Бирюков А. А.

Информационная безопасность:
защита и нападение

ДМК Пресс
издательство

УДК 004.065
ББК 32.973.26-018.2
Б59

Бирюков А. А.
Б59 Информационная безопасность: защита и нападение. – М.: ДМК Пресс, 2012. – 474 с.: ил.

ISBN 978-5-94074-647-8

В литературе по информационной безопасности (ИБ) в настоящее время не наблюдается недостатка. Однако в большинстве книг на эту тему приводится лишь малая часть тех сведений, которые необходимы для комплексного обеспечения информационной безопасности на предприятии. Например, в некоторых учебниках по ИБ основное внимание уделяется нормативным актам из области ИБ, но при этом крайне мало говорится о технической реализации угроз и защите от них.

С другой стороны, существует много книг, посвященных только техническим аспектам (так называемый взгляд «глазами хакера»). В этих книгах подробно описывается реализация конкретных защит, но не всегда понятно, в каких практических ситуациях она может пригодиться.

Данная книга представляет собой попытку преодолеть односторонний подход к теме ИБ. Книга предназначена для системных администраторов и пользователей малых и средних сетей, осуществляющих защиту корпоративных ресурсов. Здесь приводятся как техническая информация, описывающая атаки и защиту от них, так и рекомендации по обеспечению информационной безопасности с соответствующими примерами.

УДК 004.065
ББК 32.973.26-018.2

ISBN 978-5-94074-647-8

ОГЛАВЛЕНИЕ

ГЛАВА 1

Теоретические основы

В любой организации, независимо от ее размеров, всегда есть корпоративная сеть. Даже если у вас маленькая контора, в которой всего два или три компьютера, они все равно должны быть объединены в сеть и иметь доступ в Интернет. Таковы реалии современного бизнеса, всем нужен доступ к электронной почте, всем нужен доступ к информации во Всемирной информационной паутине. Однако локальные сети бывают не только в организациях. Зачастую во многих квартирах имеется по нескольку компьютеров, и каждому из них тоже необходим доступ к ресурсам Интернета. Например, у многих пользователей дома есть основной компьютер, ноутбук, карманный компьютер или коммуникатор. Всем этим устройствам в той или иной степени нужно обмениваться файлами между собой, иметь доступ в Интернет. Для организации такого доступа используют активное сетевое оборудование: маршрутизаторы, межсетевые экраны, коммутаторы, беспроводные точки доступа и концентраторы. Хотя последние встречаются все реже. Вообще, сейчас, как правило, для доступа домашних пользователей в Интернет используют устройства, сделанные по принципу «все в одном». То есть одно устройство объединяет в себе функции межсетевого экрана, простейшего маршрутизатора, коммутатора и точки беспроводного доступа. Для домашних пользователей подобное устройство является наилучшим решением, так как одна «коробка» занимает меньше места, к ней нужно вести меньше проводов, кроме того, ее легче настраивать. В корпоративных сетях, где присутствуют более 20 рабочих станций, такие решения стараются не использовать, так как при одновременном подключении большого количества рабочих станций у многофункциональных

сетевых устройств резко снижается производительность. Кроме того, в случае выхода из строя такого устройства вы лишитесь как доступа в Интернет, так и доступа во внутреннюю локальную сеть. Так что, господа системные администраторы, если ваш дешевый Dlink прекрасно работает в домашней сети, то не торопитесь советовать руководству покупать такой же дешевый Dlink для корпоративной сети. Решать проблемы, которые потом возникнут, придется прежде всего вам.

Но вернемся к вопросам сетевой безопасности. Любая локальная сеть немыслима без сетевого оборудования. А против сетевых устройств существует масса различных атак, направленных на перехват информации, проходящей по сети, захват управления устройством или временный вывод его из строя.

У читателя может возникнуть вопрос: почему, говоря о сети, я говорю только о сетевом оборудовании, ведь в сети также работает множество приложений, например серверы баз данных или электронная почта? Отвечу так: несомненно, в сети работает множество различных приложений, но в рамках обсуждения сетевой безопасности мы обсудим работу именно сетевого оборудования, так как работу приложений мы будем рассматривать в главе «Атаки на уровне приложений».

Однако, прежде чем начать обсуждение способов осуществления этих атак и средств защиты, необходимо вспомнить (я надеюсь) основы сетевых технологий, иначе материал последующих разделов может превратиться для читателя в набор непонятных терминов. Конечно, если вы можете с лету вспомнить модель OSI, знаете, что такое Spanning Tree Protocol или PVLAN, то можете смело переходить к чтению следующих разделов.

1.1. Модель OSI

При осуществлении передачи данных от компьютера к компьютеру в сети производится множество операций. При этом пользователя совершенно не интересует, как именно это происходит, – ему необходим доступ к приложению или компьютерному ресурсу, расположенному в другом компьютере сети. На самом деле вся передаваемая информация проходит много этапов обработки. Прежде всего она разбивается на блоки, каждый из которых снабжается управляющей информацией. Получившиеся в результате блоки оформляются в виде сетевых пакетов, затем эти пакеты кодируются, передаются с помощью электрических или световых сигналов по сети в соответствии с выбранным методом доступа, далее из принятых пакетов

вновь восстанавливаются заключенные в них блоки данных, блоки соединяются в данные, которые и становятся доступны другому приложению. Приведенное здесь описание является упрощенным пояснением происходящих процессов. Часть из указанных процедур реализуется только программно, другая часть – аппаратно, а какие-то операции могут выполняться как программами, так и аппаратурой. Упорядочить все выполняемые процедуры, разделить их на уровни и подуровни, взаимодействующие между собой, как раз и призваны модели сетей. Модели сетей позволяют правильно организовать взаимодействие как абонентам внутри одной сети, так и самым разным сетям на различных уровнях. В настоящее время наибольшее распространение получила так называемая эталонная модель обмена информацией открытой системы OSI (Open System Interchange). Под термином «открытая система» понимается не замкнутая в себе система, имеющая возможность взаимодействия с какими-то другими системами (в отличие от закрытой системы).

Обращаясь к истории создания иерархической модели, скажу, что модель OSI была предложена Международной организацией стандартов ISO (International Standards Organization) в 1984 году. С тех пор ее используют (более или менее строго) все производители сетевых продуктов. Модель OSI не лишена ряда недостатков, присущих универсальным моделям, а именно она громоздка, избыточна и не слишком гибка. В результате реальные сетевые средства, предлагаемые различными фирмами, не обязательно придерживаются принятого разделения функций, то есть возможны устройства, сочетающие в себе функционал различных уровней. Однако знакомство с моделью OSI позволяет лучше понять, что же происходит в сети и, соответственно, как лучше ее защищать. Все сетевые функции в модели разделены на 7 уровней. При этом вышестоящие уровни выполняют более сложные, глобальные задачи, для чего используют в своих целях нижестоящие уровни, а также управляют ими. Цель нижестоящего уровня – предоставление услуг вышестоящему уровню, причем вышестоящему уровню не важны детали выполнения этих услуг. Нижестоящие уровни выполняют более простые и конкретные функции. В идеале каждый уровень взаимодействует только с теми, которые находятся рядом с ним (выше и ниже него). Верхний уровень соответствует прикладной задаче, работающему в данный момент приложению, например веб-браузеру, нижний – непосредственной передаче сигналов по каналу связи.

Данные, которые следует передать по сети, на пути от верхнего (седьмого) уровня приложений до нижнего (первого) физического

проходят процесс инкапсуляции, то есть каждый нижеследующий уровень не только производит обработку данных, приходящих с более высокого уровня, но и снабжает их своим заголовком, а также добавляет к нему служебную информацию. Такой процесс обрастания служебной информацией продолжается до последнего (физического) уровня. На физическом уровне вся эта многооболочечная конструкция передается по кабелю приемнику. Там происходит обратный процесс – декапсуляция, то есть при передаче на вышестоящий уровень убирается одна из оболочек. Верхнего, седьмого уровня достигают уже данные, освобожденные от всех оболочек, то есть от всей служебной информации нижестоящих уровней. При этом каждый уровень принимающего абонента производит обработку данных, полученных с нижеследующего уровня, в соответствии с убираемой им служебной информацией.

В тех случаях, когда на пути между абонентами в сети включаются некие промежуточные устройства (например, концентраторы, коммутаторы, маршрутизаторы), то и они тоже могут выполнять функции, входящие в нижние уровни модели OSI. Чем больше сложность промежуточного устройства, тем больше уровней оно захватывает. В случае если между получателем и отправителем присутствует межсетевой экран, будут обработаны все семь уровней иерархической модели. Но любое промежуточное устройство должно принимать и возвращать информацию на нижнем, физическом уровне. Все внутренние преобразования данных должны производиться дважды и в противоположных направлениях. Промежуточные сетевые устройства, в отличие от полноценных абонентов (например, компьютеров), работают только на нижних уровнях и к тому же выполняют двустороннее преобразование.

Теперь поговорим подробнее о функциях разных уровней.

1.1.1. Прикладной (7) уровень (Application Layer)

Это уровень приложений, который обеспечивает услуги, непосредственно поддерживающие приложения пользователя. Примером таких приложений являются: программные средства передачи файлов (FTP), доступа к базам данных (клиенты баз дланных), средства электронной почты (Microsoft Outlook), служба регистрации на сервере (RADIUS). Этот уровень фактически управляет всеми остальными шестью уровнями. Примером может являться работа с таблицами Excel, когда пользователь сохраняет файл на сетевой ресурс. В этом случае прикладной уровень обеспечивает перемещение файла с рабочего компьютера на сетевой диск прозрачно для пользователя.

1.1.2. Представительский (6) уровень (Presentation Layer)

Это уровень представления данных, который определяет и преобразует форматы данных и их синтаксис в форму, удобную для сети, то есть выполняет функцию переводчика. Здесь же производятся шифрование и дешифрирование данных, а при необходимости и их сжатие. Стандартные форматы существуют для текстовых файлов (ASCII, HTML), звуковых файлов (MPEG, WAV), рисунков (JPEG, GIF, TIFF), видео (AVI). Все преобразования форматов делаются на представительском уровне. Если данные передаются в виде двоичного кода, то преобразования формата не требуется.

1.1.3. Сеансовый (5) уровень (Session Layer)

На этом уровне производится управление проведением сеансов связи (то есть осуществляются установка, поддержка и прекращение связи). Данный уровень предусматривает три режима установки сеансов: симплексный (передача данных в одном направлении), полудуплексный (передача данных поочередно в двух направлениях) и полнодуплексный (передача данных одновременно в двух направлениях). Сеансовый уровень может также вставлять в поток данных специальные контрольные точки, которые позволяют контролировать процесс передачи при разрыве связи. Этот же уровень распознает логические имена абонентов, контролирует предоставленные им права доступа.

1.1.4. Транспортный (4) уровень (Transport Layer)

Данный уровень обеспечивает доставку пакетов без ошибок и потерь, а также в нужной последовательности. На нем же производятся разбивка передаваемых данных на блоки, помещаемые в пакеты, и восстановление принимаемых данных из пакетов. Доставка пакетов возможна как с установлением соединения (виртуального канала), так и без. Транспортный уровень является пограничным и связующим между верхними тремя, сильно зависящими от приложений, и тремя нижними уровнями, сильно привязанными к конкретной сети.

1.1.5. Сетевой (3) уровень (Network Layer)

Производит адресацию пакетов и перевод логических имен (логических адресов, например IP-адресов) в физические сетевые MAC-

адреса (и обратно). На этом же уровне решается задача выбора маршрута (пути), по которому пакет доставляется по назначению (если в сети имеются несколько маршрутов). На сетевом уровне действуют такие сложные промежуточные сетевые устройства, как маршрутизаторы.

1.1.6. Канальный (2) уровень (Data Link Layer)

Другое название – уровень управления каналом передачи, отвечает за формирование пакетов (кадров) стандартного для данной сети (например, Ethernet) вида, включающих начальное и конечное управляющие поля. Здесь же производится управление доступом к сети, обнаруживаются ошибки передачи путем подсчета контрольных сумм и производится повторная пересылка приемнику ошибочных пакетов. Канальный уровень делится на два подуровня: верхний LLC и нижний MAC. Верхний подуровень (LLC – Logical Link Control) осуществляет управление логической связью, то есть устанавливает виртуальный канал связи. Строго говоря, эти функции не связаны с конкретным типом сети, но часть из них все же возлагается на аппаратуру сети (сетевой адаптер). Другая часть функций подуровня LLC выполняется программой драйвера сетевого адаптера. Подуровень LLC отвечает за взаимодействие с уровнем 3 (сетевым). Нижний подуровень (MAC – Media Access Control) обеспечивает непосредственный доступ к среде передачи информации (каналу связи). Он напрямую связан с аппаратурой сети. Именно на подуровне MAC осуществляется взаимодействие с физическим уровнем. Здесь производятся контроль состояния сети, повторная передача пакетов заданное число раз при коллизиях, прием пакетов и проверка правильности передачи. На канальном уровне работают такие промежуточные сетевые устройства, как, например, коммутаторы.

1.1.7. Физический (1) уровень (Physical Layer)

Это самый нижний уровень модели, который отвечает за кодирование передаваемой информации в уровни сигналов, принятые в используемой среде передачи, и обратное декодирование. Здесь же определяются требования к соединителям, разъемам, электрическому согласованию, заземлению, защите от помех и т. д. На физическом уровне работают такие сетевые устройства, как концентраторы (рис. 2).

Для того чтобы читателю стал более понятен приведенный выше материал, приведу несколько простых примеров. Что происходит,

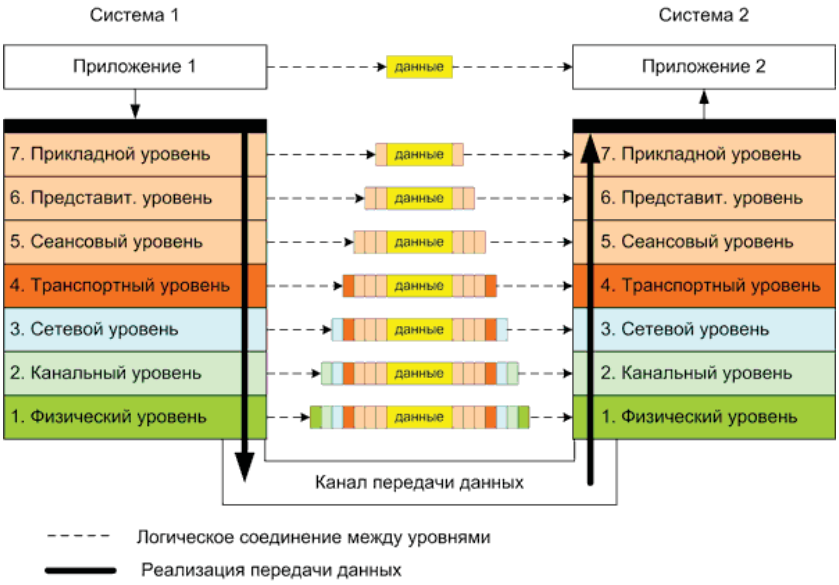

Рис. 2. *Схема пакета для различных уровней OSI*

когда вы запрашиваете какие-либо данные по сети, например HTML-страницу? Ваш веб-браузер (уровень приложений) формирует запрос по протоколу HTTP (уровень представлений и сеансовый уровень), формируются пакеты, передаваемые на порт 80 (транспортный уровень), на IP-адрес веб-сервера (сетевой уровень). Эти пакеты передаются сетевой карте вашего компьютера, которая передает их в сеть (канальный и физический уровни). По пути следования пакеты проходят через различные промежуточные устройства: коммутаторы, маршрутизаторы, межсетевые экраны. Каждое из этих устройств может осуществлять проверку пакета в соответствии со своими настройками. Например, в зависимости от IP-адреса назначения маршрутизатор перешлет пакеты в определенную сеть. А межсетевой экран разрешит или запретит передачу данных пакетов. Когда пакеты достигнут узла назначения, будет произведено обратное преобразование. Из пакетов будет извлечена информация, соответствующая каждому из уровней иерархической модели.

Говоря о том, на каком уровне данной модели работают различные устройства, хотелось бы отдельно сказать о таких устройствах безопасности, как межсетевые экраны. В общем случае, межсетевой экран работает на уровне приложений. То есть он разбирает прохо-

дящий через него пакет, выделяя из него атрибуты каждого из уровней модели и проверяя их на соответствие корпоративной политике безопасности. Выполняемые при этом действия будут выглядеть так:

- проверка IP-адреса отправителя и получателя (сетевой уровень);
- проверка порта, на который передается пакет (транспортный уровень);
- проверка соответствия сеансовым уровням и уровням представления;
- проверка, соответствует ли содержимое пакета структуре данных того протокола, который разрешен на данном порту (уровень приложений).

Например, если вы попытаетесь под видом DNS-пакета передать, скажем, HTTP-пакет (осуществить туннелирование, спрятать HTTP в DNS), то межсетевой экран выполнит алгоритм, приведенный выше. Очевидно, что IP-адрес, порт и проверка сеансового уровня будут пройдены успешно. А вот дальше в зависимости от конкретной политики межсетевого экрана на уровне представлений или на уровне приложений будет обнаружено, что в нашем DNS-пакете на самом деле находятся данные, не соответствующее структуре пакетов для данного протокола. И такой пакет должен быть заблокирован.

Но не все межсетевые экраны разбирают пакет до уровня приложений, многие дешевые модели ограничиваются проверкой данных сетевого и транспортного уровней, что не всегда безопасно.

1.1.8. Заключение

Итак, мы разобрались с устройствами, выяснили, как организовано взаимодействие между ними. Теперь поговорим о том, какие возможны атаки на сетевые устройства, работающие на определенном уровне модели OSI.

ГЛАВА 2

Классификация атак по уровням иерархической модели OSI

Рассмотрев теоретические основы уровней модели OSI, теперь перейдем к рассмотрению тех атак, которые могут быть реализованы против устройств и приложений, работающих на каждом из уровней OSI.

2.1. Атаки на физическом уровне

Атаки на физическом уровне известны давно, и, казалось бы, все знают, как с ними бороться, однако иногда и с помощью такой атаки можно получить конфиденциальную информацию.

2.1.1. Концентраторы

Начнем с простейших сетевых устройств – концентраторов (hub). Как известно, концентратор, получая пакет на один из своих портов, ретранслирует его на все остальные (рис. 3).

При этом все машины, подключенные к данному концентратору, получают отправленный пакет. При использовании этих устройств существенно снижается пропускная способность сегмента сети, и, что гораздо важнее с точки зрения информационной безопасности, любой подключенный к концентратору пользователь может без труда прослушать весь трафик, проходящий через данный сегмент.

Для того чтобы проверить это на практике, достаточно подключить к концентратору несколько машин и на одной из них запустить загрузочный диск и описанный в приложениях дистрибутив BackTrack или любой другой дистрибутив Линукс, содержащий данную утилиту.

Рис. 3. *Фото концентратора*

Здесь и далее все утилиты, приведенные в примерах, входят в состав Backtrack, если другое не указано явно.

После загрузки операционной системы Linux BackTrack необходимо в разделе BackTrack выбрать последовательно: **Privikege Escalation ⇒ Sniffers ⇒ Wirehark** (рис. 4).

Далее в открывшемся окне приложения нужно указать интерфейс, с которого будет перехватываться трафик, и нажать **Start** (рис. 5).

Ну а теперь самое интересное. Заходим с машины, подключенной к этому же концентратору, на какой-либо сайт, передающий учетные данные через веб-форму в незашифрованном виде, и вводим логин и пароль (лучше не настоящие). Отправляем эти данные на сервер и получаем сообщение о неверных учетных данных.

Затем переходим в окно **Wireshark** и просматриваем HTTP-трафик, который передавался с локальной машины на веб-сервер (рис. 6).

В своем примере я использовал веб-портал одной известной социальной сети, который до сих пор при передаче учетных данных не использует шифрование и передает логин и пароль в незашифрованном виде. В моем случае логин был test, а пароль p@ssw0rd. В логе символ @ заменен шестнадцатеричным кодом UTF-8.

Конечно, многие веб-сайты используют шифрование при передаче учетных данных, но тем не менее с помощью прослушивания трафика опытный хакер может собрать массу полезной информации. Например, по тем серверам, к которым обращается компьютер, можно

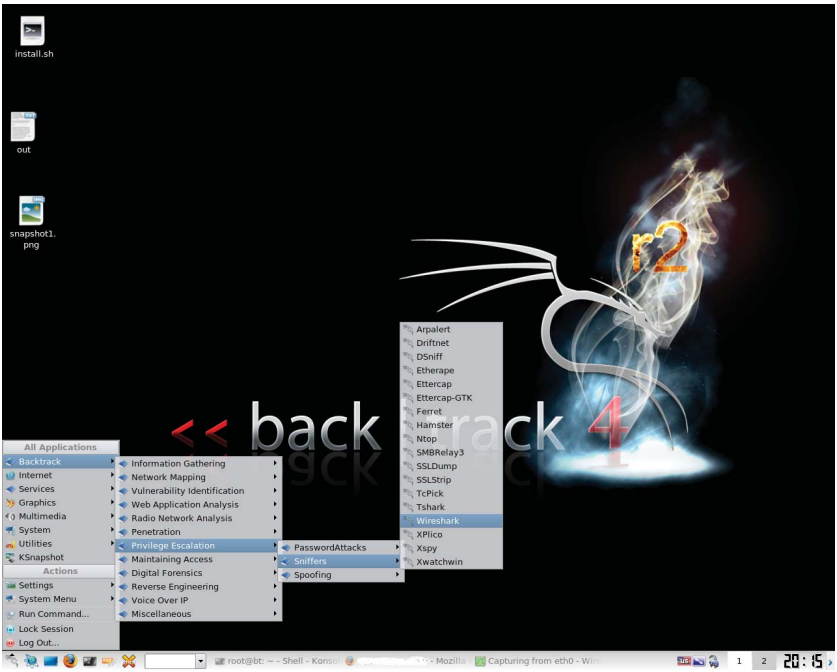

Рис. 4. *Запуск сниффера Wireshark в BackTrack*

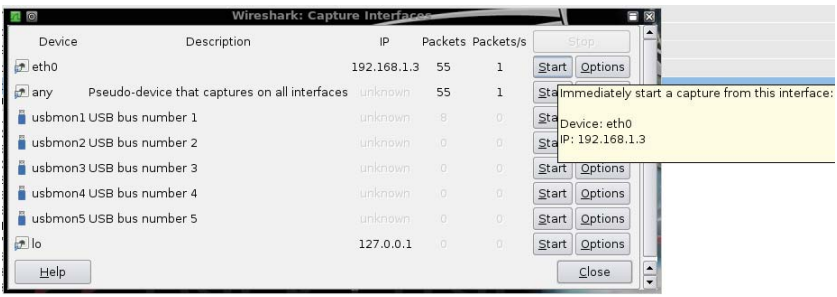

Рис. 5. *Запуск прослушивания интерфейса в Wireshark в BackTrack*

попытаться определить операционную систему, установленную на компьютере, вплоть до версий установленных обновлений. Также можно узнать об установленных приложениях. Такие атаки называются «passive fingerprint».

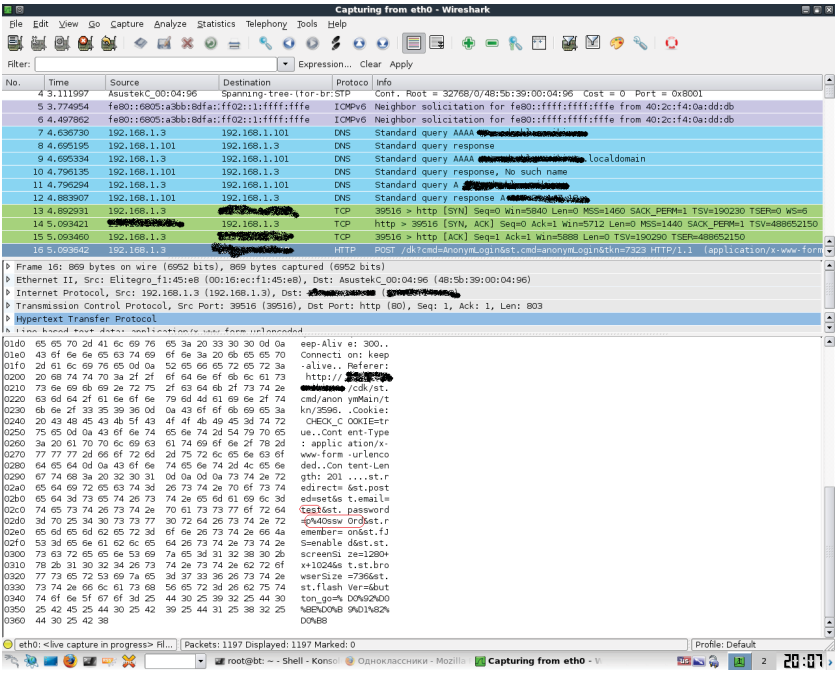

Рис. 6. *Поиск пароля в логах Wireshark*

Полученная подобным образом информация может быть впоследствии использована злоумышленником для реализации более сложных атак.

Приведенный пример является доказательством непригодности концентраторов для использования в локальных сетях. Но не торопитесь их выбрасывать. Если в вашей сети используются отказоустойчивые кластеры (необходимость использования отказоустойчивости подробно рассматривается в разделе, посвященном рискам), то вы сможете использовать концентраторы для внутреннего подключения узлов кластера. Дело в том, что узлам кластера необходимо постоянно обмениваться сообщениями вида «я живой» (I'm alive), эти сообщения передаются узлами друг другу. Так как концентратор не составляет CAM-таблиц, а шлет пакеты напрямую, то в случае выхода из строя одного из узлов второй узел быстрее узнает о выходе из строя первого. Коммутаторы составляют CAM-таблицы, которые обладают некоторым временем жизни, и при выходе из строя одного из узлов второй узнает об этом лишь по истечении этого времени жизни. Од-

нако подробнее о коммутаторах мы поговорим в следующем разделе. Для критичных бизнес-приложений, таких как электронная почта, корпоративная база данных или веб-сайт, простой продолжительностью в несколько секунд крайне вреден, а для карьеры системного администратора просто опасен. Так что использование концентраторов в кластерных системах вполне оправдано.

Для того чтобы избежать данных угроз, необходимо использовать коммутаторы, о которых речь пойдет далее.

Сейчас встретить концентратор в корпоративной сети уже не так просто. Повсеместно их заменяют коммутаторами, и это наилучший способ избежать тех угроз, которые были описаны в этом разделе. В случае если в вашей сети имеются концентраторы и в настоящий момент вы не можете от них отказаться, необходимо программными средствами заблокировать пользователям возможность прослушивать трафик. Проще всего это сделать, лишив пользователей административных привилегий на своих рабочих станциях. Также необходимо запретить работу сетевой карты в режиме, позволяющем получать весь трафик, а не только тот, который предназначен для данной машины.

2.2. Атаки на канальном уровне

На этом уровне арсенал злоумышленника уже значительно расширяется, и системному администратору нужно предпринять целый ряд мер для защиты корпоративной сети.

2.2.1. Атаки на коммутаторы

Коммутатор (switch) является более интеллектуальным устройством, чем концентратор. Как уже упоминалось ранее, коммутаторы работают на канальном уровне модели OSI. Получая пакет на один из своих портов, он, в отличие от концентратора, не пересылает его на все порты, а пересылает только на тот порт, к которому подключен получатель пакета.

Далее я приведу основные типы атак, которые применимы для коммутаторов. Существуют модели коммутаторов, поддерживающие также сетевой уровень, однако сейчас мы будем рассматривать только канальный уровень.

На канальном уровне возможны следующие типы атак:

- переполнение CAM-таблицы;

- VLAN Hopping;
- атака на STP;
- MAC-спуфинг;
- атака на PVLAN;
- атака на DHCP.

Рассмотрим каждую из атак более подробно.

2.2.2. Переполнение CAM-таблицы

Коммутатор имеет CAM-таблицу (Content Address Memory), где содержится привязка MAC-адресов к портам коммутатора. То есть в данной таблице указано, какие MAC-адреса на каком порту принимаются. CAM-таблица имеет ограниченный размер, например для коммутатора Cisco Catalyst 2960 таблица может хранить до 8192 MAC-адресов, а Catalyst 6000 серии – до 128 000 MAC-адресов.

В случае если таблица будет полностью занята, новые записи не смогут добавляться, и весь трафик будет проходить на все порты. Тогда коммутатор начнет работать как обычный концентратор, и весь трафик, проходящий через данный сегмент сети, можно будет прослушать тем же способом, который мы использовали в предыдущем разделе, с помощью утилиты Wireshark. Конечно, прослушать весь трафик в локальной сети злоумышленнику таким способом не удастся, но инсайдер, работающий в одном сегменте сети, к примеру с бухгалтерией, сможет перехватывать трафик и получить конфиденциальную информацию (рис. 7).

Реализовать данную атаку можно с помощью утилиты macchanger, которая позволяет менять MAC-адреса.

В качестве примера осуществим подмену MAC-адреса на машине, подключенной к коммутатору. Сначала включим сетевой интерфейс. По умолчанию в BackTrack сетевой интерфейс отключен.

```
root@bt:~# ifup eth0
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Listening on LPF/eth0/00:16:ec:f1:45:e8
Sending on   LPF/eth0/00:16:ec:f1:45:e8
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
DHCPOFFER of 192.168.1.3 from 192.168.1.101
DHCPREQUEST of 192.168.1.3 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.1.3 from 192.168.1.101
```

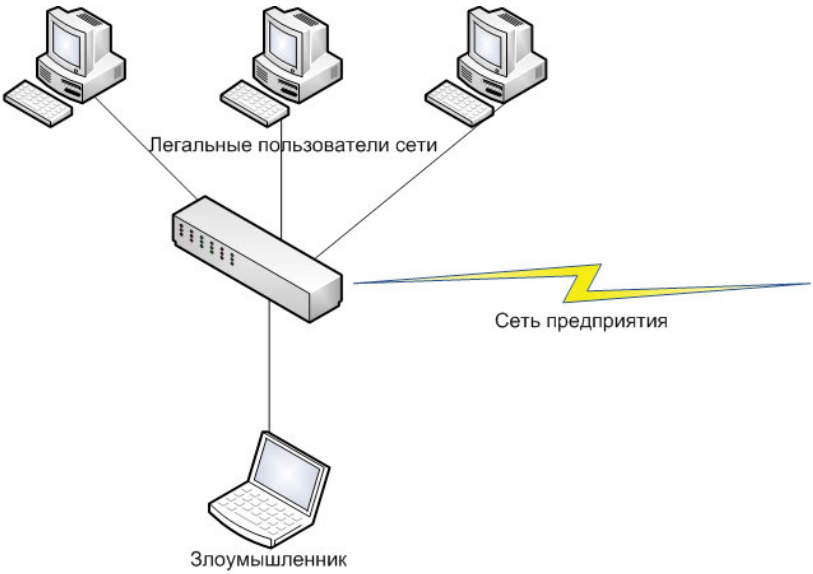

Рис. 7. *Топология сети с коммутатором*

```
bound to 192.168.1.3 -- renewal in 42928 seconds.
if-up.d/mountnfs[eth0]: waiting for interface eth1 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface eth2 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface ath0 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface wlan0 before doing NFS mounts
```

При включении был отправлен запрос на получение IP-адреса к серверу DHCP. Затем посмотрим текущее состояние сетевых интерфейсов.

```
root@bt:~# ifconfig -a
eth0    Link encap:Ethernet  HWaddr 00:16:ec:f1:45:e8
     inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0
     inet6 addr: fe80::216:ecff:fef1:45e8/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:39 errors:0 dropped:0 overruns:0 frame:0
     TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:3952 (3.9 KB)  TX bytes:1780 (1.7 KB)
     Interrupt:21 Base address:0xd800

lo   Link encap:Local Loopback
     inet addr:127.0.0.1  Mask:255.0.0.0
```

```
     inet6 addr: ::1/128 Scope:Host
     UP LOOPBACK RUNNING  MTU:16436  Metric:1
     RX packets:0 errors:0 dropped:0 overruns:0 frame:0
     TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:0
     RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

Настоящий MAC-адрес нашего сетевого интерфейса – 00:16:ec:f1:45:e8. Сейчас именно этот адрес прописан в CAM-таблице коммутатора. Теперь изменим данный адрес.

```
root@bt:~# macchanger -r eth0
Current MAC: 00:16:ec:f1:45:e8 (unknown)
Faked MAC: 04:2f:11:65:fc:0a (unknown)
```

Перестартуем сетевой интерфейс:

```
root@bt:~# ifdown eth0
root@bt:~# ifup eth0
Internet Systems Consortium DHCP Client V3.1.1
Copyright 2004-2008 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Listening on LPF/eth0/04:2f:11:65:fc:0a
Sending on   LPF/eth0/04:2f:11:65:fc:0a
Sending on   Socket/fallback
DHCPDISCOVER on eth0 to 255.255.255.255 port 67 interval 3
DHCPOFFER of 192.168.1.8 from 192.168.1.101
DHCPREQUEST of 192.168.1.8 on eth0 to 255.255.255.255 port 67
DHCPACK of 192.168.1.8 from 192.168.1.101
bound to 192.168.1.8 -- renewal in 42928 seconds.
if-up.d/mountnfs[eth0]: waiting for interface eth1 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface eth2 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface ath0 before doing NFS mounts
if-up.d/mountnfs[eth0]: waiting for interface wlan0 before doing NFS mounts
```

Снова посмотрим конфигурацию сетевых интерфейсов.

```
root@bt:~# ifconfig -a
eth0    Link encap:Ethernet  HWaddr : 04:2f:11:65:fc:0a
     inet addr:192.168.1.8  Bcast:192.168.1.255  Mask:255.255.255.0
     inet6 addr: fe80::216:ecff:fef1:45e8/64 Scope:Link
     UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
     RX packets:39 errors:0 dropped:0 overruns:0 frame:0
     TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
     collisions:0 txqueuelen:1000
     RX bytes:5801 (5.9 KB)  TX bytes:1100 (1.1 KB)
     Interrupt:21 Base address:0xd800
```