**Лабораторная работа № 12**

**Тема:** сетевые утилиты Linux.

**Цель:**изучить утилиты для захвата сетевого трафика, научиться анализировать захваченные пакеты, извлекать из них полезную информацию.

**Краткие теоретические сведения**

Утилита TCPDump представляет собой мощный и популярный инструмент для перехвата и анализа сетевых пакетов. Она позволяет просматривать все входящие и исходящие из определенного интерфейса пакеты и работает в командной строке. Её использование необходимо в случаях, когда невозможно запустить Wireshark, например, в операционных системах без графической оболочки. Данная утилита часто используется для устранения неполадок в сети, а также для обеспечения безопасности.

Если tcpdump запустить без параметров, он будет выводить информацию обо всех сетевых пакетах. С помощью параметра -i можно указать сетевой интерфейс, с которого следует принимать данные, например:

sudo tcpdump -i enp0s8

или

sudo tcpdump -i any

Чтобы посмотреть названия всех доступных для захвата интерфейсов, нужно запустить TCPDump с ключом -D.

Для более точной настройки работы TCPDump можно использовать один из перечисленных далее параметров, или их комбинацию.

* выводить сокращённую информацию: -q;
* выводить подробную информацию: -v -vv -vvv (чем больше v, тем подробнее информация);
* не преобразовывать порты: -n (вместо названий, например, http или ftp будут показываться номера, для указанных протоколов: 80 и 21);
* не преобразовывать имена: -nn (вместо имён компьютеров, например, UbuntuServer1804 будут показываться их IP-адреса);
* показывать MAC-адреса: -e;
* захватить только определённое количество пакетов: -с количество;

Для протокола, чьи пакеты должны захватываться, достаточно указать его название, например:

tcpdump icmp

или

tcpdump udp port 53

Параметр port позволяет ограничить для захватываемого трафика порт или их диапазон. Данный параметр применим только к протоколам TCP и UDP.

По умолчанию TCPDump выводит информацию о захваченных пакетах на экран, однако её можно записывать в файл. Для это используется параметр -w:

tcpdump -w /home/student/1.cap

Полученный в результате файл имеет формат, совместимый с Wireshark и может быть в дальнейшем проанализирован в данной программе с использованием более наглядного графического интерфейса, а также широких возможностей по созданию фильтров и подсчёту статистики.

Ещё одна полезная сетевая утилита — NMap («Network Mapper») предназначена для исследования сети и проверки безопасности. Результатом её работы является список просканированных целей с дополнительной информацией по каждой из них в зависимости от заданных опций. Ключевой информацией является «таблица важных портов».

Эта таблица содержит номер порта, протокол, имя службы и состояние. Состояние может иметь значение open (открыт), filtered (фильтруется), closed (закрыт) или unfiltered (не фильтруется). Открыт означает, что приложение на целевой машине готово для установки соединения/принятия пакетов на этот порт. Фильтруется означает, что брандмауэр, сетевой фильтр, или какая-то другая помеха в сети блокирует порт, и Nmap не может установить открыт этот порт или закрыт. Закрытые порты не связаны ни с каким приложением, но могут быть открыты в любой момент. Порты расцениваются как не фильтрованные, когда они отвечают на запросы Nmap, но Nmap не может определить открыты они или закрыты. Nmap выдает комбинации открыт|фильтруется и закрыт|фильтруется, когда не может определить, какое из этих двух состояний описывает порт. Эта таблица также может предоставлять детали о версии программного обеспечения, если это было указано при запуске сканирования.

Использование NMap заключается в выполнении команды:

nmap [<Тип сканирования>] [<Опции>] {<цель сканирования>}

Тип сканирования определяет используемые пакеты и подход к интерпретации результата. Для протокола TCP можно задать один или несколько из следующих типов сканирования:

-sS — TCP SYN;

-sT — TCP connect;

-sN; -sF; -sX — NULL, FIN и Xmas;

-sA — TCP ACK;

-sW — TCP Window;

-sM — TCP сканирование Мэймона.

Подробнее про их различие можно прочитать в официальной русскоязычной документации (<https://nmap.org/man/ru/man-port-scanning-techniques.html>). Чем больше типов сканирования выбрано при запуске команды, тем дольше оно будет осуществляться.

Для сканирования UDP-портов необходимо использовать параметр -sU.

По умолчанию NMap проверяет только первую тысячу портов (от 1 до 1000). При необходимости изменить данный диапазон можно использовать параметр -p, например:

nmap -sU -p 520 172.16.0.1-254

В представленной команде указывается сканировать только UDP-порт 520 на всех узлах с адресами от 172.16.0.1 до 172.16.0.254.

Возможен и противоположный вариант, например, сканировать все TCP-порты от 1 до 65535 на одном узле с адресом 10.0.0.1:

nmap -sS -p 1-65535 10.0.0.1

Последний параметр называется целью сканирования и определяет множество адресов, для которых должно выполняться сканирование. В нём допустимо указывать:

* единичный IP адрес: nmap 192.168.1.1
* доменное имя узла: nmap donstu.local
* множество IP-адресов: nmap 192.168.1.1 192.168.1.2 192.168.1.3
* диапазон IP-адресов: 192.168.1.1-254
* подсеть: 192.168.1.0/24

NMap имеет множество дополнительных опций, обеспечивающих широкие возможности сбора информации о сети. Среди них:

-sV — детальное исследование портов для определения версий служб;

-O — определять операционную систему.

Для упрощения использования NMap для неё существует официальный графический интерфейс — ZenMap.

**Задание**

1. TCPDump.

1.1. С помощью утилиты TCPDump захватите и выведите на экран информацию из входящих пакетов протоколов HTTP и ICMP.

1.2. С помощью утилиты TCPDump захватите и запишите в файл трафик, передаваемый виртуальной машиной с Ubuntu Server в процессе обращения к ней по протоколу HTPP и выполнения команды ping.

1.3. Передайте этот трафик на виртуальную машину с десктопной версией Linux и установленной программой Wireshark.

2. Wireshark

2.1. С использованием Wireshark выделите из общего трафика отдельно HTTP и отдельно ICMP-пакеты.

2.2. Определите все протоколы, чьи пакеты присутствуют в захваченном трафике.

2.3. Определите все MAC-адреса, которые присутствуют в захваченном трафике.

2.4. Определите все IP-адреса, которые присутствуют в захваченном трафике.

2.5. Определите все запрашиваемые доменные имена.

3. NMap

3.1 Просканируйте какую-либо сеть (реальную или виртуальную) с целью определить операционные системы оборудования, входящего в неё, наличие других сетевых сервисов, используемые служебные сетевые протоколы (например DHCP или протоколы маршрутизации).

**Отчёт по лабораторной работе должен содержать:**

1. Титульный лист.
2. Цель, задание.
3. Скриншоты с результатами выполнения каждого пункта задания.

**Контрольные вопросы**

1. Что такое TCPDump?
2. Приведите примеры нескольких фильтров для TCPDump.
3. Как проанализировать трафик, захваченный с помощью TCPDump в Wireshark?
4. Какая статистика может быть получена в Wireshark для сетевого трафика?
5. Что такое NMap и ZenMap?
6. Что является результатом работы NMap/ZenMap?
7. Что означает «открытый», «закрытый» и «фильтруемый» порт?