МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Юго-Западный государственный университет»
(ЮЗГУ)
Кафедра информационной безопасности

УТВЕРЖДАЮ
Проректор по учебной работе
_____О.Г. Локтионова
«___» _____2016 г.

ЛАБОРАТОРНАЯ РАБОТА № 4
«Исследование сетевых возможностей ОС Linux»

Методические указания по выполнению лабораторных и практических работ по дисциплинам «Администрирование вычислительных систем», «Администрирование вычислительных сетей» для студентов специальностей и направлений подготовки 10.05.02, 10.05.03, 10.03.01, 10.04.01.

Курск 2016

УДК 004
Составители: В.В. Гефнер, И.В. Калуцкий

Рецензент
Кандидат технических наук, доцент кафедры
защиты информации и систем связи *А.Г. Спеваков*

Исследование сетевых возможностей ОС Linux: методические указания к выполнению лабораторных и практических работ по дисциплинам: «Администрирование вычислительных систем», «Администрирование вычислительных сетей» / Юго-Зап. гос. ун-т; сост.: В.В. Гефнер, И.В. Калуцкий, Курск, 2016. 23 с.: ил. нет, Библиогр.: с. 23

Содержат сведения по вопросам сетевых возможностей в ОС GNU/Linux.

Указывается порядок выполнения лабораторных и практических работ, правила оформления, содержание отчета.

Методические указания соответствуют требованиям программы, утвержденной учебно-методическим объединением по специальностям и направлениям подготовки «Комплексная защита объектов информатизации», «Информационная безопасность», «Информационная безопасность автоматизированных систем».

Методические указания по выполнению лабораторных и практических работ по дисциплинам «Администрирование вычислительных систем», «Администрирование вычислительных сетей» для студентов специальностей и направлений подготовки 10.05.02, 10.05.03, 10.03.01, 10.04.01.дневной формы обучения.

Текст печатается в авторской редакции
Подписано в печать . 31.05.16 Формат 60x84 1/16.
Усл. печ. л. 1,2. Уч. –изд.л. 1,1. Тираж 30 экз. Заказ 590 Бесплатно.
Юго-Западный государственный университет.
305040, г. Курск, ул. 50 лет Октября, 94.

Содержание

Цель работы

Цель лабораторной работы – знакомство с сетевыми возможностями операционной системы GNU/Linux.

Порядок выполнения работы

1. Изучить теоретическую часть.
2. Выполнить задания, поставленные в данном методическом указании.
3. Сделать вывод по проделанной работе.

Содержание отчёта

1. Титульный лист.
2. Задание на лабораторную работу.
3. Ход выполнения лабораторной работы со скриншотами.
4. Вывод по лабораторной работе.

Теоретическая часть

1 Сетевые возможности операционных систем Linux

Операционные системы UNIX развивались одновременно с вычислительными сетями. Включение ЭВМ в компьютерную сеть многократно увеличивает как функциональные возможности пользователя, так и степень уязвимости системы и обрабатываемой информации по отношению к сетевым атакам. Сетевые возможности операционных систем должны быть безопасными, однако это требование гораздо легче провозгласить, чем обеспечить.

Искусство системного администратора во многом определяется его умением правильно построить и грамотно эксплуатировать вычислительную сеть. Для этого операционные системы Linux располагают самыми подходящими возможностями. Под управлением ОС Linux надежно работают и серверные приложения, и межсетевые экраны, и системы обнаружения компьютерных атак.

1.1 Контроль и настройка сетевых интерфейсов

Сетевой адаптер – это программно управляемое устройство, благодаря которому персональный компьютер или сервер превращается в интеллектуальный приемопередатчик и приобретает возможность обмена информацией с другими

компьютерами в локальной вычислительной сети. Сетевой адаптер можно использовать как устройство перехвата всех или фильтрации определенных пакетов. Компьютер с двумя сетевыми адаптерами может служить транслятором (мост, шлюз, фильтр) между двумя различными ЛВС.

Штатная утилита **ifconfig** используется для настройки любых сетевых устройств, подключенных к компьютеру, а также для получения справочной информации о состоянии и работоспособности каждого из них. Для настройки и диагностики беспроводных адаптеров Wi-Fi служит другая утилита, именуемая **iwconfig**. Но и в случае работы в беспроводной сети возможности утилиты **ifconfig** остаются востребованными.

Для получения текущей информации о состоянии сетевых интерфей-сов, в том числе и неактивных, используется команда **ifconfig –a**. Для читателя, знакомого с принципами функционирования компьютерных сетей и их терминологией, выведенная информация должна быть понятна. Утилита отображает информацию о состоянии двух физических сетевых интерфейсов: проводного **eth0**, беспроводного **ath0**, а также одного виртуального **lo**.

Для того чтобы перепрограммировать MAC-адрес, необходимо вначале отключить сетевой интерфейс от стека протоколов. Делается это с помощью команды **ifconfig eth0 down**

Затем вводится командная строка, изменяющая аппаратный адрес **ifconfig eth0 hw ether 01:02:03:04:05:06**

Наконец, адаптер вновь встраивается в стек сетевых протоколов **ifconfig eth0 up**

Вводя команду **ifconfig eth0**, нетрудно убедиться, что адрес изменен, а интерфейс активен (**up**).

Задать или изменить IP-адрес еще проще. Для этого достаточно ввести команду

ifconfig eth0 192.168.0.1 netmask 255.255.255.0

При установке или смене IP-адресов отключать и затем включать интерфейс не требуется. Маску сети можно опустить, и она будет введена по умолчанию (предполагается, что это сеть класса С).

При необходимости одному физическому адаптеру можно поставить в соответствие несколько IP-адресов (сколько именно – выяснить не удалось). Делается это с помощью любой из двух команд: **ifconfig eth0:1 192.168.0.2 ifconfig eth0 add 192.168.0.2**

Таким образом, на одном компьютере можно создать небольшую виртуальную сеть. Например, используя два адреса для сетевого обмена, можно имитировать трафик, а с третьего адреса запустить анализатор пакетов для перехвата трафика.

Следует обратить внимание на индикаторы **UP** и **RUNNING** , которые отображают состояние сетевого интерфейса. Индикатор **UP** означает, что адаптер работает в стеке сетевых протоколов в составе компьютера. Индикатор **RUNNING** указывает на подключение к сети и режим сетевого обмена. Если извлечь из

адаптера сетевой кабель, то надпись **RUNNING** не будет отображаться.

Для того чтобы сделать узел недоступным и относительно невидимым в ЛВС, можно отключить ARP-отклик. Делается это с помощью команды **ifconfig eth0 -arp**

После этого всякое участие компьютера в сетевом обмене становится невозможным. Предполагается, что компьютер станет невидимым из локальной сети, поскольку не будет отвечать на протокольные запросы о соответствии адресов. Но компьютер может быть физически подключен к сети и при этом не иметь установленного IP-адреса.

Эксплуатация беспроводных сетей Wi-Fi вызвала необходимость в еще одной утилите, получившей название **iwconfig** с помощью этой утилиты производится установка необходимых параметров.

Некоторые беспроводные адаптеры позволяют производить конфигурацию нескольких виртуальных интерфейсов на базе одного физического устройства. Так, с помощью одной команды можно эмулировать несколько беспроводных устройств различного типа на базе одного сетевого адаптера **wifi0**.

wlanconfig ath create wlandev wifi0 wlanmode

<virt_dev>

Виртуальные адаптеры могут работать в режиме точки доступа (**AP - access point**), сетевого адаптера в одноранговой сети

(**adhoc**), монитора (**monitor**) и др. Однако одновременно заставлять работать один физический радиопередатчик в несовместимых режимах нельзя. Так, невозможно одновременно эмулировать работу точки доступа и обычной точки ad-hoc.

1.2 Разведка сети

Прежде чем отправить сообщение, установить сеанс связи, требуется узнать сетевой адрес абонента, убедиться в наличии и активности сетевого узла и нужной сетевой службы. Для получения этой информации написано и успользуется множество утилит. В некоторых случаях получение информации о доступности сетей, узлов и протоколов транспортного уровня является прелюдией к сетевой атаке.

Наиболее простая и известная команда **ping** использует специальный протокол **icmp** и служит для зондирования эхо-запросами сетевых узлов для установления их наличия и доступности.

ping <параметры> <адрес_хоста> <номер_порта>

<параметры> (в зависимости от типа ОС могут использоваться иные символы)**:**

-l count или **-c count** – отправка указанного числа пакетов. По умолчанию (в зависимости от версии ОС) посылается либо один пакет, либо бесконечная серия пакетов с интервалом в одну секунду. Непрерывная отправка прерывается нажатием **Ctrl – C**,

-s count_byte – общее количество байтов в **icmp**-пакете с эхозапросом (длина заголовка **icmp**-пакета – 8 байтов),

-i timeout – временной интервал в следовании пакетов в секундах, **-f** – направление пакетов с максимально возможной скоростью (только с

правами **root**),

<адрес_хоста> – доменное имя или IP – адрес целевого компьютера, **<номер_порта>** – номер, закрепленный за сетевой службой, запущенной на удаленном компьютере (смотри файл **/etc/services**).

Например, команда

ping –c 3 –i 5 192.168.1.2 21

направляет 3 стандартных **icmp**-пакета с пятисекундным интервалом в адрес FTP-сервера (порт 21) на узле с IP-адресом 192.168.1.2.

Утилита выводит данные построчно в следующем порядке: число байтов в принятом пакете, IP-адрес исследуемого узла, порядковый номер пакета, счетчик «жизни» пакета и время возврата.

Более сложным инструментом для сетевой разведки является утилита **nmap** (netmap – карта сети). Она использует девять различных видов сканирования сетевых узлов. Принципы сканирования основаны на передаче в адрес интересующего узла сетевых пакетов с определенным «наполнением» и анализом

отклика. При этом используются особенности в реализации стека протоколов TCP/IP, присущие известным операционным системам и сетевым службам. Направляемые пакеты могут имитировать процесс установления или завершения сеанса, направление дейтаграммы, различные ошибочные ситуации и др.

Команда для сетевого сканирования выглядит так:

nmap <тип_сканирования> <параметры> <список узлов или сетей>

<тип_сканирования>

-sT – обычное TCP-сканирование с установлением соединения. Используется по умолчанию и может запускаться обычным пользователем,

-sP – обычное ping-сканирование,

-sS – TCP-сканирование с помощью сообщений SYN. Утилита инициирует установление TCP–сеанса, отправляя в адрес узел:порт первый пакет с установленным битом SYN. Адресат отвечает пакетом с установленными битами SYN и ACK, чем обозначает себя. Но вместо согласия на установление соединения утилита посылает пакет с установленным битом RST, чем разрывает соединение. Считается наилучшим из методов TCPсканирования,

-sU – UDP-сканирование, при котором в адрес каждого порта направляется пустой UDP–пакет. Если порт закрыт, адресат

отправляет клиенту пакет с установленным битом RST. Если порт открыт, он принимает пакет без ответа,

-sF – FIN-сканирование. Направляется пакет, сигнализирующий о разрыве соединения TCP (которое еще не было установлено). Если указанный порт закрыт, система отвечает пакетом с установленным битом RST, если открыт – пакет не направляется (кроме ОС Windows*),

-sN – нуль-сканирование. Направляется пакет, в котором не установлено ни одного битового флага. Результат аналогичен FIN-сканированию.

<параметры>

-O – режим изучения откликов для определения типа удаленной операционной системы. Большинство ОС обладают своей спецификой при управлении сетевыми протоколами. Для установления типа ОС программа посылает определенные пакеты в адреса конкретных портов и фиксирует реакцию на них,

-p <диапазон> – диапазон портов, которые будут сканироваться (указываются через запятую или дефис),

-v (**–vv**) – режим вывода подробной информации,

-T <число> – темп сканирования от «0» – очень медленно (один пакет в пять секунд) до «5» – максимально быстро (один пакет за 0.3 секунды).

<список узлов или сетей>

Доменные имена в списке указываются через запятую. Диапазон IP– адресов указывается в виде номера сети и сетевой маски, например 192.168.1.00/24. Любое число можно заменить символом звездочки *. Диапазон адресов в любом из октетов можно указывать в виде начального и конечного значения, через дефис, например 1–24. Наконец, нужные числовые значения можно указать через запятую, без пробела, например
192.168.2.3,7,17,24.

1.3 Перехват и анализ сетевого трафика

Утилита **tcpdump** является мощнейшим средством перехвата и анализа сетевого трафика. Эта универсальная утилита для прослушивания моноканала присутствует почти во всех дистрибутивах Linux, а для ее запуска необходимы права суперпользователя. Команда автоматически переводит сетевой адаптер в режим захвата всех пакетов в моноканале, но отображает только отфильтрованные пакеты.

tcpdump <параметры> <параметры_фильтрации>

Утилиту можно запустить без аргументов с помощью одноименной команды. В этом случае один (или единственный) сетевой интерфейс переводится в режим беспорядочного захвата пакетов, а текстовая информация о заголовках перехваченных пакетов выводится на экран. Это не очень удобно, так как приход каждого нового пакета сопровождается, как минимум, одной новой

строкой, а при их обилии продуктивное чтение и анализ трафика становятся невозможными.

Утилита очень богата возможностями, и в ее командной строке предусмотрено несколько десятков параметров. Рассмотрим наиболее важные из них.

1. С помощью параметра **-w <имя_файла>** производится запись перехваченной информации в файл специального формата. Прочитать такой файл с выводом информации на экран можно только с помощью **tcpdump**, задав для этого аргумент **-r <имя_файла>**. В то же время отфильтрованные утилитой данные можно сохранить в обычном текстовом файле, используя перенаправление вывода «>».

2. По умолчанию в каждом пакете захватывается для анализа 68 байтов. Почему выбрано именно это число? Заголовок канального уровня

(Ethernet–кадр) состоит из 14 байтов. Минимальные размеры заголовков IP и TCP пакетов составляют по 20 байтов. Еще 14 байтов отводится для распознавания инкапсулированного пакета прикладного уровня. Для явного задания длины «отрезаемой» для анализа части пакета в байтах служит аргумент **-s <длина_пакета>**. В случае необходимости перехвата всего пакета (это может посягать на конфиденциальность передаваемых данных!) его длина задается равной 1514 байтов (14 байтов заголовка кадра Ethernet + 1500 байтов как максимальный размер вложимого кадра).

3. Число перехваченных пакетов можно ограничить путем задания аргумента **-c <число_пакетов>** с завершением работы после выполнения задания.

4. При наличии в составе компьютера нескольких сетевых интерфейсов аргумент **-i <интерфейс>** позволяет определить тип сетевого адаптера (например, **–i eth1**) или модема (**–i ppp1**), с помощью которого производится перехват пакетов. Если физические интерфейсы будут заняты в сетевом обмене, для перехвата данных можно задействовать логический интерфейс обратной петли **lo**.

5. По умолчанию заголовок канального уровня не перехватывается, и внешним является IP–пакет. Для перехвата заголовка кадра Ethernet с MAC–адресами передатчика и приемника необходимо указать параметр **-e**.

6. Для вывода более подробной текстовой информации можно воспользоваться аргументами **-v, -vv, -vvv**. Стандартный формат текстовой строки анализатора может включать следующие поля:

- отметку времени перехвата, в которой три пары цифр, разделенных двоеточиями, указывают часы, минуты и секунды, а последние шесть цифр – дробную часть секунды,
- доменное имя или IP–адрес хоста–отправителя,
- номер порта получателя,

• обозначение установленных битовых флагов TCP–заголовка, которые несут информацию об этапе в установлении сеанса,

• начальный и конечный (через двоеточие) порядковые номера TCP– сегмента, а также (в скобках) – число переданных байт, размер TCP–окна в байтах.

7. Для отображения заголовков и содержимого пакетов в шестнадцатеричном коде служат аргументы **-x (–xx)**. Если возникает потребность в отображении содержимого пакетов в шестнадцатеричных и ASCII– кодах, можно воспользоваться аргументом **–X**.

<параметры фильтрации> используют несколько ключевых слов:

• протокол **(proto)** – указывает, какие именно пакеты подлежат перехвату. Среди часто используемых можно указывать ключевые слова: **ether, ip, arp, rarp, tcp, udp, icmp, ip6**,

• направление – указывает источники и получателей сообщений: **src**

(source – источник), **dst** (destination – получатель) или их комбинации: **src or dst** или **src and dst**,

• объекты прослушивания, к которым могут относиться:

host (номер или имя) – сетевой узел, являющийся источником или получателем сообщений,

net (сетевая часть адреса) – локальная сеть или ее часть,

port – номер или символическое обозначение службы, указанной в таблице **/etc/services**.

В параметры фильтрации могут входить математические выражения. Например, **‘ip[6:2] & 0x1FFF == 0’** условия фильтрации выполняются, если результат побитового логического умножения 6-го и 7-го байтов заголовка пакета с маской **0x1FFF** равен нулю. Ключевые слова и математические выражения могут объединяться с использованием логических условий: **not, and** и **or**.

Выполнение работы

1. Изучите теоретический материал, изложенный в методических указаниях.

2. Зарегистрируйтесь в системе в первой консоли с правами администратора.

3. Зарегистрируйтесь во второй консоли с правами пользователя.

4. Из консоли администратора с помощью команды **ifconfig -a** выведите на экран данные о текущем состоянии всех сетевых интерфейсов компьютера. Какую информацию из прочитанного вывода вы извлекли? Запомните, как обозначается основной Ethernet-адаптер (он может обозначаться **eth0, eth1, eth2**), и в дальнейшем используйте в сетевых командах это имя. Далее в тексте задания он упоминается как **eth0**.

5. Выведите информацию о сетевых интерфейсах с помощью команды **netstat –ai**, сравните возможности двух использованных утилит.

6. Активизируйте отключенную по умолчанию сетевую службу **telnet server**. Для этого запустите **Midnight Commander**, найдите конфигурационный файл **/etc/inetd.conf** и в режиме редактирования (F4) удалите символ комментария # перед строкой **telnet stream tcp nowait**, после чего сохраните изменения в файле. Если в системе используется демон **xinetd**, то активизация протокола

производится в конфигурационном файле **/etc/xinet.d/telnet**, строка **disable = no**. Затем следует перезапустить систему.

7. Командой **ps –ef | more**

выведите список процессов и убедитесь, что сетевой процесс **inetd** работает. Иначе его нужно запустить вручную командой **inetd**. Если исследуемая версия ОС не содержит сервера **telnet** (по причине его явной уязвимости некоторые версии Linux не предусматривают использования этого протокола), соответствующие пункты задания выполните с защищенной программной оболочкой Secure Shell (**SSH**).

8. Отключите сетевой адаптер командой **ifconfig eth0 down** (см. справку по сетевым командам). Присвойте сетевому интерфейсу временный MAC-адрес **A0:B1:C2:D3:E4:N**, где **N** – двузначный номер компьютера в классе (при использовании в ЛВС одинаковых аппаратных или сетевых адресов возможны коллизии). Подключите адаптер к сети. Убедитесь в том, что его аппаратный адрес изменен.

9. Назначьте основному сетевому интерфейсу компьютера временный IPадрес и маску подсети. Для этого введите команду

ifconfig eth0 192.168.0.N netmask 255.255.255.0 , где **N** – номер компьютера. Повторным вводом команды **ifconfig eth0** убедитесь в том, что запись введенной информации произведена. Присвоенные сетевые адреса будут действовать до перезагрузки компьютера.

10. Проверьте работоспособность петли обратной связи, послав на свой же компьютер эхо-запрос **ping 127.0.0.1**. Убедившись,

что отклики поступают, остановите зондирование комбинацией клавиш **Ctrl-C**.

11. Присвойте сетевому адаптеру дополнительный IP-адрес **192.168.0.20+N**, где **N** – номер компьютера. Проверьте прохождение ICMP-пакетов между сетевыми адресами на локальном компьютере.

12. Организуйте сеанс **telnet** на собственном компьютере, используя для этого интерфейс обратной петли или дополнительный IP-адрес. Для этого перейдите в консоль пользователя, наберите команду **telnet 127.0.0.1** (или **telnet localhost**) и после сообщения об успешном соединении введите **login** и пароль администратора. Почему вам было отказано в доступе? Почему соединение было закрыто? Можно ли считать эти меры надежной защитой, пресекающей передачу опасной информации по каналу связи в открытом виде?

13. Еще раз установите сеанс **telnet** через петлю обратной связи, используя на этот раз учетную запись обычного пользователя. После установления сеанса просмотрите список каталогов и файлов в нескольких директориях, список процессов и убедитесь, что в «удаленном» режиме доступа вы можете выполнять все команды, которые доступны пользователю, зарегистрированному на удаленном узле.

14. Перейдите в консоль администратора и с помощью команды **w** или **who** посмотрите, сколько сейчас пользователей в системе, кто они и с каких терминалов работают. Обратите внимание на то, как обозначаются локальный и удаленный терминалы.

15. С помощью команды **netstat –a** проконтролируйте список запущенных сервисов и их состояние. Найдите сеанс **telnet**.

16. Вернитесь в консоль пользователя и завершите локальный сеанс **telnet** командой **exit**. Получите сообщение о закрытии сетевого соединения. Проверьте эту информацию с помощью команды **netstat**.

17. Попробуйте войти на один из компьютеров сети в сеансе **telnet** (учетные записи пользователей на всех компьютерах должны быть одинаковы). Что потенциально опасного вы можете сделать на удаленном компьютере? Приобретите на удаленном хосте права **root**. Проверив свои возможности по манипуляции удаленным компьютером, завершите сеанс командой **exit**.

18. Поскольку на вашем компьютере **telnet**-сервер активизирован, он тоже может стать объектом доступа. Периодически с помощью команд **netstat –a, ps –ef** или **w** проверяйте, не зафиксировали ли они подозрительные соединения, процессы или удаленных пользователей.

19. Войдите на произвольный узел по протоколу Secure Shell (команда **ssh** с указанием IP-адреса хоста, после запроса необходимо ввести пароль администратора). Проверьте свои возможности по манипуляции удаленным компьютером.

20. С помощью команды **arp –a** посмотрите таблицу соответствия сетевых и аппаратных адресов. Где расположен ARP-кэш и почему он сейчас пуст?

21. С помощью утилиты **ping** исследуйте локальную сеть, к которой подключен ваш компьютер в диапазоне адресов,

идентифицирующих конкретный компьютер в сети, от 1 до 40 (**192.168.0.1/40**). Объясните, в чем уязвимость и неудобство такого метода сканирования.

22. Проверьте, обновилась ли после сканирования динамическая ARPтаблица. Если она содержит нужную вам информацию о сети, ее можно сохранить в файле командой **arp –a > /home/arp1** (через несколько минут информация о сетевых узлах будет изменена, и если тот или иной сетевой узел не проявляет активности, данные о нем в кэше будут утрачены). С помощью команды

arp –s <IP-адрес> <MAC-адрес> создайте статическую **arp**-таблицу. Выясните местоположение этой таблицы.

23. Ознакомьтесь с синтаксисом команды **nmap** (netmap – карта сети). С помощью утилиты **nmap** исследуйте локальную сеть, к которой подключен ваш компьютер. Адреса в диапазоне можно вводить с использованием символов-джокеров и через дефис. Что вы можете сказать о полученной информации?

24. Исследуйте различные виды сканирования с помощью утилиты **nmap**. На этот раз в качестве объекта выберите один из компьютеров. Типы сканирования перечислены в справке по команде **nmap.** Какую дополнительную информацию о локальной сети вы получили? Каким образом эта утилита определяет тип операционной системы, управляющей сетевым узлом?

25. Отключите ответы своего сетевого адаптера на ARP-запросы других хостов командой

ifconfig eth0 –arp

С помощью команды **ifconfig eth0** убедитесь, что настройка выполнена.

26. Выждите несколько минут для сброса ARP-таблиц на компьютерах локальной сети и с одного из компьютеров сети с помощью утилиты **ping** постарайтесь обнаружить отклик своего компьютера. Достаточно ли надежно защищает компьютер от сканирования данная мера? Насколько нарушается при этом возможность работы в сети? Включите **arp**отклик командой **ifconfig eth0 arp**.

27. Отключите сетевой интерфейс на своем компьютере с помощью команды **ifconfig eth0 down**. Повторите попытку обнаружения своего компьютера с одного из соседних узлов. Можете ли вы сами при этом проявлять какую-либо сетевую активность? Сделайте выводы. Вновь включите сетевой адаптер с помощью команды **ifconfig eth0 up**.

28. Переведите сетевой адаптер своего компьютера в режим перехвата всех пакетов с помощью команды **ifconfig eth0 promisc**. С помощью команды **ifconfig eth0** убедитесь, что настройка выполнена. При этом сетевой адаптер превращается в устройство подслушивания, но одновременно он становится очень уязвимым к сетевым атакам на отказ в обслуживании.

29. Ознакомьтесь с синтаксисом команды **tcpdump**.

30. С помощью утилиты **tcpdump** перехватите и прочитайте сетевые пакеты:

- отправленные из одного определенного адреса,
- являющиеся результатом сетевого обмена между двумя хостами,
- только Ethernet и IP-заголовки пакетов, направленных в адрес любого из компьютеров сети,
- сеансы **telnet** и **ssh** в локальной сети, ICMP-запросы в адрес вашего хоста,
- иные пакеты по указанию преподавателя.

31. Запишите несколько перехваченных пакетов в файл и просмотрите их в шестнадцатеричном коде. Найдите характерные поля и идентификаторы в заголовках канального, сетевого и транспортного уровней. Определите аппаратные и сетевые адреса, номера портов, иные характерные признаки, идентифицирующие сетевые протоколы.

32. Выясните, можно ли использовать **tcpdump** на сетевом интерфейсе, за которым не закреплен ни один IP-адрес.

33. Используя виртуальные сетевые адреса на локальном компьютере, организуйте сеанс сетевого копирования и канал наблюдения за ним. Для этого потребуется работа с трех консолей с правами администратора.

34. В первой консоли подготовьте (но пока не вводите!) команду для копирования (передачи) небольшого текстового файла, например файла паролей:

cat /etc/passwd | nc –w 2 192.168.0.22 3333

35. Во второй консоли подготовьте команду для приема копируемого файла: **nc –l –p 3333 > /home/password1**

36. В третьей консоли подготовьте команду для перехвата копируемой информации:

tcpdump –i lo –xx –vv –s 100 > /home/password2 37. После проверки синтаксиса команд произведите их поочередный запуск: вначале **tcpdump** из третьей консоли, затем команду приема данных из второй консоли и, наконец, команду передачи данных. Дождитесь завершения команд в первой и второй консолях и затем комбинацией клавиш **Ctrl+C** остановите сеанс прослушивания **tcpdump**.

38. С помощью команды **cat** или **mcedit** просмотрите результаты копирования и перехвата. Обратите внимание на то, что **tcpdump** перехватил, по меньшей мере, семь пакетов, из которых три первых и три последних предназначались для установления и завершения TCP-сеанса. По этой причине большой объем копируемых данных должен представлять собой один непрерывный поток. Для этого рекомендуется использовать утилиту блочного копирования **dd** или утилиту **tar** (см. Справочник по командам Linux). Сравните между собой содержимое скопированного и перехваченного файлов.

39. На двух произвольно выбранных компьютерах в ЛВС с моноканалом произведите копирование большого массива данных. Предварительно рекомендуется произвести контроль установленных по умолчанию параметров фиксированного жесткого диска. В

отношении HDD с IDEинтерфейсом для этого рекомендуется использовать команду **hdparm**. Для хронометража процедуры копирования рекомендуется выполнить совместно с утилитой **time**. Если результаты копирования некуда записывать, перенаправьте вывод в нулевое устройство.

40. На первом компьютере следует запустить команду

time dd if=/dev/hda count=10000|nc –w 2 192.168.0.22 3333

41. На втором компьютере следует запустить команду **time nc –l –p 3333 | dd of=/dev/null**

42. На любом из компьютеров для контроля запустите программу **tcpdump**. **tcpdump –i lo –xx –vv –c 10 –s 100 > /home/hda**

43. Оцените скорость копирования.

44. Выполните файловое копирование с помощью команд **tar** и **nc**. Команда **tar** используется для создания в качестве объекта копирования одного большого (здесь уместнее сказать – длинного) файла. **tar –czvf /home | nc –w 2 192.168.0.22 3333**

Контрольные вопросы

1. Как закрепить за одним сетевым адаптером несколько IP–адресов?
2. Как программным путем изменить аппаратный адрес сетевой карты?
3. Можно ли перехватывать трафик без установленного IP–адреса?
4. Для чего нужно отключать ARP–отклик?
5. Где находится ARP–кэш? Как долго хранятся в нем данные?
6. Перечислите известные вам виды сетевого сканирования.
7. Запишите команду перехвата шести **icmp**–пакетов, исходящих из узла с IP–адресом 192.168.0.3 .
8. Как производится сетевое копирование данных?

Библиографический список

1. Техническая электронная документация по операционной системе Linux.

2. Береснев А.Л. Администрирование GNU/Linux с нуля./А.Л. Береснев –СПб.: БВХ-Петербург, 2010 – 576 с.

3. Блум, Ричард, Бреснахэн, Кристина. Командная строка Linux и сценарии оболочки. Библия пользователя/ Ричард Блум, Кристина Бреснахэн -М. : ООО “И.Д. Вильямс”, 2012. — 784 с.

4. В.В. Бакланов Защитные механизмы операционной системы Linux: учебное пособие / В.В. Бакланов. под ред. Н.А. Гайдамакина. Екатеринбург: УрФУ, 2011. 354 с.