**Лабораторная работа №7**

**Изучение средств мониторинга и анализа сетевого трафика. Сетевой анализатор Wireshark**

**Цели работы:**

1 Знать принципы анализа сетевого трафика.

2 Научиться использовать сетевой анализатор (сниффер Wireshark).

3 Научиться анализировать сетевой трафик на примере протоколов

ARP, IP и ICMP.

**Теоретическая часть**

Sniffer (от англ. to sniff – нюхать) – это сетевой анализатор

трафика, программа или программно-аппаратное устройство,

предназначенное для перехвата и последующего анализа, либо только

анализа сетевого трафика, предназначенного для других узлов.

Перехват трафика может осуществляться:

- обычным «прослушиванием» сетевого интерфейса (метод

эффективен при использовании в сегменте концентраторов (хабов)

вместо коммутаторов (свичей), в противном случае метод

малоэффективен, поскольку на сниффер попадают лишь

отдельные фреймы);

- подключением сниффера в разрыв канала;

- ответвлением (программным или аппаратным) трафика и

направлением его копии на сниффер;

- через анализ побочных электромагнитных излучений и

восстановление таким образом прослушиваемого трафика;

- через атаку на канальном (2-й) или сетевом (3-й) уровне,

приводящую к перенаправлению трафика жертвы или всего

трафика сегмента на сниффер с последующим возвращением

трафика в надлежащий адрес.

В начале 1990-х широко применялся хакерами для захвата

пользовательских логинов и паролей. Широкое распространение хабов

позволяло захватывать трафик без больших усилий в больших

сегментах сети.

Снифферы применяются как в благих, так и в деструктивных

целях. Анализ прошедшего через сниффер трафика, позволяет:

- Отслеживать сетевую активность приложений.

- Отлаживать протоколы сетевых приложений.

- Локализовать неисправность или ошибку конфигурации.

- Обнаружить паразитный, вирусный и закольцованный трафик,

наличие которого увеличивает нагрузку сетевого оборудования и

каналов связи.

- Выявить в сети вредоносное и несанкционированное ПО,

например, сетевые сканеры, флудеры, троянские программы,

клиенты пиринговых сетей и другие.

- Перехватить любой незашифрованный (зашифрованный) пользовательский трафик с целью узнавания паролей и другой информации.

**Базовый принцип работы снифферов**

Давайте рассмотрим с вами рис. 1 На нем изображена схематично

структура сетевой подсистемы ОС. Вся базовая инфраструктура

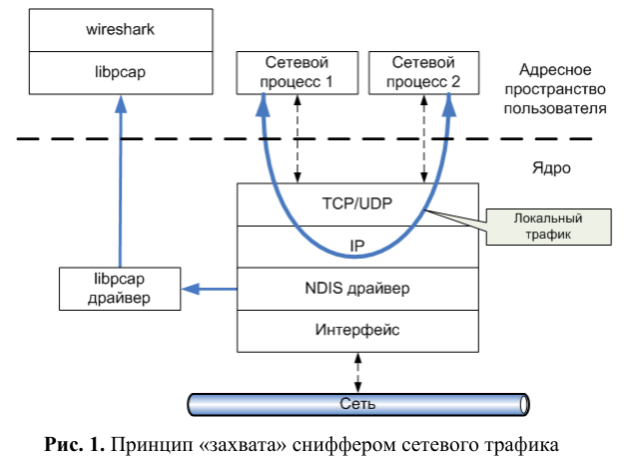
реализована в виде драйверов и работает в режиме ядра.

Пользовательские процессы и реализации прикладных протоколов, в

частности интерфейс сниффера работают в пользовательском режиме.

На рисунке отображены 2 пользовательских процесса («сетевой

процесс 1» и «сетевой процесс 2»).



Основными компонентами сниффера являются: драйвер для

захвата пакетов (libpcap драйвер), интерфейсная библиотека (libpcap) и

интерфейс пользователя (Wireshark). Библиотека libpcap (реализация

под ОС Windows носит название WinPcap - http://www.winpcap.org) –

универсальная сетевая библиотека, самостоятельно реализующая

Большое количество непосредственно с драйверами сетевых устройств. На базе данной библиотеки реализовано большое количество сетевых программ, в частности сниффер Wireshark. Сниффер использует библиотеку в режиме «захвата» пакетов, т.е. может получать копию ВСЕХ данных проходящих через драйвер

сетевого интерфейса. Изменения в сами данные не вносятся!

Основной нюанс использования сниффера заключается в том, что

он не позволяет производить анализ локального трафика, т.к. он не

проходит через драйвер сетевого устройства (см. рис 1.). Т.е., если вы

захотите проанализировать сниффером трафик между 2-ми сетевыми

процессами на локальной машине (например, ftp-сервер и ftp-клиент),

то у вас ждет разочарование. Однако, например при использовании

виртуальных машин, сниффер будет работать без проблем, т.к.

виртуальные машины эмулируют реальную среду и сетевые адаптеры,

поэтому трафик идет через драйвера как и в нормальной ситуации при

взаимодействии с другими физическими сетевыми машинами.

Также к недостаткам большинства снифферов стоит отнести и тот

факт, что, позволяя анализировать трафик, проходящий через сетевой

интерфейс, они не могут указать, какое именно приложение генерирует

или получает его. Это объясняется те, что информация об этом

хранится на сетевом (например,

большинство снифферов использует собственную реализацию стека

протоколов (например, библиотеку WinPсap), которая (как уже было

показано) работает непосредственно с драйверами устройств.

Также, снифферы вносят дополнительную нагрузку на процессор,

т.к. могут обрабатывать достаточно объемный сетевой трафик, в

особенности для высокоскоростных соединений (Fast Ethernet, Gigabit

Ethernet и др.).

Задания к работе:

1. **Протокол ICMP**
2. запустить Wireshark;
3. настроить фильтр (icmp);
4. запустить процесс захвата трафика;
5. в командной строке: ping –n 10 конечный\_узел например, ping –n 10 wireshark.org в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;
6. остановить захват трафика.

Ответьте на следующие вопросы.

1. Сколько всего пакетов захватила программа? Почему?
2. Какой IP-адрес вашего компьютера, адрес назначения?
3. Проанализируйте ping request, отправленный с вашего компьютера: укажите тип и код ICMP. Какие еще поля содержит ICMP пакет? Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?
4. Проанализируйте ping reply: укажите тип и код ICMP. Какие еще поля содержит ICMP пакет? Сколько байт занимают поля «Checksum», «Sequence number», «Identifier»?
5. запустить Wireshark;
6. настроить фильтр (icmp);
7. запустить процесс захвата трафика;
8. в командной строке: tracert конечный\_узел например, tracert wireshark.org в качестве конечного узла использовать URL, в котором присутствуют любые три буквы из фамилии студента в латинской транскрипции;
9. остановить захват трафика.

Ответьте на следующие вопросы.

1. Какой IP-адрес вашего компьютера, адрес назначения?
2. Проанализируйте пакет ICMP echo: отличаются ли эти пакеты от пакетов в первой части эксперимента? Чем?
3. Проанализируйте пакет ICMP error: какие поля в нем содержатся?
4. Чем отличаются пакеты ICMP reply (полученные) и ICMP error?
5. **Протокол Ethernet**
6. очистить кэш браузера;
7. запустить Wireshark;
8. запустить процесс захвата трафика;
9. URL: например, http://ru.wikipedia.org/wiki/Ethernet ;
10. остановить захват трафика;
11. в меню «Analyze» → «Enabled Protocols» можно снять галочку IP:тогда в списке пакетов не будет отображаться информация по протоколам верхнего уровня (после IP) — необязательный пункт.

Выберите кадр Ethernet, содержащий сообщение HTTP GET.

Ответьте на следующие вопросы.

1. Укажите 48-битный Ethernet адрес вашего компьютера.
2. Укажите 48-битный Ethernet адрес назначения. Что это за адрес? (Адрес сервера?)
3. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

Выберите кадр Ethernet, содержащий ответ HTTP. Ответьте на следующие вопросы.

1. Укажите значение Ethernet адреса источника. Какое устройство имеет такой адрес?
2. Укажите Ethernet адрес назначения: это адрес вашего компьютера?
3. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует?

**3. Протокол ARP**

1) очистить ARP кэш:

«Пуск» → «Выполнить»: netsh interface ip delete arpcache Вывести на экран ARP-таблицу можно с помощью команды: arp –a

2) очистить кэш браузера;

3) запустить Wireshark;

4) запустить процесс захвата трафика;

5) URL: например, http://ru.wikipedia.org/wiki/Ethernet в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;

6) остановить захват трафика;

7) в меню «Analyze» → «Enabled Protocols» снять галочку IP: в списке пакетов теперь не будет отображаться информация по протоколам верхнего уровня (после IP) — необязательный пункт.

Ответьте на следующие вопросы.

1. Укажите 16-чные значения адресов источника и назначения в пакете, содержащем ARP запрос (ARP ответ).
2. Укажите 16-чное значение двухбайтового поля «Type»: какому протоколу верхнего уровня оно соответствует (для ARP запроса/ARP ответа)?
3. Укажите значение поля «opcode» (для ARP запроса/ARP ответа). 4. Содержит ли ARP запрос IP-адрес источника (для ARP запроса/ARP ответа)?
4. **Протокол HTTP: Basic HTTP GET/response**
5. запустить Wireshark;
6. настроить фильтр (http);
7. запустить процесс захвата трафика;
8. URL: например, http://wiki.wireshark.org/; в URL должны присутствовать любые три буквы из фамилии студента в латинской транскрипции;
9. остановить захват трафика.

Прим.: Не принимать во внимание HTTP запрос и ответ для favicon.ico. Появление ссылки на данный файл означает, что браузер автоматически запрашивает сервер о наличии маленького значка веб-сайта, т. н. «Favicon» (отображается браузером в адресной строке перед URL страницы, а также в качестве картинки рядом с закладкой, во вкладках и в других элементах интерфейса).

В списке захваченных пакетов найдите пару HTTP сообщений (запросответ): GET сообщение и ответ сервера. В информационном поле разверните строку, содержащую HTTP, и отметьте указанную ниже информацию.

1. Версия HTTP.
2. Принимаемые браузером языки.
3. IP-адреса вашего компьютера и сервера.
4. Код состояния HTTP. Что он означает?
5. Длина тела сообщения. (Содержимое поля заголовка объекта ContentLength указывает длину тела сообщения в октетах (десятичное число), или в случае метода HEAD, размер тела объекта, который мог бы быть послан при запросе GET.)
6. Протокол транспортного уровня, который использует HTTP.

По каждому из заданий в отчете о работе должна быть представлена следующая информация.

Наименование исследуемого протокола.

Скриншот после остановки захвата трафика, если задание из двух частей - то соответственно 2 скриншота. Ответы на поставленные в задании вопросы.