Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования
«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»

КОМПЛЕКСНАЯ ЗАЩИТА ОБЪЕКТОВ ИНФОРМАТИЗАЦИИ

КНИГА 24

Д.В. МИШИН, Ю.М. МОНАХОВ

АНАЛИЗ ЗАЩИЩЕННОСТИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Идентификация ресурсов корпоративной сети передачи данных

Практикум

Владимир 2012

УДК 930.1
ББК 32.81

Редактор серии – заведующий кафедрой «Информатика и защита информации» доктор технических наук,
профессор М.Ю. Монахов

Рецензенты:
И.о. начальника кафедры специальной техники и информационных технологий Владимирского юридического института УФСИН России
доктор технических наук, профессор
Б.Ю. Житников
Профессор кафедры информационных систем и информационного менеджмента Владимирского государственного университета
доктор технических наук, профессор
Р.И. Макаров

Печатается по решению редакционного совета
Владимирского государственного университета имени Александра Григорьевича и Николая Григорьевича Столетовых

Мишин, Д.В.

Анализ защищенности распределенных информационных систем. Идентификация ресурсов корпоративной сети передачи данных : Практикум / Д.В. Мишин, Ю.М. Монахов ; Владим. Гос. Ун-т. – Владимир : Изд-во Владим. Гос. Ун-та, 2012. – 97с. (Комплексная защита объектов информатизации. Кн. 24). – ISBN .

В пособии рассматриваются теоретические и практические аспекты идентификации ресурсов корпоративных распределенных сетей TCP/IP. Особое внимание уделяется вопросам практического применения свободно распространяемых программных средств идентификации узлов сети, анализа топологии, определения версий сетевых служб и операционных систем.

Для студентов старших курсов высших учебных заведений, магистрантов, слушателей курсов повышения квалификации.

Ил.15. Табл.10. Библиогр.:18 назв.

ISBN

ОСНОВНЫЕ СОКРАЩЕНИЯ

АРМ – автоматизированное рабочее место
БД – база данных
ЗИ — защита информации
ИБ — информационная безопасность
ИС – информационная система
КИС – корпоративная информационная система
КСПД – корпоративная сеть передачи данных
ЛВС – локальная вычислительная сеть
МЭ – межсетевой экран
ОС — операционная система
ПО — программное обеспечение
РИВС — распределенная информационно-вычислительная среда
СОА – система обнаружения атак (IDS)
ARP – Address Resolution Protocol
DHCP - Dynamic Host Configuration Protocol
DNS – Domain Name System
HTTP – HyperText Transfer Protocol
ICMP – Internet Control Message Protocol
IP – Internet Protocol
ISO – International Organization for Standardization
FTP – File Transfer Protocol
NAT – Network Address Translation
NIDS – Network-based Intrusion Detection System
OSI – Open Systems Interconnection
RFC – Request for Comments
SNMP – Simple Network Management Protocol
SMTP – Simple Mail Transfer Protocol
SSH – Secure SHell
SSL – Secure Sockets Layer
TCP – Transmission Control Protocol
TTL – Time to live
UDP – User Datagram Protocol

ВВЕДЕНИЕ

Данное издание предназначено для магистрантов и студентов старших курсов, специализирующихся в вопросах проектирования комплексных систем защиты объектов информатизации, а также для ИТ-специалистов, желающих повысить свою квалификацию. Пособие содержит материалы к занятиям по изучению базового набора операций, составляющих процедуру тестирования на проникновение. Практикум ориентирован на обретение навыков осуществления типовых действий по активному аудиту распределенных информационных систем и обретение навыков использования специализированного программного обеспечения.

Организация занятий с использованием предложенного материала предполагает знание студентами основ вычислительных сетей, наличие опыта работы в операционной системе GNU/Linux, а также знание основ информатики, теории информации, архитектуры вычислительных систем, операционных систем.

Практикум ориентирован на использование специализированного программного обеспечения, программных комплексов и утилит, входящих в состав дистрибутива Backtrack версии 4 или выше. В то же время материал, изложенный в практикуме, может быть легко адаптирован для использования вместе с другими версиями соответствующего программного обеспечения, гарантирующими выполнение сходных функций.

Первая часть практикума содержит девятнадцать практических работ. Каждая работа включает в себя описание изучаемых команд и понятий; упражнения, которые необходимо выполнить в интерактивном диалоге с системой; вопросы для самоконтроля; практическое задание. В предлагаемом издании особое внимание уделяется вопросам обнаружения узлов сети, идентификации статуса портов, а также идентификации сетевых сервисов и приложений.

РАЗДЕЛ I - ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ (HOST DETECTION)

Одна из первостепенных задач при исследовании корпоративной сети (КСПД) - формирование перечня сетевых узлов для последующей работы с ними. При идентификации активных узлов (host detection) КСПД анализируются реакция – ответы узла на отправленные запросы или сообщение об ошибке. Итогом данного этапа является перечень активных узлов КСПД, подлежащих дальнейшему исследованию. Для идентификации таких узлов КСПД могут применяться следующие протоколы стека TCP/IP: ICMP, IP, UDP, TCP, ARP.

Практическая работа №1
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ. ICMP ECHO REQUEST (Утилита PING)

1. Цель работы

Овладеть основами обнаружения (идентификации) активных узлов сети с помощью протокола управляющих сообщений ICMP. Рассмотреть способы обнаружения узлов сети с помощью запроса ICMP Echo Request. Научиться применять на практике программные средства исследования сети.

2. Теоретические сведения. Методические рекомендации

Протокол ICMP

ICMP (Internet Control Message Protocol, RFC-792, RFC-1256) - протокол управляющих сообщений, входящий в стек протоколов TCP/IP. ICMP используется для передачи сообщений об ошибках и других исключительных ситуациях, возникших при передаче данных, например, запрашиваемая услуга недоступна, или узел сети (host), или маршрутизатор (router) не отвечают. Также на ICMP возлагаются некоторые сервисные функции. Сообщения ICMP передаются с

использованием базовых заголовков IP (рис. 1). Первый октет данных дейтаграммы (рис. 1) указывает тип ICMP — значение этого поля определяет формат остальной части дейтаграммы.

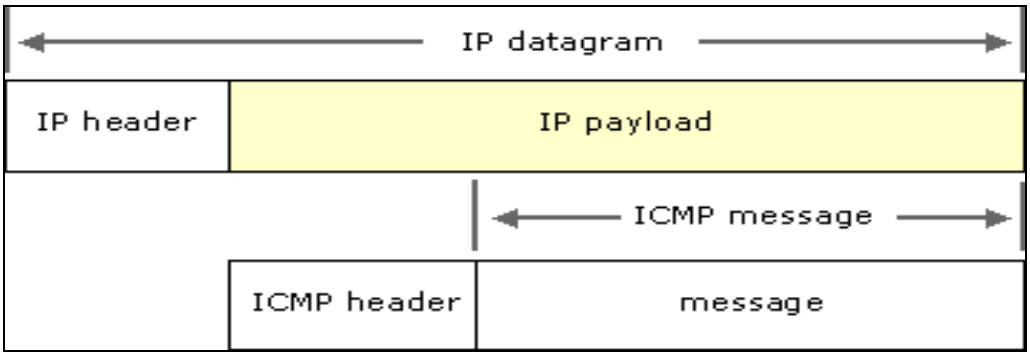


Рис. 1. IP-дейтаграмма

Так как ICMP сообщения переносятся IP-дейтаграммами, их доставка не гарантируется. Типы ICMP-сообщений различаются по заголовку ICMP.

Формат пакета протокола ICMP представлен на рис. 2 (RFC 792).

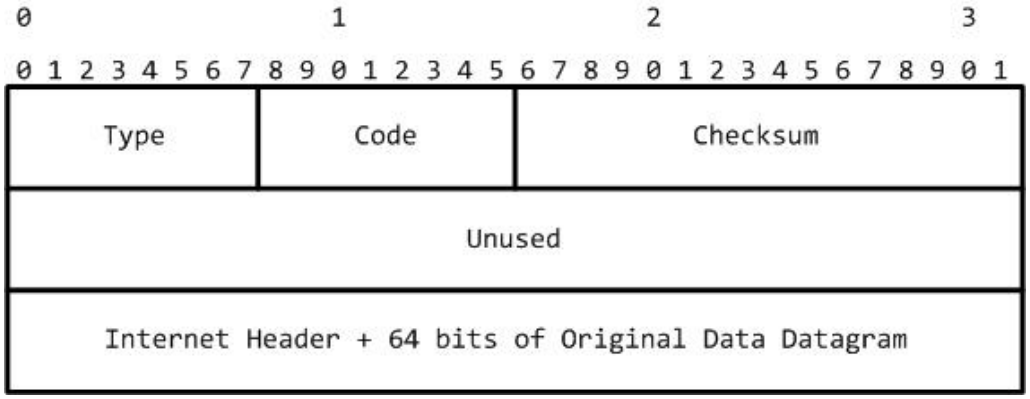


Рис. 2. Формат пакета протокола ICMP

Основные типы сообщений протокола ICMP даны в таблице 1. Для большинства сообщений об ошибках задействовано поле диагностического кода сообщений Destination Unreachable (табл. 2).

Существует два пути идентификации узлов сети средствами протокола управляющих сообщений ICMP:

- Отправка запроса (Echo Request) и ожидание ответа (Echo Reply) от запрашиваемого узла сети;
- Вызов ситуации ошибки на удаленном узле, анализ полученных диагностических сообщений.

Таблица 1 - Основные типы сообщений протокола ICMP

Тип	Сообщение	Назначение
0	Echo Reply	Ответ ICMP - эхо (ping)
1-2		Зарезервировано
3	Destination Unreachable	Получатель недостижим
4	Source Quench	Переполнение очереди источника
5	Redirect	Переадресация / изменение маршрута
7		Зарезервировано
8	Echo Request	Запрос ICMP - эхо (ping)
9		Объявление маршрутизатора (RFC-1256)
10		Запрос маршрутизатора (RFC-1256)
11	Time Exceeded	Превышение временного интервала
12	Parameter Problem	Ошибка в параметрах дейтаграммы
13	Timestamp Request	Запрос метки времени
14	Timestamp Reply	Ответ метки времени
15	Information Request	Запрос информации
16	Information Reply	Ответ информации
17	Address Mask Request	Запрос маски адреса (RFC-950)
18	Address Mask Reply	Ответ на запрос маски адреса (RFC-950)

Таблица 2 - Основные коды сообщений об ошибках протокола ICMP

Код	Тип кода	Значение
0	Network Unreachable	Хост назначения не достижим
1	Host Unreachable	Сеть назначения недостижима
2	Protocol Unreachable	Протокол недостижим
3	Port Unreachable	Порт недостижим
4	Fragmentation Need & DF set	Необходима фрагментация, однако она запрещена
5	Source Route Failed	Исходный маршрут вышел из строя
6	Destination Network Unknown	Сеть назначения неизвестна
7	Destination Host Unknown	Хост назначения неизвестен
8	Source Host Isolated	Источник изолирован
9	Communication with destination Network Administratively Prohibited	Взаимодействие с сетью назначения запрещено
10	Communication with destination Host Administratively Prohibited	Взаимодействие с узлом назначения запрещено
11	Network Unreachable for type of service	Сеть назначения недоступна для запрошенного типа сервиса
12	Host Unreachable for type of service	Хост назначения недоступен для запрошенного типа сервиса
13		Связь запрещена с помощью фильтра
14		Нарушение старшинства ЭВМ
15		Дискриминация по старшинству

Идентификация узла сети способом отправки запроса Echo Request. Утилита Ping

Активная система (узел сети - host), получившая сообщение ICMP Echo Type 8 (Echo Request), должна ответить сообщением ICMP Echo Type 0. Отсутствие ответа Echo Reply может свидетельствовать о недоступности удаленного узла или о фильтрации данного типа трафика, например средствами МЭ.

```
СКАНИРУЮЩИЙ УЗЕЛ (CLIENT) -> ICMP ECHO TYPE 8 (ECHO REQUEST)
                    ICMP ECHO TYPE 0 (ECHO REPLY) <- СКАНИРУЕМЫЙ УЗЕЛ (SERVER)
```

Для отправки сообщение ICMP Echo Request и приема ICMP Echo Reply можно использовать утилиту ping, входящую в большинство современных ОС. Утилита ping посылает сообщение ICMP Echo Request подобно клиенту, а адресуемый узел сети отвечает сообщением ICMP Echo Reply, выступая в роли сервера. Если адресуемый узел отвечает, утилита ping сообщает в стандартный выходной поток, что узел сети активен (host is alive) и завершает работу. В противном случае, после определенного интервала времени (timeout) выдает сообщение, что от узла ответа нет (no answer from host).

Основные опции утилиты ping:

-*d:* Режим отладки;

-*f:* Лавинная рассылка (Flood ping). Доступна только root;

-*i:* Время ожидания (Interval) между отправкой пакетов;

-*n:* Числовое именование узлов сети (Numeric output only);

-*p:* Шаблон (Pattern);

-*v:* Детальный вывод (Verbose output);

-*b:* Использование широковещательного адреса (Allow pinging a broadcast address);

-*c:* Количество (count) отправляемых пакетов.

Утилиту ping возможно использовать в сценариях (скриптах) для автоматизации идентификации активных узлов сети (мало применяется ввиду низкой скорости сканирования). Для опроса всех узлов сети класса C, можно написать простой сценарий, например:

```
#!/bin/sh
for i in {1..254};
do
ping -c1 192.168.0.$i;
done
exit 0
```

Утилита ping с опцией –b позволяет отправлять сообщения ICMP Echo Request по широковещательному адресу (broadcast address), таким образом можно определить доступность множества узлов сети:

```
linux:~# ping -b 192.168.1.255
```

Использование утилиты ping нерационально для опроса узлов большой сети, поскольку опрос происходит последовательно. Для исследования больших сетей рекомендуется использовать специализированные утилиты, например: fping, hping3, nmap и т.д.

3. Практическое задание

1. Проверьте наличие утилиты ping в системе:

```
linux:~# whereis ping
ping: /bin/ping /usr/share/man/man8/ping.8.gz
```

В случае необходимости, установите ПО:

```
linux:~# apt-get update && apt-get install nmap
```

2. Проверьте настройки сетевых интерфейсов;

3. Ознакомьтесь на практике с различными режимами работы утилиты ping (-f -i -v -b -c);

4. Напишите программу (bash, perl, ruby), производящую сканирование заданного диапазона адресов сети средствами утилиты ping;

5. Исследуйте подсеть лаборатории (компьютерного класса, экспериментальной установки), выявите все активные узлы при помощи разработанной программы;

6. Измерьте время сканирования диапазона из десяти, пятидесяти и ста адресов, результат оформите в виде таблицы.

Практическая работа №2
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ. ICMP ECHO REQUEST
(Утилиты FPING и NMAP)

1. Цель работы

Закрепить навыки обнаружения (идентификации) активных узлов сети с помощью протокола управляющих сообщений ICMP. Попрактиковаться в использовании программных средств исследования сети: nmap, fping, hping3.

2. Теоретические сведения. Методические рекомендации

Утилита тестирования сети fping

Fping (http://www.fping.com/) — популярная свободно распространяемая утилита для тестирования сети на основе протокола ICMP (для *NIX систем), относящаяся к инструментам, использующим технологию PING SWEEP, т.е. способна асинхронно опрашивать большое количество узлов сети, управляя временем задержки между передаваемыми ICMP-сообщениями (delay time). Утилите fping в качестве параметра можно задать более одного узла для тестирования или указать текстовый файл со списком узлов. В отличие от утилиты ping, fping широко используют в сценариях (скриптах), т.к. она представляет выходные данные в форме удобной для дальнейшего разбора (parse).

Синтаксис утилиты fping:

```
fping [options] [targets...]
```

Список наиболее используемых опций утилиты fping:

-*a:* Показывает доступные узлы сети (show targets that are alive);
-*b n:* Количество посылаемых пакетов (по умолчанию 56);
-*B f:* Устанавливает фактор задержки;
-*c n:* Количество пакетов посылаемых, каждому узлу сети;
-*e:* Выводит время выполнения (Round Trip Time);
-*f file:* Задает имя файла со списком тестируемых узлов сети;

-g: Генерирует список тестируемых узлов сети;
-i n: Задает интервал между посылаемыми пакетами (в миллисекундах);
-l: Задает опрос узлов сети в цикле (loop sending pings forever);
-p n: Задает интервал между отправкой пакетов на узлел сети;
-r n: Число попыток (по умолчанию 3);
-s: Вывод финальной статистики (print final stats);
-u: Отображает недоступные узлы сети (show targets that are unreachable).

Практический пример использования утилиты fping для идентификации активных узлов подсети 192.168.1.1-100/24:

```
linux:~# fping -r 1 -g 192.168.1.1 192.168.1.100 -s hosts
192.168.1.1 is alive
192.168.1.26 is alive
192.168.1.27 is alive
100 targets
       3 alive
      97 unreachable
       0 unknown addresses
      45 timeouts (waiting for response)
     145 ICMP Echos sent
       3 ICMP Echo Replies received
      88 other ICMP received
 0.05 ms (min round trip time)
 0.57 ms (avg round trip time)
 1.04 ms (max round trip time)
        4.394 sec (elapsed real time)
```

Утилита тестирования сети nmap

Nmap (Network Mapper, http://www.nmap.org) - утилита с открытым исходным кодом для исследования сети и проверки безопасности. Она была разработана для быстрого сканирования больших сетей. Выходные данные nmap - это список просканированных целей с дополнительной информацией по каждой в зависимости от заданных опций (http://nmap.org/man/ru/).

Синтаксис сканера nmap:

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

Список наиболее используемых опций утилиты nmap при обнаружении узлов:

-iL: <имя_входного_файла> Использовать список узлов/сетей из файла;

-sL: Сканирование с целью составления списка узлов для сканирования;

-sP: ICMP Ping сканирование - определяет, активен ли узел сети;

-n/-R: Никогда не производить DNS разрешение / Всегда производить разрешение;

-T[0-5]: Установить шаблон настроек управления временем (больше - быстрее);

-PE (Ping ICMP): Использовать ICMP запрос (ICMP Echo Request).

Для просмотра всех опций сканера nmap применяется ключ –h:

```
linux:~# nmap -h
Nmap 5.21 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
Can pass hostnames, IP addresses, networks, etc.
Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
-iL <inputfilename>: Input from list of hosts/networks
```

Чтобы произвести ICMP Echo Request сканирование средствами утилиты nmap и вывести список доступных узлов сети, т.е. узлов, которые ответили на сообщение ICMP Echo Request, используется опция -sP (тип - ICMP Ping сканирование). Пример использования утилиты nmap для обнаружения активных узлов сети подсети 192.168.1.1-254/24:

```
linux:~# nmap –sP 192.168.1.1-254 –PE –T5
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-27 15:06 MSD
Nmap scan report for 192.168.1.1
Host is up (0.00092s latency).
Nmap scan report for 192.168.1.26
Host is up (0.00089s latency).
Nmap scan report for 192.168.1.27
Host is up (0.00086s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 1.34 seconds
```

3. Практическое задание

1. Проверьте наличие всех рассмотренных утилит в системе:

```
linux:~# whereis nmap
nmap: /usr/bin/nmap /usr/lib/nmap /usr/share/nmap
/usr/share/man/man1/nmap.1.gz
```

2. Исследуйте подсеть лаборатории (компьютерного класса, экспериментальной установки), выявите все активные узлы средствами утилит fping, nmap, hping3 (ознакомиться с назначением и основными опциями утилиты hping3 необходимо самостоятельно: http://www.hping.org);

3. Сравните время сканирования всего диапазона адресов исследуемой подсети различными утилитами: ping, fping, hping3, nmap (для nmap провести сканирование тремя различными режимами управления временем), результаты оформите в виде таблицы;

4. Напишите программу (bash, perl, ruby), производящую периодическую проверку доступности некоторого множества узлов сети (с использованием любой из рассмотренных утилит) и выводящую сообщение на стандартное устройство (отправляющую сообщение на почту, адрес почты уточнить у преподавателя) в случае недоступности какого-либо из наблюдаемых узлов (перечень контролируемых узлов сети необходимо выбирать из полученного ранее файла).

Практическая работа №3
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ. ИНФОРМАЦИОННЫЕ ICMP СООБЩЕНИЯ

1. Цель работы

Изучить методы обнаружения узлов сети средством информационных запросов TimeStamp Request, Information Request, Address Mask Request. Закрепить навыки работы с программными средствами исследования сети: nmap, ping, fping, hping3, icmpush.

2. Теоретические сведения. Методические рекомендации

Кроме рассматриваемых выше сообщений протокола ICMP Echo Request и Echo Reply, для идентификации активных узлов сети можно использовать и другие информационные ICMP-сообщения. Такие сообщения, как правило, бывают более информативными, чем просто ответ ICMP Echo Reply. Наиболее используемые на практике запросы ICMP:

- TimeStamp Request (Type 13) - Запрос метки времени;
- Information Request (Type 15) - Запрос информации (Определение адреса сети для бездисковых станций);
- Address Mask Request (Type 17) - Запрос маски адреса (RFC-950).

Запрос метки времени

ICMP-запрос TimeStamp Request (Type 13) используется для получения текущего значения времени на исследуемом узле. Формат ICMP-запроса TimeStamp Request представлен на рис. 3.

На ICMP-запрос TimeStamp Request удаленный узел должен ответить ICMP-сообщением Timestamp Reply (некоторые системы могут игнорировать данный запрос). Для генерации ICMP-запросов TimeStamp Request могут быть использованы такие утилиты, как: sing (http://www.sourceforge.net/projects/sing), hping3, nmap и другие.

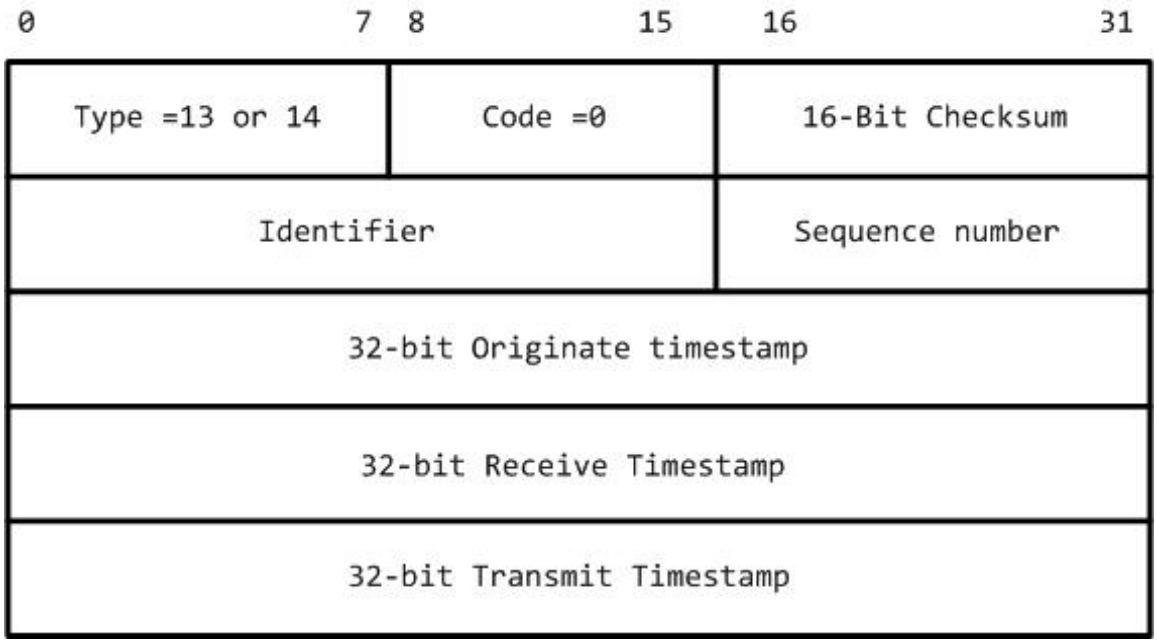


Рис. 3. Формат ICMP-запроса TimeStamp Request

Пример использования утилиты nmap для отправки сообщений TimeStamp Request:

```
linux:~$ sudo nmap -sP 192.168.1.1 -PP
[sudo] password for linux:
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-27 21:52 MSD
Nmap scan report for 192.168.1.1
Host is up (0.00039s latency).
MAC Address: 00:18:02:17:A5:E3 (Alpha Networks)
Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

Опция -PP указывает nmap использовать пакет ICMP TimeStamp Request (Type 13).

Пример использования утилиты icmpush для отправки сообщений TimeStamp Request:

```
linux:~$ sudo icmpush -tstamp 192.168.0.2
192.168.0.2 -> 09:37:59
```

Запрос Information Request

ICMP-сообщение Information Request (Type 15) используется протоколами BOOTP, DHCP и служит для определения адреса сети (для бездисковых станций). Некоторые ОС могут посылать ответ на подобные запросы. Формат ICMP-запроса Information Request представлен на рис. 4.

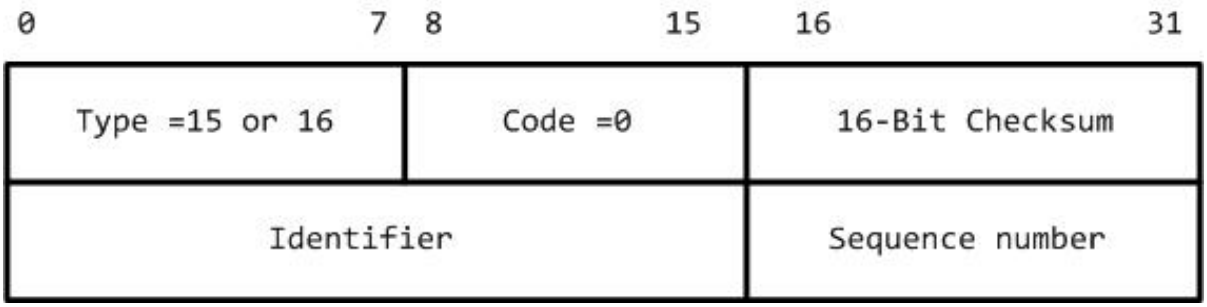


Рис. 4. Формат ICMP-запроса Information Request

Пример использования утилиты sing для отправки сообщений Information Request:

```
linux:~$ sing –info 192.168.1.1
```

Запрос Netmask Request

ICMP-сообщение Netmask Request (Type 17) – получение маски подсети, также может использоваться для идентификации узлов сети в условиях фильтрации сообщений ICMP Echo Request (различные операционные системы реагируют на данный запрос по разному). Формат ICMP-запроса Netmask Request представлен на рис. 5.

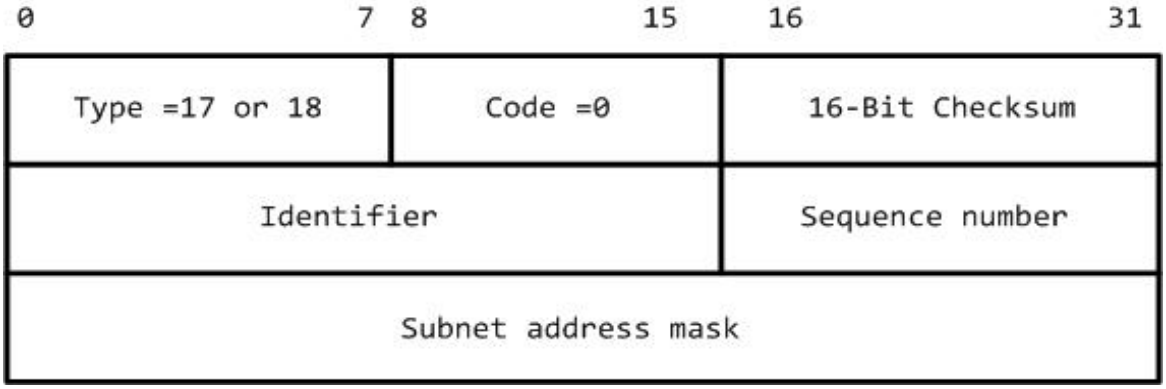


Рис. 5. Формат ICMP-запроса Netmask Request

Пример использования утилиты sing для отправки сообщений Netmask Request:

```
linux:~$ sing –mask 192.168.1.1
```

Пример использования утилиты nmap для отправки сообщений Netmask Request:

```
linux:~$ sudo nmap -sP 192.168.1.1 -PM
[sudo] password for linux:
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-27 22:52 MSD
```

```
Nmap scan report for 192.168.1.1
Host is up (0.00039s latency).
MAC Address: 00:18:02:17:A5:E3 (Alpha Networks)
Nmap done: 1 IP address (1 host up) scanned in 0.43 seconds
```

Опция -PM указывает nmap использовать пакет ICMP Netmask Request (Type 17).

3. Практическое задание

1. Исследуйте подсеть лаборатории (компьютерного класса, экспериментальной установки), выявите все активные узлы средством запросов TimeStamp Request, Information Request, Netmask Request. Выявите узлы сети, на которых запросы ICMP Echo Request фильтруются;
2. Повторите исследование, используя все рассмотренные в работе утилиты, сравните результаты;
3. Сравните реакцию различных ОС на информационные ICMP-запросы. Необходимо использовать одну из рассмотренных утилит (на выбор). Результаты занесите в таблицу (табл. 3).

Таблица 3

Операционная система	Indirect ICMP message types (broadcast)			
	Type 8	Type 13	Type 15	Type 17
Linux				
Windows 7				
…				

Практическая работа №4
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ СРЕДСТВАМИ ПРОТОКОЛА TCP (TCP-PING)

1. Цель работы

Ознакомиться с основами обнаружения (идентификации) активных узлов сети с помощью протокола TCP. Научиться использовать метод обнаружения устройств TCP-Sweep. Закрепить навыки работы с программными средствами исследования сети: nmap, hping3.

2. Теоретические сведения. Методические рекомендации

Протокол TCP

Протокол TCP (Transmission Control Protocol - протокол управления передачей, RFC 793) — один из основных протоколов Internet, предназначенный для управления передачей данных в сетях TCP/IP. Выполняет функции протокола транспортного уровня модели OSI. Протокол TCP осуществляет доставку дейтаграмм, в виде байтовых потоков с установлением соединения. Протокол TCP применяется в тех случаях, когда требуется гарантированная доставка сообщений. Он использует контрольные суммы пакетов для проверки их целостности и освобождает прикладные процессы от необходимости задержек (timeout) и повторных передач. Для отслеживания подтверждения доставки в TCP реализуется алгоритм скользящего окна (Sliding window).

Рассмотрим наиболее важные (в рамках решаемой задачи) поля сегмента TCP.

Порт отправителя (Source Port - 16 bits): идентифицирует приложение клиента, отправляющего пакеты. По возвращении данные передаются клиенту на основании номера порта источника.

Порт получателя (Destination Port - 16 bits): идентифицирует порт, на который отправлен пакет. Перечень сетевых служб и

закрепленных за ними портов TCP можно получить, используя следующую команду (в *nix системах):

```
linux:~$ less /etc/services | grep tcp
```

Наиболее популярные сетевые службы, работающие на основе TCP: 20/21 — FTP; 22 — SSH; 23 — Telnet; 25 — SMTP; 80 — HTTP; 110 — POP3; 194 — IRC (Internet Relay Chat); 443 — HTTPS (Secure HTTP).

Флаги (Control Bits - 6 bits): это поле содержит 6 битовых флагов: URG — Поле Указатель важности (Urgent pointer field is significant); ACK — Поле Номер подтверждения (Acknowledgement field is significant); PSH — (Push function) инструктирует получателя «протолкнуть» данные, накопившиеся в приемном буфере, в приложение пользователя; RST — Оборвать соединения (Reset the connection); SYN — Синхронизация номеров последовательности (Synchronize sequence numbers); FIN — флаг указывающий на завершение соединения (FIN bit used for connection termination).

Формат сегмента TCP (TCP Header Format) представлен на рис. 6 (RFC 793).

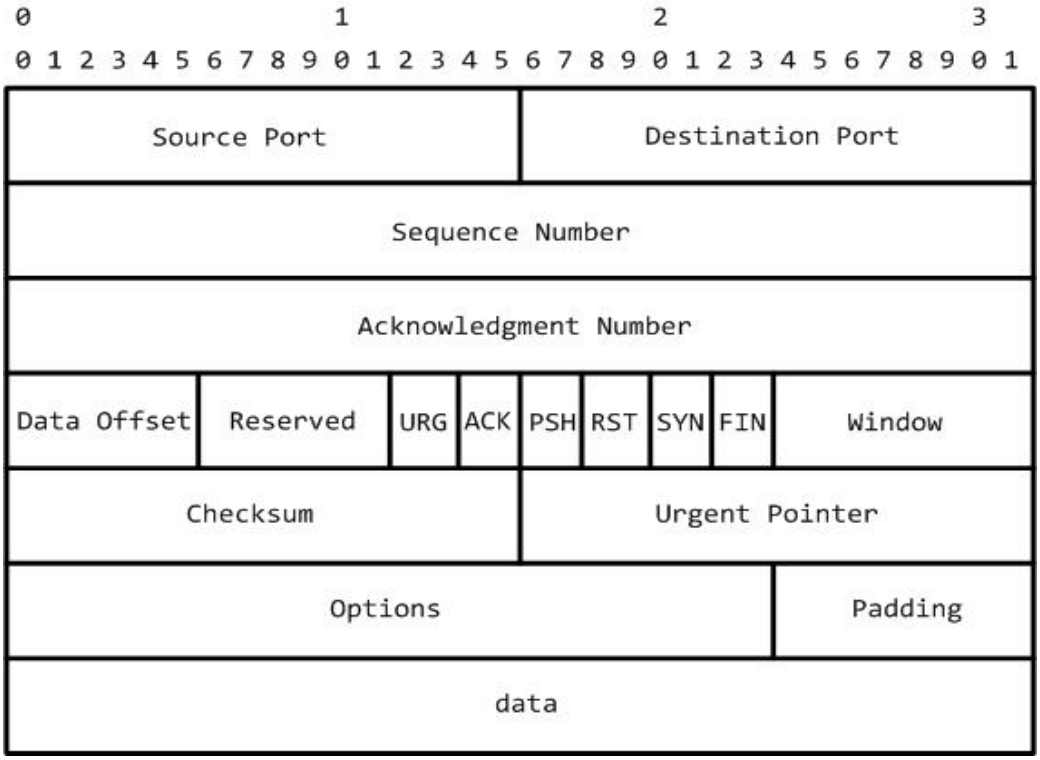


Рис. 6. Формат сегмента TCP (TCP Header Format)

TCP Sweep

Метод определения доступности узла сети с применением протокола TCP называется TCP Ping или TCP Sweep. Данный метод может использоваться в случае, когда на удаленной машине отключен ответ на ICMP-запросы (сообщения протокола ICMP фильтруются МЭ). Вместо отправки ICMP Echo Request на сканируемый узел сети, отправляется TCP-пакет с установленными флагами ACK или SYN|ACK. Реакция тестируемого узла при получении таких запросов может быть неоднозначна, она зависит от версии ОС и от настроек МЭ. Обычно узел отвечает сообщением с флагом RST, что свидетельствует о его активности в текущий момент. Возможно использование сообщений с флагом SYN. Если исследуемый узел активен и трафик не блокируется МЭ, то вероятен ответ с флагом RST или SYN|ACK.

Рассмотрим несколько популярных программных продуктов, способных генерировать пакеты с различным сочетанием TCP-флагов.

TCP Sweep средствами утилиты hping3

Утилита hping3 является мощным инструментом тестирования сетей и может передавать IP-пакеты с заданными параметрами, выводя на экран полученные от адресата отклики, подобно программам, работающим с откликами ICMP. Hping3 поддерживает фрагментацию, позволяет задавать произвольное содержимое поля данных IP-пакета, менять размер IP-пакетов и может использоваться для передачи файлов, инкапсулированных с использованием поддерживаемых протоколов.

Синтаксис утилиты hping3:

```
hping3 [options] [targets...]
```

Наиболее популярные опции hping3:

Опции общего назначения

-h (--help): Выводит на экран краткую справку о работе с программой;
-c (--count): Прекращение работы после передачи заданного числа пакетов;

--fast: Задает передачу пакетов с периодом 10 мсек;
--faster: Задает передачу пакетов с периодом 1 мксек;
--flood: Задает передачу пакетов с максимально возможной скоростью без ожидания приема откликов;

Опции выбора протокола

-0 (--rawip): Режим RAW IP, при котором hping3 будет передавать заголовки IP с данными, указанными опцией --signature и/или --file;
-1 (--icmp): Режим ICMP – программа будет передавать пакеты ICMP (по умолчанию Echo Request). Возможен выбор типа и кода ICMP с помощью опций --icmptype и --icmpcode;
-2 (--udp): Режим UDP – hping3 будет передавать пакеты UDP, адресованные на порт 0. Для управления параметрами заголовков UDP служат опции --baseport, --destport, --keep;

Опции IP

-a (--spoof): Указывает в передаваемых пакетах подставной адрес отправителя;
-t (--ttl): Задает время жизни (TTL) генерируемых пакетов (Эта опция весьма полезна при совместном использовании с опциями --traceroute и --bind);
-f (--frag): Задает режим фрагментации пакетов, который может быть полезен для тестирования стека IP или проверки МЭ;

Опции ICMP

-C (--icmptype): Задает тип ICMP (по умолчанию - ICMP Echo Request);
-K (--icmpcode): Задает код ICMP (по умолчанию - 0);
--icmp-ts: Псевдоним для --icmptype 13 (передача запросов ICMP TimeStamp Request);
--icmp-addr: Псевдоним для --icmptype 17 (передача запросов ICMP Netmask Request);

Опции TCP/UDP

-s (--baseport): Эта опция задает порт отправителя;
-p (--destport): Эта опция задает порт получателя;
-w (--win): Задает размер окна TCP (по умолчанию – 64);
-F (--fin): Устанавливает флаг TCP FIN;

-S (--syn): Устанавливает флаг TCP SYN;
-R (--rst): Устанавливает флаг TCP RST;
-P (--push): Устанавливает флаг TCP PUSH;
-A (--ack): Устанавливает флаг TCP ACK;
-U (--urg): Устанавливает флаг TCP URG;
-X (--xmas): Устанавливает флаг TCP Xmas;
-Y (--ymas): Устанавливает флаг TCP Ymas.

Более подробную информацию по утилите hping3 можно получить из документации:

```
linux:~$ man hping3
```

Пример использования hping3 (сканирование TCP-пакетом с флагом SYN):

```
linux:~$ sudo hping3 192.168.1.1 –S –p 80 –c 1
HPING 192.168.1.1 (eth0 192.168.1.1): S set, 40 headers + 0 data
bytes
len=46 ip=192.168.1.1 ttl=64 id=46780 sport=80 flags=SA seq=0
win=16000 rtt=0.9 ms
--- 192.168.1.1 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.9/0.9/0.9 ms
```

В примере отправляется один TCP-пакет с установленным флагом SYN на 80-й TCP-порт узла сети. Данный метод определения доступности узла достаточно эффективен, т.к. от доступного узла ответ придет в любом случае – в случае если TCP-порт открыт или если TCP-порт закрыт. Применение сканирования пакетами с другим сочетанием флагов может вызвать неоднозначную реакцию узла сети. Также такие пакеты могут «отбрасываться» МЭ (поддерживающими технологию Stateful Inspection).

Пример использования hping3 (сканирование TCP-пакетом с флагом SYN|ACK):

```
linux:~$ sudo hping3 192.168.1.1 –S -A –p 80 –c 1
```

Пример использования hping3 (сканирование TCP-пакетом с флагом ACK):

```
linux:~$ sudo hping3 192.168.1.1 –A –p 80 –c 1
```

Перечень опций TCP-ping сканера nmap:
-*PS/PA [список_портов]*: TCP SYN|ACK сканирование.

3. Практическое задание

1. Исследуйте подсеть лаборатории (компьютерного класса, экспериментальной установки), выявите все активные узлы методом TCP Ping, с использованием TCP-пакетов с различным сочетанием флагов (с техникой сканирования методом TCP Ping средствами nmap ознакомьтесь самостоятельно: http://nmap.org/man/ru/);
2. Выявите узлы сети, на которых запросы ICMP Echo Request фильтруются;
3. Сравните реакцию различных ОС на информационные ICMP запросы. Необходимо использовать одну из рассмотренных утилит (на выбор). Результаты занесите в таблицу;
4. Исследуйте реакцию узла сети на SYN-запросы, сравните ответы от открытого и закрытого TCP-порта. Все исследования необходимо провести средствами hping3 и nmap.

Практическая работа №5
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ СРЕДСТВАМИ ПРОТОКОЛОВ UDP (UDP-PING), IP

1. Цель работы

Ознакомиться на практике с методами обнаружения (идентификации) активных узлов сети с помощью протокола UDP. Научиться использовать метод обнаружения устройств UDP Discovery. Получить представление о практических методах обнаружения активных узлов сети с помощью протокола IP. Овладеть следующими методами: методом идентификации узла с помощью фрагментов IP-пакета, идентификации узла отправкой IP-пакета ошибочной длины, идентификации узла отправкой IP-пакета неподдерживаемого тестируемой системой протокола.

2. Теоретические сведения. Методические рекомендации

Протокол UDP

Протокол UDP (User Datagram Protocol, RFC 768) проектировался для создания в объединенной системе компьютерных сетей с коммутацией пакетов режима передачи дейтаграмм клиента. Протокол UDP предоставляет прикладной программе процедуру для отправки сообщений другим программам, причем механизм протокола минимален. Данный протокол предполагает, что нижестоящим протоколом является IP. Протокол UDP не ориентирован на транзакции, получение дейтаграмм и защита от дублирования не гарантированы.

Формат дейтаграммы UDP (UDP Header Format) представлен на рис. 7 (RFC 768).

Метод UDP Discovery

Метод обнаружения сетевых узлов с помощью протокола UDP называется UDP Discovery, UDP Scan или UDP Sweep. Его суть

заключается в том, что если в ответ на UDP-пакет от узла было получено ICMP-сообщение Destination Unreachable или Port Unreachable, то исследуемый узел доступен (UDP-порт фильтруется).

Рис. 7. Формат дейтаграммы UDP (UDP Header Format)

Если ответа получено не было, то возможны различные варианты (ввиду особенностей протокола UDP):

- Узел сети выключен или недоступен;
- ICMP-сообщения от узла блокируются средствами МЭ;
- Указанный в UDP-пакете порт открыт.

Метод UPD Discovery считается малоэффективным из-за высокой вероятности фильтрации UDP-трафика средствами МЭ и непредсказуемой реакции узла при отправке UDP-пакета на открытый порт. Пример идентификации доступности узла сети методом UPD Discovery при помощи сканера nmap:

```
linux:~$ sudo nmap -sU –p 53 192.168.1.1 –c3
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-29 00:30 MSD
Nmap scan report for 192.168.1.1
Host is up (0.000074s latency).
PORT   STATE  SERVICE
53/udp closed domain
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

В примере отправляются три пакета на 53-й UDP-порт узла сети. Опция -sU указывает nmap использовать метод UPD Discovery.

Пример идентификации доступности узла сети методом UPD Discovery при помощи утилиты hping3:

```
linux:~$ sudo hping3 192.168.1.1 –2 –p 80 –c 1
```

```
HPING 192.168.1.1 (eth0 192.168.1.1): udp mode set, 28 headers + 0
data bytes
ICMP Port Unreachable from ip=192.168.1.1 get hostname...
--- 192.168.1.1 hping statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Протокол Internet (IP)

Протокол Internet (IP – RFC 791) создан для использования в объединенных системах компьютерных коммуникационных сетей с коммутацией пакетов. Протокол Internet обеспечивает передачу блоков данных, называемых IP-дейтаграммами (IP-пакетами). Протокол Internet обеспечивает при необходимости также фрагментацию и сборку IP-дейтаграмм для передачи данных через сети с малым размером пакетов.

Формат дейтаграммы IP (Internet Header Format) представлен на рис. 8 (RFC 791).

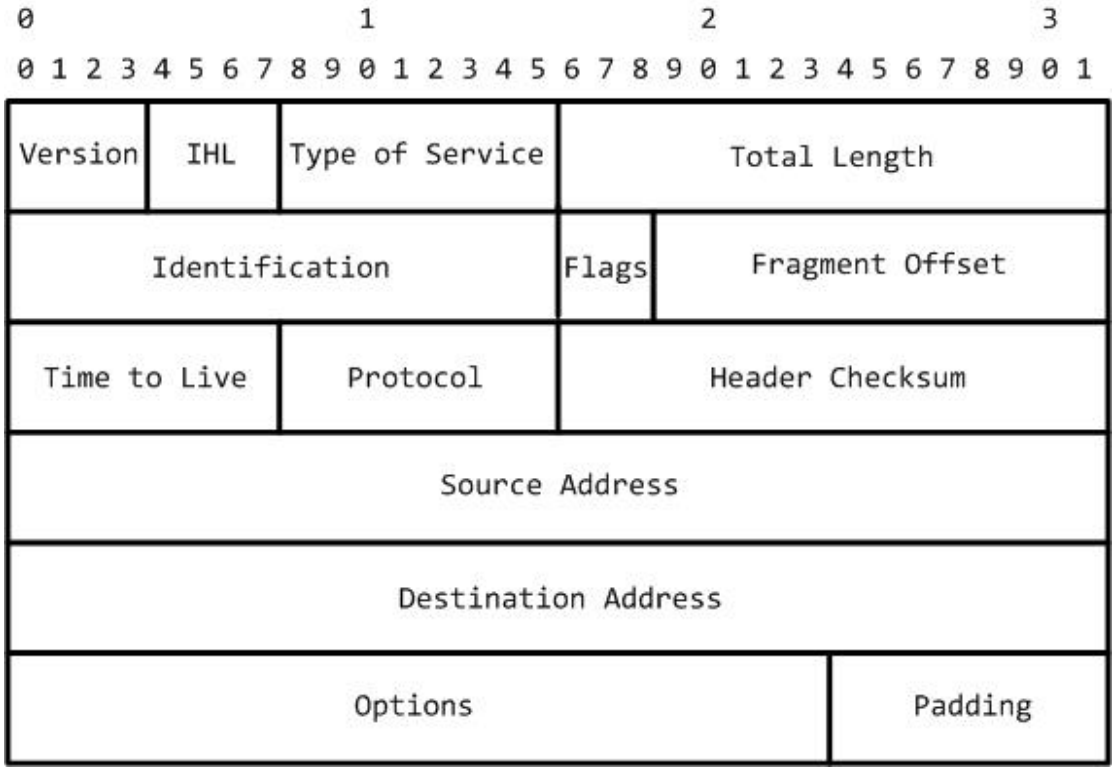


Рис. 8. Формат дейтаграммы IP (Internet Header Format)

Метод обнаружения узлов сети при помощи протокола IP основан на отправке IP-пакетов с ошибками в заголовке. Признаком

доступности узла служит ICMP-сообщение об ошибке. Ниже рассмотрены несколько методов, которые эффективно применяются для исследования доступности удаленных узлов.

Метод идентификации узла с помощью фрагментов IP-пакета

Метод основан на отправке незавершенной последовательности фрагментов IP-пакета. Когда сканируемый узел получает первый фрагмент IP-пакета, он запускает таймер. Если время ожидания превышает установленный предел, а узел так и не получил все фрагменты, он возвращает отправителю ICMP-сообщение об ошибке Fragment Reassembly Time Exceeded (превышено время на повторную сборку пакета - ICMP Type=11, Code=1). Согласно RFC 792, для отправки ICMP-сообщения Fragment Reassembly Time Exceeded обязательно должен быть получен первый фрагмент последовательности.

Пример использования метода отправки фрагмента IP-пакета при помощи утилиты hping3:

```
linux:~$ sudo hping3 192.168.1.1 –x –p 80 –c 1 –S
```

В данном примере осуществляется отправка SYN-пакета на 80-й TCP-порт узла сети. Опция –x указывает, что отправлен не последний фрагмент.

Метод идентификации узла отправкой IP-пакета ошибочной длины

Метод основан на отправке IP-пакета с ошибочным значением длины в заголовке, в ответ на такой пакет должно прийти ICMP-сообщение об ошибке Parameter Problem / Protocol Unreachable (ошибка в параметрах дейтаграммы / протокол недостижим - ICMP Type=12 Code=2).

Метод идентификации узла отправкой IP-пакета неподдерживаемого тестируемой системой протокола

Метод основан на отправке IP-пакета с неподдерживаемым тестируемой системой типом протокола, в ответ на такой пакет

должно прийти ICMP-сообщение об ошибке Destination Unreachable / Protocol Unreachable (получатель недостижим / протокол недостижим - ICMP Type=3, Code=2). Пример использования метода отправки IP-пакета с неподдерживаемым тестируемой системой типом протокола при помощи утилиты hping3:

```
linux:~$ sudo hping3 192.168.1.1 –0 –H 255 –c 1
```

В данном примере осуществляется отправка одного пакета с типом протокола 255. Опция –0 указывает на работу с протоколом IP в режиме RAW IP.

3. Практическое задание

1. Идентифицируйте узлы сети, фильтрующие сообщения ICMP Echo Request (диапазон адресов узлов сети уточнить у преподавателя). Сканирование произведите методом UPD Discovery средствами рассмотренных утилит;
2. Исследуйте реакцию узла сети на UDP-запросы, сравните ответы открытого и закрытого UDP-порта. Исследования проведите для различных ОС;
3. Идентифицируйте узлы сети, фильтрующие сообщения ICMP Echo Request, используйте рассмотренные методы протокола IP;
4. Сравните реакцию различных ОС на IP-пакеты с рассмотренными типами ошибок в заголовке.

Практическая работа №6
ОБНАРУЖЕНИЕ УЗЛОВ СЕТИ СРЕДСТВАМИ ПРОТОКОЛА ARP (ARP-PING)

1. Цель работы

Изучить методы обнаружения активных узлов сети с помощью протокола ARP. Приобрести навык использования программного средства обнаружения устройств arping.

2. Теоретические сведения. Методические рекомендации

При идентификации узлов в локальной сети (LAN) эффективным является метод обнаружения с использованием ARP-запросов. Данный метод позволяет обнаружить работающий узел даже в случае блокирования всего IP-трафика.

Протокол ARP

ARP – протокол разрешения адресов (Address Resolution Protocol, RFC 826) — использующийся в компьютерных сетях протокол низкого уровня, предназначенный для определения адреса канального уровня по известному адресу сетевого уровня. Протокол ARP работает различным образом в зависимости от протокола канального уровня сети - протокол локальной сети (Ethernet, Token Ring, FDDI) с возможностью широковещательного доступа ко всем узлам сети, или же протокол глобальной сети (X.25, frame relay), как правило, не поддерживающий широковещательный доступ.

В локальных сетях протокол ARP использует широковещательные кадры протокола канального уровня для поиска в сети узла с заданным IP-адресом. Узел, которому нужно разрешить отображение IP-адреса на локальный адрес, формирует ARP-запрос, инкапсулирует его в кадр протокола канального уровня, указывая в нем известный IP-адрес, и рассылает запрос широковещательно. Все узлы локальной сети получают ARP-запрос и сравнивают указанный

там IP-адрес с собственным. В случае их совпадения узел формирует ARP-ответ, в котором указывает свой IP-адрес и свой локальный адрес и посылает его уже направленно, так как в ARP-запросе отправитель указывает свой локальный адрес.

ARP-запросы и ARP-ответы используют один и тот же формат пакета. Формат дейтаграммы ARP представлен на рис. 9 (RFC-826).

Рис. 9. Формат дейтаграммы ARP

В методе обнаружения с помощью ARP-запросов используют nmap, arping, ettercap и другие сетевые утилиты.

Arping

Программа arping функционально аналогична утилите ping, но использует ARP-запросы. Arping выполняет запрос Echo Request, используя MAC-адрес (аппаратный адрес сетевого интерфейса), сопоставляет MAC и IP адреса, не используя ARP-кэш.

Основные опции arping:

-A: Аналогично -U, но используются пакеты ARP REPLY вместо ARP REQUEST;

-b: Отправляет только широковещательные пакеты уровня MAC;

-*c:* Счетчик;
-*D:* Режим дублированного обнаружения адреса (RFC2131);
-*f:* Завершает работу после первого приема ответа;
-*I:* Наименование сетевого интерфейса, который используется для отправки запросов ARP Request;
-*h:* Отображает страницу помощи и завершает работу;
-*U:* Предоставляет содержимое кэша ARP для обновления ARP-кэша соседних узлов.

Пример сканирования методом ARP-запросов при помощи утилиты arping:

```
linux:~$ sudo arping 192.168.0.2
[sudo] password for linux:
ARPING 192.168.0.2 from 192.168.0.9 eth0
Unicast reply from 192.168.0.2 [00:22:64:16:CB:9C]  3.132ms
Unicast reply from 192.168.0.2 [00:22:64:16:CB:9C]  0.846ms
Unicast reply from 192.168.0.2 [00:22:64:16:CB:9C]  0.877ms
^CSent 3 probes (1 broadcast(s))
Received 3 response(s)
```

Пример сканирования методом ARP-запросов при помощи утилиты nmap:

```
linux:~$ sudo nmap -sP -PR 192.168.0.2
[sudo] password for linux:
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-31 10:00 MSD
Nmap scan report for 192.168.0.2
Host is up (0.00029s latency).
MAC Address: 00:22:64:16:CB:9C (Hewlett Packard)
Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

Опция -PR сканера nmap указывает на необходимость использования ARP-сканирования.

3. Практическое задание

1. Идентифицируйте узлы сети, фильтрующие сообщения ICMP Echo Request (диапазон адресов узлов сети уточнить у преподавателя). Произведите сканирование методом ARPing с помощью рассмотренных утилит;

2. Напишите программу (bash, perl, ruby), производящую периодическую проверку на предмет появления нового устройства в сети (с неизвестным ранее аппаратным адресом). Результаты проверки должны выводиться на стандартное устройство и сохраняться в файле.

Практическое задание к разделу I

На основе результатов исследований сети лаборатории (компьютерного класса, экспериментальной установки), полученных в работах раздела, заполните (результаты по каждому из тестов в форме «удачно/неудачно») таблицу (табл. 4).

Таблица 4

№	IP, MAC	Echo Request	Ping Sweep	TimeStamp Request	Information Request	Address Mask Request	TCP Ping (22,23,53,80)	UPD Ping (22,23,53,80)	ARP Ping
1									
2									
…									
n									

Контрольные вопросы к разделу I

1. Что понимается под идентификацией узлов корпоративной сети передачи данных?
2. Какие протоколы стека TCP/IP могут применяться при идентификации узлов?
3. Протокол ICMP. Назначение, формат пакета протокола ICMP.
4. Назовите основные способы обнаружения узлов сети средствами протокола ICMP.
5. Синтаксис и основные опции утилиты ping.
6. Назовите основные недостатки применения утилиты ping при решении задачи идентификации узлов КСПД.
7. Технология PING SWEEP, достоинства и недостатки.
8. Синтаксис и основные опции утилиты fping.
9. Синтаксис и основные режимы сетевого сканера nmap.

10. Методы обнаружения узлов сети средством информационных запросов TimeStamp Request, Information Request, Address Mask Request.
11. Протокол TCP, назначение, TCP соединение, флаги.
12. Формат сегмента протокола TCP, TCP-порты.
13. TCP-sweep. Достоинства и недостатки метода TCP-sweep.
14. Синтаксис и основные опции утилиты hping3.
15. Протокол UDP. Режим передачи данных без установления соединения.
16. Формат дейтаграммы протокола UDP, UDP -порты.
17. Метод UDP Discovery. Достоинства и недостатки метода.
18. Протокол IP. Адресация.
19. Формат пакета протокола IP.
20. Метод идентификации с помощью IP фрагментов.
21. Метод идентификации отправкой IP пакета ошибочной длины.
22. Метод идентификации отправкой IP пакета неподдерживаемого протокола.
23. Протокол ARP. Адресация канального уровня ISO OSI.
24. Формат дейтаграммы ARP.
25. Синтаксис и основные опции утилиты arping.
26. Метод arping. Достоинства и недостатки.

РАЗДЕЛ II – ОПРЕДЕЛЕНИЕ ТОПОЛОГИИ СЕТИ

Следующей задачей после обнаружения сетевых объектов (узлов) является задача построения карты (топологии) исследуемой сети: определение маршрутов передачи пакетов в сетях TCP/IP, визуализация топологии исследуемой сети.

Практическая работа №7
ОСНОВНЫЕ СРЕДСТВА ОПРЕДЕЛЕНИЯ МАРШРУТОВ IP-ПАКЕТОВ - PING, TRACEROUTE

1. Цель работы

Изучить основные методы определения маршрутов передачи данных в сетях TCP/IP. Поупражняться в использовании программных средств определения маршрутов передачи данных: ping, traceroute. Провести исследование сети лаборатории с помощью рассмотренных программ.

2. Теоретические сведения. Методические рекомендации

На начальных этапах построения топологии сети необходимо определить маршруты движения пакетов между узлами и подсетями КСПД, т.е. адреса всех промежуточных маршрутизаторов при прохождении IP-пакета к целевому узлу сети. Для определения маршрутов (трассировки) служат утилиты: ping, traceroute (tracert в ОС Windows), tracepath, tracemap, tcptraceroute (tcptracert), pathping и другие. В случае проблем при доставке IP-пакетов трассировка позволяет определить, на каком именно участке сети возникли неполадки.

Метод Record Route (RR)

Метод определения маршрута Record Route основан на добавлении в заголовок IP-пакета опции RR (Record Route), в которой сохранятся адреса всех пройденных узлов (маршрутизаторов). Record

Route (записывать путь) – опция IP-пакета, которая позволяет записывать IP-адрес маршрутизатора. Опциональная часть IP-заголовка имеет в данном случае длину 39 байт и выглядит, как показано на рис. 10.

1 байт тип опции	1 байт длина опции	1 байт отступ
4*9 байт IP-адреса маршрутизаторов		

Рис. 10. Опциональная часть IP-заголовка

Тип опции будет равен 7, что означает Record Route. Длина опции – 39 байт. Изначально значение отступа равно 4. При прохождении пакета через маршрутизатор, в опциональную часть будет дописываться IP-адрес, начиная с отступа, а сам отступ будет увеличиваться на 4. Когда отступ станет равным 40, опциональная часть перестанет изменяться, т.е. количество дописываемых IP-адресов не может превышать 9.

Отправить IP-пакет Record Route можно при помощи утилиты ping с ключом –R:

```
linux:~$ ping -R –c 1 ya.ru
PING ya.ru (87.250.250.3) 56(124) bytes of data.
64 bytes from www.yandex.ru (87.250.250.3): icmp_req=14 ttl=57
time=16.1 ms
RR:   192.168.0.9
      izi.vlsu.ru (85.142.154.58)
      85.142.155.225
      vlsu.vladimir.runnet.ru (194.226.222.70)
      m9-ix.msk.runnet.ru (193.232.244.44)
      213.180.213.103
      l3-s1300-s3600.yandex.net (213.180.213.74)
      china-myt-vlan602.yandex.net (95.108.224.126)
      www.yandex.ru (87.250.250.3)
64 bytes from www.yandex.ru (87.250.250.3): icmp_req=78 ttl=57
time=234 ms     (same route)
--- ya.ru ping statistics ---
```

Различие в путях прохождения IP-пакета (до адресата и обратно) объясняется тем, что каждый маршрутизатор имеет не менее двух сетевых карт (с различными IP-адресами). В соответствии со стандартом, в опциональную часть будет дописываться IP-адрес

сетевой карты, с которой отправляется пакет (данное требование не всегда выполняется).

У метода Record Route выделяют три недостатка:

- Ограничение на количество транзитных узлов (до 9 в пути IP-пакета);
- Ошибки при модификации опциональной части IP-пакета некоторыми маршрутизаторами;
- Наличие результата только в случае успешного прохождения IP-пакета до адресата и обратно.

Принцип работы traceroute

Утилита traceroute — наиболее распространенное отладочное средство для определения маршрутов (трассировки) движения IP-пакетов между узлами и подсетями КСПД. Утилита traceroute использует ICMP-сообщения и 8-битное поле TTL в IP-заголовке. Поле TTL (время жизни – Time to live) это поле, которое отправитель устанавливает в какое-либо значение, рекомендуемое исходное значение указано в Assigned Numbers RFC и в настоящее время равно 64. С помощью поля TTL предотвращается зацикливание IP-пакетов в петлях маршрутизации.

Каждый маршрутизатор, который обрабатывает IP-пакет, уменьшает значение TTL на единицу или на количество секунд, в течение которых маршрутизатор обрабатывал IP-пакет. Современные требования к маршрутизаторам, Router Requirements RFC [Almquist 1993] (игнорируя требование RFC 1009 [Braden and Postel 1987] к маршрутизатору уменьшать значение поля TTL на количество секунд, если IP-пакет задерживается на время большее 1 секунды), позволяют маршрутизаторам использовать поле TTL в качестве счетчика пересылок (hop count).

Когда маршрутизатор получает IP-пакет со значением поля TTL равным 0 или 1, он уничтожает его и отправляет ICMP-сообщение об ошибке типа Time Exceeded (ICMP Type=11, Code=0 - Time To Live Exceeded In Transit) на исходный узел сети (источнику IP-пакета). Принцип работы traceroute (рис. 11) основан на том, что IP-пакет,

содержащий ICMP-сообщение Time Exceeded, имеет в качестве адреса источника IP-адрес маршрутизатора:

- На узел назначения отправляется IP-пакет со значением поля TTL, установленным в единицу. Первый маршрутизатор, на который попадает IP-пакет, уничтожает его (так как значение поля TTL равно 1) и отправляет ICMP сообщение Time Exceeded отправителю IP-пакета. Таким образом, определяется первый маршрутизатор в маршруте;
- На следующих этапах traceroute отправляет IP-пакеты со значением поля TTL на единицу превышающим предыдущие значения, что позволяет получить IP-адреса последующих маршрутизаторов;
- Алгоритм продолжается до тех пор, пока IP-пакет не достигнет узла назначения. Узел назначения генерирует ICMP-сообщение «порт недоступен» (port unreachable) т.к. утилита traceroute использует пакеты с несуществующими значениями UDP-портов (больше чем 30000).

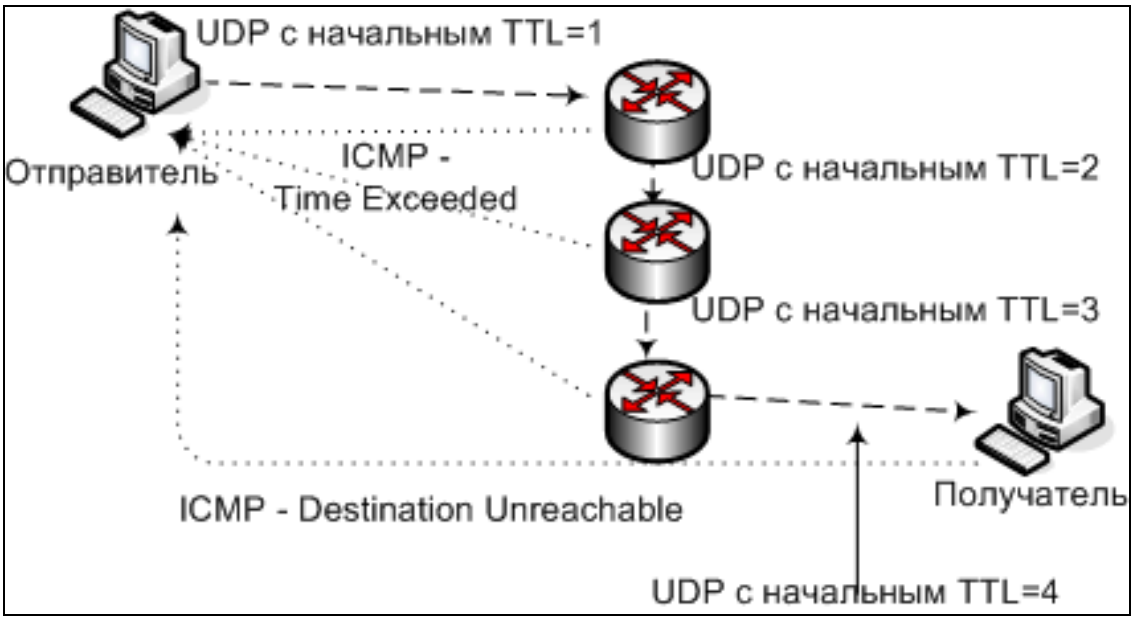


Рис. 11. Принцип работы traceroute

Пример использования протокола ICMP программой traceroute:

```
linux:~$ traceroute mail.ru
traceroute to ya.ru (87.250.250.203), 30 hops max, 60 byte packets
```

```
 1  192.168.0.2 (192.168.0.2)  0.377 ms  0.359 ms  0.348 ms
 2  85.142.154.49 (85.142.154.49)  3.167 ms  3.458 ms  3.532 ms
 3  85.142.155.226 (85.142.155.226)  1.018 ms  1.028 ms  1.024 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  7.891 ms  7.909 ms
7.903 ms
 5  msk-ix-m9.yandex.net (193.232.244.93)  7.915 ms  7.916 ms  7.909
ms
 6  213.180.213.102 (213.180.213.102)  8.004 ms  8.495 ms  8.461 ms
 7  l3-s3600-s1300.yandex.net (213.180.213.75)  9.305 ms  10.073 ms
10.069 ms
 8  www.yandex.ru (87.250.250.203)  9.467 ms  8.782 ms  8.754 ms
```

Первая строка, без номера содержит имя и IP-адрес узла назначения и указывает на то, что величина TTL не может быть больше 30.

В условиях фильтрации UDP-трафика на промежуточных узлах (распространенная ситуация, кроме UDP 53 - DNS), задача определения маршрутов усложняется. Пример работы traceroute в сети с фильтрацией UDP-трафика:

```
linux:~$ traceroute mail.ru
traceroute to mail.ru (94.100.191.204), 30 hops max, 60 byte packets
 1  192.168.0.2 (192.168.0.2)  0.316 ms  0.262 ms  0.232 ms
 2  85.142.154.49 (85.142.154.49)  0.751 ms  0.918 ms  0.948 ms
 3  85.142.155.226 (85.142.155.226)  0.668 ms  0.572 ms  0.544 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  7.848 ms  7.842 ms
7.931 ms
 5  m9-1-gw.msk.runnet.ru (194.85.40.226)  7.675 ms  7.728 ms  7.819
ms
 6  * * *
 7  * * *
 8  * * *
 9  * * *
```

Решением в таком случае может служить отказ от использования UDP-дейтаграмм. Пример использования протокола ICMP (как альтернативы UDP) программой traceroute:

```
linux:~$ traceroute -I mail.ru
traceroute to mail.ru (94.100.191.201), 30 hops max, 60 byte packets
 1  192.168.0.2 (192.168.0.2)  0.346 ms  0.335 ms  0.327 ms
 2  85.142.154.49 (85.142.154.49)  1.033 ms  1.076 ms  1.153 ms
 3  85.142.155.226 (85.142.155.226)  0.767 ms  0.829 ms  0.832 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  7.578 ms  7.659 ms
7.693 ms
```

```
 5  m9-1-gw.msk.runnet.ru (194.85.40.226)  7.568 ms  7.602 ms  7.608
ms
 6  mailru.msk.runnet.ru (194.190.254.234)  7.532 ms  7.835 ms
7.514 ms
 7  vl931.dl8.net.mail.ru (94.100.183.94)  27.175 ms  27.203 ms
27.185 ms
 8  94.100.191.201 (94.100.191.201)  7.730 ms  8.053 ms  8.023 ms
```

В случае фильтрации сообщений ICMP Echo, решение может быть следующим: необходимо использовать UDP-дейтаграмму на гарантированно открытый порт (обычно это порт 53 службы DNS). Пример использования протокола UDP программой traceroute:

```
linux:~$ traceroute  -U -p 53 mail.ru
traceroute to mail.ru (94.100.191.203), 30 hops max, 60 byte packets
 1  192.168.0.2 (192.168.0.2)  0.354 ms  0.301 ms  0.274 ms
 2  85.142.154.49 (85.142.154.49)  1.018 ms  1.037 ms  1.061 ms
 3  85.142.155.226 (85.142.155.226)  0.655 ms  0.674 ms  0.648 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  8.272 ms  8.226 ms
8.201 ms
 5  m9-1-gw.msk.runnet.ru (194.85.40.226)  8.073 ms  8.100 ms  8.079
ms
 6  mailru.msk.runnet.ru (194.190.254.234)  8.073 ms  7.817 ms
7.792 ms
 7  vl931.dl8.net.mail.ru (94.100.183.94)  7.811 ms  13.321 ms
13.319 ms
 8  94.100.191.203 (94.100.191.203)  13.250 ms  13.240 ms  13.161 ms
```

Аналогично можно использовать протокол TCP. Пример использования протокола TCP (пакет TCP SYN) программой traceroute:

```
linux:~$ traceroute  -T -p 80 mail.ru
traceroute to mail.ru (94.100.191.204), 30 hops max, 60 byte packets
 1  192.168.0.2 (192.168.0.2)  0.207 ms  0.178 ms  0.159 ms
 2  85.142.154.49 (85.142.154.49)  0.671 ms  0.808 ms  0.838 ms
 3  85.142.155.226 (85.142.155.226)  0.661 ms  0.602 ms  0.590 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  7.375 ms  7.419 ms
7.524 ms
 5  m9-1-gw.msk.runnet.ru (194.85.40.226)  7.336 ms  7.348 ms  7.453
ms
 6  * mailru.msk.runnet.ru (194.190.254.234)  7.670 ms  7.592 ms
 7  vl931.dl8.net.mail.ru (94.100.183.94)  7.877 ms  7.837 ms  7.929
ms
 8  94.100.191.204 (94.100.191.204)  7.607 ms  7.610 ms  7.538 ms
```

Подробнее об опциях traceroute можно узнать на страницах руководства:

```
linux:~$ traceroute --help
```

Существует специализированная реализация tcptraceroute, использующая TCP SYN пакеты, вместо традиционных для traceroute UDP или ICMP ECHO пакетов. Пример использования программы tcptraceroute:

```
linux:~$ tcptraceroute ya.ru
Selected device eth0, address 192.168.0.9, port 37214 for outgoing
packets
Tracing the path to ya.ru (87.250.250.203) on TCP port 80 (www), 30
hops max
 1  192.168.0.2  0.443 ms  0.335 ms  0.287 ms
 2  85.142.154.49  0.800 ms  0.672 ms  0.475 ms
 3  85.142.155.226  0.579 ms  0.627 ms  0.444 ms
 4  m9-3-gw.msk.runnet.ru (194.226.222.69)  7.744 ms  8.247 ms
7.434 ms
 5  msk-ix-m9.yandex.net (193.232.244.93)  8.300 ms  13.109 ms
7.892 ms
 6  213.180.213.102  9.756 ms  14.226 ms  8.667 ms
 7  l3-s3600-s1300.yandex.net (213.180.213.75)  9.933 ms  9.513 ms
9.770 ms
 8  www.yandex.ru (87.250.250.203) [open]  9.545 ms  9.425 ms  9.282
ms
```

3. Практическое задание

1. Произвести трассировку методом Record Route до следующих узлов сети: серверов DNS университета, пограничного маршрутизатора университета, yandex.ru, mail.ru, vkontakte.ru, elibrary.ru, google.com.
2. Составить перечень промежуточных узлов на каждом из путей прохождения IP-трафика с указанием IP-адресов их интерфейсов. Результаты занести в таблицу.

Таблица 5

№ маршрутизатора (DNS-имя)	IP-интерфейса	IP-интерфейса
1		
…		
9		

3. Произвести трассировку средствами утилиты traceroute до следующих узлов сети: серверов DNS университета, пограничного маршрутизатора университета, yandex.ru, mail.ru, vkontakte.ru, elibrary.ru, google.com.
4. Составить перечень промежуточных узлов на каждом из путей прохождения IP-трафика с указанием IP-адресов их интерфейсов. Сравнить с результатами, полученными методом Record Route. Сравнить значение RTT при трассировке утилитами traceroute и ping.

Практическая работа №8
ДОПОЛНИТЕЛЬНЫЕ СРЕДСТВА ОПРЕДЕЛЕНИЯ МАРШРУТОВ IP-ПАКЕТОВ - NMAP, TRACEMAP, MRT

1. Цель работы

Продолжить знакомство с методами определения маршрутов передачи данных в сетях TCP/IP. Научиться использовать программные средства определения маршрутов передачи данных: nmap, tracemap, MRT. Провести исследование сегмента сети университета средствами рассмотренных программных средств.

2. Теоретические сведения. Методические рекомендации

Утилиты ping и traceroute являются наиболее распространенными средствами определения маршрутов IP-пакетов, они входят в стандартную поставку большинства современных ОС. Их функционал достаточен для решения задачи трассировки маршрута в большинстве случаев. Однако существуют программные средства, обладающие дополнительными функциональными возможностями. В работе рассматриваются некоторые популярные некоммерческие программные продукты.

Сканер nmap как инструмент исследования топологии. Опция --traceroute

Сетевой сканер nmap также имеет возможность использования различных протоколов для построения трассы движения IP-пакета. Для определения маршрутов движения пакетов средствами nmap необходимо использовать опцию --traceroute. Пример применения протоколов ICMP и TCP сканером nmap:

```
linux:~$ nmap -sP 10.1.11.35 --traceroute
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-31 14:57 MSD
Nmap scan report for vla-hq-ns-01.hq.corp.vlsu.ru (10.1.11.35)
Host is up (0.00060s latency).
TRACEROUTE (using proto 1/icmp)
HOP RTT     ADDRESS
```

```
1   0.46 ms 192.168.0.1
2   1.89 ms 172.18.5.1
3   2.16 ms 10.11.11.101
4   2.13 ms 10.11.11.1
5   0.71 ms vla-hq-ns-01.hq.corp.vlsu.ru (10.1.11.35)
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds

linux:~$ nmap -sP mail.ru --traceroute -PT
Starting Nmap 5.21 ( http://nmap.org ) at 2011-08-31 14:59 MSD
Nmap scan report for mail.ru (94.100.191.204)
Host is up (0.0077s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.204
TRACEROUTE (using port 80/tcp)
HOP RTT     ADDRESS
1   0.22 ms 192.168.0.2
2   0.77 ms 85.142.154.49
3   0.64 ms 85.142.155.226
4   7.71 ms m9-3-gw.msk.runnet.ru (194.226.222.69)
5   7.68 ms m9-1-gw.msk.runnet.ru (194.85.40.226)
6   7.75 ms mailru.msk.runnet.ru (194.190.254.234)
7   7.80 ms vl931.dl8.net.mail.ru (94.100.183.94)
8   7.80 ms 94.100.191.204
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Визуализация маршрутов IP-пакета. Tracemap

Tracemap - скрипт для построения графической карты трассировки к хостам. Проверить наличие установленной программы tracemap можно командой:

```
linux:~$ whereis tracemap
```

В случае отсутствия программы tracemap ее можно получить следующим способом (необходимо подключение к сети Интернет):

```
linux:~$ wget http://xgu.ru/download/tracemap.pl
```

Для корректной работы программе могут понадобиться: traceroute, graphviz, libnet-ip-perl, программа просмотра PNG-файлов (например, gqview).

Опции программы tracemap:

-*q1:* Выполняется однократная отправка пакета для каждого TTL;

-*n:* Не выполняется обратное преобразование DNS;

-*I:* Выполняется сканирование с помощью ICMP (UDP по умолчанию).

Для построения трассы до узла достаточно передать его имя (адрес):

```
linux:~$ echo izi.vlsu.ru | perl tracemap.pl
```

Просмотреть результат (рис. 12) можно в любой программе для просмотра PNG-файлов, например:

```
linux:~$ gqview tracemap.png
```

Для того чтобы построить трассу к нескольким узлам, можно передать несколько адресов, по одному на строке:

```
linux:~$ (echo mail.ru; echo elibrary.ru; echo ya.ru; echo
google.com; echo vkontakte.ru) | perl tracemap.pl
readline() on closed filehandle PREFIXES at tracemap.pl line 44.
Tracing path to mail.ru........Done [last 7.563, total 45.46]
Tracing path to elibrary.ru.........Done [last 7.674, total 55.519]
Tracing path to ya.ru.......Done [last 8.693, total 35.131]
Tracing path to google.com.......Done [last 9.737, total 94.801]
Tracing path to vkontakte.ru.........Done [last 19.859, total
99.232]
```

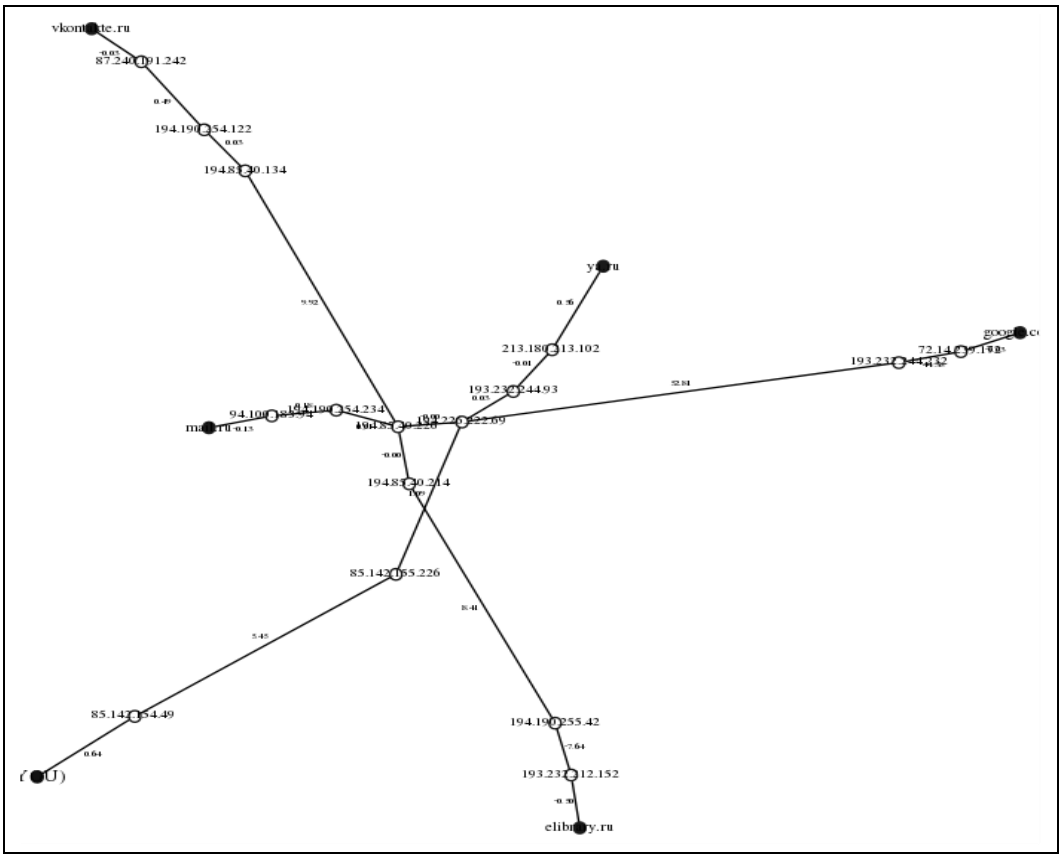


Рис. 12. Графическая карта трассировки к хостам, полученная с помощью tracemap

MTR

Рассмотренные выше средства построения путей прохождения IP-пакетов хорошо справляются со своей задачей, но их функционала не достаточно для диагностики, непрерывного мониторинга качества канала до узла, отслеживания изменения маршрутов и т.д. Подобные задачи можно решить, используя утилиту диагностики сети mtr (аналог под windows - WinMTR).

Программа mtr работает по принципу трассировок команды traceroute (tracert), однако позволяет выполнять диагностику сети в более удобном и наглядном режиме (рис. 13).

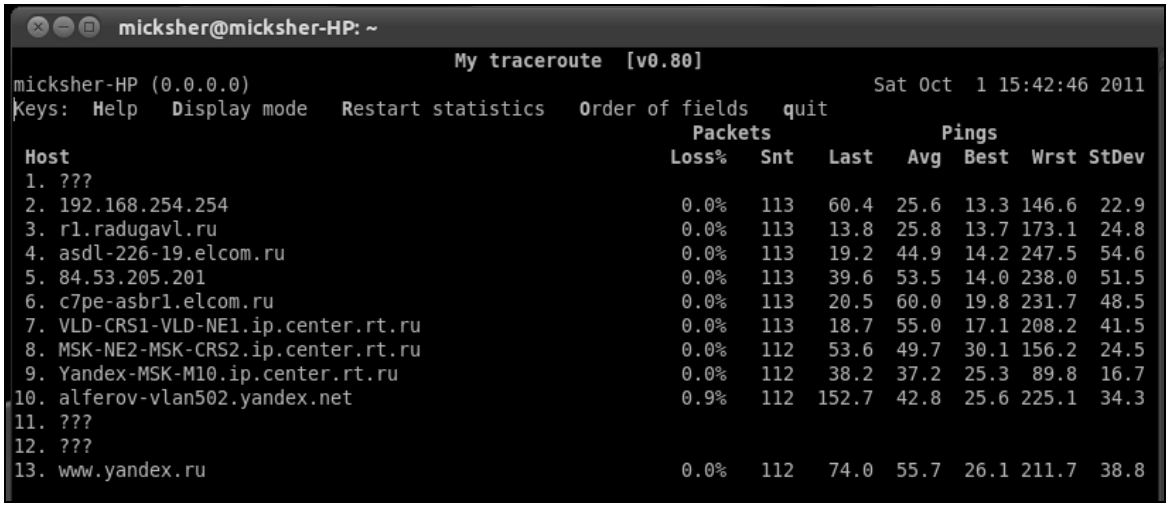


Рис. 13. Curses интерфейс mtr

Синтаксис:

```
mtr [-hvrctglspni46] [--help] [--version] [--report] [--report-
cycles COUNT] [--curses] [--split] [--raw] [--no-dns] [--gtk] [--
address IP.ADD.RE.SS] [--interval SECONDS] [--psize BYTES | -s
BYTES] HOSTNAME [PACKETSIZE]
```

Основные опции:

-h (--help): Вывод справочной информации;

-v (--version): Версия программы;

-r (--report): Помещает mtr в режим отчета. В этом режиме, mtr обработает количество циклов, определенных опцией -c, затем отобразит статистику и завершит работу. Этот режим полезен для генерации статистики о качестве сети;

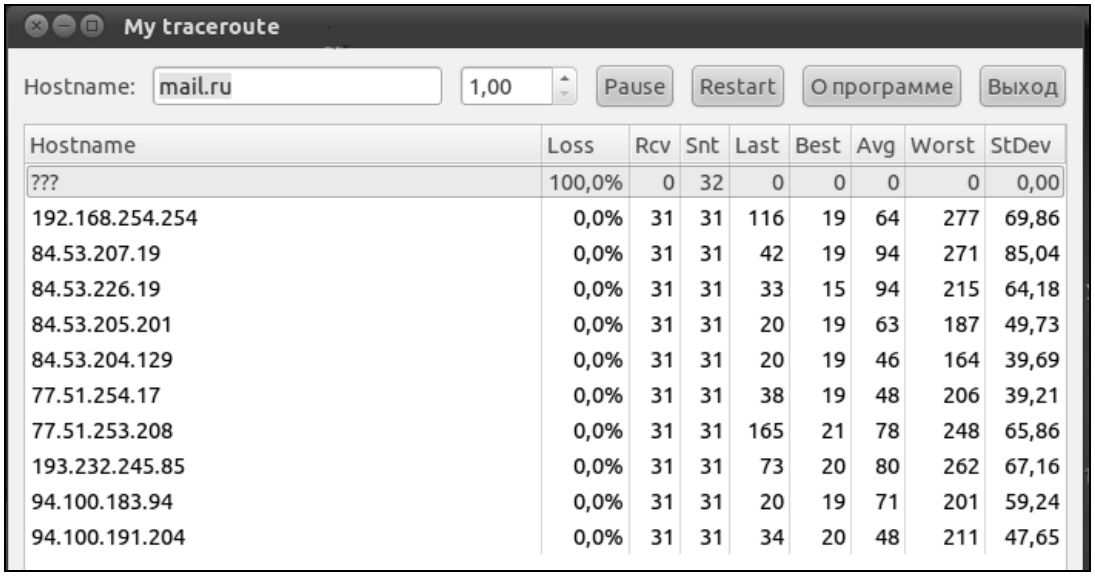
-c COUNT: Установить количество циклов, после которых mtr завершит работу;
-s BYTES: Размер посылаемых пакетов;
-t: Указывает mtr использовать curses based terminal interface (если доступно);
-n: Не использовать DNS.

Пример использования утилиты mtr:

```
linux:~$ mtr mail.ru –c 10 –n
linux:~$ mtr mail.ru –c 100 –r
```

Программа имеет графический (X) интерфейс (рис. 14):

```
linux:~$ mtr mail.ru –gtk –n
```



Hostname	Loss	Rcv	Snt	Last	Best	Avg	Worst	StDev
???	100,0%	0	32	0	0	0	0	0,00
192.168.254.254	0,0%	31	31	116	19	64	277	69,86
84.53.207.19	0,0%	31	31	42	19	94	271	85,04
84.53.226.19	0,0%	31	31	33	15	94	215	64,18
84.53.205.201	0,0%	31	31	20	19	63	187	49,73
84.53.204.129	0,0%	31	31	20	19	46	164	39,69
77.51.254.17	0,0%	31	31	38	19	48	206	39,21
77.51.253.208	0,0%	31	31	165	21	78	248	65,86
193.232.245.85	0,0%	31	31	73	20	80	262	67,16
94.100.183.94	0,0%	31	31	20	19	71	201	59,24
94.100.191.204	0,0%	31	31	34	20	48	211	47,65

Рис. 14. Графический (X) интерфейс mtr

3. Практическое задание

1. Постройте карту маршрутов IP-пакетов от одного из АРМ локальной сети до наиболее востребованных (в организации) ресурсов Интернет (не менее десяти ресурсов).
2. Средствами программы mtr исследуйте качество канала от компьютеров лаборатории (домашнего компьютера) до популярных ресурсов Интернет (ресурсов сети университета).
3. Отследите изменения маршрутов за период времени 60-180 минут (если таковые будут происходить). Выпишите все альтернативные маршруты.

Контрольные вопросы к разделу II

1. Методы определения маршрутов передачи данных в сетях TCP/IP.
2. Утилиты определения маршрутов передачи данных в сетях TCP/IP.
3. Метод определения маршрута Record Route. Достоинства и недостатки метода.
4. Утилита traceroute. Принцип определения маршрутов.
5. Использование протоколов ICMP и TCP при определении маршрутов.
6. Утилита tcptraceroute.
7. Сканер nmap как инструмент исследования топологии.
8. Утилита tracemap. Визуализация маршрутов.
9. Утилита диагностики сети mtr. Синтаксис и основные опции mrt.

РАЗДЕЛ III – ИДЕНТИФИКАЦИЯ СТАТУСА ПОРТА (PORT DETECTION)

Для идентификации (адресации) приложений на узлах сети существует система TCP (UDP) портов. Сканирование TCP (UDP) портов позволяет инвентаризировать службы сети, сделать предположения о роли узла в КСПД и установленных на нем приложениях и т.д.

Существуют несколько методов сканирования портов узла сети, в практических работах данного раздела будут рассмотрены методы: TCP-connect scanning, SYN scanning, Inverse TCP flag scanning, UDP Port Scanning.

В практических работах данного раздела будут использоваться следующие программные продукты: nmap, hping3, netcat.

Практическая работа №9
ИДЕНТИФИКАЦИЯ СТАТУСА TCP-ПОРТОВ (TCP-CONNECT. SYN-SCAN)

1. Цель работы

Изучить основы идентификации TCP-портов узла сети. Получить навык использования следующих методов идентификации статуса TCP-порта: TCP-connect Scanning, Half-open SYN flag scanning. Усвоить приемы решения задачи идентификации статуса TCP-порта с помощью утилиты nmap, hping3.

2. Теоретические сведения. Методические рекомендации

При сканировании TCP-портов (TCP Port Scanning) основным является метод сканирования с установлением соединения (TCP Connect Scanning). Метод сканирования с установлением соединения предполагает установление полноценного TCP-соединения со сканируемым узлом сети (с исследуемым TCP-портом) с

последующим разрывом этого соединения (разрыв производится стандартным образом).

На рис. 15 представлена диаграмма изменения состояний соединения TCP (TCP Connection State Diagram) описанная в документе RFC 793.

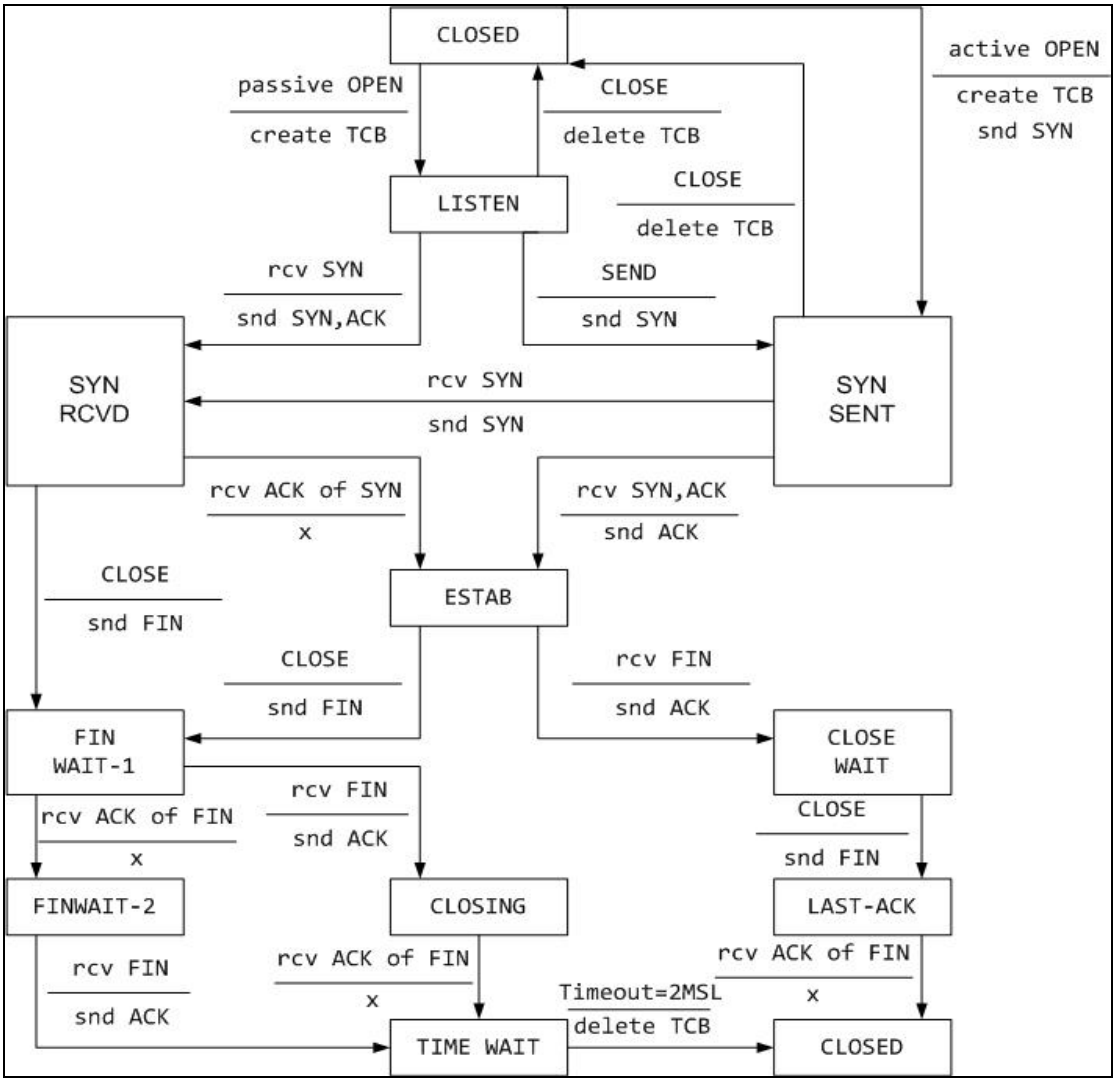


Рис. 15. Диаграмма изменения состояний соединения TCP (TCP Connection State Diagram)

Функция connect ()

Как правило, сканирование с установлением соединения реализует функция TCP connect(), данная функция присутствует в большинстве современных ОС. Функция connect() позволяет соединиться с любым TCP-портом узла сети. Если TCP-порт,

указанный в качестве параметра функции, прослушивается узлом (т.е. TCP-порт открыт для соединения), то результатом выполнения функции будет установление соединения с узлом по указанному TCP-порту. В противном случае, если соединение не установлено, то TCP-порт с указанным номером является закрытым.

Преимущества метода сканирования функцией connect ():

- Метод сканирования TCP-портов функцией connect() может применять непривилегированный пользователь (не обладающий правами root);
- Высокая скорость исследования. Возможность «параллельного просмотра» с использованием неблокированного соединения (non-blocked socket), определение состояния практически всех портов узла сети одновременно.

Основным недостатком метода сканирования TCP-портов функцией connect() является простота обнаружения и фильтрации такого рода сканирования.

Режим сканирования TCP-connect scan утилиты nmap

Режим TCP сканирования с использованием функции connect() сканера nmap устанавливается опцией –sT. Пример применения режима TCP сканирования сканером nmap:

```
linux:~$ nmap -sT mail.ru
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-01 11:46 MSD
Nmap scan report for mail.ru (94.100.191.202)
Host is up (0.0081s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.202
Not shown: 986 filtered ports
PORT     STATE  SERVICE
25/tcp   closed smtp
53/tcp   closed domain
80/tcp   open   http
88/tcp   closed kerberos-sec
110/tcp  closed pop3
143/tcp  closed imap
179/tcp  closed bgp
443/tcp  closed https
587/tcp  closed submission
993/tcp  closed imaps
Nmap done: 1 IP address (1 host up) scanned in 4.69 seconds
```

Сканер nmap имеет опции для определения портов для сканирования (по умолчанию nmap сканирует все порты до 1024 включительно), а также порядка сканирования (произвольного или последовательного):

- С помощью опции -p <диапазон портов> можно определить перечень портов для сканирования. Указание отдельных номеров портов допустимо, как и задание диапазонов портов разделенных дефисом (например, 1-1023);
- Опция -F (быстрое (ограниченные порты) сканирование) Указывает на необходимость сканирования только портов, заданных в nmap-services. Этот режим работает быстрее;
- По умолчанию nmap использует произвольный порядок сканирования портов. Обычно эта случайность нужна, но можно использовать прямой порядок сканирования, задав опцию –r (не использовать случайный порядок портов).

Пример определения диапазона портов для сканирования nmap:

```
linux:~$ nmap -sT mail.ru -p 80
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-01 12:03 MSD
Nmap scan report for mail.ru (94.100.191.203)
Host is up (0.0078s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.203
PORT   STATE SERVICE
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 0.27 seconds
```

Пример использования режима быстрого сканирования nmap:

```
linux:~$ nmap -sT -F mail.ru
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-01 12:20 MSD
Nmap scan report for mail.ru (94.100.191.202)
Host is up (0.0078s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.202
Not shown: 90 filtered ports
PORT    STATE  SERVICE
25/tcp  closed smtp
53/tcp  closed domain
80/tcp  open   http
88/tcp  closed kerberos-sec
110/tcp closed pop3
143/tcp closed imap
179/tcp closed bgp
```

```
443/tcp closed https
587/tcp closed submission
993/tcp closed imaps
Nmap done: 1 IP address (1 host up) scanned in 2.03 seconds
```

Сканирование TCP-портов флагом SYN

Метод SYN-сканирование известен как сканирование с установлением полуоткрытого соединения (Half-open SYN flag scanning), поскольку установление полного соединения с TCP-портом сканируемого узла сети не производится. Метод SYN-сканирования является одним из наиболее популярных. Сканирование в режиме полуоткрытого соединения требует возможности формировать одиночные TCP-сегменты в обход стандартного модуля TCP (необходимы права root).

При SYN-сканировании на порт сканируемого узла сети направляется SYN-сегмент. Получение ответного сегмента с флагами SYN|ACK означает, что порт открыт; получение сегмента с флагом RST означает, что порт закрыт. Получив SYN|ACK, сканер отправляет на обнаруженный порт сегмент с флагом RST, ликвидируя попытку соединения.

Режим SYN-сканирования утилиты nmap

Режим SYN-сканирования сканера nmap устанавливается опцией –sS (scan SYN). Пример применения режима SYN-сканирования:

```
linux:~$ sudo nmap -sS mail.ru
[sudo] password for linux:
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-01 17:13 MSD
Nmap scan report for mail.ru (94.100.191.201)
Host is up (0.0089s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.201
Not shown: 986 filtered ports
PORT     STATE  SERVICE
25/tcp   closed smtp
53/tcp   closed domain
80/tcp   open   http
88/tcp   closed kerberos-sec
110/tcp  closed pop3
143/tcp  closed imap
```

```
179/tcp  closed bgp
443/tcp  closed https
587/tcp  closed submission
993/tcp  closed imaps
Nmap done: 1 IP address (1 host up) scanned in 4.29 seconds
```

Для ускорения процесса опроса портов при сканировании больших сетей совместно с опцией -sS рекомендуется использовать опцию -PS <порт>, позволяющую опросить какой-либо порт на всех активных объектах сканируемой вами сети намного быстрее, чем при использовании одной опции -p:

```
linux:~$ sudo nmap -sS -n -p 80 -PS mail.ru
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-01 17:18 MSD
Nmap scan report for mail.ru (94.100.191.204)
Host is up (0.0082s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.204
PORT   STATE SERVICE
80/tcp open  http
Nmap done: 1 IP address (1 host up) scanned in 0.16 seconds
```

3. Практическое задание

1. Необходимо обнаружить все открытые порты TCP (22, 23, 80) сети лаборатории (компьютерного класса, экспериментальной установки), с использованием функции connect().
2. Повторите исследование, применяя метод Half-open SYN flag scanning.
3. Сравните время сканирования в первом и втором случае.
4. Повторите исследования с помощью утилиты hping3. Сравните результаты.

Практическая работа №10
МЕТОДЫ СКРЫТОГО СКАНИРОВАНИЯ (STEALTH TCP SCANNING METHODS)

1. Цель работы

Продолжить знакомство с основными методами идентификации TCP-портов. Научиться применять на практике утилиты nmap, hping3 для решения задачи идентификации статуса TCP-портов методами Stealth TCP scanning.

2. Теоретические сведения. Методические рекомендации

Изначально определение скрытое сканирование (Stealth TCP scanning methods) использовалось для методов сканирования, которые позволяли обходить сетевые системы обнаружения атак (ССОА) и систему протоколирования ОС. Современные ССОА способны обнаруживать Stealth TCP-сканирование. В настоящее время под Stealth TCP scanning methods понимают множество методов сканирования TCP-пакетами с нестандартным сочетанием флагов (Control Bits - 6 bits). Выделяют три группы методов скрытого сканирования:

- Инвертированное сканирование (Inverse TCP flag scanning);
- Сканирование пакетом с флагом ACK (ACK flag probe scanning);
- Сканирование фрагментированными TCP-пакетами (TCP fragmentation scanning).

Методы инвертированного сканирования

Методы инвертированного сканирования основаны на идентификации закрытых TCP-портов. Рассмотрим виды инвертированного сканирования (Inverse TCP flag scanning):

- SYN|ACK-сканирование (TCP SYN|ACK scanning);
- FIN-сканирование (TCP FIN scanning);
- XMAS-сканирование (TCP XMAS scanning);

- NULL-сканирование (TCP NULL scanning).

При Inverse TCP flag сканировании используют различные сочетания флагов заголовка отправляемого TCP-пакета. При обращении Inverse TCP пакетов к закрытым TCP-портам, исследуемый узел должен ответить RST-пакетом, все обращения таких пакетов к открытым TCP-портам должны игнорироваться (RFC 793).

Ключевой особенностью Inverse TCP flag scanning является их способность незаметно обойти некоторые не учитывающие состояние (non-stateful) МЭ. Недостатком методов инвертированного сканирования является большая вероятность ошибки, т.к. многие ОС игнорируют стандарты. Некоторые системы отвечают пакетом RST, при сканировании открытых TCP-портов, что противоречит RFC 793. Методы Inverse TCP flag scanning в настоящее время нашли свое применение при идентификации ОС (Remote OS Fingerprinting).

Сканирование TCP SYN|ACK

Метод SYN|ACK scanning считается более скрытым, чем метод SYN-сканирования. На исследуемый TCP-порт узла сети отправляется TCP-пакет с установленными флагами SYN и ACK. В случае закрытого исследуемого TCP-порта ситуация будет следующая:

```
Сканирующий узел (Client) -> TCP flags=SYN+ACK
          TCP flags=RST <- Сканируемый узел (Server)
```

В ответ на пришедший TCP-пакет с установленными флагами SYN и ACK сервер отвергнет соединение, ответив TCP-пакетом с установленным флагом сброса соединения - RST. В случае, когда исследуемый TCP-порт открыт, ситуация может быть аналогична предыдущей или будет следующая:

```
Сканирующий узел (Client) -> TCP flags=SYN+ACK
          (no response) <- Сканируемый узел (Server)
```

Пришедший на открытый TCP-порт TCP-пакет с установленными флагами SYN и ACK сервер может проигнорировать.

Метод TCP-сканирования пакетами с установленными флагами SYN и ACK малоинформативен. Если в ответ на отправленный SYN|ACK-пакет приходит RST-пакет, то считают, что фильтрация порта не осуществляется (реакция открытого TCP-порта и закрытого TCP-порта одинакова). Если ответа на отправленный SYN|ACK-пакет не последовало (или пришло ICMP-сообщение о недоступности порта), то TCP-порт считается фильтруемым. Метод SYN|ACK scanning невозможно использовать для определения статуса TCP-порта, однако данный метод позволяет отличить МЭ пакетный фильтр от МЭ экспертного уровня (StateFul Inspection).

Сканирование TCP FIN (FIN scanning)

При FIN scanning на исследуемый TCP-порт узла сети отправляется TCP-пакет с установленным флагом FIN. В соответствии с RFC 793 тестируемая система должна ответить TCP-пакетом с установленным флагом RST|ACK для всех закрытых TCP-портов, открытые порты должны игнорировать TCP(FIN)-пакеты.

Сценарий FIN scanning в случае, когда исследуемый TCP-порт закрыт:

```
Сканирующий узел (Client) -> TCP flags=FIN
     TCP flags=RST+ACK <- Сканируемый узел (Server)
```

Сценарий FIN scanning в случае, когда исследуемый TCP-порт открыт:

```
Сканирующий узел (Client) -> TCP flags=FIN
          (no response) <- Сканируемый узел (Server)
```

ОС Windows, BSDI, CISCO, HP/UX, MVS, IRIX и др. отвечают одинаково на TCP FIN пакеты во всех случаях, нарушая RFC 793.

Сканирование TCP Xmas

Метод TCP сканирования Xmas (рождественская елка) предполагает отправку TCP-пакета с установленными флагами FIN, URG, PSH. По RFC 793 тестируемая система должна ответить RST-пакетом для всех закрытых TCP-портов. Реакция различных ОС на TCP Xmas сканирование аналогична FIN scanning.

Нулевое сканирование (TCP Null scanning)

Метод TCP Null scanning основан на сбрасывании всех флагов отправляемого TCP-пакета. По RFC 793 тестируемая система должна ответить RST-пакетом для всех закрытых TCP-портов. Реакция различных ОС на TCP Null сканирование аналогична предыдущим.

Реализация Stealth scan средствами сканера nmap

Методы Stealth TCP scanning можно реализовать средствами сканера nmap, используя опции: -sF; -sN; -sX (TCP FIN, NULL и Xmas соответственно).

-sF: FIN сканирование (устанавливается только бит TCP FIN):

```
linux:~$ sudo  nmap -sF yandex.ru –p 11,22,80 -PN
```

-sN: Null сканирование (флагов в TCP-заголовке 0):

```
linux:~$ sudo  nmap -sN yandex.ru –p 11,22,80 -PN
```

-sX: Xmas сканирование (устанавливаются флаги FIN, PSH и URG):

```
linux:~$ sudo nmap -sX yandex.ru –p 11,22,80 -PN
```

3. Практическое задание

1. Исследуйте ответ различных ОС на сканирование методами Stealth scanning. Занесите в таблицу (табл. 6) результаты для различных методов сканирования и состояний TCP-порта (порт открыт, порт закрыт, порт открыт и фильтруется, порт закрыт и фильтруется).

Таблица 6

Адрес узла (IP, mac)	ОС	SYN|ACK	FIN	NULL	Xmas

2. Найдите в сети лаборатории (компьютерного класса, экспериментальной установки) узлы, защищаемые МЭ.

Практическая работа №11
МЕТОДЫ СКРЫТОГО СКАНИРОВАНИЯ
(ACK PROBE SCANNING, TCP FRAGMENTATION SCANNING)

1. Цель работы

Продолжить знакомство с основными методами идентификации TCP-портов. Научиться применять утилиты nmap, hping3 для решения задачи идентификации статуса TCP-портов методами ACK flag probe scanning, TCP Fragmenting.

2. Теоретические сведения. Методические рекомендации

ACK flag probe scanning

Этот метод сканирования не способен идентифицировать открытый порт (или даже открытый|фильтруемый). ACK flag probe scanning используется для выявления правил МЭ, определения фильтруемых TCP-портов, определения технологии МЭ (stateful, non-stateful).

Пакет запроса при ACK flag probe scanning содержит установленным только ACK-флаг. При сканировании систем, не фильтруемых МЭ, открытые и закрытые порты будут возвращать RST-пакет, т.е. они достижимы для ACK-пакетов (открыт порт или закрыт, определить невозможно). Порты, которые не отвечают при ACK flag probe scanning или отвечают ICMP-сообщением об ошибке Destination Unreachable (тип 3 - получатель недостижим, код 1, 2, 3, 9, 10 или 13), помечают как фильтруемые.

Сценарий ACK flag probe scanning в случае, когда исследуемый TCP-порт не фильтруется:

```
Сканирующий узел (Client) -> TCP flags=ACK
          TCP flags=RST <- Сканируемый узел (Server)
```

Сценарий ACK flag probe scanning в случае, когда исследуемый TCP-порт фильтруется МЭ:

```
Сканирующий узел (Client) -> TCP flags=ACK
     (no response/ICMP Type=3) <- Сканируемый узел (Server)
```

Для использования метода ACK flag probe scanning средствами сканера nmap необходимо указать опцию -sA (сканирование TCP ACK):

```
linux:~$ sudo  nmap -sA yandex.ru –p 11,22,80 -PN
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-22 23:09 MSD
Nmap scan report for yandex.ru (yandex.ru)
Host is up (0.000012s latency).
All 3 scanned ports on yandex.ru (yandex.ru) are unfiltered
Nmap done: 1 IP address (1 host up) scanned in 0.13 seconds
```

Analysis of the WINDOW field of received packets

Метод сканирования размера окна (WINDOW field of received packets) основан на анализе поля Initial Window TCP-пакета (полученного в ответ на RST пакет). В некоторых ОС открытые TCP-порты используют положительное значение поля Initial Window (даже в RST пакетах), а закрытые - нулевое. Поэтому при Window сканировании TCP-порты идентифицируют как открытые, если значение поля TCP Window положительно или закрытые, если значение равно нулю.

Для использования метода WINDOW field of received packets средствами сканера nmap необходимо указать опцию -sW (сканирование TCP Window).

3. Практическое задание

1. Исследуйте ответ различных ОС на сканирование методами ACK flag probe scanning, Window field of received packets. Результаты для различных методов сканирования и состояний TCP-порта (порт открыт, порт закрыт, порт открыт и фильтруется, порт закрыт и фильтруется) занесите в таблицу (табл. 7).
2. Необходимо обнаружить в сети лаборатории (компьютерного класса, экспериментальной установки) узлы, защищаемые МЭ.

Таблица 7

Адрес узла (IP, mac)	ОС	ACK flag probe scanning	Window field of received packets

Практическая работа №12
МЕТОДЫ СКАНИРОВАНИЯ UDP-ПОРТОВ (UDP PORT SCANNING). СКАНИРОВАНИЕ IP ПРОТОКОЛА

1. Цель работы

Освоить основные методы идентификации UDP-портов. Решить задачу идентификации статуса UDP-портов, применяя утилиты nmap, hping3.

2. Теоретические сведения. Методические рекомендации

Наиболее часто используемым транспортным протоколом для сервисов корпоративной сети является TCP. Служб, использующих в качестве транспорта UDP, значительно меньше. Самые распространенные из них: служба доменных имен DNS (Domain Name System, RFC 1034, RFC 1035; UDP-порт 53), служба управления сетью на базе протокола SNMP (Simple Network Management Protocol, RFC 1155, RFC 1212, RFC 1213, RFC 1157; UDP-порты 161/162), служба динамического назначения IP адресов DHCP (Dynamic Host Configuration Protocol, RFC 2131; UDP-порты 67/68), служба на базе простого протокола передачи файлов TFTP (Trivial File Transfer Protocol, RFC 1350; UDP-порт 69). Инвентаризация UDP-портов - UDP Port Scanning, несмотря на простоту протокола UDP, задача более сложная (в сравнении с TCP). Это связано с концепцией протокола UDP как транспортного протокола с негарантированной доставкой данных, выражающейся как в потере отдельных пакетов данных, так и в их дублировании.

Метод UDP Port Scanning заключается в следующем: на порт исследуемого узла сети отправляется пакет UDP, если узел отвечает сообщением ICMP Port Unreachable, можно делать вывод, что UDP-порт закрыт. В случае отсутствия ответа удаленного узла однозначного вывода сделать нельзя, ситуация может быть следующая: UDP-порт открыт, средствами МЭ фильтруется трафик

по UDP или ICMP протоколу, пакет потерялся (особенности протокола UDP).

Для решения проблемы потери пакетов рекомендуется увеличивать число отправляемых пакетов и время ожидания ответа при сканировании UDP Port Scanning. Для выявления фильтрации трафика по UDP или ICMP протоколу проводят предварительное сканирование узла сети. Некоторые диапазоны UDP-портов (например, 230-240 или 45200-45270) используются приложениями крайне редко, следовательно, сообщение ICMP Port Unreachable от узла должно быть получено с большой вероятностью. Отсутствие сообщений ICMP Port Unreachable может свидетельствовать о фильтрации трафика по протоколу UDP или ICMP средствами МЭ.

Еще одной особенностью UDP Port Scanning является медленная скорость сканирования. ОС узла ограничивает лимит сообщений о недостижимости порта. Например, ядро Linux ограничивает количество ICMP Port Unreachable до 80 сообщений за 4 секунды, а если это ограничение было превышено, то с простоем 0,25 секунды. К способам увеличения скорости UDP-сканирования относятся: параллельное сканирование нескольких хостов, сканирование в первую очередь наиболее популярных портов.

UDP Port Scanning средствами сканера nmap

Для использования метода UDP Port Scanning сканером nmap необходимо установить соответствующий режим опцией –sU. UDP-сканирование nmap осуществляет путем отправки пустого UDP-заголовка на каждый целевой порт удаленного узла сети. Если в ответ приходит сообщения ICMP Port Unreachable (Type 3, Code 3), nmap идентифицирует UDP-порт как закрытый. При получении ICMP-сообщения с другим кодом (Type 3, Code 1, 2, 9, 10 или 13) nmap идентифицирует UDP-порт как фильтруемый. Если после отправки нескольких пакетов, ответа получено не было, nmap идентифицирует UDP-порт как открыт|фильтруется, т.е. порт может быть открыт, или происходит фильтрация трафика МЭ.

Пример реализации UDP-сканирования сканером nmap:

```
linux:~$ sudo nmap -sU mail.ru --packet-trace
[sudo] password for linux:
Starting Nmap 5.21 ( http://nmap.org ) at 2011-09-14 23:13 MSD
Nmap scan report for mail.ru (94.100.191.201)
Host is up (0.037s latency).
Hostname mail.ru resolves to 4 IPs. Only scanned 94.100.191.201
Not shown: 999 open|filtered ports
PORT    STATE SERVICE
123/udp open  ntp
Nmap done: 1 IP address (1 host up) scanned in 65.03 seconds
```

UDP Port Scanning средствами утилит hping3, netcat

Генератор пакетов hping3 также поддерживает режимы UDP-сканирования. Для решения проблемы потери пакетов увеличим число отправляемых пакетов до десяти. Пример реализации UDP-сканирования утилитой hping3:

```
linux:~$ sudo hping3 -2 –p 53 mail.ru –c 10
HPING mail.ru (eth0 94.100.191.202): udp mode set, 28 headers + 0
data bytes
len=46 ip=10.1.11.35 ttl=124 DF id=8996 seq=6 rtt=0.5 ms
--- 10.1.11.35 hping statistic ---
10 packets transmitted, 1 packets received, 90% packet loss
round-trip min/avg/max = 0.5/0.5/0.5 ms
```

Опция hping3 -2 (--udp) задает режим UDP Port Scanning, -p (--destport) задает UDP-порт получателя, -c (--count) увеличивает число отправляемых пакетов.

Еще одной утилитой, часто применяемой для идентификации статуса портов, является netcat (или nc). Утилита netcat позволяет применять основные методы TCP- и UDP-сканирования. Параметр -z применяется для включения режима нулевого ввода-вывода (zero mode I/O). По умолчанию утилита netcat выполняет TCP-сканирование. Для указания режима UDP Port Scanning необходимо использовать параметр -u.

Синтаксис утилиты netcat:

```
nc [опции] host port
```

Некоторые опции netcat:

-*h:* Вывод справки;

-*v:* Вывод дополнительной информации (verbose);

-o [выходной файл]: Выводит дампы данных;
-i [число]: Пауза между отправляемыми данными (в секундах);
-z: Не отправлять данные (сканирование портов);
-u: Использовать UDP (по умолчанию используется TCP);
-l: Пассивный режим (прослушивание порта);
-p [число]: Локальный номер порта (для -l);
-n: Отключить разрешение DNS-имен и поиск номеров портов по /etc/services;
-w [число]: Интервал ожидания (в секундах).

Пример реализации UDP-сканирования утилитой netcat:

```
linux:~$ sudo nc -u -v -z -w2 mail.ru 1-140
[mail.ru] 53 (domain) open
```

3. Практическое задание

1. Проведите сканирование методом UDP Port Scanning некоторого количества узлов сети (уточнить у преподавателя).
2. Сравните время сканирования при различных режимах сканирования, различными инструментами (nmap, hping3, netcat). Результаты представьте в форме таблицы.
3. Исследуйте ответную реакцию на отправляемые пакеты при различных состояниях UDP-порта узла (порт открыт, порт закрыт, порт открыт и фильтруется, порт закрыт и фильтруется). Эксперимент повторите для различных ОС. Результаты оформите в виде таблицы.

Контрольные вопросы к разделу III

1. Методы идентификации TCP портов узла КСПД.
2. Состояния TCP соединения.
3. Метод TCP Connect Scanning.
4. Режим сканирования TCP Connect сканера nmap.
5. Метод Half-open SYN flag scanning.
6. Режим сканирования Half-open SYN flag сканера nmap.
7. Достоинства и недостатки методов Stealth TCP scanning.
8. Методы Inverse TCP flag scanning.
9. ACK flag probe scanning.
10. TCP fragmentation scanning.
11. Режимы Stealth TCP scanning сканера nmap.
12. Методы идентификации UDP-портов узла КСПД.
13. Реализация UDP Port Scanning сканером nmap. Опции и режимы сканера.
14. Реализация UDP Port Scanning средствами утилит hping3 и netcat. Опции и режимы.

РАЗДЕЛ IV – ИДЕНТИФИКАЦИЯ СЕТЕВЫХ СЕРВИСОВ И ПРИКЛАДНЫХ СЛУЖБ (SERVICES FINGERPRINTING)

Следующий (за определением статуса TCP/UDP портов) этап анализа защищенности – инвентаризация прикладных сетевых служб и приложений. Ввиду того, что большая часть уязвимостей приходится на уровень приложений, задача идентификации прикладных служб (сетевых приложений) является одной из важнейших. Соответствие TCP/UDP-портов прикладным службам регламентируется IANA (Internet Assigned Numbers Authority), с момента принятия в январе 2002 года RFC 3232 предусматривается ведение online базы такого соответствия (без закрепления в RFC). Актуальная версия списка размещена на сайте IANA http://www.iana.org/.

Номера TCP/UDP-портов разделены на три категории и находятся в диапазоне от 0 до 65535:

Таблица 8 – Категории TCP/UDP-портов

Номера портов	Категория	Описание
0 — 1023	Общеизвестные порты	Номера портов назначены IANA и на большинстве систем могут быть использованы исключительно процессами системы (или пользователя root).
1024 — 49151	Зарегистрированные порты	Номера портов включены в каталог IANA и на большинстве систем могут быть использованы процессами обычных пользователей или программами, запущенными обычными пользователями.
49152 — 65535	Динамически используемые порты.	Предназначены для временного использования в качестве клиентских портов, портов, используемых по согласованию для частных служб. Эти порты не могут быть зарегистрированы.

Часть TCP/UDP-портов закреплена за стандартными службами, часть используется в качестве диагностических и тестовых, для их использования не предназначены специальные программные средства, примерами таких портов являются ECHO (порт 7) и DAYTIME (порт 13). Просмотреть локальную копию официального списка соответствия между сетевыми службами и номерами портов можно в файле /etc/services в ОС Unix/Linux или в файле C:\Windows\system32\drivers\etc\services в ОС Windows.

```
linux:~$ less /etc/services
```

Однако, открытый TCP/UDP-порт не всегда соответствует стандартной прикладной сетевой службе. Причины использования нестандартного порта для службы могут быть самыми различными: отсутствие прав на использование стандартного порта службы, использование нескольких версий (например, для тестирования) одной службы, желание скрыть уязвимые службы от злоумышленников, ограничивающие доступ к защищенным портам настройки МЭ и т.д.

Для идентификации версий прикладных служб и сетевых приложений узла применяются следующие основные методы:

1. Анализ баннеров служб;
2. Исследование средствами команд прикладной службы;
3. Исследование особенностей работы служб прикладного уровня ISO OSI;
4. Эвристические методы.

Практическая работа №13
ИДЕНТИФИКАЦИЯ ПРИКЛАДНЫХ СЛУЖБ. МЕТОД АНАЛИЗА СТАНДАРТНЫХ ПРИГЛАШЕНИЙ (BANNER GRABBING)

1. Цель работы

Ознакомиться с основами идентификации прикладных сетевых служб. Получить практические навыки идентификации распространенных сетевых служб методом анализа стандартных приглашений.

2. Теоретические сведения. Методические рекомендации

Сбор баннеров (banner grabbing) - один из классических методов services fingerprinting (метод «снятия отпечатков пальцев» сетевых сервисов). Он заключается в опросе открытых в системе сервисов и

анализа возвращаемых ими стандартных приглашений (баннеров). Очень часто стандартные приглашения содержат информацию о службе и ее версии, иногда предоставляют дополнительную возможность для определения версии ОС.

Так, например, приглашение ftp-сервиса proftpd (Debian GNU/Linux) содержит следующую информацию:

```
linux:~$ telnet 192.168.0.15 21
Trying 192.168.0.15 ...
Connected to localhost.
Escape character is '^]'.
220 ProFTPD 1.3.3d Server (Debian)
421 Login timeout (300 seconds): closing control connection
Connection closed by foreign host.
```

Заголовок веб-сервера получим следующим образом:

```
linux:~$ echo ‘GET /HTTP/1.0 ’ | nc 192.168.0.9 80| grep "Server:"
Server: Apache/2.2.17 (Ubuntu)

linux:~$ echo ‘GET /HTTP/1.0 ’ | nc www.vlsu.ru 80| grep "Server:"
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny9 with Suhosin-Patch

linux:~$ echo ‘GET /HTTP/1.0 ’ | nc www.microsoft.com 80| grep
"Server:"
Server: Microsoft-HTTPAPI/2.0
```

Заголовок почтового сервера можно получить так:

```
linux:~$ telnet 192.168.0.9 25
Trying 192.168.0.9...
Connected to 192.168.0.9.
Escape character is '^]'.
220 VLAIZI427b ESMTP Postfix (Ubuntu)

linux:~$ telnet vpti.vladimir.ru 25
Trying 10.4.2.244...
Connected to vpti.vladimir.ru.
Escape character is '^]'.
220 vpti.vladimir.ru ESMTP CommuniGate Pro 4.1.8
```

Не составляют исключения и защищенные службы удаленного управления:

```
linux:~$ nc www.vlsu.ru 22
SSH-1.99-OpenSSH_5.1p1 Debian-5

linux:~$ nc www.bash.org.ru 22
```

```
SSH-2.0-OpenSSH_5.1p1 Debian-5
```

Из достоинств метода banner grabbing выделим простоту реализации, метод не требует специализированных программных инструментов - достаточно стандартных утилит ОС (telnet, netcat). Однако необходимо учитывать, что многие службы позволяют администратору модифицировать свои стандартные приветствия, то есть рассмотренный метод получения баннеров не является достоверным, его использование рекомендуется в совокупности с другими методами.

3. Практическое задание

1. Необходимо обнаружить в сети лаборатории (диапазон IP адресов уточнить у преподавателя) узлы сети, на которых открыты следующие порты: 53/UDP, 80/TCP, 22/TCP, 23/TCP, 25/TCP, 57/TCP, 110/TCP, 143/TCP, 443/TCP.
2. Сделайте предположение о версии ПО соответствующих служб на основе анализа стандартных приглашений служб. Определите версии ПО исследуемых служб локально. Сравните полученные результаты.
3. Определите версии ПО служб DNS, HTTP, HTTPS, SMTP серверов сети университета, популярных серверов Интернет (необходим доступ к сети Интернет).

Практическая работа №14
ИДЕНТИФИКАЦИЯ ПРИКЛАДНЫХ СЕТЕВЫХ СЛУЖБ МЕТОДОМ АНАЛИЗА ОСОБЕННОСТЕЙ РЕАЛИЗАЦИИ (SMTP)

1. Цель работы

Продолжить изучение методов идентификации сетевых служб. Получить практические навыки идентификации сетевых служб методом анализа их реализации на примере службы электронной почты (SMTP).

2. Теоретические сведения. Методические рекомендации

Более достоверным является метод services fingerprinting основанный на анализе особенностей работы (реализации) прикладной службы. Суть метода заключается в отправке нестандартных запросов или применении малоиспользуемых опций и команд прикладного протокола. Алгоритмы этого типа характеризуются высокой скоростью, достаточной точностью и слабой скрытностью. Т.е. попытка сбора информации может быть довольно легко обнаружена и пресечена исследуемой системой или системой обнаружения вторжений.

Протокол SMTP

SMTP (Simple Mail Transfer Protocol) — простой протокол передачи почты в сетях TCP/IP, описан в RFC 5321, RFC 821, RFC 1425, RFC 1985, используемый порт 25/TCP. Простой протокол передачи почты обеспечивает двухсторонний обмен сообщениями между локальным клиентом и удаленным сервером MTA. Стандарты RFC определяют команды SMTP, допустимые аргументы, данные, сообщения об ошибках и т.д. Основные команды SMTP представлены в таблице 9.

Таблица 9 - Основные команды SMTP

Команда	Описание
HELO	Идентифицирует модуль-передатчик для модуля-приемника (hello).
MAIL	Начинает почтовую транзакцию, которая завершается передачей данных в один или несколько почтовых ящиков (mail).
RCPT	Идентифицирует получателя почтового сообщения (recipient).
DATA	Строки, следующие за этой командой, рассматриваются получателем как данные почтового сообщения. В случае SMTP, почтовое сообщение заканчивается комбинацией символов: CRLF-точка-CRLF.
RSET	Прерывает текущую почтовую транзакцию (reset).
NOOP	Требует от получателя не предпринимать никаких действий, а только выдать ответ ОК. Используется главным образом для тестирования (No operation).
QUIT	Требует выдать ответ ОК и закрыть текущее соединение.
VRFY	Требует от приемника подтвердить, что ее аргумент является действительным именем пользователя.
SEND	Начинает почтовую транзакцию, доставляющую данные на один или несколько терминалов (а не в почтовый ящик).
SOML	Начинает транзакцию MAIL или SEND, доставляющую данные на один или несколько терминалов или в почтовые ящики.
SAML	Начинает транзакцию MAIL и SEND, доставляющие данные на один или несколько терминалов и в почтовые ящики.
EXPN	Команда SMTP-приемнику подтвердить, действительно ли аргумент является адресом почтовой рассылки и если да, вернуть адрес получателя сообщения (expand).
HELP	Команда SMTP-приемнику вернуть сообщение-справку о его командах.
TURN	Команда SMTP-приемнику либо сказать ОК и поменяться ролями, то есть стать STMP- передатчиком, либо послать сообщение-отказ и остаться в роли SMTP-приемника.

В соответствии со спецификацией команды: HELO, MAIL, RCPT, VRFY - обязаны присутствовать в любой реализации SMTP. Остальные команды SMTP могут быть реализованы дополнительно. Каждая SMTP-команда должна заканчиваться либо пробелом (если у нее есть аргумент), либо комбинацией CRLF.

Протокол SMTP требует, чтобы сервер отвечал на каждую команду SMTP-клиента. MTA-сервер отвечает трехзначной комбинацией цифр, называемой кодом ответа. Первая цифра в коде ответа означает, было ли выполнение команды успешно (2), неуспешно (5) или еще не закончилось (3). Как указано в приложении E документа RFC 821, SMTP-клиент может анализировать только первую цифру в ответе сервера, и на основании ее продолжать свои

действия. В том же RFC 821 приведены только коды команд, варианты значений ответов и описания могут отличаться в различных реализациях серверов.

Метод синтаксических искажений команд протокола SMTP

Процесс определении типа и версии программного обеспечения сервера может заключаться в отправке серверу определенного множества команд SMTP, в которых намеренно допущены различного рода синтаксические искажения (регистр, порядок команд, пропуск обязательных параметров и т.д.) и сравнении полученных данных с собранной базой ответов SMTP-серверов (профилем серверов).

Примеры синтаксического искажения команд, выявляющего особенности реализации SMTP серверов:

- Корректно заданная команда MAIL FROM без предварительной команды HELO;
- Команда HELO без указания имени домена;
- Команда MAIL FROM <имя> без символа «:»;
- Команда MAIL FROM: с пустым адресатом;
- Некорректный адрес отправителя в команде MAIL FROM.

Как правило, различаются не только значения кодов ответов (результат ответа на команду HELO, например, может быть 501, а может – 250), но и по описанию кодов («501 5.5.2 Syntax error in parameters scanning» сервера sendmail или «501 Syntax error» SMTP relay под управлением Checkpoint).

Кроме этого рекомендуется исследовать реакцию сервера на малоиспользуемые команды, например команды VRFY и EXPN (при настройке безопасности почтового сервера они могут быть отключены). По спецификации команда VRFY предназначена для проверки существования пользователя. Команда EXPN предназначена для получения расширенной информации о пользователе (в том числе его реальной фамилии и имени) и почтовых группах.

И наконец, следует проверить поддержку сервером таких команд, как: HELP, TURN, SOML, SAML, NOOP, EHLO.

3. Практическое задание

1. Установите наиболее популярные свободно распространяемые сервера электронной почты: postfix, sendmail, qmail, exim4 (перечень и версии уточнить у преподавателя).
2. Экспериментально исследуйте особенности их работы:
   - исследуйте ответы на синтаксически верные стандартные команды;
   - исследуйте ответы на команды с ошибками;
   - исследуйте ответы на команды VRFY и EXPN;
   - исследуйте поддержку малоиспользуемых команд.
3. Варианты ответов оформите в виде таблицы.
4. Подготовьте профили каждого исследуемого сервера.
5. Проведите исследование почтовых серверов Интернет / университета (адреса уточнить у преподавателя) рассмотренным методом.
6. Сравните полученные данные с подготовленными профилями. Сделайте предположения в версиях ПО исследуемых серверов.

Практическая работа №15
ИДЕНТИФИКАЦИЯ СЛУЖБЫ ЭЛЕКТРОННОЙ ПОЧТЫ МЕТОДОМ MAIL-BOUNCING

1. Цель работы

Продолжить изучение методов идентификации сетевых служб. Овладеть практическими навыками идентификации службы электронной почты (SMTP) методом mail-bouncing.

2. Теоретические сведения. Методические рекомендации

Метод mail-bouncing

Метод основан на анализе заголовков электронных писем, полученных от SMTP-сервера. Наиболее информативны электронные письма для несуществующих пользователей, в ответ на такие письма сервер возвращает уведомления о невозможности доставки (NDR, Non-Delivery Report):

```
Received: from VLA-HQ-EXC-01.hq.corp.vlsu.ru (10.1.21.21) by
mx1.vlsu.ru
 (10.1.25.32) with Microsoft SMTP Server (TLS) id 14.1.323.3; Sun,
18 Sep 2011
 12:05:13 +0400
Received: from vpti.vladimir.ru (10.4.2.244) by VLA-HQ-EXC-
01.hq.corp.vlsu.ru
 (10.1.21.21) with Microsoft SMTP Server id 14.1.323.3; Sun, 18 Sep
2011
 12:05:35 +0400
Received: from [84.53.214.152] (account testuser@izi.vlsu.ru HELO
 [192.168.1.27])  by vpti.vladimir.ru (CommuniGate Pro SMTP 4.1.8)
with ESMTP
 id 28911459 for bvrbewbewrbr@mail.ru; Sun, 18 Sep 2011 12:05:34
+0400
Subject: Test mail-bouncing
From: Test User <testuser@izi.vlsu.ru>
To: <bvrbewbewrbr@mail.ru>
Content-Type: text/plain
Date: Sun, 18 Sep 2011 12:05:21 +0400
Message-ID: <1316333121.11201.1.camel@testuser-HP>
MIME-Version: 1.0
```

```
X-Mailer: Evolution 2.32.2
Content-Transfer-Encoding: 7bit
Return-Path: testuser@izi.vlsu.ru
```

Анализ таких уведомлений позволяет получить некоторую информацию о почтовых серверах, участвующих в процессе доставки письма. Структура записи о пути электронного письма (согласно RFC 821) имеет следующую структуру:

```
Received: from ОТПРАВИТЕЛЬ by ПОЛУЧАТЕЛЬ with ПРОТОКОЛ ID, Date,
Time, GMT
```

Рассмотрим несколько фрагментов уведомлений о невозможности доставки письма от различных серверов:

```
Received: from VLA-HQ-EXC-01.hq.corp.vlsu.ru (10.1.21.21) by
mx1.vlsu.ru (10.1.25.32) with Microsoft SMTP Server (TLS) id
14.1.323.3; Sat, 17 Sep 2011 23:25:37 +0400

Received: from [192.168.1.27] ([84.53.214.255]) by mx.google.com
with ESMTPS id f15sm9509130bke.2.2011.09.17.12.35.43 (version=SSLv3
cipher=OTHER); Sat, 17 Sep 2011 12:35:44 -0700 (PDT)

Received: from [84.53.214.255] (port=60181 helo=[192.168.1.27])by
smtp8.mail.ru with psmtp id 1R50mv-0006Ho-00 for "cqk"@yandex.ru;
Sat, 17 Sep 2011 23:42:57 +0400

Received: by 10.204.143.17 with SMTP id
s17mr633005bku.207.1316327084322; Sat, 17 Sep 2011 23:24:44 -0700
(PDT)
```

Несколько фрагментов уведомлений о невозможности доставки письма от одного сервера:

```
Received: from [84.53.214.152] (account mishin@izi.vlsu.ru HELO
[192.168.1.27])  by vpti.vladimir.ru (CommuniGate Pro SMTP 4.1.8)
with ESMTP id 28911459 for bvrbewbewrbr@mail.ru; Sun, 18 Sep 2011
12:05:34 +0400

Received: from [84.53.214.152] (account mishin@izi.vlsu.ru HELO
[192.168.1.27])  by vpti.vladimir.ru (CommuniGate Pro SMTP 4.1.8)
with ESMTP id 28911463 for vbkjvbwkjvwdvbwkdbvwkdvdw@mail.ru; Sun,
18 Sep 2011 12:05:49 +0400

Received: from [84.53.214.152] (account mishin@izi.vlsu.ru HELO
[192.168.1.27])  by vpti.vladimir.ru (CommuniGate Pro SMTP 4.1.8)
with ESMTP id 28911469 for frgfwgwrgwgwrvrwvrw@mail.ru; Sun, 18 Sep
2011 12:09:25 +0400
```

Отличительным признаком может служить тег «ПРОТОКОЛ». Пример демонстрирует, что вместо требуемого стандартом with SMTP, разработчики предлагают следующее: with Microsoft SMTP Server, with ESMTPS, with psmtp, with SMTP.

Особое внимание следует уделить тегу «ID», RFC 821 формат ID описывает следующим образом: <id>::= «ID» <SP> <string> <SP> ,то есть RFC не накладывает жестких требований к формату тега «ID», и каждый разработчик почтовых серверов может использовать собственный формат.

Создадим сигнатуру, описывающую формат заголовка письма (тег «ID»), уникальную для версии SMTP сервера. Сигнатура будет представлять бинарную последовательность, каждая позиция которой будет описывать наличие (1) или отсутствие (0) некоторого характерного признака ID-тега заголовка письма.

Пример возможных признаков:

- Все символы ID-тега в верхнем регистре;
- Все символы ID-тега в нижнем регистре;
- Все символы ID-тега являются цифрами;
- Среди символов ID-тега отсутствуют цифры;
- Среди символов ID-тега присутствуют спецсимволы («-», «_», «.»,«,» и т.д.);
- Среди спецсимволов ID-тега присутствуют только символы «.»;
- Длина (количество символов) ID-тега (отведем 6 разрядов).

Опишем ID теги рассмотренных выше заголовков по предложенной схеме:

Таблица 10

ID-тег	Сигнатура	Тег «ПРОТОКОЛ»
14.1.323.3	001011001010	Microsoft SMTP Server
f15sm9509130bke.2.2011.09.17.12.35.43	010011100101	ESMTPS
1R50mv-0006Ho-00	000010010000	psmtp
s17mr633005bku.207.1316327084322	010011100000	SMTP
28911459	001000001000	ESMTP
28911463	001000001000	ESMTP
28911469	001000001000	ESMTP

3. Практическое задание

1. Предложите 3-7 возможных признаков ID-тега заголовка электронного письма.
2. Получите по 5-7 уведомлений о невозможности доставки письма от серверов электронной почты: postfix, sendmail, qmail, exim4 (используйте экспериментальные установки предыдущей работы).
3. На основе полученных уведомлений рассчитайте сигнатуры, описывающие формат заголовков письма, для исследуемых версий SMTP-серверов.
4. Сформируйте базу сигнатур (сигнатура, тег «Протокол») для дальнейшего исследования.
5. Проведите исследование почтовых серверов Интернет / университета методом mail-bouncing (адреса уточнить у преподавателя), сделайте предположение в версиях ПО исследуемых серверов Интернет / университета на основе собственной базы сигнатур. Сравните с результатами, полученными в предыдущих работах.

Практическая работа №16
СПЕЦИАЛИЗИРОВАННЫЕ ПРОГРАММНЫЕ СРЕДСТВА ИДЕНТИФИКАЦИИ ПРИКЛАДНЫХ СЛУЖБ

1. Цель работы

Продолжить изучение методов удаленной идентификации прикладных служб. Научиться применять на практике специализированные программные средства для идентификации прикладных сетевых служб: amap, strobe, nmap.

2. Теоретические сведения. Методические рекомендации

Amap

Amap (Application MAPper) – один из первых специализированных инструментов для проведения активного Services Fingerprinting в сетях TCP/IP (http://www.thc.org). Amap позволяет удаленно идентифицировать сетевые приложения и сервисы на основе методов TCP/IP Stack Fingerprinting и анализа баннеров. Несмотря на то, что методы идентификации прикладных служб в других сканерах (например, nmap) реализованы зачастую лучше, amap и сейчас может применяться совместно с другими инструментами.

Изначально Amap не был предназначен для Port detection, поэтому необходимо было определить статус портов исследуемого узла сети дополнительно. Пример совместного использования утилит amap и nmap:

```
linux:~$ sudo nmap -sU -p 1-1024 -oM result.nmap 10.1.11.35
linux:~$ sudo amap -i result.nmap -o result.amap –m
```

На первом шаге средствами nmap определяются статусы первых 1024 UDP-портов узла 10.1.11.35, результаты сканирования сохраняются в файл «result.nmap». На втором шаге средствами amap производится идентификация сетевых приложений, использующих

найденные на первом шаге UDP-порты. Результат работы утилиты amap сохраняется в файл «result.amap».

Синтаксис утилиты amap:

```
amap [Mode] [Options] <target> <port/portrange> [<port> ...]
```

Amap может работать в трех различных режимах (modes):

-*A* (Map applications): режим, при котором amap формирует специализированные запросы (triggers) на исследуемый порт и анализирует полученные ответы. Данный режим используется по умолчанию. Режим Map applications позволяет применять все опции amap.

-*B* (Banner): режим снятия стандартных приглашений прикладных служб. Могут применяться только опции, предназначенные для данного режима (в документации отмечены как «Banner»).

-*P* (Portscan): режим определения статуса порта. В режиме Portscan версии служб не определяются. Могут применяться только опции, предназначенные для данного режима (в документации отмечены как «Portscan»).

Способы задание целевого узла amap:

-*i <file>:* опция указывает amap получить целевые узлы сети для сканирования из файла. Файл может быть создан утилитой nmap;

<target> and *<port/portlist>:* целевой узел может быть задан как IP адресом, так и DNS именем. Целевой порт может быть задан числом от 1 до 65535 или диапазоном (например 22-85). По умолчанию amap работает с TCP портами удаленного узла сети.

Некоторые наиболее используемые опции amap:

-*u:* использование протокола UDP (по умолчанию используется TCP), опция применима для всех режимов;

-*6:* использование IPv6 (по умолчанию используется IPv4);

-*v:* подробный вывод, опция применима для всех режимов;

-*b:* вывод в режиме ACSII;

-*o <file>:* сохранять результаты в файл;

-*h:* краткая справка.

Пример использования утилиты amap (режимы -*P*, -*B*, -*A*):

```
linux:~$ amap -P -bqv izi.vlsu.ru 80
```

```
amap v5.4 (www.thc.org/thc-amap) started at 2011-10-01 19:51:41 -
PORTSCAN mode
Total amount of tasks to perform in plain connect mode: 1
Waiting for timeout on 1 connections ...
Port on 85.142.154.58:80/tcp is OPEN
amap v5.4 finished at 2011-10-01 19:51:42

linux:~$ amap -B -bqv izi.vlsu.ru 22
amap v5.4 (www.thc.org/thc-amap) started at 2011-10-01 19:53:34 -
BANNER mode
Total amount of tasks to perform in plain connect mode: 1
Waiting for timeout on 1 connections ...
Banner on 85.142.154.58:22/tcp : SSH-2.0-OpenSSH_5.5p1 Debian-4\r\n
amap v5.4 finished at 2011-10-01 19:53:35

linux:~$ amap -A www.vlsu.ru 80
amap v5.4 (www.thc.org/thc-amap) started at 2011-10-01 20:44:01 -
APPLICATION MAPPING mode
Protocol on 85.142.154.99:80/tcp matches http
Protocol on 85.142.154.99:80/tcp matches http-apache-2
Unidentified ports: none.
amap v5.4 finished at 2011-10-01 20:44:07
```

Утилита strobe

Утилита strobe — это популярный сканер TCP-портов. К достоинствам strobe относят быстроту сканирования, высокую точность результатов. Основной метод, реализуемый утилитой - banner grabbing. Из недостатков сканера выделяют легкое обнаружение сканирования на исследуемой системе (strobe выполняет TCP сканирование методом connect), отсутствие поддержки протокола UDP.

Синтаксис утилиты strobe:

```
strobe [Options] [host1 ... [hostn]]
```

Краткую справку по основным опциям утилиты strobe можно получить следующим образом:

```
linux:~$ strobe -h
```

Пример использования утилиты strobe:

```
linux:~$ strobe www.vlsu.ru –f –n 120
strobe 1.05 (c) 1995-1999 Julian Assange <proff@iq.org>.
www.vlsu.ru    80 http         www www-http World Wide Web HTTP
                  www          World Wide Web HTTP [TXL]
www.vlsu.ru   111 sunrpc       rpcbind SUN Remote Procedure Call

linux:~$ strobe -p 80 www.vlsu.ru www.izi.vlsu.ru www.ya.ru -o
~/out.txt
```

Результат сканирования будет сохранен в файл out.txt в домашней директории пользователя.

Сканер nmap (режим определения версий)

В предыдущих работах были рассмотрены практические примеры использования сетевого сканера nmap при обнаружении активных узлов сети, идентификации открытых портов. Для решения задачи идентификации прикладных сетевых служб, в сканере nmap предусмотрен режим определения версий (version-intensity). Сканер задействует собственные алгоритмы определения протокола службы, идентификации приложения, номер версии приложения и т.д.

Пример использования режима определения версий nmap:

```
linux:~$ nmap -sV www.vlsu.ru
Starting Nmap 5.21 ( http://nmap.org ) at 2011-10-02 00:46 MSK
Nmap scan report for www.vlsu.ru (85.142.154.99)
Host is up (0.052s latency).
rDNS record for 85.142.154.99: ip4host-154-99.vlsu.ru
Not shown: 993 closed ports
PORT    STATE    SERVICE      VERSION
22/tcp  open     tcpwrapped
53/tcp  open     domain       ISC BIND 9.6-ESV-R1
80/tcp  open     http         Apache httpd 2.2.9 ((Debian)
PHP/5.2.6-1+lenny9 with Suhosin-Patch)
111/tcp open     rpcbind      2 (rpc #100000)
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds
Service detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.85 seconds
```

Опция -sV сканера указывает на использование функции определения версий сетевых служб.

При сканировании в режиме version-intensity, nmap отправляет на исследуемый порт узла сети серию тестовых запросов, ранжированных по девятибальной шкале (запросы с низким рангом эффективны для большинства типовых служб). Ранг запросов (интенсивность сканирования), как правило, пропорционален вероятности корректной идентификации сетевой службы и обратно пропорционален времени сканирования (по умолчанию уровень интенсивности равен 7). Уровень интенсивности может быть изменен с помощью опции --version-intensity в интервале от 0 до 9.

Для отслеживания хода сканирования можно воспользоваться опциями --version-trace или --packet-trace, отображающими подробную отладочную информацию. Пример задания опций nmap:

```
linux:~$ nmap -sV www.vlsu.ru --version-intensity 9 --version-trace
–o result.txt
```

3. Практическое задание

1. Установите утилиту amap (получите архив у преподавателя или с официального сайта разработчика программы http://www.thc.org/thc-amap/):

```
linux:~$ cd ~
linux:~$ wget http://freeworld.thc.org/releases/amap-5.4.tar.gz
linux:~$ tar xzvf amap-5.4.tar.gz
linux:~$ cd amap-5.4/
linux:~$ ./configure
linux:~$ make
linux:~$ sudo make install
```

2. Получите IP адреса всех узлов сети лаборатории, на которых открыты 80-TCP и 53-UDP порты. Используйте утилиту amap в режиме Portscan.

3. Проведите исследование открытых портов одного из найденных узлов сети. Используйте утилиту amap в режимах Banner и Map applications. Сравните результаты работы программы в различных режимах.

4. Установите TCP-сканер strobe.

```
linux:~$ apt-get update && apt-get install strobe
```

5. Повторите исследование прикладной службы 80-TCP порта (узел сети предыдущего эксперимента) с помощью сканера strobe. Сравните результаты.
6. Повторите исследование средством сканера nmap. Результаты проведенных экспериментов оформите в виде таблицы. Сделайте выводы.
7. Проведите version-intensity сканирование десяти узлов сети лаборатории сканером nmap в режимах интенсивности 0, 7, 9. Сравните результаты и время сканировани. Результаты оформите в виде таблицы.

Контрольные вопросы к разделу IV

1. Задача идентификации сетевых служб. Соответствие TCP/UDP-портов и сетевых служб.
2. Методы идентификации версий прикладных служб. Services fingerprinting.
3. Метод banner grabbing. Достоинства и недостатки метода.
4. Методы анализа особенностей реализации прикладной службы.
5. Метод mail-bouncing. Идентификация службы электронной почты.
6. Сканер amap. Синтаксис, опции и режимы работы.
7. Утилита strobe. Синтаксис, опции и режимы работы.
8. Режимы идентификации версий прикладных служб сканера nmap.

РАЗДЕЛ V – ИДЕНТИФИКАЦИЯ ОПЕРАЦИОННЫХ СИСТЕМ (OS FINGERPRINTING)

Задача идентификации типа и версии ОС исследуемого узла сети (Remote OS Fingerprinting) является одной из ключевых при идентификации ресурсов КСПД. Версия ОС, как правило, определяет направление и ход дальнейшего анализа защищенности, поэтому от точности идентификации их версий зависит результат исследования всей КСПД.

Простейшие методы идентификации ОС уже упоминались в предыдущих разделах практикума: анализ стандартных приглашений сетевых служб, опрос служб прикладного уровня стандартными командами службы и т.д. В практических работах пятого раздела будут рассмотрены две группы наиболее эффективных методов идентификации ОС: методы активного исследования реализации стека протоколов TCP/IP (TCP/IP Stack Fingerprinting) и методы пассивного исследования реализации стека протоколов. Обе группы методов основаны на особенностях реализации стека TCP/IP в различных ОС.

Практическая работа №17
АКТИВНОЕ ИССЛЕДОВАНИЕ СТЕКА TCP/IP

1. Цель работы

Овладеть основами удаленной идентификации сетевых операционных систем. Изучить наиболее простые методы активного TCP/IP Stack Fingerprinting. Научиться применять на практике методы TCP|FIN сканирования, исследования флагом BOGUS и поля Window заголовка TCP, метод исследования изменения ISN.

2. Теоретические сведения. Методические рекомендации

Методы активного TCP/IP Stack Fingerprinting - это методы идентификации ОС, реализующие механизмы опроса стека TCP/IP удаленного узла сети. Как правило, ответной реакцией узла сети на любое информационное воздействие (IP-пакет данных, запрос на соединение и т.д.) является IP-пакет, адресованный источнику информационного воздействия. Реакция различных ОС на один и тот же запрос может быть различна. Совокупность ответов на некоторое множество специальным образом сформированных запросов ОС образует множество идентифицирующих признаков (сигнатур) данной ОС. В настоящее время автоматизированные средства анализа сети (сканеры сети) имеют достаточно большие базы сигнатур TCP/IP Stack Fingerprinting, что позволяет им с большой долей вероятности идентифицировать удаленную ОС.

В работе предлагается практически реализовать наиболее простые проверки, проводимые при TCP/IP Stack Fingerprinting (без использования специализированных программных средств анализа стека). Для генерации требуемых пакетов рекомендуется использовать утилиту mz (допускается использование альтернативных генераторов, например hping3).

Пример генерации единичного TCP|FIN пакета на 135 порт узла 192.168.0.11 средствами утилиты mz:

```
linux:~$ sudo mz eth0 –v –B 192.168.0.11 –p 135 –t tcp “flags=fin”
```

3. Практическое задание

Исследование 1 – метод TCP|FIN сканирования

В соответствии с RFC 793, узел сети, на пришедший (на открытый TCP-порт) TCP|FIN пакет (или иной пакет без флагов SYN и ACK), должен ответить TCP|RST пакетом. Однако многие ОС такой пакет игнорируют.

Исследуйте ответы различных ОС на входящие TCP|FIN пакеты:

1. Идентифицируйте открытые TCP порты узла сети с установленной ОС семейства Windows (адрес уточните у преподавателя).
2. Запустите одну из программ прослушивания сети (Wireshark, Tcpdump, Snort). Настройте фильтр программы на отображение TCP сессий с исследуемым узлом сети.
3. Отправьте TCP|FIN пакеты на открытые TCP порты исследуемого узла сети.
4. Пронаблюдайте ответную реакцию исследуемого узла сети.
5. Повторите эксперимент для ОС Linux, FreeBSD, встраиваемых ОС специализированных устройств (проектор, сотовый телефон и т.д.).
6. Сделайте выводы.

Исследование 2 – метод исследования флагом BOGUS

При получении SYN-пакета с установленным битом в поле BOGUS, ОС могут ответить ACK-пакетом с сохраненным изменением поля BOGUS или оборвут соединение (Термин BOGUS подразумевает установку значения поля Reserved заголовка TCP-пакета в 1000000, согласно RFC 793 значение зарезервированного поля Reserved установлено в 0000000).

Исследуйте значение поля Reserved заголовка ACK-пакетов (ответы различных ОС), полученных в ответ на входящие SYN-пакеты с установленным битом в поле BOGUS:
1. Отправьте SYN-пакеты с установленным битом в поле BOGUS на открытые TCP порты исследуемого узла сети.
2. Получите значения полей Reserved заголовка ACK-пакетов.
3. Повторите эксперимент для ОС Linux, FreeBSD, встраиваемых ОС специализированных устройств (проектор, сотовый телефон и т.д.).
4. Сделайте выводы.

Исследование 3 – метод исследования поля Window заголовка TCP

Идентификационным признаком может служить поле Window TCP заголовка принятого пакета. Значение данного поля определяет в байтах размер данных, которые получатель готов принять.

1. Исследуйте значения полей Window TCP различных ОС. Ход экспериментов аналогичен предыдущим. Сделайте вывод.

Исследование 4 – метод исследования изменения ISN

В RFC 793 определен порядок изменения поля ISN (Initial Sequence Number) ACK-пакета при установлении соединения, передаче данных и закрытии соединения. Если в полученном пакете установлен флаг SYN, то значение поля ISN принимается как начальное значение номера последовательности, и первый байт передаваемых в следующем пакете данных будет иметь номер последовательности равный ISN + 1. Если флаг SYN не установлен, то первый байт данных должен иметь тот же номер последовательности.

1. Отправьте на закрытый TCP-порт исследуемого узла TCP-пакет с установленными флагами FIN|PSH|URG. Значение ISN в поле ISS должно быть известно.
2. Получите значение ISN ответного ACK-пакета (полученного от исследуемого узла сети).
3. Повторите эксперимент, отправив TCP-пакет с установленными флагами SYN|FIN|PSH|URG.
4. Повторите эксперимент для ОС Linux, FreeBSD, встраиваемых ОС специализированных устройств (проектор, сотовый телефон и т.д.).
5. Сделайте выводы.

Практическая работа №18
СПЕЦИАЛИЗИРОВАННЫЕ ПРОГРАММНЫЕ СРЕДСТВА АКТИВНОГО ИССЛЕДОВАНИЯ СТЕКА TCP/IP

1. Цель работы

Продолжить изучение методов удаленной идентификации сетевых операционных систем. Научиться применять на практике специализированные программные средства активного исследования стека TCP/IP: xprobe2, nmap.

2. Теоретические сведения. Методические рекомендации

Утилита Xprobe2

Xprobe2 (http://xprobe.sourceforge.net/) – специализированное программное средство для активной идентификации версии операционной системы удаленного узла сети, разрабатываемое с 2001 года. Утилита использует "нечеткие" (fuzzy matching algorythm) методы сигнатурного анализа, в последних версиях присутствует модули для обнаружения honeypot и систем с намеренно модифицированными параметрами стека TCP/IP, определения версий различных устройств, реализована поддержка IPv6. Как правило, Xprobe2 подвергает исследуемый узел сети серии тестов, отправляя такие пакеты как:

-ICMP Echo Request;

-ICMP Timestamp Request;

-ICMP Address Mask Request;

-ICMP Information Request;

-Пакет UDP на закрытый порт UDP;

-Пакет TCP с флагом SYN на открытый порт TCP и т.д.

Работа Xprobe2 основана на Raw sockets (сырых сокетах), поэтому для ее использования необходимы права пользователя root.

Синтаксис утилиты xprobe2:

```
xprobe2 [ -v ] [ -r ] [ -p proto:portnum:state ] [ -c configfile ] [
-o logfile ] [ -p port ] [ -t receive_timeout ] [ -m numberofmatches
] [ -D modnum ] [ -F ] [ -X ] [ -B ] [ -A ] [ -T port spec ] [ -U
port spec ] host
```

Наиболее используемые опции xprobe2:

-*r:* отображение пути до целевого узла сети (traceroute);

-*p:* указывает номера портов и протоколов для исследования удаленного узла сети. Воможные значения для протокола *tcp* или *udp,* номера портов могут быть заданы в интервале от 1 до 65535;

-*o:* записывает результаты сканирования в файл;

-*X:* использует формат XML, используется совместно с опцией -o;

-*T:* задействует модуль для сканирования TCP-портов, интересуемый диапазон портов может задаваться следующим образом: -T20-80,110,3306;

-*U:* задействует модуль для сканирования UDP-портов;

-*v:* вывод подробной информации о ходе сканирования и загружаемых модулях;

-*M/-D:* активирует/дезактивируем указанные модули.

Пример применения xprobe2:

```
linux:~$ xprobe2 -v -p tcp:80:open www.vlsu.ru
linux:~$ xprobe2 -T 1-1024 127.0.0.1
```

В отчет сканирования xprobe2 включает список наиболее вероятных ОС (с указанием вероятности достоверности результата).

Режим идентификации ОС сканера nmap

Сканер nmap располагает обширной базой идентификационных признаков множества ОС, которые содержатся в файле nmap-os-fingerprints в каталоге Data установки Nmap.

Для активизации алгоритмов OS fingerprinting сканера nmap необходимо указать опцию -O (--osscan-):

```
nmap -O < IP или подсеть >
```

Пример применения опции OS fingerprinting nmap:

```
linux:~$ nmap -O www.vlsu.ru
Starting Nmap 5.21 ( http://nmap.org ) at 2011-10-02 16:59 MSK
Nmap scan report for www.vlsu.ru (85.142.154.99)
Host is up (0.068s latency).
```

```
rDNS record for 85.142.154.99: ip4host-154-99.vlsu.ru
Not shown: 993 closed ports
PORT    STATE    SERVICE
22/tcp  open     ssh
53/tcp  open     domain
80/tcp  open     http
111/tcp open     rpcbind
135/tcp filtered msrpc
139/tcp filtered netbios-ssn
445/tcp filtered microsoft-ds
Device type: broadband router|general purpose|WAP|storage-
misc|remote management|switch|specialized
Running (JUST GUESSING) : Linksys embedded (94%), Linux 2.6.X|2.4.X
(90%), AVM embedded (87%), D-Link embedded (87%), Dell embedded
(87%), HP embedded (87%), Linksys Linux 2.4.X (87%)
Aggressive OS guesses: Linksys WRV200 wireless broadband router
(94%), Linux 2.6.15 (Ubuntu) (90%), Linux 2.6.15 - 2.6.26 (90%),
Linux 2.6.25 (openSUSE 11.0) (90%), Linux 2.6.24 (Ubuntu 8.04)
(89%), Linux 2.6.22 (Kubuntu, x86) (88%), Linux 2.6.18 - 2.6.28
(87%), AVM FRITZ!Box FON WLAN 7170 WAP (87%), D-Link DNS-323 NAS
device or Linksys WRT300N wireless broadband router (87%), Dell
Remote Access Controller 5/I (DRAC 5/I) (87%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 11 hops
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.00 seconds
```

Для идентификации ОС эффективен следующий режим сканирования:

```
#nmap –-fuzzy –sV –F –v < IP или подсеть >
```

Для комплексного исследования удаленного узла сети рекомендуется применять опцию агрессивного сканирования -A. Данная опция активизирует режимы определения ОС (-O), определения версии (-sV), сканирование с использованием скриптов (-sC) и трассировку (--traceroute).

Пример применения опции агрессивного сканирования nmap:

```
linux:~$ nmap -A -v -T4 www.vlsu.ru
```

3. Практическое задание

1. Проверьте наличие утилиты xprobe2 в системе:

```
linux:~$ whereis xprobe2
xprobe2: /usr/bin/xprobe2 /etc/xprobe2
/usr/share/man/man1/xprobe2.1.gz
```

2. В случае необходимости установите xprobe2 на свою рабочую станцию:

```
apt-get update && apt-get install xprobe
```

3. Определите средствами xprobe2 версии операционных систем тестовых узлов сети (IP адреса тестовых узлов уточните у преподавателя).

4. Повторите исследования с различными комбинациями модулей сканирования xprobe2.

5. Проверьте наличие утилиты nmap в системе:

```
linux:~$ whereis nmap
nmap: /usr/bin/nmap /usr/lib/nmap /usr/share/nmap
/usr/share/man/man1/nmap.1.gz
```

6. Средствами nmap определите версии операционных систем тестовых узлов сети.

7. Сравните результаты работы xprobe2 и nmap.

8. Проведите исследование одного из тестовых узлов сети с применением опции агрессивного сканирования nmap.

Практическая работа №19
ПАССИВНОЕ ИССЛЕДОВАНИЕ СТЕКА В ЗАДАЧЕ ИДЕНТИФИКАЦИИ ОС

1. Цель работы

Продолжить изучение методов удаленной идентификации сетевых операционных систем. Овладеть основными методами пассивного OS Fingerprinting. Научиться применять на практике специализированные программные средства идентификации ОС удаленного узла сети: p0f.

2. Теоретические сведения. Методические рекомендации

Методы пассивной идентификации ОС (passive OS fingerprinting) базируются на анализе пакетов (как правило SYN), полученных от удаленного узла сети. В отличие от методов активного исследования, при пассивном OS fingerprinting исследующий узел сети не инициирует соединений с исследуемым. Пассивные методы широко применяются для анализа корректности TCP-сессий, удаленной идентификации открытых портов и прикладных служб, определения версий ОС.

Пассивные методы имеют определенные достоинства по сравнению активными методами OS fingerprinting:

- Отсутствие необходимости генерации дополнительного трафика, что снижает нагрузку на сеть;
- Пассивные методы троднообнаружимы для сетевых СОВ;
- В некоторых случаях пассивные методы могут выявить наличие МЭ, маршрутизаторов и NAT.

К недостаткам методов passive OS fingerprinting относят:

- Необходимость наличия программного или аппаратного сенсора в исследуемой сети;
- Сложность эффективного размещения сенсоров в топологии исследуемой сети (коммутируемые сети);

- Меньшее развитие инструментария анализа полученных данных.

p0f

p0f (http://lcamtuf.coredump.cx/p0f.shtml) – популярный сканер для идентификации типа ОС удаленного узла сети, путем прослушивания трафика. Алгоритмы p0f основаны на анализе служебных полей IP-пакетов, полученных от удаленного узла.

p0f способен определять тип ОС следующих узлов сети:

- узлы, инициирующие TCP-сессию с узлом исследователя (SYN режим);
- узлы, TCP-сессию с которыми устанавливает узел исследователя (SYN|ACK режим);
- узлы, соединения с которыми невозможны ввиду фильтрации трафика средствами МЭ (RST режим);
- узлы, трафик которых прослушивается сетевым интерфейсом узла исследователя.

Синтаксис утилиты p0f:

```
p0f p0f [ -f file ] [ -i device ] [ -s file ] [ -o file ] [ -Q
socket [ -0 ] ] [ -w file ] [ -u user ] [ -c size ] [ -T nn ] [ -e
nn ] [ -FNODVUKAXMqxtpdlRL ] [ ’filter rule’ ]
```

Наиболее используемые опции p0f:

-*i device:* прослушиваемое устройство (можно указать несколько интерфейсов);

-*s file:* анализ дампа трафика программы tcpdump (сеть не прослушивается);

-*w file:* записывает весь трафик в файл (аналог дампа tcpdump);

-*F*: задействует fuzzy matching алгоритмы;

-*U*: указывает не отображать неизвестные сигнатуры;

-*p:* переводит сетевой интерфейс в неразборчивый режим (promiscuous mode);

-*d:* переводит p0f в режим демона (фоновый режим);

-*r:* включает разрешение имен узлов сети (снижает производительность);

-*X:* указывает отображать полезную нагрузку пакета (packet payload).

Опции фильтрации p0f позволяют включать/исключать из рассмотрения отдельные узлы или диапазоны адресов, диапазоны

портов и т.д. Формат правил фильтрации аналогичен формату правил популярного сетевого анализатора tcpdump (Berkley Packet Filters):

```
'src port ftp-data'
'not dst net 10.0.0.0 mask 255.0.0.0'
'dst port 80 and (src host 195.117.3.59 or src host 217.8.32.51)'
```

Пример режимов запуска p0f:

```
linux:~$ p0f -i eth0 -F -r
linux:~$ p0f -i eth0 -U -F -p
linux:~$ p0f -i eth0 -r -X 'dst host 85.142.154.58'
```

3. Практическое задание

1. Проверьте наличие утилиты p0f в системе:

```
linux:~$ whereis p0f
p0f: /usr/sbin/p0f /etc/p0f /usr/share/man/man1/p0f.1.gz
```

2. В случае необходимости установите p0f на свою рабочую станцию:

```
apt-get update && apt-get install p0f
```

3. Практически ознакомтесь с режимами работы сканера p0f (*-F,-U,-p,-d, -r*);

4. Практически ознакомтесь с опциями фильтрации сканера p0f;

5. Определите средствами p0f версии операционных систем тестовых узлов сети (IP адреса тестовых узлов уточните у преподавателя).

6. Сравните результаты работы p0f с результатами сканеров xprobe2 и nmap.

Контрольные вопросы к разделу V

1. Задача идентификации типа и версии ОС исследуемого узла КСПД.
2. Особенности методов активного исследования реализации стека протоколов TCP/IP. Достоинства и недостатки.
3. Суть метода TCP|FIN сканирования.
4. Суть метода исследования флагом BOGUS.
5. Суть метода исследования поля Window TCP заголовка принятого пакета.
6. Суть метода исследования изменения ISN ACK-пакета.
7. Утилита Xpobe2. Синтаксис, опции.
8. Режимы OS fingerprinting сканера nmap.
9. Особенности методов пассивного исследования реализации стека протоколов TCP/IP. Достоинства и недостатки.
10. Утилита *p0f*. Синтаксис, опции.

ИСПОЛЬЗУЕМАЯ И РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Илларионов Ю.А. Введение в теорию информационной безопасности : учеб. пособие / Ю.А. Илларионов; Владим. гос. Ун-т. - Владимир : Изд-во ВлГУ, 2005. - 88 с. - (Комплексная защита объектов информатизации. Кн. 8 / под ред. М.Ю. Монахова). - ISBN 5-89368-557-1
2. RFC 768 - User Datagram Protocol, August 1980
3. RFC 791 - Internet protocol / September 1981
4. RFC 792 - Internet Control Message Protocol, September 1981
5. RFC 793 - Transmission Control Protocol, September 1981
6. RFC 821 – Simple Mail Transfer Protocol, August 1982
7. RFC 826 - An Ethernet Address Resolution Protocol -- or -- Converting Network Protocol Addresses, November 1982
8. RFC 950 - Internet Standard Subnetting, August 1985
9. RFC 1256 - ICMP Router Discovery, September 1991
10. RFC 2822 – Internet Message Format, April 2001
11. RFC 3232 - Assigned Numbers, October 1994
12. Таненбаум Э. Компьютерные сети. — Четвёртое издание. — Питер: Питер, 2007. -С. 992.
13. Network Security Assessment By Chris McNab Publisher: O'Reilly, March 2004, - 396 p. - ISBN 0-596-00611-X
14. Stuart McClure, Joel Scambray, George Kurtz : Hacking exposed 6: network security secrets&solutions, 2009,- ISBN 978-0-07-161375-0

Интернет источники

15. Remote OS Detection [Электронный ресурс] http://www.insecure.org/nmap/nmap-fingerprinting-article.html
16. Обнаружение служб и их версий [Электронный ресурс] http:// www.nmap.org/man/ru/man-version-detection.html
17. Определение ОС [Электронный ресурс] http:// www.nmap.org/man/ru/man-os-detection.html
18. Основы сканирования портов [Электронный ресурс] http:// www.nmap.org/man/ru/man-port-scanning-basics.html

ОГЛАВЛЕНИЕ

Учебное издание

Комплексная защита объектов информатизации
Книга 24

МИШИН Денис Вячеславович
МОНАХОВ Юрий Михайлович

АНАЛИЗ ЗАЩИЩЕННОСТИ РАСПРЕДЕЛЕННЫХ ИНФОРМАЦИОННЫХ СИСТЕМ

Идентификация ресурсов корпоративной сети передачи данных

Практикум