Сниффферы и методы их обнаружения

Петрухин Сергей Владимирович

2005-05-11

Эссе по курсу "Защита информации", кафедра радиотехники, Московский физико-технический институт (ГУ МФТИ), http://www.re.mipt.ru/infsec

Сниффер (sniffer) – это программа, позволяющая просматривать сетевой трафик в пределах данного сегмента сети. Ее возможности основаны на особенностях протокола Ethernet, на котором, в свою очередь, построено большинство локальных сетей. В этом протоколе все устройства используют единую магистраль для обмена данными со скоростью 10, 100, 1000 Мбит/с. Каждое устройство имеет свой уникальный MAC-адрес (Media Access Control) длиной 6 байт, прописанный в сетевой плате. При передаче по сети пакета (фрейма) Ethernet, в его заголовке указывается MAC-адрес отправителя и MAC-адрес получателя. Различные устройства сетевого обмена, как-то хабы (hub) или свичи (switch) ретранслируют этот пакет на все устройства в данном сегменте сети; таким образом, каждое устройство получает не только данные, предназначенные ему, но и множество других пакетов. Однако, по умолчанию сетевая плата отсеивает пакеты с MAC-адресом получателя, отличающимся от собственного, и таким образом организуется передача данных между двумя устройствами.

Структура кадра Ethernet:

- MAC-адрес получателя (6 байт)
- MAC-адрес отправителя (6 байт)
- Тип протокола верхнего уровня (2 байта)
- Данные
- CRC-32 (32 байта)

Данные представляют собой информацию, инкапсулированную протоколом следующего уровня. Чаще всего это IP (Internet Protocol) или ARP(Address Resolution Protocol). Протокол IP предназначен для передачи информации в глобальных сетях и содержит IP-адреса отправителя и получателя, протокол ARP предназначен для установления соответствия между IP-адресом и MAC-адресом. Поскольку при работе в глобальных сетях используется IP-адрес, то необходимо определить MAC-адрес соответствующего устройства.

Структура кадра ARP:

- Тип аппаратуры (в нашем случае Ethernet) (2 байта)
- Протокол (в нашем случае IP) (2 байта)
- Длина аппаратного адреса (в нашем случае 6 байт) (1 байт)
- Длина протокольного адреса (в нашем случае 4 байта) (1 байт)
- Вид операции (запрос или ответ) (2 байта)
- Аппаратный (MAC) адрес отправителя (в нашем случае 6 байт)
- Протокольный (IP) адрес отправителя (в нашем случае 4 байта)
- Аппаратный адрес получателя
- Протокольный адрес получателя

При формировании пакета Ethernet, устройство просматривает свою ARP-таблицу, и, если не находит соответствующего MAC-адреса, посылает ARP-запрос с IP-адресом получателя и MAC-адресом, равным FF FF FF FF FF FF. Устройство, получившее такой Ethernet-пакет, и обладающее искомым IP-адресом, посылает аналогичный ARP-пакет со своим MAC-адресом.

Структура программы-сниффера

Во-первых, необходимо каким-то образом иметь доступ к непосредственно сетевому трафику. Т.к. сетевая плата по умолчанию игнорирует пакеты, не предназначенные для нее, то необходимо перевести ее в состоянии полного прослушивания (**promiscuous mode).** Такая возможность может быть использована только при низком уровне работы с сетевой платой. Операционные системы Windows, начиная с Windows 2000, предоставляют функции для управления состоянием сетевой платы и возможность включения promiscuous mode. Таким образом, эту часть работы выполняет сама операционная система.

Во-вторых, необходимо иметь некий набор правил и фильтров, согласно которым из общего сетевого потока будут выбираться необходимые пакеты. Необходимость наличия этой части программы обуславливается тем, что в большинстве случаев нет нужды просматривать все пакеты идущие через сеть, и, к тому же, запись всех пакетов быстро приведет к заполнению буфера или места на жестком диске. Как правило, владелец машины, на которой запущен сниффер, желает просматривать трафик только от какого-либо устройства, либо трафик только одного типа (например, только протокол UDP). К тому же, в некоторых случаях может понадобиться отправка какого-либо пакета в сеть при получении заданного пакета.

В-третьих, может пригодиться подпрограмма, позволяющая зашифровывать и расшифровывать сообщения. Если информация зашифрована, но известен алгоритм шифрования, то для дальнейшей работы необходимо расшифровать сообщение. Например, известный сервис ICQ передает данные в зашифрованном виде, однако можно перехваченные данные восстановить.

В-четвертых, это внешний интерфейс программы. Получаемые пакеты можно либо просто выводить на экран, либо сохранять на жесткий диск для дальнейшего анализа.

Код для реализации прослушивания: (необходимо подключить ws2_32.lib)

```
#include <conio.h>
#include <winsock2.h>
#define MAX_PACKET_SIZE  0x10000
// Буфер для приёма данных
static BYTE Buffer[MAX_PACKET_SIZE]; // 64 Kb
void main()
{
 WSADATA    wsadata;   // Инициализация WinSock.
 SOCKET     s;         // Слущающий сокет.
 char       name[128]; // Имя хоста (компьютера).
 HOSTENT*   phe;       // Информация о хосте.
 SOCKADDR_IN sa;       // Адрес хоста
 long       flag = 1;  // Флаг PROMISC Вкл/выкл.
 // инициализация
 WSAStartup(MAKEWORD(2,2), &wsadata);
 s = socket( AF_INET, SOCK_RAW, IPPROTO_IP );
 gethostname(name, sizeof(name));
 phe = gethostbyname( name );
 ZeroMemory( &sa, sizeof(sa) );
 sa.sin_family = AF_INET;
 sa.sin_addr.s_addr = ((struct in_addr *)phe->h_addr_list[0])->s_addr;
 bind(s, (SOCKADDR *)&sa, sizeof(SOCKADDR));
```

```
  // Включение promiscuous mode.
  ioctlsocket(s, SIO_RCVALL, &flag);
  // Приём IP-пакетов.
  while( !_kbhit() )
  {
    int count;
    count = recv( s, Buffer, sizeof(Buffer), 0 );
    // обработка IP-пакета
    if( count >= sizeof(IPHeader) )
    {
      IPHeader* hdr = (IPHeader *)Buffer;
      //что-то делаем с пакетом...
    }
  }
  // Конец работы.
  closesocket( s );
WSACleanup();
}
```

Код взят из http://www.rsdn.ru/article/net/sniffer.xml

Методы обнаружения снифферов

Несмотря на кажущуюся невозможность определения устройства с программой-снифферером, это реализуемо благодаря тому, что программа изменяет стандартное поведение сетевого устройства. Администратор должен, прежде всего, заподозрить некий узел на предмет наличия на нем программы-сниффера. К этому узлу могут быть применены следующие проверки.

Проверка ping-ом. При получении ARP-пакета, содержащего соответствие IP-адреса и MAC-адреса сетевая плата прописывает его в ARP-таблицу. Если мы пошлем ARP-пакет со своим IP-адресом и несуществующим в данном сегменте MAC-адресом, но при этом адрес назначения поставим не «широковещательный» FF FF FF FF FF FF, то сетевая плата, работающая в обычном режиме, пропустит этот пакет, а работающая в promiscuous mode запишет в свою ARP-таблицу. Если после этого мы пошлем ICMP-пакет «Echo request» (ping) на «широковещательный» IP-адрес, то те машины, которые прислали ICMP-ответ без ARP-запроса, получили соответствие ARP-IP из первого, «нешироковещательного» ARP-пакета, и, следовательно, находятся в режиме прослушивания сети.

Существуют вариации этого метода. Например, можно просто послать ARP-запрос с IP-адресом подозреваемой машины, но с «нешироковещательным» MAC-адресом. Получение ARP-пакета от этого устройства будет означать, что на его сетевой карте отключен фильтр MAC-адресов. Еще один простой метод состоит в отправке ICMP-пакета «Echo request» на IP-адрес подозреваемого узла, но не с его MAC-адресом. Можно построить любые другие вариации, основание на протоколах с ответом или подтверждением, например TCP, или на протоколах с выявлением ошибок, когда ошибка в заголовке приводит к отправке обратного пакета.

При этой проверке необходимо учитывать тот факт, что многие машины могут осуществлять проверку на «широковещательный» MAC-адрес только по первому байту (т.е. «широковещательными» считаются адреса вида FF xx xx xx xx xx), некоторые адаптеры считают пакет «широковещательным», если его первый байт является нечетным числом. Все это может привести к неверному опознаванию узла в качестве сниффера.

Проверка на DNS-активность.

Многие снифферы автоматически делают запрос DNS при обнаружении неизвестного IP-адреса. Поэтому просмотр DNS-активности при посылке пакета с заведомо несуществующим IP-адресом позволяет выявить устройства с прослушивающими сетевыми платами. Этот метод можно применить как на сервере, следя за DNS-активностью и определяя активные адреса, так и локально, переключив свою сетевую плату в режим прослушивания и следя за DNS-трафиком от подозреваемых узлов.

Интересной особенностью этого метода является то, что, как правило, хакерские снифферы пытаются определить DNS сразу после получения IP-адреса, в то время как «легальные» программы откладывают это определение.

Проверка через измененный маршрут.

Для проверки необходимо создать пакет с ping-ом, предназначенный для подозреваемого узла, но в его заголовок включить маршрут через некоторый узел, про который заранее известно, что он не выполняет функции маршрутизации. Тогда промежуточный узел просто проигнорирует этот пакет, и он не дойдет до адресата. Поэтому, если после отправки данного пакета в сеть мы получили ответ от подозреваемого узла, то, значит, он получил пакет при прослушивании сети.

Если про промежуточный узел не известно, является ли он маршрутизирующим, то можно осуществить проверку через TTL-поле IP-заголовка возвращаемого пакета. В данном случае, это значение может быть использовано для определения того, был ли данный пакет перенаправлен или взят прямо из сети.

Проверка через ловушки.

Многие программы-снифферы рассчитаны на то, что по многим протоколам данные о паролях передаются в незашифрованном виде. Когда происходит авторизация, через сеть проходит пакет с паролем, и сама программа-сниффер автоматически или злоумышленник после обработки результатов, могут попытаться получить доступ к другой машине. Поэтому на какой-либо машине в сети настраивается сервис, и данные о каком-то вымышленном пользователе, который не имеет реальный привилегий, передаются в сеть. В случае попытки получить доступ от имени этого пользователя компьютер с установленным сниффером может быть вычислен путем анализа логов или с использованием систем обнаружения вторжения.

Проверка на локальной машине.

Если существует вероятность того, что машина была взломана, то злоумышленник мог оставить на ней программу, просматривающую сеть с данного компьютера. Для определения этого необходимо просмотреть открытые подключение на предмет того, находятся ли они в прослушивающем режиме.

Проверка на задержку (latency)

Если программа-сниффер нацелена только на сбор пакетов из сети и записи их на диск, то такая машина может быть обнаружена путем замера времени ответа на

стандартные ping-пакеты. Т.к. для обработки пакета и записи его на диск требуется некоторое время, то время ответа этой машины будет больше среднего времени ответа для данного сегмента сети.

Кроме этих методов, позволяющих выявить компьютер с программой-сниффером, можно применить ряд мер, делающих работу сниффера бессмысленной или усложненной.

Во-первых, можно применить кодирование передаваемой информации. Если алгоритм дешифрования не известен третьей стороне, то сниффер не сможет получить информацию. Но даже если алгоритм известен, то, при условии, что дешифрование займет много времени, сниффер не сможет производить декодирование в реальном времени.

Во-вторых, использование маршрутизаторов и свичей вместо концентраторов. Это уменьшает размер сегмента сети, в котором трафик от всех узлов общий и виден всем в данном сегменте.

Программы-снифферы – достаточно мощный инструмент, с помощью которого можно собрать информацию о сети для проведения последующей атаки или же получить имена и пароли пользователей в данном сегменте сети, т.к. большинство из них передается в незашифрованном виде. Последнее усугубляется тем, что, как правило, пользователи имеют одинаковые пароли для доступа ко многим сервисам и, перехватив открытый пароль к «несерьезному» сервису, злоумышленник может его использовать, например, для получения полного доступа к компьютеру. С другой стороны, так же снифферы могут использоваться сетевыми администраторами для проверки целостности сети и правильности ее работы, конфигурации сетевого оборудования, для предугадывания, отслеживания и предотвращения компьютерных атак.

Снифферы также могут использоваться программистами для отладки программного обеспечения, работающего с сетью, таких как системы типа клиент-сервер и распределенные системы.

ЛИТЕРАТУРА:

1. Константин Максимов. Сниффер: щит и меч. http://www.rsdn.ru/article/net/sniffer.xml
2. Ярослав Клюкин. Обнаружение пакетных снифферов. http://www.linuxcenter.ru/lib/articles/security/detect_sniff.phtml
3. Опознание снифферов. http://www.void.ru/content/652