**學 士 學 位 論 文**

**JPcap기반의 Packet Analyzer**

***A Packet Analyzer based on JPcap Library***

**김대용, 김진솔, 장용수**

**안 양 대 학 교**

**정 보 통 신 공 학 과**

**2 0 1 5 년**

**JPcap기반의 Packet Analyzer**

***A Packet Analyzer based on JPcap Library***

**김대용, 김진솔, 장용수**

위 논문을 안양대학교 정보통신공학과의

학사학위 논문으로 제출함

2015년 12월 일

**심사 위원장: (인)**

**심 사 위 원: (인)**

**심 사 위 원: (인)**

**JPcap기반의 Packet Analyzer**

***A Packet Analyzer based on JPcap Library***

**김대용, 김진솔, 장용수**

위 논문을 안양대학교 정보통신공학과의

학사학위 논문으로 제출함

2015년 12월 일

**지 도 교 수: (인)**

**국 문 요 약**

패킷 분석은 IT 보안 이슈와 네트워크 성능 이슈에 대한 원인을 빨리 찾아내기 위해 IT 전문가들이 갖춰야 하는 핵심 기술이다. 대부분 IT 전문가들은 트러블슈팅, 보안과 네트워크를 최적화하는데 패킷 분석기를 필수 도구로 사용하고 있다. JPcap은 네트워크 상의 패킷들을 전송하고 캡쳐하는데 사용되는 Java library이다. JPcap을 사용한다면 임의의 패킷만을 전송하는 애플리케이션 또는 네트워크 인터페이스로 들어오는 패킷을 캡쳐하는 애플리케이션을 개발하는데 유용하다.

본 논문에서는 JPcap JAVA API를 기반으로 한 VPS(visual packet sniffer)를 구현하였다. 사용자들이 이해하기 쉽고, 조작법이 간단한 메뉴들로 GUI(graphical user interface)를 구성하였다. Microsoft Windows(XP/7/8), Linux(Fedora/Debian/Ubuntu), Mac OS X, Solaris 등의 다양한 운영체제와 호환이 가능하기 때문에 대부분의 시스템에서 사용이 가능하다.

실질적인 구현을 테스트하기 위하여 VMware Player 기반의 Fedora Linux를 설치 후,

Mysql서버를 구축한다. 그리고, C언어 Mysql라이브러리 함수를 이용하여 만든 여행사 관리 프로그램(여행사DataBase연동)을 통하여 데이터 전송 시, 패킷의 흐름을 살펴보는

Packet Analysis Situation On Travel Agency Program(PATP)의 구현을 목표로 한다.

**목 차**

**제 1 장 서론** -------------------------------------------------------------------------------------------------- **1**

**제 2 장 관련 연구**

제 1 절 Packet Analyzer **-------------------------------------------------------------------------------- 2**

제 2 절 Travel Agency Program **---------------------------------------------------------------------- 5**

**제 3 장 모델**

제 1 절 VPS (Visual Packet Sniffer) ------------------------------------------------------------------ **7**

제 2 절 VPS 동작 환경 및 구성 --------------------------------------------------------------------- **8**

**제 4 장 설계**

제 1 절 VPS 설계 --------------------------------------------------------------------------------------- **9**

제 2 절 Infrastructure 설계 --------------------------------------------------------------------------- **12**

**제 5 장 구현**

제 1 절 구현 환경 **----------------------------------------------------------------**---------------------- **14**

제 2 절 구현 화면 **-------------------------------------------------------------------------------------- 15**

**제 6 장 결론 및 향후 계획 -------------------------------------------------------------------- 18**

**부록 Ⅰ**------------------------------------------------------------------------------------------------------------ **19**

**참고 문헌 ------------------------------------------------------------------------------------------------------ 31**

**그림 목차**

<그림 1 - 1> 공격에 이용된 포트 별 분포 **----------------------------------------------------------------- 1**

<그림 2 - 1> TCP Dump 화면 -------------------------------------------------------------------**---------------- 3**

<그림 2 - 2> Wireshark 화면 ----------------------------------------------------------**-------------------------- 3**

<그림 2 - 3> Smart Sniff 화면 ------------------------------------------------------**---------------------------- 4**

<그림 2 - 4> Travel Agency Program 메인 화면 ----------------------------------------------------------- **5**

<그림 2 - 5> Travel Agency Program 다이어그램 ---------------------------------------------------------- **6**

<그림 2 - 6> Travel Agency Program 개념적 모델링 ---------------------------------------------------**-- 6**

<그림 3 – 1> VPS Prototype Model ------------------------------------------------------**-------------------- 7**

<그림 3 – 2> VPS 동작 환경 ------------------------------------------------------**----------------------------- 8**

<그림 4 – 1> VPS 데이터 흐름도 ---------------------------------------------------**-------------------------- 9**

<그림 4 – 2> VPS 순서도 --------------------------------------------------**------------------------------------- 10**

<그림 4 – 3> Infrastructure 설계 ---------------------------------------------------**-------------------------- 12**

<그림 4 – 4> Client Local Infrastructure 설계 --------------------------------------------------**----------- 12**

<그림 4 – 5> Wide Area Infrastructure 설계 ---------------------------------------------------**------------ 13**

<그림 4 – 6> Service Local Infrastructure 설계 --------------------------------------------------**--------- 13**

<그림 5 – 1> VPS 실행 화면 ---------------------------------------------------**------------------------------- 15**

<그림 5 – 2> TCP Graph 구성도 ----------------------------------------------------**-------------------------- 15**

<그림 5 – 3> Select Dialog ------------------------------------------------------**------------------------------- 15**

<그림 5 – 4> 전체 패킷 저장 및 백업내용 --------------------------------------------------**-------------- 16**

<그림 5 – 5> 프로토콜 패킷 백업 내용 -----------------------------------------------------------**--------- 16**

<그림 5 – 6> TCP Bar Graph ---------------------------------------------------**-------------------------------- 17**

<그림 5 – 7> TCP Polygon Graph --------------------------------------------------**--------------------------- 17**

**표 목차**

<표 1 - 1> 국가별 공격 발생지 분포 **------------------------------------------------------------------------- 1**

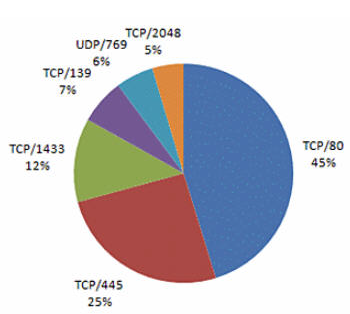
<표 2 - 1> VPS(Visual Packet Sniffer) 기능 분석 **--------------------------------------------------------------- 2**

**제 1 장 서론**

WIFI 및 LTE 기술의 등장과 다양한 단말 Device의 보급으로 매일 인터넷을 이용해 Web Surfing, Mail, Messenger 등을 하는 것은 일상이 되었다. 이러한 정보기술은 생활의 편리성과 업무의 효율성을 증대 시키지만, 반면에 온라인상에서 다양한 보안 이슈들의 발생을 초래한다. <표 1 – 1>은 네트워크 공격이 비교적 많이 발생하는 상위 10개 국가의 순위 및 공격 발생 비율을 나타낸다. 대한민국에서 가장 많은 공격이 발생하고 있음을 보여준다.

<표 1 – 2>는 공격에 이용된 포트의 상대적 비율을 나타낸다.

<표 1 - 1> 국가별 공격 발생지 분포 [1]

<그림 1 - 1> 공격에 이용된 포트 별 분포 [2]

이러한 보안 이슈의 원인을 파악하고 사전에 대응하기 위해서 Packet 분석은 필수적이다. 시중에 출시된 Packet Analyzer는 Packet을 Capture하여 분석하는 공통 기능은 제공하지만, TCP Dump의 경우 원하는 Packet을 식별하는데 어려움이 있고, Wireshark의 경우는 사용 방법의 복잡함이, Smart Sniff는 프로그램이 불안정하다는 문제점이 있다. 그래서, 사용 방법이 간단하고, Packet Header 도식도(圖式圖)를 통해 원하는 Packet을 식별하는데 편리한 Packet Analyzer VPS를 구현하였다

.

**제 2 장 관련 연구**

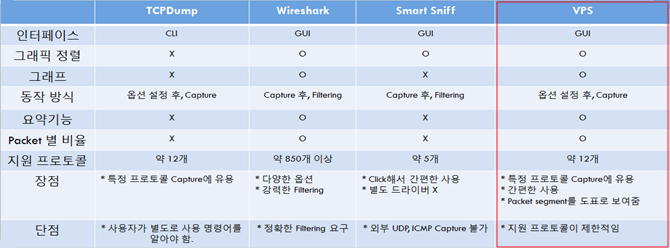
**제 1 절 Packet Analyzer**

**Ⅰ. 개요**

Packet Analyzer는 네트워크 Interface를 통해 이동하는 Packet들을 Packet Capture API를 통해 잡아내어 내부 내용을 분석하는 프로그램이다. <표 2 – 1>은 시중에 출시된 Packet Analyzer TCP Dump, Wireshark, Smart Sniff의 성능 및 장〮단점을 분석한 표이다.

**Ⅱ. 비교 분석**

<표 2 - 1> VPS(Visual Packet Sniffer) 기능 분석



**Ⅲ. Analyzer 조사**

**1. TCP Dump**

**∙ 동작 Interface**

Commend line Interface (CLI)

**∙ 동작 방식**

Option으로 원하는 Packet을 설정 후, Capture를 수행

**∙ 장점**

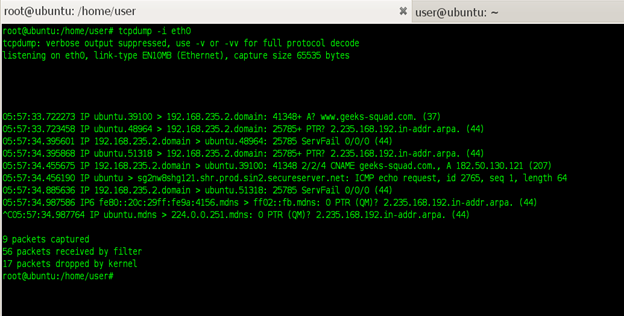
프로그램이 가벼움

특정 프로토콜만 Capture가 가능하여 시스템에 부하를 주지 않음

**∙ 문제점**

사용하기 위해 사용자가 별도로 명령어를 알아야 함

CLI 기반의 화면에서 분석할 Packet을 식별하기 어려움



<그림 2 - 1> TCP Dump 화면

**2. Wireshark**

**∙ 동작 Interface**

Graphic User Interface (GUI)

**∙ 동작 방식**

Network Interface로 이동하는 모든 Packet를 Capture함

**∙ 장점**

존재하는 Packet Analyzer중 가장 안정적임

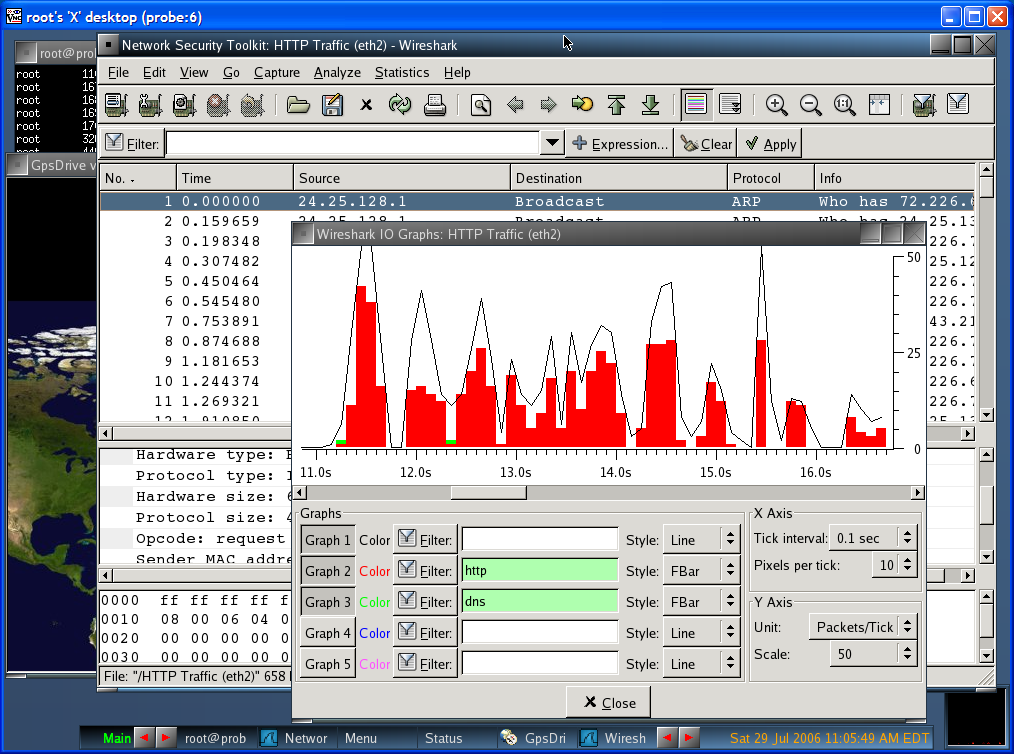
필요한 Packet을 Filtering하여 확인 할 수 있음

Graph로 기간별 통계치 확인이 가능

**∙ 문제점**

모든 Packet을 Capture하고 분석하기 때문에 시스템에 많은 부하를 발생시킴

일반 사용자가 사용하기에 사용 방법이 어려움

****

<그림 2 - 2> Wireshark 화면

**3. Smart Sniff**[3]

**∙ 동작 Interface**

Graphic User Interface (GUI)

**∙ 동작 방식**

Option으로 원하는 Packet을 설정 후, Capture를 수행

**∙ 장점**

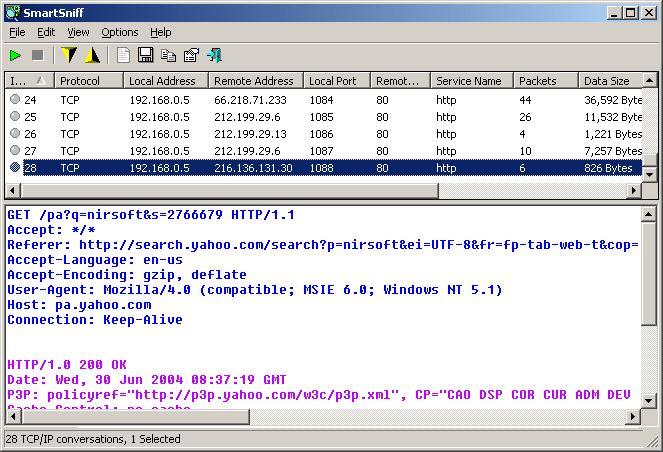
별도의 드라이버 설치 없이 네트워크 감지를 할 수 있음

특정 프로토콜만 Capture가 가능하여 시스템에 부하를 주지 않음 **∙ 문제점**

Capture할 수 있는 프로토콜 수가 적음

UDP 또는 ICMP Packet을 Capture하지 못함

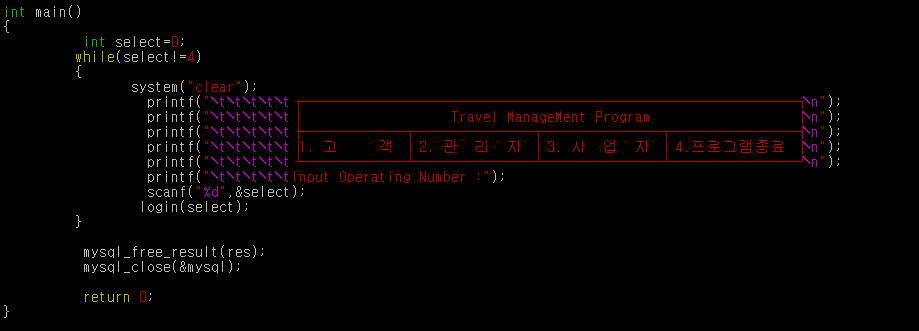
불안정한 Software 성능으로 간혹 Blue-screen 발생

****

<그림 2 - 3> Smart Sniff 화면

이처럼 TCP Dump, Wireshark, Smart Sniff는 각자 특유의 장점이 있다. 하지만, TCP Dump의 경우는 원하는 Packet을 식별하는데 불편함이, Wireshark는 사용 방법이 복잡하고 장기간 사용시 시스템에 부하를, Smart Sniff의 경우 UDP, ICMP Packet을 잡지 못한다는 단점을 갖고 있다. 따라서, 이 3가지 Packet Analyzer의 장점들을 취합하고, 단점은 보완한 Packet Analyzer VPS를 구현한다.

**제 2 절 Travel Agency Program**

****

<그림 2 - 4> Travel Agency Program 메인 화면

**Ⅰ. 개요**

Travel Agency Program은 C언어 기반의 mysql 라이브러리 내장함수를 이용하여 mysql서버 데이터베이스에 접근 할 수 있도록 사용자의 편의성을 제공하는 프로그램이다. 프로그램의 큰 틀은 3개의 View(고객,관리자,사업자)를 제공한다. 각각의 사용자는 log.h소스파일의 인증절차 후, 접근권한을 인증 받는다. 인증 후 각각의 사용자에 따라 필요기능을 수행한다.

**Ⅱ. Travel Agency Program 조사**

**∙ 동작 Interface**

Commend line Interface (CLI)

**∙ 동작 방식**

각각의 사용자의 따라 요구사항을 처리한다

**∙ 장점**

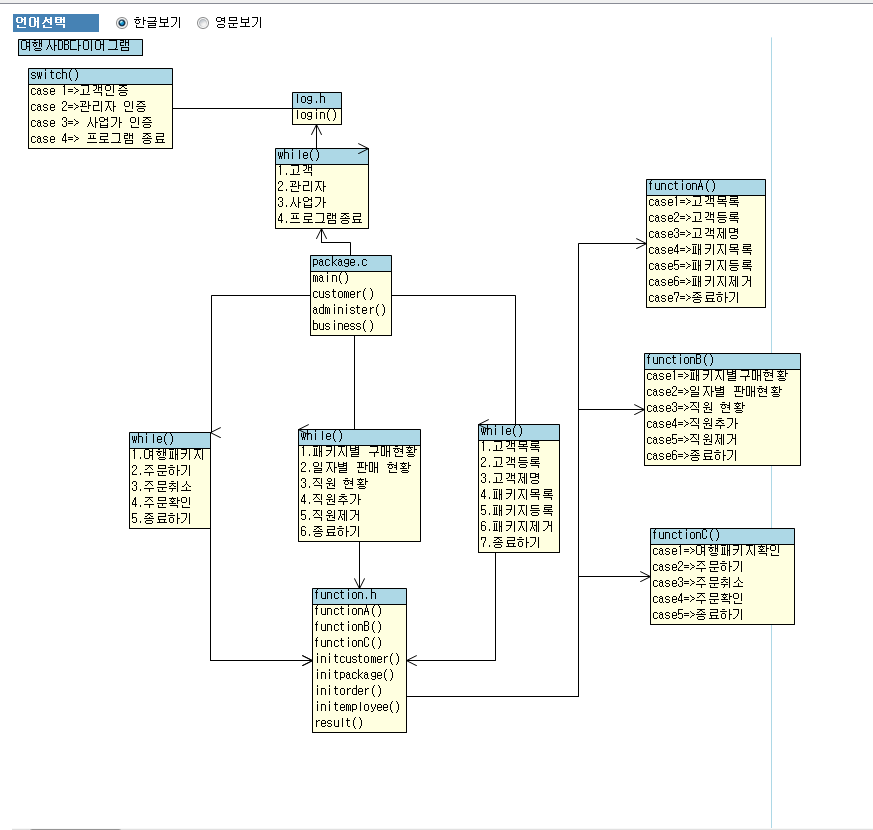
특정 사용자에게만 유용한 기능을 제공할 수 있다(보안성)

프로그램의 크기가 가볍다

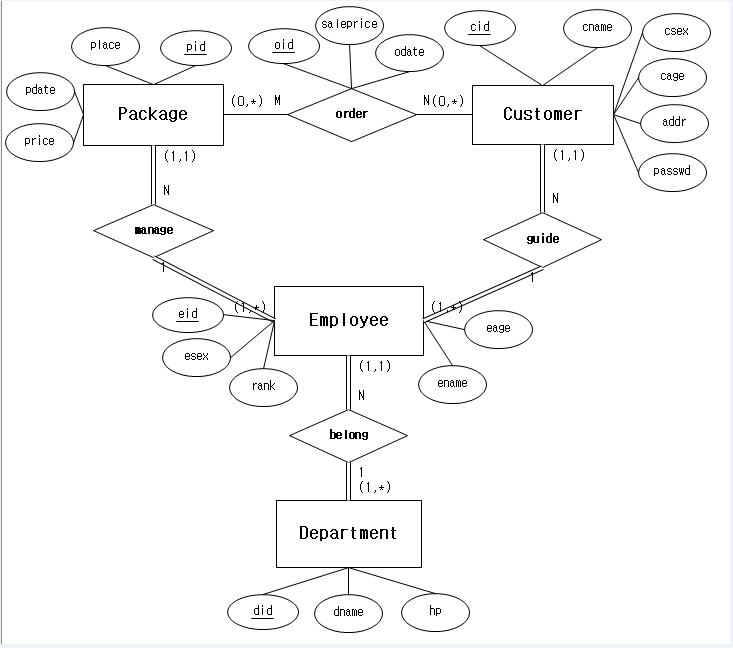
**∙ 단점**

Mysql서버 관리자의 로그인 설정이 필요하다

웹과의 연동이 미흡하다

****

<그림 2 - 5> Travel Agency Program 다이어그램

****

<그림 2 - 6> Travel Agency Program 개념적 모델링

**제 3 장 모 델**

**제 1 절 VPS (Visual Packet Sniffer)**

**Ⅰ. 개요**

VPS(Visual Packet Sniffer)는 Java JPcap Library[4]를 이용하여 옵션으로 선택한 Packet만을 Capture하여 분석한 내용을 ‘Packet Header 도식도(圖式圖)로 나타내는 Packet Analyzer이다.

**Ⅱ. VPS 사양**

**∙ 동작 Interface**

Graphic User Interface (GUI)

**∙ 동작 방식**

Option으로 원하는 Packet을 설정 후, Capture를 수행

**∙ 장점**

프로그램이 가볍고, 사용방법이 간단함

특정 프로토콜만 Capture가 가능하여 시스템에 부하를 주지 않음

Packet Header 도식도를 통해 분석할 Packet을 식별하는데 용이

**∙ 지원 프로토콜**

**Layer 3** (ARP, ICMP, IP) **Layer 4** (TCP, UDP) **Layer 7** (FTP, TELNET,SMTP,HTTP)

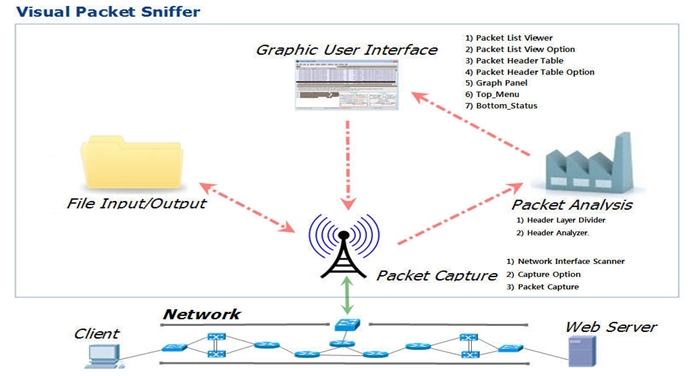


<그림 3 – 1> VPS Prototype Model

**제 2 절 VPS 동작 환경**

<그림 3 – 1>은 VPS가 동작하기 위한 환경을 나타내는 그림이다. 크게 Packet을 분석하여 화면으로 제공하는 VPS 내부 환경과 Packet 생성과 전송을 담당하는 VPS 외부 환경으로 나눌 수 있다.

본 논문에서 가장 핵심이 되는 VPS 내부 환경은 Packet Capture, File 입∙출력, Packet 분석, GUI 기능을 담당하는 4가지 모듈로 구성된다.



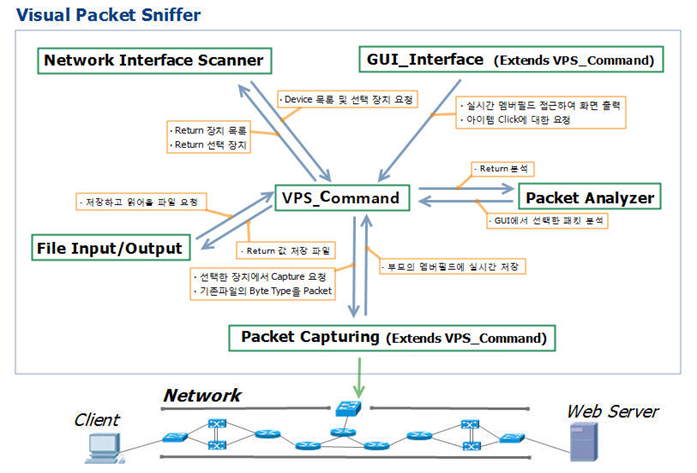
<그림 3 – 2> VPS 동작 환경

**제 4 장 설 계**

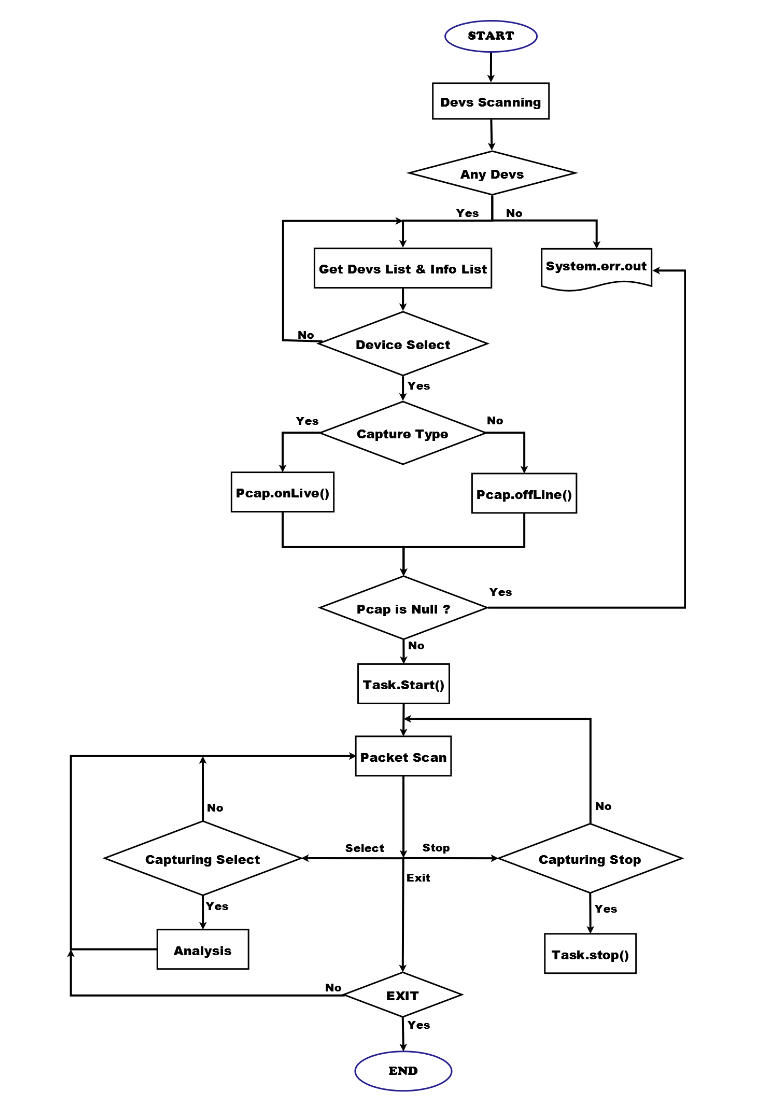
**제 1 절 VPS 설계**

**Ⅰ. VPS 데이터 흐름도(DFD)**

<그림 4 -1>은 VPS 내부 환경에서 주요 기능을 담당하는 모듈들의 데이터 흐름도를 나타낸 것이다. 중앙에서 VPS\_Command 모듈이 다른 모듈들의 신호 요청(action)을 받고 필요한 데이터(Return)값을 반환하는 흐름을 보인다.

<그림 4 – 1> VPS 데이터 흐름도

**Ⅱ. VPS 순서도**



<그림 4 – 2> VPS 순서도

**Ⅲ. VPS 화면 설계**

화면 설계는 총 3개의 패널로 구성되며, 각각의 패널은 여러가지 컴포넌트들을 포함한다.

**1) Panel 1**

총 10개의 프로토콜을 담당하며, 3개의 그래프(원형,막대,꺾은선)를 담당한다.

1. FTP : FTP 패킷의 그래프를 나타낸다.
2. Telnet : TELNET 패킷의 그래프를 나타낸다.
3. SMTP : SMTP 패킷의 그래프를 나타낸다.
4. HTTP : HTTP 패킷의 그래프를 나타낸다.
5. TCP : TCP 패킷의 그래프를 나타낸다.
6. UDP : UDP 패킷의 그래프를 나타낸다.
7. ICMP : ICMP 패킷의 그래프를 나타낸다.
8. IP : IP 패킷의 그래프를 나타낸다.
9. IGMP : IGMP 패킷의 그래프를 나타낸다.
10. ARP : ARP 패킷의 그래프를 나타낸다.

**2) Panel 2**

실질적인 패킷의 캡쳐 및 파일입출력의 연결을 처리한다.

1. Start : 패킷을 캡쳐후 Packet\_info.txt,Packet\_info\_backup.txt파일에 저장 및 각각의 프로토콜 파일에 해당되는 패킷을 저장후 TextArea에 출력한다. 또한, 현재 사용중인 네트워크 장비 및 네트워크 정보(IP,Subnet Mask 등)을 출력한다.
2. Select : 캡쳐된 패킷중 하나를 선택하여 Dialog로의 전달 및 각각의 프로토콜에 해당되는 패킷에 따라 형태를 달리하여 사용자에게 보여준다. 또한, TCP 패킷에 해당되는 패킷인경우 테이블에 나타낸다.
3. Sort : 캡쳐된 각각의 프로토콜에 해당되는 패킷을 불러들여와 TextArea에 나타내어 보여준다.
4. Clear : 현재까지 진행되어 보여진 패킷정보의 초기화 및 전체적인 그래프를 나타내어 보여준다.
5. Apply : 캡쳐된 패킷을 정렬하여 나타낸다.

**3) Panel 3**

4개의 패킷(TCP,UDP,IP,ARP)의 프로토콜 Prototype을 보여준다.

1. TCP Header Proto : TCP 헤더의 이미지
2. UDP Header Proto : UDP 헤더의 이미지
3. IP Header Proto : IP 헤더의 이미지
4. ARP Header Proto : ARP 헤더의 이미지

**제 2 절 Infrastructure 설계**

**Ⅰ. 설계 환경**

인프라 설계는 서비스 형태에 따라 크게 Client Local Network, WAN, Service Local Network 부분으로 나눠서 설계하였다.

**1) Client Local Network**

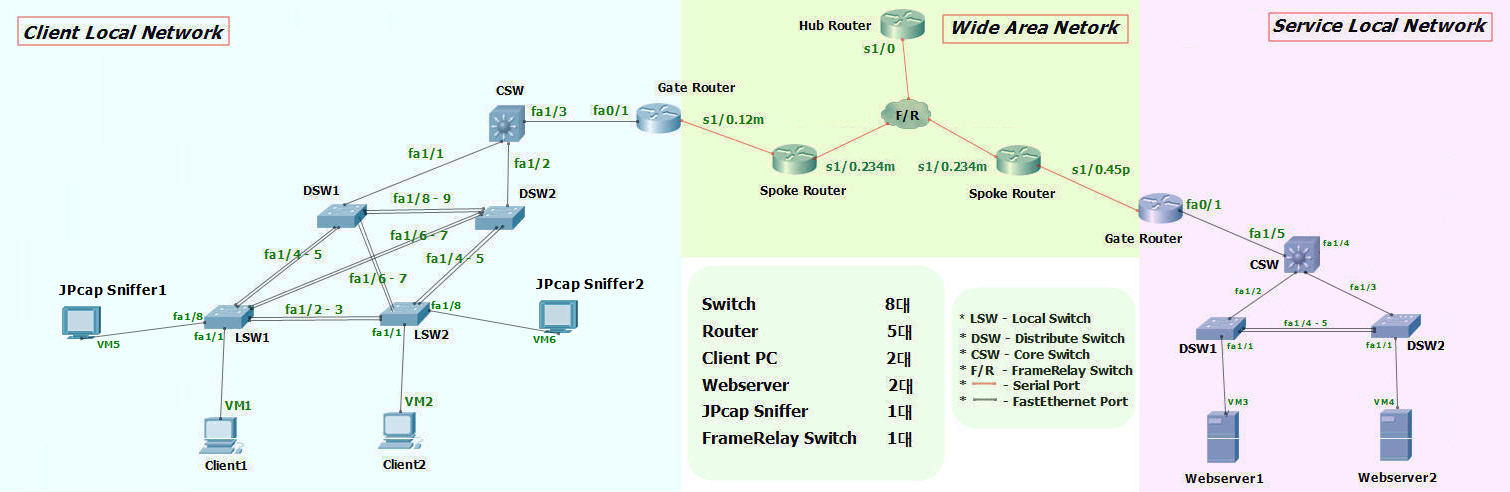
Web 서비스를 이용하는 Client와 Packet Capture하는 VPS를 Network로 연결한 Local Area Network이다.

**2) Wide Area Network**

멀리 떨어져 있는 Client Local Network와 Service Local Network를 이어주는 Network이다.

**3) Service Local Network**

Web 서비스를 하는 서버가 존재하는 Local Area Network이다.

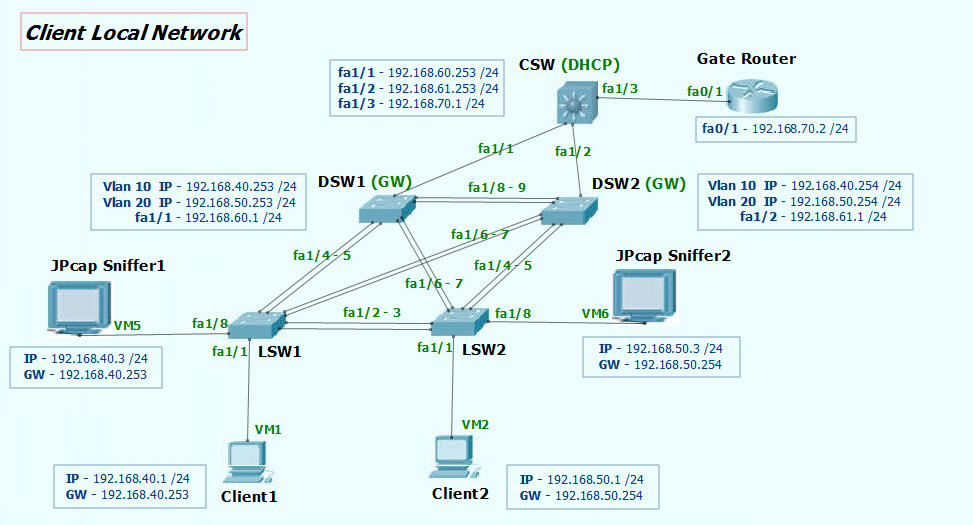


<그림 4 – 3> Infrastructure 설계

**Ⅱ. 설계 구조**

**1) Client Local Network**

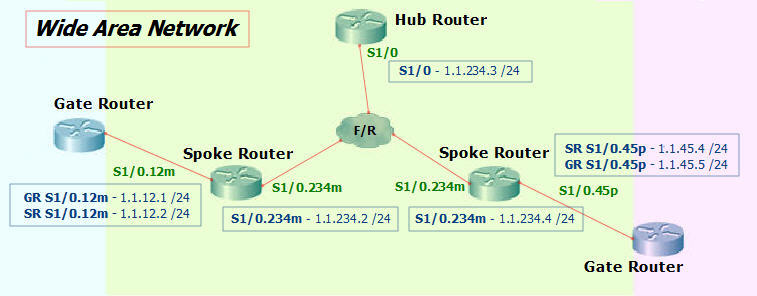
Web 서비스를 이용하는 Client와 Packet Capture하는 VPS를 Network로 연결한 Local Area Network이다.



<그림 4 – 4> Client Local Infrastructure 설계

**2) Wide Area Network**

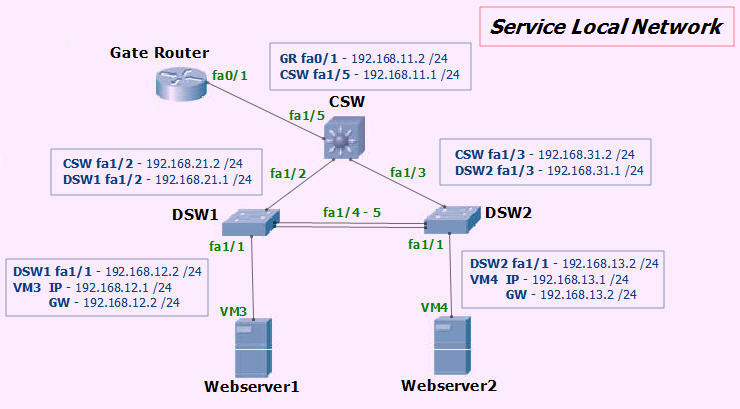
멀리 떨어져 있는 Client Local Network와 Service Local Network를 이어주는 Network이다.



<그림 4 – 5> Wide Area Infrastructure 설계

**3) Service Local Network**

Web 서비스를 하는 서버가 존재하는 Local Area Network이다.



<그림 4 – 6> Service Local Infrastructure 설계

**제 5 장 구 현**

**제 1 절 PATP 구현 환경**

**Ⅰ. Java Eclipse**[5]

sun microsystems 사가 통신망을 통해 수행할 수 있는 코드를 만들 수 있도록 개발한 프로그래밍 언어로 즉, 자바를 사용하여 실행할 수 있는 내용을 만들 수 있는 SW도구의 모음을 말한다. 자바는 멀티 플랫폼으로 유닉스나 PC의 윈도OS, 매킨토시 등 어떠한 환경에서도 동작할 수 있고, 기존 OS용 응용프로그램들이 인터넷을 통해 자바용으로 재사용될 경우에는 기존 OS 패러다임과는 매우 다른 독자적인 OS가 된다. 자바는 사용상의 편의성과 이식성이 높고, 스마트카드에서 슈퍼컴퓨터까지의 기종에 관계없이 폭넓게 활용할 수 있어 특정업체의 독점 기술이 아닌 만인의 기술로 인정받고 있으나 속도가 다소 느리다는 단점이 있다.

**Ⅱ. JPcap Library**[6]

JPcap은 오픈 소스 Java 라이브러리이다. 모든 libpcap을 자연적으로 call하는 Java Wrapper Class이다. 실시간으로 capture된 packet을 decoding하며, 최적의 packet decoding 성능을 내기 위해 native code와 Java를 혼합하여 사용한다. 사용자들은 java SDK를 이용하여 자신의 프로토콜 정의를 추가 할 수 있다.

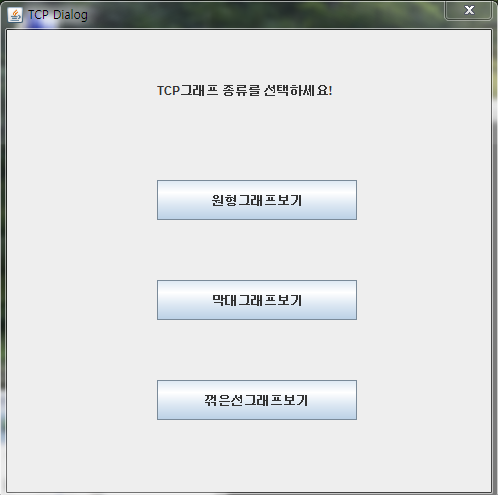
**Ⅲ. VM Ware Player->Fedora Linux(Mysql서버)**[7]

Vm ware Player은 소프트웨어로 만들어진 가상화 프로그램이다. 실제 사용중인 컴퓨터 운영체제에서 가상화 운영체제의 설치 및 네트워크 설정등 여러가지 기능을 제공한다. VM Ware Player를 이용하여 Fedora Linux를 설치후, Mysql서버 구축 및 Travel Agency Program의 컴파일 및 실행을 한 후 데이터를 보내는 과정에서의 이루어지는 패킷의 전달을 VPS로 캡쳐하기 위함이다.

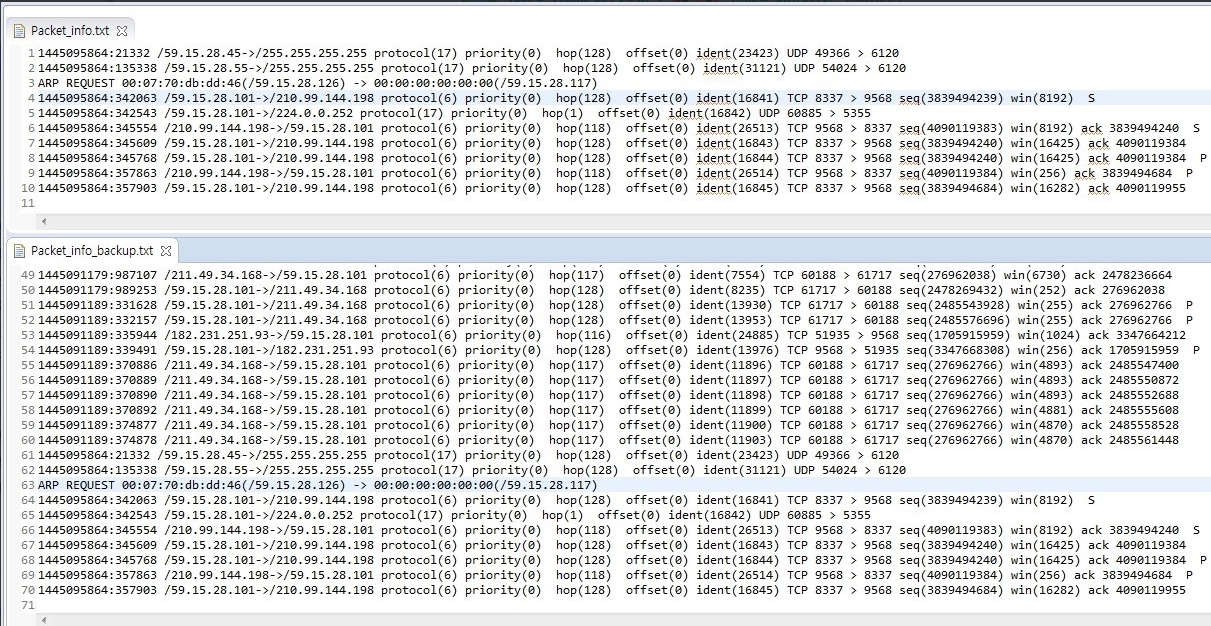
**제 2 절 PATP 구현 화면**



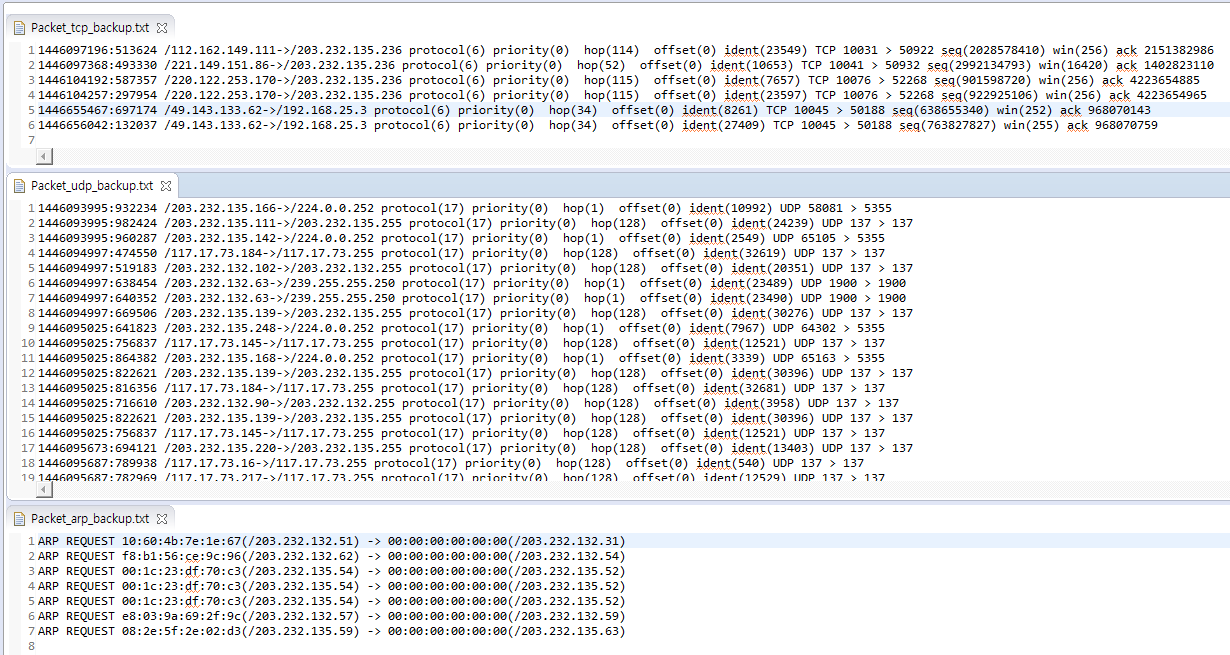
<그림 5 - 1> VPS 실행 화면



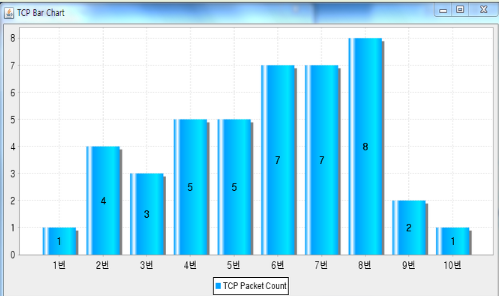
<그림 5 - 2> TCP Graph 구성도<그림 5 - 3> Select Dialog



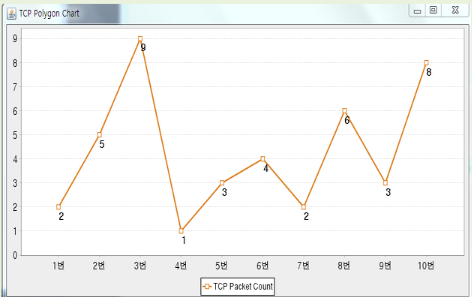
<그림 5 - 4> 전체 패킷 저장 및 백업내용



<그림 5 - 5> 프로토콜 패킷 백업내용



<그림 5 - 6> TCP Bar Graph



<그림 5 - 7> TCP Polygon Graph

**제 6 장 결론 및 향후 계획**

VPS(Visual Packet Sniffer)의 구현목표는 다음과 같다.

1. 사용자 편의성에 기반한 인터페이스

2. 프로그램 사용 간편성

현재까지 구현된 VPS는 1번의 사항은 구현에 성공하였다. 2번의 경우는 많은 부분이 미흡하다. 각각의 프로토콜 별로 구분하여 보기 쉬운 구성으로 프로그램을 만들었으나, 실제적으로 그 안의 내용물인 소스는 상당히 복잡하다.

원래의 구성 목표인 시간을 이용한 Thread 프로그램을 만들고자 하였지만, 구현하지 못하였다.

또한, 현재의 VPS는 패킷의 백업정보경로를 일일이 소스에서 지정해 주어야 하는 번거로움이 있으며, 해결하지 못하였다.이뿐 아니라, 구현되지 못한 소스부분도 많이 남아있다.

**부록 Ⅰ**

**VPS 주요코드**

**1) Window\_GUI.java**

프로그램의 전체적인 GUI의 구성 및 파일입출력을 연결한다.

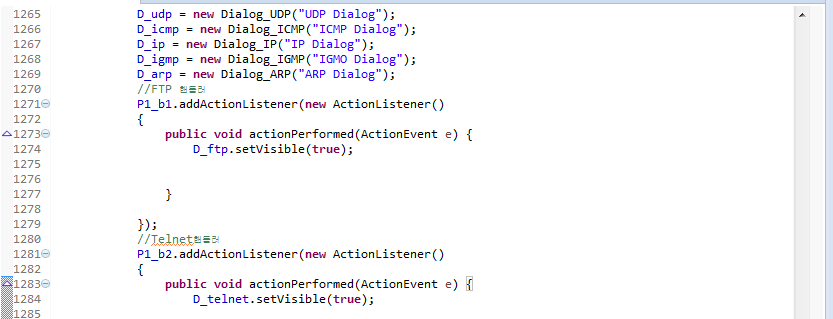
* Dialog\_TCP 클래스

TCP 그래프 다이얼로그로써, 그래프의 종류를 정한다.



* panel1 클래스

프로토콜에 맞는 다이얼로그를 보인다.

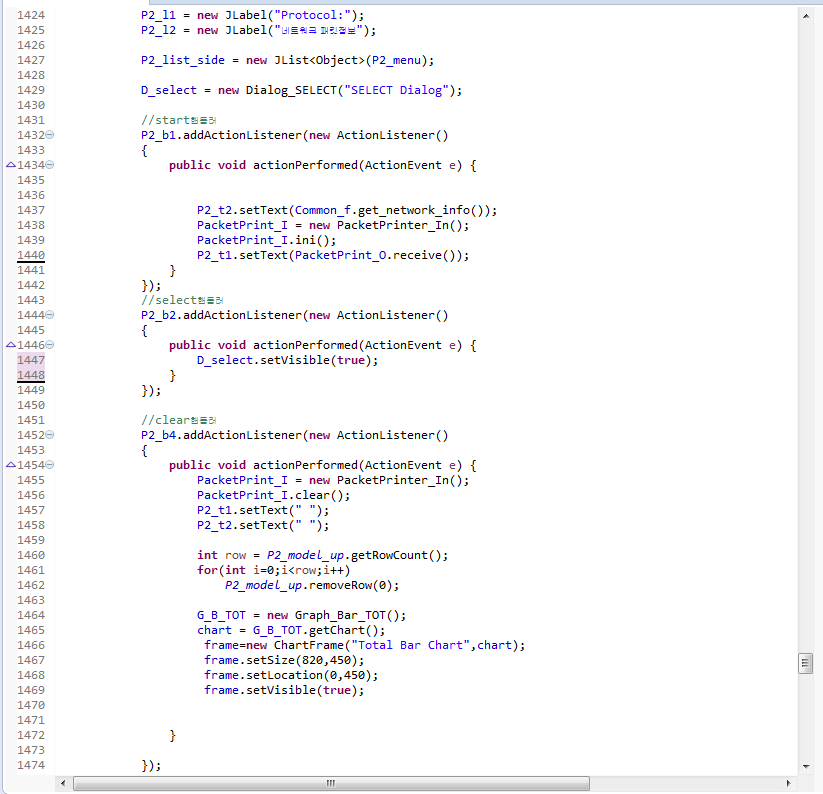


* panel2 클래스

start 버튼🡺 네트워크 장비의 정보를 받은후 파일에 저장한다. 또한, textArea에 내용을 보인다.

select 버튼🡺 select Dialog를 띄운후, 사용자에게 패킷을 선택하도록 요구한다.

clear 버튼🡺 현재까지 보여지는 내용을 모두 지운후, backup파일에 저장한다.



* panel3 클래스

프로토콜 TCP,UDP,IP,ARP패킷의 이미지를 보여준다.



**2) Write\_TCP.java**

TCP 패킷의 정보를 저장한다.

* Store() 메소드

TCP 패킷의 정보를 bufferedWriter메소드를 이용하여 저장 후 종료한다.

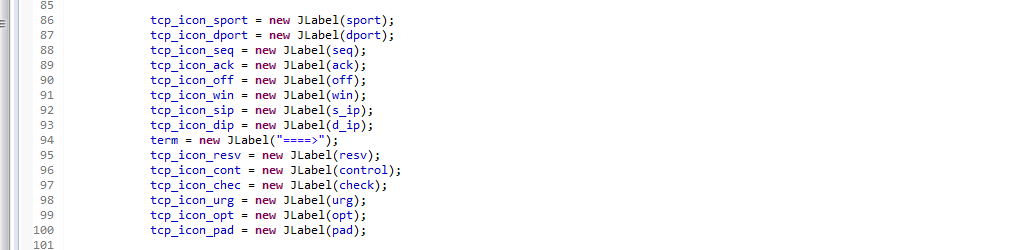
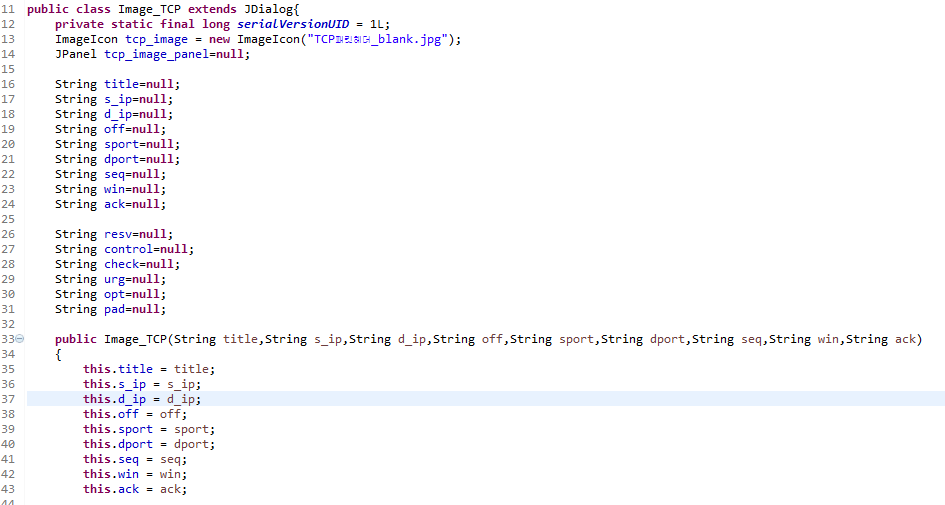


**3) Image\_TCP Class**

TCP 패킷의 정보를 패킷의 형태로 도식화한다.

* Image\_TCP() 생성자

TCP 패킷의 정보를 분석하여 Label에 저장 후 나타낸다.

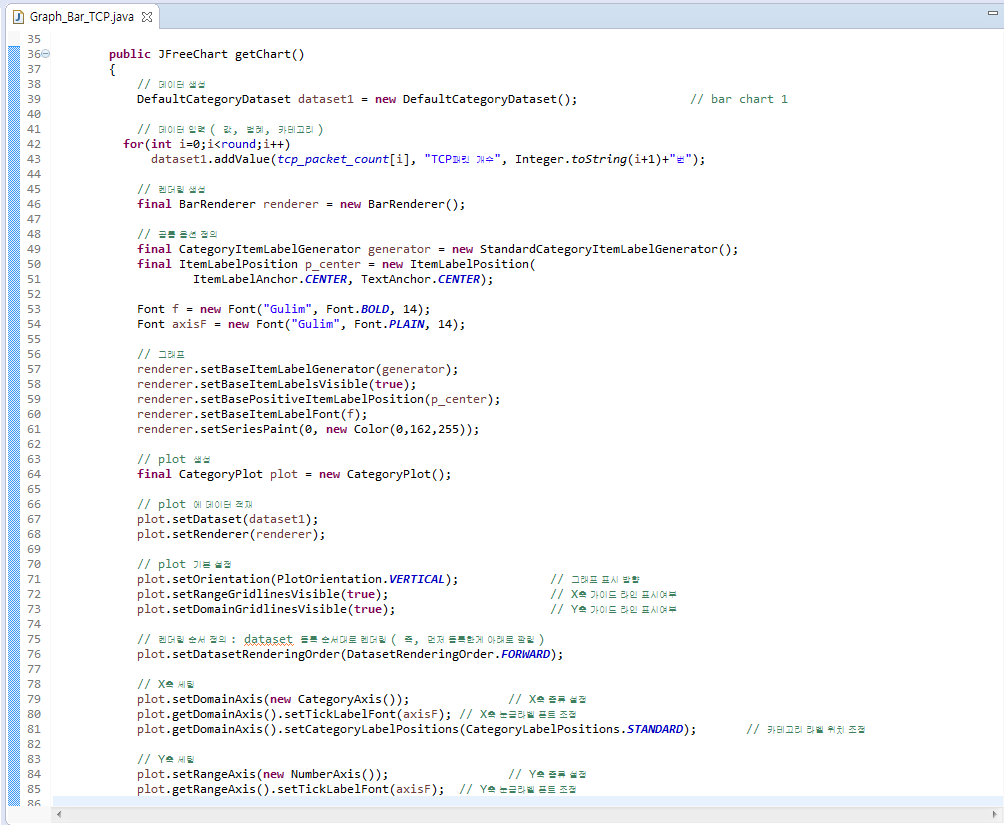


**4) Graph\_Bar\_TCP.java**

TCP 패킷의 정보를 막대그래프로 도식화한다.

* getChart() 메소드

TCP 막대그래프의 형태를 그린 후, 실행도중 입력된 값을 그래프로 보여준다.

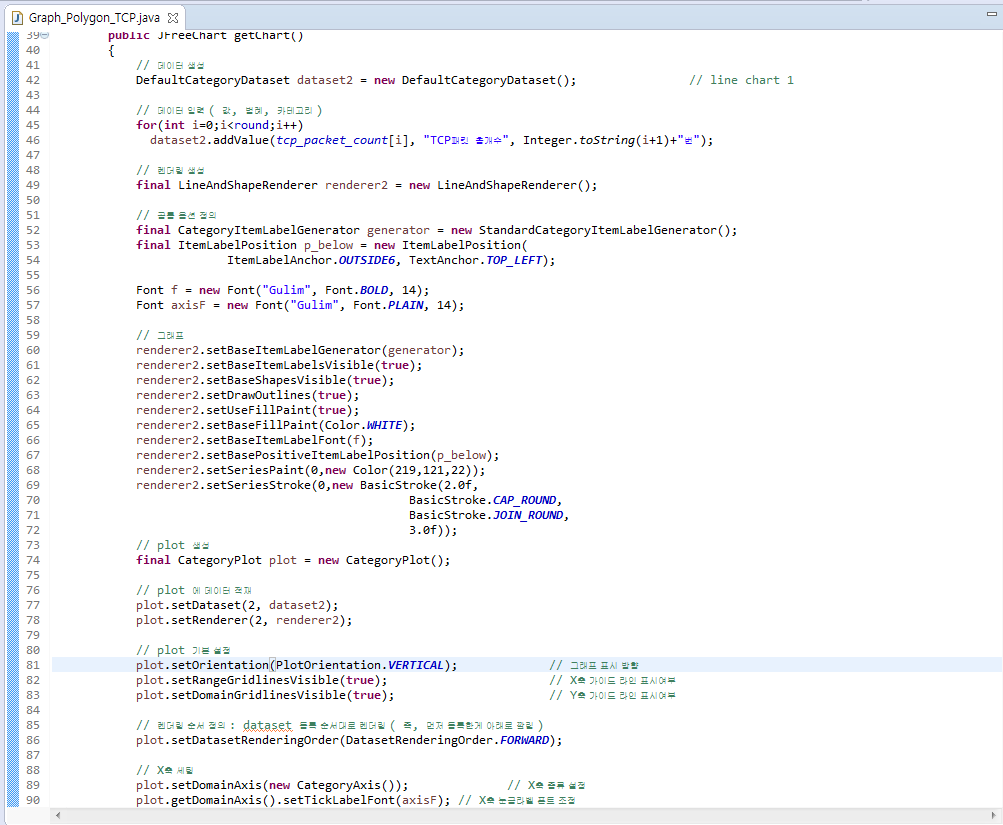


**5) Graph\_Polygon\_TCP Class**

TCP 패킷의 정보를 꺾은선그래프로 도식화한다.

* getChart() 메소드

TCP 꺾은선그래프의 형태를 그린 후, 실행도중 입력된 값을 그래프로 보여준다.



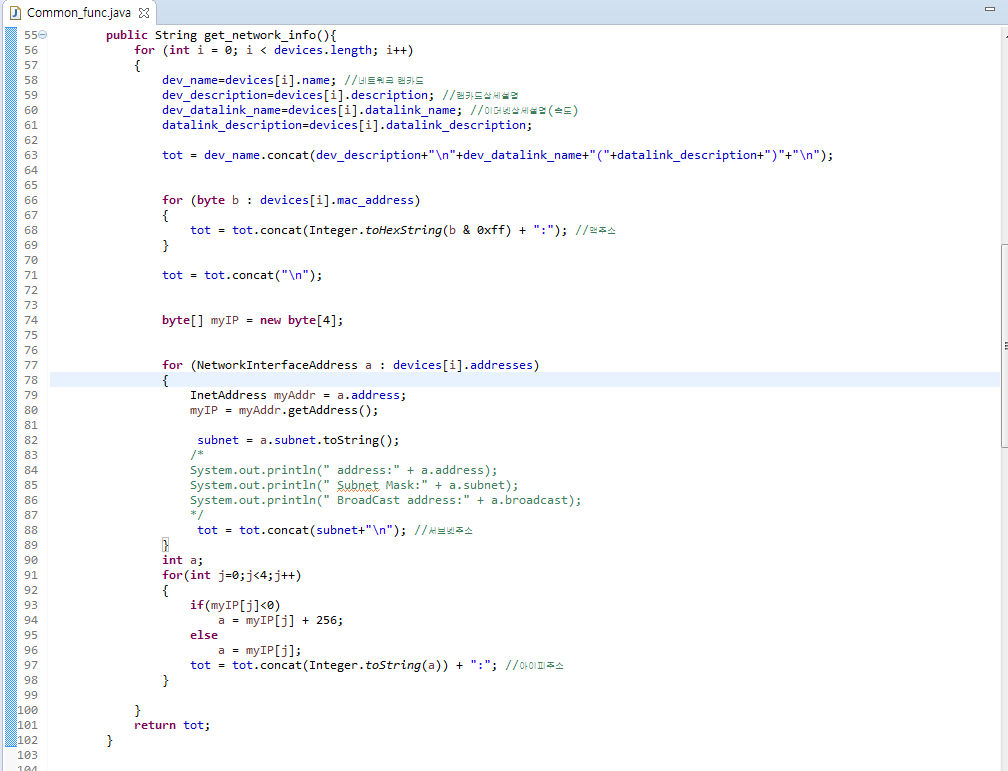
**6) Common\_func Class**

현재 통신하는 네트워크 장비의 정보(IP Address, Subnet Mask, BroadCast Address)를 얻어오며,

각각의 프로토콜에 해당되는 패킷을 분류하여 전달한다.

* get\_network\_info() 메소드

현재 네트워크 장비의 이름, 데이터링크 이름, 데이터링크 상세설명,IP주소 등의 정보를 나타낸다.



* check() 메소드

패킷의 종류를 프로토콜에 따라 비교한 후, 파일저장 및 분석하기 위하여 넘긴다.

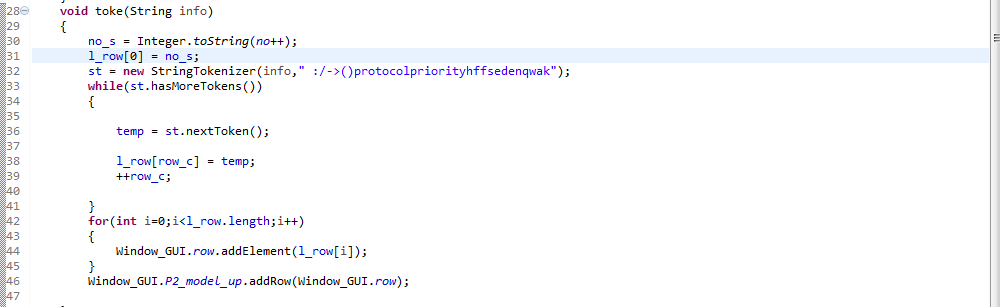


**7) L3\_TCP Class**

TCP 패킷을 Tokenizer를 이용하여 분리 후, 패킷의 도식화를 위하여 전달한다.

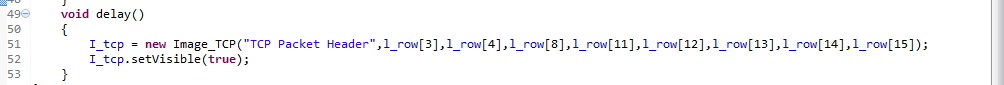
* toke() 메소드

Tokenizer메소드를 이용하여 내용을 분리 후, 배열에 저장한다. 그리고 table의 행으로 추가한다.



* delay() 메소드

분석된 정보들을 패킷의 형태로 나타내주는 Image\_TCP 클래스의 생성자로 넘긴다.



**8) PacketPrinter\_In Class**

현재 사용중인 네트워크 장치를 검색한 후 패킷을 검출한다. 검출 한 패킷은 각각의 프로토콜에 해당 될 시 개수를 증가하여 기록한다. 그 후, 전체적인 패킷의 저장 및 백업파일에 저장한다.

* ini() 메소드

현재 사용중인 네트워크 장비를 얻어온다. 그 후, 패킷을 받아온다.



* receivePacket() 메소드

JPcap의 오버라이딩 메소드로써, 패킷을 받아온 후 행해지는 기능을 서술하여야 한다. 받아온 패킷의 갯수만큼 전체 패킷의 수를 증가시킨다. 그 후, 길이를 4byte만큼 당긴 후 프로토콜의 분류를 위하여 check() 메소드로 넘긴다.



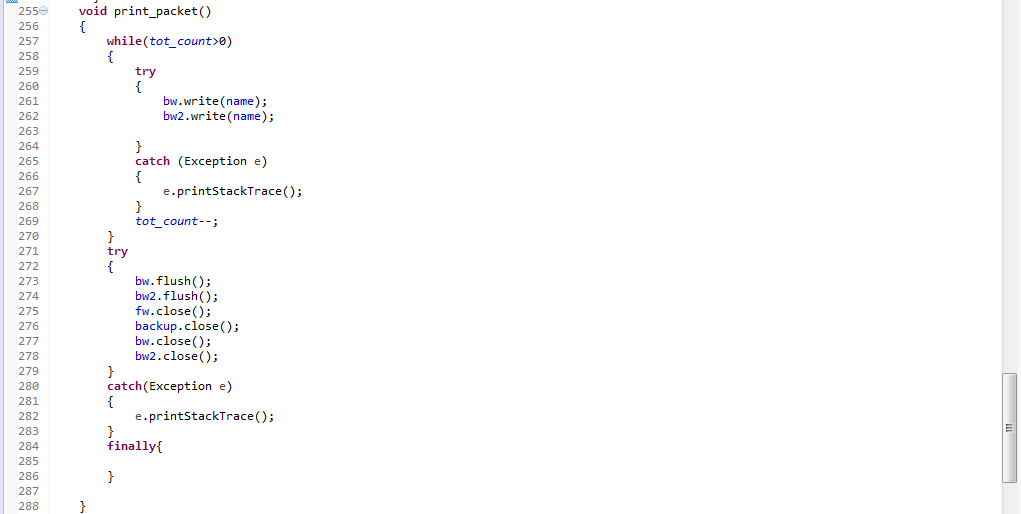
* check() 메소드

프로토콜마다 지정된 패킷에 따라 개수를 증가시킨다.



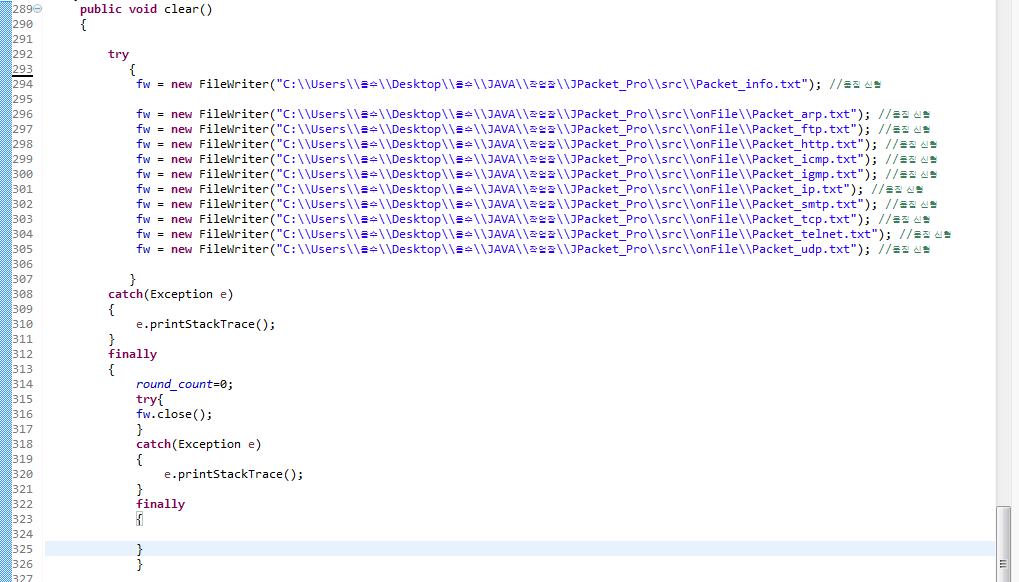
* print\_packet() 메소드

패킷의 정보를 Packet\_Info.txt파일에 저장한다.



* clear() 메소드

각각의 프로토콜 파일을 연다. 이때, 인자 값이 false이므로, 파일은 새로 만들어진다. 그러므로 지워지는 효과를 나타낸다.

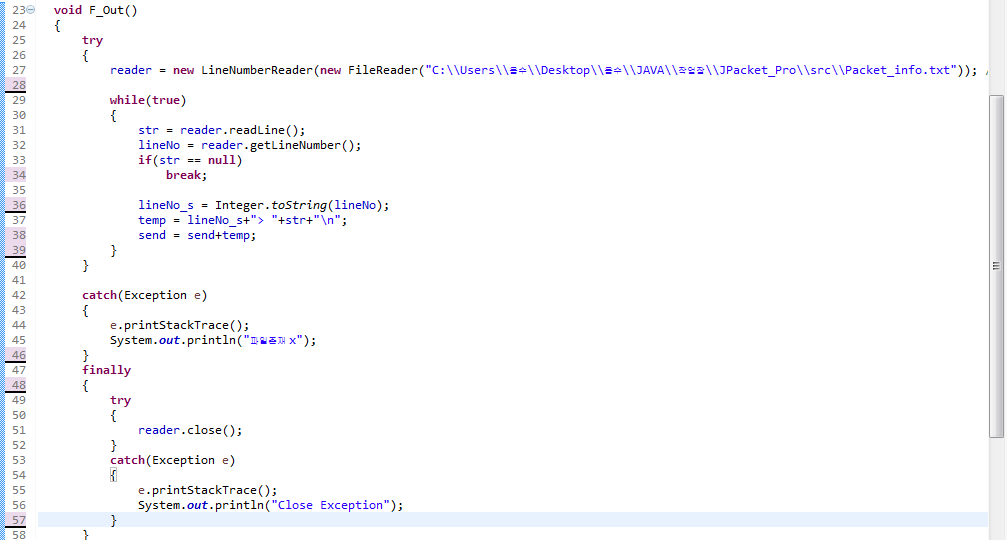


**9) PacketPrinter\_Out Class**

현재까지 파일에 저장된 패킷을 불러들인 후, TextPane에 출력한다.

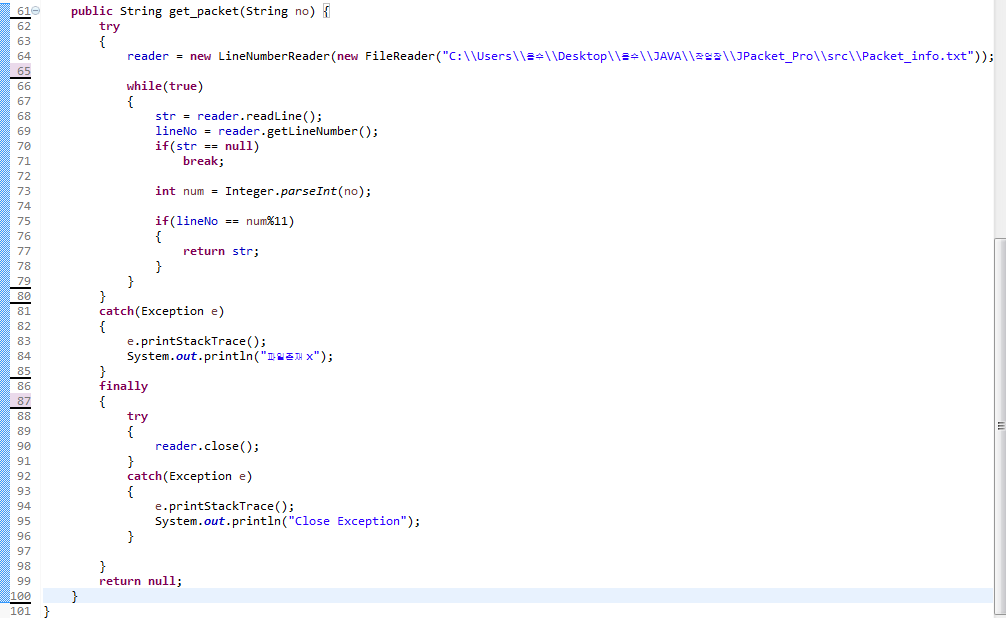
* F\_Out() 메소드

현재까지 저장된 패킷의 정보를 읽어 들인 후, “번호>”의 형식을 덧붙인다.



* get\_packet() 메소드

현재까지 파일에 저장된 패킷을 11개 단위로 끊어서 읽어 들인다.



**10) Reference Files**

환경설정 및 링크에 적용되는 프로그램 및 파일들

1. **WinPcap\_4\_1\_3.exe**[8]
2. **Jpcap.dll**[9]
3. **jpcap.tar**[10]
4. **jfreechart-1.0.19.zip**[11]
5. **jcommon-1.0.23.jar**[12]

**참고문헌**

[1] 국가별 공격 발생지 분포, http://blog.ahnlab.com/ahnlab/tag/1488

[2] 공격에 이용된 포트 별 분포, http://blog.ahnlab.com/ahnlab/tag/796

[3] Smart sniff 사용 방법, http://myfact.tistory.com/archive/200802

[4] JPcap Library API, http://jpcap.sourceforge.net/javadoc/index.html

[5] Java Eclipse, http://www.eclipse.org/

[6] JPcap Library 동영상, https://www.youtube.com/watch?v=Uump9bPIER8

[7] Vmware 설명 사이트, https://ko.wikipedia.org/wiki/VM%EC%9B%A8%EC%96%B4

[8] WinPcap 설치 사이트, http://www.winpcap.org/

[9] JPcap rundll파일 사이트, https://groups.google.com/forum/#!topic/jpcap/-vxZv0eAcp4

[10] JPcap tar파일 사이트, http://jpcap.software.informer.com/0.7/

[11] JFreeChart 라이브러리 사이트, http://sourceforge.net/projects/jfreechart/?source=typ\_redirect

[12] JFreeChart 환경설정 파일, http://mvnrepository.com/artifact/org.jfree/jcommon/1.0.23

[13] Access Restriction Error 처리, http://nota.tistory.com/7

[14] UnsatisfiedLink Error 처리, http://ohgyun.com/253

[15] “명품 Java Programming”, 황기태 김효수 저자, 2013.02.07, 생능 출판사

[16] “뇌를 자극하는 Java 프로그래밍”, 김윤명 저자, 2006.08.31, 한빛미디어

[17] JPcap Library 다운로드 설명 사이트, http://egloos.zum.com/jhpsb/v/233471