вашим действиям видимость законности - это еще и просто признак хороших манер. Ведь вы не хотите, чтобы другие системные администраторы применяли Sam Spade в вашей сети без вашего разрешения?

В [табл. 2.5](#) перечислены основные функции Sam Spade и дано их описание.

В [табл. 2.6](#) перечислены другие полезные тесты, расположенные в меню Tools.

Таблица 2.5. Основные функции Sam Spade

| Функция | Описание |
|----------|--|
| Ping | То же самое, что и встроенная функция <code>ping</code> в Windows и UNIX, но позволяет легко задавать число повторений запроса <code>ping</code> и выводит несколько более подробную выдачу |
| Nslookup | Аналог одноименной команды UNIX |
| Whois | Аналог одноименной команды UNIX |
| IPBlock | Эта команда проверяет в базе данных ARIN IP-адрес или набор IP-адресов и выдает о них некоторую полезную информацию, включая организацию, которой принадлежат IP-адреса, где они были выделены поставщиком Интернет-услуг, и различную контактную информацию, в том числе контакты для сообщений о ненадлежащем поведении, если таковое регистрируется (см. пример вывода на рис. 2.2). |
| Trace | Аналог команды <code>tracert</code> , однако генерируется дополнительная информация, такая как обратные DNS-записи и графическое отображение задержек между межсетевыми переходами. |
| Finger | Аналог команды <code>finger</code> в UNIX. |
| Time | Проверяет часы на удаленной системе. Полезно для синхронизации часов на серверах. |

Таблица 2.6. Тесты меню Tools в Sam Spade

| Тест | Описание |
|----------------|---|
| Blacklist | Проверяет, фигурирует ли ваш почтовый сервер в каком-либо из черных списков электронной почты (в базах данных, которые содержат адреса известных спамеров). Если ваш адрес каким-либо образом там оказался (например, сервер был открыт для пересылки почты), то некоторые адресаты могут не получать от вас почту |
| Abuse | Ищет официальный контакт для жалоб на ненадлежащее поведение для набора IP-адресов, чтобы вы могли подать жалобу, если у вас есть проблемы с одним из этих адресов |
| Scan Addresses | Выполняет базовое сканирование портов диапазона адресов. Этот очень простой сканер портов выявляет открытые сетевые порты. Если требуется просканировать адреса, то лучше использовать один из полнофункциональных сканеров портов, рассмотренных в лекции 4 . Помните также, что сканирование портов может рассматриваться владельцами внешних сетей как враждебная деятельность |
| Crawl website | "Утюжит" Web-сайт, выявляя все ссылки, страницы и любые другие формы или файлы, до которых можно добраться. Это полезно при поиске всех страниц, на которые ссылается web-сайт, и для выявления файлов, о которых вы не знали |

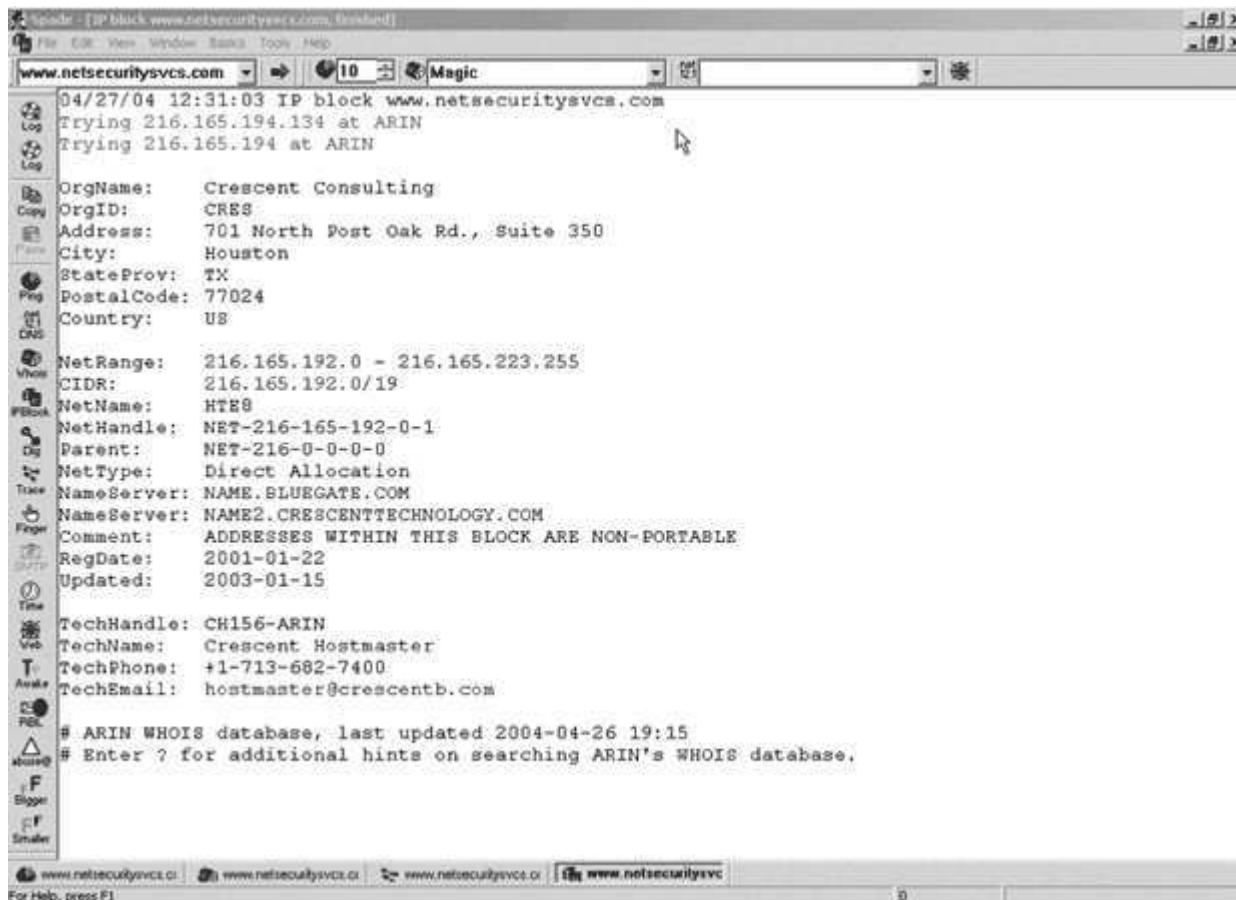


Рис. 2.2. Выдача Sam Spade IPBlock

Имеется несколько других средств, которые не являются темой данной книги, например, ликвидаторы отмеченных сообщений USENET и декодировщики URL, которые могут оказаться полезными, если вы разрабатываете Web-сайт. Sam Spade дает средства для исследования сети, аналогичные имеющимся в UNIX. Следующий инструмент, PuTTY, предоставляет возможности SSH, другой UNIX-программы, для безопасного удаленного терминального доступа в Windows

PuTTY: Клиент SSH для Windows

PuTTY

Автор/основной контакт: Sam Tatham

Web-сайт: <http://www.chiokr.greenend.org.uk/~sgtatham/putty>

Платформы: Windows 95, 98, ME, NT, 2000, XP

Рассмотренная версия 54b

Лицензия: MIT (аналогична лицензии BSD)

Другие ресурсы:

Файл справок или web-сайт.

В ближайшее время Microsoft собирается заняться этой программой и начнет поставлять встроенный клиент SSH вместе с Windows. Пока же PuTTY является отличным клиентом SSH для Windows, он также включает усовершенствованный, поддерживающий шифрование клиент Telnet. PuTTY можно использовать для защищенных коммуникаций с любым сервером, поддерживающим протокол SSH

Установка и запуск PuTTY

Загрузите файл с Web-сайта или возьмите его с приложенного к книге компакт-диска и установите, сделав на нем двойной щелчок мышью. PuTTY имеет приятный, ясный интерфейс и способен эмулировать практически любой терминал. Можно задать номер порта для входа, если сервер SSH использует нестандартный номер порта. Можно также поэкспериментировать со всеми настройками, используя меню слева.

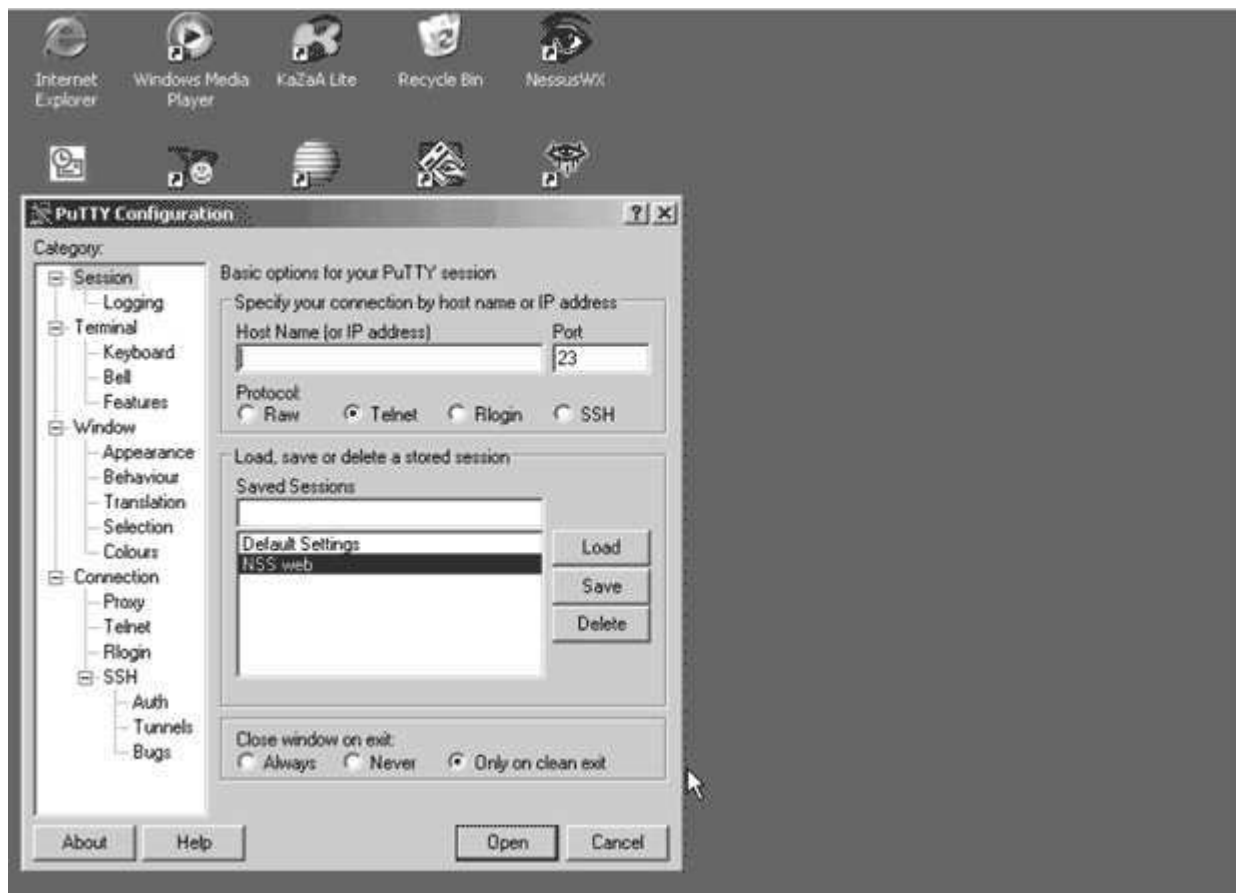


Рис. 2.3. Основной экран PuTTY

Протоколы сеансов можно записывать в текстовые файлы, что может быть весьма полезно (я использовал PuTTY для сохранения листингов всех терминальных сеансов в этой книге). Можно также до бесконечности менять конфигурацию, в том числе, набор допустимых протоколов шифрования. Вас даже будут предупреждать при попытке соединиться с сервером SSH, использующим одну из слабых версий SSH, которая может быть уязвима для взлома.

При соединении с сервером в первый раз PuTTY предупредит, что он добавляет в базу данных идентификационную метку и ключ этого сервера. Это нормально - просто проверьте, выглядит ли сертификат подходящим, примите его, и он больше не будет появляться при последующих соединениях с этим сервером.

Инструменты безопасности с открытым исходным кодом

3. Лекция: Межсетевые экраны: версия для печати и PDA

Теперь, когда у вас есть достаточно безопасная операционная система и вы освоили несколько основных приемов, перейдем к использованию некоторых более сложных защитных средств. В этой лекции описано, как настраивать и обслуживать безопасный межсетевой экран с открытыми исходными текстами. Если у вас уже есть межсетевые экраны, можно все равно прочитать данную лекцию, чтобы освежить в памяти или узнать, как они действуют. Это пригодится при изучении следующих лекций, где обсуждаются сканеры портов и уязвимостей.

Межсетевой экран - это устройство, являющееся первым рубежом передовой линии обороны против всех входящих атак или ненадлежащего использования вашей сети. Межсетевой экран может отразить или смягчить многие виды атак и экранировать (заслонить) внутренние серверы и рабочие станции от Интернета. Межсетевой экран способен также предотвратить доступ извне к машинам внутренних ЛВС. При растущем использовании случайных сканеров и автоматических червей и вирусов, экранирование внутренних машин от Интернета важно как никогда. Правильно сконфигурированный межсетевой экран существенно продвинет вас по пути защиты от внешних атак. (Защита от внутренних атак - совершенно другая проблема, которая рассматривается в лекциях с 4 по 7).

Обзор лекции

Изучаемые концепции:

- Основные понятия сетей TCP/IP
- Как работают межсетевые экраны
- Философия конфигурирования межсетевых экранов
- Бизнес-процессы для межсетевых экранов
- Примеры конфигураций межсетевых экранов

Используемые средства:

`Iptables`, `Turtle Firewall`, `SmoothWall`

В наше время мало кто сомневается, что межсетевой экран является обязательным компонентом любой инфраструктуры безопасности. Доступно множество жизнеспособных коммерческих альтернатив: Cisco, NetScreen, SonicWALL, Checkpoint - это лишь небольшая часть поставщиков высококлассных коммерческих решений, ориентированных на большие корпоративные сети с интенсивными потоками данных.

Компании Linksys (принадлежащая теперь Cisco), D-Link, NETGEAR предлагают младшие модели межсетевых экранов потребительского уровня. Как правило, подобные устройства не обладают высокой конфигурируемостью и расширяемостью: обычно они действуют как пакетные фильтры, блокируя входящие соединения и осуществляя динамическую трансляцию сетевых адресов. Они предназначены для кабельных и DSL-соединений и могут не выдержать более высокой нагрузки.

Старшие модели межсетевых экранов сделают практически все, что вы от них захотите, но это потребует денежных затрат - как минимум, нескольких тысяч долларов. Для их настройки зачастую требуется изучение нового синтаксиса или интерфейса. Некоторые из более новых моделей, такие как SonicWALL и NetScreen, поставляются с интерфейсом конфигурации на основе Web, но это обычно достигается за счет меньшей глубины конфигурационных опций.

Малоизвестный и редко афишируемый секрет некоторых коммерческих межсетевых экранов состоит в том, что в их основе лежит программное обеспечение с открытыми исходными текстами. На самом деле вы платите за высококачественную коробку и линию технической поддержки. Это может

быть оправданным для организаций, которые нуждаются в дополнительной поддержке. Однако, если вы готовы изучать еще один интерфейс, и если в коммерческом продукте используются те же технологии, которые доступны бесплатно, почему бы не создать свой собственный межсетевой экран с помощью средств с открытыми исходными текстами, представленных в этой книге, и сохранить своей фирме тысячи долларов? Даже если вы не собираетесь выбрасывать свой коммерческий межсетевой экран, лучшее понимание его работы и того, что происходит за сценой, поможет сделать его конфигурацию более безопасной.

Прежде чем мы погрузимся в инструментарий, я хочу рассмотреть основы функционирования межсетевых экранов и обработку ими различных сетевых протоколов для ограничения доступа к сети. Даже если вы не планируете использовать программное обеспечение с открытыми исходными текстами для своего межсетевого экрана, полезно знать немного больше о том, что в действительности происходит внутри этого черного ящика.

Основы архитектуры сетей

Прежде чем вы придете к подлинному пониманию сетевой безопасности, необходимо понять архитектуру сетей. Хотя эта книга не претендует на роль начального курса по сетям, в данном разделе приведен краткий обзор сетевых концепций и терминов. Я буду часто ссылаться на эти термины, и знакомство с ними поможет вам понять основы протокола TCP/IP. Если вы хорошо знакомы с сетевыми топологиями, то можете пропустить этот раздел и сразу перейти к инструментарию.

Как вы, вероятно, знаете, конструкцию каждой сети можно разделить на семь логических частей, каждая из которых решает определенную часть коммуникационной задачи. Эта семиуровневая конструкция называется Эталонной моделью взаимосвязи открытых систем (ВОС). Она была разработана Международной организацией по стандартизации (ISO) для представления логической модели описания сетевых коммуникаций, и она помогает поставщикам стандартизовать оборудование и программное обеспечение. В [табл. 3.1](#) проиллюстрирована эталонная модель ВОС и приведены примеры каждого уровня.

Таблица 3.1. Эталонная модель ВОС

| Номер уровня модели ВОС | Название уровня | Примеры протоколов |
|-------------------------|-----------------------|--|
| Уровень 7 | Прикладной уровень | DNS, FTP, HTTP, SMTP, SNMP, Telnet |
| Уровень 6 | Уровень представления | XDR |
| Уровень 5 | Уровень сеанса | RPC |
| Уровень 4 | Транспортный уровень | NetBIOS, TCP, UDP |
| Уровень 3 | Сетевой уровень | ARP, IP, IPX, OSPF |
| Уровень 2 | Канальный уровень | Arcnet, Ethernet, Token ring |
| Уровень 1 | Физический уровень | Коаксиальный кабель, оптоволокно, витая пара |

Физический уровень

Этот уровень представляет реальную физическую среду передачи данных. Для различных типов среды применяются разные стандарты. Например, коаксиальный кабель, незранированная витая пара и волоконно-оптический кабель предназначены для различных целей: коаксиальный кабель используется в более старых ЛВС, а также для подключения к Интернету через сети кабельного ТВ, витая пара - для внутренней кабельной разводки, в то время как оптоволокно обычно применяют для протяженных соединений с высокой пропускной способностью.

Канальный уровень

Этот уровень относится к различным частям оборудования сетевых интерфейсов. Он помогает кодировать данные и помещать их в физическую среду передачи. Он также позволяет устройствам идентифицировать друг друга при попытке взаимодействия с другим узлом. Примером адреса канального уровня служит MAC-адрес сетевой платы. (MAC не имеет никакого отношения к компьютерам компании Apple, это сокращение от Medium Access Control - управление доступом к среде передачи. MAC-адрес является числом, которое уникальным образом идентифицирует плату компьютера в сети.) В сетях Ethernet по MAC-адресу можно находить компьютер. В 1970-80-х годах корпорации использовали много различных типов стандартов канального уровня, определенных по большей части их поставщиками оборудования. Компания IBM использовала Token Ring для своих сетей ПК и SNA для большей части

больших машин; компания DEC применяла иной стандарт, а Apple - еще один. В наше время большинство организаций используют Ethernet, так как он широко распространен и недорог.

Сетевой уровень

Этот уровень является первой частью, которую вы действительно видите при взаимодействии с сетями TCP/IP. Сетевой уровень дает возможность взаимодействовать через различные физические сети с помощью вторичного уровня идентификации. В сетях TCP/IP для этого используется IP-адрес. IP-адрес на компьютере помогает осуществлять маршрутизацию данных при передаче из одного места в другое в сети и через Интернет. Этот адрес является уникальным числом для идентификации компьютера в IP-сети. В некоторых случаях это число уникально для компьютера; ни одна другая машина в Интернете не может иметь такой адрес. Это справедливо для обычных открыто маршрутизируемых IP-адресов. Во внутренних ЛВС машины часто используют блоки частных IP-адресов. Они зарезервированы только для внутреннего употребления и не предназначены для маршрутизации через Интернет. Эти номера не обязаны быть уникальными для различных сетей, но все равно должны быть уникальными в каждой ЛВС. В то время как два компьютера могут иметь один и тот же частный IP-адрес в различных внутренних сетях, они никогда не будут иметь один и тот же MAC-адрес, так как последний является серийным номером, присвоенным производителем сетевых плат. Существуют некоторые исключения (см. врезку "Следуйте за MAC"), но обычно MAC-адрес будет уникальным образом идентифицировать компьютер (или, по крайней мере, сетевой интерфейс этого компьютера).



Флэми Тех советует:

Следуйте за MAC!

MAC-адреса могут помочь справиться с рядом сетевых проблем. Хотя MAC-адрес не идентифицирует машину непосредственно по имени, все MAC-адреса присваиваются производителем и начинаются с особого префикса для каждого производителя, полный список которых можно найти на <http://www.macaddresses.com/>. Как правило, MAC-адреса также печатаются на самой плате.

С помощью одного из сетевых анализаторов, рассмотренных в [лекции 6](#), и используя MAC-адреса, зачастую можно проследить источник проблемного сетевого трафика. MAC-адреса обычно регистрируются серверами DHCP в Windows или межсетевыми экранами, поэтому можно сопоставить MAC-адреса с определенным IP-адресом или именем машины. Их можно использовать также для судебных доказательств - хакеры любят подделывать IP-адреса, но большинство из них не знает, как подделать MAC-адрес, и это позволяет уникальным образом идентифицировать их ПК.

Транспортный уровень

Этот уровень обеспечивает доставку пакета данных из точки А в точку В. На этом уровне располагаются протоколы TCP и UDP. TCP (Transmission Control Protocol - протокол управления передачей) по сути обеспечивает согласованность отсылки пакетов и их приема на другом конце. Он позволяет исправлять ошибки на уровне битов, повторно передавать потерянные сегменты и переупорядочивать фрагментированный трафик и пакеты. UDP (User Datagram Protocol - пользовательский дейтаграммный протокол) является менее тяжеловесной схемой, используемой для потоков мультимедийных данных и кратких взаимодействий с небольшими накладными расходами, такими как запросы DNS. Этот протокол также осуществляет обнаружение ошибок и мультиплексирование данных, но не предоставляет никаких средств для переупорядочивания данных или их гарантированной доставки. Большинство межсетевых экранов оперируют на транспортном и сетевом уровнях.

Уровень сеанса

Уровень сеанса обслуживает в основном установление соединения и его последующее закрытие. Иногда на этом уровне выполняется аутентификация, для того чтобы установить, кому разрешено участвовать в сеансе. Он используется в основном для определенных приложений, располагающихся на более высоких уровнях модели.

Уровень представления

Этот уровень обеспечивает определенное кодирование и декодирование, требующееся для представления данных в формате, понятном получателю. Некоторые формы шифрования могут рассматриваться как представление. Различие между прикладным уровнем и уровнем сеанса является тонким, и некоторые также считают, что прикладной уровень и уровень представления по сути совпадают.

Прикладной уровень

Заключительный уровень, на котором прикладные программы (FTP, HTTP, SMTP и т.п.) получают данные. На этом уровне в дело вступает некоторая программа, обрабатывающая реальные данные из пакетов. Этот уровень является головной болью профессионалов в области безопасности, так как именно на нем выявляется большинство уязвимостей.

Сети TCP/IP

Когда-то TCP/IP был малоизвестным сетевым протоколом, который использовали в основном правительственные и образовательные учреждения. На самом деле он был изобретен военным исследовательским агентством, DARPA, для обеспечения бесперебойной работы сетей. Преследовалась цель создания сети, способной выдержать отказ множества линий связи в случае катастрофического события, такого как ядерный удар. Традиционные способы передачи данных всегда полагались на одиночное прямое соединение, и если это соединение деградирует или его выводят из строя, то коммуникации нарушаются. В TCP/IP предложен способ "пакетирования" данных, позволяющий им находить собственный путь через сеть. Тем самым была создана первая отказоустойчивая сеть.

Однако большинство корпораций по-прежнему использовали сетевые протоколы, предоставляемые производителями оборудования. IBM продвигала NetBIOS или SNA; в ЛВС Novell использовался протокол IPX/SPX; в сетях Windows применялся еще один стандарт, NetBEUI, производный от NetBIOS. Хотя протокол TCP/IP стал широко использоваться в 1980-х годах, только с появлением Интернета в начале 1990-х TCP/IP превратился в стандарт передачи данных. Это привело к снижению цен на оборудование для IP-сетей, и также значительно облегчило межсетевое взаимодействие.

TCP/IP позволяет взаимодействующим узлам устанавливать соединение и затем проверять, когда передача данных начинается и завершается. В сети TCP/IP передаваемые данные разбиваются на фрагменты, называемые пакетами, и помещаются в последовательность "конвертов", каждый из которых содержит определенную информацию для следующего протокольного уровня. Пакеты помечаются 32-битными порядковыми номерами, чтобы даже в случае прихода в неправильном порядке передаваемые данные можно было собрать заново. Когда пакет пересекает различные части сети, каждый уровень открывается и интерпретируется, а затем оставшиеся данные передаются дальше согласно полученным инструкциям. Когда пакет данных прибывает в место назначения, реальные данные, или полезная нагрузка, доставляются приложению.

Говорят, аналогии обманчивы, но все-таки... Представьте себе, что вы отправляете в организацию письмо в конверте для доставки курьером. Транспортная компания использует внешний конверт для маршрутизации пакета в нужное здание. После получения пакет вскрывают и внешний конверт выбрасывают. Возможно, письмо направлено в другой внутренний почтовый ящик, поэтому его нужно вложить в межофисный конверт и переслать дальше. Наконец, письмо достигает своего получателя, который вскрывает все слои обертки и использует содержащиеся внутри данные. В [табл. 3.2](#) показано, как некоторые сетевые протоколы инкапсулируют данные.

Таблица 3.2. Пример пакета данных TCP/IP

| Протокол | Содержимое | Уровень модели ВОС |
|-------------------|----------------|--------------------|
| Ethernet | MAC-адрес | Канальный |
| IP | IP-адрес | Сетевой |
| TCP | Заголовок TCP | Транспортный |
| HTTP | Заголовок HTTP | Прикладной |
| Прикладные данные | Web-страница | Данные |

Можно видеть, что на внешнем "конверте" для наших данных написан адрес Ethernet. Он идентифицирует пакет в сети Ethernet. Внутри этого конверта находится сетевая информация, а именно, IP-адрес; еще глубже находится транспортный уровень, который устанавливает соединение и закрывает его.

Затем располагается прикладной уровень с заголовком HTTP, сообщаящим web-навигатору, как форматировать страницу. Наконец, мы доходим до реальной полезной нагрузки пакета - содержимого web-страницы. Этот пример иллюстрирует многоуровневую природу сетевых коммуникаций.

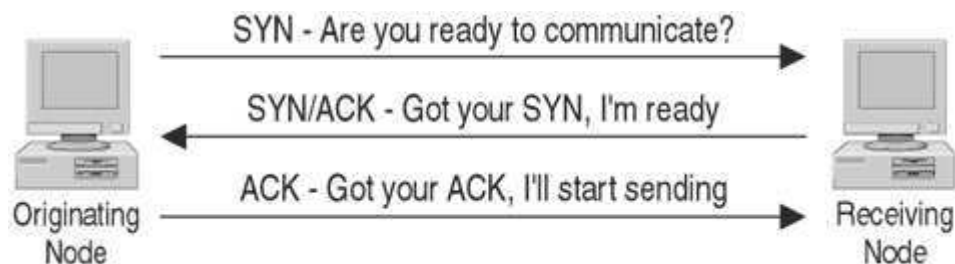


Рис. 3.1. Трехходовое квитирование установления связи

При использовании протокола TCP/IP взаимодействие между двумя узлами сети подразделяется на несколько фаз ([рис. 3.1](#)). Не вдаваясь в детали, касающиеся сервера доменных имен (DNS), и предполагая, что используются IP-адреса, а не имена хостов, в качестве первой фазы выделим порождение ARP-запроса (Address Resolution Protocol - протокол разрешения адресов) для поиска адреса Ethernet, соответствующего IP-адресу, с которым пытаются взаимодействовать. ARP преобразует IP-адрес в MAC-адрес сети Ethernet. Теперь, когда мы можем общаться с целевой машиной по протоколу IP, для формирования сеанса между машинами с помощью протокола TCP осуществляется трехходовое взаимодействие. Машина, собирающаяся послать данные другой машине, посылает пакет SYN для синхронизации или инициирования передачи. Пакет SYN, по сути, говорит: "Вы готовы к передаче данных?" Если другая машина готова принять соединение от первой, она посылает SYN/ACK, что означает: "Подтверждаю получение вашего пакета SYN, я готова." Наконец, порождающая машина отправляет пакет ACK обратно, говоря тем самым: "Отлично. Начинаю посылать данные." Такая процедура называется трехходовым квитированием установления связи в TCP. Если какой-либо из трех ходов не выполнится, то соединение не будет установлено. Осуществляя пересылку, машина снабжает пакеты данных порядковыми номерами и подтверждает получение всех пакетов с порядковыми номерами, использованными второй стороной. Когда все данные посланы, одна из сторон посылает второй стороне соединения пакет FIN. Та отвечает пакетом FIN/ACK и сама посылает пакет FIN, в ответ на который посылается последний пакет FIN/ACK для закрытия сеанса TCP/IP.

В силу способа, которым TCP/IP управляет иницированием и завершением сеанса, коммуникации TCP/IP можно назвать имеющими состояние, так как по пакетам можно определить, какая часть диалога имеет место. Это очень важно для межсетевых экранов, поскольку самым употребительным способом блокирования внешнего трафика является запрет пакетов SYN, направляемых извне на машины внутри сети. В результате внутренние машины могут общаться с внешним миром и инициировать соединения, но внешним машинам не удастся открыть сеанс. В работе межсетевых экранов имеется множество тонкостей, но по сути именно таким образом простые межсетевые экраны разрешают только однонаправленные соединения для web-навигации и аналогичных действий.

В Linux существует несколько встроенных экранирующих приложений: `Iptables` в версиях ядра 2.4x, `Ipchains` в 2.2x и `Ipfwadm` в ядре версии 2.0. Большинство межсетевых экранов на платформе Linux делают свое дело, используя одну из этих служебных программ уровня ядра.

Все три упомянутых приложения действуют аналогичным образом. У межсетевых экранов обычно имеется два или больше сетевых интерфейсов, и под Linux это достигается наличием в компьютере двух или большего количества сетевых плат. Один интерфейс обычно соединяется с внутренней ЛВС; этот интерфейс называется доверенным или собственным. Другой интерфейс предназначен для общедоступной стороны (ГВС). В большинстве небольших сетей ГВС-интерфейс подключен к Интернету. Может присутствовать и третий интерфейс, называемый ДМЗ (от военного термина ДеМилитаризованная Зона), обычно предназначенный для серверов, которые должны быть более открыты Интернету, чтобы внешние пользователи могли с ними соединяться. Каждый пакет, который пытается пройти через машину, пропускается через последовательность фильтров. Если он соответствует фильтру, над ним выполняется некоторое действие. Этим действием может быть отбрасывание пакета, пропуск пакета, или маскарад пакета ("Masq.") с помощью внутреннего собственного IP-адреса. Лучший метод конфигурирования межсетевых экранов состоит в первоначальном запрете всех пакетов с последующим выборочным разрешением необходимых потоков данных (см. врезку о философии конфигурирования межсетевых экранов).

Межсетевые экраны могут фильтровать пакеты на нескольких различных уровнях. Они могут анализировать IP-адреса и блокировать трафик, приходящий от определенных машин или сетей, проверять заголовок TCP и определять его состояние, и на более высоких уровнях анализировать приложение или номер порта TCP/UDP. Межсетевые экраны можно конфигурировать для отбрасывания целых категорий трафика, таких как ICMP. Пакеты типа ICMP, такие как `ping`, обычно отбрасываются межсетевыми экранами, поскольку они часто используются для исследования сети и атак на доступность. Нет причин, по

которым кому-то вне вашей организации должно быть позволено эхо-тестировать вашу сеть. Однако иногда разрешаются эхо-ответы, поэтому вы можете выполнять эхо-тестирование изнутри ЛВС вовне.

Бизнес-процессы безопасности

В некоторый момент, предпочтительно до того, как вы начнете загружать программное обеспечение, необходимо задокументировать бизнес-процессы своих межсетевых экранов. Это станет не только полезным средством планирования установки и конфигурирования, но может также помочь при необходимости обосновать перед руководством закупку оборудования или затраты рабочего времени. Документирование защитной деятельности позволит вам выглядеть более профессионально и подчеркнет пользу, которую вы приносите организации, что никогда не помешает. Это также упростит передачу эстафеты тому, кто придет вам на смену.

План документирует базовые процессы и процедуры, направленные на получение дивидендов от технологии. Установка межсетевого экрана - дело, конечно, хорошее, но без нужных процессов в нужном месте он рискует не обеспечить для организации обещанной безопасности. Следующие шаги описывают в общем виде бизнес-процесс установки и обслуживания межсетевого экрана.

1. Разработайте политику использования сети.

В руководстве пользователя по добропорядочному использованию компьютеров могут содержаться некоторые рекомендации, однако многие правила применения компьютеров намеренно расплывчаты и не определяют, какие приложения считаются ненадлежащими. Возможно, вам придется уточнить это у своего непосредственного начальника или у высшего руководства. Допускаются ли такие вещи, как программы мгновенного обмена сообщениями? Хотите ли вы следовать строгой политике выхода во внешний мир только посредством Web и электронной почты? Помните, что безопаснее написать правило для любого исключения, чем по умолчанию разрешить все виды деятельности. Критически важно получить ответы на эти вопросы, желательно в письменном виде, до того, как приступить к написанию правил для межсетевых экранов.

2. Составьте карту необходимых входящих и исходящих сервисов.

Если у вас еще нет карты сети, составьте ее сейчас. К каким портам каких серверов требуется обращаться извне? Есть ли пользователи, которым нужны специально открытые для них порты? (Совет: персоналу технической поддержки часто требуются FTP, Telnet и SSH.) Хотите ли вы создать демилитаризованную зону для общедоступных серверов или переадресовывать порты извне в ЛВС? Если у вас несколько сетевых сегментов или наборов общедоступных серверов, составление карты может занять больше времени, чем собственно настройка межсетевого экрана. Сейчас самый подходящий момент разобраться со всеми особыми запросами. Когда вы включите межсетевой экран и он остановит важное приложение, будет поздно.

3. Преобразуйте политику использования сети и требуемые сервисы в правила межсетевого экрана.

Именно теперь вы наконец приступаете к написанию правил для межсетевого экрана. Обратитесь к спискам допустимых внешних и требуемых внутренних сервисов, примите во внимание все исключения и создайте конфигурацию межсетевого экрана. Не забудьте использовать подход "запретить все", описанный во врезке, для отбрасывания всего, что не соответствует какому-либо из ваших правил.

4. Задействуйте и проверьте функциональность и безопасность.

Теперь можно включить межсетевой экран, откинуться на спинку кресла и ждать жалоб. Даже если ваши правила точно соответствуют политике, все равно найдутся люди, не понимающие, что использование Kazaa для загрузки фильмов противоречит политике организации. Будьте готовы проявить твердость, когда пользователи начнут просить о необоснованных исключениях. Каждое отверстие, которое вы открываете в межсетевом экране, является потенциальной угрозой безопасности.

Когда ваши пользователи будут удовлетворены работой межсетевого экрана, проверьте, что он блокирует то, что должен блокировать. С помощью двух инструментов, рассмотренных в книге далее, можно выполнить тестирование межсетевого экрана. Имеются в виду сканер портов извне и сетевой анализатор внутри. Они сообщат вам, какие пакеты проходят, а какие нет. Это полезно и для разрешения проблем в работе приложений, предположительно конфликтующих с межсетевым экраном.

5. Регулярно пересматривайте и тестируйте правила межсетевого экрана.

Прекрасная работа межсетевого экрана сегодня не означает, что так же будет завтра. Могут возникнуть новые угрозы, которые потребуют написания новых правил. Правила, которые добавлялись временно, для нужд некоторого проекта, могут слишком долго оставаться в конфигурации. Необходимо периодически пересматривать правила и сопоставлять их с текущими производственными потребностями и требованиями безопасности. В зависимости от размера и сложности конфигурации и частоты изменений, ревизия может быть ежегодной (для межсетевых экранов с небольшим числом правил - 20 и меньше) или ежемесячной (для очень сложных конфигураций). Каждый пересмотр должен включать реальные тесты с применением упомянутой выше комбинации сканер/анализатор, с использованием средств, описанных в [лекции 4](#), [5](#) и [6](#), для проверки того, что правила на самом деле работают так, как предполагалось.

Проектирование и применение подобного бизнес-процесса поможет вам получить значительно больше от установки межсетевого экрана, как в профессиональном, так и в техническом плане. Необходимо также разрабатывать бизнес-планы для других технологий, рассмотренных в этой книге, таких как сканирование уязвимостей и анализ работы сетей.



Флэми Тех советует:

"Запретить все!", когда речь идет о правилах межсетевого экрана!

Существует два метода настройки межсетевых экранов. Можно в качестве исходного принять положение "разрешить все" и затем задавать поведение, которое требуется блокировать, или же начать с положения "запретить все", после чего специфицировать то, что следует разрешить (допустимое поведение пользователей). Безусловно, предпочтительным является исходное положение "запретить все", поскольку при этом автоматически блокируются все потоки данных, если только они не разрешены явным образом. Подобный подход и более надежен, и более прост для поддержания безопасности.

Эта философия применяется в большинстве коммерческих межсетевых экранов. Лежащая в ее основе идея состоит в том, что если вам требуется определять, что считается плохим поведением, вы постоянно будете отставать, так как Интернет изменяется и развивается. Невозможно предсказать, какую форму может принять следующая, новая атака, поэтому вы будете уязвимы, пока она не будет опубликована, - только после этого вы сможете добавить новую строку в конфигурацию межсетевого экрана. Используя подход "запретить все", вы автоматически блокируете все, что не считается добропорядочной активностью.

Тип конфигурации "разрешить все" может иметь смысл в крайне либеральной среде, где накладные расходы на дополнительные разрешающие строки превышают ценность информации в сети (пример - некоммерческий или чисто информационный сайт). Но для большинства организаций подход "запретить все" более безопасен. Однако само по себе применение этого подхода не делает вашу сеть полностью безопасной. Атаки могут по-прежнему проходить через все проделанные вами отверстия, такие как доступ к web-серверу или электронной почте. Помните также, что даже при использовании подхода "запретить все" следует быть осторожным, чтобы не отменить его каким-либо правилом, разрешающим слишком многое и расположенным выше в вашей конфигурации.

Iptables: межсетевой экран с открытыми исходными текстами на платформе Linux

Iptables

Автор/основной контакт: Paul "Rusty" Russell

Web-сайт: <http://www.netfilter.org/>

Платформы: Большинство платформ Linux

Лицензия: GPL

Рассмотренная версия: 1.2.8

Ресурсы:

Списки рассылки Netfilter:

Netfilter-announce. Общий список извещений для новостей или новых выпусков и обновлений. Подписка по адресу:

<https://lists.netfilter.org/mailman/listinfo/netfilter-announce>

Netfilter-users. Общие вопросы использования Netfilter/Iptables. Сюда следует направлять общие дискуссионные заметки и вопросы. Подписка по адресу:

<https://lists.netfilter.org/mailman/listinfo/netfilter-users>

Netfilter-devel. Обсуждение разработки и участия. Подписка по адресу:

<https://lists.netfilter.org/mailman/listinfo/netfilter-devel>

В этом разделе описано, как конфигурировать межсетевой экран при помощи `Iptables` - утилиты экранирования/фильтрации пакетов, встроенной в большинство систем Linux с ядром версии 2.4 и выше. Данная утилита позволяет создать межсетевой экран, используя команды операционной системы. Она произошла от более ранних проектов межсетевых экранов в Linux. Первая система, `Ipfwadm`, позволяла создать простой набор правил пропускания или отбрасывания пакетов на основе определенных критериев. В ядро 2.2 ввели `Ipchains`, чтобы преодолеть ограничения `Ipfwadm`. Утилита `Ipchains` работала вполне приемлемо и обладала модульной архитектурой. Однако с ростом числа людей, требующих от своих межсетевых экранов выполнения многочисленных функций (например, сервера-посредника и устройства трансляции сетевых адресов), `Ipchains` также стало недостаточно. `Iptables` представляет собой обновленный вариант этих программ и допускает многочисленные применения, ставшие привычными для современных межсетевых экранов. (Отметим, что в `Iptables` используется практически тот же набор понятий и терминов, что и в `Ipchains`.)

`Iptables` является мощным, но сложным средством, и обычно рекомендуется для пользователей, знакомых с межсетевыми экранами и искусством их конфигурирования (см. врезку о написании командных файлов). Если это ваш первый межсетевой экран, я рекомендую, по крайней мере для начала, воспользоваться для создания экранирующей конфигурации одним из рассмотренных далее в этой лекции средств с автоконфигурированием. Эти средства используют `Iptables` (или продукт-предшественник - `Ipchains`) для создания меж сетевого экрана согласно заданным исходным данным. Однако, прежде чем приступить к конфигурированию с помощью одного из графических инструментов, имеет смысл разобраться в основах "подкапотной механики" `Iptables`.

Установка Iptables

В большинство систем Linux с ядром 2.4 и выше межсетевой экран `Iptables` встроен, поэтому никаких дополнительных программ устанавливать не требуется. (Если версия ядра вашей системы меньше 2.4, то в нее встроены `Ipchains` или `Ipfwadm`. Это сходные средства, но они не рассматриваются в этой книге.) Инструкции `Iptables` можно выполнять из командной строки или из командного файла (см. врезку). Чтобы проверить, что межсетевой экран `Iptables` установлен, наберите в командной строке `Iptables -L` и посмотрите, какова реакция. Должен быть выведен текущий набор правил (который, вероятно, пуст, если вы еще не сконфигурировали межсетевой экран).

Если ваша система не содержит `Iptables`, или если вы хотите получить самую свежую версию, посетите <http://www.netfilter.org/> и загрузите RPM для своей операционной системы. Можно также взять его с компакт-диска, приложенного к книге.

Если на вашем установочном диске нет Webmin RPM, зайдите на <http://www.webmin.com/> и посмотрите, доступна ли там версия Webmin для вашей операционной системы. Webmin требуется для Turtle Firewall, и существуют специальные версии для каждого дистрибутива и операционной системы. Если для вашей операционной системы нет соответствующей версии, то вы не сможете использовать Turtle Firewall, но список поддерживаемых систем весьма велик. Щелкните мышью на файле RPM в X-Window, и он будет установлен автоматически.

Использование Iptables

Идея, лежащая в основе Iptables и Ipchains, состоит в создании каналов входных данных и их обработке в соответствии с набором правил (конфигурацией вашего межсетевого экрана) с последующей передачей в выходные каналы. В Iptables правила располагаются в таблицах, а внутри таблиц - в цепочках. Основными цепочками, используемыми в Iptables, служат:

- Input.
- Forward.
- Prerouting.
- Postrouting.
- Output.

Общий формат инструкций Iptables таков:

Iptables команда спецификация_правил расширения,

где команда, спецификация_правил и расширения - это одна или несколько допустимых опций. В [табл. 3.3](#) перечислены команды Iptables, а [табл. 3.4](#) содержит спецификации правил Iptables.

Таблица 3.3. Команды Iptables

| Команда | Описание |
|--------------------------------|---|
| -A цепочка | Добавляет в конец указанной цепочки одно или несколько правил, заданных в инструкции вслед за командой |
| -I цепочка номер_правила | Вставляет правила в позицию с заданным номером в указанной цепочке. Это полезно, если вы хотите перекрыть правила, заданные ранее |
| -D цепочка спецификация_правил | Удаляет из указанной цепочки специфицированные номером или текстом правила |
| -R цепочка номер_правила | Заменяет правило в позиции с заданным номером в указанной цепочке |
| -L цепочка | Выдает все правила в цепочке. Если цепочка не задана, выдаются все цепочки |
| -F цепочка | Сбрасывает все правила в цепочке, по сути ликвидируя конфигурацию вашего межсетевого экрана. Это полезно в начале конфигурирования, чтобы гарантировать отсутствие существующих правил, способных конфликтовать с вашими новыми правилами |
| -Z цепочка | Обнуляет все счетчики пакетов и байтов в указанной цепочке |
| -N цепочка | Создает новую цепочку с заданным именем |
| -X цепочка | Удаляет указанную цепочку. По умолчанию удаляются все цепочки |
| -P цепочка политика | Задает политику для указанной цепочки |

Таблица 3.4. Спецификации правил Iptables

| Спецификация правила | Описание |
|----------------------|--|
| -p протокол | Задает протокол, которому соответствует правило. Допустимыми типами протоколов являются icmp, tcp, udp и all |
| -s адрес/маска | Задает определенный адрес или сеть для соответствия. Используется стандартная нотация с косой чертой для указания диапазона IP-адресов |

| | | |
|---------|--|---|
| -j цель | Указывает, что делать с пакетом, если он соответствует спецификациям. Допустимыми опциями для цели являются: | |
| | DROP | Отбрасывает пакет без всяких дальнейших действий. |
| | REJECT | Отбрасывает пакет и посылает в ответ пакет с уведомлением об ошибке. |
| | LOG | Протоколирует пакет в файле. |
| | MARK | Помечает пакет для дальнейших действий. |
| | TOS | Изменяет поле TOS (тип обслуживания). |
| | MIRROR | Меняет местами исходный и целевой адреса и посылает пакеты обратно, по сути "отражая" их назад отправителю. |
| | SNAT | Трансляция исходных сетевых адресов. Эта опция применяется при выполнении трансляции сетевых адресов. Исходный адрес преобразуется в другое статическое значение, определенное с помощью ключа - to-source. |
| | DNAT | Трансляция целевых сетевых адресов. Данная опция аналогична предыдущей, но применяется к целевым адресам. |
| | MASQ | Маскарад с помощью общедоступного IP-адреса. |
| | REDIRECT | Перенаправляет пакет. |

Существуют и другие команды и опции, но мы перечислили самые распространенные операции. Весь список команд можно найти в оперативной справке `Iptables`, набрав `man iptables` в командной строке.

Написание командных файлов

Часто требуется автоматизировать некоторый процесс или иметь одну команду для запуска нескольких инструкций. Применительно к межсетевому экрану обычно желательно, чтобы все его команды выполнялись при загрузке системы. Лучшим способом сделать это является написание командного файла. Это простой текстовый файл, содержащий команду или список команд. Интерпретатор командного языка выполняет команды, когда он вызывается пользователем, набравшим имя файла.

1. Чтобы создать командный файл, откройте сначала текстовый редактор, такой как `vi` или `EMACS`, и введите свои команды.
2. Введите в самом верху строку, которая выглядит следующим образом:

```
#!/bin/bash
```

Данная строка сообщает, какой интерпретатор использовать для выполнения команд. Вы должны иметь его в своей ОС, а команды, помещенные в файл, должны быть его корректными командами. Приведенный пример задает маршрутное имя интерпретатора `bash` в `Mandrake Linux`. Можно использовать другой интерпретатор, например, `Tcsh` или `Csh`. Просто задайте в первой строке его маршрутное имя. Затем сохраните файл.

3. Сделайте файл исполнимым, чтобы интерпретатор мог выполнить его как программу. Это делается с помощью команды `chmod`. Введите

```
chmod 700 имя_командного_файла
```

Такой режим доступа делает файл читаемым, записываемым и исполнимым.

Чтобы выполнить командный файл, наберите его имя в командной строке. (В `bash` необходимо задать `./` перед именем файла, расположенного в текущем каталоге.) После нажатия клавиши ввода должны выполняться команды из файла.

Вы должны находиться в том каталоге, где размещен командный файл, или задать его маршрутное имя. Чтобы он выполнялся из любого места, можно добавить этот каталог в переменную окружения `PATH` или поместить файл в один из каталогов, фигурирующих в значении `$PATH`.

Создание меж сетевого экрана Iptables

Опыт - лучший учитель, поэтому давайте рассмотрим пару команд, чтобы увидеть, как они используются в практическом приложении. Далее на примере `Iptables` показано, как создать межсетевой экран. Можно вводить команды интерактивно, по одной, чтобы сразу видеть результаты. Можно также поместить их в командный файл и выполнять его при загрузке системы, поднимая тем самым межсетевой экран (см. врезку о написании командных файлов). Набирайте их точно так же, как показано, сохраняя, в частности, регистр букв.

В следующем примере предполагается, что диапазон IP-адресов 192.168.0.1 - 192.168.0.254 принадлежит вашей подсети ЛВС, к которой подключен интерфейс `eth1`, и что интерфейс `eth0` является соединением с Интернетом или ГВС.

1. Начните с удаления всех существующих правил с помощью команды `Flush`:

```
iptables -F FORWARD
```

Это стирает все правила цепочки `FORWARD`, являющейся основной "воронкой" для всех пакетов, пытающихся пройти через межсетевой экран.

2. Очистите другие цепочки:

```
iptables -F INPUT
iptables -F OUTPUT
```

Эти команды стирают все правила на пути пакетов, направленных в локальную машину, и в выходной цепочке.

3. Поместите стандартную инструкцию "запретить все" в самое начало.

```
iptables -P FORWARD DROP
iptables -A INPUT -i eth0 -j DROP
```

4. Решение о допуске фрагментированных пакетов в `Iptables` необходимо оформить явным образом:

```
iptables -A FORWARD -f -j ACCEPT
```

5. Существует два типа распространенных атак, которые необходимо сразу заблокировать. Одна из них называется подделкой (подделываются заголовки IP-пакетов, чтобы казалось, будто внешний пакет имеет внутренний адрес). Делая это, злоумышленник может попасть в вашу сеть, даже если вы используете собственные IP-адреса. Другой тип атаки реализуется отправкой потока пакетов на широковещательный адрес сети, чтобы перегрузить ее. Это называется штормовой атакой. Атаки перечисленных типов можно блокировать с помощью двух простых инструкций:

```
iptables -A FORWARD -s 192.168.0.0/24 -i eth0 -j DROP
iptables -A FORWARD -p icmp -i eth0 -d 192.168.0.255 -j DENY
```

Первая инструкция предписывает отбрасывать все пакеты, приходящие из Интернет-интерфейса `eth0` с внутренним адресом 192.168.0.0/24. По определению ни один пакет не должен приходить из недоверенного интерфейса с внутренним, собственным исходным адресом. Вторая инструкция отвергает все приходящие извне на адрес внутренней сети широковещательные пакеты протокола ICMP.

6. Вы, как правило, желаете принимать входящие потоки данных, поступающие по соединениям, инициированным изнутри (например, кто-то просматривает web-страницу). Пока соединение, инициированное изнутри, поддерживается - все, наверное, хорошо. Можно, однако, ограничить тип пропускаемого внутрь трафика. Предположим, вы хотите разрешить сотрудникам только web-доступ и электронную почту. Можно определить типы трафика для прохода внутрь и только для уже инициированного соединения. Следующая инструкция разрешает потоки данных по web-протоколу HTTP и почтовому протоколу SMTP на основе этого критерия.

```
iptables -A FORWARD -p tcp -i eth0 -d 192.168.0.0/24 --dports
www,smtp --tcp-flags SYN, ACK -j ACCEPT
```

```
iptables -A FORWARD -p tcp -i eth0 -d 192.168.0.0/24 --dports
www,smtp --tcp-flags SYN, ACK -j ACCEPT
```

Флаг `-m multiport` извещает `Iptables`, что вы будете выдавать инструкции сопоставления с портами. Конструкция `-s sports` разрешает только трафик электронной почты и web-навигации. Опция `-syn` разрешает пакеты SYN с неустановленным флагом ACK (и RST), то есть инициирование соединений TCP, а предшествующий восклицательный знак инвертирует смысл этого условия. В результате допускаются только пакеты, не иницирующие соединений.

7. Чтобы можно было принять входящие соединения извне только на определенных портах (например, входящие соединения электронной почты с вашим почтовым сервером), и разрешить из этих же портов направлять трафик вовне, используйте следующие инструкции:

```
iptables -A FORWARD -m multiport -p tcp -i eth0 -d 192.168.0.0/24
--dport smtp --syn -j ACCEPT
```

Предполагается, что IP-адресом почтового сервера служит 192.168.0.2. Флаги `--dport` и `--sport` разрешают только почтовый трафик SMTP.

8. Можно разрешить пользователям инициировать исходящие соединения и передавать по ним данные, но только для определенных протоколов. Именно здесь вы можете запретить применение FTP и других необязательных программ.

```
iptables -A FORWARD -m multiport -p tcp -o eth0 -d
0.0.0.0 --dports www,smtp --syn -j ACCEPT
```

9. Необходимо пропускать некоторые входящие и исходящие пакеты UDP. UDP применяются для DNS, и, если эти пакеты заблокировать, то пользователи не смогут выполнять разрешение адресов. Так как, в отличие от TCP, UDP-пакеты не имеют состояния, нельзя полагаться на проверки флагов SYN или ACK. Вы хотите разрешить UDP только на порт 53, поэтому вы задаете `domain` (встроенную переменную для порта 53) как единственно допустимый порт. Это делается с помощью следующих инструкций:

```
iptables -A FORWARD -m multiport -p udp -i eth0 -d
192.168.0.0/24 --dports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth0 -s
192.168.0.0/24 --sports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth1 -d
0.0.0.0--dports domain -j ACCEPT
iptables -A FORWARD -m multiport -p udp -i eth0 -s
0.0.0.0 --sports domain -j ACCEPT
```

10. Первая из двух приведенных выше инструкций разрешает входящие дейтаграммы UDP, а вторая - исходящие. Аналогичные действия стоит проделать и для ICMP-пакетов (информационных сетевых пакетов, рассмотренных в [лекции 2](#)). Вы хотите разрешить только определенные типы пакетов, такие, например, как эхо-ответ для входящих (`--icmp-type 0`) или эхо-запрос для исходящих (`--icmp-type 8`). Этого можно добиться с помощью следующих инструкций:

```
iptables -A FORWARD -m multiport -p icmp -i eth0 -d
192.168.0.0/24 --dports 0, 3,11 -j ACCEPT
iptables -A FORWARD -m multiport -p icmp -i eth1 -d
0.0.0.0 --dports 8, 3,11 -j ACCEPT
```

11. Наконец, вы хотите установить протоколирование, чтобы, просматривая журнал, можно было увидеть, какие пакеты были отброшены. Журнал желательно периодически просматривать, даже если проблем нет, просто чтобы иметь представление о видах отброшенного трафика. Если вы видите повторно отброшенные пакеты из одной и той же сети или одного адреса, то вас, возможно, атаковали. Протоколирование всех видов трафика задается одной инструкцией:

```
iptables -A FORWARD -m tcp -p tcp -j LOG
iptables -A FORWARD -m udp -p udp -j LOG
iptables -A FORWARD -m udp -p icmp -j LOG
```

Готово! Вы получили межсетевой экран, защищающий от наиболее распространенных атак из Интернета.

IP-маскарад с помощью Iptables

Когда создавался Интернет, несколько больших блоков адресов были выделены для использования в собственных сетях. Эти адреса не маршрутизируются в Интернете, их можно использовать, не опасаясь конфликтов с другими сетями. Диапазонами собственных адресов являются

10.0.0.0 - 10.255.255.255

192.168.0.0 - 192.168.255.255

172.16.0.0 - 173.31.255.255

Используя эти адреса в своей внутренней сети и имея один внешний, маршрутизируемый IP-адрес для межсетевого экрана, вы эффективно закроете внутренние машины от внешнего доступа. С помощью Iptables несложно выстроить дополнительный защитный рубеж, используя IP-маскарад.

Межсетевой экран отсекает внутренний IP-заголовок и заменяет его заголовком, задающим экран в качестве отправителя. Затем пакет данных посылается в место назначения с исходящим IP-адресом общедоступного интерфейса межсетевого экрана. Когда пакет возвращается, экран вспоминает, по какому внутреннему IP-адресу тот направлен, и переадресует его для внутренней доставки. Этот процесс называется также трансляцией сетевых адресов (NAT). В Iptables трансляцию адресов можно организовать с помощью следующих инструкций:

```
iptables -t nat -P POSTROUTING DROP
iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

Флаг MASQUERADE можно сократить до MASQ. Одним из усовершенствований Iptables по сравнению с предыдущими системами, такими как Ipchains и Ipfwadm, является способность решения дополнительных задач, таких как трансляция сетевых адресов.

Теперь вы знаете, как создать базовую конфигурацию межсетевого экрана. Она проста, но возможные вариации бесконечны. Можно переадресовывать определенные порты на внутренние серверы, чтобы не назначать для них обязательно общедоступные IP-адреса. Можно вставить в экранирующий компьютер еще одну сетевую плату и сделать ее интерфейсом демилитаризованной зоны для серверов с общедоступными адресами. По продвинутым конфигурациям межсетевых экранов написаны целые тома и существует множество списков почтовой рассылки; один из лучших среди них - firewall-wizards. Чтобы подписаться на эту рассылку, направьте сообщение со словом "subscribe" в теле письма по адресу

firewall-wizards-request@honor.icsalabs.com

Список firewall-wizards содержит обсуждение всех уровней конфигурации межсетевых экранов и не ориентирован на определенных производителей, то есть обсуждаются все модели экранов - от открытых до коммерческих.

Если вы хотите быстро построить межсетевой экран, не набирая все эти инструкции Iptables и не запоминая их синтаксис, то существует инструмент, который создает экранирующие инструкции с помощью графического интерфейса, избавляя вас от технической работы.

Turtle Firewall: Межсетевой экран на основе Iptables с графическим пользовательским интерфейсом

Turtle Firewall

Автор/основной контакт: Andrea Frigido

Web-сайт: <http://www.turtlefirewall.com/>

Платформы: Большинство совместимых с Linux, поддерживающих Iptables

Лицензия: GPL 2.0

Контактная информация: andrea@friweb.com

Системные требования: Операционная система Linux с ядром 2.4 и выше

Perl с библиотекой expat

Сервер Webmin

Это небольшое изящное приспособление, именуемое Turtle Firewall, создал Андреа Фриджидо. По сути, Turtle является набором командных файлов Perl, которые делают за вас всю черновую работу по подготовке межсетевого экрана `Iptables` к работе. Эта программа существенно облегчает просмотр правил и проверку того, что инструкции поступают в правильном порядке. Она выполняется как служба, поэтому вам не нужно заботиться об инициализации межсетевого экрана с помощью командного файла. Она использует службу Linux Webmin, являющуюся небольшим web-сервером, позволяющим изменять конфигурацию из web-навигатора. Вообще говоря, запуск web-сервера на экранирующем компьютере несколько снижает безопасность последнего, но достигаемое упрощение конфигурирования перевешивает этот недостаток. В наше время многие коммерческие поставщики используют для конфигурирования интерфейс Web-навигатора. Важное достоинство такого подхода - возможность конфигурирования с любой машины Windows или UNIX.

Андреа предлагает также опцию коммерческой поддержки. Не более чем за 100 евро (не спрашивайте меня, сколько это долларов; во время написания книги соотношение между долларом и евро было примерно 1:1) предоставляется 30-дневная поддержка по электронной почте, так что на этапе ввода в действие без помощи вы не останетесь. Поддержка может оказаться полезной и на этапе эксплуатации, если возникают проблемы, с которыми вы не можете справиться самостоятельно.

Установка Turtle Firewall

Установка и начальная настройка Turtle Firewall очень проста, так как используется модуль администрирования Webmin, доступный на большинстве платформ Linux.

1. Если вы не установили модуль администрирования Webmin при установке ОС, следует сделать это сейчас, так как он необходим для Turtle Firewall. Найдите и запустите RPM, который должен быть на большинстве дисков с дистрибутивами Linux. Щелкните на файле RPM мышью, и он установится автоматически.
2. Когда это будет сделано, вы получите возможность входа в программу настройки межсетевого экрана, вводя его IP-адрес в окне навигатора и нажимая клавишу ввода.
3. Теперь все готово к установке Turtle Firewall. Загрузите упакованный дистрибутив с <http://www.turtlefirewall.com/> или возьмите его с компакт-диска, прилагаемого к книге, и распакуйте.
4. Перейдите в каталог `turtlefirewall` и наберите

```
./setup
```

Запустится командный файл установки, который поместит модули Perl и другие необходимые файлы в нужные места.

5. Используя Web-навигатор, войдите на сервер Webmin, указав его IP-адрес или имя хоста. Отобразится интерфейс Webmin.
6. Щелкните мышью на вкладке Module Index, и в окне отобразится основной экран Turtle Firewall ([рис. 3.2](#)).



Рис. 3.2. Основной экран Turtle Firewall

7. Щелкните мышью на иконке Firewall Items, чтобы начать конфигурирование межсетевого экрана. Сначала вам придется задать основные сведения о нем (рис. 3.3). В Turtle Firewall применяется концепция зон для определения доверенных и недоверенных сетей. Доверенная зона связывается с сетью, где работают люди, которым принято доверять (пример - ваша внутренняя сеть). Недоверенная зона - это сеть, в которой может работать кто угодно, от сотрудников до заказчиков, продавцов или даже злоумышленников. В Turtle они называются "good" и "bad", но это по сути то же самое, что доверенная и недоверенная.

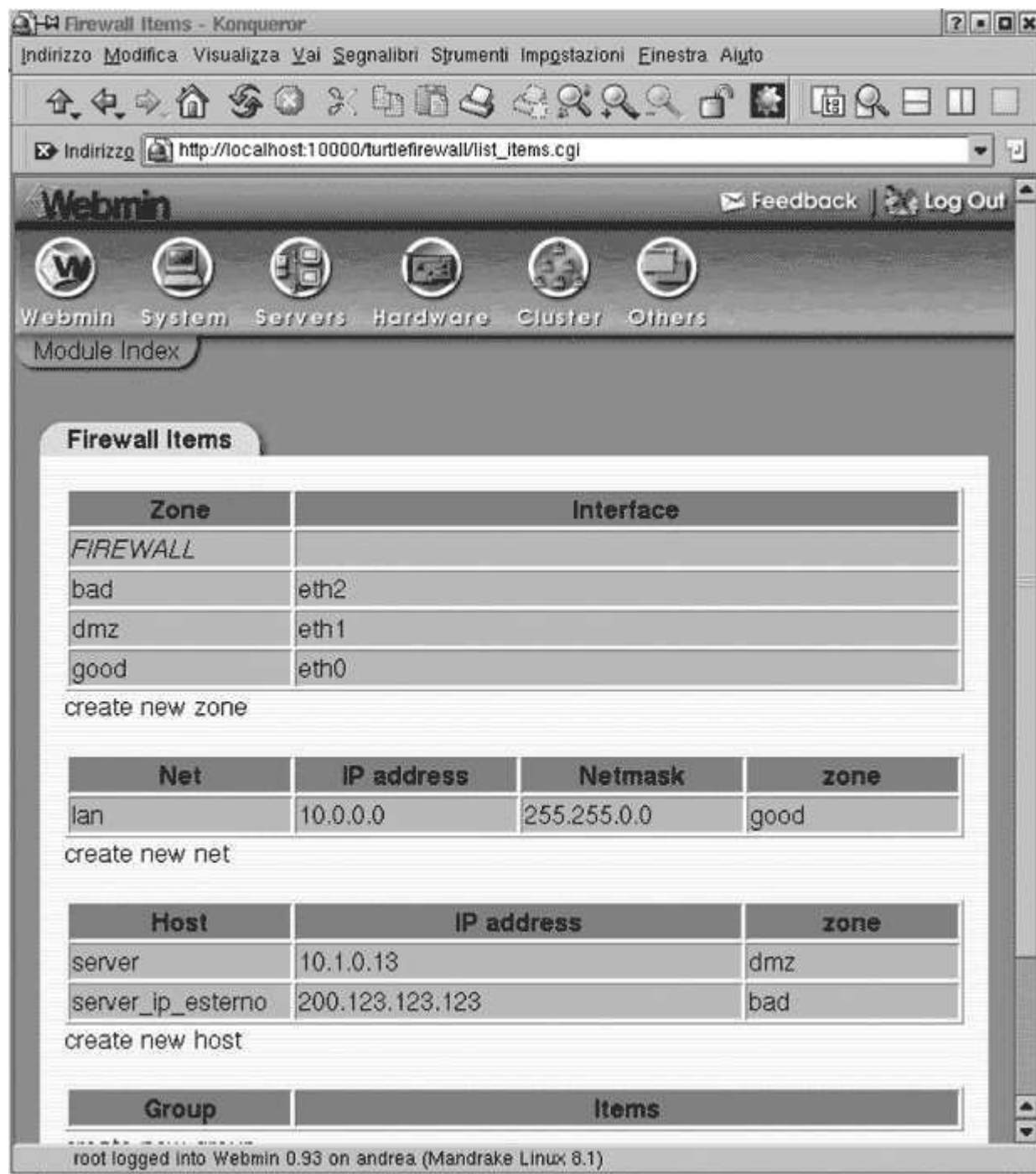


Рис. 3.3. Конфигурирование Turtle Firewall

В Turtle предусмотрен также элемент для демилитаризованной зоны (dmz), в которой располагают серверы со свободным доступом для недоверенной зоны. Задайте интерфейсы для доверенной, недоверенной и демилитаризованной (если таковая имеется) зон.

8. Затем в блоке Net следует задать адреса внутренней сети. Укажите диапазон IP-адресов с маской подсети для внутренней сети, которая будет защищаться межсетевым экраном, в предоставленном поле ([рис. 3.3](#)).
9. После этого задайте все хосты во внутренней сети и демилитаризованной зоне, требующие специального рассмотрения (пример - почтовый или Web-сервер). Сделайте это в блоке Hosts ([рис. 3.3](#)).
10. Наконец, в области Group можно определить все специальные хосты, к которым желательно подходить особым образом (пример - машины администраторов). Теперь ваш межсетевой экран готов к работе в базовом режиме.

Вероятно, вы захотите добавить некоторые дополнительные ограничения или разрешения, например, возможность кому-то извне использовать для входа SSH. Это можно сделать, написав правило под вкладкой Firewall Rules. Щелкните на ней мышью, и с вами начнут графический диалог для написания нового правила. Вы обнаружите сходство структуры диалога с форматами инструкций `Iptables` ([рис. 3.4](#)).

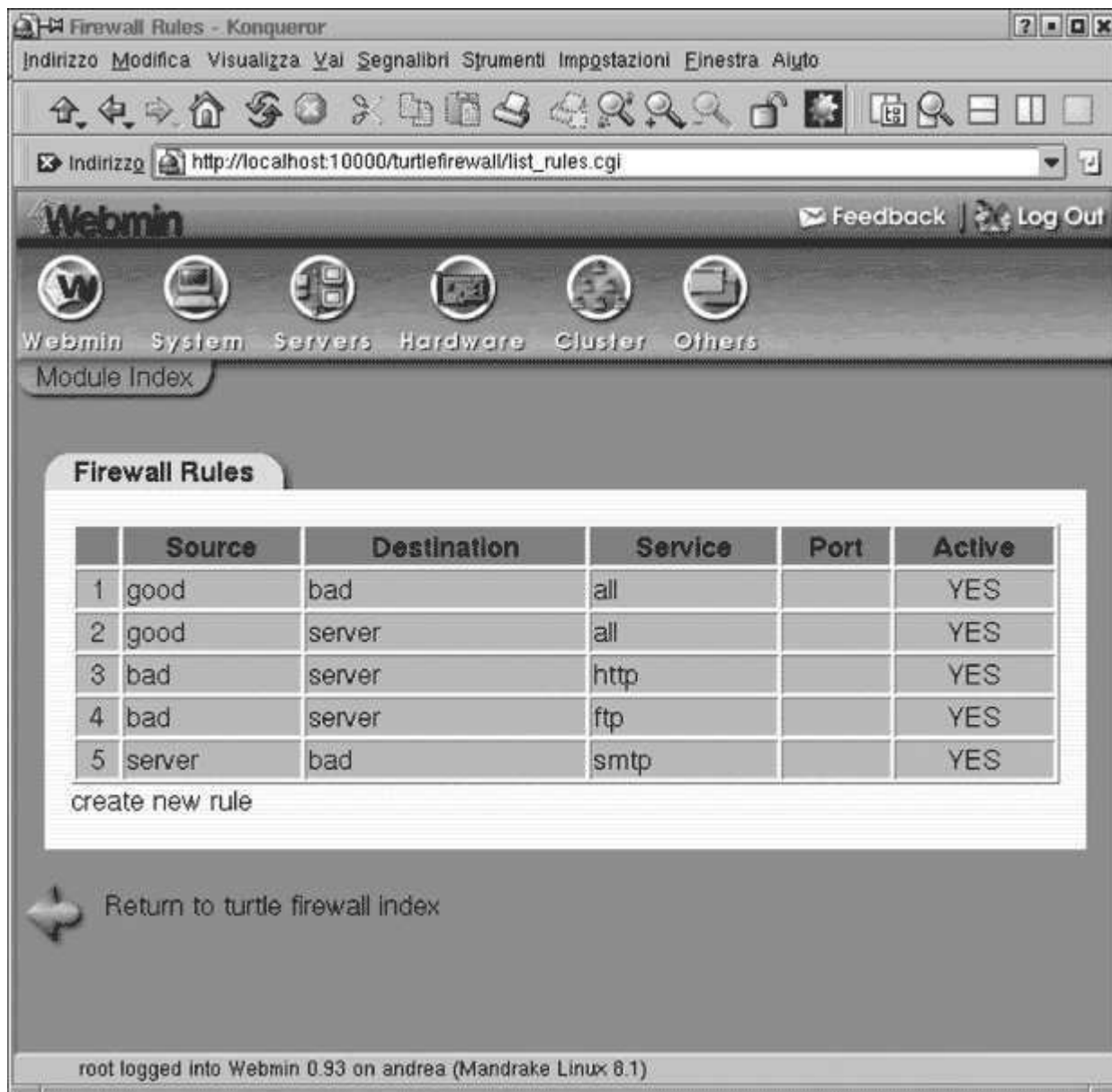


Рис. 3.4. Правила Turtle Firewall

Если вы хотите реализовать функцию маскировки *Iptables*, используя собственные IP-адреса для вашей внутренней сети, щелкните мышью на иконке "NAT and Masquerading" на основном экране. Вы сможете специфицировать, какая зона будет подвергаться маскировке (рис. 3.5). Обычно это доверенный интерфейс. Здесь же можно задать хосты, сетевые адреса которых будут транслироваться. Хост, заданный в качестве виртуального, будет служить фасадом реального хоста, и межсетевой экран будет переправлять все пакеты реальному хосту через виртуальный. Это создает дополнительный защитный рубеж для внутренних серверов.

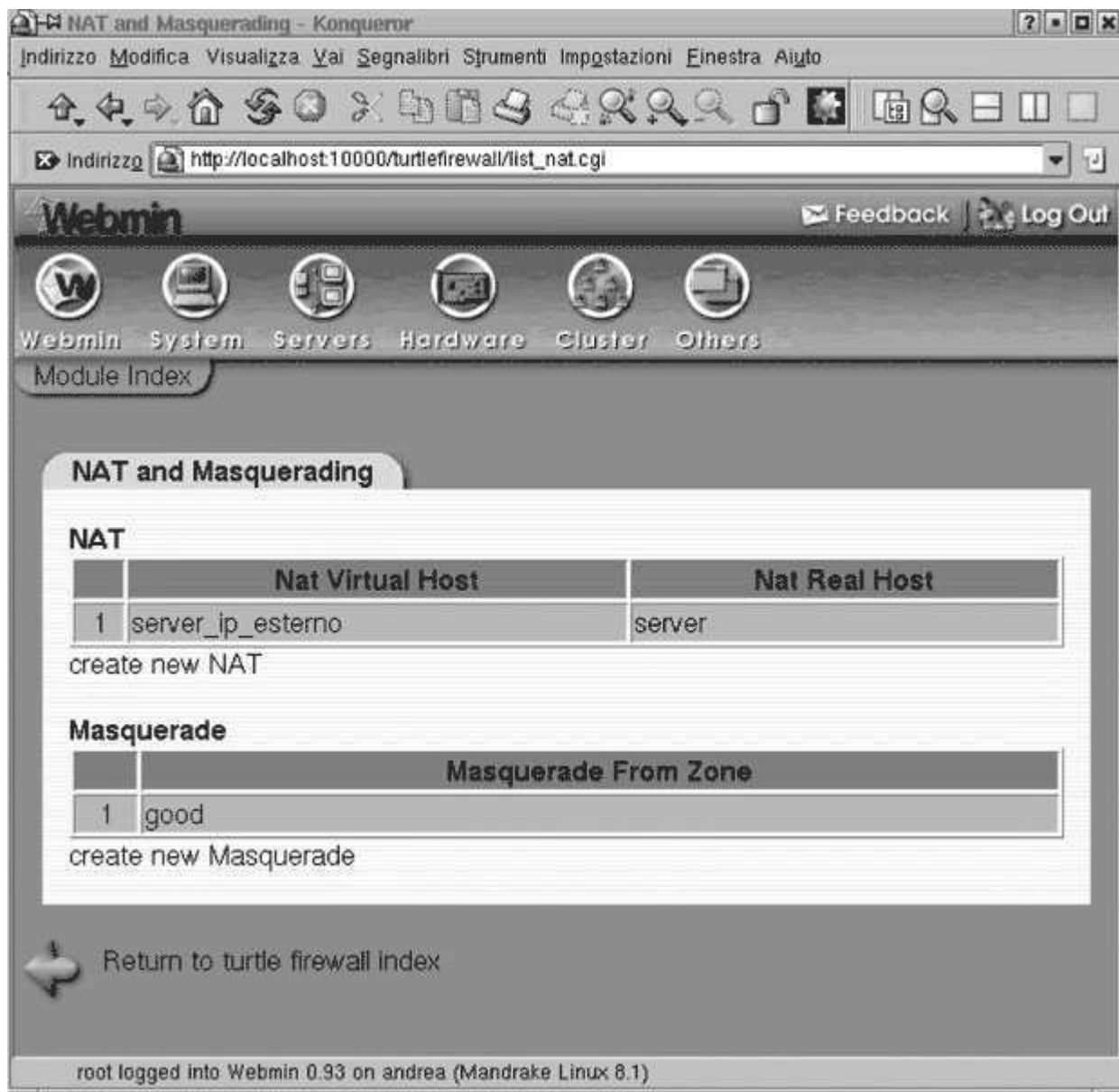


Рис. 3.5. Трансляция сетевых адресов и маскарад в Turtle Firewall

SmoothWall Express: Полный многофункциональный межсетевой экран SmoothWall Express

Автор/основной контакт: Lawrence Manning, Richard Morrel, Jon Fautley,
Tom Ellis (первоначальные авторы)

SmoothWall Limited (текущие контакты)

Web-сайт: <http://www.smoothwall.org/>

Платформа: Linux

Лицензия: GPL

Рассмотренная версия: 2.0

Web-форумы:

<http://community.smoothwall.org/forum/>

Каналы чата IRC:

Используйте сервер IRC [irc.smoothwall.org](irc://irc.smoothwall.org) 6667.

Вопросы и общая дискуссия по SmoothWall доступны на канале #help.

Списки почтовой рассылки:

Для общей поддержки и помощи при установке подпишитесь на:

<http://lists.smoothwallusers.org/mailman/listinfo/gpl>

Две обсуждавшиеся выше программы, *Iptables* и Turtle Firewall, предоставляют недорогой способ построения простого межсетевого экрана. Однако, если вам требуется сервер динамического конфигурирования хостов (DHCP-сервер), придется устанавливать его отдельно. Далее, если вы хотите использовать SSH для входа в компьютер, понадобится установить еще одну программу. SmoothWall - это межсетевой экран с открытыми исходными текстами, предлагающий мощный экранирующий пакет, в который встроены все перечисленные и многие другие возможности. Он создан компанией, которая предлагает как свободную (с лицензией GPL), так и коммерческую версии (последняя - с некоторыми дополнительными возможностями и улучшенной поддержкой). Это - еще один пример того, как продукт может использовать мощь открытого ПО и приносить компании коммерческую выгоду. Свободная версия называется SmoothWall Express и имеет на момент написания книги номер 2.0; коммерческая версия именуется SmoothWall Corporate Server, ее номер - 3.0.

SmoothWall Express содержит несколько опций, отсутствующих в *Iptables*, которые большинство организаций хотели бы иметь в полнофункциональном межсетевом экране. Конечно, можно собрать все это вместе из других программ и *Iptables*, но SmoothWall предлагает все в одной программе в простом для установки пакете. Вот некоторые из его возможностей:

- Поддержка виртуальных собственных сетей: SmoothWall объединяет виртуальные защищенные сети на основе протоколов IPsec со средствами межсетевого экранирования. Это позволяет осуществлять извне (например, из стационарного удаленного филиала или из произвольной точки маршрута коммивояжера) безопасный доступ к локальной сети по шифруемому туннелю (нестатическая виртуальная собственная IP-сеть поддерживается только в корпоративной редакции).
- Клиент и сервер DHCP: Клиент позволяет межсетевому экрану получать динамический IP-адрес для своего внешнего (в ГВС) интерфейса. Это обычная практика для предоставления Интернет-услуг через абонентские цифровые линии или кабельные модемы. Кроме того, межсетевой экран получает возможность действовать в качестве сервера DHCP для внутренней ЛВС, выдавая IP-адреса в соответствии с предварительно заданной политикой. И снова отметим, что можно, конечно, добавить эти возможности к *Iptables*, но в результате вы получите две программы, которые придется устанавливать и администрировать отдельно.
- Доступ к межсетевому экрану посредством SSH и web-навигатора: Безопасный доступ средствами командной строки и web-навигатора. Turtle Firewall предоставляет для *Iptables* web-, но не SSH-доступ. В SmoothWall встроены обе возможности без необходимости устанавливать дополнительное программное обеспечение.

- Сервер-посредник web: Возможность настроить web-посредник так, чтобы доступ ко всем web-сайтам осуществлялся через межсетевой экран. Это обеспечивает определенный уровень web-безопасности, так как любая программа, использующая уязвимости, должна будет выполняться на межсетевом экране, а не на локальной машине. Дальнейшего повышения безопасности можно добиться с помощью опции фильтрации информационного наполнения, доступной от SmoothWall Limited.
- Кэширующий web-сервер: Средство запомнить наиболее популярные web-страницы для замены удаленного доступа локальным, что дает выигрыш во времени и в пропускной способности.
- Обнаружение вторжения: SmoothWall предлагает некоторые базовые сетевые средства обнаружения вторжений.
- Графики и отчеты: SmoothWall позволяет получать простые отчеты о работе межсетевого экрана и генерировать графики на основе этих данных.
- Поддержка дополнительных типов соединений: SmoothWall поддерживает многие типы интерфейсов, включая коммутируемый, кабельный, ADSL, ISDN и Ethernet. При использовании Ipchains некоторые из этих интерфейсов требуют дополнительного программного обеспечения и конфигурирования.

Одним из существенных различий между SmoothWall и упомянутыми выше программами является необходимость функционирования SmoothWall на выделенной машине. При установке межсетевого экрана SmoothWall очищает жесткий диск и устанавливает собственную операционную систему. (По сути это урезанная версия Linux с повышенной безопасностью, но для обслуживания SmoothWall вам не нужно ничего о ней знать.) Это означает, что вы не сможете запускать на данной машине другие средства или использовать ее для иных целей (по крайней мере без больших трудностей и потенциальной опасности нарушить работу программного обеспечения SmoothWall). Не всех это устроит, но если вы ищете недорогой и быстрый способ построить готовый к немедленной эксплуатации межсетевой экран с множеством возможностей, то SmoothWall - как раз для вас.

Требования SmoothWall к оборудованию

Выше упоминалось, что SmoothWall должен функционировать на выделенной машине, но, к счастью, требования к ней невелики, так как на ней будет выполняться только программное обеспечение межсетевого экрана. Минимальная конфигурация состоит из ПК с процессором, совместимым с Intel Pentium 200 МГц, ОЗУ 32 МБ и 512 МБ дискового пространства. В качестве более оптимальной конфигурации рекомендуется процессор 500 МГц, ОЗУ 64 МБ и диск 2 ГБ. Этим спецификациям соответствуют все машины, за исключением разве что самых старых. Кроме того, нужен привод компакт-дисков и по крайней мере одна сетевая плата (обычно две, если интерфейсом в ГВС служит Ethernet).

Сравнение SmoothWall Express и SmoothWall Corporate

Если вы располагаете некоторой суммой денег на расходы и рассматриваете коммерческие альтернативы свободным средствам, есть смысл остановиться на корпоративной редакции SmoothWall. Этот межсетевой экран обладает всеми достоинствами версии Express со следующими существенными отличиями:

- Улучшенная поддержка обнаружения вторжений;
- Средства отказоустойчивости соединений;
- Поддержка роуминга в виртуальных собственных сетях (динамические IP-адреса);
- Дополнительные графики и отчеты;
- Улучшенный графический интерфейс пользователя;
- Поддержка в виртуальных собственных сетях аутентификации с применением сертификатов.

С полным списком отличий можно ознакомиться по адресу: http://download.smoothwall.org/archive/docs/promo/CorporateServer_vs_Express_Comparison_20040113.pdf.

Цена коммерческой версии вполне разумна (посмотрите текущие цены на web-сайте), она существенно меньше стоимости оборудования, на котором межсетевой экран будет функционировать. Компания SmoothWall делает и другие программные продукты - для мониторинга сети и фильтрации информационного наполнения. Изучите всю линейку предлагаемых продуктов, обратившись по адресу <http://www.smoothwall.net/>.

Установка SmoothWall

Предостережение: Помните, что при установке SmoothWall сотрет все данные на жестком диске и разместит на нем собственную операционную систему. Не запускайте его установку на компьютере, где имеются нужные вам данные или программы.

1. Сначала необходимо создать загрузочный компакт-диск. Воспользуйтесь для этого программным обеспечением для записи компакт-дисков, таким как Nero или Easy CD Creator, и создайте диск из файла с образом .iso (каталог SmoothWall на компакт-диске, приложенном к этой книге). Созданный диск будет загрузочным.
2. Сконфигурируйте ПК для начальной загрузки с компакт-диска (иначе будет просматриваться жесткий диск и загрузится найденная там операционная система). Обычно это делается в настройках BIOS на ПК, которые доступны в момент старта системы до загрузки ОС. На многих ПК для входа в этот режим служит функциональная клавиша F2.
3. Выполните загрузку с компакт-диска. Появится заглавный экран, содержащий основную информацию о лицензии и ограничении ответственности. Щелкните мышью на ОК.

У вас есть выбор между загрузкой с компакт-диска и через HTTP. Помните, что в этот режим нельзя входить, если вы не готовы к стиранию всех данных с жесткого диска и замены их программным обеспечением SmoothWall.

Выберите "CD-ROM", и установка начнется.

Вы увидите, что сначала происходит форматирование диска, а затем опробование сетевых интерфейсов компьютера (автообнаружение всех сетевых интерфейсных плат). У вас есть возможность принять или пропустить каждую из плат и задать их в качестве интерфейсов межсетевого экрана. Например, если в компьютере имеются две сетевые платы, но в качестве интерфейса межсетевого экрана желательно использовать только одну из них.

4. Задайте атрибуты всех выбранных интерфейсов. Присвойте им IP-адреса и маски подсети. После этого SmoothWall установит некоторые дополнительные файлы драйверов и попросит вас вынуть компакт-диск. Вы завершили установку программы и автоматически переходите в режим настройки.
5. В режиме настройки вам будет предложено ввести имя хоста для SmoothWall. Это имя можно использовать для доступа к машине вместо ее IP-адреса в ЛВС.
6. Затем будет предложено восстановить конфигурацию с резервной копии. Эта удобная возможность позволяет легко восстановить первоначальную конфигурацию межсетевого экрана после крушения системы (при условии, что вы делали резервные копии, см. далее в этом разделе). Не выбирайте этот пункт, если не находитесь в процессе восстановления с резервной копии.
7. Если на предыдущем шаге выбрано конфигурирование нового межсетевого экрана (не с резервной копии), вам будет предложено задать параметры сетей нескольких типов:
 - ISDN: Оставьте состояние "Disable", если не используете ISDN. В противном случае добавьте параметры, соответствующие вашей линии ISDN.
 - ADSL: Этот пункт нужен только в случае использования асимметричных цифровых абонентских линий и наличия в компьютере модема ADSL. Оставьте состояние "Disable", если вы не используете ADSL, или если провайдер предоставил вам для подключения внешний модем. В противном случае щелкните мышью на настройки службы ADSL.
 - Сетевая конфигурация: В SmoothWall зоны подразделяются на три категории:
 - Зеленая (Green): Защищаемый внутренний сетевой сегмент - ваша "доверенная" сеть.
 - Красная (Red): Внешняя сеть, отгораживаемая от ЛВС. "Недоверенная" сеть, обычно - Интернет или все, что не является вашей ЛВС.
 - Оранжевая (Orange): Это необязательный сегмент, который может содержать доверенные машины, открываемые для доступа из Интернета (упоминавшаяся ранее демилитаризованная зона). Организация этого сегмента защищает внутреннюю ЛВС в случае компрометации одного из серверов (так как по умолчанию узлы ДМЗ лишены доступа к ЛВС) и позволяет этим машинам быть доступными внешнему миру.

Выберите конфигурацию, подходящую для вашей сети. Большинство простых сетей будут зелеными (красные - это для модемов или ISDN) или зелеными и красными, если в машине две сетевые платы.

8. Теперь пришло время конфигурировать сервер DHCP. Если вы хотите сделать ваш межсетевой экран ответственным за раздачу и управление динамическими IP-адресами в ЛВС, включите эту возможность. В противном случае оставьте ее выключенной. Вы можете задать распределяемый диапазон, а также сервер имен и срок годности выданных адресов.

9. Далее вы задаете несколько паролей для различных уровней и методов доступа. Вход "root" доступен с консоли и из командной строки и действует как root в UNIX в том смысле, что вы получаете полный контроль над машиной. Затем вы выбираете пароль для пользователя "setup", который также может обращаться к системе с консоли и из командной строки, но, по сравнению с пользователем "root", имеет более ограниченные возможности и может выполнять только служебную программу настройки.
10. Наконец, создается системный счет для web-интерфейса. Это не системный счет в смысле UNIX, он недоступен из командной строки и используется исключительно для управления доступом к различным возможностям средствами web-интерфейса.
11. Перезагрузите машину, и ваш межсетевой экран SmoothWall должен запуститься и заработать. Можно войти в машину с консоли под именем root или setup. Можно с помощью SSH подсоединиться с удаленного компьютера и получить интерфейс командной строки. Однако одной из по-настоящему изящных возможностей этой программы является развитый, дружелюбный графический интерфейс, доступный из любого web-навигатора и превращающий администрирование межсетевого экрана в синекуру.

Администрирование межсетевого экрана SmoothWall

Проще всего управлять межсетевым экраном SmoothWall с помощью web-интерфейса. Это дает вам мощное средство администрирования и наращивания функциональности. Доступ к этому интерфейсу можно получить двумя способами: через порт 81 для обычных Web-коммуникаций web или через порт 441 - для защищенных с помощью SSL. В любом случае вы помещаете IP-адрес или универсальный локатор ресурсов с номером порта в поле адреса web-навигатора. Например, если сетевой интерфейс на плате вашего межсетевого экрана имеет IP-адрес 192.168.1.1, наберите в Web-навигаторе

```
http://192.168.1.1:81/
```

для обычных web-коммуникаций или

```
http://192.168.1.1:441/
```

для защищенного web-доступа.

Появится заставка SmoothWall. Чтобы получить доступ к другим экранам, необходимо ввести имя пользователя и пароль. Подразумеваемое имя пользователя - admin, с паролем, заданным при конфигурировании для web-интерфейса. На главной странице располагается несколько основных меню, включающих ряд подменю ([рис. 3.6](#)):



Рис. 3.6. Основное меню SmoothWall

- Control: основная страница межсетевого экрана, содержащая данные об авторских правах и периоде работоспособности.
- About Your Smoothie: здесь находится несколько полезных подменю:
 - Status: показывает состояние различных служб SmoothWall.
 - Advanced: отображает детальную информацию о системе.
 - Graphs: одна из самых замечательных возможностей SmoothWall. Позволяет строить графики интенсивности потоков данных, чтобы можно было анализировать сетевой трафик на различных интерфейсах в разное время дня и в разные дни. Это средство можно использовать в качестве быстрого способа выявления проблем в сети. Если вы заметите резкий рост трафика в выходные дни или поздно ночью без известной причины, то это должно вас насторожить ([рис. 3.7](#)).

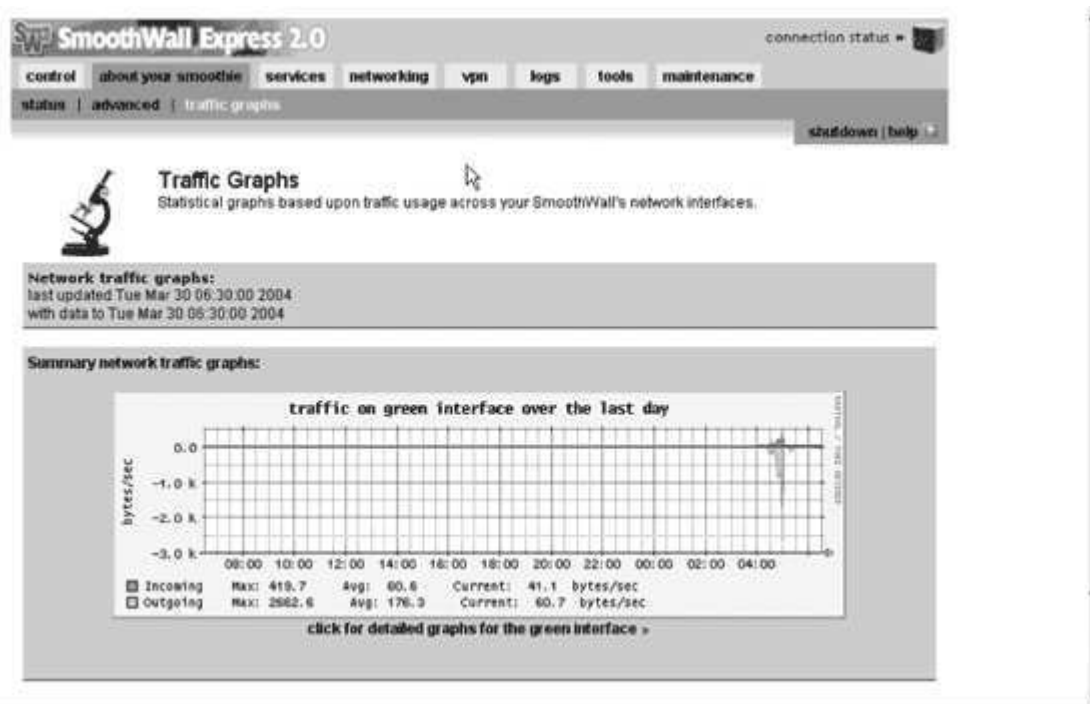


Рис. 3.7. График потоков данных через SmoothWall

- Services: здесь конфигурируются основные и дополнительные службы SmoothWall (рис. 3.8):
 - Web Proxy: если вы хотите использовать SmoothWall в качестве посредника при web-навигации, то эту функцию можно сконфигурировать здесь.
 - DHCP: здесь настраивается встроенный сервер DHCP.
 - Dynamic DNS: если поставщик Интернет-услуг присваивает вам динамический IP-адрес, но вы, тем не менее, хотите разрешить внешним сервисам внутренний доступ, можно сконфигурировать SmoothWall для автоматической замены записи DNS новым IP-адресом. Можно настроиться на использование любого из оперативных сервисов, таких как dyndns.org и dhs.org.
 - Remote Access: управление доступом к SmoothWall из любого места помимо консоли. Можно включить SSH (по умолчанию он отключен) и указать, с каких конкретных адресов разрешен доступ.
 - Time: конфигурирование параметров времени на машине. Это может быть очень важно, если вы сравниваете свои файлы журналов с другими серверами. Можно задать получение времени с общедоступного сервера времени, что делает журналы более точными.

Рис. 3.8. Экран служб SmoothWall

- **Networking:** здесь конфигурируется все, связанное с экранирующими и сетевыми функциями SmoothWall, в том числе добавление, удаление или модификация наборов правил и другие функции:
 - **Port Forwarding:** вы можете переправлять данные, посланные в определенный порт или группу портов, на внутренний защищенный хост.
 - **Internal Service Access:** щелкните мышью здесь, если требуется доступ извне к внутренним сервисам.
 - **DMZ Pinhole:** позволяет задать доступ из хоста в демилитаризованной зоне к хосту в ЛВС. Обычно это запрещено в силу предназначения ДМЗ.
 - **PPP Settings:** если вы используете SmoothWall для коммутируемого соединения с Интернет, то здесь можно задать различные телефонные настройки, такие как номер телефона, команды модема и т.д.
 - **IP Block:** удобное средство, позволяющее легко блокировать IP-адрес или диапазон IP-адресов из вашей сети без необходимости писать какие-либо правила.
 - **Advanced:** здесь располагается несколько вспомогательных сетевых настроек, таких как поддержка Universal Plug and Play (UpnP).
- **VPN:** здесь SmoothWall конфигурируется для работы в виртуальной собственной сети для безопасного удаленного доступа из других сетей. Детали рассмотрены ниже в данной лекции.
- **Logs:** этот экран упрощает доступ ко всем файлам журналов SmoothWall. Интерфейс позволяет легко просматривать любые типы журналов, как системных, так и относящихся к безопасности.
- **Tools:** здесь представлено несколько стандартных сетевых средств, включая `ping`, `traceroute` и `whois`, а также развитый клиент SSH на основе Java, позволяющий осуществлять доступ к серверам SSH из вашего web-навигатора.
- **Maintenance:** этот раздел используется для деятельности по сопровождению системы и содержит несколько подменю:
 - **Maintenance:** здесь отслеживаются все программные коррекции операционной системы SmoothWall. Своевременное наложение заплат на ОС SmoothWall очень важно. Как в любой операционной системе, в ней время от времени выявляются уязвимости, ликвидируемые с помощью корректирующих заплат. Кроме того, периодически добавляются новые возможности, повышается совместимость.
 - **Password:** здесь можно изменять входные имена и пароли для системы (при условии, что у вас есть старые пароли).
 - **Backup:** резервное копирование конфигурации SmoothWall с целью последующего восстановления после аварии системы. Необходимо сделать резервную копию, как только вы сконфигурируете SmoothWall нужным образом, чтобы сохранить свои настройки.

- Shutdown: безопасное выключение SmoothWall.

Создание виртуальной собственной сети с помощью межсетевого экрана SmoothWall

SmoothWall можно использовать для организации безопасного соединения с другой сетью посредством создания туннеля с шифрованием по спецификациям IPsec.

1. Чтобы сконфигурировать на межсетевом экране функции виртуальных собственных сетей, щелкните мышью на элементе VPN основного меню. В нем находятся два подменю ([рис. 3.9](#)).
 - Control: это основной экран, где вы можете начинать и завершать сеансы конфигурирования виртуальных собственных сетей, а также получать информацию об их состоянии.
 - Connections: здесь весьма несложным образом конфигурируются новые соединения. На SmoothWall Express (свободная версия с лицензией GPL) обе стороны должны иметь статические, общедоступные IP-адреса. Чтобы создать профиль нового соединения, перейдите на вкладку Connections основной вкладки VPN ([рис. 3.10](#)).

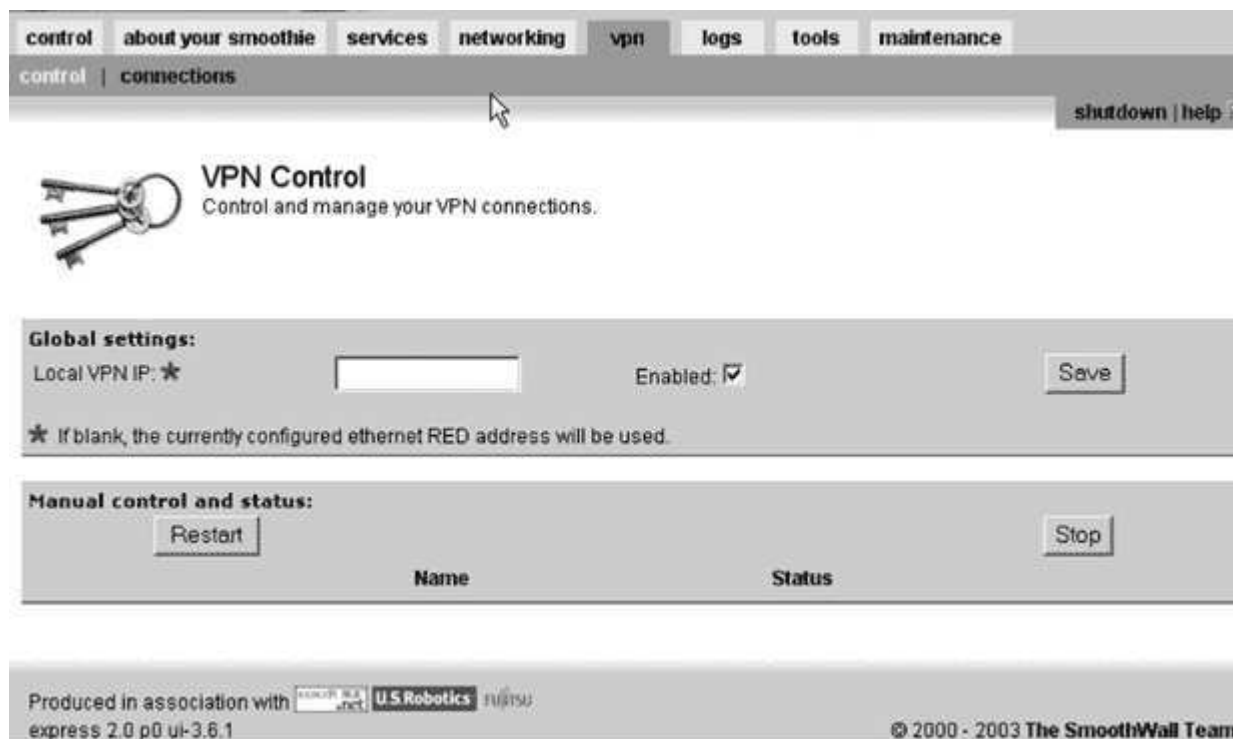


Рис. 3.9. Экран SmoothWall VPN Control



Рис. 3.10. Экран VPN Connections

2. Введите имя соединения, по возможности - mnemonicное.
3. Определите "левую" (Left) и "правую" (Right) стороны соединения. (Эти имена никак не связаны с направлением, но используются просто для ссылок на разные концы соединения. Локальная сторона обычно считается левой.) Введите IP-адрес и подсеть для локального SmoothWall на левой стороне, а также IP-адрес и подсеть удаленного SmoothWall на правой стороне.
4. Ниже вводится общий секрет, используемый для организации шифрования. Это секрет должен быть одинаковым на обоих соединяемых межсетевых экранах. Он должен быть защищенным и не передаваться небезопасным образом (например, по электронной почте). Сделайте текст секрета длиной не менее 20 символов, включите в него символы верхнего и нижнего регистров и специальные символы, чтобы сделать виртуальную собственную сеть как можно более защищенной.
5. Можно щелкнуть мышью в поле сжатия (Compression), чтобы снизить интенсивность потоков данных через виртуальную собственную сеть. Но помните, что это создает дополнительную нагрузку на процессор, которая способна превысить выигрыш от сжатия передаваемых данных и в итоге привести к замедлению работы сети.
6. Не забудьте щелкнуть мышью в поле активизации (Enable), а затем - на кнопке Add, чтобы добавить новое соединение. Теперь вы увидите его на основной странице VPN Control. Оно полностью готово к работе, если канал, с которым ассоциировано соединение, находится в работоспособном состоянии.
7. Можно экспортировать настройки виртуальной собственной сети на другой экземпляр SmoothWall, чтобы облегчить конфигурирование и избежать ошибок ввода при конфигурировании дополнительных оконечных точек. Щелкните мышью на кнопке Export. Будет создан файл с именем vpnconfig.dat, который можно перенести на удаленную машину, зайти на ту же страницу и выбрать Import. SmoothWall автоматически переставит записи для удаленного конца. Теперь ваша виртуальная собственная сеть готова к работе. Повторите этот процесс для всех производственных площадок, безопасное взаимодействие с которыми вы хотите обеспечить.

Дополнительные приложения SmoothWall

В этом разделе представлен лишь поверхностный обзор основных функций SmoothWall. Существуют другие, продвинутое функции, описанные в документации, прилагаемой к SmoothWall. Детали настройки других специальных служб, таких как web-сервер-посредник или динамический сервис имен, можно найти в справочнике администратора. Все три файла документации в PDF-формате помещены в каталог SmoothWall на компакт-диске, прилагаемом

к этой книге. Если вы располагаете свободной машиной, которую можете выделить исключительно для нужд межсетевого экранирования, то SmoothWall Express позволит вам выйти за рамки простой экранирующей функциональности и снабдить свою сеть полноценным защитным устройством.

Межсетевые экраны на платформе Windows

Ни один из межсетевых экранов, описанных в этой лекции, не работает на платформе Windows. С прискорбием приходится констатировать дефицит качественного программного обеспечения межсетевых экранов с открытыми исходными текстами для Windows. Поскольку сами исходные тексты Windows не являются открытыми, программистам нелегко создать столь сложный продукт, каковым является межсетевой экран, требующий доступа к коду уровня операционной системы. С добавлением базового межсетевого экрана в Windows XP, у программистов стало еще меньше мотивации разрабатывать альтернативные решения с открытыми исходными текстами. Это печально, так как межсетевой экран в XP хорош для индивидуальных пользователей, но не годится для решения задач корпоративного экранирования. Имеются коммерческие варианты продуктов для Windows от таких компаний, как Checkpoint, однако даже они отходят от ориентации исключительно на Windows из-за проблем безопасности данной платформы. Если требуется межсетевой экран на платформе Windows, то вам, вероятно, следует искать коммерческое решение, поскольку хорошего межсетевого экрана с открытыми исходными текстами для Windows нет. Это подчеркивает ограничения и проблемы операционных систем с закрытыми исходными текстами.

Инструменты безопасности с открытым исходным кодом

4. Лекция: Сканеры портов: версия для печати и PDA

Межсетевые экраны, помогающие защитить сеть от большинства простых атак, - обязательное средство для любой сети, подключенной к Интернету. Теперь, когда вы защитили парадный вход своей сети, рассмотрим средства, которые помогают проверить замки и окна и убедиться, что в сети нет опасных щелей.

Взглянув еще раз на модель ВОС, вы увидите, что после установления базового сетевого соединения между двумя машинами, приложение использует это соединение для выполнения функций, запрашиваемых пользователем. Приложение может загружать web-страницы, посылать электронные сообщения или осуществлять интерактивный вход с применением Telnet или SSH.

Обзор лекции

Изучаемые концепции:

- Порты TCP/UDP
- Идентификационные метки TCP
- Как работает сканирование портов
- Конфигурирование сканирования портов
- Методы сканирования портов

Используемые инструменты:

Nmap, Nmap for Windows, Nlog

| Номер уровня модели ВОС | Название уровня | Примеры протоколов |
|-------------------------|-----------------------|--|
| Уровень 7 | Прикладной уровень | DNS, FTP, HTTP, SMTP, SNMP, Telnet |
| Уровень 6 | Уровень представления | XDR |
| Уровень 5 | Уровень сеанса | RPC |
| Уровень 4 | Транспортный уровень | NetBIOS, TCP, UDP |
| Уровень 3 | Сетевой уровень | ARP, IP, IPX, OSPF |
| Уровень 2 | Канальный уровень | Arcnet, Ethernet, Token ring |
| Уровень 1 | Физический уровень | Коаксиальный кабель, оптоволокно, витая пара |

Малоизвестная, но важная организация Internet Assigned Numbers Authority (IANA) присваивает номера портов TCP/UDP. Она отслеживает множество различных стандартов и систем, обеспечивающих функционирование Интернета. Среди ее обязанностей - распределение IP-адресов и назначение ответственных за имена доменов верхнего уровня. IANA обладает значительной властью, хотя по большей части остается в тени. Немногие люди за пределами инженерных подразделений коммуникационных компаний знают о существовании IANA, но она управляет значительной частью "недвижимости" Интернета. IANA отвечает также за поддержание списка сетевых портов, по которым можно подключаться к определенным сервисам, предполагая, что приложение или операционная система соответствуют этим стандартам. Разумеется, всем компаниям, производящим программное

обеспечение, надлежит скрупулезно следовать этим стандартам, иначе их продукты могут оказаться несовместимыми с другими подключенными к Интернету системами. В [табл. 4.1](#) перечислены некоторые из наиболее употребительных TCP-портов для серверных приложений.

Полный список номеров портов представлен в приложении С. Самую свежую версию этого списка можно найти на Web-сайте IANA (<http://www.iana.org/>). Номер порта присвоен почти каждому значительному приложению. Как для TCP, так и для UDP-сервисов эти номера лежат в диапазоне от 1 до 65535. Номера портов от 0 до 1023 считаются зарезервированными для общеупотребительных приложений, обычно выполняющихся от имени пользователя root или другого привилегированного пользователя. Соответствующие им номера портов называются общеизвестными. Номера портов с 1024 по 65535 можно регистрировать в IANA для конкретных приложений. Они обычно соответствуют определенным сервисам, но подобная регистрация не имеет для производителей столь же обязательной силы, как в случае зарезервированных номеров.

Наконец, существуют недолговечные номера портов, которые операционная система выбирает случайным образом из номеров, превышающих 1024, (обычно - в верхней части диапазона). Они используются для машин, которые произвольным образом устанавливают соединения с другими машинами. Например, для загрузки web-страницы ваша машина обратится к порту 80 web-сервера. Сервер увидит входящее соединение с некоторым случайным номером порта, превышающим 1024. В таком случае сервер будет знать, что это, вероятно, пользователь, а не другое приложение, устанавливающее с ним соединение. Он также использует недолговечный номер порта для отслеживания определенного пользователя и сеанса. Например, если вы параллельно откроете два навигатора, то ваш компьютер для сеанса каждого из них создаст два разных номера порта для установления соединений, которые сервер будет считать различными.

Таблица 4.1. Общеупотребительные серверные порты

| Номер порта | Протокол | Сервис |
|-------------|----------|---|
| 21 | FTP | Протокол передачи файлов (управляющий порт) |
| 22 | SSH | Защищенный shell |
| 23 | Telnet | Telnet |
| 25 | SMTP | Почтовый сервис |
| 53 | DNS | Разрешение доменных имен |
| 79 | Finger | Finger |
| 80 | HTTP | Web-сервис |
| 135-139 | NetBIOS | Сетевые коммуникации Windows |
| 443 | SSL | Защищенный web-сервис |

То, что пакет помечен для порта 80, не запрещает ему содержать данные, отличные от web-трафика. Система номеров портов зависит от определенной "честности" машин, с которыми приходится взаимодействовать, и именно отсюда может прийти беда. На самом деле, многие приложения, такие как программы мгновенного обмена сообщениями и одноранговое ПО, которые обычно блокируются межсетевым экраном организации, нарушают эту конвенцию и проскальзывают через порт 80, который согласно конфигурации остается открытым, поскольку пользователям, находящимся позади межсетевого экрана, разрешен web-доступ.

Когда порт на компьютере открыт, он получает весь направляемый в него трафик, законный или незаконный. Посылая некорректно сформированные пакеты, пакеты со слишком большим количеством данных или с некорректно отформатированными данными, иногда можно вызвать аварийное завершение основного приложения, перенаправить поток управления в этом приложении и незаконно получить доступ к машине. Это называется переполнением буфера и составляет большой процент современных уязвимостей.

Переполнение буфера происходит, если прикладные программисты неаккуратно пишут программы и не обеспечивают должную обработку данных, "переполняющих" области памяти, отведенные входным переменным. Когда в программу поступают входные данные, не уместящиеся в отведенный буфер, они могут изменить внутренний ход выполнения программы и в результате предоставить хакеру доступ к ресурсам системного уровня.

Раньше это было технически сложной задачей, за которую могли взяться только самые квалифицированные хакеры. Но теперь, чтобы осуществить подобный взлом, уже не нужно быть высококлассным программистом. Доступны программы, которые с одного щелчка мыши автоматически выполняют переполнение буферов.

Почти все программы, независимо от размера, содержат ошибки такого рода. Современное программное обеспечение, насчитывающее миллионы строк исходных текстов, - просто-напросто слишком сложное, чтобы избежать подобных ошибок. Возможно, со временем, когда вырастут новые поколения программистов, обученных автоматически писать безопасный код, данная проблема потеряет свою остроту или исчезнет совсем. Пока же необходимо внимательно следить за тем, какие приложения или порты видны в вашей сети. Эти порты являются потенциальными "окнами" в серверах и рабочих станциях, через которые хакеры могут запускать свой вредоносный код в ваш компьютер. Поскольку именно здесь происходит большинство нарушений безопасности, очень важно понимать, что происходит на этом уровне на ваших серверах и других машинах. Этого можно легко добиться с помощью программного средства, называемого сканером портов.

Обзор сканеров портов

Сканеры портов, не мудрствуя лукаво, опрашивают набор портов TCP или UDP и смотрят, не ответит ли приложение. Если ответ получен, это означает, что некоторое приложение слушает порт с данным номером. Имеется 65535 возможных портов TCP и столько же - UDP. Сканеры можно сконфигурировать для опроса всех возможных портов или только общеупотребительных (с номерами, меньшими 1024). Веская причина для полного сканирования состоит в том, что сетевые троянские и другие вредоносные программы, чтобы избежать обнаружения, нередко используют нетрадиционные порты с номерами в верхней части диапазона. Кроме того, некоторые производители не следуют стандартам должным образом и подключают серверные приложения к портам с большими номерами. Полное сканирование охватывает все возможные места, где могут скрываться приложения, хотя и требует больше времени и пожирает несколько большую часть полосы пропускания.

Сканеры портов предстают во множестве видов от очень сложных с множеством различных возможностей до имеющих минимальную функциональность. На самом деле, вы сами можете вручную выполнить функции сканера портов, применяя Telnet и проверяя порты по очереди. Просто подключайтесь к IP-адресу, добавляя номер порта, например:

```
telnet 192.168.0.1:80
```

Данная команда использует Telnet для соединения с машиной. Номер после двоеточия (для некоторых реализаций Telnet необходимо просто оставить пробел между IP-адресом и номером порта) говорит Telnet, что для соединения надо использовать порт 80 вместо стандартного для Telnet порта 23. Вместо того чтобы получить от Telnet обычное приглашение, которое выдается при подключении к его подразумеваемому порту, вы соединитесь с web-сервером, если таковой запущен на машине. После нажатия клавиши ввода вы получите первый ответ web-сервера навигатору. Вы увидите информацию из заголовка HTTP, которая обычно обрабатывается навигатором и скрыта от пользователя. Она выглядит примерно так, как показано на [листинге 4.1](#).

```
GET / HTTP
```

```
HTTP/1.1 400 Bad Request
Date: Mon, 15 Mar 2004 17:13:16 GMT
Server: Apache/1.3.20 Sun Cobalt (Unix) Chili!Soft-ASP/3.6.2
mod_ssl/2.8.4 OpenSSL/0.9.6b PHP/4.1.2 mod_auth_pam_external/0.1
FrontPage/4.0.4.3 mod_perl/1.25
Connection: close
Content-Type: text/html; charset=iso-8859-1
```

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<HTML><HEAD>
<TITLE>400 Bad Request</TITLE>
</HEAD><BODY>
<H1>Bad Request</H1>
Your browser sent a request that this server could not understand
```

```
Request header field is missing colon separator.
<PRE>
</PRE>

</BODY></HTML>
```

Листинг 4.1. Ответ HTTP на соединение TCP

Так же можно поступить с любым открытым портом, но вы не всегда получите в ответ нечто вразумительное. По сути, именно это и делают сканеры портов: они пытаются установить соединение и ожидают ответ.

Некоторые сканеры портов пытаются также идентифицировать операционную систему на другом конце, выявляя так называемые идентификационные метки TCP. Хотя TCP/IP является стандартом сетевых коммуникаций, каждый производитель реализует его немного иначе, чем другие. Эти различия, обычно не мешающие взаимодействию, проявляются в ответах на любое воздействие, такое как эхо-тест или попытка установления TCP-соединения. Например, цифровая подпись ответа на эхо-тест от системы Windows выглядит иначе, чем в ответе системы Linux. Имеются даже различия между версиями операционной системы. На [листинге 4.2](#) приведен пример идентификационных меток TCP для Windows ME, 2000 и XP.

```
# Windows Millennium Edition v4.90.300
# Windows 2000 Professional (x86)
# Windows Me or Windows 2000 RC1 through final release
# Microsoft Windows 2000 Advanced Server
# Microsoft XP professional version 2002 on PC Intel processor
# Windows XP Build 2600
# Windows 2000 with SP2 and long fat pipe (RFC 1323).
# Windows 2K 5.00.2195 Service Pack 2 and latest hotfixes
# XP Professional 5.1 (build 2600).. all patches up to June 20, 2004
# Fingerprint Windows XP Pro with all current updates to May 2002
Fingerprint Windows Millenium Edition (Me), Win 2000, or WinXP
Tseq(Class=RI%gcd=<6%SI=<23726&>49C%IPID=I%TS=0)
T1(DF=Y%W=5B4|14F0|16D0|2EE0|402E|B5C9|B580|C000|D304|FC00|FD20|FD68
|FFFF%ACK=S++%Flags=AS%Ops=NNT|MNWNNT)
T2(Resp=Y|N%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=5B4|14F0|16D0|2EE0|B5C9|B580|C000|402E|D304|FC00|FD20|FD68
|FFFF%ACK=S++%Flags=AS%Ops=MNWNNT)
T4(DF=N%W=0%ACK=0%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=0%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(DF=N%TOS=0%IPLen=38%RIPTL=148%RID=E%RIPCK=E|F%UCK=E|F%ULEN=134%DAT=E)
```

Листинг 4.2. Идентификационные метки TCP для Windows

Тарабарщина в нижней части листинга является уникальными установками, используемыми Windows при установлении TCP-соединений. Сравнивая полученный от машины ответ с базой известных идентификационных меток TCP, можно сделать разумное предположение об операционной системе на другом конце.

Данный метод не является совершенным. Иногда программа сканера портов ошибается, поскольку некоторые производители операционных систем при реализации стека TCP заимствуют части других систем (систем UNIX в особенности). Это заставляет сканер портов считать, что перед ним - ОС-первоисточник. Существуют также необычные операционные системы, например, в коммутаторах, принтерах и сетевых устройствах, которые могут отсутствовать в базе данных сканера.

Если вашу сеть сканируют с не очень похвальными намерениями, это предоставляет злоумышленникам ценную информацию. Знание операционной системы и ее версии может послужить хорошей отправной точкой для определения того, какие зацепки и средства проникновения стоит попробовать. Это очень веская причина для регулярного сканирования своей сети, чтобы определить, какие порты в системе оставлены открытыми. Затем следует их просмотреть, закрыть неиспользуемые порты и защитить те, которые должны оставаться открытыми.

Соображения по поводу сканирования портов

При планировании сканирования портов любой сети помните, что эта деятельность создает большую нагрузку на сеть. Сканирование за короткое время десятков тысяч портов порождает в сети интенсивный трафик. Если вы используете для сканирования устаревшей сети на 10 Мбит/с мощный компьютер, это может существенно повлиять на сетевую производительность. При сканировании через Интернет данная проблема будет менее острой, так как ограничивающим фактором послужит пропускная способность промежуточных соединений, однако все равно можно снизить производительность загруженного web-сервера или почтового сервера. В крайних случаях ваша активность может даже привести к прекращению работы машин.

Независимо от способа использования, описанных выше средств обязательно получите разрешение владельца сканируемых хостов. Сканирование портов - деятельность на грани законности (в действительности вы не взламываете системы, просто опрашиваете сеть). Однако вашему начальнику может быть не до нюансов, если вы нарушите работу корпоративной сети. И прежде чем вы забавы ради решите просканировать несколько любимых web-серверов, учтите, что в контракте на предоставление Интернет-услуг могут содержаться пункты, запрещающие подобную деятельность. Операторы web-сайтов постоянно подают жалобы на поставщиков Интернет-услуг, клиенты которых регулярно позволяют себе ненадлежащее поведение. Поэтому, если вы не хотите, чтобы вас уволили или отключили от Интернета, получите письменное разрешение либо от вашего руководителя (если работаете на свою организацию), либо от клиента/добровольца (если обслуживаете третью сторону). В приложении D помещено стандартное письменное соглашение для получения разрешения от предполагаемого объекта сканирования, которое является хорошей отправной точкой для юридического прикрытия ваших позиций.

Даже при наличии разрешения необходимо принять во внимание предполагаемый эффект сканирования целевой сети. Если это интенсивно используемая сеть, вы должны выполнять сканирование ночью или в периоды наименьшей активности. Некоторые сканеры имеют возможность замедлять посылку пакетов, чтобы не очень сильно воздействовать на сеть. Это означает, что сканирование будет выполняться дольше, но в более дружественном для сети режиме.

Некоторые современные устройства, такие как межсетевые экраны и некоторые маршрутизаторы, достаточно интеллектуальны, чтобы распознать сканирование своих портов и отреагировать на него. Iptables можно сконфигурировать для этого, используя опцию multiport и устанавливая флаг приоритета. Машины могут отвечать на сканирование портов снижением скорости ответа для каждого последующего опроса. В итоге ваше сканирование может растянуться до бесконечности. Иногда можно обмануть машину на другом конце, рандомизируя порядок сканируемых портов или растягивая интервалы между запросами. Некоторые устройства, возможно, попадутся на эту удочку, другие - нет. Придется поэкспериментировать, чтобы найти работоспособный вариант.

Применение сканеров портов

Когда вы получите разрешение на сканирование, следует определить, с какой целью вы собираетесь сканировать сеть.

Инвентаризация сети

Не знаете точно, сколько машин у вас работает? Хотите узнать IP-адреса всех ваших серверов? Сканеры портов предлагают быстрый способ просмотра диапазона адресов и выявления все активных машин в этом сегменте. Можно даже воспользоваться средством Nlog (рассмотренным далее в этой лекции) для занесения результатов в базу данных и создания полезных отчетов.

Оптимизация сети/сервера

Сканер портов покажет все сервисы, запущенные в данный момент на машине. Если это серверная машина, то, вероятно, таковых окажется много, и, возможно, не все из них на самом деле нужны для выполнения основной функции машины. Помните: чем больше сервисов, тем меньше безопасности. И все эти программы могут замедлять работу перегруженного сервера. Ненужные Web-, FTP- и DNS-серверы крадут циклы процессора у основной функции

компьютера. Сканирование портов серверов с последующим анализом результатов и оптимизацией может дать немедленное увеличение скорости и сокращение времени реакции.

Выявление шпионского ПО, "троянских" программ и сетевых "червей"

Активные web-серферы нередко подцепляют на web-сайтах небольшие программы, которые пытаются отслеживать их поведение или выдавать на их компьютеры специальную всплывающую рекламу. Эти программы называются шпионским ПО, потому что нередко они пытаются следить за активностью пользователя и могут передавать собранные данные обратно на центральный сервер. Эти программы обычно не опасны, но их чрезмерное количество может существенно снизить производительность труда пользователя. Кроме того, написаны они зачастую неаккуратно и могут мешать работе других программ или даже вызывать их аварийное завершение. Они могут также помогать хакерам в поиске уязвимостей.

Другим классом сетевого программного обеспечения, которое вы определенно не хотели бы иметь в своей сети, являются "троянские" программы. Эти программы специально созданы для взлома сетей. Подобно троянскому коню из греческой мифологии, эти программы открывают хакерам и взломщикам заднюю дверь в вашу сеть. Обычно их присутствие можно обнаружить только по открытому сетевому порту, а с помощью антивирусных средств выявить их крайне сложно. Оказавшись внутри компьютера, большинство "троянских" программ пытаются вступить во внешние коммуникации, чтобы дать своему создателю или отправителю знать, что они заразили машину на этих портах. В [табл. 4.2](#) перечислены наиболее распространенные "троянские" программы и их номера портов. Многие номера портов легко распознаваемы по определенному набору цифр (например, для NetBus это 54321, а для Back Orifice - 31337, что в хакерской кодировке читается как "элита"). В целом же троянские программы стремятся использовать порты с большими, необычными, нераспознаваемыми номерами, хотя некоторые действительно хитроумные троянцы пытаются задействовать младшие зарезервированные порты, чтобы замаскироваться под обычные сервисы.

Сетевые "черви" - особо мерзкий тип вирусов. Зачастую они снабжены сетевыми средствами и открывают порты на компьютере-"хозяине". Сетевые "черви" используют сеть для распространения и поэтому иногда выявляются при сканировании портов. Сканирование портов может стать ценным подспорьем в защите от этого вида вирусов.

Таблица 4.2. Порты, используемые наиболее распространенными троянскими программами

| Номер порта | IP протокол | Известные "троянские" программы, использующие эти порты |
|---------------|-------------|---|
| 12456 и 54321 | TCP | NetBus |
| 23274 и 27573 | TCP | Sub7 |
| 31335 | TCP | Trin00 |
| 31337 | TCP | Back Orifice |
| 31785-31791 | TCP | Hack 'a'Tack |
| 33270 | TCP | Trinity |
| 54321 | UDP | Back Orifice 2000 |
| 60000 | TCP | Deep Throat |
| 65000 | TCP | Stacheldraht |

Поиск неавторизованных или запрещенных сервисов

Регулирование того, что сотрудники запускают на своих компьютерах, - трудноразрешимая проблема. Хотя с помощью политик безопасности домена можно ограничить доступ к приводам гибких и компакт-дисков, остается очевидной возможность загрузки программного обеспечения из Паутины. Кроме того, сотрудникам нравится пользоваться сервисами мгновенного обмена сообщениями, такими как ICQ или AOL Instant Messenger, для общения с друзьями, родственниками и другими людьми вне вашей сети. Если разрешить эти сервисы, то необходимо учитывать риски, которые они представляют для безопасности вашей организации. Помимо снижения производительности труда сотрудников и пожирания полосы пропускания, сети мгновенного обмена сообщениями часто оказываются средой для распространения вирусов. Известно также, что в них есть ошибки, позволяющие пользователям

осуществлять доступ к файлам на локальной машине. Даже если вы не разрешаете их официально, их применение бывает трудно отследить. Регулярное сканирование портов раскроет многие подобные сервисы, демонстрируя используемые ими открытые порты.

Существуют еще более вредоносные приложения, которые ваши пользователи могут пытаться запускать, например, программное обеспечение для одноранговой пересылки файлов, позволяющее связаться с тысячами других пользователей во всем мире для разделения файлов с музыкой, фильмами и компьютерными программами. Такое ПО способно потребить всю вашу полосу пропускания, если учесть, что пересылаться могут файлы размером в сотни мегабайт. Реальной становится и угроза судебного преследования вашей организации за нарушение авторских прав. В последнее время крупные медийные компании и программистские концерны все более настойчиво преследуют нелегальных распространителей файлов, а организации являются для них более крупными объектами преследования, чем отдельные люди. Разумеется, применение подобного ПО может открыть внешнему миру внутренности вашей сети, зачастую без явного уведомления сделать часть жесткого диска пользователя доступной другим пользователям. И конечно же, существует множество методов взлома и использования уязвимостей таких программ, позволяющих злоумышленникам пойти значительно дальше. Сухой остаток состоит в нежелательности использования в вашей корпоративной сети однорангового программного обеспечения. И с помощью хорошего сканера портов, к рассмотрению одного из которых мы переходим, можно идентифицировать всех пользователей подобного ПО и отключить их.

Nmap: Разносторонний сканер портов и средство идентификации ОС

Nmap

Автор/основной контакт: Fyodor

Web-сайт: <http://www.insecure.org/nmap>

Платформы: FreeBSD, HP/UX, Linux, Mac OS X, OpenBSD, Solaris, Windows 95, 98, 2000, XP

Лицензия: GPL

Рассмотренная версия: 3.5-1

Списки почтовой рассылки:

Хакеры Nmap:

Отправьте сообщение на nmap-hackers-subscribe@insecure.org

Разработчики Nmap:

Отправьте сообщение на nmap-dev-subscribe@insecure.org

Nmap - вне всяких сомнений, лучший сканер портов. Его главный автор - программист с псевдонимом "Fyodor", разработки которого используются во многих других программах и портированы практически на все употребительные операционные системы. На Nmap опирается сканер уязвимостей Nessus, описанный в [лекции 5](#). Доступно также несколько дополнений, включая программу Nlog, рассматриваемую далее в этой лекции. Достаточно сказать, что Nmap должен входить в инструментарий каждого администратора безопасности. Перечислим некоторые из основных достоинств Nmap:

- У него есть множество опций. Простые сканеры портов доступны с такими средствами, как Sam Spade (см. [лекцию 2](#)), однако Nmap имеет огромное число опций, предоставляющих почти неограниченное число вариантов сканирования сети. Можно понизить частоту отправки зондирующих пакетов, если вы опасаетесь замедления работы сети, или, наоборот, повысить ее, если имеется запас ширины полосы пропускания. Опции невидимости - еще один элемент репертуара Nmap. Хотя некоторые критикуют эти опции, полагая, что они необходимы только хакерам, для них имеются законные применения. Например, если необходимо проверить, насколько чувствительной является система обнаружения вторжений. Nmap позволяет сделать это, выполняя сканирование с различными уровнями невидимости. Далее, Nmap выходит за рамки простого сканирования портов и осуществляет идентификацию ОС, что полезно при установлении соответствия между IP-адресами и машинами. В данном разделе будет рассмотрено большинство основных опций, но всего их так много, что охватить все не представляется возможным.

- Он легкий, но мощный. Код Nmap невелик и будет выполняться даже на самых старых машинах (я постоянно запускаю его на Pentium 133 МГц, ОЗУ 16 МБ и уверен, что он будет работать и на более старых моделях). На самом деле, теперь он запускается даже на некоторых КПК. В небольшом объеме он концентрирует огромную энергию и без проблем сканирует очень большие сети.
- Он прост в использовании. Хотя существует множество различных способов его запуска, реализуемое по умолчанию базовое сканирование SYN делает все, что требуется большинству приложений. Имеется как режим командной строки, так и графический интерфейс для UNIX и Windows, чтобы удовлетворить запросы как круглых дураков, так и тех, кому необходима графика. Он также очень хорошо документирован и поддерживается большим числом разработчиков и оперативных ресурсов.

Установка Nmap в Linux

Если вы работаете в Mandrake, RedHat или SUSE, можно взять файлы с прилагаемого к книге компакт-диска или загрузить бинарный RPM. Чтобы загрузить файлы из Web, наберите в командной строке:

```
rpm -vhU http://download.insecure.org/nmap/dist/nmap/dist/
nmap-3.50-1.i386.rpm
rpm -vhU http://download.insecure.org/nmap/dist/
nmap-frontend-3.50-1.i386.rpm
```

Вам понадобятся два пакета: собственно программа Nmap с интерфейсом командной строки и графическая оболочка для X-Window. Приведенные выше команды загрузят RPM'ы и запустят их. Можно изменить команду, чтобы воспользоваться самой последней версией (уточните на web-сайте имя файла). После выполнения обеих команд вы будете готовы приступить к работе.

Если при выполнении команд возникнут проблемы, или если у вас другой дистрибутив ОС, то придется вручную выполнить компиляцию исходных текстов (см. врезку о компиляции). Это немного сложнее, но не слишком трудно, да и научиться этому полезно, поскольку это придется делать для других средств безопасности из данной книги. Вам будут часто встречаться такие же или похожие команды.

Компиляция исходных текстов: Краткое введение

Многие важнейшие программы UNIX написаны на Си или Си++ из соображений эффективности и мобильности. Это облегчает программистам распространение единой версии исходных текстов и позволяет пользователям компилировать их для своей специфической операционной системы. Большинство систем UNIX поставляются со встроенным Си-компилятором. Си-компилятор с открытыми исходными текстами, используемый в Linux, называется Gcc. Когда требуется построить бинарную программу из исходных текстов, вы вызываете Gcc (при условии, что программа написана на языке Си).

1. В каталоге, где были распакованы исходные тексты программы, наберите

```
./configure имя_программы
```

Запустится программа, которая проверит конфигурацию вашей системы на наличие средств, необходимых программе, и задаст так называемые параметры времени компиляции. С помощью программы configure можно задать некоторые настройки, такие как пропуск определенных частей программы или добавление необязательных элементов. При выполнении `configure` создается конфигурационный файл, называемый `makefile`, который вместе с программой `make` определяет, как и в каком порядке компилятор должен строить объектный код.

2. Выполните команду `make` для компиляции программы:

```
make имя_программы
```

Из исходных текстов она создает бинарный файл, совместимый с вашей конфигурацией. Это может потребовать некоторого времени, зависящего от программы и быстродействия компьютера.

3. Наконец, выполните следующую команду:

```
make install
```

Эта команда устанавливает бинарник, чтобы его можно было запустить на компьютере.

Описанный процесс может немного меняться от программы к программе. Некоторые программы не используют конфигурационный файл и содержат готовый к запуску makefile. У других может несколько отличаться синтаксис команд `make`. В большинстве программ с открытыми исходными текстами в основном каталоге должен лежать файл с именем `INSTALL`. Это текстовый файл, содержащий подробные инструкции по установке программы и всех потенциально полезных опций времени компиляции. Иногда эта информация содержится в файле с именем `README`.

Далее на примере Nmap показан весь процесс компиляции исходных текстов.

- Чтобы скомпилировать Nmap из исходных текстов, выполните в каталоге `nmap` следующие команды:

```
./configure  
make  
make install
```

Отметим, что для выполнения команды `make install` вы должны располагать привилегиями суперпользователя, поэтому не забудьте предварительно набрать `su root` и затем ввести пароль пользователя `root`. Не рекомендуется выполнять первые две команды от имени `root`, так как они могут повредить вашу систему, если в программах есть ошибки или вредоносный код. Необходимо выполнить представленный набор команд для каждой программы - главной программы Nmap и графической оболочки (если только вы не намерены запускать Nmap исключительно из командной строки).

- После того, как вы выполнили RPM или скомпилировали программу, вы готовы к применению Nmap. Запустите графический клиент, набрав

```
nmapfe
```

Если в вашем PATH отсутствует каталог `/usr/local/bin`, введите

```
/usr/local/bin/nmapfe
```

Отобразится основной интерфейсный экран ([рис. 4.1](#)).

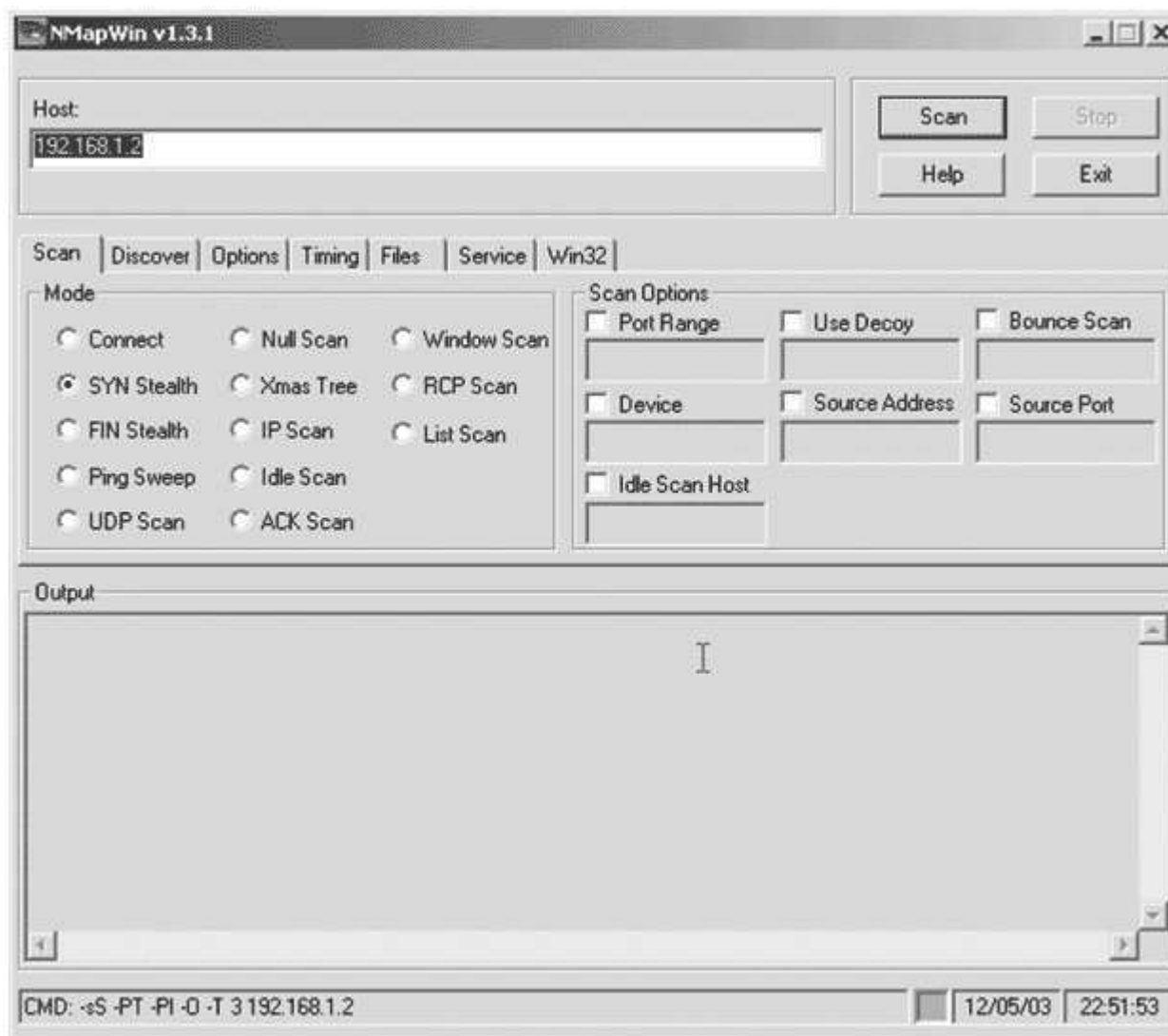


Рис. 4.1. Графический интерфейс Nmap

Совет: На рабочем столе можно создать ссылку на бинарник, чтобы для запуска программы достаточно было двойного щелчка мыши.

Установка Nmap для Windows

Nmap для Windows поддерживает Jens Vogt. Он перенес эту программу в Windows и замечательным образом держится почти вровень с версиями UNIX, отставая всего на один шаг на момент написания этой книги (версия 3.0). Правда, его продукт имеет статус бета-версии, но чего еще желать для открытого ПО? Сканер для Windows не так быстр, как для UNIX, но обладает теми же основными возможностями.

1. Возьмите файл с прилагаемого к книге компакт-диска или загрузите простой исполняемый файл установки для NMapWin с http://download.insecure.org/nmap/dist/nmapwin_1.3.1.exe
2. Если драйвер WinPcap не загружен, его необходимо установить. Если вы не знаете, есть ли он, то его, скорее всего, нет, поскольку он не является стандартным драйвером, включаемым во все версии Windows. Библиотеки WinPcap позволяют Nmap получить низкоуровневый доступ к вашей

сетевой плате, чтобы он мог перехватывать неизменные пакеты стандартным кросс-платформным образом. К счастью, пакет установки NMapWin предоставляет эти файлы. Файл установки WinPcap находится в каталоге files/nmapwin/winpcap.

Имеются две версии WinPcap. Предпочтительно использовать более новую версию, WinPcap 3.1Beta. Если у вас многопроцессорная система, следует использовать ветвь WinPcap 3.X или отключить все процессоры, кроме одного. Если это не помогает, попробуйте более старую версию или возьмите версию, которая будет работать с вашей системой, на сайте WinPcap по адресу <http://winpcap.polito.it/>

WinPcap используется многими другими программами Windows, включая открытое ПО для выявления вторжений и диагностики сети, рассматриваемое в последующих лекциях, поэтому работоспособность WinPcap важна.

Примечание: в настоящее время WinPcap работает ненадежно по коммутируемым соединениям под Windows NT, 2000 и XP. Если вы собираетесь применять сканер портов через коммутируемое соединение (что в любом случае трудно рекомендовать, учитывая ограниченную полосу пропускания для отправки зондирующих пакетов), то придется найти иное решение.

3. После установки WinPcap необходимо перезагрузить систему, чтобы все драйверы заработали. Затем запускайте NmapWin - и можно начать сканирование.

Сканирование сетей с помощью Nmap

Графический клиент Nmap предоставляет весьма простой интерфейс ([рис. 4.2](#)). Вверху имеется поле для ввода IP-адреса или диапазона IP-адресов, а чтобы начать сканирование, достаточно нажать кнопку Scan.

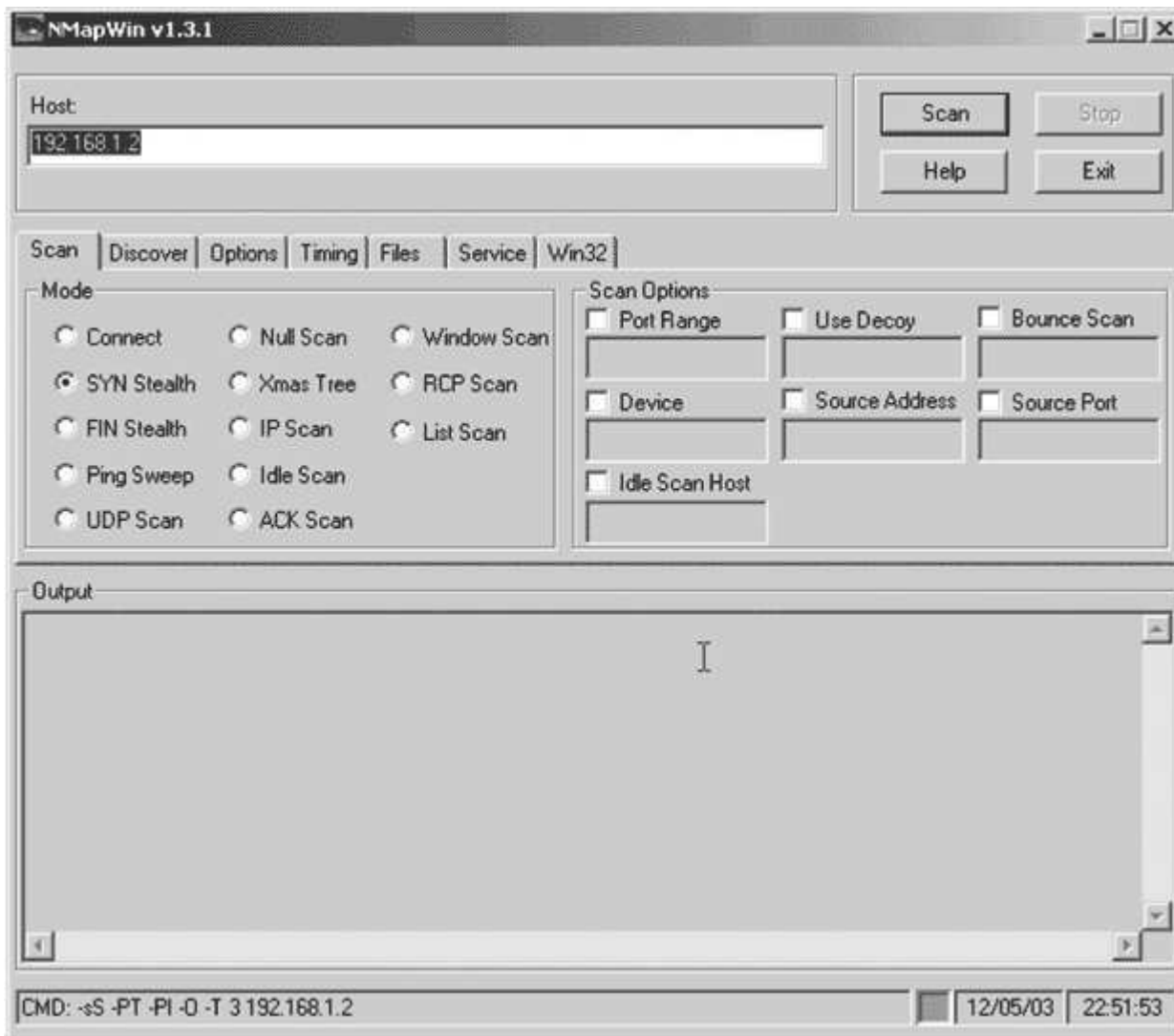


Рис. 4.2. Образ экрана NmapWin

В [табл. 4.3](#) приведены различные форматы для ввода IP-адресов. Адреса могут также извлекаться из файла, если выбрать пункт Input элемента File основного меню и задать текстовый файл с данными в подходящем для Nmap формате ([рис. 4.1](#)).



Флэми Тех учит:

Сетевые маски и нотация с косой чертой.

Вам будут часто встречаться обозначения IP-сетей или с сетевой маской, или с косой чертой и числом после нее. Это - два способа задать размер сети. Для их понимания необходимо представлять себе структуру IP-адреса. Стандартный адрес IPv4 состоит из 32 бит. Его

обычно представляют в виде четырех частей - восьмибитных октетов. Октеды для удобочитаемости обычно преобразуют в десятичные числа. Поэтому, если вы видите 192.168.1.1, то компьютер видит

11000000 10101000 00000001 00000001

Маска сети обычно представляет собой набор из четырех чисел. Она показывает, где кончается локальная сеть и начинается глобальная. Обычно маска выглядит примерно так:

255.255.255.0

Чтобы определить размер сети, представленной сетевой маской, достаточно вычесть каждый октет из 256 и перемножить полученные разности. Например, сетевая маска 255.255.255.248 описывает восьмиэлементную IP-сеть, поскольку

$$(256-255) * (256-255) * (256-255) * (256-248) = 8.$$

Сетевая маска 255.255.255.0 представляет IP-сеть из 256 узлов, так как

$$(256-255) * (256-255) * (256-255) * (256-0) = 256.$$

Наконец, сетевая маска 255.255.0.0 описывает сеть из 65536 IP-адресов, ибо

$$(256-255) * (256-255) * (256-0) * (256-0) = 65536.$$

Нотация с косой чертой чуть сложнее для понимания, но идея остается прежней. Число после косой черты показывает, сколько бит описывают глобальную сеть. Вычитая это число из 32, получаем число бит, описывающих локальную сеть. Например, запись 192.168.0.0/24 представляет сеть, начинающуюся с 192.168.0.0 и насчитывающую 256 IP-адресов. (Это такой же размер, как и у рассмотренной выше сети с маской 255.255.255.0.)

32 бита IP-адреса минус 24 бита для префикса сети дает 8 бит для локального использования, то есть 256 возможных адресов. Если вас мутит от двоичных чисел, просто воспользуйтесь для запоминания представленной ниже небольшой таблицей.

| Нотация с косой чертой | Размер сети |
|------------------------|----------------|
| /24 | 256 IP-адресов |
| /25 | 128 IP-адресов |
| /26 | 64 IP-адресов |
| /27 | 32 IP-адресов |
| /28 | 16 IP-адресов |
| /29 | 8 IP-адресов |
| /30 | 4 IP-адреса |
| /31 | 2 IP-адреса |
| /32 | 1 IP-адрес |

Таблица 4.3. Форматы IP адресов

| Формат | Пример |
|----------------------------------|-------------------------|
| Одиночный IP-адрес | 192.168.0.1 |
| IP- адреса, разделенные запятыми | 192.168.0.1,192.168.0.2 |

| | |
|--|---|
| IP-диапазон, разделенный дефисом | 192.168.0.1-255 |
| Использование стандартной нотации с косой чертой | 192.168.0.1/24 (сеть класса C из 256 адресов) |

Запуск Nmap из командной строки

Nmap можно запустить из командной строки как в UNIX, так и в Windows. Общий формат таков:

```
nmap параметры IP-диапазон
```

с любыми дополнительными настройками, заданными значениями параметров. Далее до конца лекции вслед за настройками или опциями графического интерфейса в скобках будут указываться эквивалентные опции для командной строки, например, SYN (-sS) и Bounce Scan (-n FTP_HOST).

Типы сканирования в Nmap

Nmap поддерживает множество различных видов сканирования. В табл. 4.4 перечислены наиболее употребительные. Указаны также параметры командной строки, если вы захотите использовать этот интерфейс.

Таблица 4.4. Типы сканирования в Nmap и параметры командной строки

| Тип сканирования (параметры командной строки) | Описание |
|---|---|
| SYN (-sS) | Подразумеваемый тип сканирования, пригодный для большинства целей. Он менее заметен, чем TCP Connect, то есть не будет фиксироваться большинством простых средств протоколирования. В этом режиме в каждый возможный порт посылаются одиночные TCP-пакеты с установленным флагом SYN. Если в ответ возвращается пакет SYN ACK, то Nmap делает вывод, что здесь запущен сервис. Если ответа нет, то предполагается, что порт закрыт SYN-сканирование не завершает трехходовое квитирование установления связи в TCP, так как не возвращает целевой машине пакет с установленным флагом ACK; с точки зрения сканируемой системы действующие соединения не устанавливаются. Однако, удаленная система будет удерживать эту "половинку сокета" открытой, пока не пройдет максимально допустимое время ответа. Некоторые современные серверы и программы выявления вторжений достаточно интеллектуальны, чтобы уловить подобные действия, но для большинства машин SYN-сканирование будет невидимым |
| TCP-соединение: Connect (-sT) | Это тип сканирования напоминает SYN за исключением того, что трехходовое квитирование установления связи в TCP выполняется до конца и устанавливается полноценное соединение. Подобное сканирование не только шумно, но и создает дополнительную нагрузку на сканируемые машины и сеть. Однако, если скрытность или экономия полосы пропускания не являются приоритетными, то сканированием Connect, по сравнению с SYN, можно порой получить более точные результаты. Кроме того, если у вас нет привилегий администратора или суперпользователя на машине Nmap, вы не сможете воспользоваться никаким другим типом сканирования, поскольку создание построенных особым образом пакетов для других типов сканирования требует низкоуровневого доступа к ОС |
| Эхо-тестирование: Ping Sweep (-sP) | Выполняется простое эхо-тестирование всех адресов, чтобы увидеть, какие из них ответят на ICMP-запрос. Если вас на самом деле не интересует, какие сервисы запущены, и вы просто хотите знать, какие IP-адреса активны, то данный тип позволит достичь цели много быстрее, чем полное сканирование портов. Однако некоторые машины могут быть сконфигурированы так, чтобы не отвечать на ping (например, новый межсетевой экран XP), но, тем не менее, выполнять некоторые сервисы, поэтому Ping Sweep - менее надежный метод, чем полное сканирование портов |
| UDP-сканирование: UDP Scan (-sU) | Этот тип сканирования проверяет наличие слушаемых UDP-портов. Так как UDP, в отличие от TCP, не отвечает положительным подтверждением, а отвечает на входящий пакет, только когда порт закрыт, данный тип сканирования может иногда приводить к ложным срабатываниям, однако он способен выявить троянские программы, использующие UDP- |

| | |
|--|---|
| | порты с большими номерами и скрытые RPC-сервисы. Он может быть весьма медленным, так как некоторые машины намеренно замедляют ответы на этот тип трафика, чтобы избежать перегрузки. Однако машины, выполняющие ОС Windows, не реализуют замедления, поэтому вы сможете использовать UDP для нормального сканирования хостов Windows. |
| FIN-сканирование: FIN Stealth (<code>-sF</code>) | Это скрытное сканирование, аналогичное SYN, но использующее пакеты TCP FIN. Большинство компьютеров, но не все, ответят пакетом RST, поэтому сканирование FIN сопряжено с ложными срабатываниями и пропуском положительных результатов, но может осуществляться под наблюдением некоторых программ выявления вторжений и при наличии других контрмер |
| NULL-сканирование: NULL Scan (<code>-sN</code>) | Еще одно весьма скрытное сканирование, при котором все флаги заголовка TCP сброшены (или пусты). Подобные пакеты обычно некорректны, и некоторые хосты не знают, что с ними делать. Операционные системы Windows входят в эту группу, так что их сканирование в режиме Null будет давать недостоверные результаты. Однако для серверов не под Windows, защищенных межсетевым экраном, оно может стать способом проникновения |
| XMAS-сканирование: XMAS Tree (<code>-sX</code>) | Аналогично сканированию NULL, за исключением того, что все флаги в заголовке TCP установлены, а не сброшены (отсюда и название - пакет расцвечен, как рождественская елка). Машины Windows, ввиду особенностей реализации на них стека TCP, не отвечают на подобные пакеты |
| Сканирование через отражатель: Bounce Scan (<code>-nFTP_HOST</code>) | Этот хитроумный тип сканирования использует лазейку в протоколе TCP для "отражения" сканирующих пакетов от сервера FTP во внутреннюю сеть, которая обычно недоступна. Зная IP-адрес сервера FTP, который подключен к локальной сети, вы можете проникнуть через межсетевой экран и сканировать внутренние машины. Стоит проверить и свою собственную сеть на наличие данной уязвимости. В большинстве современных серверов FTP эта дыра в защите ликвидирована. Примечание: В дополнение к сканируемым IP-адресам вы должны задать действующий сервер FTP, имеющий доступ к сети |
| RPC-сканирование: RPC Scan (<code>-sR</code>) | Этот особый тип сканирования ищет машины, отвечающие сервисам удаленного вызова процедур (RPC). Сервис RPC, при определенных условиях позволяющий удаленным командам выполняться на машине, сопряжен со значительным риском. Так как сервисы RPC могут выполняться на многих различных портах, то по результатам обычного сканирования выявить эти порты трудно. RPC-сканирование зондирует найденные открытые порты с помощью команд, показывающих имя программы и версию сервиса RPC. Неплохо время от времени проводить подобное сканирование, чтобы узнать, работают ли, и где именно, RPC-сервисы |
| Window-сканирование: Window Scan (<code>-sW</code>) | Данный тип сканирования полагается на аномалию в ответах на пакеты ACK в некоторых операционных системах, чтобы обнаружить порты, которые предположительно фильтруются. Известно, что к числу операционных систем, уязвимых для подобного сканирования, принадлежат некоторые версии AIX, Amiga, BeOS, BSDI, Cray, DG/UX, Digital UNIX, FreeBSD, HP/UX, IRIX, MacOS, NetBSD, OpenBSD, OpenStep, OpenVMS, OS/2, QNX, Rhapsody, SunOS 4.X, Tru64 UNIX, Ultrix, VAX и VxWorks |
| Реактивное сканирование: Idle Scan (<code>-sI хост-зомби:используемый_порт</code>) | Данный тип сканирования появился в Nmap версии 3.0. Это сверхскрытный метод, при применении которого пакеты сканирования отражаются от внешнего хоста. Необязательно иметь контроль над этим хостом, но он должен работать и удовлетворять некоторым требованиям. Вы должны ввести IP адрес хоста-зомби и номер используемого порта. Хотя это сканирование крайне трудно проследить до исходной точки, оно вряд ли особенно полезно для большинства администраторов, сканирующих свои собственные сети. Это одна из самых спорных опций Nmap, так как на практике она применима только для злоумышленных атак |

Опции раскрытия для Nmap

Можно настроить способ, которым Nmap выполняет раскрытие сетей и определяет, какие хосты работают. В [табл. 4.5](#) перечислены несколько различных вариантов.

Опции времени для Nmap

Nmap предоставляет средства для повышения или понижения частоты, с которой посылаются пакеты сканирования. Если вас беспокоит слишком большой сетевой трафик (или вы пытаетесь действовать скрытно), то можно понизить частоту. Помните только, что чем реже посылаются пакеты, тем дольше продлится сканирование. Для больших сетей время может вырасти экспоненциально. С другой стороны, если вы торопитесь и не обращаете внимание на

некоторый дополнительный сетевой трафик, можно поднять частоту. Различные уровни и частоты пакетов приведены в [табл. 4.6](#). В версии для Windows или с помощью опций командной строки можно устанавливать специальные частоты.

Таблица 4.5. Опции раскрытия для Nmap

| Опция | Описание |
|-----------------------------|---|
| TCP + ICMP (-PB) | Подразумеваемая настройка. Nmap обычно использует для определения статуса хоста и ICMP, и TCP-пакеты. Это наиболее надежный и точный способ, так как, если хост активен, то хотя бы по одному методу ответ, как правило, будет получен. К сожалению, это также самый шумный способ, который, скорее всего, приведет к регистрации каким-нибудь устройством сканируемой сети |
| Эхо-тестирование TCP (-PT) | Для обнаружения хостов используется только метод TCP. Многие межсетевые экраны и некоторые маршрутизаторы отбрасывают пакеты ICMP, возможно, с протоколированием. Если вы пытаетесь остаться невидимым, то метод TCP - это наилучший вариант. Однако для некоторых экзотических типов сканирования (FIN, XMAS, NULL) какие-то хосты могут остаться незамеченными. |
| Эхо-тестирование ICMP (-PE) | Использовать для раскрытия сети только пакеты ICMP. Это не лучший вариант, если вы сканируете сеть извне через межсетевой экран, так как большинство ваших пакетов будет, вероятно, отброшено. Однако внутри сети данный метод вполне надежен, хотя вы можете пропустить свой межсетевой экран и некоторые сетевые устройства, которые не отвечают на ICMP-пакеты |
| Без эхо-тестирования (-PO) | Если задается эта опция, то Nmap не будет пытаться сначала выяснить, какие хосты активны, а будет вместо этого посылать пакеты по каждому IP-адресу заданного диапазона, даже если по этому адресу машины нет. Это расточительно как с точки зрения полосы пропускания, так и времени, особенно когда сканируются большие диапазоны. Однако это может быть единственным способом просканировать хорошо защищенную сеть, которая не отвечает на ICMP-пакеты. |

Таблица 4.6. Параметры Nmap для управления частотой посылки пакетов

| Уровень частоты | Параметр командной строки | Частота пакетов | Пояснения |
|-----------------|---------------------------|---|--|
| Параноидальный | -F 0 | Раз в 5 минут | Не используйте эту опцию при сканировании большого числа хостов, иначе сканирование никогда не закончится. |
| Исподтишка | -F 1 | Раз в 15 секунд | |
| Вежливый | -F 2 | Раз в 4 секунды | |
| Нормальный | -F 3 | Со скоростью работы ОС | Используется по умолчанию |
| Агрессивный | -F 4 | То же, что и Normal, но максимальное время ожидания пакета сокращено до 5 минут на хост и до 1,25 секунды на зондирующий пакет. | |
| Безумный | -F 5 | Время ожидания 0,75 секунды на хост и 0,3 секунды на зондирующий пакет. | Этот метод не будет хорошо работать, если только вы не находитесь в очень быстрой сети и не используете очень быстрый сервер Nmap. Даже в этом случае есть риск потерять данные. |

Другие опции Nmap

В [табл. 4.7](#) перечислены некоторые другие опции Nmap, которые управляют, например, разрешением доменных имен, идентификацией ОС и т.д., и не попадают в другие категории

Существуют дополнительные опции тонкой настройки сканирования, доступные из командной строки. Подробности можно найти в оперативной справке Nmap.

Запуск Nmap в качестве службы

По умолчанию в Windows-версии Nmap запускается как служба. Это означает, что он постоянно выполняется в фоновом режиме и может вызываться другими программами, запускаться из командных файлов или заданий cron. В Windows служба Nmap управляется и конфигурируется в меню Services Tool. Для этого в меню Control Panel выберите Administrative Tools, а затем Services. Вы увидите Nmap в списке служб; можно щелкнуть на нем мышью и сконфигурировать его свойства.

Эта возможность полезна, если вы хотите, чтобы Nmap выполнял сканирование на регулярной основе. Можно настроить Nmap для сканирования вашей сети раз в неделю или раз в месяц с представлением отчетов. Можно сканировать только серверы, чтобы не пропустить значительных изменений. Если вы не планируете использовать перечисленные возможности, лучше отключить эту службу в Windows, чтобы сэкономить ресурсы и повысить безопасность. Это можно сделать, щелкнув мышью на службе Nmap в окне просмотра служб и заменив Start-up Type (тип запуска) с Automatic (автоматический) на Manual (вручную). Это изменение вступит в силу после перезагрузки машины. Можно также вручную остановить службу, щелкнув мышью на кнопке Stop.

Таблица 4.7. Прочие опции Nmap

| Опция | Описание |
|--|---|
| Не выполнять разрешение имен (-n) | Обычно Nmap пытается разрешать доменные имена для всех сканируемых IP-адресов. Это может существенно затягивать сканирование, поэтому если вас не интересуют имена хостов, разрешение имен можно отключить. Помните, однако, что знать имена хостов полезно, особенно при сканировании сетей с DHCP, где IP-адреса могут меняться. |
| Быстрое сканирование (-F) | Эта опция вызывает сканирование только портов, перечисленных в файлах употребительных портов Nmap. По умолчанию это общеупотребительные серверные порты с номерами, меньшими 1024. Данные файлы можно отредактировать и добавить в список другие порты. Подобное сканирование может оказаться значительно более быстрым, но оно не выявит троянские программы и сервисы, использующие порты с большими номерами. |
| Диапазон портов (-p диапазон_портов) | По умолчанию Nmap сканирует все 65535 возможных портов TCP. Однако, если вы хотите просканировать только определенный диапазон, можно задать его в качестве аргумента опции -p. Это полезно, если вы хотите просканировать только один тип серверов, например, порт 80 для Web-серверов, или только верхние диапазоны, чтобы найти необычные сервисы и потенциальные троянские программы. |
| Использование приманок (-D адрес_приманки_1, адрес_приманки_2 . . .) | Эта опция создает видимость, что хосты, указанные в качестве приманок, участвуют в сканировании целевых машин. Последние будут наблюдать потоки данных из нескольких источников, и им будет трудно определить, какой из них является реальным сканирующим хостом. Это еще одна опция сверхскритности, не обязательная для большинства добропорядочных применений и создающая, кроме того, существенно более высокую нагрузку на сеть. Следует учитывать также, что использование хостов в качестве приманок может привести к блокированию их доступа к сканируемой машине. На вас может обрушиться гнев людей, которых вы таким образом "подставили". |
| Фрагментация (-f) | Данная опция вызывает фрагментацию отправляемых пакетов сканирования. Это - средство обеспечения скрытности, которое можно применять, чтобы избежать обнаружения сканирования. Пакеты будут собираться на другом конце получающей их машиной, но фрагментированные пакеты могут обмануть системы обнаружения вторжений и межсетевые экраны, которые зачастую проверяют соответствие конкретным шаблонам. |
| Запрашивать информацию Idendt (-I) | Служба Idendt функционирует на некоторых (обычно - UNIX) машинах и предоставляет при запросе дополнительную информацию о хосте, например, тип операционной системы. Следует учитывать, что Nmap автоматически выполняет идентификацию ОС с помощью идентификационных меток TCP, поэтому данная опция менее полезна, чем кажется на первый взгляд. Если в вашей сети нет систем UNIX, то применение этой опции вообще теряет смысл. |
| Разрешать все адреса (-R) | При использовании данной опции Nmap пытается разрешать все адреса в диапазоне, даже когда они не отвечают. Это может быть полезно, например, в сети поставщика Интернет-услуг, где целый диапазон записей о хостах может быть присвоен потенциальным IP-адресам для пула коммутируемого доступа, но в каждый момент времени возможно использование только определенной части из них. |
| Идентификация ОС (-O) | Подразумеваемая опция. Как упоминалось ранее, каждая реализация стека TCP имеет свои особенности. При сравнении точной идентификационной метки ответов с базой данных известных идентификационных меток TCP, Nmap, как правило, может с высокой достоверностью (иногда - вплоть до диапазона версий) идентифицировать ОС, с которой общается. |

| | |
|---|--|
| | Изредка попадает что-то незнакомое, и тогда ответ TCP печатается внизу отчета. Если вы обнаружите неопределенную сигнатуру, вы сможете помочь в построении базы данных идентификационных меток ОС. Если вы точно знаете, чему она соответствует, скопируйте ее и отправьте по электронной почте на адрес группы разработчиков Nmap. Они добавят ее в базу данных, чтобы в будущем при сканировании машины такого типа ее можно было правильно идентифицировать. Все известные Nmap идентификационные метки TCP содержатся в файле nmap-os-fingerprints в каталоге Data установки Nmap. |
| Отправить через интерфейс (-e имя_интерфейса) | Эта опция заставляет пакеты сканирования отправляться через определенный интерфейс. На практике это необходимо только на машине с несколькими сетевыми платами, или если Nmap не опознает ваш сетевой интерфейс автоматически. |



Флэми Тех советует:

Дружественное сканирование Nmap!

Как упоминалось ранее, Nmap может вызывать проблемы в сетях при некорректном или неаккуратном применении. Вот несколько советов, которые помогут сделать сканирование безопасным:

- Тщательно выбирайте исходную точку сканирования. Сканирование изнутри сети даст значительно больше информации, чем сканирование извне, через межсетевой экран. Поучительно выполнить сканирование обоих видов и сравнить результаты. Не страшно, если открытый серверный порт виден изнутри сети; гораздо опаснее, если он виден извне.
- Сканирование целесообразно выполнять рано утром или поздно вечером. Таким образом вы минимизируете вероятность замедления работы жизненно важных серверов и пользовательских машин.
- Если вы беспокоитесь о перегрузке своей сети, установите в сканирующую машину старую сетевую карту на 10 Мбит/с или подключите ее через концентратор на 10 Мбит/с. Таким образом максимальный трафик, который сканирование может создать в сети, не превысит 10 Мбит/с, что вряд ли перегрузит сеть на 100 Мбит/с.

Вывод результатов Nmap

Nmap генерирует отчет, содержащий каждый обнаруженный IP-адрес, выявленные слушающие порты по этим адресам и соответствующие общеизвестные имена сервисов (при наличии таковых). Отчет также показывает, является ли порт открытым, фильтруемым или закрытым. Строго говоря, тот факт, что Nmap получил ответ из порта 80 и напечатал в отчете "http", еще не означает, что на компьютере запущен Web-сервер, хотя, скорее всего, это так. Всегда можно проверить любой подозрительный открытый порт, подключаясь с помощью telnet к нужному IP-адресу с указанием номера порта и анализируя полученный ответ. Если там выполняется web-сервер, то обычно получают ответ, вводя команду GET / HTTP. Должна быть выдана подразумеваемая домашняя страница в необработанном HTML-виде (а не как красивая Web-страница), что послужит подтверждением функционирования сервера. То же самое можно проделать с другими сервисами, такими как FTP и SMTP. Отметим, что в UNIX-версии Nmap кодирует цветом найденные порты в соответствии с их ролью ([табл. 4.8](#)).

Как можно видеть из [рис. 4.3](#), формат вывода позволяет просмотреть отчет и быстро определить, есть ли какие-то сервисы или порты, о которых следует побеспокоиться. Это не означает, что нужно игнорировать все необычные номера, которые не выделены цветом или шрифтом (в версиях UNIX). Троянские программы и ПО для общения часто отображаются как неизвестные сервисы, но вы можете поискать таинственный порт в списке общеупотребительных портов в приложении С или проверить его по списку известных плохих портов, чтобы быстро определить, требует ли он особого внимания. Если его нет в списках, то странный сервис, не использующий общеизвестные номера портов, должен вас насторожить.

Таблица 4.8. Цветовое кодирование вывода Nmap

| Цвет | Описание |
|---------|--|
| Красный | Данный номер порта присвоен сервису, который предлагает некоторую форму прямого входа в систему (как, например, Telnet или FTP). Зачастую эти сервисы оказываются наиболее притягательными для хакеров |

| | |
|-------------------|---|
| Голубой | Этот номер порта представляет почтовый сервис, такой как SMTP или POP. Подобные сервисы также часто являются объектами хакерских атак |
| Жирный черный | Эти сервисы могут предоставлять некоторую информацию о машине или операционной системе (как, например, finger, echo и т.д.) |
| Простой черный | Любые другие идентифицированные сервисы или порты |

```

Output
Starting nmap V. 3.00 ( www.insecure.org/nmap )
Interesting ports on (192.168.1.3):
(The 1597 ports scanned but not shown below are in state: closed)
Port      State      Service
22/tcp    open      ssh
111/tcp   open      sunrpc
1024/tcp  open      kdm
1241/tcp  open      msg
Remote operating system guess: Linux Kernel 2.4.0 - 2.5.20
Uptime 0.986 days (since Mon May 12 00:30:09 2003)
Nmap run completed -- 1 IP address (1 host up) scanned in 8 seconds

```

Рис. 4.3. Вывод Nmap

Журналы Nmap можно сохранять в различных форматах, включая обычный или машиночитаемый текст, и импортировать их в другую программу. Однако, если этих возможностей для вас недостаточно, то обсуждаемое далее средство Nlog может помочь придать смысл выводу Nmap. На очень больших сетях его использование может оказаться просто спасением, так как просмотр сотен страниц вывода Nmap в поисках злоумышленников может быстро сделать вас слепым, сумасшедшим или и тем, и другим.

Nlog: Средство сортировки и организации вывода Nmap

Nlog

Автор/основной контакт: H.D. Moore

Web-сайт: <http://www.secureaustin.com/nlog/>

Платформы: Большинство Linux-платформ

Лицензия: Без лицензии (подобно GPL)

Рассмотренная версия: 1.6.0

Программа Nlog помогает организовать и проанализировать вывод Nmap. Она представляет его в настраиваемом web-интерфейсе с использованием CGI-процедур. Nlog облегчает сортировку данных Nmap в единой базе данных с возможностью поиска. В больших сетях такая возможность жизненно важна, она делает Nmap действительно полезным. Остин Х.Д. Мур собрал эти программы воедино и сделал их доступными вместе с другими интересными проектами на своем web-сайте <http://www.secureaustin.com/>.

Программа Nlog расширяема: можно добавлять другие процедуры, чтобы предоставлять больше информации и запускать дополнительные тесты на обнаруживаемых открытых портах. Автор предлагает несколько таких дополнений и инструкции по созданию новых. Nlog опирается на Perl и работает с файлами журналов, сгенерированных Nmap версии 2.0 и выше.

Установка Nlog

Для установки и подготовки Nlog следуйте представленным ниже инструкциям.

1. Возьмите файлы с компакт-диска, прилагаемого к этой книге, или загрузите их с web-сайта Nlog.
2. Распакуйте файлы Nlog с помощью команды `tar -zxvf`. Она распакует и аккуратно разложит все файлы для Nlog в каталоге с именем `nlog-1.6.0` (цифры могут быть другими, они зависят от версии).
3. Можно воспользоваться командным файлом, предоставленным для автоматической установки и подготовки программы. Отметим, что перед запуском командный файл необходимо отредактировать. Перейдите в каталог Nlog и с помощью текстового редактора, такого как `vi` или EMACS, откройте файл `installer.sh` и, где требуется, задайте значения переменных, подходящие для вашей системы.

Отредактируйте следующие параметры, задав корректные для своей установки значения.

```
CGIDIR=/var/www/cgi/  
HTMLDIR=/var/www/
```

Задайте маршрут к CGI-каталогу. Выше приведены значения для подразумеваемой установки в Mandrake. Не забудьте ввести значения, корректные для вашей системы. Для других систем Linux выясните маршрут к этому каталогу, воспользовавшись командой `locate`. Эта полезная команда найдет любые файлы, содержащие введенный после нее текст.

4. Сохраните файл, а затем выполните его, набрав:

```
./install.sh
```

Командный файл установки автоматически скопирует CGI-файлы в CGI-каталог и основной файл HTML в HTML-каталог. Он также изменит права доступа к этим файлам, чтобы они могли исполняться Web-навигатором.

5. В качестве конечного шага перейдите в HTML-каталог и отредактируйте файл `nlog.html`. Измените инструкцию POST, указав в ней ссылку на ваши файлы `cgi`. Ссылка должна совпасть с приведенной выше (`/var/www/cgi/`). Сохраните файл. Теперь все готово к работе.



Флэми Тех советует:

Рекомендации по использованию текстовых редакторов в UNIX

При изучении этой книги вам понадобится редактировать текстовые файлы для задания программных переменных, конфигураций установок и для других целей. Имеется множество хороших текстовых редакторов для UNIX, включая `vi`, EMACS и Pico. У каждого из них есть свои достоинства и недостатки, но в этой книге я буду предполагать использование EMACS, так как это самый дружелюбный для X-Window, простой в использовании и доступный на большинстве систем редактор. В Mandrake Linux EMACS располагается в X-Window в меню Start, подменю Programming. EMACS можно запустить из командной строки, набрав `emacs` или `emacs имя_файла`, чтобы отредактировать файл с заданным именем.

Будьте осторожны при использовании текстовых редакторов для исполнимых или бинарных файлов. Любые изменения, сделанные в этих файлах, могут разрушить программу, которую они поддерживают. Бинарный файл можно распознать по нечитаемому содержимому. Обычно текстовые редакторы используют только для модификации текстовых файлов.

EMACS предоставляет сверху привычное меню для выбора операций с файлом, таких как сохранение и закрытие. Для перемещения по экрану и выбора меню или текста можно использовать мышь. Можно применять также ряд клавишных сокращений. Наиболее полезные из них перечислены

ниже. Примечание: "Ctrl+" означает нажатие и удержание клавиши Ctrl при нажатии следующей клавиши. Там, где перечислены две клавишные комбинации, набирайте их по очереди.

| Клавишные сокращения EMACS | Функции |
|--|--|
| CTRL+x, CTRL+c | Закрывает EMACS. Предлагает сохранить текущий файл, если это еще не сделано. |
| CTRL+g | Выход. Если вы выполняете клавишную комбинацию, из которой не можете выйти, то это сокращение вернет вас в основной буфер. |
| CTRL+x, k | Закрывает текущий файл. |
| CTRL+x, s | Сохраняет текущий файл. |
| Ctrl+x, d | Открывает список каталогов, на котором можно при помощи мыши открывать файлы и выполнять другие действия. |
| CTRL+a | Перемещает курсор в начало строки. |
| CTRL+e | Перемещает курсор в конец строки. |
| CTRL+s | Поиск введенного текста. |
| Имеется множество других клавишных комбинаций и макросов для продвинутых пользователей. Дополнительную информацию о EMACS можно найти на следующих сайтах: | |
| Домашняя страница EMACS: http://www.gnu.org/software/emacs/ | |
| Краткий справочник EMACS: http://seamons.com/emacs/ | |

Использование Nlog

В этом разделе описано использование Nlog.

1. Прежде всего, необходимо создать для просмотра файл базы данных Nlog. Для этого нужно преобразовать существующий журнал Nmap. Не забудьте предварительно сохранить журналы Nmap в машиночитаемом виде (опция `-m` в командной строке), чтобы можно было использовать их в Nlog. Затем можно воспользоваться процедурой, поставляемой вместе с Nlog, для преобразования журнала Nmap в формат базы данных Nlog. Чтобы преобразовать машиночитаемый журнал Nmap, выполните процедуру `log2db.pl`, набрав в командной строке:

```
log2db.pl журнал_Nmap
```

Замените журнал_Nmap маршрутным именем файла журнала.

2. Чтобы объединить несколько журналов в одну базу данных, выполните команду вида

```
cat каталог_журналов_Nmap/* | sort -u > сводный_журнал_Nmap
```

3. Замените каталог_журналов_Nmap на маршрутный префикс журнальных файлов Nmap и `final.db` на имя файла, в который вы хотите свести имеющиеся журналы. Утилита `sort` отсортирует данные в алфавитном порядке и исключит повторения.
4. Запустите web-навигатор и перейдите в HTML-каталог (`/var/www/` из предыдущего раздела).
5. Выберите файл базы данных Nlog, который вы хотите просмотреть, и щелкните мышью на Search ([рис. 4.4](#)).

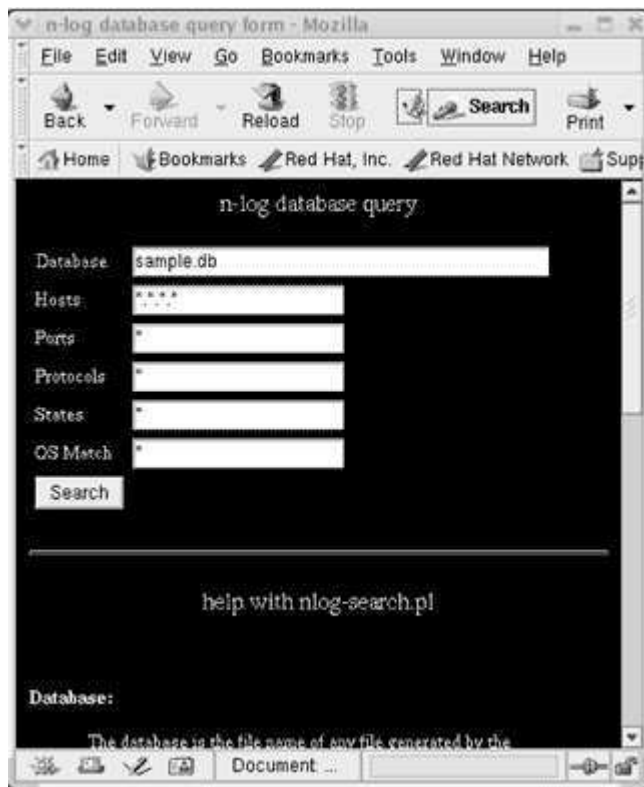


Рис. 4.4. Образ экрана Nlog

6. Теперь можно открыть базу данных Nlog и выполнить поиск по следующим критериям:

- Хосты по IP-адресу;
- Порты по номеру;
- Протоколы по имени;
- По состоянию порта (открытый, закрытый, фильтруемый);
- По выявленной ОС.

Можно также использовать любую комбинацию этих критериев. Например, можно искать по любым Web-серверам (протокол http) на системах Windows с состоянием порта "открытый".

Дополнения для Nlog

Как упоминалось ранее, Nlog легко расширяется. Можно писать дополнения для выполнения других проверок или функций на всех найденных протоколах или портах. В действительности имеется несколько дополнений, включаемых с вместе программой. Если есть доступное дополнение, то вслед за портом будет располагаться строка гипертекста, и можно щелкнуть на ней мышью, чтобы выполнить подпрограмму. В [табл. 4.9](#) перечислены встроенные расширения.

Таблица 4.9. Встроенные расширения Nlog

| Расширение | Описание |
|-------------|--|
| Nlog-rpc.pl | Это дополнение для каждого найденного сервиса RPC пытается определить, есть ли к нему какие-либо текущие присоединения и экспорты. |

| | |
|----------------|---|
| Nlog-smb.pl | Для всех узлов, выполняющих NetBIOS (такowymi будут большинство машин Windows), делается попытка выборки общих ресурсов, списков пользователей и любой другой доступной доменной информации. Используются имена пользователей, заданные в файле nlog-config.ph. |
| Nlog-dns.pl | Для IP-адресов выполняется стандартная команда <code>nslookup</code> . (Подробнее о <code>nslookup</code> см. в лекции 2) |
| Nlog-finger.pl | Выполняются запросы к выявленным службам <code>finger</code> , чтобы посмотреть, какая информация будет выдана |

Создание собственных расширений Nlog

Если вы изучите дополнительные процедуры, то увидите, что это обычные программы на языке Perl. Если у вас есть опыт работы с Perl, вы в состоянии писать собственные расширения, позволяющие выполнять для просканированных хостов практически произвольную функцию. Например, можно извлечь и вывести HTTP-заголовки для всех обнаруженных web-серверов, чтобы было легче их идентифицировать. Не стоит слишком этим увлекаться, поскольку такие программы, как Nessus (рассматривается в [лекции 5](#)), могут производить значительно более обстоятельные проверки, но если вам нужен только заголовок или какой-то небольшой фрагмент информации, то применение Nlog будет хорошим решением.

Nlog поставляется с образцом пользовательского дополнения, названным `nlog-bind.pl`. Эта процедура предназначена для опроса сервера DNS и выдачи используемой версии BIND (Berkeley Internet Name Domain), однако она не закончена и служит лишь в качестве упражнения для создания собственных дополнений. Образец находится в каталоге `/nlog*/extras/bind/`. Следующие шаги помогут вам дописать процедуру. Этот подход можно использовать для самостоятельного создания любой пользовательской процедуры.

1. Перейдите в упомянутый каталог и скомпилируйте процедуру с помощью компилятора Gcc, используя следующую команду:

```
gcc -o bindinfo binfo-udp.c
```

В каталоге будет создан бинарный файл `bindinfo`.

2. Скопируйте этот файл в каталог, где хранятся ваши Nlog-процедуры.
3. Измените у него режим доступа, сделав его исполнимым (помните, что для выполнения этой команды вы должны быть суперпользователем):

```
chmod 700 bindinfo
```

4. Откройте файл `nlog-config.ph` в текстовом редакторе.
5. Добавьте строку вида

```
$bindinfo = "маршрут_к_bindinfo";
```

Замените `маршрут_к_bindinfo` на маршрутное имя каталога, в который вы поместили бинарный файл.

6. Сохраните текстовый файл.
7. Теперь отредактируйте файл `nlog-search.pl`. Это Perl-процедура, которая создает страницу с результатами поиска.
8. Найдите раздел, который выглядит следующим образом:

```
1: # here we place each cgi-handler into a temp var for readability.
2:
3: $cgiSunRPC = "sunrpc+$cgidir/nlog-rpc.pl+SunRPC";
4: $cgiSMB = "netbios-ssn+$cgidir/nlog=smb.pl+NetBIOS";
5: $cgiFinder = "finder+$cgidir/nlog-finder.pl+Finder";
6:
7: $qcgilinks = "$cgiSunRPC $cgiSMB $cgiFinder";
```

9. Между строками 5 и 6 добавьте следующую строку:

```
$cgiBIND = "domain+$cgidir/nlog-bind.pl+BIND";
```

10. Отредактируйте строку 7, чтобы она имела следующий вид:

```
$qcgilinks = "$cgiSunRPC $cgiSMB $cgiFinder $cgiBIND";
```

В строке 7 вы можете аналогичным образом добавить ссылки на любые другие созданные вами процедуры.

11. Скопируйте файл nlog-bind.pl из этого каталога в свой cgi-каталог (var/www/cgi в Mandrake Linux), и измените режим доступа (chmod), чтобы приложение могло его прочитать.

Теперь, когда Nmap найдет открытым порт 53 (обычно это DNS-сервер), вы можете щелкнуть мышью на ссылке, которую создает Nlog, и выяснить, какая версия BIND выполняется. Следуя логике этого примера, можно написать дополнительные процедуры для расширения Nlog.

Интересные применения Nlog и Nmap

Теперь вы умеете сканировать порты с помощью Nmap, а также сортировать и анализировать результаты с помощью Nlog. Что же можно делать с этими новыми игрушками? Существует несколько интересных приложений сканеров портов. Приведем несколько реальных примеров, которые есть смысл попробовать в вашей сети (или в чьей-нибудь еще, с разрешения хозяев, конечно!) Возможно, полученные результаты вас удивят.

Выявление малоупотребительных сервисов

Если имеется сервис или номер порта, который виден только на одной или двух машинах, то, вполне возможно, это нечто чужеродное для вашей сети. Это может быть троянская программа или запрещенный сервис (например, Kazaа, ICQ или MSN) или неверно сконфигурированная машина, выполняющая сервер FTP или иной сервер общего доступа. Можно настроить Nlog, чтобы сгенерировать список интересующих вас сущностей в порядке возрастания числа их вхождений, от самых редких к самым частым. Вероятно, вы не захотите включать в это сканирование серверы своей организации, так как на них будет обнаружено множество сервисов с одиночными вхождениями. Однако не вредно просканировать эти серверы отдельно, либо для тонкой настройки, либо для исключения посторонних сервисов.

Охота на незаконные/неизвестные Web-серверы

Вероятно, если в вашей организации поддерживается несколько web-серверов, то при сканировании сети несколько раз проявится сервис HTTP. Однако, вполне возможно, что он проявится не только там, где вы ожидали его увидеть. Некоторые производители настольных компьютеров по умолчанию загружают теперь на свои машины небольшие web-серверы для использования персоналом технической поддержки. К сожалению, зачастую эти web-серверы являются убогими программами с дырами в безопасности. web-серверы можно также обнаружить на принтерах, маршрутизаторах, межсетевых экранах и даже на коммутаторах и другом специализированном оборудовании. Эти серверы могут понадобиться для настройки оборудования, но если они не используются, их необходимо отключить. По умолчанию подобные мини-серверы сконфигурированы без парольной защиты и могут предоставить хакеру плацдарм на данной машине. Они могут также открыть доступ к файлам на машинах, если нарушитель знает, как ими манипулировать. Проведите сканирование с целью выявления скрытых web-серверов и либо выключите их, либо защитите как следует. Необходимо также поискать порты, отличные от 80, которые обычно используются для HTTP. В [табл. 4.10](#) содержится краткий список номеров портов для web-сервиса.

Таблица 4.10. Употребительные альтернативные порты web-серверов

| Употребительный номер порта | Протокол |
|-----------------------------|------------------------------|
| 81 | Альтернативный web-сервис |
| 88 | web-сервис |
| 443 | Https, защищенный web-сервис |
| 8000-8002 | web-сервис |

| | |
|------|------------|
| 8080 | Web-сервис |
| 8888 | Web-сервис |

Сканирование с целью выявления серверов, выполняющихся на настольных системах

Развивая предыдущий пример, ограничим диапазон IP-адресов несерверными машинами и зададим диапазон портов от 1 до 1024. В результате будут выявляться настольные машины с сервисами, которые обычно выполняются на серверах, например, электронная почта, Web и FTP. Если для этого нет уважительной причины (например, PCAnywhere), подобные сервисы на настольных машинах запускаться не должны.

Охота на "троянские" программы

Чтобы устроить в сети охоту на "троянские" программы, выполните сканирование сети и преобразуйте результаты в формат базы данных Nlog. Откройте страницу поиска Nlog, выберите порты и задайте диапазон от 30000 до 65400. Это излюбленный диапазон для троянских программ, так как он находится в стороне от обычных сервисов и поэтому троянцы зачастую остаются незамеченными - если, конечно, вы не прибегаете к сканированию портов сети. Сам по себе факт использования сервисами портов с большими номерами не доказывает, что это троянские программы, но обратить внимание на них, безусловно, стоит. Сужая поиск до машины и номера порта, можно определиться с ними, проверяя сервисы, запущенные на этих машинах, или подключаясь посредством telnet к этим портам и проверяя, выдается ли приветственная информация сервиса.

Проверка внешнего представления сети

Поместите машину с Nmap вне вашей сети, используя коммутируемое или домашнее широкополосное соединение, и попробуйте просканировать общедоступные IP-адреса организации. Прделав это, вы увидите, какие сервисы доступны из Интернета (и, таким образом, видны для любого человека, умеющего обращаться со сканером портов). Это наиболее уязвимая часть сети, и для общедоступных сервисов следует принять дополнительные меры безопасности, используя сканер уязвимостей, например, описанный в следующей лекции. Внешнее сканирование покажет также, правильно ли межсетевой экран фильтрует порты, которые он переправляет по адресам во внутренней ЛВС.

Итак, вы познакомились с замечательными возможностями, которые предоставляют сканеры портов, в частности Nmap. С их помощью можно определить, что выполняется в сети, и какие места остаются открытыми. Но как узнать, уязвимы ли эти открытые точки? На самом ли деле безопасны сервисы, которые сознательно сделаны открытыми и, как предполагается, защищенными? Это выходит за рамки функций сканера портов и попадает в область сканирования уязвимостей - предмет следующей лекции.

Инструменты безопасности с открытым исходным кодом

5. Лекция: Сканеры уязвимостей: версия для печати и PDA

Вы защитили свой периметр с помощью межсетевого экрана и просканировали порты внутренней и внешней сетей. Что еще можно сделать, чтобы повысить безопасность сети? Межсетевые экраны не позволят легко проникать во внутреннюю ЛВС извне. Сканирование портов выявит запущенные сервисы и даст возможность исключить ненужные. Однако, как быть с сервисами, которые необходимы? Вы должны поддерживать Web-серверы и серверы электронной почты для общения с внешним миром. Могут понадобиться и другие приложения, такие как FTP, SSH, Telnet и индивидуальные приложения баз данных. Как узнать, безопасны ли эти сервисы? Чтобы оценить риски, необходимо знать угрозы и методы осуществления несанкционированного доступа к информации и ресурсам организации.

Обзор лекции

Изучаемые концепции:

- Типичные уязвимости прикладного уровня
- Настройка и конфигурирование сканирования уязвимостей
- Как выполнить безопасное и этичное сканирование уязвимостей
- Примеры конфигураций сканирования
- Чего не делает сканирование уязвимостей

Используемые инструменты:

Nessus и NessusWX

Что чаще всего делает систему уязвимой? Приложения. Взглянув на эталонную модель ВОС, вы увидите, что уровень приложений находится на вершине стека сетевых коммуникаций, что делает его самым сложным и изменчивым. Сканеры уязвимостей позволяют проверить различные приложения в системе на предмет наличия дыр, которыми могут воспользоваться. Сканер уязвимостей может также использовать низкоуровневые средства, такие как сканер портов, для выявления и анализа возможных приложений и протоколов, выполняющихся в системе.

| Номер уровня модели ВОС | Название уровня | Примеры протоколов |
|-------------------------|-----------------------|--|
| Уровень 7 | Прикладной уровень | DNS, FTP, HTTP, SMTP, SNMP, Telnet |
| Уровень 6 | Уровень представления | XDR |
| Уровень 5 | Уровень сеанса | RPC |
| Уровень 4 | Транспортный уровень | NetBIOS, TCP, UDP |
| Уровень 3 | Сетевой уровень | ARP, IP, IPX, OSPF |
| Уровень 2 | Канальный уровень | Arcnet, Ethernet, Token ring |
| Уровень 1 | Физический уровень | Коаксиальный кабель, оптоволокно, витая пара |

Выявление дыр в безопасности ваших систем

Необходимо помнить, что компьютерная безопасность сродни другим видам безопасности. Средний компьютерный нарушитель предпочитает цели подступнее и попроще. Существуют, конечно, мастера взлома систем, которые охотятся за определенными целями и разрабатывают их в течение

месяцев или даже лет, применяя физические, морально-психологические и технические средства. И, как в случае физической безопасности, если кто-то действительно захочет проникнуть в ваш компьютер, и имеет для этого достаточно денег, времени и ресурсов, то он, вероятно, добьется успеха. Однако, если вы не работаете в банке, правительственном учреждении или компании из Fortune 500, то вам, скорее всего, не стоит волноваться, что такой обихакер будет вас преследовать. Вам надо опасаться рядовых компьютерных преступников, автоматических "червей" и вирусов. Ваша работа состоит в том, чтобы в вашей сети было меньше дыр, чем у соседа, чтобы хакеры обошли вас стороной при выборе цели для взлома. Это как автомобиль с хорошей сигнализацией - только по-настоящему опытный и мотивированный угонщик будет пытаться украсть его.

В действительности только очень небольшой процент компьютерных преступников исследуют и разрабатывают собственные методы атак. Большинство хакеров действуют с помощью опубликованных и известных дыр в безопасности и средств, показывающих, как проникнуть в ваши компьютеры. Подобную информацию можно найти на бесчисленных web-сайтах, а хакерские инструменты, использующие эти дыры, доступны для загрузки.

Все основные сбои в работе Интернета, вызванные компьютерными преступлениями, возникали в результате использования дыр в безопасности, известных за некоторое время до инцидента. Обычно эпидемия распространяется через месяцы или даже годы после того, как становится известна лежащая в ее основе уязвимость. При нашествии Code Red в 2001 г. использовалась уязвимость, корректирующая заплатка для которой была доступна более года; то же с "червем" Nimda. "Червь" SQL Slammer, атаковавший базы данных SQL в феврале 2003 года, действовал спустя полгода после выпуска программной коррекции. Факт состоит в том, что большинство вторжений в компьютеры используют хорошо известные методы и уязвимости, для которых доступны заплатки или защитные решения. Так называемое мгновенное использование уязвимостей и неопубликованных дыр в безопасности - относительная редкость.

Почему люди пренебрегают простыми вещами и не заделывают дыры в безопасности своих систем? Если бы они это делали, то было бы значительно меньше компьютерных преступлений и книги, подобные этой, возможно, не существовали бы. Однако множество систем с множеством уязвимостей продолжает существовать по тысяче причин:

- Нехватка времени или персонала. Организации сокращают расходы и в трудные времена увольняют технический персонал (информационные технологии (ИТ) не приносят прибыли). Иногда функции ИТ полностью передаются сторонним организациям. И хотя зачастую это практичное решение, внешние организации, поддерживающие локальные сети, далеко не всегда считают информационной безопасностью своей первейшей обязанностью. Главное для них - бесперебойная работа сети. Удовлетворение запросов пользователей оказывается важнее безопасности.
- Опасения в отношении стабильности системы. Хорошо известно, что производители систем при выпуске заплат порой исправляют одну вещь и портят две других. Для критически важных систем затраты времени и ресурсов для надлежащего тестирования программных коррекций зачастую превышают выгоды от обновления.
- Слишком много заплат, чтобы с ними управиться. Если вы являетесь подписчиком Windows Update, сервиса коррекций Microsoft, то вы, вероятно, как минимум раз в неделю получаете уведомление о необходимости обновить или залатать систему. Для занятых системных администраторов это может быть слишком большой нагрузкой в дополнение к их обычным обязанностям. Действительно, было проведено исследование, показавшее, что расходы на корректировку программного обеспечения нередко превышают его начальную цену.
- Невежество. Системные администраторы многих организаций просто не знают о существовании проблемы и наличии заплат. Теперь, при автоматическом обновлении от Microsoft, эта проблема для систем Windows стала менее острой, но она остается для других производителей и менее известного программного обеспечения. Даже для Windows существует несколько несовместимых менеджеров заплат. Это одна из причин, почему SQL Slammer так быстро распространился, - стандартный сервис обновления Windows просмотрел его.

Еще один момент, облегчающий жизнь хакерам, состоит в том, что обычно имеется несколько различных путей проникновения в систему. На самом деле, для множества выполняемых сервисов может существовать десяток или больше потенциальных окон для входа в подключенный к Интернету сервер. Если атака одного типа не работает, всегда можно попробовать другую. В следующих разделах описаны некоторые возможные способы, с помощью которых знающий человек может вызвать разрушение системы организации. Некоторые из них могут быть неприменимы к вашей сети, но вполне возможно, что найдется по крайней мере два или три потенциальных источника уязвимости.

Переполнение буфера

Как упоминалось в [лекции 4](#), переполнение буфера является, несомненно, наиболее популярным способом взлома систем. Первым документированным использованием переполнения буфера был Интернет-"червь", выпущенный Робертом Моррисом 2 ноября 1988 г. Он был назван "червем" Morris по имени автора, и создан только для того, чтобы доказать, что это можно сделать. Он работал, используя ошибку в программе finger и распространяя себя с одной машины на другую. Для своего тиражирования он использовал плохую конфигурацию Sendmail и rsh. Предполагалось, что он копирует себя только на

несколько систем. Но при программировании "червя" Моррис сделал ошибку, и тот быстро распространился по всей Сети, состоявшей тогда лишь из нескольких тысяч систем. "Червь" поставил на колени крупнейшие университеты и другие организации, пытавшиеся справиться с быстро распространяющейся ошибкой. Это стало зарей новой эры для компьютерных хакеров и открыло глаза многим из тех, кто считал Интернет безопасным и дружелюбным местом. С тех пор возможность переполнения буфера была найдена почти во всех важных программах и часто использовалось теми, кто пытался получить несанкционированный доступ к системам.

Как защитить себя от переполнения буфера? Если вы не хотите отлаживать все применяемое вами программное обеспечение (что, между прочим, подразумевает доступ ко всем исходным текстам!), остается ждать, когда кто-то обнаружит ошибку и сообщит о ней, а затем - когда программистская компания выпустит заплату. К сожалению, отслеживание выпускаемых заплат и определение того, какие из них имеют к вам отношение, не говоря уже об их тестировании и установке, способно занять все рабочее время. Многие организации предпочитают просто не беспокоиться, а организации, прилежно устанавливающие все заплатки, зачастую не успевают делать это вовремя. Даже несколько собственных систем корпорации Microsoft пали жертвой эпидемии SQL Slammer, так как на некоторых SQL-серверах не были установлены программные коррекции, которые корпорация сама выпустила! Одним из хороших способов узнать, имеются ли условия для переполнения буфера в ваших приложениях, является их тестирование с помощью программного обеспечения сканирования уязвимостей. Это позволит обнаружить большинство известных переполнений буфера, существующих в системе, и своевременно применить корректирующие заплатки, необходимые для устранения этих условий.

Слабые места маршрутизаторов и межсетевых экранов

Эти устройства являются первой линией обороны против посторонних, пытающихся проникнуть в вашу корпоративную сеть. Однако, в связи с возрастающей сложностью устройств и изощренностью атакующих, при некорректном конфигурировании этот рубеж может оказаться слабым. Владение языком маршрутизатора для Cisco IOS - по сути отдельная специальность. Если в организации нет технических специалистов по оборудованию Cisco, то, вероятно, маршрутизаторы Cisco с точки зрения безопасности сконфигурированы не оптимально. А межсетевые экраны конфигурировать еще сложнее. Как вы узнали из [лекции 3](#), одна неверная строка конфигурации может свести на нет межсетевую защиту. Замотанный технический специалист, пытающийся побыстрее настроить доступ для сотрудников или внешних пользователей, чаще будет ошибаться в сторону расширения доступа, а не лучшей защиты.

Даже когда наборы правил написаны правильно, на маршрутизаторах нередко выполняются слабые или опасные сервисы. Многие маршрутизаторы для интерактивного входа по-прежнему полагаются на Telnet, а не на безопасное приложение, такое как SSH. Это открывает дверь для атак с помощью сетевого анализатора путем перехвата комбинации входного имени и пароля. На некоторых маршрутизаторах до сих пор выполняются finger и другие службы, способные стать каналом утечки информации.

Даже межсетевые экраны - наиболее защищенные устройства - не обладают абсолютной невосприимчивостью к атакам. Некоторые межсетевые экраны строятся поверх обычных операционных систем, таких как Windows или UNIX, и поэтому могут быть уязвимы для всех обычных атак уровня ОС. Даже если операционная система межсетевого экрана является собственной, в ней могут существовать уязвимости. Многие межсетевые экраны взаимодействуют с пользователями при помощи web-сервера, а значит, могут быть использованы дыры в web-интерфейсе. Обеспечение собственной безопасности этих средств передовой линии обороны критически важно и должно считаться одним из высших приоритетов.

Межсетевые экраны имеют также свойство обеспечивать безопасность, которая, по выражению Билла Чезвика, "тверда снаружи, мягка внутри". Это означает, что через них трудно проникнуть извне, но против атак изнутри сети почти никакой защиты не предусматривается. Необходимо добиться, чтобы внутренние системы были по крайней мере минимально защищены, а сетевая безопасность не зависела целиком и полностью от межсетевых экранов.

Использование уязвимостей Web-серверов

В наше время практически каждая компания должна иметь web-сервер. Отсутствие такового приравнивается к отсутствию телефона или факса. web-серверы печально известны наличием ошибок и дыр в безопасности. Сама идея web-сервера - возможность брать с сервера файлы без какой-либо аутентификации - создает потенциал для брешей в защите. Большое число дыр обусловлено все возрастающим числом и разнообразием протоколов и команд, с которыми приходится иметь дело web-серверам. Когда web-страницы состояли только из HTML, держать все под контролем было значительно легче. Однако сейчас Web-серверы должны интерпретировать ASP, PHP и другие типы трафика, содержащего исполнимый код, и по мере того как web-приложения становятся все сложнее, проблемы безопасности будут только обостряться.

Некоторые web-серверы защищены лучше, чем другие, но у каждого есть свои недостатки. А взлом web-сервера может вызвать не только смущение из-за обезображенной web-страницы, если этот сервер осуществляет также доступ к базе данных и другим внутренним системам, что в наше время является общепринятым.

Использование уязвимостей почтовых серверов

В электронный век электронная почта жизненно важна для взаимодействия организаций. Однако почтовые серверы традиционно служили излюбленными целями атакующих. Самый первый агент передачи почты, Sendmail, был наштапкован уязвимостями и продолжает вызывать конвульсии у профессионалов в области информационной безопасности. Немногим лучше флагманский почтовый сервер Exchange корпорации Microsoft. Обычно именно серверы Web и электронной почты оказываются наиболее уязвимыми точками организаций.

Серверы DNS

Серверы, которые управляют и поддерживают доменные имена вашей организации, являются привлекательной целью для хакеров. Основной DNS-сервер, BIND (Berkeley Internet Name Domain), постоянно находился в первой десятке наиболее эксплуатируемых хакерами сервисов. DNS - старая программа, и сама ее структура способствует наличию дыр (вместо модульной архитектуры - один монолитный бинарный файл). DNS часто запускается от имени суперпользователя, что делает его взлом еще более опасным. Кроме того, поскольку DNS трудно настраивать и его плохо понимают, он зачастую сконфигурирован неправильно и защищен плохо. Настройки межсетевых экранов для DNS нередко сконфигурированы неверно - большинство системных администраторов разрешают нефилтрованный доступ внутрь и наружу.

Web, электронная почта и другие сервисы более заметны, и технический персонал уделяет им больше внимания; в то же время, дыры в DNS предоставляют самый быстрый и легкий способ стереть вашу организацию с карты Интернета. Даже если сохраняется IP-связность с внешним миром, без корректной работы сервиса DNS для ваших доменов никто не сможет добраться до ваших web-серверов, и ни одно электронное сообщение до вас не дойдет. На самом деле, DNS считается самым слабым местом всей инфраструктуры Интернета и потенциальной целью для атак кибертеррористов. Вместо того чтобы взламывать серверы или прорываться через межсетевые экраны, атакующий может просто организовать атаку на доступность вашего сервиса DNS, эффективно отключая вашу организацию "от эфира". Или, хуже того, используя атаку типа "отравление кэша DNS", хакер может по своему выбору перенаправить потенциальных посетителей вашего web-сайта.

Использование уязвимостей баз данных

Многие web-сайты организаций предоставляют внешний доступ к своим базам данных. Например, можно дать клиентам возможность помещать и оперативно проверять статус заказов, разрешить служащим получать через Web информацию по программам социальной поддержки или предоставить поставщикам доступ к системе, чтобы автоматически обновлять время поставки. Такие функции обычно обращаются к внутренней базе данных организации. Это выводит web-сайты за рамки одномерных оперативных изданий, какими они были в ранние дни Интернета, и делает их расширением ваших систем для внешних пользователей. Однако, поступая так, вы активизируете большой потенциальный источник уязвимостей. Зачастую дополнительных мер безопасности для внешнего использования подобных систем не предпринимается. Иными словами, предполагается, что пользователи будут добропорядочными и не будут совершать явно враждебных действий. Было обнаружено, что программное обеспечение внешнего интерфейса Web, такое как ColdFusion и PHP, не обладает достаточными средствами аутентификации и содержит ошибки, приводящие, в частности, к переполнению буфера. Специально созданный универсальный локатор ресурсов может направить SQL-инструкции или другие команды базы данных прямо в сердце вашей системы. "Червь" SQL Slammer, быстро распространившийся по всему миру в начале 2003 г. и использовавший слабые места в SQL Server корпорации Microsoft, показал, как это может случиться.

Управление пользователями и файлами

Эта область является одной из самых болезненных для информационной безопасности. Вы должны предоставить пользователям доступ к системам и программам, которые нужны им для выполнения работы. Однако, ключевым принципом хорошей защиты является принцип минимизации привилегий, то есть предоставление пользователям минимально достаточного для работы доступа - и не больше. Определение этого уровня - хитрая задача. Предоставьте им слишком мало прав, и вас задергают звонками из службы помощи пользователям и жалобами; дайте слишком много прав, и вы ослабите защиту своей системы. Большинство администраторов будут отклоняться в сторону смягчения правил доступа, так как это уменьшает объем сваливающейся на них работы.

К сожалению, системы, дружелюбные пользователям, такие как Windows, также делают крен в эту сторону, давая много прав на самом слабом уровне: низкий уровень безопасности по умолчанию. В Windows есть несколько встроенных системных счетов и разделяемых ресурсов, применяемых для операций на системном уровне и имеющих больше прав, чем им на самом деле требуется. Одним из примеров служит подразумеваемый разделяемый ресурс IPC (Inter-Process Communication - межпроцессное взаимодействие), который может использовать любой пользователь, чтобы получить информацию о машине или домене. Аналогичным образом может применяться гостевой системный счет. Можно отключить или ограничить эти системные счета, но вы должны сделать это вручную после установки. К чести Microsoft, эти типы подразумеваемых системных счетов ограничены в Windows XP, но все еще существуют (поскольку они нужны для простых одноранговых сетей, которые допускает Windows). Немного лучше системы UNIX. Недостаток гранулярности в управлении системными счетами, то есть существуют только супер- и обычные пользователи, а в результате права суперпользователя получают слишком многие.

И, разумеется, поддержка актуального списка пользователей для больших сетей может требовать ежедневных усилий. Бездействующие или неиспользуемые системные счета - ценная цель для хакеров, так как их можно использовать, не беспокоясь о том, что реальный владелец заподозрит неладное.

Хороший сканер уязвимостей выявит подразумеваемые и слабые пароли, такие как стандартная комбинация входного имени и пароля "administrator/administrator" в системах Windows. Сканер будет также брать набор удостоверений и проверять, как далеко он сможет зайти. Он может выявлять неиспользуемые системные счета и пользователей, которые никогда не меняли свои пароли (в системах Windows). Это поможет разглядеть трещинки в вашей броне с точки зрения управления счетами пользователей.

Подразумеваемые системные счета производителей

Пытаясь облегчить вам жизнь, производители зачастую существенно усложняют вашу работу по обеспечению информационной безопасности. Многие производители оборудования поставляют его со стандартными подразумеваемыми входными именами и счетами пользователей для облегчения настройки. Некоторые из них добавляют также счета для технического и обслуживающего персонала. Предполагается, что сразу после установки оборудования или программного обеспечения вы измените подразумеваемые пароли, но далеко не все это делают. В результате во многие машины можно попасть после простого перебора некоторого числа комбинаций подразумеваемых входных имен и паролей. Подобные уязвимости наиболее характерны не для систем UNIX и Windows, а для маршрутизаторов, коммутаторов, телефонных систем и других типов оборудования. Однако есть и исключения. Существует целый протокол, основывающийся на идее подразумеваемых паролей. SNMP (Simple Network Management Protocol - простой протокол управления сетями) был создан для того, чтобы дать возможность программному обеспечению автоматически опрашивать устройства и получать базовую информацию о них (например, состояние включено/выключено). В некоторых случаях SNMP позволяет даже выполнить простые конфигурационные операции. В принципе, это была хорошая идея, и многие компании создали системы управления сетями на основе этого протокола. Однако при реализации производители использовали два основных подразумеваемых системных счета - "public" и "private" как цепочки символов или пароли своего сообщества. Зная эти пароли, любой имеющий доступ к сети может опросить состояние ваших устройств.

SNMP позволяет также посылать на устройства основные команды, такие как возврат маршрутизатора в исходное состояние или отключение интерфейса. Очень немногие пользователи SNMP удосужились изменить подразумеваемые цепочки символов сообщества, поскольку делать это на каждой машине - утомительно. В результате хакеры с простым инструментарием, таким как snmpwalk, который свободно доступен в Интернете, могут собрать информацию о сети, построить ее карту и, возможно, даже отключить ее, если вы применяете SNMP с подразумеваемыми строками сообщества. Чтобы подлить масла в огонь, добавим, что новые программы использования переполнения буфера в реализациях протокола SNMP позволяют хакерам полностью подчинить себе удаленную машину. Многие выполняют SNMP на своих машинах, даже если не используют его, потому что производители часто включают его по умолчанию, чтобы облегчить сетевую идентификацию.

Другим примером из области программного обеспечения является подразумеваемый системный счет sa, встроенный в SQL Server корпорации Microsoft. Этот счет используется межсистемными процессами, но он может также быть доступен командному файлу или "червю", что наглядно доказали разрушения, вызванный "червем" SQL Slammer. В Интернете существуют сайты, где перечислены все основные производители оборудования и программного обеспечения и все подразумеваемые пароли, которые могут существовать. И, конечно, имеются автоматические программы, которые могут очень быстро и без больших усилий все их проверить.

Пустые или слабые пароли

Хотя иметь системный счет с пустым паролем кажется безумием, во многих сетях делается именно это. И, хотите верьте, хотите нет, но некоторые поступают так даже со счетом администратора. Также неожиданно распространено использование комбинации пользователь/пароль вида administrator/administrator, которая служит подразумеваемой настройкой Windows. Нет ничего удивительного, что "черви" и программы взлома автоматически проверяют это условие. Если они его находят, то получают золотой приз: полный административный доступ к системе. Аналогично, когда пользователи задают пароли, они могут просто оставить их пустыми. Это дает шанс любому, имеющему список пользователей, попытаться найти счета с пустым паролем. Вы можете задать свою политику управления паролями, запрещающую подобное легкомыслие, и предъявляющую дополнительные требования к длине и сложности паролей. Целесообразно также потребовать регулярного изменения паролей и избавиться от неиспользуемых системных счетов. При сканировании уязвимостей перечисленные условия будут проверяться.

Ненужные сервисы

Подобно рудиментарному хвосту, на машинах нередко выполняются приложения, которые больше не служат никакой полезной цели. Эти сервисы могли использоваться прежними версиями библиотек, и программисты просто не удосужились своевременно их удалить. Это одна из отрицательных сторон все возрастающей вычислительной мощности и емкости памяти. Раньше программисты тщательно распределяли каждый используемый байт и не оставляли в программах ненужных строк. Однако в наш век раздутых до гигабайтных размеров операционных систем часто легче оставить устаревшие сервисы, чем рисковать нарушить функционирование некоторых программ, которые от этих сервисов зависят. Неприятный момент состоит в том, что нередко эти сервисы включены по умолчанию. В табл. 5.1 перечислены вышедшие из употребления сервисы, которые, как правило, могут быть безболезненно отключены.

Таблица 5.1. Бесплезные сервисы

| Сервис | Обычный номер порта | Функции |
|--------------------------------------|---------------------|---|
| chargen | 19 | В ответ на запрос посылает поток стандартных символов. Мало того, что данный сервис больше не применяется - он может быть использован для атаки на доступность, если заставляя его непрерывно выплевывать потоки символов |
| daytime | 13 | Возвращает время дня. На самом деле не требуется ни для каких функций современных систем |
| discard | 9 | Молча отбрасывает все, что ему посылается. Применяется в основном в тестовых целях |
| echo | 7 | Возвращает назад то, что было ему послано. Как и chargen, может использоваться в атаках на доступность, если посылать ему постоянный поток данных для эхо-отражения |
| finger | 79 | Об этом сервисе было достаточно сказано ранее. Очень полезен для хакеров |
| qotd (quote of the day - цитата дня) | 17 | При входе пользователя в систему посылает ему небольшую цитату или фразу, заданную системным администратором |

Утечка информации

В поисках способа проникнуть в систему хакеры или взломщики начинают с некоторой базовой разведки. Они пытаются как можно больше узнать о вашей системе и сети, прежде чем попытаться взломать их. Подобно тому как вор-домушник осматривает объект проникновения, они ищут электронные эквиваленты выключенного света, накопившихся газет, незакрытых окон и т.д. Делается это с помощью ряда инструментов, таких как сканеры портов, или других средствах взлома, доступные в Интернете. К сожалению, многие операционные системы охотно помогают этим незаконным сборщикам информации. Как болтливый портье, они выдают жизненно важную системную информацию, не спрашивая удостоверение личности.

Особенно грешит этим Windows. Поскольку эта ОС создавалась как самонастраивающаяся сетевая система, она предлагает всевозможную информацию любой системе, которая опрашивает ее с помощью подходящей команды. Как упоминалось выше, неправильно сконфигурированные серверы DNS могут также выдавать много информации о конфигурации сети. Наконец, огромный объем информации можно собрать при использовании общедоступных поисковых машин, таких как Google. Информацию нередко оставляют в общедоступных каталогах web-серверов, считая, что раз на нее нет ссылок с web-страниц, то она не видна поисковым машинам. Это не так, и желательно регулярно "гуглить" название вашей организации и универсальные локаторы ресурсов, чтобы посмотреть, не всплывет ли что-то интересное.

На основе этих данных внешний пользователь может сгенерировать списки пользователей, разделяемые диски и каталоги, имена систем и сотрудников и другую информацию, полезную для взлома методом грубой силы, когда с помощью автоматизированных программ пробуются различные комбинации паролей, а также для применения методов морально-психологического воздействия (см. врезку о взломе систем).

Анатомия взлома системы

Здесь приведен пример, показывающий, как взломщик может применять некоторые из перечисленных в данной лекции методов для получения несанкционированного доступа. Предположим, что хакер захотел взломать Example.com и получить доступ к данным о служащих.

1. Для начала хакер оконтурил цель. При кратком посещении web-сайта Example.com для просмотра информации о персонале он смог определить, какие IP-диапазоны используются, и узнал имена некоторых системных администраторов из раздела о технических контактах.
2. Затем он просканировал порты в выявленном диапазоне IP-адресов и определил, как системы отвечают и какие службы они выполняют.
3. Используя более сложный инструмент, такой как сканер уязвимостей из данной лекции, гипотетический хакер собрал дополнительную информацию о системах, а именно, какие из них уязвимы и для каких атак.
4. Применяя сканер портов или программу анализа защищенности, хакер смог определить, что один из серверов допускает сеансы NetBIOS null, что помогло сгенерировать список всех пользователей системы. Хакер также обнаружил, что web-сервер уязвим по отношению к переполнениям буфера и для Windows-программы использования уязвимости, открывающей доступ к любому каталогу на этом сервере.
5. Затем хакер произвел поиск в Интернете, задав ключевые слова для средств, эксплуатирующих выявленные слабые места. Он смог найти инструмент, предоставивший ему административный доступ через дыру с переполнением буфера.
6. Даже если бы оказалось, что системы не содержат уязвимостей, открывающих непосредственный доступ, хакер мог бы использовать собранную им информацию для атаки методом грубой силы на файл паролей или для применения методов морально-психологического воздействия. Он мог бы позвонить пользователю, представиться системным администратором и спросить пароль. Или позвонить в службу поддержки, заявить, что он пользователь, который забыл свой пароль, и попросить их сменить пароль на указанный им. Вариации здесь ограничиваются только изошренностью фантазии взломщика.

Атаки на доступность

Не сумев получить доступ к вашей системе, многие компьютерные преступники получают не меньше удовольствия, отключив ее, чтобы никто другой не мог ей воспользоваться. Это касается, в первую очередь, наиболее примечательных сайтов и вопросов достижения политических целей. При большом объеме операций электронной коммерции, час простоя может стоить миллионы долларов. Атаки на доступность могут принимать различные формы от простого затопления основных маршрутизаторов потоками данных до реального использования слабого места в программе для нарушения работы сервиса и всего сервера. От первого трудно защититься, но последнее вполне предотвратимо при выявлении и последующем исправлении или исключении условия, которое создает возможности для атаки на доступность.

Сканеры уязвимостей спешат на помощь

Как можно видеть, современные компьютерные сети содержат множество потенциальных областей небезопасности. Как закрыть все эти пути для атак? Вы чувствуете себя одиноким стражем, пытающимся защитить гигантский замок с множеством окон, дверей и других путей проникновения. Невозможно быть сразу везде. Можно тратить день за днем, просто проверяя вручную дыры в безопасности. Даже если попробовать автоматизировать эту деятельность с помощью командных файлов, придется использовать десятки программ. К счастью для вас и вашей психики, существуют пакеты, называемые сканерами уязвимостей, которые автоматически выявляют наличие слабых мест и помогают в решении других проблем.

Nessus: Сканер уязвимостей со встроенным сканером портов

Nessus

Автор/основной контакт: Renaud Deraison

Web-сайт: <http://www.nessus.org/>

Платформы: Linux, BSD UNIX

Лицензия: GPL

Рассмотренная версия: 2.0.10a

Другие ресурсы:

См. списки почтовой рассылки в разделе "Сеть надежной поддержки"

Nessus - действительно изумительная программа. Это великолепный пример того, как хорошо могут работать проекты с открытыми исходными текстами. Он надежен, хорошо документирован, отлично поддерживается, он лучший в своем классе. Nessus постоянно попадает в число лучших среди всех сканеров уязвимостей - коммерческих и некоммерческих. Это поразительно, если принять во внимание его конкурентов, стоящих тысячи долларов и созданных крупными компаниями. Он продолжает впечатлять и постоянно совершенствоваться и, самое главное, защищать тысячи сетей организаций. Ряд проектных решений делают Nessus уникальным и превосходящим другие сканеры уязвимостей.

Глубина тестирования

В настоящее время Nessus предлагает более 2000 отдельных тестов уязвимостей, которые охватывают практически все области потенциально слабых мест в системах. Очень немногие существующие сканеры могут конкурировать с достигнутым в Nessus уровнем тестирования, и новые тесты добавляются ежедневно всемирной сетью разработчиков. Скорость выпуска новых тестов для выявляемых уязвимостей обычно измеряется днями, если не часами. Его архитектура на основе встраиваемых модулей позволяет легко добавлять новые тесты. Ниже представлен перечень всех категорий тестов, которые выполняет Nessus:

- Потайные входы;
- Ненадлежащее использование CGI;
- Cisco;
- Атаки на доступность;
- Ненадлежащее использование Finger;
- FTP;
- Удаленный доступ к командному интерпретатору;
- Удаленное получение прав суперпользователя;
- Общие;
- Прочие;
- Netware;
- NIS;
- Сканеры портов;
- Удаленный доступ к файлам;
- RPC;
- Настройки;
- Проблемы SMTP;
- SNMP;
- Непроверенные;
- Бесплезные сервисы;
- Windows;
- Windows: управление пользователями.

Можно отключить целую категорию тестов, если они неприменимы к вашей системе или могут быть опасны для нее. Если вас беспокоят отдельные тесты, можно отключить только их. Например, естественно отключить категорию "Непроверенные", которая содержит тесты, еще не полностью оттестированные

(клиент, конечно, всегда прав, но должен соблюдать разумную осторожность). В приложении Е приведен полный список всех проверок безопасности. Помните, однако, что этот список соответствует указанной дате и постоянно изменяется по мере добавления новых встраиваемых модулей.

Архитектура клиент-сервер

Для выполнения проверок безопасности Nessus опирается на архитектуру "клиент-сервер". Сервер выполняет проверки, а клиент конфигурирует и управляет сеансами. Тот факт, что клиент и сервер могут быть разделены, предоставляет несколько уникальных преимуществ. Во-первых, сканирующий сервер можно расположить вне вашей сети, но обращаться к нему изнутри сети через клиента. Во-вторых, различные клиенты могут поддерживать разные операционные системы. В настоящее время существуют доступные клиенты UNIX и Windows, и ведутся работы по созданию дополнительных. В настоящее время предоставляется также интерфейс web-клиента, который делает Nessus полностью платформно-независимым (по крайней мере, на стороне клиента).

Независимость

Поскольку исходные тексты Nessus открыты, а встраиваемые модули написаны разнообразными группами специалистов по информационной безопасности, не приходится опасаться каких-либо конфликтов интересов, возможных в коммерческих компаниях. Например, если поставщик коммерческого сканера уязвимостей тесно связан с крупнейшим производителем ОС, то они могут быть настроены менее критически к своим продуктам и медленнее выпускать тесты для них. Проект с открытыми исходными текстами, такой как Nessus, не имеет финансовых причин не разрабатывать и не выпускать тесты немедленно. И, опираясь на его расширяемость, вы всегда можете написать собственный модуль, не дожидаясь официального.

Встроенный язык сценариев атак

В дополнение к архитектуре со встраиваемыми модулями, в Nessus имеется собственный язык сценариев атак, называемый NASL (Nessus Attack Scripting Language). Этот простой в изучении служебный язык позволяет легко и быстро писать собственные встраиваемые модули безопасности, не зная Си или всей внутренней кухни основной программы. (Далее в данной лекции приведен пример написания пользовательского встраиваемого модуля на языке NASL.)

Интеграция с другими средствами

Сканер уязвимостей Nessus можно применять сам по себе или совместно с некоторыми другими защитными средствами с открытыми исходными текстами. Часть из них рассмотрена в этой лекции, и все они являются лучшими из имеющихся. Вместо встроенного можно применить лучший в мире сканер портов Nmap. Сканер портов в Nessus быстрее и немного экономнее в расходе памяти, но Nmap, как вы узнали из [лекции 4](#), предоставляет значительно больше возможностей и настроек. Почти все параметры Nmap можно конфигурировать из клиента Nessus. Nessus также работает с Nikto и Whisker - средствами, которые выполняют более сложные проверки на web-серверах; с программами CGI; а также с Hydra - средством для проведения парольных атак на пользовательские сервисы методом грубой силы. Функциональность этих средств напрямую задана в Nessus, поэтому можно вносить изменения в конфигурацию в рамках единого интерфейса.

Интеллектуальное тестирование

Nessus можно настроить так, чтобы он не выполнял автоматически все тесты уязвимостей на всех хостах. На основе результатов сканирования портов или других исходных данных, таких как результаты предыдущих тестов уязвимостей, Nessus будет запускать только тесты, подходящие для данной машины. Например, если на сервере не запускается web-сервер, то и тесты web-сервера выполняться не будут. Nessus также достаточно интеллектуален, чтобы не предполагать автоматически, что web-серверы будут использовать порт 80; он будет проверять все возможные порты на наличие признаков web-сервера. Nessus найдет даже несколько экземпляров сервисов, подключенных к разным портам. Это особенно важно, если вы нечаянно запустили web-сервер или другой общедоступный сервер на необычных портах.

База знаний

Nessus может сохранять все результаты сканирования в базе данных, называемой базой знаний. Это позволяет использовать результаты прошлых сканирований для определения того, какие тесты выполнять, что избавляет, например, от сканирования портов при каждом запуске, так как Nessus будет помнить, какие порты были открыты в последний раз на каждом хосте, и проверять только их. Он может помнить также, какие хосты он видел в последний раз, и проверить только новые. Я не рекомендую поступать так всякий раз, поскольку можно пропустить новые порты, открывшиеся на машинах, или

новые уязвимости, проявившиеся на просканированных ранее хостах. Однако, это позволяет выполнять сканирование чаще и с меньшими требованиями к полосе пропускания и процессору, при условии, что вы регулярно проводите полное сканирование.

Множество форматов отчетов

Nessus входит в число лучших программ с открытыми исходными текстами и по возможностям генерации отчетов. Хотя они не совершенны, данные сканирования можно выдать почти в любом формате. Базовый HTML и HTML с круговыми диаграммами и графиками - два наиболее популярных формата. В отчеты входят итоговые данные, так что почти без редактирования их можно разместить на внутреннем web-сайте. Поддерживаются также форматы отчетов XML, LaTeX и обычный текст. Клиент Windows предлагает дополнительные форматы отчетов. Доступны и другие средства, обсуждаемые в последующих лекциях, которые позволяют производить дальнейшие манипуляции с данными.

Сеть надежной поддержки

Nessus имеет обширную сеть поддержки для получения помощи как по базовой установке и использованию, так и по более сложным вопросам программирования и индивидуальной настройки. Существует не менее пяти списков почтовой рассылки Nessus, каждый из которых ориентирован на свою область. Подписчики отметят, что сам главный автор, Renaud, отвечает на многие вопросы. Попробуйте получить такую поддержку от коммерческой компании! Имеется архив всех прошлых сообщений, в котором можно проверить, не давался ли уже ответ на ваш вопрос. Ниже перечислены основные списки почтовой рассылки Nessus:

- `nessus`: конечно же, список общей дискуссии о Nessus;
- `nessus-devel`: обсуждение разработки будущих версий;
- `nessu-cvs`: информирует о фиксациях CVS, сделанных в дереве Nessus;
- `nessus-announce`: модерлируемый список с небольшим трафиком, предназначенный для объявлений о доступности новых выпусков;
- `plug-ins-writers`: список, посвященный написанию новых встраиваемых модулей Nessus. Если вы хотите писать собственные проверки безопасности, вам следует подписаться на него.

Чтобы подписаться на любой из перечисленных списков, отправьте сообщение по адресу `majordomo@list.nessus.org` со следующим текстом в теле письма:

```
Subscribe имя_списка
```

Замените `имя_списка` названием списка, на который вы хотите подписаться. Чтобы отказаться от подписки, действуйте аналогичным образом, но поместите в тело текст `Unsubscribe имя_списка`.

На web-сайте Nessus размещено много документации, включая подробные инструкции по установке и основам применения, а также учебники по написанию собственных проверок безопасности на языке NASL. Насколько я знаю, никто еще не пытался собрать в одном документе полное описание всех возможностей и настроек клиента Nessus. В данном разделе мы попытаемся сделать это.

Nessus предоставляет быстрый и простой способ проверки сетей и систем на наличие уязвимостей почти всех видов, поэтому давайте установим его.

Установка Nessus для систем Linux

Имеется два необходимых условия, которые должны быть выполнены перед установкой Nessus, и два других, которые желательно выполнить заранее, чтобы в полной мере воспользоваться дополнительными возможностями.

1. Необходимым условием является установка двух программных продуктов - Gimp Tool Kit (GTK) и libpcap. Если при изучении [лекции 4](#) вы установили Nmap, то эти программы у вас уже имеются. В противном случае можно взять GTK по адресу

```
ftp://ftp.gimp.org/pub/gtk/v1.2
```

а libpcap по адресу <http://www.tcpdump.org/>

2. Двумя необязательными, но желательными программами являются OpenSSL и Nmap. Nessus может использовать Nmap в качестве сканера портов, а OpenSSL - для безопасных коммуникаций между сервером и клиентом.

Есть три способа установки Nessus в системах UNIX, как очень простых, так и несколько более сложных. В данном конкретном случае я рекомендую более длинный процесс, чтобы получить больше контроля над установкой.

Проще всего установить Nessus, запустив удаленным образом командный файл автоматической установки. Это можно сделать, набрав

```
lynx -source http://install.nessus.org | sh
```

Эта команда иницирует командный файл установки и загрузит программу в ваш компьютер. Однако на самом деле я не рекомендую делать это, так как если данный URL будет когда-либо скомпрометирован, ваш компьютер окажется открытым для атаки. Для более безопасной установки сделайте следующие шаги.

1. Загрузите командный файл автоматической установки вручную с `install.nessus.org` и выполните его с помощью команды

```
sh nessus-installer.sh
```

Если процедура автоматической установки не работает должным образом, вам придется скомпилировать программу вручную.

ПРИМЕЧАНИЕ: Я рекомендую выполнить все описанные далее шаги, даже если "процедура" автоматической установки (не совсем точный термин, поскольку этот файл на самом деле содержит целую программу и все ее элементы) вроде бы сработала. Это важно, поскольку при установке сложной программы, такой как Nessus, иногда трудно понять, что делается, и где рванет, если процедура пойдет как-то не так. По крайней мере, когда вы выполняете процесс вручную, вы лучше понимаете, как идет установка.

2. Чтобы установить Nessus вручную, необходимо сначала получить следующие четыре файла, либо с компакт-диска, либо с web-сайта Nessus, и установить их по очереди. Если нарушить порядок установки, то Nessus не будет правильно работать.
 - Nessus-libraries: Базовые библиотеки, необходимые для работы Nessus;
 - Libnasl: Модуль для NASL, встроенного языка Nessus;
 - Nessus-core: Основная программа Nessus;
 - Nessus-plug-ins: Модуль, содержащий все встраиваемые модули, выполняющие проверки безопасности. Чтобы гарантировать, что вы располагаете самыми свежими версиями встраиваемых модулей, после установки следует выполнить процедуру `nessus-update-plugins`, которая произведет необходимые обновления.
3. Перейдите в каталог `nessus-libraries` (используя команду `cd`), затем введите стандартную последовательность команд компиляции:

```
./configure  
make  
make install
```

В конце каждого процесса компиляции может потребоваться выполнение специальных инструкций. Например, для `nessus-libraries` придется добавить `/usr/local/lib` в файл `/etc/ld.so.conf`, а затем набрать `ldconfig`. В результате обновится информация о каталогах с библиотеками, так что операционная система сможет найти специальные каталоги Nessus. Убедитесь, что вы выполнили эти инструкции, прежде чем переходить к следующему шагу.

4. Сделайте то же для `libnasl`. В конце компиляции необходимо проверить, что `/usr/local/sbin` входит в список поиска `PATH`. Эта переменная окружения содержит список каталогов для поиска исполнимых файлов, который производится при вводе каждой команды. Программа установки должна сделать это автоматически, но для проверки наберите

```
echo $PATH
```

Команда выведет на экран значение PATH. Если там отсутствуют /usr/local/sbin и /usr/local/bin, можно добавить их, редактируя файл /etc/bash.rc (правильный путь для Mandrake Linux при использовании командного интерпретатора `bash`). Для других дистрибутивов маршруты могут слегка варьироваться.

5. Повторите этот процесс для двух других модулей.

В результате выполнения этих действий Nessus будет установлен. Однако прежде чем его можно будет использовать, необходимо произвести его настройку.

Настройка Nessus

Для подготовки Nessus к работе, прежде всего необходимо создать сертификат, который Nessus будет использовать для SSL-коммуникаций.

1. Наберите `nessus-mkcert`

Запустится утилита, создающая сертификат безопасности для вашей установки. Можно использовать также сторонние сертификаты, подписанные удостоверяющими центрами, такими как VeriSign.

Если вы получили сообщение об ошибке "file not found" ("файл не найден"), проверьте, что /usr/local/bin и /usr/local/sbin присутствуют в списке поиска PATH (как описано в процедуре установки).

Отвечайте на вопросы по мере их появления. Вам понадобится зарегистрировать сертификат для вашей организации. Если вы не уверены, какие значения вводить, используйте подразумеваемые.

2. Затем необходимо создать несколько счетов пользователей, чтобы можно было входить в Nessus. Поскольку продукт имеет архитектуру "клиент-сервер", то, прежде чем запустить сканирование, требуется войти в сервер с помощью клиента. У Nessus может быть любое число пользователей с индивидуальными наборами правил, которые вы должны задать на данном этапе настройки. Если вы собираетесь использовать Nessus единолично, то нужно просто определить одного пользователя без правил, но при желании можно ограничить IP-адреса, с которых он может входить. Если будет несколько пользователей, эта функция поможет отследить, кто использует ваш сервер Nessus.

Чтобы создать нового пользователя, наберите

```
nessus-adduser
```

Вам будет предложено выполнить последовательность шагов, требуемых для создания нового пользовательского счета.

3. Выполняйте эту команду всякий раз, когда необходимо создать нового пользователя. Для того чтобы использовать Nessus, необходимо создать по крайней мере одного пользователя и выполнить для него настройки.

Теперь, наконец, можно запустить Nessus.

1. Убедитесь, что выполняется X-Window (графическая среда), и запустите командный интерпретатор.
2. В командной строке наберите

```
nessusd &
```

Запустится серверный процесс Nessus. Знак & (амперсанд) предписывает выполнение программы в фоновом режиме, чтобы вы могли вводить другие команды.

3. Затем наберите команду

```
nessus
```

Запустится клиентская часть и на экране отобразится графический интерфейс Nessus.

Теперь можно начать работать с Nessus.

Входная страница Nessus

Первое, что вы увидите, будет входная страница Nessus ([рис. 5.1](#)). Вследствие архитектуры "клиент-сервер", прежде чем начать работать с Nessus, необходимо сначала войти в его сервер. Если клиент и сервер будут запускаться на одной машине, то правильными параметрами входа будут следующие:

- Сервер: localhost;
- Порт: 1241;
- Входное имя: имя, заданное при настройке Nessus;
- Пароль: пароль, заданный при настройке Nessus.

Клиент и сервер могут также выполняться на разных машинах. В этом случае замените localhost на IP-адрес или имя хоста сервера Nessus. Это дает возможность входить из дома в серверы Nessus, функционирующие на работе, и запускать сканирование поздней ночью. Кроме того, можно разместить сервер Nessus в центре обработки данных, где доступна широкая полоса пропускания, и входить в него с внутренней настольной системы, охраняемой межсетевым экраном. Подобная гибкость - важное преимущество Nessus перед некоторыми конкурирующими сканерами, она повышает его масштабируемость для крупных организаций. Некоторые локальные функции можно выполнять на клиенте, не входя в сервер Nessus. В частности, можно вызвать результаты выполненного ранее сканирования для просмотра и манипулирования, конфигурировать опции сканирования. Однако для обращений к встраиваемым модулям или разделу предпочтений без входа в сервер не обойтись, так как соответствующие данные хранятся на серверной стороне.

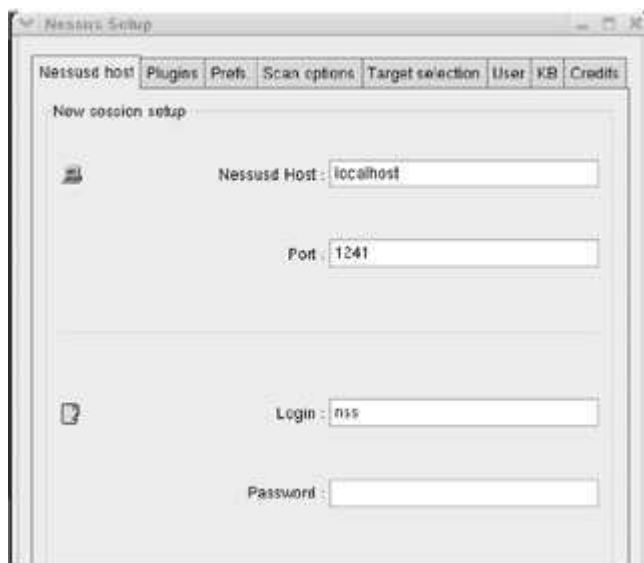


Рис. 5.1. Входной экран Nessus

Вкладка встраиваемых модулей Nessus

После входа вы получаете доступ к различным вкладкам. С помощью вкладки Plugins можно выборочно включать или отключать определенные группы или отдельные встраиваемые модули ([рис. 5.2](#)). На вкладке перечислены все категории, а когда вы щелкаете мышью на некоторой категории, то ниже появляются все ее модули. Снимая флажок справа от элемента, можно отключить категорию или модуль.

Модули, которые могут вызывать проблемы у сервиса или крах серверов, отмечены треугольником с восклицательным знаком ([рис. 5.2](#)). Кроме того, в Nessus имеются кнопки, которые позволяют быстро включить все встраиваемые модули (Enable all), включить все модули, кроме опасных (Enable all but dangerous plugins), отключить все модули (Disable all), или загрузить пользовательский встраиваемый модуль (Upload plugin...). Можно использовать

кнопку Filter для сортировки модулей по имени (Name), описанию (Description), сводке (Summary), автору (Author), идентификационному номеру (ID) или категории (Category). Как правило, рекомендуется запускать Nessus с отключенными опасными модулями; включайте их, только если вы готовы к настоящей проверке доступности и сознательно идете на риск краха некоторых серверов.

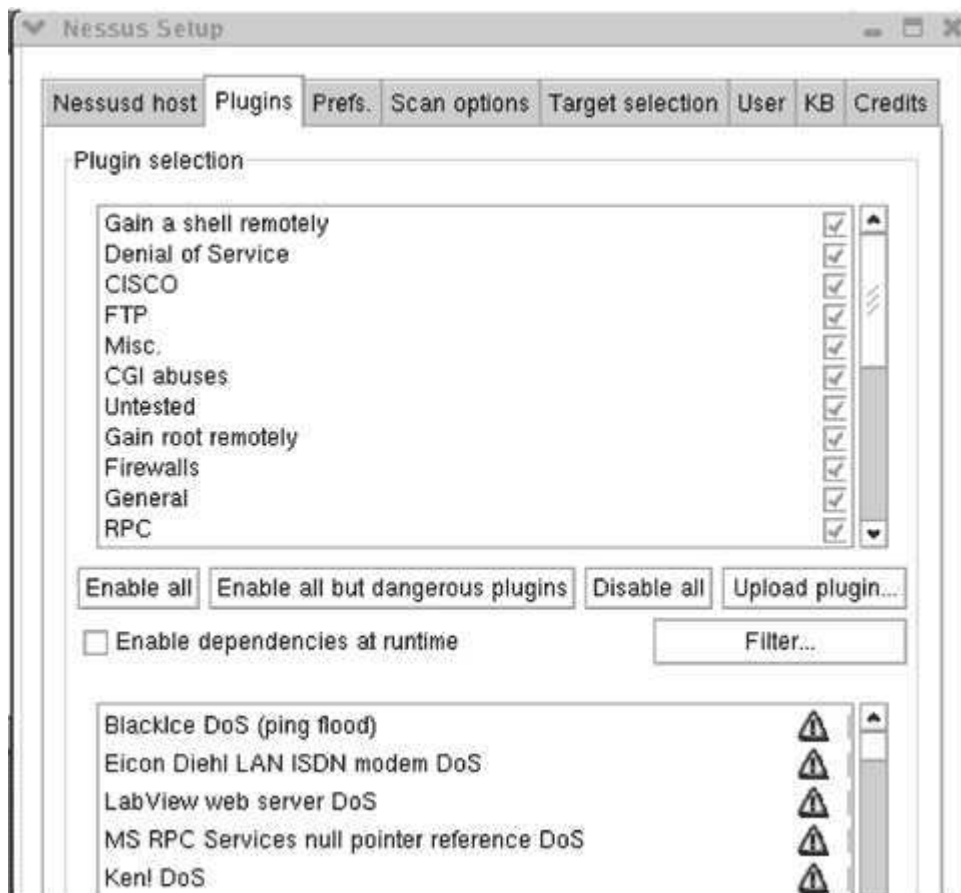


Рис. 5.2. Вкладка встраиваемых модулей Nessus (Plugins)

Вкладка предпочтений Nessus

Большинство серверных опций Nessus конфигурируются с помощью вкладки Preferences ([рис. 5.3](#)). В следующих разделах и подразделах даны подробные сведения об этих опциях.

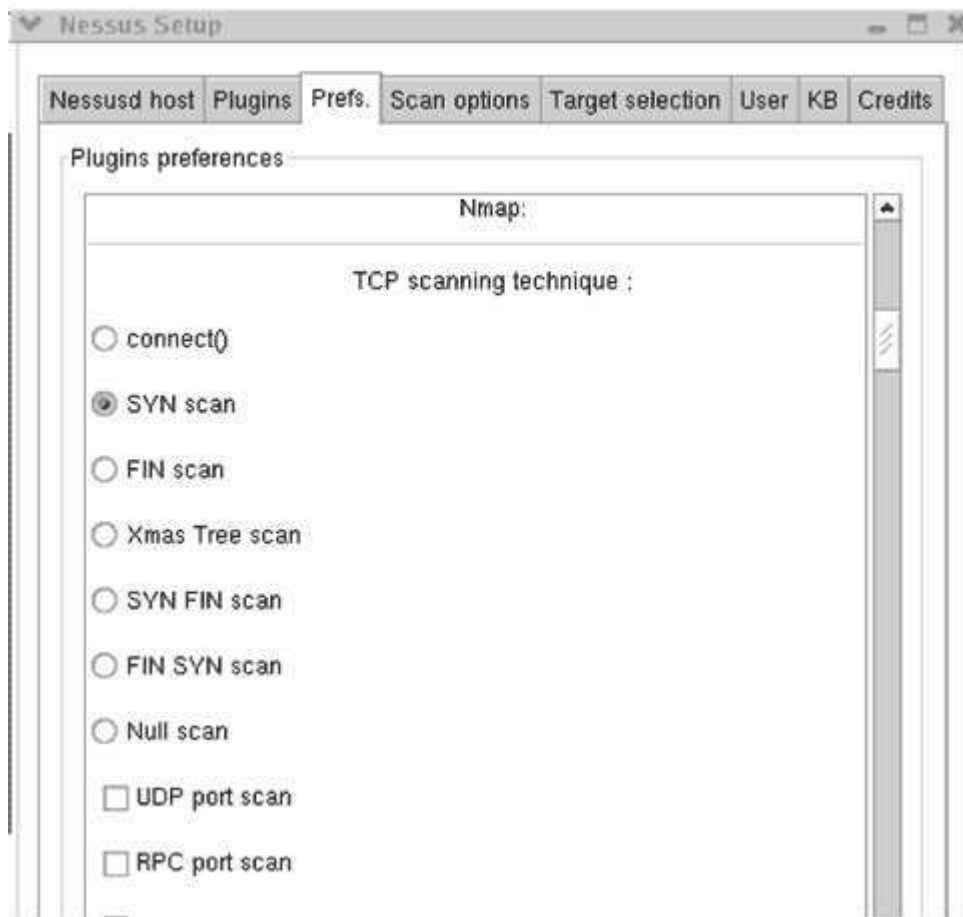


Рис. 5.3. Вкладка предпочтений Nessus (Preferences)

Nmap

Настройки Nmap используются для индивидуального конфигурирования части, отвечающей при выполнении тестов за сканирование портов. Многие из них непосредственно связаны с настройками Nmap, обсуждавшимися в [лекции 4](#), куда и следует обратиться за подробными сведениями о каждой опции.

- TCP scanning technique (Метод сканирования TCP): Задаёт требуемый тип сканирования портов, например, SYN, FIN или Connect;
- Timing policy (Политика управления частотой сканирования): См. раздел "Опции времени для Nmap" в [лекции 4](#).

Можно также ввести маршрутное имя файла результатов Nmap, чтобы Nessus мог использовать эти данные, а не выполнять новое сканирование.

Ping the remote host (Эхо-тестирование удаленного хоста)

Этот выбор позволяет эхо-тестировать машины целевой сети, чтобы прежде всего определить, работают ли они, или просто просканировать все IP-адреса в целевом диапазоне. По умолчанию, Nessus пробует эхо-тесты ICMP и TCP на портах Web и SSL. Если хост включен, он должен ответить на один из этих запросов. Данную настройку можно рекомендовать в большинстве случаев, так как нет смысла впустую тратить время и полосу пропускания, тестируя мертвые адреса. Однако, если вы сканируете извне межсетевого экрана, может оказаться желательным выполнять Nessus без эхо-тестов хостов, чтобы заведомо ничего не пропустить. Можно также задать число попыток, после отсутствия ответа на которые хост считается неработающим. Подразумеваемое значение, равное 10, вероятно, слишком велико для большинства высокоскоростных сетей. Если сканирование производится не через коммутируемое

соединение, уменьшите число попыток до 3, чтобы ускорить процесс сканирования, особенно для больших целевых сетей. Можно также указать, следует ли включать в отчет неработающие хосты. Обычно это нежелательно, поскольку искажается общая статистика сканирования, - получается, что просканировано больше хостов, чем на самом деле есть в сети. Однако включение в отчет "мертвых душ" может быть полезным, если вы хотите получить данные обо всех IP-адресах, с которыми был контакт.

Login configurations (Конфигурации входа)

В этом разделе задаются счета для входа, если вы хотите, чтобы Nessus более глубоко тестировал некоторые сервисы. Стандартное сканирование Nessus проверяет сеть без привлечения каких-либо дополнительных данных о ней, кроме IP-адресов. Однако, если задать входное имя и пароль для конкретного сервиса, то Nessus подвергнет его дополнительным проверкам. Например, если ввести входное имя для Windows-домена (SMB-счет), Nessus будет дополнительно тестировать безопасность этого домена как зарегистрированный пользователь. По умолчанию он проверяет только анонимный сервер FTP с помощью входного имени "anonymous" и стандартного пароля в виде адреса электронной почты. С действующими входными именами можно организовать тестирование FTP, HTTP, IMAP, NNTP, POP2, POP3 и SNMP.

Имеется отдельный раздел для тестирования входных форм HTTP. Можно задать определенный URL и значения заполняемых полей формы. По умолчанию будет проверяться индексный каталог с пустыми полями имени пользователя и пароля.

Brute-force login (Hydra) (Вход методом грубой силы - Hydra)

Этот раздел позволяет воспользоваться дополнительной программой Hydra, которая проверяет целостность паролей вашей системы. Вы предоставляете ей файл входных имен и паролей, а она попытается пройти по всему списку для всех указанных сервисов. Я не рекомендую применять эту опцию, если только вы не готовы иметь дело с последствиями атаки методом грубой силы, которая может блокировать счета многих пользователей после превышения максимально допустимого числа попыток входа. Предпочтительный способ проверки стойкости ваших паролей - автономное применение программ взлома к файлу паролей. Однако может быть полезно протестировать один сервис, который не часто используется, например, FTP или Telnet. Опираясь на Hydra, метод грубой силы можно применять к следующим сервисам: Cisco IOS (стандартный и разрешающий пароли), FTP, HTTP, ICQ, IMAP, LDAP, NNTP, PCNFS, POP2, Rexec, SMB (домен Windows), SOCKS 5, Telnet и VNC.

SMB use host SID to enumerate local users (Использование SMB SID хоста для перебора локальных пользователей)

В этом разделе задается диапазон числовых идентификаторов пользователей, чтобы попытаться получить дополнительную информацию об именах пользователей в домене. По умолчанию проверяются идентификаторы в диапазоне 1000-1020, который в сетях Windows всегда содержит по крайней мере счета административного и гостевого пользователей. Nessus проверит их с пустым паролем и с паролем, совпадающим с входным именем.

Services (Сервисы)

Этот раздел предназначен для тестирования сервисов SSL. Можно задать проверяемые сертификаты и получить отчет об уровне шифрования, который поддерживают ваши web-серверы. Это могут быть локальные серверы, которые все еще допускают старое 40-битное шифрование, что в наше время считается небезопасным для критически важных данных.

Web mirroring (Зеркалирование Web)

Данная настройка позволяет задать, насколько глубоко сканер будет читать web-сайт в поисках каких-либо дефектов или дыр в безопасности. Можно также изменить подразумеваемый начальный каталог.

Misc. Information on the News Server (Прочая информация о сервере телеконференций)

Если имеется сервер Network News (NNTP), расположенный на любом из IP-адресов в целевом диапазоне, Nessus проверит настройки и ограничения, установленные для посылок. Это гарантирует, что ваши серверы телеконференций не могут быть использованы для рассылки спама и иной ненадлежащей деятельности.

Test HTTP dangerous methods (Проверка опасных методов HTTP)

Тест Integrist проверяет, не допускают ли некоторые web-серверы в сети опасные команды вроде PUT и DELETE. По умолчанию данная проверка отключена, так как тест может удалить вашу домашнюю страницу, если сервер отвечает на эти команды.

Ftp writable directories (Каталоги Ftp, допускающие запись)

Проверяются серверы FTP, которые разрешают доступ для записи анонимным пользователям (что никак нельзя считать хорошей практикой). Подразумеваемая настройка означает проверку прав доступа, выданных файловой системой, и реакцию, если оказывается, что запись разрешена. Можно также задать игнорирование информации от файловой системы и пытаться в любом случае записать файл, чтобы убедиться, что доступные для записи каталоги отсутствуют. Как и с тестом Integrist, будьте осторожны с этой опцией, так как дело может закончиться перезаписью файлов на сервере FTP.

SMTP settings (Настройки SMTP)

Эти настройки используются для дополнительного тестирования почтовой системы. Nessus пытается послать поддельные сообщения, чтобы проверить, как реагирует система. Nessus.org используется как подразумеваемый домен, с которого будет приходить тестовая почта, но эту установку можно здесь изменить. Многие почтовые серверы не отвечают, если имя почтового сервера недействительно. Есть смысл изменить этот адрес, если вы являетесь внешним консультантом и хотите, чтобы ваш клиент знал, откуда приходят поддельные сообщения. Однако не используйте собственный домен, если сканируете изнутри организации; это собьет с толку почтовый сервер, который увидит поступающие от самого себя сообщения, и может повлиять на надежность результатов тестирования.

Libwhisker options (Опции Libwhisker)

Эти опции предназначены для использования с дополнительной программой Whisker, которая проверяет целостность web-серверов. Обратитесь к документации программы Whisker за объяснением данных настроек. По умолчанию эти опции отключены.

SMB use domain SID to enumerate users (Использование SMB SID домена для перебора пользователей)

В рамках этой проверки домена Windows делается попытка идентифицировать пользователей на основе их идентификаторов безопасности (SID). В типичных доменах Windows значение SID, равное 1000, назначено администратору, и несколько других стандартных назначений применяются для системных счетов, таких как гостевой. Nessus опрашивает заданный диапазон идентификаторов безопасности, пытаясь экстраполировать имена пользователей.

HTTP NIDS evasion (Обход сетевой системы выявления вторжений при тестировании HTTP)

Этот раздел предоставляет различные методы, позволяющие избежать обнаружения сетевой системой выявления вторжений (NIDS) путем создания и подделки специальных универсальных локаторов ресурсов для атак на web-серверы. Чтобы этим воспользоваться, потребуется дополнительная программа Whisker. Различные тесты пытаются послать странные URL на ваши Web-серверы, чтобы посмотреть, не дадут ли они пользователю возможность выполнить с помощью CGI-процедур ненадлежащие действия. Полное описание этих тестов можно найти в документации Whisker или в статье по адресу <http://www.wiretrip.net/rfp/libwhisker/README>.

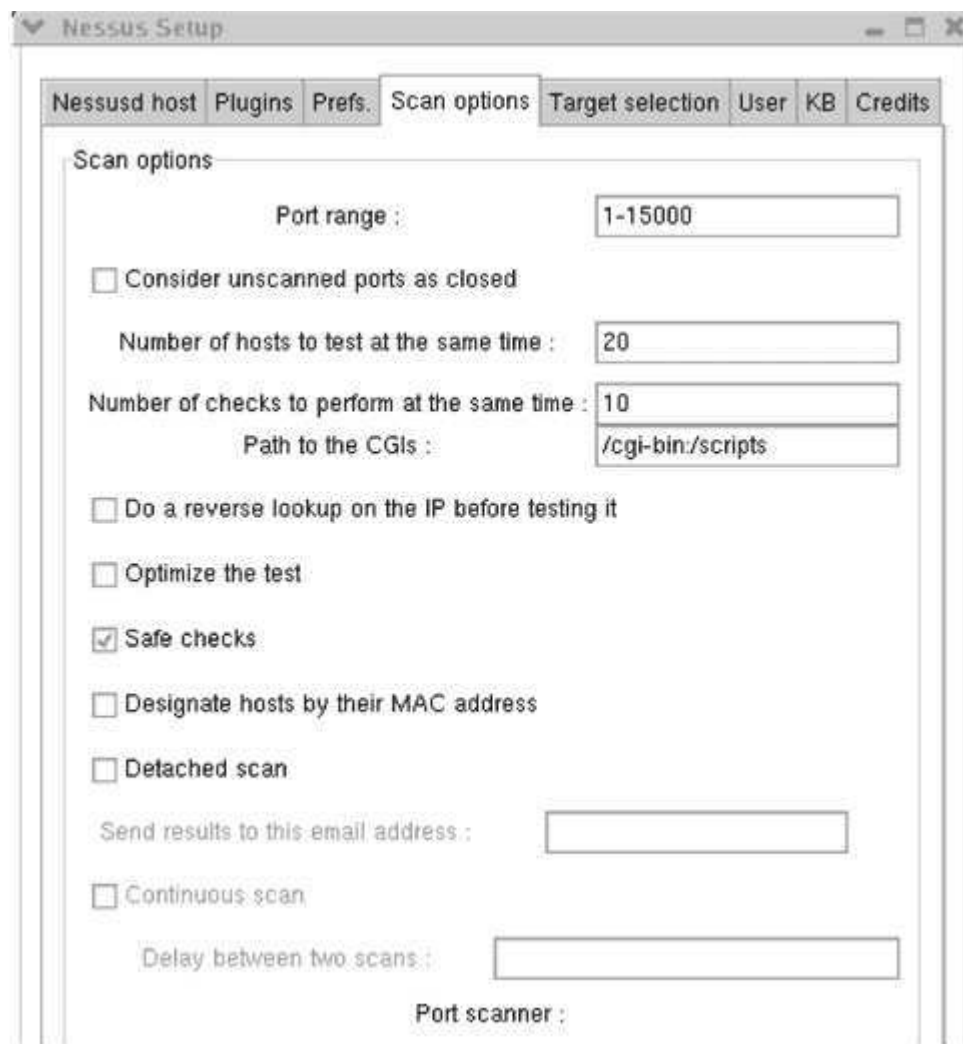
По умолчанию эти методы отключены, так как они имеют тенденцию создавать большой сетевой трафик и генерировать много ложных срабатываний. Однако, если вы применяете в своей сети системы выявления вторжений и хотите понять, работают ли они на самом деле, можно выполнить эти тесты, чтобы посмотреть, будут ли они обнаружены.

NIDS evasion (Обход сетевых систем выявления вторжений)

Этот раздел аналогичен предыдущему, за исключением того что Nessus для обхода сетевых систем выявления вторжений, которые действуют по принципу сопоставления с образцом, проделывает хитрые манипуляции с реальными пакетами TCP, а не с запросами универсальных локаторов ресурсов. Большинство современных сетевых систем выявления вторжений разоблачат эти трюки, но если у вас старая или давно не обновлявшаяся система, то стоит попробовать этот тест, чтобы посмотреть, какова будет реакция. И в этом случае также в отчеты могут попасть подозрительные данные, поэтому для обычного тестирования уязвимостей применять эту опцию не рекомендуется.

Вкладка Scan Options (Опции сканирования)

В отличие от отдельных тестов на вкладке предпочтений, эта вкладка содержит настройки, влияющие на весь процесс сканирования ([рис. 5.4](#)).



The screenshot shows the 'Nessus Setup' window with the 'Scan options' tab selected. The window has a title bar with standard OS controls. Below the title bar is a tabbed interface with tabs for 'Nessusd host', 'Plugins', 'Prefs.', 'Scan options' (active), 'Target selection', 'User', 'KB', and 'Credits'. The 'Scan options' tab contains the following settings:

- Port range :** A text input field containing '1-15000'.
- ☐ **Consider unscanned ports as closed**
- Number of hosts to test at the same time :** A text input field containing '20'.
- Number of checks to perform at the same time :** A text input field containing '10'.
- Path to the CGIs :** A text input field containing '/cgi-bin:/scripts'.
- ☐ **Do a reverse lookup on the IP before testing it**
- ☐ **Optimize the test**
- ☒ **Safe checks**
- ☐ **Designate hosts by their MAC address**
- ☐ **Detached scan**
- Send results to this email address :** An empty text input field.
- ☐ **Continuous scan**
- Delay between two scans :** An empty text input field.
- Port scanner :** A label at the bottom of the settings area.

Рис. 5.4. Вкладка Scan Options в Nessus

Port range (Диапазон портов)

Данный параметр контролирует фазу сканирования портов, задавая целевой диапазон (по умолчанию - 1-15000, что должно охватить большинство обычных сервисов). Если вы желаете поискать "тройские" программы и другие сервисы, действующие на необычно больших номерах портов, подразумеваемый диапазон необходимо расширить и сканировать все 65535 портов TCP и UDP. Следует регулярно (ежемесячно или ежеквартально, в зависимости от размеров сети) выполнять полное сканирование портов всех машин.

Consider unscanned ports as closed (Считать несканированные порты закрытыми)

Данная опция заставляет Nessus объявлять несканированные порты закрытыми. Вы можете что-то пропустить, если (посредством предыдущей опции) не задали достаточно широкий диапазон портов, но зато сканирование выполнится быстрее и с меньшим сетевым трафиком.

Number of hosts to test at the same time (Число одновременно тестируемых хостов)

Задается число хостов, которые Nessus тестирует параллельно. В большой сети возникает соблазн зависить данный параметр и тестировать все хосты одновременно. Однако с некоторого момента это становится контрпродуктивным - в действительности сканирование будет длиться дольше, а может и вообще не закончиться, если завязнет на каком-то одном хосте. На самом деле, на средних серверных машинах (до 2 ГГц) я рекомендую задавать это значение равным 10 хостам вместо подразумеваемого значения 30. Для большинства сканирований это представляется оптимальным значением. Однако, если в вашем распоряжении суперсервер, а сеть очень большая, можно пытаться наращивать это значение, пока это дает видимый эффект.

Number of checks to perform at the same time (Число одновременно выполняемых проверок)

Nessus поддерживает многозадачность не только в смысле одновременного тестирования нескольких хостов, но и в смысле одновременного выполнения нескольких проверок. Подразумеваемое значение 10 представляется разумным, однако, в зависимости от производительности сервера Nessus, можно увеличивать или уменьшать его.

Path to the CGI's (Маршрут к CGI)

Подразумеваемое место, где Nessus будет искать на удаленной системе CGI-процедуры для их тестирования. Если вы используете на машине необычную конфигурацию, то необходимо задать правильный маршрут, чтобы Nessus проверил CGI-процедуры.

Do a reverse lookup on the IP before testing it (Выполнять обратный поиск IP-адреса перед тестированием)

При использовании данной настройки перед проверкой делается попытка выполнить обратный поиск DNS и определить имя хоста для каждого IP-адреса. Это существенно замедляет сканирование и по умолчанию отключено.

Optimize the test (Оптимизировать тестирование)

По умолчанию при выполнении тестов Nessus пытается поступать разумно и не делать проверок, неприменимых к конкретному хосту. Здесь эту опцию можно отключить, чтобы Nessus выполнял все тесты на всех хостах, независимо от того, что покажет сканирование портов.

Safe checks (Безопасные проверки)

По умолчанию всегда устанавливается данный режим. Это значит, что Nessus не будет выполнять никаких небезопасных проверок, которые могут вызвать аварию или как-то иначе повредить серверу. По заголовкам и другой информации определяется, имеет ли хост определенную уязвимость. Я рекомендую всегда выбирать безопасный режим, даже если это приводит к увеличению числа ложных срабатываний.

Designate hosts by their MAC address (Обозначать хосты адресами доступа к среде передачи)

Включите эту опцию, если хотите, чтобы в отчете Nessus хосты обозначались их MAC-адресами, а не адресами IP, как делается по умолчанию. Если в вашей сети имеется хорошая база данных MAC-адресов и если вам трудно соотносить IP-адреса с определенными хостами в связи с применением протокола динамического конфигурирования хостов, подобный выбор поможет вам получить более полезный отчет.

Detached scan (Обособленное сканирование)

Данная возможность позволяет Nessus выполнять сканирование без соединения с клиентом, что полезно для сканирования в необычное время без вмешательства человека. Можно указать, чтобы отчет о сканировании направлялся по определенному адресу электронной почты.

Continuous scan (Непрерывное сканирование)

При использовании этой возможности сканирование запускается на регулярной основе, что полезно для автоматического сканирования сети по расписанию. Задайте интервал между сканированиями в секундах (86400 - для ежедневного сканирования, 604800 - для еженедельного и примерно 2592000 - для ежемесячного). Организовать регулярное сканирование можно и более удобным способом, например при помощи Командного центра Nessus (Nessus Command Center, NCC), описанного в [лекции 8](#). Однако, если вы не хотите устанавливать Web-сервер и базу данных, требующиеся для NCC, рассматриваемая возможность позволяет легко и просто организовать регулярное сканирование.

Port scanner (Сканер портов)

Здесь содержится несколько глобальных настроек для фазы сканирования портов:

- tcp connect() scan: В этом случае вместо Nmap применяется встроенный в Nessus сканер портов. Он быстрее и требует значительно меньше памяти, однако создает больше шума в сети и будет запротоколирован на большинстве сканируемых машин. Кроме того, над его настройками значительно меньше контроля, чем для Nmap.
- Nmap: В этом случае применяется Nmap и соответствующие настройки, заданные во вкладке предпочтений для сканирования портов.
- SYN Scan: Данная возможность была реализована в версии 2.0. Предлагается встроенное сканирование SYN в упомянутом выше режиме tcp connect. Это делает сканирование несколько менее шумным, но недостаток детального контроля по сравнению с Nmap все же остается.
- Ping the remote host: До всех проверок выполняется эхо-тестирование хостов в целевом диапазоне, чтобы проверить, работают ли они.
- scan for LaBrea Tar-pitted hosts: Хосты La Brea Tar-pitted служат для обнаружения сканирования портов и затягивания его до бесконечности, что может замедлить или вызвать аварию сканирования. Данная опция делает попытку обнаружить и обойти хосты с подобной защитой.

Вкладка Target Selection (Выбор цели)

На этой вкладке задаются цели сканирования ([рис. 5.5](#)). Перечислим способы задания целей сканирования:

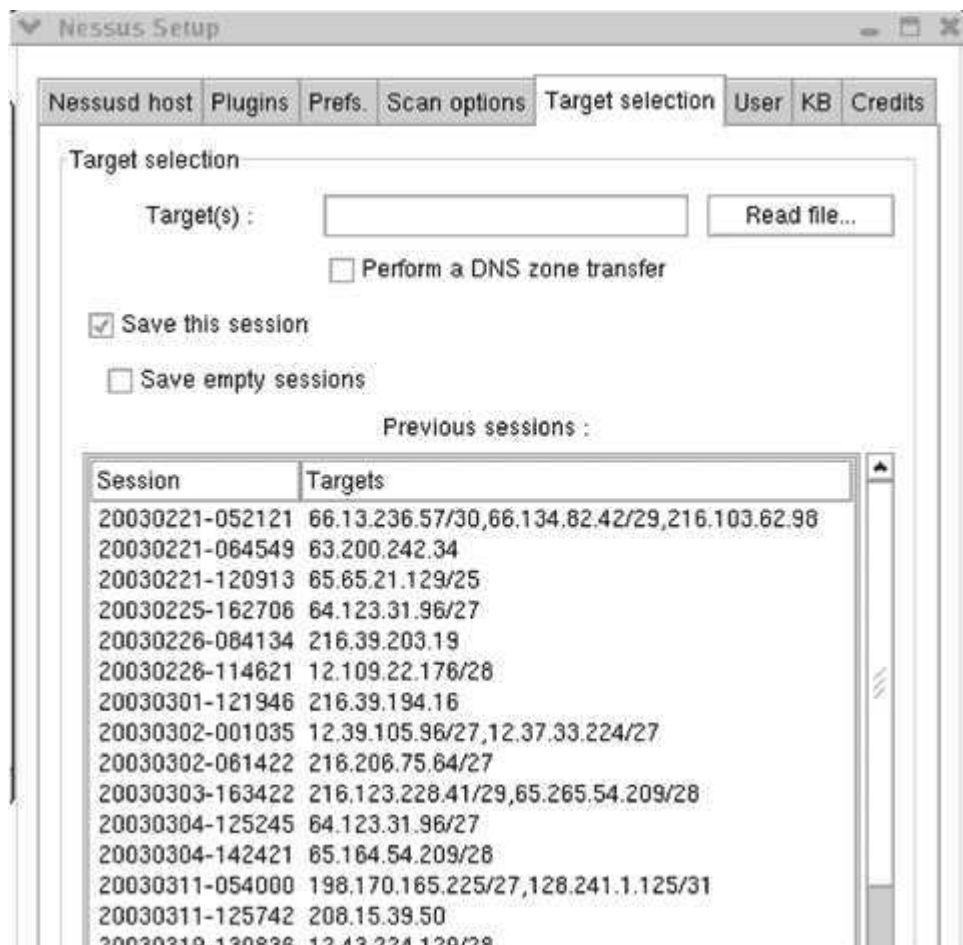


Рис. 5.5. Вкладка Target Selection в Nessus

- Один IP-адрес: 192.168.0.1.
- IP-адреса, разделенные запятыми: 192.168.0.1,192.168.0.2.
- IP-диапазоны, разделенные дефисом: 192.168.0.1-192.168.0.254.
- Стандартная нотация с косой чертой: 192.168.0.1/24 (сеть класса C из 256 .адресов)
- Имя хоста: myhost.example.com.
- Любая комбинация вышеприведенных обозначений, разделенных запятыми: 192.168.0.1-192.168.0.254,195.168.0.1/24.

На этой вкладке можно задать несколько опций.

Read file (Прочитать файл)

Щелкните мышью на этой кнопке, чтобы прочитать цели сканирования из файла. Это должен быть стандартный текстовый файл с адресами, отформатированными как в приведенных выше примерах.

Perform a DNS zone transfer (Выполнить передачу зон DNS)

Делается попытка извлечь файл зон для домена, представленного целевыми IP-адресами. На собственных (немаршрутизируемых) IP-адресах это не работает.

Save this session (Сохранять сеанс)

Поддерживает запись целей сканирования и настроек, чтобы их можно было восстановить в будущем. По умолчанию включено.

Save empty sessions (Сохранять пустые сеансы)

Сохранять сеансы, даже если они не содержат данных, например, IP-диапазон без единого работающего хоста в нем.

Previous sessions (Предыдущие сеансы)

Выдаются все предыдущие сеансы; можно загрузить любой из них, выбрав мышью соответствующий элемент списка.

Вкладка User (Пользователь)

На этой вкладке отображаются все пользователи сервера Nessus и все ассоциированные с ними правила (например, возможность входа только с определенного IP-адреса). Счета пользователей создаются с помощью процедуры `nessus-adduser`, но на этой вкладке всегда можно отредактировать или добавить правила для любого существующего пользователя.

Вкладка KB (Knowledge Base) (База Знаний)

Эта вкладка содержит конфигурацию и элементы управления для базы знаний Nessus ([рис. 5.6](#)). Это одна из самых полезных возможностей, предлагаемых Nessus. По умолчанию она отключена. Чтобы ее включить, необходимо поднять флажок "Enable KB saving". База знаний отслеживает все выполненные сканирования. Затем, когда вы захотите снова выполнить сканирование, Nessus использует сохраненные данные, чтобы определить, какие хосты сканировать и какие тесты на каждом из них выполнять. Доступные настройки описаны ниже.

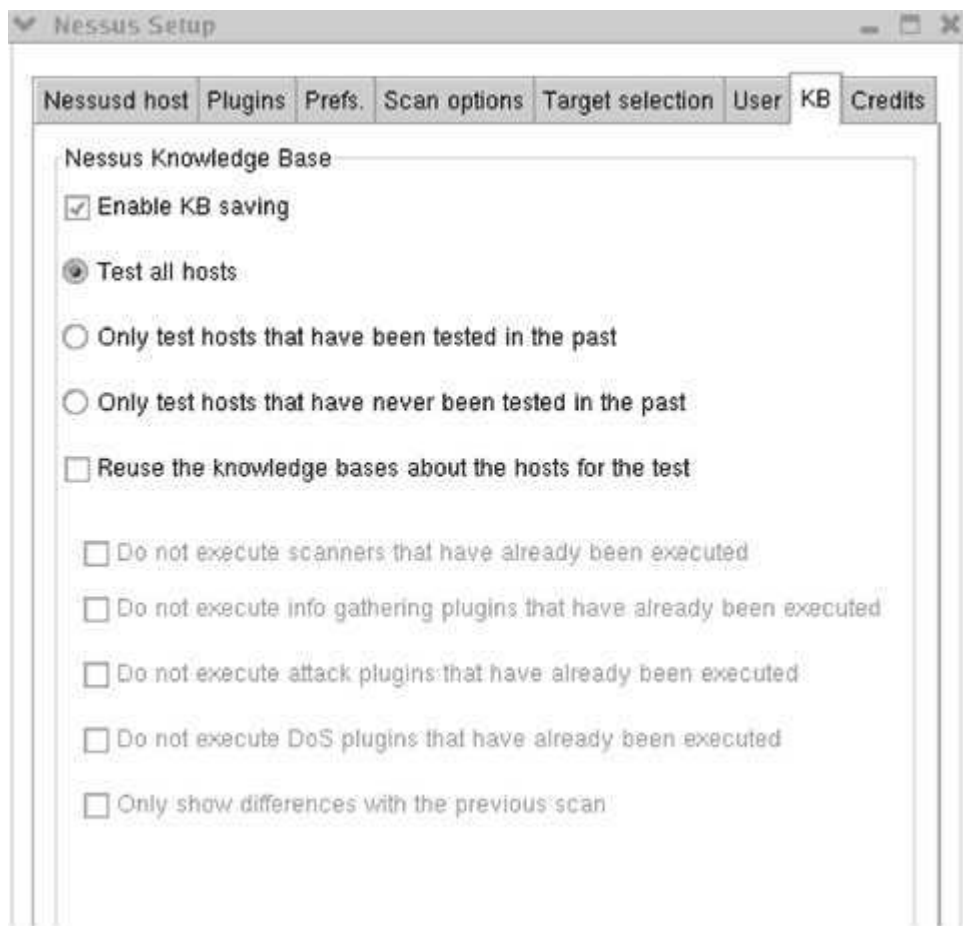


Рис. 5.6. Вкладка Knowledge Base в Nessus

Test all hosts (Тестировать все хосты)

Используется по умолчанию. Данные базы знаний будут сохраняться, но каждый хост будет тестироваться полностью.

Test only hosts that have been tested in the past (Тестировать только хосты, тестировавшиеся ранее)

Эта настройка заставляет Nessus тестировать только те хосты в целевом диапазоне, которые тестировались ранее. Это означает, что Nessus не будет искать новых хостов. Тем самым несколько снижается сетевой трафик, но машины, появившиеся в сети после предыдущего тестирования, проверены не будут.

Test only hosts that have never been tested in the past (Тестировать только хосты, не тестировавшиеся ранее)

Противоположно предыдущей настройке; в целевой сети проверяются только новые хосты. Это полезно для быстрой проверки новых машин в сети, без сканирования существовавших ранее.

Reuse the knowledge bases about the hosts for the test (Использовать при тестировании базы знаний о хостах)

Исключается выполнение определенных проверок на основе заданных параметров и того, что было найдено ранее.

- Do not execute scanners that have already been executed (Не сканировать повторно). Пропускается фаза сканирование портов, полагаясь на результаты предыдущих сканирований.
- Do not execute info gathering plug-ins that have already been executed (Не запускать повторно модули сбора информации). Nessus не будет запускать модули сбора информации, работавшие во время предыдущих сканирований. Все новые модули сбора информации, добавленные после предыдущего сканирования, будут запущены.
- Do not execute attack plug-ins that have already been executed (Не запускать повторно модули атак). То же, что предыдущая настройка, но применительно к модулям атак.
- Do not execute DoS plug-ins that have already been executed (Не запускать повторно модули атак на доступность). Аналогично предыдущей настройке.
- Only show differences with the previous scan (Показывать только отличия от предыдущего сканирования). Выполняется дифференциальное сканирование; отчет показывает различия между двумя последними сканированиями. Полезно, если нужно посмотреть, что изменилось в сети со времени последнего сканирования. То же можно сделать и с помощью Командного центра Nessus, описанного в [лекции 8](#).

Max age of a saved KB (in secs) (Максимальный возраст сохраненной базы знаний (в секундах))

Эта настройка не позволяет серверу использовать слишком старые базы знаний сканирования. Подразумеваемый максимальный возраст - 86400 секунд (одни сутки). Можно увеличивать это значение вплоть до 60 дней (5184000 секунд). Дальнейшее увеличение возраста нецелесообразно, так как будут использоваться слишком старые данные.

При помощи базы знаний можно ускорить и упростить сканирование, однако ее возможности следует применять избирательно и регулярно (желательно - ежемесячно) выполнять полное сканирование.

Опции оперативного управления процессом сканирования

Во время сканирования Nessus выдает экран, отображающий состояние процесса сканирования. Можно видеть каждый проверяемый хост и то, насколько далеко продвинулись проверки. Показывается также выполняемый в данный момент встраиваемый модуль. Обычно элементы изображения сменяются очень быстро, но иногда картинка застревает на определенном модуле. Можно прервать тестирование текущего хоста, щелкнув мышью на кнопке Stop с правой стороны экрана ([рис. 5.7](#)). Можно также щелкнуть мышью на кнопке "Stop the whole test" внизу, чтобы полностью остановить тестирование и получить отчет о полученных на текущий момент результатах.



Рис. 5.7. Экран процесса сканирования Nessus

NessusWX: Клиент Nessus для Windows

NessusWX

Автор/основной контакт: Victor Kirhenshtein

Web-сайт: <http://www.securityprojects.org/nessuswx>

Платформы: Windows 98, NT, 2000, XP

Лицензия: GPL

Рассмотренная версия: 1 .4.4

Другие ресурсы: nessuswx.nessus.org

NessusWX - это Nessus-клиент под Windows. Он представляет только клиентскую часть программы. К сожалению, Nessus пока не предлагает полное решение для тестирования уязвимостей из-под Windows. Имеющая прочные позиции компания Network Security выпускает портированную под Windows коммерческую версию Nessus, называемую NeWT. Однако, если вы не можете позволить себе ее приобретение, вам придется использовать сервер Nessus на платформе UNIX для подключения своего клиента NessusWX.

NessusWX - не просто клон UNIX-клиента. Помимо доступа к серверу Nessus с Windows-машины, NessusWX предоставляет ряд возможностей, отсутствующих у UNIX-клиента. Он также реализует некоторые настройки более логичным и удобным для применения образом. На самом деле, многие считают NessusWX лучшим вариантом использования Nessus. Помните только, что для сканирований вам все равно необходимо иметь сервер Nessus под UNIX. Кроме того, поскольку NessusWX является независимым программным продуктом, его возможности иногда будут немного отставать от "родной" UNIX-платформы. Перечислим некоторые привлекательные дополнительные возможности NessusWX.

- Поддержка MySQL: Можно экспортировать результаты сканирования Nessus в базу данных MySQL - либо прямо во время сканирования, либо сохраняя результаты в формате MySQL для последующей обработки.
- Дополнительные форматы отчетов: NessusWX позволяет сохранять отчеты Nessus как файл PDF. Запланирована поддержка формата Microsoft Word и других форматов файлов.
- Операции с отчетами: Можно уточнять результаты сканирования, например, пометить определенные тревожные сообщения как ложные срабатывания, чтобы они не появлялись в отчете. Это полезно, если ваш босс раздражается, когда видит отчет с несколькими дырами в безопасности, а вам необходимо объяснять, что это ложные срабатывания и на самом деле дыр нет.
- Более простой интерфейс пользователя: По моему мнению, пользовательский интерфейс NessusWX более дружелюбный, чем у Nessus, а параметры и предпочтения представлены упрощенным образом. Однако, если вы привыкли к применению UNIX-интерфейса, это может вас запутать, так как некоторые вещи выглядят немного иначе. Но в целом Windows-реализация лучше, чем порой путанные и избыточные опции в UNIX-клиенте.

Установка NessusWX

Установить NessusWX очень просто. Возьмите файл с компакт-диска или загрузите бинарный самораспаковывающийся файл с nessuswx.nessus.org/index.html#download.

Можно также получить пакеты с исходными текстами, если вы хотите поработать с ними и посмотреть, нельзя ли их улучшить. Но если вы не намерены это делать, то для получения исходных текстов реальной причины нет. Просто щелкните мышью на файле, и программа установки будет направлять ваши действия.

Применение Windows-клиента NessusWX

По внешнему виду интерфейс NessusWX отличается от исходного UNIX-клиента ([рис. 5.8](#)). Вы не увидите описанных ранее вкладок, но все рассмотренные конфигурационные опции доступны и в этой версии. В клиенте NessusWX проведено более четкое разграничение между настройками, которые контролирует клиент, и теми, что контролирует сервер. Контролируемые сервером настройки находятся в текстовом файле `nessus.rc` и являются глобальными настройками, в то время как настройки клиентской стороны по большей части относятся к определенным типам сканирования. Содержимое файла `nessus.rc` можно просмотреть и отредактировать, выбирая пункт Server Preferences в меню Communications.

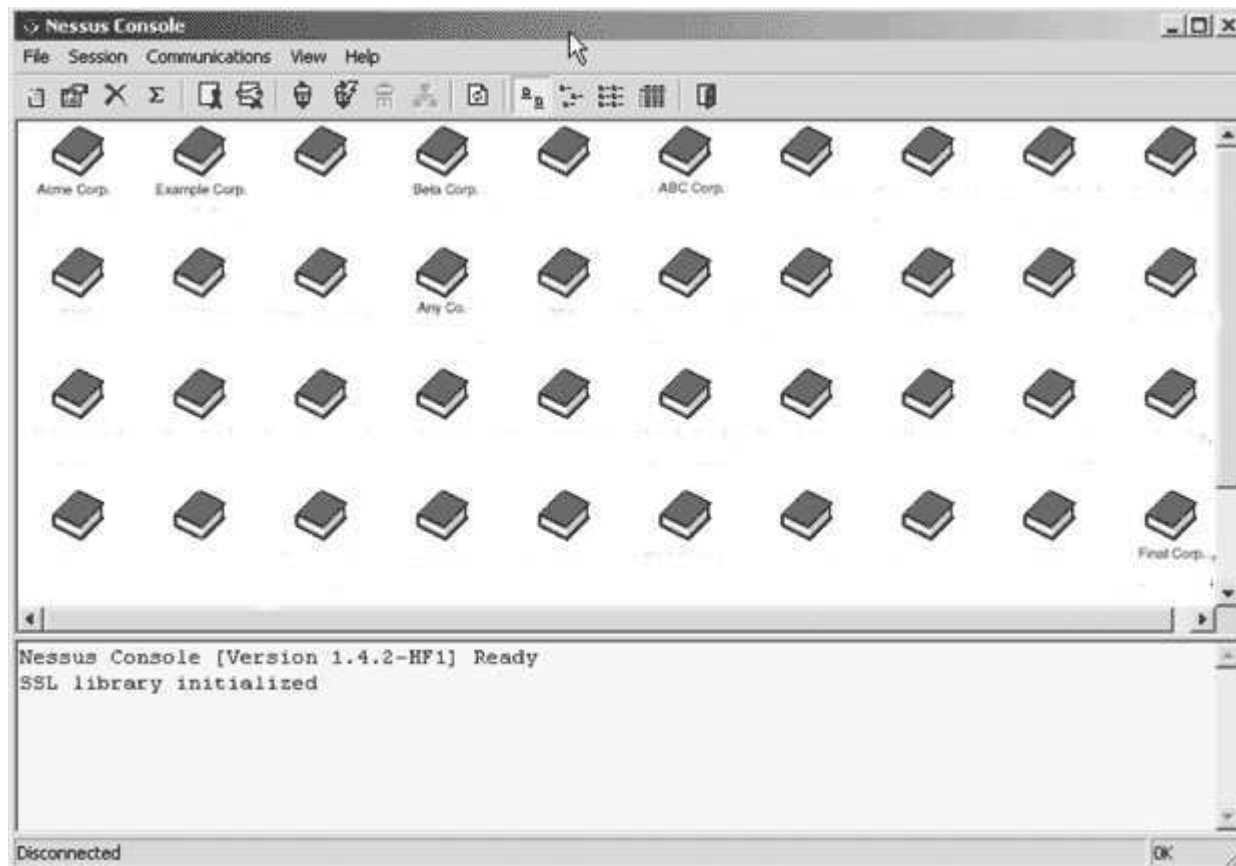


Рис. 5.8. Интерфейс NessusWX

Другое удобное свойство клиента Windows - возможность интерактивного создания конфигураций сканирования, называемых сеансами (sessions), с последующим подключением к серверу Nessus. Это означает, что конфигурации можно создавать автономно, вне соединения с сервером. Однако, чтобы начать сканирование или просмотреть и сконфигурировать предпочтения серверной стороны, необходимо подключиться к серверу и войти в него. Для этого в меню Communications щелкните мышью на Connect. Можно также использовать опцию Quick Connect (Быстрое соединение) и задать подразумеваемый сервер, чтобы всегда входить в него. Клиент также запомнит ваш пароль и входное имя, чтобы их не нужно было вводить каждый раз. (Это удобно, но, несомненно, менее безопасно!)

Создание профиля сеанса

В первую очередь целесообразно создать профиль сеанса. Это цель или совокупность целей, которые вы хотите просканировать.

1. В меню Profile (Профиль) выберите New (Новый). В появившемся диалоговом окне введите имя сеанса сканирования. Имеет смысл выбрать содержательное имя, поскольку оно будет фигурировать в шапке отчета сканирования.
2. Затем появится окно Session Properties (Свойства сеанса) ([рис. 5.9](#)). Не забудьте щелкнуть мышью на кнопке Apply (Применить) после ввода данных в каждой вкладке.

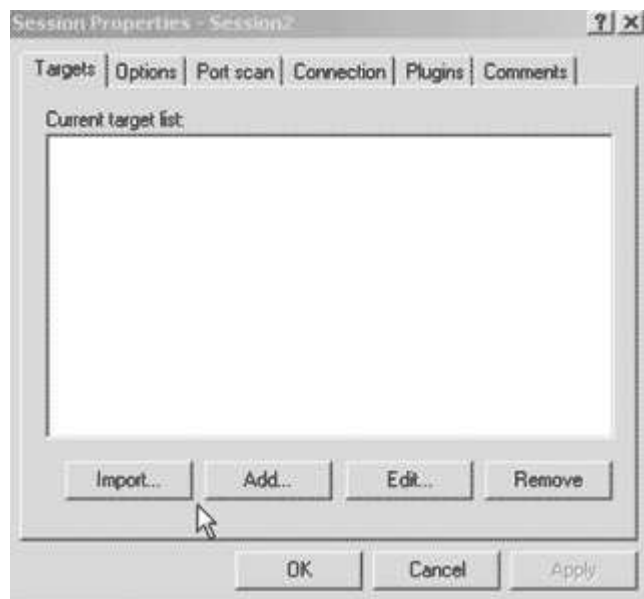


Рис. 5.9. Экран Session Properties в NessusWX

3. Щелкните мышью на Add (Добавить), чтобы задать адреса для сканирования. Отметим простой формат ввода различных диапазонов. Можно также импортировать список целей, вводя имя содержащего их текстового файла.
4. Щелкните мышью на Remove (Удалить), чтобы удалять хосты с экрана состояния, когда заканчивается их сканирование, или выберите режим без показа выполняемых модулей.
5. Затем щелкните мышью на вкладке Options ([рис. 5.10](#)), чтобы задать опции сканирования, по большей части аналогичные опциям UNIX-клиента.

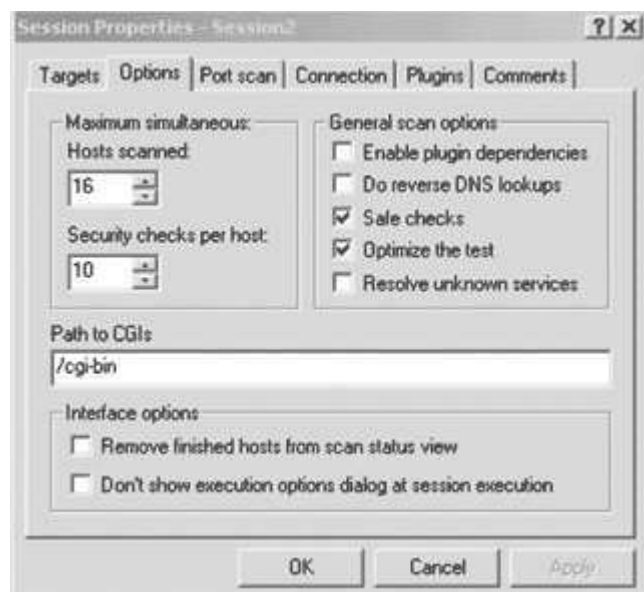


Рис. 5.10. Вкладка Scan Options в NessusWX

6. Посредством вкладки Port scan конфигурируется фаза сканирования портов (рис. 5.11). Подразумеваемая настройка включает только общепотребительные серверные порты (1-1024), а не диапазон 1-15000, как в UNIX-клиенте. Конечно, вы можете изменить данный диапазон по своему желанию. Доступны еще две настройки: Well-known services (Общеизвестные сервисы) и Specific range (Определенный диапазон). Последняя позволяет задать любой требуемый диапазон портов.

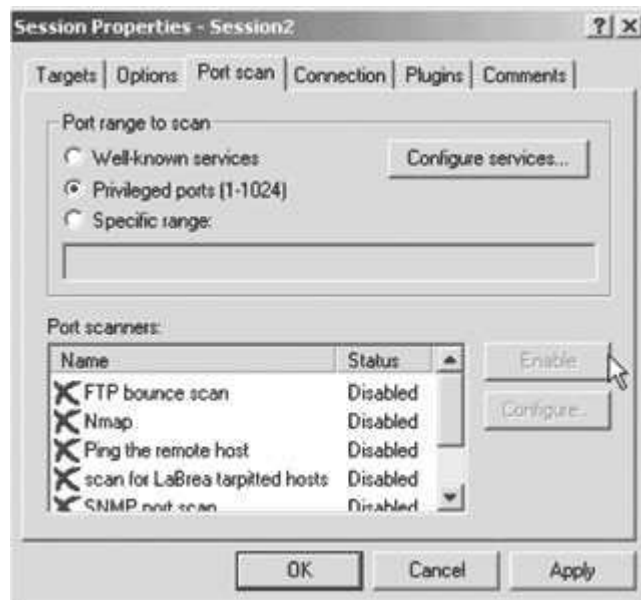


Рис. 5.11. Опции Port Scan в NessusWX

7. После входа в сервер вкладка Plugins позволит выборочно включать или отключать определенные модули или целые группы модулей. На самом деле, можно сконфигурировать некоторые параметры модулей, такие как подразумеваемый пароль, подразумеваемые каталоги и т.д., прямо из клиента, что невозможно в UNIX-клиенте.
8. Имеется также вкладка Comments. Это приятное добавление позволяет документировать различные сеансы сканирования, чтобы позднее можно было вспомнить, что вы пытались делать.
9. Щелкните мышью на OK, чтобы закрыть окно.
10. Сконфигурировав все параметры сканирования, сделайте двойной щелчок мышью на иконке профиля сканирования, который вы хотите использовать, а затем щелкните мышью на кнопке Execute. Должно начаться сканирование, ход которого отображается экране состояния (рис. 5.12)

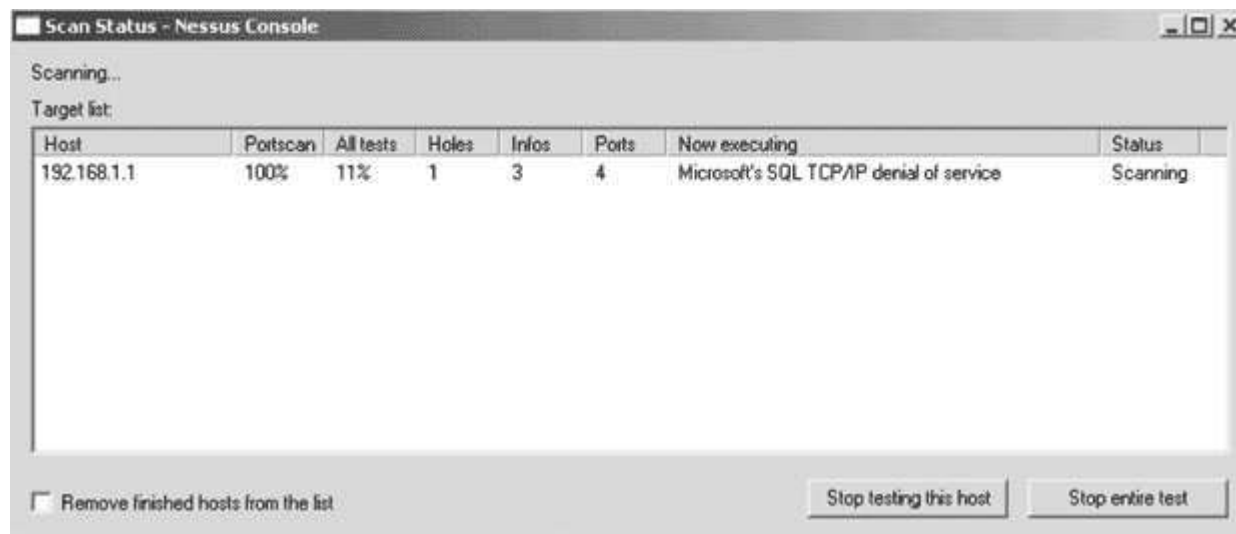


Рис. 5.12. Экран хода сканирования NessusWX

Можно заметить, что экран состояния сканирования для NessusWX подробнее, чем у UNIX-клиента. На нем отображается, в частности, процент выполненного сканирования портов. UNIX-клиент показывает прогресс сканирования только в виде полосы, что не является точным. Кроме того, показывается, сколько осталось до завершения тестов, а также текущее значение общего числа открытых портов, информационных сообщений, предостережений и найденных дыр в безопасности каждого хоста. Так же как в UNIX-клиенте, можно прервать сканирование отдельных хостов или весь тест.

Отчеты NessusWX

Чтобы получить, создать и просмотреть отчеты NessusWX, щелкните правой кнопкой мыши на профиле сканирования и выберите Results из контекстного меню. Несколько опций этого экрана позволяют управлять выводом отчетов. Можно выбрать сортировку отчета по хостам или по уязвимостям. Можно задать удаление из отчета ложных срабатываний или включение конфигурации сканирования, чтобы можно было вспомнить, какие настройки использовались для получения этих результатов. Не составляет труда сделать так, чтобы отчет содержал только открытые порты и сообщения безопасности низкого, среднего или высокого уровней серьезности, снимая выбор соответствующих флажков. Все это позволяет гибко управлять формой представления отчетов, что важно, если вы представляете эти отчеты нетехническому руководству, аудиторам, заказчикам или другим непрофессионалам.

Опции отчетов в NessusWX включают форматы .nsr (бывший "родной" формат Nessus), .nbe, html, обычный текст и .pdf. Все результаты хранятся в базе данных, поэтому можно легко извлечь результаты прошлых сканирований. Можно также сравнить результаты двух сканирований с помощью опции diff. Базовый отчет HTML включает ряд дополнений по сравнению с HTML-отчетами в UNIX. Он включает имя профиля, показывающее, что именно сканировалось, а также временной штамп и другие статистические данные, такие как длительность сканирования. Кроме того, результаты сканирования можно упорядочить по IP-адресам, как упоминалось ранее, что существенно облегчает поиск определенного хоста (это подтвердит любой, кто пытался упорядочить расположенные случайным образом результаты сканирования в UNIX). К сожалению, отсутствуют встроенные HTML-ссылки, которые имеются в UNIX-отчетах, что могло бы существенно облегчить навигацию в отчете. (Будем надеяться, что кто-нибудь сумеет объединить лучшее из отчетов UNIX и Windows.) Кроме того, все результаты сканирования можно поместить в базу данных и создать собственные отчеты с помощью NCC, как описано в [лекции 8](#).

Примеры конфигураций сканирования Nessus

При столь большом выборе настроек непросто сообразить, что делать при первом сканировании. Для детального изучения всех опций требуется некоторое время, но мы приведем несколько примеров конфигурации, которые должны давать хорошие результаты для наиболее типичных сетевых конфигураций.

Пример конфигурации 1: Внешнее сканирование множества IP-адресов без межсетевого экрана

Это простейшая из возможных конфигураций, которая требует наименьших изменений в подразумеваемой конфигурации Nessus.

- Предпочтения: Оставьте все открытым настежь; никакой скрытности на самом деле не нужно, однако сканирование SYN снизит сетевой трафик.
- Опции сканирования: При сколько-нибудь заметном числе хостов лучше применять встроенное сканирование SYN, так как сканирование с помощью Nmap может потребовать много времени.
- Оставьте все остальные опции в подразумеваемом состоянии.

Пример конфигурации 2: Внешнее сканирование сети с одним внешним IP-адресом межсетевого экрана

Это немного сложнее и требует некоторой скрытности, чтобы пакеты сканирования попали за межсетевой экран.

- Предпочтения: Используйте Nmap для SYN-сканирования с фрагментацией пакетов. Для единственного IP-адреса память и время не составляют проблемы.
- Опции сканирования: Не делайте эхо-тест хоста, так как большинство межсетевых экранов отбрасывают подобные пакеты, и вы не получите никаких результатов.
- Оставьте все остальные опции в подразумеваемом состоянии. Если вы ничего не получите в ответ, попробуйте сканирование без включенного сканирования портов.

Пример конфигурации 3: Внешнее сканирование сети с несколькими общедоступными IP-адресами - для межсетевого экрана и в демилитаризованной зоне

- Предпочтения: Используйте Nmap для SYN-сканирования с фрагментацией пакетов.
- Опции сканирования: Выполните эхо-тестирование хостов, чтобы исключить недействующие IP-адреса в демилитаризованной зоне. Для целевых сетей из более чем 20 хостов используйте встроенное сканирование SYN.
- Оставьте все остальные опции в подразумеваемом состоянии. Если вы ничего не получите в ответ, попробуйте сканирование без сканирования портов.

Пример конфигурации 4: Несколько внешних IP-адресов с сетевой системой обнаружения вторжений

- Предпочтения: Можно попробовать несколько методов обхода системы обнаружения вторжений. Можно также применить экзотические типы сканирования, такие как FIN и XMAS, если общедоступные серверы не являются Windows-машинами. Стоит попробовать растягивание интервалов между пакетами сканирования, хотя это и затянёт сканирование.
- Опции сканирования: Откажитесь от сканирования портов, иначе наверняка сработает система обнаружения вторжений.
- Встраиваемые модули: Стоит отключить самые шумные встраиваемые модули, такие как "потайные входы".

Пример конфигурации 5: Внутреннее сканирование позади межсетевого экрана

.При таком сканировании для вас важен состав генерируемых данных, а скрытность не имеет большого значения (поскольку вы уже находитесь позади межсетевого экрана).

- Предпочтения: Сработает простое SYN-сканирование, так как не нужно заботиться о прохождении через межсетевой экран. Фрагментация пакетов не нужна, поскольку это замедлит сканирование. Если вы находитесь в сети Windows, введите свои входные данные для домена, чтобы Nessus мог проверить настройки ваших пользователей Windows. Можно выполнить одно сканирование с входными данными и одно без таковых, чтобы увидеть, что получит некто, не располагающий информацией о пользователях, при простом подключении к вашей ЛВС.
- Опции сканирования: Для большого числа хостов используйте встроенное SYN-сканирование. Эхо-тестируйте удаленные хосты, чтобы быстро исключить недействующие IP-адреса.
- Встраиваемые модули: Стоит отключить некоторые категории модулей, неприменимые для внутреннего сканирования, такие как подразумеваемые системные счета UNIX (если у вас нет внутренних UNIX-машин), и то же для модулей Windows, если все машины работают под управлением UNIX.

Маршрутизаторы и межсетевые экраны брать в расчет не нужно, если только у вас нет внутренних сегментов ЛВС с межсетевыми экранами. Если вы не используете Novell Netware, отключите эту категорию модулей. Отключите все остальное, что не соответствует вашей внутренней среде ЛВС.



Уголок кодировщиков Флэми Теха:

Написание собственных процедур Nessus

Как упоминалось выше, Nessus можно индивидуализировать и расширить для ваших конкретных нужд, поскольку его исходные тексты открыты. На самом деле, Nessus даже легче расширять, чем другие программы с открытыми исходными текстами, так как он включает собственный встроенный интерпретируемый язык, называемый Nessus Attack Scripting Language (NASL). NASL позволяет легко и быстро писать новые тесты для Nessus-сканирования, не погружаясь во внутреннее устройство Nessus и не занимаясь сложным программированием.

Примечание: Тем не менее, прежде чем браться за NASL, необходимо владеть основами программирования (в частности, языком программирования Си). NASL очень похож на Си с многочисленными изъятиями - в нем нет, например, структур и деклараций переменных. Это облегчает быстрое написание новых процедур для проверки некоторых условий.

Процедуры NASL очень похожи на любые другие программы с переменными, условными инструкциями и функциями, которые можно вызывать. Благодаря Рено и его команде, создавшим множество функций, которые можно использовать для упрощения работы, вам не придется самостоятельно придумывать, как изготовить пакет или проверить открытый порт.

Каждая процедура состоит из двух частей. Первая является разделом регистрации, который Nessus использует для целей документирования. В нем вы информируете Nessus, какого типа эта процедура, и предоставляете некоторые сведения о ней, чтобы пользователи знали, что она делает. Второй раздел является разделом атаки. Именно здесь вы реально выполняете код по отношению к удаленной машине и что-то делаете с результатами.

Предположим для примера, что в вашей сети возникли реальные проблемы с Yahoo Messenger. Запуск Nessus или сканера портов показывает открытые порты, но вы хотите получить специальное уведомление, когда проявится порт Yahoo.

Для этой цели можно написать на NASL индивидуальную процедуру Nessus. На машинах с запущенной программой Yahoo Messenger открыт порт 5101, поэтому с помощью функции NASL `get_port_state()` можно легко и быстро найти машины, выполняющие эту программу, и известить об этом. Ниже представлен пример программы на NASL для решения поставленной задачи. Все строки с символом `#` в начале являются комментариями и не обрабатываются интерпретатором NASL.

```
# Это раздел регистрации
# Проверка на Yahoo Messenger
#
if(description)
{
# Это раздел регистрации и содержит информацию для Nessus
script_name(english:"Looks for Yahoo Messenger Running");
script_description(english:"This script checks to see if Yahoo Messenger is running");
script_summary(english:"connects on remote tcp port 5101");
script_category(ACT_GATHER_INFO);
script_family(english:"Misc.");
script_copyright(english:"This script was written by Tony Howlett");
exit(0);
}
# Это раздел атаки
# Проверить, открыт ли на удаленной системе порт 5101
# Если открыт, то вернуть уведомление
```



```
port = 5101;
if(get_port_state(port));
{
report = "Yahoo Messenger is running on this machine!";
security_warning(port:5101, data:report);
}
# Конец
```

Вот и все. В этой простой процедуре используются два предположения. Первое состоит в том, что порты удаленной машины были просканированы по крайней мере до номера 5101, так как функция `get_port_state` могла бы ошибочно вернуть истину для порта 5101, если его состояние было неопределенным. Предполагается также, что машина с открытым портом 5101 выполняет Yahoo Messenger, хотя это может быть и некоторое другое приложение. При желании можно добавить дополнительную логику для проверки этого, перехватывая заголовок или некоторую часть ответа и анализируя его характеристики.

Это очень простой пример. С помощью NASL можно сделать значительно больше. Обратитесь к оперативной справке NASL, чтобы получить дополнительную информацию обо всех функциях, которые можно использовать, и синтаксисе языка. Существует прекрасный учебник, написанный самим Рено, расположенный по адресу <http://www.nessus.org/doc/nasl.html>.

Особенности сканирования уязвимостей

Теперь, когда вы полностью понимаете все возможности, можно приступить к сканированию. Но прежде чем вы отправите пакеты, позвольте несколько слов об ответственном сканировании. Хотя я уже упоминал об этих вопросах в [лекции 4](#), тестирование уязвимостей имеет некоторые особенности. Сканирование портов является довольно невинной деятельностью, хотя эта активность и раздражает, когда вы видите ее зафиксированной в журналах. Однако тестирование уязвимостей может быть существенно более разрушительным, приводя к авариям серверов, разрывая соединения с Интернетом, или даже удаляя данные (например, тест `Integrist`). Многие из тестов Nessus специально созданы для организации атак на доступность. Даже с включенной опцией безопасной проверки тесты могут вызывать проблемы на некоторых системах. В связи с этим дадим несколько рекомендаций.

Не сканируйте без разрешения

Вы никогда не должны сканировать сеть, которая не находится под вашим непосредственным контролем, или если вы не имеете явного разрешения от владельца. Некоторые из видов активности, инициированной Nessus, могут юридически рассматриваться хакерским взломом (особенно при включенной опции атак на доступность). Если вы не хотите быть обвиненным по закону, отвечать в суде или отвечать на жалобы со стороны поставщиков Интернет-услуг, необходимо всегда получать разрешение на сканирование. Сторонние консультанты должны обязательно получать письменное разрешение со всеми необходимыми оговорками. Образец бланка такого документа имеется в приложении D. Внутренний персонал должен убедиться, что они имеют право сканировать все машины в целевом диапазоне. Согласуйте это должным образом с другими должностными лицами, такими как администраторы межсетевых экранов и персонал службы информационной безопасности.

Убедитесь, что все резервные копии актуальны

Вы всегда должны быть уверены, что ваши резервные копии актуальны, но это вдвойне важно при сканировании уязвимостей, на тот случай, если сканирование приведет к проблемам с сервером. Выполнение сканирования Nessus сразу после резервного копирования гарантирует, что вы сможете восстановить самую свежую версию. Кроме того, убедитесь, что вы не сканируете во время резервного копирования. Это не только может привести к порче данных резервной копии, но и существенно замедлит оба процесса.

Планируйте время сканирования

В продолжение предыдущего замечания, не забывайте координировать время проведения сканирования для получения требуемых результатов с минимальным влиянием на других служащих. Сканирование почтового сервера в 8 часов утра, когда все стремятся получить свою почту, едва ли повысит вашу популярность среди персонала. Планируйте сканирование постоянно включенных серверов на нерабочее время и старайтесь избегать наложения с

другими операциями по администрированию системы и повседневной деятельностью (сканировать сеть бухгалтерии накануне сдачи годового отчета вряд ли разумно). Сканирование внутренних машин, видимо, придется выполнять в рабочее время - либо следует договориться с каждым пользователем, чтобы он оставил машину включенной в конце рабочего дня. Для проведения сканирования в рабочее время лучше всего использовать обеденный перерыв, когда в сети будет работать минимальное количество людей.

Избегайте избыточного сканирования

Планируйте сканирование так часто, как сочтете необходимым, но не полагайте автоматически, что ежесуточное сканирование сделает вашу сеть более безопасной. Не проводите сканирование с такой частотой, если вы не в состоянии интерпретировать отчеты сканирования и реагировать на них каждый день, так как в итоге вы получите только избыточный сетевой трафик. Выбор частоты сканирования должен основываться на возможностях вашего персонала по обработке результатов. Я рекомендую делать это как минимум раз в месяц, но если ваша сеть используется очень активно, то лучше сканировать уязвимости еженедельно. Аналогично, если у вас очень небольшая внешняя сеть, то сканирование может проводиться раз в квартал. Ежедневное сканирование, скорее всего, избыточно, если только у вас нет специального персонала для проведения необходимой работы. Если вам требуется постоянная надежная защита, то используйте в дополнение к тестированию уязвимостей системы обнаружения вторжений.

Правильно размещайте сервер сканирования

Если вы хотите по-настоящему проверить внешнюю (из Интернета) уязвимость вашей информационной системы, следует разместить сервер Nessus вне вашего межсетевого экрана. Это может быть домашнее соединение с Интернетом, центр обработки данных, расположенный вне сети вашей организации или в другой организации (возможно, вы сможете договориться об использовании ресурсов другой организации для сканирования и позволить им использовать ваши для той же цели). Помните, что в силу клиент-серверной архитектуры Nessus, вы сможете по-прежнему управлять сканированием, находясь под защитой межсетевого экрана. Необходимо только включить поддержку SSL для криптографической защиты коммуникаций между клиентом и сервером.

При сканировании внутренней сети сервер необходимо разместить позади межсетевого экрана. Установка Nessus на ПК-блокноте может облегчить выполнение сканирования как изнутри, так и извне вашей сети, не требуя множества машин.

Какие уязвимости тестирование не находит

Хотя тестирование уязвимостей - полезное средство в арсенале обеспечения безопасности, не следует считать его панацеей. Имеются ситуации и области, где программа тестирования уязвимостей не может помочь. Для уменьшения рисков требуются дополнительные системы и процедуры. Ниже перечислены проблемы безопасности, которые не будут обнаружены при тестировании уязвимостей.

Логические ошибки

Логические ошибки - это дыры в безопасности в виде ошибочной программной логики. Обычно это необнаруженные или неисправленные ошибки, когда программа выполняется не так, как задумано. Пример - входная Web-страница, которая проводит аутентификацию не совсем корректно. Еще один пример - ошибка, позволяющая пользователям получить больше привилегий, чем они должны иметь. Общеизвестные логические ошибки наиболее употребительных программ могут быть включены в тесты уязвимостей Nessus, но большинство из них слишком сложно обнаружить, что удается, как правило, только специально занимающимся этим хакерам.

Необнаруженные уязвимости

Тестирование уязвимостей основывается на опубликованных отчетах об уязвимостях. Обычно после сообщения об уязвимости для системы пишется специальный дополнительный модуль. Для программ с открытыми исходными текстами это может потребовать всего лишь нескольких дней. Однако в течение этого времени окно уязвимости остается открытым, так как ваш сканер не может найти дыр в безопасности, хотя они существуют. Конечно, вы можете быстро написать собственный тест, используя NASL, пока не появится официальный выпуск.

Индивидуальные приложения

Средства тестирования уязвимостей обычно ориентированы только на опубликованные программы, коммерческие или с открытыми исходными текстами. Если вы применяете программу, которая была разработана только для внутреннего использования, тестирование уязвимостей, вероятно, ничего не сможет в ней проверить. Если она опирается на стандартные протоколы или подпрограммы, такие как HTTP, FTP, или SQL, то некоторые из этих тестов будут применимы. Существуют дополнительные программы, специально созданные для тестирования кода на его безопасность, которые вы должны применить к этим приложениям. С помощью средства тестирования, такого как Nessus, для внутреннего приложения можно написать специальные проверки.

Безопасность персонала

Все имеющиеся в мире проверки не помогут, если у вас плохая политика безопасности, или таковой нет вообще, или она не доведена до сведения персонала. Как было показано во врезке, хакеры, которым не удалось получить доступ к сети техническими средствами, могут воспользоваться методами морально-психологического воздействия, то есть попытаться договориться с кем-нибудь о предоставлении доступа. Это может оказаться удивительно легко, так как хакеры играют на природе человека, его готовности помочь другим, особенно коллегам. Существует только один способ борьбы с подобными методами, и он не требует никаких технических средств. Хорошая политика безопасности, доведение ее до сведения сотрудников, проведение ее в жизнь снизят вашу уязвимость по отношению к таким атакам.

Атаки прошлые и текущие

Тестирование уязвимостей показывает только потенциальные дыры в безопасности вашей системы; оно не сообщит, использовались ли эти дыры, и не предупредит, если атака происходит в настоящий момент. (Выявление атак, когда они происходят, является целью систем обнаружения вторжений и рассматривается в [лекции 7](#).) Такие программы, как Nessus, по своей природе предназначены исключительно для превентивных целей, и они будут эффективны только в том случае, если вы что-то предпринимаете для исправления обнаруживаемых проблем. Сканеры уязвимостей не исправляют ошибок, хотя Nessus очень полезен в том плане, что предоставляет подробные инструкции о том, как исправить любую обнаруженную проблему. И, как говорил Бен Франклин: "Грамм профилактики стоит килограмма лечения".

Инструменты безопасности с открытым исходным кодом

6. Лекция: Сетевые анализаторы: версия для печати и PDA

Теперь вы в состоянии должным образом обезопасить и укрепить свои системы и проверить сеть на наличие уязвимостей с помощью превентивных средств, помогающих поддерживать нормальное функционирование и безопасность информационных систем. Мы приступаем к рассмотрению некоторых средств, которые позволяют действовать и реагировать, когда, вопреки всем вашим профилактическим мерам, в сети проявляются компьютерные атаки или проблемы безопасности. К этой категории средств относятся сетевые анализаторы, а также системы обнаружения вторжений и беспроводные анализаторы.

Обзор лекции

Изучаемые концепции:

- Основы сетевых анализаторов
- История и функционирование Ethernet
- Как выполнить безопасный и этичный анализ сети
- Примеры конфигураций анализатора
- Приложения сетевых анализаторов

Используемые инструменты:

Tcpdump, WinDump и Ethereal

Грубо говоря, сетевой анализатор (network sniffer) прослушивает или "обнюхивает" ("sniffs") пакеты определенного физического сегмента сети. Это позволяет анализировать трафик на наличие некоторых шаблонов, исправлять определенные проблемы и выявлять подозрительную активность. Сетевая система обнаружения вторжений является ничем иным, как развитым анализатором, который сопоставляет каждый пакет в сети с базой данных известных образцов вредоносного трафика, аналогично тому, как антивирусная программа поступает с файлами в компьютере.

В отличие от средств, описанных ранее, анализаторы действуют на более низком уровне. Если обратиться к эталонной модели ВОС, то анализаторы проверяют два нижних уровня - физический и канальный.

| Номер уровня модели ВОС | Название уровня | Примеры протоколов |
|-------------------------|-----------------------|--|
| Уровень 7 | Прикладной уровень | DNS, FTP, HTTP, SMTP, SNMP, Telnet |
| Уровень 6 | Уровень представления | XDR |
| Уровень 5 | Уровень сеанса | RPC |
| Уровень 4 | Транспортный уровень | NetBIOS, TCP, UDP |
| Уровень 3 | Сетевой уровень | ARP, IP, IPX, OSPF |
| Уровень 2 | Канальный уровень | Arcnet, Ethernet, Token ring |
| Уровень 1 | Физический уровень | Коаксиальный кабель, оптоволокно, витая пара |

Физический уровень - это реальная физическая проводка или иная среда, примененная для создания сети. На канальном уровне происходит первоначальное кодирование данных для передачи через конкретную среду. Сетевые стандарты канального уровня включают беспроводной 802.11,

Arcnet, коаксиальный кабель, Ethernet, Token Ring и многое другое. Анализаторы обычно зависят от типа сети, в которой они работают. Например, для анализа трафика в сети Ethernet вы должны иметь анализатор Ethernet.

Существуют анализаторы коммерческого класса, от таких производителей, как Fluke, Network General и других. Обычно это специальные аппаратные устройства, которые могут стоить десятки тысяч долларов. Хотя эти аппаратные средства способны осуществлять более глубокий анализ, можно создать недорогой сетевой анализатор с помощью программного обеспечения с открытыми исходными текстами и недорогого ПК на Intel-платформе.

В данной лекции рассматриваются несколько сетевых анализаторов Ethernet с открытыми исходными текстами. Я решил остановиться в этой лекции на Ethernet, так как этот протокол наиболее употребителен в локальных сетях. Весьма вероятно, что ваша организация использует сеть Ethernet или она взаимодействует с подобными организациями.

Когда-то сетевой мир был очень фрагментированным в том, что касалось стандартов передачи физического и канального уровней; не существовало единого доминирующего стандарта для ЛВС. Корпорация IBM сделала свою топологию Token Ring стандартом для своих сетей ПК. Многие компании, которые первоначально использовали оборудование IBM, применяли Token Ring, потому что у них не было выбора. Arcnet, в силу своей дешевизны, был популярен в небольших компаниях. Ethernet доминировал в университетских и исследовательских средах. Существовало множество других протоколов, таких как AppleTalk компании Apple для компьютеров Macintosh. Эти протоколы обычно были специфичными для конкретного производителя. Однако с развитием Интернета стандарт Ethernet стал набирать все большую популярность. Поставщики оборудования начали стандартизацию и сосредоточились на дешевых платах Ethernet, концентраторах и коммутаторах. На сегодняшний день Ethernet стал фактическим стандартом для локальных сетей и Интернета. Большинство компаний и организаций выбрали его из-за невысокой стоимости и по соображениям совместимости.

Краткая история Ethernet

Боб Меткалф изобрел Ethernet в 1973 году в исследовательском центре Xerox в Пало Альто. (Из того же "инкубатора изобретений" вышли лазерный принтер и графический пользовательский интерфейс.) Боб и его группа разработали и запатентовали "многоточечную систему связывания данных с обнаружением коллизий", которая позже стала известна как Ethernet. Затем Боб организовал компанию, специализирующуюся на создании оборудования для этого нового протокола. Со временем она превратилась в 3Com - одну из крупнейших сетевых компаний в мире. К счастью, Ethernet был выпущен как общественное достояние, поэтому другие компании могли вносить свой вклад в эти спецификации. Для Token Ring и большинства других сетевых протоколов того времени это было не так. Если бы Ethernet остался в частной собственности или был ограничен оборудованием только одной компании, то, возможно, он не превратился бы в доминирующий сегодня стандарт. Со временем он был принят как официальный стандарт IEEE (International Electrical and Electronic Engineers - Институт инженеров по электротехнике и электронике), что практически гарантировало его широкое признание корпоративными и правительственными пользователями во всем мире. На основе Ethernet были разработаны другие стандарты, такие как Fast Ethernet, Gigabit Ethernet и Wi-Fi.

Ethernet регламентирует как управление физической средой, так и программное кодирование данных, передаваемых по сети. Так как Ethernet имеет широковещательную топологию и все компьютеры потенциально могут "говорить" одновременно, в нем предусмотрен механизм обработки коллизий - когда два компьютера одновременно посылают пакеты данных. Если обнаруживается коллизия, то обе стороны повторно посылают данные после задержки со случайной длительностью. В большинстве случаев это работает вполне успешно. Однако это также является недостатком архитектуры Ethernet. Все компьютеры, подключенные к сети Ethernet, пересылают данные по одному и тому же физическому кабелю, и плата Ethernet видит весь проходящий трафик. Плата Ethernet предназначена для обработки только тех пакетов, которые ей адресованы, но вы легко можете заметить здесь потенциальные проблемы безопасности.

Представьте себе, что почтальоны, вместо того, чтобы раскладывать корреспонденцию по почтовым ящикам адресатов, просто вываливали бы ее посреди улицы, так что каждый житель должен копаться в этой куче в поисках адресованной ему почты, оставляя другую корреспонденцию на месте. (Было бы любопытно взглянуть на подписчиков "Плейбоя" и на получателей повесток о просрочке налоговых платежей.) Подобная вымышленная система не слишком безопасна, да и время получателей расходуется не особенно эффективно, но, в сущности, именно так спроектирован Ethernet.

В наше время для повышения эффективности большинство сетей Ethernet являются коммутируемыми. Это означает, что каждый порт Ethernet видит не весь трафик, а только тот, что предназначен для подключенной к нему машины. Это помогает сгладить некоторые проблемы приватности и перегрузки, но в каждый порт все же приходит много широковещательного трафика. Обычно широковещательный трафик направляется во все порты сети с целью обнаружения или для информации. Это относится к таким протоколам, как DHCP, где машина посылает широковещательное сообщение, разыскивая в

сети какой-либо из серверов DHCP, чтобы получить у него адрес. Машины под управлением Microsoft Windows также известны тем, что генерируют обильный широковещательный трафик.

В локальных сетях Ethernet присутствуют и другие типы широковещательного трафика. Одним из них является протокол разрешения адресов (Address Resolution Protocol - ARP), который используется, когда машина впервые пытается узнать, какой адрес уровня доступа к среде передачи (MAC-адрес) соответствует определенному IP-адресу (см. врезку об IP-адресах в [лекции 3](#)). В сетях Ethernet MAC-адреса являются 12-значными шестнадцатеричными числами и присваиваются платам при производстве. Каждый производитель имеет собственный диапазон чисел, поэтому обычно по MAC-адресу можно узнать, кто сделал плату. Если машина знает IP-адрес, но не адрес Ethernet, она посылает пакеты ARP, спрашивая: "Кому принадлежит этот адрес?" Получив ответ, машина сможет затем послать остальную часть сообщения по правильному MAC-адресу. Именно этот тип трафика оставляет локальные сети Ethernet восприимчивыми для атак анализатора, даже если они используют коммутацию вместо широковещания всего трафика во все порты. Кроме того, если хакерам удастся получить доступ к коммутатору (эти устройства зачастую плохо защищены), они могут превратить свой собственный порт в порт "монитора" или "зеркала", который показывает трафик других портов.

Особенности применения сетевых анализаторов

Чтобы применять сетевые анализаторы этично и продуктивно, необходимо выполнять следующие рекомендации.

Всегда получайте разрешение

Анализ сети, как и многие другие функции безопасности, имеет потенциал для ненадлежащего использования. Перехватывая все данные, передаваемые по сети, вы вполне можете подсмотреть пароли для различных систем, содержимое почтовых сообщений и другие критичные данные, как внутренние, так и внешние, так как большинство систем не шифрует свой трафик в локальной сети. Если подобные данные попадут в нехорошие руки, это, очевидно, может привести к серьезным нарушениям безопасности. Кроме того, это может стать нарушением приватности служащих, если таковая фигурирует в политике вашей организации. Всегда получайте письменное разрешение руководства, желательно высшего, прежде чем начинать подобную деятельность. Следует также предусмотреть, что делать с данными после их получения. Помимо паролей, это могут быть другие критичные данные. Как правило, протоколы сетевого анализа должны вычищаться из системы, если только они не нужны для уголовного или гражданского преследования. Существуют документированные прецеденты, когда благонамеренных системных администраторов увольняли за несанкционированный перехват данных.

Разберитесь в топологии сети

Прежде чем настраивать анализатор, убедитесь, что вы полностью понимаете физическую и логическую организацию вашей сети. Проводя анализ в неправильном месте сети, вы либо получите ошибочные результаты, либо не сможете увидеть то, что ищете. Проверьте, что между анализирующей рабочей станцией и тем, что вы собираетесь наблюдать, нет маршрутизаторов. Маршрутизаторы будут направлять трафик в сегмент сети, только если происходит обращение к расположенному там узлу. Аналогично, если вы находитесь в коммутируемой сети, вам понадобится сконфигурировать порт, к которому вы подключились, как порт "монитора" или "зеркала". Разные производители используют различную терминологию, но по сути вам необходимо, чтобы порт действовал как концентратор, а не как коммутатор, так как он должен видеть весь трафик, идущий через коммутатор, а не только тот, что направлен на вашу рабочую станцию. Без такой настройки порт монитора будет видеть только то, что направлено в порт, к которому вы подключены, и сетевой широковещательный трафик.

Используйте жесткие критерии поиска

В зависимости от того, что вы ищете, использование открытого фильтра (то есть показ всего) сделает вывод данных объемным и трудным для анализа. Используйте специальные критерии поиска, чтобы сократить вывод, который выдает ваш анализатор. Даже если вы не знаете точно, что ищете, можно, тем не менее, написать фильтр для ограничения результатов поиска. Если вы ищете внутреннюю машину, задайте критерии для просмотра только исходных адресов внутри вашей сети. Если вы пытаетесь отследить определенный тип трафика, скажем, трафик FTP, то ограничьте результаты только тем, что приходит в порт, используемый приложением. Поступая таким образом, вы сделаете результаты анализа значительно более полезными.

Установите эталонное состояние сети

Применив сетевой анализатор во время нормальной работы и записав итоговые результаты, вы получите эталонное состояние, которое можно сравнивать с результатами, полученными во время попыток выделения проблемы. Анализатор Ethereal, рассматриваемый в этой лекции, создает для этого несколько

удобных отчетов. Вы получите также некоторые данные для отслеживания использования сети в зависимости от времени. При помощи этих данных можно определить, когда сеть насыщается и каковы основные причины этого - перегруженный сервер, рост числа пользователей, изменение типа трафика и т.п. Если есть точка отсчета, проще понять, кто и в чем виноват.

Tcpdump: Анализатор трафика Ethernet

Tcpdump

Автор/основной контакт: University of California, Lawrence Berkeley Laboratories

Web-сайт: <http://www.tcpdump.org/>

Платформы: Большинство UNIX-платформ

Лицензия: BSD

Рассмотренная версия: 3.8.1

Списки почтовой рассылки:

tcpdump-announce@tcpdump.org

Это список только для объявлений.

tcpdump-workers@tcpdump.org

Этот список для обсуждения программ. По нему также рассылаются объявления, поэтому, если вы в него входите, вам нет нужды подписываться на первый список.

Оба списка архивируются, поэтому вы можете выполнять поиск среди старых сообщений. Список обсуждения программ доступен также в формате еженедельного итогового дайджеста.

Существует много доступных анализаторов - как свободных, так и коммерческих, но Tcpdump доступен наиболее широко и недорого. Он поставляется с большинством дистрибутивов UNIX, включая Linux и BSD. На самом деле, если у вас достаточно свежий дистрибутив Linux, то, весьма вероятно, что вы уже имеете установленный и готовый к употреблению Tcpdump.

Установка Tcpdump

Tcpdump полностью оправдывает свое имя: он выдает содержимое пакетов TCP/IP, проходящих через сетевой интерфейс, на устройство вывода (обычно - на экран или в файл).

1. Чтобы анализатор Tcpdump работал, он должен иметь возможность перевести сетевую плату в так называемый режим прослушивания (или неразборчивый режим - promiscuous mode). Это означает, что сетевая плата будет перехватывать весь трафик Ethernet, а не только тот, что адресован ей. Каждая операционная система по-своему обрабатывает трафик платы Ethernet. Чтобы предоставить общую ссылку для программистов, была создана библиотека pcap. В UNIX она называется libpcap, а в Windows - WinPcap. Эти низкоуровневые драйверы могут изменять способ, которым плата обычно обрабатывает трафик. Они должны быть установлены до Tcpdump.

Если Tcpdump уже присутствует в системе, то установлены и драйверы. В противном случае возьмите их из каталога misc на компакт-диске, приложенном к этой книге, или с Web-сайта Tcpdump. Не забудьте установить их перед установкой Tcpdump.

Примечание: Для libpcap требуются также интерпретируемые языки Flex и Bison или, в качестве замены, - Lex и Yacc. Если у вас их нет, найдите их на дистрибутивном диске своей операционной системы или в Сети, и установите, чтобы установка libpcap прошла успешно.

2. Установите libpcap, распаковав ее и выполнив стандартные команды компиляции:

```
./configure
make
make install
```

Если во время компиляции вы получите предупреждение вида "Cannot determine packet capture interface" ("Невозможно определить интерфейс перехвата пакетов"), то ваша сетевая плата не поддерживает режим прослушивания, и для применения Tcprdump придется взять другую плату. Большинство современных плат должны поддерживать этот режим.

3. После установки libpcap распакуйте пакет Tcprdump и перейдите в его каталог.

4. Выполните те же команды компиляции:

```
./configure
make
make install
```

Tcprdump готов к употреблению.

Запуск Tcprdump

Существует ряд операций для фильтрации вывода, чтобы найти определенный тип трафика или снизить общий объем вывода. На самом деле, в активно используемой сети нефильтрованный вывод Tcprdump будет пролетать на экране быстрее, чем вы сможете его прочитать! Однако прямо сейчас для демонстрации возможностей Tcprdump запустим его из командной строки, набрав просто

```
tcprdump
```

Вы увидите весь нефильтрованный трафик TCP, проходящий через плату Ethernet вашей машины. Он может выглядеть примерно так, как в примере на [листинге 6.1](#).

```
12:25:38.504619 12.129.72.142.http > 192.168.1.3.3568: . ack 1418369642
  win 31856 <nop,nop,timestamp 72821542 25475802> (DF)
12:25:38.504758 192.168.1.3.3568 > 12.129.72.142.htt: . ack 1
  win 40544 <nop,nop,timestamp 25486047 72811295> (DF)
12:25:38.507753 192.168.1.3.4870 > 65.83.241.167.domain: 11414+ PTR? 1
  42.72.129.12.in-addr.arpa. (44) (DF)
12:25:38.561481 65.83.241.167.domain > 192.168.1.3.4870: 11414 NXDomain*- 0/1/0 (113)
12:25:38.562754 192.168.1.3.4870 > 65.83.241.167.domain: 11415+ PTR?
  3.1.168.192.in-addr.arpa. (42) (DF)
12:25:38.609588 65.83.241.167.domain > 192.168.1.3.4870: 11415 NXDomain 0/1/0 (119)
12:25:38.610428 192.168.1.3.4870 > 65.83.241.167.domain: 1416+ PTR?
  167.241.83.65.in-addr.arpa. (44) (DF)
12:25:38.649808 65.83.241.167.domain > 192.168.1.3.4870: 11416 1/0/0 (69)
12:25:43.497909 arp who-has 192.168.1.1 tell 192.168.1.3
12:25:43.498153 arp reply 192.168.1.1 is-at 0:6:25:9f:34:ac
12:25:43.498943 192.168.1.3.4870 > 65.83.241.167.domain: 11417+ PTR?
  1.1.168.192.in-addr.arpa. (42) (DF)
12:25:43.533126 65.83.241.167.domain > 192.168.1.3.4870: 11417 NXDomain 0/1/0 (119)
12:25:44.578546 192.168.1.1.8783 > 192.168.1.255.snmptrap: Trap(35)
  E:3955.2.2.1 192.168.1.1 enterpriseSpecific[specific-trap(1)!=0] 43525500[|snmp]
```


На первый взгляд, выдача кажется запутанной, но если разбить ее на составляющие, то смысл начинает проясняться. Первое число является временной меткой с точностью до долей секунды, так как в активно используемой сети каждую секунду проходит множество пакетов. Следующее число - это IP-адрес отправителя пакета, за которым следует > (знак больше), а затем целевой адрес. Наконец, могут присутствовать некоторые комментарии и другие данные. В примере можно видеть несколько различных видов трафика, включая трафик DNS (domain), ARP и SNMP.

По умолчанию Tcprdump выполняется, пока не будет остановлен нажатием Ctrl+C или другим сигналом прерывания. Когда Tcprdump останавливается, он выдает сводные данные о просмотренном трафике, включая:

- пакеты, полученные фильтром. Это количество пакетов, обработанных фильтром Tcprdump, а не общее число пакетов TCP в сети, если только вы не выполняете Tcprdump без критериев фильтрации;
- пакеты, отброшенные ядром. Число пакетов, которые были отброшены в связи с отсутствием ресурсов в системе. Эта возможность поддерживается не всеми системами. Даже когда она поддерживается, число может быть неточным, если сеть очень загружена или машина анализатора очень медленная.

Заголовки пакетов TCP/IP

В этом разделе описывается содержимое заголовков пакетов TCP/IP, чтобы вы могли разобраться в выводе Tcprdump. Структура пакетов TCP/IP определена в документе RFC 793 для TCP-части и в RFC 791 для IP-части. Полный текст этих спецификаций можно найти по адресу <http://www.rfc-editor.org/>. Рис. 6.1 является графическим представлением заголовков TCP и IP. Заголовки обоих типов имеют длину не менее 20 байт и обычно представляются 32-битными секциями (4 байта) с адресами, опциями и другими настройками сеансов.

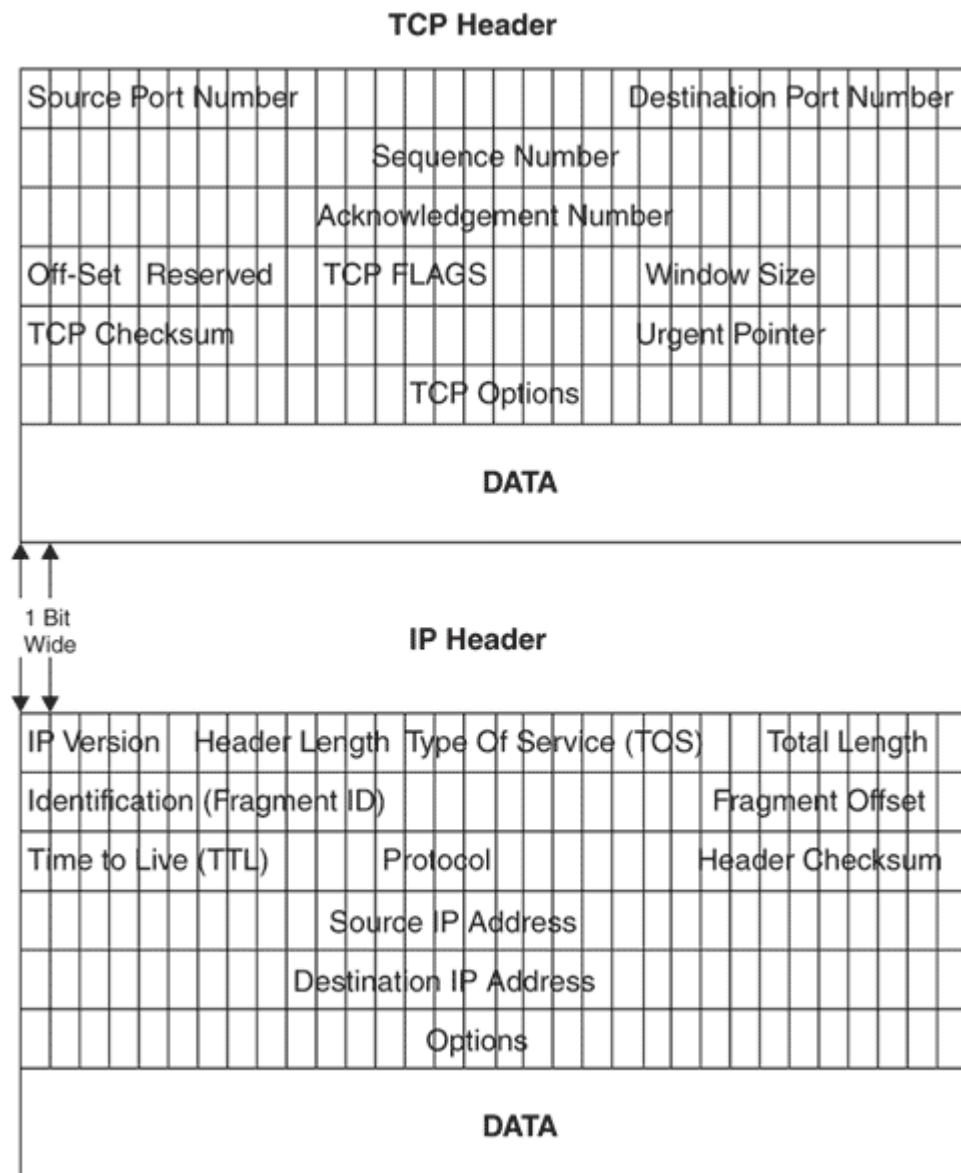


Рис. 6.1. Заголовок TCP/IP

Рассмотрим сначала IP-часть, так как это нижний уровень сетевой модели. Заголовок протокола IP содержит адреса получателя и отправителя пакета. Так как каждый адрес занимает 32 бита (4 октета по 8 бит каждый), то исходный и целевой IP-адреса вместе составляют 8 байт. В первой части заголовка помещаются различные переключатели и опции пакета. Первая строка содержит несколько бит, которые идентифицируют версию IP. Большинство сетей используют IP версии 4 (IPv4), но более новая 128-битная система IP, называемая IP версии 6 (IPv6), существует уже несколько лет и постепенно получает признание. Предполагается, что IPv6 разрешит проблемы адресного пространства IP, выделяя до 128 бит для адресной части, что должно удовлетворить любые мыслимые потребности. IPv6 решает также проблемы безопасности и верификации, имеющиеся в IPv4. Но в настоящее время вы в подавляющем большинстве случаев будете видеть пакеты IPv4. Затем следуют значения длины заголовка и типа сервиса (последний служит для

дифференциации при определении приоритетов пакетов). Заключительный фрагмент этой строки представляет общую длину заголовка, которая обычно одинакова для всех пакетов (20 байт), но может меняться для новых протоколов, таких как IPv6.

В двух следующих строках располагаются идентификатор пакета и контрольная сумма, помогающая проверять его корректность. Наконец, имеются IP-адреса отправителя и получателя и поле опций, которое может иметь переменную длину или дополняться нулями и произвольными данными.

Заголовок TCP отвечает за создание сеанса TCP и функции более высокого уровня. Обычно он имеет длину 20 байт и начинается с номеров исходного и целевого портов по 16 бит каждый. Именно поэтому номера портов не могут быть больше 65535 (2^{16} равно 65536). (Интересно, что выбор всех этих чисел, вроде бы абсолютно произвольный, всегда на чем-то основан.)

Номера портов, как упоминалось ранее, определяют программу, которой необходимо направлять пакеты на удаленной машине, и идентифицируют сеанс на локальной машине. Следующая строка содержит порядковый номер. Он используется для сборки пакетов в правильном порядке на удаленном конце, даже если они приходят в другом порядке. Это один из аспектов отказоустойчивости сеансов TCP. Кроме того, имеется номер подтверждения, также длиной 32 бита, который позволяет проверить, что пакет идет с правильной машины. Следующая строка содержит 4-битную секцию, называемую смещением данных, которая определяет, сколько 32-битных строк или "слов" имеется в заголовке (обычно 4); за ней располагаются 6 бит, зарезервированных для будущих применений. Затем следует 6-битная секция, называемая флагами TCP. Вторая половина этой строки служит для согласования размера окна и говорит получателю, сколько бит готов принять отправитель. Флаги TCP весьма важны, здесь задаются различные управляющие биты, контролирующие обработку пакетов. Каждый тип коммуникаций TCP задается одним битом, единица соответствует включению, ноль - отключению. В [табл. 6.1](#) перечислены шесть полей раздела флагов TCP и описано их применение. Примечание: Каждое "поле" имеет ширину один бит (просто - единица или ноль, включено или выключено).

Таблица 6.1. Поля флагов TCP

| Флаг TCP | Полное имя | Описание |
|----------|---------------------|---|
| URG | Указатель срочности | Показывает приоритет TCP-пакетов |
| ACK | Подтверждение | Помечает этот пакет как подтверждение получения |
| PSH | Выталкивание | Выталкивает поставленные в очередь данные из буферов |
| RST | Сброс | Сбрасывает соединение TCP по завершении или после разрыва |
| SYN | Синхронизация | Синхронизирует соединение |
| FIN | Завершение | Завершает передачу |

Обычно только один или два из этих флагов установлены (биты заданы как единица), но, как мы видели в [лекции 4](#), ничто не мешает отправить пакет со всеми битами взведенными (сканирование XMAS) или выключенными (сканирование NULL), чтобы попытаться запутать удаленную систему.

Следом располагается контрольная сумма TCP и указатель срочности. Затем идет строка со всеми TCP-опциями пакета, такими как дополнительные контрольные суммы, метки времени и т.д. Эта строка дополняется до 32 бит нулями, если опции не заполняют ее целиком. Наконец, следует полезная нагрузка - данные пакета. Может показаться, что на отправку одного пакета уходит слишком много накладных административных расходов (примерно 48 байт для каждого пакета), но это на самом деле обеспечивает относительно устойчивое соединение в сети, которая не всегда обладает сквозной надежностью (как Интернет). В действительности, чтобы избежать дополнительных расходов TCP, некоторые протоколы, не требующие соединения, используют UDP - протокол без установления соединений с меньшими накладными расходами.

В стандартном сеансе Tcprdump с обычным уровнем подробности вывода вы увидите метку времени, за которой следует порядковый номер TCP. Затем выдаются IP-части, включая исходный и целевой адреса, разделенные знаком > (больше), означающим, что пакет идет отсюда туда. В конце располагается информационное поле, которое показывает, что делает пакет. Можно использовать опции -v или -vv, чтобы получить от Tcprdump более подробные данные о заголовке (см. следующий раздел).

Обычно вы будете запускать Tcprdump с некоторыми установленными опциями или фильтрами, чтобы уменьшить и сфокусировать вывод. Общий вид инструкции запуска Tcprdump таков:

Замените опции и выражения одной или несколькими допустимыми переменными. Опции Tcprdump перечислены в [табл. 6.2](#).

Таблица 6.2. Опции Tcprdump

| Опция | Описание |
|--------------------|--|
| -a | Пытается преобразовать адреса в имена. Это создает дополнительную нагрузку на систему и может привести к потере пакетов |
| -с число | Останавливает Tcprdump после обработки заданного числа пакетов |
| -C размер_файла | Ограничивает размер выходных файлов заданным числом байт |
| -d | Выдает процедуру сопоставления пакетов с образцом в удобочитаемом виде и затем останавливается |
| -dd | Выдает процедуру сопоставления пакетов с образцом в виде фрагмента программы на языке Си |
| -ddd | Выдает процедуру сопоставления пакетов с образцом в виде десятичных чисел |
| -e | В каждой строке выдачи печатает заголовок канального уровня (в сетях Ethernet это MAC-адрес) |
| -E алгоритм:секрет | Использует встроенную в Tcprdump возможность расшифровывать на лету пакеты, зашифрованные по протоколу IPsec ESP. Разумеется, чтобы использовать эту опцию, вы должны располагать разделяемым секретным ключом. В число возможных значений параметра "алгоритм" входят des-cbc, 3des-cbc, blowfish-cbc, g3c-cbc, приведенный 128-cbc. Кроме того, оно может быть пустым. По умолчанию используется des-cbc. Значением параметра "секрет" должен служить секретный ключ ESP в текстовом виде. Дополнительную информацию об IPsec можно найти в лекции 9 |
| -F файл | Использует файл (а не сеть) для ввода данных. Это удобно для анализа событий "постфактум". |
| -i интерфейс | Читает из заданного интерфейса, когда на анализирующей машине имеется несколько сетевых интерфейсов. По умолчанию Tcprdump использует действующий интерфейс с наименьшим номером. В системах Linux можно использовать также параметр any для перехвата пакетов из всех сетевых интерфейсов |
| -n | Не преобразовывает адреса в имена |
| -N | Не печатает в именах хостов имя домена вышележащего уровня. Это полезно, если вам необходимо представить обезличенную версию вывода и вы не хотите раскрывать, чья это сеть |
| -p | Не переводит интерфейс в режим прослушивания. Используется только при исследовании трафика, направленного в анализирующий компьютер |
| -q | Печатает быстрый вывод. Печатается меньше протокольной информации, поэтому строки оказываются короче |
| -T тип | Заставляет интерпретировать пакеты, выбранные заданным в выражении фильтром, в соответствии с указанным типом |
| -t | Не печатает метку времени в каждой строке |
| -tt | Печатает неформатированную метку времени в каждой строке |
| -ttt | Печатает интервал времени между пакетами |
| -tttt | Печатает в каждой строке дату, а затем метку времени в подразумеваемом формате |
| -v | Использует чуть более подробный вывод, включающий время жизни, идентификатор, общую длину и поля опций каждого пакета |
| -vv | Предоставляет более детальный вывод. Пакеты NFS и SMB полностью декодируются |
| -vvv | Предоставляет еще более подробный вывод. Это может существенно замедлить работу анализатора |
| -w имя_файла | Записывает пакеты в указанный файл вместо вывода их на экран. Таким образом результаты "вынюхивания" без участия человека можно сохранить и проанализировать их позже. Например, если в вашей сети происходят какие-то странные вещи, вы можете запустить Tcprdump на ночь, чтобы перехватить весь необычный трафик. Не забудьте написать хороший фильтр, иначе полученный наутро файл может оказаться слишком большим |
| -x | Выводит каждый пакет (без заголовка канального уровня) в шестнадцатеричном виде. |

| | |
|----|---|
| -X | Выводит содержимое пакетов и в шестнадцатеричном, и в текстовом видах |
|----|---|

Выражения Tcprdump

Выражения Tcprdump определяют выбор отображаемых сетевых пакетов. Именно здесь происходит реальная работа Tcprdump. Выдаются только те объекты, которые соответствуют выражению; если выражения не заданы, отображаться будут все пакеты. Выражение Tcprdump состоит из одной или нескольких директив, называемых примитивами, которые, в свою очередь, состоят из идентификатора и следующего за ним квалификатора. В [табл. 6.3](#) перечислены три различных вида квалификаторов, а в [табл. 6.4](#) - доступные комбинации примитивов.

Существуют также более сложные выражения, которые можно строить с помощью булевых операций, таких как И, ИЛИ, НЕ, и операций сравнения (больше, меньше и т.п.). Обратитесь к оперативной справке Tcprdump, чтобы детальнее ознакомиться с примерами и методами применения выражений.

Таблица 6.3. Квалификаторы Tcprdump

| Квалификатор | Описание |
|--------------|---|
| тип | Определяет, к чему относится идентификатор, заданный как имя или номер. Возможными типами служат host, net и port. Например, host foo, net 128.3 или port 20 |
| направление | Определяет направление трафика от определенного идентификатора. Возможными направлениями служат src; dst; src or dst и src and dst (src обозначает исходный адрес, dst - целевой) |
| протокол | Позволяет определить протокол для фильтрации. Возможными протоколами являются ether, fddi, tr, ip, ipv6, arp, rarp, decnet, tcp и udp. Если протокол не задан, то допустимы все протоколы, совместимые с остальной частью выражения. При помощи фильтров с этим квалификатором можно определить, какая машина делает чрезмерное количество arp-запросов, или для отбрасывания на фильтре udp-запросов, которых немало во многих сетях, так как DNS использует udp |

Таблица 6.4. Допустимые комбинации примитивов

| Комбинация | Описание |
|--------------------------|--|
| dst host хост | Показывает только трафик, адресованный хосту, который может быть задан IP-адресом или именем |
| src host хост | Показывает только трафик, исходящий из хоста |
| host хост | Показывает как исходящий, так и входящий трафик хоста |
| ether dst Ethernet-хост | Показывает трафик, предназначенный для указанного Ethernet-хоста, который может быть задан либо именем, либо MAC-адресом |
| ether src Ethernet-хост | Показывает трафик, исходящий из Ethernet-хоста |
| ether host Ethernet-хост | Показывает как исходящий, так и входящий трафик Ethernet-хоста |
| gateway хост | Показывает любой трафик, использующий хост в качестве шлюза. Иными словами, трафик, переправляемый с хоста. Так происходит, когда IP-адрес отправителя или получателя не соответствует Ethernet-адресу хоста. Данную возможность целесообразно использовать, когда необходимо отследить весь трафик, проходящий через Интернет-шлюз или некоторый конкретный маршрутизатор |
| dst net сеть | Фильтрует трафик, предназначенный для конкретной сети, заданной в нотации 0.0.0.0. Аналогично ether dst Ethernet-хост за исключением того, что это может быть значительно больше, чем один хост |
| src net сеть | Фильтрует сеть отправителя |
| net сеть | То же, что и две предыдущие инструкции, но трафик разрешен как в заданную сеть, так и из нее |

| | |
|--------------------------|---|
| net сеть mask маска_сети | Сопоставляется с трафиком в заданную сеть или из нее, с указанной маской сети. Применяется для задания точного размера сети с шагом меньше, чем класс C. В этой комбинации допускается использование примитивов src и dst для указания направления потоков данных |
| net сеть/длина_маски | Сопоставляется с трафиком с сетевыми адресами из указанной сети и заданным числом бит в маске сети. Аналогична предыдущей комбинации |
| dst port порт | Фильтрует трафик TCP и UDP с заданным целевым портом. Здесь можно также специфицировать тип перехватываемого трафика, TCP или UDP. По умолчанию отображается трафик обоих типов |
| src port порт | То же, что и предыдущая комбинация, только перехватывается трафик с заданным исходным портом |
| less длина | Отображает пакеты с длиной, меньшей или равной заданной. Допустима также комбинация len <= длина |
| greater длина | То же, что и предыдущая комбинация, только перехватывается трафик с длиной пакетов больше или равной указанной |
| ip proto протокол | Перехватывает трафик заданного протокола. Допустимыми протоколами служат icmp, icmpv6, igmp, igmp, pim, ah, esp, vrrp, udp и tcp. Имена tcp, udp и icmp должны помещаться между двумя обратными косыми чертами, чтобы они не читались как ключевые слова. Пример: ip proto \tcp\ |
| ip6 proto протокол | Аналогично предыдущей комбинация, но для пакетов и типов IPv6 |
| ip6 protochain протокол | Ищет пакеты IPv6, имеющие заголовок указанного протокола |
| ip protochain протокол | То же, что и выше, но для пакетов IPv4 |
| ip broadcast | Идентифицирует только широковещательный трафик, то есть трафик, имеющий все нули или все единицы в поле целевого адреса |
| ether multicast | Регистрирует вещательные пакеты Ethernet |
| ip multicast | Регистрирует вещательные пакеты IP |
| ip6 multicast | Регистрирует вещательные пакеты IPv6 |
| ether proto протокол | Отображает трафик, который имеет указанный тип протокола Ethernet. Допустимыми именами протоколов служат ip, ipv6, arp, rarp, atalk, aarp, decnet, sca, lat, moprcl, moprsc, iso, stp, ipx и netbeui. Эти имена являются также идентификаторами, поэтому они должны быть экранированы с помощью обратных косых черт |
| decnet src хост | Перехватывает трафик DECnet с исходным адресом хоста |
| decnet dst хост | Аналогична предыдущей комбинация, но фильтрует целевой адрес хоста |
| decnet хост | Фильтрует трафик DECnet с исходным или целевым адресом хоста |
| ip | Сокращенный вариант описанной выше комбинации ether proto ip. Ловит трафик, соответствующий Ethernet-протоколу ip |
| ip6 | Сокращенный вариант описанной выше комбинации ether proto ip6. Ловит трафик, соответствующий Ethernet-протоколу ip6 |
| arp | Сокращенный вариант описанной выше комбинации ether proto arp. Ловит трафик, соответствующий Ethernet-протоколу arp |
| rarp | Сокращенный вариант описанной выше комбинации ether proto rarp. Ловит трафик, соответствующий Ethernet-протоколу rarp |
| atalk | Сокращенный вариант описанной выше комбинации ether proto atalk. Ловит трафик, соответствующий Ethernet-протоколу atalk |
| aarp | Сокращенный вариант описанной выше комбинации ether proto aarp. Ловит трафик, соответствующий Ethernet-протоколу aarp |
| decnet | Сокращенный вариант описанной выше комбинации ether proto decnet. Ловит трафик, соответствующий Ethernet-протоколу decnet |
| iso | Сокращенный вариант описанной выше комбинации ether proto iso. Ловит трафик, соответствующий Ethernet-протоколу iso |
| stp | Сокращенный вариант описанной выше комбинации ether proto stp. Ловит трафик, соответствующий Ethernet-протоколу stp |
| ipx | Сокращенный вариант описанной выше комбинации ether proto ipx. Ловит трафик, соответствующий Ethernet-протоколу ipx |
| netbeui | Сокращенный вариант описанной выше комбинации ether proto netbeui. Ловит трафик, соответствующий Ethernet-протоколу netbeui |

| | |
|-------------------------|---|
| vlan идентификатор_ВЛВС | Перехватывает пакеты на основе стандарта 802.1Q VLAN. Идентификатор виртуальной локальной сети можно опускать |
| tcp | Сокращенная форма комбинации ip proto tcp |
| udp | Сокращенная форма комбинации ip proto udp |
| icmp | Сокращенная форма комбинации ip proto icmp |
| iso proto протокол | Перехватывает пакеты BOC с заданным типом протокола - clnp, esis или isis |
| clnp | Сокращенная форма описанной выше комбинации с clnp в качестве протокола |
| esis | Сокращенная форма комбинации iso proto протокол с esis в качестве протокола |
| isis | Сокращенная форма комбинации iso proto протокол с isis в качестве протокола |

Примеры применения Tcprdump

Ниже представлены несколько практических примеров применения Tcprdump

Просмотр всего входящего и исходящего трафика определенного хоста

Если вы хотите отслеживать только входящий и исходящий трафик определенного хоста, то можно отфильтровать все остальное с помощью простого выражения "host". Например, чтобы следить за хостом с IP-адресом 192.168.1.1, нужно выполнить инструкцию

```
tcpdump -n host 192.168.1.1
```

Наблюдение за входящим и исходящим трафиком определенного порта

Если вы хотите проследить за использованием определенного приложения, можно применить Tcprdump для улавливания всего трафика, направляемого в определенный порт TCP/UDP. Если приложением, за которым вы пытаетесь наблюдать, является Telnet (порт 23), то это можно сделать с помощью следующего выражения Tcprdump:

```
tcpdump -n port 23
```

Просмотр всего входящего и исходящего трафика определенного хоста, за исключением некоторых видов трафика

Предположим, что вы хотите следить за одним хостом, как в первом примере, но желаете отфильтровать трафик SSH (если вы подключаетесь к этому хосту посредством SSH, то нефильтранный вывод Tcprdump будет отображать трафик вашего собственного соединения). Это можно сделать, добавив выражение port с булевой операцией НЕ. Вот как выглядит команда:

```
tcpdump -n host 192.163.1.1 and not port 22
```

Выявление вредоносной рабочей станции

Если возникли сетевые проблемы, и вы подозреваете, что вредоносный компьютер норовит затопить вашу сеть, можно применить Tcprdump для быстрого прослеживания виновника. Вне зависимости от того, будет ли это неисправная сетевая плата или ПК с "троянской" программой, вызывающей атаку на доступность, Tcprdump поможет пролить свет на проблему. Сначала попробуйте просто запустить Tcprdump без фильтрации и посмотреть, что порождает большую часть трафика. Используйте опции -a и -e для генерации имен и MAC-адресов.

```
tcpdump -ae
```

Отметим, что можно объединять две буквы с одним дефисом. Если вывод на экране проскальзывает слишком быстро, используйте опцию -с 1000, чтобы остановиться после получения 1000 пакетов.

Слежение за определенной рабочей станцией

С помощью Tcprdump вы легко можете запротоколировать трафик, исходящий из определенной рабочей станции, для последующего анализа (убедитесь только, что вы имеете на это законное право). Используйте инструкцию Tcprdump из первого примера с ключом -w для записи в файл. Если в сети применяется динамическое конфигурирование хостов по протоколу DHCP, то предпочтительным может оказаться использование имен SMB (Windows). Пример:

```
tcpdump -w logfile host 192.168.1.1
```

где logfile представляет файл протокола. Можно также добавить опции -с или -C для ограничения размера файла вывода.

Поиск подозрительного сетевого трафика

Если у вас вызывает беспокойство сетевая активность в нерабочее время, то можно оставить запущенный Tcprdump, отметив трафик, который вы считаете сомнительным. Можно запустить Tcprdump с установленным флагом gateway 192.168.0.1, заменяя IP-адрес на адрес своего Интернет-шлюза. Если ваша домашняя сеть использует IP-диапазон от 192.168.0.0 до 192.168.0.254, в этом случае будет помечаться весь трафик, проходящий через шлюз Интернета. Если имеется внутренний почтовый сервер, и вы не хотите протоколировать этот трафик, так как он допустим, можно добавить инструкцию

```
and host != 192.168.0.2
```

где IP-адрес является адресом почтового сервера. Восклицательный знак действует как булева операция НЕ. Будет помечаться весь входящий трафик, не предназначенный для почтового сервера. Выражение может выглядеть следующим образом:

```
tcpdump -w logfile gateway 192.168.0.1 and  
host != 192.168.1.2
```

Для выявления пользователей, применяющих определенное приложение, например, программы потокового видео или аудио, можно уточнить выражение, если известен номер порта. Если вы знаете, что используется порт TCP 1000, то можно применить примитив port для перехвата трафика подозрительного приложения. Пример:

```
tcpdump -w logfile gateway 192.168.0.1 and  
host != 192.168.1.2  
dst port 1000
```

Для более сложных сценариев обнаружения вторжений лучше применить одну из систем обнаружения вторжений, описанных в [лекции 7](#), но для быстрого предварительного анализа Tcprdump может быть очень полезным средством.

WinDump: Анализатор Ethernet-трафика для Windows

WinDump

Автор/основной контакт: Loris Degioanni

Web-сайт: windump.polito.it/install/default.htm

Платформы: Windows 95, 98, ME, NT4, 2000, XP

Лицензия: BSD

Рассмотренная версия: 3.8 alpha

Список почтовой рассылки WinPcap:

<http://www.mail-archive.com/winpcap-users@winpcap.polito.it/>

Наконец появилась программа Tcprdump для Windows. На самом деле, это настоящая UNIX-программа Tcprdump, перенесенная на платформу Windows, поэтому все функции и выражения работают точно так же.

Установка WinDump

Лорис Дижон был настолько любезен, что не только перенес Tcprdump на платформу Windows, но и сделал установку WinDump еще более простой, чем у UNIX-аналога.

1. Как и для Tcprdump в UNIX, сначала, до того как можно будет запускать WinDump, необходимо установить библиотеки перехвата пакетов. Специальная версия библиотек для Windows называется WinPcap. Она присутствует на компакт-диске в каталоге Misc. Самая свежая версия доступна также на Web-сайте программы.
2. Установите библиотеки WinPcap, щелкнув мышью на этом файле.
3. Загрузите исполнимый файл WinDump и поместите его в каталог, откуда он должен запускаться.

Никаких дополнительных действий по установке не требуется.

Применение WinDump

WinDump применяется точно так же, как и Tcprdump - из командной строки. Просто перейдите в командный режим в Windows и выполните команду в каталоге, в котором находится исполнимый файл WinDump. Все команды и выражения работают так же, но в [табл. 6.5](#) представлены несколько команд, специфических для Windows-версии.

На Web-сайте также доступны исходные тексты для тех, кто хочет внести свой вклад или сделать собственные усовершенствования. Однако, одно предостережение: данный вид программирования для Windows - удел крутых парней, хорошо разбирающихся в сетевых протоколах.

Это все, что нужно для работы в Windows или UNIX. Если вы хотите иметь что-то большее, чем просто интерфейс командной строки, воспользуйтесь описанным ниже средством, предлагающим для выноживания графический интерфейс.

Таблица 6.5. Дополнительные команды WinDump

| Команда | Описание |
|---------|---|
| -B | Устанавливает размер буфера драйвера в килобайтах для сеанса перехвата. Если пакеты теряются слишком часто, попробуйте немного увеличить это значение. По умолчанию используется 1 МБ (-B 1000) |
| -D | Печатает список доступных сетевых интерфейсов в вашей системе. Выводится имя интерфейса, его номер и описание, если таковое имеется. Эти параметры можно использовать для задания интерфейса перехвата с помощью ключа <code>Tcprdump -i</code> |

Ethereal: Анализатор сетевых протоколов для UNIX и Windows

Ethereal

Автор/основной контакт: Gerald Combs

Web-сайт: <http://www.ethereal.com/>

Платформы: Большинство UNIX, Windows 95, 98, ME, NT4, 2000, XP

Лицензия: GPL

Рассмотренная версия: 0.10.2

Списки почтовой рассылки:

Ethereal-announce

Общий список объявлений. Не принимает сообщения.

Подписка по адресу <http://www.ethereal.com/mailman/listinfo/ethereal-announce>.

Ethereal-users

Общие вопросы использования Ethereal. Отправляйте свои вопросы сюда.

Подписка по адресу <http://www.ethereal.com/mailman/listinfo/ethereal-users>.

Ethereal-dev

Дискуссии разработчиков.

Подписка по адресу <http://www.ethereal.com/mailman/listinfo/ethereal-dev>.

Ethereal-doc

Для тех, кто пишет документацию Ethereal или хочет в этом участвовать.

Подписка по адресу <http://www.ethereal.com/mailman/listinfo/ethereal-doc>.

Ethereal-cvs

Для отслеживания изменений в CVS-дереве Ethereal, в котором поддерживается самая свежая версия кода для разработчиков. Сообщения не принимаются, любые вопросы должны направляться в Ethereal-users или dev в зависимости от их содержания.

Подписка по адресу <http://www.ethereal.com/mailman/listinfo/ethereal-cvs>.

Ethereal предлагает все выгоды средства командной строки, такого как Tcpdump, а также ряд дополнительных преимуществ. Он обладает удобным пользовательским графическим интерфейсом, поэтому не нужно изучать все опции командной строки. Кроме того, он предлагает значительно больше аналитических и статистических возможностей. К числу прочих достоинств Ethereal относятся:

- Более ясный формат вывода. По сравнению с необработанными наборами перехваченных пакетов в Tcpdump, выдачу Ethereal значительно легче читать и понимать.
- Поддержка значительно большего числа форматов протоколов. Ethereal может интерпретировать более 300 различных сетевых протоколов, что охватывает почти все когда-либо изобретенные виды сетей.
- Поддержка большего числа физических форматов сетей. Сюда входят новые протоколы, такие как IP поверх ATM и FDDI.
- Возможность интерактивно просматривать и сортировать перехваченные сетевые данные.
- Возможность сохранения выдачи в виде обычного текста или в формате PostScript.
- Наличие режима фильтрации вывода с широкими возможностями, включая выделение цветом некоторых пакетов. Имеется графический интерфейс создания фильтра, облегчающий данный процесс.

- Возможность следить за потоком TCP и просматривать его содержимое в текстовом виде. Это может быть очень полезно, когда требуется читать межсерверные сообщения, чтобы отслеживать проблемы электронной почты или Web. Данная возможность позволяет оперативно следить за общением между взаимодействующими узлами.
- Возможность работать с рядом программ и библиотек перехвата. Ethereal, помимо libpcap, работает также со специализированным оборудованием. В число поддерживаемых программ входят Sniffer и Sniffer Pro от Network Associates; LANalyser от Novell; некоторые устройства от Cisco, Lucent и Toshiba; некоторые беспроводные устройства анализа, такие как NetStumbler и Kismet Wireless. Ethereal работает как встраиваемый модуль для многих из этих программ и устройств.
- Возможность сохранять сеансы в нескольких форматах. Это полезно, если вы хотите проводить дополнительный анализ с помощью других средств, таких как libcap (по умолчанию), Sun Snooper, Microsoft Network Monitor и Sniffer от Network Associates.
- Поддержка терминального режима командной строки, предназначенного для тех, кому не по душе графический интерфейс, хотя подавляющая часть полезных свойств Ethereal проистекает из его графического инструментария.

Ethereal настолько полезен в качестве сетевого средства, что он был оценен на web-сайте Insecure.org, посвященном безопасности, как занимающий вторую позицию по популярности среди доступных средств сетевой безопасности. Помимо собственно безопасности, Ethereal допускает множество применений; на самом деле его можно использовать и как универсальное средство анализа сети.

Установка Ethereal для Linux

1. Прежде чем загружать Ethereal, необходимо располагать библиотеками libpcap и библиотеками разработки GTK. Если вы загружали описанные в предыдущих лекциях сканеры портов или уязвимостей, то все библиотеки должны быть уже установлены. В противном случае следует загрузить библиотеки GTK или установить их с дистрибутивных компакт-дисков вашей ОС. libpcap можно взять с прилагаемого к книге компакт-диска или на сайте <http://www.tcpdump.org/>, GTK - на <http://www.gtk.org/>.
2. Затем необходимо решить, использовать ли RPM или компилировать исходные тексты. Существует множество пакетов RPM для различных версий Linux. Если такой пакет есть для вашего дистрибутива, то можно использовать его и пропустить процесс компиляции. В противном случае придется выполнить компиляцию.
3. Чтобы скомпилировать Ethereal, сначала загрузите и распакуйте самый свежий дистрибутив. Стандартная установка должна годиться для большинства применений. Просмотрите файл INSTALL, если захотите задать дополнительные параметры времени компиляции.
4. Перейдите в каталог установки и, как обычно, наберите

```
./configure
make
make install
```

Теперь можно запустить Ethereal, набрав `./ethereal` в командной строке или щелкнув мышью на исполнимом файле в X-Window. Чтобы запускать Ethereal в среде X-Window, необходимо быть пользователем root. Для запуска Ethereal в режиме командной строки можно набрать `./tethereal`.

Установка Ethereal для Windows

1. Прежде чем запускать Ethereal, необходимо установить библиотеки WinPcap. Если вы уже установили в своей Windows-системе описанные в предыдущих лекциях сканеры портов или уязвимостей, то библиотеки у вас уже есть и можно переходить к шагу 2. Убедитесь, что ваша версия WinPcap не ниже, чем 2.3. Если вы работаете на многопроцессорной машине или на машине с новыми процессорами Pentium с технологией многопоточности, то необходимо использовать WinPcap 3.0 или выше, но результаты все равно могут быть непредсказуемы, так как Ethereal не очень хорошо работает с несколькими процессорами.
2. Средства GTK для графического интерфейса включены в пакет установки Ethereal. Загрузите с web-сайта Ethereal самораспаковывающийся файл установки. (Я рекомендую устанавливать бинарный файл, а не заниматься компиляцией на Windows-машине. Это существенно проще и не требует компилятора под Windows.)
3. После загрузки файла сделайте на нем двойной щелчок мышью. Программа проведет вас через процесс установки. Когда это будет сделано, на рабочем столе появится иконка, после чего можно начинать работу с Ethereal.

Применение Ethereal

Независимо от применяемой версии, Windows или Linux, почти все операции и интерфейсы схожи. После запуска Ethereal вы увидите экран с тремя разделами. В этих окнах отображаются перехваченные данные и другая информация о сеансе. На [рис. 6.2](#) показан пример основного окна с активным сеансом.

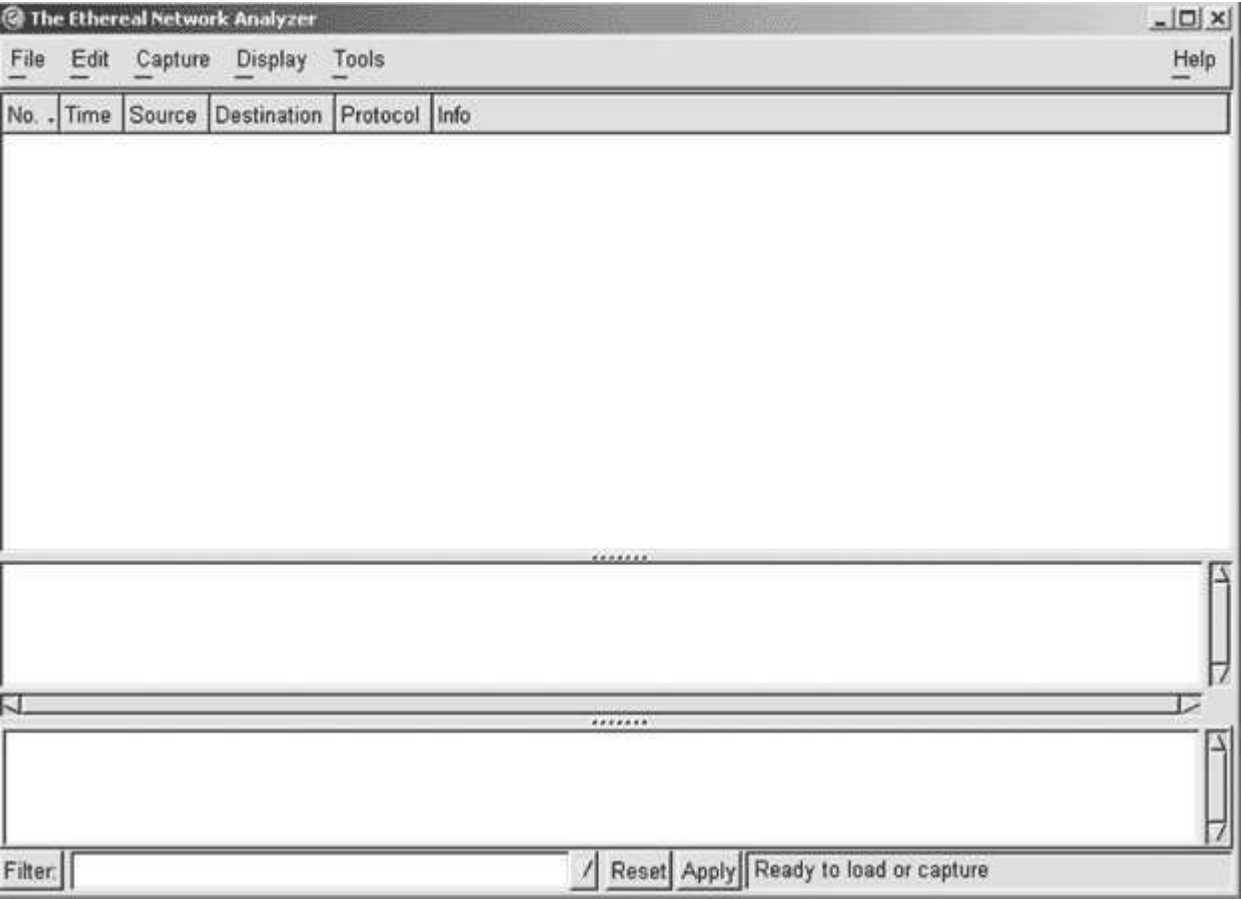


Рис. 6.2. Основной экран Ethereal

В верхней трети экрана выводится поток пакетов в порядке получения, хотя результаты можно отсортировать почти любым образом, щелкая мышью на заголовках колонок. В [табл. 6.6](#) перечислены выводимые данные для каждого пакета или кадра.

В следующем разделе экрана более детально отображается каждый выделенный пакет. Вывод организован таким образом, чтобы в целом соответствовать модели ВОС, поэтому сначала идут детали канального уровня и т.д. Небольшие символы плюс можно раскрыть, чтобы отобразить еще больше информации на каждом уровне. Удивительно, как много подробностей о каждом пакете можно увидеть. Ethereal действует как электронный микроскоп для сетевых пакетов!

В последнем разделе показано реальное содержимое пакета в шестнадцатеричном и, где возможно, в текстовом виде. Бинарные файлы и зашифрованный трафик по-прежнему будут выглядеть как мусор, но весь открытый текст станет виден. В этом проявляется мощь анализатора (и опасность его присутствия в сети).

Таблица 6.6. Данные потока пакетов

| Данные | Описание |
|--------|----------|
|--------|----------|

| | |
|----------------|---|
| Номер пакета | Присваивается Ethernet |
| Время | Время получения пакета. По умолчанию, оно устанавливается как время, прошедшее с начала сеанса перехвата, но можно сконфигурировать вывод астрономического времени, даты и времени или даже интервалов между пакетами (это полезно для анализа функционирования сети) |
| Исходный адрес | Адрес, откуда пришел пакет. В IP-сетях это IP-адрес |
| Целевой адрес | Адрес, куда направляется пакет, также обычно IP-адрес |
| Протокол | Протокол четвертого уровня, используемый пакетом |
| Информация | Некоторая сводная информация о пакете, обычно поле типа |

Запуск сеанса перехвата

Имеется множество допустимых опций и фильтров. Начните с максимально открытого сеанса перехвата. Выберите Start в меню Capture. Появится окно Capture Options ([рис. 6.3](#))

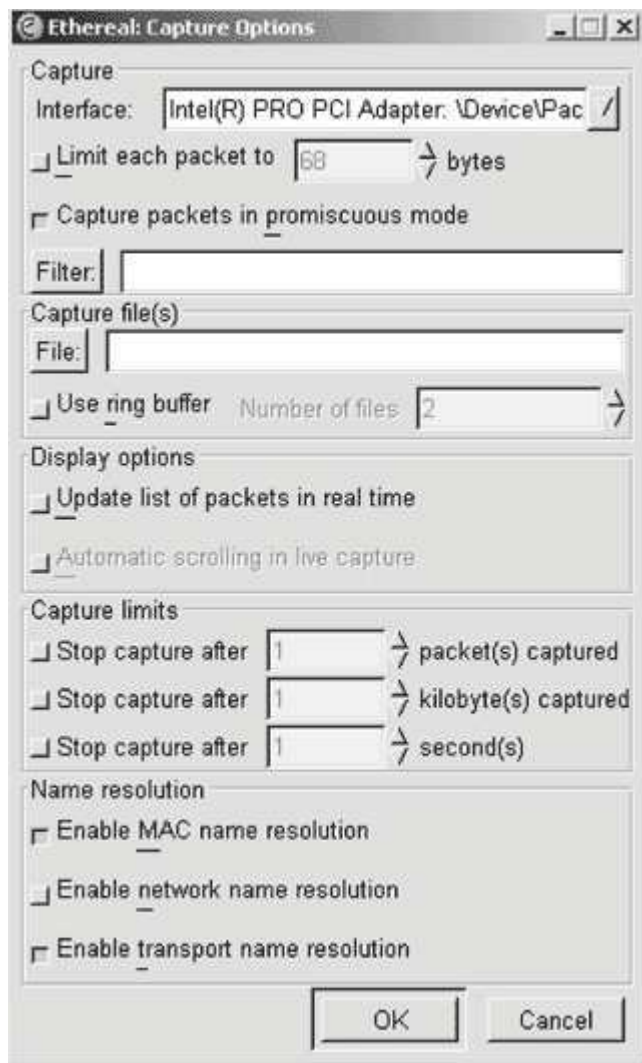


Рис. 6.3. Опции перехвата Ethereal

В [табл. 6.7](#). описаны опции, которые можно задавать перед началом сеанса.

Таблица 6.7. Опции перехвата Ethereal

| Опция | Описание |
|---|---|
| Interface (Интерфейс) | Выбирает интерфейс для перехвата из выпадающего меню. Ethereal автоматически определяет все доступные интерфейсы и выдает их. Можно также задать одновременный перехват на всех интерфейсах, совсем как в Tcpdump |
| Limit each packet to x bytes (Ограничить каждый пакет x байтами) | Задаёт максимальный размер перехватываемых пакетов. Это полезно, если есть вероятность, что некоторые из пакетов могут быть очень большими, а вы не хотите чрезмерно нагружать свою машину |
| Capture packets in promiscuous mode (Перехват пакетов в режиме прослушивания) | Подразумеваемая опция. Выключите ее, если хотите перехватывать только потоки данных, направленные в вашу машину-анализатор |

| | |
|-------------------------------------|---|
| Filter (Фильтр) | Щелкните мышью на кнопке Filter, чтобы создать фильтр, используя выражения в стиле Tcpdump. Вам будет предложено задать имя фильтра (которое можно будет использовать в будущих сеансах) и ввести выражение |
| Capture file(s) (Файлы перехвата) | Щелкните мышью на кнопке File, если хотите читать данные из файла, а не перехватывать их "вживую" |
| Display options (Опции отображения) | По умолчанию отключены, но их можно включить, если вы хотите наблюдать движение пакетов в реальном масштабе времени. Если перехват происходит в загруженной сети или ваша машина слишком медленная, то поступать так не рекомендуется, поскольку это может затопить сеанс и вызвать потерю пакетов. Однако отображение весьма полезно, если вы хотите понаблюдать за трафиком, чтобы получить общее представление о природе потоков данных в сети |
| Capture limits (Пределы перехвата) | Здесь представлено несколько дополнительных опций для задания условий завершения перехвата. Помимо остановки вручную, можно заставить Ethereal остановиться после перехвата некоторого числа x пакетов или килобайт данных или после того, как пройдет определенное число секунд |
| Name resolution (Разрешение имен) | Можно указать, должен или нет Ethereal разрешать имена на различных уровнях сетевой модели. Можно выборочно разрешать имена MAC-адресов, сетевые имена (SMB или имена хостов), и/или имена транспортного уровня. Включение всех этих опций, особенно DNS, может существенно замедлить перехват |

После установки опций щелкните мышью на ОК, и сеанс начнется. Появится окно, в котором в реальном масштабе времени отображается статистика сеанса ([рис. 6.4](#)). Если сеанс настроен для показа пакетов в реальном времени, вы будете наблюдать их в окне, по мере того как они проходят по среде передачи ([рис. 6.2](#))

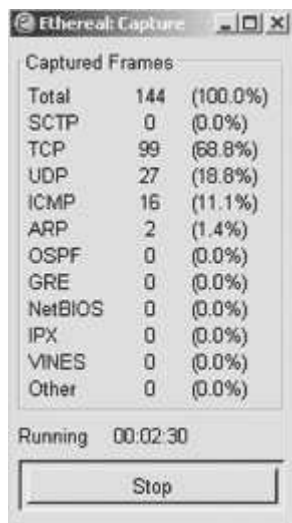


Рис. 6.4. Окно статистики сеанса Ethereal

Сеанс можно остановить в любое время, щелкнув мышью на кнопке Stop в окне статистики или выбрав Stop в меню Capture. Если вы задали опции пределов перехвата, то по их достижении сеанс остановится автоматически. Теперь можно анализировать результаты сеанса и манипулировать ими

Щелкая мышью на заголовках сверху окна, можно переупорядочить результаты по этому заголовку, так что можно сортировать вывод по исходным и целевым адресам, протоколу или информационному полю. Это помогает организовать данные, если вы ищите трафик определенного вида, например, все запросы DNS или весь трафик, связанный с почтой. Конечно, в первую очередь стоит написать фильтр для перехвата трафика определенного вида

Опции отображения

В [табл. 6.8](#) перечислены команды из меню Display, при помощи которых можно воздействовать на способ отображения пакетов на экране

Таблица 6.8. Опции меню Diplay Ethereal

| Пункт меню | Описание |
|---------------------|--|
| Подменю Options | Здесь можно установить глобальные параметры, такие как способ вычисления поля времени. Можно также включить автоматическую прокрутку трафика и разрешение имен, так как по умолчанию они отключены |
| Colorize display | Можно указать, чтобы пакеты определенных видов окрашивались определенным цветом. Это облегчает восприятие вывода и фокусирует внимание на нужной информации |
| Collapse/expand all | Показывать либо все детали каждого элемента, либо только верхний уровень |

Средства Ethereal

Вместе с Ethereal поставляется несколько встроенных аналитических средств. Данная программа построена в архитектуре со встраиваемыми модулями, поэтому другие программы могут взаимодействовать с Ethereal, или вы можете написать свою собственную. Доступ к этим возможностям находится в меню Tools ([табл. 6.9](#))

Таблица 6.9. Опции меню Tools Ethereal

| Пункт меню | Описание |
|-------------------------------|--|
| Summary | Показывает список данных верхнего уровня сеанса перехвата, например, затраченное время, число пакетов, средний размер пакета, общее количество перехваченных пакетов и среднюю плотность данных в среде передачи во время перехвата |
| Protocol hierarchy statistics | Выдает статистическое представление трафика вашей сети. Показывает, какой процент сеанса перехвата составляет каждый тип пакетов. Можно свернуть или распахнуть представление, чтобы увидеть основные уровни или второстепенные протоколы определенного уровня |
| Statistics | Содержит ряд отчетов, специфичных для определенных типов протоколов. Дополнительную информацию по этому вопросу можно найти в документации Ethereal |
| Plugins | Показывает встраиваемые модули анализатора пакетов, которые вы загрузили. Это декодировщики для новых протоколов, которые можно добавлять к Ethereal, не изменяя основной версии программы. И поскольку это архитектура со встраиваемыми модулями, можно писать свои собственные |

Сохранение вывода Ethereal

Закончив перехват и анализ данных в Ethereal, можно сохранить их либо для анализа дополнительными средствами, либо для предоставления другим пользователям. С помощью опции Save As меню File можно выбрать подходящий формат, включая libpcap (по умолчанию), Sun Snoop, LANalyser, Sniffer, Microsoft Network Monitor и Visual Networks

Приложения Ethereal

Теперь, после знакомства с основами Ethereal, представим несколько практических приложений, для которых его можно использовать.

Оптимизация сети

Выполняя широко открытый перехват сети и используя затем статистические отчеты, можно понять, насколько загружена ваша сеть и на какой вид пакетов приходится основная часть трафика. Проанализировав эти данные, можно решить, что пришло время перейти на коммутируемую сеть 100 Мбит/с или разделить два отдела на маршрутизируемые ЛВС вместо одной большой сети. Можно также определить, что требуется установить сервер WINS (слишком много запросов имен SMB, передаваемых по сети), или что некоторый сервер необходимо перенести в демилитаризованную зону или на отдельный порт маршрутизатора, чтобы удалить ассоциированный с ним трафик из сети.

Поиск дефектов в работе серверов приложений

С вашим почтовым сервером не удастся установить соединение? Возникли проблемы с DNS? Устранить подобные дефекты прикладного уровня бывает крайне сложно. Однако, применяя *Ethereal*, можно подключиться к сети и понаблюдать за коммуникациями между серверами. Можно увидеть реальные сообщения серверов для таких протоколов, как SMTP или HTTP, и, следя за потоком TCP, определить, где возникает проблема.

7. Лекция: Системы обнаружения вторжений: версия для печати и PDA

В предыдущей лекции мы ознакомились с сетевыми анализаторами и многими полезными вещами, которые можно делать с их помощью. Анализаторы можно использовать даже для выявления подозрительной активности в сети. Еще один шаг в этом направлении можно сделать, используя программное обеспечение, называемое системами обнаружения вторжений. По сути эти программы представляют собой модифицированные анализаторы, которые видят все потоки данных в сети, пытаются выявить потенциально вредный сетевой трафик и предупредить вас, когда таковой появляется. Основным методом их действия заключается в исследовании проходящего трафика и сравнении его с базой данных известных шаблонов вредоносной активности, называемых сигнатурами. Использование сигнатур очень похоже на работу антивирусных программ. Большинство видов атак на уровне TCP/IP имеют характерные особенности. Система обнаружения вторжений может выявлять атаки на основе IP-адресов, номеров портов, информационного наполнения и произвольного числа критериев. Существует другой способ обнаружения вторжений на системном уровне, состоящий в контроле целостности ключевых файлов. Кроме того, развиваются новые методы, сочетающие концепции обнаружения вторжений и межсетевого экранирования или предпринимающие дополнительные действия помимо простого обнаружения (см. врезку "Новое поколение систем обнаружения вторжений"). Однако в этой лекции основное внимание уделено двум наиболее популярным способам обнаружения вторжений в сети и системах: сетевое обнаружение вторжений и контроль целостности файлов.

Обзор лекции

Изучаемые концепции:

- Типы систем обнаружения вторжений
- Сигнатуры для систем обнаружения вторжений
- Ложные срабатывания в сетевых системах обнаружения вторжений
- Правильное размещение систем обнаружения вторжений
- Настройка систем обнаружения вторжений
- Контроль целостности файлов

Используемые инструменты:

Snort, модуль Snort Webmin, Snort for Windows, Tripwire

Сетевая система обнаружения вторжений может защитить от атак, которые проходят через межсетевого экран во внутреннюю ЛВС. Межсетевые экраны могут быть неправильно сконфигурированы, пропуская в сеть нежелательный трафик. Даже при правильной работе межсетевые экраны обычно пропускают внутрь трафик некоторых приложений, который может быть опасным. Порты часто переправляются с межсетевого экрана внутренним серверам с трафиком, предназначенным для почтового или другого общедоступного сервера. Сетевая система обнаружения вторжений может отслеживать этот трафик и сигнализировать о потенциально опасных пакетах. Правильно сконфигурированная сетевая система обнаружения вторжений может перепроверять правила межсетевого экрана и предоставлять дополнительную защиту для серверов приложений.

Сетевые системы обнаружения вторжений полезны при защите от внешних атак, однако одним из их главных достоинств является способность выявлять внутренние атаки и подозрительную активность пользователей. Межсетевым экраном защитит от многих внешних атак, но, когда атакующий находится в локальной сети, межсетевым экраном вряд ли сможет помочь. Он видит только тот трафик, что проходит через него, и обычно слеп по отношению к активности в локальной сети. Считайте сетевую систему обнаружения вторжений и межсетевым экраном взаимодополняющими устройствами безопасности - вроде надежного дверного замка и системы сигнализации сетевой безопасности. Одно из них защищает вашу внешнюю границу, другое - внутреннюю часть ([рис. 7.1](#)).

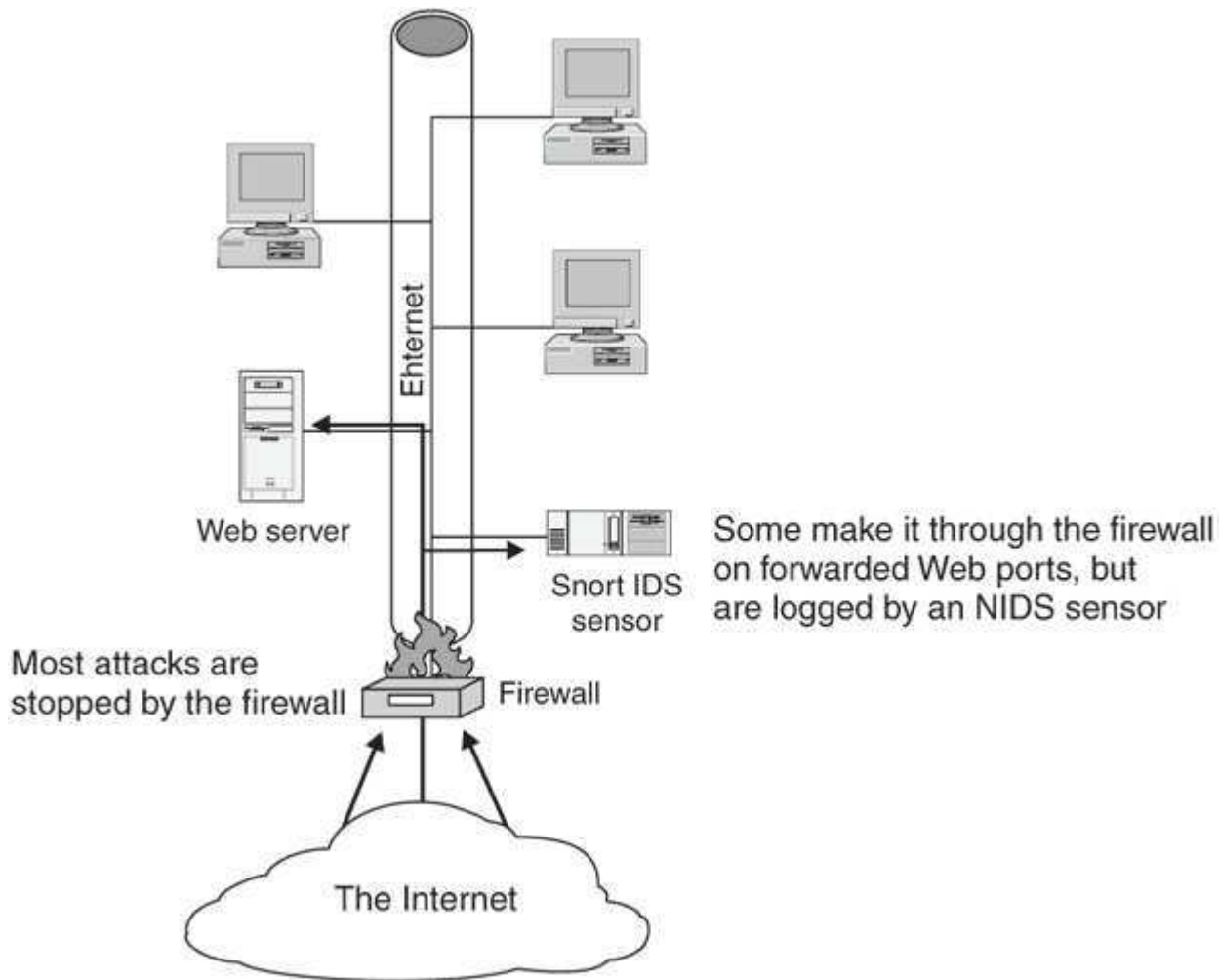


Рис. 7.1. Защита с помощью сетевой системы обнаружения вторжений и межсетевого экрана

Имеется веская причина, чтобы внимательно следить за трафиком внутренней сети. Как показывает статистика ФБР, более 70 процентов компьютерных преступлений исходят из внутреннего источника. Хотя мы склонны считать, что наши коллеги не сделают ничего, чтобы нам навредить, но иногда это бывает не так. Внутренние злоумышленники - не всегда ночные хакеры. Это могут быть и обиженные системные администраторы, и неосторожные служащие. Простое действие по загрузке файла или по открытию файла, присоединенного к электронному сообщению, может внедрить в вашу систему "троянскую" программу, которая создаст дыру в межсетевом экране для всевозможных бед. С помощью сетевой системы обнаружения вторжений вы сможете пресечь подобную активность, а также другие возможные компьютерные интриги. Хорошо настроенная сетевая система обнаружения вторжений может играть роль электронной "системы сигнализации" для вашей сети.

Новое поколение систем обнаружения вторжений

Системы обнаружения вторжений на основе выявления аномальной активности

Вместо применения статических сигнатур, с помощью которых можно выявлять только явно вредоносную деятельность, системы нового поколения отслеживают нормальные уровни для различных видов активности в сети. Если наблюдается внезапный всплеск трафика FTP, то система предупредит об

этом. Проблема с системами такого рода состоит в том, что они весьма склонны к ложным срабатываниям - то есть выдаче сигналов тревоги, когда в сети имеет место нормальная, допустимая деятельность. Так, в примере с FTP-трафиком загрузка особенно большого файла будет возбуждать сигнал тревоги.

Следует учитывать также, что системе обнаружения вторжений на основе выявления аномальной активности требуется время, чтобы построить точную модель сети. Вначале система генерирует так много тревожных сигналов, что пользы от нее почти никакой. Кроме того, подобные системы обнаружения вторжений можно обмануть, хорошо зная сеть. Если хакеры достаточно незаметны и используют протоколы, которые активно применяются в сети, они не привлекут внимания систем такого рода. С другой стороны, важное преимущество подобных систем - отсутствие необходимости постоянно обновлять набор сигнатур. Когда эта технология достигнет зрелости и достаточной интеллектуальности, она, вероятно, станет употребительным методом обнаружения вторжений.

Системы предотвращения вторжений

Новый тип сетевых систем обнаружения вторжений, называемый системами предотвращения вторжений, декларирован как решение всех проблем корпоративной безопасности. Основная идея состоит в том, чтобы при генерации тревожных сигналов предпринимать ответные действия, такие как написание на лету индивидуальных правил для межсетевых экранов и маршрутизаторов, блокирующих активность подозрительных IP-адресов, запрос или даже контратака систем-нарушителей.

Хотя эта новая технология постоянно развивается и совершенствуется, ей еще слишком далеко до проведения анализа и принятия решений на уровне человека. Факт остается фактом - любая система, которая на 100% зависит от машины и программного обеспечения, всегда может быть обманута посвятившим себя этому человеком (хотя некоторые проигравшие шахматные гроссмейстеры могут с этим не согласиться). Примером системы предотвращения вторжений с открытыми исходными текстами служит Inline Snort Джеда Хейла - свободный модуль для сетевой системы обнаружения вторжений Snort, обсуждаемой в данной лекции.

Примеры сигнатур сетевых систем обнаружения вторжений

Сетевые системы обнаружения вторжений действуют, проверяя пакеты и сравнивая их с известными сигнатурами. Хорошим примером распространенной атаки, которую можно четко идентифицировать по ее сигнатуре, является атака cmd.exe, направленная против Информационного Сервера Интернет (IIS) - web-сервера корпорации Microsoft. Эта атака применяется Интернет-"червями" и вирусами, такими как Nimda и Code Red. Атакующий "червь" или человек пытается выполнить в каталоге с правом на запись копию программы cmd.exe - командного интерпретатора Windows, используя переполнение буфера в модуле IIS, называемом Internet Server API (ISAPI). В случае успеха хакер или червь получает доступ к командной строке на этой машине и может произвести значительные разрушения. Однако команда для копирования этого файла является очевидной и нет причины для ее легального выполнения пользователями через сеть с помощью IIS. Поэтому, если вы видите подобную активность, то весьма вероятно, что это попытка вторжения. Проверка полезную нагрузку пакета и разыскивая слова cmd.exe, сетевая система обнаружения вторжений может идентифицировать данную атаку. На [листинге 7.1](#) показан один из таких пакетов. Шестнадцатеричное представление содержимого находится слева, а перевод в текст - справа.

```
length = 55
000 : 47 45 54 20 2F 73 63 72 69 70 74 73 2F 2E 2E 25 GET / scripts/..%
010 : 35 63 25 35 63 2E 2E 2F 77 69 6E 6E 74 2F 73 79 5c%5c../winnt/sy
020 : 73 74 65 6D 33 32 2F 63 6D 64 2E 65 78 65 3F 2F stem32/cmd.exe?/
030 : 63 2B 64 69 72 0D 0A c+dir..
```

Листинг 7.1. Пакет выполнения cmd.exe

Другой атакой, которую легко идентифицировать по ее сигнатуре, является переполнение буфера .ida. "Червь" Code Red распространялся с помощью этого метода. Эксплуатируется переполнение буфера в расширении .ida для web-сервера Microsoft IIS. Это расширение установлено по умолчанию, но часто не требуется. Если вы не наложили заплату на это место, оно может предоставить прямой доступ к вашей машине. По счастью, сетевая система обнаружения вторжений способна быстро идентифицировать эти пакеты, находя содержащийся в них оператор GET /default.ida. Частичный листинг атаки .ida показан на [листинге 7.2](#). В этом конкретном примере присутствуют также слова Code Red II, свидетельствующие о том, что он был создан "червем" Code Red, пытавшимся инфицировать данную машину. Даже если ваша машина полностью защищена от подобных атак, не мешает выяснить, откуда они приходят и с какой частотой.

```

length= 1414
000 : 47 45 54 20 2F 64 65 66 61 75 6C 74 2E 69 64 61 GET /default.ida
010 : 3F 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
020 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
030 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
040 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
050 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
060 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
070 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
080 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
090 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0a0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0b0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0c0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0d0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0e0 : 58 58 58 58 58 58 58 58 58 58 58 58 58 58 58 ?XXXXXXXXXXXXXXXXXX
0f0 : 58 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63 X%u9090%u6858%uc
100 : 62 64 33 25 75 37 38 30 31 25 75 75 39 30 39 25 bd3%u7801%u9090%
110 : 75 36 38 35 38 25 75 63 62 64 33 25 75 37 38 30 u6858%ucbd3%u780
120 : 31 25 75 39 30 39 30 25 75 36 38 35 38 25 75 63 l%u9090%u6858%uc
130 : 62 64 33 25 75 37 38 30 31 25 75 39 30 39 30 25 bd3%u7801%u9090%
140 : 75 39 30 39 30 25 75 38 31 39 30 25 75 30 30 63 u9090%u8190%u00c
150 : 33 25 75 30 30 30 33 25 75 38 62 30 30 25 75 35 3%u0003%u8b00%u5
160 : 33 31 62 25 75 35 33 66 66 25 75 30 30 37 38 25 31b%u53ff%u0078%
170 : 75 30 30 30 30 25 75 30 30 3D 61 20 20 48 54 54 u0000%u00=a HTT
180 : 50 2F 31 2E 30 0D 0A 43 6F 6E 74 65 6E 74 2D 74 P/1.0..Content-t
190 : 79 70 65 3A 20 74 65 78 74 2F 78 6D 6C 0A 43 6F ype: text/xml.Co
1a0 : 6E 74 65 6E 74 2D 6C 65 6E 67 74 68 3A 20 33 33 ntent-length: 33
1b0 : 37 39 20 0D 0A 0D 0A C8 C8 01 00 60 E8 03 00 00 79 .....'. ....
1c0 : 00 CC EB FE 64 67 FF 36 00 00 64 67 89 26 00 00 ....dg.6..dg.&..
1d0 : E8 DF 02 00 00 68 04 01 00 00 8D 85 5C FE FF FF .....h.....\...
1e0 : 50 FF 55 9C 8D 85 5C FE FF FF FF 50 FF 55 98 8B 40 P.U....\...P.U...@
1f0 : 10 8B 08 89 8D 58 FE FF FF FF 55 E4 3D 04 04 00 .....X....U.=...
200 : 00 0F 94 C1 3D 04 08 00 00 0F 94 C5 0A CD 0F B6 ....=.....
210 : C9 89 8D 54 FE FF FF 8B 75 08 81 7E 30 9A 02 00 ...T....u...-0...
220 : 00 0F 84 C4 00 00 00 C7 46 30 9A 02 00 00 E8 0A .....F0.....
230 : 00 00 00 43 6F 64 65 52 65 64 49 49 00 8B 1C 24 ...CodeRedII...$

```

Листинг 7.2. Сигнатура атаки .ida

Проблема ложных срабатываний сетевых систем обнаружения вторжений

Одной из главных проблем систем обнаружения вторжений является их склонность к большому числу ложных срабатываний. Ложное срабатывание имеет место, когда система генерирует сигнал тревоги на основе того, что она считает вредоносной или подозрительной активностью, но что в действительности оказывается нормальным трафиком для данной сети. Обычно в подразумеваемой конфигурации сетевая система обнаружения вторжений будет реагировать на все хоть чуть-чуть необычное. У большинства подобных систем имеются обширные используемые по умолчанию базы данных из тысяч сигнатур возможной подозрительной активности. Производители сетевых систем обнаружения вторжений не могут знать характер вашего сетевого трафика, поэтому для перестраховки они предусматривают срабатывание по каждому поводу.

Типичные причины ложных срабатываний

Работа системы мониторинга сети

Многие организации используют системы мониторинга сети, такие как HP OpenView или WhatsUp Gold, чтобы следить за системами в своей сети. Они характеризуются высокой сетевой активностью опроса и обнаружения. Для опроса состояния эти системы обычно применяют SNMP или аналогичный протокол, но они могут также использовать эхо-тестирование и другие, более назойливые проверки. По умолчанию большинство систем обнаружения вторжений рассматривают эту активность как вредоносную или, по крайней мере, подозрительную. В большой сети мониторинг может порождать тысячи сигналов тревоги в час, если система обнаружения вторжений настроена для отслеживания такой деятельности. Этого можно избежать, игнорируя активность с участием IP-адреса системы мониторинга. Можно также исключить из базы данных соответствующие сигналы тревоги, если их отслеживание не представляет для вас особой важности.

Сетевое сканирование уязвимостей/сканеры портов

Всякий раз, когда вы запускаете сетевое тестирование уязвимостей или сканирование портов с помощью таких программ, как Nessus и Nmap, ваша сетевая система обнаружения вторжений будет сходиться с ума. Эти программы созданы для выполнения именно того, что делают хакеры. На самом деле, вероятно, сигналы тревоги заданы для большинства встраиваемых модулей Nessus. И здесь также можно отключить сообщения с участием IP-адреса сервера Nessus или Nmap, но лучше всего вообще выключать систему обнаружения вторжений на время планового сканирования. В этом случае сканирующая машина будет по-прежнему защищена от атак, когда не выполняется сканирование, а база данных сигналов тревоги не будет искажена множеством данных от вашей собственной активности по сканированию.

Пользовательская активность

Большинство сетевых систем обнаружения вторжений настроены для сигнализации об опасной активности пользователей, такой как одноранговое разделение файлов, мгновенный обмен сообщениями и т.д. Однако, если подобная активность допускается либо формальной политикой, либо просто несоблюдением существующих политик, то она будет фиксироваться в журналах в виде сигналов. Это может стать основанием для проведения в жизнь или создания политик против таких видов деятельности, так как можно показать, какую часть полосы пропускания и сколько времени они занимают, не говоря уже о последствиях для безопасности. Однако, если вы намерены и далее разрешать такую активность, то необходимо закомментировать эти правила, чтобы не заполнять журналы сигналами ложных срабатываний.

Поведение, напоминающее "троянскую" программу или "червя"

Современные вирусы и вирусоподобное программное обеспечение ("черви" и "троянские" программы) нередко используют сетевые средства, пытаясь выполнять такие действия, как инфицирование других машин или массовая рассылка электронных сообщений. Подобную активность можно выявить с помощью сетевых систем обнаружения вторжений. Однако эти сигнатуры могут порождать сигналы тревоги и при нормальной деятельности. Примером служит червь Nimda, который пытается копировать на различные системы файлы с определенными расширениями, такими как .eml. К сожалению, программа Microsoft Exchange ведет себя аналогично при использовании ее web-интерфейса. Поэтому, хотя знать о подобной "троянской" активности в сети было бы полезно, можно при желании отключить сигналы, порождаемые известной нормальной деятельностью, даже когда имеется потенциальная опасность, что трафик все-таки окажется вредоносным. Это поможет избежать чрезмерного количества ложных срабатываний.

Длинные базовые цепочки аутентификации

Сигнал такого типа ориентирован на чрезмерно длинные входные строки Web, поскольку некоторые программы использования уязвимостей применяют подобный метод для переполнения буфера и несанкционированного получения доступа. Однако в последнее время многие Web-сайты набивают в это поле много информации и могут ненароком сбить с толку сетевую систему обнаружения вторжений.

Аутентификационная активность базы данных

Некоторые сетевые системы обнаружения вторжений следят за деятельностью по администрированию баз данных. Теоретически в производственных базах данных не должно наблюдаться высокой административной активности, а ее наличие может служить признаком того, что кто-то пытается что-то сделать с базой. Однако во многих базах данных использование идет параллельно с разработкой, отсюда и большой объем администрирования. Эта деятельность, хотя и вполне законная, будет порождать множество сигналов тревоги. Если ваша база данных находится в состоянии непрерывного

развития, то вам, вероятно, следует отключить эти сигналы, по крайней мере пока база не стабилизируется и не перейдет в режим производственной эксплуатации.

Существует много других причин ложных срабатываний, зависящих от конфигурации сети и уровня активности. В подразумеваемой конфигурации сетевая система обнаружения вторжений может порождать сотни ложных срабатываний в день, что способно привести системного администратора в отчаяние. В результате сигналы тревоги этих систем вскоре начинают игнорироваться, как некий посторонний шум. Однако при небольших усилиях и с помощью методов, описанных в этой лекции, сетевая система обнаружения вторжений может быстро стать полезным средством, а не электронной версией мальчика, который то и дело кричал "Волк!".

Как получить максимум пользы от системы обнаружения вторжений

Чтобы реализовать истинный потенциал системы обнаружения вторжений, необходимо сделать несколько вещей, как перед установкой, так и после нее.

Правильное конфигурирование системы

Если вы только что установили систему обнаружения вторжений и запустили ее в подразумеваемой конфигурации, то тысячи ложных срабатываний скоро переполнят чашу вашего терпения. Хотя вы сможете перенастроить систему постфактум, лучше побережь силы и нервы, уделив некоторое время упреждающему конфигурированию. Не принимайте подразумеваемые настройки, индивидуализируйте их для своей ЛВС.

Большинство систем обнаружения вторжений группирует сигналы тревоги по категориям. Просмотрите каждую группу, чтобы решить, насколько она подходит для вашей сети. Если имеется группа сигнатур для UNIX-платформ, но у вас в сети нет UNIX-систем, то, вероятно, можно безопасно отключить весь этот пакет сигналов. В некоторых системах предусмотрены сигналы тревоги, зависящие от политики и отвечающие за такие вещи, как использование мгновенного обмена сообщениями или программного обеспечения одноранговых сетей. Если у вас уже есть системы, фильтрующие подобные виды активности, или вы их разрешаете, то эти сигналы можно отключить. Необходимо тщательно проанализировать группы сигналов. Хотя вам может пригодиться большинство сигналов для Windows-платформ, некоторые из них, возможно, не имеют отношения к вашей сети или будут вызывать ложные срабатывания.

Можно также освободить некоторые хосты от контроля. Если ваша персональная машина постоянно посылает в сеть SNMP-запросы, или вы постоянно входите как администратор, то это может порождать много бесполезных сигналов тревоги. Хотя освобождение от контроля снижает уровень безопасности и может оставить критически важные машины без защиты, оно способно сделать систему обнаружения вторжений более эффективной. Уделив несколько часов тщательному конфигурированию системы до ее активации, можно сберечь много времени и сил в будущем.

Настройка системы обнаружения вторжений

Когда система запущена и работает, даже при скрупулезном конфигурировании она начнет генерировать сигналы тревоги. Вскоре, если вы найдете время для их анализа и начнете деактивировать правила, которые не подходят для вашей сети, вы сможете снизить число ложных срабатываний. Попутно у вас появится понимание того, как работает ваша сеть и какие потоки данных текут по ней, что полезно для любого сетевого администратора. Каждую неделю выделяйте некоторое время для модификации настроек системы обнаружения вторжений. Некоторые системы позволяют относительно легко пометить сигнал как источник ложных срабатываний, тогда как другие заставят вас преодолеть некоторые препятствия. В среднем требуется несколько месяцев, чтобы настроить систему обнаружения вторжений на выдачу полезных сигналов о наказуемой активности, и то только в случае целенаправленных усилий по тонкой настройке.

Средства анализа для систем обнаружения вторжений

Системы обнаружения вторжений обычно предлагают администраторам несколько различных способов получения уведомлений о срабатывании сигналов тревоги. В простейшем случае сигналы могут просто протоколироваться для последующего просмотра. На самом деле это не рекомендуется, так как заставляет администратора неусыпно следить за регистрационными журналами. Если не делать этого ежедневно, то могут пройти дни или недели, прежде чем попытки вторжения будут обнаружены. Другой возможностью извещения соответствующего должностного лица о возникновении сигнала тревоги является отправка сообщения по электронной почте или на пейджер. Однако даже с хорошо настроенной системой получение на пейджер по несколько сообщений в день может доставлять слишком много хлопот. Кроме того, электронные сообщения будут иметь формат, в котором их сложно сравнивать с прошлыми сигналами тревоги или анализировать каким-то иным образом. Лучшим способом обработки сигналов тревоги является их немедленное

занесение в базу данных, чтобы можно было выполнить углубленный анализ. Существует средство с открытыми исходными текстами для систем обнаружения вторжений, называемое ACID (Analysis Console for Intrusion Detection - консоль анализа для обнаружения вторжений). Оно подробно рассматривается в [лекции 8](#).

Теперь, ознакомившись с тем, как работают системы обнаружения вторжений, давайте построим такую систему и запустим ее в работу.

Snort: система обнаружения вторжений для UNIX с открытыми исходными текстами

Snort

Автор/основной контакт: Martin Roesch

Web-сайт: <http://www.snort.org/>

Платформы: FreeBSD, Linux, Windows и некоторые UNIX

Лицензия: GPL

Рассмотренная версия: 2.1.1

Списки почтовой рассылки:

Snort-announcements

Общие объявления о версиях и коррекциях. Не для обсуждения. Подписка по адресу lists.sourceforge.net/lists/listinfo/snort-announce.

Snort-users

Общая дискуссия о Snort. Новички приветствуются. Подписка по адресу lists.sourceforge.net/lists/listinfo/snort-users.

Snort-developers

Для разработчиков или желающих разрабатывать код ядра snort. Подписка по адресу lists.sourceforge.net/lists/listinfo/snort-developers.

Snort-sigs

Для разработчиков или желающих разрабатывать правила snort. Подписка по адресу lists.sourceforge.net/lists/listinfo/snort-sigs.

Snort-cvsinfo

Только для активных разработчиков, желающих получать уведомления при обновлении дерева CVS. Дискуссии не допускаются. Подписка по адресу lists.sourceforge.net/lists/listinfo/snort-cvsinfo.

На сайте Snort доступен архив прошлых сообщений. При возникновении вопроса целесообразно сначала поискать ответ в архиве. Вполне возможно, что кто-то встречался с вашей проблемой раньше. Посетите <http://www.snort.org/lists.html>

Существуют локальные группы пользователей, которые время от времени собираются для обсуждения различных вопросов, связанных со Snort. Список этих групп представлен на <http://www.snort.org/user-groups.html>.

Примерно в полудюжине крупных городов имеются активные группы пользователей, и еще в дюжине подобные группы находятся в стадии становления. Форма на упомянутой выше web-странице позволяет выразить заинтересованность в создании такой группы, если в ваших краях ее еще нет.

Snort - творение Мартина Реша, вышедшее, однако, далеко за пределы его авторства. В настоящее время ядро группы разработчиков насчитывает более 30 человек, не считая тех, кто пишет правила и другие части программного обеспечения. Как можно видеть из приведенных выше списков рассылки, существует много доступных источников информации о Snort. И это только бесплатные сетевые ресурсы. Имеется также несколько полноформатных книг на эту тему. Данный раздел, хотя и не является истиной в последней инстанции, дает достаточно сведений об основах, позволяет освоить Snort и работать с ним.

Snort можно отнести к системам обнаружения вторжений на основе сигнатур, хотя с добавлением модуля Spade он приобрел способность выявлять аномальную активность. Имеются также дополнительные модули, такие как Inline Snort, которые позволяют автоматически реагировать на любые сигналы тревоги.

Уникальные особенности Snort

- Открытые исходные тексты. Исходные тексты Snort открыты, он переносим практически на любую разновидность операционной системы UNIX. Доступны также версии для Windows и других операционных систем.
- Легковесность. В силу эффективной реализации Snort не требует мощного оборудования (см. врезку "Оборудование"). Это позволяет анализировать трафик в сети 100 Мбит/с практически в реальном масштабе времени, что кажется невероятным, если представить, что делается с каждым пакетом.
- Индивидуальные правила Snort. Snort предлагает простой способ расширения и индивидуализации программы путем написания собственных правил или сигнатур. Обширная документация помогает научиться этому, не говоря уже о сетевых форумах и справочных списках.

Установка Snort

Snort устанавливается довольно просто.

1. В качестве предварительного условия требуется установить пакет libpcap. Если вы загрузили любой из пакетов из лекций с 4 по 6, то libpcap уже установлен. В противном случае его можно загрузить с <http://www.tcpdump.org/>.
2. После загрузки этих библиотек просто возьмите файл с компакт-диска, прилагаемого к книге, или загрузите самую свежую версию с web-сайта.
3. Когда файл окажется в вашей машине, распакуйте его и выполните команды компиляции:

```
./configure
make
make install
```

Запуск Snort

Snort запускается из командной строки. Его можно выполнять в трех различных режимах: анализа, протоколирования и обнаружения вторжений. Последний режим является наиболее употребительным, но имеются применения и для первых двух.

Режим анализа пакетов

В этом режиме Snort действует просто как анализатор, показывая нефильтованное содержимое среды передачи. Конечно, если вам требуется только анализатор, можно применить Tcpdump или Ethereal, однако данный режим позволяет убедиться, что все работает правильно и Snort видит пакеты. В [табл. 7.1](#) перечислены ключи, которые можно использовать при выполнении Snort в режиме анализа. Необходимо включить как минимум команду `-v`, поскольку иначе Snort по умолчанию будет выполняться в одном из двух других режимов (протоколирования или обнаружения вторжений), ожидая других опций.

Испробовать этот режим можно, просто набрав в командной строке

```
snort -v
```

или

```
snort -vde
```

Выдача будет практически такой же, как от анализаторов, описанных в предыдущей лекции. Для выхода нажмите Ctrl+C, и вы увидите сводные данные сеанса анализа пакетов.

Таблица 7.1. Опции режима анализа пакетов

| Опция | Описание |
|-------|--|
| -v | Выдает на экран заголовки пакетов TCP/IP в сети Ethernet |
| -d | Аналогично предыдущей опции, но отображаются также данные прикладного уровня |
| -e | Аналогично предыдущей опции, но выдаются также заголовки канального уровня |

Требования к оборудованию для сетевых систем обнаружения вторжений

Есть ряд моментов, которые нужно учитывать при выборе оборудования для работы сетевых систем обнаружения вторжений. Поскольку системы обнаружения, как правило, активно используют процессор и дисковое пространство, настоятельно рекомендуется, чтобы сетевая система обнаружения вторжений выполнялась на специально выделенном компьютере. Однако, поскольку система функционирует на платформе Linux, она все равно потребует меньше оборудования, чем эквивалентная машина Windows. При этом предполагается, что не используется графическая среда X-Window, которая для Snort не нужна, но существенно увеличивает нагрузку на процессор.

Для работы Snort желательно иметь процессор Intel 500 МГц, хотя можно обойтись и ПК с 266 МГц. Если вы храните файлы журналов локально, вам потребуется также по крайней мере несколько гигабайт доступного дискового пространства. Должна применяться сетевая плата 100 Мбит/с, чтобы исключить возможность заторов, если вы будете анализировать сеть 100 Мбит/с. Авторы Snort утверждают, что программа будет работать в активно используемом сегменте сети 100 Мбит/с без потери пакетов. Однако, если ваша сеть перегружена, то, возможно, придется несколько повысить требования к оборудованию - до процессора 1 ГГц. Так или иначе, необходимым требованиям легко удовлетворит любая машина, кроме разве что самых старых.

Режим протоколирования пакетов

Этот режим аналогичен предыдущему, но позволяет записывать пакеты на диск для последующего анализа, аналогично функциям протоколирования в описанных выше анализаторах. Чтобы запустить Snort в режиме протоколирования, воспользуйтесь той же командой, что и для режима анализа (-v, -d и/или -e), но с добавлением ключа -l `каталог_журналов`, задающего маршрутное имя каталога журналов, в которые Snort будет записывать пакеты. Пример:

```
snort -vde -l /var/log/snort
```

Эта команда создаст файлы журналов в каталоге /var/log/snort. Убедитесь, что указанный каталог существует, иначе программа не будет загружаться правильно. Snort протоколирует пакеты по IP-адресам, создавая отдельный каталог для каждого из них. Если вы протоколируете трафик в большой локальной сети с множеством адресов, ситуация может быстро выйти из-под контроля. Поэтому можно применить другую настройку, чтобы Snort протоколировал пакеты относительно вашей домашней сети, в которой вы находитесь. Это делается с помощью команды -h `домашняя_сеть`, где `домашняя_сеть` - диапазон IP-адресов локальной сети в нотации с косой чертой. В этом случае Snort будет помещать пакеты в каталоги на основе нелокального IP-адреса в пакете, что позволяет легко распознавать "неместный" трафик. Если оба хоста, целевой и исходный, являются локальными, Snort помещает пакет в каталог, соответствующий стороне с большим номером порта, как бы отдавая предпочтение подключающемуся хосту перед серверным. В случае равенства номеров портов Snort по умолчанию использует исходный адрес в качестве каталога для размещения данных пакета. Сейчас это может показаться несущественным, но если вы протоколируете сигналы о вторжении, важно быстро определить, откуда исходит подозрительный трафик.

Учитывая приведенные соображения, командной строке для режима протоколирования пакетов целесообразно придать следующий вид:

```
snort -vde -l /var/log/snort -h 192.168.1.0/24
```

Тем самым внутренняя сеть задается диапазоном 192.168.1.1-254.

Можно также применить опцию `-b` для протоколирования всех данных в одном бинарном файле, пригодном для последующего чтения с помощью анализатора пакетов, такого как `Ethereal` или `Tcpdump`. При протоколировании с опцией `-b` нет необходимости определять домашнюю сеть, так как данные будут записываться последовательно в один большой файл. Этот метод намного быстрее для протоколирования работы активно используемых сетей или на медленных машинах. Он также облегчает анализ с помощью более развитых средств, которые приходится применять при просмотре больших объемов перехваченных сетевых данных.

Режим обнаружения вторжений

В этом режиме Snort протоколирует подозрительные или требующие дополнительного внимания пакеты. Для перевода Snort в режим обнаружения вторжений достаточно добавить к приведенной выше инструкции ключ `-c` *конфигурационный_файл*, предписывающий использовать указанный конфигурационный файл для управления протоколированием пакетов. Конфигурационный файл определяет все настройки Snort, он очень важен. Snort поставляется с подразумеваемым конфигурационным файлом, но перед запуском в него целесообразно внести некоторые изменения, отражающие специфику вашей среды. Поэтому, набрав в командной строке

```
snort -de -l /var/log/snort -h 192.168.1.0/24 -c /etc/snort/snort.conf
```

вы запустите Snort в режиме обнаружения вторжений с использованием подразумеваемого конфигурационного файла `snort.conf`. Убедитесь, что указанный конфигурационный файл существует, или задайте маршрутное имя, соответствующее его расположению в вашей системе.

Обратите внимание, что я не использовал ключ `-v` для запуска Snort в режиме обнаружения вторжений. Если, помимо сопоставления всех пакетов с сигнатурами, заставлять Snort еще и выдавать на экран сигналы тревоги, это может привести к потере пакетов, особенно в загруженных сетях. Можно также не задавать ключ `-e`, чтобы повысить производительность, если не требуется протолировать работу канального уровня. Если убрать ключ `-l`, то Snort будет использовать подразумеваемый каталог протоколов `/var/log/snort`. Опять-таки убедитесь, что этот каталог существует, иначе Snort не запустится. Можно также задать ключ `-b`, если вы хотите направить протокол в бинарный файл для последующего анализа отдельной программой. Команда для запуска Snort в режиме обнаружения вторжений в результате будет выглядеть следующим образом:

```
snort -h 192.168.1.0/24 -c /etc/snort/snort.conf
```

Режимы сигнализации Snort

При протоколировании пакетов, вызывающих сигналы тревоги, необходимо выбрать подходящий уровень детализации и формат "тревожных" данных. В [табл. 7.2](#) перечислены опции, которые можно задавать в командной строке после ключа `-A`.

Таблица 7.2. Опции режима сигнализации Snort

| Опция | Описание |
|-----------|--|
| -A full | Полная информация о сигнале, включая прикладные данные. Это подразумеваемый режим сигнализации. Он будет использоваться при отсутствии спецификаций |
| -A fast | Быстрый режим. Протоколируются только заголовки пакетов и тип сигналов. Это полезно в очень быстрых сетях, но если требуется дополнительная судебная информация, необходимо использовать опцию <code>full</code> |
| -A unsock | Посылает сигнал в UNIX-сокет с указанным номером, на котором может слушать другая программа |
| -A none | Отключает сигналы тревоги |

Имеются также опции вывода `syslog`, `smb` и `database`, но они используют не ключ `-A`, а отдельные модули вывода и предлагают более широкое разнообразие выходных форматов. Эти опции следует конфигурировать во время компиляции при помощи ключей инструкции `configure`.

- SMB посылает сигналы тревоги службе всплывающих окон Windows, поэтому вы увидите сигналы всплывающими на вашем экране или экране машины, осуществляющей мониторинг. Однако, прежде чем использовать эту опцию, желательно тщательно настроить систему обнаружения вторжений, иначе вы не сможете ничего делать, кроме как наблюдать всплывающие то и дело окна! Для того чтобы включить этот метод сигнализации, при установке Snort задайте в инструкции `configure` опцию `enable-smbalerts`. Затем нужно запустить `snort` со следующими аргументами

```
snort -c /etc/snort.conf -M рабочие_станции
```

задав после `-M` имена хостов Windows, на которые отправляются сигналы.

- Syslog посылает сигналы тревоги Syslog-серверу UNIX. Syslog - это служба, выполняющаяся на машине (обычно UNIX), которая может подхватывать и сохранять различные файлы журналов. Это помогает консолидировать журналы вашей сети в одном месте, а также затрудняет хакеру удаление протоколов вторжений. В данной книге не рассматриваются особенности настройки сервера Syslog, но если он у вас есть, то при наличии в командной строке ключа `-s` Snort будет посылать сигналы туда. Можно также определить в конфигурационном файле различные форматы Syslog, которые рассматриваются в следующем разделе.
- Snort напрямую поддерживает четыре вида вывода в базу данных посредством своих модулей вывода. К числу поддерживаемых форматов принадлежат MySQL, PostgreSQL, Oracle и unixODBC. Это должно удовлетворить потребности большинства пользователей баз данных. И, естественно, если ваша база данных не поддерживается, можно взяться за проект по написанию нужного модуля расширения. Модуль вывода в базу данных требует как параметров времени компиляции, так и настроек в конфигурационном файле. Более подробные сведения - в следующем разделе.

Конфигурирование Snort для достижения максимальной производительности

Теперь, когда система Snort установлена, а вы ознакомились с основными командами, следует отредактировать конфигурационный файл, чтобы сделать ее надежной системой обнаружения вторжений и получать требуемые результаты. Подразумеваемым конфигурационным файлом служит `snort.conf`, который по умолчанию помещается в `/etc/snort.conf`. В этом файле задаются все настройки Snort. Имя этого файла можно изменить, если при запуске Snort после ключа `-c` указать новое маршрутное имя. Данный файл можно редактировать с помощью `vi`, `EMACS` или другого текстового редактора. Многие строки в этом файле начинаются со знака `#`, за которым следуют различные комментарии. Знак `#` служит стандартным началом строк комментариев, которые многие интерпретаторы, командные или такие как Perl, игнорируют. Строки комментариев применяются для документирования программ или для отключения старого кода. Вы будете использовать их позже при тонкой настройке набора правил. Но пока единственными строками, реально воздействующими на конфигурацию, являются строки без знака `#` в начале. Остальные присутствуют только для информационных целей. Конфигурационный файл настраивается в несколько шагов.

1. Задание домашней сети.

Необходимо сообщить Snort адреса, которые представляют домашнюю сеть, чтобы он мог правильно интерпретировать внешние атаки. Это делается с помощью инструкции

```
var HOME_NET адреса
```

где адреса следует заменить на адресное пространство вашей локальной сети. Если имеется несколько сетей, то можно ввести их все, разделяя запятыми. Можно также ввести имя интерфейса, чтобы IP-адрес и маска сети, присвоенные этому интерфейсу, использовались как `HOME_NET`. Для этого служит следующий формат:

```
var HOME_NET $ имя_интерфейса
```

где `имя_интерфейса` заменяется интерфейсом, на котором слушает Snort (например, `eth0` или `eth1`).

С помощью аналогичной инструкции можно определить внешние сети, заменяя `HOME_NET` на `EXTERNAL_NET`. Подразумеваемым значением обеих переменных служит `any`. Можно оставить их в таком виде или определить одну или обе. Я рекомендую определить внутреннюю сеть, но оставить внешние сети заданными как `any`.

2. Задание внутренних серверов.

В конфигурационном файле можно определить ряд серверов, включая Web, mail, DNS, SQL и Telnet. Это уменьшит число ложных срабатываний для этих сервисов на этих машинах.

Можно также задать номера портов для этих сервисов, чтобы регистрировались только атаки на указанные порты. Все эти опции конфигурации позволяют сократить число ложных срабатываний, чтобы до вас доводилась только информация, обладающая реальной ценностью. Имеется также раздел для добавления серверов AIM, чтобы отслеживать применение AOL Instant Messenger. Это имеет смысл только в том случае, если включен класс правил Chat.

3. Конфигурирование декодировщиков и препроцессоров Snort.

Ряд ключей и настроек в конфигурационном файле управляют декодировщиками и препроцессорами Snort. Эти процедуры применяются к трафику, прежде чем он пройдет через какой-либо набор правил, обычно с целью правильного форматирования или для обработки определенного вида трафика, который проще препроцессировать, чем применять наборы правил. Примером подобного типа трафика служат фрагментированные пакеты. В Snort присутствует декодировщик, собирающий фрагментированные пакеты. Многие атаки пытаются скрыть свою истинную природу, фрагментируя пакеты, так что описываемые возможности Snort являются весьма ценными.

Другой декодировщик предназначен для пакетов сканирования портов. Так как они имеют склонность приходить группами и в большом количестве, лучше обрабатывать их заранее общей массой, чем пытаться сравнивать каждый пакет с сигнатурой. Это также делает систему обнаружения вторжений более защищенной от атак на доступность. Подразумеваемые настройки для этих подсистем должны работать хорошо, однако, получив некоторый опыт работы со Snort, вы можете попытаться изменить их, чтобы повысить производительность и достичь лучших результатов.

4. Конфигурирование модулей вывода.

Это важный шаг, если вы хотите использовать базу данных при обработке вывода Snort. Здесь вы указываете программе, как обрабатывать данные сигналов тревоги. Имеется несколько модулей вывода, которые можно применять в зависимости от требуемого формата данных: Syslog, Database и новый модуль Unified, который поддерживает универсальный бинарный формат, полезный для импорта данных другими программами. Общий формат для конфигурирования модулей вывода таков:

```
output имя_модуля: конфигурация опции
```

где `имя_модуля` следует заменить на `alert_syslog`, `database` или `alert_unified` в зависимости от используемого модуля.

Опциями конфигурации для различных модулей вывода служат:

- Syslog

Для систем UNIX/Linux нужно использовать следующую директиву:

```
output alert_syslog: LOG_AUTH LOG_ALERT
```

Для Windows-систем можно использовать любой из следующих форматов:

```
output alert_syslog: LOG_AUTH LOG_ALERT
output alert_syslog: host=имя_хоста, LOG_AUTH LOG_ALERT
output alert_syslog: host=имя_хоста:порт, LOG_AUTH LOG_ALERT
```

где `имя_хоста` и `порт` нужно заменить, соответственно, на IP-адрес и порт сервера Syslog.

- Database

Общий формат для настройки вывода в базу данных таков:

```
output database: log, тип_базы_данных, user=имя_пользователя
password=пароль dbname=имя_базы_данных host=адрес_базы_данных
```

где `тип_базы_данных` заменяется одной из допустимых для Snort разновидностей баз данных (MySQL, postgresql, unixodbc или mssql), `имя_пользователя` - допустимым именем пользователя машины базы данных, а `пароль` - его паролем. Переменная `dbname` задает имя базы данных для протоколирования. Наконец, `адрес_базы_данных` является IP-адресом сервера базы данных. Не рекомендуется устанавливать Snort и базу данных на один сервер. Помимо того, что безопаснее держать данные сигналов тревоги на другом компьютере, работа Snort и базы данных на одной машине будет существенно снижать производительность. Хотя настройка баз данных не является темой книги, базовая конфигурация базы данных MySQL для Snort и других программ обсуждается в [лекции 8](#).

- Unified

Это основной бинарный формат быстрого протоколирования и сохранения для будущего использования. Поддерживаются два аргумента - `filename` и `limit`, а вся директива может выглядеть примерно так:

```
output alert_unified: filename snort.alert, limit 128
```

5. Индивидуализация наборов правил.

Можно выполнить тонкую настройку Snort, добавляя или удаляя наборы правил. Файл `snort.conf` позволяет добавлять или удалять целые классы правил. В конце файла перечислены все наборы правил генерации сигналов тревоги. Можно отключить целые категории правил, закомментировав строки с помощью знака `#` в начале. Например, можно отключить все правила `icmp-info` для снижения числа ложных срабатываний на трафике `ping` или все правила `NetBIOS`, если в сети нет машин Windows. Имеются также общедоступные наборы правил, уже настроенные для определенных сред.

Закончив внесение изменений в конфигурационный файл, сохраните его, и тогда все будет готово для запуска Snort.

Правильное размещение сетевой системы обнаружения вторжений

Решая, где разместить сетевую систему обнаружения вторжений, следует принять во внимание, что именно вы пытаетесь защитить и как можно максимизировать эффективность и взаимную поддержку средств сетевой безопасности. Имеется несколько вариантов размещения сетевой системы обнаружения вторжений, у каждого из которых есть свои достоинства и недостатки.

- В ЛВС позади межсетевого экрана. Это наиболее распространенная конфигурация, которая предлагает наилучшую защиту как от внешних, так и от внутренних угроз. Прослушивая локальную среду передачи, можно выявлять внутреннюю активность пользователей, такую как взаимодействие между рабочими станциями или ненадлежащее применение программ. Это также обеспечивает дополнительную поддержку межсетевого экрана, позволяя обнаружить подозрительный трафик, каким-то образом сумевший проникнуть во внутреннюю сеть через фильтры экрана. В действительности, систему обнаружения вторжений можно применять для тестирования межсетевого экрана, чтобы увидеть, какой трафик он пропускает.

Однако при подобном размещении будет генерироваться много сигналов тревоги на основе потоков данных Windows, так что будьте готовы проделать большой объем работы по настройке в этой области. Далее, если у вас коммутируемая ЛВС, то понадобится возможность отражения всех портов в порт монитора, чтобы система обнаружения вторжений могла прослушивать весь трафик ЛВС.

- В демилитаризованной зоне. Можно поместить сенсор Snort в демилитаризованной зоне, чтобы отслеживать активность по отношению к вашим общедоступным серверам. Так как эти серверы наиболее открыты в вашей организации и обычно представляют собой ценные ресурсы, то весьма

разумно наблюдать за ними с помощью системы обнаружения вторжений. Проблема, которая возникает при подобной конфигурации, состоит в сортировке всех сигналов. Хотя все они могут быть оправданными сигналами тревоги, в наше время общий уровень атакующего трафика в Интернет таков, что любой общедоступный IP-адрес по несколько раз в день подвергается случайным атакам. Реагирование и попытки отследить эти сигналы будут излишними и контрпродуктивными.

Как же отличить обычных "червей", отраженных вашим сервером, от пакетов, которые действительно уносят что-то ценное? Один из возможных подходов состоит в сокращении числа сигнатур до небольшой величины, чтобы срабатывания происходили, только если компьютер действительно был скомпрометирован. Примером могут служить специальные правила для приложений, выполняющихся на этом компьютере, такие как правила для MySQL или web-iis, или правила, связанные с административным входом в систему. Можно исключить большинство сигналов зондирующего характера и не реагировать на такую деятельность, как сканирование портов и т.д.

- Между вашим поставщиком Интернет-услуг и межсетевым экраном. В этом случае будет фильтроваться весь входящий и исходящий трафик вашей ЛВС и демилитаризованной зоны. Положительная сторона этого подхода состоит в том, что вы будете перехватывать все, что направлено против ваших общедоступных серверов и внутренней ЛВС, отрицательная - в том, что вы не увидите внутренний трафик, а общий объем сигналов может быть весьма большим из-за высокого уровня фонового атакующего трафика.

Как и в предыдущем примере, попробуйте ограничить набор сигналов, оставив включенными только те, которые действительно будут отражать проблему для данного сетевого сегмента. Следует учитывать также, что сенсор, размещенный в канале между вашим поставщиком Интернет-услуг и межсетевым экраном, может стать узким горлом и одиночной точкой отказа для сетевого трафика. Возможное решение состоит в установке небольшого концентратора между двумя каналами и в подключении системы обнаружения вторжений к нему.

Мы перечислили все разумные варианты размещения системы обнаружения вторжений. Разумеется, ничто не мешает использовать их все, если у вас достаточно оборудования и времени для управления.

Отключение правил в Snort

Простейшим способом ограничения потока сигналов является отключение правил, неприменимых к вашей системе. Для этого нужно войти в компьютер, на котором выполняется Snort, и найти каталог rules (обычно в каталоге, в котором установлен Snort). В этом каталоге имеется много файлов с расширением .rules. Каждый из них содержит множество правил, сгруппированных по категориям. Можно отключить целый класс правил, закомментировав его в конфигурационном файле, или же отключать отдельные правила, если вы хотите сохранить защиту других правил этого класса. Чтобы закомментировать правило, нужно найти его в соответствующих файлах .rules и поместить символ # перед строкой этого правила. Отметим, что обычно лучше отключать отдельные правила, а не классы, за исключением случаев, когда какой-то класс целиком неприменим для вас. В [табл. 7.3](#) перечислены все имена файлов для классов правил Snort и приведено их краткое описание.

Таблица 7.3. Имена файлов классов правил Snort

| Класс правил | Описание |
|------------------------|---|
| attack-responses.rules | Это сигналы для пакетов обычных ответов после успешных атак. Они редко оказываются ложными. В большинстве случаев их следует оставить включенными. |
| backdoor.rules | Это обычные признаки использования потайных входов или "троянских" программ. Они редко бывают ложными |
| bad-traffic.rules | Эти правила представляют нестандартный сетевой трафик, который обычно не должен присутствовать в большинстве сетей |
| chat.rules | Ищет стандартные признаки многих популярных программ чата. Если чат допускается явно или неявно, то эти сигналы необходимо отключить. Отметим также, что они не являются универсальным средством для чата и могут не обнаруживать все виды трафика чата. Тем не менее, они могут быть полезны для выявления наиболее злостных нарушителей |
| ddos.rules | Ищет стандартные распределенные атаки на доступность. В демилитаризованной зоне и глобальной сети эти сигналы будут бесполезны, так как в случае распределенной атаки на доступность вы узнаете об этом, вероятно, сразу. Однако они могут быть весьма ценными в локальной сети для обнаружения машин-зомби, бессознательно участвующих в распределенной атаке на доступность другой сети |

| | |
|--------------------|--|
| dns.rules | Ищет некоторые стандартные атаки против серверов DNS. Если у вас нет собственного сервера DNS, эти правила можно отключить |
| dos.rules | Аналогично вышеупомянутому набору правил ddos.rules. |
| experimental.rules | Отключены по умолчанию. Они обычно используются только для тестирования новых правил, пока они не будут перемещены в одну из других категорий |
| exploit.rules | Предназначены для стандартного трафика использования уязвимостей и всегда должны быть включены |
| finger.rules | Эти правила сигнализируют о трафике, связанном с серверами finger. Если вы не используете finger, то можно, наверное, их отключить. Однако серверы finger часто выполняются скрытно от системного администратора, поэтому можно оставить их включенными, так как они не будут вызывать ложных срабатываний, если у вас нет серверов finger |
| ftp.rules | Аналогично finger.rules, но ищет признаки использования уязвимостей FTP. Эти правила также вполне можно оставить включенными, даже если у вас нет серверов FTP, так как они будут сигнализировать обо всех нелегальных серверах FTP |
| icmp-info.rules | Эти правила отслеживают сообщения ICMP в вашей сети, порожденные, например, утилитой ping. Они часто являются причиной ложных срабатываний, и можно, наверное, отключить весь набор, если только вы не хотите строго контролировать трафик ICMP в своей сети. Другой класс известного незаконного трафика ICMP icmp.rules перехватывает сканирование портов и сходную активность |
| icmp.rules | Охватывает незаконный или подозрительный трафик ICMP, такой как сканирование портов, и не столь часто, как icmp-info.rules, порождает ложные срабатывания. Однако, возможно, что они будут часто возникать в загруженной сети с множеством работающих диагностических сервисов |
| imap.rules | Правила, относящиеся к использованию в вашей сети протокола IMAP (Internet Message Access Protocol) |
| info.rules | Перехватывает различные сообщения об ошибках от Web, FTP и других серверов |
| local.rules | В этот файл вы добавляете индивидуальные сигнатуры для сети. По умолчанию файл пуст. В конце этой лекции имеется раздел о том, как писать индивидуальные правила Snort |
| misc.rules | Правила, которые не попадают ни в одну из других категорий или не заслуживают собственных разделов. Примером служат старые сигналы, такие как признаки использования уязвимостей сервера Gopher |
| multimedia.rules | Отслеживает использование программного обеспечения типа потокового видео. Если вы разрешаете применять приложения потокового видео или проводить видеоконференции в вашей сети, то при желании можно отключить эти правила |
| mysql.rules | Следит за административным доступом и другими важными файлами в базе данных MySQL. Если вы не используете MySQL, то можете, наверное, отключить эти сигналы. Если база данных MySQL находится в процессе создания, эти правила могут порождать много ложных срабатываний |
| Netbios.rules | Этот класс правил сообщает о различных видах активности NetBIOS в вашей ЛВС. Часть из них соответствует очевидным атакам. Однако, другая часть, например, сигналы о сеансах NULL, может иметь место в нормальных условиях в ЛВС Windows. Необходимо поэкспериментировать с этим разделом, чтобы выявить правила, подходящие для вашей ЛВС |
| nntp.rules | Правила, имеющие отношение к серверу телеконференций. Если вы их не используете, то эти правила, наверное, можно отключить |
| oracle.rules | Правила для сервера баз данных Oracle. Если вы его не используете, можно отключить эти правила |
| other-ids.rules | Эти правила связаны с использованием уязвимостей для пакетов других производителей систем обнаружения вторжений. Весьма вероятно, что в вашей ЛВС нет никаких других сетевых систем обнаружения вторжений, но если таковые есть, оставьте эти правила включенными |

| | |
|--|--|
| p2p.rules | Правила, управляющие использованием программного обеспечения однорангового разделения файлов. Эти правила будут порождать сигналы во время нормального использования данных продуктов, поэтому, если применение этого программного обеспечения допустимо, их необходимо отключить |
| policy.rules | Этот файл содержит различные сигналы, связанные с разрешенной активностью в ЛВС, такой как Go-to-my-pc и другими программами. Необходимо просмотреть эти правила и включить только те, которые соответствуют внутренним политикам |
| pop2.rules pop3.rules | Оба файла относятся к почтовым серверам. Большинство организаций при использовании POP будут применять сервер POP3. Если у вас есть какой-либо из этих двух типов серверов, оставьте эти правила включенными, если нет - отключите |
| porn.rules | Это несколько рудиментарные ловушки для web-серфинга по порнографическим сайтам. Они ни в коей мере не могут заменить хорошую систему фильтрации информационного наполнения, но способны выявить некоторые из наиболее вопиющих нарушений |
| rpc.rules | Этот класс обрабатывает сигналы тревоги, вызванные применением удаленного вызова процедур. Даже если вы считаете, что не используете подобные сервисы, следует учитывать, что они часто выполняются как часть других программ, поэтому важно знать, когда это происходит в вашей ЛВС. Удаленный вызов процедур может допускать удаленное выполнение кода, что часто используется троянскими программами и программами эксплуатации уязвимостей |
| rservices.rules | Отслеживает использование программ различных удаленных сервисов, таких как rlogin и rsh. Вообще говоря, это небезопасные сервисы, но если без них не обойтись, их следует тщательно отслеживать с помощью данного набора правил |
| scan.rules | Предупреждает об использовании программ сканирования портов. Сканирование портов является надежным индикатором ненадлежащей активности. Если вы применяете сканирование портов, то нужно либо отключать Snort в это время, либо отключить определенное правило для машины сканирования |
| shellcode.rules | Этот класс правил ищет пакеты, содержащие ассемблерный код, низкоуровневые команды, называемые также командным кодом. Эти команды являются существенной частью многих программ использования уязвимостей, таких как переполнение буфера. Перехват фрагмента командного кода зачастую служит надежным индикатором развивающейся атаки |
| smtp.rules | Управляет сигналами об использовании почтовых серверов в ЛВС. Этот раздел нуждается в тщательной настройке, так как большая часть нормальной активности почтового сервера будет вызывать сигналы тревоги |
| sql.rules | Правила для различных программ баз данных SQL. Если вы не используете никаких баз данных, эти правила можно отключить, однако неплохо оставить их включенными на тот случай, если имеется база данных SQL, о которой вы не знаете |
| telnet.rules | Отслеживает использование Telnet в сети. Telnet часто применяется на маршрутизаторах или других устройствах с интерфейсом командной строки, которые целесообразно контролировать, даже если вы не используете Telnet на своих серверах |
| tftp.rules | TFTP (trivial FTP) является альтернативным сервером FTP, часто выполняемым на маршрутизаторах. Он может применяться для загрузки новых конфигураций, поэтому стоит за ним следить |
| virus.rules | Содержит сигнатуры некоторых распространенных червей и вирусов. Этот список не является полным, поддерживается нерегулярно и не может служить заменой антивирусного программного обеспечения, но способен перехватывать некоторых сетевых "червей" |
| web-attacks.rules web-cgi.rules web-client.rules web-coldfusion.rules web- | Все эти классы относятся к различным видам подозрительной web-активности. Некоторые из них универсальны, например, класс web-attacks. Другие, такие как web-iis и web-frontpage, специфичны для определенных серверных платформ web. Даже если вы полагаете, что у вас в сети нет web-серверов Microsoft и PHP не используется, стоит |

| | |
|---|--|
| frontpage.rules web-iis.rules web-php.rules | оставить все правила включенными, чтобы обнаруживать в ЛВС любую активность такого рода, о которой вы можете и не знать. Необходимо тщательно настроить эти правила, особенно если ваши web-серверы активно развиваются. |
| X11.rules | Отслеживает применение графической среды X11 в вашей сети. |

Запуск Snort в качестве службы

Если вы собираетесь выполнять Snort на сервере, предназначенном для круглосуточной работы, то желательно запускать Snort при загрузке операционной системы, чтобы после временного выключения машины она перезагружала Snort, и система обнаружения вторжений продолжала защищать ваши ЛВС. Один из способов сделать это - включить в число стартовых процедур небольшой командный файл, запускающий Snort с параметрами командной строки. В Linux можно поместить строку для запуска Snort в файл rc.local в каталоге /etc/rc.d. Пример:

```
snort -h 192.168.1.0/24 -c /etc/snort/snort.conf &
```

Знак & (амперсанд) в конце означает запуск Snort в фоновом режиме. Можно также запускать Snort как службу, воспользовавшись командой `service snort start`.

Snort Webmin Interface: Графический интерфейс для Snort

Автор/основной контакт: Mike Baptiste/MSB Networks

Web-сайт: msnnetworks.net/snort/

Платформы: Большинство Linux

Лицензия: GPL

Рассмотренная версия: 1.1

Настраивать Snort из командной строки - скучновато. Хотя Snort не имеет пока собственного графического интерфейса, существует модуль для популярного средства управления Webmin на базе Web. Он позволяет произвести всю настройку и конфигурирование из любого web-навигатора. Свойствами этой системы являются:

- Доступ к конфигурационным файлам Snort на основе форм.
- Различные уровни доступа пользователей, что позволяет задать различные права доступа для различных пользователей.
- Возможность включать и выключать наборы правил щелчком мыши.
- Индикатор статуса для всех правил и наборов правил.
- Встроенные ссылки к внешней базе данных, такой как archNIDS, CVE и Bugtraq.
- Запись изменений в журналах.
- Средства поддержки различных языков.
- Поддержка запуска Snort в качестве службы с помощью файлов rc.d.
- Защищенное удаленное администрирование через SSL (если включено).

В [лекции 3](#) рассмотрена загрузка Webmin для администрирования межсетевого экрана. Этот же дополнительный модуль можно применять и для конфигурирования Snort. Обратитесь к [лекции 3](#), если вы еще не загрузили Webmin.

Для Snort требуется версия Webmin 0.87 или выше. Можно использовать файл Snort Webmin с компакт-диска, загрузить модуль Snort с помощью интерфейса Webmin или сначала загрузить файл, а потом установить его локально, взяв его по адресу: <http://www.msbnetworks.com/snort/download/snort-1.1.wbm>

Чтобы загрузить модуль Snort через интерфейс Webmin, выполните следующие действия.

1. Перейдите на основную страницу Webmin. Войдите с помощью имени и пароля, заданных во время установки Webmin.
2. Щелкните мышью на вкладке конфигурирования Webmin. Щелкните на Modules и выберите либо Local file, либо FTP Url, в зависимости от того, загрузили вы его уже на свою машину или хотите, чтобы Webmin взял его с web-сайта.
3. Щелкните мышью на Install module, в результате будет установлен файл модуля Snort. Модуль Snort появится как иконка на основной странице Webmin. Щелкните мышью на этой иконке, чтобы отобразить интерфейс Webmin Snort ([рис. 7.2](#)).

Snort IDS

Global Snort Configuration

Network Settings PreProcessors Alerts & Logging Edit Config File

Goto ACID

Rulesets

✓ = Enabled ✗ = Disabled

| Rule Set | Status | Action | Rule Set | Status | Action | Rule Set | Status | Action |
|---------------------------|--------|-------------------------|---------------------------|--------|-------------------------|--------------------------------|--------|-------------------------|
| backdoor | ✓ | Disable | local | ✓ | Disable | sql | ✓ | Disable |
| ddos | ✓ | Disable | misc | ✓ | Disable | telnet | ✓ | Disable |
| dns | ✓ | Disable | netbios | ✓ | Disable | virus | ✗ | Enable |
| dos | ✓ | Disable | policy | ✓ | Disable | web-cgi | ✓ | Disable |
| exploit | ✓ | Disable | rpc | ✓ | Disable | web-coldfusion | ✓ | Disable |
| finger | ✓ | Disable | rservices | ✓ | Disable | web-frontpage | ✓ | Disable |
| ftp | ✓ | Disable | scan | ✓ | Disable | web-fis | ✓ | Disable |
| icmp | ✗ | Enable | shellcode | ✓ | Disable | web-misc | ✗ | Enable |
| icmp-info | ✗ | Enable | smtp | ✓ | Disable | x11 | ✓ | Disable |
| info | ✓ | Disable | | | | | | |

Snort does not appear to be running
(If you know Snort is running, check the PID file setting in the module configuration)



Рис. 7.2. Модуль Webmin Snort

Открыв страницу Snort, вверху экрана вы увидите все основные разделы конфигурационного файла, такие как настройки сети, параметры препроцессора и опции входа. Щелкнув мышью на любом из параметров конфигурации, можно заполнить форму своей информацией, и Webmin произведет изменения в соответствующих конфигурационных файлах Snort.

Все наборы правил перечислены ниже, и можно видеть, какие из них включены или выключены. Галочка указывает, что набор правил включен, а значком X отмечены выключенные наборы. Для того чтобы отключить целый набор правил, достаточно сделать двойной щелчок мышью в поле Action. Если вы хотите просмотреть набор правил и модифицировать отдельное правило, щелкните мышью на голубом подчеркнутом тексте, и вы попадете на страницу Edit Ruleset (Редактирование набора правил) (рис. 7.3). На ней можно видеть все активные правила набора. С правилами можно выполнять действия - выключение, включение или удаление из набора правил. Если в описании сигнала тревоги имеются ссылки на внешние базы данных, такие как номера в словаре уязвимостей CVE (Common Vulnerability or Exploit), то можно получить дополнительную информацию о действии сигнала, щелкнув мышью на гиперссылке. Использование подобного интерфейса может существенно облегчить настройку правил.

Edit Ruleset - Mozilla (Build ID: 2001080104)

File Edit View Search Go Bookmarks Tasks Help Debug QA

Webmin Servers
Webmin Index
Module Config

Search docs..

Edit Ruleset

Current Rules in /etc/snort/rpc.rules

| Rule | Signature | Status | Action |
|------|--|--------|---|
| 1 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request rstad"; content:" 01 86 A0 00 00 "; reference:arachnids_10; classtype:attempted-recon; sid:583; rev:1;) | ✓ | Disable Edit Delete |
| 2 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request sadmind"; content:" 01 87 88 00 00 "; offset:40; depth:8; reference:arachnids_20; classtype:attempted-recon; sid:585; rev:1;) | ✓ | Disable Edit Delete |
| 3 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request selection_svc"; content:" 01 86 AF 00 00 "; offset:40; depth:8; reference:arachnids_25; classtype:attempted-recon; sid:586; rev:1;) | ✗ | Enable Edit Delete |
| 4 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request status"; content:" 01 86 B8 00 00 "; offset:40; depth:8; reference:arachnids_15; classtype:attempted-recon; sid:587; rev:1;) | ✓ | Disable Edit Delete |
| 5 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request tttdserv"; content:" 01 86 F3 00 00 "; offset:40; depth:8; reference:arachnids_24; classtype:attempted-recon; sid:588; rev:1;) | ✓ | Disable Edit Delete |
| 6 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request yppasswd"; content:" 01 86 A9 00 00 "; offset:40; depth:8; reference:arachnids_14; classtype:attempted-recon; sid:589; rev:1;) | ✓ | Disable Edit Delete |
| 7 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request ypsserv"; content:" 01 86 A4 00 00 "; offset:40; depth:8; reference:arachnids_12; classtype:attempted-recon; sid:590; rev:2;) | ✓ | Disable Edit Delete |
| 8 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 111 (msg:"RPC portmap request yppupdated"; content:" 01 86 DC 00 00 "; offset:40; depth:8; reference:arachnids_125; classtype:attempted-recon; sid:591; rev:2;) | ✗ | Enable Edit Delete |
| 9 | alert udp \$EXTERNAL_NET any -> \$HOME_NET 32770: (msg:"RPC rstatd query"; content:" 00 00 00 00 00 00 02 | ✓ | Disable Edit |



Рис. 7.3. Страница редактирования наборов правил Webmin Snort

Модуль Webmin Snort позволяет также управлять доступом пользователей к настройкам (рис. 7.4). На странице пользователей Webmin можно задать ряд параметров для каждого пользователя (при условии, что вы являетесь администратором Webmin). Можно предоставить определенным пользователям доступ для редактирования правил, но не для редактирования конфигурационных файлов. Можно управлять тем, к каким конфигурационным файлам они будут иметь доступ - или позволить им только просматривать файлы без редактирования или отключения. Таким образом, модуль Webmin Snort предоставляет весьма детальное управление доступом, что позволяет вам делегировать ежедневные обязанности по настройке менее квалифицированному техническому специалисту, оставляя за собой управление конфигурацией и изменениями.

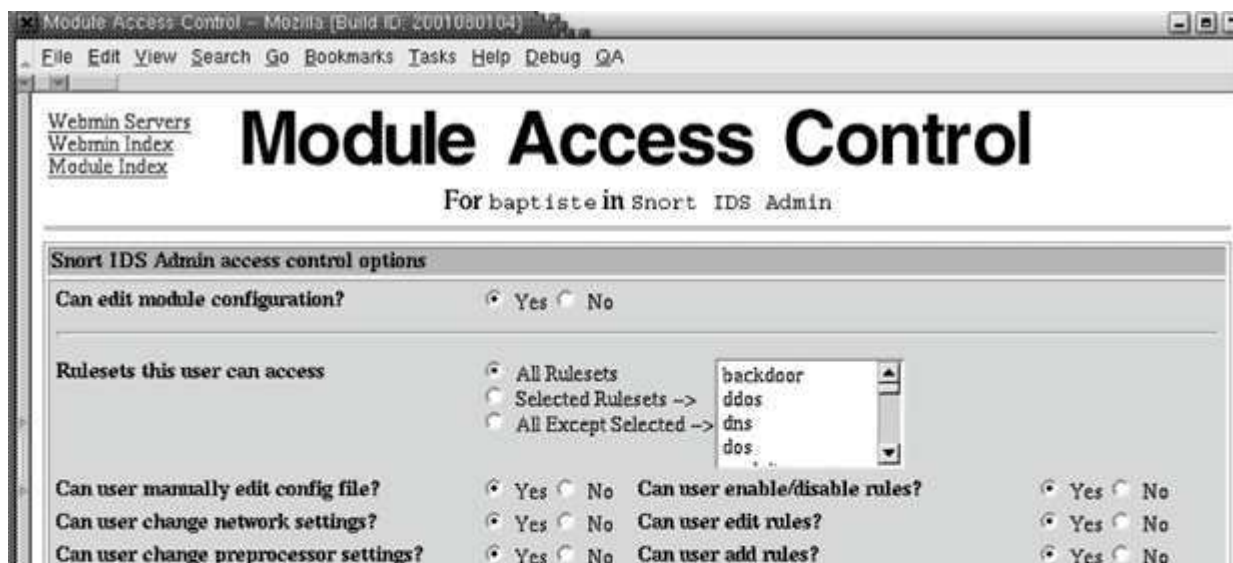




Рис. 7.4. Управление доступом для модуля Webmin Snort

Snort for Windows: система обнаружения вторжений для Windows с открытыми исходными текстами

Snort for Windows

Автор/основной контакт: Martin Roesch

Перенос в Windows: Michael Davis, Chris Reid

Web-сайт: <http://www.snort.org/>

Платформы: Windows 2000, XP

Лицензия: GPL

Рассмотренная версия: 2.0.0

Другие ресурсы: См. список в разделе "Snort: система обнаружения вторжений для UNIX с открытыми исходными текстами" выше в этой лекции.

К счастью для тех, кто не может или не хочет устанавливать версию Snort для UNIX, имеется полностью поддерживаемая версия для платформы Windows. Хотя отдача от каждого вложенного в аппаратуру доллара для UNIX-версии будет выше, Windows-версия не является просто побочным проектом - фактически, она разработана основной группой Snort и в достаточной степени синхронизирована с UNIX-версией. Она позволяет воспользоваться преимуществами простоты установки, а также другими достоинствами Windows 2000 и XP, такими как встроенная поддержка IPSec. Приятно видеть проект с открытыми исходными текстами, участники которого понимают, что имеется много компаний, использующих только Windows, которые с удовольствием применят возможности этой отличной системы обнаружения вторжений с открытыми исходными текстами.

Требования для использования Snort в Windows

Snort для Windows требует Windows 2000 или XP; на NT, 98 или 95 выполнение невозможно. Необходимы также установленные библиотеки WinPcap. Если они были установлены для программ, описанных ранее в этой книге, таких как Ethereal или WinDump, тогда все готово. В противном случае можно взять их по адресу

netgroup-serv.polito.it/winpcap

Вам может также потребоваться база данных MySQL, если вы планируете импортировать результаты в базу данных. Конкретная конфигурация MySQL для этой цели описана в [лекции 8](#).

Для того чтобы Snort для Windows демонстрировал ту же производительность, что и UNIX-версия, понадобится более мощная аппаратура,. Машина с процессором 700 МГц - это минимум, но лучше использовать процессор с частотой 1 ГГц и выше. Необходимо также убедиться, что сервер Windows хорошо защищен, на нем выполняется минимум сервисов и удалены программы, активно использующие процессор, такие как IIS. Воспользуйтесь окном Services из Administrative tools Панели управления, чтобы проверить, не запускается ли что-нибудь лишнее.

Установка Snort для Windows

Чтобы установить Snort для Windows, возьмите бинарный файл с прилагаемого к книге компакт-диска или с сайта <http://www.snort.org/>. Сделайте на нем двойной щелчок мышью, и он автоматически установится. Вас спросят, нужна ли вам определенная база данных или дополнительные модули, такие как модуль гибкого реагирования.

Настройка Snort для Windows

Процесс настройки версии Snort для Windows весьма схож с настройкой для UNIX. Все файлы конфигурации и правил находятся в тех же относительных подкаталогах. Войдите в файл `snort.conf` в подкаталоге `etc` установки Snort. Измените и отредактируйте его, как предложено в разделе о UNIX-версии. Затем перейдите в файлы правил и произведите изменения там. После этого все будет готово к запуску Snort. Обратитесь к разделу "Запуск Snort" для UNIX, чтобы получить дополнительную информацию о применении Snort для Windows, так как все команды такие же. Дополнительные настройки и рекомендации по размещению - те же, что и для исходной UNIX-версии.



Уголок кодировщиков Флэми Теха

Написание индивидуальных правил Snort

Хотя стандартные наборы правил, с которыми поставляется Snort, обеспечивают достаточную защиту от атак с известными сигнатурами, можно создавать некоторые индивидуальные правила, специфичные для вашей сети, чтобы получить от системы обнаружения вторжений максимальную отдачу. Вы можете написать правила для:

- отслеживания входящего и исходящего доступа для определенных серверов;
- поиска определенных типов или имен файлов, специфичных для вашей организации;
- наблюдения за определенными типами трафика, чужеродными для вашей сети;

Научиться писать правила для Snort несложно; это позволит быстро наращивать функциональность программы даже при отсутствии обширных программистских знаний. Как вы видели, все правила Snort являются просто текстовыми инструкциями в одном из файлов правил.

Если нужно, чтобы Snort обнаруживал некое особое поведение, которое в вашей сети будет считаться подозрительным, можно быстро закодировать правило и тут же протестировать это поведение. Правила Snort по сути представляют собой одиночные текстовые строки, начинающиеся с действия (как правило, `alert`), за которым следует несколько аргументов. В новейшей версии (2.0 и выше) можно добавить несколько строк, просто помещая \ (обратную косую черту) в конце каждой строки, кроме последней. В более сложных случаях можно также вызывать другие программы, используя инструкцию включения. Но в своей базовой форме правило Snort имеет две части: заголовок и параметры. Ниже представлен пример правила.

```
alert tcp any any 192.168.0.0/24 \ (content:"|00 05 A4 6F 2E|";msg: "Test Alert")
```

Заголовок является частью перед первой скобкой. Данная инструкция содержит действие (в нашем случае - `alert`), протокол, а также адреса и порты отправителя и получателя. Действие будет выполняться, если заданное правилом условие истинно. В данном случае будет порождаться сигнал тревоги (`alert`). Другими вариантами действий служат `Log`, `Pass`, `Activate` и `Dynamic`.

| | |
|----------|--|
| Log | Просто протоколирует пакеты |
| Pass | Игнорирует пакет. Это подразумеваемое действие для пакетов, не соответствующих правилу. |
| Activate | Сигнал тревоги, затем активация динамического правила. |
| Dynamic | Остается пассивным, пока не активируется динамическим правилом, затем действует как <code>log</code> . |

Протоколами могут быть `tcp`, `udp`, `icmp` или `ip`, что означает любой IP-протокол. (В будущем могут поддерживаться протоколы не на основе IP, такие как `IPX`). Исходный и целевой порты самоочевидны. Исходный адрес идет первым и задается в стандартной нотации с косой чертой для IP-диапазона. Можно также перечислить несколько индивидуальных адресов и сетей, разделяя их запятой без пробелов и заключая в квадратные скобки, например: `alert tcp any < [192.168.1.1,192.168.1.5,192.168.1.10] 80 \ (content: "|00 05 A4 6F 2E|";msg: "Test Alert");`

Эта инструкция ориентирована на трафик, приходящий из любых адресов, направляющийся на машины с адресами 192.168.1.1, 192.168.1.5 и 192.168.1.10 в порт 80. При условии, что это ваши web-серверы, приведенное правило будет искать идущий туда трафик, который содержит указанные шестнадцатеричные данные в разделе содержимого.

Второй частью правила Snort служат опции, задающие дополнительные детали выявляемого трафика. Можно искать по набору полей в заголовке TCP/IP (см. описания в [лекции 6](#)) или по полезной нагрузке пакета. За каждой опцией должны следовать кавычки и разыскиваемое значение. Можно добавить несколько опций, разделяя их с помощью точки с запятой. Ниже приведены допустимые опции.

| | |
|--------------|---|
| msg | Предоставляет текстовое описание сигнала тревоги |
| logto | Записывает пакет в заданный пользователем файл вместо стандартного выходного файла |
| ttl | Проверяет значение поля TTL в заголовке IP |
| tos | Проверяет значение поля TOS в заголовке IP |
| id | Сравнивает значение поля идентификатора фрагмента в заголовке IP с указанной величиной |
| ipoption | Ищет поля опций IP с определенными кодами |
| fragbits | Проверяет биты фрагментации в заголовке IP |
| dsize | Сравнивает размер полезной нагрузки пакета с указанным значением |
| flags | Проверяет флаги TCP на соответствие определенным значениям |
| seq | Сравнивает поле порядкового номера TCP с определенным значением |
| ack | Проверяет поле подтверждения TCP на соответствие определенному значению |
| itype | Проверяет поле типа ICMP на соответствие определенному значению |
| icode | Проверяет поле кода ICMP на соответствие определенному значению |
| icmp_id | Проверяет поле ECHO ID ICMP на соответствие определенному значению. |
| icmp_seq | Проверяет порядковый номер ECHO ICMP на соответствие определенному значению |
| content | Ищет определенный шаблон в полезной нагрузке пакета |
| content-list | Ищет определенный набор шаблонов в полезной нагрузке пакета |
| offset | Модификатор для опции содержимого. Задаёт смещение для начала сопоставления с образцом |
| depth | Модификатор для опции содержимого. Устанавливает максимальную глубину поиска при сопоставлении с образцом |
| nocase | Сравнивает предыдущую цепочку содержимого без учета регистра символов |
| session | Вывод информации прикладного уровня для данного сеанса |
| rpc | Следит за сервисами RPC для выявления определенных вызовов приложений/процедур |
| resp | Активный ответ. Закрывает соединение (например, разрывая его) |
| react | Активный ответ. Отвечает запрограммированным поведением (например, блокированием определенных Web-сайтов) |
| reference | Идентификаторы ссылок на внешние атаки |
| sid | Идентификатор правила Snort |
| rev | Номер версии правила |
| classtype | Классификационный идентификатор правила |
| priority | Идентификатор уровня серьезности правила |
| uricontent | Сопоставление с образцом в части URI пакета |
| tag | Дополнительные действия по протоколированию для правил |
| ip_proto | Значение протокола в заголовке IP |

| | |
|-----------|---|
| sameip | Определяет, не равны ли исходный и целевой IP-адреса |
| stateless | Применимо независимо от состояния потока |
| regex | Сопоставление с образцом с применением метасимволов |
| byte_test | Числовое сравнение |
| distance | Заставляет при относительном сопоставлении с образцом пропустить в пакете определенное число байт |
| byte_test | Числовое сопоставление с образцом |
| byte_jump | Числовое сопоставление с образцом и корректировка смещения |

Более подробную информацию о каждой из опций правил можно получить в оперативной справке. Ниже представлены несколько примеров применения этих опций для создания индивидуальных правил Snort

Пример 1 индивидуального правила

Предположим, имеется набор бухгалтерских серверов, доступ к которым может осуществляться только из внутренней сети. Можно написать правило Snort, реагирующее на трафик, идущий с любого не принадлежащего вашей сети IP-адреса и направленный на эти серверы. Пусть бухгалтерские серверы имеют IP-адреса 192.168.1.10, 192.168.1.11 и 192.168.1.12, а ваша внутренняя сеть - адреса 192.168.2.0/24. Тогда правило будет выглядеть примерно так:

```
alert tcp !192.168.1.0/24 any \
< [192.168.1.10,192.168.1.11,192.168.1.12] any \
(msg: "Попытка внешнего доступа к бухгалтерскому серверу";)
```

Знак операции ! (восклицательный знак) обозначает логическое отрицание. Смысл правила в том, чтобы выдать сигнал тревоги при обнаружении TCP-трафика, идущего не из сети 192.168.1.0/24 и направленного на указанные серверы. Не задается никаких опций, кроме msg - метки, появляющейся в журналах сигналов. Дело в том, что нас интересует любой трафик на любой порт. Будет отмечено любое обращение к бухгалтерским серверам, исходящее из внешнего мира, так как предполагается, что любой внешний трафик к этим серверам должен считаться вредоносным.

Пример 2 индивидуального правила

Опираясь на сценарий из примера 1, предположим, что следует разрешить некоторый внешний доступ к бухгалтерским серверам, но, тем не менее, гарантировать, что никто не скопирует определенные файлы. Предположим, что имеется файл с именем payroll.xls, который содержит все данные о зарплате (совершенно секретный файл, как внутри, так и вне организации). Можно написать правило, которое проследит за любым трафиком, внутренним или внешним, направленным на эти серверы и содержащим имя секретного файла. Это можно сделать с помощью опции content, осуществляющей поиск в реальном содержимом пакетов. Правило будет выглядеть примерно так:

```
alert tcp ![192.168.1.10,192.168.1.11,192.168.1.12] any <
[192.168.1.10,192.168.1.11,192.168.1.12] any
(content: "payroll.xls";msg: "Попытка доступа к файлу зарплат")
```

Отметим, что знак операции ! снова означает, что нас интересует трафик, направленный на бухгалтерские серверы из любого места, кроме этих серверов. Тем самым устраняется сигнализация о межсерверном трафике. Отметим также, что символ \ позволяет писать многострочные правила, а опция content - осуществлять поиск текста payroll.xls в пакетах. В результате серверные машины могут иметь доступ в Интернет, но если этот конкретный файл будет когда-либо выгружаться с них, вы будете об этом оповещены.

С помощью других опций можно писать правила для выявления трафика практически любого вида. Если ваши правила могут представлять интерес для других организаций, стоит послать их разработчикам Snort для вставки в официальный набор распространяемых правил. Если вы решите это сделать, постарайтесь использовать все средства документирования, такие как msg, sid, rev, classtype и priority. Также тщательно протестируйте

свои правила, чтобы гарантировать, что они действительно охватывают все виды активности, которую вы пытаетесь поймать, и не дают ложных срабатываний.

Хостовые системы обнаружения вторжений

Мы уделили много внимания сетевым системам обнаружения вторжений. Однако имеются и другие методы выявления попыток вторжения. Один из них - искать признаки вторжения в самой системе. Если машина скомпрометирована, то зачастую оказываются измененными определенные системные файлы. Например, может быть модифицирован файл паролей, добавлены пользователи, изменены системные конфигурационные файлы или режимы доступа к файлам. Обычно эти системные файлы не должны существенно меняться. Просматривая внесенные в них изменения, можно обнаружить вторжение или другую нетипичную активность.

Этот метод обнаружения вторжений может быть значительно более точным, генерирующим меньше ложных срабатываний, так как тревога поднимается только тогда, когда система на самом деле подверглась определенному воздействию. Правда, данный подход несколько сложнее проводить в жизнь, так как требуется загрузить программное обеспечение на все защищаемые системы, но поддержание безопасности критически важных систем с помощью как хостовых, так и сетевых средств обнаружения вторжений стоит затраченных сил и времени.

Преимущества хостовых методов обнаружения вторжений

- Меньшее число ложных срабатываний.
- Отслеживается активность, а не сигнатуры, поэтому не требуется постоянное обновление сигнатур.
- Менее подвержены обману.
- Требуют меньше обслуживания и настройки.

Недостатки хостовых методов обнаружения вторжений

- Необходимость загрузки и управления программным обеспечением на каждой защищаемой машине.
- Сигналы тревоги поступают после успешной атаки; сетевые системы обнаружения вторжений обеспечивают иногда более раннее предупреждение.

Tripwire: Программа проверки целостности файлов

Tripwire

Автор/основной контакт: Dr. Eugene Spafford и Gene Kim

Web-сайт: <http://www.tripwire.org/>

Платформы: Большинство UNIX

Лицензия: GPL

Рассматриваемая версия: V 2.3.47

Tripwire может служить еще одним прекрасным примером программного обеспечения с открытыми исходными текстами, совершившего переход на коммерческую платформу. Первоначально Tripwire была в чистом виде программой с открытыми исходными текстами. Со временем основатели организовали компанию для продажи и поддержки Tripwire на коммерческой основе, однако исходный базовый код они выпустили под лицензией GPL, чтобы разработка могла продолжаться в сообществе открытого ПО. Текущая открытая версия 2.3 была получена путем обновления версии 2.2.1, выпущенной в октябре 2000 года.

Имеются существенные различия между коммерческой и открытой версиями. Самые значительные из них - поддержка коммерческой версией большего числа платформ и ее закрытость. Версия с открытыми исходными текстами в настоящее время доступна только для Linux, в то время как коммерческая - на

нескольких платформах, включая Windows. Еще одно различие состоит в том, что коммерческая версия поставляется с программой, называемой twagent, которая служит для управления несколькими установками Tripwire. Коммерческая версия имеет также прекрасный графический интерфейс для управления базами данных и конфигурациями.

Обе версии Tripwire работают, создавая базу данных эталонных атрибутов важных файлов, которые предполагается отслеживать, поэтому в любое время можно сравнить текущие атрибуты с эталонными, чтобы узнать, изменилось ли что-нибудь. Это хорошо подходит для отслеживания системных бинарных файлов. Один из любимых приемов хакеров при проникновении в систему состоит в замене ключевых бинарных файлов собственными троянскими версиями. Таким образом, когда вы выполняете команду типа `ls` или `ps`, вы не увидите нелегальные файлы или процессы. Tripwire можно также применять во время судебного разбирательства, чтобы определить, где был взломщик; это напоминает исследование цифровых отпечатков.

Установка Tripwire

1. Чтобы установить Tripwire, возьмите файлы с компакт-диска или загрузите RPM или tar с web-сайта Tripwire для компиляции исходных текстов.

Для некоторых дистрибутивов существуют также доступные RPM (RPM для Mandrake и RedHat помещены на компакт-диск). Просто щелкните мышью на файле RPM, чтобы установить программу. Если у вас нет RPM для вашей операционной системы, то можно загрузить файл .tar с исходными текстами (или "tarball", как его часто называют) для последующей компиляции. Распакуйте tar-файл и перейдите в каталог src.

2. В каталоге src наберите

```
make all
```

Программа Tripwire будет скомпилирована и подготовлена для дальнейшего конфигурирования.

3. Откройте файл install.cfg и проверьте, что все подразумеваемые значения подходят для вашей системы. Этот файл управляет местом установки файлов программы и другими переменными системного уровня. Большинство подразумеваемых значений годятся для большинства систем. Проверьте, что ваш почтовый клиент задан правильно.
4. После установки переменных окружения перейдите в каталог /etc/tripwire и наберите `twinnstall.sh`.

Будет выдано лицензионное соглашение. Необходимо ввести `accept`, и затем командный файл установки скопирует файлы в предназначенные для них места и предложит ввести пароли для сайта и локального входа. Эти пароли будут использоваться для разблокирования базы данных Tripwire, поэтому они очень важны.

Выбор сильных паролей и сохранение их в надежном месте - важный момент, так как с их помощью будет проводиться шифрование базы данных и конфигурационных файлов. Предохраняйте их от компрометации. Если вы их потеряете или забудете, то в критическую минуту не сможете использовать Tripwire.

На этом процесс установки Tripwire завершается.

Конфигурирование Tripwire

Заключительным шагом, предшествующим запуску Tripwire, служит задание вашей политики. Файл политики очень важен для работы Tripwire: в нем специфицируются отслеживаемые файлы и уровень детализации. Основной файл политики, twpol.txt, находится в главном каталоге Tripwire. Строго говоря, это не сам файл политики, а копия зашифрованной версии, которую в действительности использует программа. Для большей безопасности необходимо сделать копию и удалить незашифрованную версию twpol.txt, после того как вы определите и протестируете свою политику.

В начале файла политики задано несколько системных переменных, а затем следует список различных файлов и каталогов с директивами политики для них. Эти директивы представлены либо кодовыми буквами, либо именами переменных. Они называются масками свойств и задают свойства, отслеживаемые Tripwire. В [табл. 7.4](#) перечислены элементы, которые могут отслеживаться для каждого файла, и их кодовые буквы.

Таблица 7.4. Маски свойств Tripwire

| Буква кода | Отслеживаемый атрибут |
|------------|--|
| a | Время последнего доступа |
| b | Отведенные блоки |
| c | Время создания/изменения |
| d | Идентификатор устройства, на котором располагается описатель файла |
| g | Идентификатор владеющей группы файла |
| i | Номер описателя файла |
| l | Разрешен ли рост файла |
| m | Метка времени изменения |
| n | Значение счетчика ссылок в описателе файла |
| p | Режим доступа к файлу |
| s | Размер файла |
| t | Тип файла |
| u | Идентификатор пользователя владельца файла |
| C | Хэш-код CRC32 |
| H | Хэш-код NaVal |
| M | Хэш-код MD5 |
| S | Хэш-код SHA/SHS |

Политики Tripwire действуют по принципу флагов игнорирования. Можно сконфигурировать Tripwire для отслеживания или игнорирования различных свойств файлов. Знак + (плюс) используется для отслеживания свойств, а знак - (минус) - для их игнорирования. Формат инструкций в файле политики таков:

```
имя_файла/каталога -> маска_свойств;
```

Например, следующая строка в файле политики

```
/etc/secretfile.txt -> +amcpstu;
```

предписывает Tripwire извещать вас всякий раз, когда у файла secretfile.txt в каталоге /etc изменяются время последнего доступа, время создания или модификации, режим доступа, владелец, размер или тип файла.

Существует также несколько предопределенных масок свойств. В [табл. 7.5](#) перечислены эти стандартные маски и их действие.

Таблица 7.5. Стандартные маски свойств

| Маска свойств | Действие |
|---------------|----------------------|
| \$ReadOnly | +pinugtsdbmCM-rlaSH |
| \$Dynamic | +pinugtd-srlbamcCMSH |
| \$Growing | +pinugtdl-srbamcCMSH |
| \$Device | +pugsdr-intlbamcCMSH |

| | |
|--------------|----------------------|
| \$IgnoreAll | -pinugtsdrlbamcCMSh |
| \$IgnoreNone | +pinugtsdrrlbamcCMSh |

Предопределенные переменные соответствуют поведению различных наборов файлов. Например, можно использовать `$ReadOnly` для ключевых конфигурационных файлов, так как время доступа будет изменяться, когда программы их используют, но вы не желаете, чтобы изменялись размер или содержимое. Можно использовать `$Growing` для файлов журналов, так как они будут (во всяком случае, должны) постоянно увеличиваться в размере.

В конфигурационном файле политики определяется также несколько переменных, которые являются комбинациями упомянутых выше предопределенных переменных с некоторыми добавлениями или исключениями и позволяют быстро задавать политики для различных классов файлов. Их можно немного изменить, если вы хотите проигнорировать или исследовать дополнительные моменты. Эти переменные из файла политики показаны на [листинге 7.3](#).

```
SEC_CRIT = $(IgnoreNone)-SHa ; # Критичные файлы, которые
                                # не могут изменяться
SEC_SUID = $(IgnoreNone)-SHa ; # Бинарные файлы с установленными
                                # флагами SUID или SGID
SEC_BIN = $(ReadOnly) ;       # Бинарные файлы, которые не
                                # должны изменяться
SEC_CONFIG = $(Dynamic) ;     # Конфигурационные файлы,
                                # которые редко изменяются,
                                # но часто используются
SEC_LOG = $(Growing) ;        # Файлы, которые увеличиваются,
                                # но никогда не должны менять
                                # владельца
SEC_INVARIANT = +tpug ;        # Каталоги, у которых никогда
                                # не должны изменяться режим
                                # доступа или владелец
SIG_LOW = 33 ;                 # Некритичные файлы, которые
                                # имеют минимальное влияние
                                # на безопасность
SIG_MED = 66 ;                 # Некритичные файлы, которые
                                # имеют значительное влияние
                                # на безопасность
SIG_HI = 100 ;                 # Критичные файлы, которые
                                # являются существенными точками уязвимости
```

Листинг 7.3. Переменные масок свойств

Ниже масок свойств заданы политики для различных файлов и каталогов в системе. Можно начать с подразумеваемого файла политики и посмотреть, как он работает для вашей системы. Найдите время для внимательного изучения файла, чтобы знать, какие файлы отслеживаются. Когда вы будете удовлетворены, сохраните файл и выйдите. Все готово к запуску Tripwire.

Инициализация эталонной базы данных

Первым шагом при выполнении Tripwire является формирование эталонной базы данных. Создается начальный список сигнатур, согласно которым будут применяться политики. Помните, что выполнение этого шага после того, как в системе появились подозрительные файлы, ни к чему хорошему не приведет; необходимо создать эталонную базу данных до того, как появятся какие-либо проблемы с безопасностью, лучше всего сразу после установки и конфигурирования системы. Чтобы создать начальный файл базы данных, используйте команду

```
tripwire -m i -v
```

Ключ `-m` определяет режим выполнения, в данном случае `i` означает инициализацию. Ключ `-v` задает расширенный вывод, чтобы можно было посмотреть, что происходит. Tripwire определяет все файлы, заданные в файле политики, создает базу данных в каталоге `./database` и шифрует ее с помощью пароля сайта.

Чтобы по-настоящему обезопасить Tripwire, необходимо сделать копию эталонной базы данных на некотором защищенном несетевом носителе информации - флоппи-диске, компакт-диске или магнитной ленте. Если вы будете хранить копию в сети, всегда будет существовать возможность ее изменения, хотя Tripwire и имеет от этого некоторые средства защиты.

Проверка целостности файлов

Это основной режим выполнения программы Tripwire после ввода в эксплуатацию. В этом режиме текущие атрибуты определенных файлов сравниваются с атрибутами в базе данных Tripwire. Формат запуска в этом режиме таков:

```
tripwire -m c маршрутное_имя
```

Задается маршрутное имя файла или каталогов, которые вы хотите контролировать. Эта команда будет проверять атрибуты файла согласно спецификациям файла политики и выдавать отчет обо всех изменениях.

Обновление базы данных

По мере уточнения политики и при существенных изменениях системы необходимо обновлять базу данных, чтобы она отражала реальное состояние файлов. Это важно, так как в базу данных не только будут добавляться новые файлы и каталоги, но и исключаться ложные срабатывания. Не обновляйте базу данных, если есть вероятность, что ваша система была скомпрометирована. В этом случае сигнатуры станут недействительными, а база данных Tripwire - бесполезной. Можно обновить каталоги выборочно; в конце концов, некоторые вещи, например системные бинарные файлы, будут изменяться редко. База данных Tripwire обновляется с помощью следующей команды:

```
tripwire -m u -r маршрутное_имя_отчета
```

Здесь `маршрутное_имя_отчета` соответствует самому свежему файлу отчета. Выполнение этой команды покажет все произошедшие изменения, а также правила, их обнаружившие. Рядом с файлами, в которых обнаружили изменения, будет присутствовать знак `x` в квадратных скобках. Если оставить `x` на месте, то Tripwire обновит сигнатуру для этого файла, когда вы закончите работу с отчетом. Если удалить `x`, то Tripwire будет предполагать, что исходная сигнатура правильна, и не будет ее обновлять. При выходе Tripwire внесет изменения. Можно задать ключ `-c` в командной строке, чтобы пропустить предварительный просмотр отчета. В этом случае Tripwire просто учтет обнаруженные изменения.

Обновление файла политики

Со временем вы поймете, какие правила порождают ложные сигналы, и пожелаете удалить их, изменить в них маски свойств или уточнить маски для некоторых файлов. Внесите изменения в файл политики Tripwire, сохраните его, а затем выполните следующую команду, чтобы Tripwire обновил файл политики:

```
tripwire -m p текстовый_файл_политики
```

указав новый файл политики. Прежде чем обновить политику, Tripwire запросит у вас пароли для сайта и локального входа. Когда политики Tripwire будут настроены в достаточной степени, можно будет создать задание `cron` и выполнять его ежедневно (или так часто, как захотите), чтобы проверять файловые системы в поисках следов взлома.

Инструменты безопасности с открытым исходным кодом

8. Лекция: Средства анализа и управления: версия для печати и PDA

Рассматривавшиеся до сих пор средства предоставляют много полезной информации, которая может помочь в решении проблем сетевой безопасности. Но эти замечательные программы сами создают проблему - иногда они порождают слишком много информации. Одно-единственное сканирование большой сети с помощью Nessus способно породить отчет длиной в сотни страниц. Активный сенсор Snort может выдавать тысячи сигналов в день. Даже скромный межсетевой экран в состоянии посылать записи в журнал каждый час. Отслеживание всей этой защитной информации способно занять весь рабочий день. На самом деле, в крупных организациях нередко требуется небольшая группа сотрудников исключительно для отслеживания и анализа данных системы безопасности.

С многочисленными данными, которые порождают защитные средства, очень легко перейти от неинформированности, когда ситуация с безопасностью покрыта "мраком неизвестности" (что очень плохо), к перегруженности информацией (что, возможно, еще хуже). У системного администратора может возникнуть состояние "аналитического паралича" - ощущение беспомощности, когда просто не ясно, с чего начать. Перегруженные технические специалисты зачастую вообще отказываются от действий или откладывают проблемы безопасности на потом, когда появится время, чтобы с ними разобраться.

Обзор лекции

Изучаемые концепции:

- Управление файлами журналов серверов
- Использование баз данных и Web-серверов для защитных данных
- Анализ данных систем обнаружения вторжений
- Управление данными сканирования уязвимостей
- Эксплуатация системы управления сканированием уязвимостей

Используемые инструменты:

Swatch, ACID, NPI и NCC

Чтобы избежать описанной ситуации, требуются средства, способные помочь организовать защитные данные и определить приоритеты действий. В этом отношении полезны многие коммерческие пакеты, такие как HP OpenView, BMC NetPatrol и NetIQ. К счастью для организаций с небольшим бюджетом, имеется несколько прекрасных пакетов с открытыми исходными текстами.

Хотя эти приложения с открытыми исходными текстами, строго говоря, не являются средствами безопасности, так как они активно не опрашивают и не защищают машины в сети, они во всех отношениях не менее важны, чем сетевые сканеры и системы обнаружения вторжений. Дело в том, что если вы "за деревьями не видите леса", то ваше положение ничем не лучше, чем раньше, когда защитных средств у вас не было.

Одним из примеров проблемы при анализе защитных данных служат журналы сообщений сервера. Большинство серверов, как UNIX, так и Windows, поддерживают журналы различной активности, происходящей в системе. Большая часть этой активности, такая как запуск служб, доступ пользователей и т.д., безвредна. Linux, например, поддерживает системные журналы в каталоге /var/log. Обычно имеется два общих журнала - syslog и messages - а также несколько других, более специальных. В этих текстовых файлах отражается все, что происходит в системе. На [листинге 8.1](#) показано типичное содержимое файла messages из Linux.

```
Aug 17 04:02:06 earth syslogd 1.4.1: restart.
Aug 18 21:07:57 earth sshd(pam_unix) [17904]: session opened for user join by (uid=502)
Aug 18 21:12:39 earth su(pam_unix) [17960]: session opened for user root by john (uid=502)
Aug 18 21:12:52 earth su(pam_unix) [17960]: session closed for user root
Aug 18 21:13:44 earth sshd(pam_unix) [18008]: session opened for user join by (uid=502)
Aug 18 21:14:02 earth sshd(pam_unix) [18008]: session closed for user join
Aug 18 21:23:21 earth su(pam_unix) [18482]: session opened for user root by john (uid=502)
Aug 18 21:24:12 earth su(pam_unix) [18482]: session closed for user root
Aug 18 21:39:00 earth su(pam_unix) [10627]: session opened for user root by john (uid=502)
Aug 18 21:44:57 earth httpd: httpd shutdown succeeded
Aug 18 21:44:58 earth httpd: httpd: Could not detemine the server's fully qualified domain name,
    using 127.0.0.1 for ServerName
Aug 18 21:45:00 earth httpd: httpd startup succeeded
Aug 19 23:39:02 earth sshd(pam_unix) [13219]: authentication failure:
    logname= uid=0 euid=0 tty=NODEVssh ruser= rhost=tayhou-tnt-9-216-40-228-250.isp.net user=john
Aug 22 10:31:14 earth sshd(pam_unix) [16205]: session opened for user tony by (uid=500)
Aug 22 10:31:20 earth su(pam_unix) [16240]: session opened for user root by tony (uid=500)
```

Листинг 8.1. Файл messages из Linux

Эти сообщения могут оказаться полезными при отладке системы или устранении неисправностей после установки новой программы. Можно видеть данные для каждой программы или процесса, запускаемых в системе. Видно, когда возникают какие-либо проблемы. Из приведенного примера можно понять, что процесс httpd, web-сервер, выключается. Следующая строка показывает, что имеется проблема с доменным именем, заданным для сервера. В данном случае по ошибке в качестве имени хоста был задан IP-адрес.

Файлы журналов содержат также информацию, представляющую интерес с точки зрения безопасности. Некоторые виды активности нередко служат предвестниками атаки. Неудачные попытки входа могут быть одним из таких признаков. На [листинге 8.1](#) можно увидеть, что пользователю "john" было отказано во входе. Указано даже, откуда john пытался войти - с адреса, принадлежащего Isp.net. Если это была просто одиночная неудачная попытка, и вы знаете, что "john" использует Isp.net, то можно, наверно, не беспокоиться. Вполне возможно, что john просто неправильно ввел пароль. Однако, если вы видите несколько неудачных попыток входа, или эта единственная попытка предпринята с незнакомого адреса, то ситуацию следует проанализировать внимательнее. Другой сигнал неблагополучия - самопроизвольная перезагрузка сервера в необычное время. Это лишь немногие из протоколируемых событий, требующих вашего внимания.

В идеале вы должны ежедневно просматривать все файлы журналов. К сожалению, большинство из нас не имеют времени делать это даже раз в неделю; некоторые вообще не заглядывают в журналы. В этих файлах слишком много информации, чтобы можно было легко найти действительно важные данные. Было бы здорово иметь вспомогательную программу, которая ищет подобные вещи и информирует, когда они появляются, чтобы мы могли прореагировать своевременно, а не через несколько дней или недель. К счастью, имеется программа с открытыми исходными текстами, которая именно это и делает.

Swatch: Программа мониторинга журналов

Swatch

Автор/основной контакт: Todd Atkins

Web-сайт: swatch.sourceforge.net/

Платформы: Linux, большинство UNIX

Лицензия: GPL

Рассмотренная версия: 3.0.8

Списки почтовой рассылки:

Swatch-announce

В основном для обновления версий и новых выпусков. Не принимает сообщений.

Подписка по адресу list.sourceforge.net/list/listinfo/swatch-announce.

Swatch-users

Для общих справок, вопросов и информации о разработке.

Подписка по адресу list.sourceforge.net/list/listinfo/swatch-users.

Разные люди расшифровывают название Swatch по-разному - как Simple watcher (простой наблюдатель) или как Syslog watcher (наблюдатель syslog). В любом случае, Swatch - полезная программа, которая берет на себя наблюдение за системными журналами и поднимает тревогу только при появлении в них специфицированных вами записей. Swatch - это Perl-программа, регулярно просматривающая файлы основных журналов и отыскивающая определенные ключевые слова, которые вы можете задавать. Она способна выполняться в фоновом режиме как демон или как задание cron. Можно сконфигурировать Swatch для сигнализации о любых событиях в файлах журналов messages и syslog, указывающих на проблемы безопасности. Однако Swatch можно также применять для извещения почти о любом виде активности: о запуске определенной программы, о входе определенного пользователя в систему или о других событиях, отражаемых в файле журнала. Swatch можно также сконфигурировать для наблюдения за журналами определенных приложений вместо подразумеваемых общих файлов журналов.

Возвращаясь к применению Swatch в целях безопасности, отметим, что некоторые протоколируемые события имеют к безопасности непосредственное отношение. По умолчанию Swatch отслеживает следующие события:

- Неудачные попытки входа. Критерии: В файле messages появляются слова Invalid, Repeated или Incomplete.
- Крах системы. Критерии: В файлах журналов появляются слова panic или halt.
- Перезагрузка системы. Критерии: Заголовок вашей ОС должен появляться в файлах журналов только при перезагрузке.

Вот некоторые другие связанные с безопасностью события, которые по вашему желанию может искать Swatch:

- Сообщения Snort или Nessus. Если вы записываете данные Snort в файл syslog, то можно указать, чтобы Swatch искал вхождения слова snort. Оно будет появляться каждый раз, когда генерируется сигнал Snort.
- Применение текстового редактора. Это может указывать на то, что кто-то в вашей системе пытается внести изменения в конфигурационные файлы. Конечно, если в системе часто используется текстовый редактор, будет порождаться слишком много ложных срабатываний.
- Применение FTP, SSH или Telnet. Если кто-то загружает или выгружает файлы на вашей системе, это может означать проблему. Аналогично, если кто-то удаленно входит в систему за этим необходимо проследить. Однако, если вы сами часто так поступаете, то будет больше шума, чем пользы.

Примечание: По умолчанию программа Swatch сконфигурирована для поиска элементов, специфичных для операционной системы Sun. Лишь некоторые из них могут быть общими для Linux и других операционных систем на основе UNIX. Например, при поиске перезагрузки в системе Mandrake Linux вам придется заменить аргументы поиска Sun OS Release на Linux Mandrake Release. Если вы используете другую ОС, укажите элементы из сообщения о перезагрузке вашей ОС. Чтобы узнать текст сообщения, выполните перезагрузку, а затем найдите в файле messages соответствующее специфическое словосочетание.

Установка Swatch

1. Для Swatch требуется Perl 5 или выше. Если вы используете новую установку Linux или BSD (не старше года), то у вас должна быть подходящая версия.

2. Для Swatch требуются также некоторые дополнительные подпрограммы для языка Perl: Date::Calc, Date::HiRes и Date::Format. Если у вас эти модули отсутствуют, процесс конфигурирования выдаст соответствующие сообщения. Чтобы получить модули, зайдите на web-сайт <http://www.prmfind.net/> и посмотрите, нет ли там доступного RPM для вашего дистрибутива.

Если вы используете дистрибутив Linux, то высока вероятность, что RPM для установки этих модулей присутствует на дисках дистрибутива. Если у вас нет RPM для модулей Perl, то для загрузки требуемых модулей можно воспользоваться системой Comprehensive Perl Archive Network (CPAN). (Система CPAN позволяет автоматически и без особых проблем загружать требуемые библиотеки Perl.) Чтобы сделать это, наберите следующую команду:

```
cpан -i имя_модуля
```

где `имя_модуля` заменяется на `Date::Calc` или тот модуль, который вы пытаетесь загрузить. Проверьте, что имя модуля задано правильно с учетом регистра символов и наберите оба двоеточия. Вам придется сделать это три раза - по одному для каждого требуемого модуля. Система CPAN позаботится о соединении с центральными серверами CPAN, загрузке модуля и его установке.

3. Загрузите tar-файл с прилагаемого к книге компакт-диска или с web-сайта Sourceforge и распакуйте его.
4. Так как Swatch является программой Perl, то процесс установки несколько иной по сравнению с предыдущими программами на языке Си. Последовательность вводимых команд будет следующей:

```
perl Makefile.PL
make
make test
make install
make realclean
```

Когда эти процессы завершатся, Swatch будет установлен и готов к последующим действиям.

Конфигурирование и запуск Swatch

Swatch - утилита командной строки, и запускается командой `swatch` с различными параметрами, описанными в [табл. 8.1](#).

Например, команда

```
./swatch --config-file /home/john/my-swatch-config --daemon
```

запустит Swatch с конфигурационным файлом `/home/john/my-swatch-config` вместо подразумеваемого. Программа будет выполняться как фоновый процесс или демон. Эти опции могут задаваться по отдельности или вместе.

Таблица 8.1. Опции команды Swatch

| Опция | Описание |
|---|---|
| --config-file имя_конфигурационного_файла | Swatch запускается с указанным конфигурационным файлом. По умолчанию используется <code>./swatchrc</code> . |
| --restart-time время | Swatch перезапускается в указанное время. Можно использовать знак <code>+</code> , чтобы программа перезапустилась по истечении заданного времени с текущего момента. Это полезно, чтобы освежить имеющийся образ файла журнала. |
| input-record-separator регулярное_выражение | Эта опция предписывает Swatch применять регулярное выражение для определения границ между записями и строками в файле журнала. По умолчанию используется возврат каретки, но если в вашей операционной системе применяется что-то иное, можно указать это здесь |
| daemon | Swatch запускается как системный демон. Эквивалентно запуску Swatch с ключом <code>&</code> (амперсанд) |

Таблица 8.2. Опции Swatch для файлов журналов

| Опция | Описание |
|-----------------------|---|
| --examine файл | Предписывает Swatch выполнить полный просмотр указанного файла. Применяется, когда проверяемый файл каждый раз создается заново |
| --read-pipe программа | Вместо чтения файла можно заставить Swatch осуществлять ввод непосредственно из канала от указанной программы |
| --tail файл | Читать только вновь добавленные в файл строки. Это подразумеваемый режим Swatch на файлах журналов, так как новые записи обычно добавляются в конец существующего файла. Это значительно быстрее, чем каждый раз читать весь файл, особенно для журналов, которые могут достигать больших размеров, как, например, журнал Web-сервера |

В [табл. 8.2](#) описаны некоторые дополнительные опции, которые можно применять для управления чтением файлов журналов. В каждый момент времени допускается использование только одной из них. Например, команда

```
./swatch - examine messages - daemon
```

заставит Swatch при каждом запуске выполнять поиск во всем файле messages, а не только во вновь добавленных строках

Swatch обычно просматривает UNIX-файл messages, а при отсутствии такового по умолчанию читается syslog. При помощи ключей из [табл. 8.2](#) можно заставить Swatch просматривать любые файлы журналов, например, журналы безопасности или даже файлы журналов определенных приложений, такие как nessus.messages

Конфигурационный файл Swatch

В конфигурационном файле Swatch располагаются все важные настройки. В этом файле, именуемом по умолчанию swatchrc, вы указываете программе, что искать в журналах и что делать в случае успешного поиска. Два примера файлов swatchrc поставляются вместе с программой в каталоге examples. Файл swatchrc.personal предназначен для применения на персональных рабочих станциях, а swatchrc.monitor - для мониторинга сервера. На [листинге 8.2](#) показано, как выглядит версия monitor

```
#
# Конфигурационный файл Swatch для постоянного мониторинга
#
# Неудачные попытки входа
watchfor /INVALID|REPEATED|INCOMPLETE/
    echo
    bell 3
    exec "/usr/local/sbin/badloginfinger $0"

# Температура в машинном зале
watchfor /WizMON/
    echo inverse
    bell

# Аварии и остановки машины
watchfor /(panic|halt)/
    echo
    bell
    mail
    exec "call_pager 3667615 0911"
```

```
# Перезагрузки системы
watchfor /SunOS Release/
    echo
    bell
    mail
    exec "call_pager 3667615 0411"
```

Листинг 8.2. Конфигурационный файл swatchrc.monitor

Как можно видеть из [листинга 8.2](#), базовый формат включает инструкцию watchfor, за которой следует текстовая инструкция между двумя косыми чертами, а затем одна или несколько команд действия. Текст между косыми чертами служит аргументом поиска, когда Swatch просматривает файл журнала (или его последние записи). В случае успешного поиска Swatch выполняет указанные далее действия. В [табл. 8.3](#) описаны поддерживаемые Swatch инструкции действий.

Таблица 8.3. Инструкции действий Swatch

| Инструкция действия | Описание |
|--|---|
| echo режим | Выдает искомый текст на экран. Режим указывать не обязательно; он задает цвет выводимого текста. По умолчанию используется обычный экранный цвет текста, но можно также задавать режимы blink (мигание), bold (жирный), underline (подчеркнутый), inverse (инверсный) и цвета green, blue, red, yellow, black, magenta, cyan, white или любой из перечисленных вариантов, за которым следует суффикс _h для подсвеченной цветной версии, например black_h |
| bell число | Звуковой сигнал через внутренний динамик ПК указанное число раз. По умолчанию выдается один сигнал. |
| exec команда | Выполняет указанную команду. Полезно для вызова другой программы или командного файла с целью выполнения различных действий, таких как отправка всплывающего сообщения SMB на определенную рабочую станцию. Эта опция существенно расширяет возможности Swatch. Можно даже задать вызов командного файла, который будет предпринимать дальнейшие действия согласно некоторым условиям, с учетом того, что именно было найдено в файле журнала |
| pipe команда | Передает команду в другой процесс. |
| mail addresses=адрес1:адрес2:адрес3,subject=текст | Посылает электронное сообщение, используя программу Sendmail по одному или нескольким адресам, разделенным двоеточием, с заданным текстом темы. Текст сигнала тревоги помещается в тело письма |
| write пользователь1:пользователь2 | Посылает сигнал посредством UNIX-команды write одному или нескольким пользователям. |
| throttle часы:минуты:секунды | Управляет тем, сколько раз за некоторый промежуток времени посылается сигнал для одной инструкции watchfor. Это избавляет от получения десятков сообщений, если строка текста несколько раз появляется в файле журнала в пределах заданного временного окна |

Как можно видеть, Swatch способен извещать вас об отмеченных протоколируемых событиях несколькими различными способами. Простейшим является подача звукового сигнала или вывод текста на экран. Если вы не находитесь постоянно рядом с сервером, можно настроить программу для отправки электронных сообщений. Если ваш пейджер или сотовый телефон поддерживает обмен текстовыми сообщениями по электронной почте, то вы сможете получать сообщения на своем устройстве. Можно также написать командный файл, чтобы сервер набирал номер пейджера с помощью UNIX-команды tip

Использование баз данных и web-серверов для управления защитными данными

Раз уж вы вышли за рамки простой проверки журналов серверов, вы захотите также иметь возможность анализировать выдачу программ безопасности, рассмотренных ранее. Лучший способ сделать это - импортировать результаты в базу данных. Средства, описанные в оставшейся части данной лекции, предназначены для импорта и просмотра защитных данных в БД. Для применения этих средств требуются программа базы данных и web-сервер для просмотра результатов. Хотя имеются и другие возможности, мы рекомендуем использовать базу данных MySQL и Web-сервер Apache с поддержкой PHP.

Необходимо настроить эти программы, прежде чем устанавливать любое из описываемых средств. Далее кратко описана установка и конфигурирование упомянутых базовых серверов

Настройка сервера MySQL

MySQL является базой данных на основе SQL с открытыми исходными текстами, заслужившей признание корпоративного мира за свою мощь и гибкость. Хотя эта книга не предназначена для обучения всем тонкостям применения MySQL, следующая процедура поможет настроить и выполнить некоторые основные административные задачи на базе данных MySQL, чтобы можно было использовать средства анализа.

1. Загрузите самую свежую версию MySQL с сайта <http://www.mysql.com/> или используйте RPM с дистрибутивных дисков вашей ОС. Проверьте, что версия MySQL не ниже 4.0.

Примечание: Если у вас уже имеется установленная база данных MySQL версии 4.0 или выше, перейдите к шагу 4.

2. Распакуйте файл и выполните обычные команды компиляции в созданном каталоге:

```
./configure  
make  
make install
```

3. Выполните командный файл установки, расположенный в каталоге scripts, набрав

```
mysql_install_db
```

Программа базы данных будет инициализирована и подготовлена к работе.

4. Создайте пользователя и группу mysql для базы данных, от имени которых будут выполняться задачи. Для этого следует набрать команды

```
groupadd mysql  
useradd -g mysql mysql
```

5. Чтобы MySQL могла функционировать, задайте владельца и режим доступа файла с помощью следующих команд:

```
chown -R root /usr/local/mysql  
chown -R mysql /usr/local/mysql.var  
chgrp -R mysql /usr/local/mysql  
cp /usr/local/mysql/support-files/my-medium.cnf /etc/my.cnf
```

6. Откройте в редакторе файл /etc/ld.so.conf и добавьте в него следующие строки:

```
/usr/local/mysql/lib/mysql  
/usr/local/lib
```

Сохраните файл.

7. От имени пользователя root введите

```
ldconfig -v
```

8. От имени пользователя root задайте пользователя admin для базы данных MySQL, набрав

```
/usr/local/mysql/bin/mysqladmin -u root password 123456
```

где 123456 надо заменить выбранным вами паролем. Не забудьте записать пароль и сохранить его в безопасном месте.

Когда вы это сделаете, вернитесь под свое обычное входное имя, набрав "exit" в командной строке.

9. Можно настроить MySQL для запуска в качестве демона, чтобы БД выполнялась все время, и не надо было запускать ее вручную. Для этого достаточно поместить следующую строку в конце файла rc.local, находящегося в /etc/rc.d/.

```
mysqld -user=mysql &
```

Эта команда будет запускать MySQL как системный процесс при каждой перезагрузке системы.

10. Наконец, необходимо повысить защищенность MySQL, чтобы БД не стала дырой в безопасности вашей системы. По умолчанию защищенность MySQL весьма слаба. Хотя безопасность MySQL не является темой данной книги, ниже представлено несколько советов, которыми можно воспользоваться.
- Удалите стандартных пользователей, если только у вас нет программ, которые их используют.
 - Проверьте, что пользователь root может подключаться только с небольшого числа хостов.
 - Задайте несколько правил на межсетевом экране, разрешающих соединение с сервером MySQL только ограниченному числу портов с ограниченного числа машин.
 - Создайте системные счета для запуска программ. Системный счет root или счет MySQL root (это две разные вещи) используйте только в случае крайней необходимости (к сожалению, NPI этого требует). Данная лекция включает примеры специальных счетов приложений для создания в каждом пакете описания везде, где это возможно.

Теперь сервер MySQL готов к работе. Наберите mysql в командной строке ОС. Появится приглашение для ввода имени пользователя и пароля, чтобы войти в стандартную командную строку MySQL, где можно применять стандартные команды SQL к базам данных MySQL. См. врезку о некоторых основных командах MySQL.

Основные команды MySQL

Чтобы войти в MySQL, наберите `mysql -u имя_пользователя -p пароль`, заменяя `имя_пользователя` и `пароль` соответствующими именем и паролем одного из счетов базы данных MySQL.

Примечание: Это не то же самое, что вход в систему. В данном случае вы входите в MySQL и получаете приглашение `mysql>`, после которого можно набирать команды. Не забывайте ставить в конце команды точку с запятой, прежде чем нажать клавишу ввода для ее выполнения.

Ниже представлено несколько основных команд для навигации и поиска в базе данных MySQL.

| | |
|--|---|
| <code>show databases;</code> | Отображает все доступные на сервере MySQL базы данных. |
| <code>use имя_базы_данных;</code> | Делает указанную базу данных активной, после чего над ней можно выполнять операции. |
| <code>show tables;</code> | Перечисляет все таблицы, существующие в базе данных. |
| <code>select запрос from имя_таблицы;</code> | Выдает записи, соответствующие заданному запросу в таблице с указанным именем. Имеется ряд операндов, которые можно использовать в инструкции запроса. Звездочка * в качестве запроса приведет к выводу всех записей таблицы. |

Настройка web-сервера Apache

Развитые средства анализа из данной лекции основаны на применении web-сервера в качестве как интерфейса конфигурирования, так и механизма вывода. Конечно, этот краткий раздел не претендует на исчерпывающее описание web-сервера; здесь рассмотрены только настройка и другие действия, требуемые для использования средств безопасности. Если вы намерены применять этот сервер для чего-то еще, помимо ACID и NCC, или в крупномасштабных средах, вам необходимо более глубоко ознакомиться с администрированием web-сервера. При использовании web-сервера следует учитывать вопросы безопасности - необходимо позаботиться о том, чтобы серверы были укреплены, выполняли минимум сервисов и по возможности быстро латались. Если вы желаете применять IIS или другой web-сервер, то он должен поддерживать PHP версии 4.0 или выше.

1. Загрузите самую свежую версию сервера Apache с сайта <http://www.apache.org/>. Если он есть на дистрибутивных дисках ОС или уже установлен в системе, проверьте, что его версия не ниже 1.3.

Примечание: Если сервер Apache версии 1.3 или более поздней уже установлен, перейдите к шагу 3.

2. Распакуйте программу и выполните следующие команды:

```
./configure --prefix=/www --enable-so --activate-module=src/modules/php4/libphp4.a
make
make install
```

Эти команды задают подразумеваемый каталог /www и активизируют нужные модули.

3. Запустите web-сервер, набрав в командной строке `apachectl start`. Эта команда запускает демон http и настраивает его для выполнения в качестве системного процесса.

Можно остановить Apache в любой момент, выполнив ту же команду с аргументом `stop`.

В других вариантах Linux и UNIX запуск и остановка могут осуществляться по-другому. Уточните в документации, как это делается.

4. Проверьте установку web-сервера, открывая web-навигатор и вводя IP-адрес сервера или задавая `localhost`, если вы работаете прямо на серверной машине. Если будет выведена web-страница Apache, то web-сервер успешно установлен. Корневым каталогом web-сервера, в который помещают документы для публичного просмотра, в системе Mandrake Linux служит `/usr/local/apache2/htdocs/`; различные дистрибутивы могут немного различаться.
5. Затем задайте автоматический запуск Apache при перезагрузке системы (вряд ли вы захотите перезапускать web-сервер вручную). Для этого перейдите в каталог, где находятся все стартовые командные файлы; в Mandrake Linux это `/etc/rc.d`. Каждый файл `rc` представляет свой уровень выполнения. Добавьте следующие строки в файлы `rc4.d` и `rc5.d`:

```
../init.d/httpd S85httpd
../init.d/httpd K85httpd
```

Можно протестировать внесенные изменения, перезагрузив систему и проверив, что в выдаче команды `ps -ax` присутствует процесс `httpd`.

6. Необходимо повысить защищенность Apache, чтобы предотвратить его ненадлежащее использование. web-серверы - одна из наиболее распространенных целей атакующих, поэтому если вы собираетесь разрешить доступ к этой машине извне вашей сети, требуется обеспечить ее безопасность. Ниже представлены некоторые основные рекомендации по обеспечению хорошей безопасности web-сервера:
 - Выполните сканирование уязвимостей Web-сервера сразу после завершения установки и конфигурирования, чтобы убедиться, что все корректирующие заплатки наложены и отсутствуют какие-либо очевидные дыры в безопасности.
 - Защитите все непубличные Web-каталоги с помощью какого-либо метода контроля доступа. Самым быстрым и легким способом является применение файлов `.htaccess`.
 - Шифруйте коммуникации между клиентами и сервером с помощью SSL всякий раз, когда имеете дело с информацией ограниченного доступа (данные о безопасности явно попадают в эту категорию). Если вы обращаетесь к серверу извне своей локальной сети, то есть через Интернет, справьтесь в документации web-сервера или в Интернете о необходимых настройках.

Вышеизложенное не является исчерпывающим рассмотрением проблем безопасности web-серверов, но все это необходимо выполнить, прежде чем делать сервер общедоступным.

Настройка PHP

PHP является интерпретируемым языком, предназначенным для использования в web-страницах. Он не требует компиляции, поэтому можно просто поместить PHP-процедуру в каталог, который распознает PHP, и она будет выполняться при обращении. Это упрощает написание программ, встроенных в

web-страницы. Большинство современных web-серверов распознают PHP, однако для этого может потребоваться дополнительная настройка при установке.

В силу перечисленных достоинств PHP стал предпочтительным языком реализации многих приложений на web-платформе. Он потребуется нам для трех оставшихся средств этой лекции (ACID, NPI и NCC). Установка PHP должна быть предусмотрена в директиве configure в описанной выше процедуре установки Apache. Чтобы проверить, что PHP установлен в вашей системе, и узнать, какова его версия, наберите в командной строке `php -v`. Если он присутствует, то должна появиться некоторая выдача с номером версии. Однако если вы не смогли установить его как часть Apache или хотите установить самую свежую версию, примените представленную ниже процедуру.

1. Загрузите самую свежую версию PHP с сайта <http://www.php.net/> или используйте RPM с установочных дисков операционной системы. В последнем случае проверьте, что у вас версия 4.0 или выше.
2. Распакуйте дистрибутив.
3. В каталоге установки выполните следующие команды компиляции:

```
./configure -prefix=/www/php -mysql=/usr/local/mysql \ -with-apxs2=/www/bin/apxs
-with-zlib-dir=/usr/local (all on one line)
-with-gd
make
make install
```

Инструкция configure включает несколько модулей, нужных средствам данной лекции.

4. Отредактируйте конфигурационный файл web-сервера httpd.conf, как правило, находящийся в /www. Добавьте следующие строки, а затем сохраните файл:

```
LoadModule php4_module modules/libphp4.so
AddType application/x-httpd-php.php
```

5. Чтобы проверить, что PHP работает правильно, воспользуйтесь текстовым редактором для создания небольшой процедуры в файле с именем test.php. Наберите в файле следующий текст, а затем сохраните его:

```
<?php phpinfo(); ?>
```

При выполнении этой PHP-процедуры будет выдана некоторая базовая системная информация.

6. Скопируйте тестовый файл в каталог /www/htdocs. Введите URL или IP-адрес машины, а затем наберите /test.php. Вы должны увидеть на web-странице номер версии PHP. Если все получилось, то web-сервер с поддержкой PHP готов к работе.

ACID (Консоль анализа для баз данных вторжений)

ACID

Автор/основной контакт: Roman Danyliw

Web-сайт: <http://www.andrew.cmu.edu/~rdanyliw/snort/snortacid.html>

Платформы: Большинство UNIX

Лицензия: GPL

Рассмотренная версия: .9.9b23

Список почтовой рассылки:

Список пользователей Acidlab. Подпишитесь, послав сообщение со словом "subscribe" в теле письма по адресу acidlab-users@lists.sourceforge.net.

Программа ACID (Analysis Console for Intrusion Databases - консоль анализа для баз данных вторжений) предназначена для более эффективного использования данных, генерируемых средствами обнаружения вторжений. Ее написал Роман Данылиев с коллегами в рамках проекта AirCERT, выполняемого университетом Карнеги-Меллон. Это часть более крупной деятельности CERT (Computer Emergency Response Team - Группа реагирования на нарушения информационной безопасности). CERT в течение многих лет успешно применяет эту программу для защиты Интернета и организаций. CERT отслеживает компьютерные преступления и направляет извещения в списки почтовой рассылки, когда происходит крупный инцидент. Список почтовой рассылки CERT является разновидностью системы раннего предупреждения обо всех больших кризисах или атаках, происходящих в Интернете. Как таковой он может быть весьма полезен системным администраторам. Вы можете посетить сайт CERT <http://www.cert.org/> и подписаться на почтовую рассылку.

В рамках проекта AirCERT сенсоры систем обнаружения вторжений были размещены в различных организациях с целью изучить общие тенденции вторжений. Чтобы облегчить анализ, была написана программа ACID. Поскольку исходные тексты этого проекта были сделаны открытыми, вы можете использовать их в собственных целях, не принимая участия в проекте AirCERT.

Положенная в основу ACID идея состоит в переносе всех данных об обнаруженных вторжениях в базу данных, где их можно отсортировать и организовать по приоритетам. ACID предоставляет панель управления на основе Web для сортировки, просмотра и манипулирования этими результатами.

ACID может использовать почти любую базу данных SQL и любой web-сервер и поддерживает множество сенсоров для ввода данных. Допускаются также необработанные сигналы Snort и файлы журналов в формате syslog. В настоящее время ACID работает напрямую только с одной системой обнаружения вторжений - Snort, но с помощью утилиты Logsnorter, которая доступна на web-сайте ACID, можно импортировать журналы в базу данных ACID из любого устройства, выводящего данные в формате syslog.

Для своей работы ACID требует наличия некоторых программ. Кроме базы данных, web-сервера и PHP, которые уже были рассмотрены в этой лекции, нужны также следующие библиотеки и подпрограммы.

ADODB

Этот пакет обеспечивает уровень абстракции базы данных, позволяющий PHP использовать стандартный интерфейс для множества баз данных, включая MySQL. Возьмите его по адресу <http://php.weblogs.com/adodb>, распакуйте в /www/htdocs или подходящем корневом каталоге Web, и он должен быть готов к работе. Никакой дополнительной установки не требуется.

PHPLOT

Этот пакет позволяет создавать графики с помощью ACID. Если вы хотите использовать эту возможность, возьмите модуль с <http://www.phplot.com/>. Распакуйте его в каталоге /www/htdocs и, так же как ADOdb, он должен быть готов к употреблению.

JpGraph

Эта программа позволяет PHP генерировать цветные графики. Она понадобится наряду с PHPLOT, если вы захотите представлять данные Snort в графическом виде. Возьмите ее по адресу <http://www.aditus.nu/jpgraph/> и распакуйте в корневом каталоге Web (например, /www/htdocs). Она создаст собственный подкаталог и будет доступна, когда понадобится для ACID.

GD

Этот пакет содержит библиотеки манипуляции изображениями для PHP, которые нужны также для создания графиков. Если вы установили PHP согласно данным ранее в этой лекции инструкциям, то у вас уже должна иметься эта утилита. В противном случае возьмите ее по адресу <http://www.boutell.com/gd/> и установите в каталоге /www/php. Если вы не компилировали PHP с помощью представленных выше команд, то следует также убедиться, что имеются следующие библиотеки, необходимые для GD.

- libpng. Предоставляет для GD поддержку формата PNG. Ее можно взять на <http://www.libpng.org/pub/png/> или с дистрибутивных дисков вашей ОС.
- libjpeg-6b. Это библиотека jpeg для PHP. Можно взять ее на <http://www.ijg.org/> или с дистрибутивных дисков вашей ОС.
- zlib. Эта библиотека предоставляет для GD поддержку сжатия. Можно взять ее на <http://www.ijg.org/> или с дистрибутивных дисков вашей ОС.

Конфигурирование Snort для MySQL

1. ACID предполагает, что имеется один или несколько активных сенсоров Snort, поставляющих данные. Если вы еще не создали сенсоры Snort, вернитесь к [лекции 7](#). Сенсоры Snort необходимо сконфигурировать таким образом, чтобы они записывали данные в MySQL. Для этого выполните следующие действия при установке Snort:

- При первоначальной компиляции Snort используйте следующую инструкцию configure:

```
./configure - with-mysql=/usr/local/mysql
```

Проверьте, что указан каталог, где находится MySQL.

- Отредактируйте файл конфигурации snort.conf. Найдите закомментированную строку, которая начинается с #output database. Отредактируйте ее следующим образом:

```
output database: log,mysql,user=snort password=123456 dbname=snort host=localhost
```

Замените пользователя snort и пароль 123456 на правильные имя пользователя базы данных и его пароль, которые будут применяться для ACID. ACID создаст базу данных с именем "snort", хотя можно изменить это имя, редактируя файл конфигурации ACID. Если вы подключаетесь к локальной базе данных, то оставьте у переменной host значение localhost. Если вы подключаетесь к базе данных на другой машине, задайте здесь IP-адрес или имя хоста.

2. Не забудьте удалить символ комментария # в начале строки и затем сохраните файл.

В данной лекции предполагается, что ACID и сенсор Snort устанавливаются на разных машинах. Размещение их на одной машине неудачно не только с точки зрения безопасности; работа сенсора Snort замедлится до такой степени, что от окажется бесполезным. Компьютер с ACID предпочтительно расположить в сегменте сети, отличном от сегмента с сенсорами Snort - это затруднит взломщику доступ к журналам. На [рис. 8.1](#) показаны элементы связи ACID-Snort.

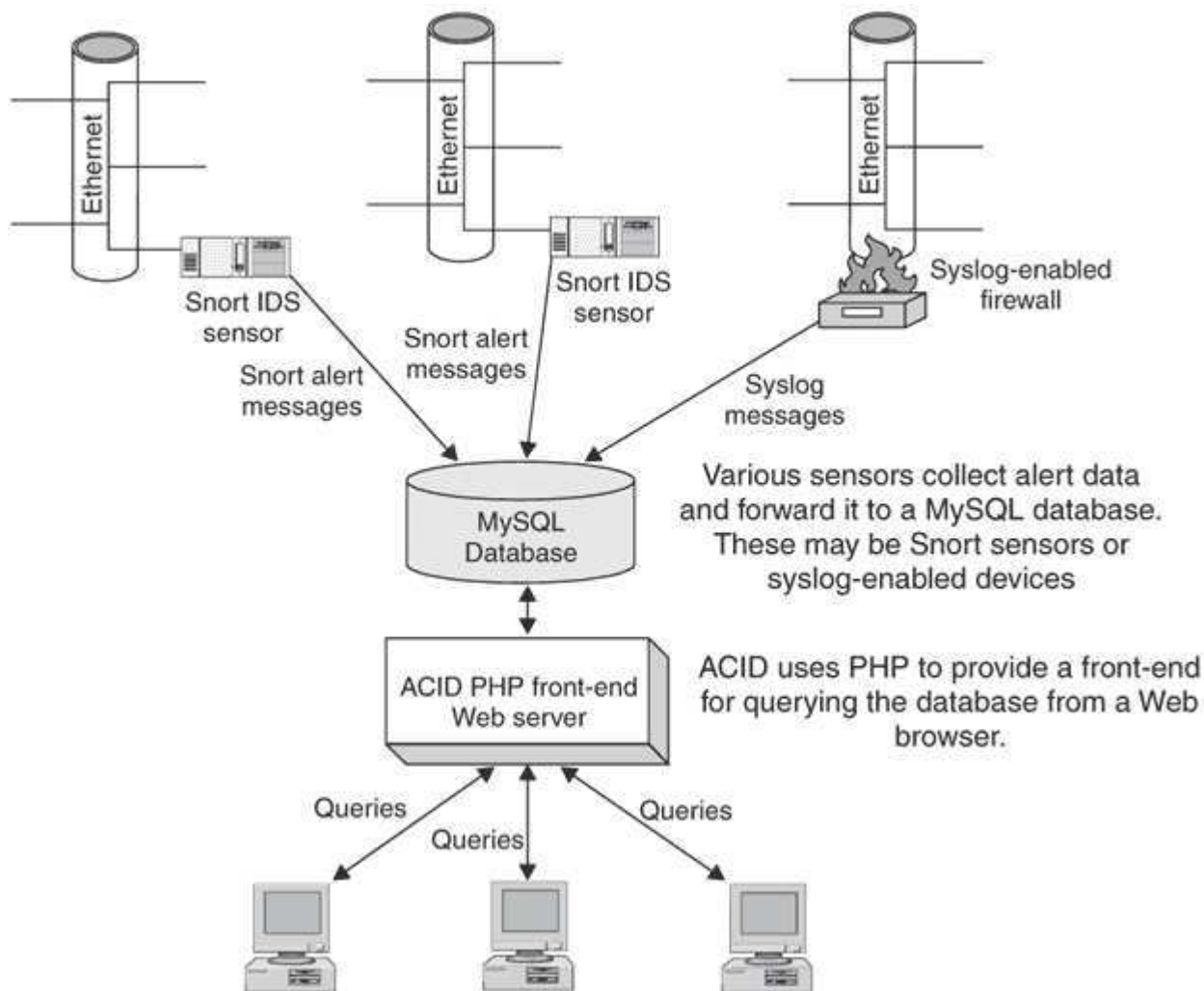


Рис. 8.1. Система обнаружения вторжений ACID-Snort

Установка ACID

После загрузки всех необходимых программ можно, наконец, установить ACID.

1. Возьмите файл программы с компакт-диска книги или с web-сайта ACID.
2. Поместите tar-файл в каталог /www/htdocs. Распакуйте его там, и он создаст собственный каталог.
3. Удалите tar-файл, поскольку все, оставленное в корневом каталоге /htdocs, может быть доступно пользователям web-сервера.

Конфигурирование ACID

1. Перейдите в каталог /htdocs/www/acid.
2. Отредактируйте файл acid_conf.php. Строки, начинающиеся с косой черты и звездочки, служат комментариями и инструкциями по конфигурированию. Строки, начинающиеся с \$, являются переменными и сообщают программе специфическую информацию о системе.

3. В инструкциях \$ задайте параметры своей системы. В [табл. 8.4](#) перечислены переменные, а также рекомендации для каждого элемента.

Таблица 8.4. Переменные для конфигурирования ACID

| Имя переменной | Описание |
|---------------------|---|
| \$DBtype | Тип базы данных, которую будет использовать ACID. По умолчанию - mysql, но можно также указать postgresql или mssql, если вы хотите применить какую-либо из этих двух баз данных |
| \$alert_dbname | Система обнаружения вторжений, данные которой использует ACID. В настоящее время поддерживается только собственный формат Snort snort_log, хотя имеются планы по расширению этого набора |
| \$alert_host | Хост, на котором будет храниться база данных сигналов. Может задаваться как IP-адрес или имя хоста. Если ACID и база данных располагаются на одной машине, то следует указать localhost. Для повышения безопасности и производительности целесообразно выделить для базы данных машину, отличную от Web-сервера PHP |
| \$alert_port | Порт, по которому происходит обращение к базе данных. Если вы размещаете ее локально, то задайте данное значение просто как " " |
| \$alert_user | Имя пользователя базы данных, которое будет применять ACID при протоколировании данных. Проверьте, что оно совпадает с именем пользователя MySQL, созданным при настройке базы данных |
| \$alert_password | Пароль пользователя базы данных. И здесь проверьте, что он совпадает с паролем MySQL для данного пользователя |
| \$archive_dbname | Имя базы данных, которую Snort использует для архивирования. Подразумеваемое имя snort_archive вполне разумно, если только вы не храните несколько баз данных на одной машине и не хотите задать более содержательные имена |
| \$archive_host | Хост, на котором будет располагаться база данных архива. Если она находится на той же машине, то значение должно задаваться как localhost |
| \$archive_port | Порт для записи на сервере базы данных. Используйте " ", если запись происходит локально |
| \$archive_user | Пользователь базы данных, от имени которого производится запись архивных данных. Обычно это значение совпадает с \$alert_user (см. выше), хотя можно создать отдельного пользователя для записи архивов |
| \$archive_password | Пароль для пользователя базы данных, от имени которого записываются архивные данные. Обычно совпадает с \$alert_password |
| \$chartlib_path | Маршрут к модулям создания графиков - /www/htdocs/jpgraph-1.11/src |
| \$chart_file_format | Формат файлов графиков. По умолчанию - png. Другими допустимыми форматами служат jpg и gif |

4. После сохранения файла с этими параметрами откройте Web-навигатор и введите /acid/acid_main.php после имени хоста или IP-адреса Web-сервера. Пример: http://localhost/acid/acid_main.php

Будет выведена страница с конфигурацией ACID. С этого момента можно применять Web-интерфейс для завершения конфигурирования ACID.

5. Щелкните мышью на кнопке Create ACID AG. Это приведет к созданию базы для данных Snort. Подразумеваемое имя этой БД - "snort".

6. Перейдите на <http://localhost/acid/>, и вы увидите основную страницу ACID для своей базы данных Snort ([рис. 8.2](#))

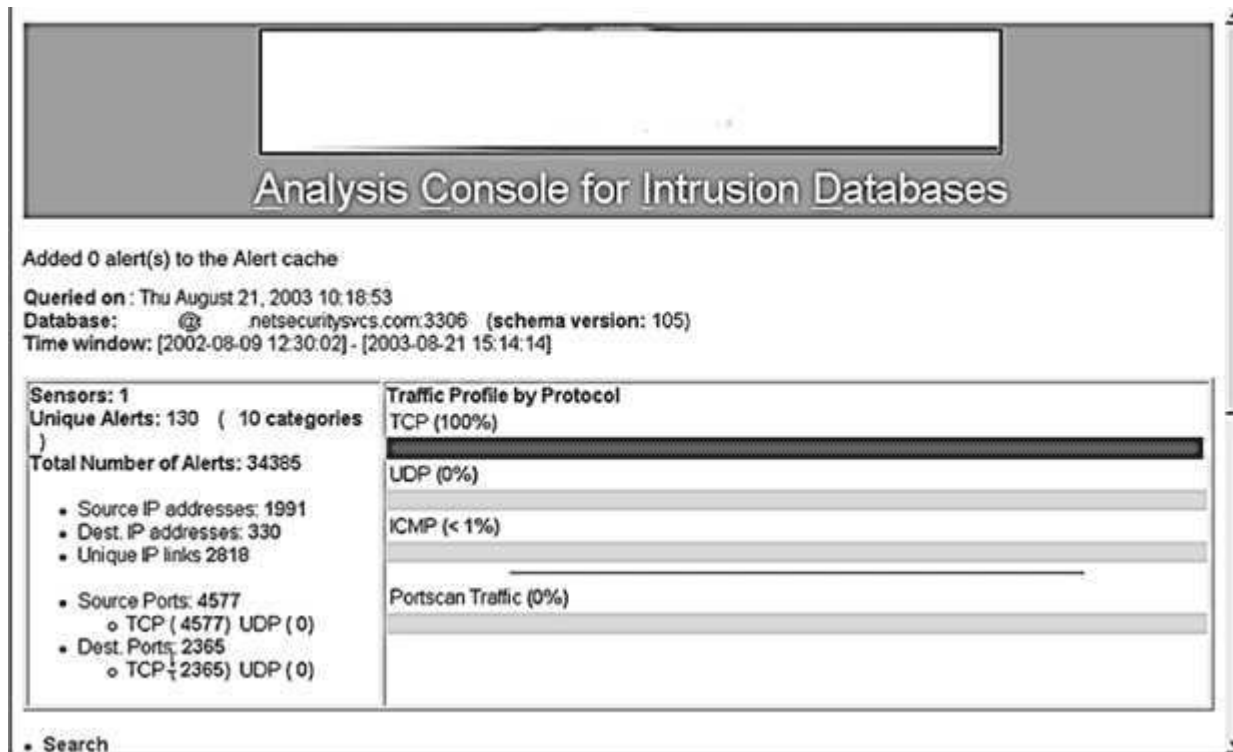


Рис. 8.2. Основной интерфейс ACID

На этом конфигурирование ACID завершается и можно приступать к использованию этой программы для управления системами обнаружения вторжений.

Основы применения ACID

При первом входе в ACID отображается основная страница ([рис. 8.2](#)). Верхняя часть основного представления базы данных показывает общую статистику просматриваемой БД, включая ее имя, временные рамки всех содержащихся в ней записей, дату и время последнего запроса.

Раздел, расположенный ниже, содержит всю сводную информацию по конкретной группе сигналов тревоги (AG - alert group). AG - это сенсор или группа сенсоров, представленных в этой базе данных. Если вы желаете отслеживать различные группы сенсоров как единое целое (например, сенсоры для различных заказчиков или подразделений), то необходимо создать отдельную базу данных или AG для каждой группы. Это важно для создания отчетов и применения архивных средств ACID. Вы сможете выполнять поиск или запросы только на отдельных AG, а не на множестве AG, поэтому необходимо организовать сенсоры различных AG подходящим образом. Для большинства организаций будет достаточно иметь одну группу сигналов тревоги для всех сенсоров. Но если вы работаете в консультационной компании или имеете дело с большим объемом операций по нескольким подразделениям, то, вероятно, удобнее распределить группы сенсоров по различным AG, чтобы можно было следить за ними по отдельности.

В прямоугольнике слева на экране можно видеть статистику для данной AG: общее число сигналов, число уникальных сигналов и число различных IP-адресов (как исходных, так и целевых), фигурирующих в базе данных. Если у вас несколько сенсоров в сети ACID, то можно щелкнуть мышью на пункте Sensors, чтобы увидеть их список. Можно ограничить поиск данными только одного сенсора. На основной странице представлены также графические профили трафика сигналов для каждого протокола и порта, чтобы можно было понять, каков вид трафика, проходящего через сенсор сетевой системы обнаружения вторжений.

Применение ACID для управления сетевыми системами обнаружения вторжений и их настройки

Прежде чем ваша сетевая система обнаружения вторжений станет сколько-нибудь полезной, ее необходимо настроить под вашу сеть, чтобы исключить ложные срабатывания. Для подобной деятельности ACID бесценна. При первом включении сетевой системы обнаружения вторжений все сигнатуры сигналов активны и ваша база данных начнет заполняться сигналами. Первоначально эти сигналы по большей части будут ложной тревогой. Для приведения сигнализации в соответствие с вашей сетью, следует удалить некоторые сигнатуры, чтобы избавиться от большей части ложных срабатываний и получать от нее только те данные, которые заслуживают внимания.

Когда вы накопите в базе данных достаточное количество сигналов (по крайней мере тысячу для загруженной сети), можно начать анализировать тревожные данные и исключать некоторые типы сигналов. Внимательно наблюдайте за своей базой данных, так как для ее заполнения может потребоваться не так много времени, особенно для подразумеваемого списка правил Snort.

Откройте ACID и щелкните мышью на кнопке Unique Alerts. Будут показаны самые свежие из полученных сигналов, сгруппированные по типу (рис. 8.3).

ACID

Alert Listing: 15 Last Alerts

Home | Search | AG Maintenance

[Back]

Added 0 alert(s) to the Alert cache

Queried DB on : Thu August 21, 2003 10:26:38

Meta Criteria

any

IP Criteria

any

Layer 4 Criteria

none

Payload Criteria

any

Displaying 15 Last Alerts

| < Signature > | < Classification > | < Total # > | Sensor # | < Src. Addr. > | < Dest. Addr. > | < First > | < Last > |
|--|------------------------|-------------|----------|----------------|-----------------|---------------------|---------------------|
| <input type="checkbox"/> [arachNIDS] [CVE] WEB-IIS ISAPI .ida attempt | web-application-attack | 3638 (11%) | 1 | 1445 | 2 | 2002-08-09 15:11:13 | 2003-08-21 15:14:14 |
| <input type="checkbox"/> WEB-IIS cmd.exe access | web-application-attack | 16630 (48%) | 1 | 1488 | 2 | 2002-08-11 17:28:02 | 2003-08-21 15:14:14 |
| <input type="checkbox"/> MS-SQL xp_reg* - registry access | attempted-user | 209 (1%) | 1 | 3 | 1 | 2002-08-09 12:33:07 | 2003-08-21 13:35:07 |
| <input type="checkbox"/> [bugtraq] [CVE] [arachNIDS] WEB-CGI formmail access | attempted-recon | 266 (1%) | 1 | 70 | 2 | 2002-10-01 14:25:20 | 2003-08-20 15:13:27 |
| <input type="checkbox"/> [arachNIDS] WEB-MISC http directory traversal | attempted-recon | 683 (2%) | 1 | 65 | 2 | 2002-08-15 10:40:09 | 2003-08-20 10:13:41 |
| <input type="checkbox"/> [CVE] DDOS mstream handler to client | attempted-dos | 17 (0%) | 1 | 3 | 1 | 2002-08-19 09:16:47 | 2003-08-20 02:39:23 |
| <input type="checkbox"/> [url] WEB-IIS CodeRed v2 root.exe | web-application- | 1103 (3%) | 1 | 228 | 2 | 2002-08-13 14:20:30 | 2003-08-19 14:54:16 |

Рис. 8.3. Список самых последних уникальных сигналов тревоги

На этой странице представлена следующая информация для каждого типа сигналов:

- Имя сигнатуры.
- Классификация сигнала.
- Общее число сигналов этого типа в базе данных.
- Номер сенсора, с которого пришел сигнал.
- Число различных исходных IP-адресов, ассоциированных с этим сигналом.
- Число различных целевых IP-адресов, ассоциированных с этим сигналом.
- Время прихода сигнала.

Можно выполнить сортировку по любому из столбцов, щелкая мышью на маленькой стрелке вверху столбца. Например, имеет смысл отсортировать список по числу сигналов и щелкнуть мышью на строке, соответствующей максимальному числу срабатываний. Это сузит список до одного типа сигналов.

Просмотрите список и попытайтесь определить, действительно ли это проблема безопасности или ложное срабатывание. Наличествуют ли какие-нибудь отличительные особенности? Замешан ли во всех сигналах этого типа один IP-адрес, исходный или целевой? Генерируются ли сигналы с регулярными интервалами или кажутся случайными? Если этот анализ не ведет к каким-либо выводам, то копайте глубже, щелкая мышью на отдельных сигналах. Это позволит увидеть реальный пакет, который вызвал сигнал, что весьма полезно с юридической точки зрения, если вы действительно были атакованы и пытаетесь в дальнейшем отреагировать или преследовать атакующих.

Будьте осторожны. Если по сети передаются секретные данные, то вы можете нечаянно их увидеть, так как перехватываете и анализируете целые пакеты данных. Убедитесь, что вам разрешено видеть эти данные. Также очень важно, чтобы база данных Snort была защищена должным образом, так как любой, кто проникнет в машину базы данных, потенциально будет иметь доступ к этой секретной информации. Другое решение этой проблемы - понижение уровня детализации данных, фигурирующих в правилах сигнала, хотя это может помешать прослеживанию виновника на основе зарегистрированных сигналов тревоги.

В примере на [рис. 8.3](#) Web-IIS cmd.exe является самым распространенным сигналом. Щелкнув мышью на данных сигнала, можно увидеть реальный пакет, который порождает этот сигнал ([рис. 8.4](#)). Показан исходный IP-адрес вместе со всеми портами TCP и настройками.

Meta

| ID # | Time | Triggered Signature |
|--------|---------------------|------------------------|
| 1 - 44 | 2002-08-11 17:28:02 | WEB-IIS cmd.exe access |

| Sensor | name | interface | filter |
|--------|------|-----------|--------|
| | iea | eth0 | none |

| Alert Group | none |
|-------------|------|
|-------------|------|

IP

| source addr | dest addr | Ver | Hdr Len | TOS | length | ID | flags | offset | TTL | chksum |
|----------------|---------------|-----|---------|-----|--------|-------|-------|--------|-----|--------|
| 133.15.196.110 | 192.168.1.206 | 4 | 5 | 0 | 99 | 50575 | 0 | 0 | 102 | 17169 |

| FQDN | Source Name | Dest. Name |
|------|-------------|---------------------------|
| www. | | Unable to resolve address |

| Options | none |
|---------|------|
|---------|------|

TCP

| source port | dest port | R1 | R0 | U | A | P | R | S | F | seq # | ack | offset | res | window | urp | chksum |
|-------------|-----------|----|----|---|---|---|---|---|---|------------|------------|--------|-----|--------|-----|--------|
| 2912 | 80 | | | X | X | | | | | 3195060598 | 2174337536 | 5 | 0 | 16560 | 0 | 14051 |

| Options | none |
|---------|------|
|---------|------|

Payload

| | | | | | | | | | | | | | | | | | | |
|-------------|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------------------|
| length = 55 | | | | | | | | | | | | | | | | | | |
| 000 | : | 47 | 45 | 54 | 20 | 2F | 73 | 63 | 72 | 69 | 70 | 74 | 73 | 2F | 2E | 2E | 25 | GET /scripts/..% |
| 010 | : | 35 | 63 | 25 | 35 | 63 | 2E | 2E | 2F | 77 | 69 | 6E | 6E | 74 | 2F | 73 | 79 | %h%o../vinnt/sy |
| 020 | : | 73 | 74 | 65 | 6D | 33 | 32 | 2F | 63 | 6D | 64 | 2E | 65 | 78 | 65 | 3F | 2F | sten32/cmd.exe?/ |
| 030 | : | 63 | 2B | 64 | 69 | 72 | 0D | 0A | | | | | | | | | | c+dir.. |

Рис. 8.4. Детали сигнала тревоги в ACID

По имени хоста можно сказать, что пакет пришел с адреса в Японии (домен верхнего уровня .jp) и определить, можно ли считать обращение к вашей сети с этого адреса нормальным. Можно копнуть глубже и увидеть реальную полезную нагрузку пакета. Слева находятся данные пакета в шестнадцатеричном виде, справа - в текстовом (если их можно представить таким образом). Это показывает реальные команды, которые отправитель пытается выполнить на вашей машине. Глядя на эти данные можно предположить, что кто-то пытается получить доступ к команде cmd.exe, иными словами, получить приглашение командной строки. Очевидно, это атака на вашу систему. К сожалению, это, скорее всего, запрограммированная атака, проводимая Интернет-"червем", а атаки такого типа случаются каждый день десятками, как можно понять по большому числу сигналов cmd.exe в базе данных. Тем не менее, стоит за этим понаблюдать и проверить, не появляется ли этот IP-адрес постоянно. Можно, по крайней мере, написать жалобу поставщику Интернет-услуг и убедиться, что атакованная машина (определяемая по целевому адресу) защищена от подобных вещей. Можно также принять дополнительные меры против IP-адреса, указанного как исходящий, например, начать юридическое преследование или вчинить гражданский иск, если

произошло реальное проникновение. По крайней мере, теперь вы точно знаете, какого вида атаки приходят в вашу сеть и что они пытаются делать. Это позволит лучше защитить сеть и реагировать, если она окажется под атакой.

Другие способы проанализировать данные сигналов тревоги с помощью ACID

Кого атакуют?

С помощью ACID найдите наиболее распространенные целевые IP-адреса, то есть IP-адреса, которые, вероятно, атакуются чаще всего и на которых необходимо сконцентрировать усилия по защите. Это поможет также отличить ложные срабатывания от реальных, поскольку вы, возможно, обнаружите, что некая машина создает огромное число сигналов из-за приложения, которое на ней выполняется. Внезапный всплеск количества сигналов на определенном IP-адресе может указать на развивающуюся атаку на эту машину. Затем можно принять меры для повышения безопасности этой машины, провести сканирование уязвимостей, проверить уровень "залатанности", отбрасывать на маршрутизаторе пакеты из враждебного источника и т.д.

Кто атакует?

Определите исходящий IP-адрес, проявляющийся чаще всего. Перейдите с основной страницы к списку исходных IP-адресов. Это покажет IP-адрес и полностью квалифицированное доменное имя и подскажет, откуда идет атака. Сортировка по количеству сигналов позволит увидеть самых злостных нарушителей в терминах порождаемых сигналов. Если IP-адреса с наибольшим количеством сигналов находятся в вашей сети, то, вероятно, имеется внутренний злоумышленник или приложение, которое включает сигнал. Используйте рассмотренный выше процесс для углубления на уровень данных сигнала и анализа сигнала. Если сигналы тревоги порождены внешними IP-адресами, то желательно определить, законен ли трафик, направленный в вашу сеть, или это реальная атака. Просмотрите отдельные сигналы, чтобы понять, что пытаются делать. Щелкните мышью на IP-адресе, будет выведена страница с дополнительной информацией об адресе и некоторыми опциями для дальнейшего анализа (см. [рис. 8.5](#)). В ACID можно применить к этому адресу различные функции, такие как обратный поиск DNS, поиск ARIN и даже поиск Sam Spade (аналогичный средству, рассмотренному в [лекции 2](#)). Выдача этих функций должна подсказать вам, какая организация владеет этими IP-адресами, контактные адреса электронной почты их центра сетевых операций, и адреса для сообщений о злоупотреблениях (если таковые имеются). Можно использовать эти контактные адреса при регистрации жалобы на выявленную активность. Если вы заметите, что какие-то адреса появляются вновь и вновь, их можно отфильтровать на маршрутизаторе или межсетевом экране.



Рис. 8.5. Детали исходного IP-адреса в ACID

Какой сервис атакуется чаще всего?

Определяя порты, на которых чаще всего возникают сигналы, можно понять, на какие сервисы направлено большинство атак. Если вы видите много сигналов на основе Web, следует уделить повышенное внимание усилению защиты web-серверов. Если сигналы показывают высокую активность NetBIOS Windows, то необходимо провести ревизию прав доступа в Windows и политики паролей. Иными словами, анализ подскажет, на каких сервисах сосредоточиться в первую очередь.

Ежедневное применение ACID

Располагая работающей программой ACID, настроенной в соответствии с конфигурацией сетевой системы обнаружения вторжений, следует взять за правило проверять как минимум раз в день, какие новые сигналы были сгенерированы. Лучше всего утром делать первую проверку, а перед уходом с работы - повторную. Если есть персонал, работающий после окончания дневной смены, то и ему можно поручить проверку базы данных сигналов ACID.

После входа в базу данных ACID можно сразу перейти в раздел Snapshot ([рис. 8.6](#)) и щелкнуть мышью на Most Recent Alerts, чтобы быстро просмотреть вновь поступившие данные о сетевой активности. Будут выведены все сигналы в хронологическом порядке. Если по-прежнему генерируется настолько много сигналов, что их анализ затруднителен, то в разделе Today's Alers выберите Unique. Будут показаны все сегодняшние сигналы, сгруппированные по типам, и можно видеть, какие из них порождают больше всего трафика. Полезны также опции Last 24 Hours и Last 72 Hours из раздела Snapshot. Они позволяют выявить наиболее часто встречающиеся сигналы, адреса, и порты за различные периоды времени.

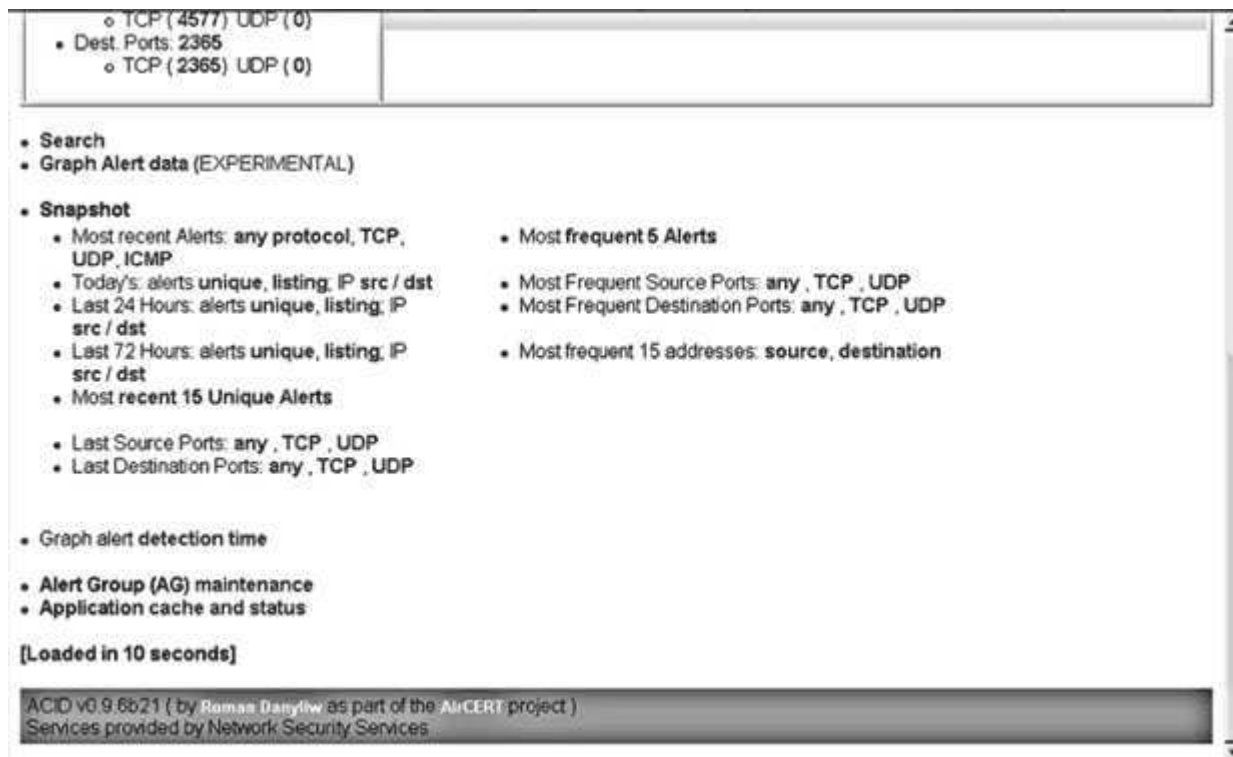


Рис. 8.6. Раздел Snapshot в ACID

Графическое представление данных ACID

Если вы предпочитаете зрительные образы, или вам нужны графики для демонстрации руководству, воспользуйтесь имеющимися в ACID средствами для построения графиков и диаграмм на основе базы данных сигналов. Эти средства пока имеют статус экспериментальных, и для работы с ними необходимы графические модули PHP, перечисленные в начале этого раздела, однако они удобны для графического вывода итоговых данных Snort. Графические средства можно вызвать, щелкнув мышью на Graph Alert Data сразу под прямоугольником со статистикой сигналов на основном экране ACID. В результате будут отображены графические опции. Данные для графиков можно организовать следующим образом:

- По времени (час, день, месяц) относительно числа сигналов;
- По IP-адресам (исходным или целевым) относительно числа сигналов;
- По портам TCP или UDP (исходным или целевым) относительно числа сигналов;

Задайте параметры с помощью раскрывающихся полей и щелкните мышью на Graph Data. Проверьте, что заполнили все поля, иначе будет выдано сообщение об ошибке. ACID построит и выведет график. На [рис. 8.7](#) показан пример графика ACID.

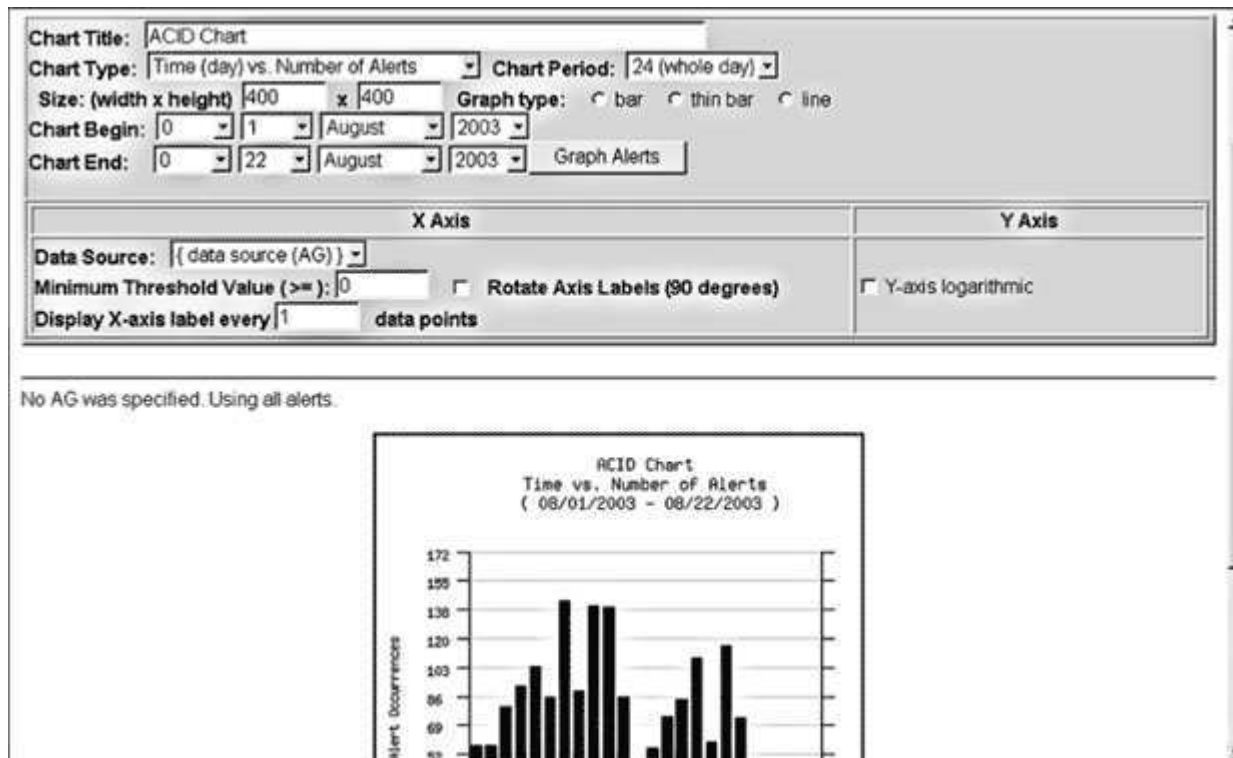


Рис. 8.7. График тревожных данных в ACID

Обслуживание базы данных ACID

По мере роста базы данных сигналов ей периодически требуется некоторое обслуживание, чтобы она не становилась слишком большой и громоздкой. Ваша статистика и графики также будут более точными, если архивировать давние сигналы с множеством ложных срабатываний. Кроме того, периодическая очистка базы ускорит обработку запросов.

Для архивирования сигналов воспользуйтесь регулятором запросов внизу основного экрана. Создайте запрос для сигналов, которые хотите архивировать, например, все сигналы, сгенерированные за последний год. Затем выберите Archive Alerts в качестве действия для запроса. Можно выборочно архивировать сигналы по дате, типу и другим критериям. Можно также выбрать простое копирование сигналов в архив или их удаление. Архивированные сигналы будут помещаться в собственную базу данных с именем, которое было задано в файле acid_conf.php в процессе конфигурирования.

Следует заархивировать все сигналы первых нескольких месяцев работы, в течение которых вы настраивали сенсоры Snort. После этого данные будут в большей степени отражать реальные атаки, а не ложные срабатывания. Неплохо также выполнять архивирование как минимум раз в год, или, возможно, ежеквартально, в зависимости от количества зарегистрированных сигналов. Как правило, нежелательно иметь в базе данных одновременно более 100000 сигналов.

Итак, теперь вы знаете, как построить законченную сеть Snort обнаружения вторжений с множеством сенсоров, протоколирующих сигналы в базу данных, которую можно применять для анализа данных и генерации отчетов. Это поможет лучше использовать данные об обнаруженных вторжениях, получить максимальную отдачу от усилий по обеспечению безопасности и запастись наглядными отчетами и графиками для демонстрации руководству. Теперь мы рассмотрим несколько средств, помогающих разобраться с результатами сканирования уязвимостей.



Флэми Тех советует:

Аккуратно используйте названия!

Будьте осторожны в разговоре со своим руководством о применении Snort или ACID на работе. Убедитесь, что руководство понимает, что это ценные программы управления, а не противозаконные наркотические вещества!

NPI (Nessus PHP Interface)

NPI

Автор/основной контакт: Kristofer T. Karas

Web-сайт: <http://enterprise.bidmc.harvard.edu/pub/nessus-php/>

Платформы: Большинство UNIX

Лицензия: GPL

Рассмотренная версия: 01a

Одна из проблем при использовании сканера уязвимостей Nessus для сканирования сетей среднего и большого размера состоит в том, что отчеты получаются весьма устрашающими. Вроде бы и форматы отчетов Nessus хороши, и перемещаться по HTML-документам легко, но, получив на анализ пару сотен страниц данных, довольно трудно выделить на фоне всего этого шума важные данные. Было бы прекрасно иметь возможность организовать результаты сканирования удобным для изучения способом. Для того чтобы анализировать результаты, следует поместить их в базу данных, а не выдавать в виде стандартного плоского файла. Желательно также иметь простой доступ к данным, например, посредством web-интерфейса. Располагая такими средствами, можно легко выделить наиболее важные данные и проанализировать изменение результатов сканирования во времени, чтобы понять, становится ли сеть более или менее защищенной.

К счастью, имеется несколько продуктов, интегрирующих Nessus с базой данных: NesQuick, Java Nessus Report Manager и Nessus PHP Interface (NPI). Для этой книги я по ряду причин выбрал NPI. Во-первых, это на самом деле продукт с открытыми исходными текстами без каких-либо коммерческих завязок. Во-вторых, он опирается на MySQL и PHP, которые мы уже применяли для других средств, таких как ACID. С помощью этих приложений NPI обеспечивает перенос данных Nessus в базу данных и их просмотр при помощи web-навигатора.

NPI по своей архитектуре аналогичен ACID. Он использует базу данных MySQL для хранения результатов и поддерживающий PHP web-сервер для просмотра и запроса результатов. На [рис. 8.8](#) показаны логические компоненты NPI. Одно из различий между архитектурами Snort и Nessus состоит в том, что в Nessus есть две отдельные части, порождающие данные: клиент, иницирующий сканирования, и выполняющий их сервер. В некоторых случаях они могут находиться на одной машине, но на рисунке изображены два различных физических сервера. Имеется также сервер базы данных, куда записываются данные сканирования, и web-сервер, предоставляющий интерфейс к данным. База данных и web-сервер также могут находиться на одной машине или на двух разных.

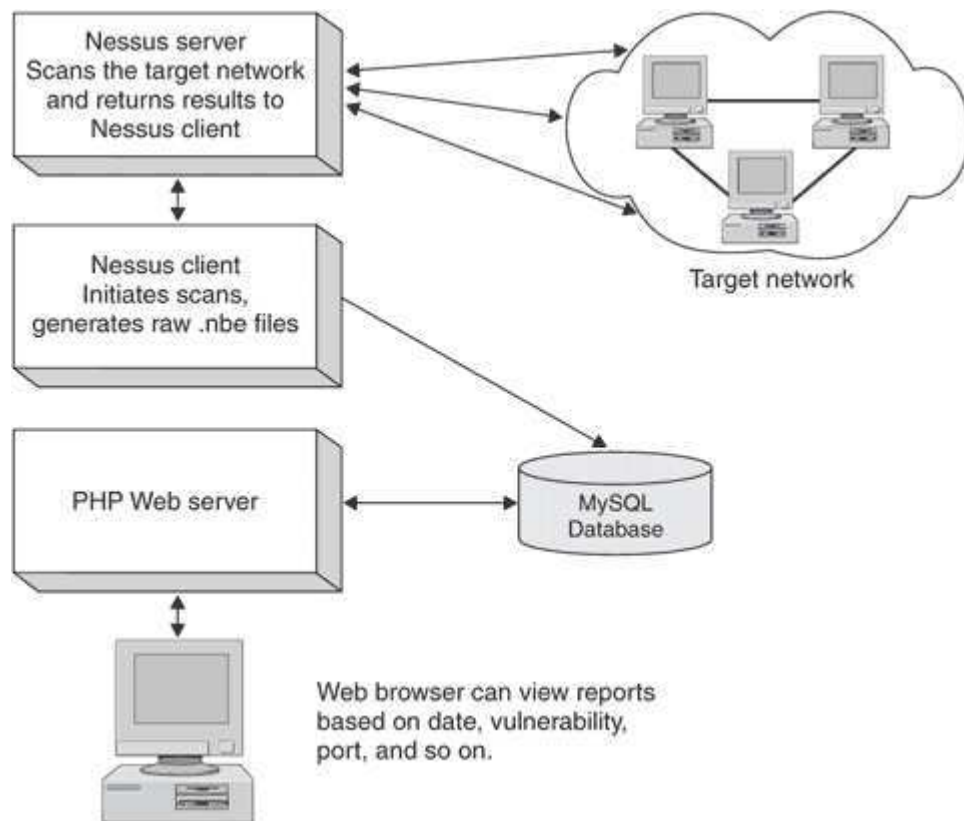


Рис. 8.8. Логическая архитектура NPI

[Рис. 8.8](#) иллюстрирует потоки данных и логические части системы NPI. Клиент Nessus входит на сервер Nessus для инициации сканирования некоторого объекта. Данные сохраняются на клиентской машине в собственном формате Nessus .nbe. Когда вы получаете необработанный файл, NPI выполняет процедуру его преобразования и импортирует данные в БД MySQL. После этого данные можно просматривать и запрашивать БД посредством любого web-навигатора через PHP-интерфейс. Этот способ анализа результатов сканирования имеет существенные преимущества, поэтому стоит немного потрудиться и установить NPI.

Установка NPI

Прежде чем приступить к установке NPI, необходимо располагать работающим сервером MySQL и web-сервером Apache с поддержкой PHP. Если они еще не установлены, то вернитесь назад к соответствующим разделам этой лекции и сделайте это. Когда серверы SQL и Web будут готовы к работе, выполните следующие действия по установке NPI.

1. Возьмите файл с прилагаемого к книге компакт-диска, распакуйте и "растарьте" программу. Поместите содержимое в отдельный каталог, имя которого не имеет значения (если все файлы находятся в одном каталоге). Например, подойдет каталог `/usr/local/nessus-php`, если, конечно, у вас есть право на запись в него.
2. Перейдите в этот каталог и отредактируйте файл `nsr.php` с помощью текстового редактора. Этот файл содержит переменные (строки, начинающиеся с `$`), которые позволяют программе контактировать с сервером MySQL. В [табл. 8.5](#) перечислены переменные, которые необходимо отредактировать, и даны рекомендации для задания их значений.

Таблица 8.5. Переменные NPI для сервера MySQL

| Переменная | Описание |
|------------|----------|
|------------|----------|

| | |
|----------------------------|--|
| <code>\$db_host</code> | Задается хост, на котором будет выполняться сервер MySQL. Если сервер MySQL выполняется на той же машине, что и web-сервер, укажите localhost. В противном случае введите IP-адрес или имя хоста машины MySQL |
| <code>\$db_user</code> | Имя пользователя, используемое для входа в базу данных MySQL. Может совпадать с именем, заданным при установке MySQL |
| <code>\$db_pass</code> | Пароль, соответствующий вышеуказанному пользователю |
| <code>\$db_database</code> | Имя базы данных, созданной для хранения данных NPI. По умолчанию NPI создает базу данных с именем nessus, но при желании его можно заменить другим |
| <code>\$db_suuser</code> | Административный пользователь базы данных MySQL. Требуется для процедуры nsr-php. По умолчанию используется root, что годится для большинства установок. Отметим, что это не то же самое, что пользователь root операционной системы |
| <code>\$db_supass</code> | Пароль пользователя root для MySQL, соответствующий вышеописанному счету \$db_suuser |
| <code>\$your_domain</code> | Список доменов, имена которых вы хотите вырезать из вывода. Это очистит отчеты, если вы хотите, чтобы в выдаче присутствовали только имена машин, а не полные имена хостов Интернета. Это необязательная переменная |

- Откройте файл nsr в текстовом редакторе и создайте в нем те же элементы, что и в шаге 2 ([табл. 8.5](#)). Проверьте, что маршрутное имя каталога встраиваемых модулей Nessus задано правильно. В большинстве установок эти модули должны находиться в подразумеваемом каталоге /usr/local/lib/nessus/plugins. Убедитесь, что это действительно так. В противном случае сделайте необходимые изменения в файле nsr.
- Отредактируйте файл nessusphp.inc, изменяя переменные так же, как и в двух предыдущих шагах.
- Создайте базу данных Nessus. Для этого нужно выполнить процедуру nsr-php:

```
php nsr-php -b
```

Для ваших данных сканирования будет создана база данных MySQL с именем nessus. Можно войти в нее и убедиться, что нужные таблицы созданы.

- Проверьте, что база данных создана должным образом.
 - Войдите в MySQL с помощью команд, описанных выше в этой лекции, во врезке, посвященной MySQL.
 - Наберите `show databases;` (не забудьте точку с запятой) в командной строке MySQL. Будет выдан список всех баз данных, и среди них должна присутствовать вновь созданная база данных Nessus.
 - Перейдите в эту базу данных, используя команду `use NESSUS;`, а затем наберите команду `show tables;`. Вы должны получить результаты, показанные на [рис. 8.9](#), - все три таблицы из базы данных nessus.

```
mysql> show tables;
+-----+
| Tables_in_nessus |
+-----+
| report            |
| scans             |
| scripts           |
+-----+
3 rows in set (0.00 sec)
```

Рис. 8.9. Вывод команды show tables

- Перенесите файлы из подкаталога www установочного каталога NPI во вновь созданный подкаталог корневого каталога документов web-сервера. Задайте правильные режимы доступа для файлов. В этом месте вы будете осуществлять доступ к базе данных Nessus. Выполните следующие команды из каталога nessus-php:

```
mkdir /usr/local/apache2/htdocs/nessus-php
mv ./www /usr/local/apache2/htdocs/nessus-php
```

```
chown -R www:www /usr/local/apache2/htdocs/nessus-php
chmod 755 /usr/local/apache2/htdocs/nessus-php/*
```

Не забудьте изменить маршрутное имя корневого каталога документов Web, если оно не такое, как в примере. При желании можно изменить имя каталога, в котором находятся файлы nessus-php. В приведенном примере страница доступа NPI помещается в подкаталог nessus-php корневого каталога документов Web. Для пользователя и группы, которые будут осуществлять доступ к базе данных MySQL, в примере использовано одно системное имя (www). Для повышения безопасности заведите специального пользователя, которому будет разрешено только чтение данных. В этом случае необходимо заменить www:www на соответствующих пользователя и группу, которым разрешено только чтение. Необходимо сделать это и для аналогичного счета MySQL.

Программа NPI установлена.

Импорт результатов сканирования Nessus в NPI

Теперь мы готовы к импорту результатов сканирования Nessus в базу данных.

1. Примените процедуру `nsr` к каждому импортируемому файлу с результатами сканирования Nessus. (Для этого, очевидно, у вас должны быть файлы проведенного сканирования, сохраненные в собственном формате `.nbe`.) NPI также допускает и преобразует более старый формат Nessus, `.nsr`. Наберите следующую команду для запуска `nsr` из командной строки:

```
./nsr ./scans/scan.nbe
```

Замените `./scans/scan.nbe` на маршрутное имя своего файла сканирования. Команда импортирует исходный файл Nessus в базу данных. Она также проверит встраиваемые модули Nessus и создает записи в базе данных для всех новых модулей, которые могли быть добавлены.

2. Теперь все готово к просмотру результатов сканирования Nessus в базе данных. Откройте web-навигатор и введите IP-адрес web-сервера NPI с маршрутным именем индексного файла Nessus-php, например: <http://localhost/nessus-php/>. Результаты должны появиться в интерфейсе PHP, позволяющем выполнять поиск и сортировку ([рис. 8.10](#)).



Рис. 8.10. Основной экран NPI

Применение NPI

Теперь можно просматривать результаты сканирования, как любую другую базу данных, сортировать их и выполнять запросы для поиска определенных уязвимостей, хостов и т.д. Имеется много способов анализа результатов Nessus с помощью NPI. Можно выполнять сортировку на основе:

- Хоста (IP-адреса) с наибольшим количеством уязвимостей;
- Наиболее распространенной уязвимости;
- Наиболее распространенной категории средств использования уязвимостей;
- Наиболее часто эксплуатируемого сервиса (номера порта);
- Даты сканирования или диапазона дат;
- Номера CVE или CAN.

Выполнение NPI-запросов позволяет сосредоточиться на областях, которые представляют наибольшую опасность для вашей сети, и максимизировать результаты вашей деятельности по повышению безопасности. Можно также быстро исключить определенные сигналы и/или машины. Посредством NPI можно произвольным образом, по вашему выбору манипулировать результатами сканирования. Некоторым недостатком NPI является ручная загрузка каждого файла сканирования в базу данных. Не составляет труда написать командный файл для автоматизации этого процесса, но мы с коллегами решили сделать в этом направлении еще один шаг.

Рождение проекта с открытыми исходными текстами

Моя консультационная компания активно применяет Nessus и Snort. Мы также используем ACID для управления системами обнаружения вторжений. Нам требовалось аналогичное средство для управления сканированиями Nessus. Хотя в NPI имеются некоторые очень удобные возможности, он все же не вполне удовлетворял наши потребности. Нам нужно было инициировать сканирования через web-интерфейс, а не только просматривать последние

результаты. Мы также пришли к выводу, что ручной импорт результатов каждого сканирования труден и отнимает много времени. У нас десятки сканирований, которые нужно выполнять в самое разное время, и, поскольку они обычно принадлежат различным организациям, их следует отслеживать отдельно. На самом деле мы хотели иметь средство, управляющее различными конфигурациями сканирования, планирующее их, выполняющее их автоматически и импортирующее результаты в соответствующую базу данных.

Мы не смогли найти средства с открытыми исходными текстами, удовлетворяющего всем перечисленным требованиям, поэтому мы оказались перед выбором: искать подходящий коммерческий продукт или выполнить собственную разработку. Как оказалось, даже коммерческие сканеры уязвимостей предлагали не совсем то, что требовалось для планирования и отслеживания данных сканирования различных клиентов. Очевидно, написание нового сканера уязвимостей "с нуля" было бы непродуктивным. Поэтому была выдвинута идея разработки дополнительного модуля для Nessus в качестве проекта с открытыми исходными текстами. Мы рассмотрели ряд вопросов, чтобы понять, насколько это целесообразно. Если вы собираетесь писать свою программу с открытыми исходными текстами, вам следует сделать то же, принимая во внимание следующие факторы.

Нет ли уже чего-то подходящего?

Прежде всего, поищите в Web, нет ли каких-либо средств, делающих то, что вам нужно. Посмотрите в таких местах, как Sourceforge.net и Freshmeat.net, используйте Google и другие поисковые машины. Велика вероятность, что для решения вашей задачи что-то уже существует. Если будет найдено "почти, но не совсем то", возможно, это можно использовать в качестве основы или вспомогательного средства при создании своей программы, как мы и сделали в случае NPI. Даже если ничего не существует, можно найти некоторые ответы на часто задаваемые вопросы или сайты с полезной для проекта информацией. Во время исследования вы можете также найти людей с аналогичной проблемой, желающих участвовать в проекте.

Имеет ли ваша программа широкую область применения?

Если решаемая проблема специфична для вашей организации, то, вероятно, нет смысла связываться с проблемами выпуска ее как продукта с открытыми исходными текстами. Пошлите несколько сообщений в подходящие телеконференции, чтобы проверить, есть ли к данной теме какой-то интерес. При отсутствии интереса можно остановиться на внутреннем проекте. Однако даже в небольших отраслях обычно имеются схожие потребности в приложениях, а Интернет сделал мир еще меньше. Подумайте обо всех случаях, когда вы искали что-то весьма необычное и все же находили, потому что кто-то в Web решил, что это стоит опубликовать. Поэтому, если есть хоть какой-то интерес, учреждайте проект с открытыми исходными текстами!

Разрешено ли вам выпускать свой продукт как открытое ПО?

Если задуманный проект входит в ваши обязанности наемного работника, убедитесь, что вам разрешат открыть исходные тексты. Если это часть большой собственной программы организации, то маловероятно, что исходные тексты позволят открыть. Однако, если это самостоятельная программа, то с учетом выгод дополнительной критики и помощи сторонних разработчиков, а также бесплатной рекламы, вполне вероятно, что руководство возражать не будет. Проясните этот вопрос, прежде чем начать распространение исходных текстов, если дорожите своей работой.

Мы прошли через этот процесс и решили, что по приведенным выше соображениям стоит разработать собственное дополнение к Nessus как проект с открытыми исходными текстами. Мы назвали эту программу Nessus Command Center (NCC). Вы присутствуете на публичной премьере этой программы.

Nessus Command Center (NCC)

Авторы/основные контакты: Tony Howlett, Brian Credeur, Matt Sisk, Lorell Hathcock

Web-сайт: <http://www.netsecuritysvcs.com/ncc>

Платформы: Linux, большинство UNIX

Лицензия: GPL

Рассмотренная версия: . 01b

Список почтовой рассылки:

Общая дискуссия и вопросы о NCC. Направьте электронное письмо со словом "subscribe" в теле или в теме по адресу ncc@netsecuritysvcs.com.

Появление NCC обусловлено имевшейся у нас потребностью в средствах управления для автоматизации сканирования и более качественного анализа результатов. Описанное выше средство NPI вполне успешно импортирует данные Nessus в базу данных, но не решает вопросов планирования, да и интерфейс оставляет желать лучшего. Чтобы не изобретать колесо, мы применили в нашем проекте web-интерфейса составные части NPI и добавили модули управления и планирования. Проект преследовал цель создания следующих сущностей:

- Платформа управления для Nessus-сканирования. Нам требовалось средство отслеживания сканирований для различных организаций с различными конфигурациями и даже различных групп организаций. Как консультационная компания, мы имеем дело с множеством различных организаций. У нас даже есть посредники, заказывающие для других организаций сканирование с использованием нашей инфраструктуры. Мы хотели получить единую панель управления всеми этими разнородными сущностями, сохраняя их разделенность.
- База данных расписаний и интерфейс для Nessus. Первой целью была разработка способов каталогизации данных сканирований, планирования сканирований и их автоматического выполнения. Мы хотели иметь возможность отслеживать различные сущности, так как сканирования будут проводиться в интересах многих организаций. Должен существовать административный уровень для создания и планирования сканирований, а также потенциальная возможность для клиентов входить и модифицировать определенные части своей конфигурации сканирования, такие как время, сканируемые хосты и т.д. База данных должна поддерживать web-интерфейс, так как наши заказчики и агенты, обращающиеся к системе для настройки своих сканирований, могут располагаться вне нашего межсетевого экрана.
- Интерфейс базы данных для результатов Nessus. Эта цель уже была частично достигнута в программе NPI, но мы хотели улучшить интерфейс, который представлялся нам рудиментарным и не поддерживал, например, многопользовательский доступ с различными уровнями полномочий. Мы планировали использовать NPI в качестве основы для этой части программы. А так как NPI имеет лицензию GPL, и наша программа будет иметь лицензию GPL, это не будет создавать никаких проблем.
- Web-интерфейс для настройки всех параметров Nessus. На самом деле, это необязательное требование. При изучении проблемы мы обнаружили, что в большинстве сканирований применяется лишь около пяти различных конфигураций. Однако было бы удобно иметь возможность настраивать все возможные параметры сканирования Nessus прямо из Web, не загружая клиента Nessus. Это позволяет вводить параметры сканирования прямо из офиса заказчика или из любого другого места. Мы осмотрелись и обнаружили еще один проект с открытыми исходными текстами, называемый Inprotect, который предлагал web-интерфейс для Nessus. Код был выпущен с лицензией GPL, поэтому мы могли использовать его в качестве руководства к нашим действиям в этой области. В связи со сложной природой данной задачи мы решили, что эта возможность не будет присутствовать в бета-версии.

Платформы для NCC

С самого начала мы решили, что будем писать программу для ряда платформ, называемого LAMP (Linux, Apache, MySQL и Perl).

- Linux: По очевидным причинам Linux предлагает наибольшую мобильность и наименьшую стоимость использования. Однако нет причин, которые препятствуют системе выполняться на других разновидностях UNIX с небольшими модификациями. Ее можно также перенести на платформу на базе Windows, такую как Perl for Windows.
- Apache: Сервер Apache был также выбран в связи с открытостью исходных текстов и потому, что он является одним из наиболее популярных web-серверов. Он явился самым логичным выбором, поскольку web-сервер должен использоваться и для других средств. Кроме него, эта система будет выполняться на любом поддерживающем PHP web-сервере, включая IIS.
- MySQL: Имеется несколько хороших баз данных с открытыми исходными текстами, включая Postgresql и другие. Мы выбрали MySQL, потому что были лучше всего с ней знакомы, а требования в лицензии были самыми простыми. Как и Apache, мы уже применяли MySQL для своих баз данных ACID.
- Perl: Конечно, существует множество интерпретируемых языков, но мы выбрали Perl потому, что он один из самых мобильных, не требует компиляции и легко модифицируем для третьих сторон.

Основываясь на архитектуре LAMP, мы начали создавать средство, удовлетворяющее нашим потребностям. Сначала мы написали план проекта, детализировавший задуманную работу. Затем мы разбили задачу на подзадачи с учетом имеющихся у нас навыков. Мы определили программные

элементы, которые потребуются для нашей системы, включая процедуры Perl, PHP и MySQL, командные файлы, а также текстовые файлы документации. [Табл. 8.6](#) содержит список всех необходимых элементов проекта вместе с описанием их назначения.

Таблица 8.6. Элементы проекта NCC

| Тип | Элемент | Описание |
|---------------------------|---|--|
| Процедура Perl | ncc.pl | Запускается с помощью cron и выстраивает очередь сканирований, готовых к выполнению |
| Процедура Perl | ncc-client.pl | Удаляет запланированные сканирования из очереди, вызывает команду для их выполнения, а затем осуществляет преобразование для переноса файлов .nbe (по мере их получения) в базу данных MySQL |
| Процедура Perl | ncc-daily.pl | Посылает ежедневные итоговые электронные сообщения и очищает очередь |
| Процедура PHP | Main.php и другие вспомогательные php-файлы | Интерфейс для ввода данных в таблицу расписания; состоит из нескольких файлов |
| Процедура PHP | Reports.php | Интерфейс для просмотра базы данных MySQL, модификация версий NPI; состоит из нескольких файлов |
| База данных MySQL | База данных NCC | Модель базы данных сканирований, внутренняя для программы базы данных MySQL |
| Процедура MySQL | ncc.mysql | Создает начальную базу данных |
| Вспомогательная процедура | install.pl | Процедура для создания элемента для cron, вызов процедуры MySQL, копирование исполнимых файлов в /bin и файла php в Web |
| Текстовый файл | ncc.ini | Переменные окружения для процедур Perl и PHP, имена баз данных, расположение файлов, адреса электронной почты для уведомлений и т.д. |
| Текстовый файл | INSTALL, README и т.д. | Несколько файлов с инструкциями по установке, эксплуатационными инструкциями и другими полезными данными |

Нам предстояло также спроектировать схему базы данных с таблицами, которые будут заполняться нашей программой. Программа NPI была отличным подспорьем в этом отношении, хотя мы добавили новые таблицы, обслуживающие планирование.

Несмотря на то, что потоки данных были аналогичны NPI, имелись и некоторые существенные различия. Мы разработали диаграмму этих потоков, чтобы можно было проследить все логические связи между системами. На [рис. 8.11](#) показана логическая архитектура NCC.

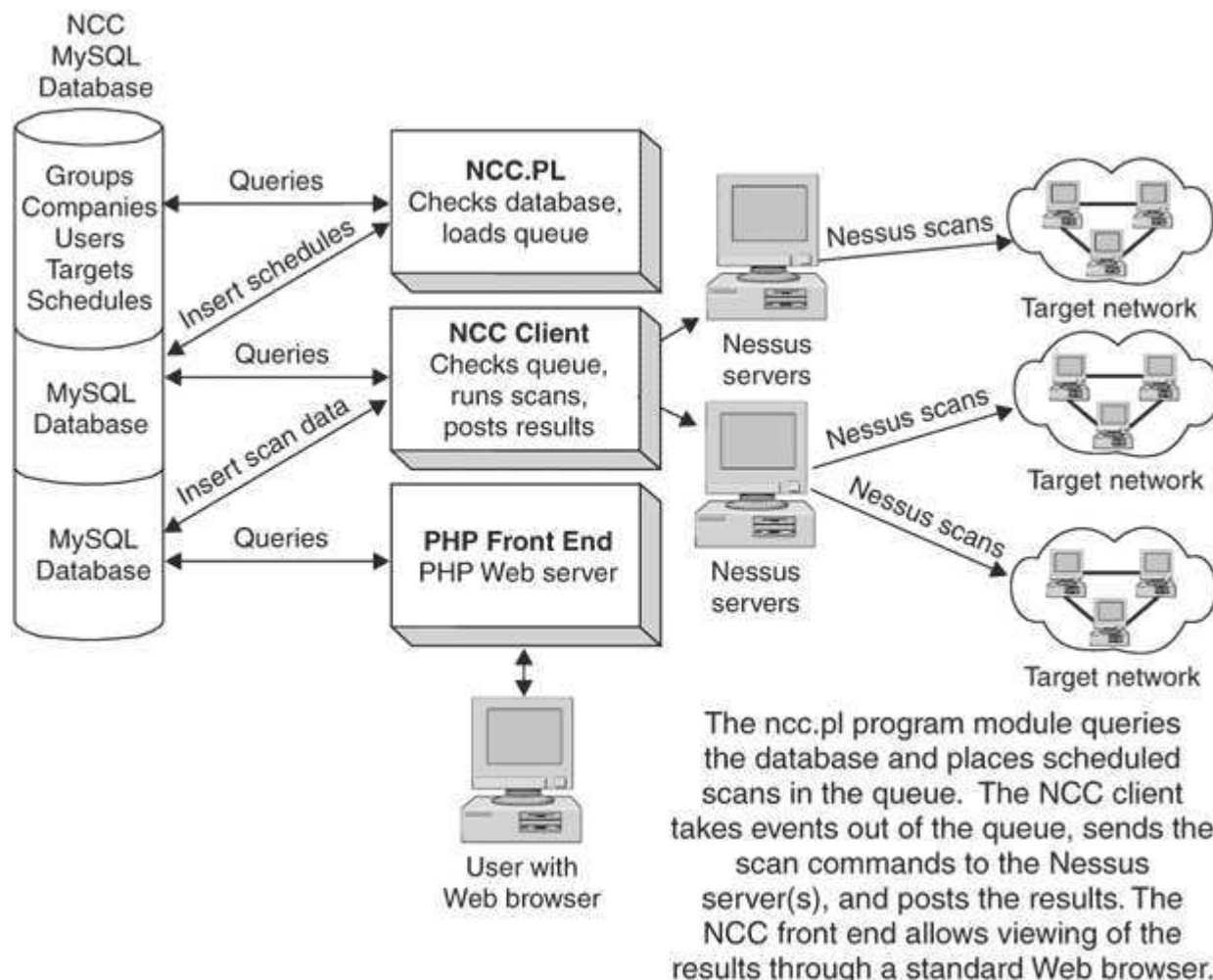


Рис. 8.11. Логическая архитектура NCC

Для проекта мы создали также web-сайт и страницу Sourceforge. Web-страница расположена по адресу <http://www.netsecuritysvcs.com/ncc>. Хотя мы решили, что в нашей группе достаточно талантов для завершения проекта, никогда не мешает познакомить других людей из сообщества открытого ПО с тем, что вы делаете. Кроме того, когда проект будет закончен, может понадобиться помощь при переносе на другие платформы и при добавлении новых возможностей.

Когда все приготовления были закончены, мы начали работу, как правило, проводя еженедельные встречи для отслеживания продвижений. Поскольку это было не основное занятие и у всех имелась другая работа, потребовалось около года для завершения программы в стадии бета-версии. Тем не менее, ее уже можно использовать и теперь, с помощью сетевого сообщества разработчиков, NCC можно расширять и улучшать. Написание NCC как проекта с открытыми исходными текстами, несомненно, требует несколько больше усилий для внешних взаимодействий, чем выполнение его как частного проекта, поскольку необходимо выполнить исследование существующих программ и интегрировать базы исходных текстов, но зато мы смогли воспользоваться существующими исходными текстами, что значительно сократило общее время разработки. Мы также знали, что если наша система станет популярной, ее можно будет перенести на другие платформы или даже использовать в качестве основы для более развитой программы, что нам только помогло бы. В конце концов, как показал опыт, в выигрыше осталась как моя компания, так и внешние пользователи.

Установка NCC

Предварительные требования NCC примерно те же, что и у средства NPI, описанного выше в этой лекции. Необходим поддерживающий PHP web-сервер (такой как Apache), база данных MySQL, клиент и сервер Nessus. Предполагается, что все это уже установлено и работает. Если вы еще этого не сделали, то обратитесь к предыдущим разделам этой лекции, посвященным настройке Apache и MySQL, и к [лекции 5](#) за инструкциями по установке Nessus.

Когда все будет на месте, можно приступить к установке NCC.

- 1. Загрузите программу или возьмите ее с прилагаемого к книге компакт-диска.
- 2. Распакуйте программу в отдельном каталоге, проверив, что он включен в ваш список поиска.
- 3. Перейдите в каталог NCC и наберите ./install.pl. Запустится процедура установки NCC. (NCC не требует компиляции, поскольку он запрограммирован на интерпретируемых языках, таких как Perl и PHP.)

Программа установки сначала проверит присутствие модулей Perl, необходимых NCC. Если она их не найдет, то придется загрузить соответствующие модули либо с дистрибутивных дисков, либо с помощью утилит CPAN, описанных в разделе "Установка Swatch" выше в этой лекции.

- 4. Программа автоматически инициализирует базу данных и скопирует все файлы в подходящие места. Во время установки будет предложено ввести дополнительную информацию. В [табл. 8.7](#) описаны эти параметры установки.

Таблица 8.7. Параметры установки NCC

| Параметр | Описание |
|--------------------------------------|---|
| Пользователь NCC | Системный счет, от имени которого будет выполняться NCC. Рекомендуется создать специальный счет пользователя только для NCC |
| Каталог установки | Можно выбрать один из двух стандартных вариантов, /usr/local/ncs или текущий каталог, либо определить свой собственный |
| Электронный адрес администратора NCC | Адрес электронной почты администратора NCC, на который будут поступать все отчеты о ежедневной активности |
| Адрес отправителя результатов | Адрес, откуда будут посылаться отчеты (важно для фильтрации спама) |
| Имя сервера MySQL | Имя хоста или IP-адрес сервера MySQL для NCC, который должен быть задан как localhost, если сервер MySQL функционирует на той же машине |
| Имя базы данных для NCC | Имя базы данных MySQL, создаваемой процедурой установки. Подразумеваемое значение ncs вполне подходит для большинства установок |
| Пользователь MySQL | Допустимый пользователь системы MySQL, специально предназначенный для NCC |
| Пароль MySQL | Пароль для указанного выше пользователя |
| Сервер Nessus | Имя хоста или IP-адрес сервера Nessus (localhost, если Nessus и NCC выполняются на одной машине) |
| Порт Nessus | Порт для подключения к серверу Nessus. Подразумеваемое значение 1241 годится, если только вы не изменили это значение на сервере Nessus |
| Имя пользователя Nessus | Допустимый пользователь на сервере Nessus |
| Пароль Nessus | Пароль для упомянутого выше пользователя. |
| Маршрут Nessus | Маршрут к исполнимым файлам Nessus. Подразумеваемое значение соответствует стандартной установке Nessus |
| Каталог Temp | Здесь NCC будет накапливать результаты сканирований, прежде чем импортировать их в базу данных. Можно заглянуть в этот каталог, если нужно найти необработанные файлы .nbe, которые были использованы |

- 5. Затем будет запрошена комбинация имени и пароля административного пользователя NCC. Этот пользователь будет администратором всей системы, поэтому тщательно выбирайте имя и пароль.

6. Создайте символическую ссылку в каком-либо из общедоступных web-каталогов, откуда вы хотите иметь доступ к NCC. Направьте ее на подкаталог html в корневом каталоге установки NCC. Это свяжет вас с основной страницей NCC и вашими общедоступными Web-каталогами и защитит от доступа другие файлы NCC.
7. Теперь вы готовы к запуску NCC. Убедитесь, что база данных и web-сервер работают, откройте web-навигатор и введите имя хоста для сервера NCC вместе с именем созданной на предыдущем шаге символической ссылки. Например, если вы назвали символическую ссылку ncc и создали ее в корневом каталоге Web, а сервер NCC имеет имя ncc.example.com, то URL будет выглядеть следующим образом:

<http://ncc.example.com/ncc>

Если вы обращаетесь к NCC на локальной машине, то сработает

<http://localhost/ncc>

Отобразится входная страница NCC.

8. Введите имя пользователя и пароль, заданные в процессе установки.

Теперь можно применять NCC для автоматизации и планирования сканирований.

Применение NCC

После входа отображается основной экран NCC (см. [рис. 8.12](#)). Здесь можно управлять всеми группами, организациями, целями сканирования и расписаниями.

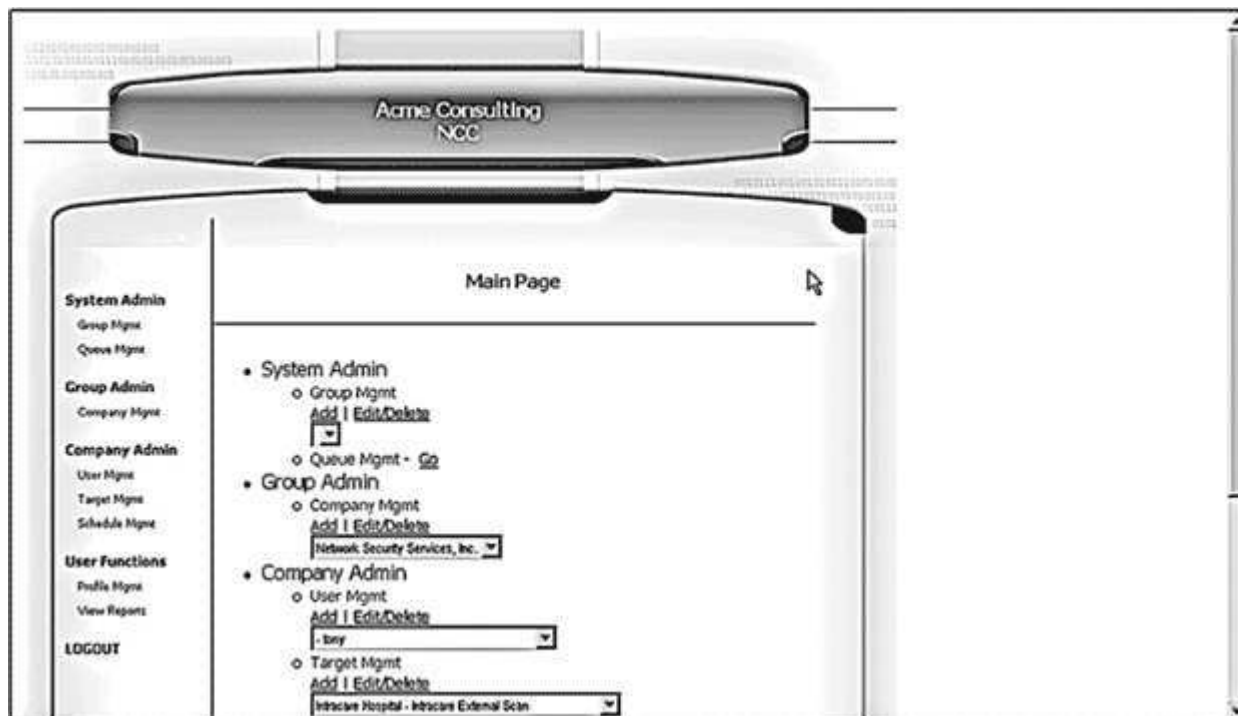


Рис. 8.12. Основной интерфейс NCC

NCC построен как модульная и расширяемая программа. Например, можно применять NCC для управления несколькими сканированиями одной организации. Однако, если вы работаете консультантом, то можете определить сканирования для нескольких организаций с разными профилями. Давайте сделаем еще один шаг и предположим, что вы хотите стать поставщиком услуг приложений безопасности. NCC позволяет задать несколько групп (каждая со своими элементами-организациями) для всех ваших индивидуальных агентов и консультантов, продающих услуги анализа защищенности. (Со временем будут созданы настраиваемые интерфейсы для управления группами, но в бета-версии эти средства отсутствуют.)

Вы можете выбрать из четырех основных опций.

- System admin. Эти опции доступны только системному администратору. Здесь можно создавать группы и выполнять другие функции системного уровня.
- Group admin: Эти опции доступны только администраторам групп, которые могут добавлять, редактировать или удалять групповые профили организаций. Вы будете применять эти функции, например, при задании различных организаций с набором целей, каждой из которых можно управлять. Каждый администратор групп будет видеть только те организации, к которым он имеет доступ.
- Company admin: Здесь вы управляете пользователями, целевыми файлами и расписаниями для каждой организации. Например, вы можете пожелать, чтобы администратор более низкого уровня выполнял сканирование для одного подразделения, но не для другого. Подобные параметры можно задавать здесь.
- User functions: Этот раздел доступен всем пользователям. Здесь отдельные пользователи могут редактировать данные своего профиля и выполнять действия со своими счетами, такие как изменение пароля. Они могут также получить доступ к данным выполненного сканирования.

Возьмем простой пример и пройдемся по всем этапам добавления пользователей, добавления целей и составления расписания сканирований. Для простоты предположим, что вам не требуются средства поддержки нескольких организаций и нескольких групп.

Добавление пользователей

1. Во-первых, необходимо добавить пользователя (отличного от системного администратора, который был добавлен ранее). В разделе Company Admin щелкните мышью на Add под строкой User Mgmt, чтобы добавить пользователя, который может запускать сканирования.
2. Выберите организацию, которой он будет принадлежать, из выпадающего списка, и щелкните мышью на Add.
3. На экране управления пользователями впишите информацию о новом пользователе ([рис. 8.13](#)).

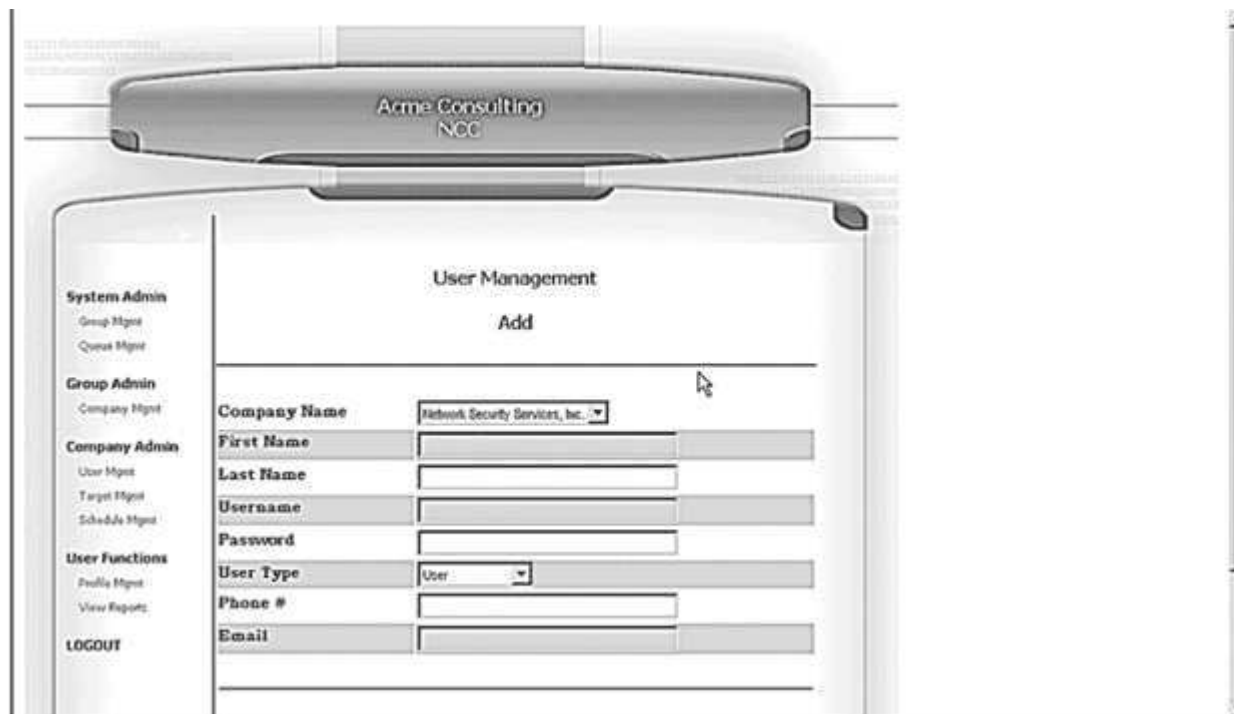


Рис. 8.13. Экран управления пользователями NCC

Здесь можно выбрать имя пользователя и пароль. Пароль при вводе заменяется звездочками и сохраняется как хэш-значение MD5, а не как обычный текст. Выберите здесь также тип пользователя: System admin, Group admin, Company admin или User. Отметим, что вы можете создавать пользователей, которые находятся на том же или нижележащем уровне, что и вы. Например, администраторы организаций не могут создавать пользователей уровня системного администратора.

Если вы хотите отредактировать или удалить существующего пользователя, щелкните мышью на Edit/delete на основном экране в разделе управления организациями (Company Management).

4. Щелкните мышью на Add, и NCC добавит в базу данных пользователя, который может теперь входить в систему и добавлять сканирования как сотрудник организации, к которой он был приписан.

Добавление целей

В NCC цель определяется как произвольный набор IP-адресов и ассоциированных настроек сканирования для этих адресов. При проектировании программы мы приняли сознательное решение разделить объекты целей и объекты расписаний. Это делает программу более модульной и повышает ее гибкость. Например, вы можете запланировать определенное сканирование на начало каждого месяца. Однако, если возникает новая уязвимость, вы можете пожелать однократно просканировать данную цель в середине месяца, чтобы проверить выявленную уязвимость. NCC позволяет добавлять событие одноразового сканирования для данной цели, вместо того, чтобы изменять, а затем восстанавливать ежемесячное сканирование в прежнем виде.

1. Чтобы добавить цель, щелкните мышью на Target Mgmt в разделе Company Admin основного экрана.
2. Раскройте контекстное меню, чтобы увидеть все цели, к которым вы имеете доступ. Если вы являетесь администратором группы, то вам покажут все цели для каждой организации, которой вы принадлежите.
3. Щелкните мышью на Add, появится экран управления целями ([рис. 8.14](#)). Здесь можно выбрать организацию, для которой добавляется цель.

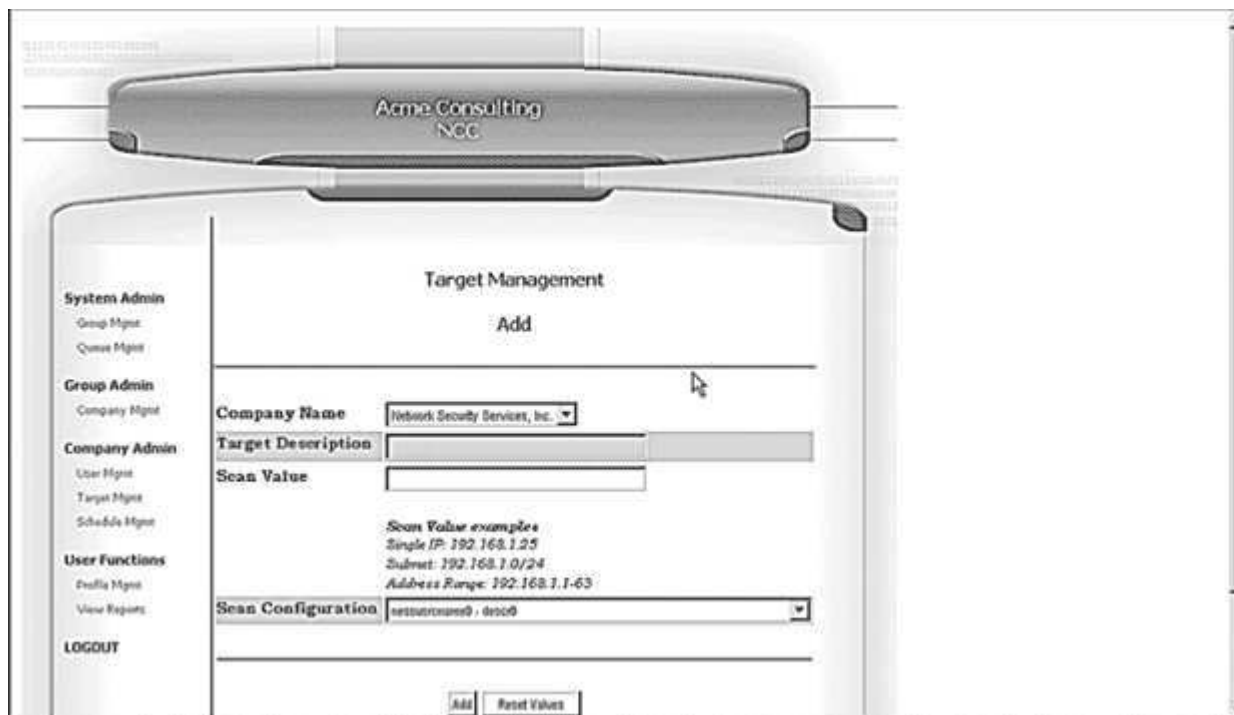


Рис. 8.14. Управление целями в NCC

Задайте для цели текстовое описание, например "серверы в демилитаризованной зоне". Это описание появится в раскрывающемся меню, поэтому оно должно быть достаточно содержательным.

4. Выберите тип сканирования: будет ли сканироваться один адрес, подсеть или диапазон адресов.
5. В поле Scan Value введите цепочку целевых IP-адресов в синтаксисе, который поддерживает Nessus. Напомним (см. [лекцию 5](#)) допустимые в Nessus форматы адресов:

| | |
|---|--|
| Один IP-адрес | 192.168.0.1 |
| IP-адреса, разделенные запятыми | 192.168.0.1,192.168.0.2 |
| IP-диапазон - пара адресов, разделенных дефисом | 192.168.0.1-192.168.0.254 |
| Стандартная нотация с косой чертой | 192.168.0.1/24 (сеть класса C из 256 адресов) |
| Имя хоста | myhost.example.com |
| Произвольная комбинация вышеприведенных элементов, разделенных запятыми | 192.168.0.1-192.168.0.254,195.168.0.1/24,192.168.0.1-192.168.0.254 |

6. Выберите конфигурацию сканирования. По умолчанию используется сканирование Nessus. Имеется до четырех других типов сканирования. (В последующих версиях будет разрешена выгрузка файла индивидуальной конфигурации и вставка его в текстовый файл.)
7. Щелкните мышью на Add, и цель будет добавлена. Теперь все готово для составления расписания сканирований.

Составление расписания сканирований

Когда созданы объекты целей, для них можно определить расписание сканирований.

1. В основном меню в разделе Company Admin щелкните мышью на Schedule Management. Появится экран управления расписанием ([рис. 8.15](#)).

System Admin
Group Mgmt
Queue Mgmt

Group Admin
Company Mgmt

Company Admin
User Mgmt
Target Mgmt
Schedule Mgmt

User Functions
Profile Mgmt
View Reports

LOGOUT

Schedule Management
Add

Schedule Description: sched_20040408_215814

Target Selection: NDS Internal Network

To add/edit/delete targets go to:

Start Now: ☐

Start Date: 28 Apr 2004

Start Time: 21:45

NOTE: This time will be used for all schedule types.

Run Once: ☐

Run Daily: ☐

Run Weekly: ☐ Sun Mon Tue Wed Thu Fri Sat
C C C C C C C

Run Monthly: ☐ Day of Month: 01

Finish Date: 28 Apr 2005

Add Reset Values

Рис. 8.15. Экран управления расписанием в NCC

2. Выберите организацию и цель в этой организации. Здесь также доступно раскрывающееся меню выбора, отражающее уровень пользователя, под именем которого вы вошли в систему.
3. Выберите дату сканирования, время, частоту, число повторов.

Можно выполнять сканирование один раз, ежедневно, еженедельно, ежемесячно, раз в два месяца, ежеквартально (в последующих версиях будут поддерживаться индивидуальные последовательности повторений в формате cron или I-cat). Можно также задать, чтобы повторение происходило только определенное количество раз, например, для заказчика, который подписал годовой контракт на ежемесячное сканирование. Допускается неограниченное число повторов, например, регулярное ежемесячное сканирование вашей собственной сети.

4. Щелкните мышью на Add, и расписание для данного сканирования будет сформировано.

Теперь можно спокойно сидеть и ждать отчетов. Пользователь, создавший сканирование, будет уведомляться по электронной почте за день до его выполнения (за исключением ежедневных сканирований, для которых уведомление происходит за час до начала). Еще одно электронное сообщение он получит, когда отчет будет готов для просмотра.

5. Когда сканирование выполнено, можно просмотреть его результаты, выбирая View reports в разделе User Functions основного меню. Отобразится экран базы данных сканирований NCC ([рис. 8.16](#)).

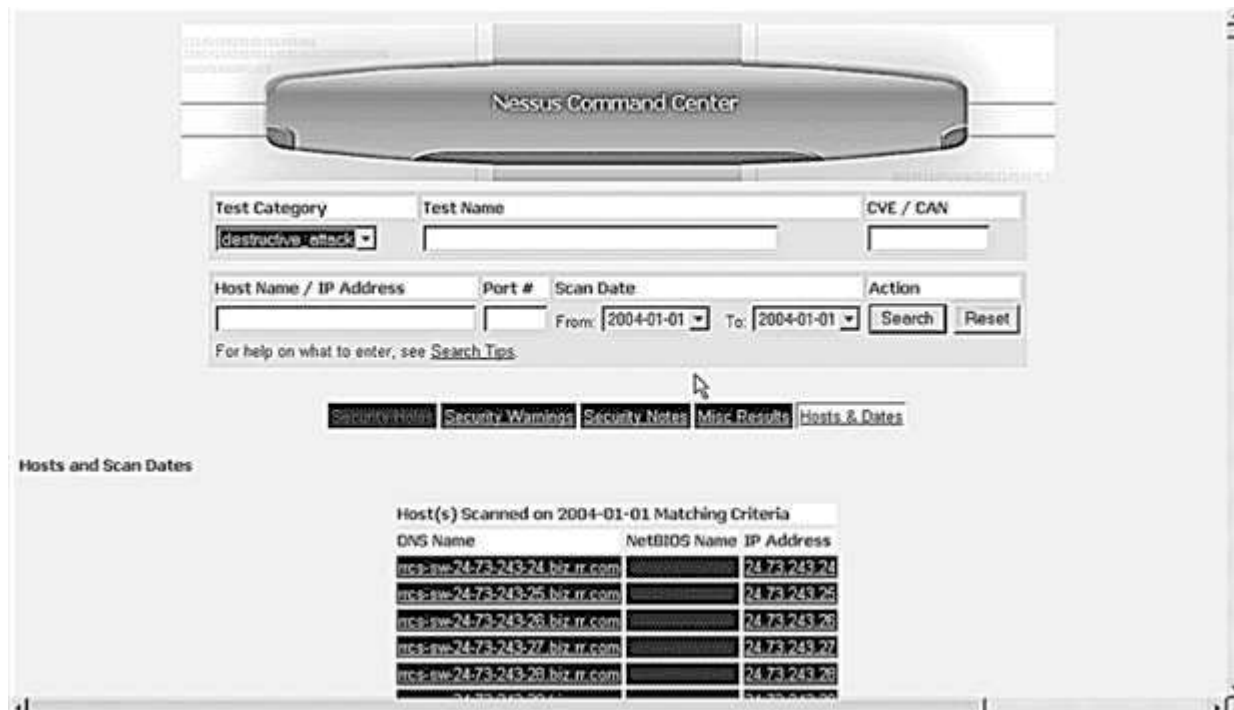


Рис. 8.16. Представление базы данных сканирований в NCC

Можно просматривать данные сканирований и генерировать индивидуализированные отчеты для заказчиков.

Заметим, что этот интерфейс похож на интерфейс NPI, который был рассмотрен выше в этой лекции. Это объясняется тем, что при создании данного раздела мы использовали фрагменты NPI. NPI имеет открытые исходные тексты с лицензией GPL, поэтому, если мы выпускаем программу с лицензией GPL и включаем информацию об авторских правах, то мы имеем право заимствовать эти тексты. При использовании открытых исходных текстов вы имеете полный доступ к любым их продвижениям и усовершенствованиям.

Может показаться, что для простого сканирования требуется слишком много работы, и это действительно так, если оно выполняется только один раз. Но когда вы управляете десятками сканирований при участии нескольких пользователей, то NCC может оказаться бесценным инструментом для контроля всей этой деятельности.

И так, у вас есть средства и знания для создания законченной системы обнаружения вторжений и сканирования уязвимостей с развитыми аналитическими возможностями. Применяя этот инструментарий, вы сможете существенно повысить безопасность вашей внутренней сети и внешних серверов. В совокупности эти средства позволяют наиболее эффективно использовать время, затрачиваемое на повышение безопасности сети. Теперь мы приступаем к рассмотрению средств шифрования, помогающих обеспечивать безопасность ваших данных внутри и вне вашей сети.

9. Лекция: Криптографические средства: версия для печати и PDA

Рассмотренные до сих пор средства применялись для защиты сетей и машин, располагающихся в этих сетях. Однако, когда данные выходят за границы сети, они оказываются вне защиты описанных средств и потенциально могут быть перехвачены злоумышленниками. Большинство современных Интернет-приложений передают свои данные в открытом виде (открытым текстом). Это означает, что при просмотре пакетов любой желающий может видеть данные. Когда данные пересекают Интернет, они проходят через различные системы, большинство из которых находятся вне вашего непосредственного контроля и поэтому должны считаться недружественными. Маршрутизаторы и коммутаторы поставщиков Интернет-услуг могут применяться как внутри, так и вне вашей сети, а почтовые и web-серверы стандартным образом обрабатывают ваши приватные данные.

Не существует способа избежать отправки данных за пределы вашей сети. Основное преимущество глобальности Интернета - возможность совместного использования информации со всеми бизнес-партнерами и заказчиками из внешнего мира. Невозможно вернуться во времена полностью собственных сетей. Как же защитить важные данные, когда они покидают уютные и безопасные пределы домашней сети? Чтобы обезопасить свои данные в Интернете, большинство организаций полагаются на криптографию, и вы также можете применять это важное средство для поддержания целостности и конфиденциальности.

Вам может понадобиться защитить данные от несанкционированного просмотра и в вашей сети, поскольку не всякая информация должна быть доступна для всеобщего обозрения даже в пределах организации. Наконец, шифрование важных данных может служить последней линией обороны против хакеров. Даже если им удастся проникнуть в сеть и подчинить себе сервер, им придется взломать шифр, чтобы добраться до ваших данных.

Обзор лекции

Изучаемые концепции:

- Симметричная и асимметричная криптография
- Различные криптографические алгоритмы
- Криптографические приложения
- Модель безопасности с удостоверяющим центром
- Модель безопасности с сетью доверия

Используемые инструменты:

PGP, GnuPG, OpenSSH, FreeS/WAN, John the Ripper

Имеется множество различных криптографических протоколов. Обратившись к эталонной модели ВОС ([табл. 9.1](#)), можно видеть, что существуют криптографические средства, действующие на нескольких различных уровнях модели. Как вы, вероятно, догадались, доступно много прекрасных криптографических средств с открытыми исходными текстами почти для любых приложений, от шифрования отдельных файлов до защиты всех ваших исходящих Интернет-соединений. На самом деле, доступность высококачественного криптографического программного обеспечения коренится в движении за открытость исходных текстов.

Таблица 9.1. Модель ВОС и криптография

| Номер уровня модели ВОС | Название уровня | Примеры протоколов |
|-------------------------|-----------------------|--------------------|
| Уровень 7 | Прикладной уровень | PGP, GnuPG |
| Уровень 6 | Уровень представления | |

| | | |
|-----------|----------------------|----------|
| Уровень 5 | Уровень сеанса | SSL, SSH |
| Уровень 4 | Транспортный уровень | |
| Уровень 3 | Сетевой уровень | IPsec |
| Уровень 2 | Канальный уровень | |
| Уровень 1 | Физический уровень | |

Виды криптографии

На сегодня имеется два основных метода шифрования. Первый метод, именуемый симметричным или шифрованием с разделяемым секретом, применялся со времен древнего Египта. В нем секретный ключ, называемый разделяемым секретом, используется для превращения данных в непонятную тарабарщину. Второй стороне разделяемый секрет (ключ) требуется для расшифрования данных в соответствии с криптографическим алгоритмом. Если изменить ключ, изменятся и результаты шифрования. Это называется симметричной криптографией, поскольку один и тот же ключ применяется обеими сторонами как для зашифрования, так и для расшифрования данных ([рис. 9.1](#)).

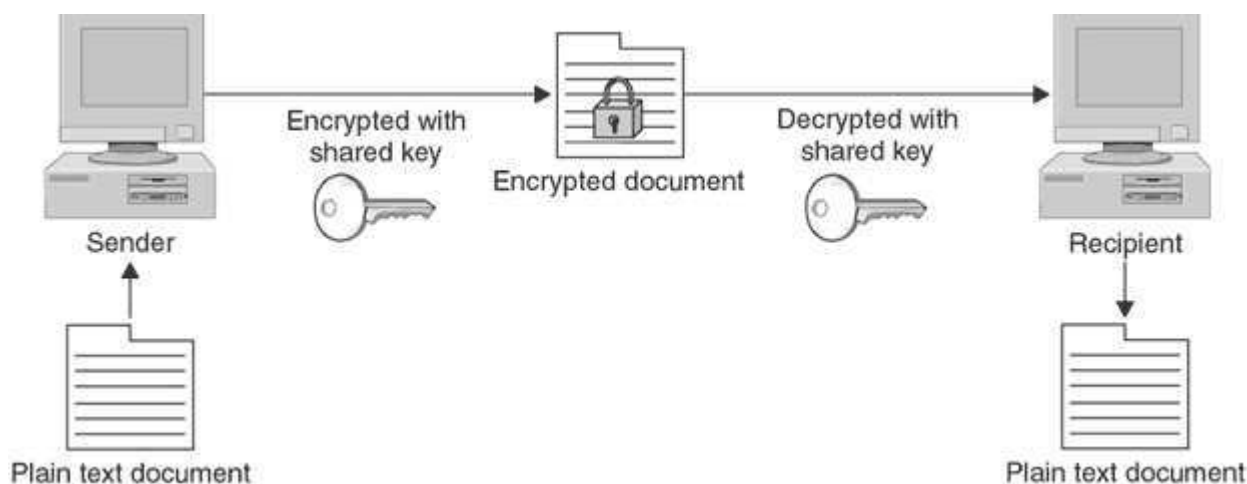


Рис. 9.1. Симметричная криптография

Проблема с этим методом состоит в том, что необходимо безопасным образом передать секретный ключ предполагаемому получателю. Если враг перехватит ключ, он сможет прочитать сообщение. Изобретались всевозможные системы с целью обойти это фундаментальное слабое место, но факт остается фактом: требуется каким-то образом передавать секретный ключ предполагаемому получателю, прежде чем можно будет начинать защищенное взаимодействие.

Революция в криптографии началась, когда Витфилд Диффи, Мартин Хеллман и Ральф Меркл изобрели криптографию с открытым ключом. (Некоторые утверждают, что в действительности британский гражданский служащий Джеймс Эллис сделал это раньше и держал в секрете, но Диффи, Хеллману и Мерклу принадлежит первая публикация, датированная 1976-м годом.) Они пытались решить старую проблему обмена ключами. Диффи интересовало, как два человека, желающие осуществить финансовую транзакцию через электронную сеть, могут сделать это безопасным образом. Он думал о далеком будущем, так как Интернет тогда был в зачаточном состоянии, а электронной коммерции еще не существовало. Если правительственные организации имеют проблемы при обмене ключами, то как может справиться с этим рядовой гражданин? Он хотел построить систему, с помощью которой две стороны могли бы легко поддерживать защищенные коммуникации и безопасные транзакции, не обмениваясь каждый раз ключами. Он знал, что если он сможет решить проблему обмена ключами, то это станет прорывом в криптографии.

Диффи сотрудничал с Мартином Хеллманом и Ральфом Мерклом. Им потребовалось несколько лет, но в конце концов они создали систему, называемую шифрованием с открытым ключом (PKE - Public Key Encryption), известную также как асимметричная криптография.

В асимметричной криптографии применяется шифрование, при котором некая величина расщепляется на два ключа меньшего размера. Один из них делается открытым, а другой сохраняется в секрете. Вы шифруете сообщение с помощью открытого ключа получателя. Получатель может затем расшифровать его с помощью своего секретного ключа. И он может сделать то же самое для вас, шифруя сообщение с помощью вашего открытого ключа, чтобы вы могли расшифровать его с помощью своего секретного ключа (рис. 9.2). Суть в том, что не требуется знать чей-то секретный ключ, чтобы послать защищенное сообщение. Применяется открытый ключ, который не нужно держать в секрете (на самом деле, его можно публиковать наравне с номером телефона). Используя открытый ключ получателя, вы знаете, что только этот человек может расшифровать сообщение при помощи своего секретного ключа. Данная система позволяет двум сущностям безопасно общаться без какого-либо предварительного обмена ключами.

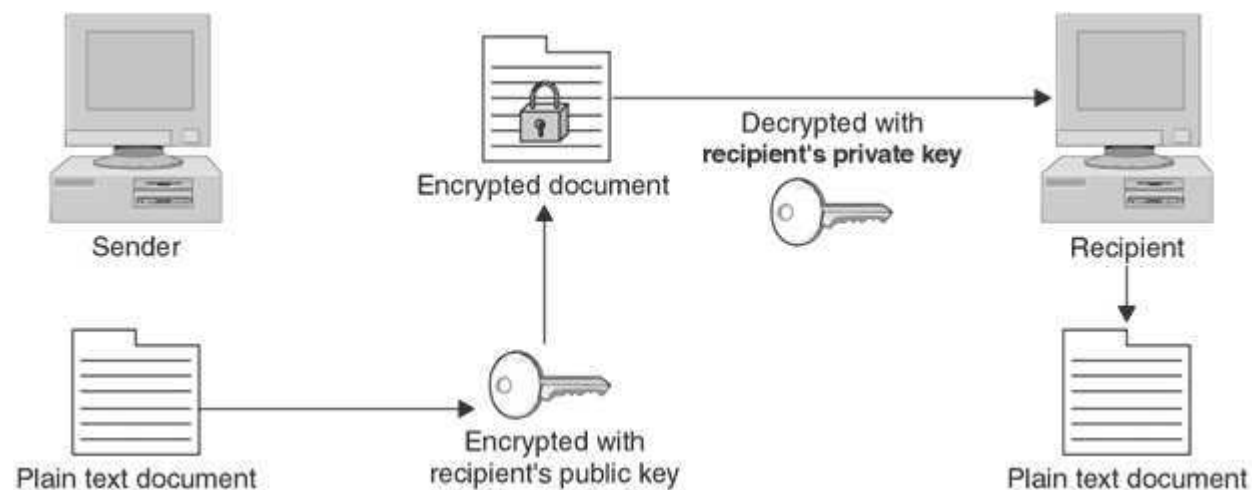


Рис. 9.2. Асимметричная криптография (открытый ключ)

Асимметричная криптография обычно реализуется с помощью односторонних функций. В математических терминах это функции, которые очень легко вычислять в одном направлении, но очень сложно - в обратном. Именно это позволяет публиковать открытые ключи, которые являются производными от секретных ключей. Очень трудно выполнить обратное преобразование и определить секретный ключ. На сегодняшний день наиболее употребительной односторонней функцией является перемножение больших простых чисел. Очень легко перемножить два больших простых числа и получить произведение, однако определение того, каким из множества возможных способов это произведение раскладывается на два множителя, является одной из трудных математических задач. Если бы кто-то изобрел метод быстрого определения множителей подобных больших чисел, это поставило бы крест на многих современных методах шифрования с открытым ключом. К счастью, имеются и другие трудновычислимые функции, обратные к легковычислимым. Например, пока неизвестны быстрые алгоритмы для логарифмирования в конечном поле или в группе точек эллиптической кривой над конечным полем, тогда как возведение в степень в обоих этих случаях выполнить очень легко.

Вскоре после публикации Диффи, Хеллмана и Меркла другая группа из трех человек разработала практическое приложение теории. Эта система для шифрования с открытым ключом была названа RSA по именам авторов: Ronald Rivest (Рональд Ривест), Adi Shamir (Ади Шамир) и Leonard Adleman (Леонард Адлеман). Они образовали компанию и начали лицензировать свою систему. Дело шло туго, и их компания почти обанкротилась, пока они не договорились с малоизвестной тогда компанией Netscape об использовании возможностей растущего поля Интернет-коммерции. Остальное уже история, и RSA на сегодняшний день - самый употребительный алгоритм шифрования с открытым ключом. Диффи и Хеллман со временем выпустили собственное практическое приложение, но оно применяется только для обмена ключами, в то время как RSA - для аутентификации и обеспечения неотказуемости.

Шифрование с открытым ключом присутствует теперь на каждом Web-сервере, предлагающем безопасные покупки. Ваша транзакция шифруется без передачи или получения секретного ключа, и все это происходит в фоновом режиме. Как пользователи, мы знаем, что в навигаторе появился маленький символ замка SSL, и мы чувствуем себя в безопасности. Нетрудно представить себе судьбу Интернет-коммерции, если бы при каждой покупке в Сети

приходилось думать о секретном ключе, шифровать сообщение, а затем как-то передавать этот ключ другой стороне. Очевидно, что без криптографии с открытым ключом электронная коммерция в ее современном виде не могла бы существовать.

Имеется много различных алгоритмов шифрования, протоколов и приложений на основе этих двух основных видов криптографии. В следующих разделах вы ознакомитесь с некоторыми из них.

Криптографические алгоритмы

В наше время сила криптографии обычно характеризуется размером ключа. Независимо от силы алгоритма, зашифрованные данные могут подвергаться атакам методом грубой силы, пробуящим всевозможные комбинации ключей. Со временем шифр может быть взломан. Для большинства современных шифров с подходящей длиной ключа время их взлома методом грубой силы измеряется тысячелетиями. Однако неизвестный дефект алгоритма, достижения в компьютерной технологии или математических методах могут резко сократить это время.

Обычно считается, что длина ключа должна быть достаточной для сохранения защиты данных на разумный период времени. Если речь идет о короткоживущих данных, таких как коммуникации на поле боя или ежедневная биржевая информация, то вполне достаточно шифра, обеспечивающего защиту в течение недель или месяцев. Однако некоторые вещи, такие как номер кредитной карты или секреты национальной безопасности, необходимо сохранять защищенными гораздо дольше, по сути - навсегда. Поэтому применение слабых алгоритмов шифрования или коротких ключей для некоторых данных приемлемо, только если полезность информации для посторонних теряется за короткое время.

Стандарт шифрования данных DES (Data Encryption Standard)

DES является исходным стандартом, который правительство США рекомендовало для правительственных и коммерческих применений. Первоначально в 1970-е годы он считался практически невскрываемым, но с ростом вычислительной мощности и снижением стоимости вычислений его 56-битный ключ функционально устарел для высокосекретной информации. Тем не менее, он все еще применяется во многих коммерческих продуктах и считается приемлемым для приложений с умеренным уровнем безопасности. Он используется также в продуктах со слабыми процессорами, таких как смарт-карты и бытовые приборы, неспособных обрабатывать более длинные ключи.

Тройной DES

Тройной алгоритм DES (TripleDES или 3DES, как его часто записывают) - более новая, усовершенствованная версия DES, а его название отражает его функциональность. Он трижды применяет DES к данным, выполняя зашифрование, расшифрование и затем снова зашифрование. В действительности он не обеспечивает трехкратного усиления шифра (так как в нем задействованы всего два ключа - первый применяется дважды для зашифрования, а второй служит для расшифрования результатов первого зашифрования), но он, тем не менее, предоставляет эффективную длину ключа в 112 бит, что более чем достаточно почти для всех пользователей.

RC4, RC5 и RC6

Это алгоритм шифрования, разработанный Рональдом Ривестом, одним из создателей RSA, первого коммерческого приложения криптографии с открытым ключом. Со временем были сделаны усовершенствования, чтобы усилить его и исправить некоторые недочеты. Текущая версия RC6 допускает длину ключа до 2040 бит и переменный размер блока до 128 бит.

AES

Когда правительство США осознало, что DES со временем достигнет конца своей полезной жизни, оно стало искать ему замену. Национальный институт стандартов и технологий США - правительственный орган стандартизации - объявил открытый конкурс на новый алгоритм, призванный стать новым правительственным стандартом. Претендентов было много, включая RC6, Blowfish известного криптографа Брюса Шнайера и другие достойные алгоритмы. Победителем стал AES, базирующийся на алгоритме Rijndael, разработанном двумя бельгийскими криптографами. То, что стандарт был выбран в результате открытого состязания, весьма примечательно, как и победа двух неамериканских разработчиков, позволившая без значительных финансовых вложений обеспечить ему мировое признание. AES быстро становится новым стандартом шифрования. Он предлагает ключ шифрования до 256 бит, что представляется более чем достаточным для обозримого будущего. Обычно AES из соображений эффективности реализуется в режиме 128 или 192 бита.

Приложения криптографии

Хэши

Хэши служат специальным применением односторонних функций для аутентификации и верификации криптографическими средствами. Исходный файл пропускается через хэш-функцию, в результате чего порождается значительно меньший файл фиксированного размера с уникальными идентификационными признаками исходного файла. В дальнейшем это позволяет удостовериться, что файл никаким образом не был изменен. Хэшируя подозрительный файл и сравнивая результат с заведомо хорошим значением, можно определить, были ли внесены какие-либо изменения. Маловероятно, что файлы с различной структурой будут порождать идентичные хэши. Даже изменение одного символа существенно меняет хэш. Вероятность того, что два различных файла дадут одинаковый хэш, пренебрежимо мала.

Хэшами часто снабжают загружаемые версии программного обеспечения, чтобы гарантировать получение подлинника. Это важно, особенно для программ с открытыми исходными текстами, которые могут попасть к вам через третьи руки или с "неродного" сайта. На официальном web-сайте обычно публикуется правильный хэш самой свежей версии. Если два значения не совпадут, то можно утверждать, что были внесены некоторые изменения, возможно, без разрешения или уведомления разработчиков программы. Наиболее популярный алгоритм хэширования называется MD5.

Цифровые сертификаты

Цифровые сертификаты - это "подпись" мира Интернет-коммерции. Они применяются для аутентификации. Они удостоверяют, что тот, с кем вы соединяетесь, действительно тот, за кого себя выдает. Проще говоря, сертификат служит "удостоверением" источника данных, содержащим открытый ключ организации. Хэш сертификата шифруется с помощью секретного ключа этой организации или уполномоченной организации - удостоверяющего центра (последнее предпочтительнее). Зная открытый ключ, парный секретному, вы можете проверить подлинность сертификата и тем самым удостовериться в принадлежности web-сайта интересующей вас организации.

Сертификаты обычно приписываются определенному домену. Они могут выпускаться удостоверяющим центром или локально, как описано выше. Имеется несколько удостоверяющих центров, самым крупным из которых является VeriSign - компания, поддерживающая также доменную систему имен. Удостоверяющие центры наделяют другие организации правом предлагать сертификаты под их ответственность. Получить сертификат от VeriSign или одной из авторизованных организаций - все равно, как если бы кто-то поручился за вас. Обычно сертификат выпускают только после проверки фигурирующих в нем данных, осуществляемой либо по телефону, либо с помощью некоторой бумажной документации, такой как устав организации. Когда вас "освидетельствуют", удостоверяющий центр берет представленную вами информацию, включая URL, для которого вы собираетесь применять сертификат, и "подписывает" ее цифровым образом с помощью своего секретного ключа. После этого Web-сервер или другая программа может использовать выданный сертификат. Когда внешние пользователи получают с сервера некоторые данные, например, Web-страницу, с присоединенным к ним сертификатом, они могут применить криптографию с открытым ключом для проверки вашей личности. Чаще всего сертификаты используются на Web-сайтах электронной коммерции, но они могут также применяться для коммуникаций произвольного вида. SSH и Nessus могут использовать сертификаты для аутентификации. В виртуальных защищенных сетях сертификаты также могут применяться для аутентификации вместо паролей.

Криптографические протоколы

IPsec

Хорошо известно, что при первоначальном проектировании IP-протокола вопросам безопасности уделяли не слишком много внимания. IP версии 4 (IPv4), доминирующий протокол IP-коммуникаций, не предоставляет никаких средств аутентификации или конфиденциальности. Полезная нагрузка пакетов посылается в открытом виде, а заголовки пакетов можно легко изменять, так как они не проверяются в месте назначения. Многие атаки эксплуатируют эту базовую незащищенность инфраструктуры Интернет. Новый стандарт IP, называемый IPv6, был разработан для обеспечения аутентификации и конфиденциальности с помощью криптографии. Он также расширяет адресное пространство IP, используя 128-битный адрес вместо применяемого сейчас 32-битного, и усовершенствован в ряде других направлений.

Полная реализация стандарта IPv6 потребует широкомасштабной модернизации оборудования, поэтому развертывание IPv6 происходит довольно медленно. Однако была предложена реализация средств безопасности для IP, называемая IPsec, не требующая значительных изменений в схеме

адресации. Производители оборудования перешли на IPsec, постепенно ставший фактическим стандартом для создания виртуальных защищенных сетей в Интернете.

IPsec - это не конкретный криптографический алгоритм, а скорее криптографический каркас для шифрования и верификации пакетов в протоколе IP. Спецификации IPsec предусматривают использование различных алгоритмов и могут быть реализованы полностью или частично. Комбинация асимметричной и симметричной криптографии применяется для шифрования содержимого пакетов, а хэширование добавляет к этому аутентификацию. Данная функция называется протоколом аутентифицирующего заголовка - Authentication Header, AH). С помощью AH создается и передается хэш IP-заголовка. Когда пакет прибывает в место назначения, хэш его заголовка перевычисляется. Если полученное значение не совпало с присланным, значит, в процессе пересылки заголовок был изменен. Тем самым обеспечивается высокая степень доверия к подлинности исходного адреса. Можно шифровать содержимое пакетов без добавления аутентифицирующего заголовка, чтобы не снижать пропускную способность и избежать проблем с трансляцией сетевых адресов и межсетевыми экранами. Имеется два различных режима работы IPsec: туннельный и транспортный.

В туннельном режиме весь пакет - заголовок и все остальное - шифруется, помещается (инкапсулируется) в другой пакет и переправляется по одному из туннелей виртуальной защищенной сети. В конечной точке этого туннеля пакет извлекается, расшифровывается и переправляется на правильный IP-адрес. Преимущество этого метода состоит в том, что посторонний наблюдатель не может даже определить пункт назначения зашифрованного пакета. Другое преимущество - возможность централизованного управления и администрирования виртуальной защищенной сети. Недостаток заключается в том, что для туннелирования требуется специально выделенное оборудование на обоих концах.

В транспортном режиме шифруется только полезная нагрузка пакетов; заголовки посылаются без изменений. Это упрощает инфраструктуру и несколько облегчает ее развертывание. Отметим, что транспортный режим можно сочетать с протоколом AH и верифицировать исходные адреса пакетов.

Протокол туннелирования точка-точка (PPTP - Point-to-Point Tunneling Protocol)

PPTP - стандарт, разработанный Microsoft, 3Com и другими большими компаниями для обеспечения шифрования. Корпорация Microsoft включила его в Windows 98 и последующие выпуски, сделав вероятным кандидатом на роль основного стандарта массовой криптографии. Однако в PPTP были обнаружены существенные дефекты, что ограничило его применение. Когда Microsoft включила поддержку IPsec в Windows 2000, это можно было считать молчаливым признанием победы IPsec как нового криптографического стандарта. Однако PPTP все-таки является полезным и недорогим протоколом для создания виртуальных защищенных сетей ПК со старыми версиями Windows.

Протокол туннелирования второго уровня (L2TP - Layer Two Tunneling Protocol)

Еще один индустриальный протокол, поддержанный компаниями Microsoft и Cisco. Хотя он часто используется в аппаратных устройствах шифрования, его применение в программном обеспечении довольно ограничено.

Защищенный протокол сеансового уровня (SSL - Secure Socket Layer)

Этот протокол специально предназначен для применения в Web, хотя он может использоваться почти для любого типа TCP-коммуникаций. Первоначально компания Netscape разработала его для своего браузера, чтобы помочь развитию электронной коммерции. SSL, опираясь на сертификаты, обеспечивает шифрование данных, аутентификацию обеих сторон и контроль целостности сообщений. В основном SSL работает в фоновом режиме при соединении с web-сервером для защиты пересылаемой информации, и его присутствие мало кто осознает. Обычно он аутентифицирует только одну сторону - серверную, так как у большинства конечных пользователей нет сертификатов.

Криптографические приложения

Фил Циммерман - программист и активный борец за права человека. Его тревожило, что все более широкое применение компьютеров и коммуникационных сетей облегчает органам национальной безопасности репрессивных режимов перехват и сбор информации о диссидентах. Фил хотел написать программное обеспечение, помогающее этим людям сохранять свою информацию в тайне и безопасности. Подобное ПО могло бы в буквальном смысле спасать человеческие жизни. Он также не вполне доверял собственному правительству. Он знал, как легко правительство может создать системы для поиска определенных ключевых слов в любых электронных сообщениях. Он хотел дать людям способ защиты и гарантии их конституционного права на тайну частной жизни.

Он назвал свою программу "Приятное уединение" (PGP - Pretty Good Privacy), так как считал, что проделал хорошую работу для защиты данных от разведок небольших стран. Однако агентство по информационной безопасности США считало по-другому. Циммерман был обвинен в нарушении федеральных законов об экспорте вооружений за предоставление возможности загружать свою программу из любой точки мира.

Первоначально Циммерман собирался учредить компанию для продажи своего изобретения. Однако, когда правительство стало его преследовать, он бесплатно распространял свое программное обеспечение через Интернет, сделав его общедоступным. Впоследствии он все-таки сформировал компанию для продвижения коммерческих версий программного обеспечения, но повсюду в Интернете имеются реализации PGP с открытыми исходными текстами. Некоторые из них более популярны, чем другие, а некоторые являются нишевыми приложениями, такими как шифрование электронной почты. В следующем разделе рассматривается официальная условно свободная версия от PGP Corporation, а также версия с полностью открытыми исходными текстами. Список всех реализаций PGP можно найти по адресу <http://www.cypherspace.org/openpgp/>.

PGP Freeware: Средство криптографии с открытым ключом

PGP Freeware

Автор/основной контакт: Phil Zimmerman

Web-сайт: <http://www.pgp.com/>

Платформы: Несколько платформ, включая все Windows и Linux

Лицензия: Бесплатно для некоммерческого применения

Рассмотренная версия: 8.0.2

Другие ресурсы:

<http://www.pgpi.com/>

Списки почтовой рассылки:

Группа поддержки PGP Freeware

Рабочая группа IETF OpenPGP

Пользовательский список рассылки PGP

Рабочая группа PGP/MIME

Список рассылки разработчиков PGPi

Список рассылки переводчиков PGPi

Список рассылки разработчиков Pgpplib

Все эти списки доступны для подписки по адресу:

<http://www.pgpi.org/links/maillinglists/en/>.

Телеконференции USENET:

Alt.security.pgp

Comp.security.pgp.announce

Comp.security.pgp.discuss

Comp.security.pgp.resources

Comp.security.pgp.tech

Официальную условно свободную версию PGP поддерживает Массачусетский технологический институт. Так как она лицензирована у Фила Циммермана и PGP Corporation, то можно не сомневаться в ее целостности и законности. Недостатком условно свободной версии PGP является то, что она лицензирована только для индивидуального применения, поэтому ее можно использовать для персональной электронной почты или в целях образования, если вы студент. Если вы собираетесь применять эту версию PGP, не забудьте внимательно прочитать лицензию. Хотя эта версия PGP имеет открытые исходные тексты и распространяется бесплатно, существуют значительные ограничения на ее использование. Помните, что открытость исходных текстов не обязательно означает бесплатность. Если вы желаете получить самую свежую версию в сочетании с простотой использования и поддержкой, то должны рассмотреть возможность покупки полной лицензии у PGP Corporation. Она стоит примерно \$125 на одного пользователя. При массовых закупках предоставляется скидка. Если вы не можете или не хотите платить, то более интересным для вас может оказаться другое средство - GnuPG, полностью свободная реализация PGP.

Официальная версия PGP от PGP Corporation обладает некоторыми замечательными возможностями:

- Встроенный клиент для виртуальных защищенных сетей на основе IPsec 3DES, пригодный для безопасных коммуникаций с любым партнером, имеющим PGP версии 8.0 или выше.
- Возможность создания саморасшифровывающихся архивов для отправки сообщений PGP тем, у кого нет установленного программного обеспечения PGP.
- Затиранье удаленных файлов, то есть возможность удалить файл с последующей многократной перезаписью области данных на диске.
- Затиранье свободного пространства, аналогичное затиранью удаленных файлов, но для свободного дискового пространства, которое может содержать следы старых данных.
- Интегрированная поддержка командной строки для тех, кто знаком со старыми командами.
- Встраиваемые модули для основных программ электронной почты: Outlook, Eudora и Claris Emailer (только в платной версии).
- Поддержка посредников, полезная для пользователей, находящихся позади межсетевого экрана (только в платной версии).
- PGPDisk - средство шифрования целых томов или частей вашего диска, так что зашифрование и расшифрование данных происходит автоматически (только в платной версии).

Прежде чем устанавливать и применять PGP, следует понять, как и на каких принципах работает программа. Данный раздел не претендует на то, чтобы научить вас разбираться во всех деталях криптографии и PGP, для этого нужно обратиться к любой из множества книг на данную тему. Но в результате прочтения этой лекции вы сможете зашифровывать и расшифровывать сообщения с помощью PGP.

Предостережение: При неправильном применении PGP может обеспечиваться лишь слабая или вообще нулевая защита. Кроме того, при неосторожном обращении с ключами расшифрования можно безвозвратно потерять свои данные (см. врезку "Не теряйте ваши ключи!").

PGP считается гибридной криптосистемой. Это означает, что для реализации своих функций она использует комбинацию симметричной и асимметричной криптографии. Криптография с открытым ключом требует значительно большей вычислительной мощности, чем симметричная, так как обычно опирается на сложные математические действия с простыми числами. В PGP криптография с открытым ключом применяется только для выработки сеансового ключа, который затем используется для шифрования сообщений с помощью традиционной симметричной криптографии. Большинство криптосистем с открытым ключом применяют подобный метод для повышения эффективности.

Вместо того чтобы при каждом использовании PGP целиком вводить свой секретный ключ, что требует немало времени и сопряжено с многочисленными ошибками, секретный ключ извлекается с жесткого диска, где он хранится в зашифрованном виде. Чтобы разблокировать свой ключ, необходимо всякий раз при использовании PGP вводить парольную фразу. Она похожа на пароль, но обычно длиннее и состоит из нескольких слов, предпочтительно из букв

и цифр. Очень важно запомнить эту парольную фразу, поскольку если вы ее потеряете или забудете, вы не сможете восстановить данные, зашифрованные с помощью PGP.

Установка PGP и генерация пары ключей открытый/секретный

1. В первую очередь, загрузите файл программы PGP с web-сайта.
2. Щелкните мышью на самораспаковывающемся zip-файле, и он автоматически начнет процесс установки.
3. У вас есть выбор между покупкой полной лицензии и оценкой продукта. Щелкните мышью на кнопке Purchase Now (Купить сейчас), если хотите получить полную версию, авторизованную для коммерческого применения. В противном случае щелкните мышью на кнопке Later (Позже), чтобы воспользоваться условно свободной версией.
4. Программа установки затем проведет вас через процесс генерации пары ключей открытый/секретный. Этот процесс очень важен, так как служит основой защиты, которую предоставляет PGP.
5. Программа попросит вас ввести имя, название организации и адрес электронной почты. Вы не обязаны вводить электронный адрес, но если вы этого не сделаете, открытый ключ не будет ассоциирован с вашим адресом на сервере ключей, и человеку, который захочет послать вам зашифрованное PGP-сообщение, будет сложно найти ваш открытый ключ, если он его еще не имеет.
6. Затем программа попросит ввести парольную фразу, позволяющую хранить ключи на диске. Не вводите здесь обычный пароль, такой как одиночное слово или набор букв. Это существенно снизит безопасность ключей. Задайте последовательность слов с комбинацией из букв и цифр. Это облегчит запоминание и в то же время гарантирует достаточную степень сложности. Хороший пример сложной парольной фразы с числами, большими и малыми буквами и другими символами - `one+one=Two`.

Примечание. Не используйте этот пример... И, конечно, это не моя парольная фраза.

После ввода парольной фразы загрузится оставшая часть программы, и установка будет завершена.

Чтобы удалить PGP со своего компьютера, следует воспользоваться предоставляемой функцией деинсталляции. Простого удаления файлов недостаточно, так как PGP вносит значительные изменения в реестр и другие базовые настройки Windows.



Флэми Тех советует:

Не теряйте ключи!

Лишиться ключей для PGP - почти то же, что потерять ключи от квартиры или автомобиля, только намного хуже. Вообразите, что при утере физических ключей ваш дом или автомобиль навсегда становятся недоступными. Именно это происходит с зашифрованными данными, если теряется секретный ключ. А поскольку секретный ключ обычно зашифрован на диске с помощью парольной фразы, то потеря последней дает обычно тот же эффект.

Не забудьте сделать резервную копию папки с секретным ключом на вашем компьютере (вы же регулярно делаете резервное копирование своих данных, не так ли?) Если вам трудно запоминать пароли, запишите парольную фразу и сохраните ее где-нибудь в безопасном месте (желательно - не на "горчичнике", прилепленном к монитору).

Помните, если вы потеряете ключ или парольную фразу, данные будут безвозвратно утеряны; даже Агентство национальной безопасности не сможет вам помочь. Думаете, это преувеличение? Если бы ваши данные легко восстанавливались, это было бы легко сделать и постороннему. Поэтому помните о своей парольной фразе и ключах.

Применение PGP

Доступ к PGP осуществляется из подменю Programs в меню Start. Там имеется несколько доступных опций, включая PGPMail и документацию. Условно свободная версия PGP снабжена отличной документацией, включая более чем семидесятистраничное введение в криптографию. Это прекрасный начальный курс для новичков в криптографии. Для PGP написано также огромное руководство пользователя.

При запуске PGPMail на экране появляется небольшая панель инструментов. Ее можно минимизировать до небольшой иконки на системной панели, когда она не используется. Простой интерфейс PGPMail предлагает несколько опций: PGPKeys, Encrypt, Sign, Encrypt and Sign, Decrypt/Verify, Wipe и FreeSpace wipe. Конкретные функции каждого элемента рассмотрены ниже.

PGPKeys

Раздел PGPKeys служит для управления как вашими открытым и секретным ключами, так и открытыми ключами ваших партнеров (рис. 9.3). Программа PGP создает на диске два каталога, называемых кольцами для ключей, так как они содержат все ключи, как открытые, так и секретные, которые нужны для применения PGP. Файл `pubring`, находящийся в основном каталоге PGP, содержит ваш открытый ключ, а также открытые ключи других людей, которым вы предполагаете посылать зашифрованные файлы. Файл `secring` содержит ваш секретный ключ, обычно в зашифрованном виде. Как правило, он содержит только один секретный ключ, но у вас может быть и несколько секретных ключей. Например, вы можете применять один из них для деловых писем, а другой - для частной корреспонденции. Помните только, что документы, зашифрованные с помощью определенного открытого ключа, можно расшифровать только с помощью соответствующего ему секретного ключа.

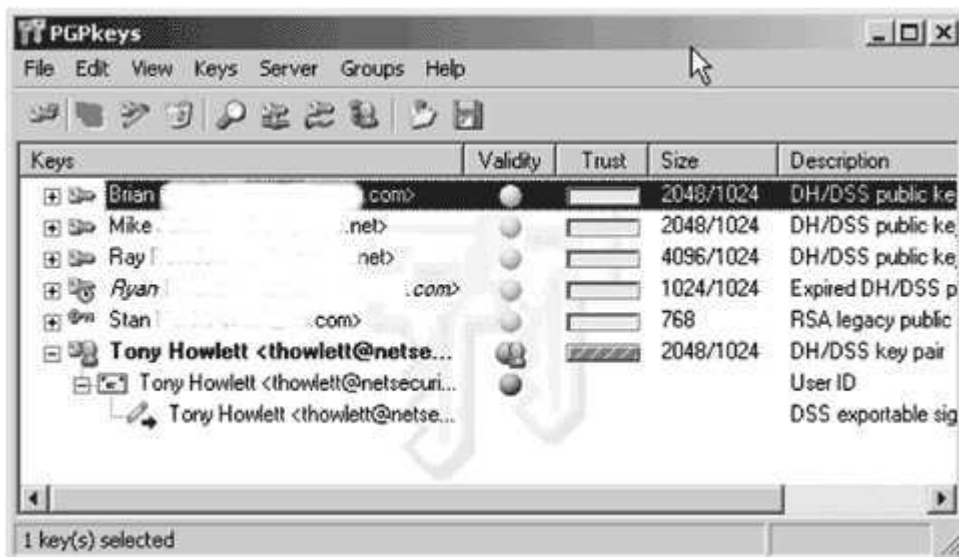


Рис. 9.3. Экран PGPKeys

Здесь можно также создавать новые ключевые пары и отзывать пары ключей, которые больше не используются. Можно загрузить свой открытый ключ на один из нескольких серверов открытых ключей. Это позволит тем, кто никогда с вами не общался, найти ваш ключ на сервере открытых ключей и послать вам сообщение PGP. Многие люди, применяющие PGP, имеют привычку размещать открытый ключ в строке подписи своих электронных сообщений, поэтому их адресаты могут легко послать им сообщение, зашифрованное PGP.

Другим способом, помогающим проверить законность некоторого персонального ключа, является его подписывание ключами других людей. Это позволяет удостовериться, что некий открытый ключ принадлежит данному человеку. Вы должны подписывать открытые ключи только тех людей, которых хорошо знаете лично, и проверять, что ключ правильный. Ваши друзья и знакомые также могут подписывать ваши ключи. Это кольцо подписывающих ключей создает неиерархическую модель доверия, называемую сетью доверия. Ее главное достоинство в том, что для ее работы не требуется центральный уполномоченный орган. Более подробно о том, как работает сеть доверия, можно узнать из раздела GnuPG далее в этой лекции.

Чтобы добавить ключи других пользователей в ваше кольцо открытых ключей, можно либо импортировать их непосредственно из файла или произвести поиск на серверах открытых ключей. Выбирая Search в меню Servers или щелкая мышью на иконке с изображением увеличительного стекла и вводя часть имени или некоторый идентифицирующий текст, можно увидеть, какие ключи на серверах открытых ключей соответствуют вашему запросу. Отбуксируйте выбранные элементы из результатов на основной экран PGPKey, и открытый ключ этого человека будет доступен для использования в сообщениях PGP. Можно также просмотреть специфические свойства любого ключа, включая подписавших этот ключ, размер ключа (в битах) и метод (обычно DH, что означает Diffie-Hellman). Наконец, можно импортировать или экспортировать ваши кольца ключей, если вы меняете компьютеры или должны восстановить данные с резервной копии.

Encrypt

Функция Encrypt устроена просто. Сначала появляется диалоговое окно, которое позволяет выбрать файл для зашифрования. Когда файл выбран, PGP попросит выбрать открытый ключ адресата из вашего кольца ключей. Если нужного ключа у вас еще нет, поищите его на серверах открытых ключей, как описано выше, и добавьте его к своему списку. Выберите открытый ключ вашего адресата и отбуксируйте ключ из поля наверху в список адресатов.

Флажки внизу слева позволяют задать несколько важных опций (рис. 9.4). Одна из наиболее важных - Wipe Original (Стереть оригинал). Установите этот флажок, если вы шифруете файл для хранения на жестком диске. В противном случае PGP просто создаст новый зашифрованный файл и оставит оригинал в открытом для просмотра текстовом виде в том же каталоге. Помните, однако, что если вы выберете эту опцию и потеряете свои ключи, то файл пропадет безвозвратно.



Рис. 9.4. Экран опций шифрования PGP

Другим важным параметром является Conventional Encryption (Обычное шифрование). Установка этого флажка отменяет шифрование с открытым ключом. Вместо этого будет производиться стандартное шифрование с разделяемым секретом, и вам нужно будет выбрать парольную фразу, чтобы зашифровать данные. Эту парольную фразу затем нужно безопасным образом передать адресату. Подобный метод ликвидирует основное достоинство PGP, но он может быть необходим, если у вас нет открытого ключа адресата. Если у адресата нет программного обеспечения PGP, выберите опцию Self-Decrypting Archive

(Саморасшифровывающийся архив). При этом будет создан файл, который сам расшифрует себя, когда получатель щелкнет на нем мышью. Конечно, получатель все равно должен знать парольную фразу, которая применялась при создании файла.

Sign

Функция Sign дает возможность подписать файл с помощью секретного ключа, позволяя впоследствии проверить, что с момента подписания файл не изменился. При этом применяется хэш-функция для преобразования файла в формат дайджеста, а затем производится шифрование с помощью секретного ключа. Это действие противоположно обычному шифрованию открытым ключом. Получатель может взять подпись и попытаться расшифровать ее с помощью вашего открытого ключа. Если хэши совпадут, то можно утверждать, что с момента подписания содержимое не изменилось. Данная функция полезна, если вы больше озабочены целостностью файла, чем конфиденциальностью информации. Примером может служить длинный контракт, который был существенно отредактирован. Можно подписать его цифровой подписью и быть уверенным, что после этого никто не сможет его изменить. Подпись можно также применять для обеспечения так называемой "неотказуемости" - если вы подписали документ, то может быть доказано, что вы это сделали, если только кто-то не заполучил ваш секретный ключ. Это равносильно юридической силе физической подписи, за исключением того, что подделать цифровую подпись значительно труднее, чем обычную.

Encrypt and Sign

Эта функция совмещает функции Encrypt и Sign, обеспечивая строгую конфиденциальность, целостность и неотказуемость.

Decrypt/Verify

Функция Decrypt/Verify применяется для обращения процесса шифрования PGP. После выбора файла для расшифрования будет предложено ввести парольную фразу, чтобы можно было использовать хранящийся на диске секретный ключ. Если парольная фраза будет введена правильно, вам предложат задать имя нового файла, в который будет помещен результат расшифрования. Эту функцию можно также применять для проверки подлинности подписи.

Wipe

Функция Wipe навсегда стирает файл с жесткого диска. Этот процесс существенно надежнее, чем функция Delete в Windows. Проблема с Windows и большинством других операционных систем состоит в том, что при удалении файла они на самом деле не удаляют данные с жесткого диска, ограничиваясь удалением записи о файле в индексе файловой системы. Данные по-прежнему остаются на дисковых пластинах. Их можно просмотреть с помощью низкоуровневого дискового редактора или восстановить с помощью легко доступных утилит, таких как Norton или DD (DD демонстрируется в [лекции 11](#)). Функция Wipe на самом деле несколько раз перезаписывает данные на диске случайными единицами и нулями. По умолчанию это делается три раза, что вполне достаточно для большинства случаев. При желании можно увеличить это число по крайней мере до десяти, если вы стираете сверхсекретные данные, поскольку специалисты по восстановлению данных в действительности могут восстановить данные даже после нескольких перезаписей. Можно увеличить число проходов до 28, и в этом случае даже Агентство национальной безопасности будет бессильно. Отметим, что при стирании больших файлов с большим числом проходов может потребоваться довольно много времени. Это высокоинтенсивная дисковая операция.

Freespace Wipe

Freespace Wipe выполняет ту же функцию, что и Wipe, но для свободного дискового пространства. Время от времени протирать свободные блоки необходимо, так как старые файлы, которые вы удаляли, но не стирали, все еще могут существовать. Кроме того, программы постоянно создают временные файлы, которые могут содержать копии данных ограниченного доступа. Они удаляются операционной системой, когда программа завершается, но все еще существуют на диске. Freespace Wipe санирует весь ваш жесткий диск. Можно запланировать автоматические регулярные санации жесткого диска.

Опции PGP

В PGP имеется ряд глобальных опций. В основном меню PGPKeys в разделе File выберите Edit, чтобы вывести диалоговое окно опций PGP ([рис. 9.5](#)). В [табл. 9.2](#) даны описания имеющихся вкладок.

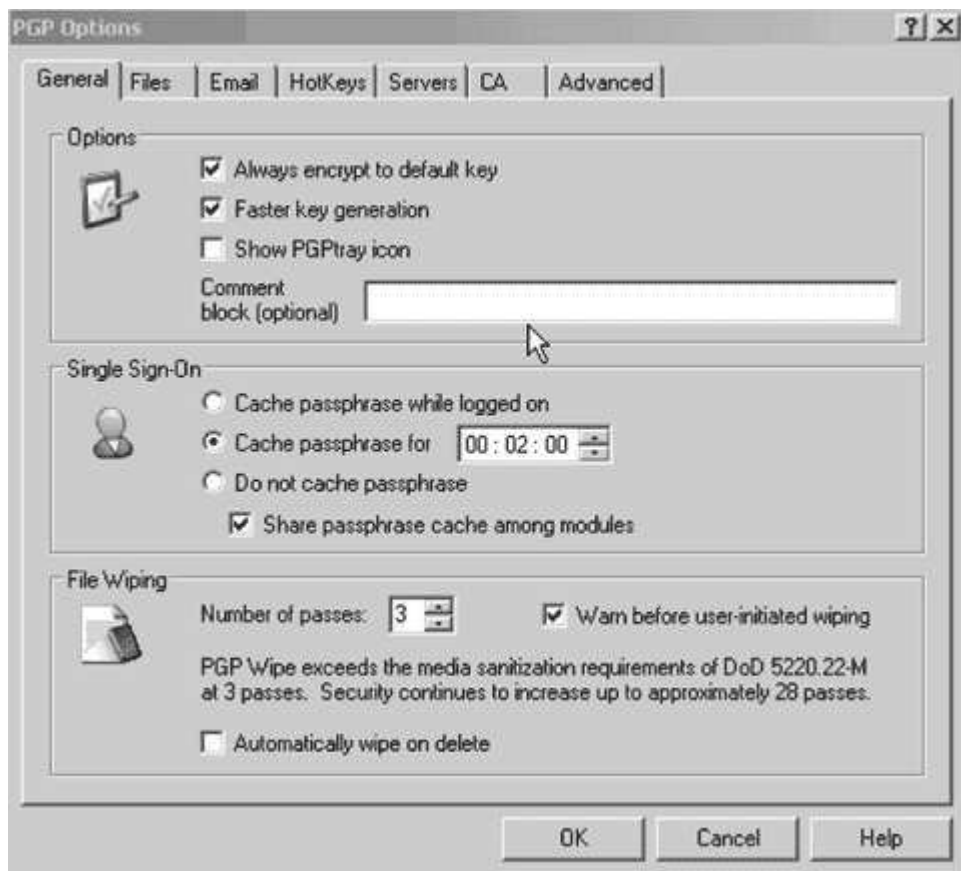


Рис. 9.5. Диалоговое окно опций PGP

Таблица 9.2. Вкладки диалогового окна опций PGP

| Вкладка | Описание |
|----------|---|
| General | Здесь можно задать запоминание вашей парольной фразы на определенное время после ее использования, чтобы не нужно было вводить ее всякий раз при расшифровании файла или подписывании документа. Подразумеваемый срок - две минуты. Можно также увеличить подразумеваемое число проходов функции Wipe и заставить Windows автоматически затирать файл после его удаления. Применяйте эту опцию с осторожностью, если хотите оставить возможность восстановления удаленных файлов. |
| Files | Здесь можно изменить подразумеваемый каталог ваших колец открытых и секретных ключей |
| Email | На этой вкладке можно задать различные опции для обработки зашифрованных электронных сообщений, включая автоматическое расшифрование сообщений PGP, подписывание всех исходящих сообщений и т.д. |
| HotKeys | Здесь можно задать быстрый доступ к основным функциям PGP с помощью горячих клавиш. Некоторые клавиши предопределены. Например, можно очистить кэш парольной фразы, нажав F12 |
| Servers | Здесь задаются серверы для поиска открытых ключей. Имеются два основных сервера, по одному в США и в Европе, но можно добавить и другие |
| CA | Если вы хотите применять цифровые сертификаты, то здесь задаются удостоверяющий центр и различные настройки |
| Advanced | Эта вкладка содержит опции процесса шифрования. Можно выбрать алгоритм для симметричной части процесса (по умолчанию применяется AES). Можно также задать опции резервного копирования колец ключей. По умолчанию при закрытии программы всегда создается файл с |

резервной копией каждого кольца. Однако необходимо периодически перемещать файлы резервных копий на безопасные носители, прожигая их на компакт-диске или перенося на флорпи-диск. Это защитит ваши ключи в случае краха жесткого диска или кражи компьютера

Приведенных сведений достаточно для запуска PGP и защиты ваших файлов и коммуникаций. Повторим, что это лишь краткий обзор продукта PGP. Более подробно с ним можно ознакомиться посредством обширной документации. Если потребуются дополнительные функции или коммерческое применение, можно рассмотреть возможность приобретения коммерческой версии.

Если вы не согласны со всеми ограничениями условно свободной лицензии PGP, но хотите применять PGP, имеется другая возможность - GNU-версия PGP, описанная ниже.

GNU Privacy Guard (GnuPG): Реализация PGP на условиях GPL

GNU Privacy Guard (GnuPG)

Автор/основной контакт: Matthew Skala, Michel Roth, Niklas Hærnæus, Remi Guyomarch, Werner Koch и другие

Web-сайт: <http://www.gnupg.org/>

Платформы: Linux, Windows, BSD и Mac

Лицензия: GPL

Рассмотренная версия: 1.2.4

Другие ресурсы: <http://www.pgpi.com/>

Списки почтовой рассылки:

Группа поддержки PGP Freeware

Рабочая группа IETF OpenPGP

Пользовательский список рассылки PGP

Рабочая группа PGP/MIME

Список рассылки разработчиков PGPi

Список рассылки переводчиков PGPi

Список рассылки разработчиков Pgp-lib

Все эти списки доступны для подписки по адресу:

<http://www.pgpi.org/links/maillinglists/en/>.

Телеконференции:

Alt.security.pgp

Comp.security.pgp.announce

Comp.security.pgp.discuss

Comp.security.pgp.resources

Comp.security.pgp.tech

GNU Privacy Guard (GnuPG) основывается на стандарте OpenPGP и является ответом на коммерческую и ограниченную условно свободную лицензионные версии PGP. В названии программы, как и в названиях большинства программ от GNU, заключена игра слов (инверсия PGP). Важное преимущество версии GNU - возможность применения для любого приложения, личного или коммерческого. Кроме того, поскольку используется лицензия GPL, программу можно расширять или встраивать в любые приложения. Недостатком же является то, что это средство командной строки, поэтому в нем отсутствуют некоторые полезные добавления, имеющиеся в коммерческой версии PGP. Если цена составляет проблему, и вы не боитесь изучать работу команд, то GnuPG - для вас. Однако, одно предостережение: GnuPG, вероятно, не лучший выбор для нетехнических пользователей, если только вы не добавите собственный интерфейс или удобные для пользователей процедуры (некоторые из них доступны в Интернет).

Установка GnuPG

Многие современные версии Linux и BSD поставляются с предустановленной системой GnuPG. Это можно проверить, набрав в командной строке `gpg --version`. Если появится листинг с информацией о программе, то можно пропустить данный раздел и сразу перейти к применению GnuPG.

Проверьте также, содержат ли ваши дистрибутивные диски файл RPM для автоматической установки. Если вы хотите получить самую свежую версию, то на web-сайте имеются RPM для многих дистрибутивов. Если там есть RPM для вашей ОС, загрузите его и просто щелкните на нем мышью, чтобы установить программу. Если RPM отсутствует, то можно загрузить .tar-файлы с прилагаемого к книге компакт-диска или с официального Web-сайта и скомпилировать их вручную с помощью следующих инструкций:

1. Распакуйте .tar-файлы, затем наберите обычные команды компиляции:

```
./configure
make
make install
```

Программа создаст структуру каталогов с корнем .gnupg в вашем пользовательском каталоге, где будут храниться ключи и другая информация.

2. (Необязательно) После установки GnuPG наберите `make clean`, чтобы избавиться от бинарных или временных файлов, созданных в процессе конфигурирования.

Создание ключевых пар

После установки программы прежде всего необходимо создать свою пару ключей открытый-секретный. Если у вас уже есть ключ и вы хотите импортировать его в GnuPG, воспользуйтесь командой:

```
gpg --import маршрутное_имя_файла_ключей
```

Следует выполнить эту инструкцию для вашего кольца открытых ключей и отдельно - для кольца секретных ключей. Обычные форматы для колец ключей - `pubring.pkr` и `secring.skr`.

Если ключей у вас еще нет, следуйте приведенной ниже процедуре.

1. Наберите `gpg --gen-key`. Будет запущен процесс, который запросит у вас некоторые данные.
2. GnuPG попросит задать длину ключей в битах (по умолчанию - 1024, что обычно достаточно для надежной криптографии с открытым ключом). Можно увеличить длину до 2048 для усиления безопасности.
3. Как правило, вам не нужно, чтобы ваши ключи имели ограниченный срок годности, но в специальном случае, когда ключи будут применяться ограниченное время, можно указать, когда истечет срок их действия.

4. GnuPG запросит у вас имя и адрес электронной почты. Эти данные важны, поскольку они определяют индексацию открытого ключа на серверах открытых ключей.
5. Наконец, GnuPG предложит ввести парольную фразу. Она должна быть достаточно длинной и сложной, но в то же время легко запоминаемой. (См. описание парольной фразы выше в этой лекции, в разделе о PGP). После двукратного ввода парольной фразы GnuPG создаст ключи. Это может занять некоторое время. В ходе этого процесса следует немного подвигать мышью. GnuPG использует случайные сигналы клавиатуры и мыши для повышения энтропии своего датчика случайных чисел.

Примечание: Еще раз - как и при работе с PGP или любым другим средством сильной криптографии, сохраняйте резервные копии пар ключей в безопасном месте и не теряйте их, иначе ваши зашифрованные данные будут безвозвратно потеряны.

Создание сертификата отзыва

После создания ключей можно создать сертификат отзыва. Он применяется, когда вы теряете ключи, или если кто-то скомпрометирует ваш секретный ключ. Тогда можно воспользоваться этим сертификатом для отзыва ключа с серверов открытых ключей. Тем не менее, вы сможете расшифровывать полученные ранее сообщения, зашифрованные с помощью старого открытого ключа (если вы не потеряли старый секретный), но никто больше не сможет зашифровывать сообщения с помощью ставших негодными открытых ключей.

Чтобы создать сертификат отзыва, введите:

```
gpg - output revoke.asc - gen-revoke пользователь
```

где пользователь заменяется его уникальной фразой в вашем кольце секретных ключей. Создается файл revoke.asc. Необходимо переместить его с жесткого диска в какое-то безопасное место. Нежелательно оставлять его рядом с секретным ключом, поскольку если кто-то получит доступ к секретному ключу, то сможет помешать и его отзыву.

Публикация открытого ключа

Желательно разместить ваш открытый ключ на сервере ключей, чтобы его можно было легко найти и послать вам сообщение. Чтобы сделать это, наберите команду:

```
gpg - keyserver имя_сервера_открытых_ключей - send-key пользователь
```

где пользователь задается адресом электронной почты, с которым ассоциируется публикуемый ключ. Можно использовать любой сервер открытых ключей PGP, так как все они регулярно синхронизируются. Выберите любой из них, и ваш открытый ключ будет распространен на все остальные. Имеется много серверов открытых ключей, например:

- certserver.pgp.com
- pgp.mit.edu
- usa.keyserver.net

Шифрование файлов с помощью GnuPG

Для зашифрования файла служит команда `-- encrypt`, имеющая следующий формат:

```
gpg -- output зашифрованный_файл -- encrypt -- recipient адрес шифруемый_файл
```

Здесь `адрес` - это электронный адрес пользователя, которому вы хотите отправить зашифрованный файл. Отметим, что вы должны иметь открытый ключ адресата в своем кольце ключей.

GnuPG можно применять и для шифрования файлов с помощью простой симметричной криптографии, что удобно для защиты локальных файлов или для отправки сообщения кому-то, чей открытый ключ у вас отсутствует. Чтобы сделать это, воспользуйтесь командой `--symmetric`:

```
gpg --output зашифрованный_файл --symmetric шифруемый_файл
```

Расшифрование файлов

Для расшифрования полученных файлов при помощи GnuPG, воспользуйтесь следующей командой:

```
gpg --output расшифрованный_файл --decrypt зашифрованный_файл
```

Чтобы расшифровать файл, необходимо иметь в своем кольце секретных ключей ключ пользователя, для которого файл был зашифрован. У вас запросят парольную фразу, и если вы введете ее правильно, GnuPG создаст расшифрованный файл.

Подписывание файлов

Как упоминалось выше, имеется еще одно применение GnuPG и PGP - подписывание документов для контроля их целостности. Это можно сделать с помощью следующей команды:

```
gpg --output подписанный_файл --sign подписываемый_файл
```

Обычно GnuPG автоматически проверяет подпись при получении файла. Верификация файла по инициативе пользователя осуществляется с помощью команды

```
gpg --verify подписанный_файл
```

Можно создавать подписи в текстовом виде и отдельно от файла, если вы хотите, чтобы пользователи без GnuPG могли получить к ним доступ. Для этого имеются две команды. Команда

```
gpg -- clearsign подписываемый_файл
```

создает текстовое дополнение к подписываемому файлу. Если вы не хотите изменять этот файл, можно создать отдельный файл подписи с помощью команды

```
gpg --output файл_подписи --detach-sign подписываемый_файл
```

Модель сети доверия PGP/GnuPG

Как упоминалось ранее, вместо иерархической системы доверия, с цифровыми сертификатами и удостоверяющим их центром, в PGP и GnuPG используется модель сети доверия. Подписывая ключи людей, которых вы знаете, вы можете проверить, что их ключ заслуживает доверия. А если они подписывают ключи других людей, которых вы не знаете непосредственно, вы создаете цепочку доверия. Такая модель основана на идее "друг моего друга - мой друг". Модель, конечно, несовершенна; кто-нибудь в дальней части цепочки доверия может оказаться плохим парнем. Но основной принцип состоит в естественном росте цепочки без какой-либо инфраструктуры, и поэтому ее нельзя легко разрушить или "раздуть". Вы создаете сеть доверия, подписывая ключи каких-то людей и предоставляя им возможность подписывать ваши. В примере на [рис. 9.6](#) Тони может неявно доверять ключам Джейн, Джо, Джона и Евы, хотя он не знает их непосредственно.

Web of Trust Model

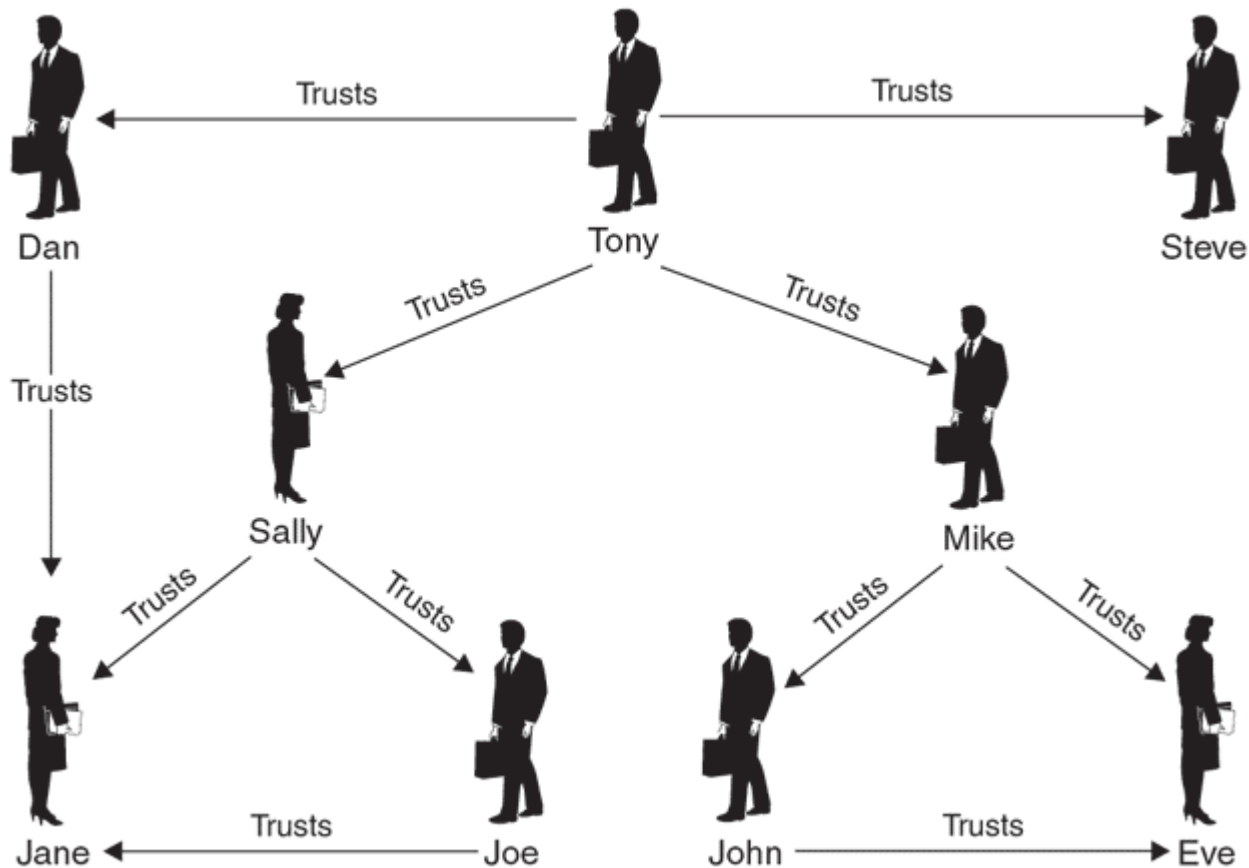


Рис. 9.6. Модель сети доверия

Подписание ключей и управление доверием к ним

В GnuPG вы подписываете ключи и управляете доверием к ним, переходя в режим редактирования ключей с помощью команды

```
gpg --edit-key доверенный_адрес
```

Здесь доверенный адрес ассоциируется с подписываемым или управляемым ключом, входящим в ваше открытое кольцо. Команда выдает основную информацию о ключе. В этом режиме наберите `fpr`, чтобы распечатать идентификационную метку данного ключа. Как и отпечатки пальцев для людей, идентификационная метка служит специфической формой идентификатора ключа. Убедитесь, что это ключ нужного человека, проверяя его либо по телефону, либо каким-то другим способом. Можно также проверить, кто еще подписал этот ключ, вводя `check`. Будет напечатан список лиц, подписавших этот ключ, что может помочь вам определить его законность.

Когда вы уверены, что это ключ нужного человека, наберите `sign`. Данная команда подписывает ключ этого человека, так что все смогут узнать, что вы ему доверяете. В этом режиме можно также отредактировать уровни доверия различных ключей вашего кольца. Войдите в этот режим из режима редактирования ключа, набирая `trust`. Появится следующее меню:

```
1 = Don't know (Не знаю)
2 = I do NOT trust (Я НЕ доверяю)
```

3 = I trust marginally (Я доверяю косвенно)
4 = I trust fully (Я доверяю полностью)
s = Please show me more information (Мне требуется дополнительная информация)
m = Back to the main menu (Вернуться в основное меню)

Выберите один из этих элементов, и ключ будет помечен должным образом. Это еще один способ сообщить о том, какие пользователи вызывают у вас наибольшее доверие, а каких вы едва знаете.

Вышеизложенное может служить хорошим введением в мир PGP и GnuPG. Данная лекция не претендует на полноту освещения этой темы, так что следует обратиться к соответствующим Web-сайтам и другим упомянутым источникам, чтобы лучше изучить эти революционные программы.

PGP и GnuPG отлично работают для шифрования файлов. Однако как быть, если вы желаете шифровать все коммуникации между двумя точками? Программа PGP в действительности не годится на эту роль (несмотря на недавнее включение в коммерческую версию клиента виртуальных защищенных сетей). Мы приступаем к обсуждению средства с открытыми исходными текстами для создания таких постоянных соединений, где все шифруется на лету.

OpenSSH (сервер)

Автор/основной контакт: Tatu Ylonen (первоначальный автор) и другие

Web-сайт: <http://www.openssh.org/>

Платформы: BSD, Linux и большинство UNIX

Лицензия: BSD

Рассмотренная версия: 2.1.1p4

Списки почтовой рассылки:

Список объявлений

Список только для чтения, который содержит общие объявления о выпуске новых версий, исправлении ошибок и т.д. Подписка по адресу <http://www.mindrot.org/mailman/listinfo/openssh-unix-announce>.

Общий список пользователей SSH

Вопросы и общая дискуссия о применении SSH. Подпишитесь, послав пустое сообщение по адресу secureshell-subscribe@securityfocus.com.

Чтобы отказаться от подписки, пошлите пустое сообщение по адресу

secureshell-unsubscribe@securityfocus.com.

Имеется также архив данного списка по адресу

<http://marc.theaimsgroup.com/?!=secure-shell&r=1&w=2>.

Список разработчиков

Обсуждение разработки SSH и программирования. Подписка по адресу

<http://www.mindrot.org/mailman/listinfo/openssh-unix-dev>.

Большинство файловых и коммуникационных утилит, до сих пор применяемых в Интернете, восходят к тем временам, когда Интернет был небольшим и безопасным. Одним из наиболее употребительных средств, помимо web-навигатора, является Telnet. Эта утилита служит для удаленного терминального доступа ко всевозможным серверам, маршрутизаторам, межсетевым экранам и другим устройствам. Большим минусом Telnet является то, что утилита посылает свои данные в открытую, поэтому, если вы применяете Telnet для входа в системы через Интернет, то кто-нибудь может перехватить ваш трафик, включая ваши пароли. Вы вольны считать, что поиск вашего пароля в потоке данных подобен поиску иголки в стоге сена, но хакеры написали программы, выполняющие поверх сетевых анализаторов, для поиска обычных входных атрибутов и записи результатов. Это справедливо и для других средств удаленного доступа, таких как FTP, TFTP и RCP.

SSH (Secure Shell, защищенный командный интерпретатор) решает эту проблему, применяя как асимметричную, так и симметричную криптографию для шифрования сеансов, начиная с первого нажатия клавиши. В этом случае злоумышленник, прослушивающий ваше соединение, получит лишь случайный шум. SSH не только обеспечивает конфиденциальность ваших данных с помощью шифрования, но предоставляет также сильную аутентификацию, препятствующую подделкам и другим маскарадным атакам. Это достигается с помощью цифровых сертификатов для аутентификации пользователей. Не

пугайте SSH с SSL - стандартом шифрования Web. Хотя они делают одно и то же дело, SSH работает с любым протоколом, в то время как SSL предназначен прежде всего для web-коммуникаций.

SSH включает также SCP - безопасный эквивалент RCP, средства удаленного копирования, и SFTP - безопасный аналог FTP. SSH можно применять и для туннелирования других протоколов, таких как HTTP и SMTP. Некоторые приложения рассмотрены в конце данного раздела. Использование этого пакета программ вместо старых аналогов гарантирует, что ваши коммуникации с серверами не будут раскрыты. Отказаться от применения в вашей сети Telnet и FTP может быть нелегко, но чем больше вы в этом преуспеете, тем в большей безопасности окажетесь.

Чтобы использовать SSH, необходимо иметь сервер SSH, выполняющийся на машине, к которой вы хотите подключиться, и клиент SSH на машине, с которой вы подключаетесь. Обычные клиенты FTP и Telnet не будут соединяться с сервером SSH. Клиент встроен в большинство современных операционных систем Linux, хотя, быть может, требуется выбрать эту опцию при установке ОС. (см. в [лекции 2](#) дополнительную информацию о клиенте SSH). Сервер SSH обычно является необязательным, и его следует выбрать при установке ОС. Чтобы проверить, установлен ли он, наберите

```
ps -ax | grep sshd
```

и посмотрите, выполняется ли процесс `sshd`. Если его нет, то необходимо установить сервер, чтобы обеспечить соединение с вашей машиной через SSH.

Установка и запуск сервера OpenSSH

1. Первым делом загрузите пакет с Web-сайта или с прилагаемого к книге компакт-диска и распакуйте его.
2. Выполните обычные команды компиляции в Linux:

```
./configure  
make  
make install
```

Произойдет сборка и установка программ SSH. Бинарные файлы и ассоциированные библиотеки будут размещены в каталоге `/usr/local/bin` (в системе Mandrake Linux, в других дистрибутивах может быть иначе). Системные демоны помещаются в `/usr/local/sbin`, а конфигурационные файлы - в `/usr/local/etc/ssh` или `/etc/ssh`, в зависимости от установки.

Можно выбрать альтернативный маршрут установки, используя аргумент `configure`

```
---prefix=маршрут
```

где `маршрут` нужно заменить желательным альтернативным местом.

3. После установки OpenSSH проверьте конфигурационный файл, находящийся в `/etc/ssh`, и убедитесь, что он соответствует параметрам вашей системы. Конфигурационный файл для серверной части называется `sshd_config`. Для внесения изменений можно воспользоваться текстовым редактором, таким как `vi` или EMACS. Необходимо проверить следующее:
 - Port: Порт, который SSH использует для входящих соединений. Подразумеваемый номер - 22. Если изменить это значение, то люди, пытающиеся с вами соединиться, должны будут вручную изменить номер порта у своих клиентов SSH.
 - Protocols: Набор протоколов, которые SSH должен принимать. По умолчанию принимаются оба типа соединений SSH1 и SSH2. Для повышения безопасности можно разрешить прием только SSH2, но тогда некоторые старые клиенты не смогут соединяться.
 - Hostkey: Задаёт расположение ключей, применяемых при проведении основанной на ключах аутентификации пользователя во время соединения с другой машиной. Это не то же самое, что ключи сервера, которые генерируются при установке.
4. Прежде чем пользователь сможет работать с SSH, он должен сгенерировать ключи. Это делается с помощью следующей команды:

```
ssh make-host-key
```

Вы получите примерно следующий ответ:


```
Generating public/private rsa key pair.
(Генерация пары RSA-ключей открытый/секретный).
Enter file in which to save the key (/home/me/.ssh/id_rsa):
(Введите файл для сохранения ключа).
Created directory '/home/me/.ssh'. Создан каталог
Enter passphrase (empty for no passphrase):
(Введите парольную фразу (пусто без фразы)).
Your identification has been saved in /home/me/.ssh/id_rsa.
(Идентификационные данные сохранены в)
Your public key has been saved in /home/me/.ssh/id_rsa.pub.
(Открытый ключ сохранен в)
The key fingerprint is :f6:41:99:d8:a5:d1:fb:e7:93:86:7e:e6:4f:01:d9:5b
(Идентификационная метка ключа).
```

Эта же команда, но с дополнительными опциями (задающими, например, неинтерактивный режим, длину ключей и имя файла для сохранения результатов), применяется и для генерации ключей хоста, необходимых серверу ssh. Идентификационная метка служит уникальным идентификатором ключей.

5. Теперь можно запустить сервер SSH из командной строки, набрав `sshd &`. Эта команда в фоновом режиме запускает `sshd` - серверный демон, постоянно слушающий попытки подключений. Если вы хотите, чтобы `sshd` запускался автоматически при загрузке системы (что предпочтительно), поместите эту строку в конце файла `rc.local`, находящегося в каталоге `/etc/rc.d/` (в Mandrake Linux, или в соответствующем стартовом файле для вашего дистрибутива).

Помните, чтобы соединиться с вашим сервером через SSH, необходимо иметь на клиентской стороне совместимую версию SSH. Инструкции по установке и применению клиента SSH можно найти в [лекции 2](#).

Переправка портов посредством OpenSSH

Хотя SSH первоначально предназначался для взаимодействия на уровне командной строки, подобно Telnet, его можно применять также для создания безопасного туннеля между двумя машинами для произвольного приложения. Можно создать безопасное соединение между двумя серверами с помощью встроенной в SSH возможности переправки порта. Чтобы это сработало, SSH должен выполняться на обоих концах соединения. Соединение можно организовать для любого сервиса с любым портом, выполнив на клиентской стороне следующую инструкцию:

```
ssh -L локальный_порт:удаленный_хост:удаленный_порт -N удаленный_хост
```

где надо заменить:

- `локальный_порт` - случайно выбранным большим номером порта для создания нового криптографически защищенного соединения;
- `удаленный_хост` - IP-адресом или именем серверного хоста на другой стороне соединения;
- `удаленный_порт` - портом сервиса, который вы желаете туннелировать на удаленную сторону;

Опция `-L` предписывает SSH слушать локальный порт на локальном хосте и переправлять любые соединения на удаленный порт удаленного хоста. Опция `-N` освобождает SSH от попыток входа; требуется просто поддерживать соединение открытым для переправляемого трафика.

При применении данного метода вам не нужно входить в удаленную систему для установления криптографически защищенного соединения с удаленным сервером. Вам понадобятся, если они требуются, соответствующие удостоверения для выполнения желаемых действий через переправляемый порт.

Ниже представлены два примера, которые показывают, как это работает.

Пример 1: Создание криптографически защищенного соединения для электронной почты с помощью OpenSSH

Обычно сообщения электронной почты пересылаются в открытом виде через порт 25. Предположим, вы желаете зашифровать это соединение. Один из способов добиться этого - создать при помощи SSH криптографически защищенный туннель для любого трафика, предназначенного для порта 25 почтового сервера. Используя вышеприведенный формат и считая, что почтовый сервер имеет IP-адрес 192.168.1.2, получим следующую команду:

```
ssh -L 5000:192.168.1.2:25 192.168.1.2 -N &
```

Эта команда задает порт 5000 на локальной машине для туннелирования почты (порт 25) на удаленный почтовый сервер. Поэтому, если вы настроите свой почтовый клиент для соединения с localhost:5000 вместо подразумеваемого почтового порта, SSH будет автоматически шифровать и переправлять трафик на порт 25 вашего почтового сервера. Теперь вы можете получать и посылать почту на эту машину, не опасаясь, что ее кто-то перехватит.

Пример 2: Создание безопасного web-соединения

Допустим, вы желаете соединиться со своим web-сервером для выполнения защищенной транзакции. Если сервер не настроен для поддержки SSL, вы все равно сможете при помощи SSH организовать безопасное туннелирование своего web-трафика на сервер. Если ваш web-сервер расположен по адресу 192.168.1.3, то командная строка будет выглядеть примерно так:

```
ssh -L 5000:192.168.1.3:80 192.168.1.3 -N &
```

Теперь вы можете соединиться, вводя localhost:5000 в web-навигаторе, и по безопасному туннелю ваш трафик будет переправляться в порт 80 удаленной машины. Можно переправлять несколько портов на одной машине. Например, команда

```
ssh -L 5000:192.168.1.2:25 -L 5001:192.168.1.2:80 -N 192.168.1.2 -N &
```

будет переправлять весь трафик с локального порта 5000 на почтовый порт и с порта 5001 - на порт 80 удаленной машины с адресом 192.168.1.2. Конечно, предполагается, что вы имеете почтовый счет на удаленном сервере.

Можно видеть, что SSH отлично подходит для создания безопасных соединений между двумя машинами практически для любого протокола. Однако, что если вы хотите шифровать весь трафик, независимо от порта или сервиса? В этом случае имеет смысл создать виртуальную защищенную сеть.

Виртуальные защищенные сети

Обычно организации создают защищенные сети, арендуя у телефонных компаний дорогие каналы "точка-точка". Обходится это в тысячи долларов в месяц, а соединенными оказываются лишь две производственные площадки. Со временем организации обрастают паутиной дорогостоящих коммуникационных линий, соединяющих их производственные площадки. С введением Интернета в производственную эксплуатацию многие сразу оценили его потенциал для межфилиальных коммуникаций. К сожалению, открытая природа Интернета создавала серьезную угрозу безопасности. На выручку пришла криптография. С ее помощью организация может создать виртуальную защищенную сеть и использовать недорогой Интернет для корпоративных коммуникаций, безопасно и без риска. Данные инкапсулируются в криптографически защищенный "туннель", поэтому злоумышленник при перехвате передаваемых пакетов не сможет добраться до полезной информации.

Имеется много производителей специализированного оборудования для создания виртуальных защищенных сетей, однако существует и решение с открытыми исходными текстами, позволяющее создать свою защищенную сеть с помощью лишь пары дополнительных ПК.

FreeS/WAN: Программное обеспечение с открытыми исходными текстами для создания виртуальных защищенных сетей на основе спецификаций IPsec

FreeS/WAN

Автор/основной контакт: John Gilmore

Web-сайт: <http://www.freeswan.org/>

Платформы: Большинство UNIX

Лицензия: GPL

Рассмотренная версия: 2.02

Списки почтовой рассылки:

Announce. Только чтение, для основных объявлений.

Briefs. Краткая сводка активности других списков.

Users. Основной список для вопросов пользователей и обсуждения.

Users-moderated. Модерируемая версия предыдущего списка с меньшим трафиком.

Design. Обсуждение только среди разработчиков.

Distros. Форум по поддержке Linux-дистрибутивов.

Bugs. Для сообщений обо всех ошибках, найденных в FreeS/WAN.

Инструкции по подписке на любой из перечисленных списков рассылки можно найти по адресу

<http://www.freeswan.org/mail.html>

Старые сообщения архивированы по адресу:

<http://www.sandelman.ottawa.33on.ca/linux-ipsec/>

Проект FreeS/WAN спонсирует и возглавляет Джон Гилмор - легендарная фигура в кругах программистов и борцов за свободу в сети, один из основателей Фонда электронной свободы (Electronic Freedom Foundation - EFF), много лет защищавшего права на свободную сильную криптографию. Участвуя в работе нескольких инновационных компаний Кремниевой Долины, в основном в Sun Microsystems, Гилмор составил состояние и теперь посвящает свое время различным проектам, многие из которых связаны с открытым ПО.

Проект FreeS/WAN зародился как попытка дать возможность шифровать свои коммуникации любому пользователю. Хотя эта цель еще не достигнута, пользователи Linux могут воспользоваться недорогим способом создания виртуальных защищенных сетей. Данное средство позволяет также соединяться с другими устройствами с помощью IPsec, так как IPsec - общепризнанный стандарт. Некоторые производители не строго следуют стандарту, поэтому возможны проблемы, если на другом конце применяется оборудование или программы других производителей. Проверьте на web-сайте FreeS/WAN список совместимости с реализациями других производителей.

При использовании IPsec все шифруется на уровне IP, независимо от приложения или порта. Именно это делает IPsec наиболее распространенной системой для создания защищенных коммуникаций. FreeS/WAN может применять и так называемое шифрование "по возможности", называемое также оппортунистическим. Это означает, что для коммуникаций с хостами, поддерживающими IPsec, применяется шифрование; с прочими хостами осуществляются обычные IP-коммуникации. Поэтому, если вы выполняете FreeS/WAN на компьютере межсетевого экрана, то вы можете получить автоматическую виртуальную защищенную сеть с сайтами, поддерживающими IPsec, при сохранении взаимодействия с другими сайтами, которые не поддерживают IPsec.

Вам потребуются две машины в качестве шлюзов. Для работы FreeS/WAN необходимы компьютеры с ОС UNIX, предпочтительно Linux. Чтобы организовать соединение IPsec между системами Windows, можно воспользоваться встроенной поддержкой IPsec (в Windows 2000 и более поздних разновидностях); FreeS/WAN для этого не нужен. Предположительно, поддержка IPsec будет встроена в новое ядро Linux. Но даже в этом случае FreeS/WAN по-прежнему

найдется применение для коммуникаций со старыми версиями и для использования возможностей оппортунистического шифрования. Команда FreeS/WAN работает также над совместимостью с предполагаемой поддержкой IPsec в ядре Linux.

Установка и запуск FreeS/WAN

FreeS/WAN предустанавливается на многих дистрибутивах Linux. Чтобы проверить наличие FreeS/WAN, введите `ipsec verify` в командной строке. Если вы получите ответ "file not found", значит, у вас нет этой системы. Даже если у вас нет RPM, можно взять исходные тексты с прилагаемого к книге компакт-диска или загрузить свежую версию, чтобы воспользоваться самыми последними криптографическими протоколами и возможностями. Для компиляции FreeS/WAN из исходных текстов следуйте приведенным ниже инструкциям.

1. Загрузите самый свежий пакет с web-сайта и распакуйте его или скопируйте файл с компакт-диска.
2. Для компиляции и установки пакета выполните следующие команды от имени пользователя `root` из каталога FreeS/WAN:

```
make oldmod
make minstall
```

3. После установки пакета FreeS/WAN необходимо перезагрузить систему, чтобы изменения вступили в силу.
4. Когда система перезагрузится, наберите в командной строке `ipsec verify`, чтобы проверить установку. Вы должны увидеть сообщение примерно такого вида:

```
Checkin your system to see if IPsec got installed and started correctly
Version check and ipsec on-path                               [OK]
Checking for KLIPS support in kernel                         [OK]
Checking for RSA private key (/etc/ipsec.secrets) [OK]
Checking that pluto is running                               [OK]
. . .
```

5. Если вы получили такую выдачу, можно запускать службу IPsec, набрав команду

```
service start ipsec
```

Служба IPsec выполняется в фоновом режиме. Теперь все готово к инициированию сеанса IPsec.

Применение FreeS/WAN

FreeS/WAN можно применять несколькими способами. Один из них предназначен для постоянного соединения "шлюз-шлюз" и называется одноранговым режимом. Этот режим подходит в случае, когда есть два офиса, желающие безопасно общаться через Интернет. Второй метод называется режимом мобильного пользователя. Он предназначен для удаленных пользователей, желающих безопасно подключаться к вашей ЛВС. Наконец, можно оперировать в режиме шифрования "по возможности", когда шифруются соединения с хостами или шлюзами, которые на это способны. Ниже описано, как задать каждый из этих режимов.

Одноранговый режим

В FreeS/WAN используются имена `Right` (Правый) и `Left` (Левый) для обозначения двух машин, соединяющихся посредством IPsec. Это никак не связано с направлением или расположением и просто позволяет ссылаться на различные стороны IPsec-соединения. По своему выбору назовите одну машину `Left`, а другую - `Right`.

1. Сначала на машине `Right` наберите следующую команду, чтобы получить ее открытый ключ:

```
ipsec showhostkey --right
```

FreeS/WAN выдаст некоторую информацию об IPsec на этой машине, в том числе ее открытый ключ. После знака равенства будет следовать длинный список случайных на вид цифр. Это и есть ключ. Перепишите это число или воспользуйтесь функцией копирования текстового редактора.

- 2. Теперь получите открытый ключ машины Left, применяя ту же самую команду, но с ключом `--left`.
- 3. Перейдите в каталог `/etc/freeswan` и отредактируйте файл `ipsec.conf` (в некоторых дистрибутивах этот файл может храниться в `/etc`). В [табл. 9.3](#) перечислены и описаны параметры, которые необходимо установить в разделе `conn net-to-net`.

Таблица 9.3. Параметры FreeS/WAN

| Параметр | Описание |
|----------------|--|
| Left | IP-адрес шлюза IPsec Left |
| Leftsubnet | Диапазон IP-адресов, прикрываемых шлюзом Left |
| Leftid | Имя хоста в формате полностью квалифицированного доменного имени и со знаком @ перед ним. Например, @gateway.example.com. |
| Leftrsasigkey | Ключ, скопированный ранее из машины Left |
| Leftnexthop | Подразумеваемый шлюз для машины Left. Подразумеваемые настройки должны работать в большинстве случаев |
| Right | То же, что Left выше, но для машины Right |
| Rightsubnet | То же, что Leftsubnet выше, но для машины Right |
| Rightid | То же, что Leftid выше, но для машины Right |
| Rightrsasigkey | То же, что Leftrsasigkey выше, но для машины Right |
| Rightnexthop | То же, что Leftnexthop выше, но для машины Right |
| Auto | Подразумеваемое значение <code>add</code> санкционирует соединение, но не иницирует его, когда загружается система. Если вы хотите, чтобы оно запускалось автоматически, замените значение этого параметра на <code>start</code> |

- 4. Оставьте остальные настройки без изменений и сохраните файл.
- 5. Скопируйте этот файл на другую машину в то же место.
- 6. Примените описанную выше команду `ipsec verify`, чтобы убедиться, что служба IPsec функционирует на обеих машинах.
- 7. Чтобы установить соединение IPsec, наберите

```
ipsec auto --up net-to-net
```

Должно появиться сообщение "IPsec SA established". Если это не так, проверьте настройки или изучите оперативную справку на предмет возможных причин неисправностей.

Если вы применяете межсетевой экран с трансляцией сетевых адресов, то, возможно, придется написать для него специальное правило, чтобы он не транслировал сетевой адрес этой машины. Многие новые модели межсетевых экранов автоматически распознают пакеты IPsec и пропускают их без изменения, поэтому этот дополнительный шаг не требуется.

- 8. Чтобы проверить соединение, попробуйте выполнить эхо-тестирование внутреннего адреса на другой стороне удаленного шлюза. Если будет получен успешный ответ, значит, туннель IPsec построен и работает.
- 9. Если вы на самом деле хотите убедиться, что пакеты шифруются, примените анализатор пакетов, такой как `Tcpdump` или `Ethereal`, чтобы попытаться прочитать эти пакеты. Если анализатор идентифицирует пакеты как пакеты ESP (ESP - один из подпротоколов IPsec), а полезная нагрузка будет выглядеть как тарабарщина, значит все работает как надо.
- 10. Если вы желаете добавить несколько межсетевых соединений, можно просто добавить еще один раздел с новым заголовком, таким как `conn office1-to-office2`. Можно также переименовать исходное соединение `net-to-net`, но оно обязательно должно быть одинаковым в конфигурационных файлах `ipsec` на обеих машинах.

Процедура практически аналогична предыдущей с некоторыми исключениями. В этом режиме под машиной Right понимается локальная машина вашего шлюза IPsec, под Left - машина удаленного пользователя.

1. На удаленной машине отредактируйте тот же файл /etc/freeswan/ipsec.conf с помощью следующего шаблона, аналогичного конфигурации net-to-net с небольшими отличиями.

```
conn road
left=%defaultroute
leftnexthop=%defaultroute
leftid=@tonyslaptop.example.com
leftrsasigkey=0sAQPIP9uI...
right=192.0.2.2
rightsubnet=10.0.0.0/24
rightid=@gateway.example.com
rightrsasigkey=0sAQOnwiBPt...
auto=add
```

В удаленной конфигурации %defaultroute служит для получения вашего динамического IP-адреса.

2. Сторона Right должна содержать информацию для шлюза. Зайдите на машину шлюза и примените следующий шаблон для файла ipsec.conf:

```
conn road
left=192.0.2.2
leftid=@gateway.example.com
leftsubnet=192.0.2.1/24
leftrsasigkey=0sAQOnwiBPt...
rightnexthop=%defaultroute
right=%any
rightid=@tonyslaptop.example.com
rightrsasigkey=0sAQPIP9uI...
auto=add
```

Обратите внимание, что строки в файле на шлюзе переставлены, Left обозначает локальную машину, а Right - удаленную, IP-адрес которой задан как %anys. Это метасимвол, допускающий произвольный IP-адрес, так как он станет известен, лишь когда удаленный пользователь попытается установить соединение.

3. Сохраните этот файл.
4. Все готово к соединению. Проверьте, что IPsec выполняется на машине шлюза, а затем наберите следующую команду на стороне удаленного пользователя:

```
ipsec auto --start road
```

Это, как и раньше, должно инициировать соединение. Если вы не получите сообщение "Ipsec SA established", проверьте настройки или обратитесь к разделу устранения неисправностей на web-сайте FreeS/WAN.

5. Протестируйте и верифицируйте соединение так же, как для конфигурации net-to-net.
6. Можно установить несколько удаленных соединений, как в предыдущей процедуре, и переименовать их содержательным образом.

Оппортунистическое шифрование

Если вы хотите воспользоваться данной возможностью FreeS/WAN, то ваш шлюзовый компьютер не должен располагаться позади межсетевого экрана, применяющего трансляцию сетевых адресов (изменение IP-адреса в заголовках нарушит режим проверки заголовков IPsec). Желательно, чтобы IP-адрес

шлюза был статическим. Шифрование "по возможности" бывает полным или частичным. В полном режиме вы можете инициировать исходящие соединения IPsec, равно как и другие хосты IPsec могут инициировать сеансы оппортунистического шифрования с вашим шлюзом. В частичном режиме инициировать соединение всегда должен ваш шлюз. В обоих режимах требуется, чтобы вы имели доступ к записи DNS для имени хоста, который вы хотите настроить.

Настройка частичного оппортунистического шифрования (только инициирование)

1. Сначала отредактируйте запись DNS для имени хоста, которое будете применять при добавлении элемента для ключа. Запись DNS должна соответствовать идентификатору в файле `ipsec.conf`. В рассмотренном выше примере с мобильным пользователем это `gateway.example.com`. Выполните следующую команду на шлюзовой машине, чтобы создать эту запись:

```
ipsec showhostkey --txt @имя_хоста-шлюза
```

Замените `имя_хоста-шлюза` именем вида `gateway.example.com`.

Будет создан текстовый файл с текстовой записью, содержащей ключ и отформатированной в соответствии с синтаксисом DNS.

2. Вставьте полученную запись в зонный файл этого домена как прямую запись TXT.

Примечание: Если вы не знаете, как редактировать записи DNS, воспользуйтесь помощью администратора DNS. Ошибка в записи DNS может легко привести к отключению всего домена.

Помните также, что распространение изменений по Интернету займет некоторое время. В зависимости от места запроса этот процесс может занять до 48 часов.

3. Убедиться, что сделанные изменения вступили в силу, можно с помощью следующего запроса:

```
ipsec verify --host gateway.example.com
```

Должен прийти ответ OK для прямой записи.

Обратный поиск записи работать не будет, но это допустимо, пока вы не пожелаете применить полное оппортунистическое шифрование. Помните, что хотя вы можете успешно опрашивать сервер DNS, другая сторона вашего соединения на это может быть еще не способна. Там также следует выполнить команду проверки.

4. Когда обе стороны смогут видеть запись DNS, остается только перезапустить службу IPsec, набрав команду

```
service ipsec restart
```

Когда она выполнится, все будет готово к работе.

Это все, что требуется, так как FreeS/WAN автоматически сконфигурирует соединение с помощью информации в записи DNS.

Настройка полного оппортунистического шифрования

Чтобы применять полное оппортунистическое шифрование, необходимо иметь на шлюзе статический IP-адрес и располагать полным контролем над записью DNS для него. FreeS/WAN использует обратный поиск DNS для проверки открытого ключа любой машины, которая пытается подключиться. Инструкции здесь точно такие же, как и для частичного режима, за исключением того, что создается еще и обратная запись DNS для имени шлюзового хоста. Создайте текстовый файл таким же образом, как и выше, и после добавления его как прямой записи, добавьте ее и как обратную, связав ее со статическим IP-адресом. Опять же, если вы не знаете, как редактировать файл DNS, попросите помощи. DNS ошибок не прощает. Когда обе записи будут видны из Интернет, следует перезапустить службу IPsec, и можно будет создавать соединения с хостами, поддерживающими оппортунистическое шифрование IPsec.

Взлом паролей

Вы ознакомились с тем, как различными криптографическими методами защитить свою информацию, и как шифровать файлы, сеансы и целые соединения с другими сайтами. В следующем разделе рассматривается средство, помогающее убедиться, что файлы паролей в безопасности. Речь идет о программе взлома зашифрованных паролей. Она выполняет работу, обратную по отношению ко всем средствам данной лекции в том смысле, что пытается расшифровать файл паролей без каких-либо ключей. Она главным образом применяется к файлам паролей, чтобы гарантировать, что у вас нет паролей, которые легко взломать.

В наше время большинство паролей не хранится на сервере в открытом виде. Хранятся хэши паролей, так что пароли не передаются по сети открытым текстом. Однако в некоторых операционных системах схема хэширования слаба и шифр легко взламывается. В худшем случае, если кто-то перехватит файл паролей, он сможет выполнить атаку методом грубой силы на хэши и выяснить некоторые пароли. Это возможно благодаря склонности многих людей выбирать простые пароли. В большинстве операционных систем к паролям можно предъявлять определенные технические требования, но пользователи все равно будут пытаться обойти ограничения с целью облегчить себе жизнь. Тестирование файлов паролей с помощью программ взлома - единственный способ точно узнать, насколько безопасны пароли пользователей.

John the Ripper: Средство взлома паролей

John the Ripper

Автор/основной контакт: Solar Designer

Web-сайт: <http://www.openwall.com/john>

Платформы: Windows и большинство UNIX

Лицензия: Freeware, аналогично BSD

Рассмотренная версия: 1.6

Утилита John the Ripper была разработана загадочным Солнечным Заговорщиком, чтобы помочь системным администраторам избавиться от слабых паролей, в основном в системах UNIX. Программа использует текстовый файл возможных паролей и проверяет хэш каждого слова из этого файла по файлу паролей. Она даже пробует варианты словарных слов, такие как cat1, cat2 и т.д. После завершения перебора всех слов из текстового файла программа переходит к методам рандомизации и пробует их, пока вы ее не остановите. Она поставляется с файлом базовых слов. Кроме того, вы можете загрузить дополнительные файлы слов для различных операционных систем или создать свои собственные.

Программа доступна для операционных систем UNIX и Windows. Так как она имеет командный интерфейс, то основные операции в обеих системах одинаковы. Ниже описаны процессы установки для Windows и UNIX.

Установка в Windows

1. Загрузите бинарный пакет Windows с Web-сайта или с прилагаемого к книге компакт-диска и распакуйте файл в отдельном каталоге.
2. На самом деле установки как таковой в Windows не требуется. Просто разместите файлы по своему выбору и запускайте программу из этого каталога с помощью соответствующих команд. При желании можно добавить этот каталог в список поиска, если вы хотите запускать John the Ripper из любого места. В противном случае перейдите в каталог john/run, чтобы получить доступ к бинарным файлам, и запустите программу.

Установка в UNIX

1. Загрузите и "растарьте" файлы исходных текстов с web-сайта или с прилагаемого к книге компакт-диска.
2. Выполните следующую команду из каталога src, который будет создан:

```
make
```


Будет выведен список поддерживаемых систем.

Примечание: если ваша система не указана, выполните на следующем шаге команду `make generic` (это должно сработать в большинстве случаев).

3. Выполните следующую команду, подставляя свой тип системы из списка поддерживаемых:

```
make тип_системы
```

Команда соберет программу и поместит основные бинарные программные файлы в каталог `john/run`.

4. Перейдите в этот каталог. Все готово к запуску John the Ripper.

Применение программы John the Ripper

1. Прежде всего, необходимо получить копию файла паролей. В большинстве систем UNIX хэши не хранятся в основном файле паролей, а содержатся в файле, называемом теневым (`shadow` в системах Linux). Это затрудняет несанкционированный доступ к хэшам, так как основной файл паролей пользователей должен быть доступен другим частям операционной системы, поэтому право на его чтение должно предоставляться всем.

Файл хэшей паролей выглядит примерно так, как показано на [листинге 9.1](#).

```
root:$1$%8_pws/,$3ABCmAmVVtBbgXc1EpAZ7.:12080:0:99999:7:::
bin:*:12080:0:99999:7:::
daemon:*:12080:0:99999:7:::
adm:*:12080:0:99999:7:::
lp:*:12080:0:99999:7:::
sync:*:12080:0:99999:7:::
apache:!!:12080:0:99999:7:::
postfix:!!:12080:0:99999:7:::
mysql:!!:12080:0:99999:7:::
tony:$1$bFIb/_R$6RFzrkqq6nY4zTkmWQ8xV0:12080:0:99999:7:::
```

Листинг 9.1. Пример файла хэшей паролей

Случайная по внешнему виду последовательность символов после имени счета - хэш пароля. Именно с ней работает John the Ripper.

2. Текстовый файл `password` в каталоге John the Ripper содержит подразумеваемый список слов. Его можно пополнять, если вы хотите, чтобы были проверены несколько специфических паролей. При желании можно заменить его собственным словарем.
3. Чтобы запустить John the Ripper, наберите команду

```
john проверяемый_файл_паролей
```

Во время работы на экран выдаются пароли, которые удалось взломать. Большинство словарных слов будет просмотрено за несколько минут. Во многих ситуациях это слишком долго, но если вы хотите выполнять программу дольше, чтобы на самом деле протестировать пароли, то процесс можно запустить в фоновом режиме.

Можно также прервать процесс тестирования и вернуться к нему позже. Нажмите `Ctrl+C` один раз, чтобы остановить тестирование и сохранить результаты в файле с именем `john.pot`. Отметим, что в результате двукратного нажатия `Ctrl+C` поиск будет завершен без сохранения результатов.

4. Можно просмотреть взломанные к текущему моменту пароли, вводя:

```
john -show проверяемый_файл_паролей
```

5. Если вы хотите возобновить процесс взлома паролей, воспользуйтесь командой

```
john -restore
```

И это почти все о программе John the Ripper. Счастливого взлома! (Только своих паролей, естественно). Если вы обнаружите слабые пароли, то можно пойти к их владельцам и попросить изменить пароль, или установить на сервере политику, требующую более сильных паролей.

Инструменты безопасности с открытым исходным кодом

10. Лекция: Средства для беспроводных сетей: версия для печати и PDA

До недавних пор сетевые администраторы должны были заботиться в основном о защите физических, фиксированных активов информационных технологий - серверов, маршрутизаторов и межсетевых экранов, составляющих кабельные сети. Однако, с появлением недорогого оборудования для беспроводных сетей возник совершенно новый спектр (и это не игра слов) проблем безопасности.

Новая технология помогла снизить стоимость развертывания сетей, предоставила доступ там, где его раньше не было, и дала возможность трактовать выражение "мобильные вычисления" буквально. Радикально изменился и периметр безопасности сетей всех размеров. Традиционно корпоративные сети соединялись с внешним миром только в нескольких местах ([рис. 10.1](#)). Это позволяло сетевым администраторам концентрироваться на защите этих критических точек доступа, размещая в них межсетевые экраны и другие средства защиты. Внутренность сети в значительной степени считалась доверенной, поскольку не существовало способа попасть туда, минуя защищенные точки.

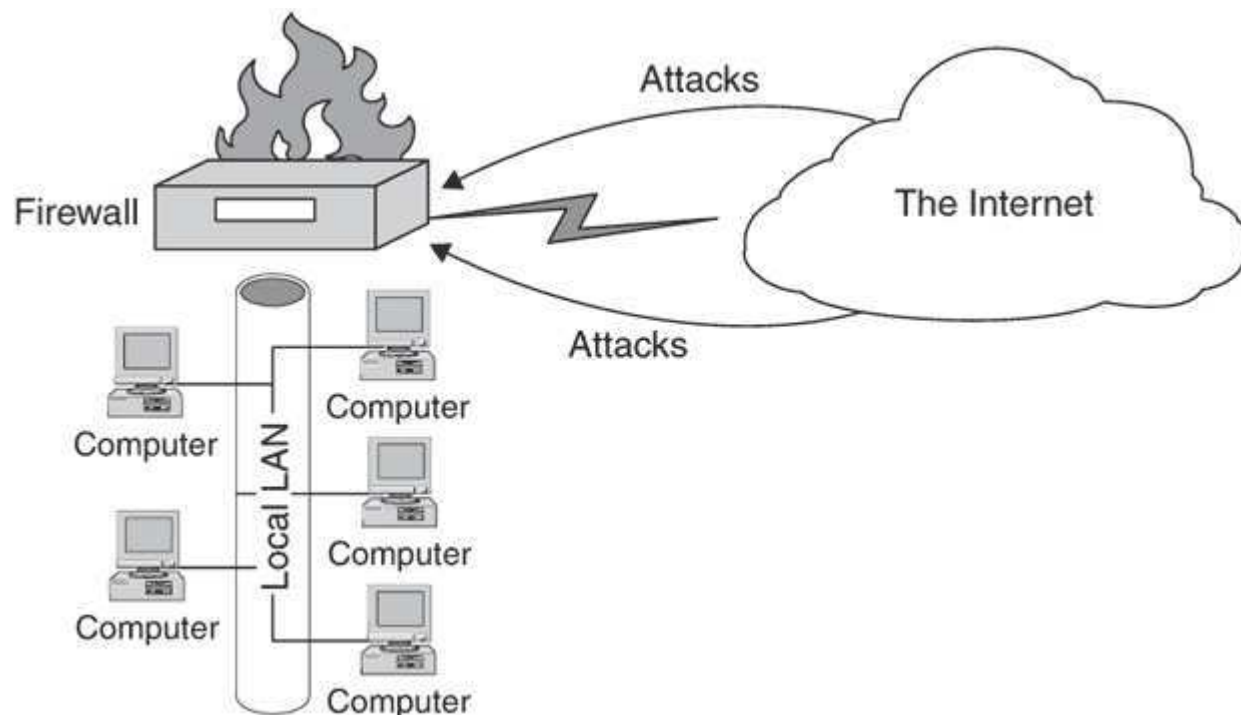


Рис. 10.1. Сетевые угрозы до появления беспроводных сетей

Обзор лекции

Изучаемые концепции:

- Терминология беспроводных ЛВС
- Протоколы 802.11

- Слабые места беспроводных ЛВС
- Оборудование для контроля беспроводных сетей

Используемые инструменты:

NetStumbler, StumbVerter, Kismet Wireless и AirSnort

Развитие технологии снова подняло планку безопасности. При развертывании беспроводной ЛВС новым периметром безопасности в буквальном смысле становится воздух вокруг вас. Такая беда, как беспроводная атака или прослушивание, может прийти "откуда не ждали", с любого направления. Если у вас развернут беспроводной доступ, то кто угодно с платой всего за полсотни долларов в принципе может прослушивать среду передачи вашей сети, даже не ступая на вашу территорию. На [рис. 10.2](#) показан новый периметр сетевой безопасности при применении беспроводных технологий. Можно видеть, что при использовании для части сети беспроводного доступа угрозы безопасности существенно возрастают. Но прежде чем можно будет надежно обезопасить беспроводную сеть, необходимо понять, как функционируют локальные беспроводные сети и где их основные слабые места.

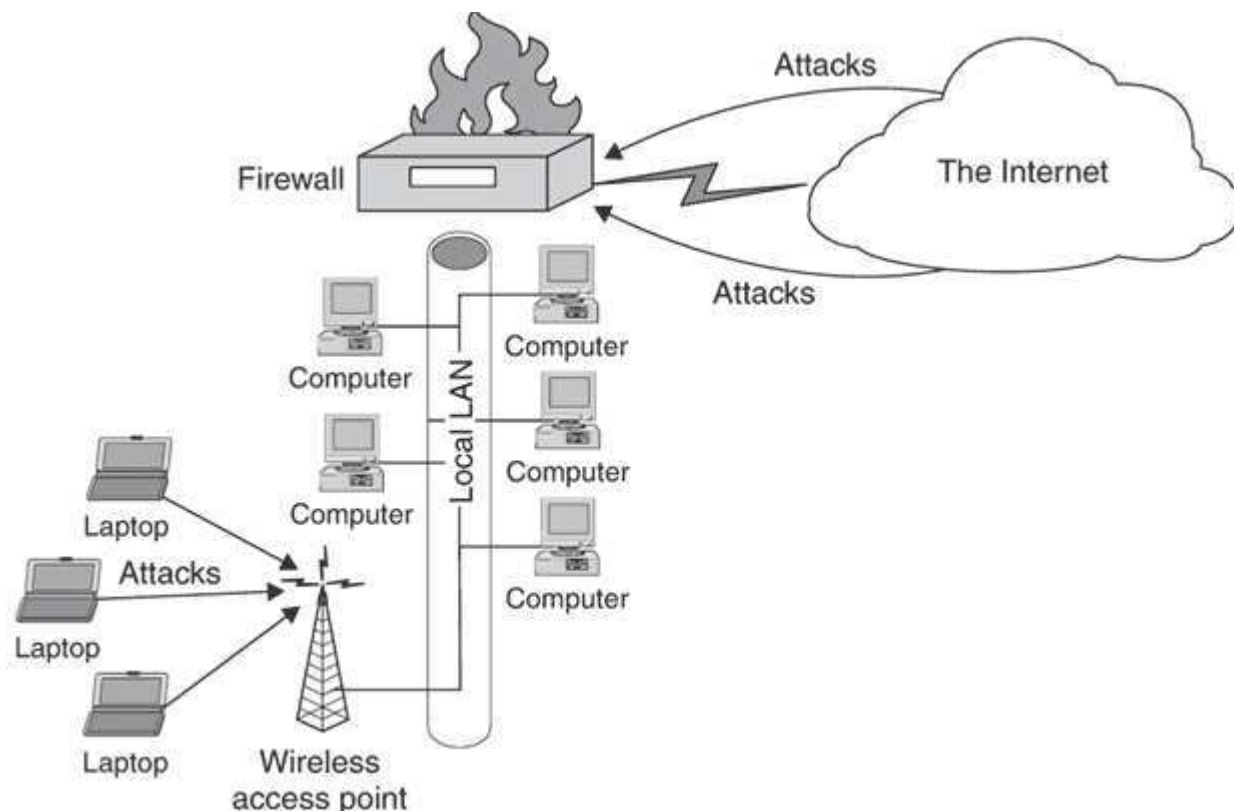


Рис. 10.2. Сетевые угрозы для беспроводных сетей

Производители оборудования для беспроводных ЛВС снизили цены настолько, что это стало разумной альтернативой домашним сетям. Вместо того, чтобы прокладывать в доме кабели Ethernet для соединения своих ПК, можно купить базовую станцию и пару плат для беспроводного соединения и использовать Интернет в любой комнате своего дома. Для участников многих деловых конференций сейчас предлагается бесплатный беспроводной доступ в Интернет. Жители проводят кампании за предоставление бесплатного доступа в Интернет для домовладений вне досягаемости цифровых абонентских линий или кабельных сетей, использующего общественные базовые станции. Широкое развертывание технологии беспроводных ЛВС, несомненно, продолжится, и рано или поздно вам придется иметь с ними дело.

Обзор технологий беспроводных ЛВС

На сегодняшний день наиболее популярный протокол для беспроводных ЛВС - несомненно, семейство спецификаций 802.11 или, в просторечии, Wi-Fi. Стандарты этого семейства по сути являются расширением протокола Ethernet, что обеспечивает отличное взаимодействие с проводными сетями Ethernet. Для передачи сигналов данных применяются частоты 2.4 ГГц для 802.11b и 802.11g, а также 5 ГГц для 802.11a. В США эти частоты принадлежат спектру общего пользования, поэтому не нужно получать лицензию на их использование. Обратная сторона состоит в том, что другие потребительские устройства также могут работать этих частотах. Некоторые беспроводные телефоны и микроволновые печи также попадают в полосу 2.4 ГГц, поэтому если в окрестности есть подобные устройства или другие сети Wi-Fi, вы можете столкнуться с некоторыми помехами.

Выбранные длины волн прекрасно подходят для ближней связи, которая и требуется для сетей Wi-Fi. Проектные параметры обеспечивают радиус действия 50 метров в помещении и более 250 метров на открытом пространстве при нормальных условиях. Однако с мощной антенной в зоне прямой видимости можно увеличить расстояние до 30 км, что отлично подходит для городских междофисных коммуникаций (предполагается отсутствие гор и доступ на крышу многоэтажного здания). В [табл. 10.1](#) описаны четыре разновидности существующих стандартов беспроводной связи 802.11.

Таблица 10.1. Стандарты беспроводной связи 802.11

| Стандарт | Описание |
|----------|---|
| 802.11a | В этой версии стандарта применяется частота 5 ГГц, принадлежащая менее используемой части спектра. Следовательно, помехи менее вероятны. Теоретический потенциал этой технологии составляет 54 Мбит/с, что является очень широкой полосой пропускания, но большинство реальных приложений до теоретического максимума не дотягивают |
| 802.11b | В настоящее время это самый популярный стандарт беспроводной связи. В нем применяется частота 2,4 ГГц, на которой работают Bluetooth и другие потребительские устройства. Он предлагает полосу пропускания до 11 Мбит/с, хотя практические приложения при неоптимальных условиях обычно получают примерно половину этого |
| 802.11g | Более новый стандарт предоставляет полосу пропускания до 54 Мбит/с, но на той же частоте 2,4 ГГц, что и стандарт 11b. Он также обратно совместим с оборудованием 11b |
| 802.11i | Этот новый протокол является, по сути, расширением 802.11b с исправлениями протокола шифрования, обеспечивающими значительно более высокий уровень безопасности. Он только недавно был одобрен IEEE, и использующие его продукты должны появиться в конце 2004 г. |

Терминология Wi-Fi

Имеется два вида беспроводных сетей. В произвольных сетях узлы соединяются напрямую. Это полезно, если вы хотите объединить несколько ПК и вам не требуется доступ в ЛВС или Интернет. Сети с инфраструктурой опираются на базовые станции (точки доступа), соединенные с вашей ЛВС. Все узлы подобной сети подключаются к ЛВС через базовую станцию. Это наиболее распространенная конфигурация в корпоративных сетях, так как она позволяет администратору централизованно контролировать беспроводной доступ. Каждой точке беспроводного доступа и плате присвоен номер, называемый идентификатором базового набора сервисов (Basic Service Set ID - BSSID). Это - MAC-адрес беспроводной стороны точки доступа. У точки доступа есть также идентификатор набора сервисов (Service Set Identifier - SSID). Это - имя беспроводной сети, с которой ассоциируются все узлы. Это имя не обязано быть уникальным среди точек доступа. На самом деле большинство производителей присваивают точкам доступа подразумеваемые идентификаторы, поэтому их можно использовать прямо из коробки. Идентификатор набора сервисов точки доступа необходим для подключения к сети. Некоторые базовые станции обладают дополнительной функциональностью, играя роль маршрутизаторов и встроенных серверов DHCP. Существуют даже интегрированные устройства, действующие как точка беспроводного доступа, межсетевой экран и маршрутизатор для домашних и малых сетей.

Узел беспроводной сети создается путем установки в компьютер беспроводной сетевой интерфейсной платы. Выпускается несколько видов подобных плат. Это может быть плата, которая вставляется в слот ПК, плата PCMCIA, внешнее устройство USB, а теперь даже компактный флэш-формат для малых слотов КПК. Беспроводная сеть 802.11 с инфраструктурой содержит точки доступа, действующие как мост между кабельной ЛВС Ethernet и одной или несколькими оконечными беспроводными точками. Точка доступа с определенной частотой включает широкоэвещательный "радиомаяк", чтобы оповещать окрестные беспроводные узлы о своем присутствии. Широкоэвещательные сигналы приглашают зарегистрироваться любые беспроводные узлы в данной области и являются одной из проблем Wi-Fi. Невозможно полностью выключить эти сигналы и таким образом скрыть факт наличия беспроводной сети в офисе. Кто угодно с платой беспроводного доступа может по крайней мере видеть сигналы радиомаяка, если находится вблизи, хотя некоторые устройства позволяют ограничить объем информации, содержащейся в этих широкоэвещательных сообщениях.

Эти сигналы содержат основную информацию о точке беспроводного доступа, включая, как правило, SSID (рис. 10.3). Если в сети не применяется шифрование или другие средства защиты, то этого достаточно для присоединения к сети. Однако даже в беспроводной сети с шифрованием SSID часто передается в открытую, а зашифрованные пакеты могут перехватываться по эфиру и подвергаться попыткам взлома.

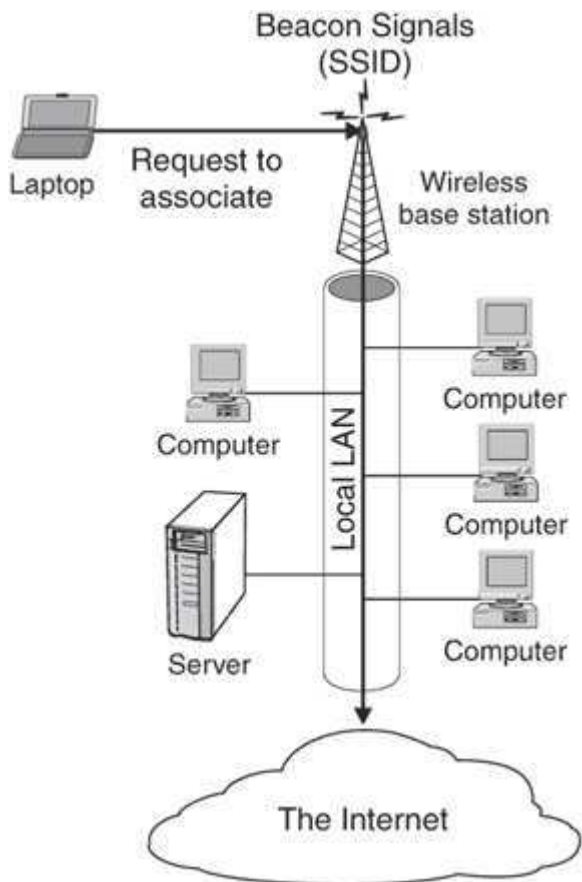


Рис. 10.3. Работа беспроводной сети

Опасности беспроводных ЛВС

Беспроводные сети характеризуются гибкостью и функциональностью, отсутствующей у кабельных ЛВС, но в то же время создают ряд угроз и новые проблемы для сетевого администратора, отвечающего за безопасность. Ниже рассмотрены некоторые аспекты, которые следует учитывать при добавлении в инфраструктуру беспроводных ЛВС.

Прослушивание

Для хакера в случае применения беспроводной сети не составит никакого труда собирать пакеты с помощью беспроводного сетевого анализатора. С этим мало что можно сделать, разве что окружить здание свинцовым экраном! Создатели беспроводных сетей учитывали это и встроили в проект стандарт шифрования, называемый WEP (Wired Equivalent Privacy - секретность, эквивалентная проводной), чтобы данные можно было шифровать. К сожалению, фундаментальным недостатком в организации работы алгоритма является его потенциальная взламываемость (один из инструментов далее в этой лекции демонстрирует это). Поэтому даже при применении WEP данные, передаваемые по беспроводной сети, потенциально подвержены несанкционированному просмотру. Кто-нибудь может прослушивать беспроводные соединения, выискивая входные имена, пароли и другие данные.

Доступ к ПК с беспроводными платами

Беспроводной канал дает потенциальным злоумышленникам наводку на машину вашей сети. Помимо точек доступа, извне могут быть видны и машины с беспроводными сетевыми платами. Используя этот способ доступа, можно развернуть атаку против машины, которая, вероятно, не защищена межсетевым экраном и не укреплена, как ваши средства защиты периметра или общедоступные серверы.

Доступ к ЛВС

Это, вероятно, наибольшая опасность, которую создают беспроводные сети. Если хакеры смогут получить доступ к вашей ЛВС через базовую станцию, то можно считать, что ключи от вашего королевства у них в кармане. В большинстве ЛВС функционирует без всяких ограничений сервер DHCP, поэтому хакеры могут получить законный IP-адрес и начать исследовать вашу сеть. Затем они могут запустить сканер уязвимостей или сканер портов, например, Nessus или Nmap, чтобы найти представляющие для них интерес машины и дыры в их защите, поддающиеся эксплуатации.

Анонимный доступ в Интернет

Даже если хакеров не интересует ваша ЛВС, они могут использовать вашу полосу пропускания для других незаконных целей. Входя в вашу сеть и затем выходя в Интернет, они могут осуществлять противоправные действия, не оставляя при этом своих следов. Любая атака или мошенничество, совершенные через это соединение, будут прослежены до вашей сети. Правоохранительные органы будут стучать в вашу дверь, а не в их. Такой метод станет более распространенным, когда хакеры осознают, как трудно проследить атаки, начинающиеся таким образом. Слишком мала вероятность перехвата злоумышленника из беспроводной сети, если только не применять заранее размещенное дорогостоящее триангуляционное оборудование. Незащищенные беспроводные ЛВС предлагают хакерам лучший анонимный доступ, какой только можно себе представить.

Специфические уязвимости 802.11

Кроме основных дыр в безопасности беспроводных ЛВС, имеется ряд проблем, специфичных для стандарта 802.11. Некоторые из них связаны с ошибками проектирования, допущенными производителем, или с подразумеваемыми конфигурациями. Другие объясняются проблемами в общей архитектуре стандарта.

Подразумеваемые идентификаторы набора сервисов

Каждая базовая станция Wi-Fi имеет специальный идентификатор, который необходимо знать, чтобы войти в сеть. При правильной реализации это обеспечивает некоторый уровень безопасности. К сожалению, многие забывают изменить подразумеваемый идентификатор набора сервисов, заданный производителем. Очень легко найти сети с подразумеваемыми SSID производителя, такими как linksys, default и т.д. Когда хакер это видит, он может предположить, что администратор тратит не слишком много времени на настройку и защиту беспроводной сети.

Вещание радиомаяка

Вещание радиомаяка - врожденная патология беспроводных сетей. Базовая станция должна регулярно извещать сигналом о своем существовании, чтобы радиоприемник конечного пользователя мог ее найти и договориться о сеансе связи, а поскольку устройства законных пользователей еще не были аутентифицированы, этот сигнал должен вещаться в открытую. Он может быть перехвачен кем угодно, и, как минимум, будет известно, что у вас имеется беспроводная ЛВС. Многие модели позволяют отключать содержащую SSID часть этого вещания, чтобы хоть чуть-чуть затруднить беспроводное подслушивание, но SSID, тем не менее, посылается при подключении, поэтому все равно существует небольшое окно уязвимости.

Применение по умолчанию нешифруемых коммуникаций

Большинство современных беспроводных сетевых устройств предлагают возможность включения встроенного стандарта беспроводного шифрования WEP. Проблема в том, что обычно его надо включать вручную; по умолчанию оно, как правило, отключено. Многие администраторы настраивают беспроводную сеть в спешке и не находят времени на активацию этой важной возможности. Если сеть настраивает не технический специалист, то почти наверняка шифрование не будет включено. Имеется также проблема управления секретными ключами пользователей, так как в WEP каждый пользователь разделяет свой секретный ключ с базовой станцией. Администрирование большого числа пользователей с беспроводным подключением может стать сущим кошмаром.

Слабые места WEP

Даже когда встроенное шифрование задействовано, остается риск, что сигнал будет прочитан. В реализации алгоритма шифрования в WEP имеются фундаментальные дефекты, позволяющие взломать его после перехвата определенного объема данных. Эти дефекты связаны со способом порождения ключей. В WEP слабы векторы инициализации, а частота их использования высока, так что со временем становится возможным взлом ключа. Когда шифрование взломано, атакующий сможет не только читать весь трафик, проходящий по беспроводной сети, но и, вероятно, войти в сеть. Поэтому, хотя WEP и предлагает некоторую базовую защиту против случайного прослушивания, любой серьезный злоумышленник наверняка запасся программным обеспечением, позволяющим при необходимости взломать шифрование.

Феномен "агрессивного объезда"

Поиск незащищенных беспроводных ЛВС стал популярным развлечением среди хакеров и любителей беспроводной связи. По аналогии с тем, как раньше хакеры осуществляли массовый или агрессивный обзвон случайного набора телефонных номеров, чтобы найти активные модемы, поиск незащищенных беспроводных ЛВС называли агрессивным объездом. Чаще всего беспроводные хакеры ездят по округе с беспроводной платой и программным обеспечением в надежде поймать сигнал сети. Программное обеспечение может зафиксировать точное расположение беспроводной сети с помощью системы глобального позиционирования (GPS), а также массу другой информации, такой как применение шифрования или отсутствие такового. Если в беспроводной ЛВС не используется шифрование или другие защитные средства, то хакеры смогут попутешествовать в Интернете или исследовать локальную сеть через беспроводной канал. Для этого не нужно большого мастерства, что и привлекает хакеров разного уровня.

Организации, использующие беспроводные сети в плотно застроенной среде вокруг своих офисов или вблизи крупных дорог, больше всего рискуют пострадать от подобной активности. В "группу риска" входят офисы в жилых и деловых районах города, где много высотных зданий. У беспроводных сетей, построенных по стандарту 802.11b, эффективный радиус действия составляет пару сотен метров. Почти наверняка это больше, чем расстояние до соседнего здания, не говоря уже о расстоянии между этажами в многоэтажном здании. В скученном деловом центре несколько незащищенных беспроводных ЛВС внутри одного здания - не редкость. С точки зрения безопасности, высотные здания - одно из худших мест для применения беспроводных ЛВС. Типичное здание со стеклянными окнами позволяет сигналам ЛВС распространяться на немалое расстояние. Если вблизи имеются другие здания, то почти наверняка в них можно перехватить некоторые сигналы. Здания в жилом районе в этом плане еще хуже. Представьте себе подростков и других бездельников, с удобствами сканирующих доступные беспроводные ЛВС прямо из своей спальни.

Недавнее исследование показало, что более 60% беспроводных ЛВС абсолютно не защищены. Хакеры даже помещают найденные точки беспроводного доступа в оперативные базы данных с картами, чтобы каждый мог найти открытые беспроводные ЛВС почти в любом месте страны. Они классифицируют их по типу оборудования, применению шифрования и т.д. Если ваша беспроводная ЛВС расположена в деловом центре крупного города, то почти наверняка она попала в подобную базу, и дело лишь за тем, чтобы какой-нибудь окрестный хакер нашел для нее немного свободного времени. Ниже приведены некоторые из оперативных баз данных, где можно проверить, не попала ли беспроводная ЛВС вашей организации в эти списки:

- <http://www.shmoo.com/gawd/>
- <http://www.netstumbler.com/nation.php>

Отметим, что большинство сайтов удалят название вашей организации из списка, если вы попросите об этом.

Оценивание безопасности беспроводной сети

Проще всего сказать, что из-за угроз безопасности беспроводных сетей вообще не следует предоставлять беспроводной доступ к вашей сети. Однако это все равно, что посоветовать вам спрятать голову в песок в надежде, что опасность вас минует. Беспроводной доступ - это не преходящая мода, это одна из наиболее активно развивающихся областей технологии, в которую делаются значительные инвестиции. Производители в жутком темпе, по все более низким ценам "выбрасывают" на рынок массу беспроводных адаптеров для всевозможных устройств. Многие компании розничной торговли, такие как McDonald's и Starbucks, устанавливают точки беспроводного доступа в своих магазинах для привлечения покупателей. В ПК-блокноты Intel Centrino встроена поддержка беспроводных сетей. Ваши пользователи жаждут свободы, которую приносит технология беспроводных ЛВС. Им нужна возможность входа в сеть со своих поддерживающих беспроводную связь ПК-блокнотов, всегда и везде. Это значит, что вам рано или поздно придется иметь дело с

безопасностью беспроводных сетей. Средства из данной лекции помогут вам оценить и, при необходимости, повысить безопасность беспроводной сети. Они также помогут вам развернуть беспроводную ЛВС более безопасным образом, если вы делаете это впервые.

Выбор оборудования

Чтобы оценить безопасность беспроводной сети, необходимо иметь как минимум беспроводную сетевую плату, машину для работы и некоторое программное обеспечение.

Беспроводные платы

Большая часть программного обеспечения, рассмотренного в этой лекции, - свободное, но необходимо купить хотя бы одну беспроводную сетевую плату. Выбор производителей широк, и цены вполне конкурентоспособны. За типичную плату придется выложить от \$40 до \$80. Следует тщательно выбирать производителя и модель, так как не все платы работают со всеми пакетами беспроводного ПО.

По сути, имеется три различных набора микросхем для устройств в стандарте 802.11b. Набор микросхем Prism II компании Intersil является, вероятно, наиболее распространенным и используется компанией Linksys, крупнейшим производителем потребительских беспроводных плат. Набор микросхем Lucent Hermes применяется в платах WaveLAN и ORiNOCO и ориентирован, в основном, на корпоративное оборудование. Cisco располагает собственным набором микросхем, обладающим некоторыми специфическими защитными возможностями. Платы Prism II будут работать с Kismet Wireless, программным обеспечением Linux, рассмотренным в этой лекции, но не на платформе Windows. Платы D-Link работают с Windows, но не с широко доступным инструментарием безопасности Windows. Может быть важен выбор конкретной модели определенного производителя. В старых платах Linksys USB применялся другой набор микросхем, и они не очень хорошо работают на Linux.

В довершение всего этого беспорядка некоторые новые протоколы еще не поддерживаются многими пакетами. Текущие версии программных пакетов, рассмотренные в этой лекции, не поддерживают новый стандарт 802.11g. Основным производителям еще предстоит выпустить интерфейсный код, чтобы разработчики программного обеспечения могли приняться за работу. Через некоторое время после того, как они это сделают, станут доступны драйверы. Вы должны изучить информацию на web-сайтах соответствующих программ, прежде чем покупать оборудование для поддерживаемых плат и протоколов. При написании данного обзора применялась плата ORiNOCO Gold PCMCIA, которая хорошо взаимодействует с программным обеспечением Windows и Linux.

Аппаратное и программное обеспечение

В качестве аппаратного обеспечения, на которое будут загружаться программы, годится почти любая машина нормальной мощности. Программное обеспечение для UNIX прекрасно работает на PII 300 с ОЗУ 64 МБ. Программное обеспечение Windows также должно работать на такой системе. Несомненно, программы следует загрузить на ПК-блокнот, так как вы будете с ним перемещаться. Существует версия Kismet Wireless для Palm OS и версия NetStumbler для Pocket PC, так что можно даже поместить программы в КПК. В наше время доступны беспроводные платы для обеих основных платформ (Palm и Pocket PC) малых КПК, способных воспользоваться этим программным обеспечением.

Необходимо также убедиться, что имеется достаточно свободного дискового пространства, если вы собираетесь взламывать ключи WEP. Требуется примерно от 500 МБ до нескольких ГБ. Не оставляйте машину без присмотра при прослушивании беспроводных данных, если нет достаточного объема свободного дискового пространства - можно легко заполнить весь жесткий диск, что приведет к аварийному останову компьютера.

Если вы проводите аудит своего беспроводного периметра и хотите знать точное местоположение, то можно приобрести небольшой карманный приемник GPS. Проверьте, что ваше устройство GPS имеет NMEA-совместимый последовательный кабель для взаимодействия с ПК-блокнотом. С помощью этого оборудования вы сможете определить точные координаты мест, откуда доступны ваши точки беспроводного доступа. Рассмотренные в этой лекции продукты способны получать данные GPS непосредственно из приемников и вставлять их в результирующую выдачу. Наконец, если у вас получится применить совместимое с GPS программное обеспечение создания карт, такое как Microsoft MapPoint, то вы сможете начертить вполне приличные карты своей оценочной активности.

Антенны

Для беспроводного "вынюхивания" встроенной антенны в непосредственной близости от офиса большинства плат будет вполне достаточно. Однако, если вы действительно хотите проверить, насколько вы уязвимы извне, вам понадобится внешняя антенна, позволяющая определить пределы досягаемости вашей беспроводной сети. В конце концов, плохие парни способны смастерить самодельную антенну дальнего действия из банки из-под чипсов Pringles и

куска провода. Вы можете купить недорогие антенны профессионального уровня в различной комплектации. Я купил набор, включающий плату ORiNOCO и внешнюю антенну, подходящую для установки на крышу автомобиля.

Это еще одна причина для тщательного выбора беспроводной платы. Некоторые платы позволяют присоединять внешние антенны, другие - нет. Необходимо проверить, что на приобретаемой плате есть порт для антенны, если вы собираетесь оценивать беспроводную сеть. Известно, что платы ORiNOCO, Cisco, Samsung и Proxim поддерживают внешние антенны.

Теперь, получив некоторые базовые знания и необходимое оборудование, перейдем к рассмотрению свободного программного обеспечения, позволяющего производить оценку беспроводной сети (вашей, естественно).

NetStumbler: Программа обнаружения беспроводной сети для Windows

NetStumbler

Автор/основной контакт: Marius Milner

Web-сайт: <http://www.netstumbler.com/>

Платформа: Windows

Лицензия: Freeware

Рассмотренная версия: 0.3.30z

Форум NetStumbler: <http://www.netstumbler.org/>

NetStumbler, вероятно, - наиболее употребительное средство для оценивания беспроводной сети, в основном потому, что оно свободно и работает на платформе Windows. На самом деле, оно настолько популярно, что его название стало синонимом "агрессивного объезда", как в выражении "Вчера вечером я нетстумблил". Подозреваю, что автор так назвал свою программу, потому что при работе с ней он "случайно" наткнулся на беспроводные сети (stumble - наткнуться, спотыкаться; русское слово "спотыкач" тоже вызывает осмысленные ассоциации - прим. перев.).

NetStumbler не считается программой с полностью открытыми исходными текстами, потому что в настоящее время автор еще не сделал их доступными. Однако она условно свободна и заслуживает упоминания в силу распространенности. Для нее написано много дополнений с открытыми исходными текстами (одно из них обсуждается далее в этой лекции). Кроме того, сообщество пользователей и web-сайт NetStumbler полностью соответствуют идеологии открытого ПО. Web-сайт весьма информативен и предоставляет много хороших ресурсов по безопасности беспроводных сетей, помимо самой программы. Имеется также база данных топографической съемки, куда пользователи NetStumbler вводят данные о точках доступа, которые они обнаружили при применении программы. Если беспроводная сеть вашей организации находится в базе данных и вы хотите, чтобы она была удалена, ваше желание с удовольствием исполнят.

Установка NetStumbler

1. Перед установкой NetStumbler убедитесь, что для беспроводной сетевой платы установлены правильные драйверы. В новых версиях Windows, таких как 2000 и XP, это обычно весьма просто. Установите программное обеспечение, которое поставляется с вашей платой, и система должна автоматически распознать плату и позволить ее сконфигурировать. Поддержка для Windows 95 и 98 может быть ненадежной. Проверьте документацию платы по особенностям применения.
2. Когда плата включена и работает, проверьте ее, попытавшись получить доступ в Интернет через точку беспроводного доступа. Если вы способны видеть внешний мир, то к установке NetStumbler все готово.
3. Процесс установки NetStumbler столь же прост, как и установка любой программы для Windows. Загрузите файл с прилагаемого к книге компакт-диска или с сайта <http://www.netstumbler.com/> и распакуйте его в отдельном каталоге.
4. Выполните файл setup в этом каталоге, и начнется обычный процесс установки в Windows.

Когда установка будет завершена, можно запустить NetStumbler.

Применение NetStumbler

После запуска NetStumbler отображается основной экран (рис. 10.4).

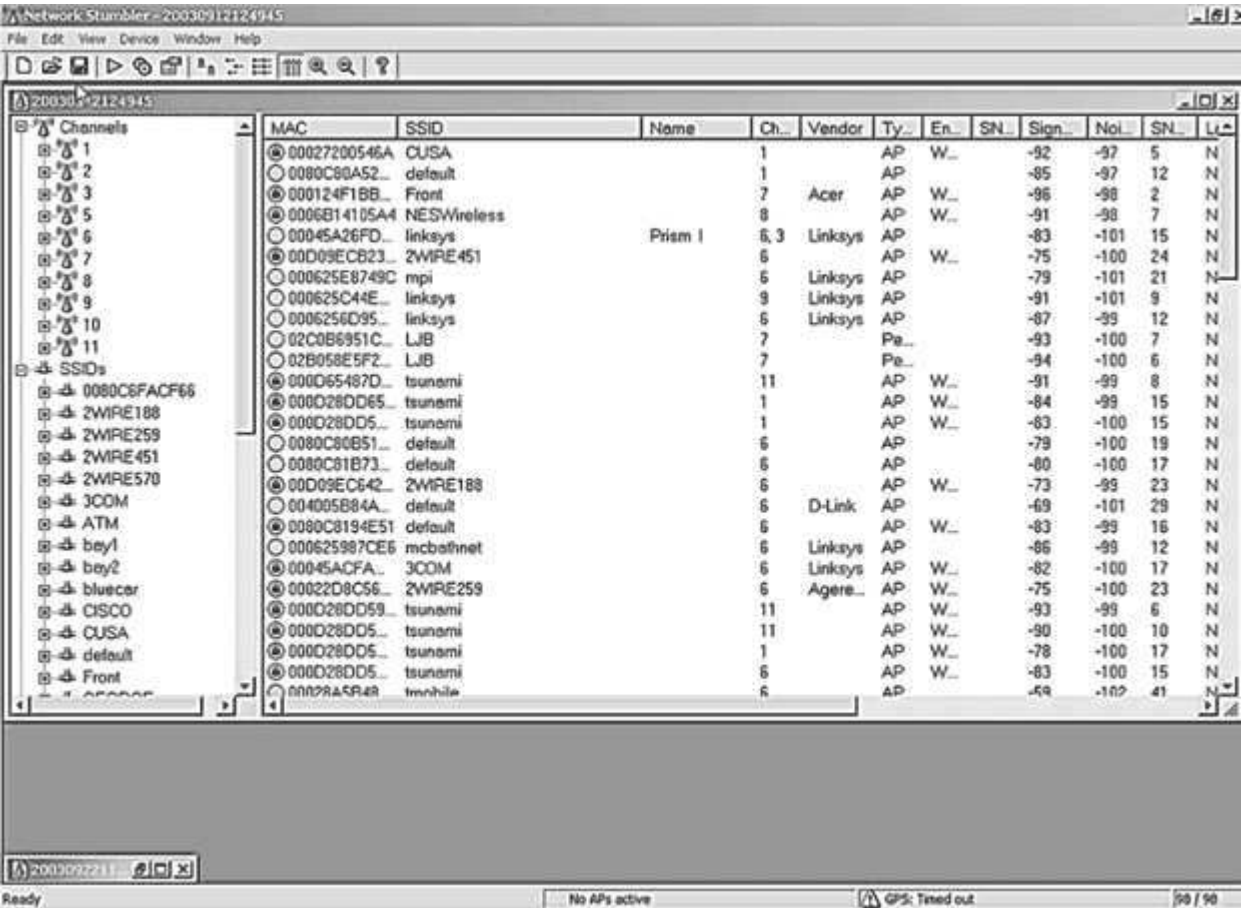


Рис. 10.4. Основной экран NetStumbler

В столбце MAC можно видеть список точек доступа, обнаруженных NetStumbler. Иконки сетей слева от MAC-адреса окрашены в зеленый цвет, если они в настоящее время в зоне досягаемости. По мере удаления от сети иконки становятся сначала желтыми, а затем красными. Иконки неактивных сетей будут серыми. Если в сети применяется шифрование, то отобразится замочек в кружке. Это позволяет быстро понять, в каких сетях используется WEP. NetStumbler собирает дополнительные данные о любой обнаруживаемой точке. В табл. 10.2 описаны выводимые поля данных и их смысл.

Таблица 10.2. Поля данных NetStumbler

| Поле данных | Описание |
|-------------|--|
| MAC | BSSID или MAC-адрес базовой станции. Это уникальный идентификатор, присвоенный производителем. Он полезен, когда у вас много станций с одним и тем же подразумеваемым SSID производителя, таким как linksys. |

| | |
|------------|--|
| SSID | Идентификатор набора сервисов, с которым настраивается каждая точка доступа. Он определяет беспроводную сеть и необходим для входа в нее. NetStumbler охотно извлечет его для вас из сигналов радиомаяка. Как отмечено в описании поля MAC, это не обязательно уникальный идентификатор, так как другие базовые станции могут иметь такое же значение SSID. Возможны проблемы, если две организации в одном здании используют одинаковые подразумеваемые значения SSID. В таком случае может случиться, что служащие используют сеть или выход в Интернет другой организации |
| Name | Необязательное описательное имя точки доступа. Иногда производитель задает его. Владелец сети может его редактировать; например, Acme Corp Wireless Network. Иногда лучше оставить это поле пустым, если вы не хотите, чтобы посторонние, исследующие сетевой радиозфир, узнали, что эта точка доступа принадлежит вам |
| Channel | Канал, в котором оперирует базовая станция. Если вы столкнулись с помехами, изменение этого значения у точки доступа может их устранить. Большинство производителей используют подразумеваемый канал. Например, для точек доступа Linksys подразумеваемым служит шестой канал |
| Vendor | NetStumbler с помощью BSSID пытается идентифицировать производителя и модель выявленного беспроводного оборудования |
| Type | Указывает, была ли найдена точка доступа, узел сети или устройство какого-то другого типа. Обычно будут находиться точки доступа, которые обозначаются AP. Беспроводные узлы показываются как Peer. Именно поэтому, даже без настроенной беспроводной сети, наличие в вашем ПК беспроводной платы может быть рискованным. В наше время многие ПК-блокноты поставляются со встроенными беспроводными передатчиками, поэтому желательно их отключить, если пользователи не собираются их применять |
| Encryption | Показывает, какой тип шифрования используется в сети (если используется). Это очень важно, поскольку, если сеть не шифруется, то посторонние могут извлечь ваш сетевой трафик прямо из эфира и прочитать его. Они могут также войти в вашу сеть, если отсутствуют другие средства защиты |
| SNR | Отношение сигнал/шум. Характеризует уровень помех и шума на входе приемника беспроводной платы |
| Signal | Уровень мощности сигнала на входе приемника |
| Noise | Уровень шума на входе приемника |
| Latitude | Широта, если вы применяете вместе с NetStumbler приемник GPS |
| Longitude | Долгота, если вы применяете вместе с NetStumbler приемник GPS |
| First seen | Показания системных часов, когда был впервые принят сигнал радиомаяка сети |
| Last seen | NetStumbler обновляет это значение всякий раз, когда вы входите в зону приема точки доступа |
| Beacon | Частота посылки сигнала радиомаяка в миллисекундах |

По мере проведения аудита, основной экран NetStumbler заполнится обнаруживаемыми беспроводными сетями. Вероятно, вас удивит количество сетей, которые проявятся вокруг вашего офиса. Еще больше вас удивит число сетей с выключенным шифрованием и подразумеваемыми идентификаторами набора сервисов.

В левой стороне экрана отображаются обнаруженные сети. Можно упорядочить их с помощью различных фильтров, выбрать их по каналам, SSID и нескольким другим критериям. Можно задать фильтры, чтобы отображались только сети, в которых шифрование включено или выключено, инфраструктурные или одноранговые (произвольные), допускающие CF-опрос (предоставление дополнительной информации при запросе) или нет, с подразумеваемым или переустановленным значением SSID.

На панели в нижней части основного экрана можно видеть состояние своей беспроводной сетевой платы. Если она функционирует нормально, то вы увидите иконку, мигающую примерно каждую секунду, и количество видимых в данный момент активных точек доступа. Если возникает проблема с интерфейсом между сетевой платой и программным обеспечением, то вы увидите это здесь. С правой стороны нижней панели находятся координаты GPS, если используется устройство глобального позиционирования.

Мигание показывает, как часто вы опрашиваете точки доступа. NetStumbler - средство активного сканирования сетей, поэтому он постоянно посылает пакеты "Hello", чтобы проверить, ответит ли какая-нибудь беспроводная сеть. Другие беспроводные средства, такие как Kismet (см. далее в этой лекции), являются пассивными, так как они только принимают сигналы радиомаяка. Недостаток активных средств - возможный пропуск некоторых точек доступа,

настроенных не отвечать на опрос; достоинство же в том, что некоторые точки доступа посылают сигналы радиомаяка так редко, что с помощью пассивных средств вы можете никогда их не поймать. Помните также, что активный опрос может вызывать срабатывание беспроводных систем обнаружения вторжения, однако очень немногие организации применяют подобные системы, а если вы используете NetStumbler только как средство оценивания собственной сети, то скрытность не должна быть важна.

Если в этом режиме щелкнуть мышью на какой-либо сети, будет показан график отношения сигнал/шум за период времени, когда вы наблюдали сеть. Это позволяет увидеть, насколько силен сигнал в различных областях ([рис. 10.5](#)).

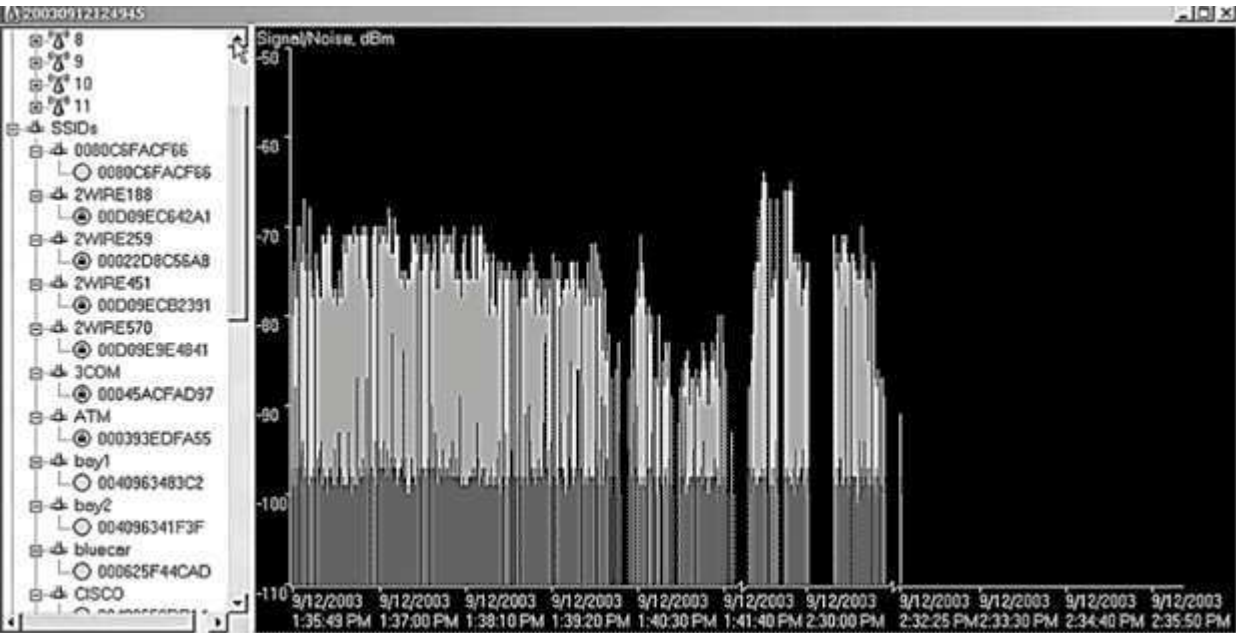


Рис. 10.5. График сигнала NetStumbler

Опции NetStumbler

Чтобы вывести диалоговое окно для задания опций NetStumbler, выберите подменю Options в меню View. В [табл. 10.3](#) перечислены вкладки и возможные значения.

Таблица 10.3. Опции NetStumbler

| Вкладка | Описание |
|-----------|--|
| General | Задаёт частоту опроса точек доступа. Можно также задать автоматическую подстройку с учетом вашей скорости, если применяется система глобального позиционирования. Имеется опция автоматического реконфигурирования вашей платы, когда найдена новая сеть, но вы, вероятно, не захотите этого делать в насыщенной области: если вокруг много точек доступа, то конфигурация платы будет изменяться каждые несколько секунд и это замедлит работу компьютера. Кроме того, программное обеспечение может сконфигурировать плату для чужой сети, и тогда вы неумышленно станете нарушителем (см. врезку "Рекомендации по эффективному и этичному аудиту беспроводных сетей") |
| GPS | Настраивает приемник GPS для взаимодействия с NetStumbler. Я использовал карманное устройство GPS Meridian с последовательным кабелем. Пришлось задать лишь правильный порт и параметры коммуникации, и NetStumbler сразу начал импортировать данные |
| Scripting | Настройка вызова внешних процедур. Можно использовать Visual Basic или любые другие языки на Windows-платформе для выполнения дополнительных действий с выдачей NetStumbler. Внешние программы также могут использовать эту функциональность |

| | |
|------|--|
| MIDI | Можно настроить NetStumbler для проигрывания отношения сигнал/шум как файла MIDI. Не думаю, что это стоит делать в области с множеством сетей, поскольку может стать очень шумно, но предполагаю, что поиск ускользающего сигнала по звуку способен быть эффективным |
|------|--|

Рекомендации по эффективному и этичному аудиту беспроводных сетей

Получите разрешения

Не забудьте получить разрешение руководства на проведение оценки беспроводной сети. Если вы - внешний консультант, то должны иметь письменное разрешение или подписанное высшим руководством соглашение. Если организация не владеет зданием, руководство должно согласовать это со службой безопасности здания, чтобы вам было разрешено находиться в помещениях.

Определите периметр беспроводной сети

Обойдите внешние границы и определите, как далеко распространяется ваш сигнал. (Хорошим практическим правилом служит обход только общедоступных мест, которые могут использоваться беспроводными взломщиками или агрессивными ездоками). Если возможно, достаньте карту и нанесите на нее свой беспроводной периметр.

Начните вне области приема, которую вы считаете нормальной, и продвигайтесь внутрь по спирали, описав сначала широкую дугу вокруг ваших рабочих помещений и попытайтесь определить, как далеко распространяется сигнал. Затем вернитесь назад и сделайте еще более широкий круг, чтобы проверить, не простираются ли некоторые зоны приема еще дальше.

Иногда особенности ландшафта или рукотворные объекты способны причудливым образом расширять распространение сигнала: он может отражаться или фокусироваться зданиями, рекламными щитами, деревьями и другими объектами. Исходите из предположения, что агрессивные ездоки этим воспользуются.

После определения периметра можно проанализировать зоны приема отклика и принять меры по их устранению или сокращению. Иногда можно уменьшить дальность распространения пакетов, перемещая точки доступа во внутренние помещения или на другую сторону здания. Как упоминалось выше, многие устройства позволяют настроить мощность сигнала, чтобы уменьшить излучение из здания.



Флэми Тех советует:

Будьте хорошим беспроводным соседом!

При проведении аудита своей сети вы можете наткнуться на другие точки и узлы беспроводного доступа в ближайшей окрестности или том же здании. Некоторые из них окажутся незащищенными.

Будьте хорошим соседом и дайте знать владельцам, что их точка доступа не защищена. Возможно, они даже не подозревают о грозящих им опасностях.

Будьте хорошим соседом и не пытайтесь проехать по их сети, чтобы продемонстрировать ее незащищенность. Это не просто очень плохое поведение, это подсудное дело, если вас поймают. Поэтому не поддавайтесь искушению и будьте хорошим беспроводным соседом.

Применение внешней антенны

Применение платы, допускающей подключение внешней антенны, резко расширяет зону доступа, и стоят они ненамного больше, чем самые дешевые беспроводные сетевые интерфейсные платы. Потребительские версии, такие как Linksys или D-Link, обычно это не поддерживают, но стоит заплатить дополнительные \$100 за лучшую плату. Если у вас совсем нет средств, посетите web-сайты, на которых рассказано, как смастерить

самодельную антенну для вашей платы. Исходите из предположения, что ваши потенциальные противники также смогут найти эти сайты и сделают антенну никак не хуже вашей.

Проводите аудит при оптимальных условиях

Дождь, сырость, туман способны повлиять на беспроводную передачу. Волны с длиной, соответствующей стандарту 802.11b, резонируют в воде, и это может приглушать сигнал во время ливня или даже просто при повышенной влажности. Листья деревьев из-за высокого содержания воды обладают таким же эффектом. Ваши результаты зимой могут отличаться от летних. Выберите для проверки ясный, сухой день, чтобы оптимизировать свои результаты.

Сохранение сеансов NetStumbler

NetStumbler автоматически начинает сохранять сеанс всякий раз, как вы его открываете. Это позволяет анализировать сеансы NetStumbler позднее. По умолчанию сеансы сохраняются в собственном формате NetStumbler. Можно также сохранять сеансы как текст для импорта в электронную таблицу или текстовый процессор, и в формате wi-scan, который является активно развивающимся файловым стандартом для журналов анализа беспроводных сетей. Можно также экспортировать их в некоторые другие форматы.

Для каждого сеанса NetStumbler выводит сверху окна уникальный номер, являющийся комбинацией даты и времени ([рис. 10.5](#)). Это полезно при отслеживании сеансов и результатов. При желании можно заменить это имя на более содержательное.

Теперь, имея множество данных о периметре беспроводной сети, желательно сгенерировать некоторые отчеты либо для руководства, либо для заказчика, если вы работаете как консультант. Если имеются данные глобального позиционирования, можно построить наглядные карты с помощью программы Microsoft MapPoint и рассмотренного ниже средства с открытыми исходными текстами.

StumbVerter: Программа преобразования карты для NetStumbler

StumbVerter

Автор/основной контакт: Michael Puchol; Sonic Security

Web-сайт: <http://www.sonar-security.com/>

Платформа: Windows

Лицензия: Freeware (аналогична GPL)

Рассмотренная версия: 1.5

Список почтовой рассылки:

Пошлите пустое сообщение по адресу stumbverter-subscribe@c2security.org

StumbVerter - небольшая изящная программа, которая берет выдачу NetStumbler и преобразует ее в исходные данные для программы Microsoft MapPoint. Ее функциональность шире, чем у базовой программы NetStumbler. Расширения включают:

- Отображение точек доступа на карте в виде небольших маяков.
- Изображение маяков различного размера и цвета в зависимости от мощности сигнала точки доступа и режима шифрования.
- Наличие кружков для записи заметок и другой информации.
- Наличие навигационной информации, такой как скорость, направление и расстояние до ближайшей известной точки доступа.
- Средство сравнения антенн.

Для применения StumbVerter необходимо иметь легальную лицензию на Microsoft MapPoint 2002. Я понимаю, что это не соответствует духу свободного программного обеспечения, но расширение функциональности вполне стоит дополнительных \$200, требуемых MapPoint. И, конечно, сама программа StumbVerter условно свободна. Развивается несколько проектов по разработке программ для преобразования файлов NetStumbler во что-нибудь свободное, такое как MapQuest или MapBlast (но ни один из них пока не достиг стадии, позволяющей включить его в публикацию). В любом случае, если необходимо представлять отчеты руководству, цветные карты будут более чем уместны.

Установка StumbVerter

1. Прежде чем устанавливать StumbVerter, убедитесь, что установлены Microsoft MapPoint и NetStumbler. Без двух этих программ установка будет некорректной. Если вы установили их только что, перезагрузите компьютер.
2. Вы должны также задействовать приемник GPS и фиксировать поступающую от него информацию в NetStumbler. Чтобы StumbVerter мог что-то сделать с данными, он должен иметь GPS-координаты беспроводных сетей. С их помощью вычисляется расположение графических элементов.
3. Загрузите StumbVerter с прилагаемого к книге компакт-диска или с web-сайта и распакуйте его.
4. Сделайте двойной щелчок мышью на файле setup, и StumbVerter будет установлен в вашу систему.

Когда все будет установлено, можно начать работать с NetStumbler и StumbVerter.

Применение StumbVerter

1. Чтобы применять StumbVerter, требуются какие-то данные для отображения. Поэтому прогуляйтесь с NetStumbler и соберите информацию о своих беспроводных сетях.
2. Сохраните сеанс в NetStumbler и экспортируйте его в текстовый сводный формат.
3. Запустите StumbVerter, сделав двойной щелчок мышью на его иконке на рабочем столе.
4. В меню сверху экрана щелкните мышью на Map, выберите Create New, а затем выберите свою область.
5. Когда загрузится карта, щелкните мышью на Import и выберите файл .nsi, представляющий сеанс NetStumbler, который вы хотите отобразить. StumbVerter выводит записанные данные в графическом формате в виде карты ([рис. 10.6](#)).

и отобразит результаты рядом друг с другом (рис. 10.7). Это может быть полезно при выборе платы или антенны, особенно если вы мастерите антенны сами.

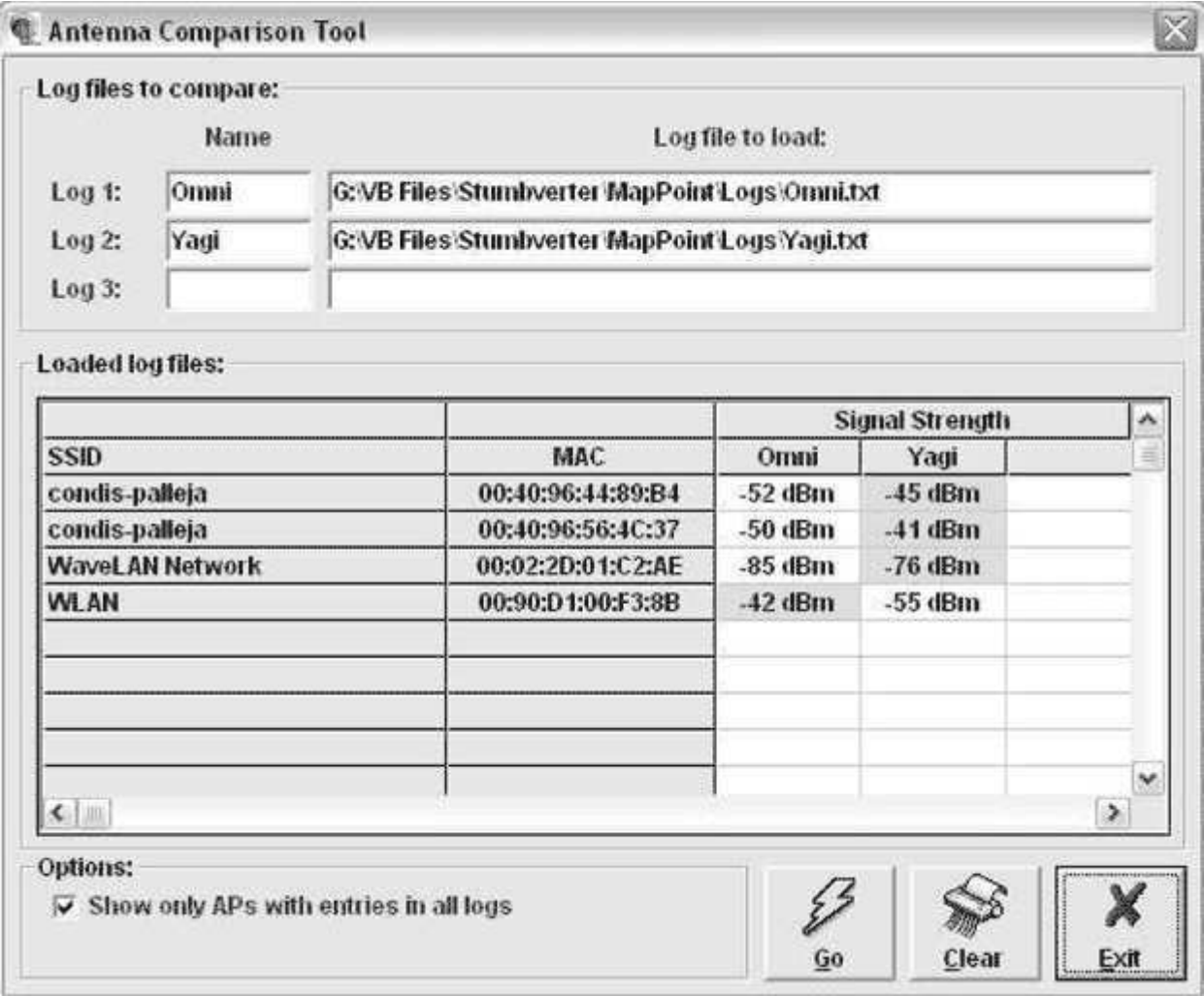


Рис. 10.7. Экран StumbVerter при сравнении антенн

Теперь, ознакомившись с замечательными средствами для Windows, рассмотрим средства для Linux. Хотя средства для Windows легче устанавливать и применять, есть некоторые вещи, которые они пока не делают (-например, пассивное сканирование и попытки взлома WEP).

Kismet Wireless: Программа обнаружения беспроводных сетей для Linux

Kismet Wireless

Автор/основной контакт: Mike Kershaw

Web-сайт: <http://www.kismetwireless.net/>

Платформы: Большинство Linux

Лицензия: GPL

Рассмотренная версия: 4.0.1

Списки почтовой рассылки:

wireless@kismetwireless.net

В основном о применении Kismet, а также предложения, обсуждение, объявление новых возможностей и т.д. Для подписки отправьте электронное сообщение со словом "subscribe" в теле письма по адресу wireless-subscribe@kismetwireless.net.

Имеется также архив обсуждений по адресу

<http://www.kismetwireless.net/archive.php>

wireless-security@kismetwireless.net

Список почтовой рассылки для обсуждения беспроводной безопасности, уязвимостей и других тем, не связанных непосредственно с Kismet. Для подписки отправьте электронное сообщение со словом "subscribe" в теле по адресу wireless-security-subscribe@kismetwireless.net.

Kismet Wireless - один из лучших анализаторов беспроводных сетей для операционной системы Linux. Есть и другие программы, в том числе AeroSniff и Prism2Dump, которые также хорошо работают в Linux. Я выбрал Kismet из-за его растущей базы поддержки и дополнительных модулей, а также совместимости с разнообразным беспроводным оборудованием. Как и Nessus, Kismet Wireless построен в архитектуре "клиент-сервер", что делает его еще более гибким.

Еще одной привлекательной чертой применения платформы Linux является возможность выполнения программ WEPcrack и AirSnort, которые на другие платформы пока не перенесены. На момент публикации не существовало никакого по-настоящему хорошего доступного программного обеспечения с открытыми исходными текстами для тестирования WEP на платформе Windows, но ситуация должна измениться.

Некоторые возможности Kismet выходят за рамки базовой функциональности такой программы, как NetStumbler. Kismet совместим с рядом других программ и может быть настроен для сбора слабых ключей шифрования для попыток взлома внешними программами. Kismet способен работать даже как система обнаружения вторжений, исходящих из вашей беспроводной сети.

Установка сетевой интерфейсной платы и драйверов

Прежде чем загружать Kismet, необходимо удостовериться, что ваша плата совместима с этой программой. В настоящее время Kismet работает со следующими беспроводными платами:

- D-Link;
- Linksys (только PCI и PCMCIA);
- RangeLan;
- Cisco Aeronet;
- ORiNOCO.

Теоретически, Kismet должен работать с любой платой, в которой использованы наборы микросхем Prism II и Hermes, а также с платами, которые можно перевести в режим rf_top или Monitor, но на практике результаты могут быть различными. Я рекомендую выбрать одну из перечисленных выше плат, чтобы избежать проблем.

Теперь начинается самое интересное. Нужно сделать несколько шагов, чтобы превратить Linux-систему в беспроводной анализатор. Для разных аппаратных и программных конфигураций эти действия немного различаются. Проверьте документацию на Web-сайте Kismet, чтобы узнать, нет ли каких-то специфических инструкций для вашего оборудования.

1. Начните с проверки актуальности ваших драйверов PCMCIA (если у вас плата PCMCIA). Если у вас не очень старая версия Linux, то, скорее всего, все будет нормально. В данном примере установки используется Mandrake Linux 9.1.
2. Если драйверы нужно обновить, зайдите на сайт <http://www.rpmfind.com/> и поищите файл pcmcia-cs для вашего дистрибутива. Запустите RPM, и он установит самые свежие драйверы.
3. Удостоверьтесь, что все подходящие для вашей платы беспроводные драйверы загружены. Беспроводные драйверы для Linux поддерживаются не так хорошо, как для Windows, и обычно не имеют удобного графического интерфейса установки. (Будем надеяться, что ситуация изменится, когда производители добавят поддержку для Linux и кто-нибудь создаст RPM для установки драйверов.)

Мне пришлось "прикручивать" собственные драйверы, и удовольствие было ниже среднего. Если возможно, выбирайте одну из поддерживаемых плат; в оперативном доступе имеются подробные инструкции и масса информации о них. Для платы ORiNOCO я скомпилировал драйвер, имевшийся на приложенном к ней диске. Самые свежие драйверы доступны также по адресу <http://www.orinocowireless.com/>, а на некоторых других сайтах предлагаются платы на основе того же набора микросхем.

Если вы используете плату Prism II, то вам потребуются драйверы Linux wlan-ng. Их можно взять на <http://www.linux-wlan.org/>.

4. Установите драйверы и все программные коррекции, необходимые для работы платы в режиме монитора, который требуется беспроводным сетевым анализаторам. Этот режим аналогичен режиму прослушивания для Ethernet, он заставляет плату принимать радиоволны, не ассоциируя их с определенной точкой доступа.

Следующие инструкции предназначены для платы ORiNOCO, которой требовались коррекции для режима монитора. Справьтесь в документации или в Интернете по поводу других плат.

- Загрузите файл или скопируйте его с прилагаемого к книге компакт-диска.
- Чтобы инициировать процесс установки, введите

```
make config
```

Процедура конфигурирования задаст несколько базовых вопросов о вашей системе. Подразумеваемые значения, как правило, годятся.

- Выполните следующие команды от имени пользователя root:

```
./Build  
./Install
```

- Для платы ORiNOCO поверх драйвера требовалось наложить заплату, чтобы обеспечить работу в режиме монитора. Для других плат это может не понадобиться. Заплату можно взять по адресу airsnort.shmoo.com/orinocoinfo.html.
- Если драйвер нуждается в программной коррекции, загрузите корректирующий файл; в противном случае перейдите к шагу 5.
- Распакуйте файл и введите следующую команду:

```
patch -p0 < текущий_корректирующий_файл
```

Должны скорректироваться все файлы, нуждающиеся в обновлении. Если ключ `-p0` не сработает, попробуйте `-p1`.

5. Войдите в файл беспроводной конфигурации и отредактируйте параметры настройки (этот файл находится в `/etc/pcmcia/config.opts`):
 - Если вы собираетесь использовать плату с Kismet, оставьте эти параметры пустыми.
 - Если вы хотите применять ее для доступа к своей локальной базовой станции, введите в этот файл подходящие для сети настройки, такие как SSID и т.д.

6. Теперь можно перезагрузить систему со вставленной в разъем беспроводной платой. Когда это произойдет, должны прозвучать два коротких сигнала, показывающие, что сетевая плата опознана и сконфигурирована.

Если вы не услышите сигналы, вернитесь к документации своей платы и убедитесь, что все шаги сделаны правильно.

7. Наберите `ifconfig` в командной строке. Вы должны увидеть интерфейс `wlan01`. Если его не видно, вернитесь к документации своей платы и убедитесь, что все шаги сделаны правильно.
8. После того, как вам удастся загрузить драйверы, удостоверьтесь, что ваша беспроводная плата действительно работает. Вы должны иметь возможность выхода в Интернет или эхо-тестирования машин в проводной ЛВС. Если вы не сможете этого сделать, придется вернуться к инструкциям по установке платы. Нужно заставить плату работать, прежде чем устанавливать программное обеспечение Kismet.
9. Необходимо также иметь свежую библиотеку `libpcap`, чтобы операционная система могла читать пакеты прямо из вашей платы. Многие описанные ранее в этой книге средства используют этот драйвер, поэтому, если вы его еще не установили, то загрузите его с прилагаемого к книге компакт-диска или с сайта <http://www.tcpdump.org/> и установите.

Наконец, вы добрались до финиша установки сетевой интерфейсной платы и драйверов, необходимых для работы Kismet.

Установка Kismet

Если все прошло успешно, можно перейти к установке программы.

1. Загрузите Kismet с прилагаемого к книге компакт-диска или с web-сайта.
2. Распакуйте дистрибутив.
3. Для компиляции Kismet наберите следующую команду с любыми подходящими конфигурационными ключами, перечисленными в [табл. 10.4](#):

```
./configure
```

Эти ключи времени компиляции можно задавать в инструкции `configure` для включения или отключения некоторых функций.

Таблица 10.4. Конфигурационные ключи Kismet

| Ключ | Описание |
|-----------------------|---|
| --disable-curses | Отключает пользовательский интерфейс на основе curses |
| --disable-panel | Отключает расширения панели ncurses |
| --disable-gps | Отключает поддержку GPS |
| --disable-netlink | Отключает перехват сокетов Linux NetLink (с заплатами для prism2/orinoco) |
| --disable-wireless | Отключает беспроводные расширения ядра Linux |
| --disable-pcap | Отключает поддержку перехвата посредством libpcap |
| --enable-syspcap | Использует системную библиотеку libpcap (не рекомендуется) |
| --disable-setuid | Отключает возможность переустановки действующего идентификатора пользователя (не рекомендуется) |
| --enable-wsp100 | Включает устройство перехвата - удаленный сенсор WSP100 |
| --enable-zaurus | Включает некоторые дополнительные возможности (такие как пьезозуммер) для КПК Zaurus. |
| --enable-local-dumper | Заставляет использовать локальные средства дампа, даже если присутствует Ethereal. |
| --with-ethereal=DIR | Поддерживает прослушивание Ethereal для протоколирования |
| --without-ethereal | Отключает поддержку прослушивания Ethereal |
| --enable-acpi | Включает поддержку продвинутого интерфейса конфигурирования и питания ядром Linux |

4. Когда процесс конфигурирования будет закончен, выполните следующие команды от имени пользователя root, чтобы закончить процесс компиляции и установить программу:

```
make dep
make
make install
```

5. Завершив установку программы Kismet, найдите файл `kismet.conf`, который по умолчанию должен располагаться в `/usr/local/etc`. В этом файле задаются ваши интерфейсные и протокольные предпочтения. В [табл. 10.5](#) описаны варьируемые параметры.

Таблица 10.5. Интерфейсные и протокольные опции Kismet

| Параметр | Описание |
|-----------------------|--|
| Capture source | Определяет, какие интерфейсы будет прослушивать Kismet. Обычно здесь уже должен быть задан основной беспроводной интерфейс (<code>wlan0</code>). Если вы хотите добавить дополнительные интерфейсы, сделайте это в формате <code>source=тип,интерфейс,имя</code> . Например, настройка <code>source=prism2,wlan0,Prism</code> предпишет Kismet слушать <code>wlan0</code> как плату типа <code>prism2</code> . В журналах соответствующие данные будут фигурировать под именем Prism |
| Fuzzy encryption | Отображает все идентифицированные пакеты как нешифрованные для станций, применяющих неопределенные или собственные методы шифрования. Обычно эту опцию оставляют выключенной, если только плата не считает заведомо шифрующие сети нешифрующими |
| Filtering packet logs | Ограничивает круг протоколируемых пакетов. Воспользуйтесь опцией <code>noiselog</code> , чтобы отбрасывать все пакеты, которые кажутся испорченными или фрагментированными из-за шума. В насыщенной области с множеством помех или при использовании платы без внешней антенны это может уменьшить размер журнала. Опция <code>beaconlog</code> отбрасывает все пакеты определенной точки доступа, кроме первого пакета радиомаяка. Настройка <code>phylog</code> отбрасывает все пакеты физического уровня, которые иногда подхватываются. Допустима любая комбинация этих настроек |
| Decrypt WEP keys | Расшифровывает перехваченные пакеты данных на лету. Для этого, однако, следует иметь ключ, который иногда можно добыть с помощью программы AirSnort (описанной далее в этой лекции). Для каждой точки доступа требуется отдельная инструкция вида <code>bssid:key</code> где <code>bssid</code> - это MAC-адрес точки доступа, а <code>key</code> - ключ для нее |
| Using an external IDS | Посылает пакеты внешней системе обнаружения вторжений для дальнейшего анализа. В этой инструкции задается именованный канал, а сетевой системе обнаружения вторжений следует предписать чтение из него. |

6. Теперь отредактируйте файл `kismet_ui.conf`, также находящийся в `/usr/local/etc`. В нем задаются некоторые настройки интерфейса. В [табл. 10.6](#) перечислены возможные варианты.
7. Сохраните оба файла.

Теперь все готово к применению Kismet для аудита беспроводной сети.

Таблица 10.6. Настройки интерфейса Kismet

| Настройка | Описание |
|-----------|---|
| Columns | Определяет, какие столбцы и в каком порядке появятся в интерфейсе Kismet. Измените значение <code>columns</code> или <code>clientcolumns</code> в соответствии с тем, что вы хотите видеть. Полный список столбцов имеется в оперативной справке Kismet |
| Colors | Определяет цвета элементов изображения. Измените значение <code>colorxxx</code> на требуемый код цвета. Придется немного поэкспериментировать с этой настройкой, чтобы правильно подобрать цвета. (Я нашел, что подразумеваемые значения приемлемы, но не для печати, и заменил их цветами, которые лучше смотрятся на бумаге.) |

Применение Kismet Wireless

Запустите Kismet, набрав имя исполнимого файла в командной строке или на терминале X-Window, поддерживающем инструментарий Curses. Отобразится основной интерфейс (см. [рис. 10.8](#)). Kismet немедленно начнет сообщать обо всех беспроводных сетях в вашей округе и выдавать информацию о них.

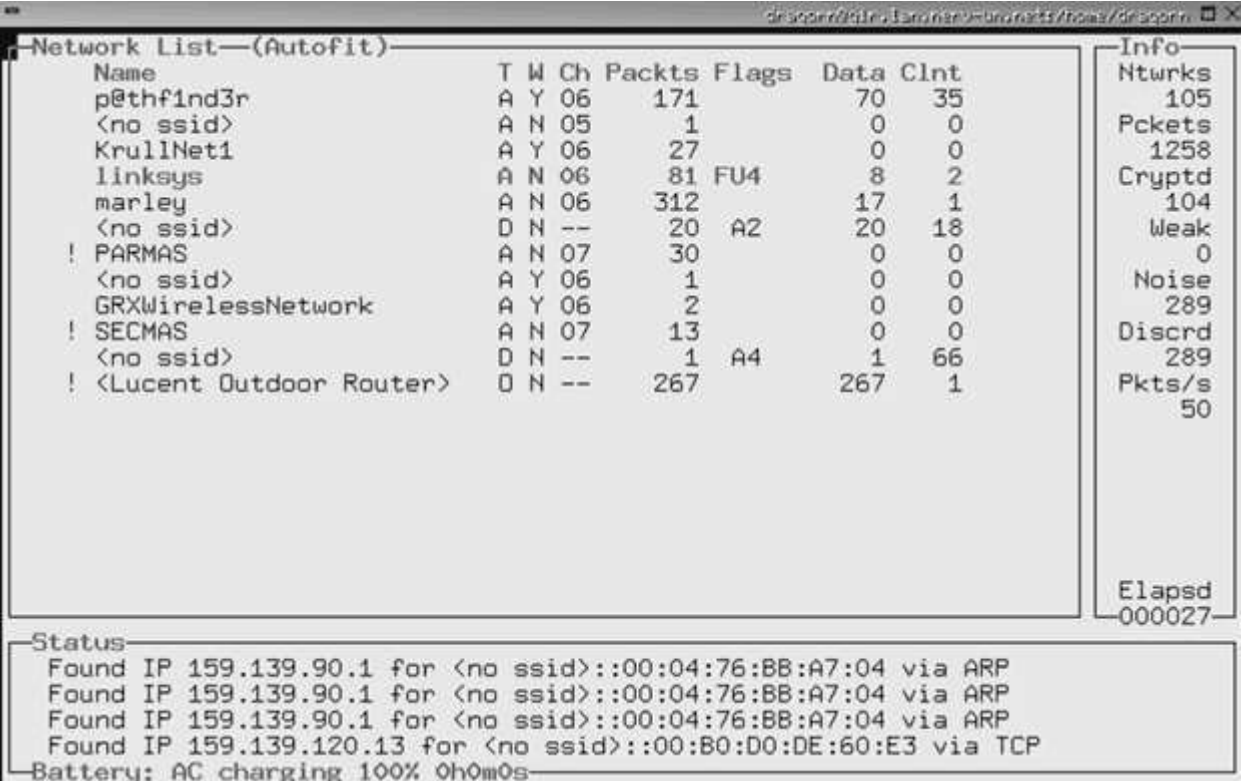


Рис. 10.8. Основной экран Kismet Wireless

В интерфейсе можно выделить три основные части. Раздел Network List слева отображает все активные в текущий момент беспроводные сети, которые Kismet смог увидеть, и основную информацию о них: SSID сети (если доступен), тип (точка доступа или узел), шифруется она или нет с помощью WEP, используемый канал вещания, число перехваченных до сих пор пакетов, любые флаги на данных и объем данных, проходящих через сеть. Вывод кодируется цветом: активные сети отображаются красным цветом, а неактивные - черным.

В поле Info справа отображается общая статистика текущего сеанса перехвата, включая общее число обнаруженных сетей, общее число пакетов, число пакетов, которые были зашифрованы, услышанные слабые сети, пакеты с высоким уровнем шума, отброшенные пакеты и среднее число пакетов в секунду.

Поле Status внизу содержит прокручивающееся представление происходящих событий. Сообщения всплывают, когда появляются новые сети или происходят другие события.

Так как Kismet - средство командной строки, хотя и с графическим интерфейсом, для управления его функциями применяются клавишные команды. В [табл. 10.7](#) перечислены клавишные команды, доступные из основного экрана.

Таблица 10.7. Клавишные команды Kismet

| Клавишная команда | Описание |
|-------------------|--|
| a | Выдает статистику числа пакетов и распределения каналов |
| c | Открывает клиентское всплывающее окно для отображения клиентов выбранной сети |
| d | Предписывает серверу начать извлечение из потока пакетов цепочек печатных символов и их отображение |
| e | Открывает всплывающее окно на серверах Kismet. Это позволяет одновременно контролировать два или несколько серверов Kismet на различных хостах (напомним, что это архитектура клиент-сервер) |
| f | Находит центр сети и отображает компас |
| g | Группирует помеченные в данный момент сети |
| h | Выдает список возможных команд |
| i | Выдает подробную информацию о текущей сети или группе |
| l | Показывает уровни сигнал/мощность/шум, если плата их сообщает |
| m | Отключает звук и речь, если они включены (или включает их, если они были перед этим выключены). Чтобы этим пользоваться, в конфигурации должны быть включены звук или речь |
| n | Переименовывает выбранную сеть или группу |
| p | Выдает типы пакетов по мере их получения |
| r | Выводит столбчатую диаграмму темпа порождения пакетов |
| s | Изменяет способ сортировки списка сетей |
| t | Помечает текущую сеть или снимает метку с нее |
| u | Исключает текущую сеть из группы |
| w | Выдает все предыдущие сигналы и предупреждения |
| z | Увеличивает панель вывода сети на весь экран (или возвращает ей нормальный размер, если она уже увеличена) |

Как отмечено выше, можно расширить представление информации о каждой обнаруженной сети, чтобы показать все детали определенной точки доступа, вводя ⁱ в командной строке. На [рис. 10.9](#) воспроизведена эта выдача.

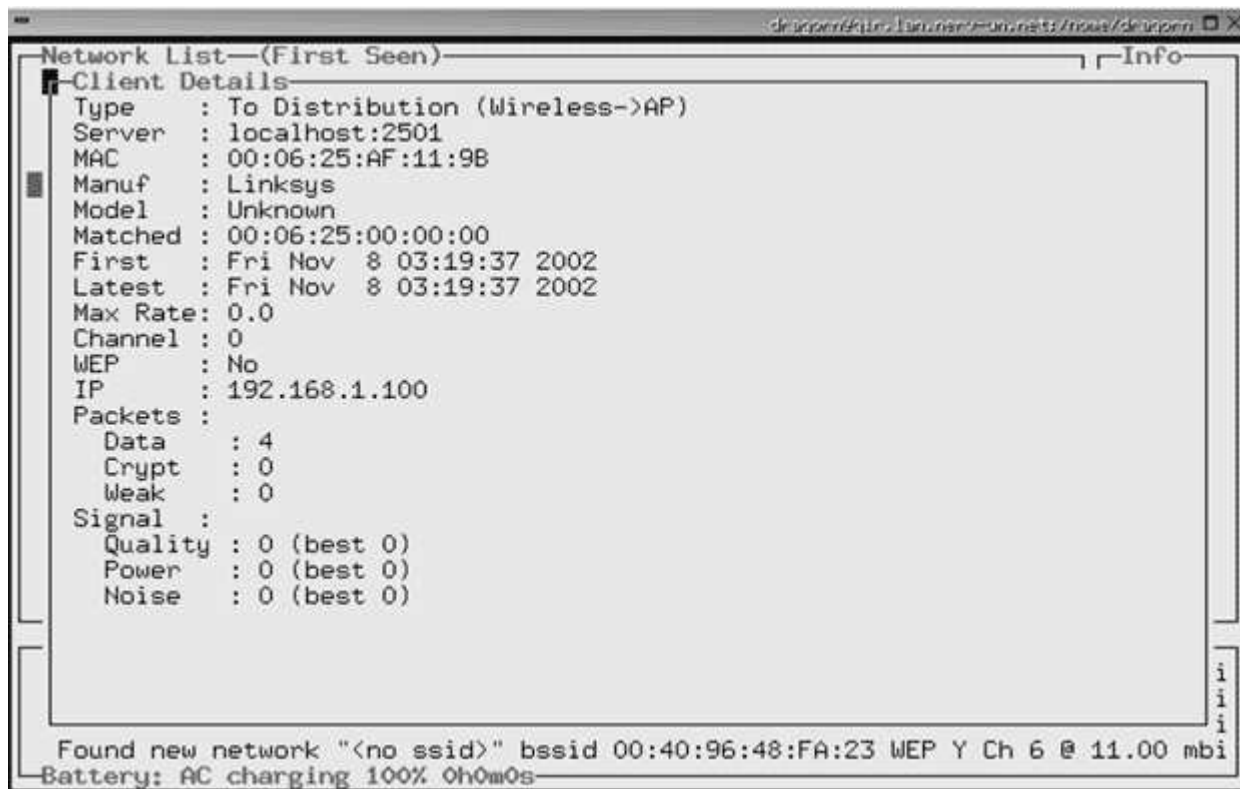


Рис. 10.9. Экран Kismet с подробными данными о сети

С помощью команды `z` можно расширить поле сети на весь экран и видеть дополнительную информацию о каждой сети, например производителя обнаруженного оборудования. Это облегчает группирование точек доступа, если вы пытаетесь следить лишь за определенной частью из них и хотите иметь возможность отфильтровывать другие. Это делается с помощью команд `g` и `u`, служащих для включения и удаления из группы соответственно.

Удобно работать с поддержкой звука - звуковой сигнал подается при обнаружении новых сетей. Звук можно отключить с помощью команды `m`, если вы то и дело входите и выходите из области приема множества сетей, иначе вы получите какофонию сигналов!

Поддержка GPS в Kismet

Kismet способен записывать данные GPS, если имеется приемник GPS, подключенный к машине. Для его чтения требуется программное обеспечение демона GPS для Kismet, GPSD. GPSD можно взять по адресу <http://russnelson.com/gpsd/>. Необходимо включить поддержку GPS при компиляции Kismet с помощью параметров времени компиляции, приведенных выше в [табл. 10.4](#). После этого Kismet будет автоматически добывать координаты услышанных сетей и протоколировать их.

Можно сделать еще один шаг и отобразить эти координаты на карте, так же как для программы в Windows. Kismet поставляется со встроенной программой GPSMAP, которая автоматически изображает собранные данные на картах в формате .gps. Для этого, правда, требуется предоставить собственную откалиброванную GPS-карту. Для построения карт в Linux имеется программа с открытыми исходными текстами GPSDrive. Ее можно загрузить со страницы <http://gpsdrive.kraftvoll.at/index.shtml>.

Kismet как система обнаружения вторжений

Kismet можно настроить как беспроводную систему обнаружения вторжений, перехватывающую входящие сигналы и обнаруживающую беспроводной трафик, ассоциированный с агрессивным объездом или иной подозрительной беспроводной активностью. Kismet обнаруживает около 10 различных видов

трафика, включая опросы NetStumbler, а также активность Airjack и других беспроводных хакерских средств. В настоящее время эти возможности Kismet довольно ограничены, но можно ожидать их развития в будущем. И поскольку исходные тексты открыты, всегда можно расширить функциональность самостоятельно, запрограммировав собственные сигналы тревоги. Еще одна возможность - передать по каналу данные Kismet традиционным системам обнаружения вторжений, таким как Snort, для более детального анализа. Функции обнаружения вторжений задаются в файле kismet.conf и по умолчанию отключены. Kismet можно настроить и для сбора известных криптографически слабых ключей для такой программы, как AirSnort, - следующего средства, представленного в этой лекции, которое анализирует беспроводные пакеты и пытается взломать шифрование WEP.

AirSnort: Программа восстановления ключей шифрования WEP

AirSnort

Исходные авторы/основной контакт: Jeremy Bruestle и Blake Hegerle

Web-сайт: <http://schmoo.airsnort.org/>

Платформы: Большинство Linux

Лицензия: GPL

Рассмотренная версия: 2.4.22

Авторы разработали AirSnort как практическое приложение для демонстрации слабых мест в WEP - протоколе шифрования для беспроводных сетей. В статье, озаглавленной "Слабые места алгоритма генерации ключей RC4", написанной специалистами по криптографии Флюхером, Мартином и Шамиром, детализированы теоретические слабости алгоритма WEP и показано, что некоторые векторы инициализации будут слабыми. Пакеты, зашифрованные с помощью слабых векторов инициализации, можно собрать и со временем накопится достаточно данных для экстраполяции разделяемого секретного ключа. Это позволяет легко расшифровывать пакеты. Вскоре после опубликования статьи были выпущены два средства, AirSnort и WEPCrack, эксплуатирующие описанные слабости для восстановления ключей WEP, фактически - взламывающие WEP. Оба средства хороши, но AirSnort обладает некоторой дополнительной функциональностью как беспроводной сетевой анализатор. AirSnort сейчас - проект с открытыми исходными текстами, базирующийся по адресу SourceForge.net, с момента своего появления он существенно расширен и улучшен. Поскольку на платформе Windows подобных средств нет, для тестирования WEP в настоящее время имеется лишь два жизнеспособных варианта - AirSnort и WEPCrack.

Применение AirSnort

Зачем применять AirSnort в собственной беспроводной сети? Может создаться впечатление, что у этой программы нет законного применения, а ее единственное назначение - служить инструментом взлома. Однако я считаю, что единственный способ узнать, каким опасностям подвержена ваша беспроводная сеть, - делать то, что делают хакеры, чтобы проверить, можно ли взломать ваше шифрование, и сколько для этого потребуется времени. AirSnort позволяет сделать именно это.

Пытаясь взломать беспроводное шифрование, можно уяснить, насколько это реально. При использовании стандартного WEP - это просто вопрос времени. Математически установлено, что в некоторой точке его можно взломать с помощью данного средства. Вопрос только в том, сколько на это потребуется времени. Если нужно много времени, то можно обоснованно считать, что вы достаточно защищены. Если трафик в вашей беспроводной ЛВС небольшой, то на взлом может уйти несколько дней или даже недель. Это делает вашу сеть практически неинтересной для большинства случайных хакеров. Однако если сеть используется интенсивно, то кто-нибудь сможет собрать достаточное количество пакетов, чтобы взломать ее через несколько часов или за день.

Знание этого поможет вам лучше обезопасить свою сеть, обосновать необходимость внедрения дополнительных средств защиты, таких как усиление физического контроля или ограничение трафика, а также обновления беспроводного оборудования. Устройство Cisco Aironet использует разновидность WEP, называемую LEAP, для улучшения и исправления слабостей исходного протокола WEP. Беспроводная сеть, основанная на этом протоколе, должна быть невзламываемой, по крайней мере с помощью легко доступных средств. Вы можете определить, что уровень вашего трафика делает непрактичным взлом шифрования. В любом случае, знание сделает ваш сон более спокойным.

Установка AirSnort

Приведение в рабочее состояние драйверов и программного обеспечения для AirSnort может быть весьма трудоемким. Требования AirSnort по сути те же, что и у Kismet. Вернитесь к разделу "Установка сетевой интерфейсной платы и драйверов" и следуйте описанной там процедуре. Когда все будет сделано, можно устанавливать AirSnort. Это - легкая половина дела.

1. Загрузите файл программы с прилагаемого к книге компакт-диска или официального Web-сайта и распакуйте его.
2. Перейдите в каталог, в который вы распаковали файл, и выполните процедуру

```
./autogen.sh
```

3. Станьте пользователем root и запустите

```
make
```

Программа будет собрана автоматически. Если не возникнет ошибок, значит, вы успешно установили AirSnort.

Запуск AirSnort

AirSnort включает три основных исполнимых файла:

- airsnort выполняет работу по сбору пакетов из некоторого источника, обычно беспроводной сетевой платы.
- gencases разбирает перехваченные данные для выявления слабых ключей.
- decrypt выполняет попытки автономного расшифрования файлов, загруженных из другого источника.

AirSnort воспринимает файлы других анализаторов беспроводных сетей, если они сохраняются в формате pcap. Kismet, наше рекомендуемое беспроводное средство для Linux, будет заранее специально вылавливать интересные для AirSnort пакеты, избавляя от этого шага.

Не обязательно собирать всю совокупность данных за один раз. AirSnort дает возможность сохранить сеанс, открыть его позже и дописать. Это делает AirSnort особенно опасным для беспроводных сетей, так как для сбора достаточного для взлома сети количества пакетов не требуется проводить весь сеанс без перерыва вблизи вашего здания. Эту деятельность можно разделить на небольшие, менее заметные интервалы, предполагая, что ключи в целевой сети меняются не очень часто.

После установки программы AirSnort можно запустить ее, набрав в командной строке airsnort. Интерфейс - сама простота: один экран, на котором отображаются интересные пакеты и общее число шифрованных и нешифрованных пакетов. В верхней части показаны такие настройки, как тип сетевой платы и т.д. Слева можно изменить некоторые настройки, такие как размах - число пробных угадываний, которое будет делать AirSnort для каждого байта ключа при попытках расшифрования для ключей в 40 или в 128 бит. По умолчанию используется 3 для 40-битного шифрования и 2 для 128-битного. Если у вас недостаточно данных или избыток вычислительной мощности, можно попробовать немного увеличить это значение, но не делайте его больше 4 или 5.

Затем можно откинуться на спинку кресла и собирать пакеты. Не ждите, что сможете взломать ключи WEP за несколько минут. Чтобы AirSnort сработал успешно, требуется примерно от 1500 до 4500 пакетов со слабыми ключами. Это соответствует примерно от 100 до 500 МБ данных. Чтобы собрать столько данных в сети с умеренной нагрузкой, может потребоваться день или больше. В менее загруженной сети на это уйдет значительно больше времени, а в более загруженной - существенно меньше. В любом случае потребуется не менее двух часов, а, возможно, и больше. Конечно, многое зависит и от удачи, поэтому ваши результаты могут варьироваться от часа до бесконечности. Как правило, на сбор данных стоит затратить примерно столько времени, сколько, по вашему мнению, затратит его средний посторонний хакер, желающий остаться незамеченным. И, конечно, возможность AirSnort возобновлять сеансы позволяет значительно сократить временное окно, так как хакеры могут собирать данные в несколько приемов.

После успешного взлома ключ WEP отображается слева на экране как в текстовом, так и в исходном шестнадцатеричном виде, и сеанс перехвата завершается. Счастливого WEP-взлома!

Что делать, если вам удастся вычислить свои ключи WEP? Не паникуйте, потому что случайные хакеры в большинстве своем не создадут вам проблем. Однако необходимо подумать об усилении защиты своей беспроводной сети, чтобы затруднить посторонним сбор этих данных. Можно принять ряд мер,

начиная от замены оборудования до реконфигурирования и изменения расположения вашей точки доступа. Исходя из критичности данных в сети следует выбрать адекватные меры.

Меры по повышению безопасности беспроводной ЛВС

Весьма вероятно, что со временем вам придется реализовать беспроводную технологию. Даже если вы не собираетесь этого делать, все равно необходимо периодически проверять сеть, чтобы убедиться, что никто не завел зловредную точку беспроводного доступа. Хотя применение любого беспроводного доступа связано с риском, можно уменьшить свою незащищенность, принимая следующие предупредительные меры.

Включите WEP

Шифруя свои данные, вы заставите хакеров затратить существенно больше времени и усилий, чтобы добраться до ваших беспроводных данных и сети. Это отведит случайных хакеров и заставит злоумышленников провести в вашем районе день или больше, увеличивая вероятность того, что они будут замечены персоналом службы безопасности или бдительными служащими.

Применяйте беспроводное оборудование с улучшенным протоколом шифрования

Как упоминалось выше, в оборудовании Cisco применяется улучшенная версия протокола WEP, называемая LEAP, которая показала себя невосприимчивой к попыткам взлома. Имеется также новый стандарт 802.11i, исправляющий проблемы WEP. К сожалению, 802.11i был одобрен как стандарт совсем недавно, и оборудование на его основе только начало появляться. Если вы можете его приобрести, то сделайте это. Цены не должны существенно отличаться от цен более старых устройств в стандартах 802.11a или 802.11b.

Требуйте, чтобы беспроводные пользователи входили через туннель виртуальных защищенных сетей

Обычно подобный туннель становится непреодолимым препятствием для возможных беспроводных взломщиков. Даже если им удастся взломать шифрование WEP, им придется побороться с шифрованием виртуальных защищенных сетей. Некоторые производители (такие как SonicWALL с Wi-FiSec) добавили такую возможность в свое оборудование. Недостатком является то, что возникает дополнительный уровень сложности для ваших пользователей, затрудняется поддержка гостевых пользователей, так как для доступа к беспроводной ЛВС им придется загружать программное обеспечение клиента виртуальной защищенной сети, а также ключ WEP.

Считайте свою беспроводную сеть недоверенной

Так как вы не можете контролировать трафик, приходящий по воздуху в точки доступа, вы должны относиться к нему так же, как к общедоступной стороне межсетевого экрана. Если позволяют средства, поместите межсетевой экран между беспроводной сетью и ЛВС (некоторые варианты с открытыми исходными текстами см. в [лекции 3](#)) или разместите ее в своей демилитаризованной зоне. Тогда у вас будет возможность отфильтровать определенные виды атакующих пакетов, ограничить некоторые виды трафика и отслеживать любую активность на этом интерфейсе.

Регулярно проверяйте свой беспроводной периметр

Это особенно важно, если вы находитесь в одной из вышеупомянутых перегруженных областей. Проверьте, насколько далеко ловится ваш сигнал и перекрывается ли ваша сеть с соседними. Даже если вы официально не разрешаете беспроводной доступ, необходимо делать это периодически, чтобы обнаружить любые неконтролируемые или "неофициальные" точки доступа. Беспроводной доступ стал настолько дешевым и простым в организации, что бездумные или безответственные менеджеры нередко просто идут в местный магазин электроники и устанавливают точку доступа для некоторой временной цели, например демонстрации в не оборудованном сетью конференц-зале, подставляя вашу сеть под беспроводную атаку. Кроме того, помните, что множество новых ПК, особенно ПК-блокнотов, поставляются со встроенными платами Wi-Fi, и их включение не составляет особого труда. Беспроводной доступ в сети может использоваться без вашего ведома. Беспроводной аудит - единственный способ прояснить ситуацию.

Переместите точки доступа

Иногда простым перемещением базовой станции во внутреннее помещение можно существенно сузить зону распространения сигнала беспроводной сети. Используйте результаты беспроводного аудита для выявления проблемных точек доступа. Поэкспериментируйте с размещением, чтобы добиться

оптимального приема внутри здания, но минимизировать прием снаружи. Например, если перед вашим зданием располагается большая автостоянка, а сзади - заросший деревьями участок, то перемещение базовой станции к задней стене здания сохранит, вероятно, ее доступность для большинства внутренних пользователей, но ограничит распространение сигнала областью, которая не так легко доступна для агрессивных ездоков.

Должным образом сконфигурируйте беспроводную сеть

Имеется много возможностей и настроек, которые позволяют существенно повысить безопасность. Не всякое оборудование поддерживает эти возможности, но вот что, тем не менее, можно сделать:

- Отключите широковещание SSID. В этом случае пользователь должен знать идентификатор набора сервисов, чтобы открыть сеанс с базовой станцией. Это действует как слабый пароль. Однако, если злоумышленник сможет взломать ваше шифрование, он сможет легко получить SSID.
- Ограничьте доступ по MAC-адресам. Это затруднит получение доступа к вашей сети через беспроводную базовую станцию. На большинстве базовых станций можно ограничить доступ для определенных аппаратных MAC-адресов. Это довольно сильный метод аутентификации, так как только пользователи с сетевыми картами подходящих серий смогут получить доступ. Однако администрирование авторизованных плат может быть обременительным, да и новым пользователям, пришедшим в офис, доступ будет разрешен не сразу. Кроме того, если атакующий узнает один из авторизованных MAC-адресов, он сможет подделать его на своей плате и замаскироваться под легального пользователя.

Обучите свой персонал

Как и вообще в компьютерной безопасности, человеческий фактор может быть самым слабым или самым сильным звеном. Убедитесь, что охрана, служащие в приемной и другой персонал знают, как определять подозрительное поведение, ассоциированное с агрессивным объездом. Например, если они заметят кого-то, длительное время сидящего в машине на вашей стоянке, возможно, со странной антенной на крыше, то, весьма вероятно, что он нацелен на вашу беспроводную сеть.

Разработайте также политику и получите санкцию на уровне компании на развертывание беспроводных ЛВС. Доведите до сведения менеджеров, что им нельзя самостоятельно устанавливать беспроводные ЛВС; им следует связаться с вами, чтобы получить официальное подключение. Они должны понимать, что таким поведением подвергают риску всю организацию. Зачастую демонстрация является лучшим способом показать опасность неофициального беспроводного доступа. Информированные сотрудники - лучшая защита.

Инструменты безопасности с открытым исходным кодом

11. Лекция: Судебные средства: версия для печати и PDA

Все средства и методы, ранее описанные в этой книге, при правильной реализации и бдительной поддержке сделают вашу сеть весьма безопасной. Но даже если все сделать верно, нельзя гарантировать абсолютную безопасность сети. Если атакующий достаточно настойчив или удачлив, он иногда сможет проникнуть внутрь. Внешние злоумышленники способны эксплуатировать еще не опубликованные уязвимости или поймать вас в окне между объявлением уязвимости и наложением корректирующей заплатки. Коварный сотрудник может применить для проникновения физические средства, такие как физический доступ к серверу или кража пароля. Могут использоваться и средства морально-психологического воздействия, чтобы с помощью излишне предупредительного сотрудника обойти все ваши меры безопасности и получить несанкционированный доступ. Что же делать, если, несмотря на все ваши приготовления, сеть или система оказались скомпрометированы?

При условии, что вас не выгнали с работы, это еще не конец света. Взломам подвергаются даже информационные системы самых крупных в мире компаний с огромным персоналом компьютерной безопасности, поэтому в этом нет ничего постыдного. Однако, теперь ваша задача - решить головоломку, определить, как все произошло, заделать дыры в безопасности и, если необходимо, выследить злоумышленников и принять дополнительные меры. В этом может помочь ряд средств с открытыми исходными текстами. Они называются судебными средствами, так как вы пытаетесь определить, что произошло, на основе доступных вам свидетельств.

Обзор лекции

Изучаемые концепции:

- Применение судебных средств
- Концепции реагирования на инциденты
- Подготовка к судебному расследованию
- Догматы надлежащего судебного расследования

Используемые инструменты:

Fport, Isof, dd, файлы журналов UNIX и Windows, Sleuth Kit, Autopsy Forensic Browser и The Forensic Toolkit

Применение компьютерных судебных средств

После атаки на систему вы захотите определить, как все происходило, чтобы предотвратить подобное в будущем. Если хакеры смогли обойти существующие электронные средства защиты, то, очевидно, где-то в броне имеется дыра. Сразу может быть неочевидно, где она находится, особенно, если злоумышленники хорошо замели следы. Судебные средства помогают обнаружить эти цифровые следы и найти дыры, чтобы их можно было залатать.

Очистка и восстановление

Если атакующие нанесли повреждения, следует точно определить, что они сделали, узнать, насколько обширны повреждения, и произвести необходимые восстановительные работы. Естественно, не в ваших интересах оставлять в сети взломанные хакерами серверы или созданные ими счета для тайного входа. Судебные средства помогают все это определить и, если атакующий удалил файлы, восстановить некоторые из них.

Уголовное расследование

Если ущерб, нанесенный атакующим, достаточно серьезен, может возникнуть желание начать его уголовное преследование. Простое искажение Web-страницы или вторжение обычно не стоят преследования из-за высоких издержек. Однако, если существенно пострадала ваша инфраструктура или корпоративная репутация, возможно, имеет смысл выдвинуть уголовное обвинение против атакующего. Ваша страховая компания может потребовать, чтобы вы представили полицейский отчет, чтобы вчинить иск. Судебные средства помогут идентифицировать атакующих, так что вы сможете представить отчет и доказательства для их судебного преследования.

Есть несколько вопросов, которые необходимо рассмотреть, прежде чем ступить на этот путь. При незначительном ущербе вы можете подать заявление в местное отделение полиции. Помните, что на местном уровне у них зачастую нет ресурсов для надлежащего расследования компьютерных преступлений, и вам, возможно, придется проводить большую часть расследования самостоятельно. Для этого можно применять средства из данной лекции. Только будьте осторожны, чтобы не испортить улики, иначе в суде они окажутся бесполезными (см. врезку о компьютерном расследовании).

Если ущерб велик или злоумышленные действия попадают в разряд федеральных преступлений (связанных, например, с межштатной или международной торговлей), можно передать дело в ФБР. Контактную информацию местного отделения ФБР можно найти в телефонном справочнике или в Web на сайте <http://www.fbi.gov/>. Если нарушены федеральные законы или материальные потери превысили \$25000, ФБР, скорее всего, займется вашим делом. В противном случае вас могут переадресовать в местные правоохранительные органы. Если вы сможете показать некую связь с терроризмом, межштатным мошенничеством (таким как кража номеров кредитных карт или маскарад), или некоторые другие элементы, которым ФБР уделяет особое внимание, вашим делом могут заняться и при меньшем ущербе. Большинство атак едва ли будет серьезно расследоваться; каждый день сообщается о слишком большом числе инцидентов, поэтому в ФБР реально уделяют внимание только по-настоящему серьезным случаям.

Если вы сумели добиться успеха и на злоумышленников заведено уголовное дело, то правильно проведенное расследование становится еще более важным. Применительно к компьютерным преступлениям очень трудно что-либо доказать. В суде весьма сложно обосновать связь между некими действиями, выполненными от имени пользователя с определенным идентификатором, и конкретным человеком. Обычно обвинители должны доказать, что человек действительно находился за клавиатурой и использовал этот системный счет, когда имела место атака. В противном случае найдется масса отговорок, таких как "Кто-то использовал мой пароль", "Меня взломали" и т.д. Повышенное внимание уделяется также режиму сохранения собранных свидетельств, то есть сведений о том, кто имел доступ к данным и мог их изменить или подменить. В подобных случаях обратитесь в правоохранительные органы, которые могут применить собственные средства сбора данных. Можно также воспользоваться услугами независимых организаций, способных оказать профессиональную помощь при взаимодействии с правоохранительными органами.



Флэми Тех советует:

Недостаток знаний опасен!

Если вы думаете о предъявлении уголовных обвинений, то не следует немедленно применять средства из этой книги. Кроме деятельности по блокированию и восстановлению, вы никоим образом не должны искажать свидетельства. Неумелый человек с помощью этих средств может стереть доказательства или сделать их бесполезными в суде. Представьте себе сыщика-новичка, бродящего на месте убийства. Никуда не годится! Пусть этим занимаются профессионалы из правоохранительных органов, а вы сможете им помочь, если понадобится, воспользовавшись инструментарием и знаниями из этой лекции.

Карьера в судебной информатике

Рост компьютерной преступности создал многообещающую область - судебную информатику с отличными перспективами карьерного роста для тех, кто ей интересуется. Потребность в компьютерно грамотных копах никогда не была столь острой. Если вас привлекает подобная деятельность, есть несколько направлений, по которым можно выдвинуться.

Местные правоохранительные органы

В полицейских управлениях крупных городов обычно имеются отделы компьютерных преступлений. Чтобы туда устроиться, может потребоваться диплом, где в большей или меньшей степени фигурирует юриспруденция или нечто аналогичное. Однако порой в полиции возникает столь острая нужда в технических специалистах, что они готовы смягчить требования к опыту работы в полиции в обмен на технические знания.

Федеральные правоохранительные органы

Наиболее перспективны должности в области судебной информатики в ФБР. Здесь вам придется работать с особо важными делами национального или международного уровня. Обычно ФБР продвигает сотрудников из собственных рядов, однако иногда делаются исключения для людей с определенным талантом или положением. Работая в ФБР, вы сможете реально влиять на компьютерную преступность.

Вооруженные силы

Если у вас склонность к военной службе, то во всех видах и родах вооруженных сил имеется персонал для борьбы с компьютерной преступностью. В этом ряду выделяется Отдел специальных расследований Военно-Воздушных Сил США. Хотя этот отдел ориентирован на преступления и инциденты в вооруженных силах, его часто привлекают и к гражданским делам, поскольку разные компьютерные преступления могут быть взаимосвязаны.

Министерство национальной безопасности

Имеется множество новых вакансий и отделов, созданных как часть Министерства национальной безопасности. Работа в правоохранительных органах или вооруженных силах зачастую оплачивается хуже, чем аналогичная деятельность в коммерческой организации, однако многие считают эти должности более престижными. Есть также крупные организации, имеющие собственный персонал для компьютерных расследований. Государственная служба может существенно повысить ваш статус, если вы захотите затем перейти к частной практике или попасть в отдел судебной информатики крупной организации.

Гражданский иск

Если вы решите, что уголовное преследование не оправдано, вы можете возбудить против хакера гражданское дело. Иногда это единственный способ заставить злоумышленника прекратить атаки. Если нападение исходит из другой организации, санкционированно, как в случае промышленного шпионажа, либо несанкционированно, как в случае неуправляемого сотрудника, обстоятельства могут вынудить вас подать гражданский иск и собрать достаточно доказательств. Хотя в случае гражданских дел требования к доказательствам не столь строги, вы все равно обязаны обосновать свои претензии. Средства из данной лекции помогут вам сделать это. Однако в случае серьезных проблем все равно лучше нанять специалиста по судебной информатике, чем пытаться сделать все самому.

Внутренние расследования

Если вы подозреваете, что источник вторжения - внутренний, его следует обязательно проследить, поскольку он крайне опасен для производственной деятельности. Внутренний хакер может нанести значительно больший ущерб, чем внешний, поскольку зачастую он знает персонал и системы, ему известна информация, раскрытие или компрометация которой может принести максимальный вред организации. Применяя судебные средства, вы можете его выследить и предоставить подтверждающие свидетельства, чтобы оправдать дисциплинарное воздействие. Ведь вы не хотите отвечать перед судом на заявление бывшего служащего о незаконном увольнении?

Жалобы поставщику Интернет-услуг

Если вы решили не возбуждать уголовного или гражданского дела, или если человек, напавший на вашу сеть, продолжает это делать, вы можете подать жалобу его поставщику Интернет-услуг и попробовать по крайней мере его отключить. Часто это единственное реальное средство, не требующее больших расходов от организации, подвергшейся атаке хакера. С помощью судебных средств из этой лекции можно проследить нарушителя по крайней мере до его поставщика Интернет-услуг, после чего вы можете подать последнему формальную жалобу с требованием принять дополнительные меры. Большинство поставщиков Интернет-услуг имеют политику надлежащего поведения для своих пользователей, которая, естественно, не санкционирует взлом чужих сетей. Если вы сможете продемонстрировать достаточно доказательств, скорее всего, будут предприняты некоторые действия, от предупреждения до ликвидации счета этого пользователя. В связи с необходимостью сохранения тайны персональных данных пользователей, они обычно раскрываются лишь

по решению суда, но некоторые поставщики Интернет-услуг охотно идут на сотрудничество для обеспечения информационной безопасности. Большинство крупных поставщиков имеет специальный электронный адрес для сообщений о ненадлежащем поведении, по которому можно направить свою жалобу.

Вы должны убедиться, что собранной информации достаточно, для того чтобы они могли найти вашего противника. Имеются в виду прежде всего IP-адреса, связанные с определенными моментами времени. Большинство поставщиков Интернет-услуг выделяют IP-адреса динамически и они меняются всякий раз, когда кто-то входит в сеть. Без информации о времени, соответствующей их журналам, вам, вероятно, не смогут помочь. Если возможно, предоставьте несколько значений времени доступа, чтобы можно было скоррелировать пользователя по нескольким точкам данных, так как их файлы журналов могут быть не синхронизированы с вашими, и время не будет в точности совпадать. Включите также любые другие имеющиеся у вас данные, такие как протоколы выполнения команд, места, куда копировались файлы, и т.д. Поставщик Интернет-услуг также может быть жертвой и ему могут потребоваться эти данные для последующего расследования.

Выработка плана реагирования на инциденты

Подобно планам резервного копирования и восстановления после аварий (они ведь у вас имеются, не так ли?), у вас должен быть план реагирования на проявления компьютерной преступности. Это поможет вам действовать правильно, как до инцидента, так и после него, чтобы иметь надежный фундамент и не создавать себе лишних проблем. Это большая тема, которой посвящены специальные книги, но по сути вы должны задокументировать последовательность действий при возникновении инцидента, чтобы вы могли ее выполнить без лишних сомнений, когда что-то произойдет.

С ведома руководства создайте план, который описывает ваши действия, если происходят определенные события. Позаботьтесь о том, чтобы высшее руководство санкционировало некоторые действия, такие как привлечение правоохранительных органов, иначе ваша работа может оказаться под угрозой. В крупных организациях в реагировании, вероятно, примут участие юристы и отдел по связям с общественностью; тогда дело может быстро уйти из ваших рук, и это хорошо, если вы понимаете свою роль в этом процессе, и другие тоже ее понимают. План действий в общих чертах может выглядеть примерно так:

1. Локализуите проблему. Убедитесь, что ваш противник не сможет нанести дополнительный ущерб.
2. Начните предварительные операции восстановления, не забывая должным образом сохранять все свидетельства.
3. Оцените размер ущерба. Попробуйте быстро определить его денежный эквивалент. Руководство обычно реагирует быстрее, когда речь идет о деньгах.
4. Сообщите о проблеме высшему руководству для принятия решения о передаче дела в правоохранительные органы или о проведении внутреннего расследования.
5. Решите, проводить ли расследование собственными силами или привлечь сторонних профессионалов.
6. Продолжите вашу деятельность, проводя внутреннее расследование или помогая официальным лицам из правоохранительных органов.

Предварительная подготовка для получения доброкачественных судебных данных

Как и в любом деле, должные предварительные действия до того, как случится беда, могут значительно облегчить вашу работу. Если протоколирование и аудит организованы плохо, то ваша судебная деятельность по меньшей мере существенно усложнится или вообще станет невозможной. Конечно, никому не нравится планировать несчастья, однако выполнение следующих рекомендаций впоследствии поможет найти необходимую информацию.

Степень подробности журналов

Если у вас достаточно дискового пространства и процессорного времени, включите протоколирование с самым высоким уровнем детализации, разумным для ваших серверов. Это предоставит значительно больше информации в случае, если необходимо извлечь что-то из журналов, и будет полезно также для устранения серверных проблем. Вы захотите, вероятно, отрегулировать настройки, чтобы найти разумный уровень детализации протоколов. В Windows степень детализации журналов задается с помощью Event Viewer в Administrative Tools. Щелкните мышью на свойствах каждого типа журналов (application, security, system), и вы сможете задать уровень детализации каждого объекта.

Используйте центральный сервер журналирования

Сохранение всех файлов журналов локально на каждом сервере плохо с нескольких точек зрения. Если атакующий сможет проникнуть в машину, то он получит доступ к файлам журналов и сможет изменить их или стереть полностью. Имеются утилиты, которые помогают взломщикам выборочно стирать

файлы журналов с протоколами их деятельности. Если журналы находятся на другом сервере, то взломщик должен будет взломать по крайней мере еще одну машину, чтобы до них добраться. Популярная утилита сервера журналов syslog - хорошее средство, и большинство серверов, маршрутизаторов, межсетевых экранов и других устройств поддерживают этот формат. С точки зрения управления значительно легче иметь все журналы на одном сервере для регулярного просмотра, и, кроме того, вы будете знать, что все они синхронизированы по одним часам. Это подводит нас к следующему пункту.

Синхронизация времени серверов

Вы должны сделать так, чтобы все ваши серверы брали время с центрального сервера, а не полагались на внутренние часы. Часы ПК известны своей неточностью и склонны к дрейфу. Можно применять сетевой протокол времени (Network Time Protocol - NTP), чтобы брать время с центрального сервера, подписаться на атомные часы в Интернете или поддерживать собственный внутренний сервер времени, чтобы иметь точное время. В этом случае протокольное время будет одинаково для всех серверов, что позволяет правильно отслеживать последовательность событий. Нет ничего более разочаровывающего, чем попытка восстановить последовательность событий атаки по журналам с множеством несогласованных часов. Настоятельно рекомендуется использовать общедоступный сервер времени. Большинство из них бесплатны и используют атомные часы для повышения точности. В этом случае ваши журналы, скорее всего, будут соответствовать внешним файлам журналов, таким как протоколы поставщика Интернет-услуг. Время общедоступных часов можно брать на следующих Web-сайтах:

- clock.isc.org
- clock.via.net
- clock.sgi.net
- ttp.nasa.gov
- tick.gpsclock.org

Где искать судебные данные

Имеются очевидные места для поиска информации после компьютерной атаки. Начинать надо с машины или машин, которые были атакованы. Протоколы и ключевые системные файлы часто содержат улики, такие как методы и идентификация нарушителя. Следует также справиться во всех задействованных системах обнаружения вторжений. Эти средства могут первыми просигнализировать об инциденте. Такие средства, как Tripwire (см. [лекцию 7](#)), могут быть бесценны при выяснении того, что было сделано, и была ли скомпрометирована система.

Важная информация нередко располагается и в самых невероятных местах, таких как каталог пользователя в случае взлома системного счета или во временных каталогах, созданных вашим противником. Если возможно, поместите систему в карантин и прочешите частым гребнем. Описанные далее в этой лекции средства помогут выполнить этот процесс.

Не ограничивайтесь только подозрительными компьютерами. Часто есть смысл поискать в других местах, помимо атакованных машин, чтобы найти информацию о злоумышленниках. Хотя они могут стереть локальные журналы на скомпрометированных машинах, иногда можно найти их следы на соседних серверах или устройствах. Атака редко бывает успешной с первого раза. Обычно атакующий вынужден проверить несколько машин, чтобы найти уязвимую. Эта активность отражается в файлах журналов соседних машин, где вы можете найти свидетельства зондирующего сканирования. Признаки нетипичной активности можно обнаружить также на маршрутизаторах и межсетевых экранах. Проверьте журналы в окрестности времени вторжения (здесь-то как раз и важна синхронизированность журнальных файлов), в том числе журналы вашего общедоступного web-сервера. Когда хакеры находят уязвимый сервер, они часто заходят на web-сайт, ассоциированный с этим доменом, чтобы проверить, кого они взломали. Попробуйте выявить IP-адреса, фигурирующие в различных журналах.

Догматы надлежащего судебного анализа

Существуют различные приемы и методы выполнения судебного анализа информационных систем и множество программных средств, способных помочь в этой деятельности. Однако можно дать некоторые основополагающие рекомендации, которым желательно следовать всегда.

Оперируйте с системой, отсоединенной от сети

Если возможно, полностью отсоедините исследуемую систему от сети во время сбора данных. Если система соединена с сетью, при сборе данных вы можете иметь дело с движущейся целью. Файлы журналов могут заполняться, дисковые области - перезаписываться, а сервисы - отключаться. В худшем

случае, если атакующие все еще имеют доступ к системе (а вы никогда не можете быть полностью уверены в обратном), они могут обнаружить вашу активность и замести следы, став неуловимыми.

Если система была отключена в результате атаки, вы можете оказаться под сильным давлением требующих вернуть ее в сеть как можно быстрее. Для производственных систем, продолжающих работать, также возможно сопротивление их отключению. Мера, конечно, непопулярная, но попробуйте отключить систему от сети, по крайней мере на время сбора данных. Подождите окончания рабочего дня, если необходимо, и объявите это периодом обслуживания системы. Сделайте копию подозрительных данных (если можно, скопируйте весь жесткий диск). Затем вы можете вернуть систему в эксплуатацию и свести к минимуму длительность ее отключения для пользователей на время выполнения вашей работы. Это подводит нас к следующему пункту.

Работайте с копиями свидетельств

Применяйте программное обеспечение создания образов данных, такое как средство dd, представленное далее в этой лекции, чтобы сделать копии свидетельств для работы с ними. Если вы планируете возбудить судебное дело, уголовное или гражданское, сделайте две копии и запечатайте одну из них в защищенный контейнер. Это обеспечит целостность свидетельств и сделает ваше дело менее уязвимым для обвинений в незаконных свидетельствах. Кроме того, если вы случайно сделаете ошибку и удалите некоторые важные свидетельства, вы всегда сможете вернуться к заведомо полноценной копии. Если можно, выполните эти начальные действия в присутствии свидетелей. Лучше всего, если это будет беспристрастная третья сторона. Отпечатайте сопроводительную записку с указанием имени создателя, даты и времени, а затем фиксируйте каждый момент передачи материалов в другие руки, с подписью и датой.

Применяйте хэши для обеспечения свидетельств целостности

При создании копий данных и получении других свидетельских файлов есть смысл создавать MD5 хэши данных и записывать их. Некоторые средства, такие как The Coroner's Toolkit (см. раздел о Sleuth Kit далее в этой лекции), делают это автоматически. Можно также применить одно из средств шифрования, рассмотренных в [лекции 9](#), такое как PGP или GnuPG. Помимо всего прочего, если аутентичность ваших данных будет оспариваться, вы сможете доказать, что копия, с которой вы работали, являлась электронной копией содержимого атакованной машины. Это поможет также выявить различия между файлами и увидеть, не были ли внесены какие-либо изменения утилитами системного уровня.

Применяйте доверенные загрузочные носители и исполнимые файлы

При проверке системы целесообразно применять для загрузки доверенные носители, такие как загрузочный флоппи-диск или компакт-диск. Их можно создать в процессе установки ОС. Некоторые из рассматриваемых средств создают собственную загрузочную среду. Это особенно важно, если вы работаете на взломанной системе. Если атакующий с помощью специальных средств сумел скомпрометировать системные бинарные файлы, то все результаты, получаемые от утилит на этом жестком диске, должны считаться подозрительными. Помимо возможной перезаписи дат файлов и других критичных данных, атакующий мог оставить некоторые временные бомбы или выполняющиеся демоны, способные вызвать дальнейшие повреждения или стереть свидетельства.

Можно создать загрузочный компакт-диск для реагирования на инциденты, содержащий все необходимые программы. Вам понадобятся диски для систем Windows и UNIX, если у вас разнородная среда.

Средства судебного анализа

Одна из проблем, с которой сталкиваются компьютерные следователи, состоит в том, что обычные файловые утилиты могут необратимо изменить файлы, фактически "смазывая" картину преступления и удаляя нужные вам свидетельства. Например, просмотр файлов с помощью обычного редактора изменит такие вещи, как временные метки. Представьте, что кто-нибудь топчется в грязных ботинках на месте реального преступления и двигает объекты по всему помещению. То же происходит при осмотре системы без подходящих средств. Вы не только лишитесь возможности принять какие-либо уголовные или гражданские меры, но можете также стереть цифровые следы атакующего. Хакеры часто применяют средства, скрывающие процессы и файлы от обычных системных утилит, поэтому вам нужны специальные инструменты, действующие вне обычной операционной системы, чтобы увидеть больше того, что видит ОС.

В последующих разделах представлены средства как для Linux, так и для Windows. Сначала мы рассмотрим несколько судебных средств уровня операционной системы, а затем - полнофункциональный инструментарий для более глубокого анализа. Помните, что применение средств уровня операционной системы может возвращать неверные или поддельные данные, если ваша ОС действительно была скомпрометирована.

Fport: Средство идентификации процессов для Windows

Fport

Автор/основной контакт: Foundstone, Inc.

Web-сайт: <http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm>

Платформы: Windows NT, 2000, XP

Лицензия: Freeware

Рассмотренная версия: 2.0

Это небольшое добавление к системе может быть полезно при исследовании машины на предмет подозрительной активности. Нередко вирус, резидентный в памяти, или "тройанская" программа проявляются как процесс, выполняющийся под странным именем или с необычным портом. Fport ищет открытые сетевые порты TCP или UDP и выдает их вместе с идентификатором ассоциированного процесса, именем процесса и маршрутным именем. Программа Fport аналогична собственной команде Windows `netstat` за исключением того, что предоставляет несколько больше информации и позволяет различным образом форматировать вывод для анализа. Это помогает отследить подозрительные программы, которые открывают сетевые порты на вашей машине. Подобное поведение служит признаком "тройанской" программы.

Конечно, не каждый неопознанный процесс является вредоносной программой, но желательно понять, что делают странные на вид сервисы, особенно с нестандартными маршрутными префиксами (отличными от системных каталогов Windows и подобных), странными или хакерскими именами.

Программа Fport создана и распространяется компанией Foundstone Corporation, занимающейся разработкой защитного программного обеспечения и оказывающей консультационные услуги. Компания предлагает несколько других свободных средств безопасности, и их web-сайт в любом случае стоит посетить. Хотя исходные тексты программы Fport не вполне открыты (распространяются только бинарные файлы), она условно свободна, и для ее применения в коммерческих целях имеются лишь незначительные ограничения.

Установка Fport

Загрузите zip-файл с web-сайта Foundstone и распакуйте его в отдельном каталоге. Там появятся два файла - исполнимый файл Fport и небольшой информационный README.

Применение Fport

Программа Fport помогает определить, была ли машина взломана и откуда пришел нарушитель. Она должна выполняться на живой системе, то есть включенной и работающей: Fport не может выполняться на статических данных.

Для запуска Fport в каталоге с исполнимым файлом введите в командной строке `fport`. Будет распечатан список всех портов, открытых в данный момент, и ассоциированных с ними приложений ([листинг 11.1](#)).

```
Port v2.0 - TCP/IP Process to Port Mapper
Copyright 2000 by Foundstone, Inc.
http://www.foundstone.com
```

```
Pid  Process      Port  Proto Path
940  svchost    -> 135   TCP  C:\WINDOWS\system32\svchost.exe
```

```

4      System    -> 139    TCP
4      System    -> 445    TCP
1348   WCESCOMM  -> 990    TCP C:\Program Files\Microsoft ActiveSync\WCESCOMM.EXE
4072   WCESMgr   -> 999    TCP C:\Program Files\Microsoft ActiveSync\WCESMgr.exe
1032   svchost   -> 1025   TCP C:\WINDOWS\System32\svchost.exe
1032   svchost   -> 1031   TCP C:\WINDOWS\System32\svchost.exe
1032   svchost   -> 1034   TCP C:\WINDOWS\System32\svchost.exe
4      System    -> 1042   TCP
4072   WCESMgr   -> 2406   TCP C:\Program Files\Microsoft ActiveSync\WCESMgr.exe
2384   websearch -> 3008   TCP C:\Program Files\websearch\websearch.exe
1144   -> 54321  TCP C:\temp\cmd.exe
4072   WCESMgr   -> 5678   TCP C:\Program Files\Microsoft ActiveSync\WCESMgr.exe
2384   websearch -> 8755   TCP C:\Program Files\websearch\websearch.exe
136    javaw     -> 8765   TCP C:\WINDOWS\System32\javaw.exe
1348   WCESCOMM  -> 123    UDP C:\Program Files\Microsoft ActiveSync\WCESCOMM.EXE
2384   websearch -> 123    UDP C:\Program Files\websearch\websearch.exe
940    svchost   -> 135    UDP C:\WINDOWS\system32\svchost.exe
1144   -> 137    UDP
1932   svchost   -> 1026   UDP C:\WINDOWS\System32\svchost.exe

```

Листинг 11.1. Выдача Fport

При просмотре этого листинга взгляд скользит по нормальным на вид выполняющимся службам и программам, пока где-то в середине не натывается на программу cmd.exe, запущенную из каталога Temp. Это бинарный файл командного интерпретатора, и ему нечего делать в каталоге Temp. Тот факт, что у службы нет имени, также подозрителен. Наконец, номер входного порта не соответствует ни одному из известных сервисов. На самом деле, если поискать его в базе данных известных троянских программ в Интернете (<http://www.simovits.com/trojans/trojans.html>), можно обнаружить совпадение с номером порта документированной троянской программы. Это служит весомым свидетельством того, что система была взломана. Теперь вы должны решить, нужно ли выключить систему, чтобы провести дополнительный судебный анализ.

В [табл. 11.1](#) перечислено несколько опций Fport для сортировки вывода. Можно также использовать опцию `-h` для вывода краткой справочной информации.

Таблица 11.1. Опции сортировки Fport

| Опция | Описание |
|-------|--|
| -a | Сортировка вывода по имени приложения |
| -ap | Сортировка вывода по маршруту приложения |
| -i | Сортировка вывода по идентификатору процесса (PID) |
| -p | Сортировка вывода по номеру порта. |

Если процессов много, можно использовать эти ключи для просмотра программ с большими номерами портов, характерными для вредоносного ПО. Можно также отсортировать вывод по маршруту приложения или имени, чтобы выявить нестандартные приложения.

Isof: Средство идентификации портов и процессов для UNIX

Isof

Автор/основной контакт: Ray Show

Web-сайт: <http://freshmeat.net/projects/Isof/>

Платформы: Linux и большинство UNIX

Лицензия: GPL

Рассмотренная версия: 4.68

Зеркалирующие сайты (допускающие анонимный доступ по FTP без обратного DNS):

thewiretapped.net/pub/security/host-security/lsof

ftp.tau.ac.il/pub/unix/admin/

Это средство аналогично только что рассмотренному Fport для Windows. LSOF (LiSt Open Files) ассоциирует открытые файлы с процессами и пользователями. Оно напоминает команду `netstat`, но выдает также сетевые порты, используемые сервисом. Это важно при попытке отследить активную программу в сети. Зачастую единственным способом найти неуловимые ошибки является наблюдение за тем, какие сетевые порты открываются.

Средство `lsof` предустанавливается в некоторых дистрибутивах UNIX и Linux и доступно в форме RPM на установочных дисках других, таких как Mandrake и RedHat Linux. Чтобы выяснить, установлено оно или нет, наберите `lsof` и посмотрите, каков будет ответ.

Установка `lsof`

1. Загрузите tar-файл с прилагаемого к книге компакт-диска или с официального web-сайта. Если IP-адрес, с которого выполняется загрузка, не имеет обратной записи DNS, то основной FTP-сайт не позволит с ним соединиться. Попробуйте один из указанных зеркалирующих сайтов.
2. Распакуйте tar-файл.
3. Вы увидите несколько текстовых файлов и еще один tar-файл, что-нибудь вроде `lsof_4.68_src`. В этом файле содержатся исходные тексты. Распакуйте его и войдите в этот каталог.
4. Прежде чем начинать процесс компиляции, необходимо выяснить сокращенный код вашего диалекта UNIX. Так как программа `lsof` переносима практически на любую версию UNIX, требуется сообщить, какая разновидность UNIX применяется, чтобы процедура конфигурирования могла настроить ее для вашей системы.

Чтобы выяснить коды различных версий UNIX, введите

```
./configure -h
```

Например, код Linux - `linux` (не правда ли, просто?)

5. Когда вы будете готовы, наберите следующую команду

```
./configure код_диалекта_UNIX
```

Программа будет сконфигурирована для компиляции.

6. Когда конфигурирование закончится, наберите:

```
make
```

7. Это завершает процесс сборки.

Программа `lsof` готова к употреблению.

Применение `lsof`

Программа lsof имеет множество применений и подробную оперативную справку, а также несколько информационных файлов README для различных приложений. Однако в данном разделе рассматривается лишь несколько специфических команд, полезных для судебных исследований.

Если вы хотите увидеть все открытые в данный момент файлы в системе и ассоциированные с ними процессы, наберите

```
lsof -n
```

Опция -n предписывает lsof не пытаться разрешать записи DNS для каждого IP-адреса, подключившегося к вашей машине. Это существенно ускоряет процесс. Примерный вид выдачи показан на [листинге 11.2](#).

| COMMAND | PID | USER | FD | TYPE | DEVICE | SIZE | NODE | NAME |
|---------|------|--------|-----|------|------------|---------|--------|-------------------------|
| xfs | 903 | xfs | 0r | DIR | 3,1 | 4096 | 2 | / |
| atd | 918 | daemon | rtd | DIR | 3,1 | 4096 | 2 | / |
| atd | 918 | daemon | txt | REG | 3,6 | 14384 | 273243 | /usr/sbin/atd |
| sshd | 962 | root | cwd | DIR | 3,1 | 4096 | 2 | / |
| sshd | 962 | root | rtd | DIR | 3,1 | 4096 | 2 | / |
| sshd | 962 | root | txt | REG | 3,6 | 331032 | 274118 | /usr/sbin/sshd |
| dhcpcd | 971 | root | cwd | DIR | 3,1 | 4096 | 2 | / |
| dhcpcd | 971 | root | rtd | DIR | 3,1 | 4096 | 2 | / |
| dhcpcd | 971 | root | txt | REG | 3,1 | 31576 | 78314 | /sbin/dhcpcd |
| xinetd | 1007 | root | cwd | DIR | 3,1 | 4096 | 2 | / |
| xinetd | 1007 | root | 5u | IPv4 | 1723 | | TCP | 127.0.0.1:1024 (LISTEN) |
| xinetd | 1007 | root | 8u | unix | 0xc37a8540 | | 1716 | socket |
| rwho | 1028 | root | cwd | DIR | 3,1 | 4096 | 61671 | /var/spool/rwho |
| rwho | 1028 | root | rtd | DIR | 3,1 | 4096 | 61671 | /var/spool/rwho |
| rwho | 1028 | tim | cwd | DIR | 3,1 | 4096 | 61671 | /var/spool/rwho |
| crond | 1112 | root | cwd | DIR | 3,1 | 4096 | 14 | /var/spool |
| crond | 1112 | root | lw | FIFO | 0,5 | | 1826 | pipe |
| | 1112 | root | 2w | FIFO | 0,5 | | 1827 | pipe |
| nessusd | 1166 | root | cwd | DIR | 3,1 | 4096 | 2 | / |
| nessusd | 1166 | root | rtd | DIR | 3,1 | 4096 | 2 | / |
| nessusd | 1166 | root | txt | REG | 3,6 | 1424003 | 323952 | |
| init | 1 | root | cwd | DIR | 3,1 | 4096 | 2 | / |
| init | 1 | root | rtd | DIR | 3,1 | 4096 | 2 | / |
| init | 1 | root | txt | REG | 3,1 | 31384 | 75197 | /sbin/init |

Листинг 11.2. Вывод команды lsof -n

Соединения на этом листинге выглядят нормально, вопросы вызывает лишь подключение через службу rwho. Стоит убедиться, что допустимый пользователь применяет команду законно. Если этот счет принадлежит кому-то из нетехнического персонала, может понадобиться дальнейшее расследование.

lsof можно также применять для поиска определенного файла. Если вы хотите увидеть, обращался ли кто-то к файлу паролей, можно воспользоваться следующей командой:

```
lsof маршрут/имя_файла
```

Замените маршрут/имя_файла маршрутом и именем файла, который вас интересует, в данном случае - /etc/passwd. Необходимо задать для lsof полное маршрутное имя, чтобы программа нашла этот файл.

Еще один вариант применения lsof - получение списка всех открытых сокетов. В этом случае можно будет увидеть работающий сервер, о котором вы не знаете. Формат этой команды таков:

```
lsof -i
```

В результате получается выдача, аналогичная представленной на [листинге 11.3](#). На нем можно видеть все выполняющиеся программы, включая sshd и nessusd - демоны для SSH и Nessus. Можно даже видеть отдельные соединения с этими службами. Похоже, что кто-то использует в данный момент сервер Nessus. Проверив IP-адрес, можно понять, что это внутренний пользователь. На самом деле это ваша собственная машина! Поэтому беспокоиться не о чем.

```
COMMAND  PID USER  FD  TYPE DEVICE SIZE NODE NAME
portmap   733  rpc   3u  IPv4  1417      UDP  *:sunrpc
portmap   733  rpc   4u  IPv4  1426      TCP  *:sunrpc (LISTEN)
sshd      962  root   3u  IPv4  1703      TCP  *:ssh (LISTEN)
xinetd  1007  root   5u  IPv4  1728      TCP  localhost.localdomain:1024 (LISTEN)
rwhod    1028  root   3u  IPv4  1747      UDP  *:who
nessusd  1166  root   4u  IPv4  1971      TCP  *:1241 (LISTEN)
nessusd  1564  root   5u  IPv4  1972      TCP  192.168.1.101:1024->192.168.1.2:1994 (ESTABLISHED)
```

Листинг 11.3. Вывод команды lsof -i

Можно задать для просмотра определенный IP-адрес или хост, помещая знак @ и адрес после ключа -i. Например, команда

```
lsof -i@192.168.1.0/24
```

отображает все соединения, исходящие из вашей сети, при условии, что ваша внутренняя сеть в нотации с косой чертой задается как 192.168.1.0/24.

Просмотр файлов журналов

Необходимо внимательно просмотреть файлы журналов, когда вы ищете признаки беды. Файлы журналов Windows можно найти в разделе Event Viewer из Administrative Tools. В Linux и BSD-вариантах UNIX файлы журналов находятся в каталоге /var/log/. В других вариантах UNIX эти файлы также могут присутствовать, но их расположение может отличаться. В [табл. 11.2](#) перечислены основные файлы журналов UNIX и их назначение.

Таблица 11.2. Файлы журналов UNIX

| Файл журнала | Описание |
|-------------------|---|
| /var/log/messages | Хранит общие системные сообщения |
| /var/log/secure | Хранит сообщения аутентификации и безопасности |
| /var/log/wtmp | Хранит историю входов в систему и выходов из нее |
| /var/run/utmp | Хранит динамический список пользователей, находящихся в данный момент в системе |
| /var/log/btmp | Только для Linux. Хранит все неудачные или неверные попытки входа. |

Эти файлы могут располагаться несколько иначе или не существовать в других версиях UNIX. Программы также часто создают собственные файлы журналов, хранящиеся обычно в каталоге /var. Для просмотра этих файлов и поиска определенных цепочек символов или чисел (таких как IP-адреса и имена пользователей) можно воспользоваться текстовым редактором.

В [табл. 11.3](#) перечислены несколько команд уровня операционной системы, которые можно применять в системах Linux и UNIX для быстрого просмотра этих файлов.

Таблица 11.3. Команды просмотра Linux и UNIX

| Команда | Описание |
|---------|----------|
|---------|----------|

| | | | | | |
|-------|---|--|--|--|--|
| users | Извлекает из файла utmp и выдает список пользователей, находящихся в данный момент в системе | | | | |
| w | Выдает детальную информацию о пользователях, находящихся в системе, в том числе: как они вошли (локально или удаленно), IP-адрес, если вход удаленный, какие команды они выполняют. Эта команда очень полезна для поимки нарушителя "с поличным". | | | | |
| last | Выдает наиболее свежие записи файла wtmp. Это также может быть весьма полезно, чтобы увидеть, кто, когда и насколько входит в систему. На листинге 11.4 приведен пример подобной выдачи | | | | |
| lastb | Только для Linux. Делает то же, что и предыдущая команда, но для файла btmp - протокола неправильных входов. Здесь нарушитель может проявиться в первую очередь, если он совершил много неудачных попыток входа | | | | |

```

tony pts/0 10.1.1.1 Sun Sep 5 23:06 still logged in
tony pts/0 10.1.1.1 Sun Sep 5 22:44 - 23:04 (00:20)
tony pts/0 10.1.1.1 Sun Sep 5 21:08 - 21:16 (00:07)
tony pts/0 10.1.1.1 Sun Sep 5 20:20 - 20:36 (00:16)
reboot system boot 2.4.18-14 Sun Sep 5 17:32 (05:34)
tony tty1 Sun Sep 5 17:29 - down (00:01)
tony pts/2 10.1.1.1 Sat Sep 4 23:02 - 23:34 (00:32)
tony pts/2 10.1.1.1 Sat Sep 4 22:36 - 22:36 (00:00)
hank pts/0 10.1.1.200 Sat Sep 4 12:13 - 12:22 (00:08)
hank pts/0 adsl-66-141-23-1 Fri Sep 3 23:53 - 23:53 (00:00)
hank pts/0 192.168.1.100 Fri Sep 3 14:47 - 14:47 (00:00)
tony pts/3 192.168.1.139 Fri Sep 3 09:59 - down (00:01)
larry pts/3 adsl-65-67-132-2 Thu Sep 2 22:59 - 23:11 (00:12)
tony pts/3 10.1.1.1 Thu Sep 2 21:33 - 21:49 (00:16)
brian pts/3 adsl-65-68-90-12 Thu Sep 2 18:23 - 18:31 (00:07)
hank pts/5 192.168.1.139 Thu Sep 2 14:29 - 15:35 (01:06)
sam pts/ dialup-207-218-2 Wed Sep 1 22:24 - 00:40 (02:16)

```

Листинг 11.4. Выдача команды last

Следует учитывать, что если ваша система была скомпрометирована, эти программы могут быть заменены "троянскими" копиями. Такие программы, как Tripwire (см. [лекцию 7](#)), способны помочь определить, были ли искажены системные бинарные файлы. Вы должны сделать заведомо хорошие копии этих бинарных файлов, чтобы было возможным исполнение с безопасного загрузочного носителя, а не из системы. Помните также, что атакующие часто будут выборочно редактировать файлы журналов, чтобы стереть все следы своей деятельности. Однако если они просто удалят файлы журналов, вы, возможно, сумеете их восстановить. Кроме того, следует проверить все файлы журналов, так как некоторые новички удаляют лишь часть из них.

Создание копий судебных свидетельств

Если вы убедились, что ваша система была атакована или взломана, то в первую очередь следует немедленно остановить атаку или ограничить риски, которым подвергается эта машина. В идеале это означает отсоединение машины от сети для проведения дальнейшего анализа. Если это невозможно, все равно желательно отключить все подозрительные системные счета, терминировать все незаконные процессы, и, возможно, заблокировать на межсетевом экране IP-адреса нарушителей, пока вы не уясните, что происходит.

После устранения непосредственной опасности необходимо сделать копии всех важных данных для просмотра в автономном режиме в соответствии с догматами надлежащего судебного анализа, сформулированными выше. Нежелательно применять ваши средства к живым данным, сделайте их полноценную копию. Для этого требуется создать образ данных, а не просто их скопировать. Нежелательно применять встроенные в операционную систему функции копирования, так как они могут изменять временные метки файлов и вставлять другую нежелательную информацию. Имеются специальные средства для получения зеркальных копий образов. К сожалению, в настоящее время не существует хорошего варианта подобных средств с открытыми исходными текстами для платформы Windows (кто-нибудь хочет включиться в хороший проект с открытыми исходными текстами для

Windows?). Наиболее популярной программой для Windows является Norton Ghost компании Symantec, которая продается примерно за \$50. В UNIX для этого существует прекрасная программа с открытыми исходными текстами dd, что означает дамп данных.

dd: Средство тиражирования дисков и файлов

dd

Авторы/основные контакты: Paul Rubin, David MacKenzie и Stuart Kem

Web-сайт: <http://mirrors.kernel.org/gnu/fileutils/>

Платформы: Большинство Linux и UNIX

Лицензия: GPL

Рассмотренная версия: Недоступна

Другие ресурсы:

Наберите man dd в командной строке.

Программу dd можно применять для буквального чтения блоков данных непосредственно с жесткого диска и создания их точных копий. Она напрямую обращается к носителю, без посредничества файловой системы, поэтому способна извлечь удаленные данные и другие вещи, которых файловая система не может видеть. Ее можно применять для создания побитных копий данных файловых систем UNIX. Поскольку UNIX трактует устройства как файлы, то можно взять весь жесткий диск и тиражировать его путем простого копирования файла устройства с помощью такого средства, как dd.

Установка dd

В большинстве операционных систем UNIX устанавливать программу dd не требуется, поскольку она является частью любой файловой системы UNIX. Наберите man dd, чтобы убедиться в ее наличии. Если по какой-то причине ее нет, можно взять ее с прилагаемого к книге компакт-диска или как часть файловых утилит GNU с приведенного выше сайта.

Применение dd

Есть два способа применения dd. Один из них - побитное копирование данных. При этом создается зеркальный образ данных на другом жестком диске или разделе диска. Другой способ - создание одного большого файла. Иногда это удобнее для целей анализа и мобильности. Для верификации можно легко вычислить хэш файла. Этот файл часто называют свидетельским, и многие судебные программы созданы для чтения данных из него.

Основной формат команды dd следующий:

```
dd -if=входной_файл -of=выходной_файл опции
```

где вместо входного файла нужно подставить копируемое устройство, вместо выходного - имя файла, в который производится копирование, а вместо опций - любые опции dd, которые вы хотите использовать. У dd много опций; в табл. 11.4 перечислены основные из них.

Таблица 11.4. Основные опции dd

| Опция | Описание |
|--------|---|
| bs= | Размер блока. Размер в байтах блока, копируемого за один раз. |
| count= | Подсчет блоков. Сколько блоков копировать. Это полезно, если вы не хотите копировать всю файловую систему, поскольку у вас очень большой жесткий диск или раздел диска, или ограниченный объем пространства на целевом носителе |
| skip= | Пропустить заданное число блоков, прежде чем начать копирование. Это также полезно при копировании части файловой системы. |

| | |
|---------|---|
| conv= | Задаёт любую из следующих подопций: |
| notrunc | - не обрезать вывод при возникновении ошибки. Рекомендуется в большинстве случаев. |
| noerror | - не останавливать чтение входного файла в случае ошибки, такой как проблемы с физическим носителем. Также рекомендуется. |
| sync | - требует перед собой команду <code>noerror</code> . Если происходит ошибка, то команда <code>sync</code> подставит на её место нули, сохраняя последовательную непрерывность данных. |

Если вы хотите с помощью `dd` скопировать устройство на приводе жесткого диска `/dev/hdc` на другой привод жесткого диска, устройство `hdd`, можно воспользоваться следующей командой:

```
dd -if=/dev/hdc of=/dev/hdd bs=1024 conv=noerror,notrunc,sync
```

Эта команда копирует содержимое устройства `/dev/hdc` (вероятно, вашего основного жесткого диска) на устройство `/dev/hdd` (вероятно, ваш вторичный жесткий диск). Убедитесь, что вы понимаете, какие диски каким устройствам соответствуют. Как поясняется во врезке о программе `dd`, ошибка здесь может обойтись очень дорого!



Флэми Тех советует:

Будьте очень осторожны с `dd`!

Не проявляйте легкомыслия при использовании низкоуровневых дисковых средств, таких как `dd`. Одна неверная команда может легко затереть весь жесткий диск. Будьте особенно осторожны в отношении входных и выходных файлов. Если их перепутать, можно перезаписать свидетельства или сделать кое-что похуже. Не играйте с `dd`, если не владеете основами работы с жесткими дисками, не знаете, что такое блок и сектор. В отличие от дружелюбной по отношению к пользователям Windows, `dd` никогда не переспрашивает дважды, когда вы собираетесь сделать какую-нибудь глупость. Поэтому, как хороший портной, семь раз прочтите руководство и один раз выполните команду.

Если вы вместо тиражирования дисков хотите создать один большой свидетельский файл, примените следующую команду для копирования файла на новое устройство:

```
dd if=/dev/hdc of=/mnt/storage/evidence.bin
```

Вероятно, вы захотите смонтировать новое устройство для сохранения этого файла. Желательно, чтобы это был совершенно новый носитель, чтобы не испортить свидетельства старыми данными. Помните, что даже стертые данные проявятся при использовании этого средства. Если у вас нет чистого носителя, убедитесь, что применяемый носитель тщательно прочищен с помощью дисковой утилиты (кстати, `dd` имеет такую возможность, прочтите о ней в оперативной справке).

Когда все свидетельства собраны, вы готовы к их дальнейшему анализу с помощью судебного инструментария. Есть много великолепных, профессионального уровня коммерческих наборов средств. Имеются также некоторые очень хорошие свободные наборы судебных средств как для Windows, так и UNIX.

The Sleuth Kit/Autopsy Forensic Browser: Набор судебных средств для UNIX

The Sleuth Kit/Autopsy Forensic Browser

Автор/основной контакт: Brian Carrier

Web-сайт: <http://www.sleuthkit.org/sleuthkit/index.php>

Платформы: Большинство UNIX

Лицензия: Публичная лицензия IBM

Рассмотренная версия: 1.70

Списки почтовой рассылки:

The Sleuth Kit User's list

Общие вопросы и обсуждение Sleuth Kit. Подписка по адресу

<http://lists.sourceforge.net/lists/listinfo/sleuthkit-users>

The Sleuth Kit Informer list

Ежемесячный бюллетень с новостями, советами и рекомендациями. Подписка по адресу

<http://www.sleuthkit.org/informer/index.php>

The Sleuth Kit Developer's list

Для вопросов и обсуждений разработчиков. Подписка по адресу

<http://lists.sourceforge.net/lists/listinfo/sleuthkit-developers>

The Sleuth Kit Announcement list

Список рассылки только для чтения с основными объявлениями или выпусками Sleuth Kit/Autopsy Forensic Browser. Подписка по адресу

<http://lists.sourceforge.net/lists/listinfo/sleuthkit-announce>

The Coroner's Toolkit (TCT) list

Информация о TCT, на котором основывается Sleuth Kit. Подписка по адресу

http://www.porcupine.org/forensics/tct.htm#mailing_list.

Sleuth Kit Брайана Карьера является собранием различных судебных средств, работающих под UNIX. Он содержит части популярного Coroner's Toolkit Дэна Фармера, равно как и вклады других людей, и предоставляет стильный Web-интерфейс на базе Autopsy Forensic Browser. Sleuth Kit предназначается для работы с файлами данных, такими как вывод дисковых утилит, аналогичных dd, и обладает широкими возможностями, фактически превосходя некоторые доступные коммерческие программы. В число ключевых функций Sleuth Kit входят:

- Отслеживание различных дел и нескольких следователей.
- Просмотр файлов и каталогов, которые были размещены в файловой системе или удалены из нее.
- Доступ к низкоуровневым структурам файловой системы.
- Генерация хронологии файловой активности.
- Сортировка по категориям файлов и проверяемым расширениям.
- Поиск в данных образов по ключевым словам.
- Идентификация графических образов и создание пиктограмм.
- Поиск в базах данных хэшей, включая судебные стандарты NIST NSRL и Hash Keeper.

- Создание заметок следователя.
- Генерация отчетов.

Установка Sleuth Kit

1. Загрузите и распакуйте файл с прилагаемого к книге компакт-диска или web-сайта.
2. В этом же каталоге наберите:

```
make
```

Программа автоматически сконфигурирует и скомпилирует себя. В процессе установки она может задать вам несколько вопросов.

Установка Autopsy Forensic Browser

Эта программа - графический интерфейсный компонент для Sleuth Kit. Ее применение вместе с Sleuth Kit существенно облегчит вашу жизнь и позволит порождать привлекательный графический вывод. При желании можно по-прежнему независимо применять средства командной строки Sleuth Kit.

1. Прежде чем начинать установку Autopsy, удостоверьтесь, что инструментарий Sleuth Kit установлен.
2. Возьмите файл Autopsy с web-сайта или из каталога /autopsy прилагаемого к книге компакт-диска.
3. Распакуйте его с помощью обычной команды `tar -zxvf`.
4. Держите маршрут к программному каталогу Sleuth Kit под рукой и подумайте о том, где разместить "сундук с уликами" - специальный каталог, где будут располагаться все данные рассматриваемого с применением Sleuth Kit дела.
5. Наберите команду `make`. Она установит программу и при этом попросит вас указать каталог для хранения данных и каталог, в который установлен Sleuth Kit.

Применение Sleuth Kit и Autopsy Forensic Browser

1. Чтобы запустить серверную программу, наберите `./autopsy &` в каталоге autopsy. Сервер будет работать в фоновом режиме с портом 9999.
2. Скопируйте универсальный локатор ресурсов, который выдается при запуске сервера. Он понадобится для входа в серверную систему.
3. Для подключения к серверу откройте web-навигатор и введите URL, который вы скопировали из адресного окна на шаге 2. Он выглядит примерно так: <http://localhost:9999/654378938759042387490587/autopsy>

Число между косыми чертами изменяется при каждом запуске Sleuth Kit. После ввода URL появится основной экран ([рис. 11.1](#)).

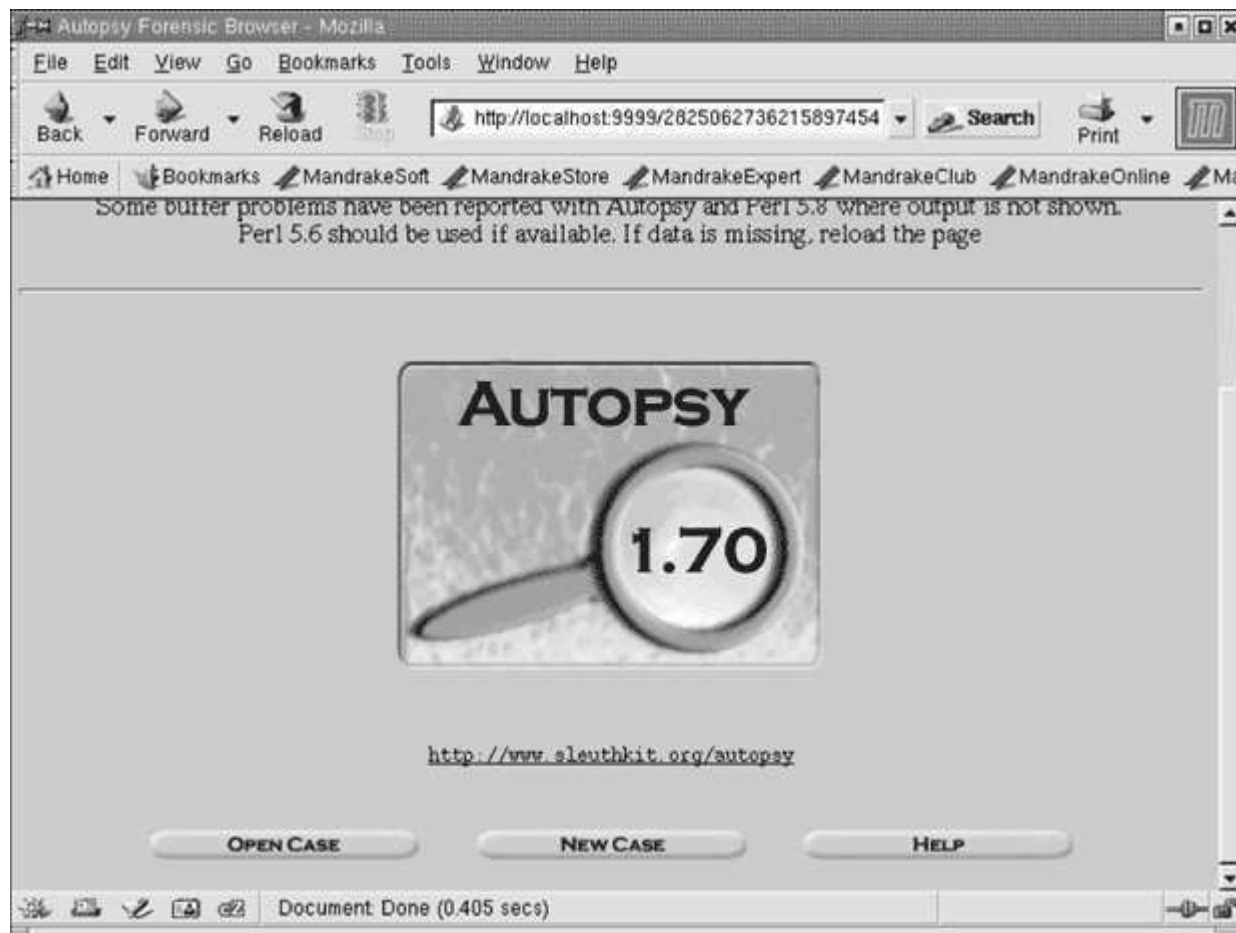


Рис. 11.1. Основной экран Autopsy Forensic Browser

Заведение и протоколирование дела

Sleuth Kit вместе с Autopsy Forensic Browser позволяет контролировать несколько дел, чтобы можно было отслеживать различные инциденты и различных заказчиков. Необходимо завести дело для хранения свидетельских файлов, прежде чем с ними можно будет работать.

1. На основном экране щелкните мышью на New Case. Появится экран заведения нового дела ([рис. 11.2](#)).

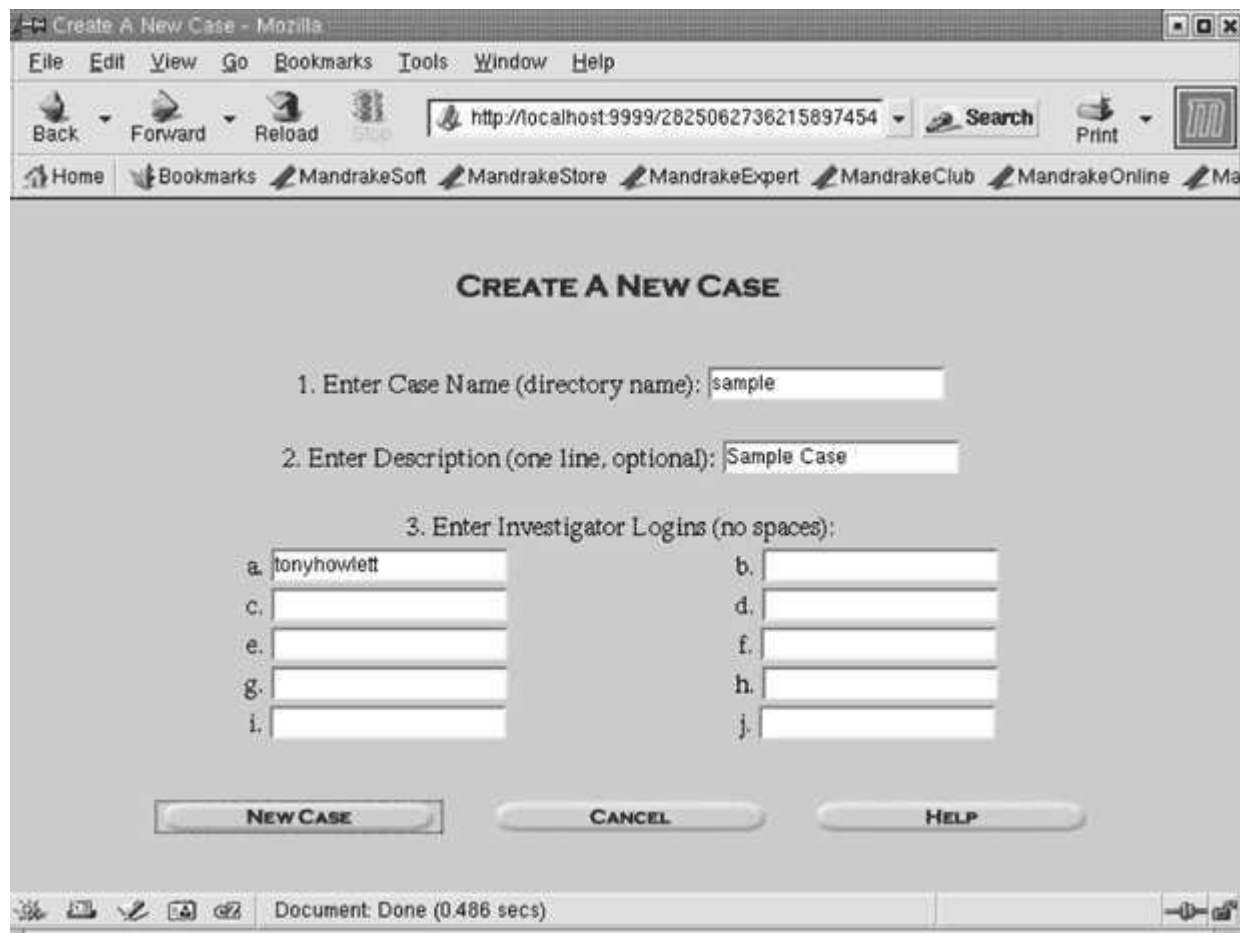


Рис. 11.2. Экран Create a New Case

- Введите название дела. Оно послужит именем каталога, в котором хранятся свидетельские данные. Этот каталог будет создан в основном каталоге для хранения свидетельств, заданном при установке.
- При желании можно дать делу полное имя, которое лучше его характеризует.
- Необходимо создать по крайней мере один идентификатор следователя для доступа к делу. В этом проявляется развитость программы. Данная возможность позволяет подключать к работе над одним делом несколько человек и отслеживать доступ и действия каждого. Щелкните мышью на New Case, чтобы завершить ввод.
- Когда дело заведено, выводится Case Gallery с отображением всех заведенных вами дел. Можно видеть детали каждого дела, включая следователей, которые с ним работают. Выберите свое новое дело, щелкните мышью на OK и войдите в него.

Теперь вы завели дело, вошли в него и готовы с ним работать.

Добавление хоста

Когда вы вошли в дело, необходимо задать по крайней мере один хост, который вы собираетесь освидетельствовать. Этот хост представляет конкретную исследуемую машину.

- В Case Gallery щелкните мышью на Add Host. Появится экран Add a New Host ([рис. 11.3](#)).

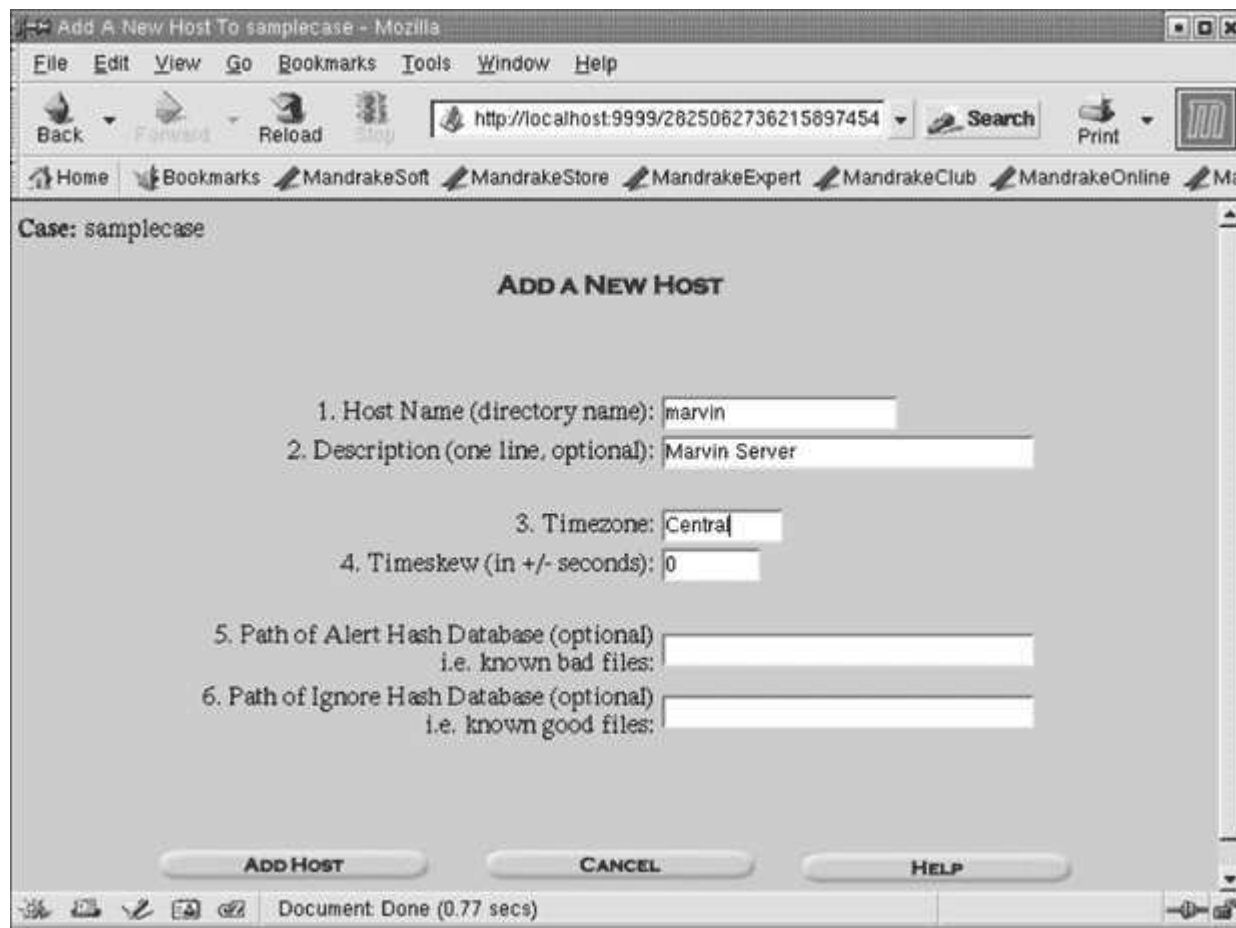


Рис. 11.3. Экран Add a New Host

2. Введите имя хоста.
3. При желании введите краткое описание хоста.
4. Задайте часовой пояс и отклонение часов - расхождение с временной меткой основного файла дела, чтобы Sleuth Kit по-особому трактовал временные метки на хосте. Это может быть очень важно при просмотре нескольких серверов с различным временем на часах.
5. При желании добавьте необязательную запрошенную информацию.
6. Щелкните мышью на Add Host, чтобы добавить хост и вернуться в Case Gallery.
7. Выполните эту процедуру для каждого хоста, на котором имеются данные.

Добавление образа

Теперь следует добавить образы данных для созданных хостов. Используйте копии данных, сделанные с помощью dd, Norton Ghost или какой-либо иной утилиты тиражирования данных.

1. Выберите хост на экране Host Gallery и щелкните мышью на OK.
2. Щелкните мышью на кнопке Add Image. Появится экран Add a New Image ([рис. 11.4](#)).

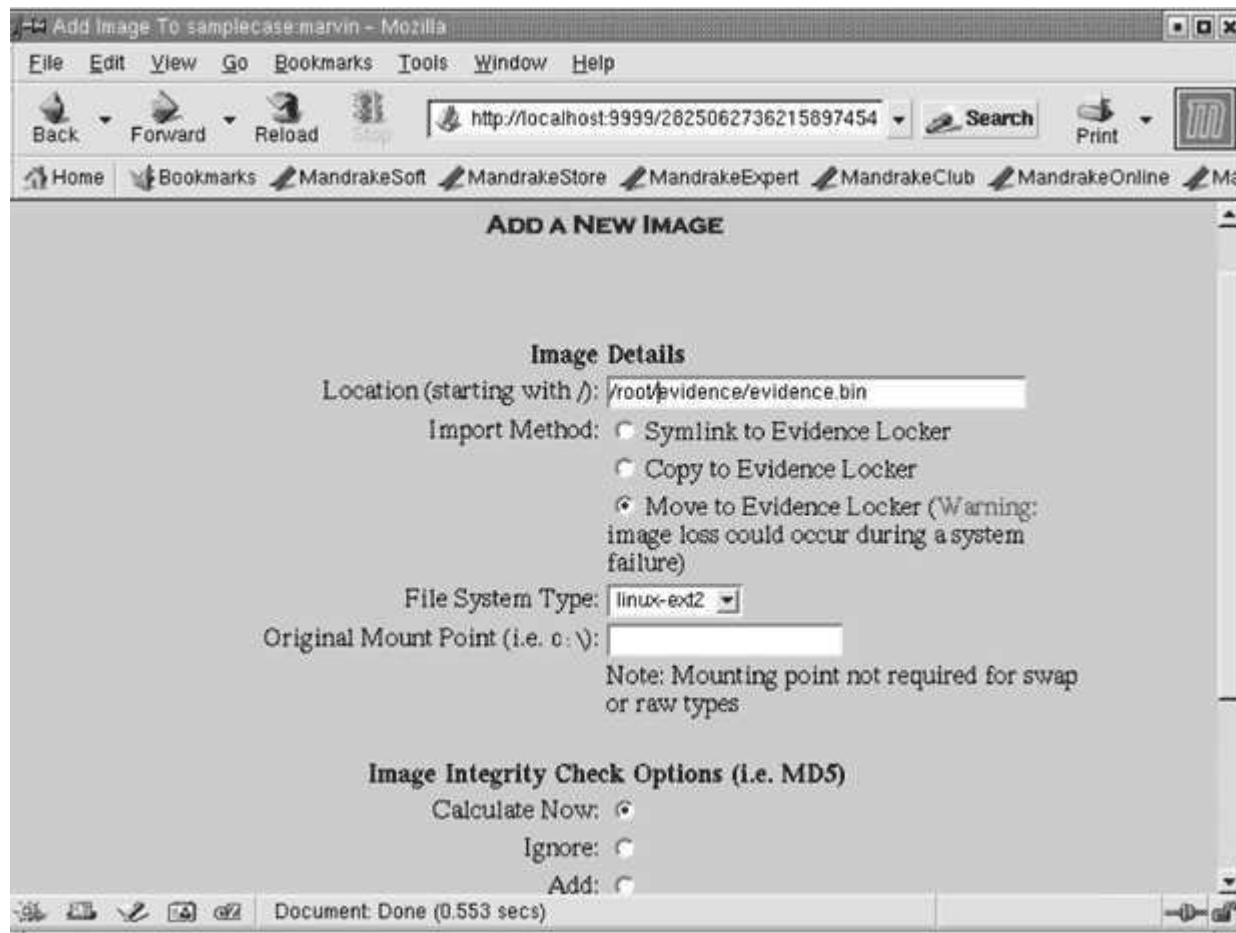


Рис. 11.4. Экран Add a New Image

3. Введите расположение и данные о файле образа. Можно скопировать файл в каталог для этого хоста в хранилище свидетельств или просто создать символическую ссылку на него. Будьте осторожны и не перемещайте файлы образов, особенно больших, слишком часто, так как это может привести к потере данных, если во время переноса возникнут проблемы.
4. Выберите тип файловой системы. Это определяет способ, которым Sleuth Kit смотрит на данные в образе.
5. Sleuth Kit автоматически создает файл хэша. Вы можете в любое время проверить соответствие хэша и данных в файле. Это значительно повышает легитимность ваших усилий в суде.
6. Для каждого хоста можно добавить несколько образов. Например, вам, возможно, понадобится разбить большой диск на несколько файлов образов. Щелкните мышью на Add Image, чтобы добавить образ и вернуться в основное окно Case Gallery.

Анализ данных

Теперь вы, наконец, готовы приступить к анализу. Может показаться, что работы по настройке слишком много, но вы оцените Sleuth Kit, когда вам придется манипулировать большим числом образов или понадобится быстро выдать определенный фрагмент данных. Перейдите в Image Gallery и щелкните мышью на образе, который хотите анализировать. В [табл. 11.5](#) перечислены типы анализа, который можно выполнять на образах данных.

Sleuth Kit в сочетании с Autopsy Forensic Browser - мощное средство организации и анализа судебных данных на уровне любой профессиональной лаборатории в стране. В этом разделе затронуты лишь некоторые основные функции, но об этом замечательном инструменте можно написать целые тома.

Здесь не рассмотрены многие команды и функции. Дополнительную информацию можно найти в оперативном руководстве и других ресурсах на веб-сайте. На сайте предлагается также ежемесячный бюллетень с интересными статьями и рекомендациями для интересующихся судебной информатикой.

Таблица 11.5. Типы проводимого в Sleuth Kit анализа

| Тип анализа | Описание |
|----------------|--|
| File Analysis | Показывает образ в виде файлов и каталогов, которые будет видеть файловая система. Здесь также видны файлы и папки, которые обычно могут быть скрыты операционной системой |
| Keyword Search | Позволяет искать во всем образе определенные ключевые слова. Это полезно, если вы ищете определенную программу или просто упоминание об определенной вещи. Юристы часто пользуются данной возможностью при поиске свидетельств инкриминируемой противозаконной деятельности на принадлежащем подозреваемому жестком диске. Это помогает довольно быстро найти иголку в стоге сена (рис. 11.5). |
| File Type | Сортирует все файлы или производит поиск по типу. Это удобно при поиске всех файлов определенного типа, например, JPEG или MP3 |
| Image Details | Выдает все детали изучаемого образа, которые могут пригодиться, например, при восстановлении данных, когда необходимо знать их физическое расположение |
| MetaData | Отображает низкоуровневые структуры каталогов и файлов в образе, полезные при поиске удаленного содержимого и просмотре других элементов, которые файловая система обычно не показывает |
| Data Unit | Позволяет углубиться в любой найденный файл и просмотреть его реальное содержимое в текстовом или шестнадцатеричном виде |

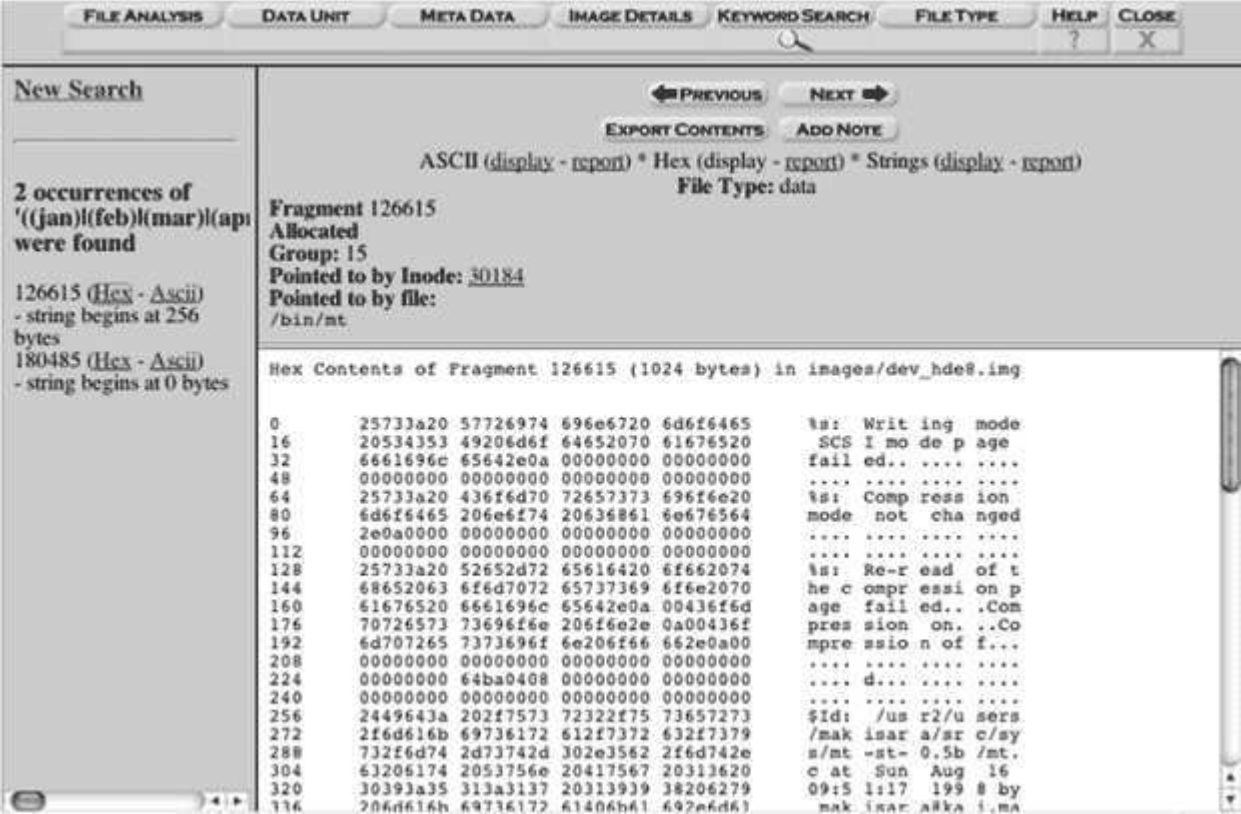


Рис. 11.5. Результаты поиска по ключевым словам

The Forensic Toolkit: Набор судебных средств для Windows

The Forensic Toolkit

Автор/основной контакт: Foundstone, Inc.

Web-сайт: <http://www.foundstone.com/index.htm?subnav=resources/navigation.htm&subcontent=/resources/freetools.htm>

Платформы: Windows NT, 2000, XP

Лицензии: Версия 1.4 - GPL, версия 2.0 - Freeware

Рассмотренные версии: 1.4 GPL, 2.0 Freeware

The Forensic Toolkit - еще одна замечательная свободная программа компании Foundstone. Этот набор средств полезен при осмотре файловых систем на платформе Windows и сборе информации для судебного расследования. Версия 1.4 программы имеет полностью открытые исходные тексты с лицензией GPL. Версия 2.0 условно свободна и может применяться для коммерческих целей, но имеет ограничения на внесение в программу дополнений и изменений, а ее исходные тексты в настоящее время недоступны.

Отметим, что эти средства работают только с файловыми системами NTFS. Если вы желаете исследовать раздел FAT32, вам придется воспользоваться другим инструментарием.

Установка The Forensic Toolkit

1. Загрузите соответствующий файл с Web-сайта (версию 1.4 или 2.0, в зависимости от того, нужны ли вам открытые исходные тексты).
2. Распакуйте файл в его собственный каталог. Это завершает установку.

Применение The Forensic Toolkit

Инструментарий включает различные утилиты командной строки, генерирующие статистические данные и информацию об исследуемой файловой системе. Чтобы выполнить команду, наберите ее в окне командной строки (вы должны находиться в соответствующем каталоге). В последующих разделах описаны отдельные средства.

Afind

Эта утилита ищет файлы по времени доступа к ним, не изменяя информацию о доступе, что отличает ее от обычных утилит Windows. Основной формат команды таков

```
afind каталог_поиска опции
```

Основные опции перечислены в [табл. 11.6](#).

Таблица 11.6. Основные опции поиска для Afind

| Опция | Описание |
|----------------|---|
| -f имя_файла | Выдает информацию о времени доступа к файлу с заданным именем |
| -s X | Отыскивает файлы, к которым обращались в течение последних X секунд |
| -m X | Отыскивает файлы, к которым обращались в течение последних X минут |
| -d X | Отыскивает файлы, к которым обращались в течение последних X дней |
| -a d/m/y-h:m:s | Отыскивает файлы, к которым обращались после указанной даты и времени |

Hfind

Это - средство поиска скрытых файлов в операционной системе Windows. Выдаются файлы, у которых установлен бит атрибута скрытости, а также файлы, скрытые с помощью специального атрибутного метода каталога/системы Windows NT. Формат таков:

```
hfind каталог_поиска
```

Команда выдает список скрытых файлов и дату и время последнего доступа к ним. Будьте осторожны при поиске по всему жесткому диску, так как на это может потребоваться много времени.

Sfind

Средство поиска на жестком диске скрытых потоков данных. Они отличаются от скрытых файлов, так как не становятся видны на жестком диске, когда вы щелкаете мышью на опции показа скрытых файлов. Скрытые потоки данных - особенность NTFS, предоставляющая определенным программам доступ к альтернативным потокам данных. Эти файлы связываются с видимым родительским файлом, но не удаляются, когда файловая система удаляет последний. Они могут применяться для сокрытия данных или вредоносного программного обеспечения. Формат команды `sfind` таков:

```
sfind каталог_поиска
```

Если вы ищете, отправляясь от корневого каталога большого диска, поиск может быть весьма длительным.

FileStat

Эта команда выдает полную распечатку атрибутов файла, включая информацию о безопасности. В каждый момент времени она работает только с одним файлом. Вывод можно направить по каналу в текстовый файл для дальнейшей обработки. Эта команда выдает довольно много информации, включая подробные сведения о файловом дескрипторе, которые обычно не сообщаются. На [листинге 11.5](#) показан пример этой информации для файла с именем test.txt.

```
Creation Time - 01/10/2004 03:18:40
Last Mod Time - 01/10/2004 03:18:40
Last Access Time - 01/10/2004 03:18:40
Main File Size - 11
File Attrib Mask - Arch
Dump complete:Dumping C:\temp\test.txt:
SD is valid.
SD is 188 bytes long.
SD revision is 1 ==SECURITY_DESCRIPTOR_REVISION1
SD's Owner is Not NULL
SD's Owner-Defaulted flag is FALSE
  SID = TONYVPRDESKTOP/Tony Howlett S-1-5-21--181663460
SD's Group-Defaulted flag is FALSE
  SID = TONYVPRDESKTOP/None S-1-5-21--181663460--953405037-
SD's DACL is Present
SD's DACL-Defaulted flag is FALSE
  ACL has 4 ACE(s), 112 bytes used, 0 bytes free
  ACL revision is 2 == ACL_REVISION2
  SID = BUILTIN/Administrators S-1-5-32-544
    ACE 0 is an ACCESS_ALLOWED_ACE_TYPE
    ACE 0 size = 24
    ACE 0 flags = 0x00
    ACE 0 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
```

```

SID = NT AUTHORITY/SYSTEM S-1-5-18
  ACE 1 is an ACCESS_ALLOWED_ACE_TYPE
  ACE 1 size = 20
  ACE 1 flags = 0x00
  ACE 1 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SID = TONYVPRDESKTOP/Tony Howlett S-1-5-21--181663460-
  ACE 2 is an ACCESS_ALLOWED_ACE_TYPE
  ACE 2 size = 36
  ACE 2 flags = 0x00
  ACE 2 mask = 0x001f01ff -R -W -X -D -DEL_CHILD -CHANGE_PERMS -TAKE_OWN
SID = BUILTIN/Users S-1-5-32-545
  ACE 3 is an ACCESS_ALLOWED_ACE_TYPE
  ACE 3 size = 24
  ACE 3 flags = 0x00
  ACE 3 mask = 0x001f01ff -R -X
SD's SACL is Not Present
Stream 1:
  Type: Security
  Stream name = ?? ??Size: 188

Stream 2:
  Type: Data
  Stream name = ?? ??Size: 11

Stream 3:
  Type: Unknown
  Stream name = ?? ??Size: 64

```

Листинг 11.5. Выдача команды FileStat

Hunt

Это средство можно применять для получения детальной информации о системе с помощью возможностей пустого сеанса Windows. При определенной степени вседозволенности в вашей системе может выдаваться важная информация, такая как списки пользователей, разделяемых ресурсов и запущенных служб. Команда имеет следующий формат:

```

hunt имя_исследуемого_хоста

```

На [листинге 11.6](#) приведен пример выдачи команды Hunt.

```

share = IPC$ - Remote IPC

```

```

share = print$ - Printer Drivers

```

```

share = SharedDocs -

```

```

share = Printer3 - Acrobat Distiller

```

```

share = Printer2 - Acrobat PDFWriter

```

```

User = Administrator, , , Built-in account for administrating the computer/domain

```

```
Admin is TONYVPRDESKTOP\Administrator
User = Howlett, , ,

User = Guest, , , Built-in account for guest access to the computer/domain

User = HelpAssistant, Remote Desktop Help Assistant Account,
      Account for Providing Remote Assistance

User = SUPPORT_388945a0, CN=Microsoft
      Corporation, L=Redmond, S=Washington, C=US, , This is a vendor's
      account for the Help and Support Service

User = Tony Howlett,
```

Листинг 11.6. Выдача команды hunt

В списке можно видеть двух пользователей, которые обычно не отображаются в разделе User Account системы Windows: HelpAssistant и SUPPORT (возможности удаленной помощи и раздражающая возможность "уведомите службу поддержки", выскакивающая всякий раз, когда программа отдает концы). Это пользователи системного уровня для внутренних программ. С помощью данного средства можно раскрыть других скрытых пользователей, спрятанных квалифицированным нарушителем.

Эта лекция не претендует на полноту описания всех возможных судебных средств, однако представленных средств достаточно, чтобы выполнять основные судебные действия практически на любой системе. Для тех, кто профессионально работает в данной области или оказался вовлеченным в расследование, имеется много других средств. Хороший список судебных средств с открытыми исходными текстами можно найти по адресу <http://www.opensourceforensics.org/>.

Инструменты безопасности с открытым исходным кодом

12. Лекция: Еще о программном обеспечении с открытыми исходными текстами: версия для печати и PDA

Теперь вы знаете, как сохранять данные в безопасности внутри и вне своей сети, как обнаруживать и расследовать атаки на ваши системы и сети. В этой книге были рассмотрены десятки защитных средств с открытыми исходными текстами, охватывающие практически все аспекты информационной безопасности. Однако мы всего лишь скользнули по поверхности того, что доступно. В каждой категории я пытался выбрать для демонстрации лучшие (по моему мнению) средства, но часто на выбор имеется масса других. Кроме того, существуют программные альтернативы с открытыми исходными текстами почти для любого типа приложений, включая текстовые процессоры, управление сетью, мультимедиа и т.д. Список можно продолжать и продолжать.

В этой, заключительной лекции представлены некоторые ресурсы для дальнейшего изучения средств безопасности с открытыми исходными текстами. Написано в ней и о том, как войти в сообщество открытого ПО.

Ресурсы открытого ПО

Если вы хотите продолжить исследование мира программного обеспечения с открытыми исходными текстами, посетите многочисленные доступные Интернет-ресурсы.

Телеконференции USENET

USENET является сетью серверов, предоставляющей место для обсуждения самых разных тем, таких как политика, развлечения и, конечно, компьютеры. Эти форумы называются телеконференциями, и они функционируют как разновидность общественных досок объявлений для людей, интересующихся определенной тематикой. Сеть USENET возникла как дискуссионная техническая группа, и по-прежнему имеется большое число групп, ориентированных на техническую тематику. Хотя спаммеры и web-форумы снизили действенность USENET, все еще остается ряд активных телеконференций USENET, связанных с открытым ПО.

Для доступа в USENET требуется специальная программа чтения новостей. В большинство современных web-навигаторов она встроена. В Internet Explorer в меню Tools выберите Mail and News, и затем выберите Read News. Для подписки также требуется действующий сервер телеконференций USENET. Поставщики Интернет-услуг обычно предоставляли этот сервис как часть своего стандартного предложения, и многие продолжают это делать. Если ваш поставщик этого не делает, то имеются общедоступные серверы USENET, к которым можно подключаться. Зайдите на сайт <http://www.newzbots.com/>, чтобы найти подобные серверы. После подписки на сервере можно заглянуть в несколько групп общего характера, которые могут представлять интерес. Есть также множество других групп, связанных с определенными операционными системами или программами:

- comp.sci.opensource
- comp.os.linux.advocacy
- comp.os.unix.bsd.freebsd.misc
- comp.os.unix.bsd.openbsd.misc

Стоит также посетить сайт Google Groups (щелкните мышью на Groups на сайте <http://www.google.com/>). Кроме доступа к текущим сообщениям и группам, там нашлось место для бывшего сайта Dejanews - архива USENET, начиная с 1992 года. Однако использование USENET постепенно сходит на нет и многие форумы перемещаются на Web-платформу или в модерируемые списки почтовой рассылки, чтобы уменьшить отношение шум/сигнал в корреспонденции.

Списки почтовой рассылки

Есть множество списков почтовой рассылки, связанных с открытым ПО. Большинство из них ориентированы на конкретные программы. Они служат для предоставления поддержки и сотрудничества в рамках проектов. Загляните на web-сайт или в документацию программы, чтобы узнать, имеются ли

посвященные ей списки почтовой рассылки, и как на них подписаться. Для средств, рассмотренных в этой книге, подобные списки существуют, они указаны в начале описания каждого средства. Имеется также несколько общих списков почтовой рассылки:

- Общее обсуждение Linux: <http://computers.rootsweb.com/>

Для подписки отправьте электронное сообщение по адресу

LINUX-L-request@COMPUTERS.rootsweb.com

поместив SUBSCRIBE в строке Subject.

- Архив почтовой рассылки BSD: <http://www.hu.freebsd.org/hu/arch/>

Web-сайты

Открытому ПО посвящена масса Web-сайтов. Есть также несколько хороших сайтов с общей информацией. Ниже представлены несколько замечательных сайтов, с которых можно начать знакомство с миром программного обеспечения с открытыми исходными текстами.

SourceForge

SourceForge (sourceforge.net) - превосходный Web-сайт, предоставляющий поддержку проектов открытого ПО и информацию о них (рис. 12.1). Его ведет Open Source Development Network, финансирующая сайт за счет рекламы и продаж своего программного обеспечения с открытыми исходными текстами. SourceForge содержит форум для обсуждения открытого ПО и множество ресурсов для проектов с открытыми исходными текстами. Если у вас есть многообещающая программа с открытыми исходными текстами, то SourceForge предоставит вам домашнюю страницу, форум, средства управления проектом, место хранения вашей программы для загрузки и много других ресурсов. Все это предоставляется бесплатно, хотя имеются некоторые ограничения, связанные с использованием ресурсов.



Рис. 12.1. Web-сайт SourceForge

Web-сайт SourceForge - отличное место для просмотра более 80000 каталогизированных проектов открытого ПО. Их можно искать по категории и платформе. Частично это, конечно, сырые идеи с минимальной поддержкой, но зато есть и тысячи полнофункциональных, проверенных временем программ. Можно включиться в любой проект, получить отклик или поддержку. SourceForge привлекает сотни тысяч пользователей и создателей самого современного программного обеспечения с открытыми исходными текстами. Если вы затеваете новый проект, то Web-сайт SourceForge - прекрасное место для поиска новобранцев.

Slashdot

Slashdot (<http://www.slashdot.org/>) - сайт с новостями обо всем, что происходит в мире открытого ПО. Он создан для крутых программистов, в основном пишущих открытый исходный код, и поддерживается их силами. Здесь можно узнать последние слухи и сплетни, сногсшибательные новости, а также познакомиться со всевозможными интересными статьями и мнениями. Частично это место общения, частично - источник горячих новостей и публикаций, насмешек и комментариев. На самом деле, в лексикон технарей даже вошло выражение "сайт послэшдотили": это когда сайт захлебывается от трафика после упоминания о нем на Slashdot.

Freshmeat

Freshmeat (<http://www.freshmeat.net/>) - серьезный сайт для обсуждения и разработки открытого ПО, некоторая комбинация Slashdot и SourceForge, но меньшего масштаба. Это может быть плюсом для тех, кто напуган размерами SourceForge и количеством его возможностей и ресурсов. Здесь также есть статьи и дискуссионные группы, равно как и непосредственные предложения загрузки многих проектов.

Open Source Initiative

Open Source Initiative (<http://www.opensource.org/>) - организация, занимающаяся продвижением и уточнением концепции разработки программного обеспечения с открытыми исходными текстами. Она предлагает формальное определение того, из чего должно состоять открытое ПО, и сертификацию такого статуса, хотя многие утверждают, что это - движущаяся цель, и открытое ПО по определению есть нечто постоянно изменяющееся и неопределяемое. До сих пор лишь небольшое число программ получили печать их одобрения, но зато это одни из наиболее известных, такие как web-сервер Apache и программа Sendmail. На мой взгляд, это движение в правильном направлении для будущего открытого ПО: только когда мир открытого ПО самоорганизуется и согласится с некоторыми стандартами, он получит значительную поддержку в корпоративной Америке. Стандартизация способствует признанию.

Free Software Foundation

Сайт Фонда открытого программного обеспечения (<http://www.fsf.org/>) - база одного из двух основных лагерей мира открытого ПО. FSF поддерживает проект GNU, а также его официальные программные продукты. Здесь можно найти лицензию GPL и узнать все о том, как она работает. Отстаиваемую Фондом точку зрения, состоящую в том, что все программное обеспечение должно быть свободным, некоторые могут счесть слишком радикальной, но здесь действительно заложили основы большей части доступного на сегодняшний день программного обеспечения с открытыми исходными текстами.

Есть множество других сайтов, посвященных открытому ПО, и все время создаются новые. Наберите в любой поисковой машине запрос "open source security" или "open source software", и вы легко в этом убедитесь.

Присоединение к движению за открытое ПО

Если вы с выгодой для себя воспользовались средствами безопасности с открытыми исходными текстами из этой книги, у вас может возникнуть желание расширить свое участие в этой деятельности. В большинстве случаев программное обеспечение свободно, и вы не обязаны что-либо делать в благодарность за полученную выгоду. Однако в создание и поддержку примененного вами программного обеспечения абсолютно добровольно вложена масса времени и сил. Открытое ПО продолжает работать и развиваться лишь за счет коллективных усилий. Для некоторых, особенно для сотрудников коммерческих программистских концернов, это звучит как-то по-социалистически, но по сути мало чем отличается от деятельности ассоциаций родителей и учителей или мелких бейсбольных лиг. Разработчики - вот кто делает мир открытого ПО прекрасным.

Участвуя в этом движении, вы не только поддержите развитие открытого ПО, но и заведете новых друзей с близкими вам интересами, установите ценные деловые контакты в своей области и по ходу дела многое узнаете об управлении проектами, совместной работе и, конечно, приобретете технических знания и опыт.

Чтобы внести свой вклад, не обязательно быть гуру в программировании. Ключевым моментом в содействии преуспеванию движения за открытое ПО является именно участие, которое может принимать различные формы - от выделения нескольких часов личного времени до превращения этого занятия во вторую работу.

Поиск ошибок/бета-тестирование

Даже если вы - простой пользователь и не интересуетесь программированием, вы можете помочь развитию средств безопасности с открытыми исходными текстами. В большинстве зрелых проектов есть списки почтовой рассылки для отслеживания ошибок, а в некоторых - более сложные системы для сообщений о проблемах. Если вы работаете с программой и обнаружите что-то, работающее неправильно, сообщите об этом и проверьте, можно ли это исправить. В процессе исправления вы поможете разработчикам найти ошибки и улучшить программу. Конечно, вы должны будете убедиться, что ваша проблема вызвана ошибкой в программе, а не вашим промахом при установке, но народ из списка рассылки с превеликим удовольствием наставит вас на путь истинный.

Чтобы надлежащим образом сообщить об ошибке, проверьте, что вы собрали значения всех переменных окружения, и попытайтесь воспроизвести ситуацию, чтобы точно определить, при каких условиях возникает ошибка. Такие вещи, как операционная система, версия программы, настройки, оборудование и т.д., важны. Не забудьте также предоставить разработчикам для анализа все сообщения об ошибках, файлы журналов или образ памяти.

Можно заняться и бета-тестированием самых свежих версий. В некоторых проектах предлагаются возможности выполнять "стабильный" или "экспериментальный" код. Хотя большинство пользователей предпочтут стабильный код, вы можете стать пионером и проверять экспериментальные или бета-версии. Помните, что при использовании этих программ могут возникать неожиданные проблемы (например, иногда новый код будет разрушать вещи, до того работавшие). Если вы собираетесь применять бета-код, то, наверное, лучше запускать его сначала на тестовой машине, а уж потом переносить на производственную.

В других проектах бета-код могут распространять ограниченному списку проверяющих, чтобы первыми пользователями кода были люди опытные, сознающие, что они используют бета-версию. Таким образом можно избежать обычных ошибок новичков, и иметь дело с пользователями, которые понимают, как работает программа, и могут точно описать свои проблемы. Поэтому вам, вероятно, не стоит предлагать себя в качестве бета-пользователя, пока вы не получите некоторый опыт работы с этим программным обеспечением. Когда вы будете готовы, попросите ключевых разработчиков внести вас в список. В этом случае вы сможете помочь улучшить программное обеспечение для будущих пользователей. В качестве дивидендов вы получите первенство в доступе к новейшим возможностям и понимание того, как дальше должен развиваться проект.

Участие в дискуссионных группах и поддержка других пользователей

В большинстве проектов с открытыми исходными текстами есть список почтовой рассылки для обсуждений и технических вопросов. Вы должны подписаться на него, даже если пока не планируете участвовать в его работе. Не обязательно часто писать сообщения, чтобы получить некоторую выгоду. Вполне допустимо просто наблюдать и читать публикуемые вопросы и ответы. Я многое узнал о программном обеспечении, со стороны наблюдая за дискуссиями в списке почтовой рассылки. Однако, одно предостережение: некоторые из подобных списков весьма активны, в них ежедневно поступают десятки сообщений. Такой объем информации может оказаться избыточным, особенно если вы и так перегружены, как большинство системных администраторов. Но даже чтение только случайных, заинтересовавших вас сообщений принесет определенную пользу. Если вы чувствуете, что получаете слишком много электронных писем, попробуйте подписаться на дайджест-версию списка - ежедневные или еженедельные сообщения, содержащие компиляцию всей корреспонденции. В этом случае вы получите только одно сообщение и сможете просмотреть его, когда найдется время. Убедитесь также, что вы знаете, как отменить подписку, прежде чем подписываться, чтобы можно было быстро отказаться от нее, если объем окажется для вас чрезмерным.

В большинстве списков почтовой рассылки при проектах открытого ПО в качестве средства управления применяется программный пакет Major Domo (это также проект с открытыми исходными текстами!). Стандартными способами для подписки и отказа от подписки в такой системе служат:

- Подписка: отправьте сообщение на адрес менеджера списка (обычно публикуемый на web-сайте) со словом "Subscribe" в теме и теле письма. Вы можете получить запрос с просьбой подтвердить свое желание быть в списке. После вашего ответа вы начнете получать сообщения.
- Отказ от подписки: отправьте сообщение на адрес менеджера списка, поместив слово "Unsubscribe" в теме и теле письма.

Списки почтовой рассылки бывают модерлируемыми и немодерлируемыми. В последнем случае все могут посылать сообщения, которые тут же будут появляться в списке. Это лучший вид списка для быстрого получения информации. Однако многие немодерлируемые списки быстро засоряются посторонними сообщениями, спорами и просто взаимной руганью. Именно поэтому большинство списков теперь модерлируются. Это означает, что человек - модератор списка - должен просматривать каждое сообщение, решать, имеет ли оно отношение к тематике списка, и одобрять его публикацию. В результате значительно уменьшается общий объем рассылаемых сообщений, которые теперь заведомо "в теме", однако ваше обращение за помощью может задерживаться на несколько дней, пока модератор до него доберется. Кроме того, модераторы обычно приостанавливают функционирование списка на время отпуска (они ведь его заслужили), поэтому получение ответов в дни отдыха может быть нерегулярным.

Когда вы поймете, что можете участвовать на равных, начинайте посылать сообщения, отвечать на некоторые простые вопросы и высказывать свое мнение. Это снимет часть нагрузки с технически более подготовленных разработчиков и расширит базу знаний всего проекта. В конце концов, вы можете иметь опыт применения определенной конфигурации или платформы, которого не имеет никто другой (возможно, вы действуете в необычной среде), или у вас может быть особое мнение по определенному вопросу. Вполне возможно, что кто-то воспользуется вашей помощью. Вам будет приятно помочь другим, и вас поразит, насколько благодарными и любезными будут люди, которым вы помогли. Если бы ваши внутренние пользователи могли быть столь же приятными и признательными!

Предоставление ресурсов для проекта

Есть нечто, что вы можете сделать даже при отсутствии способностей к программированию и опыта работы с программным обеспечением. Проекты с открытыми исходными текстами обычно лишены каких-либо поступлений для покрытия расходов на разработку и сопровождение программного обеспечения. Хотя большая часть работы выполняется добровольцами, остаются вопросы, связанные с размещением web-сайта проекта, с его аппаратным обеспечением, и многие другие. Опять-таки, обычно большую часть ресурсов безвозмездно предоставляют сами участники проекта. Если у вас есть старая машина, которую можно использовать как web-сервер, дайте знать ключевым разработчикам. Вы будете удивлены, узнав, что может делать старая машина, выполняя Linux и Apache. Если ваша организация согласится, подумайте, не могли бы вы предложить разместить web-сайт проекта в своей коммуникационной инфраструктуре. Если проект большой, ваша организация едва ли захочет с ним связываться, но для небольших проектов использование полосы пропускания, скорее всего, будет минимальным и по большей части приходящимся на нерабочее время. Если вы обладаете навыками web-дизайна, предложите создать web-сайт. Если ваш поставщик Интернет-услуг предоставляет бесплатное пространство для web-сайтов, предложите использовать его для проекта. Некоммерческие начинания обычно подпадают под условия обслуживания персонального web-пространства. В конце концов, некоторые пакеты открытого ПО принимают даже старые добрые зеленые баксы как "пожертвования" за использование программного обеспечения. Вы можете раскрутить вашу организацию на некоторое количество баксов в качестве альтернативы оплаты коммерческого ПО. В принципе, все, что придет вам в голову, может пригодиться для проекта открытого ПО. Навыки графического дизайна для создания логотипа, счета электронной почты для поддержки списков рассылки, юридическая помощь при шлифовке лицензий - все это конструктивные способы помочь вашему любимому проекту с открытыми исходными текстами.

Станьте постоянным клиентом организаций, использующих или поддерживающих открытое ПО

Хотя вы не обязаны расходовать средства из вашего бюджета на программное обеспечение, вы все-таки можете потратить деньги на другие вещи. При покупке оборудования, программного обеспечения или услуг возьмите за правило в первую очередь рассматривать производителей, использующих или поддерживающих открытое ПО. В конце концов, если компания может быть коммерчески жизнеспособной, применяя открытое ПО как ключевой компонент своих предложений, это только способствует делу. Такие компании, как Sun, IBM и Dell активно продвигают программное обеспечение с открытыми исходными текстами.

Еще о защитных средствах с открытыми исходными текстами

Теперь вы должны иметь общее представление о концепциях информационной безопасности и о том, как применять их в своей организации с помощью защитных средств с открытыми исходными текстами. Используя эти программы и информацию из настоящей книги, вы сможете значительно лучше

защитить свои системы и сети от угроз компьютерной преступности. Мы рассмотрели программы, повышающие конфиденциальность, целостность и доступность ваших сетей, систем и данных по цене, которая уложится в любой бюджет.

Надеюсь, вы понимаете, что подлинная информационная безопасность не сводится только к программам и технологии. Для нее также важны процессы и люди. Только сочетание надлежащих людей, процессов и технологии способно по-настоящему обезопасить вашу сеть. Защитные средства с открытыми исходными текстами могут дать вам лучшее программное обеспечение для построения надежного фундамента информационной безопасности.

Движение за открытое ПО ширится день ото дня, становится все более заметным и легитимным. Надеюсь, что эта книга будет способствовать вашему приобщению к делу создания качественных защитных средств на основе программного обеспечения с открытыми исходными текстами, что вы внесете в него свой вклад. Это доставит вам массу удовольствия, вы многому научитесь, вам будет приятно сознавать, что благодаря вашим усилиям Интернет и сети стали безопаснее. Возможно, в будущей редакции этой книги важнейшее место будет отведено написанному вами защитному средству с открытыми исходными текстами.