Лабораторная работа 4

Цель: *на примере протоколов SNMP-DPI и TELNET ознакомиться с уровнем приложений стека протоколов TCP/IP.*

Теоретические сведения

1. УРОВЕНЬ ПРИЛОЖЕНИЙ: ПРОТОКОЛЫ TELNET И SNMP

1.1. Уровень приложений стека протоколов TCP/IP

Уровень приложений модели стека протоколов TCP/IP выполняет следующие функции.

- Обеспечивает управление диалогом между устройствами: фиксирует какая из сторон является активной в настоящий момент, предоставляет средства синхронизации и занимается отделением данных одного приложения от данных другого приложения.

- Имеет дело с формой представления передаваемой по сети информацией, не меняя при этом ее содержания. С помощью средств данного уровня протоколы уровня приложений могут преодолеть синтаксические различия в представлении данных.

- Может выполнять шифрование и дешифрование данных.

- Предоставляет набор разнообразных протоколов, с помощью которых пользователи сети получают доступ к разделяемым ресурсам, таким как файлы, принтеры или гипертекстовые Web-страницы, а также организуют свою совместную работу, например, с помощью протокола электронной почты.

Прикладной уровень реализуется программными системами, построенными в архитектуре клиентсервер, базирующимися на протоколах нижних уровней. Протоколы прикладного уровня описывают работу конкретного приложения и не способны к передаче данных по сети. Этот уровень постоянно расширяется т.к. создаются сравнительно новые такие, например, как протокол передачи гипертекстовой информации HTTP. Уровень приложений модели стека TCP/IP соответствует совокупности трех уровней модели OSI: сеансового уровня, уровня представлений и прикладного уровня. Рассмотрены два прикладных протокола: SNMP – простой протокол управления сетью и TELNET – протокол для создания

незащищенного соединения с серверным программным обеспечением.

1.1.1. Протокол SNMP

С любой сети функционирует большое количество узлов, маршрутизаторов и имеется широкий набор программных средств. Сеть сохраняет работоспособность благодаря жесткой протокольной регламентации, требующей разработки средств контроля и управления. Функции диагностики сети возложены на ICMP, а функции управления на SNMP (Simple Network Management Protocol – RFC1157). Чаще всего управляющая прикладная программа воздействует на сеть по цепочке: SNMP, UDP, IP, физическая сеть. *Управление сетью* – это процесс управления отказами, контроля конфигураций, мониторинга производительности, обеспечения защиты и учета деятельности в сети передачи данных. Наиболее важным объектом управления обычно является маршрутизатор. Каждому управляемому объекту присваивается уникальный идентификатор.

Протокол SNMP использует UDP в качестве транспортного протокола и предназначен для использования сетевыми управляющими станциями. Он позволяет управляющим станциям собирать информацию о положении в сети. Протокол определяет формат данных, а их обработка и интерпретация остаются на усмотрение управляющих станций или менеджера сети.

Приложения управления сетью называемые *менеджерами*, общаются с программным обеспечением сетевых устройств, называемым *агентами*. SNMP – это протокол типа "запрос-отклик", то есть на каждый запрос, поступивший от менеджера, агент должен передать отклик. Под *запросом* будем понимать передачу информации от менеджера к агенту с целью получения параметров объекта управления. Под *откликом* будем понимать ответ агента, на запрос менеджера, содержащий требуемые параметры. Обмен данными между менеджером и агентом дает возможность менеджеру собирать стандартный набор информации, который определен в базе данных информации для управления сетью – MIB. Порция информации, существующая в базе данных, называется *объектом*.

Алгоритмы управления в сети обычно описывают в нотации ASN.1 (Abstract Syntax Notation). Все объекты в сети разделены на 10 групп и описаны в MIB: система, интерфейсы, обмены, трансляция адресов, IP, ICMP, TCP, UDP, EGP, SNMP. В группу "система" входит название и версия оборудования, операционной системы, сетевого программного обеспечения и пр. В группу "интерфейсы" входит число поддерживаемых интерфейсов, тип интерфейса, работающего под управлением IP (Eth-

ernet, LAPB и т.д.), размер дейтаграмм, скорость обмена, адрес интерфейса. IP-группа включает время жизни дейтаграмм, информацию о фрагментации, маски подсетей и т.д. В TCP-группу входит алгоритм повторной пересылки, максимальное число повторных пересылок и пр. [информация с сайта http://book.itep.ru/4/44/snm_4413.htm]

Протокол SNMP имеет достаточно простую структуру и включает в себя следующие команды:

- *Get-request* используется менеджером для получения от агента значения какого-либо параметра по его имени;
- *GetNext-request* используется для извлечения значения следующего объекта (без указания его имени) при последовательном просмотре таблицы объектов;
- *Set* используется менеджером для изменения значения какоголибо объекта. С помощью команды Set происходит собственно управление устройством. Агент должен понимать смысл значения объекта, который используется для управления устройством, и на основании этих значений выполнять реальное управляющее воздействие – отключить порт, установить IP адрес и т.п. Команда Set пригодна также для установки условия, при выполнении которого агент SNMP должен послать менеджеру соответствующее сообщение. Может быть определена реакция на такие события, как инициализация агента, рестарт агента, обрыв связи, восстановление связи, неверная аутентификация, потеря ближайшего маршрутизатора и др. Если происходит любое из этих событий, то агент инициализирует прерывание (*trap*). Если запросом *Set* устанавливаются значения сразу нескольких объектов, то в случае ошибки все объекты останутся без изменений;
- *Get-response* обеспечивает передачу ответа на команды *Get-request*, *GetNext-request* или *Set* от агента SNMP менеджеру. *Get-response* возвращает значения запрошенных объектов, только в случае успешного выполнения команд *Get*, *GetNext* или *Set*;
- *Trap* используется агентом для сообщения менеджеру о возникновении особой ситуации.

Схема иллюстрирующая обмен данными между SNMP менеджером и SNMP агентом представлена на рисунке 1.1. Прямоугольниками с числами обозначены порты на которых менеджер и/или агент ожидает дейтаграммы пользователя. Обычно SNMP агент использует 161 порт для ожидания запросов get, get-next или set, а SNMP менеджер – 162 порт для ожидания прерываний (trap). *Объектом управления* является любое сетевое устройство поддерживающее протокол SNMP,

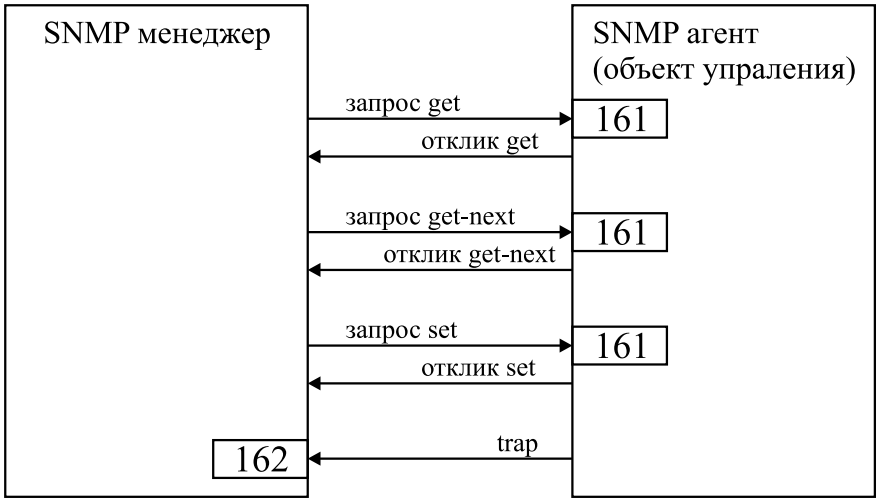


Рис. 1.1 Схема запросов/откликов SNMP.

на котором запущен SNMP агент.

В последнее время широкое распространение получила идеология распределенного протокольного интерфейса DPI (Distributed Protocol Interface). В этом случае для транспортировки SNMP-запросов используется не только UDP, но и TCP-протокол. Это дает возможность применять SNMP-протокол не только в локальных сетях. Форматы SNMP-DPI-запросов (версия 2.0) описаны в документе RFC1592. На рисунке 1.2 изображены форматы SNMP-DPI сообщений, вкладываемых в UDP-дейтаграммы для запросов *Get*, *GetNext* или *Set* На рисунке 4.2 прописными буквами латинского алфавита обозначены поля, которые содержат следующие данные:

A – длина сообщения, без первых двух байтов;

B – текущая версия протокола (для SNMPv2 это значение равно 2);

C – минимальная версия протокола совместимая используемой версией (для SNMPv2 это значение равно 2);

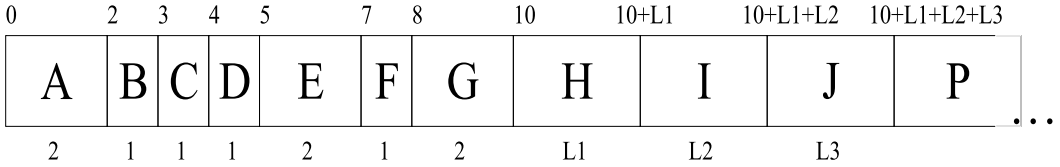
D – версия модификации основного протокола (в первой редакции протокола SNMPv2 это значение равно 0);

E – идентификатор сообщения, т.е. уникальное число, характеризующее отдельное сообщение посланный агенту и позволяющее связывать пары запрос-отклик (это необходимо при использовании UDP, т.к. возможна потеря пакетов);

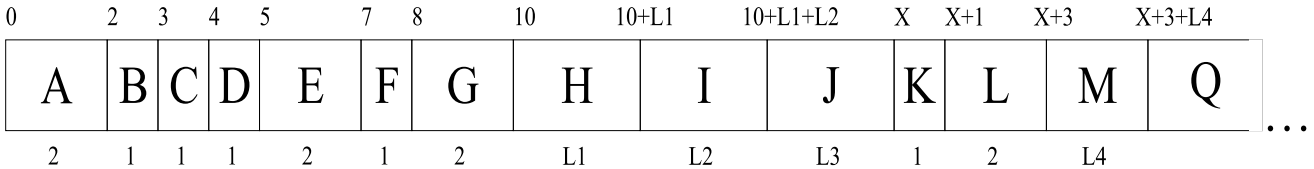
F – тип сообщения может принимать значения: SNMP_DPI_GET для get-запроса, SNMP_DPI_GETNEXT для getnext-запроса, SNMP_DPI_SET для set-запроса;

G – длина поля "имя группы доступа";

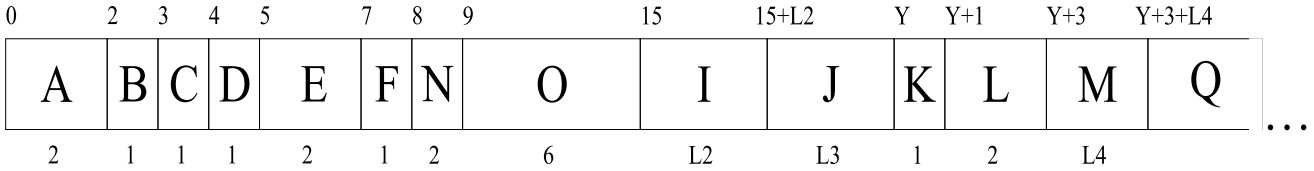
H – имя группы доступа – содержит последовательность символов, которая является пропуском при взаимодействии менеджера и объекта



(а) - формат пакета Get-Request и GetNext-Request



(б) - формат пакета Set-Request



(в) - формат пакета Get-Response

Рис. 1.2 Форматы сообщений протокола SNMP

управления (обычно это поле содержит 6-байтовую строку public);
I – идентификатор группы объекта управления – содержит последовательность чисел, определяющую путь по дереву MIB до объекта управления (последовательность должна заканчиваться значением *null*);
J – идентификатор объекта управления – лист MIB дерева, путь до которого определяет идентификатор группы объекта управления (идентификатор должен заканчиваться значением *null*);
K – тип SNMP переменной, например: SNMP_TYPE_Integer32, SNMP_TYPE_IpAddress, SNMP_TYPE_OCTET_STRING;
L – длина поля "значение SNMP переменной";
M – значение SNMP переменной соответствующего типа;
N – статус ошибки – определяет причину ошибки и может принимать следующие значения:

- noError(0) – нет ошибок;
- tooBig(1) – объект не может уложить отклик в одно сообщение;
- noSuchName(2) – в операции указана неизвестная переменная;
- badValue(3) – в команде set использована недопустимая величина или неправильный синтаксис;
- readOnly(4) – менеджер попытался изменить константу;

- genErr(5) – прочие ошибки;
- noAccess(6) – имя группы доступа содержащееся в сообщении и установленное на агенте не совпадают;
- wrongType(7) – использован неправильный тип SNMP переменной;
- wrongLength(8) – одно из полей длины не соответствует действительности;

O – индекс ошибки – порядковый номер SNMP переменной, которая привела к ошибке;
P – в случае необходимости возможна запись дополнительных SNMP переменных, т.е. пар: идентификатор группы(I), идентификатор объекта(J);
Q – в случае необходимости возможна запись дополнительных SNMP переменных со значениями, т.е. последовательностей состоящих из пяти полей: идентификатор группы(I), идентификатор объекта(J), тип SNMP переменной(K), длина поля "значение SNMP переменной"(L), значение SNMP переменной(M).
На рисунке числа и буквы над полями пакета обозначают смещение соответствующего поля от начала пакета в байтах. А числа и буквы под полями пакета обозначают длину соответствующего поля. Для краткости записи были введены следующие обозначения:
L1 – длина поля "имя группы доступа";
L2 – длина поля "идентификатор группы объекта управления";
L3 – длина поля "идентификатор объекта управления";
L4 – длина поля "значение SNMP переменной";
X = 10+L1+L2+L3;
Y = 15+L2+L3.
Видно, что сообщения SNMP имеют достаточно простую структуру. Это упрощает реализацию протокола, но ведет к тому, что имя группы доступа, предназначенное для ограничения доступа к SNMP агенту, ни как не шифруется и передается по сети в открытой форме.

1.1.2. Управляющая база данных MIB

Вся управляющая информация для контроля сетевых устройств (маршрутизаторы, коммутаторы и т.п.) концентрируется в базе данных MIB (Management Information Base, RFC1212, RFC1213). Каждая порция информации, существующая в базе данных, называется *объектом*. База данных информации для управления сетью содержит объекты, которые нужны менеджеру для управления сетью. MIB выглядит как дерево с раздельными пунктами данных в качестве выходов. Идентификатор объекта однозначно идентифицирует MIB-объект в дереве. Объект обозначается, как последовательность

чисел, разделенных точкой. Объекты организуются иерархически и их части могут принадлежать различным организациям. Верхний уровень идентификаторов MIB объектов установлен ISO/IEC. Объекты более низкого уровня выделяются специальными организациями. MIB дерево постоянно расширяется, как результат экспериментов частных разработок. Производители, например, могут определить свои личные ветви для включения образов своих продуктов. Такие деревья MIB не стандартизируются, а носят характер экспериментальных деревьев. Пример части такого дерева приведен на рисунке 1.3. Из этого рисунка

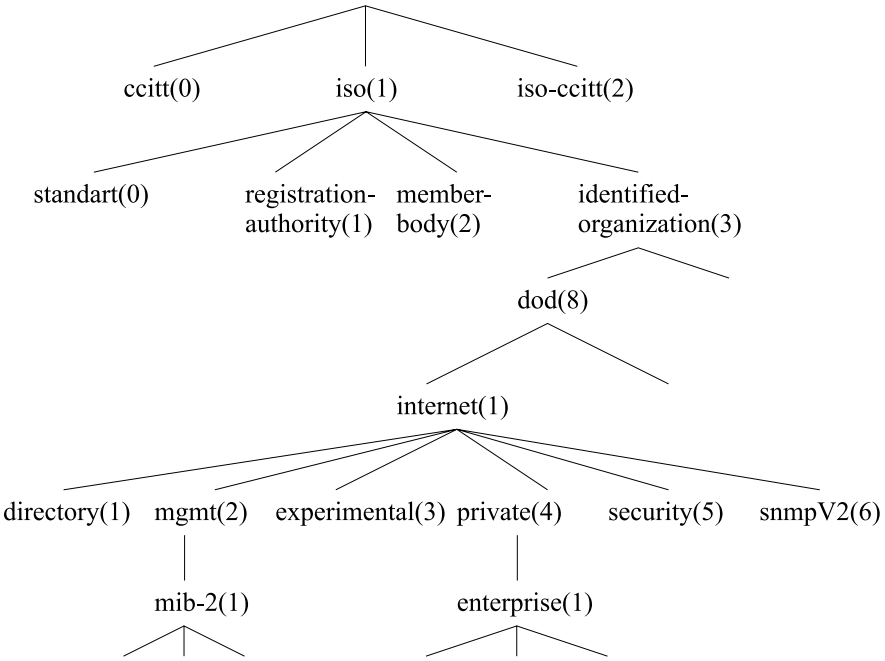


Рис. 4.3 Дерево MIB

видно, что например для узла snmpV2 идентификатор объекта будет: 1.3.8.1.6

1.1.3. Протокол TELNET

Для обеспечения удаленного доступа к сетевому устройству с помощью командного интерпретатора используется протокол TELNET (RFC854). Протокол TELNET – это сетевой протокол типа "клиент-сервер". TELNET обеспечивает незащищенное соединение, т.е. все данные передаются в открытой форме в том числе и пароли. TELNET использует TCP в качестве транспортного протокола. Общепринято, что TELNET-сервер ожидает соединения на 23 порту. TELNET позволяет пользователю установить TCP-соединение с сервером и затем передавать коды нажатия клавиш так, как если бы работа проводилась на консоли сервера. Для входа в командный режим обычно

нужна аутентификация – ввод имени пользователя и его пароля).
TELNET предлагает три услуги:

- определяет сетевой виртуальный терминал (NVT – network virtual terminal), который обеспечивает стандартный интерфейс доступа к удаленной системе;
- включает механизм, который позволяет клиенту и серверу согласовать опции обмена;
- обеспечивает соединение, при котором любая программа (например FTP) может выступать в качестве клиента.

Протокол TELNET позволяет обслуживающей машине рассматривать все удаленные терминалы как стандартные "сетевые виртуальные терминалы" строчного типа, работающие в кодах ASCII, а также обеспечивает возможность согласования более сложных функций (например, локальный или удаленный эхо-контроль, страничный режим, высота и ширина экрана и т. д.). На прикладном уровне над протоколом TELNET находится либо программа поддержки реального терминала, либо прикладной процесс в обслуживающей машине, к которому осуществляется доступ с терминала. Формат NTV достаточно прост. Для данных используются 7-битовые ASCII коды. А октеты из восьми бит зарезервированы для командных последовательностей [информация с сайта http://book.itep.ru/4/45/tlnt_453.htm].
В упрощенном варианте протокол TELNET работает следующим образом: Между клиентом и сервером устанавливается TCP соединение. Клиент посылает серверу символ перевода строки для того, чтобы сервер знал что это клиент хочет соединится по TELNET. В ответ сервер посылает приглашение ввода имени (например: login) и ждет ввода имени пользователя. После ввода сервер посылает приглашение ввода пароля (например: password) и ждет ввода пароля. Если введенные имя и пароль корректны то TELNET-сервер переходит в режим ввода. В этом режиме любой введенный текст пересылается удаленному сетевому устройству. Ввод может производиться посимвольно или построчно. При посимвольном режиме каждый введенный символ пересылается немедленно, при построчном режиме отклик на каждое нажатие клавиши производится локально, а пересылка выполняется лишь при нажатии клавиши <Enter>. В режиме ввода TELNET-сервер выдает какое-либо приглашение (например: telnet>), и ожидает ввода команд пользователя. При вводе команды quit сервер разрывает соединение.

Порядок выполнения работы

1. На компьютере К1 запустить SNMP агента. Порт и имя группы доступа выбираются студентом;
2. С компьютера К2 отправить запрос(ы) *get*, и получить переменные П1, П2, П3. Сравнить полученные значения с реальными;
3. С компьютера К2 отправить запрос(ы) *getnext* для переменных П1, П2, П3. Объяснить полученные результаты;
4. На компьютере К1 с помощью диалога "Set TCP/IP Properties" изменить IP адрес, маску подсети и шлюз по умолчанию. С компьютера К2 с помощью запросов *set* вернуть К1 в исходное состояние. Проверить результаты посредством SNMP;
5. На компьютере К2 запустить TELNET сервер. Порт и пароль выбрать самостоятельно;
6. С компьютера К3 по протоколу TELNET подключиться к компьютеру К2. Удалить все значения из таблицы маршрутизации и ARP таблицы. Добавить в таблицу маршрутизации и ARP таблицу записи необходимые для корректной работы компьютера К2;
7. С помощью команды TELNET-сервера *snmp* запустить SNMP агента на К3. Проверить работоспособность snmp-сервера: с компьютера К2 попытаться получить значение SNMP переменной П2;

В отчет необходимо включить схему сети, все вводимые параметры (порт, имя группы доступа и др.), отправляемые запросы и получаемые ответы. Для протокола TELNET необходимо привести сообщения выводимые в TELNET-консоль.

Варианты заданий

Вариант 1. Файл со схемой сети lab1_var1.jfst.
Обозначения в задании: Компьютеры К1 – OFFICE2 pc1; К2 – Boss; К3 – Hacker. SNMP переменные П1 – Counter.InputIP; П2 – IP.AllInterfaces; П3 – IP.Address_Eth0.
Вариант 2. Файл со схемой сети lab1_var2.jfst.
Обозначения в задании: Компьютеры К1 – OFFICE1 pc4; К2 – BIG BOSS; К3 – M_CH_S. SNMP переменные П1 – Counter.OutputIP; П2 – IP.ARPTable; П3 – IP.SubnetMask_Eth0.
Вариант 3. Файл со схемой сети lab1_var3.jfst.
Обозначения в задании: Компьютеры К1 – OFFICE2 pc2; К2 – Hacker; К3 – Boss. SNMP переменные П1 – Counter.ARP; П2 – IP.DefaultGateway;

П3 – SNMP.CommunityName.

Вариант 4. Файл со схемой сети lab1_var4.jfst.

Обозначения в задании: Компьютеры К1 – BIG BOSS; К2 – OFFICE1 pc1; К3 – OFFICE1 pc3. SNMP переменные П1 – Counter.InputTCP; П2 – IP.Address_Eth0; П3 – SNMP.revision.

Вариант 5. Файл со схемой сети lab1_var5.jfst.

Обозначения в задании: Компьютеры К1 – FileServer; К2 – Manager1; К3 – MegaBoss. SNMP переменные П1 – Counter.OutputTCP; П2 – IP.SubnetMask_Eth0; П3 – IP.DefaultGateway.

Вариант 6. Файл со схемой сети lab1_var6.jfst.

Обозначения в задании: Компьютеры К1 – PrintServer; К2 – Manager3; К3 – MicroBoss. SNMP переменные П1 – Counter.ReceiveDuplicatedTCP; П2 – SNMP.CommunityName; П3 – IP.ARPTable.

Вариант 7. Файл со схемой сети lab1_var7.jfst.

Обозначения в задании: Компьютеры К1 – Station1; К2 – Station4; К3 – Remote1. SNMP переменные П1 – Counter.SendDuplicatedTCP; П2 – SNMP.revision; П3 – IP.AllInterfaces.

Вариант 8. Файл со схемой сети lab1_var8.jfst.

Обозначения в задании: Компьютеры К1 – Station3; К2 – Remote1; К3 – Station2. SNMP переменные П1 – Counter.SendAckTCP; П2 – Counter.InputIP; П3 – Device.MACaddress_Eth0.

Вариант 9. Файл со схемой сети lab1_var9.jfst.

Обозначения в задании: Компьютеры К1 – PC1; К2 – PC2; К3 – PC4. SNMP переменные П1 – Counter.InputUDP; П2 – Counter.OutputIP; П3 – Device.AvailableInterfaces.

Вариант 10. Файл со схемой сети lab1_var10.jfst.

Обозначения в задании: Компьютеры К1 – PC2; К2 – PC3; К3 – PC4. SNMP переменные П1 – Counter.OutputUDP; П2 – Counter.ARP; П3 – Device.AllInterfaces.

Вариант 11. Файл со схемой сети lab1_var11.jfst.

Обозначения в задании: Компьютеры К1 – Chief; К2 – Service; К3 – Manager1. SNMP переменные П1 – Device.AllInterfaces; П2 – Counter.InputTCP; П3 – Device.Hostname.

Вариант 12. Файл со схемой сети lab1_var12.jfst.

Обозначения в задании: Компьютеры К1 – Manager3; К2 – Service; К3 – Chief. SNMP переменные П1 – Device.AvailableInterfaces; П2 – Counter.OutputTCP; П3 – Counter.OutputUDP.

Вариант 13. Файл со схемой сети lab1_var13.jfst.

Обозначения в задании: Компьютеры К1 – Remote1; К2 – Remote2; К3 – Remote3. SNMP переменные П1 – Device.Hostname; П2 –

Counter.ReceiveDuplicatedTCP; П3 – Counter.InputUDP.
Вариант 14. Файл со схемой сети lab1_var14.jfst.
Обозначения в задании: Компьютеры К1 – Remote2; К2 – Remote3; К3 – Remote1. SNMP переменные П1 – Device.MACaddress_Eth0; П2 – Counter.SendDuplicatedTCP; П3 – Counter.SendAckTCP.

Пример выполнения лабораторной работы

Файл со схемой сети: javaNetSim/labs/lab1/lab1_sample.jfst.

Задание:

1. На компьютере PC1 запустить SNMP агента.
2. С компьютера PC2 отправить запрос(ы) get, и получить переменные *ip.address_eth0*, *device.hostname*.
3. На компьютере PC2 запустить TELNET-сервер.
4. С компьютера PC1 по протоколу TELNET подключиться к компьютеру PC2. Удалить все значения из кэша ARP. Добавить туда статическую запись для узла PC1.

Порядок выполнения работы будет следующим:

1. Запустим на PC1 SNMP агент с параметрами:
   - порт на котором SNMP агент будет ожидать пакеты: *161*;
   - имя группы доступа для SNMP агента: *defgroup*.
2. Выполним с PC2 запрос SNMP-агенту на PC1 со следующими параметрами:
   - IP адрес компьютера на котором установлен SNMP агент: *172.168.0.2*;
   - порт на котором SNMP агент ожидает пакеты: *161*;
   - SNMP запрос: *get*;
   - SNMP переменные: *ip.address_eth0;device.hostname*;
   - имя группы доступа: *defgroup*.

   Результаты запроса будут выведены в консоль:

   ```
   PC2ReceivedgetResponse:
       'IP.Address_Eth0=172.168.0.2','Device.Hostname=PC1'
   ```
3. Запустим TELNET-сервер со следующими параметрами:
   - порт, на котором TELNET-сервер будет ожидать пакеты: *23*;
   - пароль для доступа к TELNET: *234*.
4. Запустим TELNET-клиент с параметрами:
   - IP адрес TELNET сервера: *10.0.0.2*;
   - порт, на котором TELNET-сервер ожидает пакеты: *23*.

   В ответ на приглашение к авторизации в системе необходимо ввести имя пользователя *root* и пароль *234*. После входа в систему будет

выведено приглашение командной строки:

```
pc1#
```

Просмотрим записи в таблице маршрутизации:

```
pc1#arp-a
InternetAddressPhysicalAddressType
10.0.0.1A2:2A:55:20:75:42Dynamic
```

Как видно из вывода команды arp, в кэше находится лишь одна динамическая запись. Ее можно удалить следующим образом:

```
pc1#arp-d10.0.0.1
```

Для добавления статической записи в кэш ARP необходимо использовать ключ -s команды arp:

```
pc1#arp-s10.0.0.1A2:2A:55:20:75:42
```

Таким образом была добавлена статическая запись для компьютера PC1. После завершения работы закрываем сеанс TELNET.

Контрольные вопросы

1. Для чего предназначен протокол SNMP?
2. Если на SNMP запрос пришел отклик с установленным флагом ошибки, то какие переменные будут содержаться в этом отклике? Если в set запросе часть переменных имеет корректные значения, а часть некорректные то какие переменные объекта управления изменятся?
3. Как обеспечивается защита в протоколе SNMP? Как вы думаете насколько безопасно применения протокола SNMP для управления реальной сетью? Что надо сделать для увеличения безопасности?
4. Для чего предназначен протокол TELNET?
5. Как работает протокол TELNET? Как обеспечивается безопасность при вводе пароля?