1.1. Лабораторные работы

Работа №1. Средства анализа пакетного трафика. Утилита Tcpdump (Windump)

Цель работы

- Ознакомление с принципами работы программ анализа пакетного трафика;
- Практическое освоение приемов сбора и анализа трафика с помощью утилиты Tcpdump (Windump);
- Получение навыков расшифровки выходных результатов работы утилиты Tcpdump.(Windump).

Программы анализа пакетного трафика

Для исследования процессов в компьютерных сетях широко используются программы перехвата и анализа пакетов. Эти программы являются основным инструментом при решении следующих задач:

- Обнаружение проблем и узких мест сети (troubleshooting).

Анализаторы трафика позволяют получить практически полную картину событий, происходящих в сети: интенсивность трафика по времени, по рабочим станциям, по протоколам, количество ошибок разных типов. Кроме того, могут быть детально исследованы различные специфические проблемы, когда, скажем, конкретной станции не удается организовать некое взаимодействие по сети, хотя внешне сеть выглядит вполне работоспособной.

- Отладка разрабатываемого сетевого ПО. Часто только тщательный разбор заголовков отправляемых и принимаемых пакетов позволяет найти причину неверного функционирования ПО;

- **Обучение**. Результаты работы анализаторов трафика являются прекрасной иллюстрацией теоретических материалов и спецификаций. В случаях, когда сетевое ПО плохо документировано или использует свои закрытые (недокументированные) протоколы; анализаторы трафика представляют собой чуть ли не единственное средство для их изучения.

- **Протоколирование сетевого трафика**. Политика безопасности многих организаций включает регистрацию входящих и выходящих пакетов для дальнейшего просмотра с целью обнаружения попыток несанкционированного доступа к информации, различного рода атак и других нежелательных явлений.

Перехват и анализ сетевого трафика осуществляется специальными программами, называемыми анализаторами трафика или снифферами (Sniffer – слушатель, точнее - вынюхиватель). Такие программы, как правило, построены на основе различных библиотек захвата пакетов (наиболее распространенными являются BPF (Berkeley Packet Filter) и LibPCap) и, в общем случае, реализуют совокупность мер по перехвату и декодированию пакетов, поступающих на доступные сетевые интерфейсы компьютера.

В результате перехвата пакетов (packet capturing) получают некий “сырой” (machine readable) дамп данных, обычно разделенный на блоки по границам кадров (пакетов).

В результате декодирования пакетов (packet decoding) пользователь получает расшифрованную версию поступивших в процессе первого этапа данных, представляющую данные в формате, удобном для чтения человеком (human readable).

Сетевые анализаторы могут перехватывать как свой трафик (относящийся к узлу, на котором функционирует анализатор), так и весь трафик сетевого сегмента.

При работе в локальных сетях, сетевые платы (например, Ethernet), в нормальном режиме захватывают только кадры со своим или

широковещательным MAC-адресом назначения. Таким образом, в нормальном режиме имеется возможность перехвата и анализа только “своего” трафика. Для перехвата пакетов всех станций сегмента сетевые анализаторы переводят сетевую карту в “неразборчивый” (promiscuous) режим, в котором карта принимает и не адресованные ей кадры.

Анализаторы трафика могут функционировать как на сетевых коммуникационных устройствах (маршрутизаторах, коммутаторах и пр.), так и на оконечных узлах сети.

Конкретные реализации анализаторов отличаются друг от друга главным образом следующими функциональными возможностями:

- поддерживаемыми физическими интерфейсами и протоколами канального уровня;
- качеством декодирования и количеством распознаваемых протоколов;
- пользовательским интерфейсом и удобством отображения;
- дополнительными возможностями: статистика, просмотр в реальном времени, генерирование или модификация пакетов и др.

Утилита TCPDUMP

Программа TcpDump является старейшим и наиболее часто используемым средством анализа сетевого трафика для Unix-систем. Утилита WinDump представляет собой версию TcpDump, работающую под управлением ОС Windows. Подробное описание и последние версии программы Tcpdump можно найти на сайте http://www.tcpdump.org.

Программа позволяет принимать и анализировать пакеты, приходящие на доступный сетевой интерфейс, выдавать полученные данные в удобном для пользователя формате, а также задавать различные условия фильтрации приходящих пакетов. Взаимодействие с пользователем основано на использовании текстового интерфейса и опций командной строки.

Программа позволяет сохранять полученные данные в файле на жестком диске в различных форматах, в том числе и с целью дальнейшей обработки другими анализаторами сетевого трафика.

Программа tcpdump использует пакетный фильтр BSD (BPF - BSD Packet Filter), существующий в современных реализациях ядра BSD для перехвата и фильтрации пакетов из сетевой платы (которая переведена в “promiscuous” режим).

На рисунке 2.3 показано взаимодействие BPF с драйвером Ethernet.

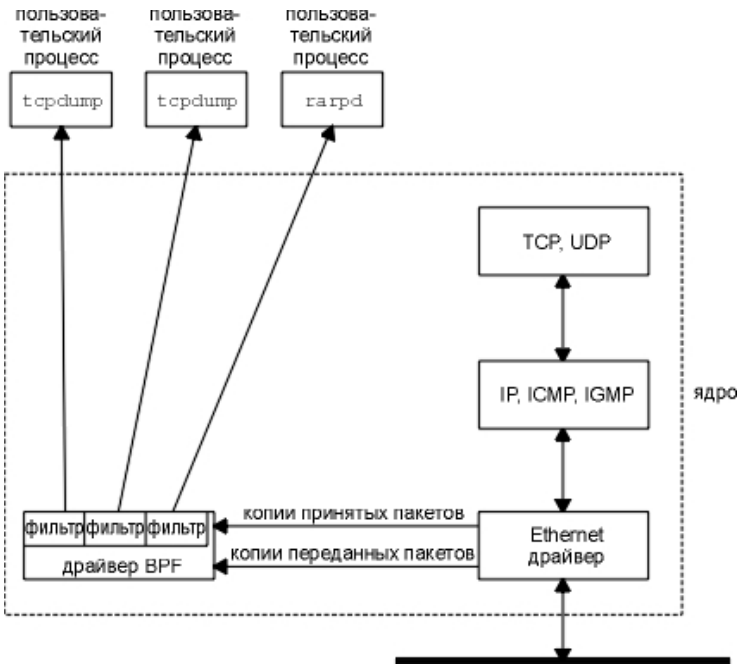

Рис.2.3 Пакетный фильтр BSD.

Драйвер BPF переводит драйвер Ethernet-карты в смешанный режим и затем получает от драйвера копию каждого полученного и отправленного пакета. Эти пакеты проходят через фильтр, указанный пользователем, таким образом, в обработку попадают только те пакеты, которые интересуют пользователя. Фильтр для tcpdump может быть указан пользователем в командной строке.

Запуск и останов программы

Запуск программы осуществляется командой

$> tcpdump <опции>

Завершение программы происходит после обработки заданного числа пакетов или при наборе комбинации Ctrl+C.

В результате программа выводит на экран последовательность символьных записей о проходящих пакетах (Рис. 2.4). Каждому пакету соответствует отдельная строка.

```
sun % tcpdump -e

tcpdump: listening on le0

09:11:22.642008 0:0:c0:6f:2d:40 ff:ff:ff:ff:ff:ff arp 60: arp who-has
svr4 tell bsdi

09:11:22.644182 0:0:c0:c2:9b:26 0:0:c0:6f:2d:40 arp 60: arp reply svr4
is-at 0:0:c0:c2:9b:26

09:11:22.644839 0:0:c0:6f:2d:40 0:0:c0:c2:9b:26 ip 60: bsdi.1030 >
svr4.discard: S 596459521:596459521 (0) win 4096 <mss 1024> [tos 0x10]

09:11:22.649842 0:0:c0:c2:9b:26 0:0:c0:6f:2d:40 ip 60: svr4.discard >
bsdi.1030:S 3562228225:3562228225 (0) ack 596459522 win 4096 <mss
1024>

09:11:22.651623 0:0:c0:6f:2d:40 0:0:c0:c2:9b:26 ip 60: bsdi.1030 >
svr4.discard: . ack 1 win 4096 [tos 0x10]
```

Рис.1 Формат вывода TCPDUMP

В начале вывода всегда указывается имя интерфейса, на котором осуществляется перехват пакетов. Каждая строка начинается с метки времени прихода пакета в формате ЧЧ:ММ:СС.СССССС. Для каждого пакета tcpdump всегда печатает имя (адрес) отправляющего хоста, затем знак "больше" (>), затем имя хоста назначения. Остальные детали записей и общий формат вывода зависят от опций командной строки и типа пакетов.

Результат можно сохранить в файле, указав в командной строке знак

“ > ” и имя файла.

Например, команда

```
% tcpdump tcp port 25 > dump.txt
```

формирует записи только о TCP-сегментах с портом источника или назначения равным 25 и сохраняет их в файле **dump.txt**.

Для запуска программы с требуемыми условиями фильтрации и вывода используется большой набор опций командной строки:

TcpDump [-adeflnNOpqStvx] [-c *count*] [-F *file*] [-i *interface*] [-r *file*] [-s *snaplen*] [-T *type*] [-w *file*] [*expression*].

-a разрешает конвертировать сетевые адреса в имена.

-c выход после обработки *count* пакетов.

-d выводит содержимое пакета в удобочитаемом виде.

-dd выводит содержимое пакета как фрагмент Си-программы.

-ddd выводит содержимое пакета в десятичном виде.

-e выводит заголовки канального уровня в каждой новой строке.

-f выводит адреса удаленных хостов без преобразования в имена.

-F указывает на использование файла *file* с описанием параметров фильтрации (дополнительные выражения в командной строке игнорируются).

-i указывает интерфейс *interface* для трассировки. Если не определен, *tcpdump* находит активный сетевой интерфейс с наименьшим номером (исключая loopback). В Windows *interface* — имя сетевого адаптера или его номер (можно узнать, запустив WinDump —D).

-l использует буферизированный вывод на stdout.

-n не преобразовывать адреса (т.е. адрес хоста, номер порта и т.д.) в имена.

-N не печатать доменное имя в имени хоста. Т.е. если использован данный флаг, *tcpdump* напечатает "nic" вместо "nic.ddn.mil".

-O не запускать оптимизатор пакетов.

-p не переводить сетевой интерфейс в "promiscuous mode".

-q выводит информацию в сокращенном виде.

-r читает пакеты из файла *file* (созданного при помощи опции -w).

-s выдает *snaplen* байт каждого пакета. 68 байт достаточно для протоколов IP, ICMP, TCP и UDP.

-T принудительная интерпретация пакетов по типу *type*.

-S выводит абсолютный номер TCP-пакета.

-t не выводит время в каждой строке.

-tt выводит неформатированное время в каждой строке.

-v детальный вывод. К примеру, TTL и тип сервиса.

-vv более детальный вывод. К примеру, вывод дополнительных полей NFS reply packets.

-w записывает raw-пакеты в *file,* который можно в дальнейшем расшифровать с использованием опции **-**r.

-x выводит каждый пакет в шестнадцатеричном виде. На вывод будет отправлено *snaplen* байт.

Дополнительные опции WinDump: [-D] [-B *size*].

-B устанавливает размер буфера драйвера *size* в килобайтах. По умолчанию размер буфера 1 мегабайт. Если в процессе работы некоторые пакеты не отображаются, следует увеличить размер буфера. Для PPP-соединений или 10 Мбит Ethernet, размер буфера можно уменьшить вдвое.

-D выводит список сетевых устройств, которые присутствуют в системе. Список имеет вид: *number* — номер сетевого устройства в системе, *name* — его имя, за ними следует описание устройства.

expression — собственно, выражение, которое задает критерий фильтрации пакетов. Если поле *expression* отсутствует, то выводятся все пакеты. В противном случае выводятся только те пакеты, которые соответствуют маске *expression.*

expression может состоять из одного или более примитивов. Примитивы часто состоят из *id* (имя или номер) определителя. Существует три ключевых типа определителей:

***type* —** определитель, задающий общую политику. Возможные типы — **host**, **net** и **port**. Например, "host foo", "net 128.3", "port 20". .

***dir* —** определитель, указывающий направление передачи пакетов. Возможные варианты **src**, **dst**, **src or dst** и **src and dst**. Т.е "src foo", "dst net 128.3", "src or dst port ftp-data". Если **dir** не указан, то по умолчанию используется **src or dst.** Для "null" соединений (это ppp или slip) используется **inbound** и **outbound** определитель для указания желаемого направления.

proto — определитель позволяет выводить пакеты конкретного протокола. Возможные протоколы: **ether**, **fddi**, **ip**, **arp**, **rarp**, **decnet**, **lat**, **sca**, **moprc**, **mopdl**, **tcp** и **udp**. Т.е. "ether src foo", "arp net 128.3", "tcp port 21".

В дополнение к вышесказанному, некоторые специальные примитивы не имеют шаблонов, это: **gateway**, **broadcast**, **less**, **greater** и арифметические выражения. Множество составных выражений фильтров используют слова **and**, **or** и **not** для объединения примитивов. К примеру "host foo and not port ftp and not port ftp-data". Для упрощения ввода некоторые определители могут быть опущены. К примеру, "tcp dst port ftp or ftp-data or domain" — то же самое, что и "tcp dst port ftp or tcp dst port ftp-data or tcp dst port domain".

Наиболее часто используются следующие выражения:

dst host *host* - пакеты, у которых поле destination IP-заголовка — *host*, При этом *host* может быть адрес или имя хоста.

src host *host* - пакеты, у которых поле source IP-заголовка — *host*.

host *host* - пакеты, у которых поля source или destination пакета — *host*. Также могут употребляться префиксы: **ip**, **arp**, или **rarp.** Если *host* — имя с несколькими IP-адресами, каждый адрес проверяется на соответствие.

ether dst *ehost* – пакеты, у которых Ethernet-адрес получателя — *ehost*. *Ehost* — любое из имен /etc/ethers или номер.

ether src *ehost* - пакеты, у которых Ethernet-адрес отправителя — *ehost*.

ether host *ehost* - пакеты, у которых Ethernet-адрес получателя или отправителя — *ehost*.

gateway *host* – пакеты к/от шлюза (*host*) Т.е. Ethernet-адрес отправителя или получателя — *host*, но ни IP-адрес отправителя, ни IP-алрес получателя не являются *host*. *Host* может быть именем, а также может находиться в /etc/hosts и /etc/ethers.

dst net *net* - пакеты, у которых адрес сети получателя — *net*. *Net* — любая запись из /etc/networks или адрес сети.

src net *net* - пакеты, у которых адрес сети отправителя — *net*.

net *net* - пакеты, у которых адрес сети отправителя или получателя — *net.*

dst port *port* - пакеты ip/tcp или ip/udp, у которых порт получателя — *port*. *port* может быть числом или присутствовать в /etc/services. Если имя используется для двух или более портов, то проверяются оба номера порта и протоколы. Если используются недопустимые номер порта или имя, то проверяются только номера портов (т.е. **dst port 513** выводит трафик tcp/login и udp/who, и **port domain** выводит tcp/domain и udp/domain).

src port *port* – пакеты, у которых порт отправителя — *port*.

port *port* – пакеты, у которых порт отправителя или получателя — *port*. Некоторые выражения можно комбинировать, к примеру: **tcp src port** *port* — только tcp-пакеты у которых порт — *port*.

less *length* - пакеты, длина которых меньше или равна *length*, что равносильно **len <=** *length*.

greater *length* - пакеты, длина которых больше или равна *length*, что равносильно **len >=** *length*.

ip proto *protocol* - IP пакеты с протоколом *protocol*. *Protocol* может иметь номер или одно из имен *icmp*, *igrp*, *udp*, *nd*, или *tcp*.

ether broadcast - широковещательные Ethernet-пакеты. Выражение *ether* является необязательным.

ip broadcast - широковещательные IP-пакеты.

ether multicast – пакеты Ethernet multicast. Выражение *ether* является необязательным. Это сокращенная запись для "**ether[0] & 1!= 0**".

ip multicast – пакеты IP-multicast.

Более полное описание см. http://www.tcpdump.org/

Примеры использования tcpdump

Выдача всех входящих и исходящих пакетов от *sundown*:

tcpdump host sundown

Выдача трафика между *helios* и одним из двух *hot* или *ace*:

tcpdump host helios and (hot or ace)

Выдача всех пакетов между *ace* и другими хостами, исключая *helios*:

tcpdump ip host ace and not helios

Выдача ftp трафика узла *onegin*:

tcpdump host onegin and (port ftp or ftp-data)

Выдача трафика не принадлежащего машинам в локальной сети (если ваша машина — шлюз в другую сеть, tcpdump не сможет выдать трафик вашей локальной сети). **tcpdump ip and not net** *localnet*

Выдача стартовых (SYN) и стоповых (FIN) пакетов TCP-соединений:

tcpdump ″tcp[13] & 3!= 0″

Соглашения о безопасности

Просмотр сетевого трафика позволяет увидеть многие детали, которые вообще-то не все должны видеть. Например, пароли, вводимые пользователями для различных приложений, таких как Telnet или FTP и передаваемые по сети именно так, как их ввел пользователь.

Мы используем tcpdump как обучающее средство, чтобы посмотреть, что в действительности передается по сети. Доступ к tcpdump и подобным ей утилитам зависит от системы (и может быть от воли системного администратора). В большинстве систем, обычный пользователь не может

запускать программы, подобные tcpdump, без санкции на то системного администратора.

Программа работы.

1. Запустить программу Tcpdump, ознакомиться с форматом вывода и основными опциями (–c, –e, –q, –n, –x).

2. Средствами программы Tcpdump определить минимальный и максимальный размер пакетов в сети.

3. Запустить программу Tcpdump с записью необработанных результатов в файл Rowdump.

4. Во время работы программы Tcpdump по п.3 выполнить операции в соответствии с индивидуальным заданием:

a) Обратиться к WEB-серверу.

b) Обратиться к FTP-серверу

c) Выполнить команду Ping . Сохранить результат в файле ping.txt

d) Выполнить команду Traceroute. Сохранить результат в файле trace.txt

Пункты 3-4 рекомендуется выполнять по подпунктам:

a) Запустить программу Tcpdump с записью необработанных результатов в файл RowdumpW. Обратиться к WEB-серверу. После ответа сервера остановить Tcpdump.

b) Запустить программу Tcpdump с записью необработанных результатов в файл RowdumpF. Обратиться к FTP-серверу. После ответа сервера остановить Tcpdump.

c) Запустить программу Tcpdump с записью необработанных результатов в файл RowdumpPing. Выполнить команду Ping . Сохранить результат (вывод команды Ping) в файле ping.txt. Остановить Tcpdump.

d) Запустить программу Tcpdump с записью необработанных результатов в файл RowdumpTrace. Выполнить команду tracert. Сохранить результат в файле trace.txt. Остановить Tcpdump.

5. Обработать файл Rowdump (файлы RowdumpW, RowdumpF, RowdumpPing, RowdumpTrace) с целью вывода только пакетов, отправленных или принятых Вашим компьютером (вывод Tcpdump для всех пакетов вашего компьютера). Сохранить результаты в файлах dumpW.txt, dumpF.txt, dumping.txt. dumptrace.txt соответственно. Пометить в файлах .txt пакеты, относящиеся к п. 4 (a, b, c, d соответственно).

В отчете представить:

1. Номер рабочего места и IP-адрес ПК.

2. Команды (с комментариями) и результаты для опций по п.1.

3. Для каждой опции привести вывод 3-4 пакетов. При выводе IP-пакетов с опцией –x, расшифровать заголовок IP-пакета.

4. Команды (с комментариями), использованные при выполнении п. 2, и полученные результаты.

5. Команды, использованные при выполнении пп. 3, 4, 5.

6. Распечатки файлов ping.txt и trace.txt.

7. Распечатки фрагментов (не более 1 стр.) файлов dumpW.txt, dumpF.txt, dumping.txt. dumptrace.txt dump2.txt с помеченными пакетами, относящимися к пп. 4 a, b, c, d, соответственно. Определить значение задержки эхо-отклика (icmp) по файлу dumpPing.txt, сравнить с результатом в файле ping.txt.

8. Выводы по всем пунктам работы.

Варианты заданий для лабораторной работы “Утилита TCPDUMP”

Вариант	WEB-сервер	FTP-сервер	Ping	Trace
1	www.spbstu.ru	ftp.dlink.ru/	list.ru	list.ru
2	www.rbc.ru	ftp.neva.ru	www.rbc.ru/	www.rbc.ru/
3	www.rtc.ru/	ftp.mccme.ru/	www.rtc.ru/	www.rtc.ru/
4	www.nlr.ru/	ftp.rbc.ru	ulmart.com	www.nlr.ru/
5	citforum.ru/	lpc1.stu.neva.ru	citforum.ru/	citforum.ru/
6	lenta.ru/	ftp://ftp.intel.com/	lenta.ru/	lenta.ru/
7	www.netasq.com	ftp.dlink.ru/	netasq.com	netasq.com
8	rfc.com.ru/	ftp.neva.ru	rfc.com.ru/	rfc.com.ru/
9	mirwifi.org/	ftp://ftp.stat.duke.edu/	mirwifi.org/	mirwifi.org/
10	www.neva.ru/	ftp.rbc.ru	www.neva.ru/	www.neva.ru/
11	forum.ru-board.com	lpc1.stu.neva.ru	yandex.ru	yandex.ru
12	edition.cnn.com/	wpc7.stu.neva.ru	edition.cnn.com/	edition.cnn.com/
13	www.vgd.ru	ftp.dlink.ru	www.vgd.ru	www.vgd.ru
14	www.podgourski.net	ftp.neva.ru	www.podgourski.net	www.podgourski.net
15	helsinki-vantaa.fi	ftp://ftp.intel.com/	rumbler.ru	rumbler.ru